Міністерство освіти і науки України

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Л. М. Олещенко

Основи комп’ютерних систем і мереж

Підручник

Затверджено Вченою радою КПІ ім. Ігоря Сікорського
як підручник для для здобувачів ступеня бакалавра, які навчаються
за спеціальністю F2 «Інженерія програмного забезпечення»

Електронне мережеве навчальне видання

Київ
«ПТФ ПРОСВІТА»
2026

УДК 004.7(075.8)
О53

Затверджено Вченою радою КПІ ім. Ігоря Сікорського
як підручник для студентів, які навчаються
за спеціальністю F2 «Інженерія програмного забезпечення»
(протокол № 12 від 15.12.2025 р.)

Відповідальний редактор — ***Легеза Віктор Петрович***, д-р техн. наук, проф., професор кафедри програмного забезпечення комп'ютерних систем КПІ ім. Ігоря Сікорського

Рецензенти — ***Кисельов Володимир Борисович***, д-р техн. наук, проф., заслужений працівник освіти України, директор навчально-наукового інституту муніципального управління та міського господарства Таврійського національного університету імені В.І. Вернадського

Соколовський Ярослав Іванович, д-р техн. наук, проф., Відмінник освіти України, професор кафедри систем автоматизованого проектування Національного університету «Львівська політехніка»

Головний редактор — ***Сирота Сергій Вікторович***, канд. техн. наук, доц., доцент кафедри прикладної математики КПІ ім. Ігоря Сікорського

Олещенко Л. М.

О53 Основи комп'ютерних систем і мереж [Електронний ресурс] : підруч. для здобувачів вищої освіти, які навчаються за спец. «Інженерія програмного забезпечення» / Л. М. Олещенко; КПІ ім. Ігоря Сікорського. – Електрон. текст. дані (1 файл 10,2 МБ). – Київ : ПТФ Просвіта, 2026. – 436 с.

ISBN 978-617-7010-40-0

Підручник розроблено для ознайомлення студентів з основними технологіями комп'ютерних систем та мереж, налаштування компонент комп'ютерних мереж, IP-адресації мережевих пристроїв, перевірки та налагодження мережевих комунікацій. Розглянуто основні типи комп'ютерних систем та мереж, апаратні та програмні компоненти мереж, мережеві протоколи і стандарти, а також технології комутації та маршрутизації. Підручник призначений для студентів, які навчаються за спеціальністю F2 «Інженерія програмного забезпечення», факультету програмних систем та прикладної математики КПІ ім. Ігоря Сікорського.

УДК 004.7(075.8)

ISBN 978-617-7010-40-0

© Л.М. Олещенко, 2026
© КПІ ім. Ігоря Сікорського, 2026

ЗМІСТ

ПЕРЕДМОВА……...…9

РОЗДІЛ І. ТИПИ КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ. КОМПОНЕНТИ МЕРЕЖІ. МЕРЕЖЕВІ ПРОТОКОЛИ І СТАНДАРТИ.....11

Тема 1. Типи комп'ютерних систем і мереж. Компоненти мережі. Мережеві протоколи і стандарти. Моделі OSI і TCP / IP..... 11

1.1. Типи комп'ютерних систем і мереж..... 12
1.2. Технології доступу в Інтернет..... 15
1.3. Вимоги до мережі. Компоненти мережі ……..... 22
1.4. Представлення мережі. Фізична та логічна топології мереж..... 27
1.5. Хмарні обчислення. Центри обробки даних ……..... 35
1.6. Мережеві протоколи і галузеві стандарти ……..…..... 37
1.7. Моделі OSI і TCP/IP …..... 41
1.8. Порівняння моделей OSI і TCP/IP……..……..44
Контрольні запитання та задачі до теми 1..... 45
Перелік теоретичних запитань 45
Перелік задач для самостійного опрацювання 46
Перелік використаної та рекомендованої літератури..... 47

Тема 2. Мережева операційна система..... 48

2.1. Мережеві ОС..... 48
2.2. Функції операційної системи Cisco IOS..... 52
2.3. Метод консольного доступу..……..……..…….. 53
2.4. Методи доступу за допомогою Telnet та SSH ……..……...…... 54
2.5. Програми емуляції терміналу ……..... 55
2.6. Режими роботи операційної системи Cisco IOS …… 58
2.7. Структура команд операційної системи Cisco IOS ……... 59
2.8. Основні команди налаштування і перевірки мережевих пристроїв 61
Контрольні запитання та задачі до теми 2..... 63
Перелік теоретичних запитань 63
Перелік задач для самостійного опрацювання 64
Перелік використаної та рекомендованої літератури..... 66

Тема 3. Фізичний та канальний рівень..... 67

3.1. Правила передавання даних у мережі..... 68
3.2. Логічна та фізична адреса пристроїв у мережі..... 71
3.3. Технології та засоби передачі даних фізичного рівня ……..…… 73
3.4. Характеристики мідних кабелів ……..…..... 76
3.5. Особливості використання оптоволоконних кабелів ……79
3.6. Особливості безпровідного середовища……..…….. 82

3.7. Канальний рівень……..……..…92
3.8. Поширені фізичні топології мереж…… 95
Контрольні запитання та задачі до теми 3..... 98
Перелік теоретичних запитань 98
Перелік задач для самостійного опрацювання 99
Перелік використаної та рекомендованої літератури..... 100
Тема 4. Кодування інформації в комп'ютерних мережах..... 101
4.1. Роль кодування в локальних мережах..... 101
4.2. Основи кодування інформації..... 102
4.3. Методи кодування на фізичному рівні OSI …….…..... 104
4.4. Кодування на канальному рівні OSI …….…..…115
4.5. Порівняння методів кодування інформації..... 116
Контрольні запитання та задачі до теми 4..... 118
Перелік теоретичних запитань 118
Перелік задач для самостійного опрацювання 119
Перелік використаної та рекомендованої літератури..... 120
Тема 5. Технології Ethernet..... 121
5.1. Підрівні канального рівня LLC і MAC..... 121
5.2. Призначення і структура MAC-адреси..... 123
5.3. Структура кадру Ethernet …….. 125
5.4. MAC- і IP-адреси ……..… 128
5.5. Функції і принципи роботи роботи протоколу ARP……..... 130
5.6. Таблиця MAC -адрес комутатора..... 133
5.7. Налаштування повнодуплексної та напівдуплексної передачі даних……..…134
5.8. Порівняння комутації рівня 2 і рівня 3……..136
Контрольні запитання та задачі до теми 5..... 137
Перелік теоретичних запитань 137
Перелік задач для самостійного опрацювання 137
Перелік використаної та рекомендованої літератури..... 138
Тема 6. Мережевий рівень..... 139
6.1. Процеси і протоколи мережевого рівня..... 139
6.2. Протокол IPv4..... 141
6.3. Протокол IPv6…… 144
6.4. Шлюз за замовчуванням ……..……..... 146
6.5. Таблиці маршрутизації ……..... 147
6.6. Будова і функції маршрутизатора ……150
Контрольні запитання та задачі до теми 6..... 152
Перелік теоретичних запитань 152
Перелік задач для самостійного опрацювання 153

Перелік використаної та рекомендованої літератури..... 153

Тема 7. Транспортний рівень..... 154

7.1. Призначення транспортного рівня..... 154
7.2. Протокол TCP..... 155
7.3. Протокол UDP ……..……..…… 158
7.4. Адресація портів TCP і UDP ……..……161
7.5. Сегментація TCP і UDP……..……...…165
7.6. Процеси і запити TCP –сервера……..166
7.7. Процеси і запити UDP –сервера…….169
7.8. Застосування, що використовують протоколи TCP та UDP……171
Контрольні запитання та задачі до теми 7..... 174
Перелік теоретичних запитань 174
Перелік задач для самостійного опрацювання 174
Перелік використаної та рекомендованої літератури..... 176

Тема 8. IP-адресація. Розбиття IP-мережі на підмережі..... 177

8.1. Мережева і вузлова частини IPv4 -адреси 177
8.2. Побітова операція AND..... 180
8.3. Одноадресна, широкомовна та багатоадресна передача ……181
8.4. Публічні та приватні IPv4 –адреси ……..…… 183
8.5. Обмеження застарілої класової адресації. Безкласова адресація…184
8.6. Привласнення IP –адрес ……..…… 186
8.7. Потреба в IPv6. Представлення IPv6……. 187
8.8. Розбиття мережі на підмережі ……..……191
Контрольні запитання та задачі до теми 8..... 196
Перелік теоретичних запитань 196
Перелік задач для самостійного опрацювання 197
Перелік використаної та рекомендованої літератури..... 198

Тема 9. Протоколи та сервіси прикладного рівня..... 199

9.1. Функції і протоколи прикладного рівня TCP/IP..... 199
9.2. Однорангові мережі..... 203
9.3. Протоколи HTTP і HTTPS …… 207
9.4. Протоколи SMTP, POP і IMAP ……..…… 211
9.5. Служба доменних імен……..……... 214
9.6. Протокол динамічної конфігурації вузла ……217
9.7. Протокол передачі файлів……..….. 220
9.8. Протокол обміну блоками серверних повідомлень ……..…… 222
Контрольні запитання та задачі до теми 9.....224
Перелік теоретичних запитань 224
Перелік задач для самостійного опрацювання 225
Перелік використаної та рекомендованої літератури..... 225

Тема 10. Засоби мережевої безпеки..... 226
10.1. Категорії загроз мережевої безпеки..... 226
10.2. Резервне копіювання, оновлення і установка виправлень…….... 232
10.3. Аутентифікація, авторизація і облік ……..……..… 234
10.4. Міжмережеві екрани ……..…… 236
10.5. Захист пристроїв комп'ютерної мережі……..…….. 239
10.6. Використання протоколу SSH ……244
10.7. Використання спеціальних команд……..…… 246
10.8. Безпека безпровідних підключень ……..…….. 249
Контрольні запитання та задачі до теми 10.....251
Перелік теоретичних запитань 251
Перелік задач для самостійного опрацювання 251
Перелік використаної та рекомендованої літератури..... 252
РОЗДІЛ ІІ. ТЕХНОЛОГІЇ КОМУТАЦІЇ ТА МАРШРУТИЗАЦІЇ..... 253
Тема 11. Локальні мережі. Технології комутації..... 253
11.1. Принципи роботи комутатора..... 254
11.2. Методи пересилання даних на комутаторі..... 257
11.3. Колізійні та широкомовні домени …….. 259
11.4. Методи зниження перевантажень мережі……..…….. 261
11.5 Базові налашування комутатора…….…. 263
11.6 Проблеми на рівні мережевого доступу…… 265
11.7. Поширені загрози безпеки локальних мереж……. 268
11.8. Тестування безпеки локальних мереж ……..…… 271
Контрольні запитання та задачі до теми 11..... 273
Перелік теоретичних запитань 273
Перелік задач для самостійного опрацювання 273
Перелік використаної та рекомендованої літератури.....274
Тема 12. Проєктування віртуальних локальних мереж..... 275
12.1. Призначення і переваги віртуальних локальних мереж..... 275
12.2. Типи віртуальних локальних мереж..... 278
12.3. Тегування кадрів Ethernet для ідентифікації мережі VLAN …… 280
12.4. Діапазони VLAN на комутаторах……..…..... 282
12.5. Створення віртуальної локальної мережі……... 284
12.6. Протокол DTP……..……..... 287
12.7. Проблематика VLAN……..…….. 289
12.8. Рекомендації з проєктування VLAN…….…. 292
Контрольні запитання та задачі до теми 12..... 295
Перелік теоретичних запитань 295
Перелік задач для самостійного опрацювання 295
Перелік використаної та рекомендованої літератури..... 296

Тема 13. Технології маршрутизації. Маршрутизація VLAN..... 297
13.1. Призначення маршрутизації..... 297
13.2. Основні функції маршрутизаторів..... 300
13.3. Налаштування основних параметрів маршрутизатора..... 304
13.4. Таблиця і записи маршрутизації 306
13.5. Статичні та динамічні маршрути 310
13.6. Маршрутизація VLAN 312
Контрольні запитання та задачі до теми 13..... 315
Перелік теоретичних запитань 315
Перелік задач для самостійного опрацювання 315
Перелік використаної та рекомендованої літератури..... 316
Тема 14. Статична та динамічна маршрутизація..... 317
14.1. Переваги та призначення статичної маршрутизації..... 317
14.2. Налаштування статичного маршруту..... 320
14.3. Статичний маршрут за замовчуванням..... 323
14.4. Призначення та роль протоколів динамічної маршрутизації 325
14.5. Принципи роботи протоколів динамічної маршрутизації 328
14.6. Класифікація протоколів динамічної маршрутизації 330
14.7. Дистанційно-векторні протоколи маршрутизації 333
14.8. Протоколи маршрутизації за станом каналу 336
Контрольні запитання та задачі до теми 14..... 339
Перелік теоретичних запитань 339
Перелік задач для самостійного опрацювання 340
Перелік використаної та рекомендованої літератури..... 341
Тема 15. Налаштування OSPF маршрутизації..... 342
15.1. Розвиток та характеристики протоколу OSPF..... 342
15.2. Компоненти та принцип роботи протоколу OSPF..... 344
15.3. OSPF для однієї та декількох областей..... 347
15.4. Інкапсуляція та типи пакетів OSPF 348
15.5. Встановлення суміжності з сусідніми пристроями 351
15.6. Синхронізація баз даних OSPF 353
15.7. Налаштування процесу OSPF 354
15.8. Перевірка даних процесу OSPF 358
Контрольні запитання та задачі до теми 15..... 358
Перелік теоретичних запитань 358
Перелік задач для самостійного опрацювання 358
Перелік використаної та рекомендованої літератури..... 359

Тема 16. Списки контролю доступу..... 360
16.1. Визначення та завдання списків контролю доступу..... 360
16.2. Сеанс зв'язку та обмін даними по протоколу TCP..... 363
16.3. Стандартні і розширені ACL -списки ……..... 366
16.4. Накладення шаблонної маски ……..…….. 367
16.5. Рекомендації з використання ACL -списків ……..……... 370
16.6. Налаштування стандартних ACL -списків ……..……... 373
16.7. Використання ACL -списка для управління доступом до VTY… 376
16.8. Створення розширених ACL -списків ……..……... 378
16.9. Приклади пошуку і усунення поширених помилок ACL ……..381
Контрольні запитання та задачі до теми 16..... 386
Перелік теоретичних запитань 386
Перелік задач для самостійного опрацювання 386
Перелік використаної та рекомендованої літератури..... 387
Тема 17. Протокол DHCP..... 388
17.1. Призначення протоколу DHCP..... 388
17.2. Операція DHCPv4..... 390
17.3. Формат повідомлень DHCPv4……..…….. 392
17.4. Налаштування DHCPv4 сервера ……..…….. 394
17.5. DHCPv4-ретрансляція …..…….. 397
17.6. Налаштування маршрутизатора в якості DHCPv4 клієнта …..… 399
17.7. Пошук і усунення помилок DHCPv4…….…. 401
17.8. Перевірка налаштувань DHCPv4 на маршрутизаторі ……..… 403
Контрольні запитання та задачі до теми 17..... 404
Перелік теоретичних запитань 404
Перелік задач для самостійного опрацювання 405
Перелік використаної та рекомендованої літератури..... 406
Тема 18. Технологія NAT..... 407
18.1. Визначення та необхідність використання технології NAT……. 407
18.2. Термінологія NAT..... 409
18.3. Механізми перетворення мережевих адрес …….. 411
18.4. Порівняння технологій NAT і PAT ……..... 416
18.5. Переваги і недоліки NAT ……..…… 417
18.6. Налаштування статичного NAT ……..… 420
18.7. Налаштування і перевірка динамічного NAT…… 422
18.8. Налаштування та перевірка PAT ..…… 424
Контрольні запитання та задачі до теми 18.....428
Перелік теоретичних запитань 428
Перелік задач для самостійного опрацювання 429
Перелік використаної та рекомендованої літератури..... 430
Термінологічний покажчик 431

ПЕРЕДМОВА

Комп’ютерні системи системи та мережі є фундаментальною основою сучасної цифрової інфраструктури, забезпечуючи ефективну взаємодію між користувачами, пристроями та інформаційними ресурсами. Знання принципів функціонування комп’ютерних систем та мереж дозволяє розуміти, як відбувається обробка, зберігання та передача даних у реальному часі в різних середовищах – від домашніх мереж до корпоративних і глобальних інфраструктур.

Програмно-апаратні рішення комп’ютерних систем і мереж допомагають організаціям створювати надійні, масштабовані та безпечні інформаційні середовища. Сучасні підходи до побудови комп’ютерних мереж дозволяють оптимізувати роботу інформаційних систем, забезпечувати високу доступність сервісів та швидке реагування на збої та навантаження. Розуміння архітектури комп’ютерних систем, протоколів передачі даних, принципів маршрутизації та методів забезпечення безпеки є необхідним для проєктування та адміністрування ефективних IT-інфраструктур. Сьогодні знання в галузі комп’ютерних систем і мереж застосовуються у хмарних технологіях, Інтернеті речей, телекомунікаціях, системах обробки великих даних та кібербезпеці.

Підручник призначений для студентів, які навчаються за спеціальністю F2 «Інженерія програмного забезпечення».

У підручнику розглянуто основні типи комп'ютерних систем та мереж, апаратні та програмні компоненти мереж, мережеві протоколи і стандарти, а також технології комутації та маршрутизації.

Вивчення дисципліни «Основи комп’ютерних систем і мереж» дозволяє сформувати у здобувачів освіти компетенції, необхідні для розв’язання практичних задач професійної діяльності, пов’язаної зі створенням та налаштуванням комп’ютерних мереж.

Метою вивчення дисципліни «Основи комп’ютерних систем і мереж» є формування у студентів здатностей аналізувати мережу за її логічною топологією, обирати апаратне та програмне забезпечення комп’ютерної мережі згідно визначених вимог, налаштовувати мережеві пристрої та сервіси, застосовувати в професійній діяльності розподілені дані, програми і ресурси комп’ютерних мереж.

Предметом дисципліни «Основи комп’ютерних систем і мереж» є технології комутації, маршрутизації, мережеві протоколи та інструменти безпеки комп’ютерних систем та мереж.

Вивчення дисципліни «Основи комп’ютерних систем і мереж» сприяє формуванню у студентів наступних фахових компетентностей:

- здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки);
- здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення (ПЗ);
- здатність розробляти програмне забезпечення інформаційно-пошукових систем.

Вивчення дисципліни «Основи комп’ютерних систем і мереж» сприяє формуванню у студентів наступних програмних результатів навчання:

- аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки;
- знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії ПЗ;
- знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних;
- знати засоби, аналізувати, вибирати, кваліфіковано застосовувати засоби забезпечення інформаційної безпеки (в тому числі кібербезпеки) і цілісності даних відповідно до розв'язуваних прикладних завдань та створюваних програмних систем.

Підручник «Основи комп’ютерних систем і мереж» складається з 2 розділів та 18 тем, в яких надаються теоретичні відомості, перелік теоретичних запитань для самоперевірки, перелік задач для самостійного опрацювання та список рекомендованої літератури.

РОЗДІЛ І.

ТИПИ КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ. КОМПОНЕНТИ МЕРЕЖІ. МЕРЕЖЕВІ ПРОТОКОЛИ І СТАНДАРТИ

Тема 1. Типи комп’ютерних систем і мереж. Компоненти мережі. Мережеві протоколи і стандарти. Моделі OSI і TCP / IP

Комп’ютерні системи бувають персональними, серверними, вбудованими та кластерними. Залежно від масштабу взаємодії, комп’ютерні мережі поділяються на локальні, міські та глобальні. Також виділяють персональні мережі для передавання даних на короткі дистанції, наприклад, Bluetooth. Різні типи мереж відповідають за передачу даних у межах одного приміщення або між континентами, забезпечуючи обмін інформацією у реальному часі. Основні *апаратні компоненти комп’ютерної мережі* включають мережеві адаптери, концентратори (hub), комутатори (switch), маршрутизатори (router), точки доступу (access points), а також сервери та клієнтські пристрої. Усі ці елементи працюють разом для забезпечення надійного з’єднання, маршрутизації трафіку та обміну даними між пристроями в мережі.

Протоколи – це набір правил, що визначають формат, послідовність і способи обміну даними між пристроями. Найпоширенішими є TCP/IP, HTTP, FTP, SMTP, DNS та інші. Стандарти, встановлені організаціями, такими як IEEE, ITU або IETF, забезпечують сумісність обладнання та уніфіковану роботу мереж у всьому світі. Наприклад, стандарт IEEE 802.3 визначає роботу Ethernet-мереж. Модель OSI (Open Systems Interconnection) є теоретичною семирівневою моделлю, яка описує, як дані проходять від одного пристрою до іншого через мережу – від фізичного передавання сигналів до прикладного рівня, де працюють користувацькі програми. Модель TCP/IP, що базується на практичному використанні мережі Інтернет, має чотири рівні: рівень мережевого доступу, мережевий, транспортний та прикладний рівні. Обидві моделі допомагають структурувати мережеву взаємодію та зрозуміти, як саме функціонує передача даних.

1.1 Типи комп’ютерних систем і мереж

У сучасному світі комп’ютерні системи є основою інформаційної інфраструктури. Вони використовуються у всіх сферах – від особистого використання до керування складними інженерними процесами, науковими обчисленнями чи підтримки бізнесу.

Комп’ютерна система – це сукупність апаратного та програмного забезпечення, яка забезпечує обробку, зберігання та передачу інформації відповідно до заданих алгоритмів. Існує кілька основних типів комп’ютерних систем: персональні комп’ютери, сервери, вбудовані системи, кластерні та хмарні обчислювальні системи.

Персональні комп’ютери (ПК) – це найпоширеніший тип систем, призначений для індивідуального користування. Вони використовуються для роботи з документами, перегляду інтернет-ресурсів, програмування та розваг. Прикладом персонального комп’ютера є ноутбук або настільний ПК, який використовується студентом для навчання або фахівцем для роботи з офісним ПЗ.

Сервери – це потужні комп’ютери, які надають ресурси іншим пристроям у мережі. Вони можуть виконувати функції зберігання даних, управління базами даних, вебобслуговування тощо. Наприклад, пошукова система Google використовує тисячі серверів для обробки запитів користувачів і доставки інформації в режимі реального часу.

Вбудовані комп’ютерні системи – це спеціалізовані пристрої з обмеженими функціями, вбудовані в інші прилади або машини. Такі системи використовуються в побутовій техніці, автомобілях, медичному обладнанні тощо. Наприклад, мікроконтролер у пральній машині керує режимами прання, контролюючи час і температуру.

Кластерні системи – це група об'єднаних комп’ютерів, які працюють як єдина система для досягнення високої продуктивності. Кластери часто застосовуються в наукових розрахунках, обробці великих даних або в моделюванні складних фізичних процесів, таких як прогноз погоди чи моделювання поведінки матеріалів.

Хмарні обчислення – це модель, за якої ресурси (сервери, сховища, додатки) надаються користувачеві як послуга через мережу Інтернет. Цей тип систем дозволяє отримувати доступ до обчислювальних потужностей без необхідності мати фізичне обладнання. Наприклад, Google Drive або Amazon Web Services дозволяють зберігати дані або

запускати складні обчислення у хмарі.

Разом із різноманіттям комп'ютерних систем розвиваються і комп'ютерні мережі – структури, які об'єднують комп'ютери для обміну інформацією. Основна класифікація мереж здійснюється за масштабом охоплення: персональні (PAN), локальні (LAN), міські (MAN) та глобальні (WAN) мережі.

PAN (Personal Area Network) – персональні мережі, які охоплюють дуже малу відстань, зазвичай кілька метрів. Найчастіше вони використовуються для з'єднання пристроїв одного користувача, наприклад, смартфона, смарт-годинника та ноутбука через Bluetooth.

LAN (Local Area Network) – локальні мережі, що охоплюють невелику географічну територію, наприклад, офіс або будівлю. Типовим прикладом LAN є мережа в університетській лабораторії, де десятки комп'ютерів підключені до одного комутатора та обмінюються інформацією з сервером.

MAN (Metropolitan Area Network) – міські мережі, які охоплюють велике місто або агломерацію. Вони використовуються, наприклад, для об'єднання філій одного банку в межах міста в єдину мережу для обміну фінансовими даними.

WAN (Wide Area Network) – глобальні мережі, які охоплюють велику територію (країни або континенти). Найвідомішим прикладом WAN є Інтернет – глобальна мережа, що об'єднує мільярди пристроїв по всьому світу.

Залежно від конфігурації мережі виділяють *одноранговi (peer-to-peer)* та *клієнт-серверні* моделі. В одноранговій мережі всі пристрої мають рівні права і можуть бути одночасно клієнтами й серверами. Це типово для домашніх мереж або невеликих офісів.

У клієнт-серверній моделі існує спеціалізований сервер, який обслуговує запити клієнтів. Така архітектура дозволяє централізовано керувати ресурсами, що особливо важливо у великих організаціях.

Також комп'ютерні мережі поділяють за типом передавання даних – *дротові* (Ethernet) та *бездротові* (Wi-Fi, LTE, 5G).

Дротові мережі зазвичай мають більшу швидкість та надійність, але потребують фізичної інфраструктури. Бездротові мережі забезпечують мобільність користувачів і зручність підключення, особливо актуальні в умовах сучасного цифрового суспільства.

Мережі SDN (Software-Defined Networking) – це підхід до проєктування комп'ютерних мереж, який відокремлює площину управління (control plane) від площини даних (data plane) (рис. 1.1).

У традиційних мережах обидві ці функції реалізовані на одному мережевому пристрої (наприклад, маршрутизаторі), тоді як у SDN-підході функція управління зосереджене в централізованому контролері. Цей контролер програмно керує мережевими пристроями за допомогою відкритих протоколів, таких як OpenFlow.

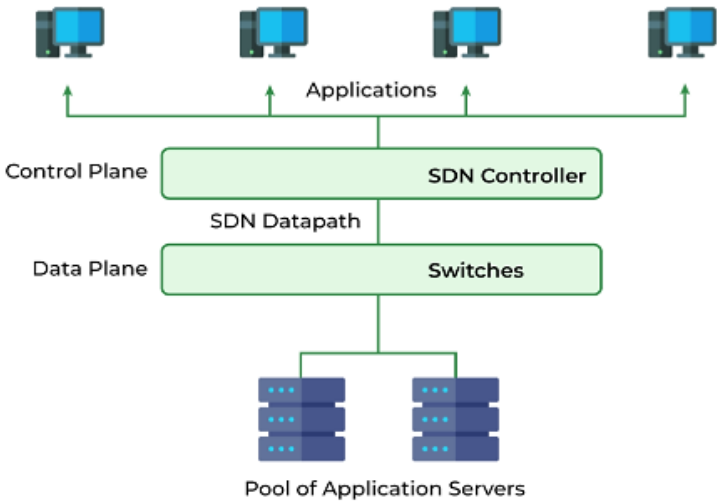


Рис. 1.1. Приклад програмно-конфігурованої мережі SDN

Основною перевагою SDN є централізоване управління мережею, що дозволяє адміністраторам програмувати та оптимізувати трафік у режимі реального часу. Це особливо корисно в дата-центрах, хмарних середовищах і великих корпоративних мережах. Гнучкість та автоматизація, які забезпечує SDN, дозволяють швидко адаптувати мережу до змін в інфраструктурі чи до вимог бізнесу. Крім того, SDN забезпечує більшу прозорість і спрощене управління політиками безпеки. Наприклад, у разі виявлення шкідливого трафіку контролер SDN може миттєво змінити маршрутизацію або заблокувати певні потоки, що значно підвищує реакцію мережі на загрози.

Однак SDN також має певні недоліки. Найбільшою проблемою є централізація: у разі збою контролера може постраждати вся мережа. Крім того, впровадження SDN вимагає оновлення мережевого обладнання та глибокої експертизи нових технологій, що може бути ресурсомістким процесом для організації.

1.2 Технології доступу в Інтернет

Історія розвитку мережі Інтернет бере початок у 1960-х роках, коли в США було створено дослідницьку мережу ARPANET. Її розробкою займалося Агентство перспективних дослідницьких проєктів Міністерства оборони США (Advanced Research Projects Agency, ARPA), і головною метою було забезпечення надійного обміну інформацією між військовими й науковими установами.

У 1969 році відбулося перше успішне з'єднання чотирьох університетських центрів через цю мережу.

У 1970-х роках були розроблені ключові протоколи передачі даних, зокрема, TCP/IP, які згодом стали основою для глобальної мережі Інтернет.

1 січня 1983 року вважається офіційною датою народження мережі Інтернет, оскільки саме тоді мережа ARPANET повністю перейшла на протокол TCP/IP. Це дозволило з'єднувати різні типи комп'ютерних мереж між собою, утворюючи єдину інформаційну систему.

Протягом 1980–1990-х років Інтернет починає активно розвиватися у наукових, освітніх та урядових установах. У 1991 році був створений Всесвітній павутинний простір (World Wide Web) Тімом Бернерсом-Лі, що зробило мережу Інтернет доступною і зрозумілою для широкого загалу. Поява веббраузерів, таких як Mosaic і Netscape Navigator, значно спростила навігацію мережею.

У 2000-х роках мережа Інтернет стала невід'ємною частиною повсякденного життя. Зростання швидкості з'єднання, поява бездротових технологій і мобільного Інтернету сприяли масовій цифровізації суспільства. З'явилися соціальні мережі, хмарні сервіси та стрімінгові платформи, які повністю змінили підхід до спілкування, навчання, роботи та дозвілля.

Сьогодні мережа Інтернет є глобальним середовищем обміну інформацією, що охоплює майже всі сфери людської діяльності. Розвиток Інтернет триває й надалі – з появою таких технологій, як 5G, Інтернет речей (IoT), блокчейн і штучний інтелект. Усе це свідчить про те, що Інтернет продовжує еволюціонувати, пристосовуючись до потреб сучасного світу. Прокладання мережі Інтернет через континенти та океани є одним з найскладніших та найважливіших завдань глобальної телекомунікаційної інфраструктури. Основу міжконтинентального

зв’язку становлять підводні волоконно-оптичні кабелі, які простягаються на тисячі кілометрів на дні океанів. Ці кабелі забезпечують переважну більшість світового обсягу міжнародного трафіку, значно випереджаючи супутникові канали як за швидкістю, так і за надійністю. Процес прокладання таких кабелів вимагає залучення спеціалізованих океанських суден, обладнаних складними системами для зберігання, розмотування та точного укладання кабелю на морське дно. Попередньо проводяться детальні геофізичні дослідження дна для вибору найбільш безпечного і стабільного маршруту. Особливої уваги потребують ділянки з тектонічною активністю, вулканічною чи глибоководною структурою.

Cable Innovator – це спеціалізоване судно для прокладання підводних кабелів, яке відіграє важливу роль у глобальному поширенні інтернет-інфраструктури. Судно належить компанії Global Marine, яка має понад 170-річний досвід у сфері прокладання та обслуговування підводних телекомунікаційних мереж. Cable Innovator має високу вантажопідйомність, велику кабельну ємність і сучасне навігаційне обладнання, що дозволяє точно прокладати кабелі на великій глибині. Судно оснащене спеціальними кабельними барабанами, лініями натягу, системами управління прокладкою, а також дистанційно керованими апаратами (remotely operated underwater vehicle, ROV) для роботи на морському дні. Cable Innovator виконує такі функції:

- прокладання нових волоконно-оптичних кабелів між континентами;
- обслуговування та ремонт вже існуючих кабельних ліній;
- технічне обстеження морського дна перед монтажем;
- участь у проєктах з відновлення комунікацій після аварій чи стихійних лих.

Судно працювало в різних регіонах світу – від Атлантики до Тихого океану, виконуючи як комерційні, так і урядові завдання.

Cable Innovator – це приклад того, як технології, морська інженерія та телекомунікації поєднуються в одному потужному інструменті для забезпечення стабільного глобального зв’язку. Його робота є невидимою частиною повсякденного Інтернету – але саме завдяки таким суднам можливий безперебійний обмін даними між континентами (рис. 1.2).

Рис. 1.2. Судно Cable Innovator
(переносить до 8500 тонн кабелів та прокладає їх під водою)

Судно Cable Innovator здатне вміщувати до 8500 тонн кабелю, це означає, що на його борт можна завантажити велику кількість волоконно-оптичного кабелю, який зберігається на спеціальних барабанах і поступово розмотується та укладається на морське дно під час місії (зазвичай це можуть бути тисячі кілометрів одного суцільного кабелю). Самі кабелі мають багатошарову будову: внутрішній сердечник містить оптичні волокна, які передають світлові імпульси, а зовнішні шари захищають їх від фізичних пошкоджень, корозії та тиску води. На окремих ділянках, особливо поблизу узбережжя, кабелі укладаються під землею або захищаються сталевими кожухами через підвищений ризик пошкоджень від якорів, риболовних сіток чи підводної активності.

Технічне обслуговування та ремонт пошкоджених кабелів – ще один складний аспект. У випадку збоїв спеціальні кораблі з глибоководними роботами локалізують пошкоджену ділянку,

піднімають кабель на поверхню, усувають несправність і знову занурюють його. Усе це потребує висококваліфікованих фахівців, точного обладнання та суворого дотримання технологій.

Сьогодні підводна кабельна інфраструктура постійно модернізується. Нові лінії мають більшу пропускну здатність, використовують передові технології оптичної модуляції, а також системи живлення, що дозволяють передавати дані на тисячі кілометрів без втрат. Будівництво таких мереж координується глобальними консорціумами операторів зв’язку, технологічних компаній і навіть урядів, адже ці кабелі є критично важливими для міжнародної економіки, безпеки та комунікацій. Прокладання міжконтинентальних підводних кабелів є ключовим етапом у розвитку глобальної інтернет-інфраструктури. Ці кабелі проходять через океани та моря та забезпечують понад 95 % глобального міжконтинентального інтернет-трафіку, з'єднуючи континенти та країни в єдину цифрову мережу.

Сучасні підводні кабелі – це високотехнологічні волоконно-оптичні системи, що складаються з кількох шарів захисту, включаючи сталеву броню та поліетиленову оболонку. Вони можуть досягати довжини понад 13 000 км, як, наприклад, кабель Firmina, що з'єднує США з Аргентиною, Бразилією та Уругваєм. Для підтримки сигналу на таких відстанях використовуються повторювачі, які встановлюються приблизно кожні 100 км.

Прокладання підводних кабелів вимагає значних зусиль та координації між країнами, компаніями та міжнародними організаціями. Процес включає вивчення морського дна, уникнення екологічно чутливих зон та забезпечення безпеки кабелів від пошкоджень, спричинених як природними факторами, так і людською діяльністю.

Існують спеціалізовані мапи, що показують усі діючі кабелі у світі, де можна переглянути маршрути, власників і рік введення в експлуатацію кожної системи. Ці мапи відображають складну глобальну мережу зв’язку, яка підтримує функціонування сучасного цифрового світу.

Для візуалізації глобальної мережі підводних кабелів можна скористатися інтерактивною картою від TeleGeography: Submarine Cable Map (рис. 1.3). Цей ресурс надає детальну інформацію про існуючі та заплановані кабельні системи, їх маршрути та точки приземлення.

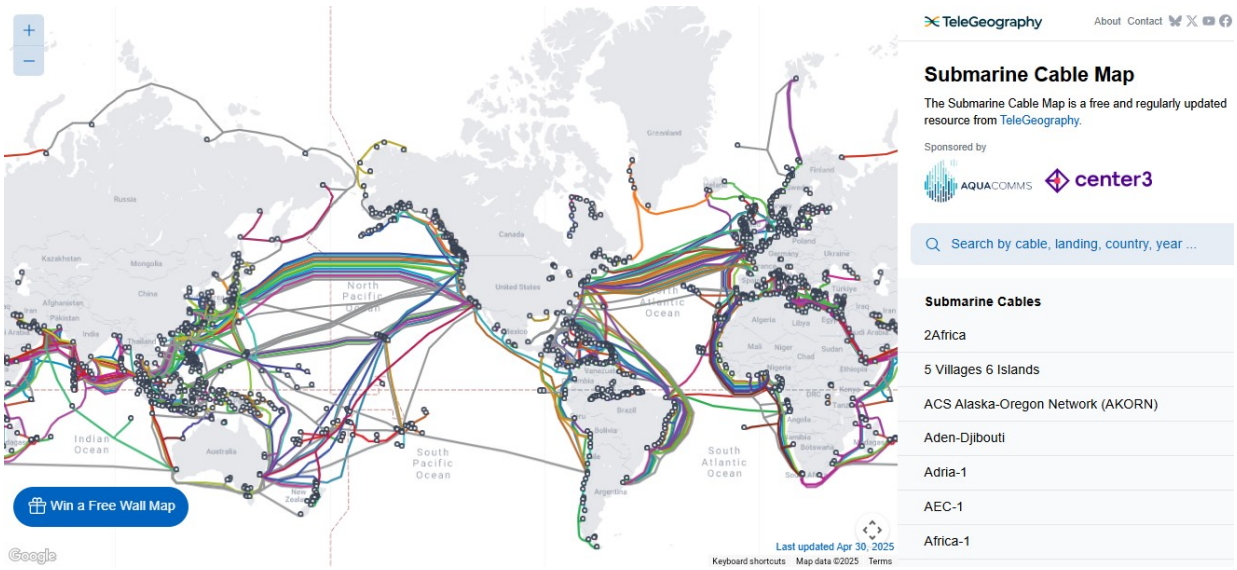


Рис. 1.3. Submarine Cable Map (https://www.submarinecablemap.com/)

Міжконтинентальні кабелі можуть прокладатися на глибинах до 8000 метрів і можуть досягати довжини понад 40 000 км, охоплюючи навіть найвіддаленіші ділянки планети.

Існує кілька типів міжконтинентальних кабелів, які відрізняються за своєю конструкцією та призначенням. Основними є волоконно-оптичні кабелі, які передають дані за допомогою світлових імпульсів. Вони складаються з декількох шарів захисту: центральне оптоволокно, гідроізоляція, металеве бронювання для захисту від механічних пошкоджень (наприклад, якорів суден або глибоководної фауни), і зовнішнє полімерне покриття. Залежно від глибини прокладки, кабель може бути легшим або важчим – наприклад, на мілководді використовується додатково захищений тип кабелю.

Посилювачі сигналу (репітери) розміщуються кожні 50–100 км уздовж кабелю для підсилення світлового сигналу, який з часом втрачає інтенсивність. Сучасні системи передають дані зі швидкістю понад 100 Тбіт/с і підтримують одночасну роботу тисяч інтернет-з’єднань.

Розгортання кабельних систем потребує високоточних навігаційних та геодезичних робіт, глибоководної техніки та суден – таких як Cable Innovator, що можуть прокладати тисячі кілометрів кабелю за один рейс. Трасування маршруту відбувається з урахуванням сейсмічної активності, океанських течій, глибини та екологічних ризиків.

Технології доступу до мережі Інтернет постійно еволюціонують, забезпечуючи швидший, надійніший і ширший доступ для користувачів по всьому світу. Залежно від технічних умов і місця розташування, використовуються різні методи підключення – від традиційних телефонних ліній до сучасних супутникових систем.

Однією з перших технологій була dial-up – доступ через комутовану телефонну мережу з використанням модема, що передавав цифрові сигнали у вигляді звуку. Хоча ця технологія забезпечувала низьку швидкість (до 56 кбіт/с), вона зробила мережу Інтернет доступною у багатьох регіонах ще у 1990-х роках. Згодом її витіснила широкосмугова передача даних – з використанням технологій DSL, ADSL, VDSL, які дозволили користуватися телефоном і Інтернетом одночасно та отримали значно вищу швидкість передачі даних.

Кабельний Інтернет з використанням коаксіальних кабелів для передачі цифрових даних став наступним кроком у розвитку доступу до мережі Інтернет. Завдяки широкій пропускній здатності, ця технологія забезпечила стабільні високі швидкості, особливо в міських районах. Ще потужнішою стала оптоволоконна технологія FTTx (Fiber To The X), що дозволяє передавати дані світлом через скляні волокна з мінімальними втратами сигналу та підтримує швидкості понад 1 Гбіт/с.

Мобільний Інтернет – один із найпоширеніших варіантів доступу сьогодні. Починаючи з 2G, мережі мобільного зв’язку пройшли етапи 3G і 4G LTE, а нині активно впроваджується 5G, що забезпечує мілісекундну затримку та швидкість до кількох гігабітів на секунду. Це відкриває нові можливості для «розумних міст», віддаленого управління та автономного транспорту.

У віддалених або важкодоступних регіонах особливо актуальні супутникові технології, такі як Starlink або OneWeb. Вони забезпечують покриття там, де традиційна інфраструктура відсутня або економічно недоцільна. Завдяки низькоорбітальним супутникам, ці системи забезпечують меншу затримку та більшу стабільність, ніж традиційні геостаціонарні сервіси. Також важливу роль відіграють технології бездротового доступу – Wi-Fi, WiMAX, точкові радіорелейні лінії та mesh-мережі. Вони дозволяють створювати локальні зони покриття або навіть забезпечувати зв’язок на рівні цілих населених пунктів без використання кабельної інфраструктури (таблиця 1.1).

Таблиця 1.1. **Порівняльний аналіз технологій доступу до мережі Інтернет**

Технологія	Середня швидкість	Затримка (ping)	Покриття	Переваги	Недоліки
Dial-up	до 56 кбіт/с	>200 мс	Всюди, де є телефонна лінія	Дешева, доступна в сільських районах	Дуже повільна, не підтримує сучасні сервіси
DSL / ADSL / VDSL	1–100 Мбіт/с	20–50 мс	Добре працює у містах і перед-містях	Стабільне з'єднання, працює по телефонних лініях	Швидкість залежить від відстані до АТС
Кабельний Інтернет	100–1000 Мбіт/с	10–30 мс	Міста, передмістя	Висока швидкість, хороший ресурс	Ділення каналу з сусідами може впливати на швидкість
Опто-волоконний Інтернет FTTH/FTTP	300 Мбіт/с – 10+ Гбіт/с	<10 мс	Великі міста, частково передмістя	Найвища швидкість, низька затримка	Дорога інфраструктура, не скрізь доступна
3G / 4G LTE	5–100 Мбіт/с	30–100 мс	Широке покриття	Мобільність, підключення без проводів	Залежить від навантаження мережі, покриття, перешкод
5G	100 Мбіт/с – 2 Гбіт/с	<10 мс	Міста, обмежене покриття	Дуже висока швидкість, низька затримка	Потребує щільної інфраструктури, менший радіус дії
Wi-Fi (локальна мережа)	до 1 Гбіт/с	<10 мс	В межах дії роутера (20–50 м)	Зручно вдома, у кафе, офісах	Залежить від інтернету провайдера і перешкод у приміщенні
WiMAX	40–100 Мбіт/с	~50 мс	Пілотні проєкти, деякі регіони	Широка зона покриття до 50 км	Обмежене розповсюдже-ння, технологія поступово втрачає актуальність
Mesh Wi-Fi	до 1 Гбіт/с	<10 мс	Всередині будівель, кампусів	Покриття великих площ без-проводове	Дорожчий за класичний Wi-Fi
Супутни-ковий Інтернет (Starlink, тощо)	20–250 Мбіт/с	20–100 мс	Віддалені регіони, глобальне покриття	Незалежний від наземної інфра-структури	Вартість, вплив погоди, іноді вища затримка

1.3 Вимоги до мережі. Компоненти мережі

Мережа, як інфраструктура для передавання даних, повинна відповідати ряду технічних та експлуатаційних вимог, щоб забезпечити надійність, ефективність і безпеку. Однією з ключових вимог є пропускна здатність (bandwidth), яка визначає обсяг інформації, що може передаватися мережею за одиницю часу. Важливою також є затримка (latency) – час, за який дані досягають одержувача, особливо критична для онлайн-ігор, відеоконференцій та фінансових операцій. Надійність – ще одна вимога, що передбачає безперервну роботу мережі без втрат даних, з резервуванням і швидким відновленням після збоїв. Масштабованість дозволяє адаптувати мережу до зростання кількості користувачів або пристроїв, а безпека гарантує захист від несанкціонованого доступу, зловмисних атак і витоків інформації.

Сучасна комп'ютерна мережа складається з кількох основних компонентів. У центрі будь-якої мережі знаходяться мережеві пристрої: маршрутизатори (роутери), які спрямовують пакети між сегментами мереж, і комутатори, що з'єднують пристрої в межах локальної мережі. Важливу роль відіграють мережеві кабелі (мідні, оптичні) або бездротові канали (Wi-Fi, 4G/5G), через які відбувається передача сигналів. Також ключовим елементом є сервери, які забезпечують спільний доступ до файлів, програм, баз даних чи ресурсів Інтернету. На кінцях мережі – клієнтські (кінцеві) пристрої: комп'ютери, смартфони, IoT-пристрої, які використовують ці ресурси. Нарешті, мережа не функціонує без програмного забезпечення, яке керує з'єднаннями, протоколами, безпекою та моніторингом.

Маршрутизатор (Router) є центральним елементом для з'єднання мереж різного типу. Він аналізує IP-адреси пакетів даних та визначає найоптимальніший шлях їх передавання до місця призначення. У домашніх умовах він поєднує локальну мережу з мережею Інтернет, а в корпоративних – забезпечує складну маршрутизацію між підмережами. Сучасні маршрутизатори підтримують бездротовий зв'язок (Wi-Fi), NAT, брандмауери, VPN та динамічно змінюють таблиці маршрутизації.

Комутатор (Switch) працює на канальному рівні (L2) моделі OSI та використовується для з'єднання пристроїв у межах локальної мережі. Він розпізнає MAC-адреси пристроїв і надсилає пакети даних лише на

порт, до якого під’єднаний конкретний отримувач. Це забезпечує вищу ефективність у порівнянні з концентраторами (hub). Сучасні комутатори можуть мати функції рівня 3 (маршрутизації), VLAN, QoS і підтримку PoE (живлення через Ethernet).

Точка доступу (Access Point) – це обладнання, що забезпечує бездротове з’єднання пристроїв із мережею. Точки доступу зазвичай підключені до комутатора або маршрутизатора через Ethernet та створюють Wi-Fi-зону для користувачів. Вони підтримують стандарти IEEE 802.11 (a/b/g/n/ac/ax), а також механізми автентифікації, шифрування (WPA3) та керування навантаженням між пристроями.

Мережевий кабель – фізичне з’єднання пристроїв у мережі забезпечується кабелями. Найпоширенішими є вита пара (UTP/STP) для локальних мереж і оптоволоконні кабелі для магістральних ліній зв’язку. Оптика забезпечує високу пропускну здатність і мінімальні затримки на великі відстані, тоді як мідні кабелі дешевші та простіші в монтажі, але обмежені за швидкістю та дальністю.

Мережеві протоколи визначають правила обміну даними між пристроями. TCP/IP – основа Інтернету, забезпечує розбиття даних на пакети, їх доставку, контроль цілісності та повторну збірку. HTTP/HTTPS керує передаванням вебсторінок. FTP/SFTP – передає файли. DNS перетворює доменні імена в IP-адреси. DHCP автоматично видає IP-адреси клієнтам. Усі ці протоколи формують "мову", якою "спілкуються" пристрої в мережі.

Брандмауер (Firewall) – це або програмний, або апаратний інструмент, що фільтрує вхідний та вихідний трафік згідно з певними правилами. Його функція – забезпечити безпеку мережі, блокуючи підозрілу активність, несанкціонований доступ або шкідливі програми (рис. 1.4).

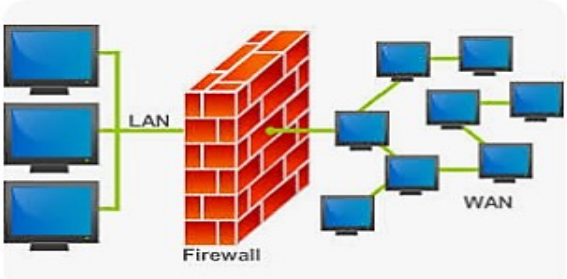


Рис. 1.4. Представлення брандмауера у фізичній топології мережі

Комп’ютерні мережі можуть значно відрізнятися за розмірами – від найпростіших, що об’єднують лише два пристрої, до глобальних систем, які включають мільйони комп’ютерів і гаджетів. Такі мережі активно використовуються людьми, які працюють віддалено – з дому або з іншого місця – для підключення до внутрішніх корпоративних ресурсів або централізованих сервісів.

Малий бізнес та домашні офіси застосовують комп’ютерні мережі для реклами своїх послуг, ведення онлайн-продажів, замовлення матеріалів та спілкування з клієнтами.

У великих компаніях мережі мають ще ширше застосування – вони дозволяють працівникам зберігати, опрацьовувати та отримувати доступ до даних, розміщених на внутрішніх серверах.

Мережеві технології також забезпечують ефективну комунікацію через використання електронної пошти, обміну повідомленнями в реальному часі та спільної роботи над документами.

Кожен пристрій, що підключений до мережі та бере участь в обміні даними, називають вузлом або кінцевим пристроєм. Такі вузли можуть як отримувати, так і передавати інформацію. Сьогодні мережеві пристрої можуть виконувати роль клієнта, сервера або обидві ролі водночас, залежно від встановленого програмного забезпечення.

Сервер – це спеціалізований комп’ютер або програмне забезпечення, що обслуговує клієнтські запити в мережі, дозволяє надавати іншим вузлам послуги, наприклад, доступ до електронної пошти або вебконтенту. Кожна служба потребує окремої серверної програми: для вебсайтів – вебсервер, для файлів – файловий сервер, тощо. Сервери можуть виконувати роль вебсерверів, файлових сховищ, баз даних, DNS, DHCP чи VPN. Вони мають високу продуктивність, надійність і часто працюють цілодобово в центрах обробки даних.

Клієнтські пристрої, у свою чергу, мають програми, які дозволяють надсилати запити й відображати отримані від серверів дані. Яскравий приклад – веббраузер, такий як Google Chrome або Mozilla Firefox. Один сервер може обслуговувати багато клієнтів одночасно. Також на одному пристрої можна встановити кілька типів серверного ПЗ, наприклад, щоб він працював і як поштовий, і як файловий сервер. Так само й клієнт може мати одночасно декілька клієнтських додатків: для перегляду сайтів, читання пошти, обміну повідомленнями,

прослуховування інтернет-радіо тощо. Завдяки цьому користувач може підключатися до багатьох серверів одночасно й отримувати необхідні сервіси в режимі реального часу.

Мережева інфраструктура охоплює три ключові групи складових: пристрої, середовище передачі (рис. 1.5) та сервіси. Ці елементи спільно забезпечують функціонування мережі, взаємодію користувачів і обмін інформацією між пристроями.

Мережеві пристрої та середовище передачі – це матеріальні компоненти, які можна побачити або відчути. До видимих елементів належать комп'ютери, ноутбуки, комутатори, маршрутизатори, бездротові точки доступу, мережеві кабелі. У випадку з бездротовими мережами, дані передаються через невидимі канали – наприклад, за допомогою радіохвиль чи інфрачервоного випромінювання.

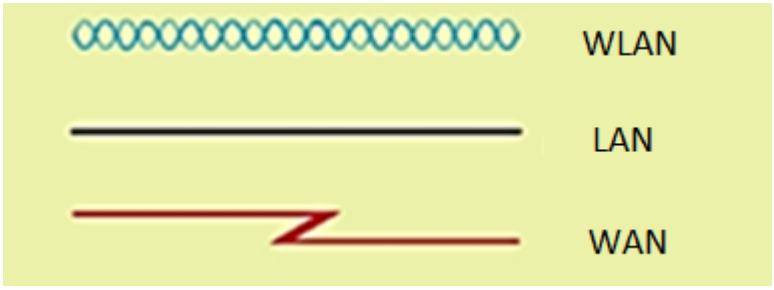


Рис. 1.5. Візуальне представлення середовища передачі даних

Сервіси – це програмні рішення, які працюють на пристроях і надають певні функції користувачам. Це можуть бути поштові служби, вебхостинг, відеозв'язок і багато інших застосунків. Крім сервісів, у мережі діють процеси, які забезпечують маршрутизацію, доставку, перерозподіл і контроль трафіку. Хоч ці процеси й не мають фізичної форми, вони критично важливі для забезпечення стабільної роботи мережі.

Кінцеві пристрої – це апаратна техніка, з якою безпосередньо взаємодіє користувач. До них належать комп'ютери, ноутбуки, сервери, смартфони, планшети, мережеві принтери, IP-телефони, камери відеоспостереження, а також спеціалізовані пристрої, такі як POS-термінали, сканери штрих-кодів, бездротові зчитувачі карток тощо (рис. 1.6). Ці пристрої або надсилають інформацію, або приймають її, виступаючи джерелом чи адресатом даних. Для того, щоб розпізнати

конкретний пристрій у мережі, кожному з них надається унікальна адреса.

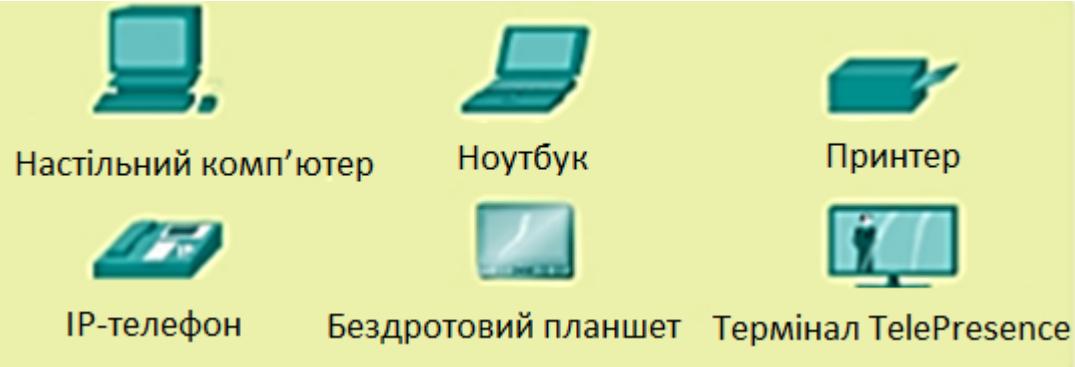


Рис. 1.6. Кінцеві пристрої

Проміжні пристрої – це технічні засоби, які організовують з'єднання між кінцевими точками. Вони забезпечують обробку, маршрутизацію, захист і контроль трафіку, працюючи у фоновому режимі. До них відносяться комутатори, точки доступу, маршрутизатори, а також апаратні брандмауери. Такі пристрої часто використовуються для поєднання різних мереж в одну інтегровану систему (рис. 1.7).

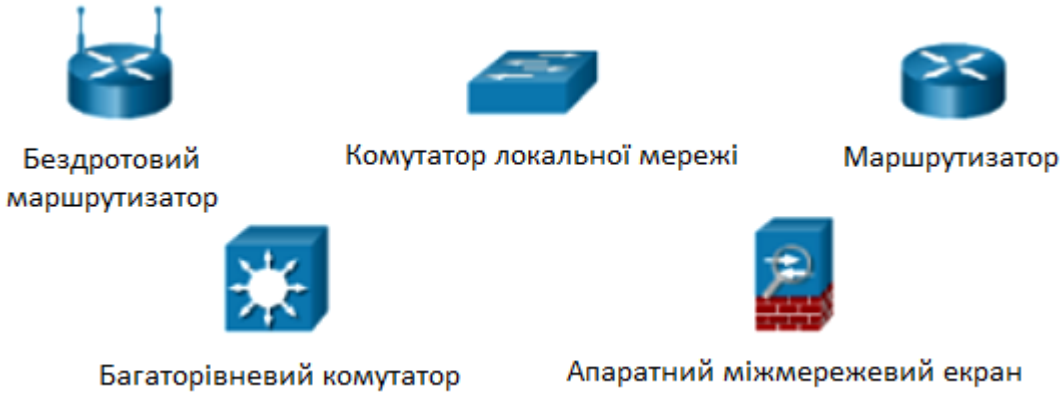


Рис. 1.7. Проміжні пристрої в середовищі моделювання Cisco Packet Tracer

Функції проміжних пристроїв включають: регенерацію сигналів, вибір оптимального маршруту для даних, реагування на збої в мережі, перенаправлення трафіку у разі проблем на одному з каналів, управління пріоритетами трафіку QoS (Quality of service), а також

контроль доступу згідно з політиками безпеки.

Для забезпечення зв'язку між пристроями потрібне середовище передачі даних. Це канал, по якому сигнал рухається від відправника до отримувача. У сучасних мережах використовують три основні типи середовищ: мідні кабелі, оптоволокно та бездротові технології. У мідних проводах інформація передається за допомогою електричних імпульсів, у волоконно-оптичних лініях – за допомогою світлових сигналів, а в бездротовому середовищі – через радіохвилі або інші види електромагнітного випромінювання.

1.4 Представлення мережі. Фізична та логічна топології мереж

Коли потрібно пояснити структуру комп'ютерної мережі, дуже корисно скористатися графічним зображенням, яке демонструє усі пристрої та канали зв'язку, що входять до складу інтегрованої мережі. Така візуалізація дозволяє зрозуміти, як між собою пов'язані окремі елементи системи. Для цього створюють спеціальні схеми, на яких за допомогою умовних позначень відображаються різні типи обладнання та з'єднань. Подібні зображення називають *топологічними схемами мережі*. Як і в будь-якій професійній галузі, у сфері мережевих технологій застосовується стандартний набір символів та термінів для опису кінцевих пристроїв, проміжного обладнання та типів середовища передачі даних. Уміння інтерпретувати логічні моделі, що відображають фізичну структуру мережі, є важливим для розуміння її побудови та функціонування.

Під час аналізу з'єднань між елементами використовуються спеціалізовані терміни, які описують ключові компоненти.

Мережева інтерфейсна карта NIC (network interface controller/card) – це пристрій, що забезпечує комп'ютеру або іншому обладнанню фізичне підключення до мережі (рис. 1.8).

Саме до цієї плати під'єднується середовище передавання даних – наприклад, кабель Ethernet.

Фізичний порт – це роз'єм на обладнанні, через який середовище зв'язку підключається до пристрою. Такі порти можна знайти, наприклад, на мережевих комутаторах або комп'ютерах.

Рис. 1.8. Мережева інтерфейсна карта

Інтерфейс – це спеціалізований порт маршрутизатора чи іншого мережевого пристрою, який з'єднується з окремою мережею.

У випадку з маршрутизаторами кожен інтерфейс відповідає за з'єднання з певною мережею, що дозволяє здійснювати передачу даних між різними сегментами.

Представлення комп'ютерної мережі охоплює її структуру, спосіб підключення пристроїв і логіку обміну даними між ними.

Для зручності розрізняють фізичну та логічну топології мережі. Кожна з них описує мережу з різних точок зору: *фізична* – фактичне розміщення пристроїв і кабелів, а *логічна* – як дані передаються всередині цієї мережі.

Фізична топологія мережі – це схема, що описує фізичне розташування кабелів, комутаторів, маршрутизаторів та інших пристроїв. До найпоширеніших фізичних структур належать «зірка», «шина», «кільце», «дерево» та «сітка».

Наприклад, у топології «зірка» всі пристрої підключаються до центрального комутатора або хаба, тоді як у «шині» всі пристрої під'єднані до одного кабелю. Найчастіше сьогодні використовується топологія «зірка» завдяки її надійності та зручності керування, тоді як основним недоліком топології «кільце» є вразливість до виходу з ладу будь-якого вузла, що порушує всю мережу.

Вибір фізичної топології впливає на витрати, складність монтажу, зручність обслуговування та масштабованість мережі (рис. 1.9).

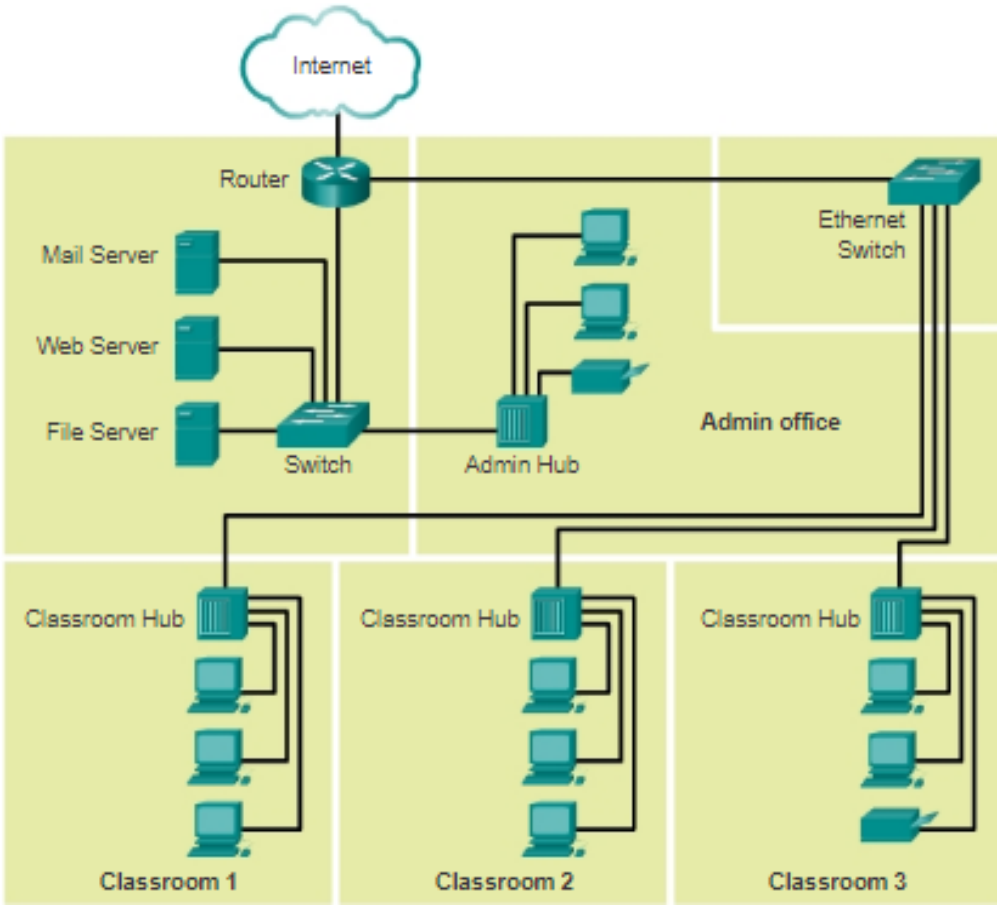


Рис. 1.9. Приклад фізичної топології мережі

Логічна топологія мережі, у свою чергу, описує шляхи, якими передаються пакети даних у мережі, незалежно від реального фізичного з'єднання.

На рис. 1.10 зображено мережеву інфраструктуру, яка складається з декількох локальних мереж, об'єднаних маршрутизатором R1 та підключених до мережі Інтернет.

Ліворуч розташована мережа 192.168.10.0, яка включає веб-, email- і файловий сервери, підключені через комутатор S1.

Нижче знаходиться мережа 192.168.11.0 з двома комп'ютерами, з'єднаними через комутатор S2.

Справа зображено три окремі мережі – 192.168.100.0, 192.168.101.0 і 192.168.102.0, кожна з яких містить по одному комутатору (S3, S4, S5) та кінцевий пристрій. Усі ці мережі з'єднуються через маршрутизатор R1, який забезпечує міжмережеву комунікацію та доступ до мережі Інтернет.

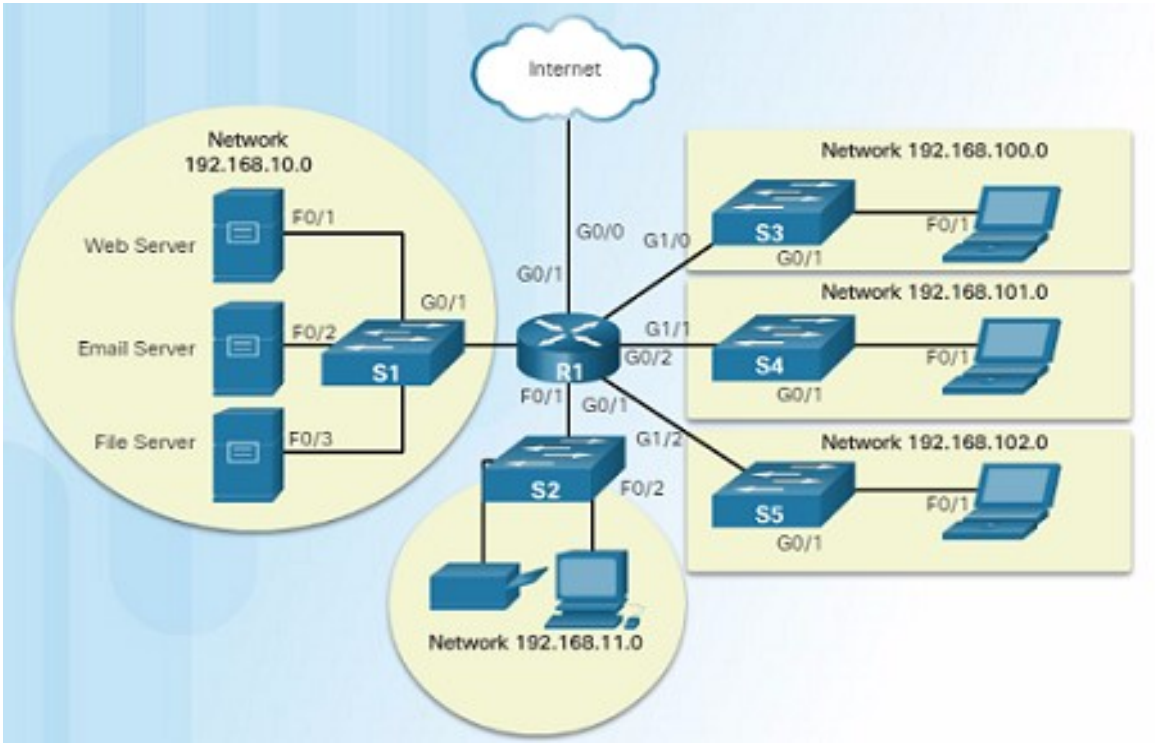


Рис. 1.10. Приклад логічної топології мережі

У контексті логічної топології, IP-адресація відіграє ключову роль у маршрутизації даних між пристроями мережі. Логічна топологія описує, як дані передаються мережею, незалежно від фізичних з'єднань, і саме IP-адреси дозволяють мережевим пристроям ідентифікувати себе та один одного для обміну інформацією.

Приклад: у мережі з фізичною топологією "зірка" пристрої фізично підключені до одного комутатора, але логічна топологія визначає, що пристрій з IP-адресою 192.168.0.5 може надсилати дані пристрою з IP-адресою 192.168.0.10, і маршрутизатор/комутатор визначає маршрут доставки пакета на основі цих IP-адрес.

Розуміння відмінностей між фізичною та логічною топологією є важливим для ефективного проєктування мереж, їхнього аналізу та усунення несправностей. Адже продуктивність, безпека та стабільність роботи фізичної інфраструктури значною мірою залежать саме від обраної топології. Середовища моделювання комп'ютерних мереж дозволяють створювати віртуальні мережеві інфраструктури для тестування, аналізу, навчання та дослідження без необхідності використання фізичного обладнання. Вони широко застосовуються в освіті, науковій діяльності та професійному адмініструванні мереж.

Розглянемо найпопулярніші середовища моделювання комп'ютерних мереж з їхніми перевагами та недоліками.

Cisco Packet Tracer – це середовище симуляції комп'ютерних мереж від компанії Cisco, яке добре підходить для навчання базовим навичкам проєктування мереж (рис. 1.11).

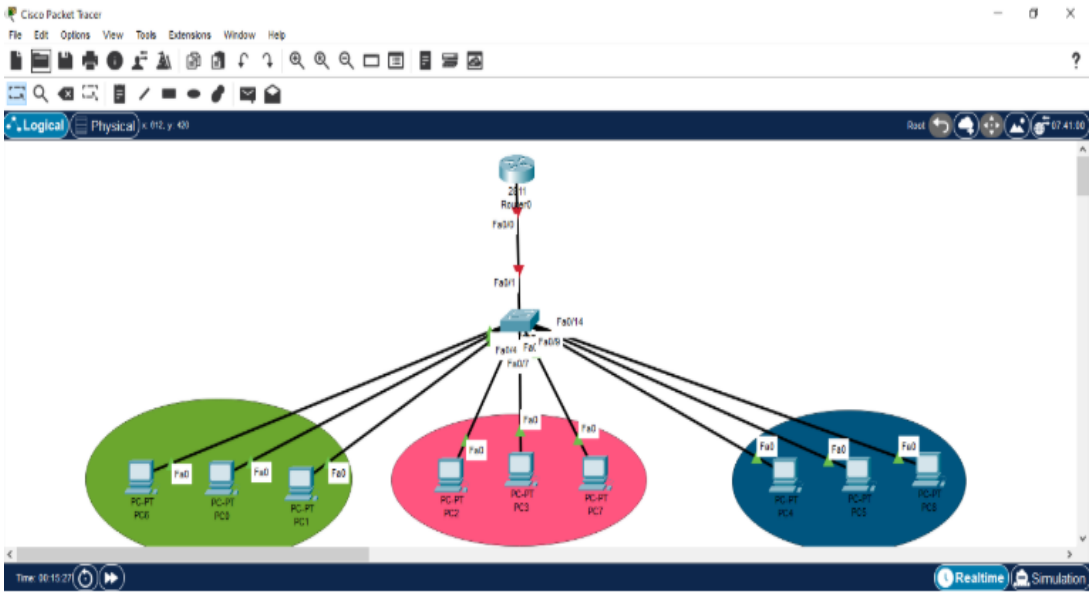


Рис. 1.11. Середовище моделювання комп'ютерних мереж Cisco Packet Tracer

Переваги Cisco Packet Tracer:

- простий у використанні графічний інтерфейс користувача;
- добре підходить для початківців та студентів мережевої академії Cisco Networking Academy;
- підтримка великої кількості пристроїв Cisco та базових протоколів;
- можливість створення сценаріїв з анімацією мережевого трафіку.

Недоліки Cisco Packet Tracer:

- обмежена функціональність щодо підтримки деяких мережевих протоколів;
- не підходить для тестування нестандартних або складних мережевих топологій.

GNS3 (Graphical Network Simulator-3) – це емулятор, який дозволяє запускати реальні образи мережевих операційних систем для створення складних топологій комп'ютерних мереж (рис. 1.12).

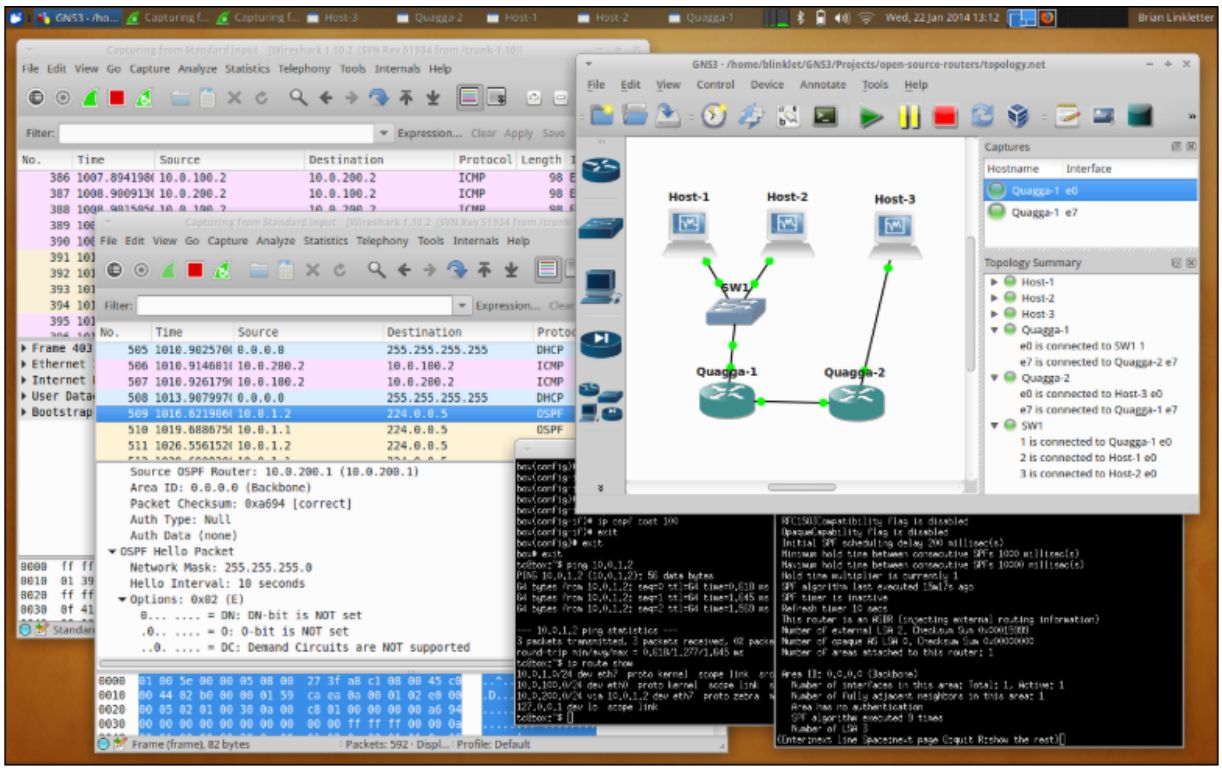


Рис. 1.12. Середовище моделювання комп'ютерних мереж GNS3

Переваги Graphical Network Simulator-3:

- можливість запускати реальні образи Cisco IOS, Juniper, MikroTik тощо;
- підтримка складних топологій та широкого спектру протоколів;
- інтеграція з VirtualBox, VMware для віртуалізації серверів та робочих станцій.

Недоліки Graphical Network Simulator-3:

- вимагає більше ресурсів комп'ютера;
- початкове налаштування складніше у порівнянні з Packet Tracer;
- потрібні ліцензії або образи IOS, що може бути обмеженням.

EVE-NG (Emulated Virtual Environment – Next Generation) – це професійна платформа віртуалізації мереж, що підтримує мультивендорну емуляцію через вебінтерфейс.

Переваги Emulated Virtual Environment – Next Generation:

- підтримка багатьох вендорів (Cisco, Juniper, Fortinet, Palo Alto та ін.), вебінтерфейс для централізованого керування лабораторіями;
- професійне середовище, яке часто використовується у корпоративному навчанні (рис. 1.13).

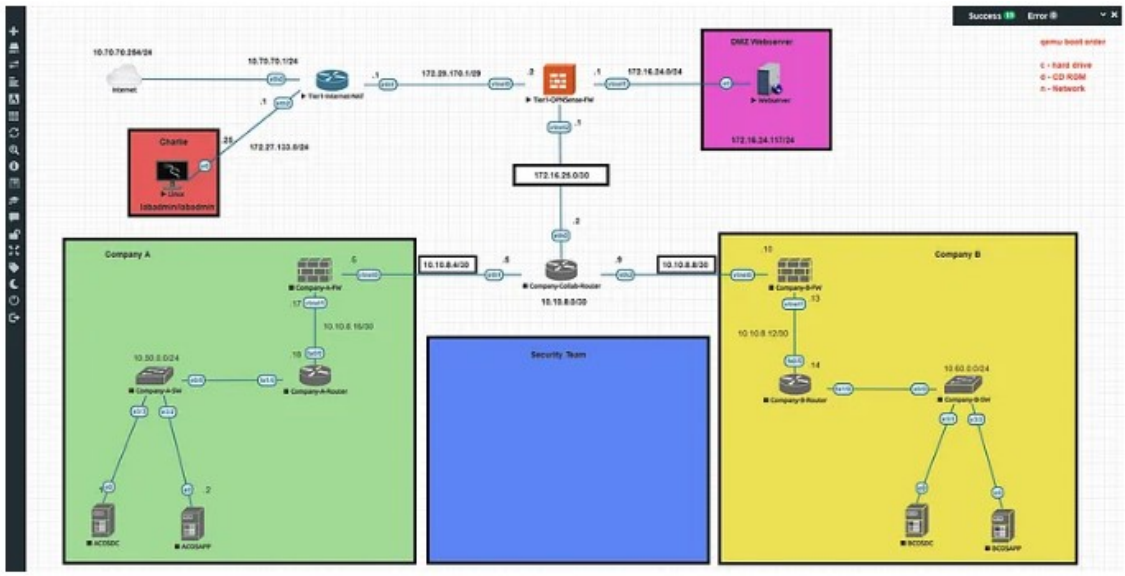


Рис. 1.13. Середовище моделювання комп'ютерних мереж EVE-NG

Недоліки Emulated Virtual Environment – Next Generation:

- встановлення та налаштування потребує часу та технічної підготовки, менш інтуїтивний графічний інтерфейс для новачків;
- для розгортання великомасштабних топологій потрібне потужне апаратне забезпечення або сервер.

NetSim – це навчальне середовище від Boson з готовими лабораторіями для підготовки до сертифікацій Cisco, яке симулює багато функцій IOS (рис. 1.14).

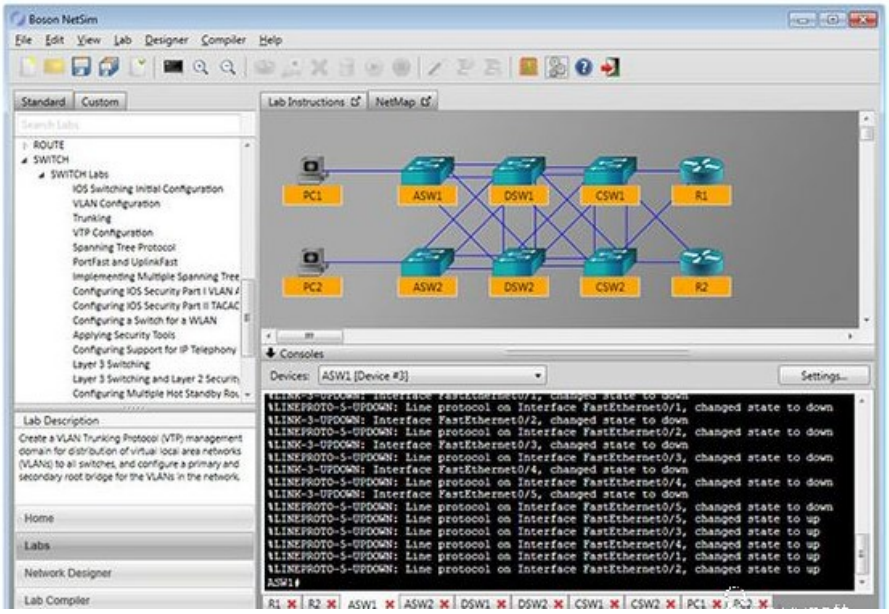


Рис. 1.14. Середовище моделювання комп'ютерних мереж EVE-NG

Переваги NetSim:

- зручний графічний інтерфейс користувача;
- симулює багато аспектів Cisco IOS.

Недоліки NetSim:

- комерційне ПЗ – потребує купівлі ліцензії;
- обмежене використання за межами навчання, не підходить для емуляції нестандартних сценаріїв.

Mininet – це середовище моделювання, орієнтоване на дослідження та навчання у сфері програмно-орієнтованих мереж SDN (рис. 1.15).

Переваги Mininet:

- призначене для моделювання програмно-орієнтованих мереж SDN;
- відкрите програмне забезпечення;
- добре підходить для досліджень з OpenFlow, SDN-контролерами (наприклад, POX, Ryu).

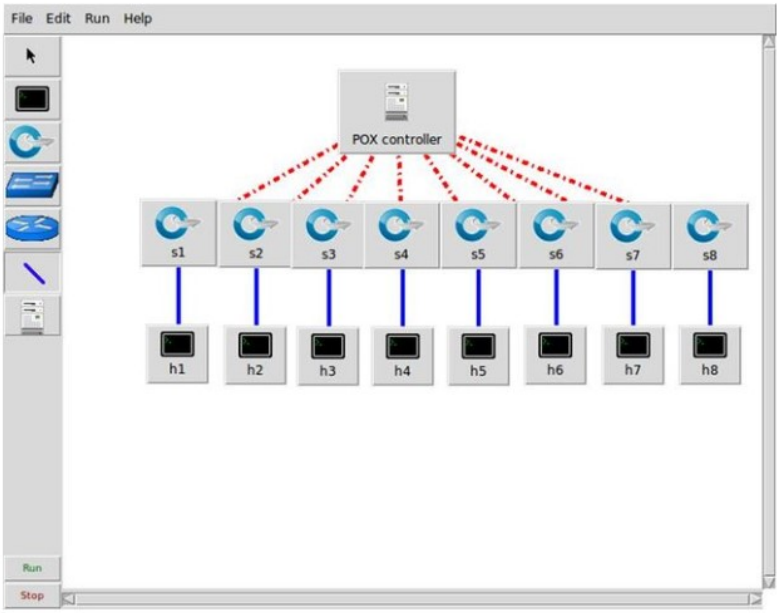


Рис. 1.15. Середовище моделювання комп'ютерних мереж Mininet

Недоліки Mininet:

- командний інтерфейс може бути незручним для новачків;
- орієнтоване переважно на Linux-середовище.

Вибір середовища моделювання залежить від цілей користувача:

для початківців і навчання оптимальним буде середовище моделювання Cisco Packet Tracer; для просунутих користувачів, які хочуть працювати з реальними образами ОС маршрутизаторів і комутаторів – GNS3 або EVE-NG. Для підготовки до іспитів зручно використовувати NetSim, а для досліджень у галузі SDN – Mininet.

1.5 Хмарні обчислення. Центри обробки даних

Хмарні обчислення – це використання обладнання та програмного забезпечення, які доступні користувачам через мережу як послуга. Компанії користуються такими ресурсами та сплачують за їх використання. При цьому обчислювальні операції не виконуються локальними комп'ютерами – замість цього обробку забезпечують віддалені сервери, які входять до складу «хмари».

«Хмара» (cloud) – це модель обчислень, у якій обробка даних, зберігання й запуск додатків здійснюються не на локальних комп'ютерах, а на віддалених серверах, доступ до яких надається через Інтернет. Вона дозволяє користувачам використовувати обчислювальні ресурси як послугу – динамічно, масштабовано та за потреби. Хмарні сервіси звільняють користувача від необхідності обслуговувати фізичну інфраструктуру, забезпечуючи гнучкість, надійність і економію ресурсів. Це значно зменшує вимоги до потужності пристроїв користувачів і до встановленого на них програмного забезпечення. Для доступу до хмарних ресурсів часто достатньо лише веббраузера.

Хмарні технології радикально змінюють підходи до зберігання та обробки даних. До хмарних обчислень належать будь-які сервіси, що надаються онлайн за підпискою або за схемою «оплата за фактом використання» в реальному часі. Вони дозволяють зберігати особисті файли або повний вміст диска на віддалених серверах. Бізнес може розширювати свої IT-потужності без потреби у значних витратах на власну інфраструктуру, наймання персоналу чи купівлю ліцензійного ПЗ. Такі сервіси надаються на запит і можуть бути доступні з будь-якого пристрою в будь-якій точці світу без втрат у функціональності або безпеці. Хмарні обчислення – це виконання обчислень через мережу Інтернет. Наприклад, онлайн-банкінг, покупки в інтернет-магазинах чи завантаження музики – усе це приклади хмарних сервісів. Зазвичай користувачі взаємодіють з хмарними сервісами через веббраузер, не

встановлюючи окреме програмне забезпечення на свої пристрої. Це робить хмарні сервіси доступними для широкого спектру пристроїв.

Розглянемо основні переваги хмарних обчислень.

Мобільність і зручність – доступ до інформації з будь-якого місця і в будь-який час через Інтернет. Швидке впровадження – IT-відділ може більше зосередитися на аналізі даних і забезпеченні взаємодії, а не на технічному обслуговуванні. Економія коштів – немає потреби витрачатися на фізичне обладнання та його налаштування. Оптимізація ресурсів – зекономлені IT-ресурси можна направити на інші потреби компанії. Гнучкість бізнесу – нові застосунки та сервіси легко інтегруються, що дозволяє швидше реагувати на зміни ринку та вимоги клієнтів.

Хмарна інфраструктура базується на центрах обробки даних (ЦОД) – спеціалізованих приміщеннях, де розміщено серверне обладнання та інші критично важливі елементи:

- резервні канали передачі даних;
- високопродуктивні сервери або кластери;
- системи зберігання даних (часто SAN – Storage Area Network);
- джерела безперебійного живлення;
- системи охолодження, пожежогасіння та безпеки.

ЦОД може займати як одну кімнату, так і цілу будівлю (рис. 1.16).

Рис. 1.16. Приклади ЦОД

Сучасні ЦОД орієнтовані на роботу з великими обсягами даних і використовують віртуалізацію – технологію, що дозволяє запускати кілька віртуальних середовищ на одному фізичному сервері. Це дає змогу ефективно розподіляти ресурси і зменшувати експлуатаційні витрати.

На відміну від звичайної багатозадачності, віртуалізація дає змогу одночасно працювати декільком операційним системам на одному фізичному процесорі.

ЦОД є дорогими в розгортанні та обслуговуванні, тому власні центри обробки даних мають переважно великі компанії. Наприклад, лікарня може зберігати електронні медичні записи у власному ЦОД. Малий бізнес зазвичай орендує серверні ресурси в сторонніх провайдерів, що дозволяє знизити загальні витрати.

1.6 Мережеві протоколи і галузеві стандарти

Мережеві протоколи – це набір правил і процедур, які визначають формат, порядок і спосіб передачі даних між пристроями в комп'ютерних мережах. Вони забезпечують узгоджене спілкування між різнорідними системами, дозволяючи комп'ютерам, маршрутизаторам і іншому обладнанню ефективно обмінюватися інформацією. Серед найпоширеніших мережевих протоколів – TCP/IP, HTTP, FTP, DNS та інші, кожен з яких виконує специфічні функції у мережевому середовищі.

Галузеві стандарти, у свою чергу, забезпечують уніфікацію технологічних рішень і сумісність обладнання та програмного забезпечення різних виробників. Такі стандарти, як ISO, IEEE 802.3 (Ethernet), ITU-T рекомендації чи IETF RFC-документи, встановлюють вимоги до проєктування, реалізації та тестування мережевих рішень. Їх дотримання є критичним для забезпечення безпеки, масштабованості та стабільної роботи сучасних інформаційних систем. Взаємозв'язок між протоколами і стандартами полягає у тому, що стандарти регламентують принципи створення протоколів, а останні реалізують ці принципи на практиці. Таким чином, гармонійне поєднання обох аспектів є основою функціонування мережі Інтернет, локальних мереж та корпоративних IT-інфраструктур. У результаті користувачі отримують доступ до стабільних і сумісних сервісів, незалежно від

апаратної платформи чи географічного розташування.

Мережеві протоколи є фундаментальною основою для взаємодії пристроїв у будь-якій обчислювальній системі. Вони визначають правила форматування, передачі, отримання та обробки даних між вузлами мережі. Зокрема, стек протоколів TCP/IP, що включає рівні фізичного, канального, мережевого, транспортного та прикладного рівнів, гарантує надійність та масштабованість мережі Інтернет. Кожен рівень розв'язує власні завдання: від поділу інформації на пакети й маршрутизації до контролю помилок і впорядкування доставки, що забезпечує цілісність та послідовність даних.

Успішна взаємодія різноманітних мережевих пристроїв – від маршрутизаторів і комутаторів до серверів і клієнтів – можлива завдяки суворим галузевим стандартам.

Такі організації, як IEEE, IETF та ITU, розробляють специфікації, які охоплюють фізичні інтерфейси (наприклад, Ethernet 802.3), бездротові рішення (Wi-Fi, стандарти 802.11), а також протоколи прикладного рівня, такі як HTTP, FTP, SMTP (рис. 1.17). Дотримання цих стандартів гарантує сумісність обладнання та програмного забезпечення різних виробників, спрощує впровадження нових технологій і сприяє безперервному розвитку мережевої інфраструктури.

Рис. 1.17. Приклади міжнародних організацій зі стандартизації

IEEE (Institute of Electrical and Electronics Engineers) – Інститут інженерів з електротехніки та електроніки – це одна з найавторитетніших міжнародних організацій, що займається розробкою технічних стандартів у широкому спектрі галузей.

IEEE охоплює такі напрями, як електроенергетика, біомедичні технології, телекомунікації, обчислювальна техніка та мережеві технології. Організація налічує десятки тисяч активних членів по всьому світу і щорічно публікує сотні стандартів, які застосовуються як у промисловості, так і в науці.

Однією з найбільш відомих серій стандартів IEEE у галузі комп'ютерних мереж є сімейство IEEE 802, яке охоплює як дротові, так і бездротові технології зв'язку. Ці стандарти описують механізми доступу до середовища передачі даних та інші аспекти мережевого функціонування на канальному рівні моделі OSI.

Зокрема, стандарт IEEE 802.3 описує принципи функціонування Ethernet – домінуючої технології для локальних мереж (LAN). Стандарт визначає, як комп'ютери можуть ділити спільне середовище передачі даних за допомогою методів керування доступом, таких як CSMA/CD (Carrier Sense Multiple Access with Collision Detection).

Ethernet використовується в офісах, дата-центрах, освітніх закладах, а також на підприємствах для побудови масштабованих і швидких дротових мереж. Наприклад, коли ви підключаєте свій комп'ютер до маршрутизатора за допомогою кабелю, швидше за все, ви використовуєте саме Ethernet відповідно до IEEE 802.3.

Інший важливий стандарт – IEEE 802.11, який визначає технології бездротових локальних мереж (WLAN), відомі як Wi-Fi. Стандарт охоплює як фізичний рівень (тобто тип модуляції сигналу), так і канальний рівень (тобто управління доступом до середовища). Різні варіації стандарту – такі як 802.11n, 802.11ac, 802.11ax – забезпечують покращення швидкості передачі, ширини покриття і енергоефективності. Наприклад, коли ви підключаєтесь до Wi-Fi у кафе, вдома чи в університеті, це відбувається через реалізацію одного з підстандартів 802.11.

ISO (International Organization for Standardization) – Міжнародна організація зі стандартизації – також відіграє ключову роль у сфері інформаційних технологій. Одним із найвідоміших внесків ISO є розробка еталонної моделі OSI (Open Systems Interconnection), яка була вперше опублікована у 1984 році. Модель OSI розділяє процес передачі даних у комп'ютерних мережах на сім логічних рівнів – від фізичного до прикладного.

Кожен рівень виконує специфічну функцію, і така структурованість допомагає розробникам створювати сумісні апаратні та програмні рішення. Метою створення моделі OSI було не лише надання теоретичної основи, але й стимулювання розробки цілісного набору протоколів, які могли б використовуватись у глобальних мережах. Хоча фактичне розгортання цих протоколів в мережі Інтернет не відбулося – замість них перемогло сімейство TCP/IP, створене за участі Вінтона Серфа та Роберта Кана – сама модель OSI залишається потужним концептуальним інструментом. Вона широко використовується у навчанні, діагностиці мережевих проблем та розробці мережевого обладнання. Протоколи, створені в рамках набору OSI, застосовуються передусім у телекомунікаційних мережах. Наприклад, такі стандарти, як X.25 чи CLNP (Connectionless Network Protocol), реалізовувалися на обладнанні провайдерів зв'язку, особливо в Європі. Крім того, ISO займається розробкою стандартів маршрутизації, включаючи протоколи внутрішньої маршрутизації, які використовуються в операторських мережах.

Internet Society (ISOC) – Інтернет-товариство – міжнародна неурядова організація, що підтримує відкритий розвиток, загальнодоступність і безпечне використання мережі Інтернет по всьому світу. ISOC просуває політики, стандарти та освітні ініціативи для збереження вільної та інклюзивної архітектури мережі Інтернет.

Internet Architecture Board (IAB) – Рада з архітектури Інтернету – орган, який відповідає за нагляд за технічним розвитком Інтернету, зокрема, за формуванням загальної архітектури та затвердженням стандартів. IAB також координує взаємодію з іншими організаціями з розробки стандартів.

Internet Engineering Task Force (IETF) – Робоча група з інженерії Інтернету – відкрита спільнота фахівців, що розробляє, вдосконалює та підтримує Інтернет-протоколи, зокрема, стек TCP/IP. IETF створює технічні специфікації, які лягають в основу роботи Інтернету.

Internet Corporation for Assigned Names and Numbers (ICANN) – Корпорація з надання імен та номерів в мережі Інтернет – організація, яка керує розподілом IP-адрес, контролює систему доменних імен (DNS) і відповідає за стабільність та унікальність мережевих ідентифікаторів в Інтернеті.

Internet Assigned Numbers Authority (IANA) – Уповноважений орган з призначення номерів в Інтернеті – підрозділ ICANN, який займається технічним управлінням глобальними унікальними ідентифікаторами: IP-адресами, номерами портів, ідентифікаторами протоколів та DNS-зонами.

Internet Research Task Force (IRTF) – Дослідницька група Інтернету – орган, що проводить довгострокові фундаментальні дослідження, пов'язані з майбутнім розвитком мережі Інтернет та протоколів TCP/IP. IRTF фокусується на інноваціях, які ще не готові до стандартизації, але мають перспективу в майбутньому.

Зростання популярності хмарних сервісів, Інтернету речей та високошвидкісних мобільних мереж 5G створює перед інженерами нові виклики щодо надійності та безпеки. Саме тому поряд із класичними протоколами з'являються специфікації для захищеної передачі даних, такі як TLS/SSL для шифрування вебтрафіку, а також MQTT і CoAP для економного обміну повідомленнями в ресурсно-обмежених IoT-пристроях.

Стандарти безпеки, наприклад, серія рекомендацій NIST, визначають методи аутентифікації, авторизації і криптографічного захисту, що допомагає захистити мережу від несанкціонованого доступу та атак на конфіденційність даних. Ще одним важливим напрямком є стандарти управління мережами й моніторингу їх роботи.

SNMP (Simple Network Management Protocol) дозволяє централізовано збирати статистичні дані про стан пристроїв, налаштовувати параметри мережевого обладнання та реагувати на аварійні події. Для більш складних середовищ розроблено протоколи автоматизованого оркестрування (наприклад, NETCONF/YANG), які підтримують уніфіковане управління конфігурацією, забезпечуючи швидке розгортання сервісів та зміни політик без необхідності ручного втручання.

1.7 Моделі OSI і TCP/IP

Модель OSI (Open Systems Interconnection) була розроблена Міжнародною організацією зі стандартизації ISO у 1980-х роках з метою створення уніфікованої концептуальної основи для опису процесу обміну даними в комп'ютерних мережах. Ця модель розділяє

мережеву взаємодію на сім рівнів, кожен з яких виконує певну функцію і взаємодіє з прилеглими рівнями. Від найнижчого фізичного рівня (який відповідає за передачу бітів по фізичному середовищу) до найвищого прикладного (де працюють такі сервіси, як електронна пошта або веббраузер), модель OSI надає чіткий і зрозумілий спосіб поділу мережевих завдань. Це робить її корисною не лише для розробки, але й для діагностики мережевих проблем, а також для навчання.

Розглянемо структуру рівнів моделей OSI, TCP/IP та їх призначення (рис. 1.18).

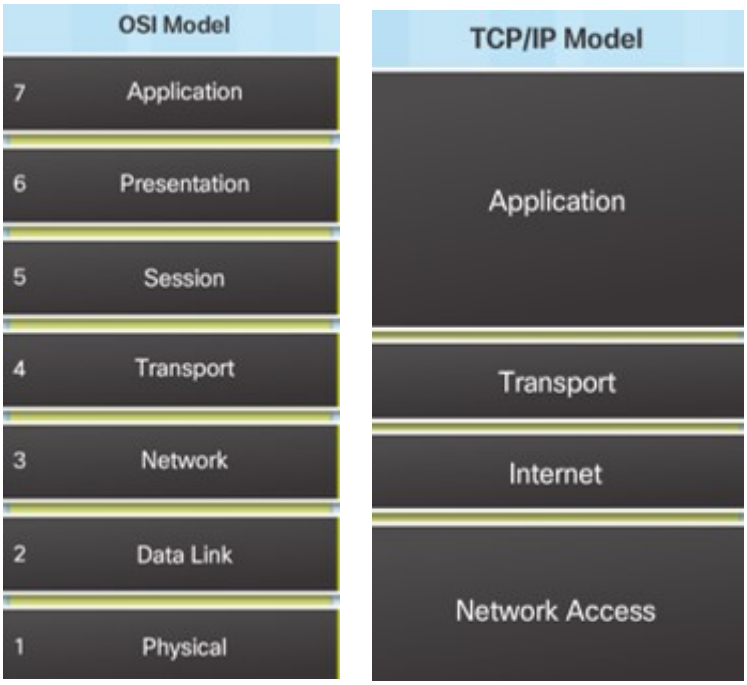


Рис. 1.18. Моделі OSI та TCP/IP

1. *Фізичний (Physical)* – забезпечує передачу бітів по каналу зв’язку, наприклад, через кабель чи радіохвилі.
2. *Канальний (Data Link)* – забезпечує надійність фізичного каналу, виявлення помилок і контроль доступу до середовища.
3. *Мережевий (Network)* – відповідає за маршрутизацію і логічну адресацію (наприклад, IP-адреси).
4. *Транспортний (Transport)* – забезпечує надійність передачі, контроль цілісності даних (TCP, наприклад, працює на цьому рівні).
5. *Сеансовий (Session)* – встановлює, управляє та завершує сеанси зв’язку між додатками.

6. *Презентаційний (Presentation)* – трансформує дані до формату, зрозумілого для прикладного рівня (наприклад, кодування, шифрування).

7. *Прикладний (Application)* – безпосередньо взаємодіє з користувачем і запускає мережеві сервіси (наприклад, HTTP, FTP, SMTP).

Модель OSI не є безпосередньо реалізованою в мережах як набір протоколів, але її структура часто використовується як еталон для аналізу і порівняння різних мережевих архітектур.

На відміну від OSI, модель TCP/IP (Transmission Control Protocol / Internet Protocol) була створена не як теоретична основа, а як практичне рішення для створення глобальної мережі. Розроблена на замовлення Міністерства оборони США у 1970-х роках, вона стала основою для сучасного Інтернету.

TCP/IP забезпечує ефективну маршрутизацію, надійність доставки даних і сумісність між різними типами комп'ютерів і мереж. Саме ця модель, а не OSI, стала домінуючою, хоча вона також має багаторівневу структуру, лише з меншою кількістю рівнів.

Розглянемо рівні моделі TCP/IP і їх функції.

Модель TCP/IP поділяється на чотири рівні, кожен з яких відповідає групі завдань.

1. *Мережевий інтерфейс (Network Access / Link)* – охоплює фізичну і канальну взаємодію з середовищем (Ethernet, Wi-Fi тощо).

2. *Мережевий (Internet)* – відповідає за адресацію та маршрутизацію, з використанням протоколу IP.

3. *Транспортний (Transport)* – забезпечує надійність і контроль доставки (TCP, або, у випадку простішої передачі – UDP).

4. *Прикладний (Application)* – охоплює всі сервіси і протоколи, з якими взаємодіє користувач (HTTP, SMTP, DNS, FTP тощо).

Модель TCP/IP демонструє гнучкість і масштабованість, необхідні для глобального використання. Наприклад, під час перегляду вебсторінки, браузер використовує протокол HTTP (прикладний рівень), який працює поверх TCP (транспортний рівень), передаючи дані через IP-мережу (мережевий рівень) і, зрештою, через фізичне з'єднання (мережевий інтерфейс).

1.8 Порівняння моделей OSI і TCP/IP

Хоча обидві моделі мають подібну мету – описати структуру мережевої взаємодії – вони мають різну філософію. OSI є формальною, деталізованою моделлю, яка більше підходить для навчання та аналізу, тоді як TCP/IP – це практична реалізація, яка була спроєктована "знизу вгору" і довела свою ефективність у реальних умовах.

Наприклад, модель TCP/IP стала основою мережі Інтернет, тоді як OSI реалізовувалась переважно у вузькоспеціалізованих телекомунікаційних системах. Розглянемо порівняння рівнів моделей OSI та TCP/IP (таблиця 1.2).

Таблиця 1.2. **Порівняння рівнів моделей OSI та TCP/IP**

Рівень OSI	Рівень TCP/IP	Функції / Приклади
7. Прикладний (Application)	4. Прикладний (Application)	HTTP, FTP, SMTP, DNS – сервіси, з якими взаємодіє користувач
6. Презентаційний (Presentation)	4. Прикладний (Application)	Кодування, шифрування, стиснення даних (наприклад, SSL/TLS)
5. Сеансовий (Session)	4. Прикладний (Application)	Управління з'єднанням між програмами (наприклад, RPC)
4. Транспортний (Transport)	3. Транспортний (Transport)	TCP, UDP – контроль доставки, відновлення помилок
3. Мережевий (Network)	2. Мережевий (Internet)	IP, ICMP – маршрутизація, логічна адресація
2. Канальний (Data Link)	1. Мережевий інтерфейс (Link)	Ethernet, PPP – передача кадрів, доступ до фізичного середовища
1. Фізичний (Physical)	1. Мережевий інтерфейс (Link)	Кабелі, роз’єми, сигнали – передача бітів у фізичному середовищі

Модель OSI часто використовується як концептуальна основа для пояснення процесу мережевої взаємодії. Наприклад, під час навчання або діагностики мережі інженери розглядають, на якому рівні моделі OSI виникла проблема – наприклад, збій на фізичному рівні (кабель пошкоджено), або на транспортному (TCP-порт заблоковано).

Інший приклад використання моделі OSI – налагодження роботи мережевого обладнання: коли налаштовується маршрутизатор, інженер

має розуміти, що IP-адреса працює на мережевому рівні (рівень 3), а MAC-адреса – на канальному (рівень 2). Модель TCP/IP використовується під час доступу до вебсайту: коли ви вводите URL у браузері, працює модель TCP/IP – DNS працює на прикладному рівні, TCP забезпечує надійність доставки (транспортний рівень), а IP відповідає за маршрутизацію (мережевий рівень).

Контрольні запитання та задачі до теми 1

Перелік теоретичних запитань

1. Які основні типи комп'ютерних мереж існують і чим вони відрізняються?
2. У чому полягає різниця між одноранговою і клієнт-серверною архітектурами мереж?
3. Які технології доступу до мережі Інтернет є найбільш поширеними на сьогодні, і які їхні переваги та недоліки?
4. Які основні вимоги пред'являються до побудови надійної комп'ютерної мережі?
5. Які компоненти входять до складу типової локальної мережі, і які функції вони виконують?
6. Що таке мережевий інтерфейсний адаптер і яка його роль у передачі даних?
7. У чому полягає різниця між фізичною і логічною топологією мережі?
8. Що таке хмарні обчислення і як вони використовуються у сучасних IT-інфраструктурах?
9. Яке призначення центрів обробки даних і які функції вони виконують?
10. Що таке мережевий протокол і які існують приклади найбільш поширених протоколів?
11. Яку роль відіграють організації IEEE та ISO у створенні мережевих стандартів?
12. Яка структура моделі OSI і яку функцію виконує кожен її рівень?
13. У чому полягає головна відмінність між моделлю OSI та TCP/IP і чому остання отримала ширше застосування?
14. Які ви знаєте середовища для моделювання комп'ютерних мереж, які їх переваги і недоліки?

Перелік задач для самостійного опрацювання

Практичне завдання 1.

Створіть схему комп'ютерної мережі для комп'ютерного класу або невеликого офісу (10-15 робочих місць), обравши тип мережі, топологію, основні компоненти (комутатори, маршрутизатори, сервери) та обґрунтуйте свій вибір.

Практичне завдання 2.

Виконайте класифікацію 10 різних мережевих протоколів за рівнями моделі OSI, вказавши, яку функцію кожен виконує та де застосовується (наприклад, HTTP, TCP, IP, DNS, FTP).

Порівняйте дві моделі – OSI та TCP/IP у вигляді таблиці, вказавши кількість рівнів, відповідність між ними, призначення кожного рівня та приклади протоколів.

Практичне завдання 3.

Зареєструйтесь на сайті https://www.netacad.com та завантажте програму Cisco Packet Tracer за посиланням https://www.netacad.com/courses/getting-started-cisco-packet-tracer?courseLang=en-US, змоделюйте просту локальну мережу з двома комп'ютерами і одним маршрутизатором. Клікніть на кожний пристрій, прогляньте його фізичну конфігурацію (рис. 1.19).

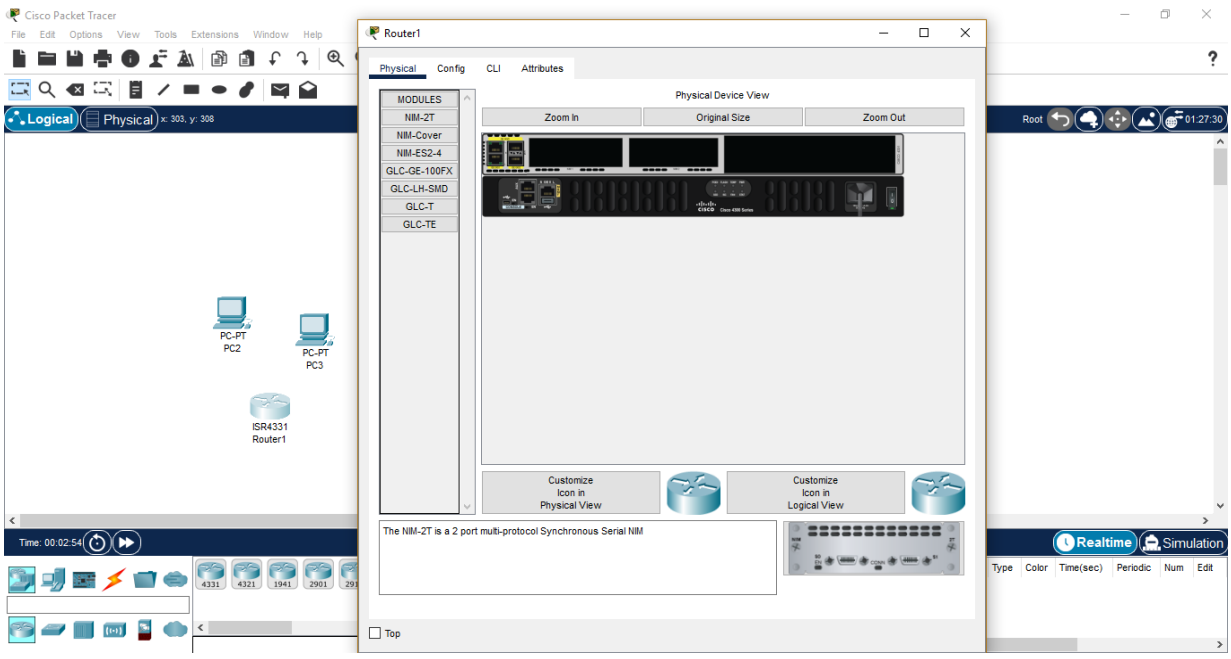


Рис. 1.19. Перегляд фізичної конфігурації маршрутизатора у середовищі моделювання Cisco Packet Tracer

Налаштуйте IP-адреси: 192.168.1.1 – для ПК1 192.168.1.2 – для ПК2, маска підмережі для кожного ПК – 255.255.255.0 (рис. 1.20). Перевірте з'єднання між комп'ютерами за допомогою команди *ping*.

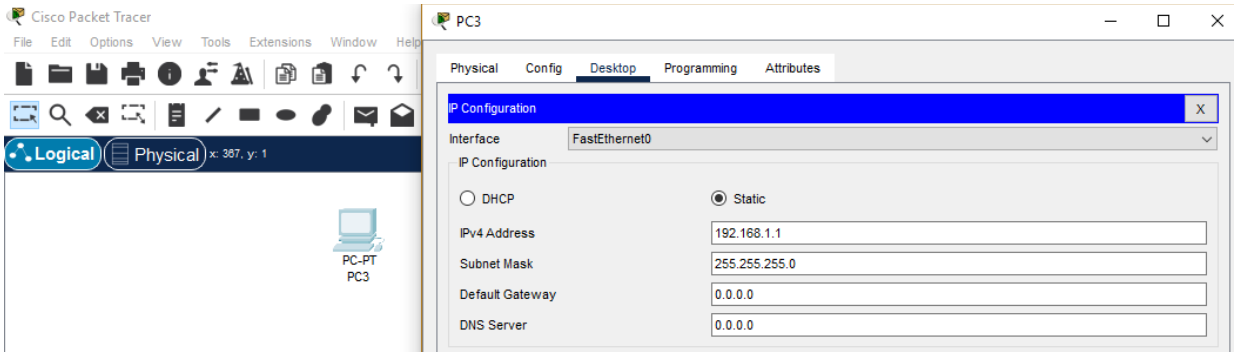


Рис. 1.20. Налаштування IP-адреси ПК у середовищі моделювання Cisco Packet Tracer

Практичне завдання 4.

Проведіть дослідження хмарних сервісів. Оберіть три популярні хмарні платформи (наприклад, Google Cloud, Microsoft Azure, Amazon Web Services) і порівняйте їх за такими критеріями: ціна, види послуг, безпека, приклади використання.

Практичне завдання 5.

Визначте фізичну та логічну топології для комп'ютерної мережі вашого навчального закладу або домашньої мережі. Побудуйте схему комп'ютерної мережі та надайте її короткий опис.

Практичне завдання 6.

Побудуйте таблицю відповідності технологій доступу до мережі Інтернет (ADSL, LTE, оптоволокно, Wi-Fi, супутниковий Інтернет) за критеріями: швидкість, вартість, стабільність, зона покриття, призначення.

Перелік використаної та рекомендованої літератури

1.1. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

1.2. Internet Engineering Task Force (IETF) RFCs [Електронний ресурс]. – Режим доступу: https://www.ietf.org

1.3. IEEE Communications Standards [Електронний ресурс]. – Режим доступу: https://standards.ieee.org

Тема 2. Мережева операційна система

Мережева операційна система (Network Operating System, NOS) – це спеціалізоване програмне забезпечення, яке керує мережею комп'ютерів та надає користувачам доступ до спільних ресурсів, таких як файли, принтери, застосунки та інші пристрої. На відміну від звичайних операційних систем, NOS зосереджується не лише на керуванні одним комп'ютером, а й на забезпеченні ефективної взаємодії між багатьма пристроями у мережі. Мережева ОС виконує функції адміністрування, захисту, маршрутизації трафіку, резервного копіювання та моніторингу мережі.

Однією з ключових функцій NOS є керування доступом до ресурсів мережі. Це включає аутентифікацію користувачів, контроль прав доступу та розмежування ролей. Наприклад, система визначає, хто може читати або змінювати певні файли, а хто має право адмініструвати мережу. Також NOS забезпечує централізоване управління, що дозволяє адміністратору змінювати конфігурації, відстежувати стан пристроїв, оновлювати програмне забезпечення або обмежувати небажану активність.

Ще однією важливою функцією є підтримка спільної роботи та надійного обміну даними між комп'ютерами. Мережева ОС організовує маршрутизацію пакетів, забезпечує стабільний зв'язок, синхронізацію даних і захист від помилок або втрати інформації. Також вона підтримує мережеві служби – сервери друку, поштові сервери, бази даних, служби каталогів (наприклад, Active Directory у Windows Server) тощо.

Сучасні приклади мережевих ОС включають Windows Server, UNIX/Linux серверні дистрибутиви (наприклад, Ubuntu Server, CentOS), Novell NetWare та інші. Вони використовуються в підприємствах, навчальних закладах і дата-центрах для побудови безпечних, масштабованих та ефективних мережевих інфраструктур.

2.1 Мережеві ОС

Операційна система (ОС) – це програмне забезпечення, яке керує апаратними ресурсами комп'ютера та забезпечує взаємодію між користувачем і обладнанням.

Класифікація мережевих операційних систем (мережевих ОС) базується на їхніх функціональних можливостях, архітектурі, сумісності з апаратним забезпеченням та сфері використання.

Мережеві ОС умовно поділяють на *серверні, вбудовані (embedded)* та *кросплатформені*.

Серверні ОС призначені для обслуговування великої кількості клієнтів у локальних та глобальних мережах.

Вбудовані ОС використовуються в спеціалізованих пристроях, таких як маршрутизатори або комутатори.

Кросплатформені ОС можуть працювати як у серверному, так і в клієнтському режимі, забезпечуючи широку гнучкість.

Однією з найвідоміших фірм, що розробляє мережеві ОС, є Microsoft, яка створила Windows Server – потужну серверну операційну систему для корпоративних мереж. Вона підтримує служби Active Directory, DNS, DHCP, файлові й принтерні сервери, віртуалізацію Hyper-V та засоби адміністрування через PowerShell. Windows Server є простим в управлінні для адміністраторів, особливо в інфраструктурах, де переважає Windows-клієнтське ПЗ.

Іншою важливою категорією є UNIX-подібні системи, серед яких Linux-дистрибутиви відіграють провідну роль. Наприклад, Ubuntu Server, Debian, CentOS (або його спадкоємець AlmaLinux/Rocky Linux), Red Hat Enterprise Linux (RHEL) використовуються як потужні мережеві ОС в дата-центрах і на вебсерверах. Ці системи підтримують широкий набір мережевих служб, є високонадійними, гнучкими та відкритими. Linux-сервери часто використовуються для розгортання вебсерверів (Apache, Nginx), поштових серверів, DNS, файлообмінних систем, VPN тощо. Окремої уваги заслуговує Novell NetWare, яка була однією з перших спеціалізованих мережевих ОС. Вона активно використовувалась у 1980–90-х роках для створення файлових і друкарських серверів. Згодом втратила популярність через розвиток Windows Server і Linux, однак історично мала значний вплив на формування уявлень про мережеві сервіси та адміністрування.

Також варто згадати Cisco IOS (Internetwork Operating System) – вбудовану мережеву ОС, яка використовується на маршрутизаторах і комутаторах Cisco. Вона не є серверною ОС у звичному розумінні, але

забезпечує керування передачею даних, маршрутизацією, безпекою та якістю обслуговування в корпоративних мережах.

Таким чином, мережеві ОС класифікуються за призначенням і середовищем використання: серверні (Windows Server, Linux), вбудовані (Cisco IOS), кросплатформені (NetWare).

Вибір тієї чи іншої системи залежить від завдань мережі, вимог до безпеки, стабільності, підтримки та бюджету організації (таблиця 2.1).

Таблиця 2.1. **Порівняння основних мережевих операційних систем з прикладами фірм-розробників і ключовими характеристиками**

Назва ОС	Фірма / Розробник	Тип ОС	Основні характеристики	Сфери використання
Windows Server	Microsoft	Серверна	Інтеграція з Active Directory, підтримка Hyper-V, GUI, централізоване адміністрування	Корпоративні мережі, офіси, державні установи
Red Hat Enterprise Linux (RHEL)	Red Hat	Серверна (UNIX-подібна)	Висока стабільність, підтримка SELinux, комерційна підтримка	Дата-центри, банки, великі компанії
Ubuntu Server	Canonical Ltd.	Серверна (UNIX-подібна)	Відкрите ПЗ, регулярні оновлення, активна спільнота, проста установка	Малі й середні бізнеси, освітні заклади
CentOS / AlmaLinux / Rocky Linux	Спільнота, спадкоємці RHEL	Серверна (UNIX-подібна)	Безкоштовна альтернатива RHEL, стабільність, зручність для хостингу	Вебсервери, VPS-хостинг, внутрішні сервери
Novell NetWare	Novell (історично)	Серверна	Спеціалізована на файлових/друкарських службах, застаріла	Архівні системи, де ще використовується
Cisco IOS	Cisco Systems	Вбудована	Підтримка мережевих протоколів (OSPF, EIGRP, BGP), CLI-інтерфейс, висока надійність	Комутатори, маршрутизатори, телекомунікації
pfSense	Netgate (FreeBSD-based)	Вбудована / серверна	Спеціалізована для мережевої безпеки, firewall, VPN, IDS/IPS	Периметрові шлюзи, офіси, навчальні установи

Операційні системи (ОС) встановлені на всіх пристроях, що підключаються до Інтернету – як кінцевих, так і мережевих.

Після увімкнення комп'ютера його операційна система завантажується з накопичувача в оперативну пам'ять (ОЗП – оперативний запам'ятовуючий пристрій). Центральним елементом ОС є ядро, яке відповідає за взаємодію з апаратними компонентами. Інша важлива частина – оболонка, що забезпечує інтерфейс між користувачем і програмами.

Оболонка може бути у вигляді командного рядка (CLI, Command Line Interface) або графічного інтерфейсу користувача (GUI, Graphical User Interface).

Використовуючи CLI, користувач вводить текстові команди через клавіатуру, а система виконує їх і виводить результат у текстовому форматі. Це надає потужний контроль, але вимагає знання команд. Натомість GUI дозволяє взаємодіяти з ОС через візуальні елементи – кнопки, вікна, піктограми – і є більш зручним для пересічного користувача. Сучасні ОС, як правило, підтримують обидва види інтерфейсів.

Більшість операційних систем, що використовуються на персональних пристроях, таких як Windows, macOS, Linux, iOS, Android, мають інтуїтивно зрозумілий графічний інтерфейс. Для конфігурації домашніх маршрутизаторів також зазвичай застосовується GUI, доступ до якого здійснюється через веббраузер. Операційна система таких пристроїв зазвичай є вбудованою мікропрограмою (firmware). У разі появи оновлень або проблем із безпекою ці мікропрограми можуть бути оновлені.

У мережевій інфраструктурі використовуються мережеві операційні системи. Наприклад, пристрої компанії Cisco працюють під керуванням Cisco IOS (Internetwork Operating System) – ОС, яка реалізує управління маршрутизаторами і комутаторами. Основний спосіб доступу до пристроїв Cisco – це інтерфейс командного рядка (CLI), який забезпечує налаштування та адміністрування мережевого обладнання.

Незважаючи на відмінності між пристроями, інтерфейс IOS для комутаторів і маршрутизаторів має багато спільного. За допомогою CLI технічні спеціалісти можуть вводити команди для конфігурації функцій пристрою, таких як маршрутизація, фільтрація трафіку, створення

VLAN тощо. Операційна система IOS може відрізнятися за функціональністю в залежності від типу пристрою і його призначення.

Файл Cisco IOS зберігається у флеш-пам’яті, яка не залежить від живлення й зберігає дані навіть після вимкнення пристрою. Це дозволяє зберігати кілька версій ОС і виконувати оновлення без заміни обладнання. Під час завантаження пристрою IOS копіюється з флеш-пам’яті в оперативну пам’ять (ОЗП), з якої працює протягом сеансу. ОЗП – енергозалежна пам’ять, тому всі дані в ній втрачаються при перезавантаженні або вимкненні пристрою.

Обсяг необхідної флеш-пам’яті та ОЗП залежить від конкретної версії IOS. Для ефективного адміністрування мережі важливо заздалегідь знати вимоги до пам’яті для кожного мережевого пристрою, особливо в разі планування оновлень чи масштабування інфраструктури.

2.2 Функції операційної системи Cisco IOS

Операційна система Cisco IOS (Internetwork Operating System) виконує низку важливих функцій, що забезпечують роботу мережевого обладнання, зокрема, маршрутизаторів і комутаторів. Це спеціалізоване програмне забезпечення, яке оптимізоване для обробки мережевого трафіку та управління апаратними ресурсами.

Однією з головних функцій IOS є управління маршрутизацією даних. Вона підтримує основні протоколи маршрутизації, такі як RIP, OSPF, EIGRP та BGP, що дозволяє автоматично знаходити найефективніші шляхи передачі даних між вузлами мережі. Це критично важливо для побудови стабільних та масштабованих мереж.

Іншою важливою функцією є забезпечення комутації трафіку на комутаторах. IOS виконує обробку кадрів Ethernet, визначення віртуальної локальної мережі VLAN, сегментацію трафіку та інші завдання, пов’язані з ефективною передачею даних у межах локальної мережі. Cisco IOS також реалізує функції безпеки, включаючи контроль доступу за допомогою списків контролю доступу ACL (Access Control Lists), підтримку шифрування VPN-з’єднань, аутентифікацію користувачів, фільтрацію трафіку та захист від атак. Це дозволяє будувати захищені мережеві середовища, як в корпоративному, так і в державному секторі. Крім того, операційна система здійснює

управління пристроєм, включаючи запуск, завантаження конфігурацій, моніторинг стану інтерфейсів, журналювання подій та підтримку віддаленого адміністрування через Telnet або SSH. Вона дозволяє мережевим адміністраторам змінювати конфігурацію пристрою в реальному часі або завантажувати готові конфігурації зі збережених файлів. Також IOS виконує керування ресурсами пам’яті і процесора, забезпечуючи ефективну багатозадачність і пріоритетність процесів. Це дозволяє одночасно обробляти великі обсяги трафіку без втрати продуктивності.

Завдяки цим функціям Cisco IOS є одним з найважливіших компонентів у сфері мережевих технологій, забезпечуючи стабільну, гнучку і безпечну роботу мережевої інфраструктури.

2.3 Метод консольного доступу

Існує кілька основних способів доступу до командного інтерфейсу CLI (Command Line Interface) мережевого пристрою Cisco, які використовуються адміністраторами для керування та налаштування обладнання.

До найбільш поширених належать:

- підключення через консольний порт,
- віддалений доступ за допомогою протоколу SSH,
- доступ через допоміжний порт AUX.

Консольний порт є спеціальним портом керування, що забезпечує позасмуговий (out-of-band) доступ до пристрою. Це означає, що з'єднання не залежить від наявності мережевих налаштувань або служб, тому його можна використовувати навіть тоді, коли мережа не працює або ще не налаштована. Основна мета консольного підключення – початкова конфігурація пристрою, технічне обслуговування або усунення неполадок. Для такого підключення використовують спеціальний консольний кабель (часто RS-232 або кабель USB-to-Serial) та програму-емулятор терміналу, наприклад, PuTTY, Tera Term або HyperTerminal.

Через консольний порт мережевий спеціаліст може переглядати процес завантаження пристрою, читати системні повідомлення, діагностувати помилки або вручну вводити конфігураційні команди.

Це зручно у випадках, коли неможливо встановити з’єднання

через мережу або при аваріях. Консольне підключення дозволяє отримати доступ до системи незалежно від її поточного стану.

Важливо враховувати, що консольний порт за замовчуванням не забезпечує шифрування даних і не використовує механізми захисту, тому слід вживати додаткових заходів безпеки. Наприклад, доступ до консольного з'єднання має бути захищений паролем, а фізичний пристрій – розміщений у захищеному приміщенні або спеціальній апаратній кімнаті, щоб уникнути несанкціонованого фізичного доступу. Якщо пароль було втрачено, існують стандартні процедури скидання пароля з використанням консольного доступу.

Таким чином, консольний доступ є ключовим інструментом у роботі мережевого адміністратора, особливо на етапах початкового налаштування або під час вирішення серйозних мережевих збоїв, коли інші способи з'єднання недоступні.

2.4 Методи доступу за допомогою Telnet та SSH

Telnet (TELecommunication NETwork) – це мережевий протокол, який дозволяє здійснювати віддалене підключення до пристроїв через текстовий інтерфейс командного рядка. Протокол працює на основі моделі «клієнт-сервер» і використовує TCP-порт 23 для встановлення з'єднання. Telnet дає змогу адміністраторам отримати доступ до інтерфейсу маршрутизаторів, комутаторів та інших мережевих пристроїв для їхньої конфігурації та моніторингу. Проте основним недоліком Telnet є відсутність шифрування даних – усі команди та паролі передаються у відкритому вигляді. Це робить Telnet непридатним для використання у відкритих або незахищених мережах, де існує ризик перехоплення трафіку.

SSH (Secure Shell) – це більш безпечна альтернатива Telnet, яка також надає віддалений доступ до командного інтерфейсу мережевих пристроїв, але забезпечує шифрування усього трафіку. SSH використовує заснований на криптографії з відкритим ключем TCP-порт 22, що забезпечує аутентифікацію користувачів та захист передаваних даних. SSH є стандартом у сучасних мережах для керування обладнанням, оскільки забезпечує конфіденційність і цілісність інформації. Крім базового керування пристроями, SSH підтримує також тунелювання трафіку, передачу файлів через SCP або

SFTP та пересилання портів, що робить його універсальним інструментом для мережевих адміністраторів.

AUX-порт (Auxiliary Port) – це фізичний резервний порт на мережевому обладнанні, зазвичай у вигляді послідовного інтерфейсу RS-232, який використовується для підключення через модем або інші термінальні пристрої. AUX забезпечує позасмуговий доступ до пристрою та дозволяє адмініструвати обладнання навіть у випадку, коли основна мережа недоступна. Це особливо корисно для аварійного відновлення або первинного налаштування. Наприклад, якщо маршрутизатор знаходиться в іншому місті та з'єднання через Інтернет втрачено, адміністратор може зателефонувати на модем, підключений до AUX-порту, і отримати доступ до CLI (Command Line Interface) пристрою. Оскільки AUX-порт зазвичай також не забезпечує шифрування, важливо застосовувати додаткові заходи безпеки, такі як автентифікація через паролі або використання VPN.

Ці три методи доступу мають свої особливості та сценарії використання. У сучасній практиці SSH є найрекомендованішим вибором для віддаленого доступу завдяки своїй безпеці, тоді як Telnet поступово виводиться з експлуатації. AUX, у свою чергу, зберігає своє значення як аварійний або резервний канал доступу до пристрою.

2.5 Програми емуляції терміналу

Для підключення до мережевого обладнання через консольний порт або за допомогою віддалених протоколів, таких як Telnet чи SSH, використовуються спеціальні програми – емулятори терміналу, які імітують роботу текстового інтерфейсу, дозволяючи користувачеві взаємодіяти з командним рядком пристрою.

Ці утиліти є незамінними інструментами для мережевих адміністраторів, оскільки забезпечують гнучкість у налаштуванні з'єднання, зручність роботи та розширені функції контролю.

До найбільш популярних програм емуляції терміналу належать:

- PuTTY,
- Tera Term,
- SecureCRT,
- HyperTerminal,

- Terminal (macOS).

Ці програми дозволяють налаштовувати параметри інтерфейсу користувача – змінювати розміри вікон, обирати шрифт і кольори, що значно покращує зручність роботи, особливо під час тривалого сеансу адміністрування.

Розглянемо особливості популярних термінальних програм.

PuTTY – це безкоштовна програма для Windows, яка підтримує кілька протоколів, зокрема SSH, Telnet і серійні з'єднання (рис. 2.1). Програма не потребує інсталяції і має гнучкі налаштування. PuTTY дозволяє зберігати профілі підключень, налаштовувати логування сесій, а також використовувати авторизацію за допомогою SSH-ключів. Через свою простоту та надійність PuTTY є однією з найпоширеніших програм для керування мережевим обладнанням.

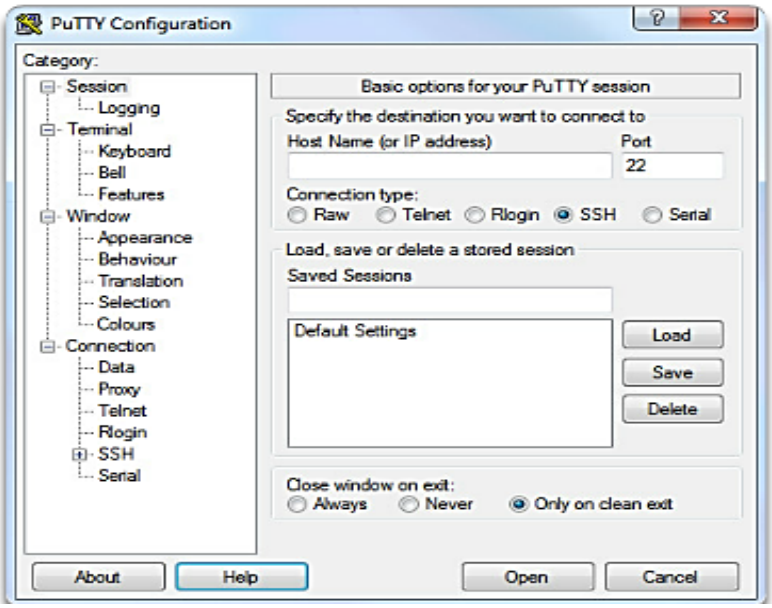


Рис. 2.1. Графічний інтерфейс користувача PuTTY

Tera Term – ще одна безкоштовна утиліта, що підтримує SSH, Telnet та COM-підключення (рис. 2.2). Вона також має функції макрозапису (автоматизації дій), що робить її зручною для повторюваних завдань. Tera Term підтримує логування, зміну кодування символів, а також підключення до модемів через COM-порт, що робить її корисною для роботи з AUX-інтерфейсами.

Рис. 2.2. Графічний інтерфейс користувача Tera Term

SecureCRT – це комерційна програма, яка надає розширені можливості управління з'єднаннями та високий рівень безпеки (рис. 2.3). Програма підтримує сучасні протоколи шифрування, дозволяє відкривати декілька вкладок одночасно, має інтеграцію зі скриптами на Python та інші функції автоматизації. SecureCRT часто використовується у великих корпоративних мережах через свою надійність та професійні інструменти керування сесіями.

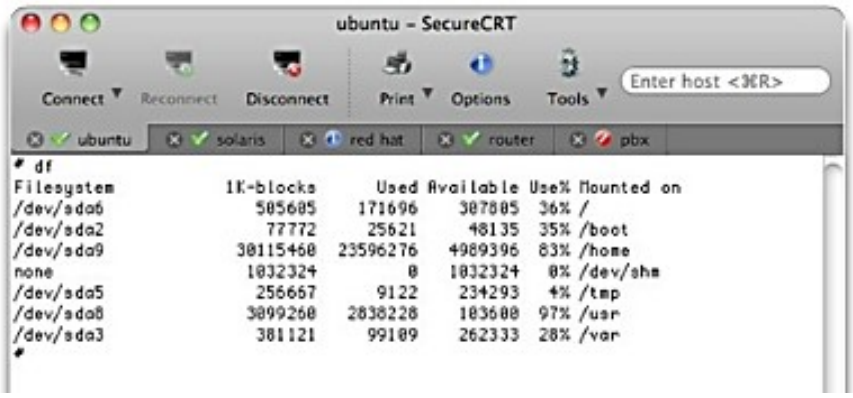


Рис. 2.3. Графічний інтерфейс користувача SecureCRT

HyperTerminal – базовий термінальний клієнт, який раніше був включений у стандартний набір програм Windows. Підтримує COM-підключення та прості Telnet-сеанси. Хоча HyperTerminal вже не поставляється в нових версіях Windows, його досі використовують для простих задач або в середовищах, де інші рішення не підходять.

Terminal (macOS) – це вбудований інструмент macOS, який підтримує SSH за замовчуванням і дозволяє працювати з командною оболонкою UNIX. Широко використовується адміністраторами, які

працюють на комп’ютерах Apple, і має інтеграцію з багатьма іншими інструментами в системі, такими як *screen*, minicom або *zsh*.

Розглянуті програми відіграють важливу роль у повсякденній роботі з мережевими пристроями, забезпечуючи надійний канал взаємодії між адміністратором і обладнанням як на локальному рівні, так і при віддаленому доступі.

2.6 Режими роботи операційної системи Cisco IOS

Після встановлення з’єднання з мережевим пристроєм спеціаліст може розпочати його налаштування. Для цього потрібно переходити між різними режимами операційної системи Cisco IOS, які мають ієрархічну структуру та однакові принципи для комутаторів і маршрутизаторів. У командному рядку ці режими згруповані за рівнем доступу: від базового до спеціалізованого. Основні з них: режим користувача, привілейований режим, режим глобального налаштування, а також додаткові – наприклад, режим налаштування інтерфейсу.

Кожен режим має свій власний формат командного рядка та набір доступних команд. Наприклад, у режимі глобального налаштування можна змінювати загальні параметри пристрою. Якщо ж потрібно налаштувати конкретний інтерфейс (наприклад, порт комутатора), слід перейти до відповідного підрежиму, де внесені зміни впливають лише на цей елемент. Щоб забезпечити безпеку, для кожного рівня доступу можна вказати окрему процедуру автентифікації.

Найважливішими режимами вважаються користувацький і привілейований. Cisco IOS розділяє доступ на два основних рівні: користувач має обмежені права, тоді як привілейований режим дозволяє повний контроль над пристроєм.

Увійшовши до CLI, користувач спочатку потрапляє в користувацький режим, який обмежується базовими командами, без можливості змінювати конфігурацію.

Цей режим позначається символом «>» (наприклад: Switch>).

Для переходу в привілейований режим використовується команда *enable*, після чого командний рядок набуває вигляду з символом «#» (Switch#), що означає розширені права. Якщо налаштовано пароль, пристрій його вимагатиме.

З привілейованого режиму можна перейти до режиму глобального налаштування, ввівши команду *configure terminal*. Після цього командний рядок зміниться на Switch(config)#. У цьому режимі можливий доступ до підрежимів конфігурації, наприклад:

- режим конфігурації інтерфейсу – для налаштування мережевих портів;
- режим конфігурації лінії – для налаштування доступу до пристрою через консоль, VTY тощо (рис. 2.4).

```
Switch# configure terminal
Enter configuration commands, one per line.
End with CNTL/Z.
Switch(config)# line vty 0 4
Switch(config-line)# interface fastethernet 0/1
Switch(config-if)# end
Switch#
```

Рис. 2.4. Приклад налаштування доступу до пристрою через VTY

Для повернення з підрежиму до попереднього використовується команда *exit*. Щоб вийти безпосередньо в привілейований режим – вводять *end* або натискають Ctrl+Z. Команда *disable* дозволяє повернутися з привілейованого режиму до користувацького (Switch# *disable* → Switch>). Командний рядок завжди містить ім'я пристрою і позначення поточного режиму, що дозволяє швидко орієнтуватися під час налаштування.

2.7 Структура команд операційної системи Cisco IOS

Усі команди в операційній системі Cisco IOS мають чітко визначену структуру – синтаксис. Кожна команда виконується лише в конкретному режимі, який відповідає її призначенню. Загальний формат команди складається з імені самої команди, а також додаткових ключових слів і параметрів, що можуть уточнювати або розширювати її функціональність. Команди задають дії, які слід виконати, тоді як ключові слова визначають, яким чином і в якому контексті ці дії мають бути здійснені.

Команди можуть мати один або декілька параметрів. На відміну від ключових слів, параметри – це не зарезервовані слова, а значення, які задає користувач. Це можуть бути IP-адреси, імена, номери інтерфейсів тощо. Синтаксис команди визначає шаблон, за яким слід вводити її в командному рядку, щоб команда була розпізнана і виконана правильно.

Наприклад, команда *ping* використовується для перевірки доступності пристрою в мережі.

Синтаксис:

Switch> ping ip-address

Приклад:

Switch> ping 10.10.10.5

У цьому випадку *ping* – це команда, а 10.10.10.5 – параметр, який задається користувачем.

Інтерфейс командного рядка IOS (CLI) має зручну функцію контекстної довідки, яка допомагає користувачам орієнтуватися в доступних командах і параметрах. Щоб скористатися цією функцією, достатньо ввести знак питання ? в потрібному місці командного рядка. Система негайно відобразить підказки без необхідності натискати Enter.

Ця допоміжна функція особливо корисна, якщо користувач не впевнений у назві команди або хоче дізнатися, чи доступна певна команда в поточному режимі роботи пристрою. Наприклад, для отримання списку всіх доступних команд у звичайному режимі введіть:

Switch> ?

Також контекстну довідку можна використовувати для перегляду команд, які починаються з певних символів. Якщо після кількох введених символів вказати ? без пробілу, IOS покаже список усіх можливих варіантів команд або ключових слів, які починаються з цієї послідовності.

Приклад:

Switch> sh?

Вивід може містити *show* – команду, яка починається на *sh*.

Ще один варіант використання знаку ? – це уточнення, які параметри або ключові слова можуть слідувати за вже введеною командою. Для цього перед ? ставиться пробіл:

Приклад:

Switch# show ?

Це відобразить перелік всіх можливих підкоманд, доступних після *show*.

CLI в IOS також підтримує гарячі клавіші й можливість скорочення команд, що значно прискорює роботу. Багато команд і ключових слів можна вводити в скороченій формі, якщо ця форма є унікальною в контексті поточного режиму. Наприклад, замість *configure* можна ввести *conf*, бо жодна інша команда не починається з *conf*. Але скорочення *con* уже не підійде, оскільки може збігатися з кількома іншими командами.

Аналогічно, ключові слова також можна скорочувати. Наприклад, команда *show interfaces* може бути введена у вигляді *show int* або навіть *sh int*.

Приклади:

Switch# show interfaces

Switch# show int

Switch# sh int

Таке скорочення команд і параметрів дозволяє швидко й ефективно працювати з CLI, особливо при регулярному налаштуванні мережевого обладнання.

2.8 Основні команди налаштування і перевірки мережевих пристроїв

Команда *show* дозволяє переглядати різну інформацію про стан пристрою. Щоб дізнатися, які саме параметри можна переглянути, достатньо ввести *show ?*. Наприклад, команда *show interfaces* надає детальні дані про всі інтерфейси маршрутизатора, включно зі статистикою їх використання. Щоб отримати інформацію про конкретний інтерфейс, використовується команда на кшталт *show interface gig0/0*.

Для перегляду збереженої конфігурації (яка зберігається у флеш-пам’яті) використовується команда *show startup-config*, а поточну конфігурацію, що міститься в оперативній пам’яті, можна побачити за допомогою *show running-config*. Якщо вивід команди не вміщується на екрані, з’являється запрошення --More--. Для продовження натискають клавішу пропуск, а для виходу – Ctrl+C.

Команда *show version* надає широкий спектр інформації, зокрема, про версію IOS, конфігураційний регістр, апаратні характеристики пристрою, обсяг пам'яті, час безперервної роботи та інше. Зокрема, параметр *Configuration register* визначає, як маршрутизатор поводиться під час запуску. Наприклад, значення 0x2102 означає, що пристрій завантажиться з пам'яті *startup-config*.

Ім'я пристрою задається командою *hostname* у глобальному режимі конфігурації. Воно повинно починатися з літери, складатися лише з літер, цифр та дефісів, не містити пробілів і бути довжиною до 63 символів. Правильне іменування допомагає в адмініструванні мережі та усуненні несправностей.

Важливо документувати місце розташування пристрою, його ідентифікатор, IP-адресу, маску підмережі та інші конфігураційні параметри. Щоб запобігти несанкціонованому доступу, необхідно розміщувати мережеве обладнання у закритих шафах або спеціалізованих приміщеннях.

Для обмеження доступу до пристрою застосовуються паролі. У Cisco IOS підтримуються такі типи паролів:

enable password – основний пароль на доступ до привілейованого режиму;

enable secret – зашифрований варіант пароля, має пріоритет над enable password;

console – пароль на доступ через консольний порт;

VTY – пароль для віддаленого доступу (SSH).

Налаштування здійснюється відповідними командами в режимі конфігурації:

line console 0

password cisco

login

Щоб усі відкриті паролі в конфігурації були зашифровані, застосовується команда:

service password-encryption

Ця команда використовує алгоритм шифрування типу 7, який забезпечує базовий рівень захисту від перегляду паролів у конфігураційному файлі.

Банери відображаються перед авторизацією користувача. Наприклад, банер типу MOTD (message of the day) використовується для попередження несанкціонованих осіб про юридичні наслідки доступу:

banner motd #Unauthorized access is prohibited.#

Поточна конфігурація зберігається в оперативній пам'яті і втрачається після перезавантаження. Щоб зберегти її у стартову (NVRAM), слід виконати команду:

copy running-config startup-config

Для завантаження конфігурації з NVRAM використовуються команда:

copy startup-config running-config

Скинути конфігурацію можна видаленням файлу:

erase startup-config

reload

Під час перезавантаження пристрій не знайде конфігураційного файлу і перейде в режим початкового налаштування.

На комутаторах інформація про віртуальні локальні мережі VLAN зберігається у файлі *vlan.dat*. Щоб повністю скинути налаштування VLAN, цей файл необхідно видалити:

delete flash:vlan.dat

reload

Контрольні запитання та задачі до теми 2

Перелік теоретичних запитань

1. Що таке мережеві операційні системи та які їх ключові характеристики?
2. Які основні функції виконує операційна система Cisco IOS?
3. У чому полягає особливість методів консольного доступу до мережевих пристроїв?
4. Які відмінності між доступом до пристрою через Telnet, SSH і AUX?
5. Які ризики безпеки пов'язані з використанням Telnet, і як їх уникнути?
6. Які переваги використання SSH у порівнянні з іншими методами доступу?
7. Які програми емуляції терміналу найчастіше

використовуються для доступу до мережевого обладнання?

8. Як налаштувати з'єднання з мережевим пристроєм за допомогою емулятора терміналу (наприклад, PuTTY чи Tera Term)?

9. Які існують режими роботи Cisco IOS і яка їхня ієрархія?

10. Як перейти з одного режиму роботи Cisco IOS в інший (наприклад, з user EXEC у privileged EXEC)?

11. З чого складається команда в Cisco IOS і як правильно її формувати?

12. Як працює контекстна довідка в Cisco IOS?

13. Яке призначення має команда *service password-encryption*?

14. Для чого використовуються банери у Cisco IOS?

15. Який синтаксис використовується для створення банера MOTD?

Перелік задач для самостійного опрацювання

Практичне завдання 1. Підключення до мережевого пристрою через консольний порт.

Крок 1. Використовуючи консольний кабель і програму-емулятор терміналу PuTTY, підключіться до маршрутизатора та виведіть базову інформацію про пристрій (команда *show version*).

Підключіть консольний кабель з порту Console на маршрутизаторі до COM-порту (або USB-COM адаптера) комп’ютера.

Відкрийте програму емуляції терміналу PuTTY.

Виберіть тип з’єднання: Serial.

У полі Serial line вкажіть номер порту (наприклад, COM3). Швидкість (baud rate) за замовчуванням: 9600.

Натисніть Open для початку сеансу.

Якщо все підключено правильно, ви побачите запрошення Router>.

Перейдіть у привілейований режим:

enable

Виведіть базову інформацію про пристрій:

show version

Ця команда покаже модель пристрою, версію Cisco IOS, кількість портів, об’єм пам’яті та іншу технічну інформацію.

Крок 2. Налаштуйте Telnet і SSH-доступ до маршрутизатора. Переконайтесь, що можна підключитися до нього з іншого пристрою.

Поясніть, у чому різниця між цими протоколами. Для налаштування SSH-доступу до маршрутизатора потрібно задати ім’я пристрою та доменне ім’я, створити обліковий запис користувача з паролем, активувати протокол SSH на віртуальних термінальних лініях (VTY), а також згенерувати RSA-ключі для SSH. Після цього на інтерфейсі призначається IP-адреса, щоб забезпечити можливість віддаленого з’єднання. Перевірити підключення можна з іншого пристрою за допомогою команди *ssh* у терміналі.

Практичне завдання 2. Використання програм-емуляторів терміналу.

Встановіть програму Tera Term або SecureCRT, підключіться до пристрою та виконайте базову конфігурацію. Зробіть знімок екрана з результатами.

Практичне завдання 3. Визначення та аналіз режимів роботи Cisco IOS.

Крок 1. Встановити та відкрити середовище моделювання Cisco Packet Tracer, створити новий проєкт і додати необхідні мережеві пристрої (маршрутизатори, комутатори, ПК), з'єднавши їх відповідними кабелями. Увімкнути пристрої, відкрити CLI (Command Line Interface) кожного з них та підготувати емуляцію термінального доступу, що дозволить виконувати налаштування й перевірку конфігурацій

Крок 2. Увійдіть у різні режими Cisco IOS (user EXEC, privileged EXEC, global configuration) та задокументуйте доступні команди у кожному з них.

Крок 3. Для створення базової конфігурації мережевого пристрою налаштуйте ім’я пристрою, банер вітання, пароль доступу та IP-адресу на інтерфейсі. Збережіть конфігурацію.

Приклад базової конфігурації мережевого пристрою (наприклад, маршрутизатора або комутатора Cisco) в Cisco IOS через CLI:

```
enable                          ! Перехід у привілейований режим
configure terminal              ! Перехід у режим глобальної конфігурації
hostname Router1                ! Налаштування імені пристрою
banner motd #
Увага! Несанкціонований доступ заборонено!
#                               ! Банер вітання
enable secret cisco123          ! Пароль доступу до привілейованого режиму
```

```
line console 0              ! Вхід у налаштування консольного доступу
 password conpass           ! Пароль для консолі
 login                      ! Активація пароля
 exit                       ! Вихід з режиму налаштування лінії
line vty 0 4                ! Налаштування доступу SSH
 password vtypass           ! Пароль для віртуальних ліній
 login                      ! Активація пароля
 exit                       ! Вихід
interface GigabitEthernet0/0   ! Перехід до налаштування інтерфейсу
 ip address 192.168.1.1 255.255.255.0  ! IP-адреса та маска
 no shutdown                ! Увімкнення інтерфейсу
 exit
exit                        ! Вихід до привілейованого режиму
write memory                ! Збереження конфігурації (альтернатива: copy
running-config startup-config)
```

Крок 4. Для перевірки мережевих параметрів використайте команди *show ip interface brief, show running-config, ping і traceroute*, щоб перевірити налаштування та з'єднання між пристроями.

Крок 5. За допомогою контекстної довідки (?) знайдіть команди, які можна використовувати в режимі конфігурації інтерфейсу.

Крок 6. Для збереження та відновлення конфігурації виконайте команду *copy running-config startup-config*, перезавантажте пристрій та перевірте, чи збереглися зміни.

Перелік використаної та рекомендованої літератури

2.1. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

2.2. SSH: The Secure Shell [Електронний ресурс]. – Режим доступу: https://www.ssh.com/academy/ssh

2.3. Configuration Fundamentals Configuration Guide [Електронний ресурс]. – Режим доступу: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/fundamentals/configuration/xe-16/fundamentals-xe-16-book.html

2.4. PuTTY [Електронний ресурс]. – Режим доступу: https://www.putty.org/

2.5. PuTTY [Електронний ресурс]. – Режим доступу: https://teratermproject.github.io/index-en.html

Тема 3. Фізичний та канальний рівень

Фізичний рівень (Physical Layer) є найнижчим рівнем у моделі OSI (Open Systems Interconnection) і відповідає за передачу бітів через фізичне середовище. Це той рівень, який забезпечує фізичне з'єднання між пристроями, передаючи дані у вигляді електричних сигналів, світлових імпульсів або радіохвиль.

Фізичний рівень визначає такі параметри, як тип кабелю (мідний, оптоволоконний), розміри пакетів, швидкість передачі даних, а також механізми, які регулюють електричні характеристики сигналу, такі як амплітуда, частота та модуляція. Оскільки фізичний рівень працює з апаратними засобами, його завдання полягає у забезпеченні стабільної та безпомилкової передачі бітів без втручання в зміст самих даних.

Канальний рівень (Data Link Layer) є другим рівнем моделі OSI і забезпечує коректну передачу даних між двома пристроями, що безпосередньо з'єднані, шляхом організації і контролю за потоком даних.

Канальний рівень відповідає за подолання проблем, пов'язаних з помилками, які можуть виникнути на фізичному рівні. Це здійснюється через додавання контрольних сум або кодів для перевірки цілісності переданих даних, а також через механізми виявлення та корекції помилок. Окрім цього, канальний рівень здійснює фреймування – процес розбиття даних на блоки (фрейми) для зручності передачі та обробки.

У контексті локальних мереж канальний рівень управляє доступом до фізичного середовища передачі через протоколи, такі як Ethernet, Wi-Fi або PPP (Point-to-Point Protocol).

Обидва рівні, фізичний і канальний, тісно пов'язані між собою, фізичний рівень забезпечує основу для передачі сигналів, а канальний рівень гарантує, що ці сигнали будуть коректно інтерпретовані, надійно передані та не будуть пошкоджені в процесі передачі даних. Вони є основою для функціонування будь-якої мережі, від локальних до глобальних, і відповідають за низку ключових процесів, таких як контроль доступу до середовища, зниження помилок і оптимізацію використання фізичних ресурсів.

3.1. Правила передавання даних у мережі

Протоколи, які використовуються в мережах зв'язку, і ті, що управляють комунікаційними процесами, мають спільні основні характеристики, що дозволяють успішно здійснювати передачу повідомлень через різні канали. Окрім визначення адрес джерела і призначення, мережеві протоколи також визначають методи передачі повідомлення від однієї точки до іншої. Основні етапи процесу передачі повідомлення включають кілька важливих аспектів:

- кодування повідомлення;
- форматування та інкапсуляція повідомлення;
- розмір повідомлення;
- часові параметри повідомлення;
- параметри доставки повідомлення.

Кодування повідомлення є першим етапом процесу передачі. Це процес перетворення даних у форму, яку можна передавати через мережу. Кодування залежить від типу використовуваного каналу зв’язку, і перед тим, як передати повідомлення, вузол-відправник перетворює його у двійкові дані (біти). Залежно від мережі ці біти можуть бути представлені через різні фізичні носії: електричні імпульси, світлові сигнали або радіохвилі. Коли повідомлення досягає приймаючого вузла, відбувається декодування – процес перетворення отриманих сигналів назад у форму, зрозумілу кінцевому користувачеві або програмному забезпеченню.

Форматування та інкапсуляція повідомлення – це ще один важливий етап. Під час передачі дані повинні бути оформлені у певному форматі, відповідно до специфікацій каналу зв'язку. Це схоже на процес укладання листа в конверт. Повідомлення, перед тим як потрапити в мережу, обробляються в кадри, кожен з яких містить заголовки з адресами відправника і одержувача, а також додаткову інформацію, необхідну для коректної доставки.

Форматування повідомлення залежить від типу даних і протоколу каналу, і правильне форматування є критично важливим для забезпечення успішної доставки повідомлення.

Розмір повідомлення також є важливим аспектом, оскільки при передачі великих обсягів даних повідомлення поділяються на менші

частини, які передаються окремо. Кожна частина повідомлення має бути в межах певного розміру, що регулюється характеристиками каналу. Процес поділу на менші блоки дозволяє ефективно використовувати мережу і називається *сегментацією*. Якщо одне повідомлення занадто велике для передачі через конкретний канал, воно розбивається на фрагменти, кожен з яких обробляється окремо. Вузол-призначення збирає ці частини разом, щоб відновити початкове повідомлення.

Часові параметри відіграють важливу роль у процесі передачі даних. Час визначає, коли саме передавати дані, як швидко передавати повідомлення і скільки часу чекати на відповідь. Це важливо для уникнення колізій, коли два пристрої намагаються передати дані одночасно, що може призвести до втрати інформації. Мережеві протоколи визначають, коли кожен вузол може почати передачу повідомлення, як реагувати на помилки, а також коли чекати на відповідь від приймаючої сторони. Важливим аспектом є тайм-аут – якщо вузол не отримав відповідь протягом певного часу, він може вважати, що зв’язок зупинено, і вжити відповідних заходів.

Параметри доставки повідомлень визначають, як і кому повинні доставлятися повідомлення. Розглянемо методи доставки повідомлень.

Одноадресна доставка (unicast) – коли повідомлення відправляється від одного вузла до одного призначеного вузла.

Багатоадресна доставка (multicast) – коли повідомлення розсилається одночасно групі вузлів.

Широкомовна доставка (broadcast) – коли повідомлення надсилається усім вузлам у мережі.

Для кожного типу доставки існують певні протоколи, які регулюють правила отримання та підтвердження доставки.

Сегментація та мультиплексування повідомлень є важливими механізмами для підвищення ефективності мережевої комунікації.

Сегментація дозволяє розділяти великі повідомлення на частини (сегменти), що забезпечує зручність обміну даними та підвищує надійність, оскільки окремі частини повідомлення можуть йти різними маршрутами в мережі.

Якщо одна з частин повідомлення втрачається або пошкоджується, лише ця частина потребує повторної передачі, а не все повідомлення.

Мультиплексування дає змогу одночасно обробляти кілька незалежних потоків даних, що дозволяє підвищити ефективність використання мережі.

Проте цей процес має свої недоліки: на кожному етапі інкапсуляції додавання нових заголовків і розбиття повідомлень на частини додає складності в обробку. Кожен сегмент потребує окремої обробки на кожному рівні стеку протоколів, що може збільшити час і ресурси для доставки та відновлення повідомлення.

Процес передачі повідомлень включає інкапсуляцію та деінкапсуляцію даних, що означає додавання або видалення заголовків протоколів на кожному рівні стека протоколів.

Інкапсуляція відбувається від верхнього до нижнього рівня, де кожен рівень додає свою інформацію до даних перед передачею.

На рис. 3.1 представлено процес передачі HTTP-даних через комп'ютерну мережу, ілюстрований моделюванням з використанням моделі TCP/IP. Користувач формує HTTP-дані, які проходять через кілька рівнів інкапсуляції: спочатку додається заголовок транспортного рівня (наприклад, TCP), потім заголовок мережевого рівня (наприклад, IP), а згодом – заголовок канального рівня (кадровий заголовок).

У результаті дані передаються у вигляді двійкового коду через мережевий адаптер на сервер. Цей процес демонструє, як дані упаковуються в кадри для передачі по мережі та поступово доповнюються заголовками нижчих рівнів.

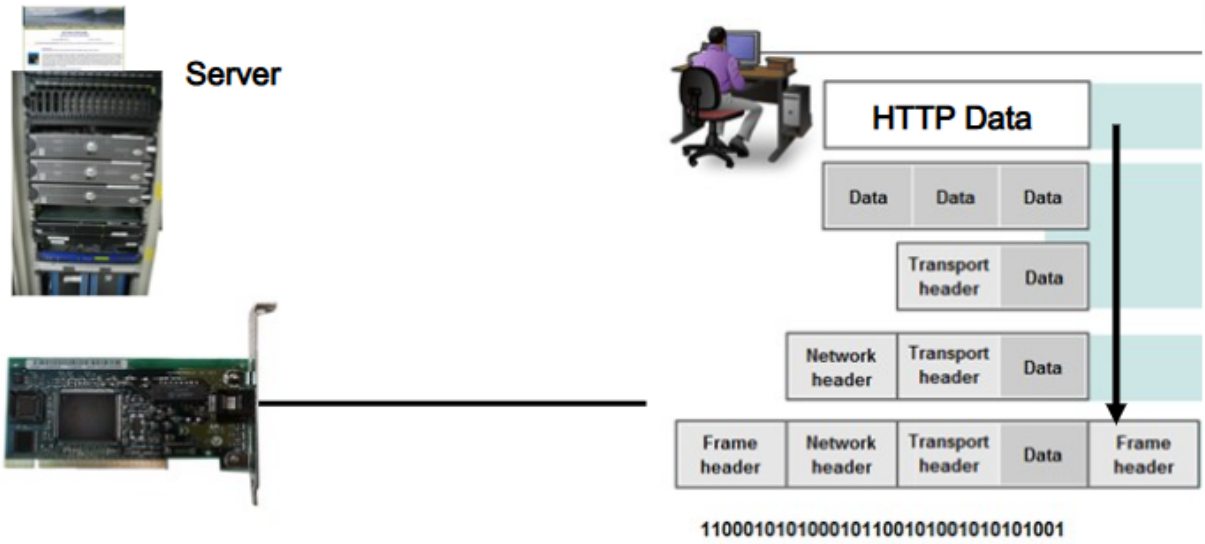


Рис. 3.1. Процес інкапсуляції повідомлень

Деінкапсуляція відбувається на приймальному боці, коли дані проходять зворотний шлях і знову інтерпретуються на кожному рівні для правильного відновлення початкового повідомлення (рис. 3.2).

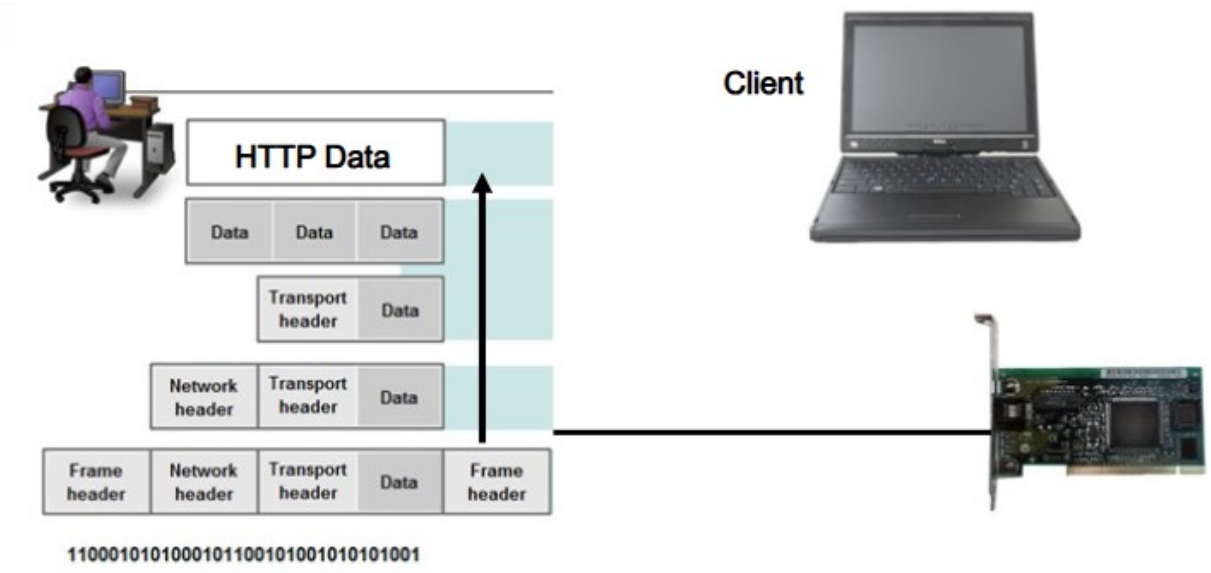


Рис. 3.2. Процес деінкапсуляції повідомлень

Мережеві та канальні рівні відповідають за доставку даних від відправника до отримувача, контролюючи адресацію та коректну передачу інформації між пристроями.

3.2. Логічна та фізична адреса пристроїв у мережі

Фізична адреса – це адресація на канальному рівні (рівень 2 моделі OSI), яка використовується для безпосередньої доставки кадрів даних між пристроями в межах однієї локальної мережі.

Найвідомішим типом фізичної адреси є MAC-адреса (Media Access Control), яка є унікальною для кожного мережевого інтерфейсу. Вона зазвичай складається з 48 бітів і присвоюється апаратно виробником мережевої карти. Фізичні адреси незмінні в межах одного пристрою і не змінюються при переміщенні між мережами, тому вони забезпечують локальну ідентифікацію пристроїв.

Логічна адреса – це адресація, яка використовується на мережевому рівні (рівень 3 моделі OSI) для ідентифікації пристроїв у межах великої мережі або мережі Інтернет. Найпоширенішим прикладом логічної адреси є IP-адреса, яка складається з двох частин:

мережевого префіксу, який визначає конкретну мережу, та ідентифікатора вузла, що вказує на конкретний пристрій у цій мережі. Логічні адреси можуть змінюватися (наприклад, при переміщенні пристрою між мережами) і використовуються для маршрутизації даних через декілька мереж, тобто для міжмережевої передачі.

Логічна адреса, яка використовується на мережевому рівні (рівень 3 моделі OSI), призначена для забезпечення доставки IP-пакета між різними пристроями в мережі. Ця IP-адреса складається з двох основних частин: мережевого префіксу та вузлового ідентифікатора.

Мережевий префікс допомагає маршрутизаторам визначити, в яку мережу потрібно доставити пакет, а вузлова частина – дозволяє фінальному маршрутизатору знайти конкретний пристрій-отримувач у тій мережі.

Кожен IP-пакет несе дві адреси: адресу джерела (тобто пристрою, що надсилає дані) та адресу призначення (тобто пристрою, що приймає дані). Саме призначення використовується маршрутизаторами на шляху для коректної доставки.

На відміну від логічної IP-адреси, фізична адреса (рівень 2), або MAC-адреса, потрібна для пересилки кадрів між мережевими інтерфейсами в межах однієї локальної мережі. Перш ніж IP-пакет можна буде передати фізичним середовищем (дротовим або бездротовим), його необхідно "запакувати" в кадр, що відповідає протоколу канального рівня – наприклад, Ethernet або Wi-Fi. Кожен тип мережі має власні особливості інкапсуляції та фізичної адресації.

На рис. 3.3 у процесі передачі IP-пакет інкапсулюється в кадр, який включає як MAC-адресу джерела, так і MAC-адресу призначення. Наприклад, якщо ПК1 з IP-адресою 192.168.1.110 передає дані до FTP-сервера з IP-адресою 192.168.1.9, MAC-адреса Ethernet-інтерфейсу ПК1 буде 0A-AA-AA-AA-AA-AA, а MAC-адреса сервера – 0C-CC-CC-CC-CC-CC. Ці адреси додаються до Ethernet-кадру, який потім передається безпосередньо, якщо обидва пристрої знаходяться в одній мережі.

Для надсилання даних всередині локальної мережі вузол повинен знати як логічну (IP), так і фізичну (MAC) адресу вузла-призначення. IP-адресу він може дізнатися вручну (наприклад, від користувача) або через DNS (Domain Name System).

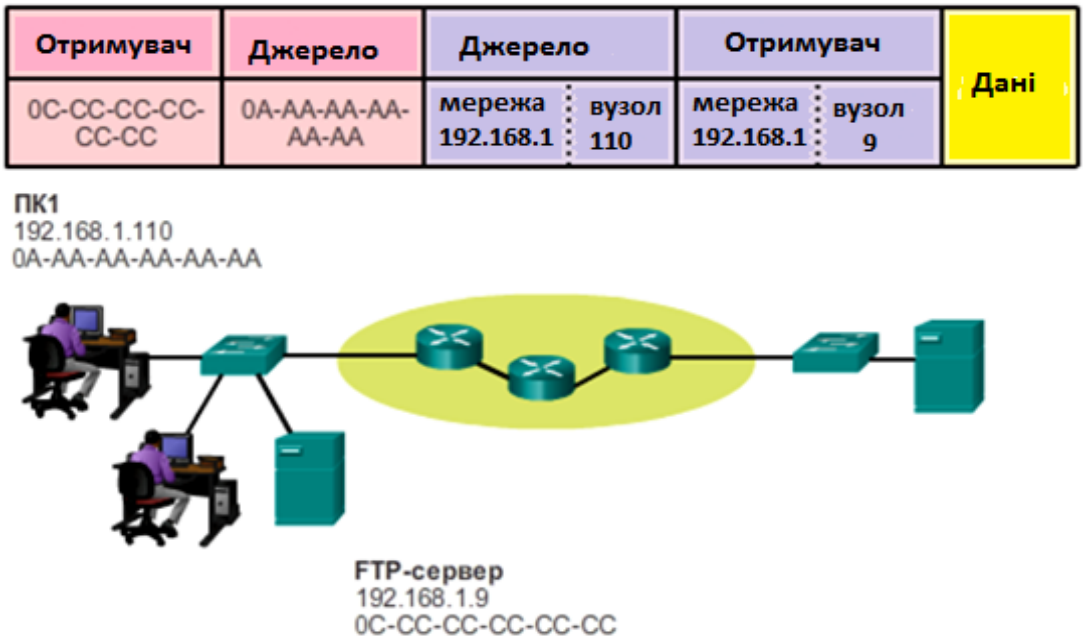


Рис. 3.3. Процес обміну даними в одній мережі ПК 1 з FTP-сервером

Для визначення MAC-адреси використовується спеціальний мережевий протокол – ARP (Address Resolution Protocol).

ARP працює шляхом надсилання широкомовного запиту, в якому зазначена IP-адреса призначення. Усі пристрої мережі отримують цей запит, і лише той, у кого така IP-адреса, повертає відповідь із власною MAC-адресою. Таким чином, вузол-джерело дізнається необхідну фізичну адресу для формування кадру.

Однак якщо вузол призначення знаходиться в іншій мережі, пряма доставки неможлива. У такому випадку використовується маршрутизатор, який виконує роль шлюзу за замовчуванням (default gateway). Це інтерфейс маршрутизатора в тій же мережі, що й вузол-джерело. Усі IP-пакети, призначені для віддалених мереж, спочатку надсилаються на MAC-адресу цього маршрутизатора.

Якщо шлюз за замовчуванням не вказано або вказано неправильно, такі повідомлення не зможуть бути доставлені за межі локальної мережі.

3.3. Технології та засоби передачі даних фізичного рівня

Мережева інтерфейсна плата (NIC) – це спеціалізований апаратний компонент, який дозволяє пристрою підключатися до комп'ютерної мережі.

Існують два основних типи NIC: для дротового з’єднання (Ethernet), та для безпровідного з’єднання – через Wi-Fi.

Деякі пристрої, зокрема, настільні комп’ютери або мережеві принтери, можуть бути оснащені лише Ethernet-інтерфейсом і потребують фізичного підключення через кабель.

Інші пристрої, такі як смартфони або планшети, зазвичай мають лише безпровідний мережевий інтерфейс, що дозволяє їм підключатися до мережі без кабелів.

Продуктивність підключення залежить не лише від типу мережевої інтерфейсної плати, але й від фізичних умов з'єднання. Наприклад, безпровідне з’єднання стає менш ефективним зі збільшенням відстані між пристроєм і точкою доступу.

Чим далі розміщений пристрій, тим слабший сигнал він отримує, що може призвести до зниження швидкості передавання даних або навіть до втрати зв'язку. Також, при одночасному використанні великою кількістю пристроїв одна точка доступу може перевантажуватись, що ще більше впливає на швидкість роботи мережі.

На відміну від безпровідних пристроїв, дротові пристрої мають виділене фізичне з’єднання у вигляді окремого кабелю Ethernet, що забезпечує стабільну швидкість і мінімальну затримку. Такий тип підключення ідеально підходить для застосунків із високими вимогами до пропускної здатності, таких як потокове відео, відеоконференції або онлайн-ігри.

Фізичний рівень моделі OSI відповідає за передачу бітів по мережевому середовищу. Цей рівень приймає кадри від канального рівня, кодує їх у форму електричних, оптичних або радіосигналів та передає по фізичному середовищу, такому як кабелі чи ефір.

На приймальному боці фізичний рівень зчитує сигнали, відтворює початкову послідовність бітів і передає їх далі – до канального рівня.

Процес передачі даних від джерела до призначення проходить кілька етапів.

Спочатку транспортний рівень розбиває дані користувача на сегменти. Далі мережевий рівень «упаковує» ці сегменти в IP-пакети. Потім канальний рівень інкапсулює пакети в кадри. Фізичний рівень перетворює кадри на сигнали – електричні, світлові або радіохвильові – залежно від середовища передачі (рис. 3.4).

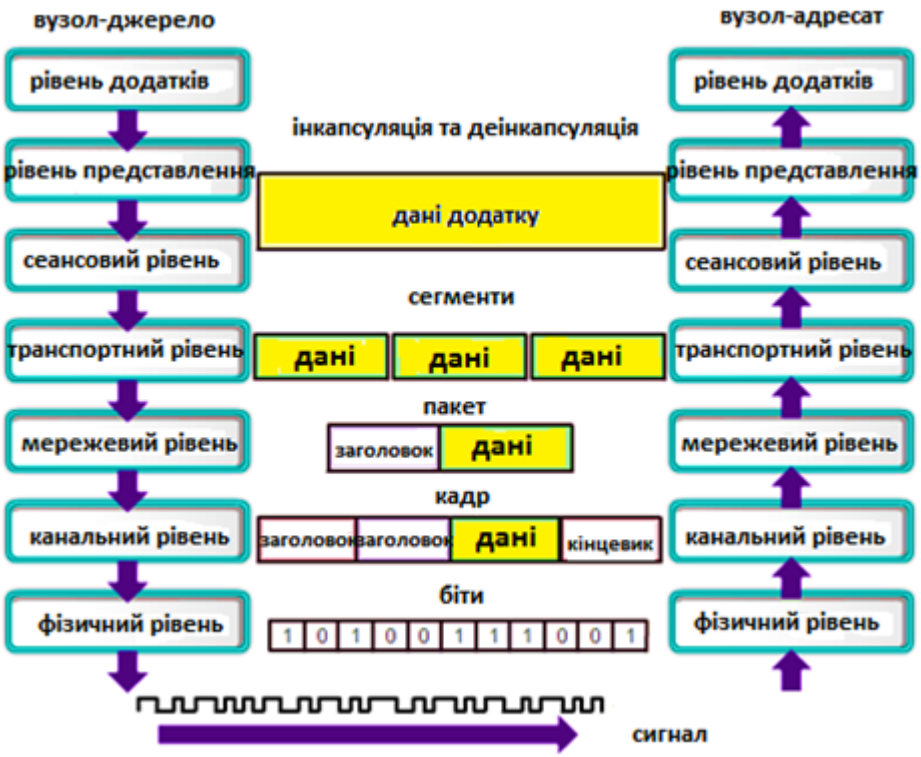


Рис. 3.4. Процес перетворення даних від джерела до адресата

Існує три основні типи фізичного середовища для передачі даних (рис. 3.5).

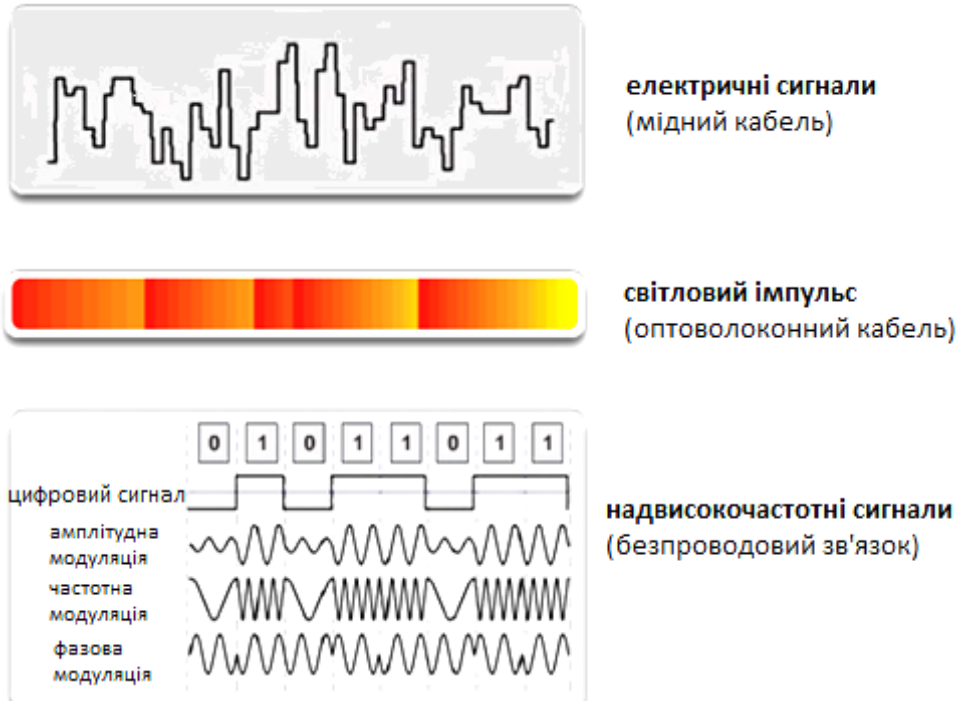


Рис. 3.5. Приклади середовищ передачі даних

Мідні кабелі – інформація передається як електричні імпульси.

Оптоволоконні кабелі – дані передаються через світлові сигнали.

Безпровідні канали – використовують радіохвилі, наприклад, Wi-Fi або Bluetooth.

Фізичний рівень адаптує передані сигнали до характеристик кожного з цих середовищ, дотримуючись стандартів, визначених організаціями зі стандартизації.

Фізичні компоненти мережі включають мережеві адаптери, з'єднувальні пристрої, кабелі та роз'єми, які фізично забезпечують обмін сигналами між пристроями. Усі ці компоненти мають відповідати міжнародним стандартам, щоб забезпечити сумісність і ефективну роботу мережі.

Кодування сигналів – це метод перетворення потоку бітів у спеціальні шаблони, які можуть бути правильно розпізнані обома сторонами – відправником і одержувачем. У мережевій комунікації кодування визначає, як змінюються електричні або інші сигнали для представлення 0 і 1.

Пропускна спроможність (bandwidth) – це теоретично можлива швидкість передачі даних через певне середовище, яка вимірюється у кілобітах або мегабітах за секунду. Однак реальна швидкість, з якою дані передаються по мережі, називається *продуктивністю (throughput)*. Вона зазвичай нижча за пропускну спроможність і залежить від багатьох факторів, таких як кількість активних пристроїв, мережевий трафік або затримки.

Затримка (latency) – це загальний час, який потрібно для передачі даних від одного вузла до іншого. Навіть якщо більшість елементів мережі підтримують високу швидкість, найповільніша ділянка маршруту стає «вузьким місцем», яке обмежує загальну продуктивність усієї мережі.

3.4. Характеристики мідних кабелів

У комп’ютерних мережах часто використовуються мідні кабелі, оскільки вони економічно вигідні, прості у монтажі та мають низький електричний опір. Проте їхня ефективність обмежується дальністю передавання сигналу та впливом перешкод. У таких кабелях інформація передається у вигляді електричних імпульсів. Мережева карта пристрою

повинна точно зчитувати сигнал, щоб розшифрувати його згідно з початковими даними. Чим довший шлях сигналу, тим сильніше він слабшає – це явище називається *затуханням*. Через це мідне середовище передавання даних підпорядковується суворим стандартам щодо максимальної довжини кабелів.

На якість сигналу впливають і зовнішні фактори. Зокрема, електромагнітні завади (ЕМЗ) або радіочастотні перешкоди (РЧП), які можуть деформувати або пошкодити передані сигнали. Такі завади можуть виникати від електронного обладнання, наприклад, флуоресцентного освітлення або електродвигунів (рис. 3.6).

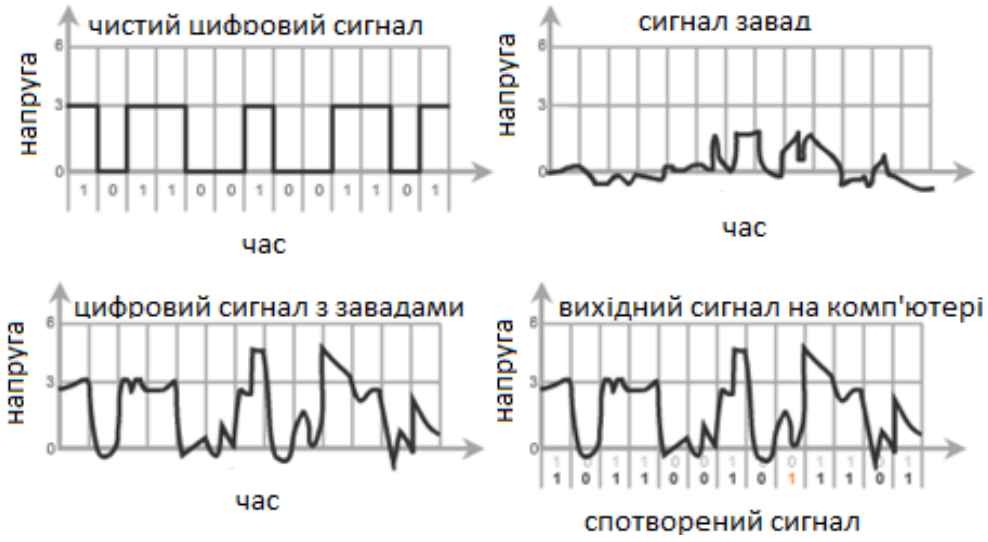


Рис. 3.6. Приклади впливу завад на цифровий сигнал

Ще одним типом перешкод є перехресні перешкоди – вони з’являються, коли сигнал з одного кабелю впливає на сигнал у сусідньому, що може викликати, наприклад, підслуховування сторонньої розмови. Це пов’язано з утворенням магнітного поля навколо провідника під час проходження струму.

Щоб зменшити вплив таких завад, деякі види мідних кабелів мають металеве екранування і вимагають правильного заземлення. Для зменшення перехресних перешкод часто використовуються кабелі з перекрученими між собою парами провідників, по яких струм тече в протилежних напрямках – це конструкція називається *витою парою*.

Захист мідних кабелів від електричних завад забезпечується шляхом:

- правильного вибору типу і категорії кабелю відповідно до умов середовища;
- правильного проєктування кабельної інфраструктури з урахуванням джерел потенційних перешкод;
- застосування спеціалізованих методів укладання кабелів.

У мережевій інфраструктурі використовують три основні типи мідних кабелів:

- неекранована вита пара (UTP),
- екранована вита пара (STP),
- коаксіальний кабель.

Ці типи кабелів застосовуються для з'єднання пристроїв у локальних мережах, зокрема, між комп'ютерами, комутаторами, маршрутизаторами та точками доступу. Кожен із них має свої вимоги, прописані в стандартах фізичного рівня (рис. 3.7).

Unshielded Twisted-Pair (UTP) Cable

Shielded Twisted-Pair (STP) Cable

Coaxial Cable

Рис. 3.7. Основні типи мідних кабелів

UTP (Unshielded Twisted Pair) є найбільш розповсюдженим видом мережевого кабелю. Він складається з чотирьох перекручених пар провідників, які вкриті пластиковою оболонкою. Така конструкція зменшує вплив перешкод. Для підключення зазвичай використовуються роз'єми типу RJ-45.

STP (Shielded Twisted Pair) кабелі забезпечують кращий захист від перешкод, ніж UTP, але є дорожчими і складнішими в монтажі. Вони також використовують роз'єми RJ-45, але потребують спеціального захищеного варіанту. У STP поєднується перекручування провідників і екранування для зменшення впливу електромагнітних і радіочастотних завад.

Коаксіальні кабелі мають два провідники, розташовані по одній осі. Вони складаються з центрального мідного провідника, ізоляції, екранувального шару з металевого обплетення або фольги і зовнішньої оболонки. Така будова забезпечує захист від зовнішніх перешкод.

Усі три типи мідних кабелів чутливі до займання і ураження електричним струмом. Тому під час прокладання кабельних мереж у будівлях слід враховувати вимоги до пожежної безпеки.

Електричні властивості мідних кабелів регламентуються Інститутом інженерів з електротехніки та електроніки IEEE.

Кабелі UTP класифікуються за категоріями залежно від швидкості передачі даних. Кабель категорії 5 (Cat5) найчастіше застосовується в мережах FastEthernet (100BASE-TX), а для сучасних гігабітних мереж використовуються щонайменше Cat5e або Cat6, останній рекомендується для створення нових мережевих інфраструктур.

3.5. Особливості використання оптоволоконних кабелів

Оптоволоконні кабелі здобули широку популярність завдяки своїй здатності ефективно з'єднувати мережеві компоненти. Вони забезпечують передачу даних на великі відстані з високою швидкістю та пропускною здатністю, що перевищує можливості традиційних засобів передачі.

Оптичне волокно – це надзвичайно тонкий, гнучкий і прозорий кабель, виготовлений з високоякісного скла (кварцу), товщиною з людське волосся. У такому кабелі інформація передається у формі світлових імпульсів. Оптичне волокно працює як провідник світла,

забезпечуючи його проходження з мінімальними втратами від одного кінця кабелю до іншого.

Одномодовий (single-mode) оптоволоконний кабель – цей тип волокна має дуже тонку скляну серцевину діаметром приблизно 9 мікрон, оточену оболонкою зі скла діаметром 125 мікрон і захисним полімерним покриттям (рис. 3.8). Світло проходить через таке волокно по одному прямому шляху, що зменшує втрати сигналу та дозволяє передавати дані на великі відстані з високою швидкістю і точністю. Одномодові оптичні волокна найчастіше використовуються в телекомунікаційних мережах і магістральних з'єднаннях, де потрібна максимальна продуктивність.

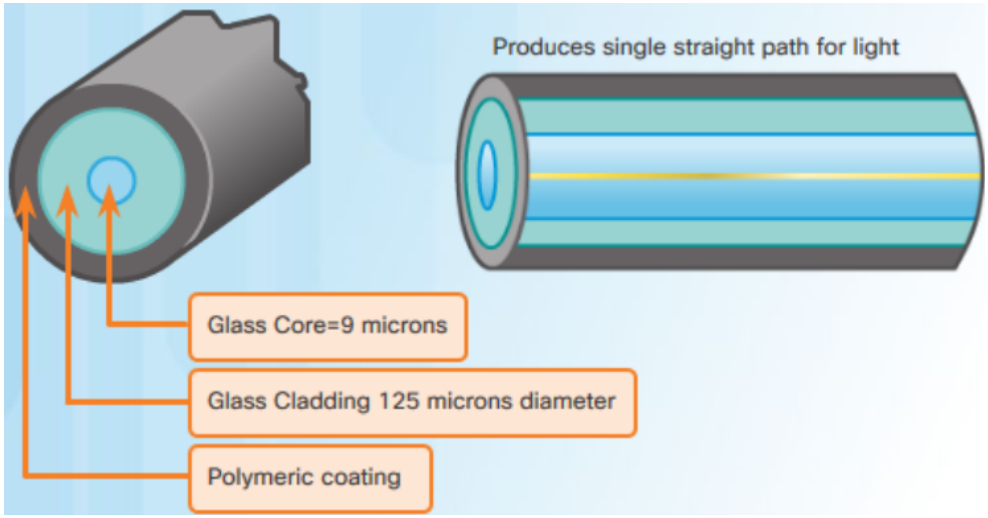


Рис. 3.8. Одномодовий (single-mode) оптоволоконний кабель

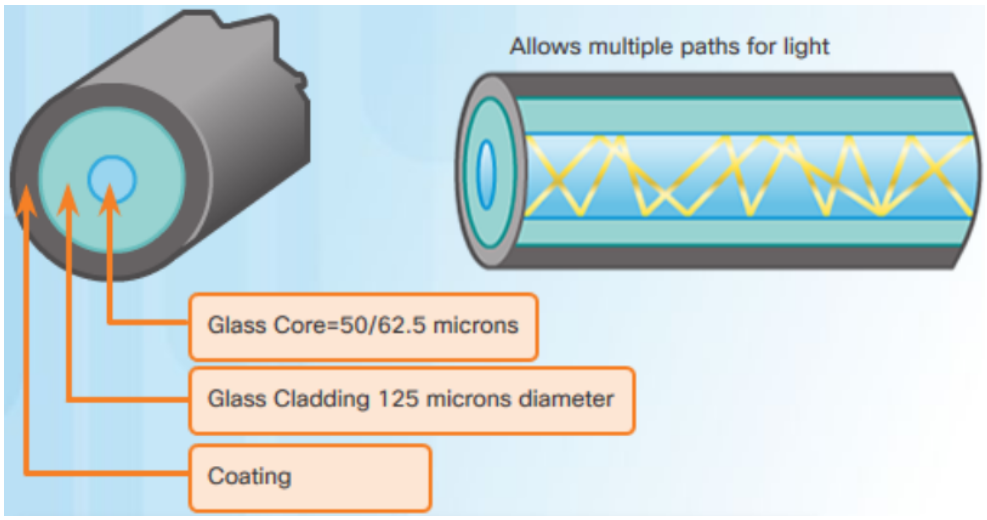


Рис. 3.9. Багатомодовий (multimode) оптоволоконний кабель

Багатомодовий (multimode) оптоволоконний кабель – цей тип оптоволокна має значно більшу серцевину діаметром 50 або 62,5 мікрон, що також оточена оболонкою 125 мікрон і захисним шаром (рис. 3.9). Завдяки ширшій серцевині світлові імпульси можуть поширюватися по кількох траєкторіях, відбиваючись від стінок оболонки. Це дозволяє використовувати дешевші джерела світла (наприклад, світлодіоди), але водночас обмежує довжину передавання через виникнення інтерференцій та загасання сигналу.

Багатомодові кабелі широко застосовуються в локальних мережах, де відстані між пристроями є відносно невеликими.

На відміну від мідних кабелів, оптичне волокно забезпечує нижчий рівень загасання сигналу та має повний імунітет до електромагнітних і радіочастотних завад.

Сьогодні оптоволоконні кабелі застосовуються в чотирьох основних напрямках.

1. *Корпоративні мережі* – оптоволоконні кабелі використовуються як основа магістральної інфраструктури для об'єднання ключових мережевих пристроїв.
2. *FTTH (Fiber to the Home)* – оптоволоконні кабелі забезпечують високошвидкісний інтернет та мультимедійні послуги для домашніх користувачів і малих офісів.
3. *Телекомунікаційні мережі* – оптоволоконні кабелі служать основою для міжміського та міжнародного зв’язку, підтримуючи швидкості до 10 Гбіт/с на відстанях до кількох тисяч кілометрів.
4. *Підводні кабелі* – спеціально створені для прокладення на дні океанів, забезпечуючи стабільне високошвидкісне з'єднання на великі відстані в складних умовах.

Незважаючи на малу товщину, оптоволоконний кабель має складну будову:

- *серцевина* – центральна частина з прозорого скла, якою проходить світло;
- *оболонка* – скляний шар навколо серцевини, що відбиває світло назад у серцевину завдяки повному внутрішньому відбиттю;
- *зовнішнє покриття* – зазвичай з ПВХ, забезпечує механічний захист;
- *додаткові елементи* – буферне покриття та зміцнюючі шари

захищають волокно від вологи, подряпин та пошкоджень.

Оптичне волокно проходить строгі тести на міцність – воно витримує тиск понад 20 000 кг/см², що робить його надійним навіть у важких умовах експлуатації.

Світлові сигнали у волокні створюються за допомогою:

- лазерів,
- світлодіодів,
- фотодетекторів, які приймають світло та перетворюють його в електричні сигнали, що далі перетворюються у мережеві кадри.

Для простої перевірки цілісності волокна можна використати яскраве джерело світла: якщо світло видно з іншого кінця – передача можлива. Для точнішої діагностики використовується оптичний тестер або рефлектометр OTDR, який дозволяє виявити пошкодження на певній відстані, вимірюючи відбиття і розсіювання світлового імпульсу.

Оптоволоконні кабелі мають низку переваг у порівнянні з мідними, вони:

- не чутливі до електромагнітних завад;
- не проводять електричний струм, отже, не потребують заземлення;
- мають мінімальні втрати сигналу, що дозволяє прокладати їх на великі відстані без підсилення;
- забезпечують високу швидкість і надійність передачі даних.

У сучасних корпоративних інфраструктурах саме оптичне волокно зазвичай використовується для побудови основної мережі, створення високошвидкісних з'єднань «точка-точка» між пристроями, а також для об'єднання будівель в один мережевий комплекс. Його технічні характеристики роблять його оптимальним рішенням для таких задач.

3.6. Особливості безпровідного середовища

Безпровідне середовище передавання даних забезпечує з’єднання мережевих пристроїв без використання кабелів. Передача сигналів у такому середовищі відбувається через електромагнітні хвилі.

Розглянемо найпоширеніші типи безпровідного середовища.

Радіохвилі Wi-Fi (Wireless Fidelity) – це найпоширеніша технологія бездротової передачі даних у локальних мережах (WLAN – wireless LAN). Передача здійснюється у діапазонах частот 2,4 ГГц і 5

ГГц відповідно до стандартів IEEE 802.11 (a/b/g/n/ac/ax).

Переваги використання WLAN:

- широкий радіус дії (залежно від стандарту та умов середовища, Wi-Fi мережа може покривати діапазон від 20–50 метрів у приміщенні до 100–200 метрів на відкритому просторі);
- підтримка великої кількості пристроїв;
- висока швидкість передавання (до кількох гігабітів за секунду у новітніх стандартах).

Основними недоліками бездротових локальних мереж є зниження рівня безпеки порівняно з кабельними мережами, оскільки бездротові сигнали можуть бути перехоплені зловмисниками. Іншим недоліком WLAN є більша чутливість до завад і перешкод, що може знижувати стабільність з'єднання; обмежена швидкість і пропускна здатність у порівнянні з дротовими мережами. Також недоліком використання WLAN є також зменшення якості сигналу зі збільшенням відстані від точки доступу або через фізичні перешкоди, таких як стіни чи меблі.

Інфрачервоне випромінювання (ІЧ/IR) та інфрачервоні сигнали використовуються для бездротової передачі даних на короткій відстані (наприклад, у пультах дистанційного керування або пристроях ІЧ-передачі).

Особливості:

- пряма видимість між передавачем і приймачем;
- низька швидкість передавання;
- обмежене застосування в сучасних мережах.

Мікрохвилі (точка-точка, супутниковий зв'язок) передаються по лінії прямої видимості, зазвичай використовуються для бездротових мостів, міських мереж або супутникового зв'язку.

Приклади застосування:

- підключення віддалених офісів;
- передавання сигналу між будівлями;
- глобальне покриття через супутники.

Bluetooth – це технологія для обміну даними на короткій відстані (до 100 м) між пристроями за стандартом IEEE 802.15.1.

Використання:

- підключення мишок, клавіатур, навушників;
- передавання файлів між смартфонами;

- створення особистих мереж (PAN).

Мобільні мережі (3G, 4G, 5G) забезпечують доступ до Інтернету через стільникові мережі операторів зв’язку.

Характеристики:

- велике покриття (доступ до мережі майже на всій території країни, включно з віддаленими та сільськими районами);
- підтримка мобільності;
- швидкість залежить від покоління мережі (до 10 Гбіт/с у 5G).

Супутниковий зв’язок забезпечує зв’язок у віддалених районах, де інші типи мереж недоступні. Передача відбувається через геостаціонарні або низькоорбітальні супутники.

Недоліки:

- висока затримка;
- залежність від погодних умов;
- вища вартість обладнання.

На відміну від мідних або волоконно-оптичних кабелів, бездротові мережі не залежать від фізичних провідників чи фіксованих маршрутів для передавання даних.

Основною перевагою бездротових технологій є висока мобільність. Крім того, кількість пристроїв, що підтримують бездротове з'єднання, постійно зростає. Саме тому такі мережі стали популярним вибором для домашнього використання.

Збільшення пропускної здатності також сприяє швидкому поширенню бездротових рішень.

В цілому бездротові мережі мають наступні недоліки.

Зона покриття – найкраще бездротовий зв’язок працює у відкритому просторі. Деякі будівельні матеріали та особливості рельєфу можуть значно обмежити ефективність сигналу.

Інтерференція – робота бездротової мережі може бути порушена іншими електронними пристроями, такими як радіотелефони, телевізори, мікрохвильові печі, флуоресцентні лампи або інші мережі.

На відміну від дротових мереж, у бездротовому середовищі сигнал може бути перехоплений сторонніми особами. Тому захист інформації та контроль доступу є критично важливими.

IEEE розробила ряд стандартів для бездротового передавання даних, які охоплюють як фізичний, так і канальний рівні моделі OSI.

IEEE 802.11 – найпоширеніший стандарт для локальних бездротових мереж WLAN, широко відомий як Wi-Fi. Використовує технологію доступу до середовища з уникненням колізій (CSMA/CA).

Різновиди:

802.11a/b/g/n/ac/ad – охоплюють еволюцію швидкості, частот та сумісності.

802.11ax (Wi-Fi 6 / 6E) – працює в діапазонах 2,4 ГГц, 5 ГГц, а також 6 ГГц (Wi-Fi 6E). Підвищує ефективність, пропускну здатність та енергоефективність, особливо у щільних середовищах.

802.11be (Wi-Fi 7) – новітній стандарт, який забезпечує надвисоку швидкість (до 30 Гбіт/с), використовуючи розширені канали (до 320 МГц), технологію MLO (Multi-Link Operation) та покращену модуляцію (4096-QAM).

IEEE 802.15 – стосується персональних бездротових мереж, більш відомий як Bluetooth. Призначений для передачі даних на невеликі відстані – до 100 метрів, включає:

802.15.1 – Bluetooth: короткодіапазонна технологія для з'єднання пристроїв.

802.15.4 – основа для таких протоколів як Zigbee, Thread, 6LoWPAN: використовується в автоматизації, IoT, системах розумного дому.

802.15.6 – бездротові мережі тіла (WBAN): розроблений для медичних застосунків, сенсорів здоров'я та носимих пристроїв.

IEEE 802.16 – або WiMAX, призначений для широкосмугового доступу в мережах типу "точка-точка". Втратив популярність через поширення LTE, однак залишається актуальним у деяких регіонах.

IEEE 802.20 – Mobile Broadband Wireless Access (MBWA): стандарт для високошвидкісного мобільного доступу, хоча не набув широкого поширення через конкуренцію з LTE.

IEEE 802.22 – Wireless Regional Area Network (WRAN): призначений для надання широкосмугового доступу у сільській місцевості, використовує «білі діапазони» телевізійних частот (TV White Spaces).

Для кожного з цих стандартів визначаються такі технічні характеристики фізичного рівня:

- кодування інформації в радіосигнал;

- частота та потужність передавання;
- параметри прийому і декодування сигналу;
- конструкція антен.

Підключення пристроїв до локальної мережі за допомогою бездротового зв'язку потребує певного обладнання.

Точка доступу (Access Point) приймає сигнали від бездротових пристроїв і підключається до основної мережі через кабель Ethernet. У домашніх умовах часто використовуються маршрутизатори, що поєднують функції точки доступу, комутатора і маршрутизатора.

Бездротові мережеві адаптери дозволяють кожному пристрою в мережі здійснювати бездротовий обмін даними.

Бездротові технології значно зменшують потребу у прокладанні кабелів та забезпечують зручність мобільного доступу. Водночас адміністраторам необхідно впроваджувати надійні політики безпеки, щоб захистити мережу від несанкціонованого доступу та можливих загроз.

Розвиток стандартів IEEE 802.11 забезпечує все вищу швидкість та стабільність бездротового зв'язку. Основні варіанти стандарту включають наступні.

IEEE 802.11a – працює на частоті 5 ГГц зі швидкістю до 54 Мбіт/с. Має менший радіус дії та менш ефективний усередині приміщень.

IEEE 802.11b – використовує частоту 2,4 ГГц, забезпечуючи швидкість до 11 Мбіт/с. Має більший радіус дії, кращу проникаючу здатність.

IEEE 802.11g – також працює на 2,4 ГГц, але підтримує швидкість до 54 Мбіт/с, поєднуючи сумісність з 802.11b і швидкість 802.11a.

IEEE 802.11n – використовує обидва діапазони (2,4 та 5 ГГц), забезпечує швидкість від 150 до 600 Мбіт/с і дальність до 70 м. Сумісний з попередніми версіями.

IEEE 802.11ac – функціонує на 5 ГГц і підтримує швидкість від 450 Мбіт/с до 1,3 Гбіт/с. Працює з пристроями 802.11a і 802.11n.

IEEE 802.11ad (WiGig) – використовує три діапазони: 2,4 ГГц, 5 ГГц і 60 ГГц. Теоретично забезпечує передачу до 7 Гбіт/с.

У таблиці 3.1 наведено порівняння технічних характеристик розглянутих стандартів.

Табл.3.1. **Характеристики стандартів 802.11**

Стандарт	Максимальна швидкість	Частота	Сумісність
802.11a	54 Мбіт/с	5 ГГц	нема
802.11b	11 Мбіт/с	2,4 ГГц	нема
802.11g	54 Мбіт/с	2,4 ГГц	802.11b
802.11n	600 Мбіт/с	2,4 ГГц або 5 Ггц	802.11a/b/g
802.11ac	1300 Мбіт/с (1,3 Гбіт/с)	2,4 ГГц і 5 Ггц	802.11a/n
802.11ad	7000 Мбіт/с (7 Гбіт/с)	2,4 ГГц, 5 Ггц і 60 ГГц	802.11a/b/g/n/ac

Li-Fi (Light Fidelity) – це новітня бездротова комунікаційна технологія, яка використовує видиме світло (а також інфрачервоне чи ультрафіолетове випромінювання) для передачі даних. Її суть полягає у використанні світлодіодних ламп (LED) для передачі інформації з високою швидкістю на частотах, непомітних для людського ока, що робить її альтернативою або доповненням до традиційного Wi-Fi.

Такий підхід забезпечує високошвидкісну передачу даних, зменшує електромагнітні перешкоди та підвищує безпеку, оскільки світло не проходить через стіни, обмежуючи зону покриття.

Li-Fi працює за принципом модуляції світла. Коли LED-лампа світить, вона може дуже швидко вмикатися і вимикатися з високою частотою, непомітною для людського ока. Ці перемикання формують бінарний код (0 і 1), який і передає дані. На приймальному кінці встановлюється фотодетектор, який реєструє зміни інтенсивності світла та декодує їх у цифрову інформацію.

Приклад: LED-лампа у приміщенні передає інтернет-сигнал на ноутбук чи смартфон, якщо вони знаходяться в полі її освітлення.

Основні компоненти системи Li-Fi:

- *джерело світла* – зазвичай LED-лампа, що підтримує модуляцію світла;
- *модулятор* – пристрій, який вмикає/вимикає лампу згідно з переданими даними;
- *фотодетектор* (на приймачі) – перетворює світлові сигнали в електричні;
- *інтерфейс декодування* – програмна або апаратна частина, що перетворює сигнал на читаємий користувачем формат.

LED-лампа передає дані до пристрою, оснащеного фотодетектором.

Світлові імпульси, що генеруються лампою, кодують інформацію, яку фотодетектор приймає та декодує, забезпечуючи бездротову передачу даних.

Li-Fi забезпечує швидкість до 9,6 Гбіт/с, що значно перевищує можливості Wi-Fi.

Оскільки світло не проходить через стіни, зловмисникам складніше перехопити дані, що робить Li-Fi ідеальним рішенням для використання в чутливих середовищах, таких як медичні чи урядові установи.

Li-Fi не створює електромагнітних завад, тому може використовуватися у середовищах, де радіочастотні сигнали можуть бути небажаними або забороненими, наприклад, в авіації або лікарнях.

У середовищах з великою кількістю підключених пристроїв, таких як фабрики або офіси, Li-Fi може забезпечити стабільне з'єднання без перевантаження радіочастотного спектру.

Li-Fi може бути інтегровано у наявні системи освітлення, що знижує витрати на впровадження та обслуговування мережі та має потенціал стати ключовою технологією для розвитку Інтернету речей, забезпечуючи швидку, безпечну та надійну передачу даних у різних середовищах.

Стандарт IEEE 802.11bb – це різновид бездротової мережевої технології, що базується на передачі даних за допомогою світла в межах прямої видимості та є частиною ширшої групи стандартів 802.11, які визначають правила взаємодії для різних типів бездротового зв’язку, зокрема, для пристроїв, що використовують технологію Li-Fi.

Цей стандарт розроблено для забезпечення дуже високих швидкостей передачі даних, які, за словами розробників, перевищують швидкість традиційного Wi-Fi.

Основна мета Li-Fi – запропонувати кращу пропускну здатність, ніж у мікрохвильових технологій, таких як Wi-Fi. Для досягнення такої високої продуктивності стандарт, ймовірно, використовуватиме деякі з рішень, типових для оптоволоконних мереж – наприклад, поділ каналів, що дозволяє досягати надвисоких швидкостей.

У рамках стандарту 802.11bb визначено використання світлових хвиль у ближньому інфрачервоному діапазоні – від 800 до 1000 нм. У цьому діапазоні пристрої можуть досягати швидкості від 10 Мбіт/с до 9,6 Гбіт/с, при цьому зберігаючи сумісність між пристроями з різними

технічними характеристиками.

Розробкою цього стандарту займалася спеціальна робоча група IEEE 802.11 з оптичних комунікацій. Серед компаній, які активно брали участь у створенні специфікацій, були pureLiFi та Fraunhofer HHI – організації, що мають великий досвід у галузі світлових бездротових технологій.

Рис. 3.10 ілюструє принцип роботи Li-Fi – технології бездротової передачі даних за допомогою світла, зокрема, через світлодіодні LED-лампи.

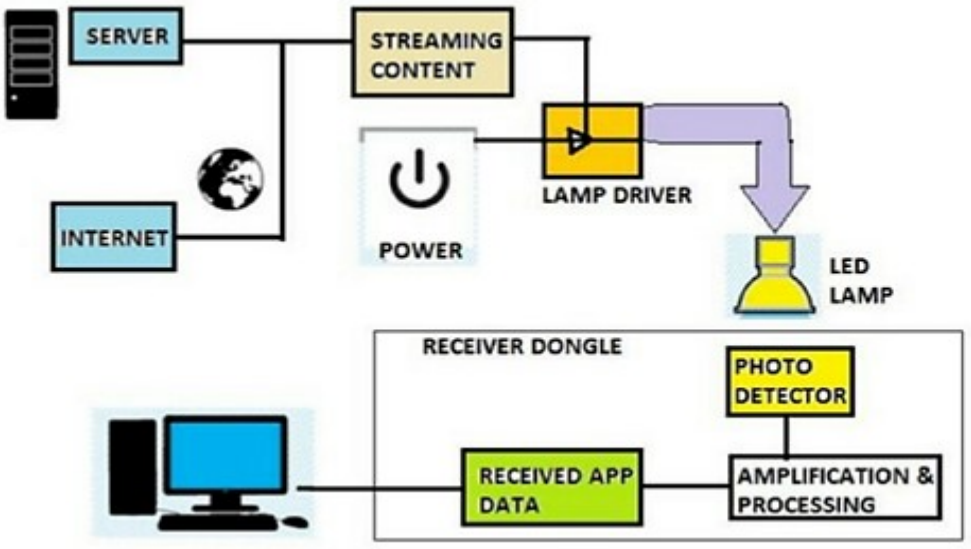


Рис. 3.10. Принцип роботи Li-Fi (Light Fidelity) – технології бездротової передачі даних за допомогою світла

У верхній частині рис. 3.10 зображено джерело даних. Сервер передає мультимедійний потік через Інтернет. Потік надходить до модуля управління лампою (Lamp Driver), який перетворює цифровий сигнал у послідовність світлових імпульсів. Для цього використовується LED-лампа, яка блимає з високою частотою, що непомітна для людського ока. Важливо, що ці імпульси не випадкові – саме в них закодовано інформацію, яку потрібно передати.

У нижній частині схеми зображено пристрій-приймач – Receiver Dongle, який складається з фотодетектора. Цей компонент вловлює зміну інтенсивності світла та перетворює її в електричний сигнал. Далі сигнал піддається посиленню та обробці, що дозволяє декодувати передані дані та передавати їх на комп’ютер або інший пристрій

користувача. Таким чином відбувається бездротова передача даних, не використовуючи радіохвилі, як це відбувається з Wi-Fi.

Li-Fi вимагає прямої видимості між LED-джерелом і приймачем, однак завдяки відсутності радіохвиль, ця технологія є безпечною для застосування у середовищах, де заборонене або небажане радіовипромінювання – наприклад, у лікарнях, літаках або на підприємствах з високими електромагнітними завадами.

Технологія Li-Fi відкриває нові перспективи для Інтернету речей, дозволяє створювати системи, у яких світильники можуть одночасно освітлювати приміщення та забезпечувати швидкий бездротовий зв'язок між пристроями. Наприклад, у «розумному» будинку LED-лампи можуть керувати передачею даних до датчиків руху, клімат-контролю, охоронних камер тощо. Завдяки великій пропускній здатності та стійкості до перешкод, Li-Fi може бути ефективним рішенням для масового підключення IoT-пристроїв, особливо там, де використання Wi-Fi є проблематичним або обмеженим.

Світло використовується як середовище для передачі даних. На відміну від Wi-Fi, який використовує радіохвилі, Li-Fi використовує видиме або ближнє інфрачервоне світло, що дозволяє уникати радіоперешкод і підвищити швидкість. При цьому Li-Fi працює лише в умовах прямої видимості між лампою та приймачем.

Переваги технології Li-Fi:

- висока швидкість передачі даних – експериментальні системи досягали понад 10 Гбіт/с, що значно швидше за Wi-Fi;
- безпека – сигнал не проходить через стіни, тому обмежується фізичною кімнатою, знижуючи ризик перехоплення;
- висока щільність покриття – кожна лампа може створювати окрему зону доступу, зменшуючи перевантаження мережі;
- енергоефективність – можна використовувати наявну систему освітлення.

Обмеження Li-Fi:

- не працює в темряві – потрібне пряме освітлення для зв'язку;
- обмежена зона дії – не проходить через стіни та непрозорі об'єкти.
- необхідність спеціального обладнання – фотодетектори, адаптери.

Розглянемо можливі сфери використання Li-Fi.

Медицина – у лікарнях, де Wi-Fi може заважати медичним пристроям.

Авіація – у салонах літаків для роздачі мультимедійного контенту без завад.

Офіси та «розумні» будівлі – для швидкої та безпечної локальної передачі даних.

Навчальні заклади – інтерактивні класи без радіохвиль.

Військові та спецслужби – зв'язок, що не виходить за межі кімнати.

Автомобілі – для обміну даними між транспортом і дорожньою інфраструктурою (VLC – visible light communication).

Станом на 2025 рік розробляються гібридні системи Li-Fi/Wi-Fi, які можуть динамічно перемикатися між мережами в залежності від умов освітлення. Такі компанії, як PureLiFi (Шотландія) і Oledcomm (Франція), уже постачають комерційні рішення, зокрема, в авіації, охороні та освіті. Також з'являються Li-Fi роутери, які працюють паралельно з LED-освітленням у звичайному будинку.

Li-Fi є перспективною технологією для IoT, зокрема в таких сценаріях:

- у лікарнях, де заборонено використовувати радіохвилі;
- у промислових приміщеннях з високим рівнем електромагнітних завад;
- у «розумних» будинках, де освітлення виконує подвійну функцію – освітлення і передавання даних;
- для підключення датчиків і пристроїв, які знаходяться в межах зони освітлення.

Завдяки високій пропускній здатності та безпечності, Li-Fi може стати основною технологією для швидкого зв'язку між IoT-пристроями, особливо в середовищах, де важлива відсутність радіовипромінювання.

Плани подальшого розвитку технології Li-Fi передбачають її інтеграцію не лише в інфраструктуру «розумного» середовища чи промисловості, а й у біомедичні системи, зокрема – у імплантовані мікрочіпи в людському тілі. Таке застосування передбачає передачу даних за допомогою світла між внутрішніми медичними пристроями та зовнішнім обладнанням для моніторингу, діагностики або лікування.

Мініатюризовані мікрочіпи, імплантовані в тіло, можуть оснащуватися мікроскопічними фотодетекторами, які можуть приймати

імпульси світла, наприклад, інфрачервоного спектра (який здатен проникати крізь м'які тканини на невеликій глибині). Ці імпульси передаватимуть дані про стан органів, температуру, рівень глюкози, артеріальний тиск або інші біомаркери.

Зовнішній приймач (наприклад, смарт-браслет, смартфон або спеціальний медичний сканер) прийматиме це світло, декодуватиме інформацію та передаватиме її далі – до лікаря або хмарного сервісу.

Технологія Li-Fi також може стати одним з каналів «внутрішнього інтернету речей» (IoB – Internet of Bodies) – де людське тіло буде не лише біологічною, а й цифровою платформою для передачі, обробки та аналізу даних.

3.7. Канальний рівень

Канальний рівень (рівень 2 у моделі OSI) відповідає за передачу даних між пристроями через фізичне середовище, тобто забезпечує обмін кадрами. Він виступає посередником між фізичним середовищем і вищими рівнями мережевої моделі, надаючи їм можливість передавати дані, не турбуючись про особливості конкретної технології або середовища.

На канальному рівні формуються кадри з мережевих пакетів, тобто об'єднуються дані з третього рівня (наприклад, IP-пакети) в зручний для передачі формат. Канальний рівень забезпечує контроль доступу до середовища, визначаючи, коли саме пристрій може передавати дані, а також виявляє помилки, які можуть виникати під час передавання.

Канальний рівень дозволяє вищим рівням не знати, через яке саме середовище (дроти, оптику, радіохвилі тощо) передаються дані. Пакети можуть переходити між різними фізичними мережами, а канальний рівень адаптує процес передачі до кожного типу середовища.

Канальний рівень має два підрівні: фізичний та логічний (рис. 3.11).

MAC (Media Access Control) – управління доступом до середовища. Контролює, як пристрої фізично передають дані, забезпечує адресацію на цьому рівні, визначає, хто і коли може передавати дані.

LLC (Logical Link Control) – логічне управління з'єднанням. Визначає, які протоколи мережевого рівня використовуються (наприклад, IPv4 або IPv6), і дозволяє їм працювати одночасно через один і той самий фізичний інтерфейс.

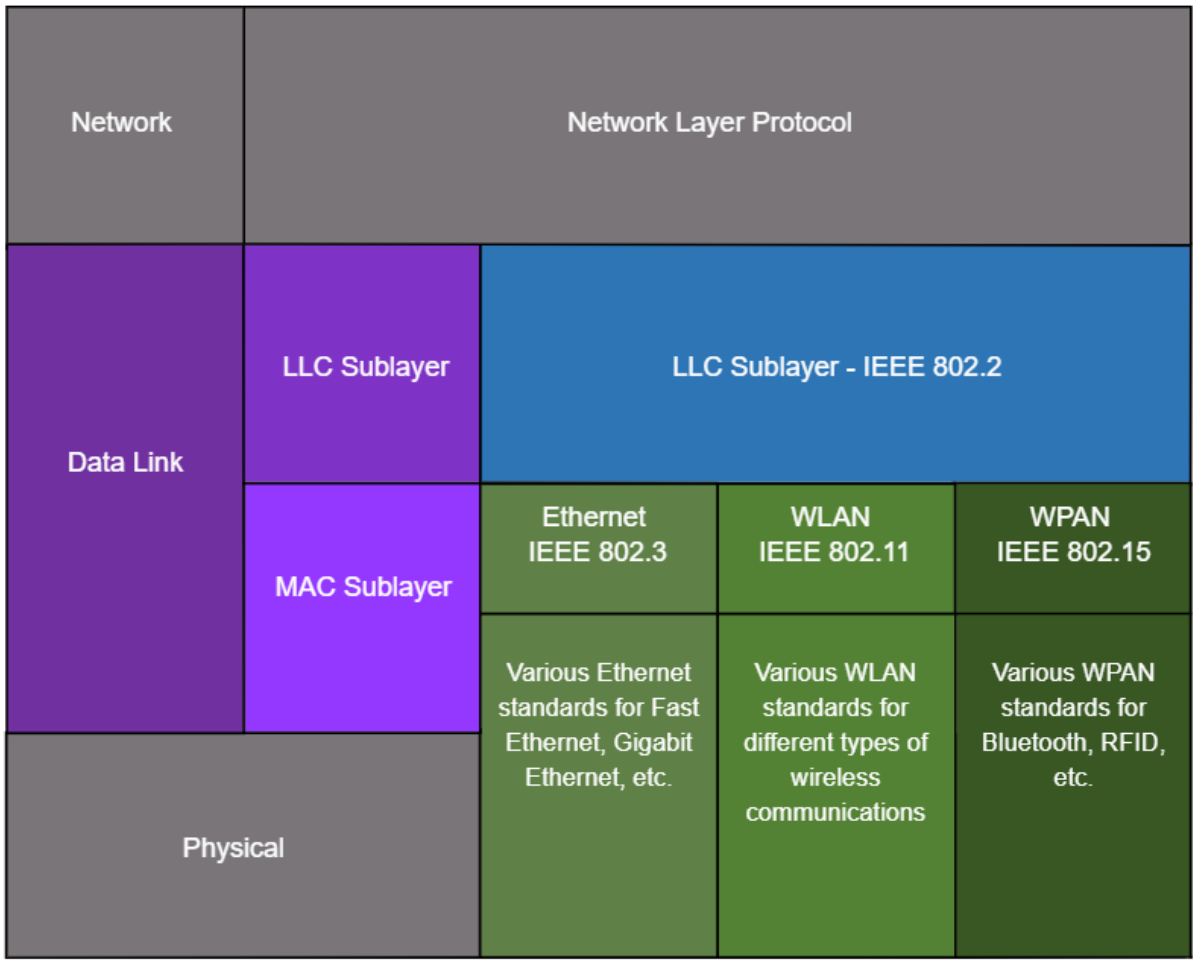


Рис. 3.11. Підрівні та основні стандарти канального рівня

Рис. 3.11 ілюструє структуру нижніх рівнів моделі OSI (Open Systems Interconnection) – фізичного, канального та мережевого. Зображено поділ канального рівня на два підрівні: LLC (Logical Link Control) та MAC (Media Access Control). Така деталізація є характерною для стандартів IEEE, зокрема для родини стандартів 802.x.

У складі канального рівня LLC-підрівень (IEEE 802.2) відповідає за управління логічними зв’язками між вузлами. Він забезпечує інтерфейс між вищим – мережевим – рівнем та MAC-підрівнем.

MAC-підрівень безпосередньо взаємодіє з фізичним середовищем передачі даних, забезпечуючи доступ до нього відповідно до певного стандарту (наприклад, Ethernet, WLAN, WPAN).

Три найбільш поширені специфікації MAC-підрівня зображено в нижній частині діаграми:

IEEE 802.3 (Ethernet) – стандарт для дротової передачі даних (Fast Ethernet, Gigabit Ethernet тощо);

IEEE 802.11 (WLAN) – стандарт для бездротових локальних мереж (Wi-Fi);

IEEE 802.15 (WPAN) – стандарт для персональних бездротових мереж (Bluetooth, RFID тощо).

У верхній частині діаграми показано, як мережевий рівень (наприклад, протокол IP) взаємодіє з підрівнем LLC. Таким чином, ми бачимо, як стандарти IEEE 802.x інтегруються в модель OSI, забезпечуючи логічну й фізичну організацію мережевих зв'язків на канальному та фізичному рівнях.

Протоколи канального рівня описують, як пакети інкапсулюються в кадри, як ці кадри передаються та як пристрої взаємодіють із передавальним середовищем. Ці правила називаються *методами доступу до середовища*.

У спільних середовищах (таких як Ethernet) пристрої конкурують за можливість передавати дані.

У з'єднаннях типу "точка-точка" передача відбувається без конкуренції.

Перед відправленням пакет інкапсулюється в кадр, який складається з:

- заголовка, що містить адресну та службову інформацію;
- даних, які включають IP-заголовок, транспортні дані та корисне навантаження;
- кінцевика, що містить контрольні коди для виявлення помилок.

Під час передачі дані перетворюються на потік бітів. Щоб розпізнати, де починається і закінчується кадр, використовуються спеціальні поля у заголовку і кінцевику (наприклад, прапори кадру).

Ключові поля кадру:

- прапори – позначають межі кадру;
- адресація – вказує джерело й призначення;
- тип – ідентифікує протокол верхнього рівня;
- управління – для керування потоком;
- дані – основне навантаження;
- перевірка помилок – допомагає виявляти помилки при передачі.

Якщо б канального рівня не існувало, мережевий рівень (наприклад, IP) мав би окремо взаємодіяти з кожним типом передавального середовища, що суттєво ускладнило б функціонування і розвиток мереж. Канальний рівень стандартизує цей процес, спрощуючи роботу всієї мережі.

Методи доступу до середовища відрізняються в залежності від топології – способу з'єднання пристроїв та типу доступу – чи використовує середовище один пристрій (точка-точка), чи кілька (спільний доступ).

При кожному переході між мережами маршрутизатор отримує кадр, витягує з нього пакет, створює новий кадр для наступної мережі та передає кадр далі.

Кожен мережевий сегмент може мати власні правила доступу до середовища, і канальний рівень адаптується до цих правил автоматично.

3.8. Поширені фізичні топології мереж

У сучасних комп'ютерних мережах найчастіше використовуються зіркоподібна та деревоподібна топології.

Зіркоподібна топологія є найпоширенішою у локальних мережах, зокрема в офісах, навчальних закладах та домашніх мережах, де кожен пристрій підключається до центрального комутатора. Така структура забезпечує простоту обслуговування, надійність та швидке виявлення несправностей.

Деревоподібна топологія, як розширення зіркової, широко застосовується в корпоративних мережах та дата-центрах, де необхідна ієрархічна структура управління трафіком і масштабованість.

У спеціалізованих випадках, наприклад, у бездротових або критичних інфраструктурах, використовується сітчаста топологія, що забезпечує високу відмовостійкість і безперервність зв'язку.

Розглянемо поширені топології комп'ютерних мереж (рис. 3.12).

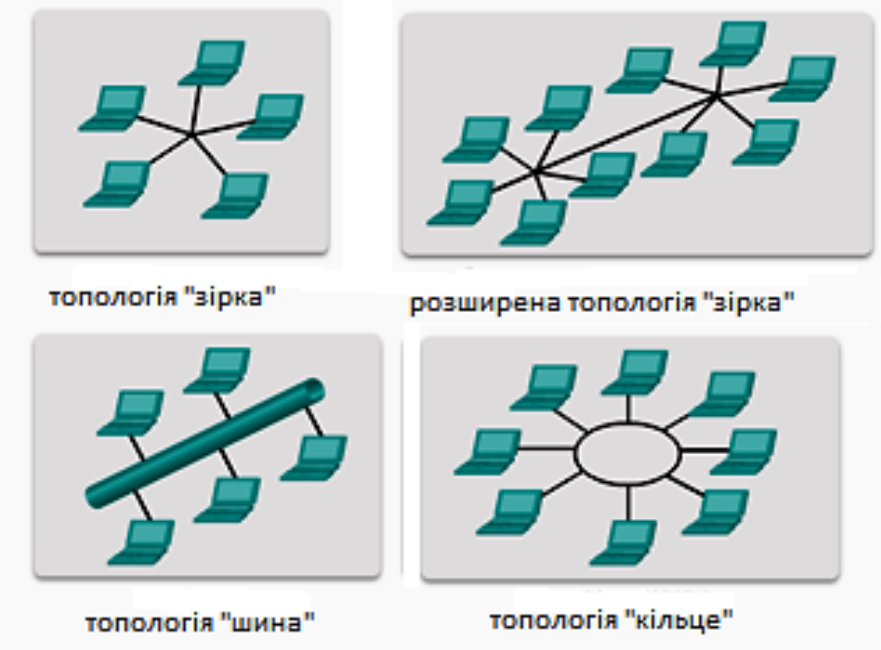


Рис. 3.12. Приклади топологій комп'ютерних мереж

Шинна топологія мережі передбачає з'єднання всіх пристроїв мережі за допомогою одного спільного кабелю, до якого підключаються всі вузли. Такий тип був поширений у перших локальних мережах, особливо в невеликих офісах або лабораторіях. Перевагою є простота реалізації і невисока вартість, однак уразливість до пошкодження кабелю, складність у виявленні несправностей і погана масштабованість обмежують її застосування в сучасних мережах.

Зіркоподібна топологія мережі передбачає підключення кожного пристрою до центрального комутатора. Ця топологія широко використовується в сучасних Ethernet-мережах завдяки високій надійності: вихід з ладу одного вузла не впливає на роботу всієї мережі. Таку структуру часто застосовують у корпоративних офісах, навчальних закладах та домашніх мережах.

Кільцева топологія мережі з'єднує всі пристрої в замкнуте коло, де дані передаються від одного вузла до іншого в одному напрямку. Прикладом реалізації може бути мережа FDDI або Token Ring. Її перевагами є передбачуваність трафіку та відсутність колізій, однак вихід з ладу одного вузла або з'єднання може паралізувати всю мережу без використання механізмів резервування.

Деревоподібна топологія мережі – це комбінація зіркової та шинної топологій, яка дозволяє будувати ієрархічну структуру мережі.

Вона є популярною в корпоративних мережах, де необхідна організація доступу за рівнями та чітка структуризація.

Сітчаста (mesh) топологія мережі передбачає наявність декількох з'єднань між вузлами, що забезпечує високу надійність і відмовостійкість. Вона застосовується в критично важливих системах, наприклад, у військових або промислових мережах, де потрібно забезпечити безперебійний зв'язок навіть у разі пошкодження окремих ліній.

Існує два основних підходи до керування доступом до спільного середовища передачі даних.

• Метод конкурентного доступу (Contention-based access) передбачає, що всі пристрої в мережі мають рівні права на доступ до середовища та змагаються за можливість передачі. У разі конфлікту (колізії) передбачено певні процедури для його вирішення.

• Керований доступ (Controlled access) дозволяє кожному пристрою використовувати середовище лише у визначений для нього час, що запобігає конфліктам.

Протоколи канального рівня встановлюють механізми керування доступом до середовища, забезпечуючи баланс між ефективною передачею кадрів, їх захистом та накладними витратами мережі.

При асоціативному (недетермінованому) доступі, пристрої можуть намагатися передати дані в будь-який момент. Щоб уникнути хаосу, застосовується технологія CSMA (Carrier Sense Multiple Access) – пристрій перевіряє, чи вільне середовище, перш ніж почати передачу. Якщо середовище зайняте, передача відкладається. Якщо ж виявлено, що середовище вільне, пристрій починає передавати дані. Цей підхід застосовується, зокрема, в Ethernet і бездротових мережах.

Проте CSMA не гарантує відсутності колізій – два пристрої можуть почати передачу одночасно, що призведе до пошкодження даних і потреби повторної передачі. Така схема не передбачає складних механізмів управління доступом чи ведення черги, але стає менш ефективною при високому навантаженні, коли зростає ймовірність конфліктів і знижується пропускна здатність.

Для підвищення ефективності були розроблені наступні методи.

• CSMA/CD (з виявленням колізій) – пристрій слухає середовище і при виявленні колізії припиняє передачу, відкладаючи її на випадковий

інтервал. Така схема застосовувалася в дротовому Ethernet.

• CSMA/CA (з уникненням колізій) – передача починається лише після попереднього повідомлення про намір передавати та підтвердження від інших пристроїв. Цей метод використовується в бездротових мережах, наприклад, у Wi-Fi (стандарт 802.11).

У мережах з множинним доступом кілька пристроїв можуть одночасно мати потенційну можливість передачі даних, але лише один з них реально передає кадр у певний момент. Хоча всі пристрої бачать цей кадр, лише адресат обробляє його. Щоб уникнути одночасного доступу кількох пристроїв, у контрольованих методах доступу застосовується черговість – наприклад, за допомогою маркера, який передається між вузлами. Тільки той, хто має маркер, може передавати дані. Після завершення передачі маркер переходить до наступного пристрою. У кільцевій топології кадр передається від одного вузла до іншого, поки не досягне адресата. Така структура дозволяє впровадити контрольований доступ у вигляді естафетної передачі, коли кадри передаються по черзі. Якщо кадр не призначений вузлу, він передається далі. Зазвичай одночасно передається лише один кадр, а коли передавати нічого, в кільці циркулює порожній кадр (маркер). Передавати дані дозволено лише після отримання маркера.

Залежно від потреб мережі, існує багато варіантів управління доступом, які можна реалізувати на основі логічної кільцевої топології – з різним ступенем контролю та складністю.

Контрольні запитання та задачі до теми 3

Перелік теоретичних запитань

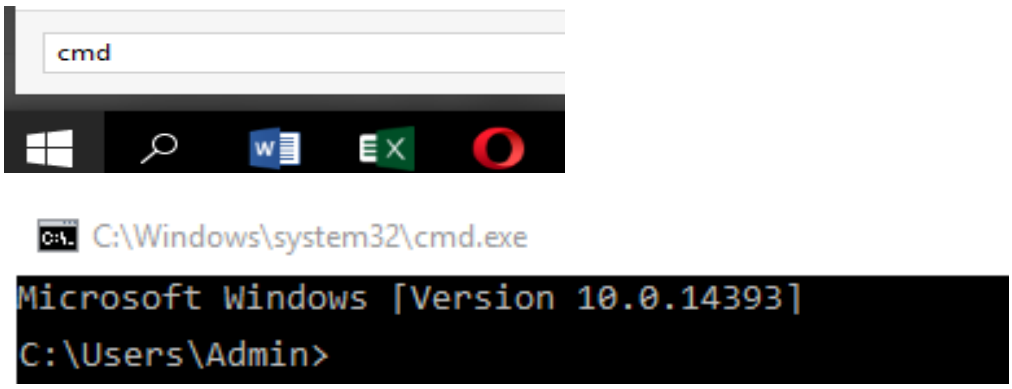
1. Які основні правила передавання даних у комп’ютерних мережах?
2. Що таке логічна та фізична адреса пристрою в мережі?
3. У чому полягає різниця між MAC-адресою та IP-адресою?
4. Які основні функції фізичного рівня моделі OSI?
5. Які типи сигналів використовуються для передачі даних на фізичному рівні?
6. Які види мідних кабелів використовуються в мережах, і які їхні особливості?
7. У чому полягає відмінність між неекранованою (UTP) і екранованою (STP) вита пара?

8. Які переваги та недоліки мають оптоволоконні кабелі порівняно з мідними?
9. Які типи оптоволоконних кабелів ви знаєте, і в яких випадках їх застосовують?
10. Які основні типи безпровідного середовища використовуються для передавання даних?
11. Які недоліки має технологія WLAN у порівнянні з кабельним з'єднанням?
12. Які функції виконує канальний рівень у мережевій моделі OSI?
13. У чому різниця між методами доступу до середовища: CSMA/CD і CSMA/CA?
14. Назвіть основні фізичні топології мереж і приклади їхнього застосування.
15. Які топології найчастіше використовуються в сучасних комп'ютерних мережах і чому?

Перелік задач для самостійного опрацювання

Практичне завдання 1. Визначення адрес та аналіз маршруту в локальній мережі.

Використовуючи командний рядок вашого комп'ютера (Пуск / *cmd*), визначте MAC-адресу та IP-адресу вашого пристрою (за допомогою команди *ipconfig/all*).



Після цього скористайтесь командою *tracert* (або traceroute в Linux) для аналізу маршруту до вебсайту (наприклад, google.com).

Дайте відповіді на питання.

– Яка фізична (MAC) адреса вашого пристрою?

– Скільки вузлів пройдено при передачі даних до сайту?

– Який тип адрес використовується на кожному етапі маршруту?

Практичне завдання 2. Вибір середовища передавання для заданої мережевої ситуації.

Уявіть, що ви проєктуєте комп'ютерну мережу для:

а) офісної будівлі;

б) заводу з електромагнітними перешкодами;

в) сільської школи з обмеженим бюджетом.

Оберіть найбільш доцільне середовище передавання даних (мідний кабель, оптоволокно, безпровідне) для кожного випадку та обґрунтуйте свій вибір.

Практичне завдання 3. Побудова фізичної топології мережі у Cisco Packet Tracer.

У середовищі моделювання Cisco Packet Tracer спроєктуйте мережу з 5 комп'ютерів і 2 комутаторів. Реалізуйте фізичну топологію «зірка». Призначте IP-адреси пристроям. Перевірте зв'язок між вузлами за допомогою команди *ping*.

Зробіть скріншот побудованої мережі та короткий звіт.

Практичне завдання 4. Аналіз роботи CSMA/CD або CSMA/CA).

Опишіть, як працює механізм CSMA/CD або CSMA/CA в ситуації, коли два пристрої одночасно хочуть передати дані. Для CSMA/CD змоделюйте (або опишіть послідовність дій) у середовищі Ethernet. Для CSMA/CA поясніть, як відбувається уникнення колізій у Wi-Fi мережі.

Перелік використаної та рекомендованої літератури

3.1. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

3.2. The Evolution of Wi-Fi Technology and Standards [Електронний ресурс]. – Режим доступу: https://standards.ieee.org/beyond-standards/the-evolution-of-wi-fi-technology-and-standards/

3.3. IEEE 802.3 Ethernet Working Group [Електронний ресурс]. – Режим доступу: https://www.ieee802.org/3/

3.4. LiFi (Light Fidelity) & its Applications [Електронний ресурс]. – Режим доступу: https://www.tec.gov.in/pdf/Studypaper/lifi%20study%20paper%20-%20approved.pdf

Тема 4. Кодування інформації в комп’ютерних мережах

У сучасному світі комп’ютерні мережі стали основою цифрового спілкування, обміну даними та функціонування більшості інформаційних систем. Щосекунди мільярди одиниць інформації передаються по мережах у вигляді сигналів. Щоб забезпечити точність, швидкість і надійність цього обміну, дані повинні бути представлені у зрозумілому для пристроїв вигляді – у формі цифрового коду. Саме тут важливу роль відіграє кодування інформації.

4.1. Роль кодування в локальних мережах

Кодування в комп’ютерних мережах – це процес перетворення даних у форму, зручну для передачі через мережеві канали. Кодування охоплює різні рівні: від перетворення символів у двійковий код до складних алгоритмів стиснення, шифрування та захисту від помилок. Залежно від контексту, кодування може слугувати різним цілям – мінімізації затримок, зменшенню обсягу передаваних даних або забезпеченню конфіденційності.

Правильне кодування є критично важливим для того, щоб дані передавалися коректно незалежно від типу середовища – чи то оптоволоконний кабель, радіосигнал або супутниковий канал. При цьому слід враховувати особливості фізичного рівня, обмеження пропускної здатності та можливі джерела перешкод. Саме завдяки кодуванню комп’ютери можуть «розуміти» один одного навіть через складні мережеві шляхи.

У цьому контексті кодування інформації виступає як один із ключових механізмів, що забезпечують стабільність та ефективність цифрових комунікацій. Розуміння принципів кодування інформації є важливим для спеціалістів у галузі ІТ, телекомунікацій, кібербезпеки та багатьох інших напрямів, пов’язаних із передаванням даних у комп’ютерних мережах.

Кодування відіграє важливу роль у функціонуванні локальних мереж, оскільки забезпечує правильне перетворення даних у сигнали, придатні для передавання по фізичному середовищу. Під час передавання кадрів по мережі дані мають бути закодовані у вигляді бітових послідовностей, які можна ефективно і точно зчитати на приймальному кінці. Завдяки цьому комп’ютери в локальній мережі

можуть "розуміти" один одного, навіть якщо використовують різне апаратне забезпечення.

Однією з ключових функцій кодування є забезпечення синхронізації між передавачем і приймачем. Різні методи лінійного кодування, наприклад, Manchester або 4B/5B, використовуються для того, щоб уникнути довгих послідовностей нулів або одиниць, які можуть ускладнити виявлення меж кадрів. Це дозволяє отримувачу правильно ідентифікувати сигнали та відновити передану інформацію без втрат.

Кодування також сприяє підвищенню надійності передавання в середовищі, де можливі перешкоди або спотворення сигналу. За допомогою спеціальних кодів виявлення або виправлення помилок (наприклад, CRC або контрольної суми) локальні мережі можуть розпізнавати пошкоджені кадри і запитувати їх повторну передачу, знижуючи ймовірність втрати важливої інформації.

У сучасних локальних мережах, які часто працюють на високих швидкостях (наприклад, 1 Гбіт/с і вище), ефективні методи кодування також допомагають мінімізувати затримки та підвищувати пропускну здатність. Це особливо важливо для додатків реального часу, таких як відеоконференції, IP-телефонія або хмарні сервіси, де затримки в передаванні можуть суттєво вплинути на якість зв’язку.

Загалом, кодування в локальних мережах є невід’ємним механізмом, що забезпечує сумісність, швидкість, надійність і безпеку передавання даних між усіма пристроями в межах локального сегмента мережі.

4.2. Основи кодування інформації

Кодування – це процес перетворення інформації з однієї форми подання в іншу для її зберігання, обробки та передавання. У контексті комп’ютерних мереж і цифрових систем кодування означає представлення даних у вигляді бітових послідовностей (нулів і одиниць), придатних для обробки технічними засобами. Завдяки кодуванню інформація може передаватися на великі відстані, зберігати точність, бути захищеною від пошкоджень і зчитуваною приймачем.

В основі кодування лежить принцип цифрового подання – тобто подання будь-якої інформації (тексту, зображень, аудіо, відео тощо) у

вигляді послідовностей бітів. Для цього використовуються різні коди: ASCII для тексту, UTF-8 для розширеного набору символів, JPEG/PNG для зображень, MP3 для аудіо та інші. Кожен тип інформації має свої стандарти кодування, що забезпечують точне перетворення та сумісність між пристроями.

Будь-яке кодування базується на обмеженому наборі символів – алфавіті. У цифрових системах основним алфавітом є двійковий (0 і 1).

Коди – це умовні правила, за якими кожному символу алфавіту джерела ставиться у відповідність певна комбінація з символів цільового алфавіту.

Наприклад, у коді ASCII літера "A" кодується як 01000001. Такий підхід дозволяє однозначно ідентифікувати передані дані.

Основними характеристиками кодування є: однозначність (кожне повідомлення має лише одне кодування), зворотність (можливість точно відновити вихідну інформацію з коду), ефективність (мінімальна довжина коду при збереженні точності), стійкість до помилок.

Залежно від мети, кодування може бути орієнтованим на стиснення, захист або структуризацію даних.

У комп’ютерних мережах кодування має вирішальне значення для передачі даних по фізичному середовищу. Перш ніж дані покинуть передавач, вони проходять кілька етапів кодування: від високорівневих форматів (наприклад, HTML або JSON) до фізичних сигналів (наприклад, імпульсів напруги, світла чи радіохвиль). Кожен рівень моделі OSI має свої специфічні механізми кодування. На фізичному рівні, наприклад, застосовують методи, що забезпечують синхронізацію та зменшення спотворень.

У процесі передавання даних можуть виникати помилки, тому важливо використовувати спеціальні схеми кодування для контролю цілісності. Найпоширенішими є контрольні суми, циклічний надлишковий код, парність та коди з виправленням помилок, наприклад, код Хеммінга або різницеві коди. Вони дозволяють не лише виявити спотворення, а й у деяких випадках виправити їх без повторного передавання.

Стиснення – це ще одна важлива форма кодування, яка дозволяє зменшити обсяг даних без втрати (наприклад, ZIP, PNG) або з втратою якості (наприклад, MP3, JPEG). Воно відіграє важливу роль у

підвищенні ефективності мережевих ресурсів, оскільки зменшує час передачі й навантаження на канали зв'язку.

Для забезпечення сумісності між різними пристроями використовуються стандартизовані способи кодування.

Наприклад, у вебтехнологіях – UTF-8, у передачі мультимедіа – MPEG, у бездротових мережах – специфічні методи кодування на фізичному рівні, такі як QAM або OFDM. Протоколи мережевого рівня також використовують кодування для структурування пакетів та маршрутизації.

4.3. Методи кодування на фізичному рівні OSI

Фізичний рівень моделі OSI відповідає за передачу бітів по фізичному середовищу – наприклад, через мідні кабелі, оптичне волокно або радіохвилі.

Однак передати самі біти (тобто послідовності 0 і 1) неможливо без відповідного перетворення в фізичні сигнали – електричні імпульси, зміни напруги, частоти, фази або світлові імпульси. Тому кодування на фізичному рівні – це перетворення бітів у сигнали, які може зчитати приймальний пристрій.

Обраний метод впливає на швидкість, надійність, відстань передачі та захищеність від шумів (рис. 4.1).

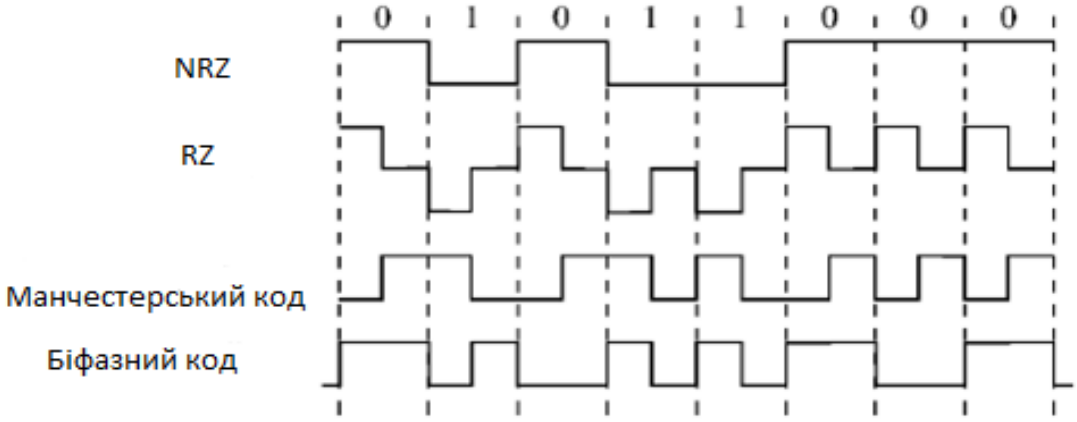


Рис. 4.1. Поширені коди передачі інформації

Небалансоване (нелінійне) бітове кодування (Unipolar) – у найпростішому вигляді кодування – біт 1 представляється як наявність напруги (наприклад, +5 В), а біт 0 – її відсутність (0 В). Такий метод

простий для реалізації, але має серйозні недоліки: постійний рівень напруги створює постійний струм, що призводить до втрат енергії, і немає чіткої синхронізації між відправником і приймачем. До того ж, при довгій послідовності однакових бітів (наприклад, багато нулів), важко визначити межі бітів.

Біполярне кодування (Bipolar) – у цьому кодуванні біт 1 кодується як позитивна або негативна напруга (чергуючись), а біт 0 – як нульова напруга. Такий підхід дозволяє краще зберігати синхронізацію, а також виявляти помилки (наприклад, дві однакові напруги поспіль замість чергування). Біполярні методи, такі як AMI (Alternate Mark Inversion), активно використовуються в телекомунікаціях. Проте й вони неефективні при довгих послідовностях нулів, де немає змін сигналу.

NRZ (Non-Return to Zero) – без повернення до нульового рівня – це одна з форм лінійного кодування, при якій сигнал не повертається до нульового рівня між бітами. У варіанті NRZ-Level біт 1 – це високий рівень напруги, біт 0 – низький. У NRZ-Inverted (NRZI) зміна сигналу позначає біт 1, а відсутність зміни – біт 0. NRZI дає кращу синхронізацію, особливо при довгих послідовностях однакових бітів, і використовується, наприклад, у USB.

Кодування NRZ є одним із найпростіших способів цифрового представлення даних. У цьому методі використовуються лише два стани сигналу, які відповідають логічним значенням "0" і "1", без переходу до нейтрального рівня між ними. Логічний нуль може бути закодований одним рівнем напруги, а логічна одиниця – іншим. При цьому зміна рівня сигналу протягом бітового інтервалу не відбувається.

У такому кодуванні логічна одиниця може відповідати, наприклад, низькому рівню напруги, а нуль – високому (або навпаки, залежно від реалізації). Допускаються як рівні однакової полярності, так і протилежної. Під час передачі кожного біта сигнал залишається стабільним на визначеному рівні протягом усього інтервалу.

Основною перевагою NRZ є його технічна простота – сигнал можна безпосередньо передавати та приймати без додаткового кодування або декодування. Крім того, цей метод вимагає мінімальну смугу пропускання серед усіх способів кодування. Наприклад, при передаванні послідовності типу 101010... із частотою 10 Мбіт/с (один біт – 100 нс), зміна сигналу відбувається кожні 200 нс, тож необхідна смуга

пропускання становить 5 МГц (рис. 4.2).

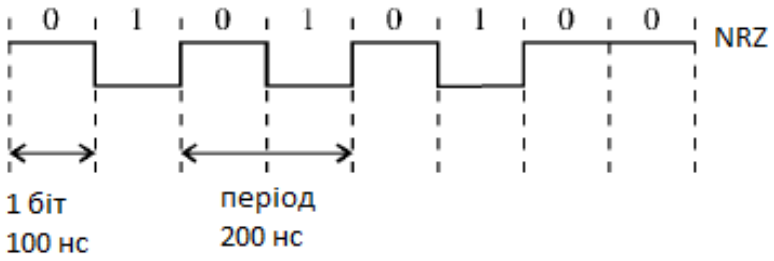


Рис. 4.2. Швидкість передачі і необхідна пропускна спроможність при кодуванні NRZ

Найсуттєвіший недолік NRZ полягає у відсутності синхронізуючих сигналів протягом тривалих послідовностей однакових бітів. Якщо передаються лише нулі або лише одиниці, приймач втрачає орієнтир і може не визначити коректно межі бітів. Оскільки синхронізація відбувається лише на стартовому біті, подальше декодування залежить від точності внутрішнього годинника приймача. Якщо його частота навіть трохи відрізняється від частоти передавача, накопичена помилка за довгий пакет може призвести до зсуву розмітки біта і втрати даних. Наприклад, для пакету з 10 000 біт похибка синхронізації має бути меншою за 0,01%, щоб уникнути збоїв, навіть якщо сигнал ідеальний. Один із варіантів вирішення цієї проблеми – використання окремої лінії для передачі синхросигналу. Проте це ускладнює інфраструктуру, збільшує витрати на кабель, приймачі та передавачі. Тому NRZ здебільшого застосовується для передачі коротких пакетів – зазвичай до 1 Кбіт.

Ще один недолік NRZ – необхідність у фіксованій довжині переданих повідомлень. Приймач не може самостійно визначити, чи триває передача, чи вже завершилася. Для цього використовується спеціальний стартовий біт, рівень якого відрізняється від лінії в пасивному стані (наприклад, якщо без сигналу лінія в стані "0", то стартовий біт – "1"). Завершення прийому настає після фіксованої кількості біт, які рахує приймач.

Найвідомішим прикладом використання NRZ є стандарт RS-232C, що застосовується в послідовних портах комп’ютерів. Тут дані передаються у вигляді байтів (8 біт), яким передують стартовий і завершальний біти.

На відміну від NRZ, такі методи як RZ, манчестерський і біфазний коди мають спеціальні переходи сигналу всередині кожного бітового інтервалу. Це дозволяє приймачеві синхронізуватися по кожному біту окремо, відстежуючи зміну сигналу. У результаті навіть при незначній невідповідності частот годинників передавача і приймача прийом даних залишається стабільним. Такі методи називаються *самосинхронізуючими*. Вони фактично поєднують передавання даних із синхросигналом. При ще вищих швидкостях передачі застосовують складніші схеми кодування, такі як 4B/5B.

RZ (Return to Zero) – з поверненням до нульового рівня – кожен біт сигналу складається з двох частин: активного рівня (наприклад, +5 В для біт 1) і повернення до нуля. Це дає чіткіший поділ між бітами і полегшує синхронізацію, але водночас зменшує ефективність передачі – на кожен біт витрачається більше часу. Цей метод менш енергоефективний, але підходить для середовищ з високими вимогами до точності.

Код RZ – це трирівневий метод кодування, який отримав свою назву через особливість сигналу повертатися до середнього (нульового) рівня після активного стану у першій половині бітового інтервалу. У кожному інтервалі біта сигнал повертається до нейтрального значення – наприклад, до нульового потенціалу. При цьому логічна одиниця може відповідати позитивному імпульсу, а логічний нуль – негативному (або навпаки), причому лише в першій половині інтервалу.

Оскільки у середині кожного біта обов’язково відбувається зміна сигналу (у бік нуля), це дозволяє приймачеві легко виділяти синхроімпульси. Такий тип кодування забезпечує синхронізацію не лише на початку пакету, як у NRZ, а й у межах кожного біта, що суттєво знижує ризик втрати синхронізації при передачі даних будь-якої довжини.

Ще однією перевагою RZ є виявлення початку і завершення потоку біт. Приймач орієнтується на наявність чи відсутність зміни сигналу в межах бітового інтервалу. Якщо в інтервалі відсутній перехід, це свідчить про завершення послідовності біт (рис. 4.3).

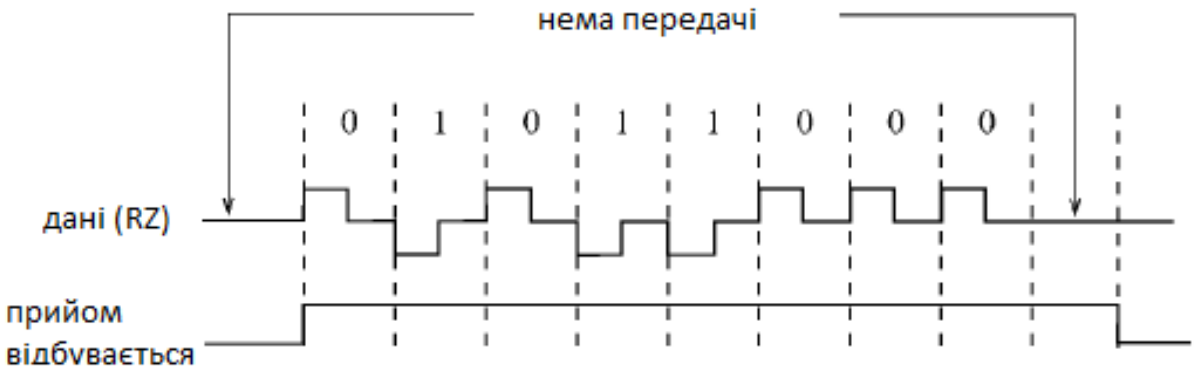


Рис. 4.3. Визначення початку і кінця прийому у коді RZ

Втім, цей метод має і недоліки. Зокрема, для передачі тієї ж кількості інформації, що й код NRZ, код RZ вимагає у два рази ширшої смуги пропускання, адже на кожен біт припадає два зміни сигналу. Наприклад, для швидкості передачі 10 Мбіт/с потрібна смуга пропускання в 10 МГц, тоді як для NRZ вистачає 5 МГц (рис. 4.4).

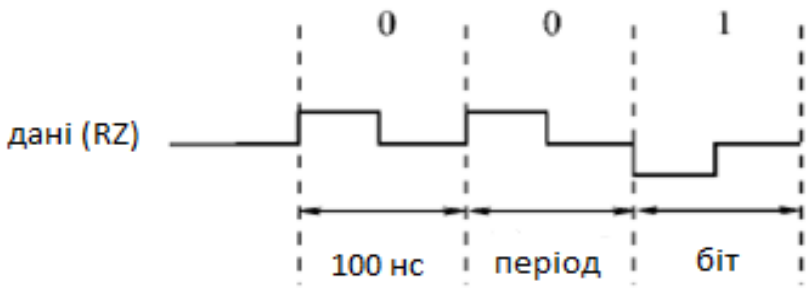


Рис. 4.4. Швидкість передачі і пропускна спроможність при коді RZ

Ще однією проблемою при використанні RZ є необхідність обробки трьох рівнів сигналу, що ускладнює електроніку передавачів і приймачів. Проте цей тип кодування знаходить застосування не лише у дротових мережах, але й в оптоволоконних системах.

У випадку оптики замість позитивних і негативних рівнів використовуються варіації світлового потоку: повна відсутність світла, помірна яскравість і сильне освітлення.

Це дозволяє постійно підтримувати світловий сигнал у лінії, навіть коли дані не передаються, забезпечуючи таким чином контроль за цілісністю з’єднання без додаткових засобів (рис. 4.5).

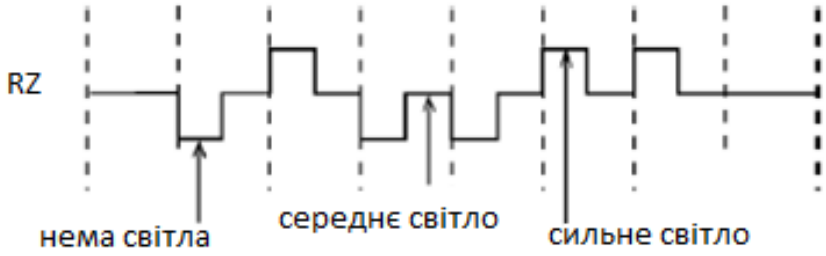


Рис. 4.5. Використання коду RZ в оптоволоконних мережах

Манчестерське кодування (Manchester Encoding) поєднує кодування і синхронізацію, кожен біт представлений переходом сигналу: від високого до низького (1) або навпаки (0). Завдяки обов'язковому переходу на кожному біті, підтримується тактова синхронізація між передавачем і приймачем. Цей метод використовується в Ethernet (10 Мбіт/с), RFID-системах і магнітних картках, в технології Near Field Communication (NFC). Основним недоліком є подвоєна частота передачі (тому що кожен біт потребує двох переходів), що збільшує ширину спектра сигналу. Манчестерське кодування – це метод, при якому логічний нуль передається як перехід сигналу від високого до низького рівня, а логічна одиниця – як перехід від низького до високого.

Цей код належить до класу самосинхронізуючих, але, на відміну від коду RZ, у ньому використовується лише два рівні сигналу. Це не лише покращує захищеність від перешкод, а й спрощує конструкцію приймачів і передавачів. Логічний нуль кодується переходом від низького до високого рівня в середині бітового інтервалу, а логічна одиниця – переходом від високого до низького (або навпаки, залежно від реалізації). Як і у випадку з RZ-кодуванням, обов'язковий перехід сигналу в центрі кожного біта дає можливість приймачу з високою точністю виділяти тактову частоту та підтримувати синхронізацію протягом тривалих послідовностей даних. При цьому допустиме відхилення частоти внутрішнього генератора приймача від частоти передавача може становити до 25%.

Однак, через те, що в кожному бітовому інтервалі гарантовано відбувається зміна сигналу, для передачі із швидкістю 10 Мбіт/с потрібна смуга пропускання до 10 МГц – удвічі більше, ніж для NRZ (рис. 4.6).

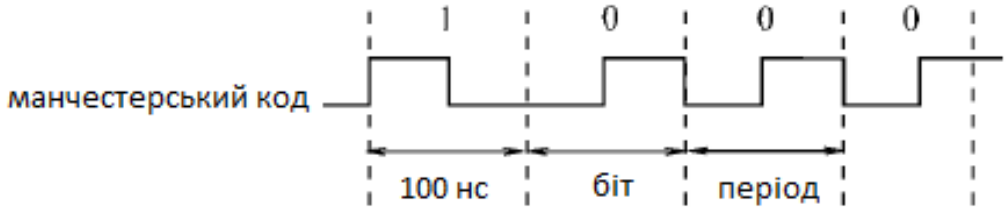


Рис. 4.6. Швидкість передачі і пропускна здатність при манчестерському кодуванні

Крім того, наявність переходу сигналу дозволяє визначати як початок, так і завершення переданої бітової послідовності. Якщо протягом бітового інтервалу відсутній перехід, це сигналізує про завершення передачі. Таким чином, манчестерський код підтримує передавання послідовностей змінної довжини (рис. 4.7).

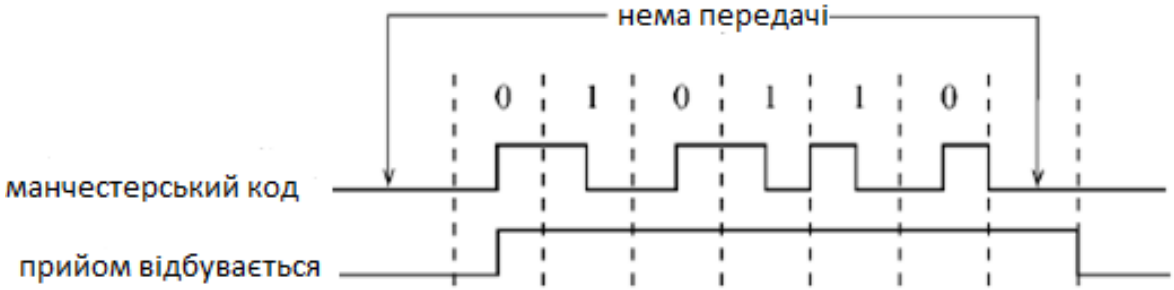


Рис. 4.7. Визначення початку і кінця прийому при манчестерському коді

Манчестерське кодування використовується як для електричних, так і для оптичних середовищ. У випадку з оптоволокном один із рівнів відповідає наявності світлового імпульсу, інший – його відсутності.

Однією з важливих переваг манчестерського сигналу є наявність постійної складової, що виникає в результаті того, що сигнал протягом кожного біта проводить половину часу в кожному з двох рівнів. Якщо ці рівні симетричні за амплітудою (наприклад, +A і –A), середнє значення сигналу буде нульовим. Це дозволяє реалізовувати гальванічну розв’язку за допомогою трансформаторів без необхідності в окремому живленні лінії, а також знижує вплив низькочастотних завад, які трансформатор не пропускає.

У разі, якщо один з рівнів має значення 0 (як, наприклад, в Ethernet), середнє значення сигналу дорівнює приблизно половині його

амплітуди. Це використовується для виявлення колізій (зіткнень пакетів) у мережі: зміна постійної складової сигналу дозволяє системі фіксувати помилкові ситуації.

Спектр частот при манчестерському кодуванні обмежується лише двома частотами: для швидкості 10 Мбіт/с – це 10 МГц (при передачі одних лише нулів або одиниць) та 5 МГц (при передачі бітів, які чергуються, наприклад, 101010...). Завдяки цьому можна застосовувати смугові фільтри для ефективного пригнічення завад, шумів та сторонніх сигналів.

Диференційне манчестерське кодування (Differential Manchester) – у цьому варіанті зміна сигналу в середині кожного біту обов'язкова (як у звичайному манчестерському), але значення біта визначається наявністю чи відсутністю зміни на початку. Це робить метод менш чутливим до інверсії сигналу під час передачі даних. Диференційне манчестерське кодування застосовується в стандарті Token Ring і деяких інфраструктурах безпеки.

У деяких сучасних мережах, наприклад, Fast Ethernet або Gigabit Ethernet, використовується блочне кодування, яке перетворює групи бітів (наприклад, 4 або 8) у довші, але контрольовані коди (5 або 10 біт відповідно). Це дозволяє уникати довгих послідовностей нулів або одиниць і гарантує достатню кількість переходів сигналу для синхронізації. Наприклад, 8B/10B кодування активно використовується у Gigabit Ethernet, Fibre Channel і PCI Express.

Усі сучасні коди передавання даних намагаються знайти оптимальний баланс між шириною смуги пропускання, необхідною для заданої швидкості, та здатністю до самосинхронізації. Розробники прагнуть зберегти самосинхронізуючі властивості, уникаючи при цьому подвоєння вимог до смуги пропускання, як це спостерігається у випадках із кодами RZ, манчестерським або біфазним кодом.

Найпоширенішим підходом є додавання до потоку даних спеціальних бітів синхронізації. Наприклад, може додаватися один синхробіт на кожні 4, 5 або 6 інформаційних бітів, або два на вісім. Водночас це не є простою вставкою додаткових бітів – інформаційні біти перетворюються в більші за розміром коди, які передаються мережею, після чого приймач виконує зворотне перетворення, відновлюючи оригінальну інформацію. Такий підхід також спрощує

виявлення наявності передачі.

Наприклад, для мереж FDDI (зі швидкістю 100 Мбіт/с) використовується код 4B/5B, де кожна 4-бітова група інформації перетворюється в 5-бітову послідовність. Це дозволяє синхронізуватися не на кожному біті, як у манчестерському кодуванні, а раз на 4 біти. При цьому смуга пропускання зростає лише на 25% порівняно з NRZ (до 62,5 МГц замість 100 МГц). Аналогічні схеми використовуються й в інших кодах, таких як 5B/6B (у стандарті 100VG-AnyLAN) або 8B/10B (у Gigabit Ethernet).

У варіанті 100BASE-T4 мережі Fast Ethernet застосовується інший підхід – кодування 8B/6T. Тут інформація передається паралельно по трьох витих парах у вигляді трирівневих сигналів. Це дозволяє досягати 100 Мбіт/с навіть на дешевих кабелях категорії 3 з шириною смуги лише 16 МГц. Однак цей метод вимагає більших витрат на кабелі й більше передавально-приймальних пристроїв, а також забезпечення однакової довжини дротів для уникнення різниці в затримках.

Іноді закодовані дані додатково обробляються для полегшення синхронізації. Найчастіше для цього використовують коди NRZI (Non-Return to Zero Invert) у оптоволоконних мережах (FDDI, 100BASE-FX) та MLT-3 у мережах з витою парою (TPDDI, 100BASE-TX). Обидва коди не є самосинхронізуючими (рис. 4.8).

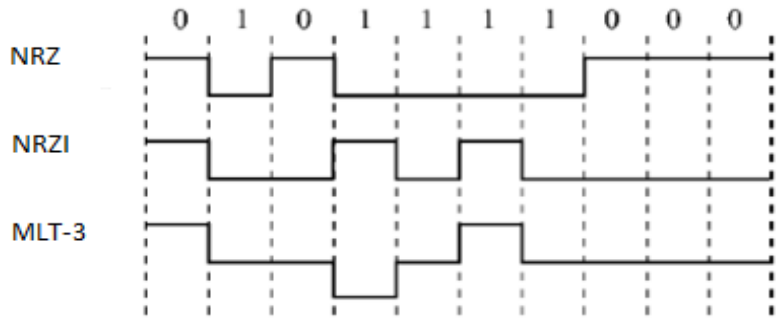


Рис. 4.8. Кодування NRZI і MLT - 3

Код NRZI передбачає зміну рівня сигналу на протилежний лише при передачі одиниці; нуль передається без зміни рівня. Це дозволяє мати переходи у випадку послідовних одиниць, що полегшує синхронізацію порівняно з NRZ, у якому немає переходів ані при нулях,

ані при одиницях.

MLT-3 (Multi-Level Transition – 3) використовує три рівні сигналу (+U, 0, –U). При передачі нуля рівень не змінюється, а при одиниці – переходить на наступний рівень у циклі. Завдяки цьому частота зміни сигналу зменшується вчетверо порівняно зі швидкістю передачі, що дозволяє знизити вимоги до смуги пропускання нижче, ніж у NRZ.

Усі ці коди використовують прямокутні імпульси на двох або трьох рівнях, що безпосередньо передаються мережею.

Аналогове кодування – у деяких мережах застосовується інший підхід – модуляція цифрових імпульсів високочастотним аналоговим сигналом (зазвичай синусоїдальним).

Такий метод дозволяє значно збільшити пропускну спроможність каналу при широкосмуговій передачі, оскільки забезпечує одночасну передачу кількох бітів. Крім того, на відміну від цифрових сигналів, аналогові (синусоїдальні) сигнали не зазнають значних спотворень форми, лише зниження амплітуди (рис. 4.9).

Модуляція – це зміна одного параметра несучої хвилі (амплітуди, частоти або фази) під впливом інформаційного сигналу.

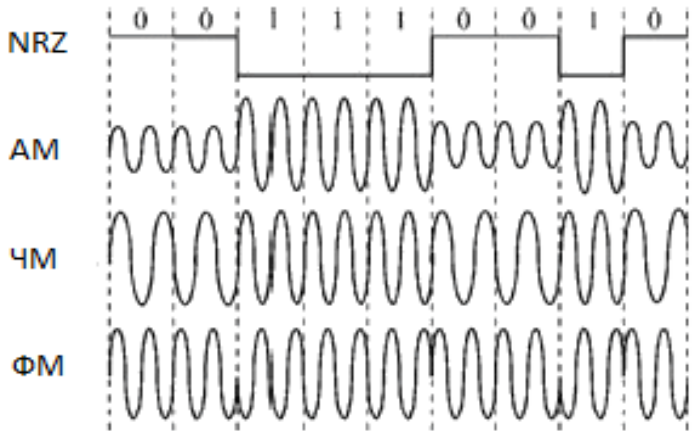


Рис. 4.9. Аналогове кодування цифрової інформації

Найпростіші способи аналогового кодування – це амплітудна, частотна та фазова модуляція.

Амплітудна модуляція (АМ) – логічній одиниці відповідає наявність сигналу або більша амплітуда, нулю – менша або відсутність сигналу.

Частота залишається сталою. Недолік – висока чутливість до шумів. Переваги – простота реалізації та вузький спектр частот.

Частотна модуляція (ЧМ) – одиниця і нуль кодуються сигналами різних частот при сталій амплітуді. Частота сигналу змінюється відповідно до логічного значення. Амплітуда залишається сталою. Цей спосіб менш чутливий до шумів.

Фазова модуляція (ФМ) – логічним змінам відповідають стрибки фази сигналу з постійною амплітудою та частотою.

Існують складніші способи модуляції, які комбінують вищезгадані методи. Аналогове кодування зазвичай використовується в середовищах з обмеженою смугою пропускання (наприклад, телефонні лінії у глобальних мережах) або в радіоканалах, де необхідно обслуговувати кількох користувачів. У локальних кабельних мережах аналогове кодування майже не застосовується через високу вартість та складність обладнання.

Імпульсно-кодова модуляція (ІКМ) перетворює аналоговий сигнал (наприклад, голос) у цифровий шляхом дискретизації його амплітуди і представлення цих значень у двійковій формі. Частота дискретизації повинна бути вдвічі більшою за найвищу частоту спектра сигналу (відповідно до теореми Котельникова).

На фізичному рівні дані передаються у вигляді електричних, оптичних або радіосигналів, які представляють логічні «1» і «0». Це відбувається через зміну певних параметрів сигналу – наприклад, рівня напруги чи тривалості імпульсу. Наприклад, довший імпульс може означати одиницю, коротший – нуль. Це подібно до використання азбуки Морзе, де символи передаються за допомогою різної тривалості світлових або звукових сигналів.

Асинхронний метод передачі – передача сигналів без тактового синхронізуючого сигналу. Інтервали між переданими символами можуть бути нерегулярними, тому потрібно спеціальне маркування початку та кінця кадру (наприклад, прапори).

Синхронний метод передачі – передача з чітко визначеним тактовим сигналом, що регламентує точні моменти передачі кожного біта.

Більш складні методи модуляції поєднують ці способи. Аналогове кодування особливо корисне при обмеженій смузі пропускання

(наприклад, у телефонних мережах) або в радіозв'язку. У локальних мережах аналогове кодування використовується рідко через складність апаратури.

У бездротових мережах та оптоволоконних каналах сигнали кодуються за допомогою модуляції: амплітудної, частотної, фазової або їх комбінацій (QAM – квадратична амплітудна модуляція). Наприклад, Wi-Fi (802.11n/ac/ax) і 4G/LTE використовують складні схеми, такі як 64-QAM або 256-QAM для ефективної передачі великого обсягу даних.

4.4. Кодування на канальному рівні OSI

Кодування на канальному рівні – це процес перетворення цифрових даних, що надходять від мережевого рівня, у форму, придатну для передавання фізичними засобами, з урахуванням обмежень фізичного середовища. Це кодування забезпечує надійну передачу даних, синхронізацію, контроль помилок та інші функції, які є ключовими для безперебійного обміну інформацією між мережевими вузлами. Однією з основних задач кодування на канальному рівні є забезпечення синхронізації між передавачем і приймачем. Синхронізація дозволяє приймачу визначити межі бітів у потоці даних. Якщо передавати дані без належного кодування (наприклад, довгі послідовності нулів або одиниць), приймач може втратити орієнтацію в потоці бітів. Самосинхронізуючі коди, такі як манчестерський або біфазний, вирішують цю проблему, вставляючи переходи сигналу, які дають змогу відновити тактовий сигнал приймача.

Багато традиційних самосинхронізуючих кодів, наприклад, манчестерський або RZ-коди, подвоюють кількість необхідної смуги пропускання, оскільки передбачають часті зміни рівня сигналу.

Щоб зменшити перевантаження каналу, були розроблені більш ефективні методи кодування, які вводять біти синхронізації в обмеженій кількості, зберігаючи при цьому самосинхронізацію. Прикладом є коди типу 4B/5B, 5B/6B або 8B/10B, де групи інформаційних бітів замінюються кодованими комбінаціями з контрольованими властивостями. У мережі FDDI та 100BASE-FX (оптоволоконні канали) застосовується 4B/5B-кодування – кожні 4 біти інформації замінюються 5-бітовим кодом, який забезпечує достатню кількість переходів сигналу для синхронізації. У Gigabit Ethernet використовується 8B/10B-

кодування, яке додає лише 25% надлишковості, але при цьому гарантує часті переходи та контроль дисбалансу сигналу. У деяких технологіях, наприклад, 100BASE-T4, реалізовано багаторівневе кодування – код 8B/6T, який передає дані трьома сигналами одночасно по трьох витих парах. Це дозволяє знизити частотні вимоги до кожної пари, використовуючи дешеві кабелі категорії 3. Ще один приклад – код MLT-3 (Multi-Level Transition-3), який змінює рівень сигналу лише при передачі одиниць і тільки між трьома рівнями (+U, 0, -U). Його використовують у 100BASE-TX – він зменшує частоту зміни сигналу, а отже, і смугу пропускання.

Часто після основного кодування застосовується ще й додаткове кодування, таке як NRZI (Non-Return to Zero Inverted) – зміна рівня сигналу при передачі одиниці, без зміни для нуля. Воно дозволяє краще виявляти переходи, ніж звичайне NRZ-кодування. Цей підхід, наприклад, також використовується у мережах FDDI.

Кодування на канальному рівні є критично важливою частиною функціонування фізичного рівня передачі даних. Воно забезпечує:

- надійну синхронізацію між відправником і одержувачем;
- ефективне використання пропускної здатності каналу;
- мінімізацію впливу шумів;
- виявлення та, частково, виправлення помилок;
- збереження енергетичного балансу сигналу (щоб уникнути зміщення постійної складової).

Кожен стандарт Ethernet або інший мережевий протокол обирає той чи інший метод кодування залежно від швидкості, середовища передачі й вимог до вартості обладнання.

4.5. Порівняння методів кодування інформації

Розглянемо порівняння відомих методів кодування інформації (табл. 4.1). Порівняльна таблиця демонструє, як різні методи кодування вирішують ключові задачі фізичного та канального рівнів: синхронізацію, ефективність використання смуги пропускання та простоту апаратної реалізації.

Таблиця 4.1. **Порівняльний аналіз методів кодування інформації в комп'ютерних мережах**

Метод кодування	Принцип роботи	Переваги	Недоліки	Приклади використання
Unipolar (уні-полярне)	1 = напруга, 0 = відсутність	Простота реалізації	Високе енергоспоживання, погана синхронізація	У деяких навчальних системах
Bipolar (біполярне, AMI)	1 = чергування + та - напруги, 0 = 0 В	Виявлення деяких помилок, зниження постійної складової	Проблеми з довгими послідовностями нулів	Телефонія, T1-лінії
NRZ-Level	1 = високий рівень, 0 = низький	Енергоефективне	Відсутність переходів при однакових бітах	Старі інтерфейси, RS-232
NRZ-Inverted (NRZI)	1 = зміна сигналу, 0 = без зміни	Покращена синхронізація	Проблеми при довгих послідовностях нулів	USB, IBM SDLC
RZ (Return to Zero)	Повернення до нуля після кожного біта	Краща синхронізація	Знижена пропускна здатність	Теоретичні моделі, менш поширене
Manchester	Перехід у середині біта: ↑ (0) або ↓ (1)	Вбудована синхронізація, просте розпізнавання	Вдвічі більше переходів, ширший спектр сигналу	Ethernet 10 Мбіт/с, RFID
Differential Manchester	Перехід у середині біта завжди; початковий перехід визначає біт	Менш чутливий до інверсій, збереження синхронізації	Трохи складніший для реалізації	Token Ring, деякі промислові системи
4B/5B	4 біти → 5-бітовий код без довгих нулів	Гарантія переходів, надійна синхронізація	Надмірність, складність кодування	Fast Ethernet (100Base-TX)
8B/10B	8 біт → 10-бітовий код з контрольованою частотою переходів	Виявлення помилок, хороша синхронізація	Зниження ефективності (20% надлишку)	Gigabit Ethernet, Fibre Channel, PCIe
Модуляція (ASK, FSK, PSK, QAM)	Кодування бітів через зміну амплітуди, частоти або фази	Висока швидкість, адаптація до різних середовищ	Складність приймача, чутливість до шумів	Wi-Fi, LTE, 5G, оптоволокно

Манчестерське кодування забезпечує чудову самосинхронізацію, адже передбачає зміну сигналу в кожному бітовому інтервалі. Проте його головний недолік – подвоєна смуга пропускання, що робить це кодування менш ефективним для високошвидкісних каналів.

У свою чергу, коди типу 4B/5B, 8B/10B зменшують навантаження на смугу пропускання порівняно з манчестерським, але забезпечують часті переходи для збереження синхронізації. Вони трохи ускладнені через таблиці відповідності, проте широко використовуються завдяки хорошому балансу між ефективністю та синхронізацією. Особливо вигідно вони виглядають в оптоволоконних каналах.

Методи типу MLT-3 і NRZI орієнтовані на зменшення частоти змін сигналу, що дозволяє знизити вимоги до смуги пропускання. Але при цьому синхронізація стає складнішою, особливо при довгих послідовностях нулів, що потребує додаткових рішень (наприклад, комбінування з кодуванням на кшталт 4B/5B). Таким чином, жодне кодування не є ідеальним – кожне має свої компроміси між швидкістю, надійністю та вартістю реалізації. Вибір методу залежить від конкретних вимог до мережі: середовища передавання, пропускної здатності, стійкості до шумів і вартості обладнання.

Контрольні запитання та задачі до теми 4

Перелік теоретичних запитань

1. Яке призначення кодування в локальних комп’ютерних мережах?
2. Чому важливо забезпечувати синхронізацію під час передачі даних?
3. Що таке інформаційне кодування і як воно впливає на якість зв’язку?
4. Які основні характеристики визначають ефективність методу кодування?
5. У чому полягає відмінність між самосинхронізованими і несамосинхронізованими кодами?
6. Як працює манчестерське кодування і в яких мережах воно застосовується?
7. Що таке кодування NRZ і NRZI, та які їхні особливості?
8. У чому переваги кодування типу 4B/5B та 8B/10B на фізичному рівні?

9. Яким чином кодування допомагає виявляти та виправляти помилки на канальному рівні OSI?

10. Які функції кодування реалізуються на канальному рівні у моделі OSI?

11. Як використання багаторівневих сигналів (MLT-3) впливає на смугу пропускання?

12. У чому полягає різниця між аналоговим і цифровим кодуванням у комп'ютерних мережах?

13. Які типи модуляції застосовуються для передачі цифрових даних у фізичному середовищі?

14. Як порівнюються різні методи кодування за критеріями пропускної здатності та складності реалізації?

15. Чому в локальних мережах рідко використовується аналогове кодування, попри його переваги?

Перелік задач для самостійного опрацювання

Практичне завдання 1.

У мережі використовується код 4B/5B. Скільки бітів буде передано по фізичному каналу при передачі 400 інформаційних бітів? Яка надмірність такого кодування у відсотках?

Практичне завдання 2.

Порівняйте смугу пропускання, необхідну для кодування NRZ, Manchester та 4B/5B при однаковій швидкості передачі 100 Мбіт/с. Який з варіантів є найбільш ефективним за цим параметром?

Практичне завдання 3.

Припустимо, що в системі використовується код MLT-3. Визначте максимальну частоту зміни сигналу при передачі послідовності з одних одиниць на швидкості 100 Мбіт/с.

Практичне завдання 4.

Сформуйте двійковий сигнал за кодуванням NRZI для наступної інформаційної послідовності: 11010011. Початковий рівень сигналу – низький. Побудуйте графік зміни сигналу.

Практичне завдання 5.

Порівняйте два методи кодування (наприклад, 8B/10B і Manchester) за наступними критеріями:

– надмірність кодування;

– синхронізація;

– вимоги до смуги пропускання;

– апаратна складність реалізації;

Оформіть результати порівняння у вигляді таблиці.

Практичне завдання 6 (тест).

Дайте відповіді на запитання (множинний вибір або одна правильна відповідь)

1. Що з наведеного є головною перевагою кодування Manchester?

А) Висока щільність кодування

Б) Простота апаратної реалізації

В) Надійна синхронізація

Г) Низька надмірність

2. Який код перетворює 8 інформаційних бітів у 10 передаваних?

А) 5B/6B

Б) NRZI

В) 4B/5B

Г) 8B/10B

3. У чому полягає принцип NRZ-кодування?

А) Зміна рівня при кожному біті

Б) Рівень змінюється лише при передачі «1»

В) Сигнал повертається до нуля після кожного біта

Г) Кожен біт передається двома імпульсами

Перелік використаної та рекомендованої літератури

4.1. Analog Devices [Електронний ресурс]. – Режим доступу: https://www.analog.com/media/en/technical-documentation/application-notes/60662154968765an746_0.pdf

4.2. 100BASE-TX TP-PMD [Електронний ресурс]. – Режим доступу: https://www.iol.unh.edu/sites/default/files/knowledgebase/fe/100BASE-TX_PMD.pdf

4.3. Implementation of 8B/10B encoder-decoder for Gigabit Ethernet Frame [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/6923049

Тема 5. Технології Ethernet

Ethernet – це одна з найпоширеніших технологій локальних мереж (LAN), яка стала стандартом де-факто для обміну даними між пристроями в межах обмеженого простору, таких як офіси, підприємства, кампуси чи домашні мережі.

Від моменту своєї появи у 1970-х роках технологія пройшла значну еволюцію: від оригінальної коаксіальної архітектури до сучасних волоконно-оптичних і високошвидкісних гігабітних рішень.

Основою успіху Ethernet стало поєднання простоти реалізації, масштабованості, економічної ефективності та сумісності з іншими мережевими технологіями. Ethernet працює в межах фізичного та канального рівнів моделі OSI, використовуючи різні методи кодування, модуляції та організації доступу до середовища передавання.

На сьогодні технології Ethernet охоплюють швидкості від 10 Мбіт/с до 400 Гбіт/с і вище, що робить їх універсальними як для домашніх користувачів, так і для дата-центрів.

5.1. Підрівні канального рівня LLC і MAC

Фізичний рівень моделі OSI відповідає за передавання бітів – найменших одиниць інформації – через фізичне середовище, таке як мідний кабель, оптоволокно чи радіохвилі. У контексті технології Ethernet цей рівень забезпечує основу для передавання цифрових сигналів, з яких формуються кадри канального рівня. Ethernet, що є найпоширенішим стандартом у світі для побудови локальних мереж, функціонує саме на фізичному та канальному рівнях, забезпечуючи надійний і уніфікований спосіб обміну даними між пристроями.

Ethernet належить до родини технологій, регламентованих стандартами IEEE 802.2 (для логічного управління каналом) і IEEE 802.3 (для фізичної реалізації та контролю доступу до середовища). Ці стандарти охоплюють різні рівні швидкості передавання даних, починаючи від 10 Мбіт/с (класичний Ethernet) і доходячи до 100 Гбіт/с, що забезпечує гнучкість і масштабованість у різних сферах застосування – від домашніх мереж до центрів обробки даних.

Канальний рівень у Ethernet поділяється на два логічних підрівні: управління логічним каналом LLC і контроль доступу до середовища MAC (рис. 5.1).

Підрівень LLC функціонує як посередник між програмним забезпеченням (наприклад, IP-протоколом) і апаратним забезпеченням. Він додає до пакета керуючу інформацію, необхідну для правильної доставки, і незалежно від типу пристрою реалізується у вигляді драйвера мережевої карти.

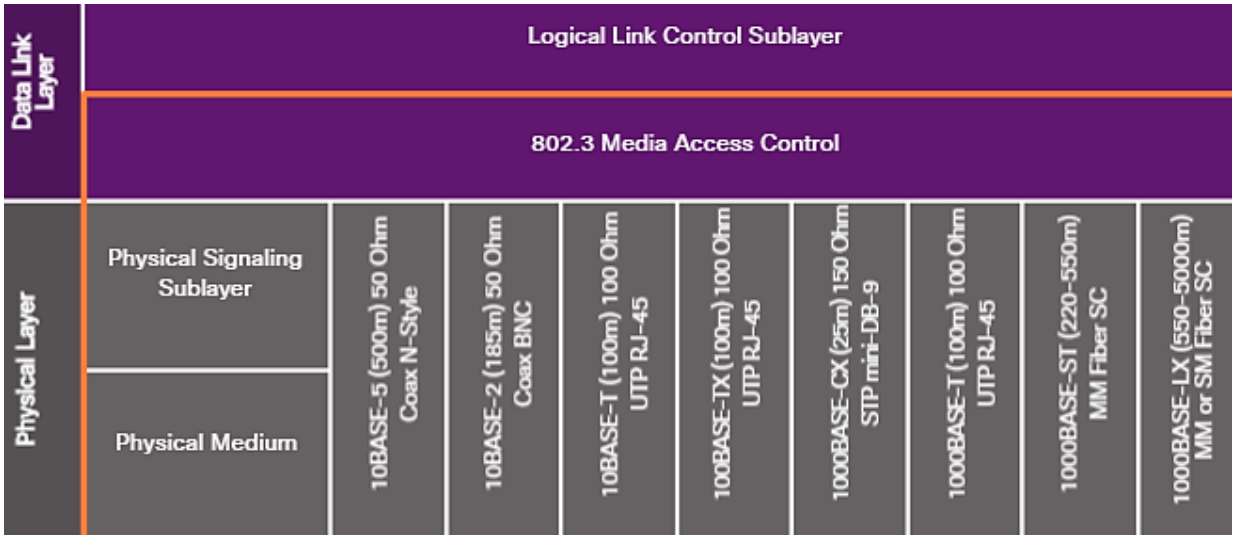


Рис. 5.1. Структура підрівнів канального рівня

Підрівень MAC реалізований апаратно (в мережевих адаптерах) і регламентується стандартом IEEE 802.3. Його ключові функції – інкапсуляція даних у кадри Ethernet і управління доступом до середовища передавання. При інкапсуляції додається заголовок, що містить MAC-адресу, і кінцеву частину з CRC-кодом для виявлення помилок. Також MAC забезпечує синхронізацію між передавальними та приймальними вузлами, що дозволяє чітко визначати межі кадру.

У підході CSMA/CD пристрій перевіряє, чи зайняте середовище, перед початком передачі. Якщо середовище вільне – передає, інакше чекає. У разі одночасної передачі двох вузлів виникає колізія, і дані потрібно передавати повторно. Проте сучасні комутатори і повнодуплексні з’єднання дозволяють уникнути таких ситуацій, що робить CSMA/CD фактично застарілим для дротових LAN.

Оскільки CSMA/CD не вимагає складних механізмів управління чергою доступу, цей метод є ефективним у мережах з низьким навантаженням. Проте в умовах великої кількості вузлів і високої інтенсивності трафіку зростає частота колізій, що знижує ефективність

роботи мережі. У бездротових мережах (Wi-Fi) натомість застосовується схожий принцип – CSMA/CA (запобігання колізіям), де перед передачею пристрій посилає повідомлення про намір використати середовище. Це дозволяє зменшити ймовірність конфлікту, особливо в умовах, коли неможливо точно визначити стан каналу.

Технологія Ethernet, хоч і бере початок у 70-х роках, досі залишається основою сучасних комп'ютерних мереж, її архітектура, заснована на розподіленому контролі доступу і простоті реалізації, була адаптована під нові вимоги: високі швидкості, комутація, бездротові середовища. Розуміння принципів роботи Ethernet, особливо функцій фізичного та канального рівнів, є фундаментальним для фахівців у сфері IT і мережевих технологій.

5.2. Призначення і структура MAC-адреси

Логічна топологія мережі Ethernet заснована на принципі спільного використання одного середовища передачі даних, подібно до шини з множинним доступом. У такій мережі усі пристрої приєднані до одного фізичного або логічного каналу, по якому передаються всі дані. Це означає, що кожне повідомлення, яке надсилається будь-яким пристроєм, потрапляє до всіх інших пристроїв у мережі. Усі вузли можуть бачити кожен кадр, що проходить по мережі. Виникає логічне питання: як окремий пристрій розуміє, що саме він є запланованим одержувачем кадру, не витрачаючи ресурси на повну обробку та деінкапсуляцію даних, щоб дістатися до IP-адреси?

Ця проблема стає ще більш критичною у великих мережах, де існує високий рівень трафіку. У таких умовах значна кількість кадрів постійно циркулює по мережі, і було б вкрай неефективно, якби кожен пристрій змушений був повністю аналізувати кожен кадр, незалежно від того, чи є він адресатом. Для усунення цього недоліку була впроваджена система унікальної ідентифікації пристроїв – MAC-адресація.

MAC-адреса (Media Access Control address) є унікальним ідентифікатором, що дозволяє точно визначити джерело та одержувача в межах однієї Ethernet-мережі. Ці адреси працюють на канальному рівні моделі OSI і мають фіксовану довжину – 48 біт.

У більш зручному для читання вигляді вони записуються як 12 шістнадцяткових символів, кожен з яких відповідає 4 бітам.

Унікальність MAC-адрес забезпечується завдяки спеціальним правилам, які регулюються Інститутом інженерів з електротехніки та електроніки (IEEE). Виробники мережевого обладнання мають бути зареєстровані в IEEE, щоб отримати унікальний ідентифікатор організації (OUI) – перші 24 біти (3 байти) MAC-адреси. Це дозволяє гарантувати, що жоден інший виробник не зможе використовувати цей самий ідентифікатор.

Згідно з правилами IEEE, кожна MAC-адреса складається з двох частин:

- перші 3 байти – це унікальний код виробника (OUI), який визначає компанію-виробника;
- останні 3 байти – це унікальний серійний номер, призначений виробником для конкретного пристрою.

Оскільки MAC-адреса фізично закодована у пристрої, її ще часто називають *апаратною адресою або BIA (Burned-In Address)*. Зазвичай вона зберігається в постійній пам'яті (ROM) мережевої плати і не змінюється протягом усього терміну служби пристрою.

Водночас сучасні операційні системи та мережеві адаптери дозволяють змінити MAC-адресу програмним способом. Це може бути корисним, наприклад, для обходу систем безпеки, які обмежують доступ до мережі на основі MAC-адреси. Проте така можливість знижує надійність фільтрації трафіку за MAC-адресами.

Кожен пристрій у мережі, який може передавати або приймати дані, – робочі станції, сервери, принтери, комутатори, маршрутизатори – отримує власну MAC-адресу. Вони використовують її для ідентифікації при обміні інформацією. Різні виробники можуть записувати ті ж самі MAC-адреси у різних форматах, наприклад:

00-00-05-9A-3C-78-00,

0000:05:9A:3C:78:00,

0000005.9A3C.7800.

При завантаженні комп'ютера мережева плата копіює MAC-адресу з ПЗП в оперативну пам'ять. Після цього вона використовується для формування заголовків Ethernet-кадрів.

Коли пристрій відправляє дані, він додає до кожного кадру адресу джерела (свою власну) і адресу призначення.

У процесі передачі кадру всі пристрої в мережі перевіряють

вказану в ньому MAC-адресу призначення. Ця перевірка здійснюється на підрівні MAC канального рівня.

Якщо MAC-адреса в кадрі збігається з апаратною адресою пристрою, кадр приймається і передається на обробку у вищі рівні моделі OSI. Якщо ж збігу немає – кадр просто ігнорується. Завдяки цьому значно зменшується навантаження на систему, оскільки пристрої обробляють лише ті кадри, які адресовані саме їм.

5.3. Структура кадру Ethernet

На канальному рівні моделі OSI структура Ethernet-кадру залишається практично однаковою незалежно від швидкості мережі. Це означає, що незалежно від того, наскільки швидко передаються дані (наприклад, 10 Мбіт/с або 1 Гбіт/с), спосіб формування кадру, який передає ці дані, залишається незмінним.

Основне призначення кадру Ethernet полягає в інкапсуляції повідомлення, тобто в «упакуванні» даних з третього рівня (мережевого рівня) у спеціальну структуру для подальшої передачі по фізичному середовищу.

Кожен Ethernet-кадр має певну структуру – набір обов'язкових полів, які називаються полями кадру. Вони поділяються між заголовком, самими даними та кінцевиком.

Існує два головні формати кадрів Ethernet: один визначений стандартом IEEE 802.3, інший – так званий Ethernet II (також відомий як DIX Ethernet, створений компаніями DEC, Intel та Xerox). Хоча вони схожі, ключовою різницею є наявність у стандарті IEEE 802.3 спеціального байту – початку роздільника кадру (SFD) – та використання поля «довжина» замість «типу».

Формат Ethernet II став стандартом де-факто для мереж TCP/IP, тому найчастіше зустрічається саме він.

Преамбула (7 байтів) та Початок роздільника кадру (1 байт) використовуються для синхронізації. Ці байти дозволяють приймаючим пристроям «почути» про наближення нового кадру, налаштуватися на його прийом та правильно зчитати інформацію (табл. 5.1).

MAC-адреса призначення (6 байтів) вказує, кому адресовано кадр. Кожен пристрій у мережі порівнює цю адресу зі своєю власною, щоб вирішити – приймати кадр чи ігнорувати його.

Таблиця 5.1. **Структура кадру Ethernet (IEEE 802.3)**

Поле	Розмір (байт)
Преамбула	7
Початок роздільника кадру	1
MAC-адреса призначення	6
MAC-адреса джерела	6
Довжина	2
Дані	46–1500
Контрольна послідовність	4

MAC-адреса джерела (6 байтів) повідомляє, хто відправив кадр – тобто яка мережева плата чи інтерфейс.

Поле довжини (2 байти) в стандарті IEEE 802.3 використовується для вказівки розміру поля даних. Якщо це значення перевищує 1536 (0x0600), то воно трактується як вказівка на тип протоколу (EtherType), що ближче до формату Ethernet II. Якщо ж значення менше або дорівнює 1500 (0x05DC), це типовий кадр IEEE 802.3.

Дані (від 46 до 1500 байтів) – це основна частина кадру, в якій міститься інформація з вищого рівня, зазвичай IP-пакет.

Якщо розмір переданих даних менший за 46 байтів, додаються спеціальні байти-заповнювачі (padding), щоб забезпечити мінімальну довжину кадру – 64 байти.

Контрольна послідовність кадру (FCS, 4 байти) – використовується для виявлення помилок під час передачі. Циклічний контроль надмірності (CRC) дозволяє виявити, чи не було спотворено дані в процесі надсилання.

Якщо обчислення CRC не співпадають – кадр вважається пошкодженим і відкидається.

У середовищі Ethernet використовуються різні типи MAC-адрес залежно від типу передавання.

Одноадресна MAC-адреса – це унікальна адреса, яка призначена конкретному пристрою. Вона використовується для направлення кадру від одного відправника до одного одержувача (рис. 5.2).

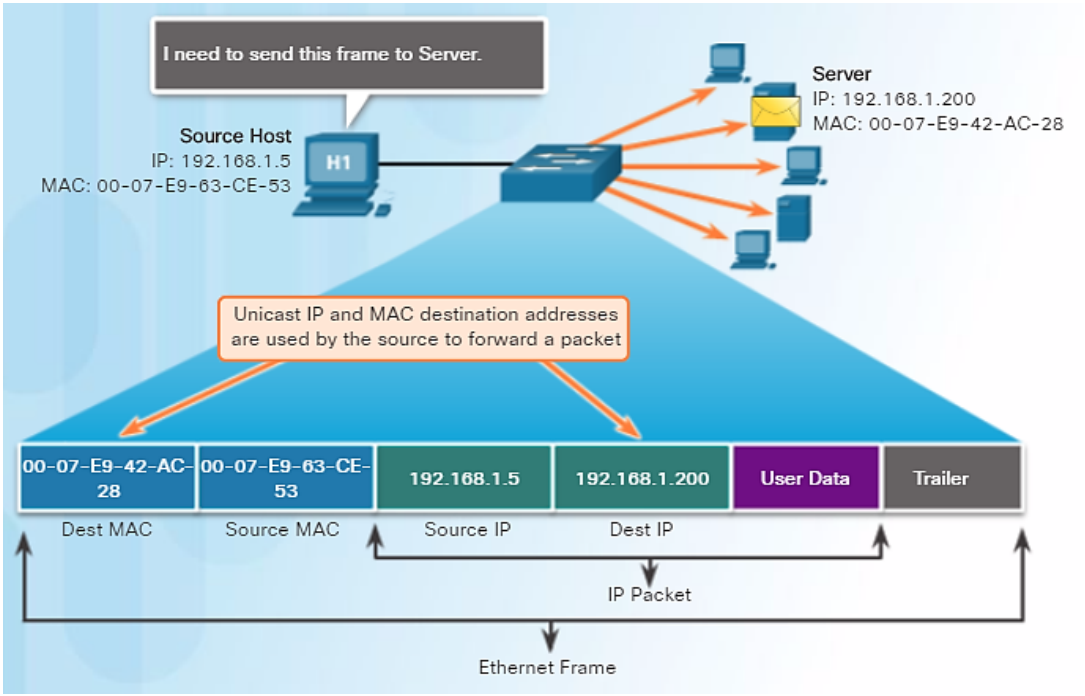


Рис. 5.2. Приклад одноадресної розсилки

Широкомовна (broadcast) MAC-адреса – спеціальна адреса, яка складається повністю з одиниць. Вона використовується, коли кадр потрібно доставити усім пристроям у мережі. Наприклад, протоколи ARP і DHCP покладаються на широкомовні запити для пошуку інших пристроїв у мережі (рис. 5.3).

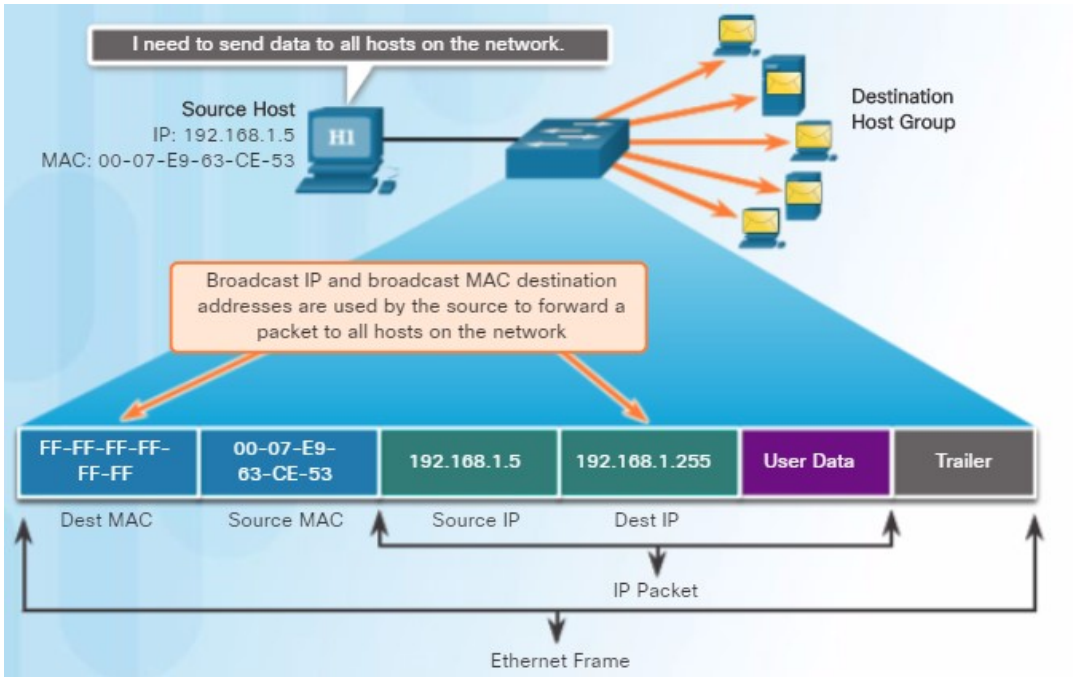


Рис. 5.3. Приклад широкомовної розсилки

Багатоадресна (multicast) MAC-адреса – дозволяє відправляти кадр групі пристроїв у мережі (рис. 5.4).

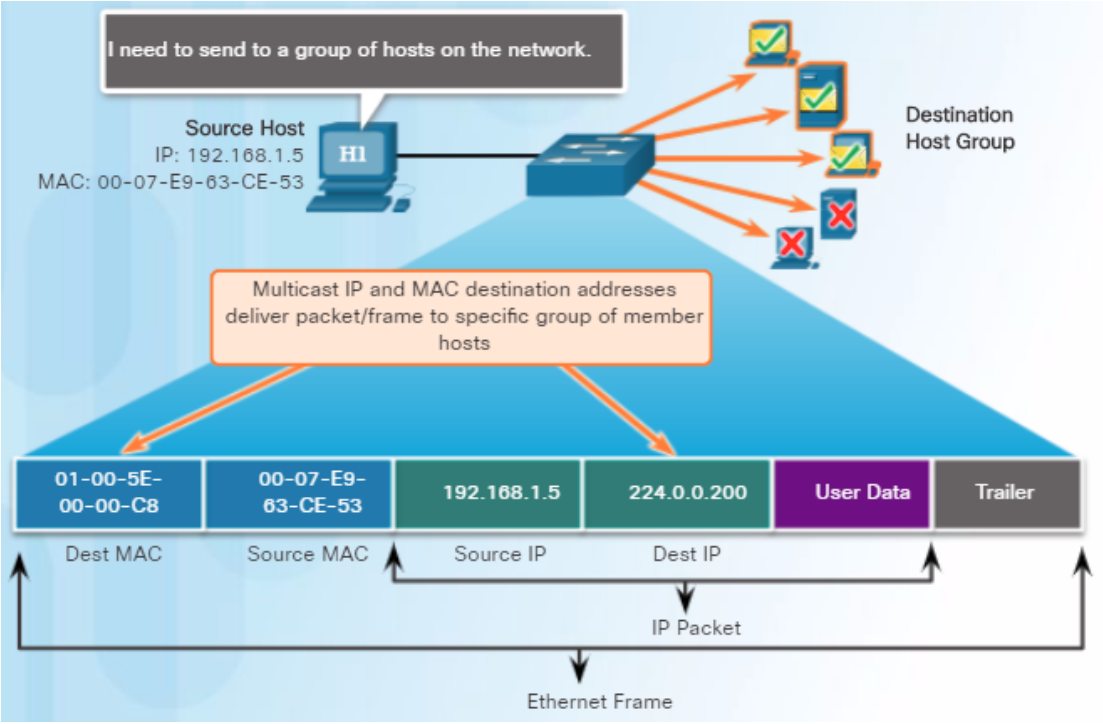


Рис. 5.4. Приклад багатоадресної розсилки

Наприклад, багатоадресна розсилка використовується під час онлайн-ігор або відеоконференцій, коли дані потрібно передати лише певному набору користувачів. Для таких кадрів IP-адреса отримувача лежить в діапазоні 224.0.0.0–239.255.255.255. Відповідна MAC-адреса формується зі спеціального префікса 01-00-5E, а решта байтів генерується з нижчих 23 біт IP-адреси.

У Windows MAC-адресу мережевого адаптера можна переглянути, скориставшись командою *ipconfig /all* у командному рядку.

5.4. MAC- і IP-адреси

У комп'ютерній мережі кожному пристрою (вузлу) призначаються дві головні адреси, які дозволяють ідентифікувати пристрій: MAC-адреса (фізична адреса) та IP-адреса (логічна адреса).

Ці адреси виконують різні, але взаємопов’язані функції.

Щоб краще зрозуміти їхню роль, можна провести аналогію з поштовим листом: MAC-адреса – це ім’я одержувача, а IP-адреса – його

поштова адреса. Щоб доставити лист, потрібно знати і те, і інше.

MAC-адреса (Media Access Control) – це унікальний апаратний ідентифікатор, що присвоюється мережевій платі пристрою під час її виготовлення. Вона «вшита» в обладнання і, як правило, не змінюється незалежно від того, в яку мережу підключено пристрій. Наприклад, ноутбук, підключений до Wi-Fi вдома або в кафе, матиме ту саму MAC-адресу. Ця адреса використовується для ідентифікації пристроїв у межах локальної мережі.

IP-адреса (Internet Protocol) – це логічна адреса, яка визначає розташування пристрою у мережі. Вона може змінюватися залежно від мережі, до якої підключено пристрій. Наприклад, телефон отримає одну IP-адресу у домашній Wi-Fi мережі і зовсім іншу – у мобільній мережі або в офісі. IP-адреси призначаються або вручну адміністратором, або автоматично через DHCP-сервер (Dynamic Host Configuration Protocol).

Коли пристрій хоче надіслати дані, він формує IP-пакет, який містить IP-адресу отримувача. Але для того, щоб дані дійшли до конкретного пристрою у локальній мережі, потрібно знати і MAC-адресу одержувача. Тут у справу вступає протокол ARP (Address Resolution Protocol) – він дозволяє визначити MAC-адресу за відомою IP-адресою. Наприклад, якщо комп’ютер хоче зв’язатися з принтером у мережі, він спочатку запитає через ARP: «Хто має IP-адресу 192.168.1.10?». У відповідь пристрій із цією IP-адресою повідомить свою MAC-адресу.

Коли передається IP-пакет, у локальній мережі він інкапсулюється в Ethernet-кадр, в якому вказуються MAC-адреса джерела і MAC-адреса призначення. Наприклад, якщо комп’ютер з MAC-адресою 00:1A:2B:3C:4D:5E відправляє файл принтеру з MAC-адресою 00:1A:2B:3C:4D:6F, ці адреси вказуються в Ethernet-кадрі.

Усі інші пристрої в мережі також отримають цей кадр, але тільки той, чия MAC-адреса збігається з адресою призначення, обробить повідомлення.

IP-адреса, у свою чергу, використовується маршрутизаторами, щоб визначити найкращий маршрут до вузла призначення у глобальній мережі. Наприклад, при зверненні до сайту www.cisco.com пристрій використовує DNS-сервер для визначення IP-адреси сайту (наприклад, 209.165.200.225).

Далі IP-пакети подорожують мережею до маршрутизатора, який знає, як доставити ці дані до кінцевого вузла. Але на кожному окремому кроці ця інформація все одно передається через MAC-адреси відповідних інтерфейсів, на базі локальних Ethernet-сегментів.

Таким чином, MAC-адреси працюють на канальному рівні (2 рівень моделі OSI), забезпечуючи доставку в межах локальної мережі, тоді як IP-адреси функціонують на мережевому рівні (3 рівень), забезпечуючи глобальну маршрутизацію даних. Обидва типи адрес є необхідними для коректної роботи мережі, оскільки кожен з них виконує свою унікальну і незамінну функцію.

5.5. Функції і принципи роботи роботи протоколу ARP

Щоб мережевий пристрій, наприклад комп'ютер, міг відправити дані в мережі Ethernet, йому необхідно вказати як свою MAC-адресу (фізичну), так і свою IP-адресу (логічну) у відповідних полях заголовка пакету. Крім того, пристрій повинен знати MAC- і IP-адреси одержувача. Наприклад, якщо користувач надсилає електронний лист, система має знати не лише ім'я домену (логічну адресу), а й де знаходиться фактично комп'ютер одержувача (фізичну адресу в межах мережі).

Навіть якщо IP-адреса призначення вже відома, пристрій відправника має дізнатися, яка MAC-адреса відповідає цій IP-адресі, аби сформувати кадр Ethernet. Саме цю задачу виконує протокол ARP.

ARP (Address Resolution Protocol) – протокол, який зіставляє IP-адреси з MAC-адресами.

ARP працює за допомогою спеціальних типів повідомлень у мережі – широкомовних (broadcast) та одноадресних (unicast).

Запит ARP – це широкомовне повідомлення, яке надсилається до всіх пристроїв локальної мережі з метою дізнатися MAC-адресу, яка відповідає заданій IP-адресі. У відповідь пристрій, який має відповідну IP-адресу, надсилає одноадресне повідомлення зі своєю MAC-адресою.

Основними функціями ARP є:

1. встановлення відповідностей між IP-адресами (зазвичай IPv4) та MAC-адресами;

2. зберігання цих відповідностей у спеціальній таблиці – таблиці ARP або кеші ARP.

Коли пристрій хоче відправити кадр у мережі, він спочатку звертається до своєї таблиці ARP, яка зберігається в оперативній пам'яті. Якщо в таблиці вже є відповідність IP-адреси одержувача до MAC-адреси, ця інформація використовується для інкапсуляції пакету в кадр. Наприклад, якщо комп'ютер вже обмінювався даними з маршрутизатором, MAC-адреса цього маршрутизатора, ймовірно, вже є в кеші ARP.

Існує два основні способи наповнення таблиці ARP.

1. *Пасивне спостереження за мережею*. Коли пристрій отримує кадри від інших пристроїв, він може автоматично запам'ятовувати IP- та MAC-адреси джерела.

2. *Активний запит ARP*. Якщо запису немає в таблиці, пристрій відправляє ARP-запит на адресу FFFF.FFFF.FFFF, який отримують усі пристрої в локальній мережі. Відповідь надсилає лише той вузол, чия IP-адреса відповідає запиту.

Якщо ARP-запит не дає результату – жоден пристрій не відповідає, то пакет не може бути інкапсульований, і він відкидається. У випадку, коли відправник – це маршрутизатор, він може надіслати ICMP-повідомлення про помилку назад до пристрою-джерела.

При створенні кадру система повинна визначити, чи знаходиться вузол призначення у тій же локальній мережі. Якщо так, то використовується MAC-адреса цього вузла. Якщо ж ні – кадр відправляється до шлюзу за замовчуванням (маршрутизатора), і використовується його MAC-адреса як адреса призначення. Наприклад, якщо комп'ютер намагається звернутись до сервера в іншій мережі, він відправить кадр маршрутизатору, який виступає посередником.

Термін зберігання записів у таблиці ARP обмежений і керується таймером. В ОС Windows, наприклад, записи зберігаються 2 хвилини, але якщо адреса знову використовується, цей час подовжується до 10 хвилин. Є також можливість вручну очистити кеш ARP за допомогою спеціальних команд. Наприклад, у Windows використовується команда *arp -a*, а в маршрутизаторах Cisco – *show ip arp* (рис. 5.5).

```
C:\> arp -a

Interface: 192.168.1.67 --- 0xa
  Internet Address      Physical Address      Type
  192.168.1.254         64-0f-29-0d-36-91     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 10.82.253.91 --- 0x10
  Internet Address      Physical Address      Type
  10.82.253.92          64-0f-29-0d-36-91     dynamic
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Рис. 5.5. Відображення таблиці ARP на ПК з ОС Windows

Запити ARP є широкомовними, тобто їх отримують усі пристрої в сегменті локальної мережі. У невеликих або корпоративних мережах це зазвичай не створює значного навантаження, але при одночасному запуску великої кількості пристроїв може спостерігатись тимчасове зниження продуктивності. Проте ARP має вразливості. Наприклад, зловмисник може виконати так званий ARP-спуфінг – надіслати фальшиві ARP-повідомлення, щоб змусити інші пристрої запам’ятати неправильну відповідність між IP і MAC-адресою. Це дозволяє перехоплювати або перенаправляти трафік (рис. 5.6).

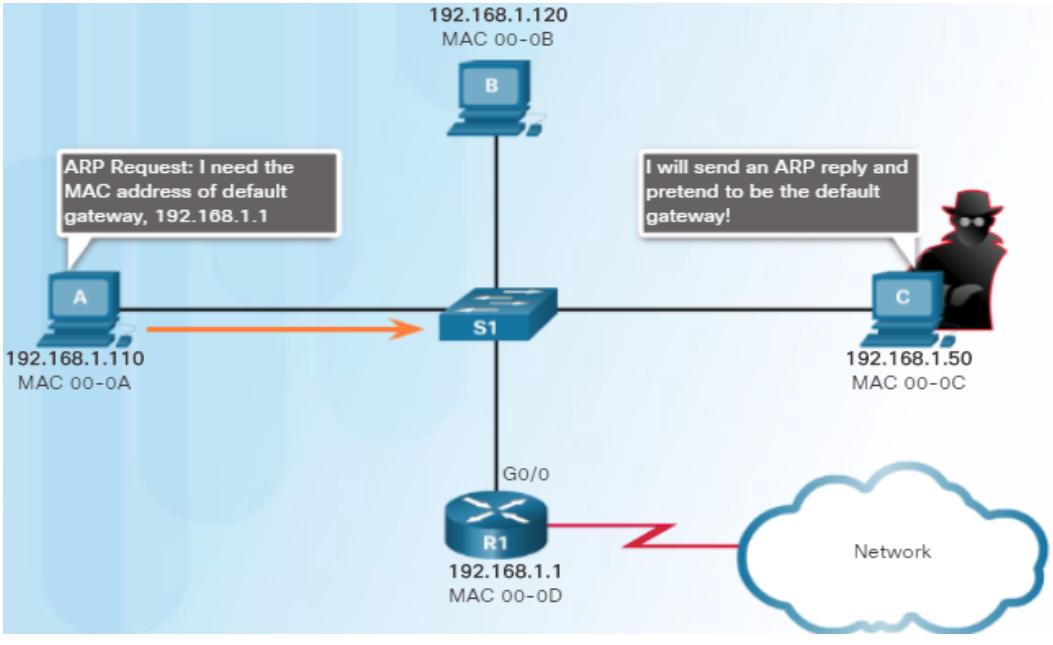


Рис. 5.6. Приклад атаки ARP-спуфінгу

Для запобігання таким атакам використовують статичні ARP-записи, які не змінюються автоматично. Також сучасні комутатори можуть зменшити кількість широкомовних кадрів і забезпечити ізоляцію одержувачів. Кожен порт комутатора створює окремий колізійний домен, що дозволяє ефективніше використовувати пропускну здатність та зменшує вплив ARP на всю мережу.

5.6. Таблиця MAC -адрес комутатора

Комутатори Ethernet використовують MAC-адреси для передачі даних у межах локальної мережі. Їх головне завдання – визначити, на який порт слід передати кадр, що надходить, аби він досяг вузла призначення. Для цього використовується внутрішня комутуюча матриця – апаратно-програмна система, яка керує переміщенням кадрів усередині комутатора, визначаючи оптимальний шлях передачі даних між портами.

Щоб комутатор міг правильно скеровувати кадри, він веде таблицю MAC-адрес (її ще називають таблицею комутації). У цій таблиці кожен запис містить MAC-адресу пристрою, порт, через який ця адреса доступна, а також час останньої активності. Коли комутатор отримує кадр, він читає MAC-адресу джерела з поля заголовка кадру. Якщо цієї MAC-адреси ще немає в таблиці, комутатор додає новий запис, зіставляючи цю адресу з портом, на якому було отримано кадр (рис. 5.7).

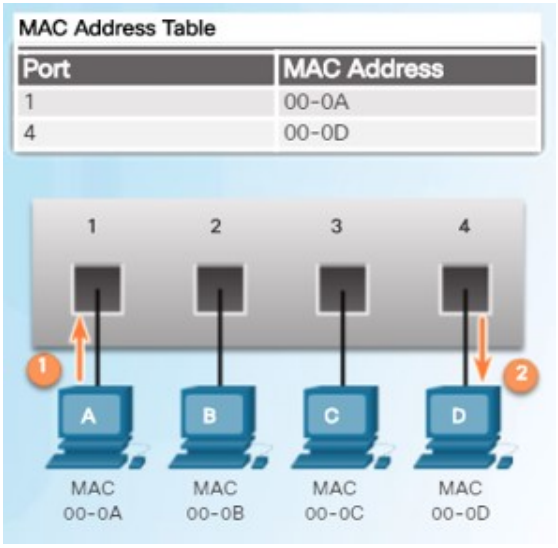


Рис. 5.7. Формування таблиці MAC-адрес на комутаторі

Коли комутатор приймає кадр, він перевіряє таблицю MAC-адрес, щоб знайти порт, через який можна доставити кадр вузлу призначення. Якщо MAC-адреса призначення є у таблиці, кадр пересилається безпосередньо на відповідний порт – відбувається одноадресна передача. Це дозволяє уникати непотрібного навантаження на інші порти і зменшує обсяг мережевого трафіку.

Якщо ж MAC-адреса призначення відсутня в таблиці, комутатор вдається до методу, що називається *флудинг*: кадр дублюється та розсилається на всі порти (крім порту прийому). Якщо вузол призначення існує в мережі, він відповість, і комутатор використає MAC-адресу відповіді для створення нового запису в таблиці. Відтак, у майбутньому кадри для цього вузла вже будуть направлятися конкретно через відповідний порт.

У складніших топологіях, де кілька комутаторів з'єднані між собою, на портах, які служать для з'єднання з іншими комутаторами, можуть накопичуватися кілька MAC-адрес. Це пов'язано з тим, що за одним портом може бути доступна велика кількість пристроїв – ті, що знаходяться у підключеній до цього порту підмережі. Таким чином, комутатор знає, що для досягнення певної MAC-адреси потрібно передати кадр через інший комутатор.

Усі ці механізми забезпечують ефективну передачу кадрів на канальному рівні, знижують навантаження на мережу і дають змогу швидко реагувати на зміни в топології або складі пристроїв, підключених до мережі.

5.7. Налаштування повнодуплексної та напівдуплексної передачі даних

Не дивлячись на те, що комутатори прозорі для мережевих протоколів і користувацьких додатків, вони здатні функціонувати в різних режимах, що може як позитивно, так і негативно відобразитися на пересилці кадрів Ethernet по мережі.

Одним з базових параметрів комутатора є дуплексний режим для кожного окремого порту, підключеного до кожного головного пристрою. Порт на комутаторі має бути налагоджений так, щоб співпадати з параметрами дуплексного режиму певного типу середовища передачі даних.

Для обміну даними в мережах Ethernet використовуються два типи налаштувань дуплексного режиму: напівдуплексний і повнодуплексний.

Напівдуплексний зв'язок використовує однонапрямлений потік даних, коли відправка і отримання даних не виконуються в один і той же час. Це подібно до використання рації, коли одноразово може говорити тільки одна людина. Якщо хто-небудь намагається говорити під час розмови іншої людини, відбувається колізія. В результаті при напівдуплексному зв'язку використовується множинний доступ з визначенням колізій, що дозволяє понизити вірогідність появи колізій і виявити їх у разі виникнення. При напівдуплексному зв'язку можливе зниження продуктивності, викликане постійним перебуванням в режимі очікування, оскільки дані можуть передаватися одночасно тільки в одному напрямі. Напівдуплексні з'єднання, як правило, зустрічаються на старішому устаткуванні, наприклад, на концентраторах (*концентратор* або *хаб*) – це мережевий пристрій, який з'єднує кілька комп'ютерів у мережі та передає отримані дані на всі порти без розрізнення адресата).

При повнодуплексному зв'язку потік даних передається в обидві сторони, що дозволяє одночасно відправляти і отримувати інформацію. Підтримка двосторонньої передачі даних підвищує продуктивність за рахунок скорочення часу очікування між передачами. Більшість мережевих адаптерів Ethernet, що продаються сьогодні (Fast Ethernet і Gigabit Ethernet) працюють в повнодуплексному режимі. У повнодуплексному режимі детектор колізій відключений. При цьому унеможливлено зіткнення кадрів, що пересилаються двома пов'язаними кінцевими вузлами, оскільки ці вузли використовують два окремі канали зв'язку в мережевому кабелі.

Кожне повнодуплексне з'єднання використовує тільки один порт. Повнодуплексним з'єднанням потрібен комутатор, який підтримує повнодуплексний режим, або пряме підключення між двома вузлами, кожен з яких підтримує повнодуплексну передачу даних. Вузли, які безпосередньо підключені до виділеного порту комутатора за допомогою мережевих адаптерів, що підтримують повнодуплексний зв'язок, повинні підключатися до портів комутатора, налагоджених для роботи в повнодуплексному режимі.

5.8. Порівняння комутації рівня 2 і рівня 3

Комутатори рівня 2 працюють на канальному рівні моделі OSI. Вони здійснюють комутацію та фільтрацію трафіку, використовуючи MAC-адреси пристроїв для визначення, на який порт передати кадр.

Цей тип комутаторів не має уявлення про IP-адреси, підмережі чи маршрутизацію. Якщо пристрої знаходяться в різних IP-підмережах, то комутатор рівня 2 не може самостійно передати дані між ними – для цього потрібен маршрутизатор, який працює на мережевому рівні і має змогу ухвалювати рішення на основі IP-адрес.

Наприклад, Cisco Catalyst 2960 – типовий комутатор рівня 2.

Комутатори рівня 3 поєднують функції традиційних комутаторів і маршрутизаторів. Вони здатні працювати як на канальному, так і на мережевому рівні OSI. Тобто, крім комутації кадрів за MAC-адресами, вони також можуть маршрутизувати пакети на основі IP-адрес.

Замість того, щоб просто визначати, яка MAC-адреса закріплена за певним портом, комутатори рівня 3 можуть аналізувати IP-адреси, визначати підмережі та пересилати IP-пакети між різними VLAN або мережами без потреби у зовнішньому маршрутизаторі.

Ці комутатори зазвичай мають апаратне прискорення маршрутизації, що дозволяє виконувати переадресацію пакетів майже з тією ж швидкістю, з якою виконується звичайна комутація.

Наприклад, Cisco Catalyst 3560 – комутатор рівня 3.

Комутатори рівня 3 підтримують різні інтерфейси, що дозволяють здійснювати маршрутизацію IP-пакетів.

SVI (Switch Virtual Interface) – це логічний інтерфейс, який асоціюється з певною VLAN. Він дає змогу здійснювати маршрутизацію між VLAN на самому комутаторі, без зовнішнього маршрутизатора.

Маршрутизований порт (Routed Port) – це фізичний порт, який налаштований як інтерфейс маршрутизатора, а не як звичайний порт комутатора. На такий порт можна призначити IP-адресу і використовувати його для з'єднання з іншими мережами.

EtherChannel рівня 3 – це логічний інтерфейс, який об'єднує кілька фізичних маршрутизованих портів у групу. Цей інтерфейс забезпечує більшу пропускну здатність і резервування, працюючи як один логічний маршрутизований інтерфейс.

Отже, комутатори рівня 3 значно розширюють функціональність порівняно з рівнем 2, дозволяючи:

- здійснювати маршрутизацію між підмережами;
- зменшувати потребу у виділених маршрутизаторах в LAN;
- підтримувати складніші топології з високою швидкістю обробки трафіку.

Це робить їх ідеальними для великих корпоративних мереж, де потрібна висока швидкість комутації разом із можливістю маршрутизації між численними підмережами.

Контрольні запитання та задачі до теми 5

Перелік теоретичних запитань

1. Які складові має канальний рівень OSI?
2. У чому полягає принцип дії методів доступу до середовища CSMA, CSMA/CD та CSMA/CA?
3. Яке призначення має MAC-адреса і яка її будова?
4. Які правила слід дотримуватися під час формування MAC-адрес?
5. У якому форматі зазвичай записується MAC-адреса?
6. Які існують різновиди MAC-адрес?
7. Які поля містить кадр Ethernet і яка їх функція?
8. Поясніть відмінності в структурі та використанні MAC-адреси і IP-адреси.
9. Яку роль виконує протокол ARP і як він працює?
10. Що таке ARP-спуфінг і чим він небезпечний для мережі?
11. Навіщо комутатору потрібна таблиця MAC-адрес і як вона використовується?
12. Чим відрізняється повнодуплексна передача даних від напівдуплексної?
13. Яка головна різниця між функціональністю комутаторів 2-го і 3-го рівнів?

Перелік задач для самостійного опрацювання

Завдання 1. Аналіз кадру Ethernet.

Розпишіть поля кадру Ethernet, вкажіть їхнє призначення та значення за замовчуванням.

Завдання 2. Визначення типу MAC-адреси і таблиці MAC-адрес.

MAC-адреса 01:00:5E:00:00:FB – визначте, до якого типу вона належить (уні-, мульти-, або широкомовна). В середовищі моделювання Cisco Packet Tracer створіть мережу з 4 вузлів і одного комутатора. На основі топології створіть приклад заповненої таблиці MAC-адрес комутатора.

Завдання 3 (тест).

Дайте відповіді на запитання.

1. Що визначає MAC-адреса?

A) IP-розташування пристрою
B) Фізичну адресу мережевого інтерфейсу
C) Тип трафіку
D) Швидкість з'єднання

2. Що робить комутатор, коли не знає MAC-адресу отримувача?

A) Ігнорує кадр
B) Відправляє кадр на всі порти
C) Відправляє кадр на порт за замовчуванням
D) Надсилає кадр до маршрутизатора

3. ARP-протокол використовується для:

A) Шифрування MAC-адрес
B) Пошуку MAC-адреси за IP-адресою
C) Надсилання кадрів між VLAN
D) Розділення трафіку між маршрутизаторами.

Перелік використаної та рекомендованої літератури

5.1. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

5.2. IEEE Standard for Ethernet [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/9844436

5.3. MAC Address [Електронний ресурс]. – Режим доступу: https://www.cloudns.net/blog/mac-address-everything-you-need-to-know/

5.4. What is MAC Address and How to find it [Електронний ресурс]. – Режим доступу: https://www.pynetlabs.com/what-is-mac-address/

Тема 6. Мережевий рівень

Мережевий рівень є третім рівнем у моделі OSI (Open Systems Interconnection) і виконує ключову роль у процесі передачі даних між пристроями, які можуть бути розташовані в різних мережах.

Основне завдання цього рівня – забезпечити маршрутизацію пакетів даних від джерела до призначення, незалежно від їх фізичного розташування, а також вибір оптимального маршруту через складну мережеву інфраструктуру.

На мережевому рівні здійснюється логічна адресація, що дозволяє ідентифікувати кожен пристрій у мережі. При цьому найпоширенішим типом логічної адресації є IP-адресація (IPv4 або IPv6).

Мережевий рівень також відповідає за фрагментацію пакетів, обробку помилок у маршрутизації, контроль перевантаження мережі та взаємодію з протоколами, такими як ICMP, ARP та протоколами маршрутизації (наприклад, OSPF, RIP, BGP).

Таким чином, мережевий рівень забезпечує критично важливу функцію – передачу даних між незалежними мережами, підтримуючи надійність, ефективність та масштабованість комунікацій у сучасних комп’ютерних системах.

6.1. Процеси і протоколи мережевого рівня

У сучасних комп’ютерних мережах застосунки, запущені на одному пристрої, повинні мати змогу обмінюватися даними з аналогічними сервісами на інших пристроях, часто розташованих в різних мережах або навіть країнах. Щоб забезпечити швидку та ефективну доставку цих даних незалежно від типу середовища чи відстані, використовується мережевий рівень (рівень 3) моделі OSI.

Мережевий рівень відповідає за доставку пакетів даних між пристроями в мережі. Його основні завдання – логічна адресація, маршрутизація, інкапсуляція/деінкапсуляція даних і фрагментація, коли це необхідно. Для передачі даних мережевий рівень використовує протоколи, головним із яких є IP – Internet Protocol.

Щоб передача даних була можлива, мережевий рівень виконує кілька важливих процесів.

Кожен пристрій у мережі отримує унікальну IP-адресу. Ця адреса

дозволяє іншим пристроям знаходити і надсилати дані саме туди, куди потрібно. Наприклад, комп'ютер з IP-адресою 192.168.1.10 може передати дані на сервер із адресою 172.16.0.5, і мережевий рівень забезпечить, щоб пакети дійшли до потрібного пристрою.

Дані, отримані з транспортного рівня (наприклад, TCP-сегмент), інкапсулюються на мережевому рівні в IP-пакет. До них додається IP-заголовок із вказанням адреси відправника і отримувача. Це нагадує відправку листа: ви вкладаєте повідомлення в конверт, на якому зазначено, кому його доставити.

Після створення IP-пакету мережевий рівень визначає маршрут до пристрою-призначення. Якщо цей пристрій знаходиться в іншій мережі, дані передаються через маршрутизатори. Наприклад, якщо користувач у Києві хоче звернутися до сервера в Лондоні, його запит пройде через кілька мережевих вузлів – це і є процес маршрутизації.

Коли пакет прибуває до адресата, мережевий рівень перевіряє, чи збігається IP-адреса в заголовку з адресою пристрою. Якщо так, IP-заголовок видаляється, а дані передаються на транспортний рівень для подальшої обробки.

У більшості мереж використовуються два основних протоколи мережевого рівня – IPv4 та IPv6. Вони виконують однакові функції, але мають різну структуру адрес і можливості. Наприклад, IPv4 використовує 32-бітні адреси (як 192.168.0.1), а IPv6 – 128-бітні (2001:0db8:85a3::8a2e:0370:7334), що забезпечує більший простір для адресації.

IP є протоколом без встановлення з'єднання, тобто він не перевіряє, чи готовий одержувач прийняти дані. Він також не гарантує доставку пакетів – це ненадійний протокол. Якщо дані загубляться або пошкодяться, IP не забезпечує їх повторну відправку – за це відповідає транспортний рівень, зокрема, протокол TCP.

Відсутність механізмів підтвердження доставки в IP дозволяє йому працювати дуже швидко та з мінімальним навантаженням. Це робить його придатним для великого обсягу різнорідного трафіку. Наприклад, відеопотік чи онлайн-ігри часто використовують ненадійні протоколи (UDP), що працюють поверх IP, для мінімізації затримок.

Ще один важливий процес на мережевому рівні – фрагментація. Якщо пакет перевищує максимальний розмір, дозволений середовищем

передачі (MTU), маршрутизатор може розділити його на частини. Це дозволяє адаптувати передавання до обмежень фізичних мереж.

Наприклад, якщо MTU у мережі – 1500 байт, а пакет – 2000 байт, маршрутизатор розіб'є його на два пакети і додасть спеціальні заголовки, щоб приймач зміг зібрати їх назад у вихідні дані.

Мережевий рівень взаємодіє з транспортним і канальним рівнями. Він отримує дані з TCP/UDP, інкапсулює їх у пакети, а потім передає на канальний рівень, який займається доставкою всередині локальної мережі. Завдяки цьому забезпечується модульність, гнучкість і масштабованість мережі.

Коли ми відкриваємо вебсайт, браузер надсилає запит до вебсервера. Запит спочатку проходить через транспортний рівень, далі – через мережевий, де отримує IP-заголовок. Потім він передається через кілька маршрутизаторів до вебсервера. Після обробки відповіді дані повертаються назад до нашого комп'ютера аналогічним шляхом. Цей обмін забезпечується саме механізмами мережевого рівня.

6.2. Протокол IPv4

Протокол IPv4 (Internet Protocol версії 4) був офіційно впроваджений у 1983 році в рамках мережі ARPANET – першої мережі, яка стала прообразом сучасної мережі Інтернет. З того часу IPv4 став основним протоколом для маршрутизації даних у глобальній мережі. Його широке застосування пояснюється простотою реалізації, ефективністю та підтримкою численними пристроями і сервісами по всьому світу.

Наприклад, коли ми переглядаємо вебсайт, відправляємо електронного листа або передаємо файл, усі ці дії, швидше за все, опираються саме на протокол IPv4.

Розглянемо структуру пакету IPv4.

Кожен пакет у протоколі IPv4 має чітку структуру, що складається з двох основних частин:

- *заголовок IP (IP Header)* – ця частина містить службову інформацію про пакет, зокрема, його призначення, джерело, тип вмісту та інші параметри, необхідні для коректної доставки;
- *корисне навантаження (Payload)* – це сегмент транспортного рівня (наприклад, TCP або UDP) разом з фактичними даними, які

користувач намагається передати (наприклад, вміст електронного листа або частина відео).

Розглянемо ключові поля заголовка IPv4 (рис. 6.1).

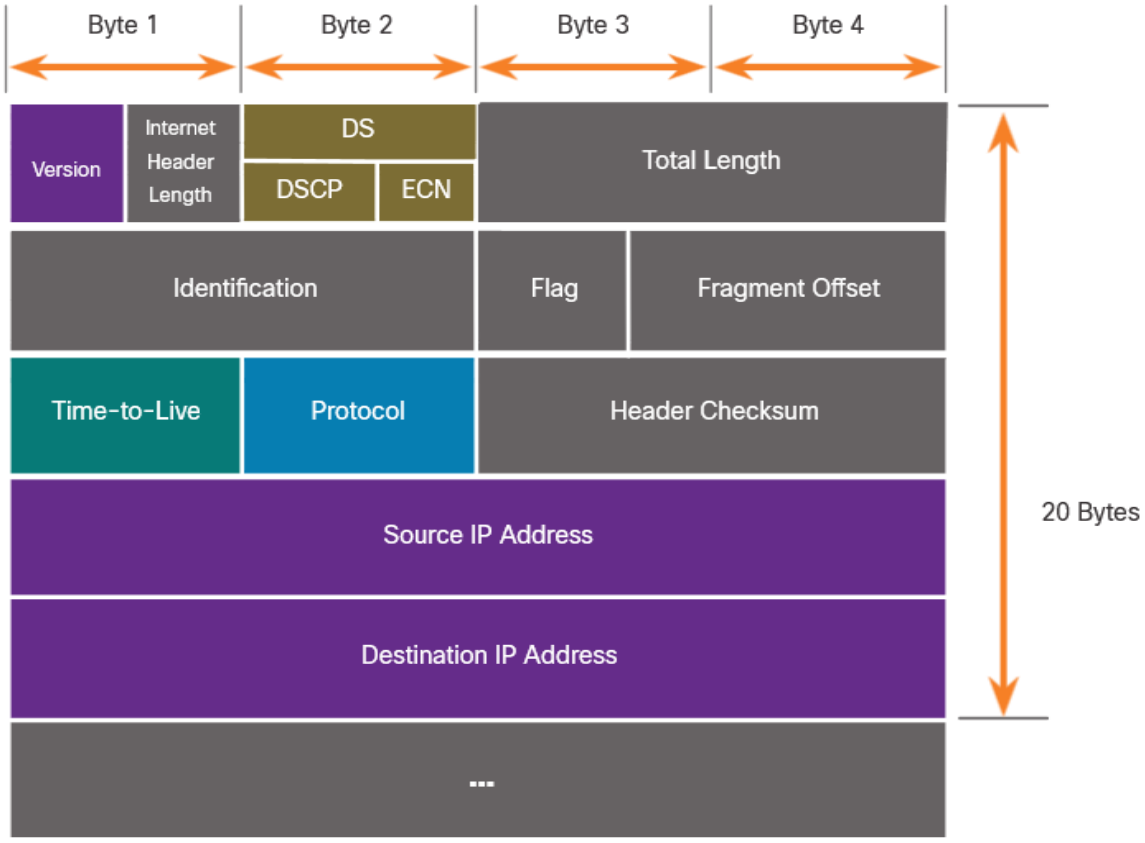


Рис. 6.1. Поля заголовка пакета IPv4

IPv4 має декілька важливих полів, що дозволяють маршрутизаторам та іншим мережевим пристроям правильно обробляти і доставляти пакети.

Версія – це поле займає 4 біти та вказує, яку версію IP-протоколу використовує пакет. Для IPv4 це значення завжди буде 0100 (тобто 4 у двійковій системі).

Диференційовані сервіси (DS) – 8-бітове поле, яке дозволяє встановлювати пріоритети для різних типів трафіку. Наприклад, відеоконференція може отримати вищий пріоритет, ніж завантаження файлів. Частина цього поля також використовується для запобігання перевантаженням у мережі (Explicit Congestion Notification).

Час існування (TTL, Time to Live) – значення, що обмежує кількість переходів (hop), які може здійснити пакет у мережі.

Кожного разу, коли пакет проходить через маршрутизатор, TTL зменшується на 1. Якщо TTL досягає нуля, пакет знищується. Це запобігає безкінечному циркулюванню помилкових пакетів у мережі. Наприклад, команда *traceroute* використовує TTL, щоб визначити маршрут проходження пакетів.

Протокол – вказує, які дані містяться в корисному навантаженні. Наприклад, якщо це TCP-сегмент, то поле матиме значення 6, для UDP – 17, для ICMP – 1.

IP-адреса джерела і призначення – це поля по 32 біти кожне, які вказують адресу пристрою, що відправив пакет, і адресу пристрою, якому призначений пакет. Наприклад, якщо ми переглядаєте сайт, наш комп'ютер буде джерелом, а сервер сайту – призначенням.

Розглянемо інші поля для перевірки та ідентифікації пакетів.

Довжина заголовка (IHL) – визначає розмір заголовка у 32-бітових словах. Мінімально – 5 (20 байт), максимально – 15 (60 байт), залежно від наявності додаткових опцій.

Загальна довжина – повний розмір пакету, включаючи заголовок і дані. Максимальна довжина пакету – 65 535 байт, хоча на практиці обмеження можуть бути меншими через обмеження середовища передачі.

Контрольна сума заголовка – використовується для виявлення помилок у заголовку. Якщо контрольна сума не співпадає із обчисленим значенням, пакет відкидається як пошкоджений.

Хоча протокол IPv4 є дуже популярним, проте він має суттєві обмеження.

IPv4 підтримує приблизно 4,3 мільярда унікальних адрес. Враховуючи стрімке зростання Інтернету речей, смартфонів, ПК та інших підключених пристроїв, цього ресурсу вже недостатньо. Наприклад, кожен будинок може мати десятки пристроїв, які потребують IP-адреси.

Збільшення розмірів таблиць маршрутизації – з розширенням Інтернету зростає кількість мереж і маршрутів, які повинні зберігатися у таблицях маршрутизаторів. Це вимагає більше оперативної пам'яті й обчислювальних ресурсів.

Через нестачу адрес часто використовується технологія *NAT (Network Address Translation – перетворення адрес)*, що дозволяє

кільком пристроям використовувати одну публічну IP-адресу. Однак NAT ускладнює встановлення прямих з'єднань, які потрібні, наприклад, для онлайн-ігор або деяких VoIP-застосунків.

Протокол IPv4 залишається основою функціонування Інтернету, але його технічні обмеження стали поштовхом для створення IPv6 – новішої версії протоколу з розширеними можливостями та практично необмеженим простором адрес.

6.3. Протокол IPv6

На початку 1990-х років Інженерна група з розвитку Інтернету (IETF) усвідомила обмеження протоколу IPv4, зокрема вичерпність адресного простору, і почала розробку нового протоколу – IPv6. Основною метою IPv6 стало усунення недоліків попередньої версії, забезпечення масштабованості, довговічності та підтримка сучасних і майбутніх вимог до мережевої інфраструктури. Протокол IPv6 запроваджує низку технічних і функціональних покращень. Найсуттєвіше з них – розширення адресного простору. Якщо IPv4 оперує 32-бітовими адресами (що дає приблизно 4,3 мільярда можливих унікальних адрес, з яких реально використовуються близько 3,7 мільярда), то IPv6 використовує 128-бітову ієрархічну адресацію. Це забезпечує $\approx 3{,}4\times10^{38}$ унікальних IP-адрес (340 ундециліонів), що майже відповідає кількості піщинок на планеті Земля. Така кількість адрес дає змогу кожному пристрою мати унікальну глобальну адресу, уникаючи необхідності трансляції мережевих адрес (NAT), яка широко застосовувалася в IPv4. Відмова від NAT спрощує організацію наскрізного з'єднання, що критично для низки сучасних сервісів, таких як VoIP, потокове відео та P2P-додатки.

Ще одним важливим покращенням є оптимізована обробка пакетів завдяки спрощеній структурі заголовка IPv6. Заголовок IPv6 має фіксовану довжину в 40 байт, у порівнянні з мінімум 20 байт в IPv4, які могли збільшуватися до 60 байт залежно від кількості опцій.

У IPv6 заголовок містить лише 8 основних полів, що знижує навантаження на маршрутизатори та прискорює обробку трафіку. Зокрема, усунено необхідність обчислення контрольної суми на кожному вузлі маршруту, що скорочує час обробки (рис. 6.2).

Додатково, в IPv6 передбачено нове поле – "Мітка потоку" (Flow

Label), яке дозволяє ідентифікувати потоки даних для додатків реального часу, забезпечуючи маршрутизацію з мінімальною затримкою без потреби аналізу транспортного рівня.

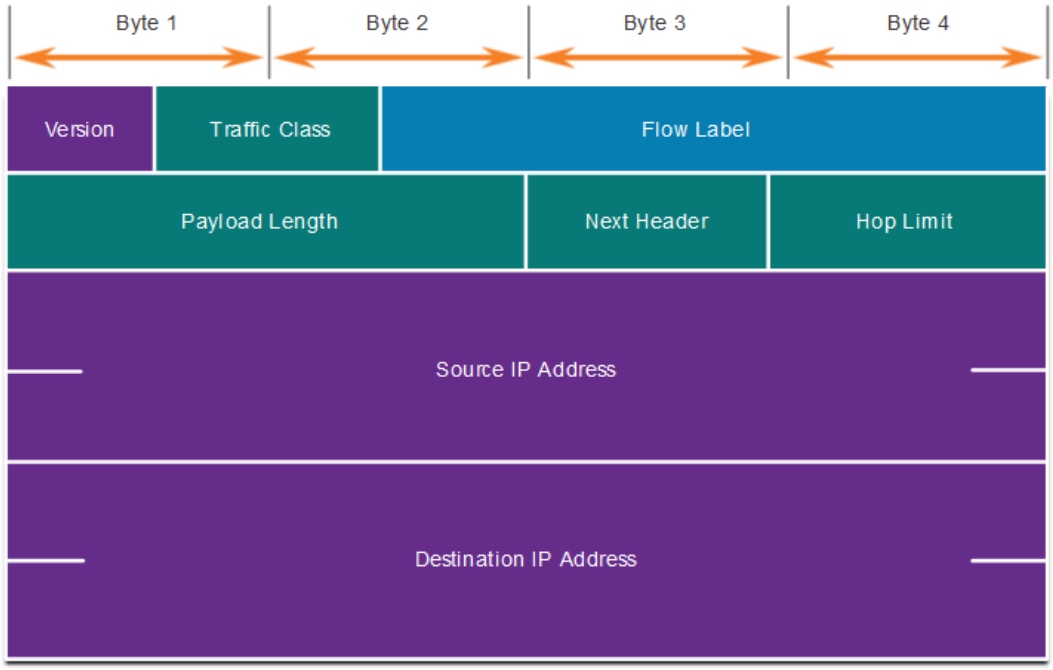


Рис. 6.2. Поля заголовка пакета IPv4 у заголовку пакета IPv6

У заголовку IPv6 використовуються такі основні поля: поле "Версія" (4 біти) завжди має значення 0110, що вказує на шосту версію протоколу; "Клас трафіку" (8 біт), що відповідає полі DS у IPv4, дозволяє реалізувати механізми якості обслуговування (QoS) через DSCP і управління перевантаженням через ECN; "Мітка потоку" (20 біт) визначає спеціальні потоки даних; "Довжина корисного навантаження" (16 біт) вказує обсяг даних після заголовка; "Наступний заголовок" (8 біт) вказує тип корисного навантаження або наявність розширених заголовків; "Межа переходу" (8 біт) – еквівалент TTL у IPv4 – зменшується на кожному маршрутизаторі; "Адреса джерела" та "Адреса призначення" (по 128 біт кожна) вказують на відправника і одержувача пакету відповідно.

Крім основного заголовка, IPv6 дозволяє додавати розширені заголовки, які надають додаткову інформацію – для фрагментації, безпеки, підтримки мобільності тощо. Ці заголовки розташовуються між основним заголовком і корисним навантаженням та є

опціональними, що забезпечує гнучкість і розширюваність протоколу.

IPv6-адреси подаються у шістнадцятковому форматі та розділяються двокрапками.

Приклад: 2001:0db8:85a3:0000:0000:8a2e:0370:7334.

Для зручності допускається скорочення нульових блоків і послідовностей нулів, наприклад: 2001:db8:85a3::8a2e:370:7334. Це дозволяє компактно представити довгі адреси без втрати інформаційного змісту. Таким чином, IPv6 не лише розширює технічні можливості мереж, а й враховує проблеми, з якими стикаються сучасні інтернет-системи, забезпечуючи основу для подальшого розвитку глобального зв'язку.

6.4. Шлюз за замовчуванням

Шлюз за замовчуванням – це пристрій, який забезпечує маршрутизацію трафіку з локальної мережі до зовнішніх, віддалених мереж. У більшості домашніх мереж або малих підприємств роль шлюзу зазвичай виконує маршрутизатор, що з'єднує локальну мережу з Інтернетом. Коли вузол намагається відправити пакет до пристрою, який знаходиться поза межами його власної IP-мережі, він не може доставити цей пакет безпосередньо. У таких випадках пакет передається шлюзу за замовчуванням, оскільки саме він має інформацію про зовнішні маршрути, збережену у своїй таблиці маршрутизації.

Вузли локальної мережі, навпаки, зберігають лише мінімальну інформацію про безпосередньо підключені мережі, тому вони не здатні самостійно здійснювати маршрутизацію за межами локальної підмережі. Таблиця маршрутизації, яка зберігається в оперативній пам'яті маршрутизатора, містить дані про наявні маршрути як до локальних, так і до віддалених мереж.

Ця таблиця дозволяє маршрутизатору швидко і ефективно визначати оптимальний шлях для доставки пакетів до вузлів призначення. Однак і сам вузол має локальну таблицю маршрутизації, яка дозволяє йому приймати рішення про те, чи потрібно направити пакет безпосередньо в локальну мережу, чи скористатися шлюзом за замовчуванням. У типовій локальній таблиці вузла міститься запис про маршрут до інтерфейсу *loopback* (127.0.0.1), запис про маршрут до локальної мережі, до якої підключений вузол, а також запис про

маршрут за замовчуванням. Саме останній запис визначає, куди слід направляти всі пакети, призначені для мереж, про які вузол не має конкретної інформації.

Адреса шлюзу за замовчуванням – це IP-адреса інтерфейсу маршрутизатора, який з'єднаний з локальною мережею. Цю адресу можна налаштувати вручну на кожному вузлі, або вона може бути надана автоматично, наприклад, через DHCP. Важливо зазначити, що шлюз за замовчуванням використовується лише у тому випадку, коли вузол намагається встановити з'єднання з віддаленим ресурсом.

Якщо пристрій, наприклад мережевий принтер або сканер, працює виключно в межах локальної мережі, налаштування шлюзу за замовчуванням є необов'язковим. Якщо на такому пристрої налаштовано лише IP-адресу і маску підмережі, цього цілком достатньо для взаємодії з іншими локальними вузлами. Більше того, навмисне виключення адреси шлюзу за замовчуванням може бути заходом безпеки: пристрій не зможе підключитися до Інтернету і, відповідно, не стане об'єктом потенційних зовнішніх атак.

Таким чином, налаштування шлюзу за замовчуванням слід виконувати з урахуванням функцій і потреб конкретного пристрою.

6.5. Таблиці маршрутизації

На комп'ютерах під управлінням ОС Windows для перегляду таблиці маршрутизації можна скористатися командами *route print* або *netstat -r* (рис. 6.3).

```
C:\Users\PC1> netstat -r

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    192.168.10.1   192.168.10.10      25
        127.0.0.0        255.0.0.0         On-link         127.0.0.1     306
        127.0.0.1  255.255.255.255         On-link         127.0.0.1     306
  127.255.255.255  255.255.255.255         On-link         127.0.0.1     306
    192.168.10.0    255.255.255.0         On-link    192.168.10.10     281
   192.168.10.10  255.255.255.255         On-link    192.168.10.10     281
  192.168.10.255  255.255.255.255         On-link    192.168.10.10     281
        224.0.0.0        240.0.0.0         On-link         127.0.0.1     306
        224.0.0.0        240.0.0.0         On-link    192.168.10.10     281
  255.255.255.255  255.255.255.255         On-link         127.0.0.1     306
```

Рис. 6.3. Приклад відображення таблиці маршрутизації для ПК1

Після введення наведених команд виводиться інформація про поточні мережеві підключення TCP/IP, що включає три основні розділи: список інтерфейсів, таблицю маршрутизації IPv4 і таблицю маршрутизації IPv6. У списку інтерфейсів відображається MAC-адреса і номер кожного мережевого інтерфейсу пристрою, включно з адаптерами Ethernet, Wi-Fi та Bluetooth.

Таблиці маршрутизації IPv4 і IPv6 містять усі відомі маршрути для відповідних протоколів, включаючи маршрути з прямим підключенням, локальні маршрути та маршрути за замовчуванням.

Кожна таблиця маршрутизації складається з п'яти основних стовпців: *мережа призначення (Network Destination), маска мережі (Netmask), шлюз (Gateway), інтерфейс (Interface)* і *метрика (Metric).* Стовпець мережі призначення містить список досяжних мереж, маска мережі вказує, як розділити IP-адресу на мережеву і вузлову частини.

У стовпці шлюзу зазначено адресу, через яку має бути доставлено пакет; якщо вузол призначення безпосередньо доступний, вказується «On-link». Стовпець інтерфейсу визначає, через який фізичний інтерфейс буде надіслано пакет, а метрика показує вартість маршруту – менше значення означає кращий маршрут.

Коли пакет прибуває на інтерфейс маршрутизатора, він перевіряє свою таблицю маршрутизації, щоб визначити, куди його далі переслати. Таблиця маршрутизатора містить як маршрути з прямим підключенням, так і віддалені маршрути (рис. 6.4).

Рис. 6.4. Мережі з прямим підключенням і віддалені мережі маршрутизатора R1

Маршрути з прямим підключенням створюються автоматично, коли інтерфейс маршрутизатора активується з IP-адресою. Кожен інтерфейс підключений до окремого сегмента мережі, і маршрутизатор зберігає цю інформацію в таблиці.

Віддалені маршрути надаються іншими маршрутизаторами або налаштовуються вручну адміністратором. Дані про такі маршрути можуть передаватися динамічно з використанням протоколів маршрутизації, таких як EIGRP або OSPF.

Таблиця маршрутизації вузла включає лише маршрути з прямим підключенням, тому для пересилки даних у віддалену мережу необхідно мати налаштований шлюз за замовчуванням. У свою чергу, таблиця маршрутизатора дозволяє зберігати як локальні, так і віддалені маршрути. У ній кожен маршрут має код джерела, який позначає спосіб його отримання.

Наприклад, C – це маршрут з прямим підключенням, L – маршрут локального каналу, S – статичний маршрут, який було створено вручну, D – маршрут, отриманий через EIGRP, а O – маршрут, отриманий через OSPF (рис. 6.5).

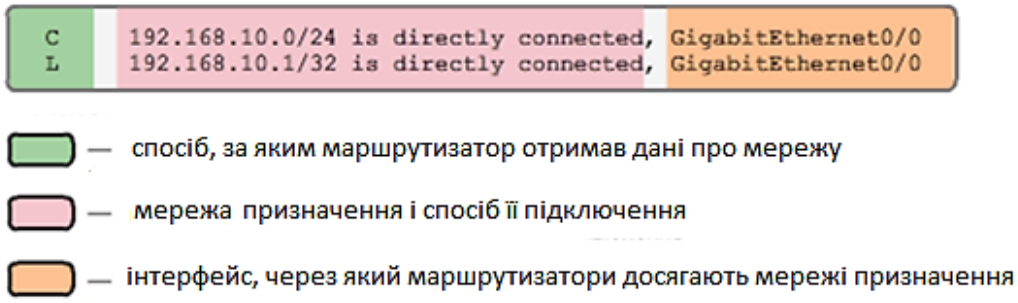


Рис. 6.5. Відображення інтерфейсів маршрутизатора з прямим підключенням

Записи в таблиці маршрутизації маршрутизатора для віддалених мереж містять кілька ключових елементів: джерело маршруту, мережу призначення, адміністративну відстань (яка показує достовірність маршруту), метрику, IP-адресу наступного переходу, інтерфейс виходу та відмітку часу останньої активності маршруту.

Наступний перехід – це IP-адреса пристрою, до якого маршрутизатор повинен надіслати пакет. Для звичайного вузла наступним переходом виступає адреса шлюзу за замовчуванням.

Для кожного маршруту до віддаленої мережі в таблиці маршрутизатора зазначено відповідний наступний перехід (рис. 6.6).

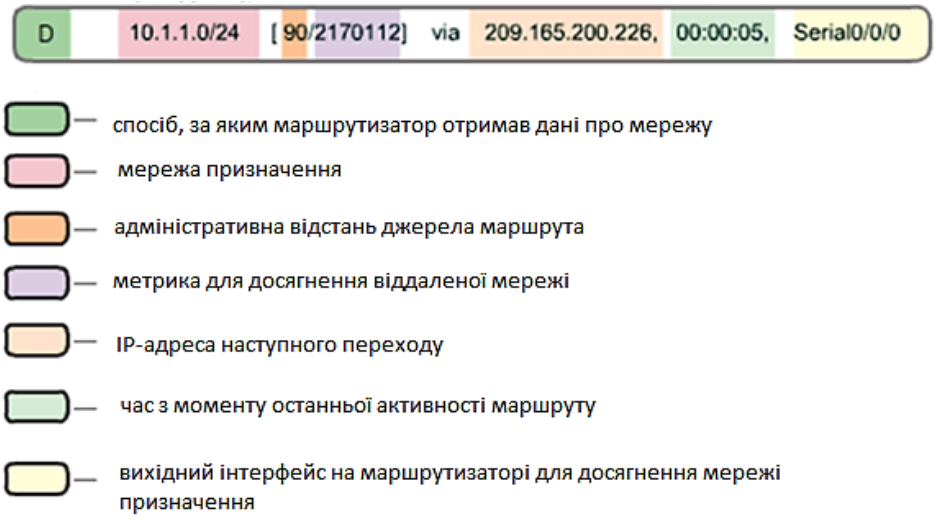


Рис. 6.5. Приклад відображення запису таблиці маршрутизації на маршрутизаторі для маршруту до віддаленої мережі 10.1.1.0

Якщо пакет потрапляє на маршрутизатор, і в таблиці маршрутизації знайдено відповідний маршрут, маршрутизатор пересилає пакет на вказаний IP-адресу наступного пристрою, використовуючи зазначений інтерфейс. Проте, якщо маршрут до мережі призначення відсутній, маршрутизатор не може доставити пакет і відкидає його. Щоб уникнути втрати пакетів, які не відповідають жодному з відомих маршрутів, маршрутизатор можна налаштувати для використання статичного маршруту за замовчуванням, який слугуватиме своєрідним шлюзом «останньої надії».

6.6. Будова і функції маршрутизатора

Незалежно від функціональних можливостей, розмірів або рівня складності, всі маршрутизатори за своєю будовою дуже подібні до звичайних комп’ютерів.

Подібно до настільних ПК, ноутбуків, планшетів або смартпристроїв, маршрутизатори мають типові апаратні компоненти, до яких належать:

- *операційна система (ОС)* – забезпечує взаємодію між апаратним забезпеченням маршрутизатора та його функціями;

- *центральний процесор (ЦП)* – виконує обробку даних та керує всіма процесами в пристрої;
- *оперативна пам'ять* – зберігає тимчасові дані, необхідні під час роботи пристрою;
- *постійна пам'ять* – містить інструкції для початкового завантаження маршрутизатора.

Крім того, маршрутизатори мають спеціалізовані типи пам'яті, зокрема, флеш-пам'ять і NVRAM (незалежну пам'ять), яка зберігає налаштування конфігурації навіть після вимкнення живлення.

Центральний процесор маршрутизатора потребує операційної системи для виконання функцій маршрутизації й комутації.

Незважаючи на різноманіття моделей маршрутизаторів, їх апаратна структура залишається в цілому однаковою (рис. 6.6).

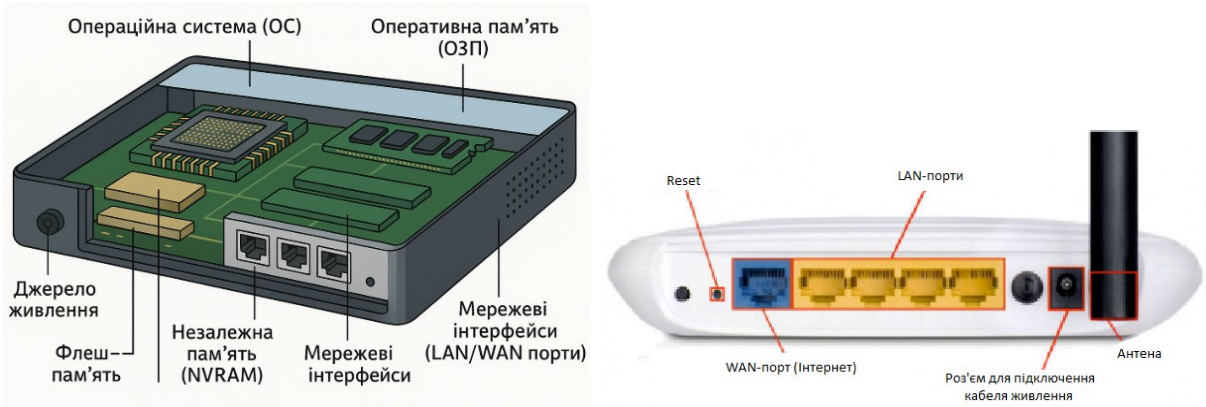


Рис. 6.6. Узагальнена схема будови маршрутизатора

Розглянемо типові компоненти маршрутизатора.

WAN-порт (Інтернет) – порт для підключення до глобальної мережі Інтернет.

LAN-порти – порти для підключення локальних пристроїв, таких як комп'ютери, принтери тощо.

Антена – забезпечує бездротове з'єднання Wi-Fi.

Кнопка скидання (Reset) – скидає налаштування маршрутизатора до заводських.

Кнопка живлення – вмикає або вимикає пристрій.

Індикатори стану – світлодіоди, що показують статус роботи маршрутизатора та його з'єднань.

До основних функцій маршрутизаторів належать наступні:

- адресація – визначення та обробка IP-адрес для пересилки даних;
- інтерфейси – забезпечення підключення до різних типів мереж;
- маршрутизація – вибір оптимального шляху для пересилки пакетів;
- безпека – фільтрація трафіку, контроль доступу та інші механізми захисту;
- QoS (якість обслуговування) – управління трафіком для пріоритетності певних типів даних;
- управління ресурсами – моніторинг і оптимізація використання апаратних і мережевих ресурсів.

Контрольні запитання та задачі до теми 6

Перелік теоретичних запитань

1. Які основні функції виконує мережевий рівень у моделі OSI?
2. Яку роль відіграє протокол IP у мережевому рівні?
3. Чим відрізняються протоколи IPv4 і IPv6?
4. Які типи адрес існують у протоколі IPv4?
5. Що таке мережевий префікс і як він використовується в IP-адресації?
6. Які проблеми адресного простору вирішує IPv6 у порівнянні з IPv4?
7. Які основні формати запису IPv6-адреси?
8. Що таке шлюз за замовчуванням і яку функцію він виконує в мережі?
9. Як пристрій визначає, чи передавати дані локально, чи через шлюз?
10. Що таке таблиця маршрутизації та яку інформацію вона містить?
11. Як маршрутизатор приймає рішення про пересилання пакетів?
12. У чому полягає відмінність між статичною і динамічною маршрутизацією?
13. Які основні елементи має маршрутизатор і як вони взаємодіють?
14. Як відбувається процес інкапсуляції та декапсуляції на мережевому рівні?
15. Чим маршрутизатор відрізняється від комутатора за принципом роботи?

Перелік задач для самостійного опрацювання

Завдання 1. Налаштування IP-адресації в мережі IPv4.

У середовищі моделювання Cisco Packet Tracer створіть просту мережу з двох ПК і одного маршрутизатора. Налаштуйте IPv4-адресацію, перевірте з'єднання за допомогою команди *ping*.

Завдання 2. Побудова мережі з IPv6-адресацією.

Побудуйте мережу з трьома пристроями в середовищі моделювання Cisco Packet Tracer. Призначте їм глобальні IPv6-адреси. Використайте команду *ping* для перевірки зв'язку між вузлами.

Завдання 3. Вивчення таблиці маршрутизації.

Створіть мережу з двох маршрутизаторів і кількох ПК у середовищі Cisco Packet Tracer. Налаштуйте статичну маршрутизацію. Перевірте таблиці маршрутизації та поясніть їхній вміст.

Завдання 4. Налаштування шлюзу за замовчуванням.

У мережі з одним маршрутизатором і кількома ПК задайте шлюз за замовчуванням. Перевірте зв'язок з пристроєм в іншій мережі.

Завдання 5. Тестування протоколів IPv4 і IPv6.

У середовищі моделювання Cisco Packet Tracer створіть дві мережі – одну з IPv4, іншу з IPv6. Проведіть тестування з'єднань та зробіть висновки про відмінності у форматах адрес, налаштуванні, діагностиці.

Перелік використаної та рекомендованої літератури

6.1. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

6.2. IPv4 (RFC 791) [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc791

6.3. IPv6 (RFC 8200) [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc8200

6.4. Components of the Router Architecture [Електронний ресурс]. – Режим доступу: https://learn.microsoft.com/en-us/windows/win32/rras/components-of-the-router-architecture

6.5. What is a WiFi Router [Електронний ресурс]. – Режим доступу: https://www.netgear.com/hub/technology/what-is-a-wifi-router/

Тема 7. Транспортний рівень

Транспортний рівень моделі OSI є ключовим елементом, що забезпечує надійне та впорядковане передавання даних між пристроями у мережі. Саме на цьому рівні реалізуються механізми керування потоком, виявлення та виправлення помилок, а також встановлення з'єднань між застосунками на кінцевих вузлах. Основними протоколами транспортного рівня є TCP (Transmission Control Protocol) і UDP (User Datagram Protocol), які підтримують різні способи передавання даних залежно від вимог до надійності та швидкості. У цьому розділі розглядаються принципи роботи транспортного рівня, відмінності між TCP і UDP, а також роль портів у маршрутизації даних між програмами.

7.1. Призначення транспортного рівня

Транспортний рівень моделі OSI виконує важливу роль у процесі передавання даних між комп'ютерами в мережі. Саме тут відбувається підготовка інформації, отриманої від прикладного рівня, до передавання мережею. Він забезпечує встановлення зв'язку між пристроями-учасниками обміну, розподіл даних на зручні для передавання сегменти, контроль за їх доставкою та збиранням на приймаючому боці. Таким чином, транспортний рівень створює умови для того, щоб дані були надійно та правильно доставлені незалежно від типу мережі чи характеристик передавального середовища.

Під час встановлення з'єднання між відправником і одержувачем транспортний рівень визначає правила сегментації даних, забезпечує механізми виявлення втрат інформації та перевірку цілісності отриманих фрагментів. Його основне завдання – гарантувати передачу повного і точного потоку даних від одного прикладного процесу до іншого. При цьому зберігається незалежність від конкретної мережевої інфраструктури та її параметрів, таких як пропускна здатність або завантаженість.

Протоколи транспортного рівня дозволяють зібрати фрагментовані сегменти в логічний потік, який без помилок потрапить до адресата. У стеку протоколів TCP/IP для виконання цих завдань використовують два основні протоколи – TCP і UDP. TCP забезпечує контроль доставки і гарантує надійність з'єднання, а UDP – простіший і швидший, хоча й

не надає механізмів контролю доставки, тому більше підходить для таких задач, як потокове відео чи голосовий зв'язок.

Кожен потік даних між додатками на різних пристроях створює окрему *сесію* зв'язку. На одному вузлі може бути запущено багато застосувань, і всі вони можуть одночасно комунікувати з іншими пристроями у мережі. Транспортний рівень мусить керувати кожною з цих сесій окремо.

Для ідентифікації кожного окремого прикладного процесу на комп'ютері використовуються *номери портів* – цифрові ідентифікатори, що дозволяють даним потрапити до правильного отримувача всередині вузла.

Передавання великих обсягів даних як одного потоку може створити навантаження на мережу, блокуючи інші процеси, що одночасно передають інформацію.

Щоб уникнути цього, транспортний рівень розділяє дані на менші частини – сегменти, кожен з яких інкапсулюється із заголовком, що містить службову інформацію. Завдяки цьому стає можливим *мультиплексування*, тобто одночасне передавання різних потоків даних через одну мережу, не заважаючи роботі окремих додатків.

На приймальному боці транспортний рівень виконує об'єднання сегментів у повноцінний потік, який потім передається прикладному рівню. Заголовок кожного сегмента містить біти, що активують різні функції протоколу – наприклад, перевірку цілісності, керування порядком доставлення чи повторну передачу втрачених даних.

Таким чином, транспортний рівень не тільки забезпечує розподіл даних для ефективного передавання мережею, але й виконує критично важливі функції з керування зв'язком, підтримки надійності та забезпечення адресності передачі для конкретних застосувань.

Залежно від вимог до швидкості або надійності передачі, вибір між TCP і UDP дозволяє гнучко адаптувати роботу мережевих застосувань до умов конкретного середовища.

7.2. Протокол TCP

TCP (Transmission Control Protocol) є одним із двох основних протоколів транспортного рівня в стеку TCP/IP, і саме він забезпечує надійну передачу даних між пристроями в комп'ютерній мережі.

Основна мета TCP – гарантувати, що кожен фрагмент даних, який було відправлено з одного пристрою, успішно прибув до іншого, і якщо якісь фрагменти були втрачені чи пошкоджені в процесі, то вони будуть повторно надіслані. Саме завдяки таким функціям TCP вважається протоколом з надійною доставкою.

Для досягнення надійності TCP використовує три ключові механізми. Перший – це відстеження усіх сегментів, які були надіслані. Кожному сегменту призначається порядковий номер, що дозволяє чітко фіксувати, який саме фрагмент повідомлення було передано. Другий механізм – підтвердження отримання даних. Приймаюча сторона надсилає повідомлення-підтвердження відправникові після отримання кожного сегменту. І, нарешті, третій механізм – це повторна відправка всіх сегментів, які не були підтверджені. Якщо TCP не отримує підтвердження в межах певного часу, він вважає сегмент втраченим і пересилає його знову. При цьому не пересилається усе повідомлення цілком, а лише ті сегменти, які не дійшли до адресата.

Передача даних за допомогою TCP базується на попередньому встановленні з'єднання, яке ще називають сеансом. Це означає, що перед тим, як розпочати передачу даних, TCP встановлює логічний зв'язок між пристроями-учасниками.

Сеанс дозволяє узгодити обсяг трафіку, який можна передавати одночасно, та налаштувати параметри обміну. Таке з'єднання триватиме доти, поки не буде передано усе повідомлення. Приклади застосувань, які користуються TCP, – це FTP (протокол передавання файлів) та HTTP (протокол передавання гіпертексту).

Завдяки функціям встановлення і підтримання сеансів, підтвердження доставки, повторної передачі та контролю черговості, TCP забезпечує стабільну і безпечну комунікацію між пристроями. Цей протокол здатен впорядковувати отримані сегменти у правильній послідовності, навіть якщо вони прибули до адресата не в тому порядку, в якому були надіслані. Це особливо важливо в мережах, де дані можуть проходити різними маршрутами з різними швидкостями.

Ще однією важливою функцією TCP є управління потоком. У випадках, коли ресурси приймаючого пристрою – наприклад, оперативна пам'ять або пропускна здатність – перевантажені, TCP може уповільнити потік вхідних даних, щоб уникнути перевантаження. Це

робиться шляхом регулювання кількості сегментів, які дозволено передати без підтвердження. Таким чином, TCP не лише запобігає втраті даних, а й підвищує ефективність мережі загалом.

TCP є протоколом з контролем стану з'єднання, тобто TCP підтримує актуальну інформацію про стан кожного встановленого з'єднання та відслідковує, які сегменти вже були надіслані, які з них підтверджено та отримано, і які ще потрібно повторно надіслати. Такий контроль дозволяє TCP надійно завершити кожен сеанс зв'язку. Усі ці функції потребують додаткових обчислювальних ресурсів, а також створюють додаткове навантаження в заголовках TCP-сегментів.

Заголовок TCP має розширену структуру та містить щонайменше 20 байт службової інформації. До заголовку TCP входять, зокрема, порядковий номер, що дозволяє впорядкувати сегменти, і номер підтвердження, який вказує, що певний обсяг даних вже отримано. Довжина заголовка вказує на обсяг службової частини сегменту, а поле «зарезервовано» залишено для майбутніх оновлень (рис. 7.1).

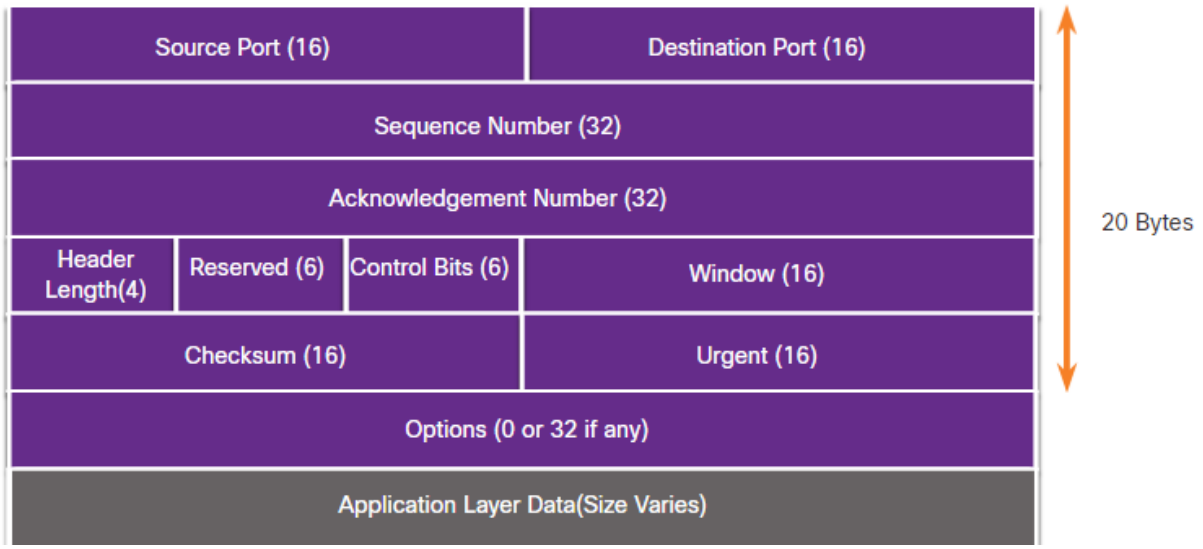


Рис. 7.1. Заголовок TCP

Біти керування (так звані *прапори*) визначають тип і мету сегмента, наприклад, ініціалізація з'єднання або завершення сеансу. Розмір вікна вказує на кількість байтів, які приймаюча сторона може опрацювати без підтвердження. Контрольна сума дозволяє виявити помилки в сегменті, а поле терміновості сигналізує про необхідність пріоритетної обробки даних (табл. 7.1).

Таблиця 7.1. **Поля заголовка TCP**

Поле заголовка TCP	Опис
Порт джерела	16-бітове поле, яке використовується для ідентифікації вихідної програми за номером порту.
Порт призначення	16-бітове поле, яке використовується для ідентифікації цільової програми за номером порту.
Порядковий номер	32-розрядне поле, яке використовується для повторного збирання даних.
Номер підтвердження	32-розрядне поле, яке використовується для вказівки на те, що дані отримані та наступний байт очікується від джерела.
Довжина заголовка	4-бітове поле, відоме як «зміщення даних», яке вказує довжину заголовка сегмента TCP.
Зарезервовано	6-бітове поле, яке зарезервовано для майбутнього використання.
Контрольні біти	Використовується 6-бітне поле, яке включає бітові коди або прапори, які вказують на призначення та функцію сегмента TCP.
Розмір вікна	16-бітове поле, яке використовується для вказівки кількості байтів, які можуть бути прийняті за один раз.
Контрольна сума	16-бітове поле, яке використовується для перевірки помилок заголовка сегмента та даних.
Терміново	16-бітове поле, яке використовується для вказівки терміновості даних, що містяться.

У порівнянні з UDP, протокол TCP надає набагато ширший набір функцій, що робить його придатним для передачі критично важливої інформації, яка має бути доставлена точно, у правильному порядку і без втрат. Проте через більший обсяг службової інформації TCP є більш ресурсоємним і має більшу затримку, ніж UDP, що є ціною за його надійність.

7.3. Протокол UDP

Протокол TCP забезпечує високу надійність під час передачі даних, що дозволяє застосуванням стабільно взаємодіяти між собою навіть у складних мережевих умовах. Це досягається за рахунок механізмів підтвердження отримання даних, повторної передачі

втрачених сегментів та відновлення послідовності. Однак ці функції вимагають додаткових обчислювальних ресурсів, що може спричинити затримку, особливо у випадках передачі великих обсягів даних або в умовах обмеженої пропускної здатності мережі.

Таким чином, між рівнем надійності та навантаженням на мережу виникає певний компроміс. Для деяких застосувань надмірна надійність навіть може негативно впливати на продуктивність і користувацький досвід, тому в таких випадках доцільніше використовувати альтернативний транспортний протокол – UDP (User Datagram Protocol).

Заголовок UDP набагато простіший, ніж заголовок TCP, оскільки він має лише чотири поля та вимагає 8 байтів, тобто 64 біти (рис. 7.2, табл. 7.2).

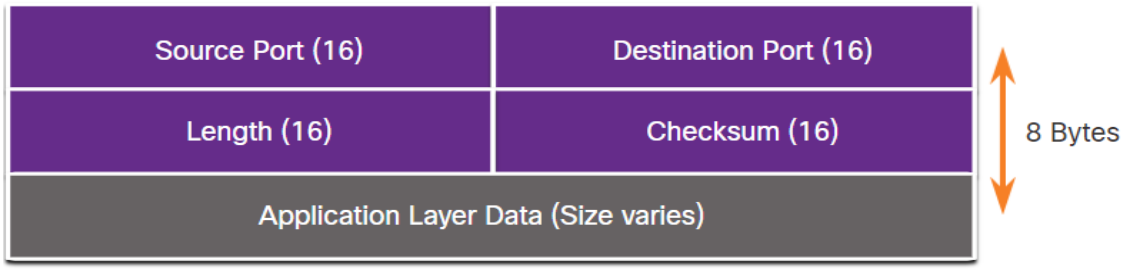


Рис. 7.2. Заголовок UDP

Таблиця 7.2. **Поля заголовка UDP**

Поле заголовка UDP	Опис
Порт джерела	16-бітове поле, яке використовується для ідентифікації вихідної програми за номером порту.
Порт призначення	16-бітове поле, яке використовується для ідентифікації цільової програми за номером порту.
Довжина	16-бітне поле, що вказує довжину заголовка дейтаграми UDP.
Контрольна сума	16-бітове поле, яке використовується для перевірки помилок заголовка дейтаграми та даних.

Протокол UDP надає лише базові можливості для передачі даних між застосуваннями без встановлення з’єднання і без гарантії доставки. Він не передбачає підтвердження отримання пакетів, не повторює передачу втрачених даних і не відновлює їхній початковий порядок. Це робить UDP протоколом із мінімальним використанням ресурсів і низькими затримками. Такий підхід називають "негарантованою доставкою", що означає відсутність контролю за успішністю передачі. UDP не задіює внутрішні механізми транспортного рівня, які відслідковують, чи досягли дані вузла призначення.

Механізм дії UDP можна порівняти з надсиланням звичайного листа поштою без реєстрації. Відправник не має жодного способу дізнатися, чи отримав адресат лист, а поштове відділення не забезпечує відстеження доставки. Аналогічно, у мережі UDP-протокол передає пакети без зворотного зв’язку та відповідальності за результат доставки.

Офіційна специфікація UDP описана в документі RFC 768. Хоча UDP також підтримує процеси сегментації і збору даних, подібно до TCP, цей протокол не реалізує функцій управління потоком і забезпечення надійності.

Кожен фрагмент даних, який надсилається через UDP, називається датаграмою. Датаграми надсилаються без перевірки доставки або спроби відновити пошкоджені чи втрачені частини. Якщо деякий пакет загубиться чи буде пошкоджений, він не передається повторно.

UDP характеризується кількома ключовими властивостями.

По-перше, UDP не встановлює з’єднання між пристроями, що дозволяє передавати дані без початкового обміну службовою інформацією.

По-друге, цей протокол не гарантує доставку – немає механізмів підтвердження, виявлення помилок чи повторної передачі.

По-третє, UDP не передбачає збереження початкової послідовності даних, тому отримані пакети можуть бути впорядковані інакше, ніж вони були надіслані.

Нарешті, протокол UDP не забезпечує контроль за потоком даних, тому перевантаження вузла призначення може призвести до втрати інформації без подальших спроб її відновлення.

Завдяки простоті та низькому навантаженню на ресурси, UDP часто використовується в мережевих застосуваннях, для яких швидкість

передачі важливіша за точність. Серед них – служба доменних імен (DNS), потокове відео, VoIP (передача голосу через IP) та інтернет-радіо. Такі застосування зазвичай допускають втрату деякої кількості даних, яка майже не впливає на загальну якість. Наприклад, в інтернет-радіо втрата кількох аудіопакетів призводить лише до короткочасного зникнення звуку, що є менш критичним, ніж затримка, яка могла б виникнути через повторну передачу.

Оскільки UDP не відстежує стан сеансу, ні клієнт, ні сервер не зобов'язані підтримувати інформацію про попередню взаємодію. Це дозволяє значно знизити складність реалізації застосування, що працює з UDP, і зменшити витрати часу та ресурсів на передачу.

Обидва транспортні протоколи – TCP і UDP – є ефективними у своїх сферах застосування. Вибір між ними залежить від конкретних вимог програмного забезпечення. Якщо застосування вимагає доставки всіх даних без втрат і в правильному порядку, TCP буде кращим вибором. Це характерно для баз даних, веббраузерів і поштових клієнтів, де навіть незначна втрата інформації може призвести до помилок або пошкодження файлів. У таких випадках надійність важливіша за швидкість, тому розробники застосовують саме TCP, незважаючи на його більший вплив на ресурси мережі.

З іншого боку, якщо для застосування важлива швидка передача даних у реальному часі, а втрата частини інформації не є критичною, доцільніше використовувати протокол UDP. Затримки, пов'язані з підтвердженнями або повторною передачею, у таких випадках тільки шкодитимуть. Саме тому UDP є основним протоколом для VoIP, онлайн-відео, потокового аудіо та інших служб, де необхідна безперервність передавання і мінімальна затримка.

7.4. Адресація портів TCP і UDP

Кожне повідомлення, що передається між мережевими застосуваннями через протоколи TCP або UDP, супроводжується службовою інформацією, яка дозволяє коректно встановити зв'язок між пристроями. Одним із ключових елементів такої інформації є *номери портів* – числові ідентифікатори, що вказуються в заголовках сегментів (у TCP) або датаграм (у UDP). Вони вказують, з якого саме порту на відправляючому пристрої було надіслано повідомлення (порт джерела)

та на який порт на приймаючому пристрої слід його доставити (порт призначення). Завдяки цьому система знає, до якого саме застосування на обох кінцях має бути спрямовано дані.

Коли клієнт ініціює звернення до якогось сервісу, він вказує номер порту призначення, який відповідає конкретній службі. Наприклад, якщо клієнт хоче звернутися до вебсайту, він зазначає порт 80, який зарезервований для протоколу HTTP. У свою чергу, сервер, що обробляє запити, може одночасно працювати з декількома сервісами, кожен з яких має власний порт. Наприклад, той самий сервер може обслуговувати HTTP через порт 80 і FTP через порт 21, розмежовуючи трафік за допомогою номерів портів.

Номер порту джерела зазвичай генерується динамічно – тобто випадковим чином – клієнтським пристроєм під час кожного нового сеансу зв'язку. Це дозволяє одному пристрою підтримувати кілька одночасних сеансів із тим самим або різними серверами.

Наприклад, користувач може відкривати кілька вкладок з вебсайтами в браузері одночасно, і кожен із запитів буде позначено окремим портом джерела. Саме завдяки цим номерам транспортний рівень мережевої моделі може коректно розпізнати, який саме сеанс ініціював відповідь, що надходить у відповідь від сервера.

Коли дані відправляються через мережу, інформація про порт джерела та порт призначення додається до транспортного сегменту або датаграми. Далі цей сегмент інкапсулюється в IP-пакет, в якому, крім портів, також вказуються IP-адреси джерела і призначення.

Поєднання цих чотирьох параметрів – IP-адреси відправника і одержувача разом з відповідними портами – утворює унікальний ідентифікатор, що називається сокетом.

Сокет дозволяє точно встановити, яке застосування на якому пристрої бере участь у сеансі зв'язку.

Сокет клієнта може виглядати як 192.168.1.5:1099, де IP-адреса – це адреса пристрою-клієнта, а 1099 – номер порту, з якого відправляються дані. Водночас сокет сервера, що надає вебдоступ, може бути представлений як 192.168.1.7:80. Разом вони утворюють пару сокетів: 192.168.1.5:1099 – 192.168.1.7:80. Така пара точно ідентифікує канал зв'язку між клієнтом і сервером, забезпечуючи можливість передавання даних без плутанини навіть у разі великої кількості

одночасних з'єднань. Система портів дозволяє мережевим застосуванням на одному пристрої одночасно обробляти декілька потоків даних від різних джерел. Коли клієнт надсилає запит, порт джерела виконує функцію зворотної адреси. Транспортний рівень запам'ятовує цей номер, щоб при отриманні відповіді з сервера дані могли бути направлені до правильного застосування або процесу.

Всі порти розподіляються згідно з класифікацією, яку визначає Інтернет-адміністрація адресного простору IANA.

Існує три основні категорії портів (табл. 7.3).

Таблиця 7.3. **Групи номерів портів**

Група портів	Діапазон номерів	Опис
Відомі порти	**від 0 до 1023**	Ці номери портів зарезервовано для звичайних або популярних служб і програм, таких як веббраузери, клієнти електронної пошти та клієнти віддаленого доступу. Визначені добре відомі порти для звичайних серверних програм дозволяють клієнтам легко визначити необхідну пов'язану службу.
Зареєстровані порти	**1,024 до 49,151**	Ці номери портів призначає IANA суб'єкту, який надіслав запит, для використання з певними процесами чи програмами. Ці процеси – це переважно окремі програми, які користувач вирішив інсталювати, а не звичайні програми, які отримуватимуть добре відомий номер порту.
Приватні та/або **динамічні порти**	**49,152 до 65,535**	Ці порти також відомі як *тимчасові порти*. ОС клієнта зазвичай призначає номери портів динамічно, коли ініціюється підключення до служби. Потім динамічний порт використовується для ідентифікації клієнтської програми під час зв'язку.

Перша категорія – це відомі порти, номери яких лежать у діапазоні від 0 до 1023. Вони зарезервовані для стандартних сервісів, таких як HTTP (порт 80), SMTP (порт 25 для електронної пошти), IMAP (порт 143 для доступу до поштових скриньок) чи Telnet (порт 23). Ці порти найчастіше використовуються саме на стороні сервера.

Клієнтські програми зазвичай вже знають, який порт відповідає певному сервісу, і використовують його для встановлення з'єднання (табл. 7.4).

Таблиця 7.4. **Відомі номери портів**

Номер порту	Протокол	Застосування
20	TCP	Протокол передачі файлів (FTP) - дані
21	TCP	Протокол передачі файлів (FTP) - керування
22	TCP	Безпечна оболонка (SSH)
23	TCP	Telnet
25	TCP	Простий протокол передачі пошти (SMTP)
53	UDP, TCP	Служба доменних імен (DNS)
67	UDP	Протокол динамічної конфігурації хоста (DHCP) - сервер
68	UDP	Протокол динамічної конфігурації хоста - клієнт
69	UDP	Тривіальний протокол передачі файлів (TFTP)
80	TCP	Протокол передачі гіпертексту (HTTP)
110	TCP	Протокол поштового відділення версії 3 (POP3)
143	TCP	Протокол доступу до повідомлень Інтернету (IMAP)
161	UDP	Простий протокол керування мережею (SNMP)
443	TCP	Захищений протокол передачі гіпертексту (HTTPS)

Друга категорія – зареєстровані порти, які мають номери від 1024 до 49151. Їх використовують застосування, які не є частиною стандартного набору інтернет-сервісів, але все ж мають фіксовані порти, зареєстровані для забезпечення унікальності. Найчастіше такі порти асоціюються із застосуваннями, встановленими користувачем або сторонніми сервісами.

Третя категорія – це динамічні або приватні порти, з номерами від 49152 до 65535. Вони призначені для тимчасового використання під час ініціації з'єднання. Коли клієнт запускає нове з'єднання, його система динамічно обирає один із цих портів як порт джерела. Саме ці порти найчастіше використовуються для клієнтських з'єднань. Хоча у

деяких випадках приватні порти можуть бути використані і для серверного зв’язку – наприклад, у системах однорангового обміну файлами, де кожен вузол може виконувати роль як клієнта, так і сервера.

Отже, структура портів і сокетів є ключовою складовою роботи транспортного рівня в сучасних комп’ютерних мережах. Вона дозволяє одночасно обробляти величезну кількість з’єднань, точно визначати кінцеві точки обміну даними та гарантувати доставку інформації потрібному застосуванню.

7.5. Сегментація TCP і UDP

Розбиття даних на сегменти є фундаментальним принципом роботи транспортного рівня в комп’ютерних мережах. Цей процес дозволяє передавати великі обсяги інформації частинами, що відповідає можливостям мережевого середовища, а також забезпечує одночасну роботу кількох застосувань на одному пристрої. Завдяки такому розподілу даних декілька програм можуть використовувати одне й те саме з'єднання з мережею без взаємного конфлікту – кожне з них отримує свій окремий потік даних, що проходить через механізм мультиплексування.

Коли дані передаються за допомогою протоколу TCP, кожен сегмент, який надсилається в мережу, має у своєму заголовку спеціальний порядковий номер. Це число дозволяє приймаючій стороні, тобто вузлу призначення, точно зібрати отримані сегменти у правильному порядку, незалежно від того, якою послідовністю вони фактично прибули. Завдяки цій функції застосування, що приймає дані, отримує їх саме в такій формі, в якій їх сформував відправник. Це критично важливо для багатьох застосувань, зокрема веббраузерів, систем передачі файлів чи електронної пошти, де навіть невелика помилка порядку байтів може призвести до некоректної роботи.

Протокол TCP забезпечує не лише контроль порядку, але й гарантію доставки, перевірку цілісності даних, відновлення втрачених сегментів тощо. Така надійність має свою ціну – TCP вимагає складнішої обробки даних, ведення обліку стану сеансу, підтверджень прийому, повторних передач при втраті сегментів. Усе це створює додаткове навантаження на ресурси системи й збільшує час на обробку.

На відміну від TCP, протокол UDP є більш «простим» варіантом передачі даних. UDP також дозволяє передавати інформацію між застосуваннями та підтримує ідентифікацію окремих сеансів, проте не виконує контроль порядку доставки та не гарантує збереження підключення. У заголовку UDP-сегментів відсутні поля з порядковими номерами, а отже, вузол призначення не може зібрати отримані частини у правильній послідовності, якщо вони прибули не в тому порядку, в якому були відправлені.

UDP не передбачає встановлення стійкого з'єднання – кожен пакет передається окремо, незалежно від інших. Завдяки такій спрощеній структурі протокол забезпечує вищу швидкість передачі даних і менше навантаження на мережеву інфраструктуру. Це робить UDP ідеальним для тих застосувань, де критично важлива швидкість, а не гарантія доставки, таких як відеотрансляції, голосовий зв'язок або онлайн-ігри.

Оскільки в мережі пакети можуть рухатись різними маршрутами, цілком можливо, що вони досягнуть точки призначення в іншому порядку, ніж були відправлені. У випадку з UDP це не розглядається як помилка – застосування, що використовує цей протокол, повинне бути здатним працювати з даними, отриманими в довільному порядку. Тобто обробка повинна бути або нечутливою до порядку, або містити власні механізми впорядкування за потреби.

Отже, вибір між TCP і UDP залежить від конкретних вимог застосування: якщо потрібна висока надійність – використовується TCP; якщо ж важлива мінімальна затримка та висока швидкість – тоді перевага надається UDP.

7.6. Процеси і запити TCP –сервера

Процеси застосунків зазвичай запускаються на серверних комп'ютерах, які здатні одночасно обробляти декілька таких процесів. Кожен із них очікує на ініціативу з боку клієнта – зокрема, запит на передачу даних, отримання інформації або доступ до певного сервісу. Це означає, що саме клієнт розпочинає взаємодію, а сервер реагує на запит, використовуючи відповідний застосунок, який вже запущений і готовий до обробки даних.

Для забезпечення коректної роботи між серверами та клієнтами кожен серверний процес прив'язується до певного номера порту. Цей номер може бути призначений за замовчуванням або встановлений вручну системним адміністратором. Важливо розуміти, що на одному й тому ж сервері не можуть функціонувати одночасно два сервіси з однаковими номерами портів в межах одного транспортного протоколу. Наприклад, вебсервер і FTP-сервер, що працюють на одному сервері, не можуть одночасно використовувати TCP-порт 8080. Кожен з них має мати свій унікальний порт.

Якщо певному застосунку призначено конкретний порт і він активний, вважається, що цей порт «відкритий». Це означає, що транспортний рівень мережевої моделі може приймати пакети, адресовані на цей порт, і передавати їх далі до відповідного застосунку. На одному сервері може бути відкрито кілька портів – по одному на кожний запущений сервіс. У більшості випадків сервери одночасно надають кілька типів сервісів, наприклад, можуть працювати як вебсервери та FTP-сервери.

Для підвищення безпеки серверів часто застосовують практику обмеження доступу лише до тих портів, які потрібні для роботи дозволених сервісів. Це дозволяє зменшити ризики атак на мережу, оскільки зловмисники не можуть підключитися до сервісів, які не мають відкрити портів або не повинні бути доступні зовнішнім користувачам.

Встановлення з'єднання між двома вузлами в мережі можна порівняти з процесом рукостискання, що є загальноприйнятим жестом вітання між двома людьми.

У комп'ютерних мережах аналогом такого жесту є *процедура «трибічного рукостискання» (three-way handshake)*, яка використовується в TCP-з'єднаннях (рис. 7.3).

Перший етап – ініціювання з'єднання клієнтом за допомогою запиту синхронізації SYN.

Другий етап – підтвердження ACK цього запиту сервером, разом з пропозицією встановити зворотний зв'язок.

Третій етап – остаточне підтвердження ACK з боку клієнта, після чого обидві сторони вважають з'єднання встановленим (FIN).

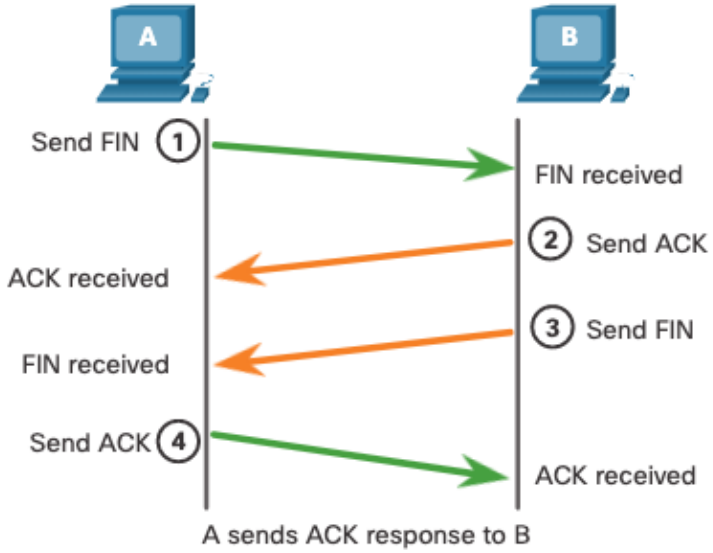


Рис. 7.3. Процедура завершення процесу «трибічного рукостискання» (three-way handshake), яка використовується в TCP-з'єднаннях

Перш ніж дані почнуть передаватися між вузлами, трибічне рукостискання гарантує, що пристрій призначення доступний у мережі, що він має активний сервіс на запитуваному номері порту, та що з'єднання може бути безпечно встановлене. Процес рукостискання забезпечується за допомогою спеціальних бітів управління в заголовку TCP. Зокрема, біти SYN та ACK беруть участь у цьому процесі: SYN використовується для ініціації з'єднання, ACK – для підтвердження.

Коли передаються дані, важливо, щоб вони були доставлені в правильному порядку, навіть якщо окремі сегменти потрапляють до адресата різними маршрутами. TCP забезпечує це за допомогою порядкових номерів, які вказуються в кожному сегменті.

На початку кожного сеансу встановлюється початковий порядковий номер ISN, який використовується як базове значення для підрахунку байтів у потоці даних. Усі подальші сегменти збільшують цей номер відповідно до обсягу переданих даних.

На приймаючому боці TCP розміщує сегменти у буфері згідно з їхніми порядковими номерами. Якщо деякі сегменти прибувають пізніше або поза порядком, вони зберігаються до моменту, коли можна зібрати повну послідовність.

Це дозволяє повністю відновити вихідне повідомлення і передати його на рівень застосунків без втрати чи спотворення інформації.

TCP гарантує доставку кожного сегменту, використовуючи систему підтверджень. Кожне підтвердження містить номер ACK – це очікуваний номер наступного байта, який приймаючий вузол розраховує отримати. Таким чином, відправник знає, які дані вже успішно доставлені, а які ще потребують повторної передачі. Порядкові номери SEQ та номери підтверджень ACK використовуються одночасно в обох напрямках обміну даними.

Протокол TCP також використовує поняття розміру вікна, яке визначає кількість байтів, що можуть бути передані до отримання наступного підтвердження. Це дозволяє контролювати об'єм переданих даних і підтримувати стабільну та безпечну передачу, знижуючи ймовірність перевантаження мережі або буфера приймача. Якщо підтвердження вчасно не надходить, TCP вважає, що сегмент було втрачено, і пересилає його повторно.

Управління потоком – ще один важливий механізм, який використовує TCP для забезпечення надійної роботи. Завдяки цьому механізму протокол може регулювати швидкість передачі даних відповідно до можливостей мережі та обмежень пристрою-отримувача. Одним зі способів управління потоком є *динамічна зміна розміру вікна* – в залежності від умов мережі, TCP може зменшити розмір вікна, вимагаючи частіших підтверджень і, відповідно, знижуючи темп передачі.

Отримуючий пристрій передає відправнику значення розміру вікна, яке вказує, скільки байтів він готовий обробити. Якщо буфер майже заповнений, розмір вікна може бути зменшено, що слугує сигналом для джерела про необхідність зниження швидкості.

Усе це дозволяє ефективно управляти передачею даних, запобігати втратам і уникати повторних пересилок, що суттєво підвищує якість обміну інформацією в мережі.

7.7. Процеси і запити UDP –сервера

UDP є одним із основних протоколів транспортного рівня, який забезпечує передачу даних без створення з'єднання. Його основна відмінність від TCP полягає у простоті: він не вимагає встановлення стабільного з'єднання між пристроями перед передачею інформації, не використовує механізми контролю потоку, перевірки доставки або

повторної передачі втрачених даних. Це дозволяє значно зменшити обчислювальне навантаження та затримки, що особливо важливо для реального часу або систем, де швидкість важливіша за повну надійність.

Тим не менш, відсутність вбудованих механізмів забезпечення надійності не означає, що UDP – це ненадійний або "неповноцінний" протокол. Надійність у разі потреби може бути реалізована на прикладному рівні або за допомогою інших механізмів.

Таким чином, UDP підходить для ситуацій, коли сам протокол не повинен втручатися в обробку помилок або втрат пакетів, і коли це доречно робити на рівні застосування.

Хоча обсяг трафіку, який передається через UDP, як правило, менший, ніж через TCP, протокол використовується в ряді критично важливих сервісів. Наприклад, служба доменних імен DNS для швидких запитів і відповідей, протокол DHCP для динамічного призначення IP-адрес, SNMP для моніторингу мереж, RIP – один із старіших протоколів маршрутизації, TFTP для спрощеної передачі файлів, а також UDP широко використовується в IP-телефонії (VoIP) і онлайн-іграх, де затримки критично важливі, а втрати кількох пакетів не мають вирішального значення.

Оскільки UDP не вимагає встановлення з’єднання, як це робить TCP, обмін повідомленнями між пристроями виглядає як послідовність незалежних транзакцій. Якщо прикладна програма має дані для надсилання, вона просто передає їх без попереднього узгодження умов зв’язку. Це спрощує процес і скорочує час між запитом і відповіддю, що особливо важливо в системах із жорсткими обмеженнями часу.

У випадках, коли одержувач отримує кілька UDP-дейтаграм, вони можуть приходити не в тому порядку, в якому були надіслані, оскільки різні пакети можуть проходити через різні маршрути. Протокол UDP не виконує відстеження порядку доставки, як TCP. Тому застосування, яке використовує UDP, повинне самостійно впорядковувати пакети, якщо це необхідно для його коректної роботи.

Серверні застосування, які працюють через UDP, також використовують зареєстровані або відомі порти, які призначаються їм системою. Коли така програма запущена, вона чекає на прийом дейтаграм, що приходять на конкретний номер порту. Якщо UDP

отримує пакет, адресований одному з цих портів, він передає його прикладній програмі згідно з номером порту.

Як і з TCP, ініціатива для зв'язку, як правило, виходить від клієнта. Клієнтська програма випадковим чином вибирає порт джерела з динамічного діапазону, який буде використано для передачі, тоді як порт призначення – це зазвичай заздалегідь визначений, зареєстрований порт сервера. Використання випадкових портів джерела також сприяє підвищенню безпеки, оскільки ускладнює зловмисникам прогнозування відкритих портів для потенційної атаки.

Після вибору портів і формування даних UDP негайно передає пакет на нижчий, мережевий рівень для обробки та надсилання по мережі. У заголовку кожної дейтаграми міститься інформація про порт джерела та порт призначення. Для відповіді від сервера використовується та сама пара портів, але в зворотному порядку, що дозволяє організувати двосторонній обмін повідомленнями навіть без створення постійного з'єднання.

7.8. Застосування, що використовують протоколи TCP та UDP

Багато сучасних мережевих застосувань вимагають гарантованої доставки даних, надійності з'єднання та повного контролю за передачею інформації. У таких випадках оптимальним вибором є протокол TCP. TCP забезпечує всю необхідну функціональність для коректної та впорядкованої доставки даних між віддаленими пристроями. Хоча протокол TCP передбачає певні затримки, пов'язані з перевіркою цілісності даних, повторною передачею втрачених сегментів та підтвердженням доставки, цей протокол підходить для сценаріїв, де точність та завершеність переданої інформації мають вирішальне значення.

Одна з головних переваг TCP полягає в тому, що він бере на себе більшість обов'язків з обробки даних, які інакше мали би виконуватись прикладними програмами. Застосуванням не потрібно самостійно турбуватись про розбиття потоку даних на сегменти, відновлення послідовності або повторну передачу втрачених пакетів – усі ці функції реалізуються на рівні TCP.

Таким чином, прикладна програма віддає потік даних транспортному рівню, а TCP дбає про надійність та правильну доставку. Це значно спрощує розробку застосувань і зменшує ризик помилок у роботі мережі.

Прикладами добре відомих мережевих застосувань, які працюють на базі TCP, є протокол HTTP, що забезпечує передачу вебсторінок у браузері, FTP – для обміну файлами між комп'ютерами, SMTP – для надсилання електронної пошти, а також Telnet, який надає можливість віддаленого доступу до серверів (рис. 7.4).

Усі ці протоколи потребують максимальної гарантії доставки даних, навіть якщо це передбачає певні затримки у передачі.

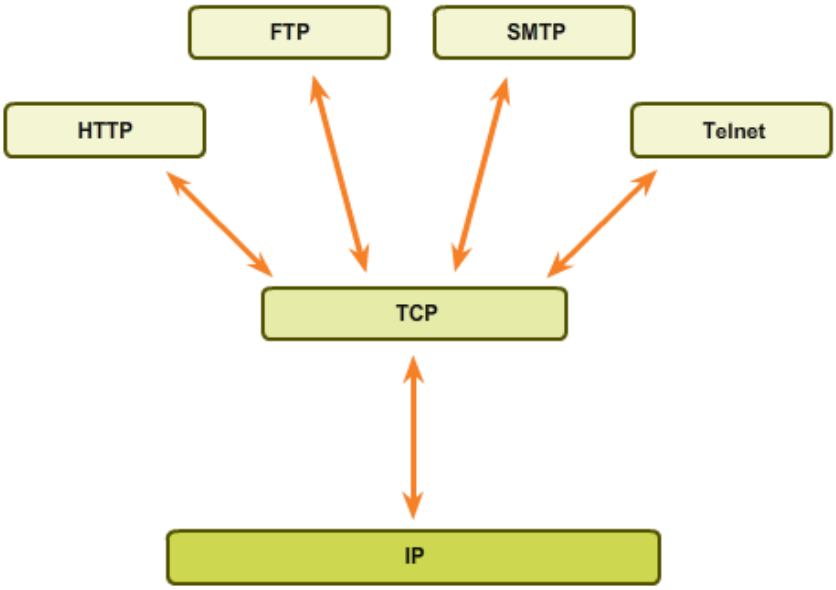


Рис. 7.4. Застосування, що використовують протокол TCP

Однак не всі типи застосувань потребують такої суворої надійності. Існує чимало сценаріїв, де швидкість має значно більше значення, ніж безпомилкова передача кожного байта. У таких випадках перевага надається протоколу UDP, який є особливо корисним для таких трьох типів застосувань: по-перше, для тих, що допускають втрату деяких пакетів, але мають високу чутливість до затримок; по-друге, для простих запитно-відповідних операцій, де можлива втрата відповіді; і по-третє, для однонапрямлених потоків, у яких відправник не очікує жодного підтвердження або відповіді від одержувача.

Яскравим прикладом застосувань, які використовують UDP, є служби передачі відео й голосу, зокрема, IP-телефонія та потокове інтернет-телебачення. Для них втрата кількох пакетів не має суттєвого значення для кінцевого користувача, але надмірна затримка, що виникає через механізми повторної передачі TCP, може істотно погіршити якість відео чи звуку. Саме тому UDP дозволяє досягти вищої ефективності в мультимедійному середовищі.

Інший клас застосувань, який добре працює з UDP, включає прості запити та відповіді, де ймовірність втрати пакету є прийнятною. До таких протоколів належать DHCP (динамічне отримання IP-адрес), DNS (служба доменних імен, яка в певних випадках також використовує TCP), SNMP (протокол керування мережею) і TFTP (спрощений протокол передачі файлів) (рис. 7.5). Ці протоколи орієнтовані на ефективність і мінімальні витрати, і тому застосування складнішого TCP було б у них надмірним.

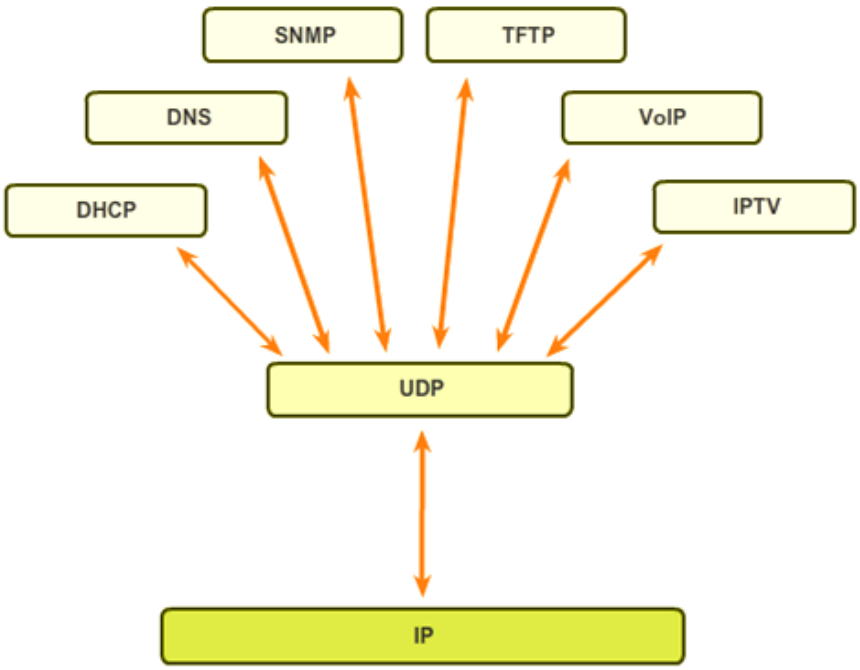


Рис. 7.5. Застосування, що використовують протокол UDP

Деякі застосування, як, наприклад, TFTP, самостійно реалізують функції надійності, що зазвичай притаманні TCP. У таких випадках їм не потрібні сервіси TCP, тому вони використовують UDP.

Протокол TFTP має власні механізми для контролю потоку, виявлення помилок, підтвердження отримання та повторної передачі.

Контрольні запитання та задачі до теми 7

Перелік теоретичних запитань

1. Яке головне призначення транспортного рівня в мережевій моделі та які основні сервіси він надає для взаємодії прикладних програм?
2. У чому полягають ключові переваги протоколу TCP порівняно з іншими транспортними протоколами?
3. Які особливості та переваги характеризують протокол UDP і в яких випадках його доцільно використовувати?
4. Як реалізується система портів у протоколах TCP та UDP і яка її роль у маршрутизації даних між процесами?
5. Які категорії портових номерів існують (відомі, зареєстровані, динамічні) та які приклади можна навести для кожної?
6. Яким чином відбувається розбиття даних на сегменти у TCP і датаграми в UDP?
7. Які етапи і механізми використовуються TCP-сервером для обробки клієнтських запитів?
8. Які особливості має обробка запитів на стороні UDP-сервера і чим вона відрізняється від TCP?
9. Які прикладні протоколи або сервіси зазвичай використовують TCP, а які – UDP, і чому саме?
10. Як TCP забезпечує надійність доставки даних – за рахунок яких технічних механізмів?
11. Чим є "трикратне рукостискання" (three-way handshake) і яку роль воно відіграє у встановленні з’єднання TCP?
12. Яким чином протокол TCP реалізує управління потоком і контроль перевантаження?
13. Як UDP забезпечує мінімальні затримки та які є приклади його використання в реальному часі?
14. У чому полягає різниця між контрольними сумами TCP та UDP, і як вони використовуються для виявлення помилок?
15. Як працює мультиплексування на транспортному рівні, та яку роль у цьому відіграє нумерація портів?

Перелік задач для самостійного опрацювання

Завдання 1. Порівняльний аналіз протоколів TCP і UDP.

Створіть таблицю, у якій порівняйте основні характеристики

протоколів TCP та UDP за такими критеріями: надійність, швидкість, контроль потоку, контроль помилок, впорядкування даних, прикладні протоколи. Зробіть висновок: які переваги має кожен протокол залежно від типу застосування?

Завдання 2. Аналіз мережевого трафіку за допомогою програми Wireshark.

Wireshark – це відкритий мережевий аналізатор, який дозволяє здійснювати детальний моніторинг, фільтрацію та аналіз мережевого трафіку в реальному часі або збережених записів.

Програма підтримує велику кількість мережевих протоколів, має зручний графічний інтерфейс користувача, що дає змогу розшифровувати пакети на різних рівнях мережевої моделі OSI.

Wireshark використовується адміністраторами комп'ютерних мереж, спеціалістами з кібербезпеки, розробниками та викладачами для діагностики неполадок, виявлення вразливостей, вивчення принципів роботи протоколів і перевірки налаштувань мережевого обладнання.

Завантажте програму Wireshark за посиланням: https://www.wireshark.org/download.html (рис. 7.6).

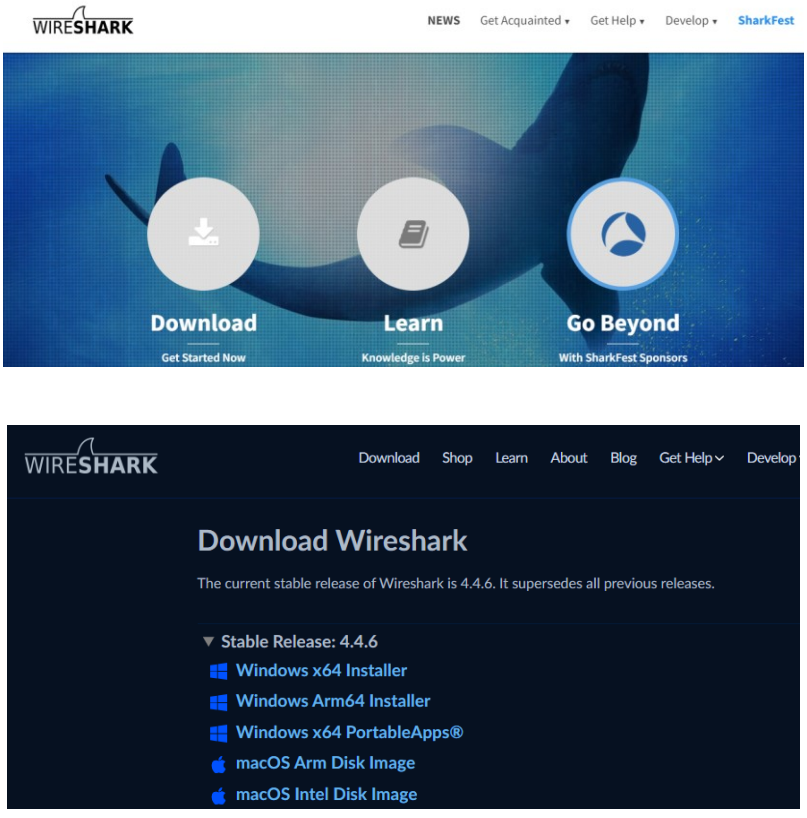


Рис. 7.6. Вікно завантаження програми Wireshark

За допомогою програми Wireshark захопіть фрагмент мережевого трафіку. Відфільтруйте TCP- і UDP-пакети, проаналізуйте структуру заголовків (порти, контрольна сума, прапорці TCP тощо).

Зробіть короткий звіт. Опишіть, які порти використовуються, які застосування передають дані.

Завдання 3. Дослідження портів.

Складіть список найбільш поширених портів (відомих і зареєстрованих), які використовуються різними мережевими сервісами.

За допомогою утиліт (наприклад, *netstat, ss, nmap*) перевірте відкриті порти на власному комп'ютері або віртуальному середовищі.

Зробіть висновки про активні сервіси та рівень мережевої безпеки.

Завдання 4. Візуалізація процесу встановлення TCP-з'єднання.

Зобразіть діаграму процесу встановлення та завершення TCP-з'єднання («трикратного рукостискання» та процес його завершення).

Поясніть значення кожного кроку (SYN, SYN-ACK, ACK, FIN). Укажіть, що відбувається у випадку неповного завершення сесії (наприклад, збої, тайм-аути).

Перелік використаної та рекомендованої літератури

7.1. RFC 793 – Transmission Control Protocol (TCP) [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc793

7.2. RFC 768 – User Datagram Protocol (UDP) [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc768

7.3. IANA Service Name and Transport Protocol Port Number Registry [Електронний ресурс]. – Режим доступу: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

7.4. RFC 9293 – Updated TCP Specification (2022) [Електронний ресурс]. – Режим доступу: https://www.rfc-editor.org/rfc/rfc9293.html

7.5. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

Тема 8. IP-адресація. Розбиття IP-мережі на підмережі

У сучасних комп'ютерних мережах ідентифікація пристроїв та ефективне управління адресним простором є критично важливими завданнями. Саме для цього використовується IP-адресація – система, яка забезпечує унікальність кожного вузла в мережі й дозволяє передавати дані від джерела до призначення. Кожен пристрій у мережі, що використовує протокол IP (Internet Protocol), має власну IP-адресу, яка складається з двох основних частин: мережевої та хостової. Правильне планування та розподіл IP-адрес забезпечує не лише коректну маршрутизацію, а й оптимізацію використання мережевих ресурсів.

Зі зростанням кількості пристроїв у мережах виникає необхідність у поділі великих IP-мереж на менші логічні частини – підмережі. Цей процес, який називається *розбиттям на підмережі*, дозволяє підвищити гнучкість управління мережею, покращити безпеку, зменшити обсяг широкомовного трафіку та ефективніше використовувати обмежений адресний простір, особливо у випадку з IPv4.

8.1. Мережева і вузлова частини IPv4 -адреси

Щоб визначити, чи належать два пристрої до однієї IP-мережі, необхідно розуміти принципи двійкової системи числення. Це пов'язано з тим, що IP-адреса – це структурована адреса, яка складається з двох основних частин: мережевої та вузлової.

У 32-бітному поданні адреси частина бітів визначає саму мережу, а інша – конкретний пристрій (вузол) у цій мережі. Усі пристрої, що належать до однієї мережі, повинні мати однакові біти в мережевій частині своєї IP-адреси.

Водночас, біти вузлової частини мають бути різними для кожного пристрою, щоб забезпечити їх унікальність в межах однієї мережі.

Навіть якщо десяткові значення IP-адрес схожі, це не гарантує, що пристрої знаходяться в одній мережі. Визначальним є збіг бітових комбінацій у тій частині адреси, яка визначена як мережева. Якщо ця частина у двох IP-адрес однакова – вузли належать до однієї мережі.

Для правильного функціонування мережевого пристрою недостатньо лише призначити йому IP-адресу – також обов'язково вказується маска підмережі.

Маска підмережі, як і сама IP-адреса, складається з 32 біт і використовується для визначення межі між мережею і вузлом у структурі адреси. Під час роботи маска підмережі порівнюється з IP-адресою за допомогою побітової операції зліва направо: біти, які мають значення «1», вказують на мережеву частину, а біти з «0» – на частину, що відноситься до конкретного вузла (рис. 8.1).

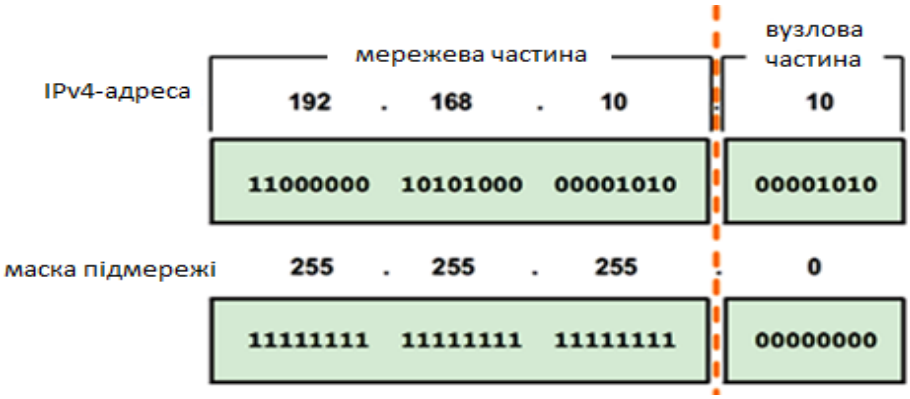


Рис. 8.1. Мережева і вузлова частини IP –адреси

На практиці і IP-адреса, і маска підмережі найчастіше подаються у зручному форматі – чотири десяткові числа, розділені крапками, по 8 біт (октет) у кожному.

Маска задається разом із IP-адресою на мережевому пристрої і дозволяє йому визначити, яка саме частина адреси відноситься до мережі. У таблиці 8.1 наведено приклади можливих варіантів масок підмереж, які використовуються в IPv4 для поділу адресного простору.

Таблиця 8.1. **Допустимі маски підмережі**

Значення підмережі	Значення біта							
	128	**64**	**32**	**16**	**8**	**4**	**2**	**1**
255	1	1	1	1	1	1	1	1
254	1	1	1	1	1	1	1	0
252	1	1	1	1	1	1	0	0
248	1	1	1	1	1	0	0	0
240	1	1	1	1	0	0	0	0
224	1	1	1	0	0	0	0	0
192	1	1	0	0	0	0	0	0
128	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Довжина префікса – це альтернативний спосіб позначення маски підмережі, що дозволяє спростити запис IP-адреси разом із інформацією про мережу та вказує кількість бітів, які у масці підмережі мають значення "1". Ця форма запису виглядає як число після символу косої риски ("/"). Наприклад, маска 255.255.255.0 в двійковому вигляді має 24 одиниці, тому її префіксна форма записується як /24. Таким чином, як традиційний формат маски (наприклад, 255.255.255.0), так і префікс (/24) означають одне й те саме – межу між мережею та вузлом.

Subnet Mask	32-bit Address	Prefix Length
255.0.0.0	11111111.00000000.00000000.00000000	/8
255.255.0.0	11111111.11111111.00000000.00000000	/16
255.255.255.0	11111111.11111111.11111111.00000000	/24
255.255.255.128	11111111.11111111.11111111.10000000	/25
255.255.255.192	11111111.11111111.11111111.11000000	/26
255.255.255.224	11111111.11111111.11111111.11100000	/27
255.255.255.240	11111111.11111111.11111111.11110000	/28
255.255.255.248	11111111.11111111.11111111.11111000	/29
255.255.255.252	11111111.11111111.11111111.11111100	/30

Рис. 8.2. Представлення маски підмережі через довжину префікса

Однак не кожна мережа має префікс саме /24. Довжина префікса вибирається відповідно до кількості пристроїв, які потрібно розмістити в підмережі. Менший префікс (наприклад, /16) дозволяє більше адрес для вузлів, тоді як більший префікс (наприклад, /30) – значно менше. Ця варіативність префіксів змінює доступний діапазон IP-адрес у межах мережі, зокрема кількість адрес для пристроїв та значення широкомовної адреси.

У кожній підмережі IPv4 можна виділити три основні типи IP-адрес. Перший – мережева адреса, яка ідентифікує саму мережу. Вона містить тільки ті біти, які відповідають мережевій частині, решта бітів – нулі. Ця адреса не призначається жодному пристрою, а використовується для загального позначення мережі. Другий тип – адреси вузлів, тобто адреси, які можуть бути присвоєні конкретним пристроям у мережі для їхньої ідентифікації. І третій тип – широкомовна адреса, яка дозволяє одному вузлу надіслати повідомлення всім іншим пристроям у межах тієї ж мережі. Широкомовна IP-адреса є останньою можливою адресою

в межах кожної мережі, і вона формується шляхом заповнення вузлової частини всіма одиницями. Наприклад, у масці /24 широкомовна адреса матиме значення 255 в останньому октеті. Така адреса використовується для одночасної доставки повідомлень усім пристроям у межах певної підмережі. Цей процес називається *прямою широкомовною передачею*, і він є важливим елементом роботи з локальними мережами.

Щоб гарантувати, що всі пристрої мають унікальні IP-адреси в межах мережі, слід визначити допустимий діапазон адрес вузлів. Перша адреса після мережевої – це мінімальна можлива адреса для пристрою, а остання – це максимальна адреса перед широкомовною. Ці адреси розраховуються з урахуванням двійкового представлення, де остання адреса вузла завжди має вузлову частину, що складається з одиниць, окрім крайнього справа біта, який має значення нуль. Значення цієї адреси завжди на одиницю менше, ніж у широкомовної. Таким чином, точне визначення меж дозволяє коректно планувати адресний простір і уникнути конфліктів між пристроями.

8.2. Побітова операція AND

Коли пристрою надається IPv4-адреса, разом із нею він отримує маску підмережі, яка дозволяє визначити, до якої мережі належить цей пристрій. Мережева адреса служить ідентифікатором для групи пристроїв, що функціонують у межах однієї мережі.

Під час передавання пакетів IP-вузол використовує цю інформацію, щоб вирішити, чи можна надіслати пакет безпосередньо, чи його слід передати через шлюз за замовчуванням. Для цього він порівнює свою мережеву частину адреси з адресою призначення, і якщо вони збігаються, передача здійснюється локально.

У випадку відмінності мережевих частин, дані направляються до шлюзу для подальшої маршрутизації. Однією з основних логічних операцій, що використовується при роботі з IP-адресами, є побітова операція AND, яка виконує порівняння двох бітів і повертає результат лише тоді, коли обидва біти дорівнюють одиниці. Таким чином:

1 AND 1 = 1
1 AND 0 = 0
0 AND 1 = 0
0 AND 0 = 0

Щоб дізнатися мережеву адресу пристрою, кожен біт IP-адреси проходить через логічну операцію AND разом із відповідним бітом маски підмережі. Це дозволяє «відсіяти» вузлову частину адреси і залишити лише мережеву.

Під час цієї операції біти IP-адреси, які порівнюються з одиницями в масці підмережі, зберігають своє значення, тоді як біти, що зіставляються з нулями, стають нулями. Внаслідок цього вузлова частина IP-адреси в результаті операції AND занулюється, і формується мережева адреса. Така адреса складається із справжніх бітів мережі, тоді як вузлова частина представлена лише нулями.

Застосування цієї логіки дозволяє визначити, до якої мережі належить IP-адреса, а також які ще адреси знаходяться в межах тієї ж підмережі. Якщо дві IP-адреси після застосування маски підмережі мають однакову мережеву частину, це означає, що вони можуть обмінюватися інформацією безпосередньо, без маршрутизатора. Якщо мережеві частини різні – для передачі даних потрібен пристрій третього рівня, наприклад, маршрутизатор.

У процесі налагодження або перевірки мережі важливо мати змогу визначити, чи належать два пристрої до однієї локальної мережі. Через неправильне налаштування IP-параметрів пристрій може помилково вважати себе частиною іншої мережі. У більшості комп’ютерних мереж крайові пристрої – це комп’ютери, телефони, принтери, планшети тощо, і саме їм потрібно надавати найбільшу кількість IP-адрес. Призначення IP-адрес цим пристроям може здійснюватися як вручну (статично), так і автоматично (динамічно) за допомогою DHCP.

8.3. Одноадресна, широкомовна та багатоадресна передача

У мережах з використанням IPv4 пристрої можуть обмінюватися даними трьома основними способами.

Перший – це одноадресна передача, коли дані надсилаються з одного вузла на інший конкретний пристрій. Другий – широкомовна передача, при якій один вузол надсилає пакет усім пристроям у межах однієї мережі. І третій спосіб – багатоадресна передача, коли дані спрямовуються не всім підряд, а лише обраній групі вузлів, які можуть знаходитись як у тій самій мережі, так і в інших.

У всіх трьох випадках в заголовку пакета вказується IP-адреса пристрою-відправника як адреса джерела. Одноадресна передача є найбільш поширеним методом зв'язку в мережах типу «клієнт-сервер» або однорангових мережах. У цьому випадку в поле адреси призначення пакета вказується IP-адреса конкретного отримувача, і пакет може проходити через проміжні мережі до досягнення цілі.

Широкомовна передача застосовується, коли необхідно звернутися до всіх пристроїв локальної мережі одночасно. Для цього в IP-пакеті вказується адреса, у якій вузлова частина заповнена одиницями, що сигналізує про широкомовне призначення.

Кожен пристрій у локальній мережі, отримавши такий пакет, обробляє його, як і будь-який звичайний. Цей метод використовується в багатьох мережевих протоколах, зокрема, в DHCP, де широкомовні повідомлення допомагають, наприклад, при початковому отриманні IP-адреси. Сфери застосування широкомовних повідомлень включають трансляцію адрес з одного рівня до іншого, надсилання запитів на адресу, а також інші дії, які потребують контакту з усіма пристроями в мережі. Важливо відзначити, що, на відміну від одноадресної передачі, широкомовні пакети зазвичай не виходять за межі локальної мережі. Це обмеження визначається конфігурацією маршрутизатора та типом широкомовного трафіку.

Багатоадресна передача призначена для підвищення ефективності передачі даних у мережі. Вона дозволяє одному пристрою надсилати лише один пакет, який прийматимуть лише ті вузли, що входять до певної групи мультисповіщення. Це значно знижує навантаження на мережу порівняно з одноадресною передачею, де кожному отримувачу надсилається окремий пакет. Наприклад, один пакет може бути одночасно отриманий тисячами вузлів, які підписалися на відповідну групу.

Такий підхід використовується в різних сценаріях – від трансляції відео або аудіо до розповсюдження оновлень програмного забезпечення, організації мережевих ігор та передачі даних у протоколах маршрутизації. У протоколі IPv4 передбачений спеціальний діапазон адрес для багатоадресної передачі – від 224.0.0.0 до 239.255.255.255. Ці адреси поділяються на різні категорії, включаючи локальні і глобальні.

Маршрутизатори, які працюють у мережі, здатні розпізнати, що пакети призначені для локальних груп багатоадресної розсилки, і не передають їх за межі своєї мережі. Такі локальні групи часто використовуються в протоколах маршрутизації для обміну інформацією між маршрутизаторами.

Вузли, які беруть участь у багатоадресній взаємодії, називаються *клієнтами багатоадресної розсилки*. Щоб отримувати потрібні дані, програма на вузлі ініціює підписку на певну групу мультисповіщення. Кожна з таких груп має власну IPv4-адресу, яка ідентифікує її в мережі як отримувача певного трафіку.

8.4. Публічні та приватні IPv4 –адреси

Хоча більшість IPv4-адрес, що використовуються в мережах, є загальнодоступними й призначеними для відкритого доступу через інтернет, існують також спеціальні адреси, які не передбачені для прямого виходу в глобальну мережу. Такі адреси називають приватними, і вони використовуються в закритих або обмежених мережах.

До діапазонів приватних адрес належать такі блоки:

- **10.0.0.0 до 10.255.255.255** (діапазон /8)
- **172.16.0.0 до 172.31.255.255** (діапазон /12)
- **192.168.0.0 до 192.168.255.255** (діапазон /16)

Ці діапазони визначені в стандарті RFC 1918, тому їх іноді називають RFC 1918-адресами. Унікальною рисою приватних адрес є те, що вони можуть повторюватися в різних локальних мережах без конфліктів, оскільки не призначені для використання в публічному інтернеті. Пакети з такими адресами не повинні виходити за межі локальної мережі, а мережеві пристрої, зокрема маршрутизатори та міжмережеві екрани, повинні блокувати або трансформувати ці адреси, наприклад, через NAT.

Загалом, більшість одноадресних IPv4-адрес, які використовуються в мережі Інтернет, є публічними, унікальними та дозволяють пристроям бути досяжними з глобальної мережі. Однак існує також ряд адрес, які або взагалі не призначаються пристроям, або мають обмеження у використанні.

Однією з таких спеціальних адрес є **127.0.0.1** – це так звана *адреса зворотного зв'язку (localhost)* або адреса *loopback*, яка використовується для того, щоб пристрій міг надсилати пакети самому собі. Це дуже зручно для тестування мережевих програм, вебзастосунків, що працюють на одному комп'ютері, або для внутрішньої взаємодії між локальними сервісами. Фактично зарезервований увесь діапазон від 127.0.0.0 до 127.255.255.255, хоча найчастіше застосовується саме адреса 127.0.0.1. Жодна з адрес цього блоку не повинна передаватися мережею – вони існують лише всередині пристрою.

Окрім приватних та loopback-адрес, існує ще один спеціальний блок – це так звані *автоматично призначені адреси (APIPA)*, які використовують діапазон 169.254.0.0 до 169.254.255.255 (тобто 169.254.0.0/16). Операційна система призначає такі адреси у випадку, коли пристрій не може отримати IP-адресу через DHCP-сервер. Це дозволяє організувати мінімальну однорангову мережу без централізованого налаштування.

Комунікація за допомогою локальних адрес цього типу можлива лише в межах однієї фізичної мережі – такі пакети не можуть бути переадресовані маршрутизаторами, а їхній час життя TTL повинен обмежуватись одиницею, щоб уникнути спроб виходу за межі локального сегменту. Проте багато програм і сервісів, особливо з клієнт-серверною або одноранговою архітектурою, можуть нормально працювати у такому середовищі з використанням локальних IP-адрес.

8.5. Обмеження застарілої класової адресації. Безкласова адресація

У минулому IP-адреси поділяли на групи, які називалися класами: A, B, C, D і E. Така класифікація описана в стандарті RFC 1700.

Класи A, B і C використовувалися для одноадресних мереж, тобто для безпосереднього зв'язку між окремими пристроями.

Клас D був зарезервований для групової (мультикаст) передачі, а клас E – для експериментальних або майбутніх цілей. За цією схемою компаніям чи організаціям надавався повноцінний блок адрес певного класу, що відповідало певному розміру мережі. Такий підхід до розподілу називався *класовою адресацією*.

Клас A був призначений для великих мереж, де потрібно понад 16 мільйонів унікальних адрес. Адреси цього класу мали фіксований префікс /8, тобто перший октет вказував на мережу, а три інші – на окремі вузли. Адреси цього класу починалися з бітової послідовності 0, що обмежувало їхню кількість до 128 можливих мереж – від 0.0.0.0/8 до 127.0.0.0/8. Незважаючи на те, що клас A займав половину всього адресного простору, його можна було розподілити лише між приблизно 120 організаціями.

Клас B створено для середніх і великих мереж – він дозволяє мати до 65 000 вузлів у мережі. У цьому класі два перші октети представляли мережеву частину адреси, а два інші – вузли. Початкові біти адрес класу B – це «10», а діапазон охоплює адреси від 128.0.0.0/16 до 191.255.0.0/16. Такий підхід дозволив розподілити 25% всього простору IPv4 між близько 16 000 мереж.

Клас C був найпоширенішим і орієнтований на малі мережі. Його блоки містили максимум 254 адреси вузлів, оскільки три перших октети позначали мережу, а лише останній – пристрої. Всі адреси цього класу починалися з бітової послідовності «110» і охоплювали діапазон від 192.0.0.0/24 до 223.255.255.0/24. Хоча цей клас займав лише 12,5% всього адресного простору IPv4, він дозволив створити понад два мільйони окремих мереж.

Проблеми класової адресації полягали у відсутності гнучкості. Наприклад, якщо компанії потрібно лише трохи більше ніж 254 адреси (що перевищує можливості класу C), їй доводилось отримувати блок класу B, який містить понад 65 000 адрес – більшість з яких залишалася невикористаною. Це призводило до великого розсіювання й втрати IP-адрес. Через ці обмеження система класів поступово вийшла з ужитку наприкінці 1990-х років.

Проте елементи класового підходу ще залишилися: операційні системи досі можуть розпізнавати клас адреси за її значенням і відповідно до цього автоматично встановлювати маску підмережі.

Сучасна практика базується на так званій *безкласовій адресації* або CIDR (Classless Inter-Domain Routing). На відміну від фіксованих довжин префіксів /8, /16 чи /24, які відповідали класам A, B і C, CIDR дозволяє використовувати будь-яку довжину префікса – наприклад, /13

або /20. Такий гнучкий підхід дозволяє ефективніше використовувати обмежений простір IPv4.

CIDR було впроваджено у 1993 році інженерною групою IETF як тимчасове рішення для зменшення витрат адресного простору. Проте вже тоді стало зрозуміло, що IPv4 не зможе довго задовольняти потреби зростаючої кількості пристроїв. Тому в 1994 році було розпочато розробку нового Інтернет-протоколу, яким згодом став протокол IPv6.

8.6. Привласнення IP –адрес

Щоб компанія або організація могла розмістити мережеві вузли, таких як вебсервери, їй необхідно мати публічні IP-адреси. Такі адреси повинні бути унікальними у всій мережі Інтернет і призначаються окремо для кожної організації. Це правило стосується як IPv4-, так і IPv6-адрес. Контроль за їх розподілом здійснюється централізовано.

Головним регулятором IP-адрес є організація IANA (Internet Assigned Numbers Authority), яка координує виділення як IPv4-, так і IPv6-адрес. До середини 1990-х років саме IANA розподіляла увесь доступний IPv4-адресний простір. Згодом, щоб полегшити управління адресами в різних частинах світу, ці функції було передано регіональним інтернет-реєстраторам (*RIR – Regional Internet Registry*), кожен з яких відповідає за певний географічний регіон.

Станом на сьогодні діє п’ять основних регіональних інтернет-реєстраторів (рис. 8.3).

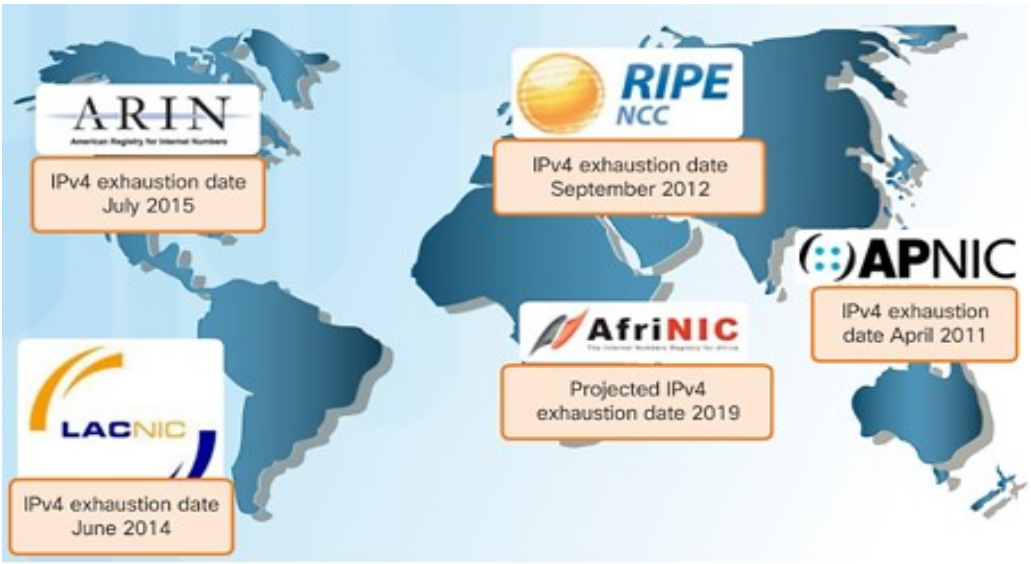


Рис. 8.3. Регіональні інтернет-реєстратори

AfriNIC – відповідає за Африканський континент (www.afrinic.net);

APNIC – охоплює країни Азії та Тихоокеанського регіону (www.apnic.net);

ARIN – працює в Північній Америці (www.arin.net);

LACNIC – обслуговує Латинську Америку і Карибський регіон (www.lacnic.net);

RIPE NCC – відповідає за Європу, Близький Схід і частину Азії (www.ripe.net).

Ці регіональні організації розподіляють IP-адреси інтернет-провайдерам (ISP), які надають адреси своїм клієнтам – компаніям або приватним користувачам. Як правило, компанія отримує невеликий пул адрес (наприклад, 6 або 14), що вистачає для типових потреб. Якщо ж потрібна більша кількість, її можна замовити за додаткову плату.

Фактично, інтернет-провайдер тимчасово надає адреси своїм клієнтам. У разі переходу до іншого провайдера, нова компанія виділяє адреси зі свого власного пулу, а старі повертаються до попереднього провайдера для подальшого використання.

У випадку з IPv6-адресами компанії також можуть отримати їх через провайдера або безпосередньо від одного з регіональних реєстраторів, якщо це передбачено політикою RIR.

Для підключення до Інтернету організація повинна скористатися послугами провайдера, який забезпечує доступ до глобальної мережі через свою інфраструктуру. Провайдери не тільки забезпечують зв’язок, а й пропонують додаткові сервіси: систему доменних імен DNS, електронну пошту, хостинг вебсайтів тощо. Рівень сервісу залежить від класу провайдера та потреб клієнта – від базових послуг до повноцінного інфраструктурного обслуговування.

8.7. Потреба в IPv6. Представлення IPv6

Протокол IPv6 був створений як наступник IPv4, з урахуванням усіх його недоліків. Основною перевагою IPv6 є величезний 128-бітовий адресний простір, якого достатньо для приблизно **340 ундециліонів унікальних адрес** – це число 340 з 36 нулями (рис. 8.4). Але IPv6 – це не лише про більшу кількість адрес.

Number Name	Scientific Notation	Number of Zeros
1 Thousand	10^3	1,000
1 Million	10^6	1,000,000
1 Billion	10^9	1,000,000,000
1 Trillion	10^12	1,000,000,000,000
1 Quadrillion	10^15	1,000,000,000,000,000
1 Quintillion	10^18	1,000,000,000,000,000,000
1 Sextillion	10^21	1,000,000,000,000,000,000,000
1 Septillion	10^24	1,000,000,000,000,000,000,000,000
1 Octillion	10^27	1,000,000,000,000,000,000,000,000,000
1 Nonillion	10^30	1,000,000,000,000,000,000,000,000,000,000
1 Decillion	10^33	1,000,000,000,000,000,000,000,000,000,000,000
1 Undecillion	10^36	1,000,000,000,000,000,000,000,000,000,000,000,000

Legend

There are 4 billion IPv4 addresses

There are 340 undecillion IPv6 addresses

Рис. 8.4. Порівняння адресного простору IPv4 та IPv6

У процесі розробки спеціалісти з IETF також виправили технічні обмеження IPv4 і додали нові можливості для підвищення ефективності роботи мереж.

Основним стимулом для переходу на IPv6 стало те, що адресний простір IPv4 почав стрімко вичерпуватись. Особливо це відчутно в умовах зростаючого попиту на Інтернет у регіонах, таких як Азія і Африка.

31 січня 2011 року IANA передала останні два /8-блоки IPv4-адрес регіональним інтернет-реєстраторам.

Максимальна кількість IPv4-адрес теоретично обмежена 4,3 мільярдами. Для сповільнення вичерпання IPv4 використовуються приватні адреси (RFC 1918) у поєднанні з технологією NAT (трансляція мережевих адрес), але NAT обмежує можливість прямого з'єднання між пристроями.

Щоб полегшити перехід до IPv6, були розроблені спеціальні технічні рішення, які дозволяють поступово адаптувати мережі.

Розглянемо три основні методи переходу до протколу IPv6.

1. *Подвійний стек* – дозволяє одночасну підтримку як IPv4, так і IPv6 в одному середовищі. Пристрої з подвійним стеком можуть працювати з обома протоколами паралельно.

2. *Тунелювання* – полягає в інкапсуляції IPv6-пакетів у пакети IPv4 для передачі через вже існуючу IPv4-мережу.

3. *Перетворення NAT* – за допомогою NAT64 пакети з IPv6 можуть бути перетворені в IPv4 і навпаки, що дозволяє взаємодію між пристроями з різними протоколами.

IPv6-адреса має довжину 128 біт і записується як рядок з 32 шістнадцяткових цифр. Кожні 4 біти представляються однією цифрою у шістнадцятковій системі, незалежно від регістру літер. Стандартна форма запису має вигляд: x:x:x:x:x:x:x:x, де кожне "x" – це чотиризначне шістнадцяткове число, що відповідає 16 бітам.

Оскільки повний запис адреси часто буває громіздким, передбачено два правила скорочення.

1. *Правило опущення початкових нулів* – у кожному сегменті можна не писати ведучі нулі (рис. 8.5). Наприклад, 009F0 скорочується до 9F0.

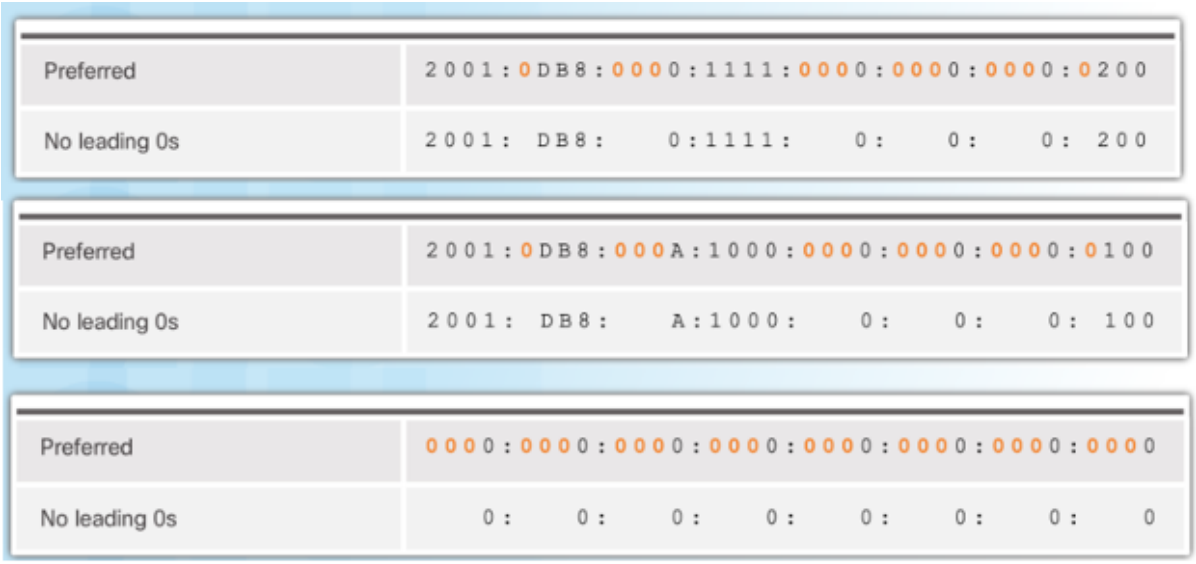


Рис. 8.5. Правило опущення початкових нулів в IPv6-адресі

2. *Правило стиснення нульових блоків* – поспіль розташовані сегменти, що складаються лише з нулів, можуть бути замінені подвійною двокрапкою ::. Це скорочення дозволене лише один раз у

кожній адресі, щоб уникнути неоднозначностей. Таке представлення називається *стислим форматом* (рис. 8.6).

Рис. 8.6. Правило стиснення нульових блоків в IPv6-адресі

IPv6 підтримує три типи адрес:

- *унікальні (Unicast)* – використовуються для ідентифікації окремого пристрою в мережі;
- *групові (Multicast)* – дозволяють відправити пакети одразу кільком пристроям;
- *будь-які (Anycast)* – адреса присвоюється кільком пристроям, але трафік надсилається лише до найближчого.

На відміну від IPv4, протокол IPv6 не має широкомовної розсилки (broadcast). Натомість він використовує спеціальну групову адресу, яка забезпечує подібну функціональність.

IPv6 використовує префікси для вказівки мережевої частини адреси. Тут не застосовується десяткове представлення маски, як у IPv4. Замість цього вказується довжина префікса у вигляді /число, наприклад, /64. Це означає, що перші 64 біти адреси належать до мережі, а решта – ідентифікує конкретний інтерфейс.

Глобальні унікальні IPv6-адреси, які можуть маршрутизуватися через Інтернет, еквівалентні публічним адресам IPv4. Організація IANA виділяє блоки таких адрес регіональним інтернет-реєстраторам. Наразі призначаються лише адреси з префіксом 2000::/3, тобто з першими трьома бітами 001. Це становить приблизно одну восьму всього адресного простору IPv6.

Типова глобальна унікальна адреса складається з трьох наступних компонентів.

Префікс глобальної маршрутизації – визначає мережу, до якої належить пристрій.

Ідентифікатор підмережі – використовується для поділу адресного простору всередині організації.

Ідентифікатор інтерфейсу – індивідуальна частина, яка визначає сам пристрій.

Наприклад, адреса 2001:0DB8:ACAD::/48 має префікс глобальної маршрутизації довжиною 48 біт (2001:0DB8:ACAD). Подвійна двокрапка вказує на сегменти з нулями, які не записуються повністю.

Для автоматичної конфігурації IPv6-адрес використовується технологія SLAAC (Stateless Address Autoconfiguration), завдяки якій пристрій може самостійно отримати адресу, довжину префікса і шлюз за замовчуванням, базуючись на інформації, отриманій із повідомлень ICMPv6 Router Advertisement, без необхідності залучення DHCPv6-серверів.

8.8. Розбиття мережі на підмережі

Сегментація мережі шляхом поділу її на менші логічні частини називається *розбиттям мережі на підмережі*. Адміністратори можуть формувати підмережі за географічним принципом (наприклад, обладнання на третьому поверсі), за організаційною структурою (наприклад, відділ продажів), типом обладнання (сервери, принтери,

маршрутизатори) або іншим логічним критерієм. Такий підхід дозволяє зменшити навантаження на мережу і підвищити її ефективність.

Для забезпечення зв'язку між пристроями з різних підмереж потрібен маршрутизатор. Кожен пристрій у мережі надсилає трафік за межі своєї підмережі через інтерфейс маршрутизатора, який виступає в ролі шлюзу за замовчуванням.

Якщо трафік призначено пристрою, що знаходиться в іншій мережі, маршрутизатор обробляє цей трафік і перенаправляє його у потрібному напрямку. Для визначення, чи є призначення локальним або віддаленим, використовується маска підмережі.

Розбиття мережі на підмережі дозволяє розділити одну мережеву адресу або пул адрес на декілька логічно ізольованих мереж. Кожна підмережа має власний адресний простір. Пристрої, що належать до певної підмережі, повинні мати IP-адресу, відповідну маску підмережі та правильний шлюз за замовчуванням.

Обмін трафіком між підмережами можливий лише за допомогою маршрутизатора. Кожен його інтерфейс повинен мати IP-адресу, яка належить до тієї підмережі, до якої він підключений.

Кожна мережева адреса охоплює певний діапазон можливих адрес вузлів. Усі пристрої в межах однієї мережі мають IP-адреси з цього діапазону, а також спільну маску підмережі або префікс.

Префікс і маска підмережі – це два способи представлення однієї і тієї ж інформації про мережеву частину IP-адреси (рис. 8.7).

Prefix Length	Subnet Mask	Subnet Mask in Binary (n = network, h = host)	# of hosts
/8	255.0.0.0	nnnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh 11111111.00000000.00000000.00000000	16,777,214
/16	255.255.0.0	nnnnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh 11111111.11111111.00000000.00000000	65,534
/24	255.255.255.0	nnnnnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh 11111111.11111111.11111111.00000000	254

Рис. 8.7. Приклади представлення префікса та маски підмережі

Маска підмережі (наприклад, 255.255.255.0) показує, які біти адреси належать до мережі, а які – до вузлів.

Префікс підмережі (наприклад, /24) – це скорочене числове представлення кількості бітів, що належать до мережевої частини. Обидва варіанти використовуються для визначення меж підмережі та розрахунку діапазону доступних IP-адрес у ній.

Щоб створити підмережі в IPv4, частина бітів з вузлової частини адреси перетворюється на частину мережі – це і є «запозичення» бітів. Чим більше бітів запозичується, тим більше підмереж можна створити. Наприклад, 1 запозичений біт дозволяє створити 2 підмережі, 2 біти – 4, 3 біти – 8 тощо. Однак при цьому зменшується кількість доступних адрес вузлів у кожній підмережі. Запозичення бітів можливе лише з вузлової частини адреси. Мережева частина фіксується постачальником послуг і не може бути змінена.

Перші три октети IP-адреси становлять мережеву частину IP-адреси (n – network), четвертий октет – вузлову частину (h – host).

Для визначення кількості підмереж використовується формула *2^n*, де *n* – кількість запозичених (borrowing) бітів (рис. 8.8).

Кількість вузлів у кожній підмережі визначається як *2^n - 2*, де *n* – кількість бітів, що залишилися для адресації вузлів (рис. 8.9).

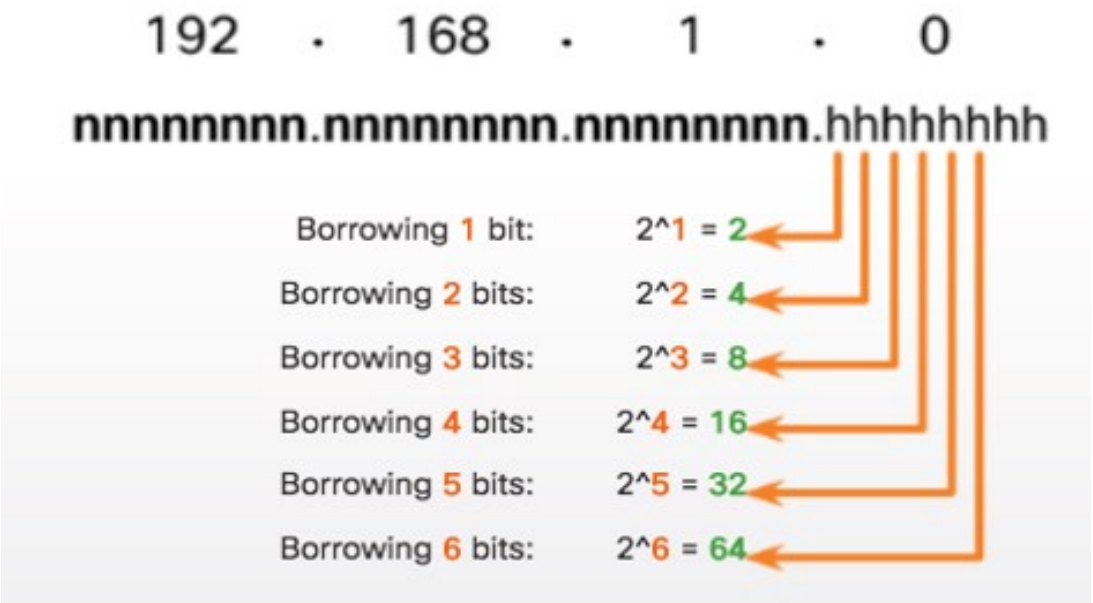


Рис. 8.8. Приклади запозичення бітів з вузлової частини IP-адреси (h – host).

Prefix Length	Subnet Mask	Subnet Mask in Binary (n = network, h = host)	# of subnets	# of hosts
/25	255.255.255.128	nnnnnnnn.nnnnnnnn.nnnnnnnn.nhhhhhhh 11111111.11111111.11111111.10000000	2	126
/26	255.255.255.192	nnnnnnnn.nnnnnnnn.nnnnnnnn.nnhhhhhh 11111111.11111111.11111111.11000000	4	62
/27	255.255.255.224	nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnhhhhh 11111111.11111111.11111111.11100000	8	30
/28	255.255.255.240	nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnhhhh 11111111.11111111.11111111.11110000	16	14

Рис. 8.9. Приклади розрахунку кількості підмереж та вузлів

У деяких випадках важливіше мати конкретну кількість підмереж, ніж багато адрес для кожної з них.

Наприклад, якщо потрібно логічно розділити мережу за внутрішньою структурою компанії – об’єднати всі пристрої технічного відділу в одну підмережу, а пристрої керівництва – в іншу. У такій ситуації першочерговим критерієм стає кількість підмереж.

Під час планування адресації мережеві адміністратори повинні передбачити достатній простір для майбутнього розширення – кожна підмережа має вміщувати не лише наявні, а й потенційні пристрої.

У традиційній схемі підмережування кожній підмережі надається рівна кількість IP-адрес. Це ефективно, коли всі підмережі мають схожі вимоги до кількості вузлів. Однак такий підхід не завжди раціональний – іноді багато адрес залишаються невикористаними.

Наприклад, для трьох WAN-з’єднань потрібно по дві адреси в кожній підмережі, але стандартна маска /27 надає 30 адрес, з яких 28 будуть зайвими. Це не лише неекономно, а й зменшує загальну кількість підмереж, які можна створити. Така неефективність властива класичному підходу до розбиття на підмережі.

На рис. 8.10 показано, як при традиційному розбитті на підмережі всі підмережі мають однаковий розмір та використовують єдину маску підмережі.

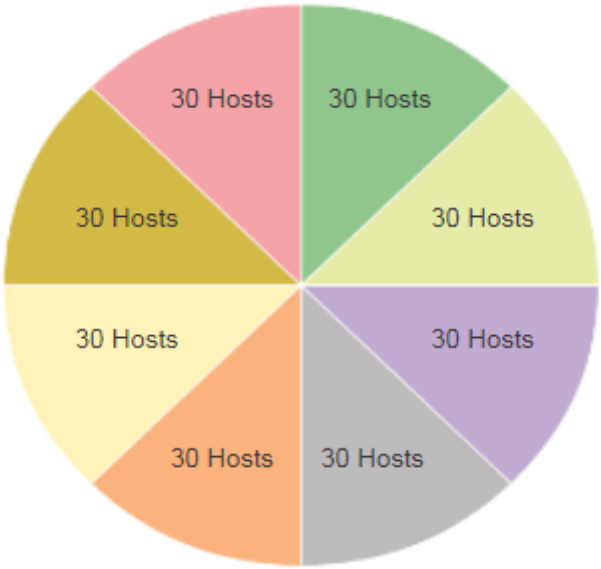


Рис. 8.10. Традиційне розбиття на підмережі
(створення підмереж однакового розміру)

На відміну від традиційного підходу, розбиття на підмережі із застосуванням маски змінної довжини VLSM (Variable Length Subnet Mask) дозволяє ефективніше використовувати адресний простір. Цей підхід дає змогу створювати підмережі різного розміру, залежно від кількості пристроїв (рис. 8.11).

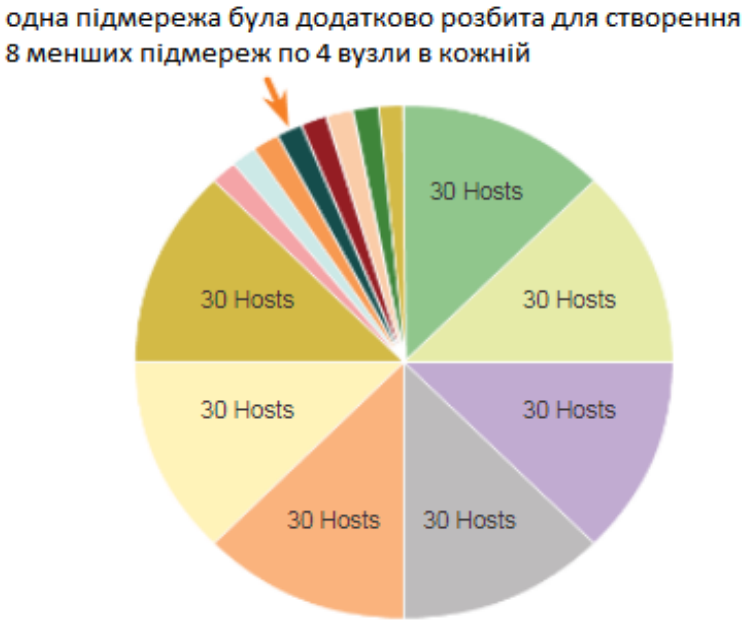


Рис. 8.11. Підмережі змінного розміру

На рис. 8.11 показано, що VLSM дозволяє задавати різну довжину маски підмережі в залежності від кількості запозичених бітів, тобто кожна підмережа може мати свій розмір.

Як і при традиційному розбитті на підмережі, для VLSM використовуються ті ж самі формули для обчислення кількості підмереж і адрес вузлів.

Основна різниця полягає в тому, що VLSM дозволяє поетапно ділити мережу: спочатку – на великі підмережі, потім – кожну з них ще на менші. Цей процес можна повторювати стільки разів, скільки потрібно.

При плануванні адресного простору в організаційній мережі слід врахувати наступні ключові аспекти.

Уникнення дублювання адрес: кожен вузол повинен мати унікальну адресу. Відсутність чіткої схеми може призвести до конфліктів і втрати доступу до мережі.

Контроль доступу: адреси серверів часто використовуються для керування доступом. Якщо адреса обрана випадково і не задокументована, забезпечити належний контроль буде складно.

Моніторинг безпеки та продуктивності: правильне планування адресації дозволяє швидко виявляти пристрої, що спричиняють підозрілу або надмірну активність у мережі.

У мережах використовується різноманітне обладнання, зокрема:

- клієнтські пристрої користувачів;
- сервери та периферійні пристрої;
- вузли, доступні з Інтернету;
- проміжні мережеві пристрої;
- шлюзи.

Під час розробки схеми IP-адресації доцільно використовувати уніфіковану структуру призначення адрес кожному типу пристроїв. Це полегшує масштабування, фільтрацію трафіку та ведення документації.

Контрольні запитання та задачі до теми 8

Перелік теоретичних запитань

3. Що таке IP-адреса і які її основні функції в мережі?
4. Як розділяється IPv4-адреса на мережеву та вузлову частини?

5. Яке призначення побітової операції AND у визначенні мережевої адреси?

6. У чому полягає різниця між одноадресною (unicast), широкомовною (broadcast) та багатоадресною (multicast) передачею?

7. Які IP-адреси вважаються приватними, а які – публічними? Наведіть приклади.

8. Які обмеження має класова IP-адресація та в чому переваги безкласової?

9. Що таке маска підмережі та як вона використовується?

10. Як працює префікс у позначенні підмережі? У чому його різниця з маскою підмережі?

11. Яка формула використовується для розрахунку кількості підмереж та кількості вузлів у підмережі?

12. Що таке VLSM і в чому перевага VLSM над традиційною схемою розбиття?

13. Які існують способи призначення IP-адрес (динамічне, статичне, DHCP)?

14. Чому виникла потреба в IPv6 і як виглядає його формат?

15. Які типи адрес підтримує IPv6?

Перелік задач для самостійного опрацювання

Завдання 1.

Визначити мережеву адресу, маску підмережі та діапазон доступних вузлів для IP-адреси 192.168.10.25/27. Виконати побітову операцію AND між IP-адресою і маскою підмережі.

Завдання 2.

Визначити, чи є IP-адреса 10.0.0.25 публічною чи приватною.

Завдання 3.

Представити IPv6-адресу у скороченій формі: 2001:0db8:0000:0000:0000:ff00:0042:8329.

Завдання 4.

Обчислити кількість доступних вузлів у мережі з префіксом /26.

Завдання 5.

Університет у Польщі (університет А) використовує технологію відеоконференцій для встановлення взаємодії студентів з університетом

В в Україні. Відеоконференція проводиться між двома кінцевими пристроями через мережу Інтернет. Адміністратор мережі університету А налаштовує кінцевий пристрій з IP-адресою 209.165.201.10 та надсилає запит на IP-адресу для кінцевого пристрою в університеті В і отримує таку відповідь – 192.168.25.10. Жоден університет не використовує VPN. Адміністратор університету А відразу розуміє, що ця IP-адреса не працюватиме. Чому? Відповідь пояснити.

Завдання 6.

У вас є мережа з адресою 192.168.1.0/24, яку потрібно розділити на 4 однакові підмережі для розміщення в різних відділах компанії.

Визначте нову маску підмережі.

Вкажіть діапазон IP-адрес для кожної з 4 підмереж (початкова IP-адреса, кінцева IP-адреса, мережева та широкомовна адреси).

Намалюйте схему підключення зображаючи 4 відділи як окремі підмережі.

Перелік використаної та рекомендованої літератури

8.1. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

8.2. Internet Standard Subnetting Procedure [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc950

8.3. Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc4632

8.4. CIDR and Classful Routing [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc1817

8.5. Configure IP Addresses and Unique Subnets for New Users [Електронний ресурс]. – Режим доступу: https://www.cisco.com/c/en/us/support/docs/ip/routing-information-protocol-rip/13788-3.html#toc-hId--1412502249

Тема 9. Протоколи та сервіси прикладного рівня

Сучасні комп'ютерні мережі забезпечують користувачам широкий спектр послуг – від перегляду вебсторінок і передачі електронної пошти до відеоконференцій та обміну файлами. Усі ці можливості реалізуються завдяки роботі протоколів та сервісів прикладного рівня, які розташовані на верхньому рівні моделі OSI та TCP/IP.

Прикладний рівень забезпечує безпосередню взаємодію користувача з мережею, використовуючи програмні застосунки та стандартизовані протоколи обміну даними. На цьому рівні працюють такі поширені протоколи, як HTTP/HTTPS (вебдоступ), FTP (передача файлів), DNS (резолюція доменних імен), SMTP/POP3/IMAP (електронна пошта) тощо. Вони дозволяють різним програмам і пристроям узгоджено обмінюватися даними, дотримуючись визначених правил і форматів. Знання роботи прикладного рівня є необхідною основою для розуміння принципів роботи мережевих застосунків та побудови ефективних мережевих сервісів.

У цьому розділі розглядаються основні функції прикладного рівня, найпоширеніші протоколи та принципи їх роботи. Особлива увага приділяється тому, як прикладні сервіси використовуються у мережевому середовищі, а також ролі клієнт-серверної та одноранговой моделей взаємодії.

9.1. Функції і протоколи прикладного рівня TCP/IP

Прикладний рівень є найвищим рівнем моделі OSI і знаходиться найближче до кінцевого користувача. Саме на цьому рівні реалізується взаємодія між програмами користувача та мережею, що забезпечує передачу даних.

Як проілюстровано на рис. 9.1, прикладний рівень відповідає за організацію комунікації між застосунками на різних пристроях – наприклад, між веббраузером користувача і вебсервером або між поштовим клієнтом і поштовим сервером.

Таким чином, саме завдяки цьому рівню ми можемо переглядати вебсторінки, надсилати листи електронною поштою або передавати файли по мережі.

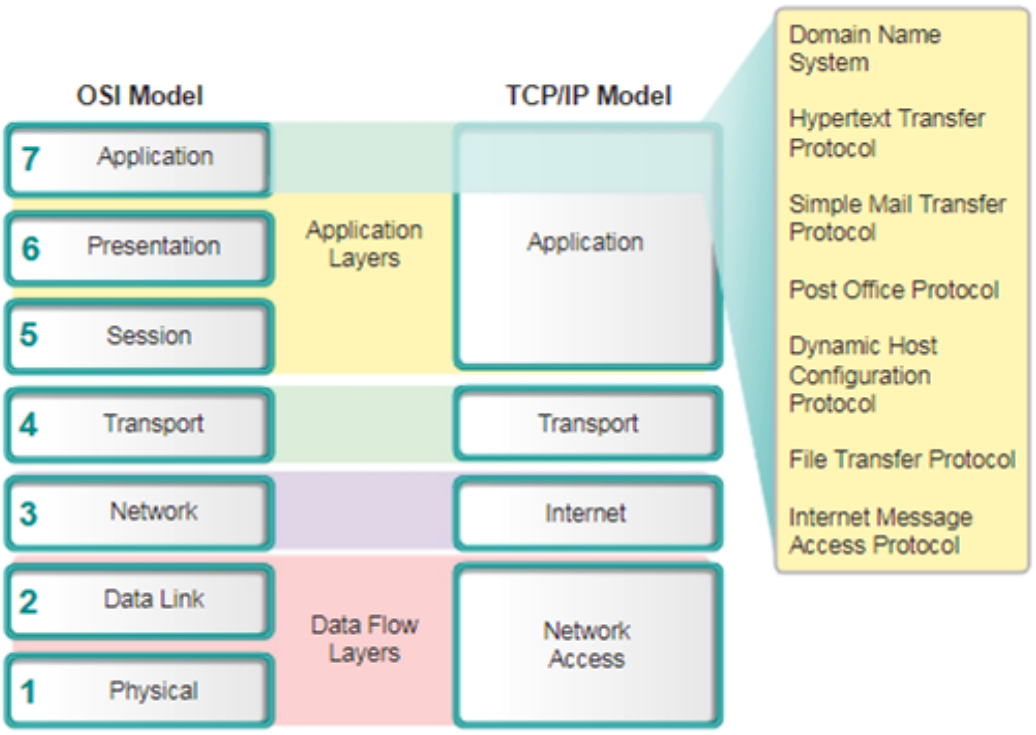


Рис. 9.1. Протоколи прикладного рівня

Клієнт-серверна модель – це мережева архітектура, у якій клієнтський пристрій ініціює запит на доступ до певного ресурсу або послуги, а сервер обробляє цей запит і надає відповідь. У цій моделі сервери зазвичай централізовано зберігають дані, керують обліковими записами користувачів і забезпечують доступ до ресурсів, тоді як клієнти виступають як споживачі цих послуг (рис. 9.2).

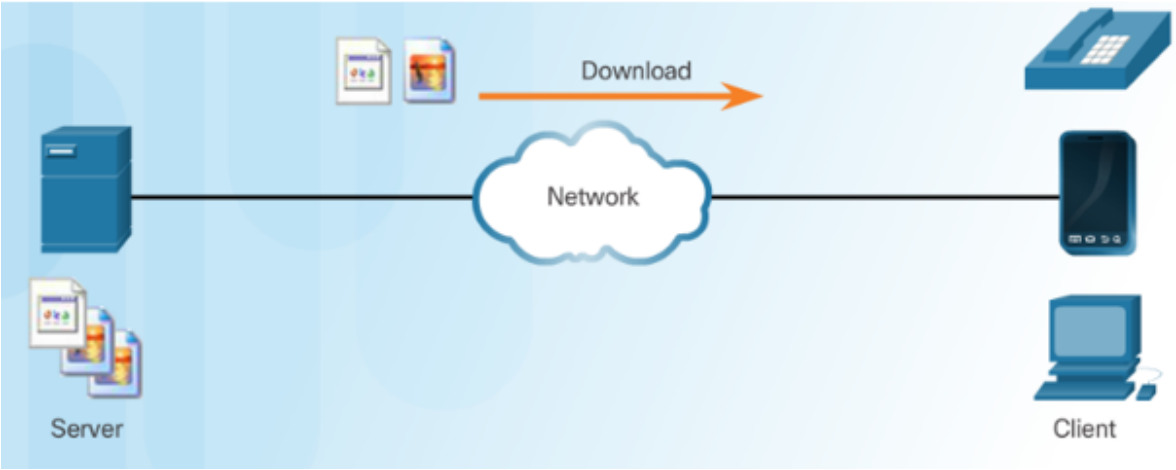


Рис. 9.2. Клієнт-серверна модель взаємодії

Передача даних може відбуватися в обох напрямках, але частіше за все – від сервера до клієнта. Клієнт-серверна модель широко застосовується у вебсервісах, електронній пошті, базах даних та корпоративних системах, забезпечуючи ефективне управління, безпеку та масштабованість мережі.

Протоколи прикладного рівня – це набір правил, що регулюють спосіб обміну даними між застосунками. До найвідоміших належать:

- HTTP (HyperText Transfer Protocol) – забезпечує передачу вебсторінок через мережу Інтернет;
- FTP (File Transfer Protocol) – дозволяє передавати файли між комп'ютерами;
- TFTP (Trivial File Transfer Protocol) – варіант FTP, який не потребує встановлення з'єднання;
- IMAP (Internet Message Access Protocol) – використовується для доступу до електронної пошти, дозволяє працювати з поштою без її завантаження;
- DNS (Domain Name System) – перетворює доменні імена, наприклад www.example.com, на IP-адреси, зрозумілі мережевому обладнанню.
- SMTP (Simple Mail Transfer Protocol) – передає електронну пошту.
- DHCP (Dynamic Host Configuration Protocol) – автоматично присвоює IP-адресу пристрою.
- POP (Post Office Protocol) – дозволяє отримувати пошту.

Наступним є рівень представлення, який виконує такі основні функції:

- *форматування даних* – перетворення інформації у формат, зрозумілий пристрою-одержувачу (наприклад, текстовий документ може бути перетворений у PDF чи HTML);
- *стиснення* – зменшення обсягу даних для ефективнішої передачі (наприклад, стиснення зображень у JPEG);
- *шифрування* – забезпечення безпеки передачі (наприклад, HTTPS шифрує дані HTTP за допомогою SSL/TLS).

На цьому рівні встановлюються стандарти для різних типів даних. Наприклад, QuickTime і MPEG – це популярні формати відеофайлів. QuickTime використовується переважно на пристроях Apple, тоді як MPEG став універсальним стандартом стискування відео (рис. 9.3).

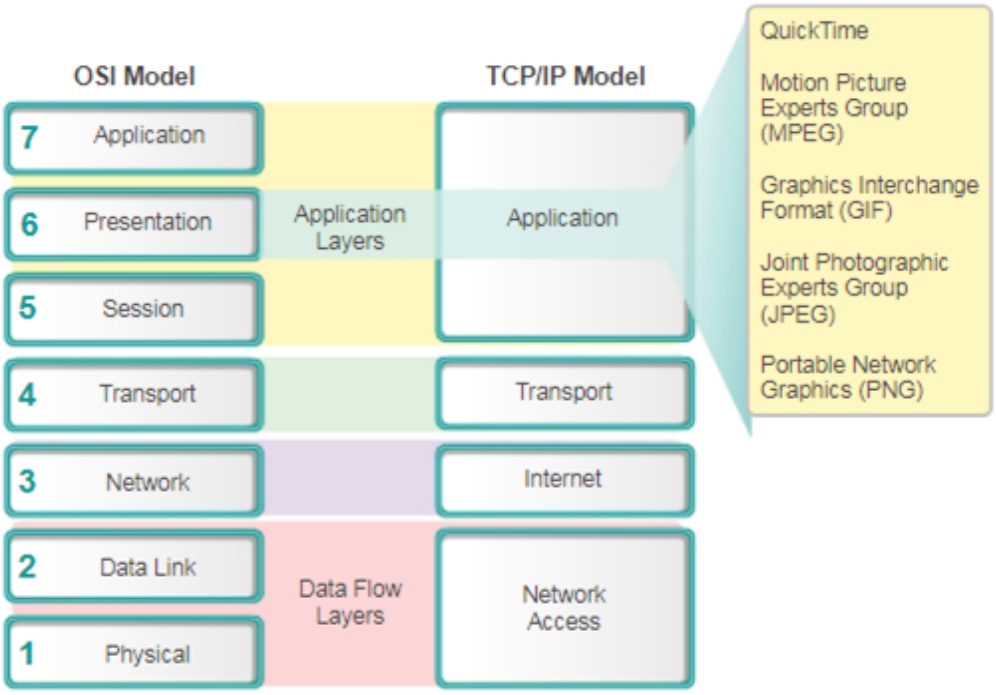


Рис. 9.3. Стандарти рівня представлення

Для графічних зображень використовуються такі формати:

- *GIF (Graphics Interchange Format)* – для простих анімацій;
- *JPEG (Joint Photographic Experts Group)* – для фотографій із високим стисненням;
- *PNG (Portable Network Graphics)* – підтримує прозорість і є вдосконаленим форматом, що замінив формат GIF.

На сеансовому рівні встановлюється і підтримується зв’язок між програмами. Цей рівень дозволяє, наприклад, браузеру підтримувати активну сесію з вебсервером, або дозволяє сервісу передачі файлів відновити сеанс після переривання.

У моделі TCP/IP функції трьох рівнів OSI – прикладного, представлення і сеансового – об'єднані, і більшість застосунків реалізує їх одночасно.

Для повсякденних користувачів найчастіше використовуються:

HTTP – для перегляду вебсторінок;

SMTP – для надсилання пошти;

POP – для отримання пошти.

Наприклад, коли ми вводимо адресу в адресному рядку браузера (наприклад, http://www.cisco.com/index.html), браузер використовує протокол HTTP для зв'язку з сервером. Він розпізнає:

http – протокол;

www.cisco.com – доменне ім'я сервера;

index.html – конкретний файл, який потрібно отримати.

Ім'я сервера перетворюється через DNS у IP-адресу. Потім браузер надсилає запит GET на сервер, отримує HTML-код сторінки й відображає її користувачу.

Веббраузери можуть відображати як простий текст, так і складні HTML-сторінки. У деяких випадках для перегляду специфічного вмісту потрібні додаткові програми або розширення.

Щоб браузер міг зрозуміти, який тип даних надходить, сервер вказує це в заголовку відповіді. Такий підхід до передачі даних – з використанням протоколів прикладного рівня – є основою функціонування мережі Інтернет, електронної пошти, обміну файлами й усіх популярних вебсервісів.

9.2. Одноранговi мережі

У сучасних комп'ютерних мережах дані, до яких звертається користувач, не завжди зберігаються на його пристрої.

Наприклад, якщо ми відкриваємо документ з хмарного сховища або переглядаємо відео на стримінговій платформі, самі дані фізично розташовані на іншому пристрої, часто віддаленому.

У такому випадку відбувається запит на отримання даних до пристрою, де вони зберігаються. Цей запит може бути оброблений централізовано (через сервер) або децентралізовано – в *одноранговій моделі* (peer-to-peer, P2P).

На рис. 9.4 показано роботу одноранговоі мережі, у якій два комп'ютери – Peer1 і Peer2 – виконують як клієнтські, так і серверні функції. Peer1 надає доступ до файлів для Peer2 і надсилає завдання на друк через Peer2, який виступає як сервер друку. Peer2, у свою чергу, звертається до файлів на Peer1 і друкує файл, отриманий від Peer1, використовуючи локально під'єднаний принтер. Така конфігурація демонструє, як комп'ютери в одноранговій мережі можуть обмінюватися ресурсами без використання централізованого сервера.

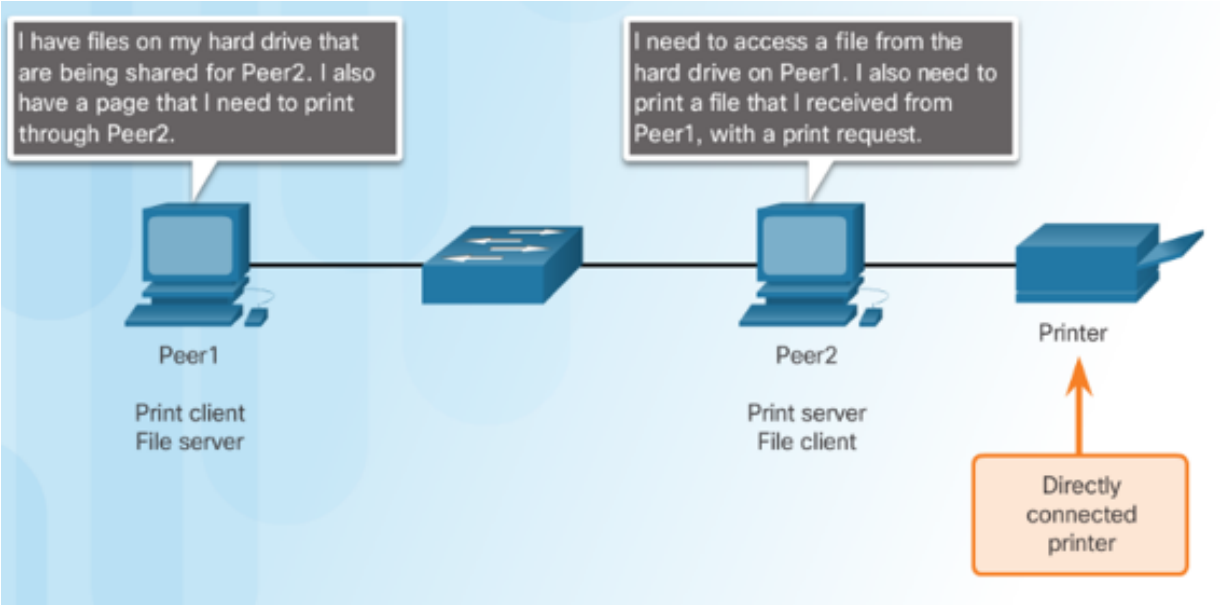


Рис. 9.4. Приклад однорангової мережі

P2P-мережа – це така мережа, в якій усі пристрої мають однакові права і можуть як надавати, так і отримувати ресурси без централізованого сервера. Наприклад, у домашній локальній мережі комп'ютери можуть обмінюватися файлами або надсилати документи на загальний принтер без використання окремого сервера.

Усі пристрої – кінцеві або крайові (тобто знаходяться на «краю» мережі) – можуть бути одночасно і клієнтами, і серверами.

Наприклад, ПК може надавати доступ до музичних файлів (сервер) і водночас отримувати відео з іншого пристрою (клієнт).

Децентралізованість є ключовою перевагою P2P, оскільки дані не зберігаються на одному центральному сервері, а розподілені по різних пристроях. Наприклад, у студентській мережі гуртожитку користувачі відкрили загальний доступ до музичної теки на своїх ноутбуках. Інші користувачі можуть слухати ці файли безпосередньо, без використання серверної системи.

Однак є й недоліки: відсутність централізованого управління, що ускладнює застосування політик безпеки, наприклад, обмеження доступу до певних файлів. Також права доступу потрібно налаштовувати окремо на кожному пристрої.

P2P-додаток – це програма, яка дозволяє пристроям безпосередньо обмінюватися даними у рамках однієї сеансної сесії. Такий додаток запускає користувацький інтерфейс; відкриває фонові сервіси для

прийому й передачі даних; дозволяє працювати в режимі "і клієнт, і сервер одночасно".

BitTorrent – популярний P2P-додаток для обміну файлами. Користувач завантажує частини файлу з різних комп'ютерів, одночасно надаючи іншим доступ до тих частин, які вже є у нього.

У деяких випадках P2P-додатки поєднують децентралізовану передачу даних із централізованим пошуком. Така модель має центральний індекс-сервер, який:

- зберігає інформацію про місце розташування файлів;
- допомагає знайти, на якому пристрої знаходиться файл;
- після цього дозволяє пристроям обмінюватися даними безпосередньо. Наприклад, клієнт eMule звертається до серверу індексації, щоб знайти, хто має потрібний файл, а далі безпосередньо підключається до цього пристрою для завантаження.

Таблиця 9.1. **Типові P2P-додатки**

Назва	Призначення
eDonkey / eMule	Обмін великими файлами
Shareaza	Підтримує різні P2P-мережі
BitTorrent	Децентралізована передача частин файлу
Bitcoin	Мережа для криптовалютних транзакцій
LionShare	Академічна мережа обміну ресурсами
Gnutella-клієнти (BearShare, LimeWire тощо)	Доступ до файлів через Gnutella-протокол

Gnutella – це відкритий протокол P2P-мережі, який дозволяє:

- підключатися до інших пристроїв через Інтернет;
- обмінюватися файлами без централізованого сервера (рис. 9.5).

Популярні Gnutella-клієнти:

- Gnucleus;
- BearShare;
- LimeWire;
- Morpheus;
- WinMX;
- XoloX.

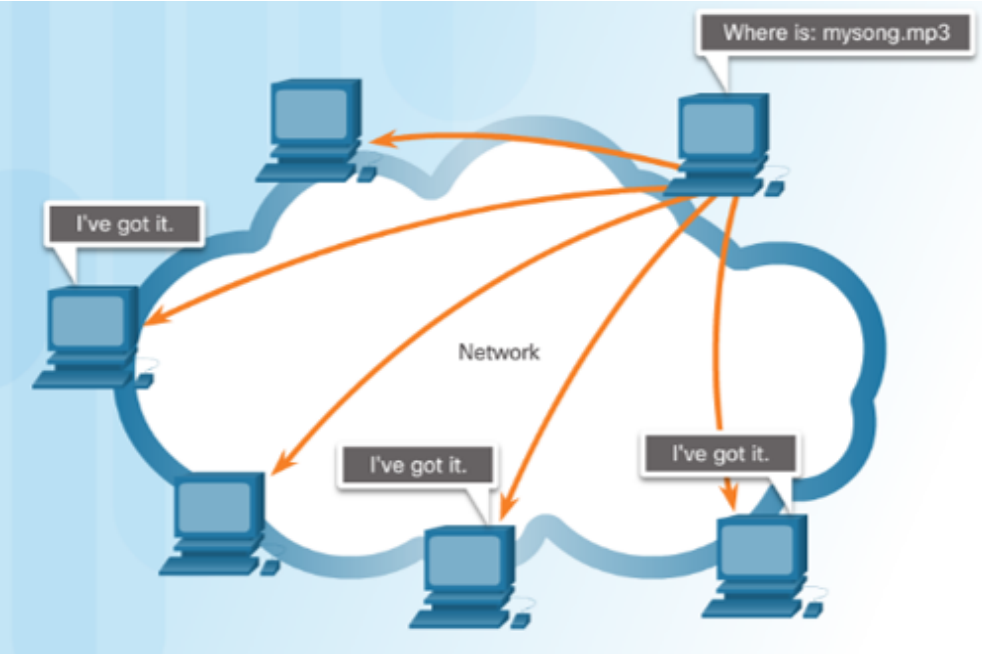


Рис. 9.5. Приклад використання протоколу Gnutella

Розробники часто створюють власні розширення протоколу Gnutella, щоб додати нові функції або підвищити швидкість роботи своїх застосунків.

P2P-додатки можуть працювати не лише у P2P-мережах, а й також у мережі Інтернет. Наприклад, клієнт ініціює запит (відкриття листа). Сервер обробляє запит і надсилає відповідь (наприклад, повідомлення). Дані можуть передаватися в обидва боки, але часто у більшій кількості від сервера до клієнта.

Наприклад, поштовий клієнт (Outlook або Thunderbird) звертається до сервера (наприклад, Gmail), щоб отримати список нових листів. Сервер обробляє запит і відправляє відповідь.

Крім самих даних, такі протоколи можуть включати:

- аутентифікацію користувача;
- ідентифікацію файлів;
- формати запитів/відповідей (SMTP, POP3, IMAP).

Модель P2P забезпечує гнучку, децентралізовану взаємодію пристроїв без потреби у сервері, що робить її популярною для обміну файлами, медіа, криптовалют та інших цифрових ресурсів. У той час як клієнт-серверна модель забезпечує централізовану, контрольовану інфраструктуру, P2P підходить для мереж, де важлива автономність та розподіленість ресурсів.

9.3. Протоколи HTTP і HTTPS

HTTP (Hypertext Transfer Protocol) є одним з найважливіших і найпоширеніших протоколів прикладного рівня в моделі TCP/IP, який призначений для передачі гіпертекстових документів, зокрема, HTML-сторінок, і є основою функціонування всесвітньої павутини.

Спочатку протокол HTTP розроблявся лише для запитів і передачі HTML-документів між веббраузером і вебсервером, однак його гнучкість дозволила перетворити його на універсальний засіб обміну даними в розподілених інформаційних системах.

Суть роботи протоколу HTTP полягає в моделі "запит–відповідь", коли клієнт (веббраузер) надсилає запит, а сервер повертає відповідь із вмістом або повідомленням про помилку (рис. 9.6).

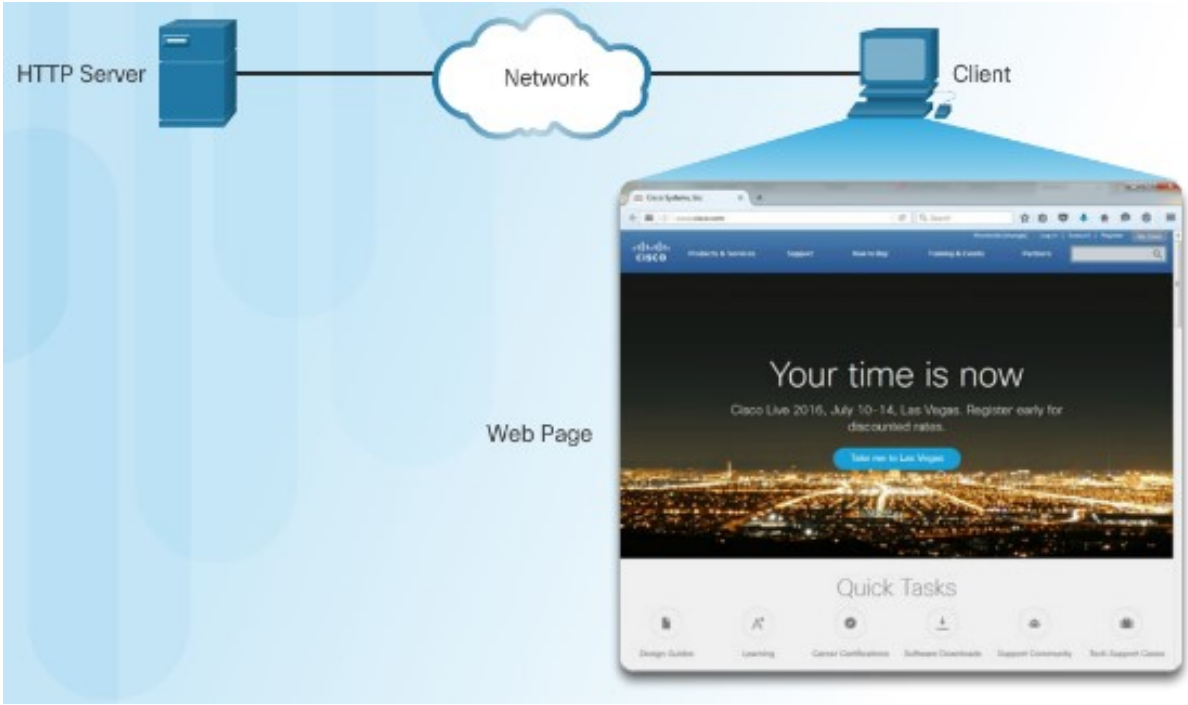


Рис. 9.6. Приклад відображення HTML-документу у браузері як вебсторінки

Основні типи HTTP-запитів включають GET, POST і PUT. Коли браузер надсилає запит GET, він зазвичай вимагає HTML-сторінку, зображення або інші ресурси з вебсайту.

Якщо ресурс доступний, сервер відповідає повідомленням "HTTP/1.1 200 OK" і надсилає потрібний файл, наприклад, HTML-код сторінки (рис. 9.7).

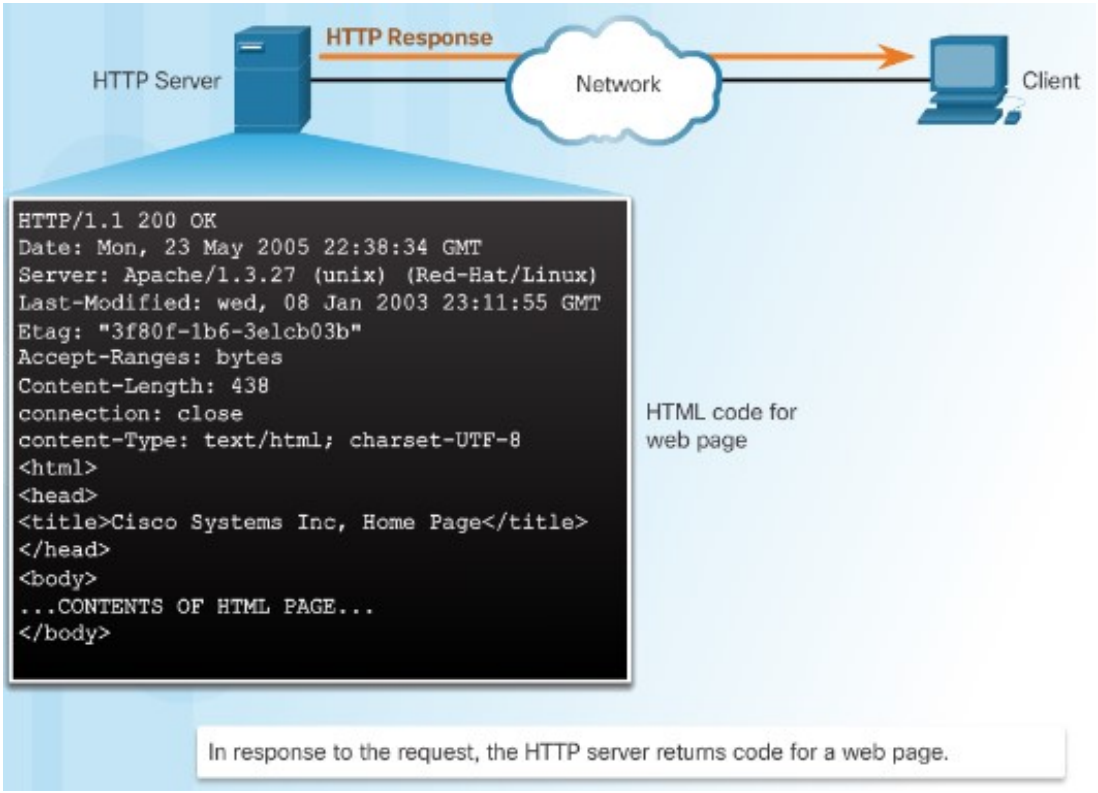


Рис. 9.7. Приклад відповіді від HTTP-сервера

HTTP-коди відповідей сервера класифікуються за п’ятьма основними групами залежно від типу відповіді.

Кожен код складається з трьох цифр: перша цифра вказує на клас відповіді, а дві інші – на конкретний статус.

Розглянемо класифікацію HTTP-коди відповідей сервера.

1xx – Інформаційні відповіді.

Ці повідомлення означають, що запит прийнятий і обробка триває.

100 (Continue) – клієнт може продовжувати надсилати запит.

101 (Switching Protocols) – сервер прийняв запит на зміну протоколу.

102 (Processing) – сервер обробляє запит, але відповідь ще не готова.

2xx – Відповіді «успіху».

Повідомляють, що запит клієнта було успішно оброблено.

200 (OK) – стандартна відповідь при успішному запиті.

201 (Created) – запит завершено, і ресурс створено (наприклад, після PUT).

202 (Accepted) – запит прийнято для обробки, але ще не завершено.

204 (No Content) – відповідь не містить тіла (наприклад, після успішного видалення ресурсу).

3xx – Перенаправлення.

Потрібні додаткові дії з боку клієнта для завершення запиту.

301 (Moved Permanently) – ресурс переміщено на іншу адресу назавжди.

302 (Found) – тимчасове перенаправлення на інший ресурс.

304 (Not Modified) – запитуваний ресурс не змінювався, можна використати кеш.

4xx – Помилки клієнта.

Клієнт надіслав некоректний запит або не має прав доступу.

400 (Bad Request) – неправильний синтаксис запиту.

401 (Unauthorized) – потрібна автентифікація.

403 (Forbidden) – доступ заборонено, навіть якщо автентифікація успішна.

404 (Not Found) – ресурс не знайдено на сервері.

405 (Method Not Allowed) – метод не дозволено для запитуваного ресурсу.

5xx – Помилки сервера.

Запит клієнта правильний, але сервер не зміг його обробити.

500 (Internal Server Error) – загальна помилка сервера.

501 (Not Implemented)– сервер не підтримує потрібний метод.

502 (Bad Gateway) – сервер отримав помилкову відповідь від іншого сервера.

503 (Service Unavailable) – сервер тимчасово недоступний (перевантаження або техобслуговування).

504 (Gateway Timeout) – сервер не дочекався відповіді від іншого сервера.

Ці коди є невід'ємною частиною протоколу HTTP і дозволяють клієнтам (наприклад, браузерам) правильно інтерпретувати результат запиту та вживати відповідних дій.

На рис. 9.8 наведено приклад *запиту GET* на відкриття вебсторінки.

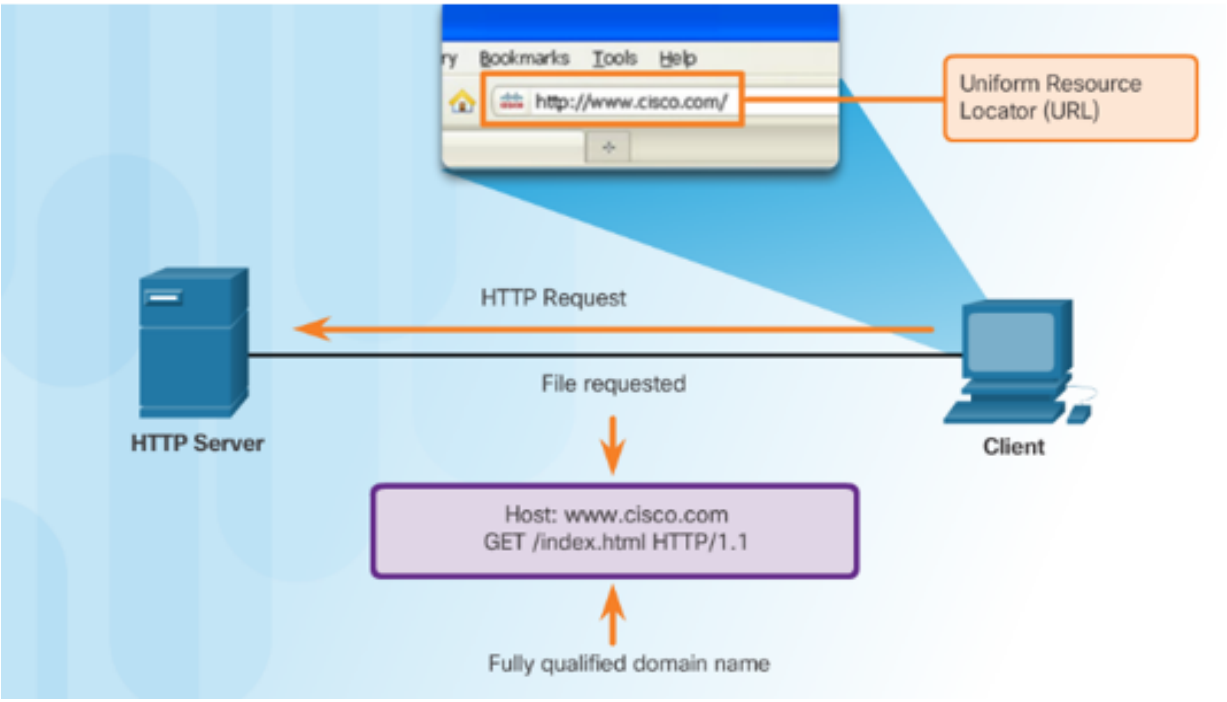


Рис. 9.8. Приклад надсилання запиту від клієнта на сервер на відкриття вебсторінки

Якщо ж ресурс відсутній або переміщено, сервер може відповісти, наприклад, повідомленням "The location of the requested file has changed".

Запити POST використовуються для передачі на сервер даних, введених користувачем у форми, наприклад, при реєстрації або оформленні замовлення. У цьому випадку дані передаються як частина тіла запиту.

Запит PUT призначений для завантаження файлів чи іншого контенту на сервер, наприклад, при надсиланні зображень або документів на вебсайт.

Попри свою універсальність, HTTP не є безпечним: усі дані передаються у відкритому тексті, що створює ризики перехоплення й зчитування інформації. Щоб вирішити цю проблему, використовується HTTPS – захищена версія HTTP, яка забезпечує шифрування і аутентифікацію за допомогою SSL/TLS. У взаємодії через HTTPS структура запиту така ж, як у HTTP, однак передача даних відбувається у зашифрованому вигляді, що захищає дані користувача від стороннього втручання. Це особливо важливо під час обміну конфіденційною інформацією, такою як паролі, дані платіжних карток тощо.

З точки зору моделі TCP/IP, процес передачі HTTP-запиту клієнтом до вебсервера проходить декілька етапів.

На рівні застосувань створюється HTTP-запит (наприклад, GET). Ці дані можуть бути додатково стиснуті або зашифровані. Потім вони передаються на транспортний рівень, де розбиваються на сегменти з додаванням заголовків із номерами портів – наприклад, порт 80 для HTTP чи 443 для HTTPS. Транспортний рівень забезпечує надійність доставки даних і контроль їх порядку.

Далі дані потрапляють на мережевий рівень, де кожному сегменту призначається IP-адреса відправника й одержувача. Це дозволяє маршрутизаторам визначати шлях до кінцевого вузла. Після цього на рівні мережевого доступу (або канальному рівні в моделі OSI) формуються кадри, які містять фізичні адреси відправника та наступного вузла на шляху до адресата.

Потім кадри кодуються у послідовності бітів і передаються як електричні сигнали або оптичні імпульси по фізичному середовищу, наприклад, по Ethernet-кабелю чи через оптичне волокно.

Якщо сервер і клієнт розміщені в різних мережах, дані проходять через кілька маршрутизаторів, кожен з яких розпаковує кадр, аналізує IP-заголовок, визначає найкращий маршрут і пересилає пакет далі, загортаючи його в новий кадр. Цей процес триває, доки пакет не досягне вузла призначення. На кінцевому пристрої відбувається деінкапсуляція – послідовне зняття заголовків протоколів кожного рівня, поки дані не потраплять назад на рівень застосувань, де знову перетворюються у зрозумілий вигляд (наприклад, у вебсторінку). Коли пристрій отримує дані, транспортний рівень за допомогою номера порту визначає, якому саме застосуванню вони призначені. На одному комп’ютері може працювати одразу декілька мережевих застосувань: веббраузер, поштовий клієнт, месенджер, служба резервного копіювання тощо.

Наприклад, користувач читає вебсторінку, тоді як у фоновому режимі його поштова програма отримує листи, а месенджер приймає повідомлення – усі ці процеси є окремими, але взаємодіють через спільну мережу завдяки використанню портів і стеку протоколів TCP/IP.

9.4. Протоколи SMTP, POP і IMAP

Однією з важливих послуг, які надають Інтернет-провайдери, є розміщення серверів електронної пошти. Саме завдяки цій послузі користувачі можуть обмінюватися повідомленнями через мережу,

причому швидко, зручно і з будь-якого пристрою, підключеного до інтернету. Електронна пошта суттєво трансформувала способи комунікації, замінивши традиційні засоби зв'язку. Однак для повноцінної роботи електронної пошти на комп'ютері чи смартфоні необхідно використовувати спеціальні застосунки та служби, які забезпечують відправку, зберігання та доставку повідомлень.

Електронна пошта – це комплекс програмних рішень і протоколів, що дають змогу доставляти повідомлення від одного користувача до іншого, зберігати їх у базах даних на поштових серверах, а також здійснювати доступ до них за потреби.

Зазвичай Інтернет-провайдери встановлюють власні сервери електронної пошти, на яких розміщені облікові записи численних клієнтів. Повідомлення зберігаються на цих серверах до моменту завантаження користувачем або знищення.

Коли користувач відправляє листа, його поштовий клієнт (програма або застосунок) звертається до поштового сервера, вказаного в налаштуваннях. Сервер перевіряє, чи знаходиться домен одержувача серед локально відомих. Якщо такого домену немає, сервер виконує DNS-запит, щоб отримати IP-адресу сервера, що обслуговує пошту цього домену. Потім лист пересилається за цією адресою. Клієнти електронної пошти не спілкуються безпосередньо один з одним – для пересилання завжди використовується посередництво поштових серверів. Це справедливо навіть у випадку, коли обидва користувачі мають облікові записи в одному домені.

Обмін електронною поштою базується на використанні трьох основних протоколів прикладного рівня: SMTP, POP і IMAP.

SMTP (Simple Mail Transfer Protocol) застосовується для надсилання повідомлень: як від поштового клієнта до сервера, так і між серверами. Коли клієнт створює листа і натискає «Надіслати», використовується протокол SMTP для передавання цього повідомлення серверу. Повідомлення формується у форматі, який містить заголовок із зазначенням адреси відправника та одержувача, а також тіло листа з текстом і, можливо, вкладеннями.

SMTP використовує порт 25, до якого підключається клієнтський застосунок для передачі листа. Сервер може або помістити повідомлення у чергу облікового запису користувача, або переслати

далі іншому серверу. Якщо цільовий сервер недоступний (наприклад, через перевантаження або технічні роботи), SMTP надає можливість тимчасового зберігання листа з подальшими спробами повторної доставки. У разі якщо декілька спроб завершуються невдачею, лист повертається відправнику з відповідним повідомленням про помилку доставки. Отримання електронної пошти клієнтськими застосунками здійснюється за допомогою одного з двох протоколів – POP або IMAP.

POP (Post Office Protocol), зазвичай версії POP3, дозволяє клієнтові завантажити листи із сервера на локальний пристрій. Після завантаження листи часто автоматично видаляються із сервера. Це означає, що копії листів зберігаються тільки на одному пристрої. POP працює на порту TCP 110. Після встановлення з'єднання клієнт і сервер обмінюються командами, необхідними для автентифікації, отримання листів та завершення сеансу.

З одного боку, такий підхід зменшує навантаження на поштові сервери провайдера, оскільки не вимагає тривалого зберігання даних. Це зручно для провайдерів, які не хочуть утримувати великі обсяги інформації. З іншого боку, POP не підходить для підприємств, які прагнуть до централізованого зберігання і резервного копіювання листів, оскільки кожен лист зникає з сервера одразу після завантаження.

IMAP (Internet Message Access Protocol) пропонує іншу модель роботи. Його головна відмінність полягає в тому, що при роботі з листами користувач оперує не оригіналами, а копіями. Повідомлення залишаються збереженими на сервері, а клієнт отримує лише їхні копії. Завдяки цьому можна працювати з поштою з кількох пристроїв: наприклад, перевірити листи з офісного комп'ютера, відповісти зі смартфона, а згодом пересортувати їх на домашньому ноутбуці – і вся інформація залишиться синхронізованою.

IMAP дозволяє створювати структуру папок на сервері, організовуючи листування у зручну ієрархію. Зміни, внесені на одному пристрої, автоматично застосовуються на інших. Наприклад, якщо користувач видалив лист на телефоні, він зникне і на комп'ютері. Така модель ідеально підходить для малого і середнього бізнесу: пошта зберігається централізовано, захищена резервним копіюванням і доступна з будь-якої точки світу. Проте використання IMAP вимагає значно більших ресурсів на стороні провайдера. Сервер має забезпечити

тривале зберігання великого обсягу листів, підтримку складної структури поштових скриньок і механізмів синхронізації. Якщо клієнти вимагають ще й регулярного створення резервних копій, вартість обслуговування таких послуг для провайдера істотно зростає. Тому для великого потоку абонентів провайдери частіше обирають POP як менш затратний варіант.

9.5. Служба доменних імен

У комп'ютерних мережах кожен пристрій має унікальну числову IPадресу, яка дозволяє йому надсилати й отримувати дані. Наприклад, сервер може мати адресу 198.133.219.25.

Проте запам'ятати такі числа звичайним користувачам доволі складно. Саме тому були введені доменні імена – прості й зрозумілі назви, які значно легше сприймаються людиною. Наприклад, замість того, щоб запам'ятовувати вказану IP-адресу, користувач вводить знайоме ім'я вебресурсу *www.cisco.com*.

Ці доменні імена пов'язані з відповідними IP-адресами за допомогою спеціальної служби – системи доменних імен, або DNS (Domain Name System). Завдяки цій службі, коли користувач вводить адресу сайту в браузері, DNS автоматично знаходить і повертає відповідну IP-адресу. Таким чином, у разі зміни IP-адреси сайту, користувачі цього навіть не помітять – достатньо лише оновити запис DNS, не змінюючи ім'я сайту.

Спочатку, коли мережі були маленькими, такі відповідності можна було задавати вручну. Однак із розширенням мережі Інтернет та кількості пристроїв така практика стала незручною. Зараз це завдання виконує розподілена система серверів DNS, яка автоматично зіставляє доменні імена з відповідними IPадресами.

Кожен DNS-сервер зберігає базу даних із записами, які використовуються для перетворення імен. Протокол DNS визначає формат повідомлень, які використовуються для обміну інформацією між клієнтами та серверами. Ці повідомлення містять запити, відповіді, повідомлення про помилки, а також записи ресурсів.

Для забезпечення роботи DNS-серверів часто використовується програмне забезпечення BIND (Berkeley Internet Name Domain), яке розробили студенти Каліфорнійського університету в Берклі ще на

початку 1980-х років. Формат DNS-повідомлень, реалізований у BIND, став стандартом в Інтернеті.

У базах даних DNS зберігаються різноманітні записи, які забезпечують зв'язок між доменними іменами та IP-адресами.

Основні типи таких записів:

A – відображає доменне ім'я на IPадресу вузла (наприклад, серверу чи маршрутизатору);

NS – вказує на авторитетний сервер імен, який обслуговує конкретну зону;

CNAME – канонічне ім'я, яке дозволяє мати кілька доменних імен для одного пристрою або сервісу (це запис у DNS, який вказує, що певне доменне ім'я є псевдонімом (alias) іншого, основного (канонічного) імені, наприклад, www.example.com може бути псевдонімом для server1.example.com, де останнє є канонічним іменем);

MX – запис поштового обміну, що визначає сервери для обробки електронної пошти домену.

Коли пристрій у мережі потребує перетворення доменного імені в IP-адресу, клієнт DNS надсилає запит до призначеного серверу. Якщо запитане ім'я присутнє в локальній базі записів цього серверу, адреса повертається негайно. Якщо ж ні – сервер переадресовує запит до інших DNS-серверів, поки не знайде відповідний запис. Після отримання IP-адреси вона зберігається в кешпам'яті серверу на певний час – це дозволяє повторно використовувати знайдену адресу без потреби знову звертатися до мережі. Такий підхід значно зменшує навантаження на інфраструктуру DNS та знижує час відповіді на запити.

У системах на базі ОС Windows також реалізовано локальне кешування DNS.

Служба «Клієнт DNS» зберігає нещодавно отримані відповіді у пам'яті, а командою *ipconfig /displaydns* можна переглянути їх список.

DNS побудована як ієрархічна система, що нагадує перевернуте дерево: вгорі знаходиться кореневий сервер, а нижче – сервери доменів верхнього, другого і наступних рівнів. Наприклад, для доменного імені mail.cisco.com послідовність запиту виглядатиме так: спочатку запит йде до кореневого серверу, який знає, де шукати домен .com. Сервер .com перенаправить запит до DNS-серверу, відповідального за домен cisco.com, який, своєю чергою, містить запис MX для mail.cisco.com.

Кожен DNS-сервер відповідає лише за свою частину ієрархії, яку називають «зоною». У межах цієї зони сервер зберігає всі необхідні записи і є довіреним (авторитетним) для цієї інформації. Наприклад, DNS-сервер у домені *cisco.netacad.net* не відповідальний за ім'я *mail.cisco.com*, бо це інший домен і, відповідно, інша зона.

DNS є децентралізованим, масштабованим сервісом. Він дозволяє ефективно керувати адресацією в глобальній мережі (рис. 9.9).

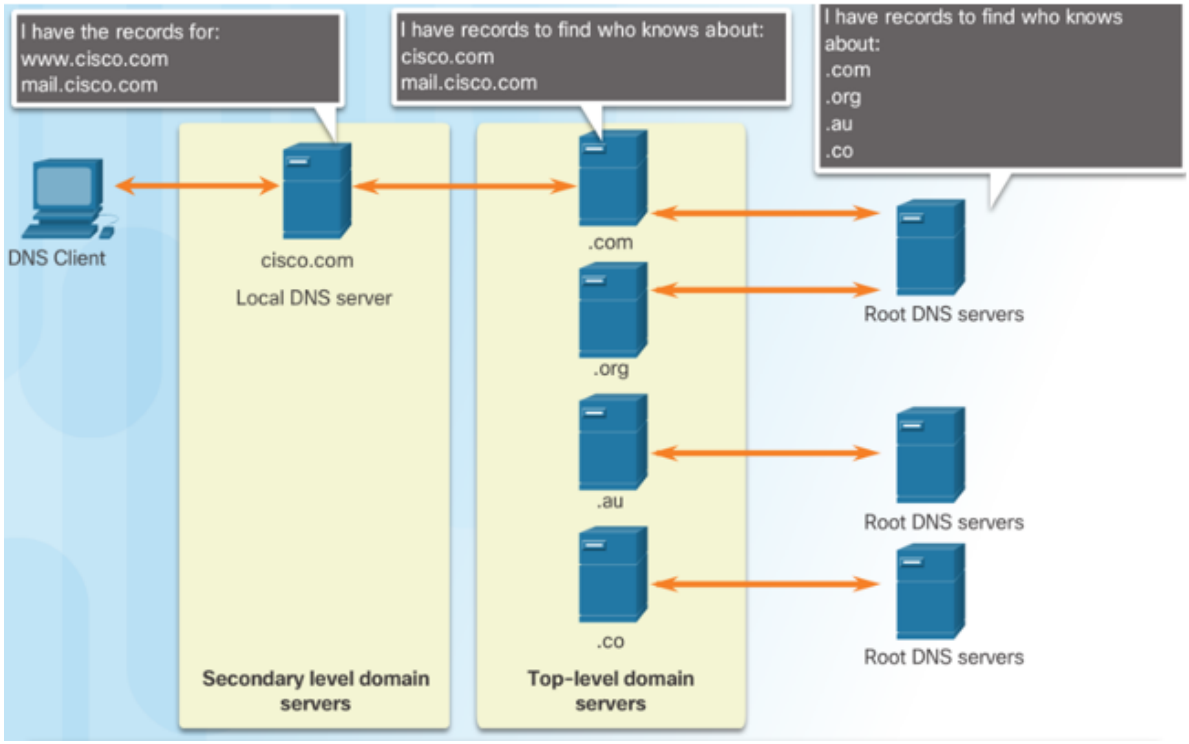


Рис. 9.9. Ієрархічна система DNS

Доменні імена формуються з урахуванням ієрархії. Наприклад:

.com – комерційні організації;

.org – некомерційні організації;

.jp – Японія;

.au – Австралія;

.co – Колумбія.

Під доменами верхнього рівня можуть існувати домени другого і третього рівнів. Повне доменне ім'я складається з усіх цих рівнів, від найнижчого до кореневого. Наприклад, у записі www.example.com домен example є другим рівнем, а www – третім.

На відміну від традиційних клієнтських застосунків, таких як веббраузер або поштовий клієнт, DNS-клієнт – це фоновий сервіс, що працює постійно і обслуговує усі програми, яким потрібно перетворити ім’я в IP-адресу.

Коли ми налаштовуємо мережеве підключення, зазвичай ми вказуємо одну або кілька адрес DNS-серверів – це можуть бути сервери провайдера або публічні DNS-сервіси (наприклад, 8.8.8.8 від Google).

У ОС Windows доступна утиліта *nslookup*, яка дозволяє вручну перевірити правильність роботи DNS, з’ясувати, яка IP-адреса відповідає певному домену, або визначити, які записи містить певний сервер імен. Це корисний інструмент для діагностики мережевих проблем, зокрема – з іменуванням.

9.6. Протокол динамічної конфігурації вузла

У комп’ютерних мережах одним з найзручніших способів надання пристроям IP-адрес та інших мережевих параметрів є використання служби динамічного конфігурування хостів, або DHCP (Dynamic Host Configuration Protocol). Завдяки цій службі кожен пристрій, що підключається до мережі, автоматично отримує унікальну IP-адресу, маску підмережі, адресу шлюзу за замовчуванням, DNS-серверів та інші важливі параметри. Усе це дозволяє значно спростити процес конфігурації мережі, адже не потрібно кожного разу вручну вносити дані на кожному пристрої.

Процес, коли IP-адреси присвоюються автоматично, називається *динамічною адресацією*. Альтернативним варіантом є *статичне призначення адрес*, при якому адміністратор особисто вводить усі необхідні параметри в налаштуваннях кожного вузла.

У невеликих мережах це ще можливо, але у більших, де кількість пристроїв постійно змінюється, таке рішення стає непридатним.

DHCP, у свою чергу, дозволяє автоматично обслуговувати пристрої, що регулярно підключаються до мережі – як нові комп’ютери, так і мобільні телефони, планшети, ноутбуки (рис. 9.10).

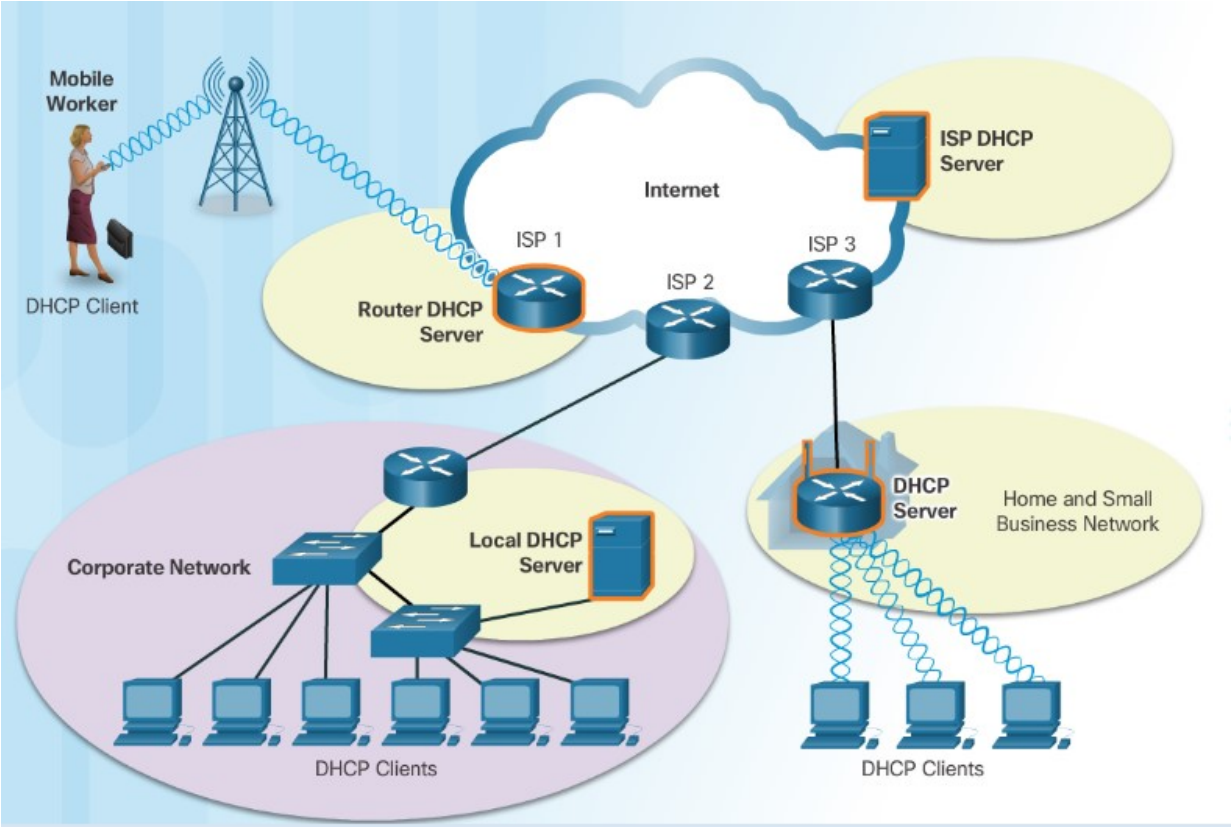


Рис. 9.10. Обслуговувати пристроїв DHCP

Коли пристрій з підтримкою DHCP приєднується до мережі – наприклад, до офісної Wi-Fi або домашньої локальної мережі – він автоматично надсилає широкомовний запит DHCPDISCOVER, запитуючи доступну адресу. Сервер DHCP, який отримує цей запит, формує у відповідь повідомлення DHCPOFFER, в якому пропонує певну IP-адресу з наявного діапазону, що називається пулом адрес. Крім самої IP-адреси, у повідомленні пропонуються також інші параметри: маска підмережі, шлюз за замовчуванням, адреси DNS-серверів та тривалість оренди адреси.

Якщо в мережі присутні кілька DHCP-серверів, клієнт може отримати кілька таких пропозицій. У відповідь він вибирає одну з них і надсилає DHCPREQUEST – запит, в якому повідомляє серверу, яку саме адресу він хоче прийняти. Сервер, що зробив цю пропозицію, підтверджує її повідомленням DHCPACK, після чого клієнт починає використовувати призначену адресу. У випадку, якщо адреса вже зайнята або термін дії пропозиції закінчився, сервер може відповісти повідомленням DHCPNAK (негативне підтвердження), і весь процес почнеться знову (рис. 9.11).

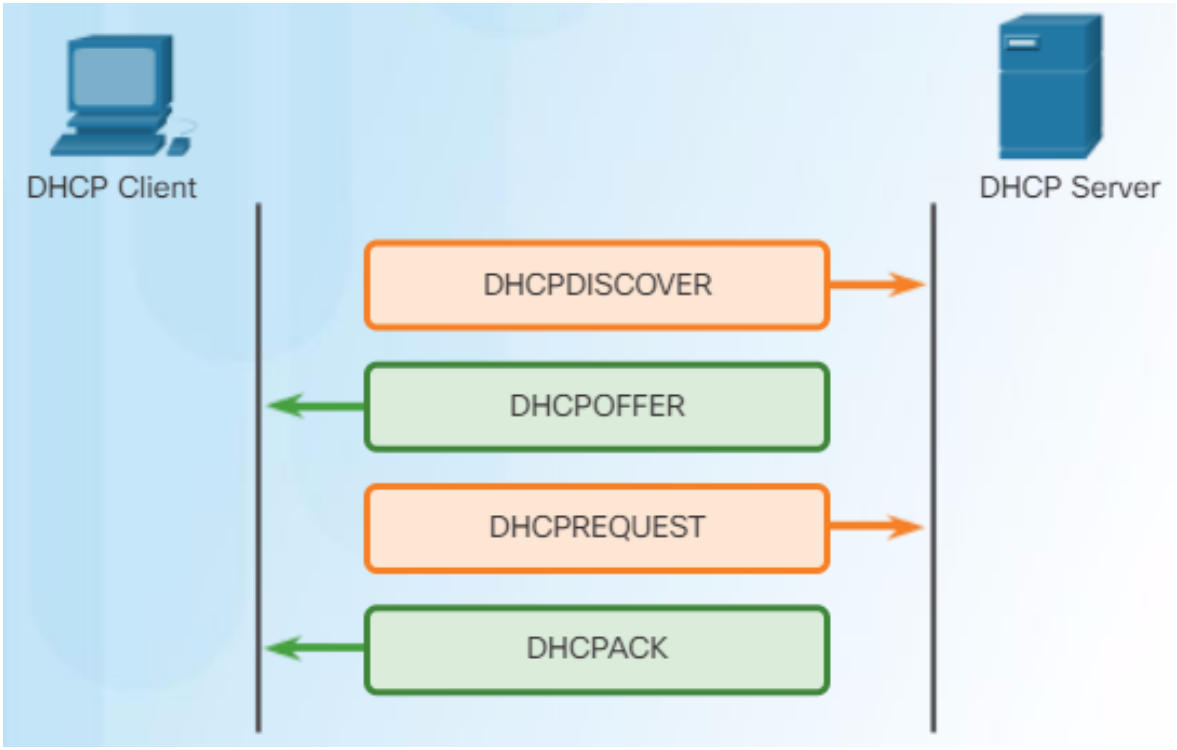


Рис. 9.11. Взаємодія клієнта і сервера DHCP

Надана адреса не закріплюється за пристроєм назавжди – вона видається на певний період, після якого потребує продовження.

Якщо клієнт залишається в мережі, він повинен вчасно надіслати новий запит DHCPREQUEST для оновлення оренди.

Якщо пристрій відключається від мережі або вимикається, його адреса повертається в пул і знову стає доступною для інших користувачів. Це дуже зручно в умовах, коли пристрої постійно підключаються та відключаються, наприклад, у бібліотеках, кав'ярнях, аеропортах – всюди, де є вільний безпровідний доступ до мережі.

Оскільки DHCP-клієнт є вбудованою функцією більшості сучасних операційних систем, процес конфігурації є повністю прозорим для користувача. Наприклад, коли користувач приходить у кафе з ноутбуком і підключається до Wi-Fi, DHCP-сервер автоматично видає йому всі необхідні параметри – і вже за кілька секунд пристрій має доступ до Інтернету.

Слід зауважити, що DHCP не лише спрощує підключення, але й запобігає конфліктам IP-адрес у мережі. Кожному пристрою призначається унікальна IP-адреса, і система контролює, щоб одна адреса не була видана одночасно двом різним клієнтам. Це гарантує

стабільність роботи мережі. Крім того, адміністратори можуть змінювати налаштування IP-адрес на сервері DHCP, і ці зміни автоматично застосовуватимуться до клієнтів – без необхідності змінювати конфігурацію вручну на кожному пристрої.

Більшість інтернет-провайдерів також використовують протокол DHCP для автоматичної видачі адрес своїм клієнтам. Таким чином, при підключенні домашнього роутера до мережі провайдера, він, як DHCP-клієнт, отримує зовнішню IP-адресу і починає працювати в мережі Інтернет. У внутрішній домашній мережі той самий роутер виконує вже функцію DHCP-сервера, призначаючи адреси смартфонам, комп'ютерам, телевізорам і іншим пристроям.

Для діагностики і перевірки роботи служби DHCP користувачі можуть використовувати спеціальні утиліти.

В ОС Windows є команда *ipconfig /all*, яка показує, чи отримано IP-адресу через DHCP, а також усі параметри оренди.

Команда *ipconfig /release* дозволяє звільнити IP-адресу, а *ipconfig /renew* – запитати нову. Це корисно при усуненні проблем із мережею або під час переходу між різними точками доступу.

Завдяки своїй гнучкості, масштабованості та простоті в обслуговуванні, протокол DHCP є однією з ключових технологій сучасних комп'ютерних мереж, що забезпечує ефективну і безперебійну роботу для користувачів будь-якого рівня.

9.7. Протокол передачі файлів

Протокол передачі файлів FTP (File Transfer Protocol) є одним із найпоширеніших протоколів прикладного рівня, який активно використовується для обміну файлами між комп'ютерами в мережі. Основна мета цього протоколу полягає в тому, щоб забезпечити надійний і гнучкий спосіб передачі файлів між клієнтським пристроєм і сервером, що виконує функцію зберігання або розповсюдження даних.

Для роботи з протоколом FTP потрібен спеціальний клієнт – програма, яка запускається на комп'ютері користувача та забезпечує інтерфейс для підключення до FTP-сервера. За допомогою цієї програми користувач може переглядати файли на сервері, завантажувати їх собі або, навпаки, передавати на сервер.

Наприклад, веброзробник може використовувати FTP-клієнт для завантаження файлів свого сайту на хостинговий сервер. Існує багато різних FTP-клієнтів, серед яких популярні програми FileZilla, WinSCP або вбудовані засоби командного рядка операційних систем. Для ефективного функціонування FTP використовує два окремі мережеві з'єднання. Перше з них призначене для обміну командами та відповідями між клієнтом і сервером. Це управлінське з'єднання, через яке клієнт надсилає команди, такі як "увійти", "переглянути список файлів", "видалити файл", "почати передачу" тощо. У відповідь сервер надсилає повідомлення про успішність або помилки виконання запитів.

Друге з'єднання створюється для фактичної передачі файлів. Саме через це з'єднання надсилаються дані – або від сервера до клієнта, або навпаки. Наприклад, коли користувач завантажує фотографії з FTP-сервера на свій комп'ютер, друге з'єднання відповідає за передачу цих зображень. Особливістю цього процесу є те, що передача даних виконується окремо від командного каналу, що дозволяє підтримувати паралельну роботу: клієнт може надалі надсилати нові команди, поки дані передаються. Цікаво, що ці два з'єднання – управлінське та дане – мають різний життєвий цикл.

Управлінське з'єднання зазвичай зберігається протягом усього сеансу FTP, поки триває робота клієнта з сервером. Натомість з'єднання для передачі даних створюється щоразу заново при кожній операції передачі – наприклад, при завантаженні одного або кількох файлів.

FTP підтримує як завантаження файлів із сервера (цей процес часто називають "завантаженням" або "download"), так і відправлення файлів на сервер ("вивантаження", або "upload"). Наприклад, студент може завантажити з університетського FTP-сервера навчальні матеріали або, навпаки, передати туди власні проєкти. У великих організаціях FTP часто використовується для резервного копіювання даних, коли файли зберігаються на віддаленому FTP-сервері.

Важливо зазначити, що класичний FTP не забезпечує шифрування даних, тож при передаванні конфіденційної інформації краще використовувати захищені версії протоколу – такі як FTPS (FTP over SSL/TLS) або SFTP (Secure File Transfer Protocol, який працює через SSH). Такі рішення дозволяють не лише обмінюватися файлами, а й захищати передані дані від перехоплення та несанкціонованого доступу.

9.8. Протокол обміну блоками серверних повідомлень

Протокол обміну блоками серверних повідомлень, відомий як SMB (Server Message Block), є одним із ключових протоколів прикладного рівня, який дозволяє організувати спільне використання ресурсів у мережі. Його створено ще наприкінці 1980-х років компанією IBM, і з того часу він активно розвивався та інтегрувався в сучасні мережеві технології. Основною функцією SMB є забезпечення обміну файлами, папками, принтерами та іншими пристроями між клієнтськими пристроями та серверами в мережі (рис. 9.12).

Протокол SMB працює за принципом «запит – відповідь». Це означає, що клієнт надсилає певний запит до сервера (наприклад, запит на відкриття або копіювання файлу), а сервер у відповідь надсилає результат обробки цього запиту. Такий підхід дозволяє організувати чітке та контрольоване спілкування між пристроями в локальній мережі, що особливо важливо у великих корпоративних середовищах.

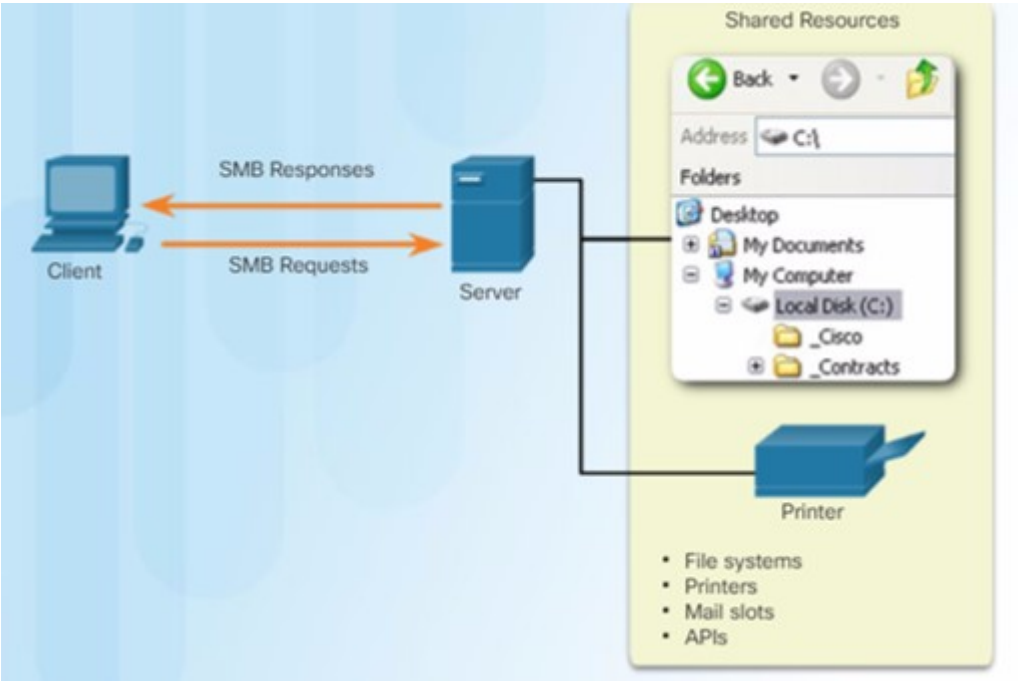


Рис. 9.12. Взаємодія з сервером за допомогою протоколу SMB

Структура повідомлень у протоколі SMB суворо регламентована. Кожне повідомлення має стандартний формат, який включає фіксовану довжину заголовка – це частина, де міститься службова інформація про запит, тип операції, ідентифікатори сесії тощо. Після заголовка йдуть

змінні параметри та дані, що передаються – наприклад, вміст файлу або ім’я каталогу.

SMB надає багато функцій, які виходять за межі простої передачі файлів. Наприклад, протокол підтримує аутентифікацію користувачів під час встановлення з’єднання – користувач повинен ввести свої облікові дані, щоб отримати доступ до спільного ресурсу. Це дає змогу контролювати, хто і до яких файлів або пристроїв має доступ. Також можливе завершення сесії, коли користувач більше не потребує підключення, і це дозволяє серверу ефективніше використовувати ресурси. Окрім файлів, SMB дозволяє керувати мережею принтерів. Наприклад, кілька користувачів можуть підключатися до одного мережевого принтера через SMB, надсилати документи на друк і бачити чергу завдань. Цей механізм значно полегшує адміністрування офісного обладнання в компаніях, де принтери не прив’язані до конкретного комп’ютера. Іншою важливою функцією SMB є можливість передавання повідомлень між пристроями. Наприклад, додаток на одному комп’ютері може надіслати службове повідомлення іншому пристрою в мережі – це використовується для адміністративного інформування або службових повідомлень у локальній мережі.

Своє особливе поширення SMB отримав завдяки інтеграції в операційні системи Microsoft. Починаючи з Windows 2000, протокол SMB тісно пов’язаний зі стеком TCP/IP і використовує систему доменних імен DNS для виявлення ресурсів. Це означає, що сучасні комп’ютери з Windows можуть знаходити один одного по імені (наприклад, server.local) замість використання IP-адрес, що значно зручніше для користувачів.

У попередніх версіях Windows (до 2000 року) протокол SMB покладався на альтернативні способи іменування, які не були сумісні з TCP/IP або DNS. Це створювало труднощі в сумісності з іншими платформами. Проте з розвитком технологій, Microsoft зробила SMB відкритішим та універсальнішим. На відміну від FTP, який зазвичай встановлює короткотривалі з’єднання лише на час передачі файлів, SMB підтримує постійне з’єднання з сервером. Це означає, що після одного підключення користувач може працювати з файлами так, ніби вони зберігаються локально на його комп’ютері. Наприклад, відкритий

у Word документ, що розміщується на сервері, можна редагувати, зберігати і друкувати, не копіюючи його на локальний диск.

SMB не обмежується лише системами Windows. Операційні системи сімейства UNIX та Linux також можуть взаємодіяти з ресурсами Windows за допомогою спеціального програмного забезпечення під назвою *Samba* – цей пакет дозволяє Linux-системам як виступати в ролі SMB-клієнтів, так і самостійно роздавати ресурси в мережі. Наприклад, на сервері з Ubuntu можна налаштувати спільний доступ до папки, до якої зможуть підключатися користувачі з Windows.

Також комп'ютери Apple підтримують роботу з протоколом SMB. У macOS реалізовано можливість підключення до спільних мережевих папок, що знаходяться на Windows-серверах або на NAS-пристроях, які підтримують SMB. Таким чином, протокол SMB став справжнім стандартом для обміну ресурсами у змішаних мережах з різними операційними системами. Отже, SMB – це універсальний і багатофункціональний протокол, що дозволяє ефективно організувати обмін файлами, використання принтерів і спільну роботу в мережах. Його підтримка на різних платформах і гнучкість у налаштуваннях роблять його незамінним у сучасному мережевому середовищі.

Контрольні запитання та задачі до теми 9

Перелік теоретичних запитань

1. Які основні функції виконує прикладний рівень моделі TCP/IP?
2. Чим прикладний рівень відрізняється від інших рівнів мережевої моделі?
3. Яка роль протоколу HTTP у взаємодії користувача з вебресурсами?
4. Чим відрізняється протокол HTTPS від HTTP, і які механізми безпеки він використовує?
5. Які функції виконує протокол SMTP у процесі передавання електронної пошти?
6. У чому полягає відмінність між POP3 та IMAP при отриманні електронної пошти?
7. Як працює служба доменних імен DNS і яке її значення у функціонуванні мережі Інтернет?
8. Яку роль виконує протокол DHCP у динамічному налаштуванні

мережевих пристроїв?

9. Як здійснюється передавання файлів за допомогою протоколу FTP? Які типи з'єднань використовуються?

10. У чому полягає специфіка роботи FTP-клієнта і FTP-сервера?

11. Що таке протокол SMB і як він реалізує спільний доступ до ресурсів у мережі?

12. Як організовано обмін повідомленнями у протоколі SMB? Які ресурси він дозволяє спільно використовувати?

13. У чому полягає різниця між клієнт-серверною та одноранговою моделлю мережі?

14. Які прикладні сервіси є типовими для однорангових мереж?

15. Які протоколи прикладного рівня використовуються для організації спільної роботи в мережі (приклади з реального життя)?

Перелік задач для самостійного опрацювання

Завдання 1.

На своєму ПК дослідити роботу DNS-запитів за допомогою утиліти *nslookup*. Відслідкувати і записати перетворення доменного імені на IP-адресу.

Завдання 2.

Створити просту мережу з двох комп'ютерів з використанням DHCP-сервера, з аналізом отриманих IP-адрес.

Завдання 3.

Налаштувати спільний доступ до папки у локальній мережі з використанням SMB (у Windows або через *Samba* у Linux). Перевірити доступ з іншого пристрою.

Перелік використаної та рекомендованої літератури

9.1. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

9.2. HTTP/HTTPS [Електронний ресурс]. – Режим доступу: https://developer.mozilla.org/en-US/docs/Web/HTTP

9.3. HTTP request methods [Електронний ресурс]. – Режим доступу: https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Methods

Тема 10. Засоби мережевої безпеки

У сучасному цифровому світі, де обмін інформацією в електронному вигляді став повсякденним явищем, питання захисту комп'ютерних мереж набуває особливої актуальності.

Розвиток інформаційних технологій супроводжується зростанням ризиків, пов'язаних із несанкціонованим доступом до даних, крадіжками конфіденційної інформації, кібератаками, шкідливим програмним забезпеченням та іншими загрозами. Будь-яка вразливість у мережевій інфраструктурі може призвести до серйозних наслідків – від втрати даних до порушення безперервності бізнес-процесів або функціонування цілих організацій.

Мережева безпека – це комплекс організаційних і технічних заходів, спрямованих на забезпечення конфіденційності, цілісності та доступності інформації, що передається мережевими каналами. Для досягнення цієї мети застосовуються спеціалізовані засоби, які дозволяють контролювати мережевий трафік, запобігати несанкціонованому доступу, виявляти та реагувати на спроби вторгнення, а також забезпечувати шифрування переданої інформації.

До основних засобів мережевої безпеки належать брандмауери, системи виявлення та запобігання вторгненням (IDS/IPS), антивірусне програмне забезпечення, засоби шифрування (VPN, TLS), засоби автентифікації та авторизації, а також системи управління доступом. Кожен із цих компонентів виконує специфічну функцію у багаторівневій системі захисту і вимагає правильного налаштування та постійного оновлення відповідно до нових загроз.

10.1. Категорії загроз мережевої безпеки

Мережева безпека є однією з найважливіших складових сучасної інформаційної інфраструктури, вона охоплює заходи, спрямовані на захист комп'ютерних мереж від несанкціонованого доступу, зловмисних атак, витоку або втрати даних.

Для забезпечення ефективного захисту необхідно враховувати специфіку середовища, доступні інструменти, типи загроз та особливості вимог до продуктивності й надійності мережі (табл. 10.1).

Таблиця 10.1. **Класифікація мережевих загроз**

Ознака	Тип загрози	Джерело	Ціль	Приклади
За способом реалізації	Шкідливе ПЗ (Malware)	Зовнішнє	Порушення цілісності, доступу	Emotet, LockBit, RedLine Stealer
	Фішинг	Зовнішнє	Викрадення облікових даних	E-mail з підробленим посиланням на Microsoft 365
	Атаки типу «відмова в обслуговуванні» (DoS/DDoS)	Зовнішнє	Паралізація ресурсів	Атака DDoS на сервіси Cloudflare, Amazon Web Services
	Атаки нульового дня	Зовнішнє	Використання невідомих уразливостей	CVE-2023-36884 (уразливість у Microsoft Office)
	SQL-ін’єкція, XSS	Зовнішнє	Отримання доступу до баз даних	Злом сайтів через SQL injection
	Внутрішні порушення (інсайдери)	Внутрішнє	Витік або знищення даних	Співробітник копіює базу клієнтів перед звільненням
За спрямованістю	На конфіденційність	Внутрішнє/ Зовнішнє	Викрадення особистих даних	Злом акаунтів користувачів Facebook або Google
	На цілісність	Внутрішнє/ Зовнішнє	Зміна або знищення даних	Шкідливе ПЗ, що модифікує бази даних
	На доступність	Зовнішнє	Порушення роботи систем	DDoS-атака на банківські сервіси України
За об’єктом впливу	Сервери	Зовнішнє	Виведення з ладу / злом	Злом сервера з відкритим SSH-доступом
	Робочі станції	Внутрішнє/ Зовнішнє	Встановлення троянців	Троян, що викрадає паролі з браузера
	Маршрутизатори / мережеве обладнання	Зовнішнє	Контроль трафіку	Зловмисник перепрошиває роутер для прослуховування
	IoT-пристрої	Зовнішнє	Побудова ботнет-мереж	Mirai Botnet атакує IP-камери та роутери

Забезпечення мережевої безпеки включає комплекс технологій, методів, програмного забезпечення та обладнання, покликаних захистити дані від втрати, перехоплення або модифікації.

До основних інструментів захисту належать міжмережеві екрани, антивірусне програмне забезпечення, системи виявлення і запобігання вторгненням (IDS/IPS), а також механізми шифрування. Крім того, важливо впроваджувати політики контролю доступу, навчати користувачів принципам «інформаційної гігієни» й здійснювати моніторинг активності в мережі в реальному часі.

Сучасні загрози безпеці найчастіше надходять із зовнішнього середовища, передусім через мережу Інтернет. Серед найпоширеніших видів шкідливого програмного забезпечення найбільш поширені віруси, черв'яки та троянські програми.

Вірус, як правило, проникає до системи разом із виконуваним файлом і активується під час запуску.

Черв'як же, на відміну від вірусу, може поширюватися самостійно через мережу без участі користувача, використовуючи відомі уразливості в системі. Наприклад, хробак може сканувати локальну мережу на наявність відкритих портів та автоматично копіювати себе на вразливі пристрої.

Троянські програми маскуються під легітимне програмне забезпечення, але насправді відкривають доступ до системи для зловмисника.

Окремим видом загроз є шпигунське та *рекламне ПЗ*, яке непомітно встановлюється на пристрій користувача і збирає конфіденційну інформацію, наприклад, історію перегляду або дані кредитних карток.

Зростаючу небезпеку становлять *атаки «нульового дня»*, які експлуатують уразливості в програмному забезпеченні ще до того, як про них дізнаються розробники. Внаслідок цього системи залишаються беззахисними впродовж критичного періоду часу.

Ще одним серйозним типом загроз є *хакерські атаки*, які можуть бути спрямовані на отримання контролю над пристроями, злам серверів або викрадення облікових даних користувачів. Прикладом є DoS-атака «відмова в обслуговуванні», що спричиняє перевантаження серверів або мережевих пристроїв, виводячи їх із ладу. До прикладу, хакери можуть

здійснити масове надсилання запитів на вебсервер, що призведе до його «зависання» і недоступності для реальних користувачів.

DoS (Denial of Service) або атака на відмову в обслуговуванні – це тип кібернападу, основною метою якого є виведення з ладу комп'ютерної системи, сервера або мережі, зробивши їх недоступними для користувачів. Це досягається шляхом надмірного навантаження цільового ресурсу такою кількістю запитів, яку він фізично або програмно не здатен обробити. У результаті користувачі не можуть отримати доступ до послуг, що надаються, наприклад, до вебсайту, електронної пошти чи онлайн-додатків.

DDoS-атака (Distributed Denial of Service) – це тип кібернападу, під час якого велика кількість скомпрометованих пристроїв, об'єднаних у ботмережу, одночасно надсилають запити до цільового сервера або мережевого ресурсу з метою перевантажити його та зробити недоступним для законних користувачів.

На відміну від звичайної DoS-атаки, DDoS має розподілений характер, що ускладнює виявлення та блокування шкідливого трафіку, адже він надходить із багатьох різних джерел (рис. 10.1).

Такий тип атаки часто використовується для виведення з ладу вебсайтів, онлайн-сервісів або інфраструктури компаній, що може призвести до серйозних фінансових та репутаційних збитків.

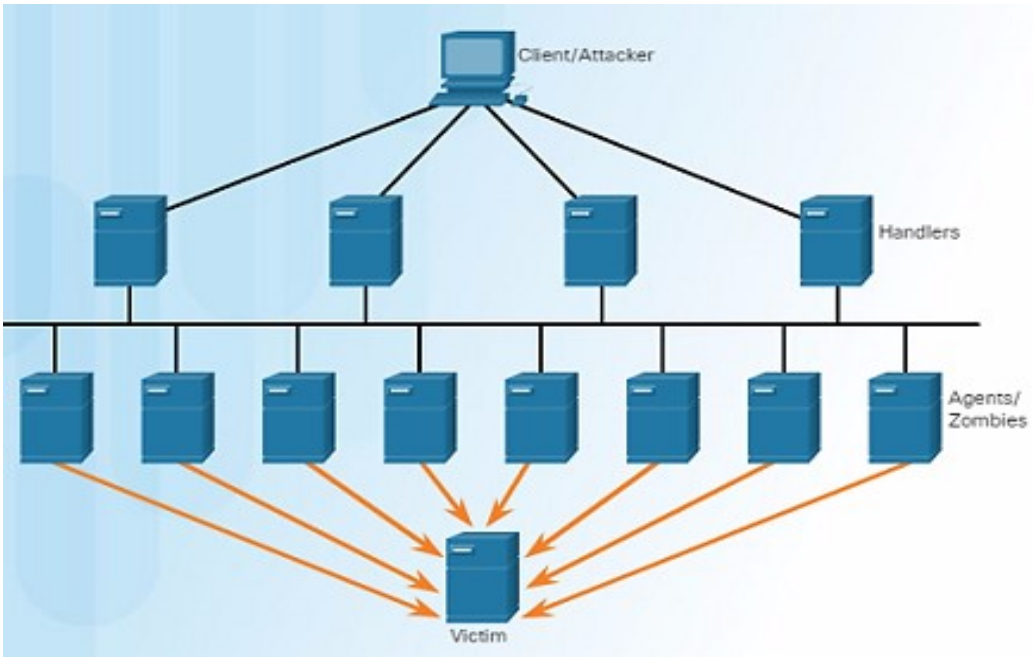


Рис. 10.1. Приклад DDoS-атаки

Найпростішим прикладом атаки DoS є ситуація, коли атакувальний пристрій надсилає величезну кількість запитів до сервера одночасно. Сервер, намагаючись обробити всі ці запити, швидко вичерпує свої ресурси – оперативну пам’ять, процесорний час або мережеву пропускну здатність. Як наслідок, сервер або перестає відповідати на будь-які запити, або істотно сповільнюється, що по суті означає відмову в обслуговуванні.

Існує кілька методів реалізації DoS-атак, один із найпоширеніших – *флудинг (flooding)*, тобто «затоплення» сервера великою кількістю запитів. Наприклад, ICMP-флуд (також відомий як "ping flood") використовує ICMP-пакети для перевантаження цільового хоста. Ще один приклад – UDP-флуд, при якому атакувальний пристрій надсилає величезну кількість UDP-пакетів на випадкові порти, змушуючи сервер витрачати ресурси на обробку й перевірку відповідей.

Інший вид – атаки на рівні протоколів, наприклад, *SYN flood*, яка використовує особливості TCP-з’єднання. Зловмисник надсилає велику кількість TCP SYN-запитів для встановлення з’єднання, але не завершує триетапний процес встановлення TCP-з’єднання («рукостискання»). Через це сервер резервує ресурси для з’єднання, яке насправді не буде використано, і з часом не має змоги обробляти нові запити.

DoS-атаки можуть бути одиночними – тобто здійснюваними з одного пристрою, або розподіленими (DDoS – Distributed Denial of Service), коли атака виконується одночасно з багатьох пристроїв, часто заражених ботами і об'єднаних у так звану ботмережу (botnet). DDoS-атаки є значно складнішими для відбиття, оскільки атакувальні пакети надходять з багатьох джерел, що ускладнює їх блокування без ризику обмежити мережевий трафік.

Наслідки DoS-атаки можуть бути різноманітними: від короткочасної недоступності вебсайту до фінансових втрат, зниження довіри клієнтів, порушення угод про рівень обслуговування SLA та репутаційних збитків для компаній. Наприклад, великі корпорації або державні установи можуть стати ціллю DoS-атак у рамках кібервійни або політичного протесту.

Захист від DoS-атак вимагає комплексного підходу. До основних методів належать фільтрація підозрілих запитів, використання міжмережевих екранів (firewalls), систем виявлення вторгнень IDS,

балансувальників навантаження (load balancers), а також послуг провайдерів DDoS-захисту (наприклад, Cloudflare, Akamai, Radware). Багато сучасних рішень включають автоматичне розпізнавання шаблонів трафіку, геолокаційний аналіз, а також використання штучного інтелекту для прогнозування потенційних атак.

З розвитком Інтернету речей загроза DoS-атак зросла, оскільки безпека багатьох "розумних" пристроїв залишається на низькому рівні. Зловмисники можуть легко захопити управління десятками тисяч таких пристроїв і використати їх для запуску масштабних DDoS-атак. Саме тому безпека як серверів, так і клієнтських пристроїв є критично важливою складовою сучасної кібергігієни.

Не менш небезпечними є атаки, спрямовані на перехоплення або крадіжку інформації. У корпоративному середовищі це може включати витік персональних даних клієнтів, комерційних таємниць або конфіденційної внутрішньої документації. Злочинці часто використовують соціальну інженерію для отримання доступу до паролів або введення користувачів в оману.

Жодна система безпеки не є ідеальною, тому для досягнення прийнятного рівня захисту рекомендується застосовувати багаторівневий підхід. Наприклад, якщо антивірус не виявить загрозу, її може заблокувати міжмережевий екран або система виявлення вторгнень. Такий підхід дозволяє компенсувати недоліки окремих елементів захисту та підвищує загальну надійність системи.

У домашньому середовищі політика безпеки зазвичай передбачає прості рішення – наприклад, вбудований брандмауер у маршрутизаторі, антивірусне ПЗ на комп’ютері та базові налаштування безпеки Wi-Fi. У деяких випадках користувачі можуть покладатися на засоби захисту, надані інтернет-провайдером. Натомість у корпоративних мережах застосовуються розгалужені системи захисту з гнучкими правилами фільтрації, контролем доступу до ресурсів, шифруванням даних і сегментацією мережі.

Для малих офісів до базового рівня захисту зазвичай належать антивірусне ПЗ, антишпигунські інструменти та міжмережеві екрани, які можуть бути інтегровані в маршрутизатори або окремі апаратні пристрої. Для великих підприємств до цього додаються спеціалізовані міжмережеві екрани з розширеними функціями фільтрації, списки

контролю доступу ACL, системи IPS для виявлення складних атак, а також VPN-технології для безпечного віддаленого доступу.

Безпека мережі повинна враховувати не лише сучасний стан, а й бути готовою до змін і масштабування. З розвитком Інтернету речей, хмарних технологій та мобільних пристроїв вимоги до захисту постійно зростають. Пристрої та користувачі мережі стають дедалі мобільнішими, що потребує адаптивних і динамічних рішень безпеки.

Уразливості можуть бути як програмного, так і апаратного характеру. Фізичні загрози включають пошкодження мережевого обладнання, вплив екстремальних температур або вологості, перебої з електроживленням, а також людський фактор – неправильне обслуговування обладнання чи нехтування базовими правилами безпеки. В усіх цих випадках важливо не лише мати технічні засоби захисту, але й забезпечити належний рівень організаційної культури в галузі інформаційної безпеки.

Таким чином, комплексний підхід до побудови захищених мережевих інфраструктур передбачає використання як технічних засобів, так і організаційних заходів, які разом формують надійну систему захисту від численних зовнішніх і внутрішніх загроз.

10.2. Резервне копіювання, оновлення і установка виправлень

Антивірусне програмне забезпечення здатне виявляти більшість шкідливих програм, таких як віруси і троянські програми, а також ефективно перешкоджати їх подальшому поширенню в мережі. Впровадження антивірусу може відбуватися як безпосередньо на комп’ютерах користувачів, так і централізовано на рівні всієї мережі організації. Важливо регулярно оновлювати антивірусні бази даних, адже це забезпечує захист від нових загроз, які постійно з’являються у вигляді нових вірусів або троянців. Підприємствам рекомендується уважно слідкувати за оновленнями і вчасно встановлювати їх, щоб підтримувати ефективність захисту.

Для зменшення ризику зараження вірусами-черв’яками потрібні злагоджені дії як системних адміністраторів, так і мережевих фахівців. Вони повинні впроваджувати комплекс заходів, спрямованих на мінімізацію можливого поширення таких загроз у мережі. До таких

заходів відносять стримування, тобто обмеження розповсюдження черв'яків в межах внутрішньої мережі організації, що допомагає локалізувати загрозу і не допустити її масштабування. Крім цього, важливо проводити регулярну протидію – своєчасно оновлювати операційні системи, виправляти виявлені вразливості та здійснювати сканування систем для виявлення потенційних загроз.

Не менш важливо відслідковувати комп'ютери, які могли бути заражені, та ізолювати їх у своєрідний карантин, щоб уникнути передачі вірусу іншим пристроям мережі. У разі виявлення зараження проводиться лікування – видалення шкідливого програмного забезпечення та відновлення пошкоджених систем. Іноді для повного усунення вірусу-черв'яка доводиться навіть перевстановлювати операційну систему, особливо якщо зараження глибоке і вплинуло на ключові компоненти.

Одним із найефективніших способів знизити наслідки атак вірусів-черв'яків є своєчасне отримання оновлень безпеки від виробника операційної системи та їх встановлення на всі вразливі копії систем у мережі. Це може бути складним завданням, особливо якщо в локальній мережі працює багато неконтрольованих користувачем пристроїв.

Для спрощення керування великою кількістю комп'ютерів рекомендується створювати образ стандартного програмного забезпечення – це заздалегідь налаштована операційна система з усіма потрібними додатками, авторизованими для використання. Такий образ розгортають на нових або оновлених машинах, що допомагає стандартизувати середовище і підвищити рівень безпеки.

Однак безпекові вимоги змінюються, і навіть для розгорнутих систем часто необхідно встановлювати нові оновлення.

Для автоматизації цього процесу може використовуватися центральний сервер оновлень, з яким комп'ютери мережі зв'язуються через певні інтервали часу.

Всі нові виправлення безпеки завантажуються з цього сервера і встановлюються автоматично, без необхідності втручання користувача. Такий підхід значно підвищує швидкість реакції на нові загрози і знижує ризики, пов'язані із затримкою встановлення важливих оновлень.

10.3. Аутентифікація, авторизація і облік

Служби мережевої безпеки, такі як аутентифікація, авторизація та облік, становлять фундаментальну інфраструктуру для контролю доступу до мережевих пристроїв. Вони разом формують комплексний механізм, що дозволяє не лише перевіряти особу користувача (аутентифікація), але й визначати, які саме дії він може виконувати після входу в мережу (авторизація), а також вести детальний запис його дій у процесі роботи з ресурсами (облік).

Такий підхід значно підвищує рівень безпеки і дає змогу ефективно керувати доступом у порівнянні з простими методами, які базуються лише на локальних налаштуваннях пристроїв, наприклад, через консольний доступ або базові команди аутентифікації.

AAA (Authentication, Authorization, Accounting) – це модель управління доступом до мережевих ресурсів, яка забезпечує комплексний підхід до безпеки.

Аутентифікація (Authentication) визначає, хто користувач, тобто перевіряє його особу за допомогою пароля, сертифіката, біометричних даних чи інших механізмів.

Авторизація (Authorization) визначає, що саме дозволено користувачу робити після підтвердження його особи – наприклад, доступ до певних сервісів чи файлів.

Облік (Accounting) фіксує дії користувача: коли він увійшов у систему, які ресурси використовував і як довго.

Ці три компоненти разом дозволяють забезпечити контроль доступу, відстеження активності та формування звітності, що є критично важливим для безпеки корпоративних мереж.

Для підтвердження особи користувачів і адміністраторів використовуються різні способи, які можуть включати введення логіна та пароля, проходження через системи «запит-відповідь», застосування карток-маркерів та інші методи ідентифікації.

У невеликих мережах зазвичай застосовується локальна аутентифікація, коли кожен мережевий пристрій має власну базу даних користувачів із їхніми обліковими записами.

Наприклад, якщо в офісі кілька маршрутизаторів і комутаторів, то на кожному з них зберігаються імена користувачів та паролі для

доступу. Однак у разі збільшення кількості пристроїв, наприклад, до сотні, керування такими обліковими записами стає дуже складним – потрібно повторно додавати і оновлювати інформацію на кожному окремому пристрої. Це значно ускладнює масштабування та підтримку безпеки.

У великих корпоративних мережах для вирішення цієї проблеми рекомендується впроваджувати зовнішню аутентифікацію, яка дозволяє централізовано зберігати всі облікові записи користувачів на одному сервері, що суттєво спрощує управління ними.

Користувачі проходять перевірку особи не на кожному окремому пристрої, а через цей зовнішній сервер, який виконує відповідні функції.

Найпоширенішими стандартами для таких серверів є RADIUS і TACACS+.

RADIUS – це відкритий стандарт, який має невеликі вимоги до ресурсів процесора і пам'яті, тому він широко підтримується різноманітними пристроями, такими як комутатори, маршрутизатори і бездротові точки доступу.

TACACS+ забезпечує більш модульний підхід, окремо керуючи аутентифікацією, авторизацією і обліком через сервер безпеки, що підвищує гнучкість і контроль над доступом.

Після успішної аутентифікації відбувається авторизація – визначається, які ресурси і функції доступні конкретному користувачу.

Наприклад, адміністратор може налаштувати, щоб користувач із іменем «student» мав дозвіл підключатися до сервера *serverXYZ* виключно через протокол SSH, але не мав доступу до інших сервісів. Це гарантує, що користувачі не матимуть зайвих прав, що підвищує безпеку системи.

Облік же фіксує всі дії користувачів: які ресурси вони використовували, скільки часу були підключені, які зміни зробили. Це дозволяє адміністраторам детально відстежувати активність, аналізувати потенційні загрози і виявляти порушення.

Аналогія з кредитною карткою є досить влучною: як карта визначає, хто може нею користуватися, які суми можна витратити, і фіксує всі покупки, так і AAA-системи керують доступом користувачів і ведуть повний облік їх дій у мережі. Це дає змогу підвищити рівень захищеності і краще контролювати використання мережевих ресурсів.

10.4. Міжмережеві екрани

Крім захисту окремих комп'ютерів або серверів, які підключені до мережі, важливо також контролювати увесь мережевий трафік, що проходить між різними сегментами мережі. Такий контроль допомагає запобігти проникненню небажаного або шкідливого трафіку, а також забезпечує безпеку всього середовища. Одним із найпоширеніших і найефективніших інструментів для захисту мережі від зовнішніх загроз є *міжмережевий екран (firewall)*, який встановлюється на межі між двома або більше мережами і виконує ретельний контроль над усім трафіком, що рухається між ними.

Міжмережевий екран (брандмауер) – це програмний або апаратний засіб захисту комп'ютерної мережі, який контролює вхідний та вихідний мережевий трафік на основі заданих правил безпеки. Основна мета міжмережевого екрану – не допустити несанкціонований доступ до приватної мережі ззовні, а також обмежити небезпечну або непотрібну активність усередині мережі. Для цього використовуються різні методи фільтрації та перевірки трафіку.

Брандмауер аналізує мережеві пакети та визначає, чи дозволити чи заборонити їхню передачу, виходячи з політики безпеки. Він працює на основі визначених критеріїв: IP-адреси відправника і одержувача, портів, типу протоколу, вмісту пакетів та інших параметрів.

У сучасних мережах міжмережевий екран є обов'язковим елементом для побудови захищеної IT-інфраструктури (рис. 10.2).

Рис. 10.2. Міжмережевий екран (брандмауер) – апаратна реалізація

Розглянемо основні типи міжмережевих екранів.

Пакетні фільтри (Packet Filtering Firewalls) – базовий тип брандмауера, який перевіряє кожен IP-пакет окремо, не враховуючи контекст з'єднання. Рішення приймається на основі адрес, портів і протоколів. Вони дуже швидкі, але мають обмежену функціональність.

Становий брандмауер (Stateful Inspection) відслідковує стан з'єднання (стан TCP-з'єднань, сесій тощо). Такі екрани зберігають інформацію про активні з'єднання і приймають рішення з урахуванням контексту. Це значно підвищує безпеку порівняно з фільтрами пакетів.

Проксі-брандмауери (Application-level Gateways) працюють на прикладному рівні і можуть аналізувати вміст трафіку. Проксі виступає посередником між користувачем і зовнішнім ресурсом, перевіряючи запити на наявність шкідливого коду чи порушень політики.

Брандмауери наступного покоління (Next-Generation Firewalls, NGFW) поєднують функціональність станового фільтрування з поглибленим аналізом трафіку, виявленням загроз, фільтрацією вмісту, DPI (deep packet inspection), контролем додатків та іноді інтегрованим антивірусом. Це найефективніший тип для захисту від складних атак.

Хмарні міжмережеві екрани реалізовані в середовищі хмарних провайдерів. Забезпечують гнучкість і масштабованість у віртуалізованому середовищі. Такі рішення особливо актуальні для компаній, які використовують хмарні інфраструктури (наприклад, AWS, Azure, Google Cloud).

Розглянемо основні компоненти міжмережевого екрана.

Інтерфейси зв'язку (мережеві порти) призначені для прийому та передачі даних між внутрішньою мережею та зовнішніми мережами.

Процесор і оперативна пам'ять відповідають за обробку політик фільтрації, перевірку стану з'єднань, кешування даних і забезпечення роботи програмного забезпечення екрана.

Операційна система або прошивка керує роботою пристрою, виконує основні функції фільтрації трафіку, логування та управління політиками.

Політики безпеки ACL (Access Control Lists) – правила, які визначають, який трафік дозволено, а який – заборонено.

Модуль логування та моніторингу веде журнал подій, що дозволяє адміністратору аналізувати інциденти, перевіряти підозрілу активність та покращувати політику безпеки.

Деякі міжмережеві екрани, особливо апаратні NGFW, також можуть включати вбудовані антивірусні сканери, системи запобігання вторгненням, VPN-шлюзи, засоби контролю програм і фільтрацію вебконтенту. Таким чином, міжмережевий екран – це багатофункціональний інструмент, який забезпечує як базову, так і глибоку мережеву безпеку, завдяки поєднанню інтелектуального аналізу трафіку з гнучкими політиками контролю доступу.

Одним із найпростіших методів є фільтрація пакетів, коли доступ дозволяється або забороняється на основі IP-адрес або MAC-адрес джерела чи отримувача. Наприклад, якщо в мережі є відомі IP-адреси зловмисників, їхній трафік можна повністю блокувати. Інший підхід – фільтрація за типом застосунку, що базується на номерах портів. Таким чином можна, наприклад, заборонити доступ до певних служб, таких як FTP чи BitTorrent, якщо вони не потрібні в організації.

Також широко застосовується фільтрація за URL-адресами, яка обмежує доступ до конкретних вебсайтів або груп сайтів за ключовими словами. Це часто використовується для блокування небажаного контенту, наприклад, соцмереж або ресурсів із шкідливим програмним забезпеченням.

Ще одним складнішим і більш ефективним методом є аналіз пакетів з урахуванням станів з'єднань, так званий *SPI (Stateful Packet Inspection)*, що перевіряє, чи є вхідний пакет частиною вже встановленого з'єднання або відповіддю на запит із внутрішньої мережі. Якщо пакет не відповідає цим критеріям, він відхиляється. Ця технологія допомагає блокувати різні атаки, наприклад, DoS (відмову в обслуговуванні), коли мережа атакована численними непотрібними запитами, що призводить до її перевантаження.

Міжмережеві екрани можуть використовувати один або декілька таких методів одночасно, підвищуючи тим самим рівень захисту. Крім того, багато з них підтримують технологію трансляції мережевих адрес NAT, що перетворює внутрішні IP-адреси приватної мережі на зовнішні публічні адреси. Це приховує справжню структуру мережі і ускладнює потенційним зловмисникам доступ до внутрішніх ресурсів.

Існують різні варіанти реалізації міжмережевих екранів залежно від їх призначення та масштабу.

Апаратні міжмережеві екрани – це спеціалізовані пристрої, які часто називають «пристроями захисту» і встановлюються між мережею організації та мережею Інтернет. Вони мають високу продуктивність і можуть обробляти великі обсяги трафіку.

Інший варіант – серверні міжмережеві екрани, які працюють як програмне забезпечення в операційних системах, наприклад, UNIX, Windows або Novell. Такі рішення підходять для захисту серверів і мереж невеликого чи середнього масштабу.

Ще один тип – інтегровані міжмережеві екрани, які додають функції фільтрації до існуючих пристроїв, таких як маршрутизатори. Це дозволяє оптимізувати використання обладнання, поєднуючи маршрутизацію та захист в одному пристрої.

Окремо варто виділити персональні міжмережеві екрани, які встановлюються безпосередньо на робочих станціях користувачів. Вони не призначені для захисту всієї мережі, а оберігають конкретний комп'ютер від шкідливого трафіку. Такі брандмауери можуть бути вбудовані у сучасні операційні системи або додані як сторонні програми. Наприклад, популярні персональні міжмережеві екрани є у Windows або сторонніх антивірусних пакетах.

10.5. Захист пристроїв комп'ютерної мережі

Безпека мережі завжди залежить від її найслабшої ланки – якщо хоч один елемент системи вразливий, це може поставити під загрозу всю мережу. Зазвичай у засобах масової інформації основну увагу приділяють зовнішнім загрозам, таким як інтернет-черв'яки або атаки типу DoS (відмова в обслуговуванні). Проте, не менш важливою є безпека внутрішньої мережі, адже саме там відбувається основна робота користувачів і розташовані ключові ресурси організації. Внутрішня мережа зазвичай складається з так званих кінцевих (крайових) мережевих пристроїв – це окремі комп'ютери, ноутбуки, планшети, сервери або навіть смартфони, які виступають клієнтами мережі.

Якщо кінцеві пристрої, якими користуються співробітники, не захищені належним чином, жодні засоби безпеки периметра мережі не зможуть гарантувати її надійний захист. Наприклад, якщо працівник встановить на свій ноутбук шкідливу програму або не оновить операційну систему, вірус може поширитися далі всередину мережі,

уражаючи інші комп’ютери та сервери. Саме тому захист крайових пристроїв – одна з найскладніших і найважливіших задач для мережевого адміністратора. Тут великого значення набуває людський фактор: співробітники мають знати, як правильно користуватися пристроями і дотримуватися встановлених правил безпеки.

Компанії потрібно розробити детальні внутрішні політики безпеки, які будуть включати рекомендації щодо використання антивірусного програмного забезпечення, налаштувань системи, правил доступу та інших заходів. Важливо не лише створити ці документи, а й провести навчання для працівників, щоб кожен розумів свої обов’язки. Наприклад, політика може вимагати, щоб на всіх пристроях було встановлено антивірусне ПЗ з регулярним оновленням баз даних, а також щоб користувачі не відкривали підозрілі листи чи не встановлювали неперевірені програми.

Крім того, захист крайових пристроїв також повинен включати безпеку мережевої інфраструктури другого рівня. Це важливо, оскільки існують специфічні атаки, які працюють саме на цьому рівні, наприклад, *спуфінг MAC-адрес* – коли зловмисник підмінює свою адресу на адресу довіреного пристрою, або атаки типу «мережевий шторм», які викликають перевантаження мережі великою кількістю трафіку.

Для боротьби з такими атаками застосовується мінімізація ризиків – набір методів і правил, які дозволяють швидко виявляти і блокувати підозрілі дії в мережі.

Коли на пристрій встановлюють нову операційну систему, її безпекові налаштування зазвичай встановлені за замовчуванням і часто є недостатньо надійними. Наприклад, типово можуть залишатися стандартні імена користувачів та паролі, що створює вразливість. Тому першим кроком має бути негайна зміна цих налаштувань. До інших простих, але важливих заходів відносяться обмеження доступу до системних ресурсів лише для користувачів із відповідними правами, а також відключення або видалення будь-яких служб і програм, які не використовуються. Наприклад, якщо на сервері не потрібен FTP-сервер, його краще відключити, щоб уникнути можливих вразливостей.

Постійне оновлення пристроїв є ще одним критично важливим аспектом. Часто пристрої, які доставляються із заводу, не мають останніх виправлень безпеки, оскільки вони могли тривалий час

зберігатися на складі. Тому перед тим, як їх підключити до мережі, потрібно інсталювати всі актуальні оновлення. Це допомагає захистити пристрої від відомих вразливостей і зменшує ризик злому.

Ще один ключовий момент – використання надійних паролів. Існують стандартні рекомендації, яких слід дотримуватися, щоб знизити ризик підбору пароля зловмисниками. Пароль повинен бути довжиною не менше восьми символів, а краще – десяти і більше. Важливо створювати складні комбінації, що включають великі й малі літери, цифри, а також спеціальні символи та, за можливості, пропуски. Наприклад, замість простого пароля «password123» краще обрати «P@ssw0rd!2025». Не рекомендується використовувати паролі, що складаються з простих слів, імен рідних чи домашніх улюбленців, дат народження або послідовностей типу «12345» чи «abcdef».

Цікавою порадою є навмисне внесення помилок у пароль, наприклад, заміна «Ivanenko» на «1vAnENko» або «Security» на «S3cur1ty». Це робить пароль менш передбачуваним для автоматизованих систем підбору. Крім того, паролі потрібно регулярно змінювати – навіть якщо здається, що вони залишилися в таємниці, це значно знижує шанси їхнього використання зловмисниками у майбутньому.

VPN (Virtual Private Network) – віртуальна приватна мережа – це технологія, яка забезпечує захищене з'єднання через публічні мережі, зокрема, через мережу Інтернет. VPN дозволяє створювати зашифрований "тунель" між пристроєм користувача і віддаленим сервером або приватною мережею, забезпечуючи конфіденційність, цілісність і захист переданої інформації (рис. 10.3).

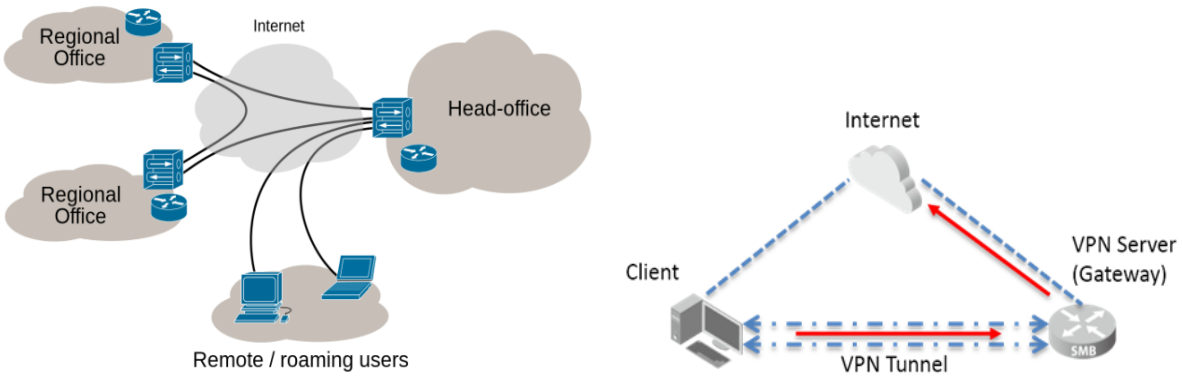


Рис. 10.3. Приклади використання технології VPN

Коли користувач підключається до VPN, його трафік спочатку шифрується і надсилається до VPN-сервера, а вже звідти – у відкриту мережу. Таким чином, зовнішні спостерігачі, провайдери чи зловмисники не можуть побачити реальний вміст запитів або сайтів, які відвідує користувач. Адреса користувача замінюється IP-адресою сервера VPN, що також забезпечує анонімність.

Наприклад, працівник компанії, перебуваючи у відрядженні, підключається до корпоративного VPN. Таким чином працівник отримує доступ до внутрішніх ресурсів компанії (серверів, баз даних, внутрішніх сайтів), ніби він перебуває у локальній мережі офісу.

Розглянемо основні типи VPN.

Remote Access VPN – забезпечує безпечне підключення окремого користувача до приватної мережі.

Site-to-Site VPN – об'єднує дві або більше локальних мереж через Інтернет, дозволяючи їхнім пристроям обмінюватися даними як у межах однієї мережі.

Client-based VPN – передбачає встановлення спеціального програмного клієнта на пристрій користувача.

Browser-based VPN – доступний через розширення у браузері (наприклад, у Google Chrome).

Наприклад, мережа філій банку у різних містах може бути об'єднана за допомогою Site-to-Site VPN, що дозволяє автоматизувати обмін інформацією між офісами через захищений тунель.

Розглянемо переваги використання VPN.

- *Захист персональних даних* – шифрування захищає від перехоплення логінів, паролів, платіжних даних.
- *Доступ до заблокованого контенту* – VPN дозволяє обійти регіональні обмеження.
- *Анонімність в мережі Інтернет* – IP-адреса користувача маскується, ускладнюючи його ідентифікацію.
- *Безпечна робота в публічних мережах* – VPN особливо актуальний при використанні Wi-Fi у кафе, аеропортах тощо.

Розглянемо недоліки використання VPN.

- *Зменшення швидкості з'єднання* через шифрування трафіку та віддалене розташування сервера.

- *Довіра до провайдера VPN* – особливо якщо використовуються безкоштовні сервіси.
- *Юридичні обмеження* – у деяких країнах використання VPN регулюється законодавством або заборонене (Іран, ОАЕ, Туркменістан).

Розглянемо приклади програмного забезпечення VPN.

OpenVPN – відкритий стандарт для створення VPN-з'єднань.

Cisco AnyConnect – корпоративне рішення для безпечного доступу до VPN.

NordVPN, ExpressVPN, ProtonVPN – комерційні сервіси для індивідуального використання VPN.

IPsec (Internet Protocol Security) – це набір протоколів для забезпечення безпеки передавання даних через IP-мережі шляхом шифрування і автентифікації кожного IP-пакета на мережевому рівні.

IPsec підтримує два режими роботи: транспортний (шифрується лише корисне навантаження) і тунельний (шифрується весь пакет), що дозволяє використовувати IPsec як основу для створення захищених VPN-з'єднань між окремими пристроями або мережами.

IPsec застосовується в корпоративних мережах для створення безпечних каналів зв’язку через мережу Інтернет, забезпечуючи конфіденційність, цілісність даних та автентифікацію джерела.

Для перевірки надійності паролів мережеві адміністратори можуть користуватися тими ж інструментами, що і хакери, наприклад, програмами для брутфорс-атак (методи «грубої сили»). Це дозволяє виявити слабкі місця у системі захисту і покращити її до реального впровадження.

Крім того, при налаштуванні мережевих пристроїв важливо слідувати внутрішнім політикам безпеки організації. Наприклад, імена пристроїв мають бути зручними для документування та обліку, але водночас не повинні розкривати зайвої інформації, яка може допомогти зловмисникам. Назва пристрою не повинна, наприклад, містити тип або модель обладнання, чи інформацію про його розташування.

Для збереження паролів у секреті використовують додаткові налаштування, наприклад, команду *service password-encryption* на мережевих пристроях, яка автоматично шифрує всі незашифровані паролі у файлах конфігурації. Це унеможливлює їх простий перегляд у вигляді звичайного тексту сторонніми особами.

Щоб забезпечити, що всі паролі мають мінімально необхідну довжину, застосовують команду *security passwords min-length*, яка задає мінімальний розмір пароля. Це автоматично блокує можливість встановлення занадто коротких і слабких паролів.

Ще один спосіб захисту – це обмеження кількості спроб входу в систему. Наприклад, команда

Router(config)# login block-for 120 attempts 3 within 60

означає, що після трьох невдалих спроб входу протягом 60 секунд доступ буде заблоковано на 120 секунд. Це допомагає запобігти атакам методом підбору пароля (брутфорс).

Важливо також використовувати банерні повідомлення, які інформують користувача про правила доступу до системи. Наприклад, команда:

Router(config)# banner motd # Unauthorized access is prohibited #

виводить повідомлення, що попереджає про відповідальність за незаконний доступ, що може стати корисним у разі судових розглядів.

Ще один корисний захід – налаштування тайм-ауту сесії, щоб автоматично розривати з'єднання користувачів, якщо вони тривалий час неактивні. Наприклад, за допомогою команд:

Router(config)# line vty 0 4

Router(config-line)# exec-timeout 10

можна встановити, що після 10 хвилин бездіяльності користувач буде автоматично відключений від пристрою. Це допомагає запобігти несанкціонованому використанню залишених без нагляду сесій.

Таким чином, для забезпечення безпеки мережі необхідно комплексно підходити до захисту кожного її елемента – від кінцевих користувачів і їх пристроїв до налаштувань мережевих пристроїв і операційних систем.

10.6. Використання протоколу SSH

Протокол Telnet, який раніше використовувався для віддаленого керування мережевими пристроями, сьогодні вважається застарілим і небезпечним. Основна проблема Telnet полягає у тому, що дані, які передаються під час сесії, не шифруються – вони відправляються у відкритому вигляді. Це означає, що будь-хто, хто має доступ до мережевого трафіку, може легко «підслухати» інформацію, включно з

паролями, використовуючи спеціалізовані програми, наприклад, програму Wireshark. Через це Telnet створює серйозну вразливість у безпеці мережі, і використання цього протоколу для керування критичними пристроями не рекомендується.

Для більш надійного і захищеного віддаленого доступу використовується протокол SSH (Secure Shell), який забезпечує шифрування даних, що передаються між клієнтом і сервером, тому навіть якщо трафік перехоплять, отримати корисну інформацію буде дуже складно.

Перехід від Telnet до SSH – це важливий крок для підвищення безпеки мережі. Налаштування SSH на мережевому пристрої, наприклад, на маршрутизаторі, складається з кількох основних кроків.

Спочатку потрібно призначити унікальне ім’я цьому пристрою, щоб ідентифікувати його в мережі.

Потім потрібно встановити доменне ім’я для IP-мережі, що допомагає у процесі генерації ключів шифрування. Для цього використовується команда *ip domain-name* в режимі глобальної конфігурації, наприклад:

ip domain-name example.com

Наступним етапом є створення криптографічних ключів, які будуть використовуватися для шифрування інформації в SSH-сесіях. Важливо визначити розмір ключа, який впливає на рівень безпеки і швидкість обробки даних. Зазвичай рекомендують використовувати ключ довжиною не менше 1024 біт.

Приклад команди для генерації ключа:

crypto key generate rsa general-keys modulus 1024

Чим більший розмір ключа, тим безпечнішим буде з’єднання, але це також збільшує навантаження на пристрій при обробці даних.

Далі потрібно створити користувача з паролем у локальній базі даних маршрутизатора. Це необхідно для аутентифікації користувачів, які будуть підключатися через SSH. Команда виглядає так:

username admin secret StrongPassword123

Вона створює користувача з іменем «admin» і захищеним паролем.

Нарешті, слід дозволити використання SSH-сесій для віддаленого доступу через віртуальні термінальні лінії VTY.

Для цього призначають локальний метод входу та вказують, що дозволено підключення лише по SSH, наприклад:

```
line vty 0 4
login local
transport input ssh
```

Після цих налаштувань до маршрутизатора можна підключатися за допомогою будь-якого SSH-клієнта, наприклад PuTTY або OpenSSH. Це забезпечує більш безпечний віддалений доступ, захищаючи інформацію від перехоплення зловмисниками.

Наприклад, у великій компанії, де кілька мережевих адміністраторів керують маршрутизаторами, перехід від Telnet до SSH дозволяє значно знизити ризики компрометації паролів і захистити внутрішню інфраструктуру від потенційних атак через незашифровані канали. Таким чином, налаштування SSH є одним із базових, але дуже важливих кроків у забезпеченні безпеки сучасних мереж.

10.7. Використання спеціальних команд

Після створення та налаштування комп'ютерної мережі важливо, щоб мережевий адміністратор міг перевірити її працездатність. Ця перевірка допомагає переконатися, що всі пристрої підключені правильно і мережа працює без збоїв. Окрім цього, адміністраторам рекомендується ретельно документувати конфігурації та топологію мережі – це полегшує подальшу підтримку та виявлення проблем.

Одним із найпростіших і найпопулярніших способів перевірити доступність інших пристроїв у мережі є команда *ping*. Вона використовується для відправки спеціальних пакетів, що дозволяють визначити, чи відповідає віддалений пристрій і чи працює зв'язок між ними. Таку перевірку часто називають *тестуванням стеку протоколів*, оскільки команда *ping* тестує взаємодію мережевого рівня (третій рівень OSI) з канальним (другим) та фізичним (першим) рівнями. За допомогою протоколу ICMP (Internet Control Message Protocol) відправляється ехо-запит, і якщо віддалений вузол активний і доступний, він повертає ехо-відповідь.

Проте команда *ping* не завжди дає повну картину про природу проблем у мережі. Вона швидше допомагає локалізувати зону, де сталася помилка, що є першим кроком для її подальшого вирішення.

Наприклад, якщо *ping* не отримує відповіді, це може означати, що вузол відсутній, маршрутизація порушена або що пакети блокуються між пристроями.

Крім перевірки доступності, *ping* можна використовувати для контролю правильності IP-конфігурації на локальному комп'ютері або мережевому пристрої. Особливо корисною є перевірка за допомогою «loopback»-адреси 127.0.0.1. Ця команда не відправляє пакети в мережу, а тестує внутрішній мережевий стек пристрою, перевіряючи роботу мережевих протоколів від верхніх рівнів до фізичного і назад. Наприклад, виконання команди:

ping 127.0.0.1

надасть відповідь, що показує кількість надісланих та отриманих пакетів, час відгуку і параметр TTL – час життя пакета, який визначає, скільки разів пакет може бути пересланий мережею, перш ніж буде знищений.

При виконанні команди *ping* в операційній системі Cisco IOS на екрані відображаються спеціальні символи, які допомагають зрозуміти результати перевірки.

Знак оклику «!» означає успішне отримання відповіді від пристрою, що підтверджує працездатність з'єднання на мережевому рівні. Крапка «.» вказує на затримку або відсутність відповіді, що може бути пов'язано з проблемами у маршрутизації або блокуванням пакетів системою безпеки. Символ «U» означає, що пристрій на маршруті повідомив про недосяжність вузла призначення, тобто пакет не може бути доставлений за вказаною адресою.

Окрім *ping*, існують більш детальні інструменти для діагностики мережі, такі як команда *tracert* (у Windows) або *traceroute* (на маршрутизаторах). Вони дозволяють простежити весь шлях, яким проходить пакет від вашого пристрою до цільового вузла, показуючи список проміжних маршрутизаторів (хопів).

Це допомагає зрозуміти, де саме відбувається збій або затримка у мережі. Наприклад, при запуску команди

tracert 10.1.0.2

ми можемо побачити, через які маршрутизатори проходить пакет, і чи всі вони відповідають. Якщо на якомусь етапі з'являються зірочки «* * *» і повідомлення про таймаут, це свідчить про те, що

маршрутизатор або пристрій на цьому кроці не відповідає. Окрім відсутності відповіді маршрутизатора, зірочки «* * *» у виведенні команди *tracert* можуть свідчити про інші причини: фільтрацію ICMP-пакетів на маршрутизаторі або брандмауері, перевищення часу очікування (timeout) через перевантаження мережі або затримки, а також налаштування пристрою, що не дозволяє надсилати відповіді на запити трасування. Іноді це може бути ознакою несправності або неправильного маршруту до вузла.

Для отримання детальної інформації про мережеві налаштування на комп'ютері з Windows часто використовують команду *ipconfig*, яка показує IP-адресу, маску підмережі, шлюз за замовчуванням та інші основні параметри. Якщо потрібно більше деталей, у тому числі фізичну (MAC) адресу мережевого інтерфейсу, застосовують команду *ipconfig/all*. Це корисно, наприклад, для визначення виробника мережевого обладнання за частиною MAC-адреси (OUI), інформацію про яку можна знайти в онлайн-базах даних.

Ще одним важливим інструментом є команда *arp*, яка відображає відповідності між IP-адресами та фізичними адресами (MAC) у кеші пристрою. За допомогою команди *arp –a* адміністратор бачить усі поточні записи, а команда *arp –d* дозволяє очистити кеш, щоб він оновився з новими даними. Команда *show ip interface brief* показує коротку зведену інформацію про всі інтерфейси маршрутизатора або комутатора, їх статус і IP-адреси. Це зручно для швидкого огляду стану пристрою. Варто пам'ятати, що комутатори, на відміну від маршрутизаторів, можуть працювати на другому рівні моделі OSI, передаючи кадри без необхідності мати IP-адресу.

IP-адреса потрібна тільки у разі, якщо адміністрування комутатора здійснюється через мережу, наприклад, за допомогою SSH.

У таких випадках для забезпечення зв'язку з комутатором, особливо якщо адміністратор працює з віддаленої мережі, необхідно налаштувати шлюз за замовчуванням.

Таким чином, поєднання інструментів *ping, traceroute, ipconfig, arp* і команд моніторингу інтерфейсів дозволяє мережевому адміністратору не лише перевірити працездатність мережі, але й локалізувати проблеми та контролювати налаштування для ефективного управління інфраструктурою. Наприклад, якщо після налаштування нової мережі

адміністратор не може підключитися до віддаленого сервера, він спочатку перевіряє його доступність за допомогою команди *ping*, потім простежує маршрут з *traceroute*, дивиться IP-налаштування через *ipconfig* і перевіряє відповідність MAC-адрес з *arp*, що дозволяє швидко знайти та виправити помилки у конфігурації.

10.8. Безпека безпровідних підключень

Заходи з безпеки бездротової мережі слід ретельно продумувати і впроваджувати ще до підключення точки доступу до локальної мережі або до мережі інтернет-провайдера. Це необхідно, щоб уникнути вразливостей, які можуть бути використані зловмисниками з самого початку роботи мережі. В першу чергу важливо змінити стандартні параметри, які зазвичай встановлені виробником і часто є загальнодоступними. Наприклад, слід обов'язково змінити стандартний SSID – ім'я бездротової мережі, а також логіни та паролі, які використовуються для доступу до пристрою. Відсутність таких змін робить мережу вразливою для злому. Крім того, рекомендується вимкнути широкомовну трансляцію SSID, що ускладнює виявлення мережі для випадкових користувачів або потенційних нападників. Також дуже важливим є налаштування шифрування, яке захищає дані під час передачі по повітрю, перетворюючи їх у незрозумілий код, що унеможливлює прочитання навіть при перехопленні сигналу.

Одним із перших і найпоширеніших протоколів шифрування є WEP (Wired Equivalent Privacy), який використовує ключі довжиною 64 або 128 біт (іноді 256 біт) для шифрування трафіку. Ключ WEP можна вводити вручну у вигляді послідовності букв і цифр або використовувати так звані фрази-паролі – прості слова або фрази, за допомогою яких автоматично генерується ключ. Для роботи мережі всі бездротові пристрої і точка доступу повинні використовувати однаковий WEP-ключ, що дозволяє їм розпізнавати й обробляти інформацію. Наприклад, якщо у нас вдома є ноутбук і смартфон, щоб обидва пристрої могли підключатися до Wi-Fi, вони мають бути налаштовані на однаковий WEP-ключ. Проте WEP має суттєві недоліки. Основна проблема полягає в тому, що ключ є статичним і однаковим для всіх користувачів мережі. Це створює вразливість, бо існують спеціальні програми, які дозволяють зловмисникам за короткий час

розшифрувати WEP-ключ і отримати доступ до всієї мережі, такі програми зловмисник може знайти в Інтернеті. Це означає, що якщо наш ключ залишатиметься незмінним, будь-хто з відповідними знаннями зможе підключитися до нашої мережі, переглянути передану інформацію або навіть викрасти її. Щоб зменшити ризик компрометації даних у мережі, рекомендується часто міняти WEP-ключ. Але краще перейти на більш сучасні та надійні протоколи.

Найпопулярнішим і більш безпечним є WPA (Wi-Fi Protected Access). Він використовує довші ключі – від 64 до 256 біт – і, на відміну від WEP, автоматично генерує нові динамічні ключі кожного разу, коли пристрій підключається до мережі. Це значно ускладнює злом, бо ключі змінюються постійно і їх практично неможливо передбачити. Наприклад, коли наш смартфон підключається до роутера з WPA, він отримує унікальний ключ шифрування на цю сесію, і цей ключ буде відрізнятися від ключа, який отримає наш ноутбук при наступному підключенні. Крім шифрування, для захисту мережі можна використовувати фільтрацію за MAC-адресами – унікальними ідентифікаторами мережевих пристроїв. Це дозволяє дозволити доступ лише певним відомим пристроям і заблокувати всі інші. Також часто впроваджують додаткові заходи аутентифікації та фільтрації трафіку, щоб запобігти несанкціонованому доступу і знизити ризики атаки.

У сучасних бездротових мережах все частіше застосовують нові протоколи безпеки, які значно перевищують за надійністю WPA. Наприклад, *WPA3* – останній стандарт, який впроваджує більш складні методи шифрування і забезпечує сильнішу аутентифікацію користувачів. Він використовує алгоритми, які захищають від атак на підбір паролів, і підтримує індивідуальне шифрування даних навіть у відкритих мережах. Також популярні рішення для корпоративного рівня – такі як IEEE 802.1X, забезпечують автентифікацію на основі серверів RADIUS, що дає можливість контролювати доступ окремих користувачів та пристроїв. Ще одним сучасним напрямом є використання протоколів захисту для бездротових мереж нового покоління, таких як Wi-Fi 6 (802.11ax), що підтримують покращене шифрування і управління доступом. Окрім цього, з’являються рішення для бездротових мереж Інтернету речей, де безпека реалізується через

спеціалізовані протоколи, адаптовані під обмежені ресурси пристроїв, наприклад, Thread або Zigbee з розширеними засобами захисту.

Контрольні запитання та задачі до теми 10

Перелік теоретичних запитань

1. Які основні категорії загроз існують для мережевої безпеки?
2. Що таке шкідливе програмне забезпечення (malware) і які його типи?
3. Як соціальна інженерія використовується для компрометації мережевої безпеки?
4. У чому полягає роль резервного копіювання в забезпеченні інформаційної безпеки?
5. Яке значення мають оновлення програмного забезпечення і виправлення безпеки?
6. Що таке AAA (Authentication, Authorization, Accounting) і як вони працюють?
7. Поясніть різницю між аутентифікацією та авторизацією.
8. Для чого використовуються міжмережеві екрани (брандмауери) у комп'ютерних мережах?
9. Які є типи брандмауерів і чим вони відрізняються?
10. Які методи захисту пристроїв комп'ютерної мережі від атак використовуються на практиці?
11. Що таке SSH і чому він є безпечнішим за Telnet?
12. Як налаштувати SSH-доступ до маршрутизатора або комутатора?
13. Які команди використовуються для перевірки та діагностики стану безпеки мережевих пристроїв?
14. Які заходи можна застосовувати для підвищення безпеки безпровідної мережі?
15. У чому відмінність між протоколами WEP, WPA та WPA3?

Перелік задач для самостійного опрацювання

Тестове завдання 1.

Який протокол використовується для безпечного керування мережевими пристроями?

а) Telnet

б) FTP

в) SSH

г) HTTP

Тестове завдання 2.

Який із наведених варіантів є методом аутентифікації?

а) Брандмауер

б) Пароль

в) Антивірус

г) Шифрування

Тестове завдання 3.

Яка основна функція брандмауера?

а) Компресія даних

б) Визначення маршруту

в) Фільтрація трафіку

г) Відновлення резервних копій

Тестове завдання 4.

Яка адреса використовується для доступу до маршрутизатора через SSH?

а) 127.0.0.1

б) IP-адреса пристрою

в) 255.255.255.0

г) MAC-адреса

Перелік використаної та рекомендованої літератури

10.1. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

10.2. Workshop on Foundational Cybersecurity Activities for IoT Device Manufacturers [Електронний ресурс]. – Режим доступу: https://doi.org/10.6028/NIST.IR.8572

10.3. Superior security outcomes [Електронний ресурс]. – Режим доступу: https://www.cisco.com/site/us/en/products/security/index.html

10.4. ENISA (European Union Agency for Cybersecurity) [Електронний ресурс]. – Режим доступу: https://www.enisa.europa.eu

10.5. Virtual Private Networking [Електронний ресурс]. – Режим доступу: https://docs.opnsense.org/manual/vpnet.html

10.6. What is IPSec [Електронний ресурс]. – Режим доступу: https://aws.amazon.com/what-is/ipsec/

РОЗДІЛ ІІ.
ТЕХНОЛОГІЇ КОМУТАЦІЇ ТА МАРШРУТИЗАЦІЇ

У сучасних комп’ютерних мережах ефективне передавання даних є критично важливим для забезпечення швидкої, надійної та безперервної комунікації між пристроями. Саме тому технології комутації та маршрутизації відіграють ключову роль у побудові та функціонуванні локальних і глобальних мереж. Ці технології визначають, яким чином інформація передається від джерела до місця призначення, якими маршрутами вона проходить, і які рішення приймаються для оптимізації цього процесу. Комутація забезпечує передачу даних усередині локальної мережі, дозволяючи пристроям обмінюватися інформацією з мінімальною затримкою, тоді як маршрутизація забезпечує зв’язок між різними мережами, визначаючи найефективніший шлях для проходження пакетів. Успішна інтеграція цих технологій є основою продуктивної та масштабованої мережевої інфраструктури.

Тема 11. Локальні мережі. Технології комутації

Локальні обчислювальні мережі LAN є основою більшості сучасних інформаційних систем. Вони забезпечують об’єднання комп’ютерів, серверів, принтерів та інших пристроїв в межах обмеженої географічної зони – наприклад, офісу, навчального закладу або підприємства. Основна мета локальних мереж – забезпечити швидкий, надійний і безпечний обмін даними між пристроями та спільне використання ресурсів. Ефективне функціонування таких мереж значною мірою залежить від технологій комутації, які дозволяють організувати швидку доставку даних між кінцевими вузлами.

Технології комутації реалізуються за допомогою спеціалізованих пристроїв – комутаторів, які працюють на канальному рівні моделі OSI. Вони приймають рішення про передачу кадрів даних на основі MAC-адрес, що дозволяє зменшити колізії в мережі та підвищити її продуктивність.

Сучасні комутатори підтримують численні додаткові функції, такі як віртуальні локальні мережі VLAN, пріоритезація трафіку (QoS) та захист від атак, що робить їх ключовими елементами будь-якої локальної мережевої інфраструктури.

11.1. Принципи роботи комутатора

Комутатори забезпечують ефективну передачу даних між різними вузлами локальної мережі. Залежно від функціональності, комутатори поділяються на *некеровані (unmanaged)*, *керовані (managed)* та *смарт-комутатори (smart switches)*.

Некеровані комутатори прості у використанні, не потребують конфігурації, і найчастіше застосовуються в домашніх чи малих офісних мережах. *Керовані комутатори*, навпаки, мають розширену функціональність – вони підтримують VLAN, QoS, SNMP, моніторинг, резервування, безпеку, і дозволяють адміністрування через CLI або вебінтерфейс. *Смарт-комутатори* займають проміжне місце між двома попередніми типами – вони мають обмежені можливості налаштування, підходять для невеликих компаній і доступні за ціною.

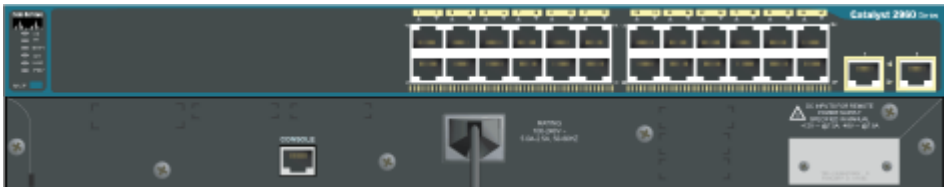
На ринку існує кілька основних виробників мережевого обладнання, серед яких Cisco, TP-Link, D-Link, HPE (Aruba), MikroTik, Netgear, Ubiquiti, Juniper та інші. Найвідомішою в корпоративному сегменті є компанія Cisco – її пристрої відзначаються високим рівнем надійності, підтримкою стандартів безпеки та гнучким управлінням. HPE Aruba також орієнтована на підприємства і пропонує «розумні» рішення для бездротових і провідних мереж. D-Link та TP-Link мають широкий асортимент бюджетних моделей, що підходять для малого бізнесу або домашніх користувачів. MikroTik вирізняється гнучкістю налаштувань і функціональністю при доступній ціні, хоча вимагає досвіду в конфігуруванні. Ubiquiti ставить акцент на централізоване управління і добре підходить для побудови Wi-Fi-мереж.

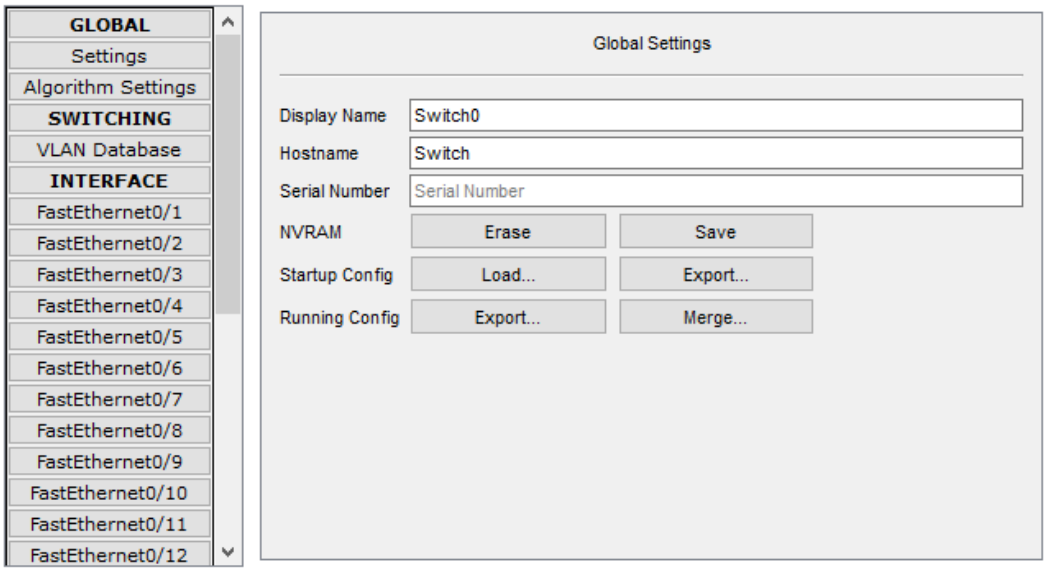
Відмінності між комутаторами різних виробників полягають не лише в ціні чи дизайні, а й у прошивці, підтримці протоколів, можливостях масштабування, інтерфейсі керування, а також екосистемі. Наприклад, Cisco використовує власну операційну систему IOS, яка має гнучкий CLI-інтерфейс, тоді як TP-Link або D-Link частіше застосовують вебінтерфейси з базовим набором функцій.

MikroTik базується на RouterOS, що дозволяє будувати складні маршрутизовані мережі навіть із простих пристроїв.

Ubiquiti підтримує систему централізованого керування UniFi Controller, яка забезпечує візуальний контроль усієї мережі.

На рис. 11.1 представлено приклад графічного представлення для налаштування інтерфейсів комутатора Cisco Catalyst 2960 у середовищі моделювання Cisco Packet Tracer. У верхній частині розміщене зображення самого комутатора з 24 портами FastEthernet та 2 гігабітними портами, що дає уявлення про фізичну модель пристрою.





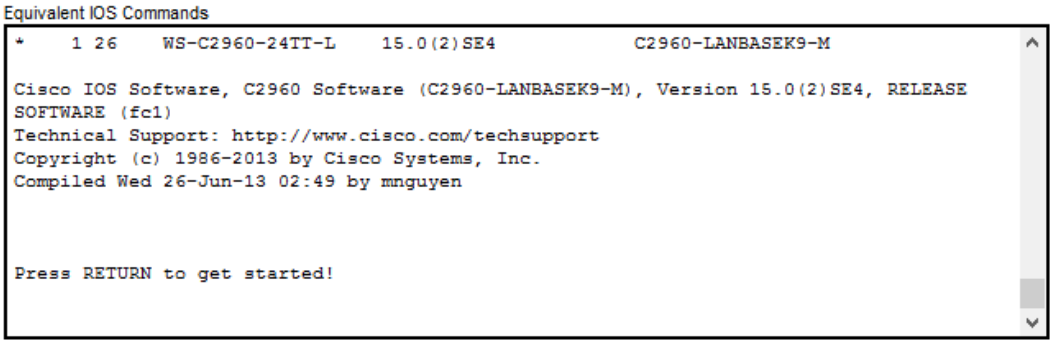


Рис. 11.1. Відображення моделі комутатора Cisco Catalyst 2960 в середовищі моделювання Cisco Packet Tracer

У центральній частині зображення знаходиться вкладка "Global Settings", яка дозволяє задати основні параметри пристрою. Зокрема, тут можна змінити ім’я пристрою (Display Name), ім’я хоста (Hostname), а також переглянути або задати серійний номер.

Нижче розміщені кнопки для роботи з конфігураціями: збереження та видалення конфігурацій із пам’яті NVRAM, завантаження або експорт початкової (Startup Config) та поточної (Running Config) конфігурації комутатора.

Ліва частина інтерфейсу містить вкладки для доступу до інших функціональних розділів: "GLOBAL", "SWITCHING", "VLAN Database", а також список інтерфейсів FastEthernet, які можна налаштовувати індивідуально. Це дозволяє адміністратору швидко перемикатись між портами комутатора та виконувати необхідні конфігураційні дії.

У нижній частині рисунка наведено текстову інформацію, яка містить еквівалентні команди Cisco IOS для CLI (інтерфейсу командного рядка). Тут вказано модель комутатора (WS-C2960-24TT-L), версію програмного забезпечення 15.0(2)SE4 та назву образу ПЗ (C2960-LANBASEK9-M). Відображена також дата компіляції прошивки та технічна підтримка. Цей блок підказує, що користувач може почати роботу, натиснувши клавішу Enter у CLI-режимі.

В сучасних комп’ютерних мережах важливим завданням є забезпечення швидкого, надійного й захищеного обміну даними між різними пристроями. Ця взаємодія вимагає ефективної роботи як на рівні локальної мережі, так і на рівні міжмережевої комунікації. Комутатори відіграють ключову роль у локальних мережах LAN, оскільки відповідають за підключення кінцевих пристроїв – комп’ютерів, принтерів, IP-телефонів та інших вузлів – до внутрішнього середовища мережі. Вони керують трафіком усередині локального середовища, забезпечуючи ефективне переміщення даних між пристроями. Маршрутизатори здійснюють передачу даних між різними мережами, наприклад, між кількома LAN або між локальною мережею й мережею Інтернет, і зазвичай не працюють безпосередньо з окремими вузлами мережі.

Ключова концепція роботи комутаторів ґрунтується на механізмах комутації та пересилки кадрів. Незалежно від типу мережі – локальної, глобальної або телефонної – комутатори приймають рішення про подальшу передачу кадру, керуючись двома основними параметрами: портом, через який кадр надійшов (вхідний порт), і MAC-адресою пристрою призначення.

Комутатор визначає, через який вихідний порт передати дані, щоби доставити їх до конкретного вузла. У разі невизначеності щодо призначення кадру, комутатор використовує метод широкомовної розсилки, надсилаючи кадр усім портам, окрім вхідного.

Функціональність комутатора базується на його здатності аналізувати трафік і створювати *таблицю MAC-адрес* – структуру, що зберігає відповідність між MAC-адресами пристроїв і портами, до яких вони підключені.

Щоразу, коли комутатор отримує кадр, він зчитує MAC-адресу джерела та додає її до таблиці, якщо такої ще немає. З часом комутатор формує повну картину топології підключених пристроїв.

У мережах з кількома об'єднаними комутаторами один порт може відповідати кільком MAC-адресам, якщо він веде до іншого комутатора або сегмента мережі.

Для зберігання та швидкого доступу до інформації в таблиці MAC-адрес комутатори використовують спеціальний тип пам’яті – CAM (Content Addressable Memory), або асоціативну пам’ять.

Цей тип пам’яті дозволяє здійснювати миттєвий пошук за вмістом, що значно пришвидшує процес маршрутизації кадрів на канальному рівні моделі OSI.

Завдяки CAM, комутатор може ефективно пересилати трафік до конкретного вузла без зайвого навантаження на мережу.

Високопродуктивні комутатори використовують CAM-пам’ять у поєднанні зі спеціалізованими мікросхемами та прошивками для досягнення максимальної швидкості та точності обробки даних.

11.2. Методи пересилання даних на комутаторі

У міру того, як корпоративні мережі зростали, їх продуктивність почала суттєво знижуватись через збільшення навантаження та частоту колізій. Щоб зменшити ці проблеми, у мережеву інфраструктуру стали додавати мости Ethernet, які дозволяли обмежувати розмір колізійного домену. Проте з розвитком технологій у 1990-х роках мости були витіснені Ethernet-комутаторами.

Завдяки вдосконаленню інтегральних мікросхем стало можливим перенести процес прийняття рішень про пересилку кадрів з програмного

забезпечення на апаратний рівень, а саме – на спеціалізовані інтегральні мікросхеми ASIC.

ASIC (Application-Specific Integrated Circuit) – це інтегральна мікросхема, спроєктована для виконання конкретного завдання або набору завдань, на відміну від універсальних мікросхем, таких як процесори загального призначення. Завдяки вузькій спеціалізації, ASIC забезпечує високу продуктивність, енергоефективність та швидкодію в конкретних застосуваннях, зокрема в мережевому обладнанні, криптомайнінгу, обробці сигналів та мобільних пристроях.

Одним із основних методів, який використовують сучасні комутатори, є *комутація з проміжним зберіганням (store-and-forward switching)*. Цей підхід передбачає повне приймання кадру комутатором, його збереження у буфері, перевірку на наявність помилок (зокрема, перевірку FCS – frame check sequence), і лише потім пересилання до вихідного порту.

Якщо виявлено помилку, кадр відкидається, що забезпечує високу якість передачі даних і запобігає поширенню пошкоджених кадрів у мережі. Завдяки буферизації цей метод також дозволяє ефективно працювати з різними швидкостями інтерфейсів, наприклад, коли кадр надходить через порт 100 Мбіт/с, а передається через порт 1 Гбіт/с.

Основною перевагою комутації з проміжним зберіганням є її здатність виявляти пошкоджені кадри, покращуючи загальну надійність мережі. Проте цей підхід може спричинити деяке підвищення затримки, оскільки комутатор чекає завершення прийому всього кадру перед тим, як ухвалити рішення про його пересилку.

У той же час, *наскрізна комутація (cut-through switching)* працює швидше – вона починає передавати кадр практично одразу після зчитування MAC-адреси призначення, не чекаючи завершення отримання всього кадру. Це дозволяє мінімізувати затримки, але не забезпечує виявлення помилок, тому пошкоджені кадри можуть бути передані далі в мережу.

Наскрізна комутація підходить для сценаріїв, де важлива мінімальна затримка, зокрема, у високопродуктивних обчислювальних системах або в середовищах, що обробляють великі обсяги трафіку в реальному часі. Наскрізна комутація має серйозний недолік – за

відсутності перевірки кадрів на помилки, навіть некоректні пакети можуть навантажувати мережу, що знижує ефективність роботи.

Ще однією варіацією наскрізної комутації є так званий *безфрагментний режим (fragment-free switching)*. У цьому режимі комутатор чекає прийому перших 64 байт кадру перед тим, як розпочати пересилку. Це дозволяє уникнути пересилання кадрів, пошкоджених через колізії, оскільки більшість таких помилок трапляється на початку передачі. Безфрагментна комутація поєднує в собі переваги обох підходів: вона забезпечує вищу швидкість, ніж «store-and-forward», і водночас дозволяє фільтрувати деякі помилкові пакети, чого не робить звичайна наскрізна комутація.

Загалом, Cisco та інші виробники зазвичай використовують комутацію з проміжним зберіганням як основний метод у своїх комутаторах через його надійність і точність.

Проте інші методи можуть бути доцільними залежно від специфіки мережі, вимог до швидкодії та характеру переданих даних.

11.3. Колізійні та широкомовні домени

У традиційних мережах Ethernet, які використовують концентратори (hub), усі підключені пристрої поділяють спільне середовище передачі даних. Це означає, що лише один пристрій може передавати дані в певний момент часу.

Якщо два або більше пристрої намагаються передавати одночасно, виникає *колізія* – ситуація, коли дані змішуються і стають пошкодженими, що змушує пристрої повторювати передачу. Такі частини мережі, де можливі колізії, називають *колізійними доменами*.

Наприклад, у комп’ютерному класі з 10 ПК, підключених до концентратора, усі 10 пристроїв перебувають в одному колізійному домені, і кожна нова передача може конфліктувати з іншою.

З появою комутаторів (switch) та маршрутизаторів (router) ситуація значно покращилася. Ці пристрої працюють не тільки на фізичному рівні, як концентратори, але й на канальному рівні (рівень 2) або мережевому рівні (рівень 3) моделі OSI чи TCP/IP.

Комутатори здатні розділяти мережу на менші логічні частини – сегменти, кожен з яких має власний колізійний домен.

Таким чином, коли кожен ПК або сервер підключений до окремого

порту комутатора, кожне з’єднання стає індивідуальним колізійним доменом. Це дозволяє кожному пристрою використовувати повну смугу пропускання порту, що зменшує ймовірність колізій і підвищує загальну ефективність. Такий підхід називається *мікросегментацією*.

Незважаючи на те, що комутатори фільтрують більшість кадрів, адресованих конкретним MAC-адресам, широкомовні кадри (broadcast) – тобто ті, що призначені всім пристроям у мережі – не фільтруються. Коли в мережі надсилається широкомовна розсилка, комутатор копіює та пересилає такий кадр на всі порти, крім того, з якого кадр надійшов.

Наприклад, якщо ПК надсилає ARP-запит для з’ясування MAC-адреси маршрутизатора, цей запит отримає кожен пристрій, підключений до комутатора. Така сукупність пристроїв, які отримують один і той самий широкомовний кадр, утворює широкомовний домен.

Широкомовний домен (broadcast domain) на рівні 2 обмежується комутаторами, оскільки вони пересилають широкомовні кадри на всі порти. Якщо в мережі з'єднано кілька комутаторів, ці домени об’єднуються, і кожен новий пристрій у такій мережі збільшує розмір широкомовного домену. Це може призводити до перевантаження мережі широкомовним трафіком, особливо у великих мережах.

Наприклад, велика кількість одночасних ARP-запитів або DHCP-відкликів може серйозно вплинути на продуктивність, оскільки вони розповсюджуються на всі пристрої у домені.

На відміну від комутаторів, маршрутизатори не пересилають широкомовні кадри між мережами. Вони функціонують на рівні 3 (мережевому) і здатні ізолювати широкомовні домени, тим самим обмежуючи поширення широкомовного трафіку.

Наприклад, якщо два офісні відділи підключено до різних підмереж і розділено маршрутизатором, широкомовні запити з одного відділу не потраплятимуть в інший, що зменшує навантаження і підвищує ефективність.

Таким чином, правильне проєктування мережі – з використанням комутаторів для мікросегментації та маршрутизаторів для меж широкомовних доменів – дозволяє оптимізувати як продуктивність, так і безпеку. Це особливо важливо для великих корпоративних мереж або структур із високими вимогами до швидкодії.

11.4. Методи зниження перевантажень мережі

Комутатори, які використовуються в локальних мережах, відіграють ключову роль у підвищенні ефективності та зменшенні перевантажень мережевого середовища. Однією з головних їхніх переваг є можливість сегментувати мережу на окремі колізійні домени. Кожен порт комутатора функціонує як самостійний колізійний домен, надаючи повну смугу пропускання пристрою, підключеному до нього.

Наприклад, якщо комп’ютер підключено до порту комутатора, він отримає виділену пропускну здатність – на відміну від концентратора, де всі пристрої поділяють одну й ту ж смугу. Це значно знижує ймовірність колізій і сприяє стабільності передавання даних.

Ще однією важливою функцією комутаторів є підтримка повнодуплексного зв’язку.

Повнодуплексний режим дозволяє пристроям одночасно передавати та приймати дані, на відміну від напівдуплексного режиму, де передача й прийом відбуваються по черзі. Така можливість значно підвищує швидкодію та зменшує затримки у мережі. Повнодуплексні з'єднання особливо важливі для сучасних високошвидкісних стандартів, таких як Gigabit Ethernet (1 Гбіт/с) і 10 Gigabit Ethernet (10 Гбіт/с), які широко застосовуються в дата-центрах і корпоративних мережах.

Комутатори також виконують функцію перенаправлення кадрів на основі MAC-адрес, використовуючи спеціальні таблиці (MAC-таблиці), які зберігають відповідність між MAC-адресами пристроїв і портами, до яких вони підключені.

Наприклад, якщо комутатор отримує кадр, адресований певному комп’ютеру, він не передає його по всіх портах, а лише по тому, який веде до призначення. Це зменшує навантаження в мережі та дозволяє уникнути непотрібного трафіку.

Серед апаратних характеристик комутаторів, які також сприяють зменшенню перевантаження, важливу роль відіграє висока щільність портів. Наприклад, у стандартному корпусі висотою 1,75 дюйма (1U) може вміщуватись 24 або навіть 48 портів. Це дозволяє ефективно використовувати простір у телекомунікаційній стійці, підключати більше пристроїв і масштабувати мережу без необхідності встановлення додаткового обладнання.

Комутатори корпоративного рівня можуть підтримувати навіть

сотні портів і одночасно обслуговувати велику кількість користувачів.

Ще одна перевага – наявність великих буферів кадрів. Буфери дозволяють тимчасово зберігати трафік у разі перевантаження портів. Наприклад, якщо порт, до якого підключено сервер, отримує велику кількість запитів, комутатор може тимчасово зберегти вхідні дані в буфері, а не відкинути їх, поки не звільниться пропускна здатність. Це особливо корисно в умовах високої активності або при підключенні до портів вузлів із великою кількістю паралельних з'єднань.

Комутатори також підтримують різні швидкості портів, зокрема 100 Мбіт/с, 1 Гбіт/с, 10 Гбіт/с і навіть 100 Гбіт/с. Це дає змогу підключати пристрої з різними технічними можливостями, оптимізуючи витрати без втрати продуктивності. Наприклад, звичайний офісний ПК може працювати на швидкості 1 Гбіт/с, тоді як сервер або мережевий пристрій – на 10 Гбіт/с.

Щоб забезпечити високу швидкодію, комутатори обладнані механізмами швидкої внутрішньої комутації. Вони використовують внутрішні шини або спільну пам'ять для обробки та пересилання трафіку між портами. Це дозволяє обробляти великий обсяг даних із мінімальними затримками – наприклад, переміщення файлів між двома серверами в одній мережі відбуватиметься практично миттєво.

Ще одним важливим фактором є низька вартість кожного порту, що дозволяє будувати масштабовані мережі без значних фінансових витрат. Наприклад, при проєктуванні мережі з високою щільністю користувачів комутатор з 48 портами дозволить зменшити кількість обладнання, зберігши високу швидкість з'єднань для кожного пристрою.

Загалом, комутатор – це центральний елемент локальної мережі, який забезпечує з'єднання між пристроями, контроль над потоками даних та оптимізацію пропускної здатності.

У правильно налаштованій мережі комутатори дозволяють ефективно розподіляти ресурси, забезпечуючи стабільний зв'язок. Їх можна налаштовувати як локально, так і віддалено.

Наприклад, адміністратор може підключитися до комутатора з іншого міста, якщо пристрою призначено IP-адресу і вказано шлюз за замовчуванням. Це спрощує адміністрування великих мереж і дозволяє швидко реагувати на проблеми без фізичної присутності на місці.

11.5. Базові налашування комутатора

Щоб забезпечити можливість віддаленого управління комутатором, необхідно виконати низку налаштувань, аналогічних тим, що застосовуються до звичайних комп'ютерів або серверів у мережі. Насамперед потрібно призначити IP-адресу та маску підмережі – це дозволить комутатору ідентифікувати себе в мережі та взаємодіяти з іншими пристроями.

Наприклад, IP-адреса 192.168.1.10 із маскою 255.255.255.0 дозволить комутатору працювати в межах підмережі 192.168.1.0/24.

Якщо управління комутатором передбачається з віддалених мережевих сегментів, потрібно налаштувати шлюз за замовчуванням. Це пристрій (зазвичай маршрутизатор), через який комутатор зможе надсилати пакети за межі своєї локальної мережі. Наприклад, якщо шлюз має адресу 192.168.1.1, комутатор буде надсилати весь трафік, призначений іншим підмережам, саме на цю IP-адресу. Налаштування шлюзу виконується просто – командою *ip default-gateway 192.168.1.1*.

Для реалізації таких налаштувань зазвичай використовується *SVI (Switched Virtual Interface)* – віртуальний інтерфейс, прив'язаний до певної VLAN (віртуальної локальної мережі), через який здійснюється керування комутатором.

VLAN (Virtual LAN) – це логічно розділені групи портів на одному комутаторі, VLAN використовується для логічної сегментації мережі. Наприклад, порти 1–10 можуть бути призначені для VLAN 10, а порти 11–20 – для VLAN 20, навіть якщо фізично вони знаходяться на одному пристрої. Усі налаштування VLAN поширюються на відповідні порти, які до неї належать.

За замовчуванням більшість комутаторів налаштовані на керування через VLAN 1, до якої автоматично належать усі порти. Проте з міркувань безпеки не рекомендується використовувати VLAN 1 як мережу управління, оскільки вона зазвичай активна за замовчуванням і може бути ціллю для атак. Краще створити окрему VLAN, наприклад VLAN 99, спеціально для цілей адміністрування.

Крок 1. Налаштування інтерфейсу управління.

Щоб підготувати комутатор до віддаленого управління, слід створити VLAN, яка слугуватиме для адміністрування, і налаштувати її інтерфейс. Наприклад:

interface vlan 99

ip address 192.168.99.10 255.255.255.0

no shutdown

Цей набір команд активує віртуальний інтерфейс VLAN 99 і призначає йому IP-адресу. Проте цей інтерфейс не набуде стану “up/up” доти, поки VLAN 99 не буде створена і хоча б один порт комутатора не буде приєднаний до цієї VLAN.

Щоб створити VLAN 99 та прив’язати порт, наприклад FastEthernet0/1, до цієї VLAN, використовуються такі команди:

vlan 99

name Admin_VLAN

exit

interface FastEthernet0/1

switchport access vlan 99

Це означає, що до порту FastEthernet0/1 може бути підключений комп’ютер адміністратора, і він зможе керувати комутатором у межах VLAN 99.

Крок 2. Налаштування шлюзу за замовчуванням.

Для забезпечення зв'язку з пристроями за межами локального сегмента, необхідно вказати комутатору, куди направляти трафік. Наприклад, якщо маршрутизатор, який виконує функцію шлюзу, має адресу 192.168.99.1, то на комутаторі треба виконати команду:

ip default-gateway 192.168.99.1

Ця команда визначає, куди надсилати пакети, призначені для інших підмереж. Без цього налаштування комутатор не зможе зв’язатися з мережами за межами VLAN 99.

Після всіх налаштувань доцільно зберегти конфігурацію, щоб вона не зникла після перезавантаження пристрою. Це робиться командою:

copy running-config startup-config

Крок 3. Перевірка конфігурації

Після завершення налаштувань важливо переконатися, що все працює належним чином. Для цього можна використати команду:

show ip interface brief

Вона покаже коротку інформацію про всі інтерфейси комутатора – фізичні й віртуальні – включаючи стан інтерфейсів, IP-адреси та їхню активність. Якщо інтерфейс VLAN 99 перебуває в стані “up/up”, це

свідчить про його готовність до роботи.

Додатково для глибшого аналізу можна застосовувати команду:

show interfaces

Ця команда виводить детальну інформацію про кожен порт, включаючи швидкість, кількість переданих і прийнятих пакетів, помилки, тип з'єднання тощо. Наприклад, якщо певний порт має високий рівень помилок, це може свідчити про проблеми з кабелем або обладнанням.

Правильне налаштування VLAN, SVI та шлюзу за замовчуванням дозволяє ефективно та безпечно керувати комутатором із будь-якої точки мережі. Це особливо важливо в корпоративних мережах, де адміністратори часто працюють із десятками або сотнями комутаторів, які можуть бути розташовані на великій відстані один від одного.

11.6. Проблеми на рівні мережевого доступу

Команда *show interfaces* в операційній системі IOS комутатора або маршрутизатора Cisco є потужним інструментом для діагностики стану мережевих інтерфейсів. За її допомогою можна не лише перевірити поточний стан інтерфейсу, але й виявити типові проблеми в середовищі передачі даних.

Насамперед слід звернути увагу на два ключові параметри виводу цієї команди:

FastEthernet0/1 is up – цей параметр свідчить про те, що фізичний рівень (Layer 1 моделі OSI) функціонує нормально: кабель підключено, інтерфейс активний.

Line protocol is up – це вже характеристика канального рівня (Layer 2). Якщо цей рядок показує "up", це означає, що між пристроями йде обмін службовими повідомленнями, зокрема *keepalive*.

На основі команди *show interfaces* можна діагностувати кілька типових ситуацій.

Інтерфейс увімкнено, але протокол канального рівня вимкнено (*is up, line protocol is down*). Це може свідчити про проблему з інкапсуляцією (наприклад, невірно вказано тип *encapsulation*), або інтерфейс на іншому боці лінії вимкнений чи несправний.

Наприклад, на одному кінці вказано encapsulation PPP, а на іншому – HDLC. Така невідповідність призведе до збою лінка.

Обидва параметри вимкнені (*is down, line protocol is down*): ймовірно, мережевий кабель не підключено, роз'єм пошкоджено, або порт на іншому кінці лінії вимкнений.

Інтерфейс вимкнено вручну (*administratively down*): це означає, що інтерфейс було свідомо деактивовано адміністратором командою shutdown. Його можна активувати командою *no shutdown*.

Не всі проблеми призводять до повного зникнення зв'язку. Часто мережа працює, але з помітним зниженням продуктивності через помилки введення/виводу.

Помилки введення (input errors) – це сумарний лічильник усіх проблем, що виникають під час отримання кадрів:

Карликові кадри (runts) – Ethernet-кадри, розмір яких менше 64 байтів. Зазвичай виникають через несправність мережевої карти або надмірні колізії.

Гігантські кадри (giants) – перевищують допустиму довжину Ethernet-кадру (зазвичай понад 1518 байт). Причини схожі на runts, часто пов'язані з апаратними збоями.

CRC-помилки (CRC errors) – вказують на пошкодження кадрів через електричні завади, погані з'єднання, неправильно підібраний тип кабелю. Якщо таких помилок багато – необхідно перевірити кабель або замінити його.

Overruns, frame errors, ignored – свідчать про втрати даних через перевантаження або помилки на рівні буферизації.

Помилки виводу (output errors) – це лічильник проблем, пов'язаних із передачею кадрів:

Колізії (collisions) – характерні для напівдуплексних мереж, у повнодуплексних бути не повинні. Якщо колізії спостерігаються в повнодуплексному режимі – ймовірно, неправильно налаштовано тип з'єднання.

Пізні колізії (late collisions) – відбуваються після передавання 64 байтів. Найчастіше спричинені занадто довгими кабелями або конфліктом дуплексних режимів між пристроями.

Наприклад, один порт працює в повнодуплексному режимі, інший – у напівдуплексному. Результат – пізні колізії та зниження якості зв'язку.

Для безпечного керування мережевими пристроями, таких як комутатори, протокол SSH (Secure Shell) є рекомендованим стандартом.

На відміну від Telnet, який передає дані, включаючи паролі, у відкритому вигляді, SSH забезпечує шифрування всієї сесії, включаючи ідентифікацію користувача.

SSH використовує TCP-порт 22, тоді як Telnet – порт 23. SSH значно безпечніший і більш прийнятний для реального використання в корпоративних мережах.

Розглянемо налаштування SSH на комутаторі Cisco Catalyst 2960.

Крок 1. Перевірка підтримки SSH.

Перевірте, чи підтримується SSH вашою версією IOS. Для цього введіть команду:

show ip ssh

Якщо команда не розпізнається, ймовірно, у вас встановлено версію IOS без підтримки криптографії. Перевірити це можна також командою *show version* – якщо у назві образу присутнє *k9*, то підтримка криптографії є.

Крок 2. Присвоєння доменного імені.

ip domain-name mynetwork.local

Це потрібно для генерації RSA-ключів.

Крок 3. Генерація ключів RSA та увімкнення SSH.

crypto key generate rsa

Після цього буде запитано розмір ключа. Мінімум – 1024 біти. Для покращення безпеки бажано використовувати 2048 біт.

Щоб увімкнути тільки другу версію SSH (безпечнішу), використовуємо команду:

ip ssh version 2

Крок 4. Створення користувача.

username admin secret StrongPassword123

Це дозволяє локальну аутентифікацію для SSH-сесій.

Крок 5. Налаштування VTY-ліній.

line vty 0 15

transport input ssh

login local

Ці команди обмежують доступ лише по SSH і вимагають локальної аутентифікації. Щоб переконатися, що SSH працює, можна знову використати команду:

show ip ssh

Для підключення до комутатора з ПК можна скористатися SSH-клієнтом, наприклад, PuTTY. У ньому необхідно вказати IP-адресу комутатора, вибрати тип з'єднання SSH і ввести облікові дані, задані на комутаторі.

11.7. Поширені загрози безпеки локальних мереж

Звичайні налаштування безпеки мережевого комутатора не здатні ефективно протистояти складним атакам. Для забезпечення надійного захисту необхідно впроваджувати багаторівневу систему безпеки, яка постійно вдосконалюється та адаптується до нових загроз. Чим краще фахівці з мережевої безпеки обізнані про потенційні загрози та їх наслідки, тим ефективніше вони можуть захистити інфраструктуру. Розглянемо деякі поширені типи атак та методи їх запобігання.

1. Переповнення таблиці MAC-адрес (MAC Flooding або CAM Table Overflow)

Комутатори зберігають MAC-адреси пристроїв у спеціальній таблиці (CAM-таблиці), що дозволяє ефективно маршрутизувати трафік. Атака типу MAC Flooding полягає в надсиланні великої кількості кадрів з підробленими MAC-адресами, що перевищує ємність таблиці. У результаті комутатор переходить у режим широкомовної передачі, надсилаючи трафік на всі порти, що дозволяє зловмиснику перехоплювати дані інших користувачів.

Наприклад, зловмисник запускає скрипт, який генерує тисячі кадрів з фальшивими MAC-адресами, заповнюючи CAM-таблицю комутатора. Після цього комутатор починає розсилати трафік на всі порти, що дозволяє зловмиснику підслуховувати мережеві дані.

Використання функції Port Security дозволяє обмежити кількість MAC-адрес, які можуть бути прив'язані до конкретного порту.

У разі перевищення встановленого ліміту порт може бути автоматично відключений або переведений у режим обмеженої функціональності.

2. Атаки на протокол DHCP: виснаження ресурсів та підміна (DHCP Spoofing)

Протокол DHCP автоматично призначає IP-адреси пристроям у мережі. Атака типу "виснаження ресурсів" полягає в надсиланні великої кількості запитів на отримання IP-адрес, що призводить до вичерпання

пулу доступних адрес на DHCP-сервері. У результаті нові пристрої не можуть отримати IP-адресу та підключитися до мережі.

Атака DHCP Spoofing передбачає розміщення зловмисником фальшивого DHCP-сервера в мережі, який видає неправильні мережеві налаштування, такі як IP-адреса шлюзу або DNS-сервера. Це дозволяє зловмиснику перенаправляти трафік через свій пристрій для подальшого аналізу або модифікації.

Наприклад, зловмисник підключає до мережі пристрій, який імітує DHCP-сервер, і починає відповідати на DHCP-запити швидше за легітимний сервер, надаючи клієнтам неправильні налаштування.

Впровадження функції DHCP Snooping дозволяє комутатору фільтрувати DHCP-пакети, приймаючи їх лише з довірених портів. Це запобігає роботі фальшивих DHCP-серверів у мережі.

3. Використання вразливостей протоколу CDP (Cisco Discovery Protocol)

CDP – це пропрієтарний протокол Cisco, який дозволяє пристроям обмінюватися інформацією про себе. Однак, оскільки CDP передає дані у відкритому вигляді, зловмисники можуть перехоплювати ці повідомлення для збору інформації про мережу, такої як IP-адреси, версії ПЗ та інші параметри.

Наприклад, зловмисник підключається до мережі та використовує інструменти для прослуховування CDP-повідомлень, отримуючи детальну інформацію про мережеву інфраструктуру, що може бути використано для подальших атак.

У цьому випадку рекомендується відключити CDP на портах, де його використання не є необхідним, за допомогою команди *no cdp run* в глобальній конфігурації пристрою.

4. Атаки на Telnet: перебір паролів та відмова в обслуговуванні (DoS)

Telnet – це протокол для віддаленого доступу до пристроїв, який передає дані у відкритому вигляді, що робить його вразливим до атак. Зловмисники можуть використовувати методи перебору паролів (brute-force) для отримання несанкціонованого доступу.

Крім того, вразливості в реалізації Telnet можуть бути використані для проведення DoS-атак, що призводить до недоступності сервісу.

Наприклад, зловмисник запускає скрипт, який автоматично перебирає можливі комбінації логінів і паролів для підключення до пристрою через Telnet. У разі успіху він отримує повний доступ до системи.

У цьому випадку рекомендується використовувати більш безпечні протоколи, такі як SSH, які забезпечують шифрування даних.

Також слід впроваджувати політику використання складних паролів, обмежувати кількість спроб входу та використовувати списки контролю доступу ACL для обмеження доступу до Telnet.

Розглянемо загальні рекомендації з безпеки локальної мережі.

1. *Розробка політики безпеки*

Створіть чітку політику безпеки, яка визначає правила доступу, обов'язки персоналу та процедури реагування на інциденти.

2. *Використання надійних паролів*

Забезпечте використання складних паролів, які регулярно змінюються, для всіх облікових записів.

3. *Обмеження фізичного доступу*

Забезпечте фізичний захист мережевого обладнання від несанкціонованого доступу.

4. *Відключення невикористовуваних сервісів*

Вимикайте всі непотрібні сервіси та порти на мережевих пристроях.

5. *Використання шифрування*

Застосовуйте шифрування для захисту конфіденційних даних під час передачі та зберігання.

6. *Регулярне оновлення ПЗ*

Своєчасно оновлюйте програмне забезпечення та прошивки мережевих пристроїв для усунення відомих вразливостей.

7. *Навчання персоналу*

Проводьте регулярні тренінги для співробітників з питань інформаційної безпеки та протидії соціальній інженерії.

Дотримання цих рекомендацій допоможе створити надійну систему захисту мережевої інфраструктури та мінімізувати ризики, пов'язані з кіберзагрозами.

11.8. Тестування безпеки локальних мереж

Засоби безпеки відіграють ключову роль у виявленні та усуненні вразливостей у комп'ютерних мережах. Вони дозволяють адміністраторам не лише перевіряти поточний стан безпеки, але й моделювати потенційні атаки для оцінки стійкості системи до загроз.

Аудит безпеки мережі передбачає систематичну перевірку конфігурацій, політик доступу та інших аспектів, що можуть впливати на безпеку. Наприклад, адміністратор може використовувати інструменти для перевірки, які дані можуть бути перехоплені зловмисником при отриманні доступу до мережевого трафіку.

Тестування на проникнення (penetration testing) – це імітація реальних атак з метою виявлення слабких місць у системі безпеки. Це може включати спроби обходу міжмережевих екранів, експлуатацію вразливостей у протоколах або перевірку наявності неправильних конфігурацій. Інструменти для такого тестування часто постачаються з детальною документацією та прикладами команд для проведення різних типів атак.

Співробітники, відповідальні за налаштування та контроль засобів безпеки, повинні мати глибокі знання в таких питаннях:

- мережева безпека;
- міжмережеві екрани;
- системи запобігання вторгненням;
- операційні системи;
- програмування;
- мережеві протоколи (наприклад, TCP/IP).

Ці знання необхідні для ефективного виявлення та усунення вразливостей, а також для правильного налаштування засобів захисту.

Одним із прикладів атак є *MAC Flooding*, коли зловмисник надсилає велику кількість пакетів з підробленими MAC-адресами, перевантажуючи таблицю MAC-адрес комутатора. Це змушує комутатор пересилати трафік на всі порти, що може призвести до витоку конфіденційної інформації.

Для виявлення таких атак адміністратор може використовувати інструменти, які перевантажують таблицю MAC-адрес помилковими записами та аналізують реакцію мережі.

Важливим аспектом є облік часу, оскільки різні комутатори мають різну ємність таблиці MAC-адрес та періоди їхнього застарівання. Неправильне налаштування може призвести до того, що "отруєні" MAC-адреси застаріють під час аудиту, що обмежить ефективність перевірки.

Для підвищення безпеки мережі рекомендується відключати невикористовувані порти комутатора. Наприклад, якщо на комутаторі Catalyst 2960 є 24 порти, а використовується лише 3, слід відключити решту 21 портів за допомогою команди *shutdown*. У разі необхідності порт можна знову активувати командою *no shutdown*.

Функція безпеки порту (Port Security) дозволяє обмежити кількість допустимих MAC-адрес на одному порту та визначити, які пристрої мають право на підключення. Існують різні типи захисту MAC-адрес.

Статичний захист – MAC-адреси вручну налаштовуються на порту та зберігаються в конфігурації.

Динамічний захист – MAC-адреси автоматично вивчаються та зберігаються в таблиці адрес, але видаляються при перезавантаженні комутатора.

Захист на основі прив'язки (Sticky) – динамічно вивчені MAC-адреси автоматично додаються до конфігурації та зберігаються після перезавантаження.

У разі порушення безпеки (наприклад, підключення пристрою з невідомою MAC-адресою) порт може реагувати в один із трьох режимів.

Protect – пакети з невідомими адресами відкидаються без повідомлення.

Restrict – пакети відкидаються, але генерується повідомлення про порушення.

Shutdown – порт вимикається, і його потрібно вручну активувати.

Точна синхронізація часу є критичною для безпеки мережі, оскільки неправильні часові мітки можуть ускладнити аналіз журналів подій та виявлення атак. Протокол NTP (Network Time Protocol) дозволяє синхронізувати час між пристроями в мережі з точністю до мілісекунд. Для цього використовуються NTP-сервери, які можуть бути внутрішніми (локальними) або зовнішніми (в мережі Інтернет).

Щоб налаштувати синхронізацію часу, необхідно вказати IP-адресу NTP-сервера в конфігурації мережевого пристрою. У випадку використання еталонного генератора часу, він повинен бути

синхронізований з всесвітнім координованим часом UTC (Coordinated Universal Time) за допомогою GPS (Global Positioning System) або радіосигналу. Загалом, впровадження та правильне налаштування засобів мережевої безпеки, включаючи аудит, тестування на проникнення, захист портів та синхронізацію часу, є необхідними заходами для забезпечення стійкості мережі до сучасних загроз.

Контрольні запитання та задачі до теми 11

Перелік теоретичних запитань

1. Що таке локальна мережа? Які її основні характеристики?
2. Яка роль комутатора у локальній мережі?
3. Поясніть принцип роботи комутатора рівня 2 моделі OSI.
4. Які основні методи пересилання даних використовуються в комутаторах?
5. Що таке колізійний домен і як він утворюється?
6. Що таке широкомовний (broadcast) домен і як він впливає на продуктивність мережі?
7. Як комутатори зменшують колізійні домени у порівнянні з концентраторними (hub) мережами?
8. Які існують методи зниження перевантаження мережі (congestion)?
9. Які базові налаштування потрібно виконати для нового комутатора Cisco?
10. Для чого використовується VLAN і як вона допомагає у сегментації мережі?
11. Які проблеми можуть виникати на рівні мережевого доступу?
12. Які існують типи атак на рівні локальних мереж (наприклад, MAC-флудинг)?

Перелік задач для самостійного опрацювання

Практичне завдання 1. Налаштування базової локальної мережі.

У середовищі моделювання Cisco Packet Tracer створіть мережу з 2-х комутаторів, 4 ПК. Призначте IP-адреси кожному ПК.

Налаштуйте базову комутацію – перевірте зв’язок між ПК, використовуючи команду *ping*. Використайте команду *show mac address-table* для перегляду MAC-таблиці комутатора.

Практичне завдання 2. Виявлення та усунення проблем.

Створіть мережу з неправильно з'єднаними пристроями або конфліктами IP-адрес.

Використовуйте інструменти Packet Tracer (Simulation mode, команди *ping, ipconfig, show*) для діагностики та усунення проблем.

Тестове завдання 3. Принцип роботи комутатора.

Що відбувається, коли комутатор отримує кадр з MAC-адресою призначення, яка ще не записана в його MAC-таблиці?

а) Комутатор скидає кадр

б) Комутатор передає кадр на всі порти, крім порту, з якого отримав кадр

в) Комутатор пересилає кадр тільки на порт, де виявлено джерело

г) Комутатор направляє кадр до маршрутизатора

Тестове завдання 4. VLAN.

Яке з тверджень про VLAN є правильним?

а) VLAN дозволяє фізично відокремити пристрої один від одного

б) Усі пристрої в одній VLAN завжди перебувають у різних широкомовних доменах

в) VLAN використовується для логічної сегментації мережі

г) VLAN знижує швидкість передачі даних у мережі

Перелік використаної та рекомендованої літератури

11.1. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

11.2. IEEE Standard for Ethernet [Електронний ресурс]. – Режим доступу: https://standards.ieee.org/ieee/802.3/7071/

11.3. IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks. [Електронний ресурс]. – Режим доступу: https://standards.ieee.org/ieee/802.1Q/6844/

11.4. Cisco Packet Tracer [Електронний ресурс]. – Режим доступу: https://www.netacad.com/cisco-packet-tracer

11.5. Technical Guide to Information Security Testing and Assessment [Електронний ресурс]. – Режим доступу: https://csrc.nist.gov/pubs/sp/800/115/final

Тема 12. Проєктування віртуальних локальних мереж

У сучасних комп’ютерних мережах дедалі більшого значення набуває логічна сегментація мережевих ресурсів. Одним із найефективніших засобів досягнення цього є використання віртуальних локальних мереж (VLAN, Virtual Local Area Networks). VLAN дозволяють об'єднувати пристрої, розташовані в різних фізичних сегментах мережі, в єдиний логічний домен, забезпечуючи при цьому підвищений рівень керованості, безпеки й ефективності передачі даних.

Проєктування VLAN є важливим етапом побудови корпоративної мережі, оскільки правильно реалізована сегментація дозволяє мінімізувати широкомовний трафік; зменшити навантаження на мережеву інфраструктуру; підвищити рівень інформаційної безпеки; спростити адміністрування та масштабування мережі.

У цьому розділі розглядаються принципи проєктування VLAN, типові сценарії їх впровадження, правила присвоєння портів комутаторів до VLAN, а також особливості взаємодії між VLAN за допомогою маршрутизаторів або комутаторів третього рівня. Також увага приділяється використанню протоколів, таких як 802.1Q, які забезпечують передачу трафіку VLAN через магістральні канали.

12.1. Призначення і переваги віртуальних локальних мереж

У сучасних об'єднаних мережах, побудованих із використанням комутованої інфраструктури, віртуальні локальні мережі VLAN відіграють ключову роль у логічному структуруванні мережі. Завдяки VLAN адміністратори отримують можливість гнучко сегментувати мережу незалежно від фізичного розташування пристроїв, що забезпечує більш ефективне управління ресурсами, підвищує безпеку й полегшує масштабування.

На відміну від традиційної мережі, де пристрої взаємодіють у межах фізично з'єднаних сегментів, VLAN дозволяє створювати логічні групи пристроїв, які можуть розміщуватися в різних частинах будівлі, офісу або навіть міста. Наприклад, співробітники відділу кадрів, розташовані на різних поверхах офісу, можуть бути об'єднані в одну

VLAN, що дозволить їм обмінюватися даними, ніби вони підключені до одного комутатора (рис. 12.1).

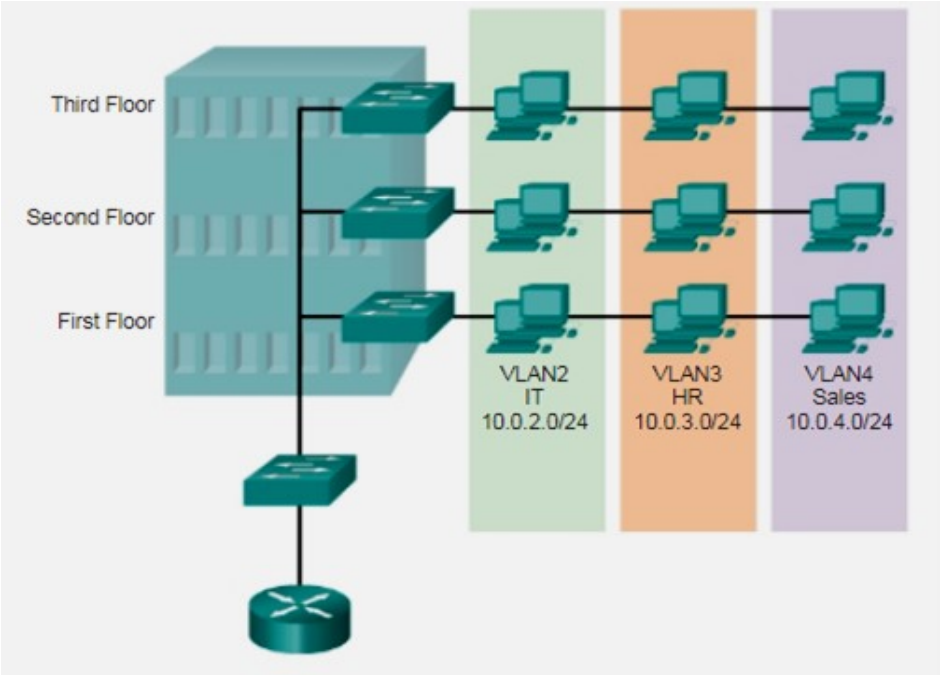


Рис. 12.1. Логічне відокремлення ПК організації до різних VLAN

Принцип дії VLAN ґрунтується на логічному підключенні: кожен порт комутатора може бути призначений певній VLAN, незалежно від фізичної топології. Усі пристрої в межах однієї VLAN сприймають мережу так, ніби вони перебувають в одному широкомовному домені. Це означає, що широкомовні повідомлення (broadcast), одноадресні або багатоадресні пакети передаються лише в межах тієї VLAN, у якій вони були згенеровані, й не потрапляють до інших VLAN.

Для прикладу, якщо комп’ютер бухгалтерії відправляє широкомовний кадр, то цей кадр отримає лише інше обладнання в тій самій VLAN – скажімо, інший комп’ютер бухгалтерії, а не сервер технічного відділу, навіть якщо той підключений до того ж самого комутатора.

Оскільки VLAN створюють логічні межі між сегментами мережі, передача трафіку між різними VLAN можлива лише через спеціальні пристрої, які підтримують маршрутизацію (наприклад, маршрутизатор або комутатор третього рівня). Це дозволяє ефективно контролювати доступ між сегментами, реалізовувати політики безпеки та обмеження

доступу. Кожен порт комутатора може бути членом лише однієї VLAN, за винятком особливих випадків, наприклад, портів, що використовуються для передачі трафіку декількох VLAN (транкові порти), або портів, до яких підключено IP-телефони – у таких випадках порт може обслуговувати як голосовий, так і звичайний трафік.

Розглянемо переваги використання VLAN.

Безпека – VLAN дозволяє ізолювати трафік між логічно розділеними сегментами. Наприклад, трафік фінансового відділу може бути фізично відокремлений від гостьового Wi-Fi, що значно знижує ризик несанкціонованого доступу.

Оптимізація витрат – замість створення окремих фізичних мереж для кожного відділу, можна використовувати одну інфраструктуру з підтримкою кількох VLAN, що зменшує витрати на обладнання та кабелі.

Підвищення продуктивності – VLAN допомагає зменшити навантаження широкомовного трафіку. Коли кожна VLAN є окремим широкомовним доменом, це знижує загальну кількість переданих широкомовних кадрів і підвищує ефективність роботи мережі.

Скорочення широкомовних доменів – розділення великої мережі на менші VLAN дозволяє обмежити зону дії широкомовних кадрів і уникати перевантаження мережі.

Простота управління – коли користувачі з однаковими ролями або потребами з'єднані в одну VLAN, налаштування мережевих політик значно спрощується. Наприклад, при підключенні нового пристрою до певного порту адміністратор просто призначає йому відповідну VLAN, і всі потрібні налаштування автоматично набирають чинності.

Гнучкість для проєктів – VLAN дозволяє організовувати мережу відповідно до бізнес-логіки: наприклад, проєктна група може бути об'єднана в одну VLAN незалежно від того, в якому офісі або місті знаходяться її учасники.

Отже, використання VLAN не тільки спрощує логічну організацію мережі, а й дозволяє адаптувати її до структурних, адміністративних і безпекових вимог будь-якої організації.

Проєктування VLAN – це стратегічний етап створення масштабованої та безпечної мережевої інфраструктури, що відповідає сучасним вимогам бізнесу.

12.2. Типи віртуальних локальних мереж

Віртуальна локальна мережа, призначена для передачі користувацького трафіку (тобто звичайних даних – інтернет, файли, програми), називається VLAN для даних (Data VLAN). Вона відрізняється від VLAN, які використовуються для передачі голосового трафіку (наприклад, IP-телефонія) або керування мережевими пристроями (наприклад, адміністрування комутаторів). VLAN для даних іноді називають *користувацькою VLAN*, оскільки вона обслуговує звичайних користувачів мережі. Наприклад, у навчальному закладі можна створити окрему VLAN для студентів, іншу – для викладачів, і ще одну – для адміністративного персоналу.

Коли комутатор запускається вперше, усі його порти автоматично належать VLAN за замовчуванням (Default VLAN), яка в обладнанні Cisco позначається як VLAN 1 (рис. 12.2). Це означає, що всі підключені до цих портів пристрої належать до одного широкомовного домену, тому вони можуть без обмежень обмінюватися даними.

```
Switch# show vlan brief

VLAN Name                Status    Ports
---- ------------------- --------- -----------------------
1    default             active    Fa0/1,  Fa0/2,  Fa0/3,  Fa0/4
                                   Fa0/5,  Fa0/6,  Fa0/7,  Fa0/8
                                   Fa0/9,  Fa0/10, Fa0/11, Fa0/12
                                   Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                   Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                   Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                   Gi0/1,  Gi0/2
1002 fddi-default        act/unsup
1003 token-ring-default  act/unsup
1004 fddinet-default     act/unsup
1005 trnet-default       act/unsup
```

Рис. 12.2. VLAN за замовчуванням (VLAN 1)

Наприклад, якщо ми підключимо комп’ютери до портів нового комутатора без додаткової конфігурації, всі вони зможуть бачити одне одного в мережі, тому що знаходяться в одній VLAN.

Однак VLAN 1 не можна перейменувати або видалити, і з міркувань безпеки не рекомендується використовувати її для передачі критичних даних або адміністрування.

Native VLAN – це VLAN, призначена транковому порту, через який проходить трафік кількох VLAN.

Транкові порти (trunk ports) використовуються для з'єднання комутаторів між собою або з іншими пристроями, які підтримують VLAN, і дозволяють передавати трафік з багатьох VLAN одночасно. За замовчуванням, усі нетеговані Ethernet-кадри (які не містять тегу VLAN) передаються через native VLAN, якою зазвичай є VLAN 1. Наприклад, якщо транковий порт отримує кадр без тегу, він вважає, що цей кадр належить до native VLAN.

Згідно зі стандартом IEEE 802.1Q, native VLAN забезпечує зворотну сумісність із застарілими пристроями, які не підтримують тегування. Проте з міркувань безпеки рекомендується створити окрему невикористовувану VLAN і призначити її як native VLAN на всіх транкових портах, щоб уникнути несанкціонованого доступу або ін'єкції трафіку в основну VLAN.

Управляюча VLAN (Management VLAN) – це VLAN, через яку здійснюється доступ до функцій управління комутатором або іншим мережевим пристроєм. Наприклад, адміністратор може віддалено підключитися до комутатора через інтерфейси Telnet, SSH або вебінтерфейс за допомогою IP-адреси, призначеної інтерфейсу SVI цієї VLAN. За замовчуванням це VLAN 1, але з метою підвищення безпеки доцільно створити окрему VLAN для управління, наприклад VLAN 10, і дозволяти до неї доступ лише з певних IP-адрес або портів.

Якщо управляюча VLAN збігається з native VLAN, це створює потенційні вразливості безпеки, адже управлінський трафік може бути випадково або зловмисно перехоплений з інших сегментів мережі.

Голосова VLAN – це окрема VLAN, спеціально призначена для передачі VoIP-трафіку (Voice over IP). Такий трафік пред’являє особливі вимоги до мережі: низька затримка (менше 150 мс), мінімальна втрата пакетів, гарантована пропускна здатність та пріоритет обробки. Наприклад, якщо в офісі використовується IP-телефонія, кожен IP-телефон може бути підключений до комутатора через порт, який одночасно обслуговує як голосову VLAN (для передачі голосу), так і VLAN для даних (для підключення комп’ютера через той самий порт).

Для реалізації цього потрібна правильна конфігурація QoS (якість обслуговування), щоб голосовий трафік мав вищий пріоритет порівняно з трафіком звичайного інтернет-серфінгу чи відео.

Транкові з'єднання – це кабельні канали типу "точка-точка", що використовуються для з’єднання мережевих пристроїв і передачі трафіку з кількох VLAN. Вони є необхідними, якщо в мережі використовується більше однієї VLAN, і потрібно, щоб пристрої в одній VLAN, але підключені до різних комутаторів, могли взаємодіяти без маршрутизатора.

Наприклад, якщо в мережі є VLAN 10 (відділ кадрів), VLAN 20 (відділ IT) і VLAN 30 (відділ продажів), а комутатори розміщені на різних поверхах будівлі – потрібно налаштувати транкові порти між цими комутаторами, щоб розширити ці VLAN на всі поверхи.

Транкові з'єднання працюють за стандартом IEEE 802.1Q, який дозволяє позначати (тегувати) кадри, що передаються, і вказувати, до якої VLAN вони належать. Транк не є частиною жодної конкретної VLAN – він просто передає трафік усіх VLAN, які мають бути доступні з обох боків каналу.

12.3. Тегування кадрів Ethernet для ідентифікації мережі VLAN

Стандартний кадр Ethernet, згідно з початковими специфікаціями, не містить у своєму заголовку жодної інформації про VLAN, до якої належить. Це означає, що в локальній мережі без VLAN всі пристрої працюють у межах одного широкомовного домена, і немає можливості логічно розділити трафік. Проте в мережах, де використовується VLAN і кадри передаються між комутаторами через транкові порти (тобто через канали, які обслуговують декілька VLAN одночасно), виникає потреба позначити, до якої VLAN належить кожен кадр.

Цю задачу вирішує тегування – процес додавання спеціальної позначки до кадру Ethernet. Така позначка дозволяє мережевим пристроям коректно маршрутизувати трафік відповідно до VLAN.

Для реалізації тегування застосовується стандарт IEEE 802.1Q. У рамках цього стандарту до кадру Ethernet додається тег розміром 4 байти, який вбудовується між полем MAC-адреси отримувача та полем

типу/довжини кадру. Тег містить критично важливу інформацію, що дозволяє ідентифікувати VLAN, а також налаштувати пріоритет кадру для якості обслуговування (QoS).

Наприклад, якщо комп'ютер у VLAN 10 надсилає кадр до іншого комп'ютера в тій самій VLAN, але на іншому комутаторі, то цей кадр має пройти через транковий канал. У такому разі комутатор додає до нього тег 802.1Q з ідентифікатором VLAN 10.

Тег VLAN за стандартом IEEE 802.1Q складається з кількох полів.

TPID (Tag Protocol Identifier) – це 2-байтове поле, що вказує, що кадр має тег VLAN. Для Ethernet це значення зазвичай 0x8100 у шістнадцятковій формі.

Пріоритет користувача (User Priority) – 3-бітове поле, яке визначає рівень пріоритету кадру. Це дозволяє системі надавати перевагу важливим типам трафіку, наприклад, голосовому або відео.

CFI (Canonical Format Indicator) – 1-бітове поле, яке використовується для визначення формату MAC-адрес і сумісності між різними типами мереж. Зараз воно в основному зарезервоване.

VID (VLAN Identifier) – це 12-бітовий ідентифікатор VLAN, який може приймати значення від 1 до 4094 (оскільки 0 і 4095 зарезервовані). Це дозволяє підтримувати до 4094 різних VLAN у мережі.

Коли кадр надходить на порт комутатора, який налаштовано в режимі доступу (*access mode*) і призначено певній VLAN, комутатор не додає тег у заголовок кадру. Але якщо такий кадр потрібно переслати через транковий порт, комутатор додає до нього тег VLAN, перераховує контрольну суму FCS (Frame Check Sequence) та відправляє кадр далі. Це гарантує, що наступний комутатор або мережевий пристрій правильно розпізнає VLAN, до якої належить кадр.

Деякі пристрої можуть надсилати нетеговані кадри навіть у транкових каналах. Наприклад, старіші пристрої або пристрої без підтримки VLAN можуть не додавати тег до кадру. Щоб забезпечити зворотну сумісність, у стандарті 802.1Q передбачено поняття *native VLAN* – це VLAN, до якої автоматично належать усі нетеговані кадри, що надходять у транковий порт.

За замовчуванням в комутаторах Cisco це VLAN 1. Наприклад, якщо порт Fa0/24 налаштовано як транковий і на нього надходить нетегований кадр, то цей кадр вважається частиною native VLAN –

тобто VLAN 1. Якщо у мережі VLAN 1 не використовується або на ній немає активних пристроїв, то такий кадр просто відкидається.

У деяких випадках мережеві пристрої можуть додавати тег навіть для трафіку *native VLAN*, хоча він зазвичай має бути нетегованим. Якщо транковий порт отримує тегований кадр, що належить до тієї ж VLAN, що і native VLAN, то комутатор Cisco та інші пристрої можуть відхилити цей кадр, вважаючи його помилковим. Це захисний механізм для запобігання конфліктам або потенційним атакам.

Щоб уникнути таких ситуацій і зменшити ризики безпеки, рекомендується створювати окрему, «невикористовувану» VLAN, наприклад VLAN 999, і призначати її як *native VLAN* на всіх транкових портах. Таке рішення дозволяє ізолювати нетегований трафік і чітко визначати, з яких VLAN надходять кадри.

Кожен порт комутатора має параметр *PVID (Port VLAN ID)* – це номер VLAN, до якої належить нетегований трафік, що надходить у порт. У транкових портах значення PVID зазвичай відповідає номеру native VLAN. Таким чином, комутатор використовує PVID для визначення, до якої VLAN належить нетегований кадр, і обробляє його відповідно.

12.4. Діапазони VLAN на комутаторах

Комутатори, залежно від моделі, підтримують різну кількість віртуальних локальних мереж VLAN. Такі базові моделі, як Catalyst 2960, забезпечують підтримку достатньої кількості VLAN для задоволення потреб більшості організацій, зокрема підприємств малого й середнього масштабу.

Наприклад, як комутатори серії Catalyst 2960, так і більш потужні моделі, такі як Catalyst 3560, можуть працювати з *понад 4 000 VLAN*, що дозволяє створити велику кількість логічно ізольованих сегментів мережі для різних підрозділів, сервісів або клієнтів.

Всі можливі VLAN на комутаторах Cisco поділяються на два основні діапазони: *стандартний (normal range)* та *розширений (extended range)*. Цей поділ ґрунтується на ідентифікаторах VLAN (VID), які можуть набувати значень від 1 до 4094, що обумовлено 12-бітним полем VLAN ID у заголовку тегу IEEE 802.1Q (максимум 2^{12}=4096, з яких деякі

зарезервовані).

Стандартний діапазон VLAN охоплює ідентифікатори від 1 до 1005 і використовується найчастіше в локальних мережах компаній.

Цей діапазон підходить для більшості малих і середніх організацій, повністю підтримується протоколом VTP (VLAN Trunking Protocol), який дозволяє централізовано керувати VLAN на кількох комутаторах.

Конфігураційні дані таких VLAN зберігаються у спеціальному файлі *vlan.dat*, який розміщується у флеш-пам'яті комутатора.

VLAN з ідентифікаторами 1002–1005 зарезервовані для застарілих технологій, таких як Token Ring і FDDI, і не можуть бути змінені або видалені.

VLAN 1 створюється автоматично і використовується за замовчуванням для деяких службових процесів (наприклад, управління CDP, VTP тощо).

Наприклад, якщо є компанія з чотирма відділами – бухгалтерія, кадри, IT і служба підтримки, – то можна створити VLAN 10, 20, 30 і 40 для кожного з них відповідно. Всі ці VLAN будуть належати до стандартного діапазону.

Розширений діапазон VLAN охоплює ідентифікатори від 1006 до 4094. Ці VLAN використовуються в більш складних або масштабних мережах, зокрема:

У мережах операторів зв'язку або великих підприємств, яким потрібно обслуговувати сотні або навіть тисячі підрозділів або клієнтів.

У мережах із приватною MPLS-інфраструктурою (Multiprotocol Label Switching), де ізоляція VLAN для кожного клієнта є критичною.

Проте VLAN цього діапазону мають обмеження: їх не підтримує протокол VTP, тому керування такими VLAN відбувається вручну або через централізовані системи налаштування, також конфігурація VLAN не записується у vlan.dat, а зберігається безпосередньо у файлі поточної конфігурації (*running-config*), і при збереженні – у *startup-config*.

Деякі старіші функції, доступні для VLAN стандартного діапазону, можуть бути недоступні або обмежені для VLAN з розширеного діапазону.

Наприклад, у великій телеком-компанії, яка обслуговує понад 3 000 клієнтів, кожен клієнт може отримати власну VLAN з унікальним

ідентифікатором із розширеного діапазону, наприклад, VLAN 2001, VLAN 3050 тощо.

Максимальна теоретична кількість VLAN, яку може підтримувати обладнання Cisco відповідно до стандарту IEEE 802.1Q, становить 4096. Проте в дійсності доступними для використання є 4094 ідентифікатори, оскільки значення 0 і 4095 зарезервовані для службових цілей (наприклад, кадри без тегу або з помилковим тегом).

12.5. Створення віртуальної локальної мережі

Коли ми створюємо віртуальні локальні мережі VLAN стандартного діапазону на комутаторі Cisco, інформація про ці VLAN автоматично зберігається у спеціальному файлі з назвою *vlan.dat*, який знаходиться у флеш-пам'яті пристрою. Ця пам'ять є постійною (non-volatile), отже, навіть після перезавантаження комутатора дані про створені VLAN не будуть втрачені. Завдяки цьому не обов'язково вручну виконувати команду *copy running-config startup-config*, щоб зберегти лише VLAN.

Однак, у більшості випадків разом зі створенням VLAN вносяться й інші зміни в конфігурацію комутатора – наприклад, налаштування портів, IP-адрес, протоколів або безпеки.

У таких ситуаціях рекомендується зберігати всю поточну конфігурацію в постійній пам'яті за допомогою команди:

copy running-config startup-config

Це гарантує, що після перезапуску комутатор працюватиме згідно з усіма внесеними налаштуваннями.

Для зручності адміністратору дозволено створювати кілька VLAN одночасно. Це робиться за допомогою команди *vlan* у глобальному режимі конфігурації. Можна вказати як окремі ідентифікатори VLAN, розділені комами, так і діапазон VLAN через дефіс.

Наприклад, щоб створити VLAN 100, 102, 105, 106 і 107, можна використати одну команду:

S1(config)# vlan 100,102,105-107

Це значно спрощує і прискорює процес налаштування у випадку, коли потрібно створити десятки або сотні VLAN.

Після створення VLAN наступним кроком є прив'язка портів комутатора до відповідних VLAN. Кожен порт, що працює в режимі

access (доступу), може належати лише одній VLAN одночасно. Це дозволяє логічно розділяти трафік між підрозділами компанії, навіть якщо фізично вони підключені до одного пристрою.

Винятком є випадок, коли до порту підключено IP-телефон Cisco, що підтримує передачу одночасно двох VLAN: одна VLAN використовується для голосу (voice VLAN), інша – для даних (data VLAN), які передає комп'ютер, підключений до телефону.

Приклад команди для налаштування порту:

S1(config-if)# switchport access vlan 20

S1(config-if)# switchport voice vlan 100

Для налаштування кількох інтерфейсів одночасно застосовується команда *interface range*. Це дуже корисно при конфігурації великої кількості портів, що мають однакові параметри:

S1(config)# interface range fa0/1 - 10

S1(config-if-range)# switchport access vlan 10

Транк (trunk) – це спеціальний тип підключення між двома комутаторами, який дозволяє передавати трафік з декількох VLAN через один фізичний канал. Транкові з'єднання працюють на 2 рівні моделі OSI і необхідні для того, щоб VLAN могли "перетинати межі" комутаторів і забезпечували логічну сегментацію в розподіленій мережі.

За замовчуванням, транк передає трафік всіх VLAN, якщо його конфігурація не обмежена вручну за допомогою дозволеного списку VLAN.

Щоб перевести порт комутатора в транковий режим, використовується команда:

S1(config-if)# switchport mode trunk

Це примусово встановлює порт у режим *trunk* і активує протокол DTP (Dynamic Trunking Protocol), який спробує погодити транковий режим із пристроєм на іншому кінці з'єднання.

Навіть якщо інший пристрій не підтримує або не налаштований на транковий режим, порт все одно ініціює функціонування як транковий, намагаючись передавати трафік VLAN відповідно до локальної конфігурації. Це може викликати конфлікти, тому важливо переконатися, що обидва боки каналу налаштовані однаково.

На прикладі (рис. 12.3) порт F0/1 комутатора S1 налаштовується як транковий.

```
S1(config)# interface FastEthernet0/1
S1(config-if)# switchport mode trunk
S1(config-if)# switchport trunk native vlan 99
S1(config-if)# switchport trunk allowed vlan 10,20,30,99
S1(config-if)# end
```

Рис. 12.3. Приклад налаштування транкового каналу

У цій конфігурації встановлена native VLAN – це VLAN 99 (використовується для нетегованого трафіку).

Транковий порт пропускає трафік лише VLAN 10, 20, 30 і 99, що вказується через обмеження списку допустимих VLAN.

Приклад команди для обмеження VLAN:

S1(config-if)# switchport trunk allowed vlan 10,20,30,99

Щоб перевірити, як налаштований порт, можна використати команду:

S1# show interfaces fa0/1 switchport

У виведенні буде видно, чи порт у режимі *trunk*, яка встановлена native VLAN, які VLAN дозволені для проходження через транк.

Це дозволяє переконатися в правильності налаштування та діагностувати можливі проблеми.

На рис. 12.4 зображено приклад налаштування інтерфейсу FastEthernet 0/1 на комутаторі Cisco як транкового порту, який передає трафік кількох VLAN.

На порті F0/1 комутатора S1 встановлено транковий режим (*switchport mode trunk*) і призначено native VLAN 99 командою *switchport trunk native vlan 99*.

Вивід команди *show interfaces f0/1 switchport* підтверджує, що порт працює в транковому режимі з *encapsulation dot1q*, native VLAN 99 активна і теги для неї дозволено (Administrative Native VLAN tagging: enabled).

На інтерфейсі активовано передачу всіх VLAN, тобто порт сконфігуровано у транковому режимі, який дозволяє пропускати трафік усіх віртуальних локальних мереж тобто будь-який VLAN, налаштований у мережі, потенційно може передаватися через цей порт, якщо інші пристрої мережі також це підтримують.

```
S1(config)# interface f0/1
S1(config-if)# switchport mode trunk
S1(config-if)# switchport trunk native vlan 99
S1(config-if)# end
S1# show interfaces f0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 99 (VLAN0099)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
```

Рис. 12.4. Приклад перевірки налаштування інтерфейсу fa0/1

12.6. Протокол DTP

На комутаторах Cisco, зокрема серій Catalyst 2960 і 3560, інтерфейси Ethernet підтримують різні режими роботи, що визначають їхню участь у створенні транкових з’єднань.

Кожен порт може бути налаштований так, щоб працювати:

- у транковому режимі, коли передається трафік декількох VLAN;
- у звичайному (нетранковому) режимі, коли порт належить одній VLAN;
- у динамічному режимі, коли порт автоматично узгоджує тип з’єднання з портом сусіднього пристрою.

Таке узгодження виконується за допомогою спеціального протоколу – DTP.

DTP (Dynamic Trunking Protocol) – це власний (пропрієтарний) протокол компанії Cisco, який дозволяє двом сумісним пристроям автоматично домовитись, чи буде їх з’єднання транковим. Цей протокол працює лише між Cisco-пристроями, і зазвичай ввімкнений за замовчуванням на інтерфейсах Ethernet комутаторів Catalyst.

Наприклад, коли два комутатори Cisco з’єднані через порти в

динамічному режимі, DTP може автоматично визначити, що трафік між ними має бути транковим, і активує trunk-режим без додаткових налаштувань.

Однак інші пристрої, які не підтримують DTP (наприклад, комутатори інших виробників або маршрутизатори), можуть неправильно обробляти або ігнорувати DTP-кадри, що може призвести до помилок конфігурації.

У випадках, коли порт Cisco-комутатора з'єднується з пристроєм, що не підтримує DTP (наприклад, з сервером, маршрутизатором або комутатором іншого бренду), рекомендується вимкнути DTP вручну. Це допоможе уникнути некоректної взаємодії і небажаних транкових домовленостей.

Для цього використовується команда:

switchport nonegotiate

Ця команда блокує відправлення DTP-кадрів, і ви повинні вручну налаштувати порт у транковий режим або режим доступу.

Приклад повної конфігурації порту для з'єднання з іншим пристроєм у транковому режимі без DTP:

```
interface fa0/1
switchport mode trunk
switchport nonegotiate
```

Розглянемо основні режими DTP на Cisco-портах.

Інтерфейси Ethernet на Catalyst 2960 і 3560 підтримують кілька режимів, які визначають поведінку порту щодо DTP:

switchport mode access

Команда переводить порт у постійний нетранковий режим. Такий порт не буде намагатися встановити транкове з'єднання навіть якщо підключений пристрій працює в *trunk*-режимі.

Використовується для підключення комп'ютера або принтера:

```
interface fa0/2
switchport mode access
switchport mode dynamic auto
```

Порт очікує, що сусідній пристрій запропонує транковий режим. Якщо інший порт буде в режимі *trunk* або *dynamic desirable*, то канал стане транковим.

Це стандартний режим для портів Ethernet за замовчуванням.

switchport mode dynamic desirable

Порт активно ініціює транкове з'єднання, тобто надсилає DTP-запити. Якщо інший порт налаштований як *access*, з'єднання залишиться нетранковим. Якщо інший порт підтримує DTP, з'єднання стане транковим.

switchport mode trunk

Порт негайно переводиться у транковий режим і намагається погодити транкове з'єднання з іншим портом, незалежно від його налаштувань.

Перевагою є забезпечення стабільного транкового з'єднання, навіть якщо інший пристрій не підтримує DTP (у поєднанні з nonegotiate).

Команда *switchport nonegotiate* вимикає протокол DTP для даного інтерфейсу. Цю команду потрібно застосовувати лише разом із ручним налаштуванням режиму (наприклад, *access* або *trunk*), і вона не може бути використана в динамічному режимі.

Щоб перевірити, у якому режимі працює порт і чи активний на ньому DTP, використовується команда:

show dtp interface

Це дозволяє отримати важливу інформацію для діагностики та підтвердження правильності налаштування портів, зокрема, режим порту (*trunk, access, dynamic*), чи відправляє або приймає порт DTP-кадри, стан узгодження з іншим портом.

12.7. Проблематика VLAN

У кожної VLAN повинна бути власна унікальна IP-підмережа. Це означає, що всі пристрої, які знаходяться в одній VLAN, мають належати до однієї і тієї ж IP-підмережі – наприклад, VLAN 10 може використовувати підмережу 192.168.10.0/24. Якщо хоча б один пристрій матиме адресу з іншої підмережі, наприклад 192.168.20.5/24, він не зможе обмінюватися даними з іншими пристроями VLAN 10. Така помилка досить поширена й призводить до втрати зв'язку всередині VLAN. Щоб її виправити, адміністратор має перевірити IP-адреси і маски всіх пристроїв VLAN і внести відповідні зміни, щоб вони відповідали потрібній підмережі.

Якщо IP-адресація вже перевірена й виправлена, але пристрої все одно не можуть обмінюватися даними, потрібно виконати додаткову діагностику. Насамперед варто використати команду *show vlan* на комутаторі, щоб переконатися, що кожен порт призначений до правильної VLAN. Наприклад, якщо комп'ютер підключено до порту Fa0/2, а цей порт знаходиться у VLAN 20 замість очікуваної VLAN 10 – пристрій не зможе нормально функціонувати.

Для зміни VLAN використовується команда *switchport access vlan [номер_VLAN]* у режимі конфігурації інтерфейсу.

Далі слід скористатися командою *show mac address-table*, яка дозволяє перевірити, які MAC-адреси були виявлені на кожному порту та до яких VLAN вони належать. Це допомагає зрозуміти, чи дійсно пристрій працює у мережі. Якщо VLAN, до якої прив'язаний порт, була видалена з конфігурації, то порт автоматично переходить у неактивний стан. У такому разі допоможе команда *show interfaces switchport*, яка покаже статус VLAN порту. Щоб порт знову став активним, потрібно створити VLAN командою *vlan [номер]* у глобальному режимі конфігурації.

Неполадки можуть також виникати при налаштуванні транкових каналів. Іноді порт комутатора поводиться як транковий, навіть якщо він налаштований як *access*. Наприклад, доступний порт Fa0/3 може почати приймати трафік з кількох VLAN, якщо сусідній комутатор помилково надсилає йому транковий трафік. Це свідчить про проблему з некоректно налаштованими транковими портами.

Для діагностики транкових каналів слід насамперед використати команду *show interfaces trunk*, яка показує, чи співпадають налаштування native VLAN з обох боків з'єднання. Якщо на одному кінці використовується VLAN 99 як native VLAN, а на іншому – VLAN 2, виникає конфлікт. У такому випадку трафік VLAN 99 на одному кінці буде прийнятий як VLAN 2 на іншому, що викликає некоректну маршрутизацію даних.

Про таку помилку також може повідомити протокол CDP (Cisco Discovery Protocol). У випадку невідповідності native VLAN на портах Fa0/1 обох комутаторів, у консолі з'явиться повідомлення:

**Mar 1 06:45:26.232: %CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/1 (2), with S2 FastEthernet0/1 (99).

Такі помилки призводять до втрати управління і можуть не тільки порушити з'єднання, а й становити загрозу для безпеки мережі, оскільки адміністративний трафік передається не в той сегмент.

Розглянемо основні помилки конфігурації транкових каналів.

Невідповідність native VLAN: транкові порти мають різні значення native VLAN.

Невірно узгоджені режими портів: наприклад, один порт у режимі trunk, а інший – у *dynamic auto*.

Неправильний список дозволених VLAN: транк не дозволяє VLAN, яка повинна передаватися.

Для перевірки дозволених VLAN на транку знову використовується *show interfaces trunk*. Якщо, наприклад, VLAN 30 не дозволена, хоча повинна бути, то пристрої з VLAN 30 не отримають доступ до інших сегментів мережі. Щоб вручну задати список VLAN, які дозволено передавати через транк, застосовується команда:

switchport trunk allowed vlan 10,20,30.

Безпека VLAN – ще один важливий аспект. Хоча VLAN дозволяють ізолювати трафік і підвищити ефективність мережі, вони не позбавлені ризиків.

Однією з поширених атак є *VLAN Hopping*. Її суть полягає у тому, що зловмисник отримує доступ до трафіку VLAN, яка йому не належить.

Один з видів VLAN Hopping – *атака через спуфінг комутатора*. У цьому випадку зловмисник підключається до порту комутатора, налаштованого в динамічному режимі (*dynamic auto*) і починає імітувати інший комутатор, ініціюючи створення транкового з'єднання за допомогою підроблених DTP або 802.1Q кадрів. У результаті комутатор перетворює порт на транковий і відкриває доступ до всіх VLAN, дозволених на цьому каналі.

Для захисту від такої атаки потрібно вимикати DTP на всіх портах, де транк не потрібен, за допомогою *switchport nonegotiate*, виконувати ручне налаштування транку замість автоматичного (*switchport mode trunk*), встановлювате порти за замовчуванням у режим доступу (*access*).

Інший тип загрози – *атака з подвійним тегуванням (Double-Tagging)*. Тут зловмисник вставляє два 802.1Q теги в кадр. Зовнішній тег – VLAN, яка збігається з native VLAN транкового порту, внутрішній тег – VLAN, яку хоче атакувати зловмисник. Перший комутатор видаляє зовнішній тег (оскільки це native VLAN), але не бачить внутрішній. Кадр без зовнішнього тега передається на наступний комутатор, який вже читає внутрішній тег і пересилає кадр до VLAN жертви. Цей тип атаки особливо небезпечний, бо працює навіть на портах доступу, якщо вони належать до тієї ж VLAN, що й *native VLAN*.

Захист полягає у тому, щоб використовувати окрему, унікальну VLAN як native VLAN, ніколи не призначати користувацьким портам native VLAN, заборонити прийом тегованих кадрів на *access*-портах (наприклад, *switchport access vlan 10, switchport mode access*).

12.8. Рекомендації з проєктування VLAN

VLAN 1 – це VLAN за замовчуванням на комутаторах Cisco, яка автоматично використовується для всіх портів, якщо інше не вказано. Вона також використовується для керування комутатором, передачі деяких службових протоколів (наприклад, CDP, VTP) та іншого внутрішнього трафіку.

Через це VLAN 1 є вразливою з точки зору безпеки. Якщо залишити порти в цій VLAN, зловмисник може потенційно перехопити службовий трафік або навіть отримати доступ до налаштувань мережі.

Щоб зменшити такий ризик, усі порти слід вручну перевести в інші VLAN, які використовуються для звичайної роботи.

Найбільш ефективна практика – уникати використання VLAN 1 взагалі для користувацького трафіку. Для цього рекомендується налаштувати всі невикористані порти на спеціальну "мертву" VLAN (іноді її називають "black hole VLAN" або VLAN-«чорна діра»), яка не має виходу до жодної з активних частин мережі. Наприклад, можна створити VLAN 999 і перевести туди всі порти, які не задіяні:

```
Switch(config)# vlan 999
Switch(config)# interface range fa0/10 - 24
Switch(config-if-range)# switchport mode access
```

Switch(config-if-range)# switchport access vlan 999

Switch(config-if-range)# shutdown

Таким чином, навіть якщо до такого порту хтось підключиться, він не отримає доступу до мережі.

Додатково рекомендується вимикати невикористані порти (*shutdown*), що повністю запобігає підключенню до них.

Ще один важливий захід безпеки – відокремлення адміністративного трафіку від користувацького. Адміністративний трафік – це трафік, пов’язаний з управлінням мережевим обладнанням: доступ до CLI, вебінтерфейсу, SNMP, логування тощо. За замовчуванням цей трафік також передається через VLAN 1. Щоб уникнути потенційного доступу до цих даних зловмисниками, потрібно створити окрему VLAN, наприклад VLAN 100, і призначити її як керуючу (management VLAN). Комутатору необхідно надати IP-адресу саме в цій VLAN, наприклад:

Switch(config)# interface vlan 100

Switch(config-if)# ip address 192.168.100.2 255.255.255.0

Switch(config-if)# no shutdown

У такому випадку користувачі з інших VLAN зможуть підключитися до комутатора лише через маршрутизований шлях, і тільки якщо дозволено в міжвлановому фаєрволі або ACL.

Для ще більшої безпеки рекомендується налаштувати комутатор так, щоб він приймав лише захищені сеанси віддаленого управління, наприклад через SSH. Це забезпечує шифрування трафіку між адміністратором і пристроєм, що унеможливлює його перехоплення. Telnet, навпаки, передає дані відкритим текстом, тому його варто повністю відключити. Для налаштування SSH потрібно створити ключі шифрування, встановити ім’я користувача та увімкнути службу SSH:

Switch(config)# ip domain-name mynetwork.local

Switch(config)# crypto key generate rsa

Switch(config)# username admin secret MySecurePassword

Switch(config)# line vty 0 4

Switch(config-line)# transport input ssh

Switch(config-line)# login local

Окрему увагу варто приділити протоколу DTP (Dynamic Trunking Protocol), який автоматично узгоджує транкові з'єднання між

комутаторами. DTP має кілька режимів:

access – порт завжди працює як порт доступу (в одній VLAN),

trunk – порт завжди працює як транковий,

dynamic auto – очікує ініціативи від іншого порту,

dynamic desirable – активно намагається створити транкове з'єднання.

Автоузгодження може призвести до помилок або навіть атак, таких як VLAN Hopping. Щоб цього уникнути, рекомендується явно вказувати режим порту і вимикати DTP:

Switch(config-if)# switchport mode access

Switch(config-if)# switchport nonegotiate

Це гарантує, що порт не стане транковим навіть у разі помилкової взаємодії з іншим пристроєм.

Голосовий трафік, такий як трафік з IP-телефонів, вимагає особливої обробки через вимоги до затримки, втрат і пріоритету (QoS). Якщо комп’ютери користувачів та IP-телефони підключені до однієї VLAN, може виникнути ситуація, коли обидва пристрої намагаються використовувати всю доступну смугу пропускання, і це призводить до затримок у передачі голосу або обривів дзвінків. Щоб цього уникнути, рекомендується для IP-телефонів створити окрему голосову VLAN, наприклад VLAN 110. Це дозволяє комутаторам пріоритезувати голосовий трафік за допомогою QoS.

Порт у цьому випадку можна налаштувати так:

Switch(config-if)# switchport mode access

Switch(config-if)# switchport access vlan 10

Switch(config-if)# switchport voice vlan 110

Тут VLAN 10 використовується для комп’ютера користувача, а VLAN 110 – для IP-телефону, який підключений до цього ж порту, але через IP-телефон.

Отже, VLAN 1 не слід використовувати для звичайного або адміністративного трафіку; невикористані порти варто вимикати і переводити у «мертву» VLAN; адміністративну VLAN треба винести в окрему підмережу; управління комутатором має відбуватися лише через SSH; DTP слід вимикати на всіх access-портах; голосовий і користувацький трафік мають бути розділені в різні VLAN для QoS.

Контрольні запитання та задачі до теми 12

Перелік теоретичних запитань

1. Що таке віртуальна локальна мережа VLAN? Яке її основне призначення?
2. Які переваги використання VLAN у корпоративних мережах?
3. У чому полягає принцип логічної сегментації мережі через VLAN?
4. Які типи VLAN існують? Наведіть приклади для кожного типу.
5. Яка роль керованої VLAN (Management VLAN) у мережевій інфраструктурі?
6. Що таке голосова VLAN (Voice VLAN) і чому її доцільно виділяти окремо?
7. Як відбувається тегування кадрів Ethernet у VLAN (802.1Q)?
8. Які обмеження існують для ідентифікаторів VLAN? Які діапазони VLAN підтримуються Cisco-комутаторами?
9. Які відмінності між стандартними, розширеними та зарезервованими VLAN?
10. Як створити VLAN на комутаторі Cisco? Які команди використовуються?
11. У чому полягає призначення команд *switchport mode access* і *switchport access vlan*?
12. Як працює протокол DTP (Dynamic Trunking Protocol)? Які режими він підтримує?
13. Які типові проблеми можуть виникнути при роботі VLAN?
14. Як перевірити конфігурацію VLAN і транкових портів на комутаторі?
15. Які рекомендації з проєктування VLAN слід враховувати для побудови ефективної та безпечної мережі?

Перелік задач для самостійного опрацювання

Практичне завдання 1. Базове створення VLAN.

У середовищі моделювання Cisco Packet Tracer створіть топологію мережі з двома комутаторами та чотирма ПК.

Налаштуйте VLAN 10 та VLAN 20, прив’яжіть по два ПК до кожної VLAN, перевірте комунікацію в межах однієї VLAN за допомогою команди *ping*.

Практичне завдання 2. Налаштування транкового з'єднання.

Додайте третій комутатор і з’єднайте всі три між собою транковими лінками. Налаштуйте *trunk*-порти вручну, використовуючи команди *switchport mode trunk, switchport trunk allowed vlan*. Упевніться, що трафік VLAN 10 і 20 проходить між усіма пристроями.

Практичне завдання 3. Управління VLAN та безпека.

Створіть VLAN 99 як Management VLAN. Вимкніть DTP на всіх портах, окрім тих, що використовуються для транків (використовуючи команду *switchport nonegotiate*). Відключіть всі невживані порти.

Тестове завдання 4.

Яка VLAN використовується за замовчуванням на більшості комутаторів Cisco?

а) VLAN 0

б) VLAN 10

в) VLAN 1

г) VLAN 99

Тестове завдання 5.

Який діапазон зарезервованих VLAN?

а) 1–1000

б) 1002–1005

в) 1006–4094

г) 0–15

Тестове завдання 6.

Яка команда використовується для перевірки активних VLAN на комутаторі?

а) show interface vlan

б) show ip vlan

в) show vlan brief

г) vlan status

Перелік використаної та рекомендованої літератури

12.1. IEEE 802.1Q [Електронний ресурс]. – Режим доступу: https://standards.ieee.org/ieee/802.1Q/6844/

12.2. Що таке VLAN і як налаштувати його на бездротовому маршрутизаторі ASUS [Електронний ресурс]. – Режим доступу: https://www.asus.com/ua-ua/support/faq/1049415/

Тема 13. Технології маршрутизації. Маршрутизація VLAN

У комп'ютерних мережах маршрутизація є ключовим процесом, який забезпечує доставку даних між різними мережами. Вона дозволяє пакетам, що надходять з однієї IP-мережі, досягати призначення в іншій, навіть якщо ці мережі фізично віддалені або належать до різних логічних доменів. Без ефективної маршрутизації неможливе функціонування глобальної мережі Інтернет, а також великої кількості корпоративних і локальних мереж.

Технології маршрутизації охоплюють набір протоколів, методів і пристроїв, які визначають, як саме трафік переміщується мережею. Вони включають як статичну маршрутизацію, де маршрути задаються вручну, так і динамічну маршрутизацію, яка дозволяє маршрутизаторам автоматично обмінюватися інформацією про шляхи до мереж. Серед протоколів динамічної маршрутизації найпоширенішими є RIP, OSPF, EIGRP та BGP. Кожна технологія маршрутизації має свої особливості, переваги й обмеження. Наприклад, статична маршрутизація проста в налаштуванні, але не підходить для великих або динамічних мереж. Натомість протоколи на кшталт OSPF забезпечують адаптивність і високу масштабованість, що є критично важливим у складних корпоративних середовищах. Сучасні мережі вимагають гнучкої та надійної маршрутизації, яка може адаптуватися до змін у топології, балансувати навантаження та забезпечувати якісне обслуговування (QoS). Тому вивчення технологій маршрутизації є важливим етапом для фахівців у галузі мережевих технологій.

13.1. Призначення маршрутизації

Комутатори Ethernet працюють на другому рівні моделі OSI – канальному рівні. Їх основне завдання полягає в пересиланні кадрів Ethernet між пристроями в межах однієї локальної мережі LAN. Наприклад, коли комп'ютер надсилає файл на локальний принтер або передає дані серверу в тій самій підмережі, комутатор визначає MAC-адресу призначення й пересилає кадр на відповідний порт. Проте, коли джерело та пункт призначення перебувають у різних IP-мережах або VLAN, комутатор не може самостійно передати ці дані – у такому випадку кадр потрібно передати на маршрутизатор.

Маршрутизатор функціонує на третьому рівні моделі OSI – мережевому. Його призначення – з'єднувати різні мережі та забезпечувати пересилання IP-пакетів між ними. Наприклад, якщо комп'ютер у київському офісі надсилає запит на вебсервер, що знаходиться в Німеччині, цей запит пройде через маршрутизатори різних провайдерів, перш ніж дістатися до сервера. Так само, навіть у межах великої корпоративної мережі, маршрутизатор може використовуватися для передавання трафіку між відокремленими сегментами, наприклад, від мережі бухгалтерії до мережі серверів.

Щоб знайти оптимальний маршрут для доставки пакету, маршрутизатор використовує таблицю маршрутизації (рис. 13.1).

```
IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0         10.0.0.1        10.0.0.32     35
         10.0.0.0    255.255.255.0         On-link         10.0.0.32    291
        10.0.0.32  255.255.255.255         On-link         10.0.0.32    291
       10.0.0.255  255.255.255.255         On-link         10.0.0.32    291
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link         10.0.0.32    291
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link         10.0.0.32    291
===========================================================================
Persistent Routes:
  None

IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
  1    331 ::1/128                  On-link
```

Рис. 13.1. Приклад фрагменту таблиці маршрутизації

У цій таблиці зберігається інформація про напрямки до різних мереж і відповідні інтерфейси. Наприклад, якщо маршрутизатор знає, що мережа 192.168.10.0/24 доступна через інтерфейс GigabitEthernet0/1, то всі пакети, призначені для цієї мережі, він направить саме туди. Оптимізація маршрутів дозволяє зменшити затримки, уникати перевантажених сегментів і забезпечувати швидку доставку даних.

Якщо комп’ютер (вузол) намагається надіслати пакет в іншу IP-мережу, він не має змоги зв’язатися з призначенням безпосередньо. У цьому випадку пакет передається на шлюз за замовчуванням – зазвичай це IP-адреса інтерфейсу маршрутизатора, підключеного до тієї ж підмережі. Наприклад, якщо комп’ютер із IP 192.168.1.10 має шлюз за замовчуванням 192.168.1.1, то весь трафік, призначений для зовнішніх мереж (наприклад, Інтернет), буде спрямований на цей шлюз. Такий механізм дозволяє ізолювати внутрішні маршрути і централізовано контролювати зовнішні з’єднання.

Для забезпечення ефективної маршрутизації важливо враховувати кілька ключових характеристик мережі. Однією з них є топологія. Вона може бути фізичною або логічною. Фізична топологія відображає реальне розміщення кабелів, пристроїв та портів (наприклад, «зірка» або «кільце»), тоді як логічна топологія описує, як дані фактично рухаються мережею, незалежно від фізичного з’єднання – наприклад, логічна шина в мережі Ethernet.

Ще одним критичним фактором є швидкість передавання даних, що визначається як обсяг інформації, який може бути переданий за секунду (наприклад, 100 Мбіт/с або 1 Гбіт/с). Від швидкості залежить продуктивність мережі, особливо при передаванні великих обсягів даних чи в роботі з відеоконференціями, стримінгом або базами даних.

Вартість – це ще один аспект проєктування мережі, який включає в себе ціну на обладнання (маршрутизатори, комутатори, кабелі), витрати на інсталяцію, обслуговування, навчання персоналу тощо. Наприклад, побудова мережі на основі високопродуктивних маршрутизаторів Cisco коштує більше, ніж на недорогих пристроях, проте може забезпечити вищу надійність і розширюваність.

Безпека є невід’ємною складовою сучасних мереж. Йдеться як про захист від несанкціонованого доступу, так і про запобігання витоку або втраті даних. Прикладом засобів безпеки є фільтрація трафіку за IP-адресами, використання міжмережевих екранів, шифрування, наприклад, VPN (Virtual Private Network) або контроль доступу за допомогою списків ACL.

Доступність означає, що мережа повинна бути доступна для користувачів тоді, коли вона їм потрібна. Наприклад, віддалений офіс має мати стабільне з'єднання з головним офісом без частих збоїв. Для

досягнення цього можуть використовуватись резервування каналів зв'язку, відмовостійкі маршрутизатори, джерела безперебійного живлення UPS тощо.

Масштабованість визначає здатність мережі зростати – тобто додавати нових користувачів, пристрої або підмережі без потреби в повній перебудові. Якщо мережа спроєктована з урахуванням зростання (наприклад, використання підмереж чи модульної структури), то її розширення буде менш затратним і простішим.

Надійність описує стабільність роботи мережевих пристроїв – комутаторів, маршрутизаторів, серверів тощо. Високий показник середнього часу безвідмовної роботи MTBF (mean time between failure) свідчить про те, що обладнання не схильне до частих збоїв, що дуже важливо в критичних інфраструктурах (медицина, фінанси, енергетика). Наприклад, сервер з MTBF у 200 000 годин зазвичай працює без збоїв протягом багатьох років.

13.2. Основні функції маршрутизаторів

Маршрутизатор – це не просто «чорна коробка», яка передає дані з одного сегмента мережі в інший. Це повноцінний спеціалізований комп'ютер, який має центральний процесор (ЦП), оперативну та постійну пам'ять. Як і в звичайному комп'ютері, маршрутизатор виконує операції відповідно до інструкцій операційної системи (ОС). Ці інструкції охоплюють запуск пристрою, основні діагностичні процедури, а також функції маршрутизації та пересилання даних. Наприклад, під час вмикання маршрутизатора активується програма ініціалізації, яка перевіряє його працездатність і запускає процес завантаження ОС.

Для обробки й зберігання даних маршрутизатори використовують кілька типів пам'яті, кожна з яких має своє призначення.

Оперативна пам'ять забезпечує тимчасове збереження інформації, необхідної для поточної роботи пристрою. Тут зберігається активна версія ОС (наприклад, Cisco IOS), конфігураційні файли, таблиці маршрутизації, ARP-таблиці, а також буфери обробки трафіку. Наприклад, якщо маршрутизатор обробляє вхідний пакет, він тимчасово розміщує його в буфері ОЗП перед прийняттям рішення про

пересилання. Варто пам’ятати, що ОЗП – це енергозалежна пам’ять: усі дані зникають після вимкнення живлення.

Постійна пам’ять містить базові інструкції, необхідні для початкового завантаження пристрою. У ПЗП, наприклад, може зберігатися «урізана» версія IOS, яку використовують у випадках, коли основна ОС недоступна. Цей тип пам’яті схожий на BIOS у звичайному ПК. Вона не змінюється й зберігає вміст навіть після вимкнення живлення.

NVRAM (Non Volatile Random Access Memory) – незалежна оперативна пам’ять використовується для постійного збереження конфігураційного файлу, з якого пристрій завантажується після перезавантаження – так званого файлу *startup-config*. Наприклад, якщо ми налаштували IP-адреси або маршрути вручну, вони зберігаються саме в NVRAM.

Флеш-пам’ять виконує роль «жорсткого диска» маршрутизатора. У ній зберігається повнофункціональна операційна система IOS, а також інші файли, необхідні для роботи. Після запуску маршрутизатора ця ОС копіюється у ОЗП для швидкого доступу. Флеш-пам’ять є незалежною, тобто не втрачає дані після відключення живлення. Наприклад, якщо у нас кілька версій IOS, ви можете перемикатися між ними, використовуючи флеш-пам’ять.

Маршрутизатори відрізняються від звичайних ПК не тільки своєю функціональнісю, але й обладнанням. Вони не мають відео- чи звукових карт, оскільки не призначені для роботи з мультимедіа. Замість цього маршрутизатори оснащені спеціальними мережевими інтерфейсами (LAN, WAN), які забезпечують підключення до інших мереж. Наприклад, один інтерфейс може бути підключений до локальної мережі офісу, а інший – до мережі провайдера.

Більшість пересічних користувачів навіть не підозрюють, скільки маршрутизаторів бере участь у передачі їх даних. Відправляючи повідомлення чи відкриваючи вебсторінку, вони не замислюються, що кожен пакет, можливо, проходить через десятки маршрутизаторів, перш ніж дійде до сервера в іншому місті або країні. Саме маршрутизатори забезпечують надійне й ефективне переміщення пакетів між мережами.

Маршрутизатор може мати декілька інтерфейсів, кожен з яких підключений до окремої IP-мережі. Наприклад, маршрутизатор із

трьома інтерфейсами може з’єднувати локальну мережу, гостьову мережу та мережу провайдера. Коли на один з інтерфейсів надходить пакет, маршрутизатор аналізує його IP-адресу призначення, визначає оптимальний маршрут і вирішує, через який інтерфейс його передати.

Інтерфейси маршрутизатора можуть вести як у локальні мережі LAN, так і у глобальні мережі WAN. Наприклад, локальний інтерфейс з'єднує маршрутизатор із внутрішньою мережею компанії, де знаходяться комп’ютери співробітників. Інший інтерфейс може бути підключений до глобальної мережі через оптоволоконне підключення провайдера. Таким чином маршрутизатор стає вузлом між різними типами мереж.

Основні функції маршрутизатора зводяться до двох ключових дій: визначення найкращого шляху для доставки пакетів і фактична пересилка цих пакетів. Для цього використовується таблиця маршрутизації – спеціальна база даних, у якій зберігається інформація про доступні мережі, пов’язані інтерфейси та метрики.

Наприклад, якщо маршрутизатор бачить, що шлях через інтерфейс GigabitEthernet0/0 до певної мережі має меншу затримку, ніж інший маршрут, то він вибирає саме його.

Коли маршрутизатор отримує пакет, він виконує інкапсуляцію – тобто обгортає пакет у новий кадр відповідно до типу інтерфейсу. Наприклад, пристрій може отримати дані в Ethernet-кадрі, а передати їх далі в кадрі PPP (Point-to-Point Protocol), якщо вихідний інтерфейс підключений до WAN-лінії.

Маршрутизатори підтримують різні типи мережевого середовища: Ethernet, Frame Relay, DSL, бездротові стандарти Wi-Fi (802.11), Bluetooth тощо. Це дозволяє їм працювати в різноманітних умовах, від домашніх мереж до великих корпоративних і міжконтинентальних з’єднань.

Щоб пристрій міг використовувати маршрутизатор для виходу в інші мережі, він має бути налаштований з певними параметрами IP.

IP-адреса – унікальний ідентифікатор пристрою в мережі. Наприклад, ПК у мережі може мати адресу 192.168.1.10.

Маска підмережі – визначає межі підмережі, з якою пристрій може обмінюватися даними безпосередньо.

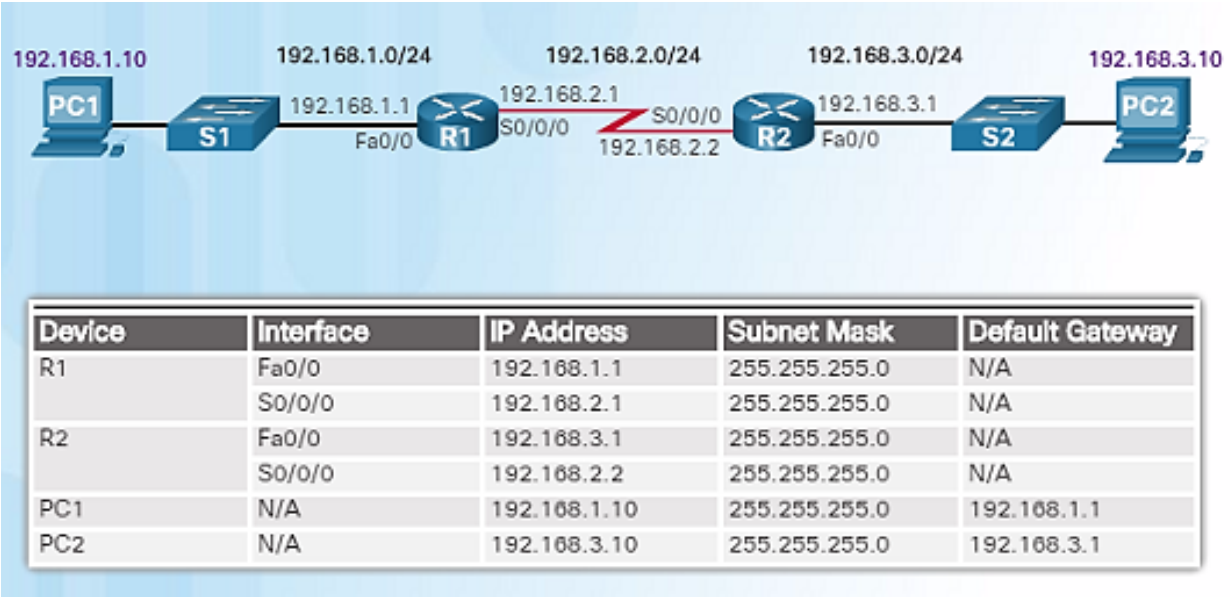
Шлюз за замовчуванням – вказує, куди надсилати пакети, якщо їх IP-адреса знаходиться за межами локальної мережі. Як правило, це адреса інтерфейсу маршрутизатора в цій же мережі, наприклад, 192.168.1.1. Якщо пристрій відправляє пакет до іншого вузла в тій же мережі (наприклад, з 192.168.1.10 до 192.168.1.20), то він передається безпосередньо. Якщо ж пакет призначено для іншої мережі (наприклад, 8.8.8.8), він відправляється на шлюз – маршрутизатор, який уже вирішує, як його переслати далі.

Під час проєктування або аналізу мережі важливо мати добре документовану інфраструктуру. Документація повинна включати назви пристроїв, IP-адресацію, маски підмереж, конфігурації інтерфейсів та шлюзи за замовчуванням. Це значно полегшує обслуговування, масштабування та усунення несправностей.

Таблиця адресації – таблична форму, в якій вказано імена пристроїв, інтерфейси, IP-адреси, маски підмереж і шлюзи (табл. 13.1, рис. 13.2).

Таблиця 13.1. **Приклад таблиці адресації**

Пристрій	Інтерфейс	IP-адреса	Маска підмережі	Шлюз за замовчуванням
Router1	G0/0	192.168.1.1	255.255.255.0	–
PC1	NIC	192.168.1.10	255.255.255.0	192.168.1.1



Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	Fa0/0	192.168.1.1	255.255.255.0	N/A
	S0/0/0	192.168.2.1	255.255.255.0	N/A
R2	Fa0/0	192.168.3.1	255.255.255.0	N/A
	S0/0/0	192.168.2.2	255.255.255.0	N/A
PC1	N/A	192.168.1.10	255.255.255.0	192.168.1.1
PC2	N/A	192.168.3.10	255.255.255.0	192.168.3.1

Рис. 13.2. Приклади таблиці адресації з логічною топологією мережі

13.3. Налаштування основних параметрів маршрутизатора

Однією з ключових відмінностей між комутаторами та маршрутизаторами є різні типи інтерфейсів, які вони підтримують. Комутатори рівня 2, що працюють на канальному рівні моделі OSI, зазвичай призначені для з'єднання пристроїв в межах локальної мережі (LAN). Саме тому більшість таких комутаторів мають декілька портів FastEthernet (100 Мбіт/с) або Gigabit Ethernet (1 Гбіт/с).

Наприклад, звичайний настільний комутатор на 24 порти може забезпечити підключення комп'ютерів, принтерів і серверів у локальній офісній мережі. Натомість маршрутизатори мають підтримку як локальних мереж LAN, так і глобальних мереж WAN, що дає змогу їм об'єднувати різні типи мереж, у тому числі мережі з різними технологіями передачі даних. Це вимагає наявності великої кількості інтерфейсів різного типу. Наприклад, маршрутизатори серії Cisco ISR G2 (Integrated Services Routers Generation 2) мають один або два вбудовані інтерфейси Gigabit Ethernet, а також слоти для встановлення модулів HWIC (High-Speed WAN Interface Cards). Такі модулі дозволяють додавати порти для ADSL, кабельного інтернету або навіть старіших послідовних інтерфейсів, які використовуються в корпоративних мережах.

Щоб інтерфейс на маршрутизаторі міг передавати дані, його необхідно правильно налаштувати. По-перше, йому призначається IP-адреса разом із маскою підмережі, якщо використовується протокол IPv4. Це робиться командою *ip address*, наприклад:

Router(config-if)# ip address 192.168.1.1 255.255.255.0

По-друге, інтерфейс потрібно увімкнути, оскільки за замовчуванням усі інтерфейси маршрутизатора вимкнені (у стані *shutdown*). Для активації використовується команда *no shutdown*. Це подібно до подачі живлення на порт. Також фізичне з'єднання є обов'язковим – кабель має бути підключений до іншого пристрою, інакше інтерфейс залишатиметься в неактивному стані.

Для кращої організації конфігурації мережі рекомендовано задавати короткий опис кожного інтерфейсу за допомогою команди *description*. Наприклад:

Router(config-if)# description Підключення до офісного комутатора

Цей опис полегшує ідентифікацію інтерфейсу, особливо в складних мережах або при усуненні неполадок. Якщо інтерфейс веде до зовнішньої мережі, наприклад, до провайдера, доцільно вказати контактні дані провайдера або номер договору.

Окрім фізичних інтерфейсів, маршрутизатори також підтримують логічні інтерфейси, зокрема, інтерфейс *loopback*. Це віртуальний (програмний) інтерфейс, що існує всередині пристрою і не пов'язаний з фізичним портом. Особливістю *loopback* є те, що він завжди перебуває у стані UP, доки працює сам маршрутизатор. Це робить його корисним для тестування, моніторингу та ідентифікації маршрутизатора.

Інтерфейс *loopback* часто використовується в динамічних протоколах маршрутизації, таких як OSPF, як основна IP-адреса маршрутизатора. Це забезпечує стабільну ідентифікацію, навіть якщо фізичні інтерфейси можуть бути недоступними. Наприклад:

Router(config)# interface loopback 0

Router(config-if)# ip address 10.0.0.1 255.255.255.255

Можна створити декілька *loopback*-інтерфейсів, кожен з унікальною IP-адресою. Це зручно для імітації підключених мереж або використання в тестових цілях.

Після налаштування інтерфейсів необхідно перевірити їхній стан. Для цього використовуються команди Cisco IOS:

show ip interface brief – надає стислий огляд усіх інтерфейсів, показуючи IP-адресу, стан (up/down) та чи активований інтерфейс;

show ip route – відображає таблицю маршрутизації, що зберігається в ОЗП, тут можна побачити мережі, до яких маршрутизатор має доступ, а також які інтерфейси використовуються для передавання пакетів;

show running-config interface [interface-id] – показує поточну конфігурацію конкретного інтерфейсу.

Також існують додаткові команди для більш детального аналізу:

show interfaces – виводить статистику трафіку, помилок, колізій та інші параметри для кожного інтерфейсу;

show ip interface – деталізує налаштування IPv4, включаючи фільтри доступу, підтримку NAT, MTU та інші параметри.

Для перевірки налаштувань у мережі IPv6 використовуються ті ж команди, адаптовані до IPv6.

Наприклад, ping для IPv6 виконується так само, як і для IPv4, але з використанням IPv6-адреси, наприклад:

Router# ping 2001:db8::1.

13.4. Таблиця і записи маршрутизації

Маршрутизатор – це мережевий пристрій, основним завданням якого є визначення найкращого маршруту для доставки пакетів до місця призначення.

Коли на маршрутизатор надходить IP-пакет, він аналізує його адресу призначення та виконує пошук у таблиці маршрутизації – спеціальному наборі записів, який вказує, через який інтерфейс і до якого маршрутизатора далі слід переслати пакет.

Основна функція маршрутизатора – вибір оптимального шляху.

Типи маршрутів: пряме підключення, віддалена мережа та маршрут за замовчуванням. Під час пошуку відповідного маршруту маршрутизатор може знайти три типи шляхів.

Мережа з прямим підключенням (Connected network)

Якщо IP-адреса призначення належить до тієї ж підмережі, що й один з інтерфейсів маршрутизатора, пакет передається безпосередньо до кінцевого пристрою (рис. 13.3).

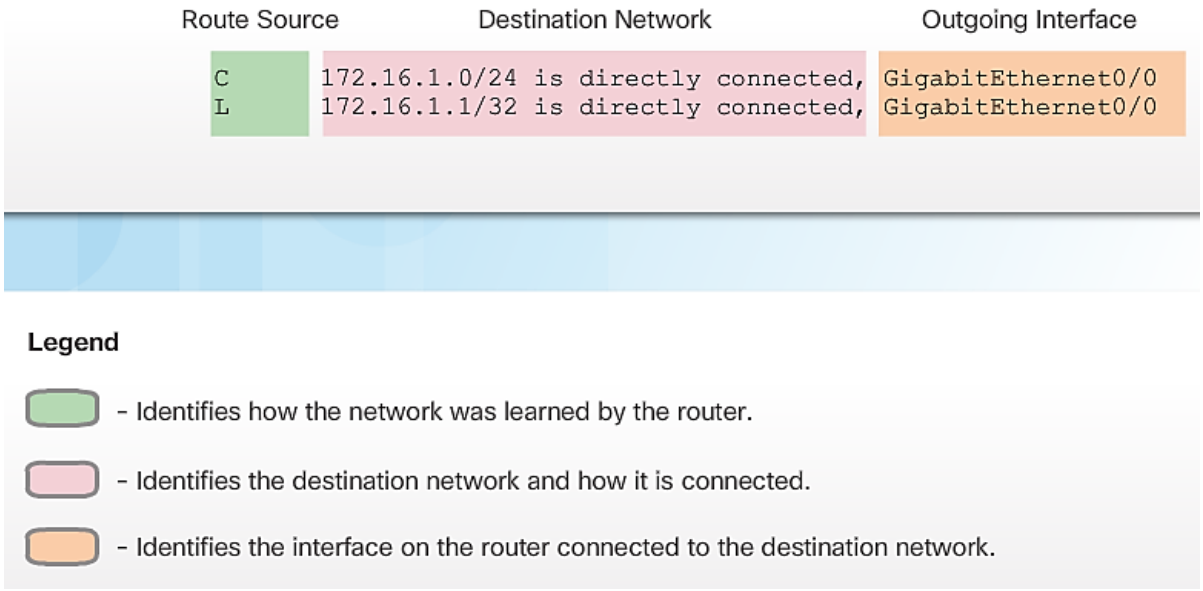


Рис. 13.3. Приклад записів таблиці маршрутизації для мереж з прямим підключенням

Наприклад, якщо інтерфейс маршрутизатора має адресу 192.168.1.1/24, і пакет надійшов для 192.168.1.45, то передача відбудеться безпосередньо – без участі інших маршрутизаторів.

Віддалена мережа (Remote network)

Якщо адреса призначення належить до мережі, яка не є безпосередньо підключеною, маршрутизатор передає пакет іншому маршрутизатору – далі по шляху (рис. 13.4).

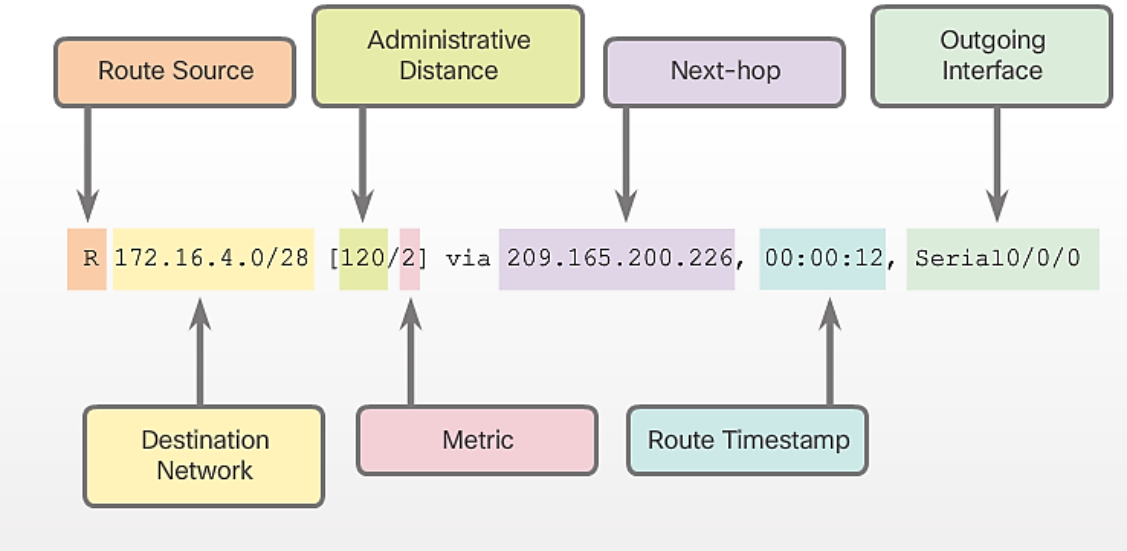


Рис. 13.4. Приклад запису таблиці маршрутизації для віддаленої мережі

Наприклад, для адреси 10.10.10.10 маршрутизатор пересилає пакет через інтерфейс, підключений до іншого маршрутизатора, який уже знає, як досягти цієї адреси.

Невідомий маршрут (Route not found)

Якщо в таблиці маршрутизації немає запису, що відповідає IP-адресі призначення, маршрутизатор шукає маршрут за замовчуванням (default route). Якщо такий маршрут задано (наприклад, ip route 0.0.0.0 0.0.0.0 192.168.0.1), пакет пересилається на так званий «шлюз останньої надії». Якщо ж такого маршруту немає – пакет відкидається, а відправнику надсилається ICMP-повідомлення про недоступність мережі.

У ситуації, коли до однієї мережі існує декілька маршрутів, маршрутизатор вибирає найкращий шлях на основі метрики – числового значення, яке характеризує «вартість» шляху. Метрика залежить від

параметрів, таких як кількість переходів, швидкість каналу, затримка, надійність, навантаження тощо.

Різні протоколи маршрутизації використовують різні метрики.

RIP (Routing Information Protocol) – використовує кількість переходів (hops). Максимум – 15, більше – вважається недосяжним.

OSPF (Open Shortest Path First) – базується на ширині смуги пропускання шляху; пріоритет надається маршрутам із більш високою пропускною здатністю.

EIGRP (Enhanced Interior Gateway Routing Protocol) – враховує кілька факторів: затримку, ширину смуги, надійність, навантаження.

Коли два або більше маршрути до одного пункту призначення мають однакову метрику, маршрутизатор може використовувати всі ці маршрути одночасно. Це явище називається *розподілом навантаження з рівною вартістю (equal-cost load balancing)*. У таблиці маршрутизації буде один запис призначення, але кілька вихідних інтерфейсів. Це дозволяє підвищити ефективність використання мережевих ресурсів.

Наприклад, якщо до мережі 10.0.0.0/24 є два однаково «дешеві» шляхи – через FastEthernet0/0 та GigabitEthernet0/1, маршрутизатор буде чергувати пересилку пакетів між ними.

Тільки протокол EIGRP також підтримує *розподіл навантаження з нерівною вартістю (unequal-cost load balancing)*, коли маршрутизатор використовує кілька шляхів з різними метриками, але з різною частотою передачі.

У разі, коли маршрутизатор має кілька шляхів до одного пункту призначення, отриманих з різних джерел (наприклад, RIP і EIGRP), він не лише порівнює метрики, а й враховує адміністративну відстань (AD) – показник довіри до джерела маршруту: чим менше значення AD, тим надійніший маршрут.

Наприклад:

пряме підключення має AD = 0;

статичний маршрут – AD = 1;

EIGRP – AD = 90;

OSPF – AD = 110;

RIP – AD = 120.

Якщо є два маршрути – статичний і динамічний (OSPF), маршрутизатор вибере статичний, бо його AD нижче.

Таблиця маршрутизації зберігається в оперативній пам’яті і містить повну інформацію про те, які мережі доступні, яким шляхом до них дістатися, і через який інтерфейс.

Усі маршрути мають специфічні коди:

L – локальна IP-адреса, призначена інтерфейсу маршрутизатора.

C – маршрут до мережі з прямим підключенням.

S – статичний маршрут.

D – маршрут, отриманий за допомогою EIGRP.

O – маршрут, отриманий за допомогою OSPF.

Наприклад, у таблиці можуть бути такі записи:

C 192.168.1.0/24 is directly connected, FastEthernet0/0

L 192.168.1.1/32 is directly connected, FastEthernet0/0

O 10.0.0.0/8 [110/20] via 192.168.1.2, FastEthernet0/0

Для перегляду таблиці маршрутизації використовується команда:

Router# show ip route

Ця команда відображає:

- типи маршрутів (прямі, локальні, статичні, динамічні),
- метрики,
- вихідні інтерфейси,
- маршрутизатори наступного переходу.

У сучасних версіях IOS також видно локальні записи (L), які вказують на IP-адреси, безпосередньо призначені інтерфейсам – вони використовуються для самоідентифікації пристрою.

На рис. 13.5 зображено фрагмент таблиці маршрутизації маршрутизатора R1, отриманої за допомогою команди *show ip route*. У записах таблиці вказано, що маршрут за замовчуванням (default route) направляється через інтерфейс Serial0/0/1 до IP-адреси 209.165.200.234.

Таблиця містить маршрути до локальних і безпосередньо підключених мереж (позначені як L і C), а також динамічно отримані маршрути за допомогою протоколу RIP (R) до мереж, таких як 172.16.3.0/24, 172.16.4.0/28, 209.165.200.224/30 та інших.

Інтерфейси GigabitEthernet0/0, Serial0/0/0 та Serial0/0/1 активно використовуються для з'єднання з різними мережами.

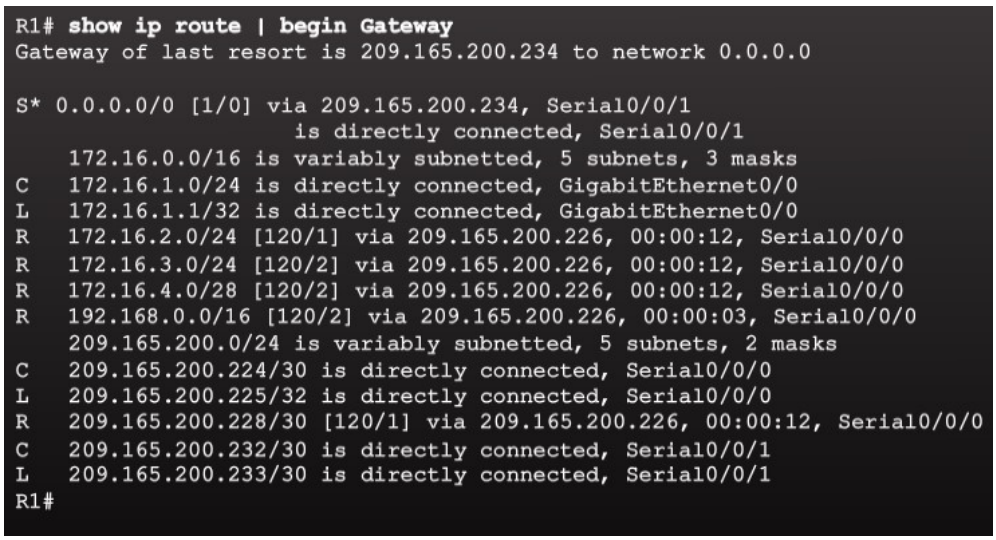


Рис. 13.5. Фрагмент таблиці маршрутизації маршрутизатора R1, отриманої за допомогою команди *show ip route | begin Gateway*

14.5. Статичні та динамічні маршрути

Після початкового налаштування інтерфейсів маршрутизатора, коли вони вже функціонують і з'явилися у таблиці маршрутизації як безпосередньо підключені, наступним етапом є організація маршрутизації між віддаленими мережами. Для цього використовують два основні підходи – статичну або динамічну маршрутизацію. Обидва способи дозволяють маршрутизатору передавати дані за межі безпосередньо підключених мереж, проте реалізуються по-різному.

Статична маршрутизація полягає у ручному налаштуванні маршрутів адміністратором. У цьому випадку кожен маршрут задається командою, яка вказує, куди слід пересилати пакети для досягнення певної мережі. Наприклад, щоб налаштувати маршрут до мережі 192.168.5.0/24 через маршрутизатор з IP 10.0.0.2, адміністратор використовує команду:

ip route 192.168.5.0 255.255.255.0 10.0.0.2

Такі маршрути мають позначку «S» у таблиці маршрутизації. Основною перевагою цього підходу є передбачуваність, безпека та ефективне використання ресурсів: він не навантажує процесор і не споживає зайвої пропускної здатності на обмін службовими повідомленнями. Однак, якщо мережа змінюється – додаються нові маршрутизатори, зникають зв'язки чи оновлюється топологія – адміністратору доведеться вносити зміни вручну, інакше пакети не

знайдуть потрібний шлях.

Окремим випадком є *статичний маршрут за замовчуванням*, який використовується, коли невідомо, у якій мережі розташований конкретний вузол. У такому випадку маршрутизатор перенаправляє трафік на вказаний шлюз. Типовий приклад:

ip route 0.0.0.0 0.0.0.0 209.165.200.234

Це означає, що будь-який трафік, який не відповідає іншим маршрутам у таблиці, буде надіслано на IP-адресу 209.165.200.234. Такий маршрут зручно використовувати в ситуаціях, коли маршрутизатор має лише один вихід в інші мережі – наприклад, до інтернет-шлюзу чи центрального маршрутизатора компанії.

Динамічна маршрутизація, на відміну від статичної, передбачає автоматичний обмін інформацією про маршрути між маршрутизаторами за допомогою спеціальних протоколів – таких як RIP, OSPF чи EIGRP. Вони дозволяють маршрутизаторам дізнаватися про існуючі мережі в реальному часі, аналізувати шляхи до них та автоматично оновлювати свої таблиці маршрутизації при зміні топології. Наприклад, при використанні RIP маршрутизатори кожні 30 секунд надсилають оновлення сусідам, повідомляючи про нові або змінені маршрути.

Важливою перевагою динамічної маршрутизації є її здатність швидко реагувати на зміни в мережі без участі адміністратора.

Якщо певний маршрут стає недоступним, протокол знаходить альтернативний шлях і оновлює таблицю. Це робить мережу більш гнучкою та надійною. Однак такі протоколи потребують більше ресурсів – вони споживають смугу пропускання і процесорний час для обчислень і обміну даними.

Щоб дізнатися, які протоколи маршрутизації підтримує операційна система IOS на маршрутизаторі, можна скористатися командою:

router ?

У випадку з IPv6 протоколи маршрутизації мають адаптовану структуру. Для ввімкнення маршрутизації IPv6 використовується команда:

ipv6 unicast-routing

А для налаштування статичного маршруту за замовчуванням в

середовищі IPv6:

ipv6 route ::/0 <ipv6-next-hop або вихідний інтерфейс>

Отже, вибір між статичною і динамічною маршрутизацією залежить від конкретних вимог до гнучкості, масштабу мережі та доступних ресурсів. У невеликих або стабільних мережах доцільніше використовувати статичні маршрути, тоді як у великих і динамічних – перевагу надають динамічній маршрутизації.

14.6. Маршрутизація VLAN

У сучасних комп'ютерних мережах VLAN (віртуальні локальні мережі) дозволяють розділяти одну фізичну мережу на логічні сегменти. Однак для передачі даних між такими сегментами необхідна маршрутизація на третьому рівні моделі OSI. Цей процес забезпечує керований обмін трафіком між VLAN і може здійснюватися за допомогою маршрутизатора або комутатора третього рівня (багаторівневого комутатора). Наприклад, якщо комп’ютер з VLAN 10 хоче надіслати дані до пристрою у VLAN 20, то без маршрутизатора або багаторівневого комутатора цей обмін не відбудеться.

Оскільки VLAN утворює окремий широкомовний домен, то вузли, які належать до різних VLAN, не можуть безпосередньо спілкуватися між собою. Для обміну інформацією між цими сегментами необхідний пристрій маршрутизації, який здатен аналізувати IP-адреси й пересилати пакети на основі інформації з таблиці маршрутизації. Такими пристроями можуть бути як традиційні маршрутизатори, так і багаторівневі комутатори, які підтримують маршрутизацію між VLAN. Сам процес обміну трафіком між VLAN отримав назву «маршрутизація між VLAN» (inter-VLAN routing).

Класичний метод маршрутизації між VLAN передбачав підключення окремих фізичних інтерфейсів маршрутизатора до відповідних VLAN на комутаторі. Наприклад, для трьох VLAN було потрібно три окремі інтерфейси на маршрутизаторі й три порти на комутаторі. Такий підхід вимагав значної кількості портів і кабелів, що ускладнювало масштабування та збільшувало вартість.

Більш сучасним та ефективним методом маршрутизації між VLAN є технологія "router-on-a-stick". Вона передбачає використання одного фізичного інтерфейсу маршрутизатора, який працює в режимі транка.

Цей інтерфейс з'єднується з транковим портом комутатора і може приймати трафік з мітками VLAN. Для цього на маршрутизаторі створюються підінтерфейси (subinterfaces), кожен з яких відповідає певній VLAN. Наприклад, підінтерфейс GigabitEthernet0/0.10 може обслуговувати VLAN 10, а GigabitEthernet0/0.20 – VLAN 20. Кожному з цих підінтерфейсів задається IP-адреса для відповідної підмережі.

Після отримання тегованого кадру з VLAN, маршрутизатор розпізнає VLAN за міткою, обробляє пакет відповідно до IP-адреси призначення, і відправляє його назад у відповідну VLAN через той же фізичний інтерфейс. Цей метод є економічно вигідним, оскільки не вимагає багато портів, але має обмеження щодо пропускної здатності, адже вся маршрутизація проходить через один інтерфейс.

Альтернативним і більш продуктивним варіантом є використання багаторівневих комутаторів, які поєднують функції комутатора другого рівня і маршрутизатора третього рівня. Вони можуть здійснювати маршрутизацію між VLAN без залучення зовнішнього маршрутизатора. Для цього на комутаторі активується функція IP-маршрутизації за допомогою команди *ip routing*. Завдяки апаратній обробці даних багаторівнева маршрутизація забезпечує вищу швидкодію і менші затримки.

Великою перевагою багаторівневих комутаторів є те, що весь трафік маршрутизується безпосередньо в межах пристрою, що зменшує навантаження на мережу і знижує вимоги до пропускної здатності ліній зв’язку. Наприклад, якщо вузли VLAN 10 і VLAN 20 підключені до одного багаторівневого комутатора, то весь обмін між ними відбувається локально. Проте, багаторівневі комутатори є дорожчими за традиційні маршрутизатори, що може бути обмеженням для невеликих мереж.

У минулому маршрутизація між VLAN виконувалася програмно, що суттєво уповільнювало обробку пакетів порівняно з комутацією. Тому мережі проєктувалися таким чином, щоб максимізувати використання другого рівня, залишаючи маршрутизацію лише для з’єднання із зовнішніми мережами. Це призводило до складної топології з великою кількістю петель, для уникнення яких застосовувався протокол STP (Spanning Tree Protocol). STP дозволяв блокувати надлишкові зв’язки для уникнення циклів, але при цьому знижував

ефективність використання каналів.

Завдяки розвитку апаратної маршрутизації, сьогодні стало можливим реалізовувати маршрутизацію навіть на рівні ядра мережі без втрати продуктивності. Це дозволяє скорочувати обсяг широкомовного трафіку і збільшувати масштабованість мережі.

Як приклад, кожна VLAN є окремою підмережею, і логічно зробити так, щоб комутатори розподілу виконували функції шлюзів третього рівня для користувачів VLAN, підключених до комутаторів доступу. Тобто комутатори розподілу повинні мати IP-адреси, які відповідають усім VLAN, що використовуються на комутаторах доступу.

Між рівнями розподілу і ядра зазвичай використовуються маршрутизовані порти третього рівня, що підвищує продуктивність і дозволяє обійтися без протоколів типу STP. Оскільки в цій частині мережі немає фізичних петель, потреба в STP зникає.

У таких мережах зручно використовувати інтерфейси типу SVI (Switched Virtual Interface). SVI є логічними інтерфейсами, що асоціюються з VLAN і дозволяють комутаторам здійснювати маршрутизацію між VLAN.

Розглянемо основні причини використання SVI.

- SVI виступає шлюзом VLAN для маршрутизації в інші мережі.
- Дозволяє налаштовувати IP-з’єднання третього рівня на комутаторі.
- Підтримує протоколи маршрутизації та функції моста.

Переваги використання SVI:

- значно вища швидкість маршрутизації, ніж у методу "router-on-a-stick", оскільки маршрутизація виконується апаратно;
- відсутність потреби у фізичних з'єднаннях з маршрутизатором;
- можливість створення декількох високошвидкісних з’єднань між комутаторами, наприклад, з використанням EtherChannel;
- менші затримки, оскільки весь трафік маршрутизується безпосередньо всередині комутатора.

У невеликих мережах доцільно використовувати "router-on-a-stick", тоді як для середніх і великих мереж краще підходять багаторівневі комутатори з підтримкою SVI.

Контрольні запитання та задачі до теми 13

Перелік теоретичних запитань

1. Що таке маршрутизація і яке її основне призначення в комп’ютерних мережах?
2. Які функції виконує маршрутизатор у мережі?
3. Чим відрізняються маршрутизатори третього рівня від комутаторів другого рівня?
4. Які існують типи маршрутів? Наведіть приклади.
5. Що таке статичний маршрут? Як він налаштовується?
6. Які переваги та недоліки має статична маршрутизація?
7. Як працює динамічна маршрутизація? Наведіть приклади протоколів.
8. Які є переваги динамічної маршрутизації порівняно зі статичною?
9. Як виглядає запис статичного маршруту в конфігурації Cisco?
10. Що таке маршрут за замовчуванням і коли він використовується?
11. Які функції виконує таблиця маршрутизації?
12. Як перевірити таблицю маршрутизації на маршрутизаторі Cisco?
13. Що таке VLAN і як вона ізолює трафік у мережі?
14. Чому між різними VLAN потрібна маршрутизація?
15. Які методи маршрутизації між VLAN існують? Поясніть метод Router-on-a-Stick і використання SVI?

Перелік задач для самостійного опрацювання

Практичне завдання 1. Статична маршрутизація між двома мережами.

У середовищі моделювання Cisco Packet Tracer потрібно налаштувати два маршрутизатори для зв’язку двох окремих VLAN за допомогою статичних маршрутів. Створіть дві мережі (наприклад, 192.168.1.0/24 та 192.168.2.0/24), кожна з яких підключена до окремого маршрутизатора. З’єднайте маршрутизатори інтерфейсами та призначте їм IP-адреси. Додайте статичні маршрути на кожному з маршрутизаторів. Перевірте зв’язок між ПК у різних VLAN.

Практичне завдання 2. Реалізація маршрутизації VLAN за допомогою Router-on-a-Stick.

У середовищі моделювання Cisco Packet Tracer потрібно налаштувати маршрутизацію між VLAN на одному фізичному інтерфейсі маршрутизатора. Створіть 2 VLAN (наприклад, VLAN 10 та VLAN 20). Призначте відповідні порти комутатора до цих VLAN. Створіть підінтерфейси на маршрутизаторі для кожної VLAN. Призначте IP-адреси на підінтерфейсах. Налаштуйте інтерфейс комутатора як *trunk*. Перевірте зв'язок між хостами у різних VLAN.

Практичне завдання 3. Налаштування SVI на багаторівневому комутаторі.

У середовищі Cisco Packet Tracer потрібно забезпечити маршрутизацію між VLAN за допомогою SVI на L3-комутаторі. Створіть VLAN 10, 20 і призначте порти. Увімкніть *IP-routing*. Створіть інтерфейси VLAN 10, VLAN 20 з IP-адресами. Налаштуйте порти комутатора як *access*. Перевірте маршрутизацію між VLAN.

Тестове завдання 4.

Яке твердження найточніше описує статичний маршрут?

а) Встановлюється автоматично протоколом

б) Вимагає ручного налаштування

в) Використовується лише в IPv6

г) Не змінюється навіть при збої в мережі

Тестове завдання 5.

Яка команда використовується для налаштування статичного маршруту?

а) ip default-gateway

б) ip route

в) ip static

г) ip add route

Перелік використаної та рекомендованої літератури

13.1. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

13.2. Requirements for IP Version 4 Routers [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc1812

13.3. What is routing [Електронний ресурс]. – Режим доступу: https://aws.amazon.com/what-is/routing/

Тема 14. Статична та динамічна маршрутизація

У сучасних комп’ютерних мережах маршрутизація відіграє ключову роль у забезпеченні зв’язку між різними підмережами. Завдяки маршрутизації можлива передача даних між локальними мережами, між VLAN, а також з'єднання з глобальною мережею Інтернет. В основі маршрутизації лежить принцип прийняття рішень про те, куди і яким шляхом направити пакети даних. Цей процес відбувається на мережевому рівні (3-й рівень моделі OSI), і для його реалізації використовуються маршрутизатори або багаторівневі комутатори.

Маршрутизація буває двох основних типів: статична і динамічна.

У статичній маршрутизації всі маршрути налаштовуються вручну адміністратором. Це забезпечує повний контроль над рухом трафіку, але вимагає постійного втручання при зміні топології мережі. Такий підхід добре підходить для невеликих або стабільних мереж, де рідко відбуваються зміни. На відміну від цього, динамічна маршрутизація використовує спеціальні протоколи, які дозволяють маршрутизаторам автоматично обмінюватися інформацією про маршрути. Це забезпечує високу гнучкість та адаптивність мережі до змін, таких як збої з'єднань чи поява нових вузлів. Протоколи динамічної маршрутизації, такі як RIP, OSPF або EIGRP, дозволяють мережі автоматично знаходити оптимальні шляхи для передачі даних. Розуміння відмінностей, переваг і недоліків обох типів маршрутизації є важливим для проєктування надійних і масштабованих мереж.

У цій темі ми детально розглянемо принципи роботи статичної та динамічної маршрутизації, способи їх налаштування та ситуації, у яких кожен із підходів буде найефективнішим.

14.1. Переваги та призначення статичної маршрутизації

У комп’ютерних мережах маршрутизація забезпечує процес передачі даних між різними мережами, переміщуючи пакети від відправника до одержувача через серію проміжних пристроїв.

Ключовими пристроями, що виконують ці функції, є маршрутизатори. Вони аналізують інформацію, закладену в заголовках пакетів, і визначають найкращий маршрут для передачі даних далі – у напрямку до цільової мережі.

Наприклад, якщо комп’ютер в офісі з IP-адресою з підмережі 192.168.1.0/24 намагається зв’язатися з вебсервером в Інтернеті, маршрутизатор отримає цей пакет, визначить, що адресат знаходиться в зовнішній мережі, і перенаправить його до інтернет-провайдера або до наступного маршрутизатора.

Маршрутизатори дізнаються про існування віддалених мереж двома способами: статично і динамічно.

Статично – адміністратор вручну вводить інформацію про маршрути до певних мереж, вказуючи або IP-адресу наступного переходу, або вихідний інтерфейс маршрутизатора. Наприклад, команда:

ip route 10.10.0.0 255.255.0.0 192.168.1.1

вказує маршрутизатору, що мережа 10.10.0.0/16 досяжна через шлюз 192.168.1.1.

Динамічно – маршрутизатор автоматично отримує та оновлює маршрути за допомогою спеціальних протоколів маршрутизації, таких як RIP, OSPF, або EIGRP. Такі протоколи дозволяють пристроям обмінюватися маршрутною інформацією та адаптуватися до змін у мережі без ручного втручання.

У більшості сучасних мереж використовується комбінація обох способів – для досягнення балансу між контролем і автоматизацією.

Статична маршрутизація передбачає повний ручний контроль з боку адміністратора. Вона є простою для реалізації у малих або стабільних мережах, де топологія змінюється рідко.

Наприклад, у невеликому офісі, де всі пристрої мають один вихід в Інтернет, можна налаштувати єдиний статичний маршрут за замовчуванням.

Переваги статичної маршрутизації:

- маршрути не передаються мережею, що зменшує ризик перехоплення інформації;
- не використовуються ресурси процесора або пропускна здатність для обробки оновлень маршрутизації;
- адміністратор точно знає, яким шляхом рухаються дані.

Недоліки статичної маршрутизації:

- при зміні топології маршрути потрібно оновлювати вручну;

- у великих мережах складно підтримувати актуальність вручну заданих маршрутів;
- кожен новий маршрут вимагає додаткового запису.

На відміну від статичної, динамічна маршрутизація автоматично будує таблиці маршрутизації, забезпечуючи кращу гнучкість у великих або складних мережах. Протоколи, такі як OSPF, будують карту мережі та автоматично вибирають найкоротші або найшвидші маршрути.

Наприклад, при відмові одного з маршрутизаторів, динамічна система автоматично знайде альтернативний шлях, не потребуючи втручання адміністратора (табл. 14.1).

Таблиця 14.1. **Порівняння динамічної і статичної маршрутизації**

Критерій	Динамічна маршрутизація	Статична маршрутизація
Складність налаштування	Як правило, не залежить від розміру мережі	Зростає зі збільшенням розміру мережі
Зміни топології	Автоматично адаптується до змін	Потрібне втручання адміністратора
Масштабованість	Підходить для простих і складних топологій	Підходить лише для простих топологій
Безпека	Менш безпечна	Більш безпечна
Використання ресурсів	Використовує CPU, пам’ять, пропускну здатність каналу	Додаткові ресурси не потрібні
Прогнозованість маршруту	Маршрут залежить від поточної топології	Маршрут до пункту призначення завжди однаковий
Автоматичне оновлення маршрутів	Так	Ні
Резервування маршрутів	Автоматичне, залежить від протоколу	Через плаваючі статичні маршрути з підвищеною адміністративною відстанню
Контроль адміністратора	Частковий, залежить від протоколу	Повний контроль
Типові приклади використання	Великі корпоративні мережі з багатьма маршрутизаторами	Невеликі мережі, тупикові мережі, маршрути за замовчуванням
Адміністративна відстань (AD)	Вища (OSPF – 110, RIP – 120)	1 (має вищий пріоритет)

Плаваючий статичний маршрут – це резервний маршрут, який активується лише у разі недоступності основного. Його налаштовують із вищим значенням адміністративної відстані AD, щоб він не конфліктував з основним маршрутом до тієї ж мережі.

Статичний маршрут за замовчуванням (default route) використовується для пересилання трафіку до будь-якої адреси, яка не співпадає з жодним іншим маршрутом у таблиці. У IPv4 такий маршрут має адресу призначення 0.0.0.0/0. Наприклад:

ip route 0.0.0.0 0.0.0.0 192.168.1.1

Це інструкція для маршрутизатора: якщо не знаєш, куди направити трафік – відправляй на 192.168.1.1.

Розглянемо приклади застосування статичної маршрутизації.

Статична маршрутизація часто використовується у таких випадках:

- невеликі або тупикові мережі, де є лише один шлях до зовнішнього світу;
- пограничні маршрутизатори, що з’єднують локальну мережу з Інтернетом;
- VPN-з'єднання, де потрібен контроль над маршрутами трафіку.

Також можливо агрегувати статичні маршрути, тобто об'єднувати кілька мереж у один узагальнений маршрут, якщо вони мають спільний префікс. Це дозволяє скоротити кількість записів у таблиці маршрутизації.

У мережевому адмініструванні важливо правильно комбінувати статичну і динамічну маршрутизацію. Статичні маршрути забезпечують простоту, контроль і безпеку в передбачуваних умовах. Натомість динамічна маршрутизація дозволяє автоматично пристосовуватись до змін, що робить її ідеальною для великих або часто змінюваних мереж. Правильне поєднання обох методів дозволяє створити стабільну, масштабовану й адаптивну інфраструктуру.

14.2. Налаштування статичного маршруту

Статичні маршрути конфігуруються вручну адміністратором з використанням команди глобального режиму конфігурації *ip route*.

Розглянемо загальний синтаксис цієї команди.

Router(config)# ip route <адреса_мережі_призначення> <маска_підмережі> {<IP-адреса наступного переходу> | <вихідний інтерфейс>} [distance] [name] [permanent] [tag]

При створенні маршруту потрібно обов'язково вказати адресу мережі призначення та маску підмережі.

Адреса мережі призначення (network-address) – це IP-адреса мережі, до якої потрібно доставити пакети. Її також називають префіксом.

Маска підмережі (subnet-mask) – визначає розмір мережі призначення. Наприклад, 255.255.255.0 означає мережу з 256 адресами. Іноді використовується скорочене CIDR-нотація, наприклад, /24.

Наприклад:

ip route 192.168.10.0 255.255.255.0 192.168.1.1

Це означає, що усі пакети, призначені для мережі 192.168.10.0/24, потрібно пересилати через маршрутизатор з IP 192.168.1.1.

Для маршрутизації пакету необхідно знати, куди його направити. Це може бути:

- *IP-адреса наступного переходу* – IP найближчого маршрутизатора, до якого треба направити пакет;
- *вихідний інтерфейс* – інтерфейс маршрутизатора, через який буде передано пакет (наприклад, GigabitEthernet0/1).

Можна вказати лише одне з них або обидва. Наприклад:

тільки IP:

ip route 10.0.0.0 255.255.255.0 192.168.1.2

тільки інтерфейс:

ip route 10.0.0.0 255.255.255.0 GigabitEthernet0/0

обидва:

ip route 10.0.0.0 255.255.255.0 GigabitEthernet0/0 192.168.1.2

Маршрут наступного переходу – вказується лише IP-адреса сусіднього маршрутизатора. Наприклад:

ip route 192.168.5.0 255.255.255.0 10.0.0.1

Безпосередньо підключений маршрут – вказується тільки інтерфейс, через який треба передавати пакети:

ip route 192.168.5.0 255.255.255.0 Serial0/0/0

Такий маршрут дозволяє уникнути додаткового рекурсивного пошуку наступного переходу.

Повністю заданий маршрут – вказується інтерфейс, і IP-адреса наступного переходу:

ip route 192.168.5.0 255.255.255.0 GigabitEthernet0/1 10.0.0.2

Використовується для мереж з множинним доступом (наприклад, Ethernet), де важливо точно знати, з ким саме потрібно встановити з'єднання.

Плаваючі статичні маршрути – іноді потрібно створити резервний маршрут, який буде використовуватися лише в разі недоступності основного маршруту. Для цього вказують адміністративну відстань, яка має бути вищою, ніж у динамічного маршруту:

ip route 192.168.5.0 255.255.255.0 10.0.0.2 150

Тут число 150 – це адміністративна відстань. Основні маршрути мають нижчі значення (наприклад, 1 для звичайного статичного, 110 для OSPF).

У випадку, якщо вказано лише IP-адресу наступного переходу, маршрутизатор повинен виконати рекурсивний пошук, щоб з'ясувати, через який інтерфейс передавати пакети. Це споживає більше ресурсів, тому за можливості краще вказувати вихідний інтерфейс.

В Ethernet-мережах, які мають множинний доступ (тобто можуть містити багато пристроїв), вказування лише інтерфейсу може викликати проблеми. Наприклад:

ip route 192.168.2.0 255.255.255.0 GigabitEthernet0/1

Цей маршрут не дає чіткого уявлення, до якого саме пристрою в мережі треба передавати пакет. У такому випадку краще використовувати повністю заданий маршрут.

Після налаштування маршруту важливо переконатися, що він працює. Для цього використовують наступні команди:

ping – перевірка досяжності вузла;

traceroute – відображення шляху до вузла;

show ip route – перегляд таблиці маршрутизації;

show ip route static – перегляд лише статичних маршрутів;

show ip route <мережа> – перегляд маршруту до конкретної мережі.

Розглянемо приклад повної конфігурації.

R1(config)# ip route 192.168.2.0 255.255.255.0 GigabitEthernet0/1 10.1.1.2

R1# show ip route static

У цьому прикладі R1 пересилає пакети до мережі 192.168.2.0/24 через інтерфейс GigabitEthernet0/1, вказуючи на маршрутизатор із адресою 10.1.1.2.

14.3. Статичний маршрут за замовчуванням

У комп'ютерних мережах статичний маршрут за замовчуванням – це спеціальний запис у таблиці маршрутизації, який використовується для пересилки всіх пакетів, призначення яких не збігається з жодним конкретним маршрутом у таблиці. Такий маршрут часто називають маршрутом «останньої надії», оскільки він служить резервним шляхом у разі відсутності точніших маршрутів.

Наприклад, якщо маршрутизатор отримує пакет з адресою призначення 203.0.113.5, а в таблиці маршрутизації немає жодного маршруту, що покриває цю IP-адресу, він передасть пакет на вказаний шлюз за замовчуванням. Це дає змогу зменшити обсяг таблиці маршрутизації, особливо в тих випадках, коли маршрутизатор не має потреби зберігати окремий маршрут до кожної можливої мережі.

Найчастіше маршрути за замовчуванням застосовуються у двох основних сценаріях.

Наприклад, внутрішній маршрутизатор підприємства може передавати весь інтернет-трафік через один зовнішній інтерфейс, підключений до провайдера. У такому разі зручно використовувати маршрут за замовчуванням замість опису кожної зовнішньої мережі окремо.

Якщо маршрутизатор має лише один сусідній маршрутизатор або один можливий вихід у зовнішню мережу, він може передавати весь трафік за замовчуванням на цього сусіда. Це особливо актуально для невеликих філій або віддалених офісів.

Синтаксис команди для IPv4

Щоб створити статичний маршрут за замовчуванням у мережі IPv4, використовується команда:

ip route 0.0.0.0 0.0.0.0 {ip-адреса-наступного-переходу | вихідний-

інтерфейс}

Адреса 0.0.0.0 з маскою 0.0.0.0 (тобто маска /0) означає, що цей маршрут відповідає будь-якій IP-адресі, оскільки не вимагається збігу жодного біта. Такий запис охоплює всі можливі мережі, яких немає в таблиці.

Приклад:

R1(config)# ip route 0.0.0.0 0.0.0.0 192.168.1.1

У цьому прикладі маршрутизатор R1 передає всі пакети, для яких не знайдено кращого маршруту, через IP-адресу 192.168.1.1, яка є шлюзом до зовнішньої мережі.

Статичні маршрути за замовчуванням в IPv6

Для протоколу IPv6 команда для створення маршруту за замовчуванням має інший формат, але подібний зміст. Для цього використовується команда:

ipv6 route ::/0 {ipv6-адреса-наступного-переходу | вихідний-інтерфейс}

Префікс ::/0 у цьому випадку відіграє ту ж роль, що й 0.0.0.0/0 в IPv4 – він означає, що маршрут охоплює всі IPv6-адреси.

Приклад:

Router(config)# ipv6 route ::/0 2001:db8:1::1

Цей запис змушує маршрутизатор пересилати всі IPv6-пакети, що не мають точнішого збігу, через вузол з адресою 2001:db8:1::1.

Інші типи статичних маршрутів в IPv6

Окрім маршруту за замовчуванням, у мережах IPv6 можна налаштовувати:

- звичайні статичні маршрути – аналогічні до IPv4;
- сумарні маршрути (aggregate routes) – дозволяють об’єднувати кілька маршрутів в один;
- плаваючі маршрути (floating routes) – резервні маршрути з вищою адміністративною відстанню;
- повністю визначені маршрути – з одночасним зазначенням вихідного інтерфейсу і адреси наступного переходу.

Увімкнення маршрутизації IPv6

Перш ніж маршрутизатор зможе пересилати IPv6-пакети, необхідно активувати маршрутизацію для цього протоколу за

допомогою команди:

Router(config)# ipv6 unicast-routing

Без цієї команди маршрутизатор не зможе обробляти IPv6-маршрути, навіть якщо вони правильно налаштовані.

Маршрут за замовчуванням – це потужний інструмент для оптимізації таблиці маршрутизації. Його правильне налаштування дозволяє ефективно управляти трафіком, особливо в мережах із доступом до інтернету або обмеженою кількістю сусідів.

Незалежно від того, чи використовується IPv4, чи IPv6, принципи роботи маршруту за замовчуванням залишаються подібними, хоча синтаксис команд дещо відрізняється.

14.4. Призначення та роль протоколів динамічної маршрутизації

У сучасних мережах, які складаються з багатьох підмереж, управління трафіком через статичні маршрути може стати надто складним і ресурсозатратним завданням. Щоб з’єднати всі вузли в такій мережі, адміністратору доводиться вручну налаштовувати маршрути до кожної окремої підмережі, а також постійно оновлювати конфігурацію у випадку змін. Наприклад, при додаванні нової підмережі або відмові лінії зв'язку, потрібно змінювати записи в таблиці маршрутизації на декількох пристроях. Це не тільки забирає час, але й підвищує ризик помилок. Рішенням є застосування динамічних протоколів маршрутизації, які автоматично виявляють зміни в мережі та оновлюють маршрути без участі адміністратора. Такі протоколи значно покращують масштабованість мережі та зменшують навантаження на технічний персонал.

Динамічні протоколи почали активно застосовуватися з кінця 1980-х років. Одним із перших був протокол RIP (Routing Information Protocol), заснований на алгоритмах, які ще в 1969 році використовувалися в мережі ARPANET – попередниці сучасного Інтернету. Версія RIPv1, що з’явилася у 1988 році, мала обмежену функціональність: вона не підтримувала маску підмережі та була непридатна для складних сучасних топологій.

Щоб адаптувати протокол до нових вимог, згодом було

розроблено RIPv2, який додав підтримку VLSM (змінної довжини маски підмережі) та більш безпечний обмін інформацією. Але навіть ця версія виявилася недостатньо масштабованою для великих корпоративних мереж.

Для великих і складних мереж були створені нові, більш ефективні протоколи.

OSPF (Open Shortest Path First) – відкритий протокол, який працює за алгоритмом найкоротшого шляху (*Dijkstra*) і підходить для ієрархічних мереж з великою кількістю маршрутизаторів. Наприклад, велика університетська мережа може використовувати OSPF для ефективного розподілу трафіку між кампусами.

IS-IS (Intermediate System to Intermediate System) – ще один протокол, схожий на OSPF, але частіше використовується в провайдерських або магістральних мережах.

EIGRP (Enhanced Interior Gateway Routing Protocol) – протокол розроблений компанією Cisco, поєднує переваги протоколів відстань-вектор і стан каналу. Підходить для великих корпоративних мереж, де важлива швидка адаптація до змін у топології.

BGP (Border Gateway Protocol) – це єдиний протокол, який використовується для маршрутизації між різними автономними системами, тобто між провайдерами або великими компаніями. Наприклад, якщо Google та Amazon мають різні мережі, вони обмінюються маршрутами через BGP.

Динамічні протоколи маршрутизації – це сукупність алгоритмів, структур даних і типів повідомлень, що дозволяють маршрутизаторам взаємодіяти один з одним у режимі реального часу. Основні функції цих протоколів:

Наприклад, маршрутизатор A дізнається від маршрутизатора B, що існує мережа 10.0.2.0/24, і додає її у свою таблицю маршрутизації.

При зміні мережі, наприклад, через заміну обладнання або зміну топології, маршрутизатори автоматично обмінюються актуальними даними.

За допомогою різних метрик (наприклад, кількість переходів, затримка, пропускна здатність) протоколи вибирають найкращий шлях.

Якщо з ладу виходить певний канал, протокол миттєво шукає інші можливі шляхи, щоб зберегти доступність мережі.

Динамічні протоколи складаються з кількох ключових компонентів.

Наприклад, маршрутизатор зберігає інформацію в таблицях маршрутизації, базах даних сусідів або топології. Усе це розміщується в оперативній пам’яті.

Для взаємодії маршрутизатори надсилають пакети певного формату – *hello*-повідомлення, оновлення маршрутів, підтвердження тощо.

Кожен протокол має свій алгоритм для вибору найкращого маршруту – наприклад, алгоритм Беллмана-Форда (у RIP) або алгоритм Дейкстри (у OSPF).

До головних переваг динамічної маршрутизації належать:

- автоматичне оновлення маршрутів;
- висока гнучкість і адаптивність;
- менше ручного втручання;
- здатність до масштабування.

Наприклад, у великій компанії з десятками філій, підключених через VPN, протокол OSPF дозволяє автоматично оновлювати маршрути без необхідності вручну змінювати конфігурації на кожному пристрої.

Проте динамічна маршрутизація має і недоліки:

- споживає ресурси маршрутизатора – процесорний час, оперативну пам’ять, і мережеву смугу;
- має нижчий рівень безпеки – пакети з оновленням маршрутів можуть бути перехоплені або підроблені, якщо не вжито додаткових заходів (наприклад, аутентифікації);
- налаштування складніші, ніж у випадку статичної маршрутизації, і вимагають глибших знань з боку адміністратора.

У невеликих мережах, де структура стабільна й не змінюється часто, доцільно використовувати статичну маршрутизацію – вона проста, безпечна та не витрачає ресурси. У середніх мережах можна застосувати гібридний підхід: використовувати статичні маршрути для критичних вузлів і динамічні для решти. У великих і динамічних мережах перевагу слід надавати динамічній маршрутизації.

Динамічні протоколи маршрутизації є важливою складовою

масштабованих і стійких мереж. Вони дають змогу адаптуватися до змін у реальному часі, обирати найоптимальніші шляхи та мінімізувати адміністративне навантаження. Хоча вони потребують складнішої конфігурації й певних системних ресурсів, у більшості випадків їх використання повністю виправдане.

14.5. Принципи роботи протоколів динамічної маршрутизації

Усі протоколи динамічної маршрутизації були створені з головною метою – автоматично виявляти віддалені мережі та швидко адаптуватися до змін у топології мережі. Це особливо важливо в складних, багаторівневих мережах, де ручне оновлення маршрутизаторів було б надзвичайно трудомістким.

Наприклад, якщо до великої корпоративної мережі додається нова філія з власною підмережею, завдяки протоколу динамічної маршрутизації ця зміна буде автоматично врахована всіма маршрутизаторами без втручання адміністратора.

Загалом робота динамічного протоколу маршрутизації проходить у кілька основних етапів. Спочатку маршрутизатор передає і приймає спеціальні службові повідомлення через свої активні інтерфейси. Ці повідомлення містять інформацію про відомі маршрути або зміни в мережі. Далі маршрутизатор поширює ці дані серед інших пристроїв, які використовують той самий протокол – наприклад, OSPF або EIGRP. Таким чином маршрутизатори обмінюються маршрутною інформацією, щоб будувати повну картину топології мережі.

Коли в мережі стається зміна – наприклад, вихід з ладу каналу або поява нової мережі – маршрутизатор, який її виявив, негайно повідомляє інші маршрутизатори. Протокол маршрутизації оновлює свою базу даних, а нова інформація передається далі по мережі, щоб усі пристрої синхронізували свої таблиці маршрутизації. Це дозволяє швидко знайти альтернативні шляхи доставки трафіку.

На момент увімкнення маршрутизатор ще не має уявлення про структуру мережі, до якої він підключений. Він лише знає свою початкову конфігурацію, яка зберігається у незалежній пам’яті NVRAM. Після завантаження операційної системи маршрутизатор

починає аналізувати свої інтерфейси і виявляти безпосередньо підключені до них мережі. Наприклад, якщо інтерфейс FastEthernet 0/0 має IP-адресу з підмережі 192.168.1.0/24, ця мережа одразу буде додана в таблицю маршрутизації.

Якщо на маршрутизаторі активовано протокол динамічної маршрутизації, наступним кроком є ініціація обміну маршрутною інформацією з іншими маршрутизаторами. Пристрій починає надсилати повідомлення оновлення по всіх активних інтерфейсах. У цих повідомленнях міститься поточна інформація з таблиці маршрутизації – спочатку це лише безпосередньо підключені мережі.

Одночасно маршрутизатор приймає аналогічні оновлення від сусідніх пристроїв. Якщо в отриманому повідомленні містяться дані про мережу, якої ще немає в таблиці, маршрутизатор додає її. Наприклад, маршрутизатор R1, отримавши оновлення від R2, може дізнатися про існування мережі 10.2.0.0/16 і записати її з відповідним шляхом. Такий процес поширення маршрутної інформації продовжується до моменту, поки всі маршрутизатори не будуть мати повну ідентичну інформацію.

Після кількох циклів обміну маршрутами кожен маршрутизатор має повні дані про топологію, включаючи як свої локальні мережі, так і віддалені. Щоб уникнути зациклення маршрутів, у дистанційно-векторних протоколах (таких як RIP) використовується метод розділення горизонту (split horizon). Наприклад, якщо маршрутизатор R2 дізнався про мережу 10.1.0.0 через інтерфейс Serial 0/0/0, він не буде передавати інформацію про цю мережу назад у тому ж напрямку.

Після завершення обміну маршрутизатори вважаються такими, що досягли збіжності. Це означає, що всі маршрутизатори мають однакову і актуальну інформацію про мережу. Тепер кожен з них може використовувати свої таблиці маршрутизації для визначення найкращого шляху до будь-якої IP-адреси в межах мережі. Наприклад, якщо користувач з мережі 192.168.1.0/24 звертається до сервера в 172.16.0.0/16, маршрутизатор автоматично вибирає оптимальний шлях, враховуючи поточні умови мережі.

Збіжність – це процес, під час якого маршрутизатори обмінюються інформацією, обчислюють маршрути та синхронізують таблиці маршрутизації. Важливо розуміти, що досягнення повної збіжності є критичним для стабільної роботи мережі.

Поки збіжність не завершена, маршрути можуть бути недоступні, або трафік може спрямовуватись по застарілих шляхах. Тому одним з ключових показників ефективності протоколу є *час збіжності*.

Протоколи можна оцінювати за швидкістю і надійністю збіжності. Чим швидше маршрутизатори приходять до спільної картини мережі, тим кращим вважається протокол. Наприклад, застарілі протоколи, такі як RIP, мають повільну збіжність і працюють на основі періодичних оновлень (кожні 30 секунд), що затримує реакцію на збої. У той час як сучасні протоколи OSPF або EIGRP реагують на зміни майже миттєво, використовуючи подієво-орієнтовану модель обміну повідомленнями.

Таким чином, завдяки динамічній маршрутизації маршрутизатори здатні адаптуватися до змін у мережі без втручання адміністратора, забезпечуючи оптимальні маршрути, швидке відновлення після збоїв і підтримку складних топологій.

Цей підхід особливо цінний у великих підприємствах, де зміни в мережевій інфраструктурі відбуваються регулярно.

14.6. Класифікація протоколів динамічної маршрутизації

Протоколи маршрутизації поділяються на кілька типів залежно від їхніх функцій, принципів роботи, способу обробки маршрутної інформації та області застосування. Така класифікація дає змогу обирати найбільш придатний протокол під конкретні умови мережі.

	Interior Gateway Protocols				Exterior Gateway Protocols
	Distance Vector		Link-State		Path Vector
IPv4	RIPv2	EIGRP	OSPFv2	IS-IS	BGP-4
IPv6	RIPng	EIGRP for IPv6	OSPFv3	IS-IS for IPv6	BGP-MP

Рис. 14.1. Класифікація протоколів динамічної маршрутизації

IGP (Interior Gateway Protocol) – протоколи для внутрішньої маршрутизації, що працюють в межах однієї автономної системи. Наприклад, компанія використовує OSPF або EIGRP у своїй корпоративній мережі для ефективного обміну маршрутами між офісами.

EGP (Exterior Gateway Protocol) – протоколи для зовнішньої маршрутизації між різними автономними системами. Основним прикладом є BGP, що використовується між провайдерами або великими організаціями.

Протоколи маршрутизації можна розділити на дистанційно-векторні та за станом каналу.

Дистанційно-векторні (Distance Vector) – передають всю таблицю маршрутів своїм сусідам, не маючи повного уявлення про мережу. Прикладом є RIPv2. Такий підхід простий, але має повільну збіжність.

За станом каналу (Link-State) – маршрутизатори обмінюються інформацією про свої безпосередні підключення і формують повну карту мережі. OSPF та IS-IS належать до цього типу.

За вектором маршруту (Path Vector) – використовують інформацію про весь шлях до призначення. BGP є основним представником цього класу і використовується в Інтернеті.

Класові протоколи (Classful) – не передають інформацію про маску підмережі. Наприклад, RIPv1 і IGRP. Вони базуються на класах IP-адрес (A, B, C), що нині вважаються застарілими.

Безкласові протоколи (Classless) – передають разом з маршрутом і маску підмережі, що дозволяє використовувати VLSM (змінну довжину маски) та CIDR (агрегацію маршрутів). Це такі протоколи, як RIPv2, EIGRP, OSPF, IS-IS і BGP.

RIPv1 – класовий, дистанційно-векторний IGP. Простий, але не підтримує VLSM.

RIPv2 – безкласовий, дистанційно-векторний IGP з підтримкою VLSM.

EIGRP – безкласовий гібридний IGP, поєднує риси векторного і канального підходів. Розроблений Cisco.

OSPF – безкласовий IGP за станом каналу. Підходить для великих мереж з високими вимогами до масштабованості.

IS-IS – безкласовий IGP, схожий на OSPF, але має іншу архітектуру.

BGP – єдиний стандартний протокол зовнішньої маршрутизації (EGP), який працює по вектору маршруту.

Автономна система (AS) – це сукупність мереж, що керуються єдиною адміністрацією.

Наприклад, корпоративна мережа великої компанії або мережа інтернет-провайдера. В межах Інтернету кожна AS має свій унікальний номер і виконує маршрутизацію трафіку або всередині себе, або між собою.

Внутрішня маршрутизація (IGP) використовується в межах однієї AS.

Зовнішня маршрутизація (EGP) потрібна для обміну маршрутами між різними AS. Тут використовується тільки BGP, тому термін "протокол зовнішньої маршрутизації" часто просто замінюють на "BGP".

У класових протоколах, таких як RIPv1 і IGRP, маршрутизатори не передають маску підмережі. Це означає, що вони не можуть:

- працювати з VLSM;
- агрегувати маршрути за допомогою CIDR;
- розпізнавати “розірвані” мережі (наприклад, підмережі, які не з’єднані безпосередньо і мають іншу мережу між собою).

У сучасних мережах, де IP-адреси виділяються гнучко, такі протоколи непридатні. Сьогодні використовуються тільки безкласові протоколи, які передають і маску, і префікс.

У протоколах IPv6, класова адресація взагалі не застосовується – усі маршрутизаційні протоколи IPv6 безкласові за своєю природою.

Швидкість збіжності (convergence) визначає, як швидко маршрутизатори обмінюються інформацією і синхронізують таблиці після змін у мережі. Протокол з високою швидкістю збіжності, наприклад OSPF, здатний швидше реагувати на відмови або додавання нових маршрутів.

Наприклад, RIP має повільну збіжність, що може спричинити петлі маршрутизації в динамічних мережах.

Масштабованість показує, наскільки добре протокол працює в великих мережах. OSPF та BGP добре масштабуються завдяки ієрархічній структурі, тоді як RIP підходить лише для малих мереж.

Підтримка VLSM (безкласовість)

Протоколи, які підтримують VLSM, дозволяють гнучко розподіляти IP-адреси, оптимізуючи їхнє використання. Наприклад, EIGRP дозволяє мати мережі з різними масками в межах одного домену маршрутизації.

Споживання ресурсів включає використання оперативної пам’яті, процесора, і пропускної здатності. *Link-state* протоколи, такі як OSPF, споживають більше ресурсів, оскільки будують повну карту мережі. *Distance-vector* протоколи, наприклад RIPv2, менш вимогливі, але менш ефективні. Прості протоколи, такі як RIP, легко налаштовуються, але мають обмежену функціональність. OSPF потребує глибших знань, але забезпечує гнучкість і масштабованість.

Метрики маршрутизації

Коли маршрутизатор має кілька маршрутів до одного пункту призначення, він повинен вибрати найкращий шлях. Це здійснюється за допомогою метрик – числових значень, які визначають вартість маршруту. Наприклад, RIP використовує кількість переходів (*hops*), EIGRP – складну формулу, яка враховує пропускну здатність, затримку та інші параметри, OSPF – використовує "вартість" інтерфейсу, яка зазвичай базується на пропускній здатності.

Метрики унікальні для кожного протоколу і не переносяться між ними. Саме тому два різні протоколи можуть обрати різні маршрути до тієї самої мережі.

14.7. Дистанційно-векторні протоколи маршрутизації

Термін "дистанційно-векторний" (англ. *distance-vector*) описує принцип роботи певного класу протоколів маршрутизації, у такому підході кожен маршрутизатор визначає маршрут до мережі призначення, базуючись лише на інформації, отриманій від сусідів – інших маршрутизаторів, з якими він безпосередньо пов'язаний. Ця інформація включає *дистанцію* і *вектор*.

Дистанція (відстань) – це числове значення, що відображає "вартість" шляху до мережі призначення.

Найчастіше використовуються такі метрики, як *кількість переходів (hops), пропускна здатність лінії, затримка передачі пакетів, надійність каналу, навантаження* тощо. Наприклад, у протоколі RIP використовується лише кількість переходів, а у EIGRP – комбінована метрика на основі кількох показників.

Вектор – це напрямок, у якому слід надсилати пакети, щоб дістатися до мережі призначення. Зазвичай це означає IP-адресу

наступного маршрутизатора або локальний інтерфейс, через який доступна віддалена мережа.

До дистанційно-векторних протоколів внутрішньої маршрутизації IPv4 відносяться наступні протоколи.

RIPv1 (Routing Information Protocol, версія 1) – найперша реалізація протоколу маршрутизації, яка не підтримує маску підмережі, а значить – не здатна працювати з VLSM, є застарілою і не рекомендованою до використання.

RIPv2 – удосконалена версія RIP, що підтримує VLSM, CIDR, а також групову розсилку оновлень, проста в налаштуванні та підходить для невеликих мереж.

IGRP (Interior Gateway Routing Protocol) – власний протокол компанії Cisco, який використовував комбіновану метрику, також є застарілим.

EIGRP (Enhanced Interior Gateway Routing Protocol) – більш досконалий протокол, що поєднує риси дистанційно-векторних та протоколів за станом каналу, підтримує швидку збіжність, VLSM, багатопоточність і високий рівень оптимізації.

Усі ці протоколи здійснюють обмін маршрутною інформацією лише з безпосередньо підключеними сусідами. Наприклад, якщо маршрутизатор має три підключені маршрутизатори-сусіди, то саме з ними він обмінюється таблицями маршрутів. Цей процес не передбачає повного розуміння топології всієї мережі – кожен пристрій "бачить" лише своїх сусідів і доступні через них мережі.

У деяких протоколах, таких як RIP, оновлення маршрутизації відбуваються регулярно – кожні 30 секунд, незалежно від змін у мережі. Ці оновлення надсилаються у вигляді широкомовних повідомлень (broadcast) на адресу 255.255.255.255, що змушує всі пристрої в мережі їх обробляти. Це створює навантаження на мережу та обчислювальні ресурси.

Більш сучасні протоколи, наприклад RIPv2 та EIGRP, використовують групову розсилку (multicast).

Наприклад, RIPv2 надсилає оновлення на адресу 224.0.0.9, що дозволяє отримати їх лише тим пристроям, які дійсно беруть участь у протоколі. EIGRP може навіть надсилати оновлення лише конкретному маршрутизатору, що робить обмін більш ефективним. До того ж, EIGRP

відправляє оновлення лише у випадку змін, а не за фіксованим інтервалом.

У серці кожного маршрутизуючого протоколу лежить алгоритм, що відповідає за:

- відправлення й прийом оновлень;
- розрахунок найкращих маршрутів;
- реагування на зміни в мережевій топології.

Для прикладу, RIP використовує алгоритм Беллмана-Форда, який дозволяє визначати найкоротший шлях до мережі на основі кількості переходів.

IGRP та EIGRP базуються на алгоритмі *DUAL (Diffusing Update Algorithm)*, розробленому доктором Хосе Гарсія-Луна-Асевесом.

DUAL дозволяє зберігати резервні маршрути та забезпечує швидке перемикання у разі збоїв.

RIP (Routing Information Protocol) – протокол маршрутизації першого покоління, вперше описаний у RFC 1058, його перевага – простота конфігурації, що робить його корисним для невеликих мереж. Проте RIP має обмеження:

- широкомовна розсилка оновлень кожні 30 секунд;
- максимальна кількість переходів – 15; мережа, яка знаходиться на відстані 16 або більше переходів, вважається недоступною;
- усі оновлення відправляються незалежно від змін у мережі.

У 1993 році з'явився RIPv2, який підтримує:

- VLSM і CIDR (маску підмережі в оновленнях), що дозволяє економніше використовувати адресний простір;
- групову розсилку на адресу 224.0.0.9, що знижує навантаження;
- підтримку аутентифікації, яка забезпечує захист від несанкціонованих оновлень;
- можливість агрегування маршрутів, що дозволяє скоротити обсяг таблиць.

RIP працює через UDP, порт 520, що робить його простим в реалізації, RIPng – використовується для IPv6. У 1997 році з’явилась версія RIP для IPv6 – RIPng, що базується на RIPv2, але адаптована до особливостей IPv6. Як і попередні версії, вона обмежена 15 переходами та має адміністративну дистанцію 120.

IGRP, створений Cisco у 1984 році, використовував складену метрику, яка включала:

- пропускну здатність,
- затримку,
- надійність,
- завантаженість.

IGRP передавав оновлення кожні 90 секунд через широкомовну розсилку. Проте через обмеження був згодом витіснений протоколом EIGRP.

EIGRP був представлений у 1992 році як заміна IGRP. Його ключові переваги:

- підтримка VLSM, CIDR та кількох протоколів (IPv4, IPv6, IPX, AppleTalk);
- подієві оновлення – інформація надсилається лише при зміні топології;
- Hello-повідомлення для перевірки доступності сусідів;
- зберігання всіх маршрутів у таблиці топології, що дозволяє мати резервні маршрути;
- миттєва збіжність завдяки DUAL;
- мінімальне використання ресурсів.

14.8. Протоколи маршрутизації за станом каналу

Протоколи маршрутизації за станом каналу (Link-State Routing Protocols) будують повну карту мережі, що дозволяє кожному маршрутизатору самостійно обчислювати найкоротший шлях до всіх вузлів. Найвідомішими протоколами цього типу є OSPF (Open Shortest Path First) та IS-IS (Intermediate System to Intermediate System).

Збирання інформації про безпосередніх сусідів

Маршрутизатори використовують спеціальні повідомлення (наприклад, Hello в OSPF), щоб виявити сусідів і встановити з ними зв’язок.

Оголошення про стан каналу

LSA – Link-State Advertisement – кожен маршрутизатор створює повідомлення LSA, яке містить інформацію про його з’єднання та метрики.

Поширення інформації про топологію

LSA розсилаються по всій області (area) з використанням надійної доставки (flooding). Усі маршрутизатори в області зберігають однакову топологічну базу даних.

Обчислення маршрутів

Використовується алгоритм Дейкстри (Shortest Path First, SPF), який знаходить найкоротший шлях до всіх мереж.

OSPF (Open Shortest Path First) – це відкритий стандарт (розроблений IETF), який працює на рівні 3 моделі OSI.

Особливості OSPF:

- підтримка VLSM, CIDR, аутентифікації;
- розділення на області (Areas) – покращує масштабованість.

Типи маршрутизаторів в OSPF:

- Internal Router – усі інтерфейси в межах однієї області;
- Backbone Router – маршрутизатор у області 0 (основна);
- Area Border Router (ABR) – з'єднує різні області;
- Autonomous System Boundary Router (ASBR) – з'єднує OSPF з іншими протоколами.

Типи пакетів OSPF:

- Hello – встановлення сусідства;
- DBD (Database Description) – обмін описом бази даних;
- LSR (Link State Request) – запит до сусіда про відсутню LSA;
- LSU (Link State Update) – надсилання LSA;
- LSAck – підтвердження отримання LSA.

Типи LSA:

- LSA типу 1 – про маршрутизатор;
- LSA типу 2 – про DR (Designated Router) в широкомовному сегменті;
- LSA типу 3 – міжобласні маршрути (створюється ABR);
- LSA типу 4 – описує ASBR;
- LSA типу 5 – зовнішні маршрути з інших протоколів.

Наприклад, маршрутизатор R1 в області 0 виявляє нову мережу 192.168.2.0/24, створює відповідну LSA і поширює її всім сусідам. Інші маршрутизатори оновлюють свою базу і обчислюють новий найкоротший шлях.

IS-IS (Intermediate System to Intermediate System) – протокол, розроблений ISO, працює на рівні 2 OSI (мережеві служби CLNP). Основні відмінності між IS-IS та OSPF полягають у тому, що IS-IS є незалежним від IP протоколом, працює безпосередньо на рівні 2 моделі OSI, має простішу ієрархію рівнів (Level 1 і Level 2) та краще масштабується, тоді як OSPF є IP-залежним, працює на рівні 3 та використовує поділ на області (Areas) з окремим маршрутизатором ядра (табл. 14.2).

Таблиця 14.2. **Порівняння протоколів OSPF та IS-IS**

Характеристика	OSPF	IS-IS
Рівень OSI	3	2
Стандартизація	IETF (відкритий)	ISO
Формат адрес	IP	CLNP (може містити IP)
Топологія	Areas з backbone (Area 0)	Levels 1, 2
Придатність до мереж	Корпоративні мережі	Великі, операторські мережі
Розширення	OSPFv3 для IPv6	IS-IS natively supports IPv6
Залежність від IP	Висока	Низька

В TCP/IP-середовищах IS-IS використовує IPv4 та IPv6 через розширення. Не залежить від IP, гнучкий у підтримці інших протоколів. Швидкий у великих мережах (наприклад, у провайдерів).

Level 1 (Intra-area) – маршрутизація в межах області.

Level 2 (Inter-area) – маршрутизація між областями.

Level 1-2 маршрутизатори – аналог ABR в OSPF.

Розглянемо типи повідомлень IS-IS.

- *Hello* – встановлення сусідства.
- *LSP (Link State PDU)* – містить LSA.
- *CSNP (Complete Sequence Number PDU)* – контроль повноти бази.
- *PSNP (Partial SNP)* – запит або підтвердження окремих LSP.

Розглянемо переваги IS-IS:

- менше залежить від IP;
- стабільність і масштабованість у великих мережах;
- добре працює у мультипротокольному середовищі (IPv4, IPv6,

MPLS).

Переваги протоколів маршрутизації за станом каналу:

- у разі змін в мережі маршрутизатори швидко перебудовують таблицю маршрутів;
- повна картина мережі дозволяє кращу оптимізацію маршрутів;
- ієрархія дозволяє масштабування;
- підтримка аутентифікації.

Недоліки протоколів маршрутизації за станом каналу:

- складність налаштування та супроводу;
- більше споживання ресурсів (оперативна пам'ять, CPU);
- початкове навантаження при побудові топології.

Протоколи за станом каналу (OSPF, IS-IS) – це потужні механізми маршрутизації, які забезпечують швидке реагування на зміни в мережі та точну маршрутизацію. Вибір між OSPF і IS-IS залежить від типу мережі, вимог до масштабованості, сумісності та обслуговування.

Сьогодні серед протоколів динамічної маршрутизації найчастіше використовуються OSPF (Open Shortest Path First), EIGRP (Enhanced Interior Gateway Routing Protocol), IS-IS (Intermediate System to Intermediate System) та BGP (Border Gateway Protocol). Кожен з них має свої переваги й сферу застосування залежно від типу мережі, масштабу, необхідної гнучкості та вимог до конвергенції.

BGP – основний протокол для маршрутизації між автономними системами в Інтернеті, його використовують провайдери, дата-центри та великі компанії з глобальними мережами. Хоча BGP не призначений для швидкої конвергенції, його здатність до складної політики маршрутизації робить його незамінним у глобальній маршрутизації.

У сучасних мережах вибір протоколу часто залежить від розміру мережі, сумісності з обладнанням, потреб у масштабуванні та вимог до керованості й безпеки.

Контрольні запитання та задачі до теми 14

Перелік теоретичних запитань

1. Що таке маршрутизація в комп'ютерних мережах?
2. Які основні відмінності між статичною та динамічною маршрутизацією?

3. У яких випадках доцільно використовувати статичну маршрутизацію?

4. Які переваги та недоліки має статична маршрутизація?

5. Як налаштовується статичний маршрут на маршрутизаторі Cisco?

6. Що таке маршрут за замовчуванням і коли його використовують?

7. Як налаштувати маршрут за замовчуванням у Cisco IOS?

8. Які основні функції протоколів динамічної маршрутизації?

9. У чому полягає принцип роботи динамічного маршрутизатора?

10. Як відбувається обмін маршрутною інформацією між маршрутизаторами?

11. Як класифікуються динамічні протоколи маршрутизації?

12. Що таке дистанційно-векторні протоколи? Наведіть приклади.

13. Як працює RIP? Які його переваги та обмеження?

14. Що таке протоколи маршрутизації за станом каналу? Назвіть приклади.

15. У чому полягають переваги OSPF над RIP?

Перелік задач для самостійного опрацювання

Практичне завдання 1. Налаштування статичної маршрутизації.

У середовищі моделювання Cisco Packet Tracer потрібно забезпечити взаємозв’язок трьох локальних мереж через маршрутизатори з використанням статичних маршрутів.

Топологія мережі: 3 маршрутизатори (R1, R2, R3), кожен з підключеною локальною мережею. Налаштувати IP-адресацію. Створити статичні маршрути на кожному маршрутизаторі для досягнення всіх мереж.

Перевірити зв'язок за допомогою команди *ping*.

Практичне завдання 2. Маршрут за замовчуванням.

У середовищі моделювання Cisco Packet Tracer потрібно реалізувати маршрут за замовчуванням для доступу до Інтернету.

Топологія мережі: два маршрутизатори (R1 – внутрішня мережа, R2 – інтернет-шлюз).

Налаштувати маршрут за замовчуванням на R1 (ip route 0.0.0.0

0.0.0.0 <gateway>).

Перевірити, що з локальної мережі є доступ до хостів "в Інтернеті".

Практичне завдання 3. Протокол маршрутизації RIP.

У середовищі моделювання Cisco Packet Tracer потрібно налаштувати динамічну маршрутизацію за допомогою RIP.

Топологія мережі: три маршрутизатори, з'єднані в лінію.

Використати команду *router rip* і оголосити всі мережі.

Перевірити таблиці маршрутизації (*show ip route*) та зв'язок.

Тестове завдання 4.

Що є основною перевагою динамічної маршрутизації над статичною?

а) Простіша конфігурація

б) Краща безпека

в) Автоматичне оновлення маршрутів

г) Вища швидкість передачі

Тестове завдання 4.

Яка команда використовується для налаштування статичного маршруту в маршрутизаторах Cisco?

а) ip static route

б) ip routing static

в) ip route

г) set static route

Перелік використаної та рекомендованої літератури

14.1. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

14.2. Routing Information Protocol [Електронний ресурс]. – Режим доступу: https://www.cisco.com/site/us/en/learn/topics/networking/what-is-routing.html

14.3. OSPF Version 2 [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc2328

14.4. OSI IS-IS Intra-domain Routing Protocol [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc1142

14.5. IP Routing [Електронний ресурс]. – Режим доступу: https://www.cisco.com/c/en/us/tech/ip/ip-routing/index.html

Тема 15. Налаштування OSPF маршрутизації

OSPF (Open Shortest Path First) – це один із найпоширеніших протоколів динамічної маршрутизації, який належить до категорії протоколів за станом каналу (link-state), забезпечує швидку, надійну та масштабовану маршрутизацію в мережах середнього і великого розміру. Завдяки використанню алгоритму Дейкстри (Shortest Path First), OSPF обчислює найоптимальніший маршрут до кожної мережі в автономній системі.

На відміну від дистанційно-векторних протоколів, OSPF не надсилає всю таблицю маршрутизації, а обмінюється інформацією про стан безпосередньо підключених мереж. Це зменшує навантаження на мережу та прискорює процес збіжності – досягнення узгодженого бачення маршрутизованої мережі між усіма вузлами.

У цій темі розглядаються основні принципи налаштування OSPF у середовищі Cisco, включаючи створення OSPF-процесу маршрутизації; оголошення мереж; призначення ідентифікаторів маршрутизаторів (Router ID); розділення на області (areas), зокрема використання backbone area (area 0); перевірку та усунення несправностей у конфігурації OSPF. Знання принципів і практики налаштування OSPF є ключовим для фахівців з мережевих технологій, оскільки цей протокол часто використовується в корпоративних та провайдерських мережах.

15.1. Розвиток та характеристики протоколу OSPF

Протокол OSPF є прикладом протоколу маршрутизації за станом каналу (link-state), створеного як вдосконалена альтернатива раннім протоколам динамічної маршрутизації, зокрема, RIP (Routing Information Protocol). У той час як RIP був одним із перших широко використовуваних протоколів, його обмеження швидко стали очевидними в міру зростання складності комп'ютерних мереж.

Одним з основних недоліків RIP було використання кількості переходів (*hop count*) як єдиної метрики для обчислення найкращого маршруту. Наприклад, маршрут із меншою кількістю переходів, але через повільні або перевантажені канали зв'язку, вважався кращим за швидший шлях із більшою кількістю переходів. Такий підхід негативно впливав на ефективність і масштабованість мереж, особливо коли йшлося про великі мережеві інфраструктури з різнорідними каналами.

На відміну від RIP, OSPF забезпечує набагато точніший аналіз мережевої топології. Він використовує алгоритм Дейкстри (SPF – Shortest Path First) для побудови дерева найкоротших шляхів від маршрутизатора до кожної доступної мережі. Завдяки цьому OSPF здатний швидко реагувати на зміни в мережі, а його збіжність (тобто час досягнення консенсусу між маршрутизаторами) є значно швидшою.

Розробку OSPF розпочала робоча група IETF (Інженерна група з розвитку Інтернету) у 1987 році у відповідь на потребу в надійнішому протоколі маршрутизації для великих IP-мереж. На той час Інтернет використовувався переважно науковими установами та дослідницькими центрами, і існуючі рішення не задовольняли вимоги до масштабованості та надійності.

У 1989 році з'явилася перша специфікація – OSPFv1, опублікована в RFC 1131. Було реалізовано дві версії: одна – для маршрутизаторів, інша – для UNIX-систем, що згодом трансформувалася у сервіс GATED, який тривалий час використовувався в UNIX-середовищах. Проте OSPFv1 залишався експериментальним і не здобув широкого впровадження.

Ситуація змінилася у 1991 році, коли Джон Мой представив OSPFv2 – значно покращену версію протоколу. Вона була стандартизована у 1998 році в RFC 2328, що залишається актуальним і до сьогодні. Для підтримки IPv6 у 1999 році було створено OSPFv3, який надалі був оновлений у RFC 5340 (2008). OSPFv3 адаптований до нових вимог мережевої адресації, включаючи роботу з IPv6.

OSPF є безкласовим протоколом маршрутизації, що означає підтримку *CIDR (Classless Inter-Domain Routing)* та *VLSM (Variable Length Subnet Masking)*. Це дозволяє створювати гнучкі схеми адресації, в яких мережі можуть мати маски різної довжини – наприклад, мережа 192.168.1.0/24 може співіснувати з підмережею 192.168.1.128/25 без конфлікту.

Одна з ключових особливостей OSPF – це можливість ієрархічної організації мережі шляхом поділу її на області (areas). Центральною областю завжди *є area 0 (backbone area)*.

Наприклад, у великій компанії можна створити окремі області для кожного офісу або філії, які підключаються до центральної області для оптимізації маршрутизації.

Розглянемо основні властивості протоколу OSPF.

- Підтримка VLSM та CIDR дає змогу оптимізувати використання IP-простору.
- Замість регулярної передачі повних таблиць маршрутизації (як у RIP), OSPF надсилає лише оновлення про зміни (LSA – Link State Advertisements), що значно знижує навантаження на мережу.
- У разі змін у топології (наприклад, вихід з ладу маршрутизатора) OSPF оперативно перебудовує маршрути, зазвичай протягом кількох секунд.
- Підтримка областей дозволяє будувати великі й складні мережі без втрати ефективності.

OSPF підтримує аутентифікацію повідомлень маршрутизації, зокрема з використанням MD5-хешування, що унеможливлює прийняття фальшивих оновлень від зловмисників.

У мережевих пристроях кожному джерелу маршруту присвоюється адміністративна дистанція (Administrative Distance, AD) – числове значення, що визначає надійність джерела.

Для OSPF це значення дорівнює 110, тоді як, наприклад, для RIP – 120, а для статичних маршрутів – 1. Тобто, у разі конфлікту між маршрутами з однаковою ціллю, пристрій обере той, що надійшов від протоколу з меншою AD.

15.2. Компоненти та принцип роботи протоколу OSPF

Незалежно від типу чи призначення, усі динамічні протоколи маршрутизації мають спільні структурні елементи. Вони використовують спеціалізовані повідомлення для обміну інформацією про доступні маршрути в мережі. Ці повідомлення забезпечують маршрутизаторам можливість збирати, оновлювати та обробляти дані про мережу. У процесі роботи вони створюють структури даних, які обробляються за допомогою алгоритмів маршрутизації для прийняття рішень щодо пересилання пакетів.

У протоколі OSPF дані про мережу зберігаються у трьох ключових базах даних, кожна з яких відіграє свою роль у процесі маршрутизації.

База суміжності (Adjacency Database) – містить інформацію про сусідні маршрутизатори, з якими було встановлено логічне з’єднання.

Наприклад, якщо маршрутизатор R1 має фізичне підключення до R2 і обидва використовують OSPF, вони спочатку визначають одне одного за допомогою "hello"-пакетів і потім створюють запис у БД суміжності.

База даних про стан каналів (LSDB – Link State Database) – зберігає детальний опис топології мережі. Вона наповнюється інформацією з оголошень про стан каналу LSA, які передаються всіма маршрутизаторами в області.

База пересилання (Routing Table) – використовується для створення таблиці маршрутизації, яка містить найкращі шляхи до всіх доступних мереж.

На її основі маршрутизатор приймає рішення, куди відправляти IP-пакети.

Всі ці таблиці зберігаються в оперативній пам’яті маршрутизатора і регулярно оновлюються на основі отриманих повідомлень.

OSPF використовує п’ять типів спеціалізованих повідомлень (пакетів) для обміну інформацією між маршрутизаторами:

Hello-пакети – використовуються для виявлення сусідів. Наприклад, кожні 10 секунд маршрутизатор відправляє *hello*-пакет, щоб побачити, хто ще знаходиться в локальній мережі й працює за OSPF.

Database Description (DBD) – пакети, що передають опис вмісту LSDB, використовуються для синхронізації інформації між маршрутизаторами.

Link State Request (LSR) – запити про відсутні записи у БД, якщо після обміну DBD-пакетами маршрутизатор виявив нестачу інформації.

Link State Update (LSU) – пакети, що містять фактичні оголошення про стан каналів (LSA), які маршрутизатори передають своїм сусідам.

Link State Acknowledgment (LSAck) – пакети підтвердження отримання LSA, необхідні для надійності передачі.

Ці повідомлення забезпечують повноцінне й точне бачення мережі для кожного маршрутизатора.

Алгоритм SPF – основа OSPF

Після того, як маршрутизатор зібрав повну інформацію про мережу, він використовує алгоритм SPF (Shortest Path First), відомий також як алгоритм Дейкстри, для обчислення найкоротших маршрутів.

Розглянемо, як працює цей алгоритм.

1. Вважаємо маршрутизатор коренем дерева.
2. Обчислюємо вартість шляху до кожного іншого вузла (інших маршрутизаторів або мереж), враховуючи метрики OSPF (наприклад, пропускну здатність каналу).
3. Побудовується SPF-дерево, де кожен шлях є найоптимальнішим.
4. На основі цього дерева заповнюється таблиця маршрутизації з найефективнішими маршрутами.

Наприклад, якщо R1 має два шляхи до мережі 192.168.10.0/24: через R2 (вартість 10) і через R3 (вартість 20), то в таблиці маршрутизації буде вибраний шлях через R2.

Щоб досягти стану збіжності – коли всі маршрутизатори мають однакове уявлення про топологію – OSPF виконує кілька етапів.

Встановлення суміжності

Кожен маршрутизатор розсилає hello-пакети на всі інтерфейси, де увімкнено OSPF. Після виявлення інших маршрутизаторів встановлюється стійке з'єднання, або суміжність (adjacency).

Обмін оголошеннями про стан каналів (LSA)

Після встановлення суміжності маршрутизатори обмінюються LSA-повідомленнями, у яких міститься інформація про всі підключені інтерфейси, IP-адреси, метрики тощо. LSA розповсюджуються всією мережею за принципом "затоплення" (flooding).

Побудова бази даних топології (LSDB)

Коли маршрутизатор отримав усі LSA, він створює повну карту мережі – базу даних про стан каналів. Це дозволяє візуалізувати, які маршрутизатори підключені один до одного і як.

Виконання SPF-алгоритму

На основі LSDB виконується розрахунок SPF-дерева для визначення найкоротших маршрутів до кожної мережі. Результат цього обчислення використовується для формування таблиці маршрутизації, яка вже застосовується для реального пересилання пакетів.

Цей механізм робить OSPF одним з найнадійніших і найефективніших протоколів внутрішньої маршрутизації, який чудово масштабується, швидко реагує на зміни і зберігає стабільність мережі в будь-яких умовах.

15.3. OSPF для однієї та декількох областей

Протокол OSPF (Open Shortest Path First) підтримує ієрархічну структуру маршрутизації, яка забезпечує кращу масштабованість і ефективність в обробці маршрутизованих даних у великих мережах. Основною ідеєю цієї архітектури є розподіл мережі на логічні сегменти, що називаються областями (areas).

Кожна область включає групу маршрутизаторів, які підтримують однаковий набір даних про топологію LSDB, тобто мають єдине бачення внутрішньої структури своєї області. Це дозволяє зменшити обсяг обробки інформації на кожному маршрутизаторі.

OSPF з однією областю (single-area OSPF) – у цьому випадку всі маршрутизатори знаходяться в єдиній логічній області, яка зазвичай має номер 0 (називається *нульовою або магістральною областю*). Це спрощена структура, яка підходить для невеликих або середніх мереж. Наприклад, офісна мережа з кількома маршрутизаторами без складної структури може працювати в межах однієї області.

OSPF з кількома областями (multi-area OSPF) – використовується в великих або географічно розподілених мережах. У цьому випадку мережу ділять на кілька областей, кожна з яких повинна мати з'єднання з магістральною областю (областю 0). Зв'язок між областями забезпечують пограничні маршрутизатори (ABR – Area Border Routers).

Наприклад, якщо компанія має філії в різних містах, кожну філію можна винести в окрему область, а центральний офіс – в область 0.

Використання кількох областей дозволяє розділити велику автономну систему (AS) на менші логічні частини, що підвищує ефективність. Основна ідея – локалізувати зміни та обчислення в межах області, не поширюючи їх на всю мережу. Наприклад, якщо в області 1 змінився стан з'єднання (канал вийшов з ладу), тільки маршрутизатори в цій області виконують повторний розрахунок SPF, у той час як інші області не змінюють свою базу топології, а просто оновлюють записи в таблицях маршрутизації.

Це означає, що ресурси центрального процесора використовуються раціонально: зміни не змушують усю мережу перераховувати шляхи, лише відповідну частину. Алгоритм SPF, який є обчислювально складним, застосовується локально, що зменшує час на перебудову таблиці маршрутизації та знижує загальне навантаження.

Уявімо, що в мережі з трьома областями (0, 1, 2) в області 2 відбувається зміна – маршрутизатор втрачає з'єднання з іншим вузлом. У такому разі тільки маршрутизатори області 2 перебудовують свої LSDB і запускають SPF-алгоритм. Інші області (0 і 1) отримують лише коротке резюме у вигляді агрегованої інформації, яка оновлює їхню таблицю маршрутизації, але не вимагає запуску SPF. Таким чином, вплив обмежується лише тією областю, де сталася зміна.

Зменшення навантаження на ЦП і обсяг баз даних

У великих мережах, якщо всі маршрутизатори перебувають в одній області, LSDB може стати надмірно великою. Чим більше записів у базі стану каналу – тим більше ресурсів споживає маршрутизатор, особливо під час перерахунку топології. Тому поділ на області дозволяє розбити великі бази даних на менші частини, що оптимізує обсяг пам'яті, спрощує адміністрування і скорочує час реагування на зміни.

Переваги багатозонної структури OSPF

Ієрархічна архітектура OSPF забезпечує кілька ключових переваг.

- Менші таблиці маршрутизації – завдяки тому, що адреси з інших областей можуть агрегуватися, кількість записів у таблиці маршрутизації значно зменшується. Наприклад, кілька підмереж можна представити однією зведеною адресою (192.168.0.0/16 замість 192.168.1.0/24, 192.168.2.0/24 і т.д.).
- Менше навантаження на ЦП і пам'ять – оновлення LSA не передаються між областями повністю, тому зменшується обсяг трафіку і не всі маршрутизатори виконують повторні обчислення.
- Локалізація змін і зниження частоти запуску SPF – зміни в топології ізольовані в межах області, завдяки чому відбувається швидша стабілізація мережі та скорочується обсяг розсилки LSA-повідомлень.

15.4. Інкапсуляція та типи пакетів OSPF

Протокол OSPF є протоколом внутрішньої динамічної маршрутизації, який використовує власну структуру пакетів для обміну інформацією між маршрутизаторами. Ці пакети передаються через мережу Ethernet і містять декілька рівнів заголовків. Починається усе з заголовка Ethernet-кадру, де вказується MAC-адреса призначення. Для групових (multicast) повідомлень OSPF використовуються MAC-адреси типу 01-00-5E-00-00-05 (для загальних повідомлень) або 01-00-5E-00-

00-06 (для призначених лише певним маршрутизаторам, таких як DR і BDR).

Далі йде заголовок IP-пакета, у якому важливим є поле «протокол», що має значення 89 – це вказує, що пакет належить до OSPF. Також у цьому заголовку зазначається групова IP-адреса. Для широкомовлення OSPF-повідомлень маршрутизатори використовують IP-адресу 224.0.0.5 (усі маршрутизатори OSPF), а для повідомлень, які призначено тільки DR та BDR – 224.0.0.6.

Наступним рівнем є заголовок пакету OSPF, де зазначається тип пакету (наприклад, hello, DBD, LSU тощо), а також ідентифікатор маршрутизатора, що його надіслав, і ідентифікатор області, в якій цей пакет згенеровано. Це дозволяє однозначно визначити джерело інформації та її контекст у межах ієрархічної архітектури OSPF.

У залежності від типу OSPF-пакету в останній частині розміщуються дані, специфічні для кожного типу.

Наприклад, для пакету типу 1 (*hello*) це можуть бути параметри суміжності, а для пакету типу 4 (LSU) – структура LSA (Link-State Advertisement). Усі ці пакети дозволяють OSPF встановлювати та підтримувати зв’язки між маршрутизаторами та динамічно обмінюватися інформацією про зміни в мережі.

OSPF використовує п’ять основних типів пакетів, які разом реалізують весь процес маршрутизації за станом каналу. Розглянемо їх по черзі.

Hello-пакети (тип 1) – це перші пакети, які маршрутизатори надсилають для виявлення сусідів у мережі. Вони допомагають встановити стосунки суміжності між маршрутизаторами. Наприклад, якщо в мережі з’явився новий маршрутизатор, він надсилає *hello*-пакет до адреси 224.0.0.5, і якщо інші маршрутизатори відповідають з тими ж параметрами, між ними встановлюється сусідство.

DBD-пакети (Database Description, тип 2) – передають короткий зміст (заголовки) бази даних стану каналу, якою володіє маршрутизатор. Наприклад, коли два маршрутизатори встановили зв’язок, вони звіряють свої бази даних за допомогою цих DBD-пакетів.

LSR-пакети (Link-State Request, тип 3) – використовуються маршрутизатором для запиту детальної інформації про ті записи, які є в DBD-пакеті сусіда, але відсутні або застарілі в локальній БД.

LSU-пакети (Link-State Update, тип 4) – це пакети, що містять оновлення мережевої топології, у вигляді LSA. Наприклад, якщо у мережі з'явилася нова мережа або маршрут, маршрутизатор надсилає LSU до своїх сусідів, аби повідомити їх про зміну.

LSAck-пакети (Link-State Acknowledgment, тип 5) – слугують підтвердженням отримання LSA-повідомлень. Наприклад, після отримання LSU маршрутизатор надсилає LSAck, щоб повідомити відправника, що оновлення успішно отримане.

Найважливішим типом для початкового встановлення зв'язку є *hello*-пакет. Він виконує дві основні функції: по-перше, виявляє суміжні маршрутизатори в межах однієї підмережі, а по-друге, гарантує, що параметри маршрутизаторів (часові інтервали, ідентифікатори тощо) узгоджені. Якщо будь-який з параметрів не збігається – наприклад, інтервал *hello* – суміжність не встановлюється.

Hello-пакет включає в себе кілька важливих полів.

Тип пакету – встановлює тип OSPF-повідомлення, наприклад, 1 для hello, 2 для DBD.

Ідентифікатор маршрутизатора – 32-бітове значення у форматі IPv4, наприклад, 192.168.1.1.

Ідентифікатор області – дозволяє визначити, у межах якої області відбувається обмін.

Маска підмережі – потрібна для перевірки, чи маршрутизатори знаходяться в одній мережі.

Інтервал hello – зазвичай 10 секунд. Наприклад, якщо один маршрутизатор надсилає hello-пакети кожні 10 секунд, а інший – кожні 20, то вони не стануть сусідами.

Пріоритет маршрутизатора – значення від 0 до 255, використовується при виборі DR (Designated Router) і BDR (Backup DR). Пріоритет 0 означає, що маршрутизатор ніколи не буде DR.

Dead-інтервал – час, після якого маршрутизатор вважає сусіда недоступним, якщо не отримує від нього *hello*. Зазвичай дорівнює 4×hello, тобто 40 секунд.

Ідентифікатори DR і BDR – якщо такі є, маршрутизатор вказує їх у *hello*-пакеті, щоб сусіди знали, з ким встановлювати з'єднання.

Список сусідів – вказує, з якими маршрутизаторами вже встановлено стосунки суміжності.

На прикладі великої локальної мережі, де є десятки маршрутизаторів, *hello*-пакети дозволяють OSPF-мережі самостійно визначити, хто буде DR, а хто – BDR, і хто з ким буде обмінюватися топологією безпосередньо. Це дозволяє мінімізувати кількість трафіку і централізувати обмін даними через виділені маршрутизатори.

Таким чином, структура та функції пакетів OSPF є основою ефективного обміну маршрутизованою інформацією. Від *hello* до LSAck – кожен тип пакету має свою чітко визначену роль у процесі виявлення, узгодження і обміну маршрутизованими даними між пристроями.

15.5. Встановлення суміжності з сусідніми пристроями

Коли на інтерфейсі маршрутизатора активується протокол динамічної маршрутизації OSPF, першим кроком є встановлення зв'язку з іншими пристроями, що також підтримують OSPF у цій самій мережі.

Для цього маршрутизатор розсилає спеціальні пакети привітання (*hello packets*) з усіх активних інтерфейсів, на яких увімкнено OSPF.

Ці пакети містять у собі ключову інформацію, зокрема унікальний ідентифікатор маршрутизатора – OSPF Router ID. Цей ідентифікатор, як правило, є IP-адресою, яка дозволяє однозначно розпізнати кожен маршрутизатор у межах конкретної області OSPF.

Наприклад, якщо маршрутизатор має IP-адресу 192.168.1.1, ця адреса може виступати його ідентифікатором у процесах OSPF.

Коли сусідній маршрутизатор, який також працює під управлінням OSPF, отримує пакет привітання від іншого пристрою, чий ідентифікатор ще не включений до його списку відомих сусідів, він ініціює процес *встановлення відносин суміжності (adjacency)*.

Цей процес є критично важливим для обміну інформацією про маршрути – тільки суміжні маршрутизатори обмінюються докладними даними про топологію мережі. У звичайному режимі маршрутизатори продовжують надсилати один одному пакети *Hello* через регулярні інтервали, щоб підтримувати стабільність зв’язку та швидко виявляти неполадки або зміни.

У випадку мереж з множинним доступом, таких як Ethernet, де декілька маршрутизаторів підключені до одного загального сегменту, протокол OSPF стикається з рядом проблем.

Однією з головних проблем є те, що кожен маршрутизатор потенційно може встановити суміжність з усіма іншими маршрутизаторами в межах мережі.

У результаті, при наявності *n* маршрутизаторів кількість можливих суміжних з'єднань досягає *n(n - 1)/2*.

Наприклад, у мережі з 10 маршрутизаторами може виникнути до 45 суміжностей. Це створює надмірне навантаження на мережу, оскільки кожен маршрутизатор змушений обмінюватися інформацією про маршрути з усіма іншими пристроями.

Ще однією серйозною проблемою є "лавинна" розсилка *LSA (Link-State Advertisement)* – спеціальних повідомлень, які інформують про зміни в топології мережі.

Кожного разу при ініціалізації OSPF або при виявленні змін у конфігурації, маршрутизатори починають розсилати свої LSA всім своїм сусідам, що в умовах множинного доступу може призвести до надмірного спаму в мережі й суттєвого зниження продуктивності.

Щоб уникнути цих проблем, протокол OSPF передбачає механізм вибору виділеного маршрутизатора (Designated Router, DR). У межах кожної мережі з множинним доступом DR виконує роль центру координації обміну інформацією.

Усі інші маршрутизатори надсилають свої LSA не безпосередньо кожному з сусідів, а лише DR, який потім розсилає агреговану інформацію далі. Це значно зменшує трафік і кількість суміжностей у мережі.

Для забезпечення відмовостійкості обирається також резервний виділений маршрутизатор (Backup Designated Router, BDR), який бере на себе функції DR у разі його виходу з ладу. Інші маршрутизатори, що не є ні DR, ні BDR, отримують статус DROther і взаємодіють лише з DR і BDR, а не між собою.

Цей підхід дозволяє зберігати ефективність і масштабованість OSPF у складних мережах з великою кількістю пристроїв, мінімізуючи навантаження та підвищуючи стабільність маршрутизації.

15.6. Синхронізація баз даних OSPF

Після досягнення стану Two-Way, коли два маршрутизатори підтвердили взаємну наявність один одного шляхом обміну пакетами

Hello, протокол OSPF переходить до наступного етапу – синхронізації бази даних про стан каналів (LSDB – Link State Database). На цьому етапі маршрутизатори встановлюють більш тісні зв’язки (adjacency) з вибраними сусідами (зокрема з DR та BDR у мережах з множинним доступом), щоб забезпечити повну відповідність топологічної інформації між ними. Пакети *Hello* більше не є основним інструментом обміну даними. Замість них у гру вступають інші типи пакетів OSPF – DD (Database Description), LSR (Link State Request), LSU (Link State Update) та LSAck (Link State Acknowledgement) – які забезпечують синхронізацію інформації про мережу.

На стадії ExStart два маршрутизатори домовляються, хто з них буде головним (*master*), а хто підлеглим (*slave*) у подальшому обміні даними. Це необхідно для упорядкування передачі пакетів DD, які описують вміст бази даних про стан каналів. Як правило, маршрутизатор із вищим Router ID (ідентифікатором OSPF) стає провідним. Наприклад, якщо два маршрутизатори мають ID 10.1.1.1 та 10.2.2.2, то провідним буде останній.

Після цього, у стані Exchange, маршрутизатори обмінюються DD-пакетами, які вміщують зведення про всі LSA, відомі кожному з них. На основі цього зведення маршрутизатори визначають, яких саме LSA їм не вистачає або які з них є застарілими. Для отримання відсутньої інформації вони надсилають LSR-пакети, вказуючи які саме LSA їм потрібні. Коли маршрутизатор отримує відповідь у вигляді LSU-пакета на всі свої запити LSR, і ця відповідь підтверджується LSAck-пакетом, обидва маршрутизатори вважаються повністю синхронізованими.

У такому разі їхній стан переходить у *Full* – найвищий рівень сусідських відносин в OSPF. У цьому стані маршрутизатори мають однакову копію бази даних LSDB, що гарантує правильну побудову маршрутів. Після досягнення стану *Full* маршрутизатори продовжують надсилати один одному пакети Hello з певним інтервалом (наприклад, кожні 10 секунд у мережах Ethernet), щоб підтверджувати "живість" з'єднання. Якщо *Hello*-пакети не надходять у встановлений проміжок часу (зазвичай 40 секунд – таймер *Dead Interval*), маршрутизатор вважає сусіда недоступним, і відбувається повторна синхронізація або реконструкція маршруту.

Після початкової синхронізації топологічної бази даних подальші LSU-пакети надсилаються лише у двох випадках:

- коли в мережі зафіксовані зміни топології, наприклад, нове з'єднання, відмова інтерфейсу або зміна вартості маршруту (cost); тоді генеруються інкрементні оновлення, тобто лише ті LSA, які зазнали змін;
- коли спливає таймер *refresh* – стандартно кожні 30 хвилин (1800 секунд), навіть якщо змін не було, це потрібно для забезпечення актуальності та стабільності бази даних.

Цей механізм дозволяє OSPF працювати дуже ефективно, особливо у великих мережах, де важливо мінімізувати обсяг переданих даних та гарантувати цілісність інформації про мережу.

15.7. Налаштування процесу OSPF

Для того щоб визначити, які інтерфейси маршрутизатора братимуть участь у маршрутизації в межах певної області OSPF, використовується команда *network*, яка дозволяє вказати діапазон IP-адрес, до яких належать інтерфейси, що повинні підтримувати протокол OSPF. Після активації цієї команди відповідні інтерфейси стають здатними надсилати й приймати OSPF-пакети. Це також означає, що підмережа, до якої належить інтерфейс, буде включена в таблицю маршрутизації OSPF і розповсюджена серед інших маршрутизаторів у домені.

Наприклад, якщо потрібно активувати OSPF на інтерфейсі з IP-адресою 192.168.10.1/24, команда виглядатиме так:

network 192.168.10.0 0.0.0.255 area 0.

Тут 0.0.0.255 – це шаблонна маска (wildcard-mask), яка вказує, які біти адреси аналізуються при пошуку збігів.

Команда має наступний базовий синтаксис:

network [мережева-адреса] [шаблонна-маска] area [ідентифікатор-області].

Ідентифікатор області (area-id) є логічною ознакою приналежності маршруту до певного сегмента в OSPF. Для простих конфігурацій зазвичай використовується область з номером 0 – вона є головною або «кістяковою» (backbone) у архітектурі OSPF. Це спрощує майбутнє масштабування мережі та її поділ на декілька областей.

OSPF є безкласовим протоколом маршрутизації, що означає: йому не потрібно враховувати клас IP-адреси, але завжди необхідно вказати шаблонну маску. Ця маска – це двійковий шаблон, який показує, які біти IP-адреси мають бути точним збігом, а які можна ігнорувати.

Шаблонна маска розраховується як обернене значення до звичайної маски підмережі. Наприклад, для маски 255.255.255.0 шаблонною буде 0.0.0.255, тобто 255.255.255.255 мінус 255.255.255.0.

У новіших версіях Cisco IOS з'явилася можливість вказувати звичайну маску підмережі замість шаблонної – система самостійно виконує перетворення. Це спрощує налаштування, особливо для менш досвідчених адміністраторів. Проте, з точки зору безпеки і продуктивності, OSPF-повідомлення повинні передаватися лише через ті інтерфейси, які безпосередньо підключені до інших пристроїв з активованим OSPF. Інакше можна втратити продуктивність або ненавмисно розкрити структуру мережі стороннім вузлам.

OSPF використовує метрику під назвою *«вартість» (cost)* для визначення найоптимальнішого маршруту.

Вартість маршруту залежить від пропускної спроможності інтерфейсу: чим вона більша – тим менше значення вартості. Наприклад, інтерфейс Ethernet із пропускною здатністю 10 Мбіт/с має вищу вартість, ніж інтерфейс FastEthernet із 100 Мбіт/с.

Формула розрахунку стандартної вартості така:

вартість = 100000000 / пропускна_здатність_інтерфейсу

Отже, інтерфейс із пропускною здатністю 100 Мбіт/с матиме вартість:

100000000 / 100000000 = 1.

Однак, виникає проблема, коли пропускна здатність інтерфейсу перевищує 100 Мбіт/с (наприклад, 1 Гбіт/с або 10 Гбіт/с). У такому разі всі такі інтерфейси отримують однакову вартість – 1, що унеможливлює точну диференціацію шляхів. Щоб усунути цю проблему, потрібно вручну змінити еталонну пропускну здатність (reference bandwidth), використовуючи команду:

auto-cost reference-bandwidth [значення_в_Мбіт/с]

Наприклад:

для 1 Гбіт/с: auto-cost reference-bandwidth 1000

для 10 Гбіт/с: auto-cost reference-bandwidth 10000

Цю команду слід налаштувати на всіх маршрутизаторах у домені OSPF, щоб забезпечити узгодженість обчислень.

Також існує можливість явно вказати вартість для кожного інтерфейсу за допомогою команди:

ip ospf cost [число].

Це особливо корисно в гетерогенних мережах, де використовуються маршрутизатори різних виробників (наприклад, Cisco, MikroTik, Juniper), які можуть по-різному обчислювати вартість. Явне зазначення вартості гарантує однакову інтерпретацію в OSPF-домені незалежно від платформи.

15.8. Перевірка даних процесу OSPF

Для моніторингу стану протоколу OSPF на маршрутизаторі використовується команда *show ip ospf*. Вона надає загальну інформацію про конфігурацію процесу OSPF, зокрема відображає *ідентифікатор процесу (process ID)* – локальне значення, яке асоціюється з екземпляром OSPF на маршрутизаторі, а також ідентифікатор маршрутизатора (router ID). Окрім цього, команда показує, які області OSPF налаштовані, а також час останнього запуску алгоритму SPF (Shortest Path First). Це дає змогу зрозуміти, коли востаннє мережа перебудовувалася, що може бути корисно для діагностики нестабільних мереж або частих змін топології.

Приклад:

Router# show ip ospf

У відповідь буде виведена інформація на кшталт:

Routing Process "ospf 1" with ID 1.1.1.1
Supports only single TOS(TOS0) routes
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs

Для швидкої перевірки налаштувань на рівні конкретних інтерфейсів OSPF доцільно скористатися командою *show ip ospf interface*. Ця команда відображає розширену інформацію щодо кожного інтерфейсу, який бере участь у OSPF-маршрутизації. Вона показує, до якої області належить інтерфейс, який пріоритет має, встановлену вартість (cost), таймери Hello/Dead, тип мережі (broadcast, point-to-point тощо), а також наявність сусідів. Така деталізація особливо корисна для

перевірки правильності використання команд *network* при конфігурації маршрутизатора.

Приклад:

Router# show ip ospf interface GigabitEthernet0/0

Якщо потрібно переглянути стиснуту інформацію про всі інтерфейси, що задіяні в OSPF, рекомендується команда *show ip ospf interface brief*. Вона виводить короткий звіт про всі активні інтерфейси OSPF, зазначаючи їхній стан (up/down), значення cost, до якої області належать, а також чи є на інтерфейсі OSPF-сусіди.

Це зручний інструмент для швидкого огляду загального стану OSPF на маршрутизаторі, без надмірної деталізації.

Приклад:

Router# show ip ospf interface brief

Ще однією важливою командою є *show ip protocols*, яка використовується для перегляду зведеної інформації про всі активні протоколи маршрутизації, у тому числі й OSPF.

За допомогою команди *show ip protocols* можна дізнатися:

- які процеси OSPF активні;
- які мережі були оголошені (через команду *network*);
- ідентифікатор маршрутизатора;
- які таймери використано;
- які фільтри чи маршрути застосовано (якщо такі є).

Це корисно для комплексної перевірки налаштувань OSPF, особливо якщо в мережі використовується кілька протоколів маршрутизації.

Для протоколу OSPFv3, який застосовується з адресацією IPv6, конфігурація виконується не в режимі глобального маршрутизатора, як у OSPFv2, а безпосередньо на інтерфейсах. Це суттєва відмінність.

Перед тим як активувати OSPFv3, необхідно переконатися, що на інтерфейсах налаштовані адреси *link-local*, оскільки саме через них здійснюється обмін повідомленнями між сусідніми пристроями.

Крім того, протокол OSPFv3 вимагає активованої одноадресної маршрутизації (unicast routing) для IPv6. Без цього протокол не функціонуватиме. Ще однією обов’язковою умовою є наявність 32-бітного router ID – він повинен бути унікальним у межах домену

OSPFv3. Якщо маршрутизатор його не встановить автоматично, адміністратор повинен вручну вказати його через команду router-id.

Приклад налаштування OSPFv3 для інтерфейсу:

```
Router(config)# ipv6 unicast-routing
Router(config)# interface GigabitEthernet0/0
Router(config-if)# ipv6 ospf 10 area 0
Router(config-if)# ipv6 ospf router-id 2.2.2.2
```

Контрольні запитання та задачі до теми 15

Перелік теоретичних запитань

1. Що таке протокол OSPF і які його основні характеристики?
2. У чому полягає різниця між OSPF та RIP?
3. Яку роль відіграє Dijkstra SPF-алгоритм у роботі OSPF?
4. Що таке область (area) в OSPF і для чого вона використовується?
5. Які типи маршрутизаторів існують в OSPF (ABR, ASBR, DR, BDR)?
6. Як відбувається встановлення суміжності (adjacency) між OSPF-маршрутизаторами?
7. Поясніть механізм *Hello*-пакетів в OSPF.
8. Які типи пакетів використовує OSPF? Опишіть кожен з них.
9. Що таке LSA (Link-State Advertisement) та їх типи?
10. Як відбувається синхронізація баз даних OSPF?
11. Що таке процес OSPF в Cisco IOS? Як його запустити?
12. У чому полягає різниця між внутрішньообласним та міжобласним маршрутом в OSPF?
13. Як виконується інкапсуляція даних у OSPF-пакетах?
14. Які команди використовуються для перевірки стану OSPF-суміжності?
15. Як можна оптимізувати OSPF-мережу з кількома областями?

Перелік задач для самостійного опрацювання

Практичне завдання 1. OSPF в одній області.

У середовищі моделювання Cisco Packet Tracer потрібно налаштувати OSPF в одній області для трьох маршрутизаторів.

Топологія мережі: R1 ---- R2 ---- R3

Присвоїти IP-адреси інтерфейсам. Активувати OSPF (*router ospf 1*)

на всіх пристроях. Налаштувати мережі з маскою та Area 0.

Перевірити сусідство (*show ip ospf neighbor*) та маршрути (*show ip route ospf*).

Практичне завдання 2. OSPF з кількома областями.

У середовищі моделювання Cisco Packet Tracer потрібно реалізувати OSPF із двома областями (area 0 та area 1) з ABR.

Топологія мережі:

R1(Area 0) ---- R2(ABR) ---- R3(Area 1)

R2 має інтерфейси в обох областях.

Налаштувати правильне призначення мереж до відповідних областей. Перевірити LSA у кожному маршрутизаторі. Перевірити маршрутизацію між кінцевими пристроями.

Тестове завдання 3.

1. Який тип пакету використовується для встановлення OSPF-сусідства?

а) LSR

б) Hello

в) LSU

г) DBD

Тестове завдання 4.

Який тип маршрутизатора з'єднує дві області в OSPF?

а) DR

б) ASBR

в) ABR

г) IR

Перелік використаної та рекомендованої літератури

15.1. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

15.2. OSPF Version 2 [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc2328

15.3. OSPF for IPv6 [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc5340

Тема 16. Списки контролю доступу

У сучасних комп'ютерних мережах безпека є одним з найважливіших аспектів проєктування та експлуатації. Одним із базових інструментів контролю доступу в мережах є списки контролю доступу (Access Control Lists, ACL).

ACL дозволяють адміністраторам мереж визначати, які пакети мають право проходити через маршрутизатор або комутатор, а які – ні. З їхньою допомогою можна обмежити доступ до певних ресурсів, реалізувати політики безпеки, відфільтрувати трафік, здійснити контроль за доступом до мережевих сервісів і навіть організувати базову форму міжмережевого екранування.

ACL виконують роль фільтрів, які застосовуються до мережевого трафіку на основі заданих критеріїв – таких як IP-адреса джерела або призначення, номер протоколу, номер порту тощо. Кожен запис у списку контролю доступу визначає одну умову для обробки пакету: дозволити або заборонити його передавання.

ACL поділяються на кілька типів, серед яких найпоширенішими є стандартні, розширені та іменовані списки. Крім того, для роботи з IPv6 існують окремі списки контролю доступу, адаптовані до специфіки цієї версії IP-протоколу.

Застосування ACL має вирішальне значення для побудови надійної, керованої та безпечної мережевої інфраструктури. Наприклад, за допомогою ACL можна заборонити доступ співробітників з певного підрозділу до внутрішніх серверів бухгалтерії, або обмежити доступ до інтернету лише визначеним пристроям.

16.1. Визначення та завдання списків контролю доступу

Фахівці з проєктування та адміністрування мереж часто застосовують *міжмережеві екрани (firewall)* для створення бар'єру між внутрішньою корпоративною мережею та зовнішніми небажаними джерелами. Ці міжмережеві екрани можуть бути реалізовані як окремі апаратні пристрої або як програмні модулі, інтегровані, наприклад, у маршрутизатори. Їхнє головне завдання – обмежити або дозволити мережевий трафік на основі певних правил. Одним із простих способів створення фільтрації трафіку є застосування списків контролю доступу ACL на маршрутизаторі. Такий підхід дозволяє, наприклад, блокувати

доступ до локальної мережі з підозрілих IP-адрес або дозволяти тільки певні протоколи, такі як HTTP і DNS.

На рис. 16.1 показано, що фільтрація пакетів у мережах здійснюється на 3 (мережевому) та 4 (транспортному) рівнях моделі OSI.

Рис. 16.1. Рівні OSI, на яких працює фільтрація пакетів

Списки контролю доступу є послідовністю правил, які вказують маршрутизатору, які пакети потрібно пропускати, а які – відкидати. Кожне правило в ACL аналізує мережевий трафік за певними параметрами: IP-адресами джерела й призначення, портами, типом протоколу та іншими атрибутами. Наприклад, можна створити правило, яке дозволяє доступ до сервера баз даних тільки з IP-адрес, що належать певному підмережевому діапазону, заблокувавши всі інші запити.

Однією з основних причин використання ACL у мережі є підвищення рівня безпеки. Контроль доступу дозволяє захистити критичні ресурси від несанкціонованого доступу, мінімізувати загрозу внутрішніх атак і знизити ризики, пов’язані з помилками конфігурації або зловмисною активністю.

Наприклад, можна заборонити користувачам з Wi-Fi сегменту мережі підключення до адміністративних серверів, які мають бути доступні лише для персоналу IT-відділу.

Технічно ACL реалізується як набір команд у Cisco IOS, які впливають на обробку вхідного або вихідного трафіку на інтерфейсі маршрутизатора. Коли ACL застосовується, маршрутизатор оцінює кожен пакет, що проходить через інтерфейс, і вирішує, чи дозволити йому подальше пересилання, чи заблокувати його. Таким чином, маршрутизатор перестає бути лише пристроєм маршрутизації, а починає виконувати функції базового мережевого захисту.

ACL також виконують низку важливих функцій у мережі.

Оптимізація використання пропускної здатності.

Наприклад, якщо корпоративна політика забороняє перегляд відео з YouTube у робочий час, можна створити ACL, яка блокує трафік до відеосервісів. Це дозволяє зменшити навантаження на інтернет-канал та покращити якість зв'язку для критичних бізнес-застосунків.

Контроль за розповсюдженням оновлень маршрутизації.

ACL дозволяють обмежити обсяг інформації, яка розповсюджується через протоколи маршрутизації, такі як OSPF чи EIGRP. Наприклад, маршрутизатор може бути налаштований так, щоб передавати маршрути тільки до певних підмереж, зменшуючи тим самим зайве навантаження на інші вузли.

Організація доступу на основі ролей або зон мережі.

За допомогою ACL можна дозволити, наприклад, лише працівникам відділу кадрів доступ до HR-сервера, водночас повністю обмеживши цей доступ для інших співробітників. Це дозволяє впровадити модель поділу мережі на зони з різними рівнями довіри (*trusted/untrusted zones*).

Фільтрація за типами трафіку.

ACL можуть бути налаштовані так, щоб дозволяти лише конкретні протоколи. Наприклад, може бути дозволений трафік SMTP і IMAP для електронної пошти, але заборонено Telnet, як небезпечний протокол віддаленого доступу.

Управління доступом до сервісів.

За допомогою ACL можна встановити обмеження доступу до певних типів ресурсів, таких як файлові сервери, вебсервіси або FTP-сховища. Наприклад, користувачам певного VLAN можна дозволити лише HTTP-доступ до зовнішнього вебсайту, але обмежити будь-які інші запити.

Варто зазначити, що за замовчуванням ACL не налаштовані, і маршрутизатор працює за принципом: "весь трафік дозволено", спираючись лише на таблицю маршрутизації. Тільки після активації ACL на конкретному інтерфейсі маршрутизатор починає фільтрувати пакети згідно з правилами доступу.

Крім базового фільтрування, ACL можуть бути застосовані для поглибленого аналізу або обробки трафіку. Наприклад, у QoS (Quality of Service) ACL можуть використовуватись для класифікації трафіку за пріоритетом. Це дозволяє виділяти особливо важливий трафік – наприклад, голосовий або відеоконференції – і забезпечувати йому вищий пріоритет обробки, як у випадку VIP-квитка на подію: такий трафік «обслуговується» швидше й ефективніше, ніж звичайний.

15.2. Сеанс зв’язку та обмін даними по протоколу TCP

Списки контролю доступу ACL є одним із ключових інструментів, який використовують мережеві адміністратори для регулювання *вхідного (inbound)* та *вихідного (outbound)* трафіку на маршрутизаторах і комутаторах. Ці списки дозволяють як просто дозволяти або блокувати трафік, так і здійснювати більш детальне управління передачею даних на основі різних мережевих параметрів. Наприклад, можна створити правило, яке дозволяє доступ лише до певного сервера з конкретної IP-адреси у визначені години або блокувати підключення до певного порту TCP, що часто використовуються в атаках.

Щоб краще зрозуміти, як ACL фільтрує трафік, уявіть типовий процес доступу користувача до вебсторінки. Коли користувач вводить адресу сайту в браузері, його пристрій створює TCP-з’єднання з вебсервером. Протокол IP (Internet Protocol) визначає маршрут, яким трафік буде передаватися між джерелом (комп’ютером користувача) і призначенням (сервером), а протокол TCP (Transmission Control Protocol) відповідає за встановлення стабільного з’єднання і керує порядком передавання сегментів даних між застосунками. Саме в цьому діалозі між двома пристроями ACL може бути застосовано для дозволу або блокування таких з’єднань. Протокол TCP використовується в більшості сучасних мережевих сервісів – від відправки електронної пошти до завантаження файлів. Його функція полягає в тому, щоб розбити потік даних на сегменти, які зручно транспортувати мережею, і

зібрати їх на приймаючій стороні у правильному порядку. TCP гарантує надійну доставку: дані не губляться, не дублюються та не приходять у неправильній послідовності. TCP орієнтований на встановлення з'єднання, тобто перед передачею даних дві сторони встановлюють логічне з'єднання, яке забезпечує контроль за передачею.

Наприклад, якщо комп'ютер ініціює з'єднання з сервером банку, TCP гарантує, що всі передані пакети даних будуть доставлені саме в тому вигляді, в якому були надіслані.

Списки ACL використовують інформацію з TCP/IP-з'єднання для прийняття рішень про фільтрацію трафіку. Це робиться через механізм, який називається *фільтрацією пакетів (packet filtering)*. У процесі фільтрації маршрутизатор аналізує заголовки пакетів, які надходять або виходять з мережевого інтерфейсу, і порівнює ці дані з заданими умовами. Наприклад, можна створити правило, яке блокує всі пакети з IP-адреси 192.168.1.10 або дозволяє тільки трафік з певного TCP-порту, наприклад, 443 (HTTPS). Якщо пакет відповідає хоча б одному правилу, він або дозволяється, або блокується відповідно до цього правила.

Коли маршрутизатор виконує функції фільтрації, він витягує ключову інформацію з кожного пакету – наприклад, IP-адресу джерела, адресу призначення, тип транспортного протоколу (TCP, UDP, ICMP) і, при потребі, порти. Ця інформація порівнюється із записами в ACL. Таким чином, ACL виконує роль списку умов (або правил), які вказують, які пакети дозволити, а які – відхилити.

Наприклад, маршрутизатор може бути налаштований на блокування всього трафіку, що надходить на порт 23 (Telnet), для запобігання несанкціонованому віддаленому доступу.

Кожен ACL складається з набору правил, які називаються записами ACE (Access Control Entries). Кожен запис ACE – це окрема умова, яка описує, який тип трафіку має бути дозволений або заборонений. Наприклад, ACE може виглядати так: *permit tcp 192.168.10.0 0.0.0.255 any eq 80* – цей запис дозволяє всім пристроям у мережі 192.168.10.0/24 доступ до будь-якого ресурсу через TCP-порт 80 (HTTP). Коли трафік проходить через інтерфейс маршрутизатора, система порівнює кожен пакет з ACE-правилами у порядку зверху донизу, поки не знайде перший збіг. Якщо знайдено відповідність – пакет обробляється відповідно до дії (*permit* або *deny*).

Для перевірки трафіку ACL «витягує» з пакету ключову інформацію рівня 3 (мережевого), зокрема:

- IP-адресу відправника (джерела);
- IP-адресу отримувача (призначення);
- тип повідомлення у разі протоколу ICMP.

Якщо ACL налаштовано для більш детальної фільтрації, він також може аналізувати дані рівня 4 (транспортного), включаючи:

- номер порту TCP/UDP джерела;
- номер порту TCP/UDP призначення.

Це дозволяє, наприклад, блокувати лише вхідний FTP-трафік, не впливаючи на HTTP чи HTTPS.

ACL також відіграють важливу роль у безпеці: вони дозволяють не тільки контролювати доступ до певних сегментів мережі, а й запобігати атакам типу «сканування портів», вторгнення, DoS-атакам тощо. Вони можуть бути налаштовані для фільтрації вхідного або вихідного трафіку залежно від того, коли саме потрібно застосувати контроль.

Вхідні ACL (Inbound ACL) застосовуються до трафіку ще до того, як маршрутизатор визначить його подальший маршрут. Це означає, що пакет може бути одразу відкинутий, що дозволяє зменшити навантаження на процес обробки маршрутизації. Наприклад, якщо на інтерфейс надходить небажаний трафік з певної IP-адреси, його можна заблокувати ще до обробки.

Вихідні ACL (Outbound ACL), навпаки, застосовуються після того, як маршрутизатор вже вирішив, куди слід переслати пакет. Такі ACL ефективні, коли необхідно фільтрувати трафік, який виходить з інтерфейсу, незалежно від того, з якого саме інтерфейсу він був отриманий. Наприклад, можна дозволити тільки вебтрафік з локальної мережі до Інтернету через зовнішній інтерфейс маршрутизатора.

Кожен список ACL завжди завершується неявним правилом «*deny all*» – воно не відображається явно, але завжди діє наприкінці списку. Якщо жодне з правил не відповідає пакету, то пакет автоматично буде відкинутий. Тому дуже важливо правильно налаштувати хоча б одне дозволяюче правило, інакше вся передача даних буде заблокована. Це правило забезпечує базову безпеку, але водночас вимагає уважності при конфігурації.

16.3. Стандартні і розширені ACL -списки

У мережевих пристроях Cisco для контролю доступу до ресурсів і регулювання трафіку існують два основні типи списків контролю доступу ACL для протоколу IPv4: стандартні ACL і розширені ACL. Кожен із цих типів має своє призначення та рівень гнучкості у фільтрації трафіку, що дозволяє мережевим адміністраторам ефективно застосовувати відповідні правила залежно від конкретних вимог до безпеки та продуктивності.

Стандартні ACL є базовим типом списків контролю доступу. Вони дозволяють або блокують трафік виключно на основі IPv4-адреси джерела. Тобто стандартний ACL не враховує адресу призначення пакета, протокол, порти чи інші характеристики – рішення про дозвіл або заборону приймається лише за IP-адресою, з якої надходить трафік.

Через таке обмеження ці списки найкраще підходять для простих сценаріїв фільтрації, наприклад, коли потрібно заборонити доступ до певної частини мережі з однієї IP-адреси або групи адрес.

Наприклад, ми хочемо заборонити будь-який трафік від хоста 192.168.1.100 до всієї мережі:

access-list 10 deny 192.168.1.100

access-list 10 permit any

Цей тип ACL створюється в глобальному режимі конфігурації на маршрутизаторі й прив’язується до потрібного інтерфейсу.

На відміну від стандартних, розширені ACL (Extended ACL) пропонують набагато більшу гнучкість. Вони можуть аналізувати й фільтрувати пакети за кількома критеріями одночасно, серед яких:

- тип транспортного або мережевого протоколу (наприклад, IP, ICMP, TCP, UDP);
- IP-адреса джерела;
- IP-адреса призначення;
- порти джерела і призначення для TCP або UDP (наприклад, 80 – HTTP, 443 – HTTPS, 21 – FTP);
- інші параметри, такі як прапорці TCP (наприклад, ACK або SYN), що дає змогу застосовувати ACL у ще більш точкових сценаріях.

Це дозволяє реалізовувати складні політики безпеки та обмежень трафіку.

Наприклад, ми хочемо дозволити лише вебтрафік з підмережі 192.168.10.0/24 до вебсервера 10.0.0.10 (порт 80), блокуючи інші з'єднання.

access-list 101 permit tcp 192.168.10.0 0.0.0.255 host 10.0.0.10 eq 80

access-list 101 deny ip any any

Розширені ACL також створюються в глобальному режимі конфігурації і, як правило, застосовуються ближче до джерела трафіку, що дозволяє зменшити обсяг небажаного трафіку в мережі.

Щоб організувати та ідентифікувати ACL, Cisco IOS дозволяє використовувати два підходи: нумеровані та іменовані списки.

У нумерованому форматі кожен ACL визначається числовим діапазоном, який також вказує на тип списку:

- *стандартні ACL* – номери в діапазоні 1–99 або 1300–1999 (розширення діапазону),
- *розширені ACL* – номери в діапазоні 100–199 або 2000–2699.

Приклад нумерованого ACL:

access-list 100 deny ip any any

Хоча нумеровані ACL є простими у реалізації, вони не дають змоги зрозуміти призначення списку без додаткових коментарів, що ускладнює обслуговування великих конфігурацій.

Саме тому з версії Cisco IOS 11.2 була введена підтримка іменованих ACL. Такий підхід дозволяє давати кожному ACL зрозуміле ім’я, яке описує його функцію або область застосування. Це полегшує адміністрування мережі, особливо в складних середовищах із великою кількістю правил.

Приклад іменованого ACL:

```
ip access-list extended WEB-ACCESS
permit tcp 192.168.10.0 0.0.0.255 host 10.0.0.10 eq 80
deny ip any any
```

Іменовані ACL підтримують додавання, видалення та редагування окремих правил без видалення всього списку, що є особливо корисним при внесенні змін у діючу політику доступу.

16.4. Накладення шаблонної маски

У конфігурації ACL для IPv4 використовується особливий тип маски – так звана *шаблонна маска (Wildcard Mask)*. Це 32-бітна бінарна

послідовність, яка дозволяє маршрутизатору визначити, які біти IP-адреси потрібно враховувати при порівнянні, а які – ігнорувати. Такий механізм потрібен, щоб можна було легко описати як окремі IP-адреси, так і цілі діапазони адрес, до яких застосовуються правила доступу.

У списках ACL для IPv6 не використовуються шаблонні маски. Натомість, у IPv6 діапазон адрес визначається за допомогою довжини префікса, аналогічно до позначення підмереж (наприклад, /64).

Хоча шаблонні маски виглядають схожими на звичайні маски підмережі (наприклад, 255.255.255.0), їхнє призначення та логіка роботи відрізняються. У масках підмереж біти 1 означають частину мережі, яка має бути точною, а біти 0 – частину адреси, що може змінюватись.

У шаблонних масках навпаки:

біт 0 в шаблонній масці означає, що цей біт у адресі має збігатися з відповідним бітом у заданій адресі;

біт 1 означає, що цей біт можна ігнорувати, він не враховується при порівнянні.

Тому шаблонні маски іноді називають *зворотними (інверсними) масками*, оскільки логіка роботи з бітами у них протилежна до звичайних масок підмереж.

Найпростіший спосіб обчислити шаблонну маску – відняти маску підмережі від 255.255.255.255. Розглянемо кілька прикладів.

Приклад 1. Потрібно дозволити доступ до всієї мережі 192.168.3.0/24.

Маска підмережі – 255.255.255.0.

Шаблонна маска: 255.255.255.255 - 255.255.255.0 = 0.0.0.255

Команда:

access-list 10 permit 192.168.3.0 0.0.0.255

Приклад 2. Дозвіл для 14 користувачів у підмережі 192.168.3.32/28.

Маска підмережі – 255.255.255.240

Шаблонна маска: 255.255.255.255 - 255.255.255.240 = 0.0.0.15

Команда:

access-list 10 permit 192.168.3.32 0.0.0.15

Приклад 3. Охоплення двох суміжних мереж – 192.168.10.0 та 192.168.11.0.

Спільний префікс можна охопити маскою підмережі 255.255.252.0,

шаблонна маска: 255.255.255.255 - 255.255.252.0 = 0.0.3.255

Команда:

access-list 10 permit 192.168.8.0 0.0.3.255

Розглянемо приклад конфігурації ACL.

Уявімо, що потрібно дозволити доступ до 16 підмереж – від 192.168.16.0 до 192.168.31.0.

Замість того, щоб прописувати окремі правила для кожної з підмереж, як показано нижче:

access-list 10 permit 192.168.16.0

access-list 10 permit 192.168.17.0

...

access-list 10 permit 192.168.31.0

Можна використати єдину команду з шаблонною маскою:

access-list 10 permit 192.168.16.0 0.0.15.255

Це не лише скорочує конфігурацію, а й знижує навантаження на маршрутизатор при обробці ACL.

Щоб уникнути помилок при ручному введенні шаблонних масок, можна використовувати ключові слова *host* та *any*.

Host – використовується для точного збігу з однією конкретною IP-адресою. За ним автоматично підставляється шаблонна маска 0.0.0.0.

Наприклад:

access-list 10 permit host 192.168.1.5

дорівнює:

access-list 10 permit 192.168.1.5 0.0.0.0

Any – означає “будь-яка IP-адреса”, відповідає шаблонній масці 255.255.255.255.

Наприклад:

access-list 10 permit any

дорівнює:

access-list 10 permit 0.0.0.0 255.255.255.255

Ці ключові слова також підтримуються у списках ACL для IPv6, що робить конфігурацію ще більш універсальною.

У великих мережах конфігурація ACL може стати досить складною.

Наприклад, якщо маршрутизатор має два інтерфейси, налаштовані як для IPv4, так і для IPv6, то для кожного інтерфейсу потрібні:

1 ACL для вхідного трафіку IPv4;

1 ACL для вихідного трафіку IPv4;

1 ACL для вхідного трафіку IPv6;

1 ACL для вихідного трафіку IPv6.

Таким чином, загалом потрібно 8 ACL-списків для повного покриття обох інтерфейсів.

ACL не обов'язково має бути заданим як для вхідного, так і для вихідного трафіку. Все залежить від конкретних вимог доступу.

Крім того, напрямок та номер списку вказуються безпосередньо при застосуванні ACL до інтерфейсу за допомогою команди *ip access-group* або *ipv6 traffic-filter*.

16.5. Рекомендації з використання ACL -списків

Для ефективного застосування списків контролю доступу ACL в мережевій інфраструктурі варто дотримуватися певних практичних рекомендацій.

Розглянемо поради щодо налаштування та розміщення ACL.

Захист на кордоні мережі

Одним з основних місць застосування ACL є міжмережеві екрани (firewalls), розташовані на межі між внутрішньою мережею підприємства та зовнішньою мережею, наприклад, Інтернетом.

На таких маршрутизаторах ACL дозволяє обмежити доступ до внутрішніх ресурсів лише певним IP-адресам або типам трафіку. Наприклад, можна дозволити лише вебтрафік (порт 80 або 443) з певного діапазону зовнішніх IP-адрес.

Сегментація внутрішньої мережі

Якщо в організації є декілька логічних або фізичних сегментів, ACL можуть використовуватися на маршрутизаторах між цими частинами мережі. Це дозволяє обмежити доступ, наприклад, відділу маркетингу до серверів фінансового відділу.

Фільтрація на периферійних пристроях

Рекомендується застосовувати ACL на маршрутизаторах, розташованих на периферії мережі (edge routers), які розділяють мережу

на зони з різними рівнями довіри або з різною політикою безпеки. Це створює додатковий рівень ізоляції для захисту чутливих зон мережі.

ACL для кожного протоколу

Необхідно створювати ACL окремо для кожного протоколу, що використовується на інтерфейсі маршрутизатора. Наприклад, один ACL може регулювати лише IPv4-трафік, інший – IPv6 або ICMP-запити.

Принцип "один до одного" при конфігурації ACL

Існує правило "три одиниці", яке визначає логіку застосування ACL на маршрутизаторах.

Один ACL на один протокол

Для кожного типу протоколу (наприклад, IP, ICMP, TCP) має бути окремий ACL. Це дозволяє точно контролювати різні види трафіку.

Один ACL на один напрямок

ACL працює лише в одному напрямку – або вхідному (inbound), або вихідному (outbound). Наприклад, ACL на інтерфейсі GigabitEthernet 0/1 може контролювати лише трафік, що заходить в інтерфейс, і не впливає на вихідний трафік.

Один ACL на один інтерфейс

На кожен інтерфейс маршрутизатора можна прив'язати окремий ACL. Це дозволяє гнучко керувати політиками безпеки на різних мережевих сегментах.

Планування та точне розміщення ACL

Перед тим, як створювати ACL, важливо скласти чіткий план, який описує мету, типи трафіку, які потрібно контролювати, і місце застосування фільтрації. Неправильно налаштовані ACL можуть призвести до відмови в обслуговуванні, блокування легітимного трафіку, зниження продуктивності або навіть втрати зв'язку між мережевими вузлами.

Наприклад, якщо компанія хоче заборонити перегляд відео в офісі, варто фільтрувати трафік за TCP-портами, які використовують відеосервіси (наприклад, YouTube – 443/80, Netflix – 443), і застосувати ACL максимально близько до користувача.

Розміщення ACL

Розміщення списків контролю доступу ACL безпосередньо впливає на ефективність мережі.

Розглянемо основні правила розміщення ACL.

Розширені ACL – розміщувати ближче до джерела: ці списки мають можливість фільтрувати трафік за кількома параметрами (джерело, призначення, порти, протоколи). Їх розміщують якомога ближче до джерела, щоб негайно відкидати небажаний трафік і не пропускати його через мережеву інфраструктуру.

Наприклад, якщо сервери в сегменті 192.168.1.0/24 не повинні отримувати Telnet-запити, ACL слід розмістити на маршрутизаторі, що обслуговує відправників цих запитів.

Стандартні ACL – розміщувати ближче до місця призначення: оскільки вони можуть фільтрувати лише за IP-адресою джерела, їх ефективніше розміщувати ближче до місця призначення, щоб мінімізувати ризик блокування іншого важливого трафіку.

Додаткові чинники розміщення ACL

Вибір місця застосування ACL може залежати також від таких факторів:

- якщо адміністратор не має повного контролю над усіма маршрутизаторами, розміщення ACL обмежується доступними пристроями;
- у мережах з обмеженою пропускною здатністю важливо знижувати об’єм трафіку ще на рівні джерела, щоб уникнути перевантаження.

Іноді вигідніше створити один простий ACL на маршрутизаторі біля місця призначення, ніж кілька розширених ACL на кожному джерелі. Це спрощує конфігурацію, але не завжди ефективно з точки зору навантаження. Маршрутизатор аналізує вхідний трафік відповідно до списку Access Control Entries (ACE) – записів у ACL – послідовно, зверху вниз. Як тільки відбувається перше співпадіння з правилом, решта записів не розглядається.

У кінці кожного ACL за замовчуванням знаходиться неявне правило *deny all*, яке блокує весь трафік, що не відповідає жодному з правил вище. Тому до списку обов'язково має входити хоча б одне правило *permit*, інакше весь трафік буде заблоковано.

Наприклад, якщо ACL містить лише *deny tcp any any*, маршрутизатор відкине всі TCP-з'єднання.

Щоб дозволити, наприклад, лише трафік до сервера HTTP, варто додати *permit tcp any host 192.168.0.10 eq 80*.

16.6. Налаштування стандартних ACL -списків

Для реалізації контролю доступу за допомогою стандартних нумерованих ACL на маршрутизаторах Cisco необхідно виконати два основні кроки: спочатку створити сам список правил доступу, а потім застосувати його до потрібного інтерфейсу маршрутизатора.

Створення стандартного ACL здійснюється в режимі глобальної конфігурації за допомогою команди *access-list*, яка вказує номер ACL у межах діапазону від 1 до 99.

Починаючи з версії Cisco IOS 12.0.1, було додано розширений діапазон номерів для стандартних ACL – від 1300 до 1999, що дає змогу створити до 798 додаткових ACL.

Синтаксис створення ACL:

Router(config)# access-list <номер> {permit | deny | remark} <джерело> [маска_узагальнення] [log]

При цьому дозволяється або забороняється трафік з конкретної IP-адреси або діапазону адрес.

Приклади:

Дозвіл трафіку від окремого хоста з IP-адресою:

R1(config)# access-list 10 permit host 192.168.10.10

Дозвіл трафіку з цілої мережі (наприклад, 192.168.10.0/24):

R1(config)# access-list 10 permit 192.168.10.0 0.0.0.255

Щоб повністю видалити створений ACL, використовується команда:

R1(config)# no access-list 10

Для перегляду всіх активних ACL на пристрої можна скористатися:

R1# show access-lists

Прив'язка ACL до інтерфейсу

Після створення ACL його потрібно прикріпити до відповідного інтерфейсу маршрутизатора. Це здійснюється в режимі конфігурації інтерфейсу за допомогою команди:

Router(config-if)# ip access-group <номер або ім'я ACL> {in | out}

in – застосування правил до вхідного трафіку;

out – застосування до вихідного.

Видалити ACL з інтерфейсу можна так:

Router(config-if)# no ip access-group <номер>

Використання іменованих стандартних ACL

На відміну від нумерованих ACL, іменовані ACL дозволяють використовувати зрозумілі імена замість номерів. Це значно покращує читабельність конфігурації, особливо у великих мережах.

Наприклад, список, який блокує FTP-з'єднання, можна назвати NO_FTP.

Створення іменованого ACL

Вхід у режим конфігурації ACL:

Router(config)# ip access-list standard NO_FTP

Ім’я має бути унікальним, враховується регістр літер (напр., No_FTP ≠ NO_FTP).

Додавання правил:

Router(config-std-nacl)# deny host 192.168.1.100

Router(config-std-nacl)# permit any

Застосування до інтерфейсу:

Router(config-if)# ip access-group NO_FTP in

Коментарі в ACL: ключове слово remark

Для зручності адміністрування до списку можна додати коментарі:

Router(config)# access-list 10 remark

Це правило дозволяє доступ для адміністратора.

Коментар не впливає на трафік, але допомагає іншим інженерам розуміти призначення правил. Коментар не повинен перевищувати 100 символів.

Редагування ACL

Оскільки Cisco IOS не підтримує пряме редагування окремих записів у нумерованому ACL, існує два основні способи внесення змін.

Метод 1. Використання текстового редактора

- Використайте команду *show running-config* для виведення повної конфігурації.
- Скопіюйте потрібний ACL у текстовий редактор (наприклад, Notepad).
- Внесіть зміни.
- Видаліть старий ACL з пристрою.
- Вставте змінену версію назад.

Деякі версії IOS можуть або пропускати весь трафік, або блокувати його повністю, якщо ACL видалено з конфігурації, але не відключено з

інтерфейсу. Рекомендується спершу прибрати ACL з інтерфейсу (*no ip access-group*), і лише потім видаляти або змінювати список.

Метод 2. Використання порядкових номерів у іменованих ACL

Цей метод дозволяє точкове редагування.

Виведіть ACL:

Router# show access-lists 1

Наприклад, рядок з помилковою адресою 192.168.10.99 має номер 10.

Видаліть його:

Router(config-std-nacl)# no 10

Додайте правильний запис:

Router(config-std-nacl)# 10 deny host 192.168.10.10

Валідація ACL і облік статистики

Після застосування ACL слід перевірити його дію. Команда:

Router# show access-lists

відображає статистику спрацювань кожного правила.

Зверніть увагу, що у кожному ACL за замовчуванням наприкінці є приховане правило *deny all*. Воно не відображається у виводі, але блокує увесь трафік, що не відповідає жодному правилу. Щоб зробити його явним:

access-list 10 deny any log

Але слід бути обережним: якщо додати його не останнім, це може призвести до блокування легального трафіку.

Порядок обробки ACE-записів

Хоча записи додаються у порядку введення, IOS обробляє їх по-іншому: спочатку йдуть оператори вузлів (*host*), які впорядковуються внутрішньою хеш-функцією (*хешування (англ. hashing)* – це перетворення вхідного масиву даних довільної довжини у вихідний бітовий рядок фіксованої довжини, *хеш-функція* – це функція, що перетворює вхідні дані будь-якого (як правило великого) розміру в дані фіксованого розміру).

Потім йдуть оператори діапазону, у порядку їх введення.

Порядкові номери показують лише порядок додавання, а не обробки. Цей механізм впорядкування діє тільки на стандартні ACL IPv4. У розширених ACL та ACL для IPv6 обробка йде послідовно, без хешування.

16.7. Використання ACL -списка для управління доступом до VTY

Компанія Cisco рекомендує використовувати протокол SSH (Secure Shell) для безпечного віддаленого адміністративного доступу до маршрутизаторів і комутаторів.

Протокол SSH дозволяє захистити облікові дані адміністратора та іншу конфіденційну інформацію, шифруючи весь трафік між клієнтом і пристроєм Cisco.

На відміну від застарілого протоколу Telnet, який передає дані у відкритому вигляді, SSH забезпечує надійний захист від перехоплення даних (наприклад, за допомогою атаки «*man-in-the-middle*»).

Розглянемо приклад.

Якщо адміністратор підключається до маршрутизатора з IP-адресою 192.168.1.1, використовуючи SSH, то передані логін і пароль не зможе перехопити зловмисник, навіть якщо він має доступ до одного з проміжних маршрутизаторів або комутаторів у мережі.

Однак у випадках, коли образ Cisco IOS не підтримує протокол SSH (наприклад, у застарілих або спрощених версіях прошивки), рекомендується додатково обмежити доступ до віртуальних термінальних ліній VTY – каналів, через які здійснюється адміністративне підключення. Одним із надійних способів контролю є фільтрація за допомогою списків контролю доступу ACL, що дозволяє обмежити IP-адреси, з яких можливе підключення до пристрою по Telnet (або SSH, якщо він доступний).

Наприклад, ми можемо дозволити Telnet-доступ лише з IP-адреси адміністративної станції 192.168.10.10, заблокувавши всі інші звернення.

Для цього використовується команда *access-class*, яка налаштовується безпосередньо в режимі конфігурації лінії VTY. Ця команда дозволяє обмежити як вхідні, так і вихідні з'єднання між пристроєм Cisco та вказаними в ACL IP-адресами.

Це забезпечує додатковий контроль над тим, хто може управляти пристроєм, і навіть над тим, до кого пристрій може ініціювати з'єднання.

Синтаксис команди:

Router(config-line)# access-class <номер_ACL> { in [vrf-also] | out }

in – обмежує вхідні з’єднання до маршрутизатора (найчастіше використовується).

out – обмежує вихідні з’єднання, які ініціює сам маршрутизатор.

Важливі моменти щодо конфігурації доступу до VTY-ліній

Можна використовувати лише нумеровані ACL (наприклад, access-list 10 permit 192.168.10.10) для *access-class*. Іменовані ACL тут не підтримуються.

ACL має бути однаковим на всіх VTY-лініях. Cisco-пристрій зазвичай має кілька VTY-ліній (наприклад, з 0 по 4 або 0 по 15), і користувач може випадково або навмисно підключитися через будь-яку з них. Якщо обмеження встановлені лише на частину VTY, це створює вразливість у системі безпеки.

Стандартні ACL, що фільтрують лише IP-адреси джерела, підходять для застосування з *access-class*, навіть якщо мова йде про SSH чи Telnet, оскільки команда працює на логічному рівні ліній доступу, а не безпосередньо з мережевим стеком.

Приклад конфігурації:

Router(config)# access-list 10 permit 192.168.10.10

Router(config)# line vty 0 4

Router(config-line)# access-class 10 in

Ця конфігурація дозволяє підключатися до пристрою через будь-яку з ліній VTY (0-4) лише з IP-адреси 192.168.10.10. Усі інші адреси будуть автоматично відкинуті через неявне правило *deny*.

Обмеження ACL на VTY не замінюють розширені ACL, що використовуються на інтерфейсах маршрутизатора. Звичайні ACL, що застосовуються до IP-трафіку на інтерфейсах (вхідному або вихідному), не впливають на локально ініційовані з'єднання маршрутизатора. Тобто, навіть якщо існує ACL, яка блокує Telnet на інтерфейсі, маршрутизатор усе одно може ініціювати Telnet-з’єднання до іншого пристрою. Для контролю цього потрібно використовувати *access-class* out на VTY.

Приклад проблемної ситуації

Маршрутизатор має розширений список ACL, що блокує Telnet-з'єднання на зовнішньому інтерфейсі, але адміністратор виявляє, що маршрутизатор все ще може самостійно підключатися до інших пристроїв через Telnet. Це пов'язано з тим, що ACL не блокує трафік,

згенерований самим маршрутизатором. Для вирішення цієї проблеми потрібно використовувати *access-class out*.

Застосування команди *access-class* у поєднанні зі списками контролю доступу – ефективний спосіб убезпечити адміністративні канали доступу до мережевого пристрою. Це дозволяє створити додатковий рівень захисту, незалежно від того, використовується Telnet чи SSH, особливо в умовах, коли неможливо повністю відмовитися від небезпечних протоколів у старих середовищах.

16.8. Створення розширених ACL -списків

Щоб забезпечити гнучкіший і точніший контроль над мережевим трафіком, адміністратори мереж можуть застосовувати *розширені списки контролю доступу (Extended ACLs)*.

На відміну від стандартних ACL, які фільтрують трафік лише за IP-адресами джерела, розширені ACL дозволяють фільтрацію за цілим рядом додаткових параметрів, включаючи IP-адресу призначення, мережеві протоколи (TCP, UDP, ICMP тощо), а також номери портів або відомі служби, наприклад HTTP, FTP, Telnet.

Діапазон нумерації

Розширені ACL можуть бути нумерованими або іменованими. Для нумерованих списків зарезервовані діапазони з 100 до 199 і з 2000 до 2699, що забезпечує до 799 можливих ACL. Наприклад, ACL номер 110 може використовуватися для фільтрації трафіку HTTP та FTP, дозволяючи або забороняючи доступ залежно від мережевої політики.

Уявімо ситуацію: адміністратор хоче дозволити користувачам з підмережі 192.168.1.0/24 надсилати електронну пошту через SMTP (порт 25), але при цьому заборонити їм перегляд вебсторінок (HTTP – порт 80).

У такому випадку можна налаштувати розширений ACL наступним чином:

```
access-list 110 permit tcp 192.168.1.0 0.0.0.255 any eq 25
access-list 110 deny tcp 192.168.1.0 0.0.0.255 any eq 80
access-list 110 permit ip any any
```

Третій рядок дозволяє весь інший трафік, крім забороненого вище. Без нього наприкінці списку автоматично діє неявна заборона (*deny*

any), яка б блокувала будь-який трафік, що не відповідає попереднім правилам.

Фільтрація за портами і службами

Розширені ACL дозволяють деталізовано вказувати, які саме служби (на основі портів) слід дозволити чи заборонити. Це дозволяє мережевим адміністраторам створювати дуже конкретні правила, що відповідають бізнес-вимогам або політикам безпеки.

Приклад:

access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 20

access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 21

Ці записи дозволяють вихід з мережі 192.168.20.0/24 до будь-якого сервера за портами 20 (FTP Data) і 21 (FTP Control).

Встановлення ACL на інтерфейсі

Як і у випадку зі стандартними ACL, після створення розширеного ACL, його потрібно застосувати до конкретного інтерфейсу маршрутизатора. Це виконується за допомогою команди *ip access-group* в режимі конфігурації інтерфейсу:

interface GigabitEthernet0/1

ip access-group 110 in

Це означає, що ACL 110 буде застосовано до всього вхідного трафіку на інтерфейсі G0/1.

Складні випадки: повернення трафіку HTTP

Протокол HTTP є двостороннім: коли внутрішній клієнт надсилає запит до вебсайту, сервер повертає відповідь. Щоб уникнути блокування таких відповідей, можна скористатися параметром *established*, який дозволяє лише відповіді на раніше встановлені з'єднання:

access-list 104 permit tcp any 192.168.10.0 0.0.0.255 established

Така команда дозволяє повернення трафіку лише у випадку, якщо TCP-з'єднання вже встановлено, тобто пакети мають біт ACK або RST.

Іменовані ACL

Іменовані ACL створюються не за номером, а за текстовим ідентифікатором. Це зручно для керування великими конфігураціями, де імена можуть краще описувати функцію ACL.

Приклад створення іменованого розширеного ACL:

ip access-list extended BLOCK_HTTP

```
permit tcp 192.168.1.0 0.0.0.255 any eq 25
deny tcp 192.168.1.0 0.0.0.255 any eq 80
permit ip any any
```

Після створення ACL застосовується на інтерфейсі так само, як і нумерований список:

```
interface GigabitEthernet0/1
 ip access-group BLOCK_HTTP in
```

Для перегляду ACL використовується команда:

```
show access-lists BLOCK_HTTP
```

Правила обробки ACL

ACL обробляються послідовно – від першого до останнього запису. Як тільки знайдено відповідний запис ACE (Access Control Entry), інші не перевіряються. Якщо жоден запис не підходить – застосовується неявна команда *deny any*, яка відкидає всі інші пакети.

Змінити розширений ACL можна кількома способами.

Метод 1. Видалити старий список (*no access-list 110*), відредагувати його у текстовому редакторі, а потім заново вставити.

Метод 2. Для іменованих ACL можна використовувати порядкові номери рядків у CLI для вставки, зміни або видалення окремих ACE.

ACL можуть застосовуватись як на вхідному (*in*), так і на вихідному (*out*) трафіку інтерфейсу. Процес обробки пакету наступний.

1. Кадр надходить на інтерфейс.
2. Маршрутизатор знімає інформацію рівня 2.
3. Якщо на інтерфейсі є ACL, пакет перевіряється на відповідність.
4. Якщо пакет дозволено – виконується перевірка таблиці маршрутизації.
5. Якщо є вихідний інтерфейс, перевіряється, чи прив’язаний до нього ACL.
6. Якщо так – знову перевірка відповідності, і лише потім – передача пакета.
7. Якщо пакет не відповідає жодному дозволяючому запису, він відкидається і не досягає місця призначення.

Цей підхід до управління доступом дозволяє гнучко реалізовувати політику безпеки в корпоративних мережах, забезпечуючи не лише фільтрацію за IP-адресами, а й обмеження доступу до конкретних сервісів і додатків.

16.9. Приклади пошуку і усунення поширених помилок ACL

Команди перегляду *show* є ключовим інструментом при діагностиці та усуненні помилок у списках контролю доступу ACL. Завдяки їм можна виявити найпоширеніші проблеми, серед яких – неправильний порядок правил ACE або некоректне визначення умов доступу, таких як IP-адреси, протоколи чи порти.

Приклад помилки 1. Неправильний порядок записів у ACL

Ситуація: хост з IP-адресою 192.168.10.10 не може встановити з'єднання з хостом 192.168.30.12. Перевірка команди *show access-lists* на маршрутизаторі R3 показує, що перше правило (ACE 10) вже спрацювало, і саме воно забороняє доступ (рис. 16.2).

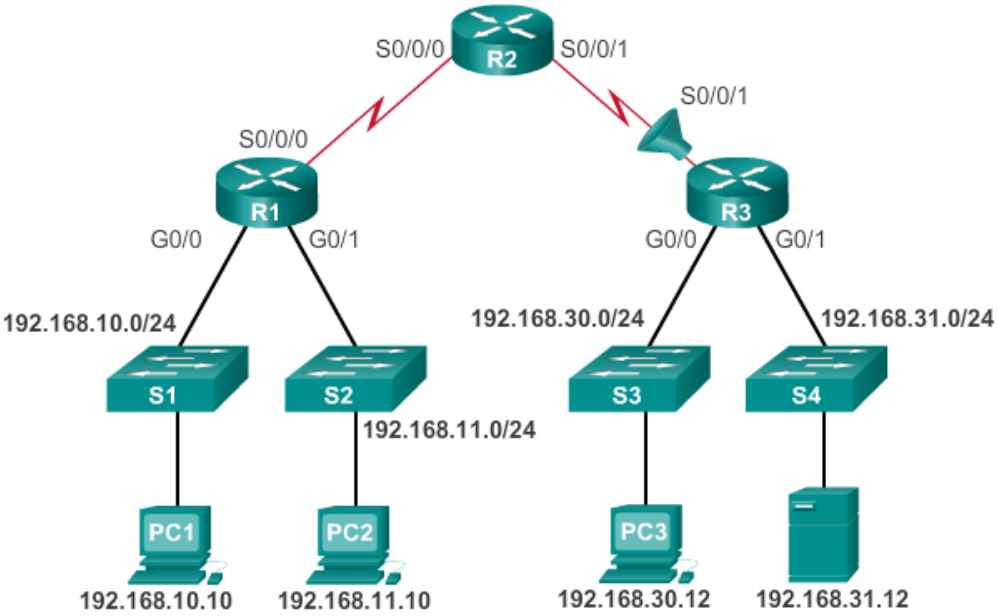


Рис. 16.2. Перевірка ACL

Причина: усі ACL-правила обробляються послідовно – з самого початку списку. Як тільки трафік відповідає певному ACE, решта правил більше не перевіряються. У даному випадку правило 10 одразу

блокує доступ для хоста 192.168.10.10, тому навіть якщо нижче у списку (наприклад, у правилі 20) дозвіл існує, він ніколи не буде врахований.

Рішення: необхідно переглянути порядок правил у ACL. Наприклад, дозволити потрібні з'єднання перед заборонними правилами:

access-list 101 permit ip host 192.168.10.10 host 192.168.30.12

access-list 101 deny ip any any

Приклад помилки 2. Неправильний протокол у дозволяючому правилі

Ситуація: мережа 192.168.10.0/24 не може скористатися TFTP для підключення до мережі 192.168.30.0/24. При цьому в ACL 120 є запис, що дозволяє TCP-з'єднання.

Причина: протокол TFTP (Trivial File Transfer Protocol) використовує UDP, а не TCP. Тому правило типу *permit tcp any any* не охоплює TFTP. Через неявне правило *deny all*, трафік, який не відповідає жодному з записів у ACL, автоматично блокується, навіть якщо це явно не показано в *show access-lists* (рис. 16.3).

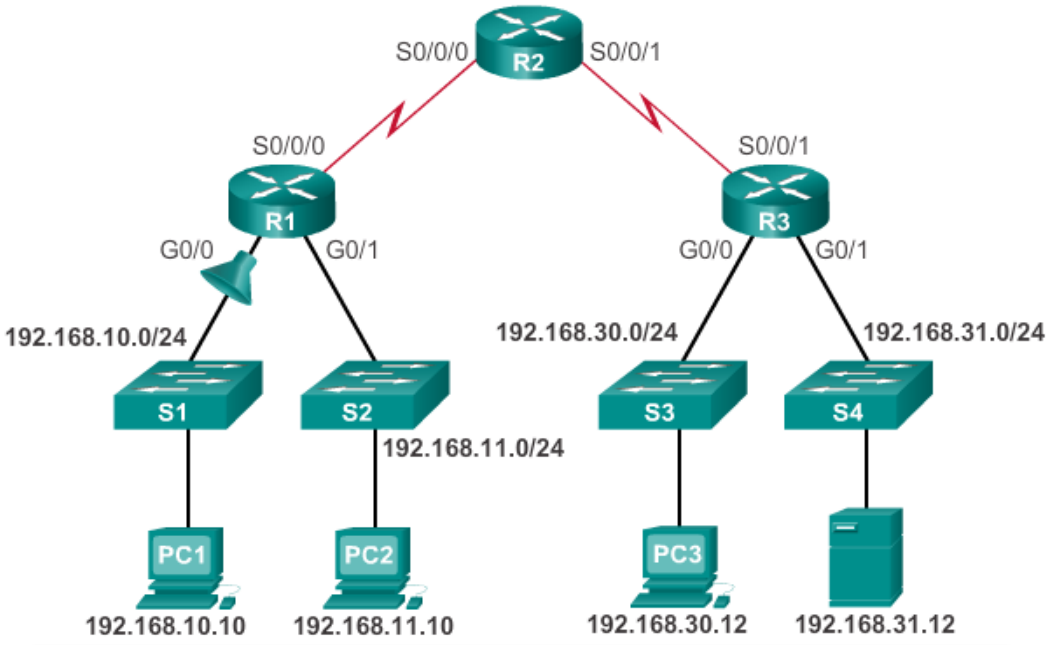


Рис. 16.3. Перегляд ACL 120

Рішення: оскільки TFTP працює на UDP-порті 69, потрібно дозволити UDP, або ж дозволити весь IP-трафік:

access-list 120 permit ip any any

Згідно з найкращими практиками, розширені ACL слід розміщувати якнайближче до джерела трафіку. У цьому випадку – на інтерфейсі G0/0 маршрутизатора R1.

Приклад помилки 3. Невірне використання порту в ACL

Ситуація: мережа 192.168.11.0/24 успішно встановлює Telnet-з'єднання з мережею 192.168.30.0/24, хоча згідно політики безпеки це заборонено. Команда *show access-lists 130* показує, що Telnet-з'єднання дозволено.

Причина: у записі списку контролю використаний помилковий напрямок фільтрації по порту. Наразі правило 10 намагається заборонити Telnet по джерелу порту, але на практиці фільтрацію слід виконувати по порту призначення, оскільки сервер Telnet слухає вхідні з'єднання на порту 23 (рис. 16.4).

```
R1# show access-lists 130
Extended IP access list 130
    10 deny tcp any eq telnet any
    20 deny tcp 192.168.11.0 0.0.0.255 host 192.168.31.12 eq smtp
    30 permit tcp any any (12 match(es))
```

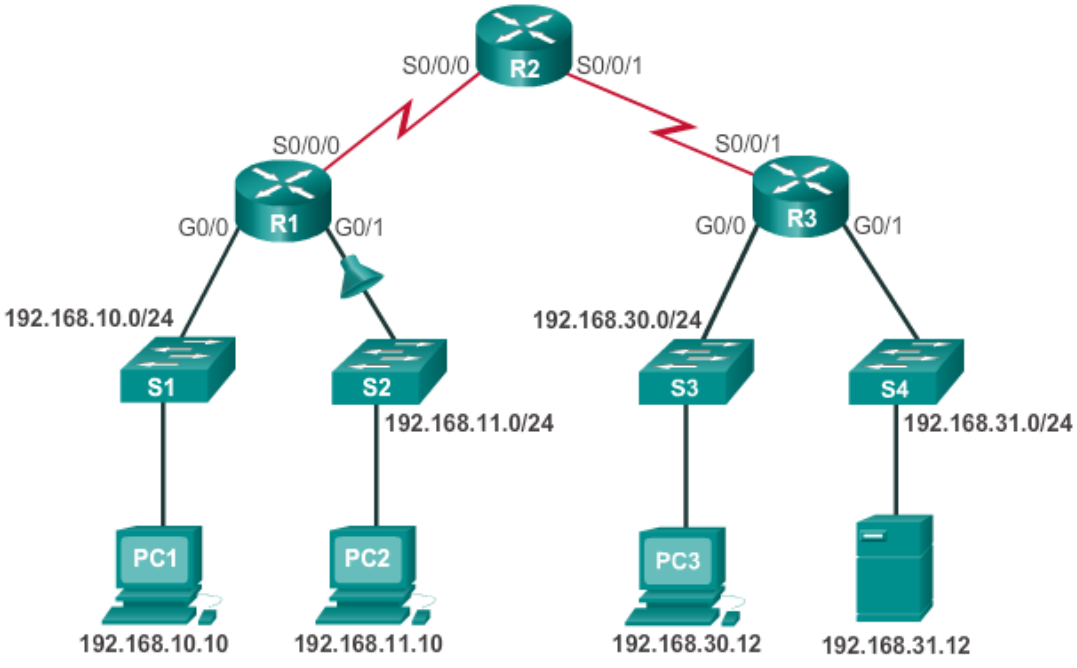


Рис. 16.4. Перегляд ACL 130

Рішення: правильно вказати порт призначення у ACL:

access-list 130 deny tcp any any eq 23

Приклад помилки 4. Відсутність конкретного правила блокування

Ситуація: хост 192.168.30.12 має змогу використовувати Telnet для підключення до 192.168.31.12, попри заборону згідно політики компанії.

Команда *show access-lists 140* показує, що з'єднання дозволене (рис. 16.5).

```
R3# show access-lists 140
Extended IP access list 140
    10 deny tcp host 192.168.30.1 any eq telnet
    20 permit ip any any (5 match(es))
```

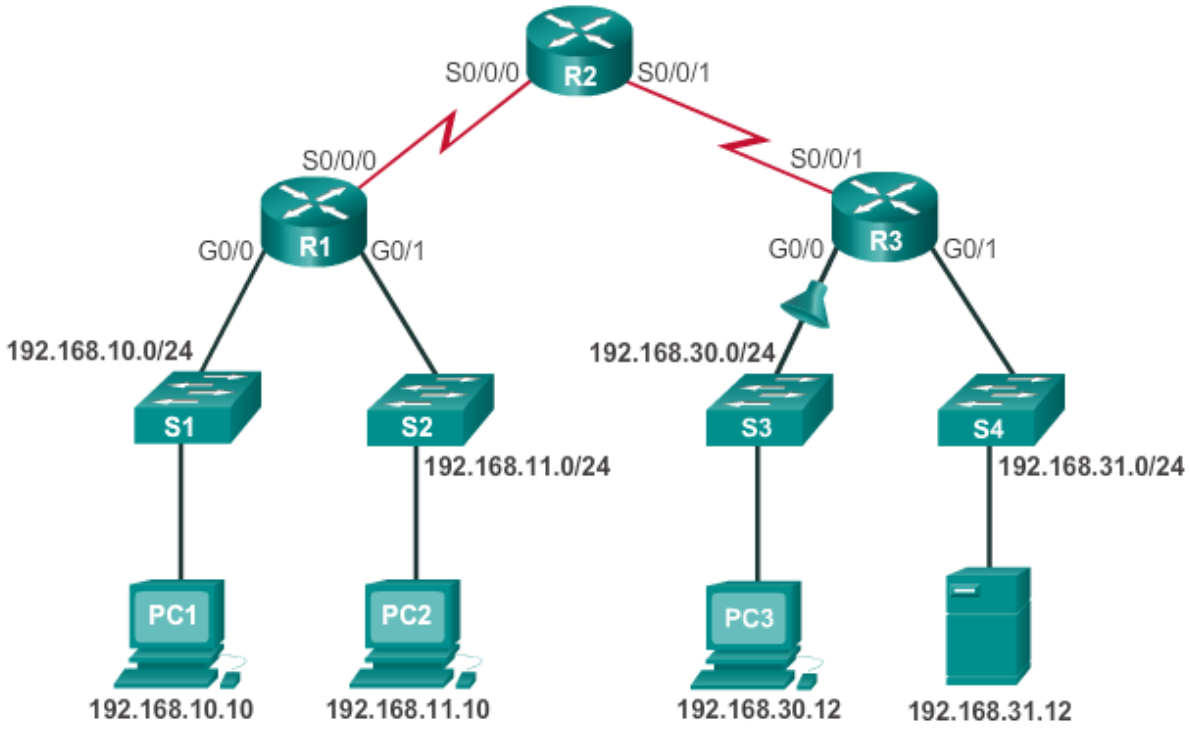


Рис. 16.5. Перегляд ACL 140

Причина: у ACL немає чіткого правила, яке забороняє підключення саме від 192.168.30.12. Поточний запис у ACL, можливо, стосується іншого трафіку або некоректно вказаного інтерфейсу.

Рішення: додати правило, яке явно забороняє з'єднання саме від 192.168.30.12:

access-list 140 deny tcp host 192.168.30.12 any eq 23

Приклад помилки 5. ACL застосований у неправильному напрямку

Ситуація: трафік Telnet від хоста 192.168.30.12 до хоста 192.168.31.12 не блокується, хоча правило в ACL 150 начебто забороняє таке з'єднання.

Команда *show access-lists 150* не фіксує спрацьовування правила.

Причина: список контролю застосований до інтерфейсу G0/1 у вхідному напрямку, тоді як фільтрування Telnet у цьому випадку має відбуватись у вихідному напрямку, оскільки трафік покидає маршрутизатор через G0/1.

```
R2# show access-lists 150
Extended IP access list 150
    10 deny tcp any host 192.168.31.12 eq telnet
    20 permit ip any any
```

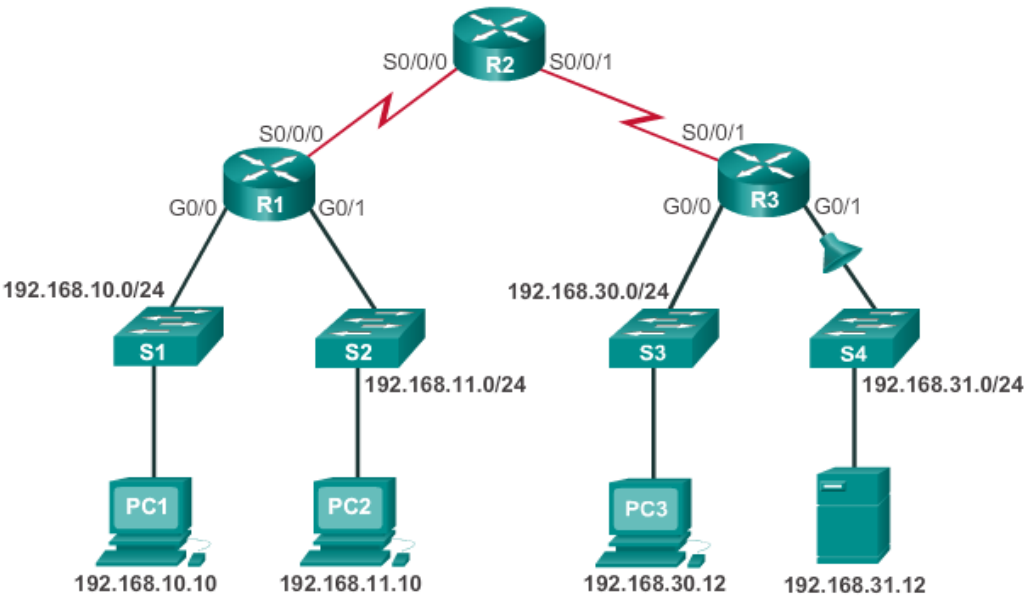


Рис. 16.6. Перегляд ACL 150

Рішення: застосувати ACL у відповідному напрямку, наприклад:

interface G0/1

ip access-group 150 out

Команда *show access-lists* дозволяє виявити типові помилки в ACL. Найчастіші помилки включають неправильний порядок записів, некоректне зазначення протоколів або портів, помилки у визначенні джерела та призначення трафіку та застосування ACL не на тому інтерфейсі або в неправильному напрямку.

Контрольні запитання та задачі до теми 16

Перелік теоретичних запитань

1. Що таке список контролю доступу ACL і яке його основне призначення?
2. У чому полягає відмінність між стандартними та розширеними ACL?
3. Яка структура TCP-сеансу та як ACL впливають на обмін даними?
4. Що таке шаблонна (wildcard) маска та як вона застосовується в ACL?
5. Як ACL взаємодіють з маршрутизатором Cisco при обробці трафіку?
6. Що означає принцип обробки ACL «зверху вниз»?
7. Яка відмінність між вхідним (inbound) і вихідним (outbound) фільтруванням у ACL?
8. Як реалізується управління віддаленим доступом до пристроїв Cisco за допомогою ACL?
9. Які правила слід дотримуватися при створенні списку доступу для VTY-каналів?
10. Чим відрізняються іменовані ACL від нумерованих?
11. Як ACL використовуються для обмеження доступу до певних протоколів, таких як Telnet чи SSH?
12. Як ACL впливають на продуктивність маршрутизатора?
13. Як можна уникнути типових помилок під час конфігурування ACL?
14. Яка роль прихованого правила *deny all* у кінці кожного ACL?
15. Чому важливо застосовувати розширені ACL якомога ближче до джерела трафіку?

Перелік задач для самостійного опрацювання

Практичне завдання 1. Створення та тестування стандартного ACL.

У середовищі моделювання Cisco Packet Tracer створити локальну мережу, заборонити доступ для однієї IP-адреси до мережі. Топологія мережі: R1 підключений до мережі 192.168.1.0/24. Налаштувати ACL, який забороняє хосту 192.168.1.100 доступ до цієї мережі. Верифікація через *ping* та команду *show access-lists*.

Практичне завдання 2. Розширений ACL для блокування Telnet.

У середовищі моделювання Cisco Packet Tracer створити локальну мережу, заборонити трафік Telnet з підмережі 192.168.10.0/24 до сервера в 192.168.30.12. Визначити точку застосування ACL. Перевірити з'єднання Telnet до сервера. Додати виключення для однієї IP-адреси, наприклад, 192.168.10.5.

Практичне завдання 3. Управління доступом до VTY.

У середовищі моделювання Cisco Packet Tracer дозволити лише одній IP-адресі доступ до VTY-каналів маршрутизатора. Налаштувати ACL з використанням команди *access-class*. Перевірити можливість віддаленого доступу через Telnet.

Тестове завдання 4.

Яке з наступного є правильним твердженням щодо ACL?

а) ACL застосовуються лише до вхідного трафіку

б) ACL автоматично застосовуються до всіх інтерфейсів

в) ACL потрібно явно застосувати до інтерфейсу

г) ACL не можуть фільтрувати за протоколами

Тестове завдання 5.

Яка команда застосовує ACL до інтерфейсу?

а) ip apply access-list 10 in

б) ip access-group 10 in

в) apply access-list 10

г) access-class 10 out

Перелік використаної та рекомендованої літератури

16.1. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

16.2. Assigned numbers [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc1700

16.3. Що таке VLAN [Електронний ресурс]. – Режим доступу: https://www.asus.com/ua-ua/support/faq/1049415/

Тема 17. Протокол DHCP

У сучасних комп'ютерних мережах ефективне управління IP-адресацією є критично важливим завданням для забезпечення надійної роботи пристроїв і сервісів. Одним із ключових інструментів, що автоматизує процес призначення IP-адрес, є протокол DHCP (Dynamic Host Configuration Protocol), який дозволяє пристроям отримувати не лише IP-адресу, а й інші мережеві параметри, необхідні для коректного функціонування у мережі – зокрема, адресу шлюзу за замовчуванням, DNS-сервери, маску підмережі тощо.

DHCP значно зменшує навантаження на системних адміністраторів, оскільки усуває необхідність у ручному налаштуванні кожного окремого пристрою, особливо в середовищах з великою кількістю користувачів або мобільних пристроїв. Протокол підтримується як у мережах IPv4 (DHCPv4), так і в мережах нового покоління IPv6 (DHCPv6), що робить його універсальним інструментом в адмініструванні мереж будь-якого масштабу.

Вивчення принципів роботи DHCP, його варіантів реалізації, переваг і недоліків, а також типових конфігурацій є важливою частиною підготовки фахівців з мережевих технологій.

17.1. Призначення протоколу DHCP

У комп'ютерних мережах кожен пристрій повинен мати унікальну IP-адресу для коректної ідентифікації та обміну даними. Зазвичай мережеві адміністратори вручну призначають постійні (статичні) IP-адреси пристроям, які мають незмінне фізичне або логічне розташування. До таких пристроїв належать маршрутизатори, сервери, принтери та інше критичне мережеве обладнання, яке забезпечує стабільне функціонування служб і сервісів. Статична адресація дозволяє адміністраторам легко отримувати віддалений доступ до обладнання, оскільки відома наперед IP-адреса спрощує пошук пристрою в мережі.

Однак в умовах динамічного середовища – наприклад, у великих офісах або підприємствах, де користувачі часто змінюють місце роботи або використовують мобільні пристрої – постійне оновлення IP-адрес вручну стає вкрай складним і неефективним. Кожне переміщення комп'ютера або зміна мережевого сегмента вимагає повторного конфігурування параметрів адресації. Така процедура вимагає часу,

зусиль і може призводити до помилок, особливо при розширенні інфраструктури або роботі з віддаленими користувачами.

Для автоматизації процесу налаштування адресації використовується протокол динамічного призначення IP-адрес – DHCP, який дозволяє автоматично видавати IP-адреси та супутні параметри (наприклад, шлюз за замовчуванням, DNS-сервери) клієнтам у мережі без необхідності втручання адміністратора. Реалізація DHCP-сервера в локальній мережі значно спрощує процес налаштування як стаціонарних комп'ютерів, так і ноутбуків або смартфонів, що підключаються до мережі тимчасово або з різних місць.

Використання централізованого DHCP-сервера дозволяє ефективно контролювати розподіл IP-адрес в організації, включаючи філії або віддалені офіси. Це забезпечує єдину політику управління адресацією, запобігає конфліктам адрес і забезпечує зручний аудит використання мережевих ресурсів. DHCP підтримується як для протоколу IPv4 (DHCPv4), так і для IPv6 (DHCPv6), що дозволяє адаптуватися до сучасних потреб мережевого простору.

Протокол DHCPv4 забезпечує три основні способи призначення IP-адрес, кожен з яких може бути використаний залежно від конкретних вимог.

Ручний (manual) спосіб призначення IP-адрес – адреса прив'язується до MAC-адреси пристрою, і DHCP лише доставляє її клієнту. Це дозволяє забезпечити постійність адреси, як у випадку статичної конфігурації, але із централізованим управлінням.

Автоматичний (automatic) спосіб призначення IP-адрес – сервер самостійно обирає адресу з пулу адрес та закріплює її за пристроєм на необмежений термін. Це зручно для постійних клієнтів, але без адміністративного втручання.

Динамічний (dynamic) спосіб призначення IP-адрес – найпоширеніший варіант, коли IP-адреса видається на певний час (оренду), після чого або оновлюється, або змінюється.

Динамічний розподіл особливо зазвичай використовується у середовищах з великою кількістю тимчасових клієнтів – наприклад, в університетських кампусах, готелях, коворкінгах. Термін оренди зазвичай встановлюється на 1 день, кілька днів або навіть тиждень. Після завершення терміну клієнт може запитати нову адресу або

повторно отримати ту ж саму, якщо вона ще вільна. Це дозволяє ефективно використовувати адресний простір і мінімізує ручну конфігурацію з боку системного адміністратора.

17.2. Операція DHCPv4

Протокол DHCPv4 (Dynamic Host Configuration Protocol версії 4) працює за принципом взаємодії за моделлю «клієнт-сервер», що дозволяє автоматизовано призначати IP-адреси та інші мережеві параметри пристроям у локальній мережі (рис. 17.1).

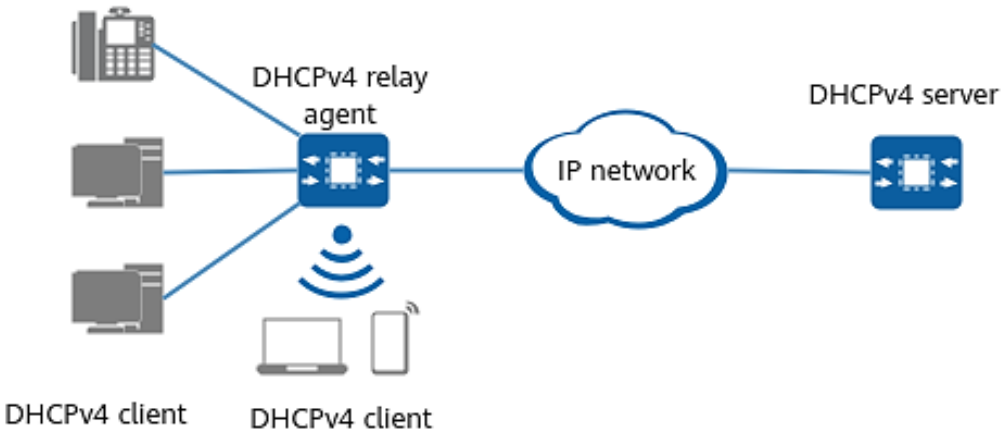


Рис. 17.1. Приклад взаємодії «клієнт-сервер» для DHCPv4

Якщо комп'ютер або інший мережевий пристрій (клієнт) приєднується до мережі, він не має власної IP-адреси, тому звертається до DHCP-сервера з проханням отримати її. DHCP-сервер у відповідь призначає клієнту IP-адресу на певний час – «в оренду».

Цей механізм дозволяє оптимально використовувати обмежений пул IP-адрес. Якщо клієнт відключається або залишає мережу, його адреса згодом автоматично повертається в загальний пул для повторного використання. Це особливо корисно в динамічних середовищах, таких як офіси, готелі або Wi-Fi-мережі, де пристрої часто приєднуються та від'єднуються.

Коли клієнт щойно підключився до мережі (наприклад, після завантаження або увімкнення мережевого інтерфейсу), він ініціює 4-кроковий процес DHCP-взаємодії для отримання IP-адреси.

Крок 1. DHCPDISCOVER – виявлення DHCP-сервера.

Клієнт надсилає широкомовне повідомлення DHCPDISCOVER у спробі знайти доступні DHCP-сервери. Оскільки у клієнта ще немає IP-адреси, він використовує широкомовну адресу 255.255.255.255 на рівні 3 (IP) та FF:FF:FF:FF:FF:FF на рівні 2 (Ethernet MAC). Це повідомлення містить MAC-адресу клієнта, щоб сервер міг відрізнити запитувача.

Наприклад, ноутбук підключається до Wi-Fi у кав'ярні, і автоматично надсилає DHCPDISCOVER, щоб отримати мережеву адресу.

Крок 2. DHCPOFFER – пропозиція від сервера.

DHCP-сервер, що отримав DHCPDISCOVER, резервує одну з доступних IP-адрес у своєму пулі і формує відповідь DHCPOFFER. У цій відповіді сервер вказує пропоновану IP-адресу, маску підмережі, шлюз за замовчуванням, DNS-сервери та термін оренди.

Крім того, DHCP-сервер створює запис ARP, який пов'язує MAC-адресу клієнта із запропонованою IP-адресою. Повідомлення DHCPOFFER надсилається одноадресно клієнту, використовуючи його MAC-адресу.

Наприклад, сервер пропонує IP-адресу 192.168.1.45 клієнту з MAC-адресою 00:1A:2B:3C:4D:5E.

Крок 3. DHCPREQUEST – запит на оренду.

Після отримання однієї або кількох пропозицій від серверів, клієнт обирає одну з них (зазвичай першу отриману) і надсилає повідомлення DHCPREQUEST, яке служить офіційним підтвердженням прийняття саме цієї пропозиції. Повідомлення надсилається у вигляді широкомовної розсилки, щоб проінформувати всі DHCP-сервери у мережі – обраний сервер визнається, інші – ігноруються.

Наприклад, клієнт отримав пропозиції від двох серверів, але обрав ту, яка запропонувала найкращі параметри. DHCPREQUEST сигналізує всім, що вибір зроблено.

Крок 4. DHCPACK – підтвердження призначення.

Отримавши DHCPREQUEST, обраний сервер виконує перевірку на конфлікт IP-адрес, надсилаючи запит ICMP Echo (*ping*) на запропоновану IP-адресу. Якщо відповіді немає, адреса вважається вільною. Після цього сервер надсилає клієнту повідомлення DHCPACK, яке остаточно підтверджує оренду.

Клієнт, отримавши DHCPACK, зберігає отриману IP-адресу та інші налаштування, а також проводить власну ARP-перевірку: надсилає ARP-запит, щоб переконатися, що в мережі ніхто інший не використовує цю адресу.

Наприклад, якщо на ARP-запит не надійшла відповідь, клієнт безпечно використовує отриману адресу (наприклад, 192.168.1.45).

Оновлення терміну оренди IP-адреси

IP-адреса не призначається клієнту назавжди – вона має визначений термін дії (lease time). Наприклад, DHCP-сервер може видавати адресу на 8 годин. Щоб продовжити оренду, клієнт повинен періодично надсилати запити DHCPREQUEST безпосередньо тому ж серверу, який видав адресу.

Якщо DHCP-сервер отримує такий запит, він відповідає DHCPACK – це означає, що оренда подовжена. Якщо відповідь не надходить упродовж певного часу (наприклад, сервер не в мережі), клієнт надсилає широкомовний запит DHCPREQUEST, щоб знайти альтернативний DHCP-сервер, здатний продовжити або переоформити оренду.

Переваги механізму оренди

Оренда адрес забезпечує гнучке, ефективне та динамічне використання адресного простору. Наприклад, якщо співробітник залишає офіс з ноутбуком або смартфон втрачає з’єднання з Wi-Fi, адреса не залишатиметься зайнятою, а буде повернена до пулу DHCP-сервера після завершення терміну оренди.

Наприклад, в офісі на 50 працівників достатньо пулу з 60 IP-адрес – навіть якщо кілька людей одночасно приходять і йдуть, DHCP динамічно перерозподіляє адреси.

17.3. Формат повідомлень DHCPv4

Усі транзакції в межах протоколу DHCPv4 відбуваються з використанням уніфікованого формату повідомлень, який забезпечує взаєморозуміння між клієнтськими пристроями та сервером DHCP. Ці повідомлення не передаються безпосередньо, а інкапсулюються всередині транспортного протоколу UDP (User Datagram Protocol), що є простим протоколом без встановлення з’єднання. При цьому клієнт відправляє повідомлення із використанням UDP-порту 68 як джерела, а сервер отримує його через порт призначення 67. Це дозволяє

автоматизувати процес запиту та отримання IP-адрес без необхідності ручного налаштування.

Повідомлення DHCPv4 складається з низки полів, кожне з яких виконує окрему функцію. Наприклад, поле "код операції (OP)" визначає тип повідомлення: чи це запит від клієнта (значення 1), чи відповідь від сервера (значення 2). Це дозволяє системі правильно ідентифікувати напрямок комунікації. Поле "тип устаткування" (наприклад, 1 для Ethernet) вказує на фізичний інтерфейс, яким користується клієнт, аналогічно до форматів у протоколі ARP. Також повідомлення містить поле "довжина фізичної адреси", що повідомляє про кількість байтів, які займає MAC-адреса клієнта – зазвичай це 6 байтів для Ethernet. Поле "переходи" (*hops*) використовується у випадку, якщо повідомлення проходить через декілька агентів-ретрансляторів. Значенням за замовчуванням є 0, що вказує на безпосереднє підключення клієнта до локального сегменту мережі.

"Ідентифікатор транзакції" (transaction ID) – унікальне випадкове число, що генерується клієнтом і допомагає узгоджувати повідомлення у випадку, коли кілька серверів надсилають відповіді. Це дає змогу точно визначити, яка відповідь відповідає якому запиту.

Поле "секунди" дозволяє серверу зрозуміти, скільки часу минуло з моменту початку запиту клієнта. Наприклад, якщо клієнт чекає відповіді надто довго, сервер може надати йому пріоритет у відповіді.

"Прапори" застосовуються, якщо клієнт ще не має IP-адреси. Зокрема, використовується лише 1 біт з 16, який визначає, чи потрібно відповідь надсилати як широкомовну розсилку. Якщо встановлений біт дорівнює 1, то відповідь сервер надсилає на адресу 255.255.255.255.

Деякі поля повідомлення заповнюються лише в окремих випадках. Наприклад, поле "IP-адреса клієнта" містить адресу, яку вже має пристрій, якщо він звертається за продовженням оренди. Якщо це первинний запит, то в полі стоїть 0.0.0.0. "IP-адреса сервера" може вказувати на DHCP-сервер, з якого клієнт повинен продовжувати конфігурацію. Проте справжня адреса DHCP-сервера вказується окремо – в полі DHCP Server Identifier. У випадку, якщо сервер і клієнт знаходяться в різних підмережах, використовується "IP-адреса шлюзу" – саме за її допомогою DHCP-агент пересилає пакети через маршрутизатор до відповідного сегменту мережі.

"Фізична адреса клієнта", або MAC-адреса, є важливою для ідентифікації пристрою в мережі та для фільтрації або резервування IP-адрес. Інші поля використовуються за потребою. Наприклад, "ім’я сервера" може містити псевдонім або доменне ім’я DHCP-сервера (наприклад, *dhcp1.company.local*), хоча це не обов’язково. Поле "ім’я файлу завантаження" важливе у випадках, коли клієнт завантажує образ операційної системи через мережу (PXE-завантаження), – воно вказує ім’я файлу, який потрібно отримати.

Найважливішим є поле DHCP-опції, в якому розміщуються параметри, критичні для функціонування протоколу: IP-адреса шлюзу, адреси DNS-серверів, термін оренди, тип конфігурації тощо. Це поле є змінної довжини, і в ньому закладена велика частина функціональності протоколу.

У реальному середовищі, коли пристрій підключається до мережі (наприклад, ноутбук у офісі або смартфон вдома через Wi-Fi), він, не маючи IP-адреси, розсилає в локальну мережу повідомлення DHCPDISCOVER на широкомовну адресу 255.255.255.255. При цьому адреса джерела в IP-заголовку встановлюється в 0.0.0.0, бо пристрій ще не має жодної IP-адреси.

Сервер DHCP, що отримав цей запит, відповідає повідомленням DHCPOFFER, у якому пропонує клієнту IP-адресу, маску підмережі, шлюз, DNS-сервер, а також вказує тривалість оренди адреси. Наприклад, клієнт може отримати адресу 192.168.1.100 на 24 години разом з параметрами доступу до зовнішньої мережі.

За потреби, сервер може додати інші параметри, такі як час синхронізації або параметри проксі. Завдяки цьому DHCP дає змогу пристроям підключатися до мережі без необхідності втручання користувача чи адміністратора, що особливо корисно в масштабних корпоративних середовищах.

17.4. Налаштування DHCPv4 сервера

Маршрутизатор з операційною системою Cisco IOS може виконувати роль DHCPv4-сервера, що дозволяє автоматично видавати IP-адреси клієнтам у мережі. У такій конфігурації маршрутизатор не лише розподіляє IP-адреси з визначеного пулу, а й керує їх орендою,

запобігає конфліктам і автоматизує процес налаштування клієнтських пристроїв.

Розглянемо етапи конфігурації DHCPv4-сервера на маршрутизаторі Cisco.

Крок 1. Виключення певних IPv4-адрес з пулу.

За замовчуванням маршрутизатор намагається використовувати всі IP-адреси в заданому пулі для динамічного розподілу. Проте в реальних мережах деякі адреси мають бути зарезервовані для важливих пристроїв, налаштованих вручну – таких як маршрутизатори, сервери, мережеві принтери або камери відеоспостереження.

Щоб ці IP-адреси не видавалися іншим пристроям випадково, їх потрібно виключити з DHCP-пулу за допомогою команди:

ip dhcp excluded-address <start-IP> [<end-IP>]

Приклад: якщо IP-адреси з 192.168.10.1 до 192.168.10.10 використовуються для вручну налаштованих пристроїв, введіть:

ip dhcp excluded-address 192.168.10.1 192.168.10.10

Це запобігає конфліктам адрес і гарантує стабільну роботу ключових компонентів мережі.

Крок 2. Створення пулу DHCPv4-адрес.

Наступний етап – створення DHCP-пулу, з якого маршрутизатор буде розподіляти IP-адреси клієнтам. Це виконується за допомогою команди:

ip dhcp pool <pool-name>

Після її введення маршрутизатор переходить у спеціальний режим конфігурації DHCP (Router(dhcp-config)#), у якому можна задати параметри пулу.

Приклад:

ip dhcp pool OFFICE-POOL

Тут OFFICE-POOL – довільна назва пулу, яка допоможе адміністратору орієнтуватися в конфігурації, особливо якщо в мережі кілька пулів (наприклад, для різних підмереж або VLAN).

Крок 3. Налаштування параметрів DHCP-пулу.

Усередині DHCP-пулу необхідно вказати основні мережеві параметри, які будуть надсилатися клієнтам разом з IP-адресою.

1. Діапазон доступних IP-адрес.

Задається командою *network*, яка вказує мережу та маску підмережі:

network <network-address> <subnet-mask>

Приклад:

network 192.168.10.0 255.255.255.0

Це означає, що адреси з мережі 192.168.10.0/24 будуть доступні для оренди (за винятком виключених).

2. Основний шлюз.

Зазвичай – це IP-адреса інтерфейсу маршрутизатора, що підключений до клієнтів. Задається командою:

default-router <gateway-IP>

Приклад:

default-router 192.168.10.1

Можна вказати до 8 шлюзів, якщо в мережі передбачено резервування або складні маршрути.

3. DNS-сервер.

Клієнти отримають цей параметр для розпізнавання доменних імен. Використовується команда:

dns-server <DNS-IP>

Приклад:

dns-server 8.8.8.8 1.1.1.1

Це дозволить клієнтам використовувати Google DNS та Cloudflare DNS.

4. Доменне ім'я.

Встановлюється за допомогою команди:

domain-name <domain>

Приклад:

domain-name corp.local

Це домен, у якому працюють клієнти – він може бути внутрішнім або зовнішнім.

5. Термін оренди IP-адреси.

Стандартна тривалість оренди становить 1 день, але її можна змінити командою:

lease <days> [<hours> <minutes>]

Приклад:

lease 0 12 0

Це встановить термін оренди на 12 годин.

6. NetBIOS (WINS) сервер.

Для застарілих або змішаних середовищ, де використовується NetBIOS, можна вказати WINS-сервер:

netbios-name-server <IP>

Приклад:

netbios-name-server 192.168.10.5

У Cisco IOS служба DHCP за замовчуванням увімкнена. Якщо з якихось причин потрібно тимчасово вимкнути функціональність DHCP-сервера, використовується команда глобальної конфігурації:

no service dhcp

Для повторного увімкнення:

service dhcp

Просто увімкнення або вимкнення служби DHCP не матиме жодного ефекту, якщо попередньо не створено жодного DHCP-пулу конфігурації.

17.5. DHCPv4-ретрансляція

У великих корпоративних мережах, які мають складну ієрархічну структуру, сервери зазвичай не розташовуються випадково або поодиноко, а об'єднуються в централізовані обчислювальні центри – так звані *серверні ферми (англ. server farms або data centers)*. Ці спеціалізовані сегменти мережі, як правило, виділені в окремі підмережі та обладнані високопродуктивними пристроями для обробки великої кількості запитів. У межах таких ферм можуть функціонувати сервери, що надають різні мережеві сервіси – наприклад, динамічне призначення IP-адрес DHCP, система доменних імен DNS, спрощений протокол передачі файлів TFTP або FTP повнофункціональний протокол передачі файлів.

Оскільки користувачі (клієнти мережі) часто фізично або логічно відокремлені від серверів – тобто знаходяться в інших підмережах – виникає проблема з досягненням служб, які очікують широкомовні запити, зокрема DHCP. За стандартами IPv4, широкомовні повідомлення не передаються за межі локальної підмережі – це означає, що DHCPDISCOVER не може дійти безпосередньо до сервера, який знаходиться, наприклад, в дата-центрі.

Для вирішення цієї проблеми в комп'ютерних мережах застосовується налаштування *допоміжної адреси* (англ. *ip helper-address*), яка дозволяє маршрутизатору пересилати широкомовні пакети DHCP до віддаленого DHCP-сервера. Коли клієнт надсилає DHCP-запит на адресу 255.255.255.255, маршрутизатор, який отримує цей запит на своєму інтерфейсі, не ігнорує його, а виступає в ролі *агента ретрансляції* (*DHCP relay agent*), який приймає запит, інкапсулює його та пересилає уніказованим способом (unicast) безпосередньо до IP-адреси DHCP-сервера, яка вказується в параметрі *ip helper-address*.

Наприклад, якщо в офісі знаходиться принтер, що підключається до мережі через маршрутизатор, і DHCP-сервер розташований у головному дата-центрі, налаштування *ip helper-address 10.10.10.5* на інтерфейсі маршрутизатора дозволить цьому принтеру отримати IP-адресу з головного DHCP-сервера.

Слід зауважити, що DHCP – не єдина служба, яку може обслуговувати механізм ретрансляції.

За замовчуванням команда *ip helper-address* налаштована на пересилання широкомовних повідомлень для восьми стандартних UDP-служб, які часто використовуються в корпоративних мережах.

1. *UDP порт 37*: протокол Time – використовується для синхронізації годинників клієнтів і серверів.
2. *UDP порт 49*: TACACS – протокол контролю доступу, який використовується для автентифікації в мережевих пристроях.
3. *UDP порт 53*: DNS – забезпечує перетворення доменних імен у IP-адреси.
4. *UDP порт 67*: DHCP/BOOTP client – запити від клієнтів.
5. *UDP порт 68*: DHCP/BOOTP server – відповіді серверів.
6. *UDP порт 69*: TFTP – легкий протокол для передачі конфігураційних файлів і прошивок.
7. *UDP порт 137*: NetBIOS Name Service – дозволяє знайти ім'я комп'ютера в мережі Windows.
8. *UDP порт 138*: NetBIOS Datagram Service – служить для надсилання повідомлень у стилі Windows у локальній мережі.

Таким чином, використання команди *ip helper-address* спрощує адміністрування, покращує централізованість, і дозволяє зменшити кількість серверів DHCP, необхідних у мережі.

17.6. Налаштування маршрутизатора в якості DHCPv4 клієнта

У деяких ситуаціях маршрутизатори Cisco, особливо ті, що використовуються в невеликих офісах, домашніх мережах SOHO або віддалених філіях, повинні не надавати IP-адреси, а навпаки – отримувати їх автоматично від провайдера. Тобто виступати в ролі DHCPv4-клієнта, подібно до звичайних комп'ютерів або принтерів у мережі (рис. 17.2).

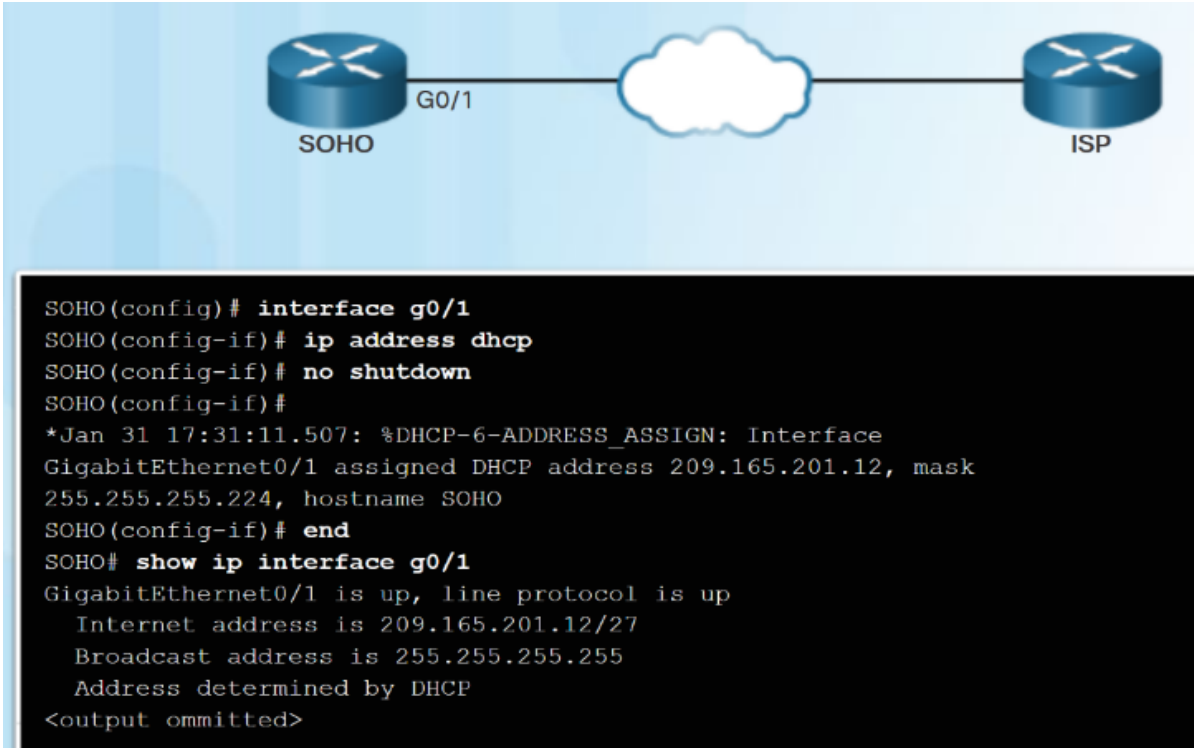


Рис. 17.2. Приклад налаштування маршрутизатора як DHCPv4-клієнта

Це часто трапляється при підключенні до кабельного або DSL-провайдера, де IP-адресу для зовнішнього інтерфейсу видає обладнання провайдера – наприклад, модем або шлюз.

Метод підключення та тип інтерфейсу

У більшості простих інсталяцій для підключення до Інтернету використовується Ethernet-інтерфейс маршрутизатора. Це може бути інтерфейс типу GigabitEthernet 0/1 або інший, залежно від конкретної моделі пристрою.

Щоб такий інтерфейс працював як DHCP-клієнт, достатньо виконати базову конфігурацію, вказавши, що IP-адреса має бути

отримана автоматично. Для цього в режимі конфігурації інтерфейсу застосовується команда:

ip address dhcp

Ця команда вказує маршрутизатору, що даний інтерфейс повинен ініціювати DHCP-запит до сервера, отримати доступну IP-адресу, маску підмережі, шлюз за замовчуванням та, за потреби, додаткові параметри (наприклад, DNS).

Уявімо, що інтернет-провайдер (ISP) надає абонентам IP-адреси з діапазону 209.165.201.0/27, що включає 30 доступних адрес. Провайдер налаштував своє обладнання як DHCP-сервер і призначає клієнтам динамічні IP-адреси зі згаданого діапазону. На стороні абонента (тобто в домашньому офісі або філії) використовується маршрутизатор, який підключається до модему ISP через інтерфейс GigabitEthernet 0/1. Для того, щоб цей інтерфейс отримав IP-адресу автоматично, виконується наступна конфігурація:

interface GigabitEthernet0/1

ip address dhcp

no shutdown

Команда *no shutdown* активує інтерфейс, якщо він був вимкнений.

Перевірка налаштування DHCP-клієнта

Після завершення налаштування можна перевірити, чи успішно отримано IP-адресу від DHCP-сервера, за допомогою команди:

show ip interface g0/1

У виводі цієї команди буде видно:

- поточну IP-адресу інтерфейсу (виданий сервером DHCP);
- маску підмережі;
- статус інтерфейсу (він має бути «up»);
- додаткову інформацію, таку як шлюз за замовчуванням (за допомогою команди *show ip route*) або отримані DNS-сервери.

Цей механізм дозволяє гнучко та швидко підключати маршрутизатори до провайдера, не вводячи вручну статичні параметри адресації. Це особливо зручно в сценаріях з масовим розгортанням пристроїв або при використанні провайдерських послуг із динамічною адресацією. Такий підхід зменшує ризик помилок при ручному налаштуванні та спрощує адміністрування мережі.

17.7. Пошук і усунення помилок DHCPv4

У процесі роботи протоколу DHCPv4 можуть виникати різноманітні проблеми. Їхні причини досить різноманітні – від апаратних збоїв до помилок у налаштуваннях програмного забезпечення. Наприклад, неполадки можуть бути викликані збоєм у роботі операційної системи, драйвера мережевого адаптера, або неправильною роботою DHCP-агента ретрансляції, що перенаправляє запити клієнтів на сервер у різних підмережах. Водночас практика свідчить, що найпоширенішою причиною несправностей є неправильна або неповна конфігурація параметрів DHCP. Через велику кількість потенційних джерел несправностей під час діагностики роботи DHCPv4 важливо застосовувати систематизований, крок за кроком підхід, перевіряючи як фізичні, так і логічні параметри.

1. Вирішення конфліктів IPv4-адрес

Одна з типових проблем – це конфлікти IP-адрес, коли двом клієнтам випадково надається одна й та ж IP-адреса. Така ситуація може виникати, наприклад, якщо термін оренди адреси у одного клієнта завершився, але з якоїсь причини він не звернувся до сервера для її продовження. DHCP-сервер у такому випадку може повторно призначити цю ж адресу іншому клієнту. Після перезавантаження перший клієнт може спробувати використовувати "стару" адресу, що і спричиняє конфлікт. Для виявлення таких ситуацій для DHCPv4-сервера використовується команда *ping*, щоб перевірити доступність адреси в мережі. У свою чергу, клієнт може виявляти конфлікти за допомогою протоколу ARP (Address Resolution Protocol). Якщо конфлікт зафіксовано, сервер видаляє проблемну IP-адресу з пулу розподілу доти, доки адміністратор вручну не вирішить конфлікт. Для перегляду списку зафіксованих конфліктів використовується команда:

R1# show ip dhcp conflict

Ця команда дозволяє побачити IP-адреси, що викликали конфлікт, метод їх виявлення (наприклад, Ping або Gratuitous ARP) та точний час виявлення.

2. Перевірка фізичного з'єднання

Наступним кроком має бути перевірка фізичної доступності мережі, а саме – чи працює інтерфейс маршрутизатора, до якого під'єднаний клієнт. Для цього використовується команда:

show interfaces interface

Якщо інтерфейс не має статусу *up*, жодні дані, включно із DHCP-запитами, не будуть передаватися, і клієнт не зможе отримати IP-адресу. Наприклад, вимкнутий порт на комутаторі або пошкоджений кабель можуть повністю блокувати обмін даними.

3. Перевірка зв'язності зі статичною IP-адресою

Щоб визначити, чи пов'язана проблема саме з DHCP-сервером, доцільно призначити статичну IP-адресу на клієнтському пристрої. Якщо після цього підключення до мережевих ресурсів усе одно неможливе – наприклад, не відкривається вебсайт або не пінгується шлюз – проблема, ймовірно, не в DHCP, а в загальній мережевій інфраструктурі. Такий метод дозволяє відокремити проблему DHCP від інших мережевих неполадок.

4. Перевірка налаштування порту комутатора

Інколи причина несправності криється в некоректному налаштуванні порта комутатора, до якого підключено клієнт. Наприклад, порт може бути переведений у неправильний VLAN або мати обмеження на передачу широкомовних запитів, до яких належать DHCP-запити. У таких випадках слід вручну ініціювати запит DHCP (*ipconfig /renew* у Windows або *dhclient* у Linux) і проаналізувати, чи приходить відповідь від сервера.

5. Діагностика роботи DHCP в межах однієї підмережі або VLAN

Важливим етапом є перевірка, чи функціонує DHCP-сервер правильно, якщо клієнт знаходиться в тій же підмережі або VLAN.

Це дозволяє виключити можливі проблеми з агентами ретрансляції DHCP (DHCP relay agents), які використовуються для передавання DHCP-повідомлень між підмережами. Якщо DHCP працює коректно лише в межах однієї VLAN, проблема може полягати у неправильному налаштуванні ретранслятора (наприклад, відсутність IP-адреси сервера в параметрах *ip helper-address*). Якщо ж DHCP не працює навіть у локальній мережі, найімовірніше, проблема саме в конфігурації або програмному забезпеченні DHCP-сервера.

Таким чином, діагностика DHCPv4 потребує комплексного підходу, що охоплює перевірку конфігурацій, фізичних з'єднань, VLAN-ідентифікації, а також здатності клієнта й сервера правильно взаємодіяти.

17.8. Перевірка налаштувань DHCPv4 на маршрутизаторі

Коли сервер DHCPv4 знаходиться в іншій локальній мережі, відокремленій від клієнтської LAN, маршрутизатор відіграє ключову роль у забезпеченні комунікації між ними. У такому випадку інтерфейс маршрутизатора, що з'єднаний з мережею клієнтів, має бути налаштований на ретрансляцію DHCPv4-запитів у сторону віддаленого DHCP-сервера. Для цього використовується спеціальна команда налаштування – *ip helper-address*, яка вказує маршрутизатору, куди саме (на яку IP-адресу) потрібно пересилати широкомовні запити, що надходять від клієнтів.

Наприклад, якщо DHCP-сервер має IP-адресу 192.168.10.1, то на інтерфейсі маршрутизатора, підключеному до сегменту мережі з клієнтами, слід виконати команду: *ip helper-address 192.168.10.1*. Якщо ця команда не буде вказана або буде задана некоректно (наприклад, з помилковою IP-адресою), то жодні запити від клієнтів до DHCP-сервера не надійдуть, і клієнти не отримають IP-адреси автоматично. Щоб перевірити правильність такої конфігурації маршрутизатора, адміністратор має виконати кілька діагностичних дій.

Крок 1. Спочатку необхідно впевнитися, що команда *ip helper-address* налаштована на правильному інтерфейсі, вона повинна бути застосована саме до того інтерфейсу, який приймає DHCP-запити від клієнтів – зазвичай це інтерфейс, що з'єднаний з локальною мережею клієнта. Адреса, зазначена у команді, повинна точно відповідати IP-адресі сервера DHCP. Для перевірки можна скористатися командою *show ip interface*, яка покаже, чи активовано пересилання DHCP-запитів на конкретному інтерфейсі.

Крок 2. Далі слід упевнитися, що на маршрутизаторі не відключено глобальні сервіси DHCP. Існує команда *no service dhcp*, яка деактивує функціонал як DHCP-сервера, так і ретранслятора. Якщо ця команда була введена, то маршрутизатор не зможе обробляти чи пересилати DHCP-запити. Важливо зазначити, що служба DHCP активована на маршрутизаторі за замовчуванням, тому відсутність команди *no service dhcp* у конфігурації вважається нормальною.

Щоб перевірити це, можна виконати команду *show running-config | include no service dhcp*. Якщо у відповіді нічого не відображається, це означає, що служба активна. Якщо ж у вихідних даних буде рядок *no*

service dhcp, то потрібно його видалити, щоб DHCP-функціональність запрацювала.

У випадку, коли маршрутизатор налаштовано як ретранслятор, але DHCP-клієнти все одно не отримують адреси, слід перевірити, чи взагалі запити доходять до маршрутизатора.

Якщо запити не потрапляють на маршрутизатор, або не передаються далі, процес автоматичної конфігурації IP-адрес клієнтами через DHCP буде неможливим. Тому адміністратору потрібно провести діагностику, щоб упевнитися, що маршрутизатор отримує запити й правильно їх обробляє.

Один з найефективніших інструментів для діагностики проблем із DHCP на маршрутизаторі – це команда *debug ip dhcp server events*, яка дозволяє відстежувати події, пов'язані з роботою DHCP-сервера, зокрема, коли адреса призначається клієнту, коли оновлюється база даних оренди адрес,та коли DHCP-запит отримано або передано.

За допомогою цієї команди можна в реальному часі побачити, чи маршрутизатор приймає запити від клієнтів і як він на них реагує. Наприклад, якщо при активації цієї команди не спостерігається жодних подій у відповідь на клієнтські запити, це може свідчити про помилку в маршрутизації, фільтрації трафіку або неправильне налаштування VLAN.

Такий підхід дозволяє точно локалізувати проблему і впевнено усунути її, забезпечуючи належну роботу служби автоматичної IP-конфігурації в складних мережах із сегментацією.

Контрольні запитання та задачі до теми 17

Перелік теоретичних запитань

1. Яке основне призначення протоколу DHCPv4 у комп'ютерних мережах?
2. Які етапи включає процес динамічного отримання IP-адреси через DHCPv4?
3. Поясніть послідовність повідомлень DHCPDISCOVER, DHCPOFFER, DHCPREQUEST і DHCPACK.
4. У чому полягає різниця між статичним, автоматичним і динамічним виділенням адрес у DHCPv4?

5. Які поля входять у формат DHCPv4-повідомлень і яку інформацію вони містять?

6. Що таке оренда (lease) IP-адреси і які параметри її контролюють?

7. Як правильно налаштувати DHCPv4-сервер на маршрутизаторі Cisco?

8. Як працює механізм DHCPv4-ретрансляції (relay agent)?

9. Яка роль команди *ip helper-address* у ретрансляції DHCP-запитів?

10. Які типові помилки виникають при налаштуванні DHCPv4?

11. Як перевірити, чи працює служба DHCPv4 на маршрутизаторі?

12. Яке призначення команди *service dhcp*?

13. Як налаштувати маршрутизатор Cisco як DHCPv4-клієнт?

14. Яка команда дозволяє здійснювати діагностику подій DHCPv4 на маршрутизаторі?

15. Яка команда використовується для перевірки IP-налаштувань інтерфейсу маршрутизатора?

Перелік задач для самостійного опрацювання

Практичне завдання 1.

У середовищі моделювання Cisco Packet Tracer створіть мережу з трьох пристроїв: один маршрутизатор Cisco, один комутатор, один ПК.

Налаштуйте DHCPv4-сервер на маршрутизаторі.

ПК має автоматично отримати IP-адресу, шлюз і DNS. Перевірте командою *ipconfig* на ПК отримані параметри.

Практичне завдання 2.

Створіть дві локальні мережі (LAN1 і LAN2), підключені через маршрутизатор. DHCP-сервер має бути лише в LAN1. ПК у LAN2 має отримати IP-адресу через ретрансляцію.

Налаштуйте команду *ip helper-address* на відповідному інтерфейсі. Перевірте, чи ПК з LAN2 отримує IP-адресу.

Практичне завдання 3.

Надається мережа з неправильно налаштованим DHCP. Знайдіть і виправте помилки у конфігурації DHCP. Використовуйте команди *show running-config, show ip interface, debug ip dhcp server events*.

Тестове завдання 4.

Що робить команда *ip helper-address*?

а) Вмикає DHCP на маршрутизаторі

б) Додає додатковий шлюз

в) Пересилає широкомовні запити до вказаного сервера

г) Призначає адресу клієнту

Тестове завдання 5.

Яке з повідомлень ініціює процес DHCP?

а) DHCPOFFER

б) DHCPACK

в) DHCPDISCOVER

г) DHCPREQUEST

Тестове завдання 6.

Що відбудеться, якщо на маршрутизаторі вказано *no service dhcp*?

а) DHCP-сервер буде вимкнений

б) DHCP працює у тестовому режимі

в) DHCP працює лише на певних портах

г) Нічого, це дефолтна команда

Тестове завдання 7.

Яка команда відображає конфігурацію DHCP на маршрутизаторі?

а) show dhcp lease

б) show ip dhcp pool

в) ip dhcp excluded-address

г) debug dhcp server status

Перелік використаної та рекомендованої літератури

17.1. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

17.2. Dynamic Host Configuration Protocol [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc2131

17.3. Dynamic Host Configuration Protocol (DHCP) [Електронний ресурс]. – Режим доступу: https://www.fortinet.com/resources/cyberglossary/dynamic-host-configuration-protocol-dhcp

17.4. Configuring DHCPv4 [Електронний ресурс]. – Режим доступу: https://info.support.huawei.com/hedex/api/pages/EDOC1100413634/FEN1022J/01/resources/en-us_topic_0000001307928725.html

Тема 18. Технологія NAT

У сучасних комп'ютерних мережах спостерігається постійне зростання кількості пристроїв, підключених до мережі Інтернет. Це, у свою чергу, спричинило швидке вичерпання наявного простору IPv4-адрес. Одним із ключових рішень для подолання цієї проблеми стала технологія NAT (Network Address Translation) – трансляція мережевих адрес, що дозволяє локальним пристроям використовувати приватні IP-адреси, які не маршрутизуються в глобальній мережі, і водночас мати доступ до зовнішніх ресурсів Інтернету через одну або кілька публічних адрес. NAT є критично важливою технологією, яка використовується практично в усіх локальних мережах, що підключаються до Інтернету. Її основна функція полягає у заміщенні адреси джерела (або призначення) IP-пакетів, які проходять через маршрутизатор, що виконує роль шлюзу. Це дозволяє не тільки економити глобальні IP-адреси, але й підвищує рівень безпеки, приховуючи внутрішню структуру мережі від зовнішніх користувачів.

18.1. Визначення та необхідність використання технології NAT

Усі пристрої, які мають вихід в мережу Інтернет за допомогою публічних IPv4-адрес, повинні використовувати адреси, зареєстровані в одному з регіональних інтернет-реєстраторів RIR (Regional Internet Registry), таких як ARIN, RIPE NCC, APNIC тощо. Це забезпечує унікальність кожної публічної адреси в глобальній мережі. Хоча більшість організацій не купують IP-адреси безпосередньо, вони можуть орендувати їх у свого інтернет-провайдера. Але навіть у цьому випадку лише той, хто є офіційним власником (зареєстрованим у RIR) публічної адреси, має право призначити її якому-небудь пристрою.

Обсяг IPv4-адресного простору технічно обмежений: усього можливо призначити близько 4,3 мільярда унікальних адрес (2^{32}). Коли в 1981 році Боб Кан та Вінт Серф створювали протокол TCP/IP, включаючи IPv4, об'єм такого простору здавався достатнім. Проте з початком масового використання комп'ютерів, мобільних пристроїв та розгортання Інтернету, стало очевидним, що цього ресурсу недостатньо.

У відповідь на швидке зростання числа пристроїв, які підключаються до Інтернету, було розроблено кілька тимчасових рішень для збереження IPv4-адресного простору. Одним із таких рішень став механізм *NAT (Network Address Translation)* – трансляція мережевих адрес, а також введення діапазонів приватних IPv4-адрес, визначених у стандарті RFC 1918. Наприклад, адреси з діапазонів 192.168.0.0/16, 10.0.0.0/8 або 172.16.0.0/12 не можуть бути використані безпосередньо в мережі Інтернет.

Приватні адреси та призначення NAT

Через нестачу публічних IP-адрес більшість локальних мереж побудовані на основі приватних адрес, які дозволяють організаціям використовувати один і той самий діапазон приватних адрес у різних мережах без конфліктів. Але оскільки ці адреси не глобально унікальні, вони *не можуть* бути маршрутизовані через Інтернет. Щоб пристрій із приватною IP-адресою міг отримати доступ до мережі Інтернет, його адреса має бути перетворена в публічну. Цей процес і виконує NAT – маршрутизатор змінює вихідну адресу пакета на публічну, що дозволяє коректну маршрутизацію в глобальній мережі.

Наприклад, домашній роутер може мати публічну адресу 203.0.113.10, яка використовується для всього трафіку Інтернету, хоча пристрої в мережі мають адреси виду 192.168.1.x. Завдяки NAT, один такий роутер може обслуговувати сотні пристроїв одночасно.

Значення NAT і його вплив на Інтернет

NAT фактично дозволив відтермінувати вичерпання адресного простору IPv4. Без цього механізму кількість доступних публічних IP-адрес закінчилася б ще до 2000 року. Завдяки NAT одна публічна адреса може використовуватись одночасно сотнями або навіть тисячами пристроїв, які мають приватні адреси.

Втім, технологія NAT не позбавлена недоліків. Наприклад, NAT ускладнює встановлення вхідних з’єднань, що створює труднощі для деяких сервісів (ігрові сервери, VoIP тощо). Крім того, NAT порушує принцип "кінець-до-кінця" (end-to-end), закладений у початкову архітектуру Інтернету.

Переваги NAT – економія і безпека

Окрім збереження IP-адрес, NAT підвищує конфіденційність і безпеку локальної мережі. Внутрішні адреси не видно зовні, а отже,

зловмисник не може безпосередньо атакувати приватний хост через Інтернет.

У NAT-маршрутизаторі налаштовується так званий пул публічних адрес – тобто список публічних IP-адрес, якими маршрутизатор може оперувати. Коли пристрій з приватною адресою відправляє запит у глобальну мережу, NAT-маршрутизатор замінює його адресу на одну з адрес з пулу і запам'ятовує відповідність у NAT-таблиці.

NAT на межі тупикової мережі

Зазвичай NAT виконується на пограничному маршрутизаторі, який з'єднує локальну мережу з провайдером або іншими мережами. Часто такий маршрутизатор знаходиться на межі так званої *тупикової мережі* (*stub network*), яка має лише один маршрут до зовнішньої мережі. Коли внутрішній пристрій потребує виходу в Інтернет, трафік спрямовується до цього пограничного маршрутизатора, який виконує трансляцію адрес, дозволяючи пристрою успішно спілкуватись із зовнішнім світом.

Перехід до IPv6 – остаточне рішення

Хоча технологія NAT ефективно продовжила життя IPv4, остаточне вирішення проблеми вичерпання адрес криється в переході на IPv6 – новий інтернет-протокол, що має 128-бітний адресний простір (близько $3{,}4\times10^{38}$ адрес). Цей обсяг практично безмежний для сучасних і майбутніх потреб Інтернету. З поступовим впровадженням IPv6 використання NAT ставатиме менш необхідним, а мережі зможуть працювати за оригінальним принципом "кінець-до-кінця".

18.2. Термінологія NAT

У контексті роботи з перетворенням мережевих адрес NAT терміни «внутрішня мережа» та «зовнішня мережа» використовуються для позначення двох логічно розділених частин мережевої інфраструктури.

Внутрішньою мережею вважається та частина мережі, яка містить пристрої з приватними IP-адресами (зазвичай з діапазонів, визначених у RFC 1918: 10.0.0.0/8, 172.16.0.0/12, або 192.168.0.0/16). Саме ці адреси не маршрутизуються в Інтернеті. Ця мережа підлягає трансляції NAT.

Зовнішньою мережею називається все, що знаходиться за межами внутрішньої мережі, включаючи глобальну мережу Інтернет, з яким пристрої повинні обмінюватися даними.

Коли використовується NAT, IP-адреси можуть мати різне значення залежно від контексту – чи йде мова про приватну внутрішню мережу, чи про публічну глобальну мережу.

Крім того, важливо враховувати напрямок трафіку – чи це вхідний трафік із зовнішнього середовища до внутрішньої мережі, чи вихідний – від внутрішнього клієнта до Інтернету. NAT виконує перетворення адрес залежно від цього контексту.

Щоб точніше описати, як адреси змінюються при перетворенні NAT, використовується спеціалізована термінологія, яка включає чотири основні типи адрес (рис. 18.1).

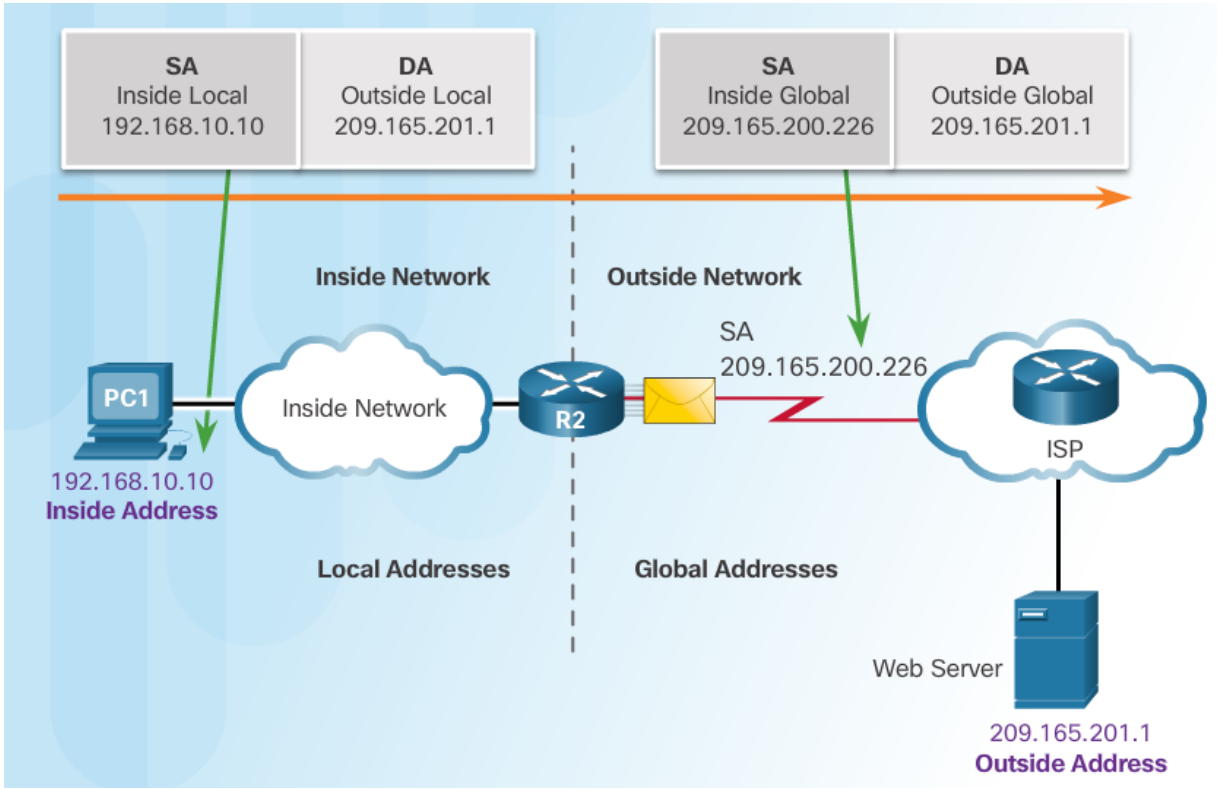


Рис. 18.1. Внутрішня локальна адреса (Inside Local), внутрішня глобальна адреса (Inside Global), зовнішня глобальна адреса (Outside Global) та зовнішня локальна адреса (Outside Local) при перетворенні NAT

Внутрішня локальна адреса (Inside Local) – це IP-адреса пристрою, який знаходиться всередині локальної мережі. Вона зазвичай є приватною.

Наприклад, комп’ютер у мережі має IP-адресу 192.168.10.10.

Внутрішня глобальна адреса (Inside Global) – це публічна IP-адреса, яка призначається NAT-сервером для відображення внутрішнього пристрою в Інтернеті.

Наприклад, той самий комп’ютер 192.168.10.10 зсередини може бути представлений у мережі як 209.165.200.226.

Зовнішня глобальна адреса (Outside Global) – це справжня публічна IP-адреса вузла призначення в Інтернеті, до якого внутрішній клієнт надсилає запити.

Наприклад, коли клієнт у внутрішній мережі звертається до сайту Google, його IP-адреса (наприклад, 142.250.190.78) виступає як зовнішня глобальна.

Зовнішня локальна адреса (Outside Local) – це адреса призначення, яка відображається з точки зору внутрішньої мережі. В деяких випадках NAT може перетворювати цю адресу, якщо використовується політика NAT, яка змінює зовнішні IP.

Наприклад, внутрішній пристрій може бачити IP зовнішнього сайту як 10.1.1.1, хоча насправді це лише трансляція справжньої зовнішньої глобальної адреси.

Ключовим моментом є те, що всі ці терміни слід трактувати з точки зору пристрою, IP-адреса якого підлягає трансляції NAT. Якщо це внутрішній клієнт, то "внутрішні" адреси відносяться до нього, а "зовнішні" – до його призначення. І навпаки, у разі вхідного трафіку (наприклад, коли сервер розміщений у внутрішній мережі, але має бути доступний з Інтернету), ці ролі можуть змінюватися.

18.3. Механізми перетворення мережевих адрес

Існує три основні методи перетворення мережевих адрес, які дозволяють пристроям у приватних мережах взаємодіяти з зовнішньою мережею Інтернет, використовуючи публічні IP-адреси. Ці методи забезпечують ефективне використання обмеженого адресного простору IPv4 і підвищують гнучкість управління мережею.

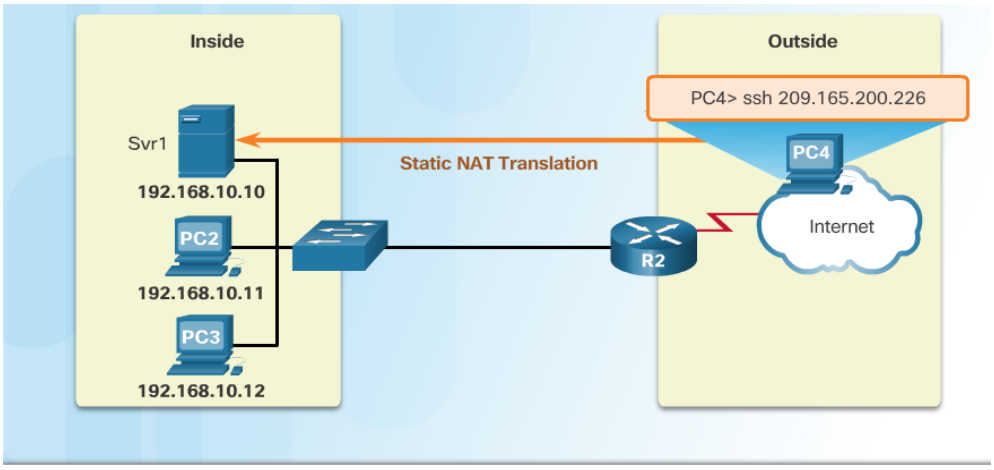
1. Статичне перетворення мережевих адрес (Static NAT)

Статичне перетворення NAT забезпечує пряме й постійне зіставлення між приватною IP-адресою пристрою всередині мережі та публічною IP-адресою, яка використовується для доступу з Інтернету.

Тобто кожному внутрішньому пристрою призначається конкретна публічна адреса, і це зіставлення залишається незмінним.

Наприклад, компанія має вебсервер з приватною IP-адресою 192.168.1.10. Щоб надати до нього доступ користувачам з Інтернету, адміністратор статично прив'язує до нього публічну адресу, скажімо, 203.0.113.10. Всі запити до цієї публічної адреси будуть автоматично перенаправлені на вебсервер. Такий підхід зручний для служб, які потребують постійного доступу, наприклад, вебсервери, поштові сервери або FTP.

На рис. 18.2 представлена схема роботи статичного перетворення NAT, де внутрішні приватні IP-адреси з локальної мережі (Inside) перетворюються на публічні глобальні IP-адреси для доступу ззовні (Outside). У цьому прикладі три пристрої у внутрішній мережі мають приватні IP-адреси: 192.168.10.10, 192.168.10.11 та 192.168.10.12. На маршрутизаторі R2 налаштовано статичне відображення кожної з них на унікальну публічну IP-адресу: 209.165.200.226, 209.165.200.227 та 209.165.200.228 відповідно. Коли користувач із зовнішньої мережі (ПК4) підключається до 209.165.200.226 через SSH, запит автоматично перенаправляється на сервер 192.168.10.10 всередині мережі. Така схема забезпечує постійне, фіксоване зіставлення між внутрішніми та зовнішніми IP-адресами, що є характерною рисою статичного NAT.



Static NAT Table

Inside Local Address	Inside Global Address - Addresses reachable via R2
192.168.10.10	209.165.200.226
192.168.10.11	209.165.200.227
192.168.10.12	209.165.200.228

Рис. 18.2. Статичне перетворення мережевих адрес (Static NAT)

Статичне перетворення NAT також дозволяє віддаленому персоналу безпечно підключатися до конкретного пристрою в мережі, не відкриваючи всю локальну інфраструктуру для зовнішнього світу. Наприклад, мережевий адміністратор може під’єднатися з дому через SSH до внутрішнього пристрою компанії, використовуючи постійно призначену публічну адресу.

Однак недоліком статичного NAT є те, що він вимагає наявності стільки ж публічних IP-адрес, скільки одночасно активних з'єднань, що неефективно за умов дефіциту IPv4-адрес.

2. Динамічне перетворення мережевих адрес (Dynamic NAT)

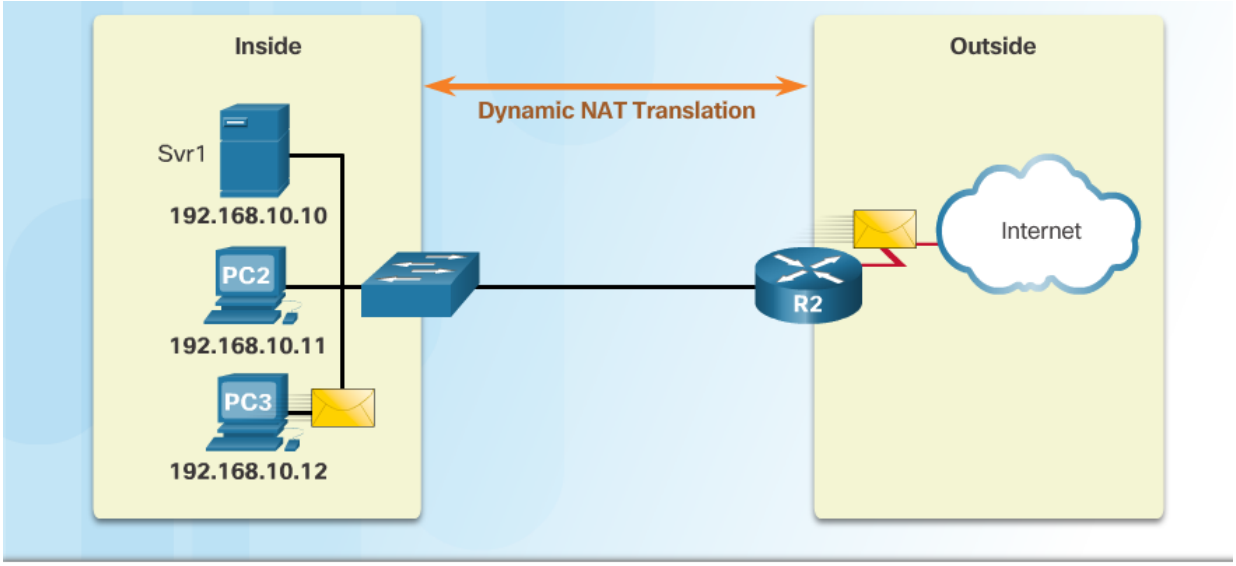
Динамічне перетворення NAT, на відміну від статичного, не використовує фіксоване зіставлення між приватними і публічними адресами. Натомість у маршрутизаторі налаштовується пул публічних адрес, з якого автоматично й тимчасово призначається адреса кожному пристрою, що намагається вийти в Інтернет.

Наприклад, офісна мережа має 50 комп’ютерів і лише 10 публічних IP-адрес у пулі. Коли перші 10 пристроїв ініціюють зовнішні з'єднання, їм тимчасово надаються публічні адреси. Якщо з'єднання закінчується – адреса повертається до пулу і може бути призначена іншому пристрою.

На рис. 18.3 показано роботу динамічного перетворення NAT, коли внутрішні (приватні) IP-адреси з локальної мережі перетворюються на глобальні (публічні) адреси зі спеціального пулу для доступу до Інтернету. У цьому прикладі комп’ютер PC3 з локальною адресою 192.168.10.12 використовує глобальну адресу 209.165.200.226, щоб надсилати дані в зовнішню мережу через маршрутизатор R2.

Таблиця внизу демонструє пул глобальних адрес (від 209.165.200.226 до 209.165.200.230), які маршрутизатор R2 може динамічно призначати внутрішнім пристроям при виході в Інтернет. Такий підхід дозволяє економно використовувати обмежений пул публічних IP-адрес, автоматично зв'язуючи їх із внутрішніми адресами лише на час активного з’єднання.

Цей метод зменшує потребу в кількості публічних адрес, однак має обмеження: кількість одночасних з’єднань не може перевищувати розмір пулу публічних адрес.



IPv4 NAT Pool

Inside Local Address	Inside Global Address Pool - Addresses reachable via R2
192.168.10.12	209.165.200.226
Available	209.165.200.227
Available	209.165.200.228
Available	209.165.200.229
Available	209.165.200.230

Рис. 18.3. Динамічне перетворення мережевих адрес (Dynamic NAT)

3. Перетворення адрес портів (PAT, NAT Overload)

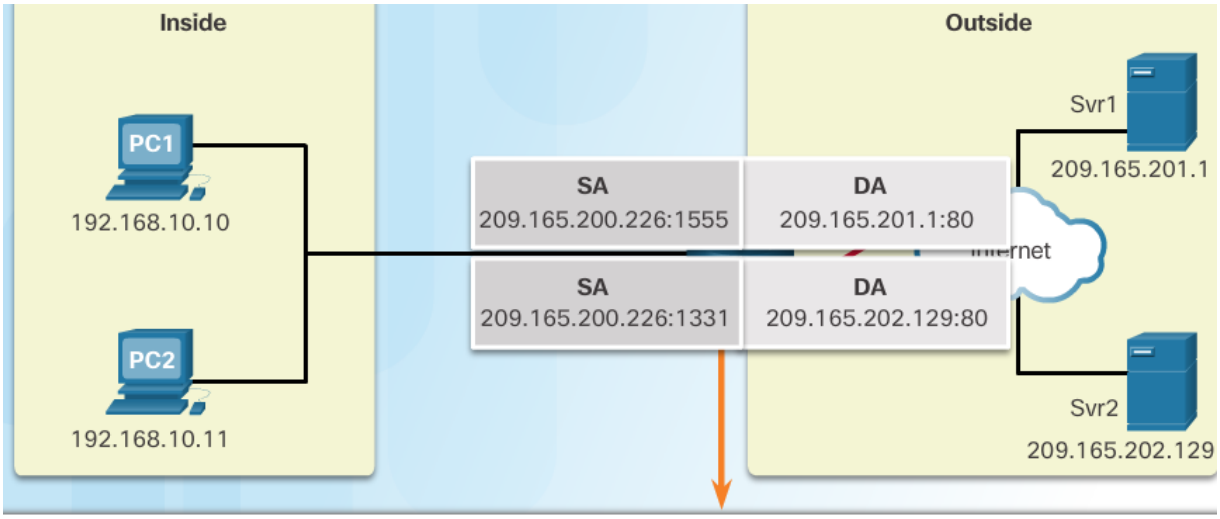
PAT (Port Address Translation) дозволяє великій кількості внутрішніх пристроїв використовувати одну або кілька публічних адрес завдяки механізму, що додає до кожного з'єднання унікальний номер порту. Таким чином, NAT-пристрій веде облік не тільки IP-адрес, але й портів.

Цей механізм дозволяє одночасно підключатися до мережі Інтернет багатьом пристроям з однією публічною IP-адресою. Наприклад, у типовому домашньому господарстві усі пристрої (смартфони, ноутбуки, смарт-ТВ) можуть виходити в Інтернет через один маршрутизатор, якому провайдер виділив лише одну публічну IP-адресу. Коли смартфон надсилає запит до вебсайту, маршрутизатор записує, що IP-адреса 192.168.0.2 з порту 45678 звернулася до певного ресурсу, і замінює адресу на публічну, додавши новий порт.

Коли відповідь приходить, маршрутизатор знає, якому пристрою її передати, аналізуючи номер порту.

На рис. 18.4 показано роботу PAT (або NAT з перевантаженням), де декілька пристроїв локальної мережі можуть використовувати одну глобальну IP-адресу для виходу в мережу Інтернет.

У прикладі ПК1 (192.168.10.10) і ПК2 (192.168.10.11) одночасно звертаються до двох зовнішніх серверів (209.165.201.1 і 209.165.202.129) на порт 80 (HTTP), але мають різні номери портів джерела (1555 і 1331), що дозволяє маршрутизатору R2 розрізняти трафік. В таблиці показано, як NAT з перевантаженням зіставляє внутрішні локальні IP-адреси та порти з однією глобальною IP-адресою (209.165.200.226), додаючи унікальні номери портів, щоб забезпечити ідентифікацію кожного з'єднання. Це дозволяє зекономити публічні IP-адреси, підтримуючи одночасні з'єднання багатьох внутрішніх хостів.



NAT Table with Overload

Inside Global IP Address	Inside Local IP Address	Outside Local IP Address	Outside Global IP Address
209.165.200.226:1555	192.168.10.10:1555	209.165.201.1:80	209.165.201.1:80
209.165.200.226:1331	192.168.10.11:1331	209.165.202.129:80	209.165.202.129:80

Рис. 18.4. Перетворення адрес портів (PAT, NAT Overload)

PAT є найпоширенішим типом NAT, оскільки забезпечує максимальну ефективність використання публічних адрес і дозволяє масштабувати мережі без додаткових витрат на адресний простір. Крім того, PAT підвищує безпеку, приховуючи внутрішню структуру мережі від зовнішнього світу. Пристрої, що не ініціюють з’єднання, не можуть

бути досягнуті зовні без спеціальних налаштувань (наприклад, Port Forwarding).

Таким чином, кожен з методів NAT має своє призначення:

- *статичний NAT* – для постійного доступу до конкретних внутрішніх ресурсів;
- *динамічний NAT* – для зменшення використання публічних IP, але з обмеженнями на кількість одночасних сесій;
- *PAT* – як найефективніше й найбільш універсальне рішення для більшості мереж, включаючи домашні та корпоративні.

18.4. Порівняння технологій NAT і PAT

Щоб краще зрозуміти, як працюють різні методи трансляції адрес у комп'ютерних мережах, важливо розібратися в ключових відмінностях між *NAT (Network Address Translation)* та *PAT (Port Address Translation)* – двома основними підходами до перетворення IP-адрес. Обидва методи використовуються для забезпечення зв'язку між пристроями з приватними IP-адресами та зовнішнім середовищем (наприклад,мережею Інтернет), однак реалізують це по-різному.

Почнемо з NAT у класичному вигляді. Цей метод трансляції працює за принципом відповідності "один до одного" між приватною та публічною IPv4-адресою. Тобто кожному пристрою всередині локальної мережі, який потребує доступу до Інтернету, призначається окрема публічна IP-адреса. Наприклад, якщо у мережі є три комп'ютери з IP-адресами 192.168.1.10, 192.168.1.11 і 192.168.1.12, то для кожного з них потрібно зарезервувати публічну IP – скажімо, 203.0.113.1, 203.0.113.2 та 203.0.113.3 відповідно. Це можливо лише у випадку, якщо є доступ до великого пулу публічних адрес, що у сучасних умовах – рідкість через їх обмежену кількість.

На противагу цьому, PAT (який також називають NAT Overload) дозволяє багатьом внутрішнім пристроям користуватися однією публічною IP-адресою, додаючи до неї унікальні номери портів.

Тобто PAT виконує трансляцію не лише IP-адреси, але й порту джерела. Це дає змогу одночасно підтримувати велику кількість з'єднань іззовні при використанні лише однієї публічної адреси.

Наприклад, п'ять комп'ютерів у внутрішній мережі з адресами 192.168.0.2 – 192.168.0.6 можуть усі виходити в Інтернет через одну

публічну адресу 203.0.113.10, але кожен використовуватиме різний порт: 203.0.113.10:5001, 203.0.113.10:5002, і так далі. Цю відповідність портів PAT записує у спеціальну таблицю на маршрутизаторі, що дозволяє розпізнати, який внутрішній клієнт відповідає яким зовнішнім запитам – цей механізм називається відстежуванням з'єднань.

У процесі PAT маршрутизатор записує у свою таблицю всі активні з'єднання між внутрішніми адресами і зовнішніми, включаючи порти TCP або UDP. Завдяки цьому вхідні пакети з Інтернету можуть бути точно перенаправлені на потрібний пристрій у внутрішній мережі, навіть якщо всі пристрої "маскуються" під одну публічну IP-адресу.

Але що відбувається з трафіком, який не використовує TCP або UDP, тобто не має портів транспортного рівня (Layer 4)? Сюди, наприклад, належать пакети, що використовують ICMPv4 (протокол контролю передачі, який використовується, наприклад, при виконанні команд типу *ping*). Ці пакети не мають номерів портів, а отже, PAT не може використовувати звичний механізм трансляції за портами.

У таких випадках PAT використовує інші поля в заголовках. Зокрема, для ICMPv4 використовується спеціальне поле, що називається *Query ID (ідентифікатор запиту)* – унікальний ідентифікатор, який присвоюється кожному запиту ICMPv4 (наприклад, ехо-запиту) і який зростає з кожним новим запитом.

PAT використовує цей ідентифікатор для того, щоб відрізнити і маршрутизувати відповідь на конкретний запит – аналогічно до того, як він використовує номер порту для TCP або UDP.

Наприклад, комп'ютер з IP 192.168.1.10 надсилає ехо-запит (*ping*) до зовнішнього сервера. PAT використовує Query ID = 130, щоб потім розпізнати відповідь і передати її назад саме цьому комп'ютеру.

Таким чином, PAT забезпечує підтримку не тільки стандартних транспортних протоколів, але й здатен обробляти інші типи мережевого трафіку, застосовуючи для кожного з них відповідний метод ідентифікації та трансляції.

18.5. Переваги і недоліки NAT

Перетворення мережевих адрес NAT стало важливим механізмом у сучасних мережах, особливо в умовах обмеженого простору IPv4, і має як численні переваги, так і певні недоліки.

Розглянемо переваги NAT.

1. Економія публічних IP-адрес та ефективне використання портів.

Однією з головних переваг NAT є можливість зберегти дефіцитний ресурс – публічні IPv4-адреси. За допомогою NAT з перевантаженням (PAT) багато внутрішніх пристроїв можуть використовувати одну загальну публічну IP-адресу для виходу в Інтернет. Наприклад, у домашній мережі маршрутизатор отримує одну публічну адресу від провайдера, але дозволяє одночасний доступ до Інтернету для всіх пристроїв у домі – ноутбуків, смартфонів, смарт-ТВ тощо. У цьому випадку кожен сеанс відрізняється унікальним портом, що дозволяє підтримувати множинні з'єднання.

2. Гнучкість і масштабованість підключень.

NAT дозволяє адміністраторам створювати різні пули адрес: основні, резервні та для балансування навантаження. Наприклад, у великій корпоративній мережі можуть бути налаштовані кілька пулів публічних адрес, що забезпечують відмовостійкість. Якщо один пул вичерпано або недоступний, NAT автоматично перейде до іншого. Це забезпечує безперебійний доступ користувачів до Інтернету навіть у разі збоїв.

3. Незалежність внутрішньої адресації від провайдера.

Завдяки NAT організація може використовувати приватну схему IP-адресації, яка не змінюється при зміні постачальника Інтернету. Наприклад, якщо компанія вирішує перейти від одного провайдера до іншого, змінюються лише зовнішні публічні адреси на маршрутизаторі, тоді як всі внутрішні IP-адреси залишаються незмінними. Це зменшує витрати й ризики, пов'язані з масштабною перенастройкою мережі.

4. Покращення безпеки.

NAT частково виконує функцію захисту, приховуючи внутрішню топологію мережі. Пристрої з приватними IP-адресами не видно в глобальному Інтернеті, що зменшує ймовірність зовнішніх атак.

Наприклад, хакер не може безпосередньо ініціювати з'єднання з пристроєм, IP-адреса якого не публікується.

Водночас слід пам'ятати, що NAT не замінює повноцінний міжмережевий екран, а лише доповнює систему захисту.

Розглянемо недоліки NAT.

1. Зниження продуктивності та збільшення затримок.

Процес NAT вимагає обробки кожного IP-пакета маршрутизатором. Зокрема, маршрутизатор повинен змінити IP-адресу в заголовку пакета, змінити порти TCP/UDP і перерахувати контрольні суми. Це додає затримку, особливо при першому з'єднанні.

Наприклад, при здійсненні дзвінка через протокол реального часу VoIP, може виникнути затримка, яка негативно впливає на якість звуку. У великих мережах або при високому навантаженні ці затримки стають ще помітнішими.

2. Втрата наскрізної адресації.

Деякі протоколи і застосунки потребують прямого з'єднання між джерелом і вузлом призначення. Наприклад, IP-телефонія, однорангові додатки (P2P), ігри, відеоконференції – усе це може не працювати через NAT або вимагати спеціального налаштування. Також виникають проблеми з цифровими підписами, які базуються на незмінності IP-адреси. Якщо адреса підмінюється NAT, підпис втрачає чинність.

3. Складність діагностики та трасування.

NAT ускладнює трасування маршрутів пакетів через мережу. Коли пакети змінюють адресу кілька разів під час проходження через різні NAT-пристрої, стає важко точно з'ясувати, де виникла проблема. Наприклад, у разі збоїв підключення адміністратору доводиться аналізувати NAT-таблиці та відповідність портів, що значно ускладнює діагностику.

4. Несумісність з деякими тунельними протоколами.

NAT може створювати конфлікти з протоколами тунелювання, зокрема, з пртоколом IPsec. Це пов'язано з тим, що NAT змінює заголовки пакетів, що порушує цілісність пакета, яку перевіряє IPsec. В результаті VPN-з'єднання можуть не встановлюватися або постійно обриватися без належного налаштування NAT Traversal (NAT-T).

5. Порушення роботи деяких служб.

Програми, які вимагають ініціалізації з'єднання ззовні (наприклад, вебсервери чи FTP у активному режимі), можуть не працювати через NAT без явного перенаправлення портів (*port forwarding*). Також сервіси, що використовують нестандартні або нестабільні порти, можуть бути блоковані NAT-пристроєм.

Наприклад, якщо два пристрої за різними NAT-мережами намагаються з’єднатися один з одним у грі чи відеочаті, з’єднання може не відбутися взагалі.

Отже, NAT – це потужний інструмент, який дозволяє ефективно використовувати обмежений адресний простір і підвищує гнучкість мережі. Проте NAT вносить складності у взаємодію з деякими службами, впливає на продуктивність і ускладнює діагностику.

Для повноцінної роботи мережі NAT повинен застосовуватися з урахуванням усіх технічних особливостей і в поєднанні з іншими технологіями, такими як фаєрволи, VPN та спеціальні налаштування NAT для окремих сервісів.

18.6. Налаштування статичного NAT

Статичне перетворення NAT (Static Network Address Translation) – це вид трансляції мережевих адрес, при якому внутрішній IP-адресі постійно й однозначно відповідає певна зовнішня (публічна) IP-адреса. Тобто кожному внутрішньому пристрою (наприклад, серверу) закріплюється фіксована публічна IP-адреса, через яку він доступний іззовні. Таке зіставлення встановлюється вручну адміністратором і не змінюється під час роботи мережі.

Цей підхід особливо корисний у ситуаціях, коли зовнішні користувачі повинні мати стабільний доступ до певного внутрішнього ресурсу – наприклад, вебсервера, FTP-сервера або камери відеоспостереження.

Наприклад, якщо вебсервер у внутрішній мережі має адресу 192.168.10.254, йому можна призначити публічну адресу 209.165.201.5. У такому випадку будь-який користувач із зовнішньої мережі (Інтернету), який надсилає запити на адресу 209.165.201.5, фактично буде звертатися до сервера 192.168.10.254.

Розглянемо основні кроки налаштування статичного NAT.

Крок 1. Створення відповідності адрес.

На першому етапі адміністратор конфігурує однозначну відповідність між внутрішньою локальною адресою (адресою пристрою у локальній мережі) та внутрішньою глобальною адресою (яка буде видима з мережі Інтернет).

Це налаштовується командою на маршрутизаторі, наприклад:

ip nat inside source static 192.168.10.254 209.165.201.5

Це означає, що будь-який пакет, відправлений з IP 192.168.10.254, буде виглядати ззовні як пакет з IP 209.165.201.5.

Крок 2. Визначення NAT-інтерфейсів.

Далі слід вказати, які інтерфейси маршрутизатора є внутрішніми, а які – зовнішніми щодо NAT. Наприклад:

interface Serial0/0/0

ip nat inside

interface Serial0/1/0

ip nat outside

У цьому прикладі інтерфейс Serial0/0/0 працює з локальною мережею (вхідні пакети від внутрішніх клієнтів), а Serial0/1/0 – з зовнішньою (з мережею Інтернет).

Пакети, які надходять із внутрішньої мережі на інтерфейс Serial0/0/0 від IP-адреси 192.168.10.254, будуть транслюватися маршрутизатором у публічну адресу 209.165.201.5 перед відправкою в Інтернет. Навпаки, коли відповіді з Інтернету надходять на публічну адресу 209.165.201.5 через зовнішній інтерфейс (Serial0/1/0), маршрутизатор змінює адресу призначення на внутрішню (192.168.10.254) і передає трафік вебсерверу у внутрішній мережі.

Після налаштування можна перевірити його працездатність за допомогою таких діагностичних команд:

show ip nat translations – відображає таблицю всіх активних трансляцій NAT, у випадку статичного NAT зіставлення завжди присутнє у таблиці, навіть якщо немає активного трафіку;

show ip nat statistics – показує загальну статистику роботи NAT, зокрема кількість успішних трансляцій.

Щоб протестувати трансляцію, варто спочатку скинути статистику:

clear ip nat statistics

Після цього можна ініціювати з’єднання з вебсервером ззовні (наприклад, зробити HTTP-запит), і команда *show ip nat statistics* покаже збільшення кількості збігів, що свідчитиме про успішну трансляцію.

Перевага статичного NAT – це передбачуваність і постійна доступність внутрішніх ресурсів для зовнішніх користувачів. Його часто використовують для серверів, розташованих усередині організації, які повинні бути досяжні ззовні в будь-який час.

18.7. Налаштування і перевірка динамічного NAT

1. Принцип роботи динамічного NAT

На відміну від статичного NAT, який створює фіксоване, постійне відображення між конкретною внутрішньою (приватною) IP-адресою та конкретною глобальною (публічною) IP-адресою, динамічний NAT виконує автоматичне відображення. У цьому випадку маршрутизатор динамічно зіставляє приватні IP-адреси з доступними публічними адресами з певного пулу. Наприклад, якщо у внутрішній мережі є 10 пристроїв, які потребують доступу до Інтернету, а в наявності лише 5 публічних адрес у пулі – лише 5 пристроїв зможуть одночасно підключитися до Інтернету. Інші чекатимуть, поки звільниться адреса.

2. Вимоги до конфігурації інтерфейсів

Як і у випадку зі статичним NAT, динамічний NAT потребує визначення, які інтерфейси маршрутизатора будуть вважатися внутрішніми, а які – зовнішніми. Внутрішні інтерфейси – це ті, що підключені до локальної мережі, а зовнішні – ті, що підключені до глобальної мережі (наприклад, Інтернету). Це дозволяє маршрутизатору зрозуміти, в якому напрямку виконувати перетворення адрес.

3. Принцип "першим прийшов – першим обслужений"

Під час роботи динамічного NAT, усі пристрої у внутрішній мережі мають рівний доступ до пулу публічних адрес. Коли приватний пристрій хоче вийти в Інтернет, маршрутизатор шукає вільну адресу в пулі. Якщо така є – пристрій отримує її на час сесії. Якщо ж усі адреси вже зайняті, пристрій повинен чекати. Наприклад, якщо у компанії є 100 комп’ютерів, а в пулі – лише 50 публічних адрес, одночасний вихід в Інтернет можливий лише для 50 користувачів.

4. Кроки налаштування динамічного NAT

Крок 1. Створення пулу публічних адрес. Це здійснюється за допомогою команди *ip nat pool*, де вказуються початкова й кінцева адреси, а також маска підмережі або довжина префікса. Наприклад:

ip nat pool MYPOOL 203.0.113.10 203.0.113.20 netmask 255.255.255.0

Крок 2. Створення ACL (Access Control List), який визначає, які внутрішні адреси мають бути перетворені. Наприклад:

access-list 1 permit 192.168.1.0 0.0.0.255

Крок 3. Прив’язка ACL до пулу. Це робиться за допомогою команди:

ip nat inside source list 1 pool MYPOOL

Крок 4. Позначення внутрішніх інтерфейсів NAT:

interface FastEthernet0/0

ip nat inside

Крок 5. Позначення зовнішніх інтерфейсів NAT:

interface Serial0/0/0

ip nat outside

5. Перевірка і моніторинг перетворень

Після налаштування NAT, можна використовувати команду *show ip nat translations*, щоб переглянути активні динамічні (та статичні) перетворення адрес. Тут відображаються вихідна й призначена IP-адреса, порт, протокол та інше.

Приклад:

Pro Inside global Inside local Outside local Outside global

tcp 203.0.113.10:1025 192.168.1.100:1025 198.51.100.5:80 198.51.100.5:80

6. Тайм-аут і очищення перетворень

Динамічні записи мають обмежений термін дії – за замовчуванням це 24 години. Його можна змінити за допомогою:

ip nat translation timeout 3600

Щоб вручну очистити записи NAT, використовується команда *clear ip nat translation*. Якщо її викликати без параметрів, буде очищено всі динамічні перетворення. Наприклад:

clear ip nat translation *

Це корисно під час тестування конфігурації NAT, щоб оновити таблицю й уникнути конфліктів.

7. Додаткові команди для діагностики

Команда *show ip nat statistics* дозволяє переглянути статистику NAT: кількість активних перетворень, скільки адрес у пулі та скільки з них вже використовуються.

Команда *show running-config* допоможе переглянути поточну конфігурацію NAT, у тому числі списки доступу, призначення інтерфейсів, пул адрес тощо.

Хоча найбільш поширений випадок – це перетворення приватних адрес у публічні, NAT також може використовуватись між іншими парами адрес, наприклад, між двома приватними мережами або двома публічними діапазонами для сегментації мережевих ресурсів.

18.8. Налаштування та перевірка PAT

Перетворення адрес портів PAT (Port Address Translation), також відоме як NAT з перевантаженням, дозволяє значно економити глобальні (публічні) IP-адреси. Його суть полягає в тому, що маршрутизатор може використовувати одну публічну IPv4-адресу для обслуговування багатьох пристроїв, що мають приватні IP-адреси всередині локальної мережі.

Наприклад, якщо у компанії є лише одна публічна IP-адреса, PAT дозволяє одночасно підключити до Інтернету десятки або навіть сотні внутрішніх клієнтів, використовуючи цю єдину адресу.

Щоб досягти цього, PAT розрізняє з'єднання не лише за IP-адресами, а й за номерами портів TCP або UDP. Тобто, якщо внутрішні вузли надсилають пакети з однаковою публічною адресою, маршрутизатор створює унікальні відповідності, додаючи до них різні номери портів.

Наприклад, три пристрої з адресами 192.168.1.10, 192.168.1.11 і 192.168.1.12 можуть бути представлені в Інтернеті як 203.0.113.5:10001, 203.0.113.5:10002 і 203.0.113.5:10003 відповідно. Таким чином, маршрутизатор знає, якому пристрою належить кожен пакет.

Теоретично, одна публічна IP-адреса може підтримувати до 65 536 унікальних комбінацій портів (оскільки діапазон TCP/UDP портів – від 0 до 65535). Але з практичної точки зору, кількість внутрішніх адрес, які реально можуть бути обслуговані, складає приблизно 4000. Це пов'язано з обмеженнями на використання певних портів, а також з потребами самих програм.

Існує два варіанти налаштування PAT, залежно від того, скільки публічних адрес виділяє провайдер. У першому випадку організація отримує кілька публічних IP-адрес. Вони об’єднуються в пул і PAT використовує кожну адресу з цього пулу разом із портами. Наприклад, якщо провайдер виділив три адреси, PAT може використовувати їх для обслуговування сотень внутрішніх пристроїв.

У другому випадку, коли виділяється лише одна публічна IP-адреса (часто це адреса інтерфейсу маршрутизатора, підключеного до Інтернету), PAT виконує всі перетворення через неї. Це типовий сценарій для домашніх мереж або малих офісів.

Процес PAT однаковий незалежно від того, використовується одна адреса чи пул. Для перевірки налаштувань використовуються ті ж самі команди, що й для NAT – наприклад, *show ip nat translations*, яка виводить таблицю відповідностей між внутрішніми адресами і зовнішніми портами.

Переадресація портів (Port Forwarding)

Переадресація портів – це окрема технологія, яка дозволяє направляти вхідні запити з Інтернету на конкретний вузол у приватній мережі. Наприклад, якщо потрібно зробити локальний вебсервер (з адресою 192.168.1.100) доступним ззовні, ми можемо налаштувати маршрутизатор на переадресацію порту 80 (HTTP) на цю адресу. Таким чином, усі звернення до 203.0.113.5:80 автоматично перенаправлятимуться до 192.168.1.100:80.

Це особливо важливо для програм однорангового обміну файлами (P2P), онлайн-ігор, FTP-серверів та інших сервісів, які потребують вхідних з'єднань. Оскільки NAT зазвичай не дозволяє ініціацію з'єднань з Інтернету до приватної адреси, без переадресації такі програми не працюватимуть.

Налаштування переадресації портів зазвичай включає вказання публічної IP-адреси (зовнішнього інтерфейсу), порту, який буде відкритий, та IP-адреси внутрішнього вузла, на який перенаправляється трафік. Наприклад, для доступу до внутрішнього вебсервера через нестандартний порт 8080, можна вказати URL у вигляді: http://203.0.113.5:8080.

Команди налаштування переадресації портів в IOS подібні до статичного NAT, з тією різницею, що додається ще і номер порту. Для моніторингу – також можна використовувати *show ip nat translations*.

NAT у середовищі IPv6

З переходом до IPv6 потреба у NAT значно зменшилась, оскільки адресного простору в новому протоколі більше ніж достатньо – 340 ундециліонів унікальних адрес. Це дозволяє кожному пристрою мати

свою глобальну унікальну IP-адресу без необхідності приховування за одним публічним обліковим записом.

Однак у деяких випадках все ж використовується певна форма NAT для IPv6. Зокрема, для створення приватних мереж з обмеженим доступом до Інтернету застосовуються унікальні локальні адреси (ULA). Вони мають префікс FC00::/7 (зазвичай FD00::/8) і використовуються для внутрішньої взаємодії в організації. Вони не маршрутизуються в Інтернеті, однак допомагають організовувати адресний простір усередині об'єкта без конфліктів.

NAT для IPv6 найчастіше використовується не для приховування адрес, а як засіб транспортування між IPv6 і IPv4. Наприклад, технології NAT64, DNS64, Dual Stack або тунелювання (6to4, ISATAP) дозволяють організаціям поступово переходити на новий протокол, забезпечуючи сумісність зі старими системами.

Усунення несправностей NAT

Якщо виникають проблеми з NAT (у тому числі з PAT або переадресацією портів), важливо мати чіткий план діагностики:

З’ясувати мету NAT – яку функцію він має виконувати, які типи NAT використовуються.

Перевірити таблицю перетворень командою *show ip nat translations*. Вона покаже, чи взагалі створюються перетворення, і які саме.

Очистити таблиці NAT та застосувати команди *debug ip nat* або *debug ip nat detailed*, щоб побачити, як маршрутизатор обробляє кожен пакет.

Перевірити маршрутизацію – переконатися, що пакети після перетворення NAT прямують у правильному напрямку.

Для моніторингу загального стану NAT доцільно використовувати команду *show ip nat statistics*, яка показує кількість активних перетворень, налаштування пулу адрес тощо.

У складних мережах команду *debug ip nat detailed* можна використовувати для точнішого пошуку помилок, оскільки вона надає детальну інформацію про кожне перетворення та можливі проблеми, такі як відсутність доступних публічних адрес.

У сучасних умовах більшість інтернет-користувачів в Україні, як і в багатьох інших країнах світу, все ще активно користуються

технологією NAT (Network Address Translation) через обмежену кількість доступних публічних IPv4-адрес.

Незважаючи на існування IPv6, повний перехід на нову версію протоколу відбувається повільно через технічні, економічні та організаційні бар'єри. Провайдери в Україні часто використовують NAT не лише на рівні домашніх маршрутизаторів, а й у вигляді так званого Carrier-Grade NAT (CGNAT), коли одна публічна адреса розподіляється на сотні абонентів.

У багатьох європейських країнах, таких як Німеччина, Франція чи Польща, ситуація подібна: масове використання NAT залишається стандартною практикою, особливо серед домашніх користувачів та малого бізнесу. Проте деякі провайдери вже активно впроваджують IPv6, пропонуючи клієнтам як IPv6-адреси, так і подвійний стек (Dual Stack) – одночасну підтримку IPv4 та IPv6. Це дозволяє поступово відмовлятися від NAT та зменшувати пов'язані з ним обмеження, зокрема проблеми з прямим доступом до пристроїв, що важливо, наприклад, для роботи серверів, IP-камер або ігрових сервісів.

У США та деяких країнах Азії, зокрема Японії та Південної Кореї, спостерігається більша динаміка впровадження IPv6. Великі оператори мобільного зв'язку вже повністю перейшли на IPv6 для мобільних користувачів, а технологія NAT все частіше використовується лише як проміжне рішення.

У Китаї ситуація з використанням NAT є особливо показовою через величезну кількість інтернет-користувачів – понад 1 мільярд людей. Через обмежений пул глобальних IPv4-адрес китайські провайдери змушені масово впроваджувати Carrier-Grade NAT (CGNAT), що дозволяє використовувати одну публічну IP-адресу для тисяч користувачів одночасно. Це рішення є економічно доцільним, але водночас створює технічні обмеження, зокрема ускладнює реалізацію прямих з'єднань (наприклад, для P2P-сервісів, IP-камер, VoIP та онлайн-ігор), погіршує продуктивність і ускладнює моніторинг трафіку.

Водночас Китай є одним із світових лідерів у просуванні та впровадженні IPv6. Ще з 2017 року уряд КНР прийняв стратегію активного розширення використання IPv6 у державному та приватному секторах. Однією з цілей було повністю перейти на IPv6 до 2030 року. В результаті цього плану великі китайські провайдери, такі як China

Telecom, China Unicom та China Mobile, почали впроваджувати Dual Stack-рішення – одночасну підтримку IPv4 та IPv6 – у своїх мережах, особливо в мобільному секторі.

Однак, попри значний прогрес у розгортанні IPv6, більшість користувачів у Китаї все ще отримують доступ до Інтернету через NAT-механізми. Причинами цього є наявність величезної кількості застарілих пристроїв, що не підтримують IPv6, програмне забезпечення, не оптимізоване для нового протоколу, та інерція корпоративного сегменту, де перехід на нову архітектуру вимагає значних інвестицій та навчання персоналу.

Таким чином, хоча NAT залишається домінуючим механізмом доступу до Інтернету в більшості країн, світова тенденція полягає в поступовому відмовленні від нього на користь повноцінного використання IPv6. Проте повний перехід потребує значного часу та ресурсів, тому NAT ще довго залишатиметься важливою частиною мережевої архітектури як в Україні, так і в глобальному масштабі.

Контрольні запитання та задачі до теми 18

Перелік теоретичних запитань

1. Що таке NAT (Network Address Translation) і яка його головна мета?
2. Чому виникла необхідність у використанні NAT у сучасних мережах?
3. Що таке приватні та публічні IP-адреси? Які діапазони IP-адрес вважаються приватними?
4. Поясніть терміни: inside local, inside global, outside local, outside global.
5. У чому полягає відмінність між статичним NAT, динамічним NAT і PAT?
6. Як працює механізм статичного перетворення NAT? Наведіть приклад.
7. Яким чином динамічний NAT призначає публічні адреси?
8. Що таке пул NAT-адрес, і як його створити?
9. У чому особливість технології PAT (Port Address Translation)? Як вона оптимізує використання адрес?
10. Які переваги дає NAT з точки зору безпеки мережі?

11. Які недоліки має NAT з точки зору роботи мережевих протоколів?
12. Як налаштувати статичне NAT на маршрутизаторі Cisco?
13. Як налаштувати динамічне NAT з використанням ACL?
14. Як перевірити статус NAT за допомогою команд CLI?
15. Якими командами можна очистити або переглянути таблицю NAT на маршрутизаторі?

Перелік задач для самостійного опрацювання

Практичне завдання 1. Налаштування статичного NAT.

У середовищі моделювання Cisco Packet Tracer потрібно створити мережу, де вебсервер із приватною IP-адресою буде доступний ззовні за публічною адресою. Умови: сервер: 192.168.1.10, публічна IP-адреса: 203.0.113.10. Налаштуйте статичне NAT так, щоб *ping* з зовнішнього ПК проходив до сервера.

Практичне завдання 2. Налаштування динамічного NAT.

У середовищі моделювання Cisco Packet Tracer потрібно забезпечити доступ комп'ютерам у внутрішній мережі до Інтернету через пул публічних адрес. Умови: внутрішні ПК: 192.168.1.0/24, пул публічних адрес: 203.0.113.10 - 203.0.113.20. Застосувати ACL та перевірити перетворення командою *show ip nat translations*.

Практичне завдання 3. Налаштування PAT (NAT Overload).

У середовищі моделювання Cisco Packet Tracer потрібно забезпечити спільний доступ усіх клієнтів до Інтернету через одну публічну IP-адресу. Умови: внутрішня мережа: 192.168.2.0/24, зовнішня адреса інтерфейсу: 198.51.100.1. Налаштувати NAT overload і перевірити роботу з Internet Simulation Mode.

Тестове завдання 4.

Яке твердження вірне щодо статичного NAT?

а) Використовує пул адрес

б) Працює тільки з PAT

в) Встановлює постійне відображення між IP-адресами

г) Не потребує конфігурації інтерфейсів

Тестове завдання 5.

Що означає ip nat inside source list 1 pool MYPOOL?

а) Визначає NAT таймаут

б) Зв'язує ACL зі списком маршрутизації

в) Пов’язує список дозволених адрес із пулом NAT

г) Очищає записи NAT

Тестове завдання 6.

Яка команда призначає інтерфейс як внутрішній для NAT?

A. ip nat outside

B. ip nat inside

C. ip nat pool

D. ip access-list standard

Тестове завдання 7.

4. Що буде, якщо пул NAT вичерпаний при динамічному NAT?

A. Внутрішні адреси перетворяться автоматично

B. Пристрій отримає помилку або чекатиме в черзі

C. Буде використано PAT

D. Всі з’єднання перериваються

Перелік використаної та рекомендованої літератури

18.1. Cisco Networking Academy [Електронний ресурс]. – Режим доступу: https://www.netacad.com

18.2. IP Network Address Translator (NAT) Terminology and Considerations [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc2663

18.3. Traditional IP Network Address Translator (Traditional NAT) [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc3022

18.4. IANA IPv4 Special-Purpose Address Registry [Електронний ресурс]. – Режим доступу: https://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml

ТЕРМІНОЛОГІЧНИЙ ПОКАЖЧИК

AAA (Authentication, Authorization, Accounting) ……..…….. 234
ACE (Access Control Entries) ……364
ACL (Access Control Lists) …….. 360
ARP (Address Resolution Protocol) ……73, 130
ARPANET ……...15
ASIC (Application-Specific Integrated Circuit) …… 258
AUX-порт (Auxiliary Port) …… 55
BGP (Border Gateway Protocol) ……...…… 326
BitTorrent ……. 205
Bluetooth ……83
Cable Innovator ……16
CAM (Content Addressable Memory) ……...257
CIDR (Classless Inter-Domain Routing) ……..185
Cisco IOS (Internetwork Operating System) ……51
CLI (Command Line Interface) …….. 51
CSMA (Carrier Sense Multiple Access) ……. 97
CSMA/CA (з уникненням колізій) …… 98
CSMA/CD (з виявленням колізій) …… 97
DDoS (Distributed Denial of Service) ……..…… 229
DHCP (Dynamic Host Configuration Protocol) ……. 217,388
Dial-up …… 20
DNS (Domain Name System) …….. 214
DoS (Denial of Service) …… 229
DTP (Dynamic Trunking Protocol) ……...…… 287
Dynamic NAT ……... 413
EGP (Exterior Gateway Protocol) ……. 331
EIGRP (Enhanced Interior Gateway Routing Protocol) ……...…….. 326,334
EtherChannel рівня 3 ……...……. 136
FTP (File Transfer Protocol) ……. 220
FTTx (Fiber To The X) ……..……. 20
Gnutella ……. 205
GUI (Graphical User Interface) …….. 51
HTTP (Hypertext Transfer Protocol) …… 207
HyperTerminal …… 57

IAB (Internet Architecture Board) …….. 40
IANA (Internet Assigned Numbers Authority) ……..….. 41,186
ICANN (Internet Corporation for Assigned Names and Numbers) …… 40
IEEE (Institute of Electrical and Electronics Engineers) …… 38
IEEE 802.11 …… 84
IEEE 802.15 …… 85
IEEE 802.16 …… 85
IEEE 802.1Q …….. 280
IEEE 802.20 …… 85
IEEE 802.22 …… 85
IETF (Internet Engineering Task Force) …… 40
IGP (Interior Gateway Protocol) …… 330
IGRP (Interior Gateway Routing Protocol) ……... 334
IMAP (Internet Message Access Protocol) …… 213
IPsec (Internet Protocol Security) …….. 243
IPv4 ……... 141
IPv6 ……... 144
IP(Internet Protocol) …….. 129
IRTF (Internet Research Task Force) …….. 41
IS-IS (Intermediate System to Intermediate System) ……. 326,338
ISO (International Organization for Standardization) …… 39
ISOC (Internet Society) ……... 40
LAN (Local Area Network) ……..…….. 13
Li-Fi (Light Fidelity) ……... 87
LLC (Logical Link Control) …… 93
MAC (Media Access Control) …… 92, 123, 129
MAN (Metropolitan Area Network) …… 13
NAT (Network Address Translation) ……..…….. 407
NIC (network interface controller/card) …….. 27
NRZ (Non-Return to Zero) …… 105
NVRAM (Non Volatile Random Access Memory) ……... 301
OSI (Open Systems Interconnection) …….. 41
OSPF (Open Shortest Path First) …… 326, 337
OSPF з кількома областями (multi-area OSPF) …….…… 347
OSPF з однією областю (single-area OSPF) …… 347
P2P-мережа …….. 203

PAN (Personal Area Network) …… 13
PAT (Port Address Translation) …… 414
POP (Post Office Protocol) …… 213
PuTTY …… 56
RADIUS …… 235
RIP (Routing Information Protocol) …….. 334
RZ (Return to Zero) ……... 107
SDN (Software-Defined Networking) …… 14
SecureCRT …….. 57
SLAAC (Stateless Address Autoconfiguration) …… 191
SMB (Server Message Block) ……... 222
SMTP (Simple Mail Transfer Protocol) …… 212
SNMP (Simple Network Management Protocol) …… 41
SSH (Secure Shell) …….. 54, 244
Static NAT …….. 411
STP (Shielded Twisted Pair) …… 79
Submarine Cable Map …… 19
SVI (Switch Virtual Interface) ……... 136
TCP (Transmission Control Protocol) ……... 155
TCP/IP (Transmission Control Protocol / Internet Protocol) …….. 43
Telnet (TELecommunication NETwork) …… 54
Tera Term …… 56
Terminal (macOS) ……... 57
UDP (User Datagram Protocol) …….. 158
UTP (Unshielded Twisted Pair) …….. 79
VLAN (Virtual Local Area Networks) …….. 275
VLSM (Variable Length Subnet Mask) …….. 195
VPN (Virtual Private Network) …….. 241
WAN (Wide Area Network) …… 13
WEP (Wired Equivalent Privacy) …….. 249
Wireshark …….. 175
WPA (Wi-Fi Protected Access) …… 250
Амплітудна модуляція (AM) ……..…….. 113
Аналогове кодування …….. 113
Асинхронний метод передачі …….. 114
Асоціативний доступ …… 97

Багатоадресна доставка 69
Багатомодовий оптоволоконний кабель 81
Біполярне кодування 105
Блочне кодування 111
Брандмауер 23
Вбудовані комп'ютерні системи 12
Віртуалізація 37
Деінкапсуляція 71
Деревоподібна топологія мережі 96
Диференційне манчестерське кодування 111
Зіркоподібна топологія мережі 96
Імпульсно-кодова модуляція (ІКМ) 114
Інкапсуляція 70
Інтерфейс 28
Інфрачервоне випромінювання 83
Керований доступ 97
Кільцева топологія мережі 96
Кінцеві пристрої 25
Кластерні системи 12
Клієнт-серверна архітектура 13
Клієнтські пристрої 24
Коаксіальні кабелі 79
Кодування в комп'ютерних мережах 101
Кодування сигналів 76
Комп'ютерна система 12
Комп'ютерна мережа 13
Комутатор 22
Комутатори рівня 2 136
Комутатори рівня 3 136
Логічна адреса 71
Логічна топологія мережі 29
Манчестерське кодування 109
Маршрутизатор 22, 300
Маска підмережі 178
Мережева інтерфейсна карта 27
Мережева операційна система 48

Мережевий кабель 23
Мережеві протоколи 23,37
Метод конкурентного доступу 97
Міжконтинентальні кабелі 19
Міжмережевий екран 236
Мікрохвилі 83
Мобільний Інтернет 20
Мобільні мережі (3G, 4G, 5G) 84
Модуляція 113
Мультиплексування 70
Напівдуплексний зв'язок 135
Небалансоване (нелінійне) бітове кодування 104
Одноадресна доставка 69
Одномодовий оптоволоконний кабель 80
Однорангова мережа 13
Оперативна пам'ять 300
Операційна система (ОС) 48
Оптичне волокно 79
Персональні комп'ютери (ПК) 12
Повнодуплексний зв'язок 135
Постійна пам'ять 301
Принцип цифрового подання 102
Проміжні пристрої 26
Пропускна здатність/ спроможність 22,76
Радіохвилі Wi-Fi 82
Розширені ACL 366
Сегментація 69
Сервер 12,24
Сервіси 25
Середовище передачі даних 25
Синхронний метод передачі 114
Сітчаста топологія мережі 97
Сокет 162
Стандартні ACL 366
Супутниковий зв'язок 84
Тегування 280

Точка доступу …… 23
Фазова модуляція (ФМ) …….. 114
Фізична адреса ……... 71
Фізична топологія мережі ……. 28
Фізичний порт …… 27
Флеш-пам’ять …….. 301
Хмара …….. 35
Хмарні обчислення ……... 12,35
Центри обробки даних (ЦОД) …….. 36
Частотна модуляція (ЧМ) ……... 114
Шинна топологія мережі …….. 96
Широкомовна доставка ……. 69
Шлюз за замовчуванням ……. 146

Електронне мережне навчальне видання

Олещенко Любов Михайлівна

ОСНОВИ КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

Підручник

Комп'ютерна верстка *А. А. Мушницький*
Коректура *Н. В. Мурашова*

Відповідальний за випуск Сирота С.В.

"ПРОСВІТА"
КИЇВ

Промислово-торговельна фірма «Просвіта»
у формі товариства з обмеженою відповідальністю.
01032, Київ, бульвар Т. Шевченка, 46,
тел. (067) 440 29 96, E-mail prosvita.kyiv@gmail.com
Свідоцтво ДК № 221 від 16.10.2000 р.

Підп. до публікації 08.04.2026.
Обл. вид арк. 18,8

Електронні текстові дані (1 файл: 10,2 Мбайт).