125-річчя від дня заснування
КПІ ім. Ігоря Сікорського
30-річчя від дня заснування
кафедри телекомунікацій

Сергій КРАВЧУК,
Дмитро МІНОЧКІН,
Олена ГЕТЬМАН

ОСНОВИ ПІДТРИМКИ МОБІЛЬНОСТІ В ІНФОКОМУНІКАЦІЙНИХ СИСТЕМАХ. МОДЕЛЮВАННЯ ПАКЕТНОГО САМОПОДІБНОГО ТРАФІКУ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Присвячено 125-річчю від дня заснування КПІ ім. Ігоря Сікорського,
30-річчю від дня заснування кафедри телекомунікацій

С. О. Кравчук, Д. А. Міночкін, О. В. Гетьман

ОСНОВИ ПІДТРИМКИ МОБІЛЬНОСТІ В ІНФОКОМУНІКАЦІЙНИХ СИСТЕМАХ. МОДЕЛЮВАННЯ ПАКЕТНОГО САМОПОДІБНОГО ТРАФІКУ

Навчальний посібник

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня магістра
за освітньою програмою «Інженерія та програмування інфокомунікацій»
за спеціальністю 172 «Телекомунікації та радіотехніка»

Електронне мережне навчальне видання
Київ
КПІ ім. Ігоря Сікорського
2023

Рецензент: *Скулиш М.А., докт. техн. наук, с.н.с., КПІ ім. І. Сікорського*

Відповідальний редактор *Кравчук С.О., докт. техн. наук, професор*

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
(протокол № 1 від 07.09.2023 р.)
за поданням Вченої ради навчально-наукового інституту телекомунікаційних систем
(протокол № 6 від 26.06.2023 р.)

Кравчук С.О.
Основи підтримки мобільності в інфокомунікаційних системах. Моделювання пакетного самоподібного трафіку. [Електронний ресурс]: навчальний посібник для студ. спеціальності 172 «Телекомунікації та радіотехніка» / С.О. Кравчук, Д.А. Міночкін, О.В. Гетьман; КПІ ім. Ігоря Сікорського. – Електронні текстові данні (1 файл: 3 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2023. – 116 с.: Іл.

Навчальний посібник містить поглиблений опис методів і способів дослідження самоподібного трафіка, розробка й використання нових підходів до адекватного моделювання трафіка сучасних пакетних телекомунікаційних мереж, особливо у світлі стрімкого розвитку й поширення високошвидкісних мереж, у тому числі безпроводових, що пред'являють дуже жорсткі вимоги до забезпечення високої якості обслуговування. Посібник структурно має два розділи, два додатки і побудований так, що складність матеріалу зростала з кожним наступним розділом. Навчальний посібник призначено для поглибленого вивчення самоподібного трафіка в інфокомунікаційних системах для здобувачів ступеня магістра за спеціальністю 172 «Телекомунікації та радіотехніка» з дисципліни «Основи підтримки мобільності в інфокомунікаційних системах», буде також корисним для аспірантів, наукових та інженерно-технічних працівників за напрямом інформаційно-телекомунікаційних систем та технологій.

УДК 621

Обсяг 7,25 авт. арк.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Перемоги, 37, м. Київ, 03056
https://kpi.ua
Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів
і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

© С. О. Кравчук, Д.А. Міночкін, О.В. Гетьман, 2023
© КПІ ім. Ігоря Сікорського, 2023

ЗМІСТ

Вступ 7
1. ТРАДИЦІЙНИЙ ПІДХІД ДО МОДЕЛЮВАННЯ ТРАФІКУ 9
1.1. Пуасонівська модель трафіку та успіх її застосування до телефонних мереж загального користування 9
1.2. Неадекватність пуасонівської моделі реальному трафіку пакетних мереж. 11
2. ФРАКТАЛЬНА МОДЕЛЬ ТРАФІКУ 17
2.1. Загальні положення 17
2.2. Розподіли з важкими хвостами 19
2.2.1. Розподіл Парето 21
2.2.2. Розподіл Вейбулла 21
2.2.3. Логарифмічно-нормальний розподіл 22
2.3. Самоподоба 23
2.3.1. Повільно спадна залежність 27
2.3.2. Властивості самоподібних процесів (з ПСЗ) 30
2.3.2.1. Повільно спадна дисперсія 30
2.3.2.2. Ефект Херста 32
2.3.3. Методи оцінки параметра Херста 35
2.3. 3.1. Метод *R*/*S* 35
2.3.3.2. Оцінка Уіттла 36
2.3.4. Фрактальний Броунівський рух 38
2.3.5. Сходження суперпозиції РВХ-процесів до самоподоби 40
2.4. Фрактальна природа трафіку реальних телекомунікаційних мереж 41
2.4.1. «Всесвітня павутина» 41
2.4.2 Загальна канальна сигналізація №7 42
2.4.3. Використання протоколів TCP, FTP і Telnet 43
2.4.4. Цифровий відеотрафік з VBR 44
2.4.5. Інші 44
2.5. Вплив самоподібного трафіку на продуктивність мереж 45

2.6. Обмеження самоподоби. Мультифрактали 52
3. Короткий огляд і класифікація методів 57
ВИСНОВКИ 59
Список використаної літератури 60
Додаток А. МЕТОДИ МОДЕЛЮВАННЯ САМОПОДІБНОГО ТРАФІКУ 66
1 Точні методи 66
1.1 Метод Хоскінга 66
1.2 Метод Холецького 69
1.3 Метод Девіса й Харте 72
2. Наближені методи 76
2.1. Метод стохастичного подання фрактального Броунівського руху 76
2. 2. Агрегування процесів джерел (*ON*-/*OFF*- метод) 77
2.3. Випадкове зміщення проміжних точок і вдалий випадковий приріст 79
2.4. Спектральне моделювання, метод Паксона і наближений циклічний метод 84
2.4.1. Спектральне моделювання 84
2.4.2. Метод Паксона 86
2.4.3. Наближений циклічний метод 89
2.5. Моделювання, засноване на вейвлетах 90
2.6. Інші методи 99
2.6.1. Метод хаотичних карт 99
2.6.2. Метод Декройсфонда та Лавода 100
2.6.3. Метод Джапарідзе і ван Зантена 101
Додаток Б . УТИЛІТИ ГЕНЕРАЦІЇ САМОПОДІБНОГО МЕРЕЖНОГО ТРАФІКУ 103

Перелік скорочень

АСШЗ – асимптотично самоподібний у широкому значенні;

ШПФ – швидке пертворення Фур’є;

ПСЗ – повільно спадна залежність;

ЗКС№7 – загальна канальна сигналізація №7;

РВХ – розподіл з важким хвостом;

ВЗСТ – випадкове зміщення проміжних точок процесу;

ССВЗ – строго самоподібний у вузькому значенні;

ССШЗ – строго самоподібний в широкому значенні;

ТМЗК – телефонна мережа загального користування;

ВВП – вдалий випадковий приріст;

ФБР – фрактальний Броунівський рух;

АТМ - Asynchronous Transfer Mode (асинхронний режим передачі)

FTP – File Transfer Protocol (Протокол передачі файлів)

IP – Internet Protocol (протокол Інтернету)

ISDN - Integrated Services Digital Network (цифрова мережа з інтеграцією послуг)

TCP – Transmission Control Protocol (протокол управління передачею)

Telnet - Протокол мережевого теледоступу

VOIP - Voice over IP (голос поверх IP)

WWW - WorldWideWeb («Всесвітня павутина»)

VBR - Variable Bit Rate (змінна бітова швидкість)

JPEG – Joint Photographic Experts Group (стандарт стиснення статичних зображень Об’єднаної групи експертів з фотозображень)

Вступ

Із часу появи перших інфокомунікаційних мереж розробка і застосування адекватних реальним фізичним процесам моделей трафіку було визначальним для їхнього розгортання та росту. Високоточні, з погляду максимального наближення до реальних процесів, моделі мережевого трафіку необхідні для планування і економічно-вигідного розподілу мережних ресурсів, а також забезпечення необхідної якості обслуговування.

Завдяки численним експериментам і дослідженням, проведеним за останнє десятиліття вченими різних країн світу, сьогодні можна із упевненістю стверджувати, що трафік сучасних інфокомунікаційних мереж з комутацією пакетів має особливу структуру, що не дозволяє використовувати при її проектуванні традиційні методи, засновані на Марківських моделях і формулах Ерланга, які добре себе зарекомендували в телефонних мережах з комутацією каналів. Особливості, про які йде мова, прийнято називати проявом ефекту самоподоби телетрафіку і у першу чергу вони проявляються в специфічній формі трафіку: у його реалізації завжди присутня деяка кількість достатньо сильних викидів на тлі відносно низького середнього рівня. Описане явище значно погіршує характеристики (збільшує втрати, затримки, джиттер) проходження совоподібного трафіку через мережу навіть у випадках, коли середня інтенсивність трафіку набагато нижче потенційно досяжної швидкості передачі в даному каналі.

До теперішнього часу відомо, що совоподібною структурою володіє телетрафік у проводових мережах при роботі широко поширених протоколів Ethernet, голос поверх IP VoIP (Voice over IP), керування передачею TCP (Transmission Control Protocol), загальної канальної сигналізації №7 (ЗКС№7) і ін. Також виявлені аналогічні ефекти в стільникових телефонних мережах з комутацією пакетів.

Використання традиційних моделей при аналізі мереж із совоподібним характером трафіка призводить до невірних висновків про продуктивність, переоцінку якості обслуговування й неправильному визначенню необхідних мережевих ресурсів при проектуванні, розгортанні й керуванні мережами. Тому актуальною і перспективною є розробка методів і способів дослідження совоподібного трафіка, розробка й використання нових підходів до адекватного моделювання трафіка сучасних пакетних телекомунікаційних мереж, особливо у світлі стрімкого розвитку й поширення високошвидкісних мереж, у тому числі безпроводових, що пред'являють дуже жорсткі вимоги до забезпечення високої якості обслуговування.

Посібник призначено для *поглибленого* вивчення тематики самоподібного трафіку в інфокомунікаційних системах та методів його наочного моделювання для здобувачів ступеня магістра за спеціальністю 172 «Телекомунікації та радіотехніка», буде також корисним для аспірантів, наукових та інженерно-технічних працівників за напрямом інформаційно-телекомунікаційних систем та технологій.

1. ТРАДИЦІЙНИЙ ПІДХІД ДО МОДЕЛЮВАННЯ ТРАФІКУ

1.1. Пуасонівська модель трафіку та успіх її застосування до телефонних мереж загального користування

Теорія телекомунікаційного трафіку бере свій початок з розробок перших телефонних мереж з комутацією каналів (комутованих телефонних мереж загального користування). Робота Ротрі (M. C. Rotry) з AT&T моделювала телефонні дзвінки як такі, що мають однакову тривалість і обчислювала імовірності блокування викликів використовуючи біноміальне розкладання імовірностей виникнення або не виникнення окремих викликів. Моліна (E. C. Molina), також з AT&T, розширив цю модель, коли він вивів вираз для процесу надходження викликів, що (як незабаром з'ясувалося) вже був описаний Симеоном Пуасоном, чиє ім'я він і носить. Ця модель, остаточно сформована роботою Ерланга (A. K. Erlang), що ввів експоненціальну тривалість зайнятості, стала основою теорії телефонного трафіку [1]. Пуасонівський процес – це просто випадковий процес надходження заявок із взаємно незалежними експоненціально розподіленими тривалостями інтервалів між моментами надходження заявок і тривалостями телефонних розмов. Експоненціальний розподіл має вигляд [2]

$$F(x) = 1 - e^{-\lambda x}, \tag{1.1}$$

а відповідна щільність розподілу

$$f(x) = \lambda \cdot e^{-\lambda x}, \tag{1.2}$$

де $\lambda = M[x]^{-1}$ – середня інтенсивність процесу.

Відповідно, процес $X(t)$, що визначає кількість викликів, що надійшли за інтервал часу t, задовольняє умові

$$P\left[X(t)=n\right]=\frac{e^{-\lambda t}\cdot(\lambda\cdot t)^{n}}{n!} \qquad (1.3)$$

Унікальною характеристикою процесу є те, що його дисперсія дорівнює середньому. Характерною рисою Пуасонівського процесу є його випадковість – повна відсутність післядії або пам'яті.

До появи перших пакетних комп'ютерних мереж, Ерлангівська модель Пуасонівського розподілу надходження викликів і експоненціально розподілених тривалостей обслуговування більше 50 років управляла телетрафіком. Формули Ерланга разом з іншими математично розв'язними і витончено отриманими результатами з теорії черг досить довго застосовувалися при розробці телефонних мереж. Більше того, вони сприймалися як природні, універсальні закони.

Традиційна теорія телетрафіку, застосована до телефонних мереж загального користування (ТМЗК), була одним з найбільш успішних застосувань математичних методів в індустрії. Серед головних причин її приголомшливого успіху в традиційній телефонії були дуже статичний характер ТМЗК і обмежена мінливість, властива гомогенним системам, в яких можна говорити про «типових» користувачів і «характерну» поведінку, завдяки чому середні величини адекватно описують процеси. Ще одна важлива причина успіху теорії полягає в її особливій привабливості для телекомунікаційного співтовариства, у першу чергу завдяки фізичній інтерпретації і простоті моделей, що вимагають усього декількох параметрів, що легко визначаються на практиці.

1.2. Неадекватність пуасонівської моделі реальному трафіку пакетних мереж.

Перші зміни в «статичному» світі телефонії відбулися в 1980-х роках з появою факсів і тривали надалі, стаючи усе більш радикальними зі збільшенням популярності Web. Ключовою зміною стало те, що телефонні виклики, що використовувались для факсимільної передачі та доступу до Інтернет мали статистичні характеристики, що сильно відрізняються від типових розмовних. Вони ставали значно більшими і мінливими за своєю тривалістю, і їхня кількість постійно і вражаюче зростала. Обидва типи викликів руйнували існуючу інженерну інфраструктуру ТМЗК, розраховану лише на розмовні виклики. Іноді блокування викликів збільшувалося до неприпустимих рівнів, особливо у вечірній час, і були потрібні спеціальні інженерні методи для запобігання насичення ТМЗК викликами Інтернет-доступу. Ясно, що теорія вже не відповідала дійсності.

З огляду на величезну успішність і ефективність застосування Пуасонівської моделі в телефонії, її елегантність і простоту, і «віру» в універсальність Пуасонівської природи трафіку та в експоненціальні розподіли, а також у відсутність часових залежностей, що з'явилася за 50 років, не дивно, що на початку, і досить довго, незважаючи на архітектурні та прикладні зміни, зовсім не замислювалися про переоцінку застосовності Пуасонівських моделей до трафіку нових мереж, мереж передачі даних, що з'явилися за останні десятиліття [3]. Надходження викликів просто замінили надходженням пакетів, а тривалості викликів часом обслуговування (проходження). Це і дотепер є переважною моделлю пакетного трафіку в посібниках з мереж й університетських курсах [4].

Однак, зараз зрозуміло, що старі правила не підходять до нових мереж. Як причину цього Уолтер Віллінгер (Walter Willinger) і Верн Паксон (Vern

Paxson) у роботі [3] визначили чотири істотних розбіжності між мережами передачі даних і телефонними мережами:

- Мережі передачі даних використовують пакетну комутацію замість комутації каналів.
- Трафік даних є набагато більш мінливим – тривалості індивідуальних з'єднань коливаються від дуже коротких до дуже довгих, і необхідні смуги (швидкості) коливаються від дуже вузьких до дуже широких.
- Буферизація пакетів, що очікують на передачу по каналу при його завантаженості, і можлива втрата пакетів при тривалому перевантаженні через переповнення буфера.
- Для подолання перевантажень протоколи (наприклад, *TCP*) використовують механізми наскрізного контролю перевантажень, що зменшують швидкості передачі при виявленні перевантажень. Ці механізми вносять складні кореляції.

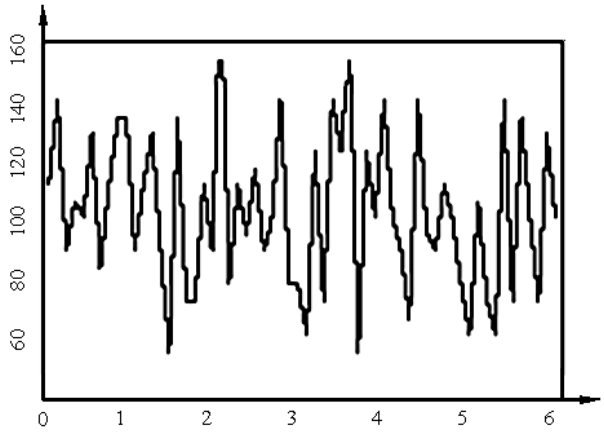
Шкідливою спадщиною впливу телефонії на дослідження мереж передачі даних була практично повна відсутність в 1970-х й 1980-х спроб підтвердити застосовність найважливіших припущень моделювання, незважаючи на реальні вимірювання трафіку мереж передачі даних.

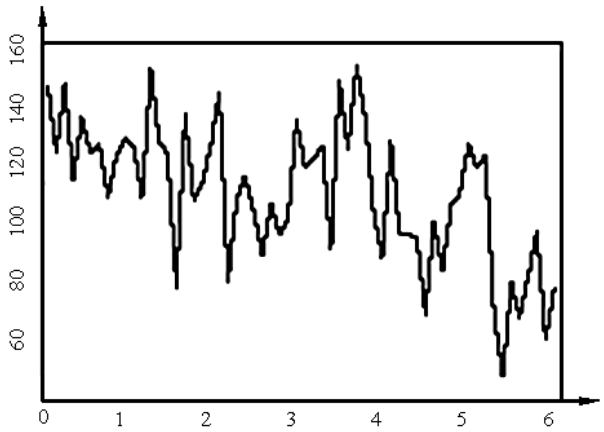
Ці вимірювання показують, що трафік даних є надзвичайно мінливим, або вибухоподібним (*bursty*), і має масштабно-інваріантні властивості. Інакше кажучи, вибухоподібність - це відсутність гладкості, плавності.

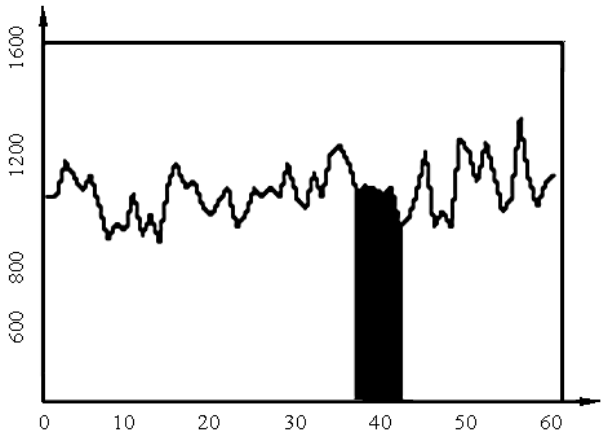
У телефонії часовий масштаб сплесків пов'язаний з інтенсивністю λ Пуасонівського процесу. Наприклад, якщо $\lambda = 100$ 1/с, то часовий масштаб вибухоподібності (у якому відбуваються значні флуктуації) близько 10 мс. Періоди тривалих (більше середнього) активностей і тривалого (менш середнього) затишшя при значному збільшенні або зменшенні часового масштабу відбуваються з імовірністю, що наближається до нуля. Таким чином, Пуасонівский трафік мінливий лише в одному визначеному масштабі, або в дуже обмеженому діапазоні часових масштабів.

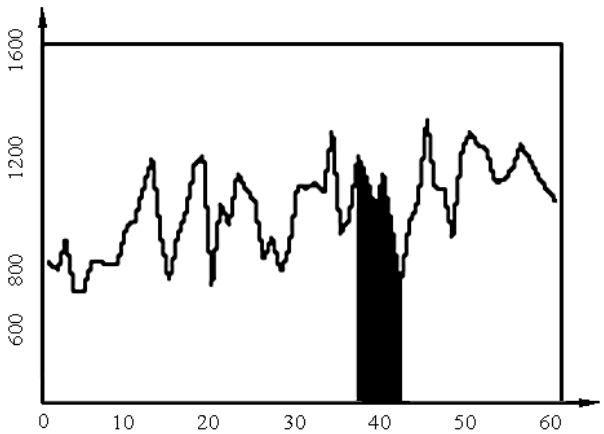
Однак численні дослідження показують, що сплески пакетного трафіку відбуваються в багатьох різних часових масштабах, і така багатомасштабна вибухоподібність зовсім не вписується в світ традиційного Пуасонівського моделювання трафіку. Властивість масштабної інваріантності можна розглядати як відсутність якого-небудь характерного часового або амплітудного масштабу, що описує флуктуації трафіку (сплески), або, аналогічно, відсутність характерної довжини сплесків; пакетний трафік має відмінну рису – структуру «сплеск всередині сплеску» [5].

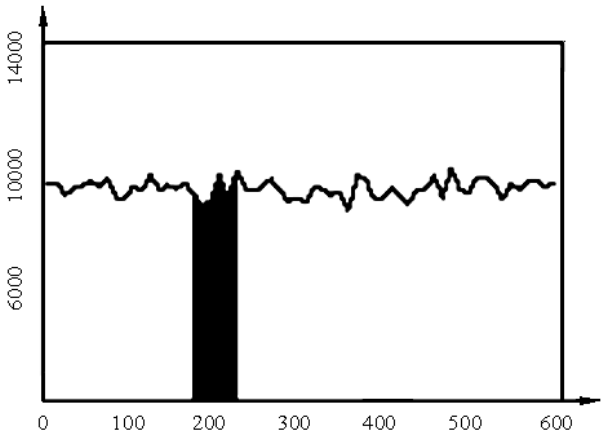
На рис. 1.1 наочно демонструються ці властивості реального трафіку даних і нездатність Пуасонівської моделі їх описати. Права колонка графіків побудована на підставі реальних вимірювань інтенсивності Інтернет-трафіку протягом однієї години. На підставі цих даних за допомогою пуасонівської моделі була сгенерована штучна вибірка інтенсивностей надходження пакетів, що має такі самі середнє та дисперсію (ліва колонка). На графіках у верхньому ряді показані обрані випадковим чином підінтервали кожної вибірки довжиною 6 с у часовому масштабі 100 мс, тобто кожна точка графіку відображає кількість пакетів, що надійшли протягом інтервалу 100 мс. Наступний ряд показує зміну інтенсивності трафіку в десятикратно збільшеному масштабі (тепер кожна точка відображає кількість пакетів за 1 с). Масштаб по осі інтенсивності збільшується пропорційно зміні масштабу по осі часу. Таким чином, точка кожного наступного графіку отримується шляхом усереднення 10 відповідних точок попереднього. Підінтервал, обраний для побудови верхнього графіка, позначений на вихідному графіку чорним кольором. У третьому ряді масштаб знову збільшений в 10 разів, і, нарешті, графіки в нижньому ряді отримані шляхом збільшення масштабу на цей раз у шість разів і охоплюють весь інтервал спостереження довжиною одна година.

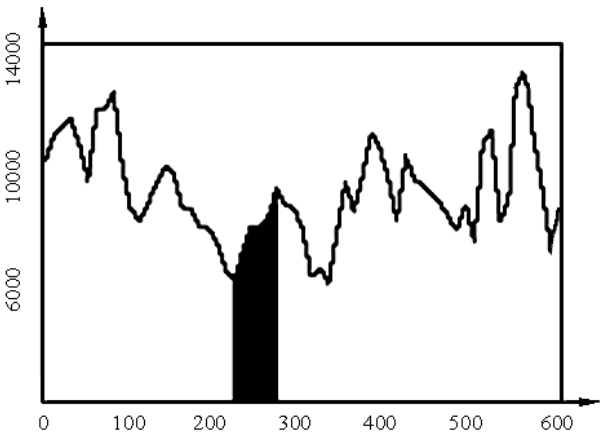

160
140
120
100
80
60
0
1
2
3
4
5
6


160
140
120
100
80
60
0
1
2
3
4
5
6


1600
1200
800
600
0
10
20
30
40
50
60


1600
1200
800
600
0
10
20
30
40
50
60


14000
10000
6000
0
100
200
300
400
500
600


14000
10000
6000
0
100
200
300
400
500
600

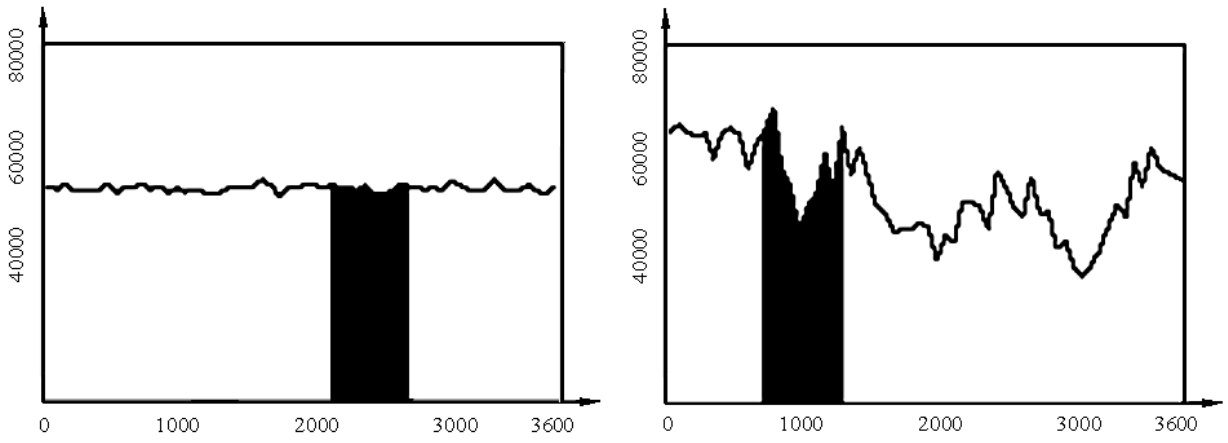


Рисунок 1.1 Порівняння пуасонівського і реально виміряного трафіку

Розбіжності між реальним трафіком і змодельованим Пуасонівською моделлю очевидні. У той час як пуасонівський трафік згладжується при збільшенні масштабу, і стає майже рівномірним, реальний трафік поводить себе по іншому. При будь-яких, навіть досить великих часових масштабах він виглядає майже однаково, проявляючи значну нерівномірність. Це розходження є принципово важливим: трафік у лівій колонці не створює ніяких проблем при проектуванні та керуванні мережами. Вище певного масштабу часу інтенсивність трафіку зводиться до відомої середньої швидкості прибуття пакетів. Немає необхідності в буферах більших розмірів, не виникає складності з вибором пропускної здатності магістральних ліній, забезпечення необхідної якості обслуговування не становить ніяких проблем. З виміряним трафіком все абсолютно інакше, і це розходження буде докладно розглянуто нижче.

Таким чином (що підтвердили численні подальші дослідження різних пакетних мереж, у тому числі *ATM* (*Asynchronous Transfer Mode* - асинхронний режим передачі), ЗКС№7, *Ethernet, ISDN* (*Integrated Services Digital Network* - цифрова мережа з інтеграцією послуг) і інші), реальні пакетні послідовності не підпорядковуються Пуасонівським розподілам і не можуть бути адекватно змодельовані за допомогою традиційних моделей.

Згодом були запропоновані кілька узагальнень Пуасонівської моделі для поліпшення її точності, включаючи суми експоненціальних розподілів і Марковсько-модульований Пуасонівський процес. Більшість із них мали обмежений успіх, у першу чергу, тому що ґрунтувалися все ще на експоненціальних розподілах.

Питання для самоперевірки

1. Що представляє собою Пуасонівська модель трафіку та успіх її застосування до телефонних мереж загального користування.

2) В чому полягає неадекватність пуасонівської моделі реальному трафіку пакетних мереж.

2. ФРАКТАЛЬНА МОДЕЛЬ ТРАФІКУ

2.1. Загальні положення

Необхідно було відмовитися від традиційного розрахунку телетрафіку та шукати нові ідеї, підходи. І так само як трафік даних виявився радикально відмінним від телефонного, так само відрізнялися й основні математичні ідеї та моделі. Математична основа старого підходу – обмежена мінливість як у часі – процеси трафіку або незалежні, або їх кореляції спадають експоненціально швидко, так і за амплітудою – розподіли величин, пов'язаних із трафіком, мають експоненціально спадні хвости. Однак трафіку даних властива висока і вкрай висока мінливість. Висока часова мінливість (берстність) виникає із залежностей, що зберігаються протягом великих проміжків часу, і описується повільно спадною залежністю (ПСЗ), тобто автокореляціями, що підпорядковуються степеневому закону спадання. Походження ПСЗ зараз вважається досить зрозумілим і пояснюється важкими хвостами розподілів розмірів файлів, що передаються по мережі та періодів активності користувачів.

З іншого боку, граничні форми іншого типу мінливості – величини флуктуацій за величиною або амплітудою, і тому стосуються малих часових масштабів, описуються розподілами з важкими хвостами з нескінченною дисперсією.

Ці два типи мінливості цілком різні, і необхідно пам'ятати, що статистичні характеристики трафіку є функціями як часової, так і амплітудної берстності.

Варто звернути увагу, що хоча на великих масштабах часу (секунди і більше) були знайдені прості, і відносно зрозумілі закони, цього не можна сказати про малі масштаби.

На рис. 2.1 у вигляді діаграми показана розроблена класифікація згаданих процесів, розглянутих далі докладно.

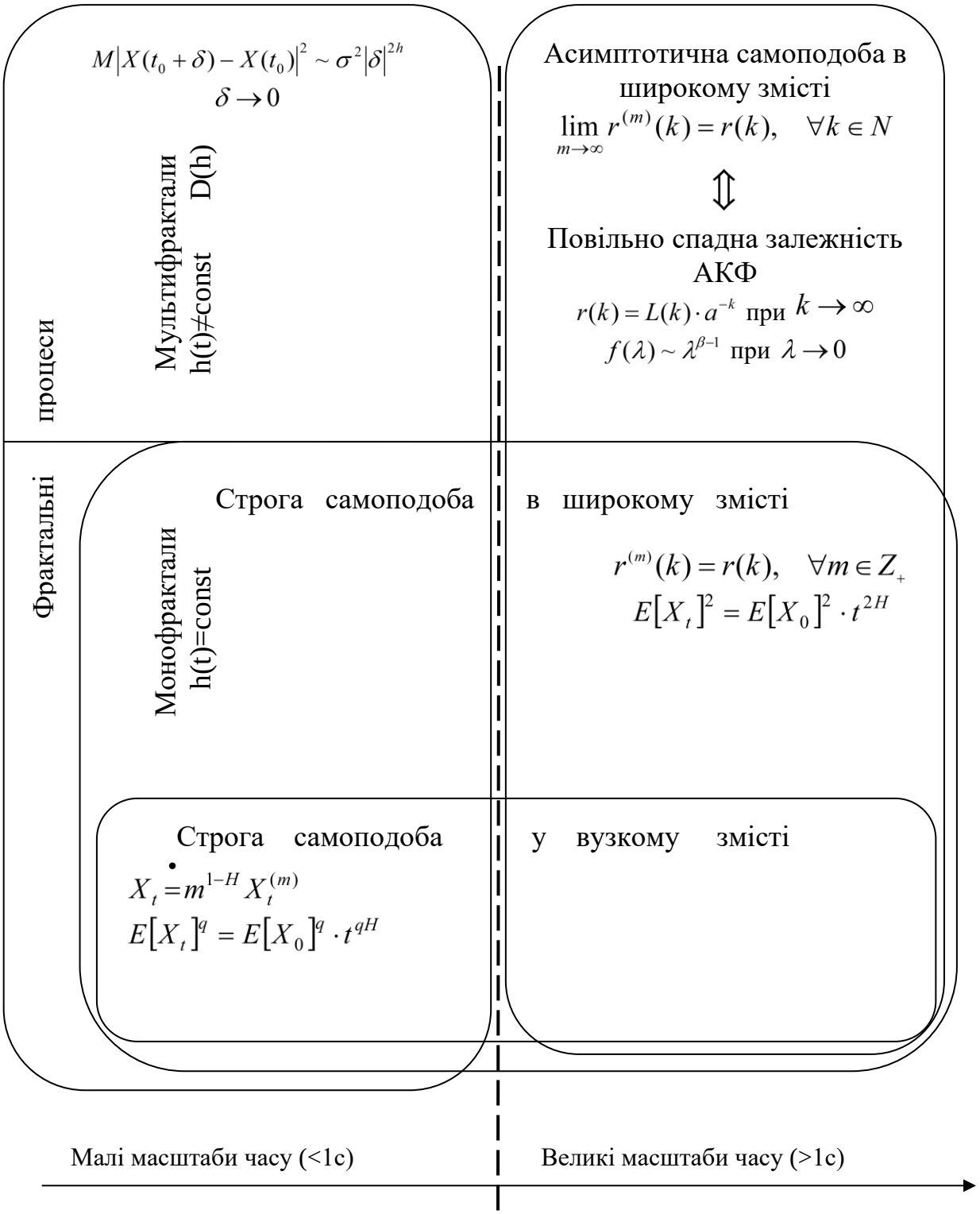


Рисунок 2.1. Класифікація фрактальних процесів

2.2. Розподіли з важкими хвостами

Розподіли з важкими хвостами дуже тісно пов'язані із самоподібними процесами. Як уже було зазначено, вони розглядаються як причини виникнення масштабно інваріантних властивостей трафіку пакетних мереж.

Випадкова величина X має розподіл з важким хвостом (РВХ), якщо

$$P[X > x] \approx L(x) \cdot x^{-\alpha} \quad \text{при } x \to \infty, \qquad (.)$$

де $0 < \alpha < 2$ – коефіцієнт ваги хвоста або параметр форми, L – повільно мінлива на нескінченності функція.

РВХ мають хвости, що спадають асимптотично за степеневим (гіперболічним) законом, на відміну від розподілів з легкими хвостами, хвости яких спадають експоненціально. Основна властивість випадкової величини, розподіленої відповідно до РВХ – її дуже висока мінливість. При $0 < \alpha < 2$ РВХ має нескінченну дисперсію, а при $0 < \alpha \le 1$ ще й нескінченне середнє. На рис. 2.2. представлені графіки щільностей розподілу величин, що мають РВХ, для цих трьох випадків, і для порівняння – експоненціальної щільності ймовірностей $(\lambda = 1)$. При розгляді фрактального трафіку (мереж) найцікавішим є випадок $1 < \alpha < 2$. У цьому випадку вибірка з РВХ являє собою здебільшого відносно невеликі значення, однак також містить і достатню кількість дуже великих значень.

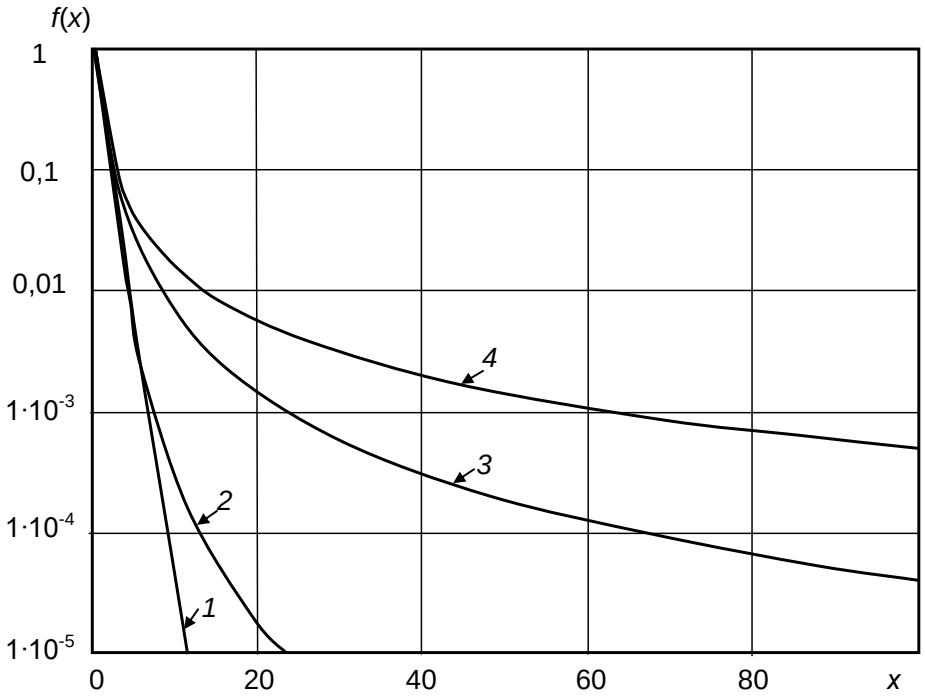


Рис. 2.2. Щільності імовірностей РВХ:

1 - експоненціальний розподіл; 2 - α = 3; 3 - α = 1,25; 4 - α = 0,5

Висока мінливість (і наявність відносно великих значень) тривалостей активності користувачів, або розмірів файлів, що завантажуються користувачами викликають тривалі та досить сильні часові залежності в потоках пакетів, що при об'єднанні цих потоків в агрегатний потік проявляються в його масштабно-інваріантному характері (відсутня характерна тривалість сплесків) і повільно спадної функції автокореляції, тобто до самоподоби трафіку. Це стане очевидним з подальшого викладу при розгляді цих понять. При цьому буде представлена строга залежність між РВХ і викликаною нею самоподобою.

Крім цього, як буде показано нижче, існує і формальний зворотний зв'язок. Самоподібний трафік призводить до РВХ довжин черг у буферах та інших величин, пов'язаних з якістю обслуговування й продуктивністю мереж. Таким чином, можна сказати, що РВХ є не просто елегантною математичною моделлю, але й чітко відбиває реалії пакетного мережевого трафіку. Розглянемо найпоширеніші моделі РВХ.

2.2.1. Розподіл Парето

Найпростішим і найбільш часто використовуваним РВХ є розподіл Парето із щільністю розподілу

$$f(x) = \alpha \frac{k^{\alpha}}{x^{\alpha+1}} \tag{2.1}$$

і функцією розподілу

$$F(x) = 1 - \left(\frac{k}{x}\right)^{\alpha} \tag{2.2}$$

де k – параметр, що визначає мінімальне значення, що може набувати випадкова змінна, $x \geq k > 0$.

Середнє і дисперсія величини, яка має розподіл Парето, описуються наступними виразами:

$$M[X] = \frac{\alpha}{\alpha - 1} k \quad (\alpha > 1) \tag{2.3}$$

$$\sigma_X^2 = \frac{ak^2}{(\alpha - 1)^2 (\alpha - 2)} \tag{2.4}$$

2.2.2. Розподіл Вейбулла

Розподіл Вейбулла також досить часто застосовується при моделюванні самоподібних процесів. Його недоліком є наявність двох параметрів, що вносить додаткові складності при оцінці їх значень. Цим обумовлюється його менш часте використання. Функція щільності розподілу імовірностей розподілу Вейбулла має вигляд

$$f(x) = \alpha\beta\, x^{\beta-1} e^{-\alpha x^{\beta}} \tag{2.5}$$

Функція розподілу

$$F(x) = 1 - e^{-\alpha x^{\beta}} \tag{2.6}$$

Математичне очікування величини, що має розподіл Вейбулла, має вигляд:

$$M[X] = \alpha^{-\frac{1}{\beta}} \Gamma\left(1 + \frac{1}{\beta}\right)$$

Дисперсія:

$$\sigma_X^2 = \alpha^{-\frac{2}{\beta}} \left(\Gamma\left(1 + \frac{2}{\beta}\right) - \Gamma^2\left(1 + \frac{1}{\beta}\right) \right)$$

де $\Gamma(x)$ – гамма-функція Ейлера.

2.2.3. Логарифмічно-нормальний розподіл

Визначення логарифмічно-нормального розподілу ґрунтується на нормальному розподілі [2]: випадкова величина X є логарифмічно-нормально розподіленою, якщо випадкова величина $Z = \ln(X)$ має нормальний розподіл. Функція щільності розподілу імовірностей величини X має вигляд

$$f(x) = \frac{1}{x\sqrt{2\pi}\sigma} e^{-\frac{(\ln(x)-\zeta)^2}{2\sigma^2}} \qquad (x > 0) \tag{2.7}$$

а функція розподілу

$$F(x) = \frac{1}{\sqrt{2\pi}\sigma} \int_0^{\ln x} e^{-\frac{t-\zeta}{2\sigma^2}} dt \qquad (x > 0)$$

де ζ – середнє, а σ – середньоквадратичне відхилення *Z*. Вибірку логарифмічно-нормальної *X* можна одержати, сгенерувавши вибірку гаусової величини *Z*, і трансформувавши її відповідно до співвідношення $X_i = e^{Z_i}$

Середнє та середньоквадратичне відхилення логарифмічно-нормального розподілу дорівнюють відповідно $E[X] = e^{\zeta + \frac{\sigma^2}{2}}$ й $\sigma_X = e^{\zeta}\sqrt{e^{\sigma^2}(e^{\sigma^2} - 1)}$

Логарифмічно-нормальний розподіл був одним з перших не-експоненціальних розподілів, що використовувалися при моделюванні мережевого трафіку. Він пропонувався для моделювання тривалостей телефонних розмов, міжпакетних інтервалів у локальних мережах, розмірів (у пакетах) з'єднань протоколами мережевого теледоступу *Telnet* і передачі файлів *FTP* (*File Transfer Protocol*) [2].

2.3. Самоподоба

Розрізняють детерміновану та стохастичну самоподобу. У випадку детермінованої самоподоби, математичний об'єкт є самоподібним, якщо він може бути розділений на зменшені копії себе самого, тобто структура цілого збігається зі структурою складових частин. Класичний приклад природного об'єкта, що має ознаки детермінованої самоподоби – лист папороті (рис. 2.3).

Рисунок 2.3. Лист папороті – приклад детермінованого самоподібного об'єкта (джерело ілюстрації [6])

Однак трафік даних розглядається як стохастичний процес, і тому щодо нього ми можемо говорити лише про стохастичну самоподобу. Стохастична само подоба означає, що статистичні характеристики процесу залишаються незмінними при розгляді процесу в різних часових масштабах (інваріантні до часового масштабу). Тобто якісно процес виглядає однаково при будь-яких (і досить великих) масштабах часової осі (рис. 2.4)

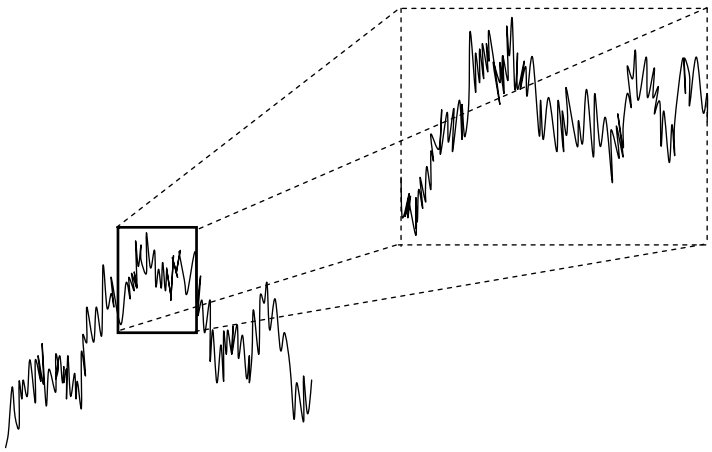


Рис. 2.4. Приклад стохастичної самоподоби

Дамо математичне визначення самоподібним процесам. Слід зазначити, що теорія самоподібного трафіку проходить досить ранню стадію розвитку, і тому існують значні відмінності в термінології та визначеннях.

Нехай $X = (X_t : t = 0, 1, 2, \ldots)$ – стаціонарний у широкому змісті стохастичний процес дискретного аргументу. Тобто, середнє $\mu = E[X_t]$ і дисперсія $\sigma^2 = E\left[(X_t - \mu)^2\right]$ процесу не залежать від часу, а автокореляційна функція $r(k) = \dfrac{E[(X_{t+k} - \mu)(X_t - \mu)]}{\sigma^2}$ залежить лише від «часової затримки» k

.

Позначимо як $X^{(m)} = (X_t^{(m)} : t = 0, 1, 2, \ldots)$ процес, отриманий з X шляхом усереднення за непересічними сусідніми блоками довжини $m, m \in \mathrm{N}$. Його елементи можуть бути виражені так:

$$X_t^{(m)} = \frac{1}{m} \sum_{i=tm}^{tm+m-1} X_i \qquad (2.8)$$

Надалі будемо називати такий ряд агрегованим. Відзначимо, що при $\forall m$ агрегований процес $X^{(m)}$ також буде стаціонарним у широкому змісті. Позначимо як $r^{(m)}(k)$ автокореляційну функцію процесу $X^{(m)}$.

Процес X називається *строго самоподібним у вузькому змісті* (ССВЗ) (*strictly self-similar*) з параметром Херста (Hurst) $H \in (0;1]$ якщо виконується вираз

$$X_t \overset{\bullet}{=} m^{1-H} X_t^{(m)} \qquad (2.9)$$

який розглядається в змісті рівності всіх скінченновимірних розподілів.

Статистичні характеристики всіх порядків самоподібних у вузькому змісті процесів підпорядковуються наступному співвідношенню

$$M[X_t]^q = M[X_0]^q \cdot t^{qH} \qquad (2.10)$$

де q – порядок статистичної характеристики.

У той же час на практиці часто розглядаються лише статистичні характеристики другого порядку (як ті, що найбільш впливають на значимі для нас параметри трафіку). Тому виникло поняття самоподоби в широкому змісті (другого порядку), що припускає масштабну інваріантість процесу як мінімум відносно статистик другого порядку.

Процес X називається *строго самоподібним у широкому змісті* (ССШЗ) (*exactly second-order self-similar*) з параметром Херста $H \in (0;1]$, якщо

$$r^{(m)}(k) = r(k), \quad \forall k \in N, \forall m \in Z_+ \tag{2.11}$$

тобто функція автокореляції процесу не змінюється після агрегації по блоках довжини m. Відповідно для ССШЗ вимога (2.10) обмежується другим порядком:

$$M[X_t]^2 = M[X_0]^2 \cdot t^{2H} \tag{2.12}$$

Тепер стала зрозумілою назва ССВЗ – воно є частковим, більш вузьким випадком ССШЗ. Асимптотично самоподібні процеси – ще більш широкий клас.

Процес X називається *асимптотично самоподібним у широкому змісті* (АСШЗ) (*asymptotically second-order self-similar*) з параметром Херста $H \in (0;1]$, якщо

$$\lim_{m \to \infty} r^{(m)}(k) = r(k), \quad \forall k \in N \tag{2.13}$$

Асимптотично самоподібні процеси проявляють властивості самоподоби лише у великих масштабах часу.

Параметр Херста (що часто називають параметром самоподоби) показує ступінь самоподоби, інтенсивність прояву процесом масштабно-інваріантних властивостей. Значення H = 1/2 вказує на їх відсутність. По мірі наближення H до одиниці вони проявляються все сильніше. При H < 1/2 говорять про антисамоподібну поведінку процесу. Вимірювання реального трафіку

телекомунікаційних мереж показали, що показник самоподоби для нього знаходиться у в межах $H \in (0{,}5;1)$, тому саме цей випадок є для нас цікавим. Розглянемо поняття повільно спадної залежності, тісно пов'язаної із самоподобою при $H \in (0{,}5;1)$.

2.3.1. Повільно спадна залежність

Визначення повільно спадної залежності (*long-range dependence*) ґрунтується на поведінці функції автокореляції процесу при зростанні часової затримки. Для багатьох процесів з її збільшенням автокореляція швидко зменшується. Загалом, процес зі швидко спадною залежністю задовольняє умові, що полягає в тому, що його автокореляція спадає щонайменше так само швидко, як експонента:

$$r(k) = L(k) \cdot a^{-k} \text{ при } k \to \infty, \qquad (2.14)$$

У традиційних моделях трафіку даних, як правило, використовуються тільки короткостроково залежні процеси. Очевидно, що для короткостроково залежного процесу сума $\sum_{k=0}^{\infty} r(k)$ є скінченою: тобто, залежність процесу від свого минулого нетривала і не має значного впливу.

Стаціонарний процес дискретного часу X називається процесом з повільно спадною залежністю (ПСЗ), тривалою пам'яттю, або сильною залежністю, якщо його функція автокореляції

$$r(k) = L(k) \cdot k^{-\beta} \text{ при } k \to \infty, \qquad (2.15)$$

де $\beta \in (0;1)$, L – повільно мінлива на нескінченності функція, тобто, $\lim_{x\to\infty} L(ax)/L(x) = 1$ при всіх $a > 0$.

Таким чином, процеси з ПСЗ характеризуються автокореляційною функцією (АКФ), що спадає гіперболічно (за степеневим законом) при збільшенні часової затримки (лага).

Із цього визначення випливає, що АКФ спадає так повільно, «минуле процесу настільки вагоме», що її сума розходиться, тобто, $\sum_{k=0}^{\infty} r(k) = \infty$ (на відміну від процесів із сумовними кореляціями). Інтуїтивно, якщо присутня повільно спадна залежність, сильно запізнілі кореляції можуть бути індивідуально невеликими, але їхній кумулятивний ефект є значним. На рис. 2.5 представлені графіки автокореляційної функції реального трафіку, виміряного в мережі *LAN FastEthernet* [7]. На нього накладена АКФ строго самоподібного процесу, що апроксимує емпіричну.

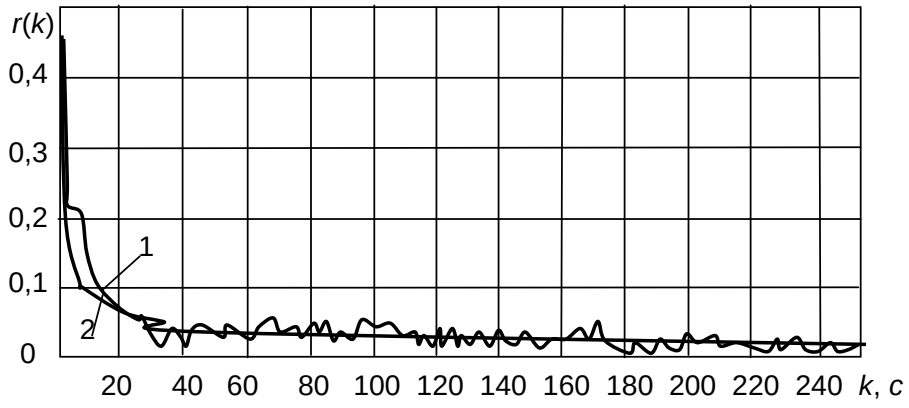


Рис. 2.5. Графік автокореляції реального трафіку

Параметр β у визначенні ПСЗ тісно пов'язаний з параметром Херста:

$$\beta = 2 - 2H \tag{2.16}$$

Таким чином, умова $\beta \in (0;1)$ еквівалентна умові $H \in (0{,}5;1)$, що як раз відповідає випадку, що нас цікавить. Можна сказати, що властивість ПСЗ – це асимптотичний прояв самоподоби у великих масштабах часу. Вона відбиває явище інерційності самоподібних процесів, тобто кластеризації пакетів, і так звану «кластеризацію кластерів». При $H \in (0{,}5;1)$ поняття ПСЗ еквівалентно

АСШЗ (визначеному для великих масштабів часу), а отже, ССШЗ і ССВЗ процеси мають повільно спадну залежність. Тому часто в літературі ці поняття вживаються як синонімічні й взаємозамінні. На рис. 2.6 представлені графіки автокореляційних функцій при різних параметрах Херста.

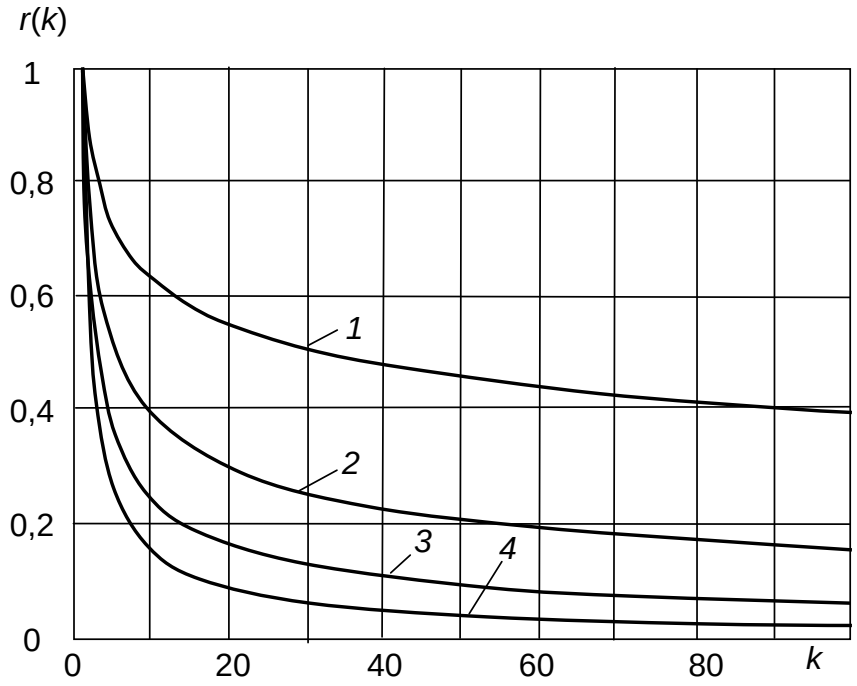


Рис. 2.6. Повільно спадна залежність:

1 – $H = 0{,}9$; 2 – $H = 0{,}8$; 3 – $H = 0{,}7$; 4 – $H = 0{,}6$

Видно, що при збільшенні параметра H (зменшенні β) збільшується залежність процесу від його порівняно давнього минулого. ПСЗ визначає цікаву з точки зору прогнозування властивість - тривалу пам'ять. На інтуїтивному рівні дану властивість можна пояснити в такий спосіб: майбутнє процесу визначається його минулим, причому з все меншим ступенем впливу в міру того, як минуле віддаляється від поточного. Таким чином, процес з тривалою пам'яттю, як би поступово “забуває” своє відносно давнє минуле в міру просування часу в майбутнє.

У частотній області ПСЗ виражається у вигляді характерного степеневого закону поведінки спектральної щільності процесу

$$f(\lambda) = \sum_{k=-\infty}^{\infty} r(k) \cdot e^{ik\lambda} \qquad (2.17)$$

біля початку координат:

$$f(\lambda) \sim \lambda^{\beta-1} \text{ при } \lambda \to 0 \qquad (2.18)$$

Таким чином, з погляду спектрального аналізу процес із ПСЗ (крім того, що має несумовну АКФ) має спектральну щільність із особливістю в нулі (тобто спектральна щільність $f(\lambda)$ такого процесу прямує до нескінченності, в той час як частота λ прямує до нуля, $f(0) = \sum_{k=-\infty}^{\infty} r(k) = \infty$). З іншого боку, ШСЗ процес, навпаки, характеризується спектральною щільністю, скінченною при $\lambda = 0$. Це відбувається при $\beta = 1$, або $H = 1/2$.

2.3.2. Властивості самоподібних процесів (з ПСЗ)

2.3.2.1. Повільно спадна дисперсія

Важливою особливістю самоподібних процесів є те, що дисперсія агрегованого процесу $X^{(m)}$ зменшується повільніше, ніж пропорційно розміру блоку агрегації m, як це було б у випадку процесів з ШСЗ (наприклад, Пуасонівського), для якого

$$D\left[X^{(m)}\right] = D\left[\frac{1}{m}\left(X_1 + X_2 + \ldots + X_m\right)\right] = \frac{1}{m} \cdot m \cdot \left(\frac{1}{m} D[X_1]\right) = \frac{1}{m} D[X_1] \qquad (2.19)$$

для $\forall m$.

Однак завдяки властивості інерційності у випадку самоподібного ПСЗ процесу

$$D\left[X^{(m)}\right] \sim \frac{1}{m^{\beta}} \quad \text{при } m \to \infty,\ \beta = 2(1-H) \tag{2.20}$$

Таким чином, самоподібний процес при досить великих m виглядає менш згладженим, більш нерівномірним, (тобто має більшу мінливість) ніж ШСЗ процес, що, як ми бачили, відповідає властивостям трафіку, виявленим експериментально (див. рис. 1.1).

Якщо прологарифмувати (2.20), одержимо

$$\lg\left(D\left[X^{(m)}\right]\right) \sim \lg\left(D[X]\right) - \beta \lg(m) \tag{2.21}$$

Оскільки $\lg(D[X])$ є константою, що не залежить від m, то графік залежності $D\left[X^{(m)}\right]$ від m у логарифмічному масштабі буде являти собою пряму лінію з нахилом, рівним $-\beta$. Графік легко побудувати за вибіркою X, якщо обчислити агрегований процес при різних рівнях агрегації m, а потім обчислити дисперсію. Після цього вже нескладно визначити значення параметра H. Тангенс кута нахилу від -1 до 0 відповідає наявності самоподоби (ПСЗ). Дослідження показали, що експериментальні результати дійсно лягають на пряму лінію з від'ємним нахилом, меншим (за модулем) 1. На рис. 2.7 показаний приклад такої побудови для даних, виміряних у мережі АТМ [8]. Нахил апроксимативної прямої становить 0,40, що відповідає значенню $\beta = 0{,}4;\ H = 0{,}8$. Для порівняння лінією показаний нахил -1, що відповідає ШСЗ.

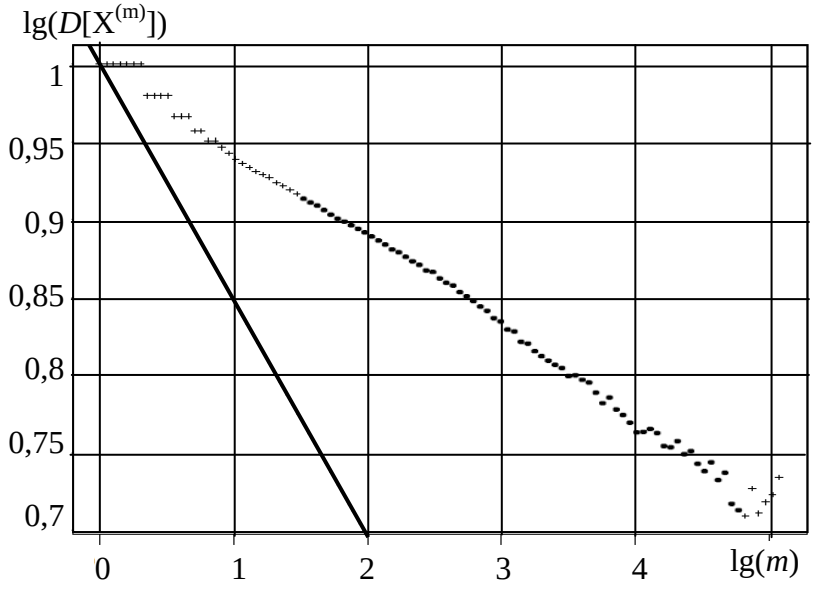


Рис. 2.7. Графік залежності дисперсії від масштабу

2.3.2.2. Ефект Херста

Параметр Херста, як це не дивно, був названий за іменем Х. Е. Херста (*H. E. Hurst*), що присвятив своє життя вивченню Ніла та інших річок, а також проблемам водних резервуарів. Серед інших питань Херст виявив, що рівень води в Нілі за 30-річний період проявляє ознаки самоподоби. Цей розділ знайомить із історією створення параметра *H*, що деяким чином прояснює вплив цього параметра на поведінку процесу.

Гідролог за освітою, Херст досліджував довгострокові характеристики потоків води різних рік. Він помітив, що деякі ріки (наприклад, Рейн) мають відносно м'які флуктуації. Інші ріки, такі як Ніл, проявляють зовсім іншу картину. Херст помітив, що довгострокові та короткострокові характеристики подібні. У короткостроковій перспективі, як і слід було сподіватися, рівень води в Нілі змінювався рік у рік. Також можна було очікувати, що ці флуктуації будуть мати тенденцію до усереднення, так що за більш довгий період рівень Нила повинен був залишатися в досить

вузькому діапазоні з гарними та поганими періодами, що часто змінюють один одного. Однак у дійсності Ніл поводиться не так. У довгостроковій перспективі довгі періоди посухи слідують за довгими періодами високої води, під час яких повені відбувалися ледве не щороку. Мандельброт (розробник теорії фракталів) назвав це явище *ефектом Йосифа (Joseph*), відповідно до уривка з Біблії (Книга Буття 41:25, 29-30): «І сказав Йосип до фараона — ось приходять сім літ – великий достаток у всім краї єгипетськім. А по ним настануть сім літ голодних». Можливо, це не кращий термін, оскільки довгострокове поводження Ніла не проявляє такої регулярності. У теперішній дійсності не спостерігається якої-небудь стійкої циклічності в будь-якому часовому масштабі. Однак статистична схожість довгострокової мінливості на короткострокову існує.

Для оцінки цієї схожості Херст розробив нормалізовану безрозмірну характеристику *R*/*S*. Докладно вона буде розглядатися в розділі (2.3.3.1). Відзначимо лише, що як *R*, так й *S* характеризують мінливість даних. Величина *R* лінійно залежить від даних, тоді як *S* ураховує квадрати вихідних значень. Херст називав це відношення *діапазоном масштабування* (*rescaled range*), і емпірично виявив, що для багатьох часових рядів, що спостерігаються у природі, відношення *R*/*S* асимптотично підпорядковується такому співвідношенню:

$$M\left[\frac{R(n)}{S(n)}\right] \sim n^H, \quad n \to \infty \tag{2.22}$$

де n – розмір досліджуваної вибірки.

У цьому нескладно переконатися візуально, якщо побудувати графік залежності *R*/*S* від n у логарифмічному масштабі. Всі точки лягають на пряму лінію, нахил якої являє собою параметр *H*.

Для будь-якого процесу з незалежними приростами відношення *R*/*S* асимптотично пропорційно $n^{1/2}$. Однак Херст виявив, що для багатьох природних (гідрологічних, геофізичних, кліматичних) явищ значення *H*

знаходяться у діапазоні від 0,7 до 0,9, що відповідає більшій мінливості у великих масштабах. Для них точки графіка лягали на лінію, що відхиляється (має більший нахил) від лінії з $H = 1/2$, і цей ефект став відомий як ефект Херста. Цей ефект виявляється і у даних отриманих при вимірюванні інтенсивності трафіку сучасних телекомунікаційних мереж.

Варто відмітити, що у випадку $½ < H < 1$ говорять про *персистенте* (підтримоване) поводження процесу або про те, що процес має тривалу пам'ять. Інакше кажучи, якщо протягом деякого часу в минулому спостерігалися додатні прирости процесу, тобто відбувалося збільшення, то і надалі в середньому буде відбуватися збільшення. Інакше кажучи, імовірність того, що процес на $i+1$ кроці відхиляється від середнього в тому ж напрямку, що й на i кроці настільки велика, наскільки параметр H близький до 1. Тобто персистентні стохастичні процеси виявляють чітко виражені тенденції зміни при відносно малому “шумі”.

При $H = 1/2$ відхилення процесу від середнього є дійсно випадковими і не залежать від попередніх значень, що відповідає випадку звичайного Броунівського руху.

У випадку $0 < H < 1/2$ говорять про *антиперсистентний* процес. Тут високі значення процесу слідують за низькими, і навпаки. Інакше кажучи, імовірність того, що на $i + 1$ кроці процес відхиляється від середнього в протилежному напрямку (відносно відхилення на i кроці) настільки велика, наскільки параметр H близький до 0.

Ці властивості проілюстровані на рис. 2.8 у вигляді графіків самоподібних процесів з $H = 0,2; 0,5; 0,8$.

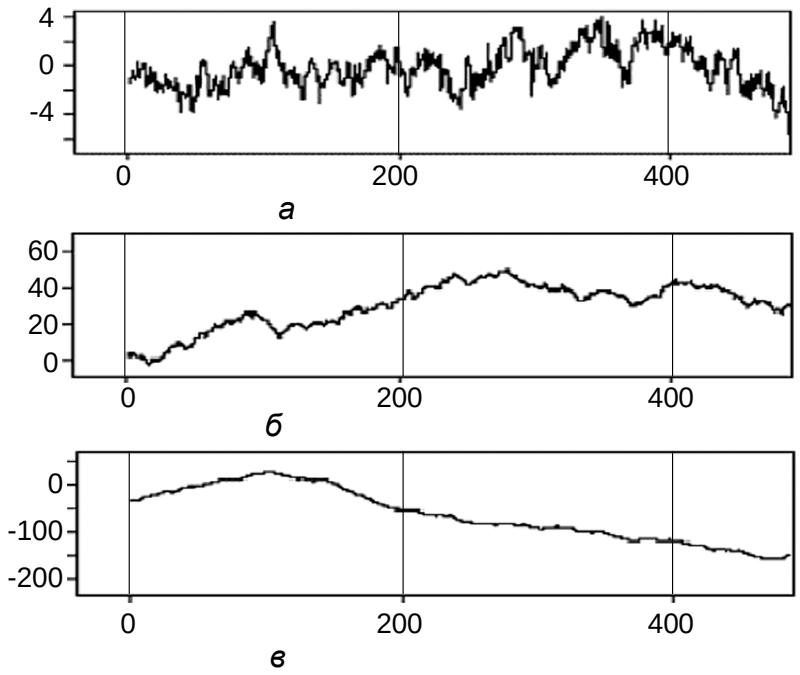


Рис. 2.8. Ефект Херста: *а* – $H = 0{,}2$; *б* – $H = 0{,}5$; *в* – $H = 0{,}8$

2.3.3. Методи оцінки параметра Херста

Метою моделювання самоподібного трафіку є одержання вибірок, що максимально відповідають реальному трафіку мереж. Тому дуже важливим є завдання точного визначення параметра H процесів, як для вимірювання ступеня самоподоби емпіричних пакетних послідовностей, так і для оцінки точності застосовуваних алгоритмів моделювання. Крім розглянутого методу побудови графіка дисперсії від масштабу часу, для визначення значення параметра Херста експериментальних і штучних самоподібних вибірок, використовується величезна кількість інших методів: метод, що ґрунтується на вейвлетах, періодограмах, статистичних моментах та ін. [9]. Розглянемо метод *R*/*S*-статистики та оцінки Уіттла.

2.3. 3.1. Метод *R*/*S*

Цей метод у загальних словах описаний раніше. Тепер конкретизуємо їх. Для стохастичного процесу $X = (X_t : t = 0, 1, 2, \ldots)$ *діапазон масштабування* в інтервалі спостереження N визначається в такий спосіб:

$$R(N)/S(N) = \frac{\max_{1 \le j \le N} \left\{ \sum_{k=1}^{j} (X_k - M(N)) \right\} - \min_{1 \le j \le N} \left\{ \sum_{k=1}^{j} (X_k - M(N)) \right\}}{\sqrt{\frac{1}{N} \sum_{k=1}^{N} (X_k - M(N))^2}} \qquad (2.23)$$

,

де $M(N) = \frac{1}{N} \sum_{j=1}^{N} X_j$ – вибіркове середнє при довжині вибірки N.

У знаменнику цього відношення – середньоквадратичне відхилення за вибіркою, а в чисельнику – діапазон між максимальним і мінімальним відхиленням *кумулятивного процесу* від середнього значення за весь час спостереження. У певному сенсі, як $R(N)$, так і $S(N)$ вимірюють мінливість процесу, однак по різному. Як було зазначено, для самоподібних процесів відношення *R/S* добре описується формулою (2.22) для великих значень N. Якщо прологарифмувати обидві частини (2.22), одержимо:

$$\log(R/S) \sim H \log(N) - H \log 2 \qquad (2.24)$$

При побудові графіка залежності *R/S* від *N* у логарифмічному масштабі точки лягають на пряму лінію. Нахил цієї лінії являє собою параметр *H*. Цей результат підтверджений рядом експериментів.

2.3.3.2. Оцінка Уіттла

Графіки залежності дисперсії від масштабу часу і *R/S* від *N* є евристичними, або візуальними, методами, тобто їх варто застосовувати лише для грубої оцінки параметра Херста, і для визначення того, чи має цей набір

даних самоподібні властивості. Тепер розглянемо оціночну формулу, що дозволяє одержати більш точну статистичну оцінку, але потребує прийняття певних припущень щодо даних, які розглядаються. Спочатку розглянемо оціночну формулу спектральної щільності.

Функцію автокореляції вибірки процесу $X(t)$ можна оцінити в такий спосіб:

$$\hat{R}_N(k) = \frac{1}{N}\sum_{n=1}^{N} X(n+k)X(n) \qquad (2.25)$$

Оскільки спектральна щільність $f(\lambda)$ являє собою перетворення Фур'є функції автокореляції $R(k)$, то можна розраховувати, що в результаті перетворення Фур'є (2.25) вийде відповідна оцінка спектральної щільності. Таким чином, оцінка спектральної щільності вибірки $X(t)$ може бути отримана так:

$$P(\lambda) = \frac{1}{2\pi N}\left(\sum_{j=1}^{N} X_j e^{ij\lambda}\right)^2 \qquad (2.26)$$

Цей вираз зазвичай називається періодограмою, або функцією інтенсивності.

Припустимо, що вибірка взята із самоподібного стохастичного процесу з параметром H. Тоді спектральна щільність може бути виражена як $f(\lambda, H)$, де параметр H невідомий. Відповідно до критерію максимальної правдоподібності Уіттла $\hat{H}$, оцінка параметра H, визначається як таке значення H, що мінімізує $Q(H)$:

$$Q(H) = \int_{-\pi}^{\pi} \frac{P(\lambda)}{f(\lambda, H)} d\lambda \qquad (2.27)$$

Інтеграл можна замінити дискретним сумуванням по частотах $\omega_i = 2\pi/N, 4\pi/N, \ldots, 2\pi$ при довжині вибірки N.

Крім того, метод дозволяє обчислити оцінку дисперсії, що у свою чергу дозволяє обчислити величини довірчих інтервалів:

$$D\left[\hat{H}\right] = 4\pi\left(\int_{-\pi}^{\pi}\left(\frac{\partial \log f(\omega)}{\partial H}\right)^2 d\omega\right)^{-1} \qquad (2.28)$$

На відміну від попередніх двох методів, що використовуються для перевірки часових послідовностей на самоподобу й одержання грубої оцінки параметра *H*, оцінка Уіттла припускає, що часова послідовність є самоподібним процесом певного виду, і дозволяє оцінити *H* у довірчому інтервалі.

2.3.4. Фрактальний Броунівський рух

Фрактальний Броунівський рух (ФБР) є однією з формальних моделей самоподоби, і найбільш часто застосовується для моделювання самоподібних ПСЗ процесів. Власне кажучи, ФБР є узагальненням звичайного Броунівського руху при послабленні вимог до незалежності його приростів. Середнє і автоковаріаційна функція унікально визначають ФБР, оскільки це гаусів процес. Тому немає необхідності розглядати статистичні характеристики вище другого порядку (ССШЗ), і саме завдяки цьому процес є дуже привабливим з погляду моделювання.

Фрактальний Броунівський рух визначається стохастичним поданням [10],

$$B_H(t) = \frac{1}{\Gamma(H+1/2)}\left(\int_{-\infty}^{0}\left((t-s)^{H-1/2} - (-s)^{H-1/2}\right)dB(s) + \int_0^t (t-s)^{H-1/2}\,dB(s)\right) \qquad (2.29)$$

,

де Γ – гамма-функція $\Gamma(\alpha)=\int_0^\infty x^{\alpha-1}e^{-x}dx$

$0 < H < 1$ – параметр Херста. ФБР із параметром Херста H є самоподібним з параметром Херста H.

Інтегратор B – це стохастичний процес, звичайний Броунівський рух. Відзначимо, що B відновлюється з B_H при $H = 1/2$. Це відповідає випадку відсутності ПСЗ і тому дійсно являє собою звичайний Броунівський рух, що характеризується незалежними приростами.

Фрактальний Броунівський рух $B_H = (B_H(t):t = 0, 1, 2, \ldots)$ з $0 < H < 1$ унікально характеризується такими властивостями, (порівняєте, наприклад, з [11]):

- $B_H(t)$ має стаціонарні прирости;
- $B_H(0) = 0$;
- $B_H(t)$ підпорядковується Гаусовому розподілу при $t \geq 0$
- $M\left[B_H(t)\right]=0$ при $t \geq 0$;
- $M\left[B_H^2(t)\right]=t^{2H}$ при $t \geq 0$;

Функцію автокореляції ФБР знайдемо в такий спосіб:

$$M\left[\left(B_H(t)-B_H(s)\right)^2\right]=M\left[B_H^2(t)+B_H^2(s)-2B_H(t)B_H(s)\right]. \qquad (2.30)$$

Звідси одержуємо

$$r(t,s)=M\left[B_H(t)B_H(s)\right]=\frac{1}{2}\left(M\left[B_H^2(t)\right]+M\left[B_H^2(s)\right]-M\left[B_H(t)-B_H(s)\right]\right)= \qquad (2.31)$$
$$=\frac{1}{2}\left(t^{2H}+s^{2H}-(t-s)^{2H}\right)$$

При $0 < s \leq t$.

Далі розглянемо поняття фрактального гаусового шуму (ФГШ), під яким мають на увазі інкрементний процес $X=\left(X_t:t=0,1,2,\ldots\right)$ фрактального Броунівського руху:

$$X_t = B_H(t+1) - B_H(t) \tag{2.32}$$

Ясно, що X_t має стандартний нормальний розподіл при будь-якому k, але немає (у загальному випадку) незалежності. Його функція автокореляції має вигляд

$$r(k) = \frac{1}{2}\left[(t-1)^{2H} - 2t^{2H} + (t+1)^{2H}\right] \tag{2.33}$$

при $k \in N$

Якщо $H = 1/2$, всі автокореляції рівні 0 (крім, звичайно, при $k = 0$), що узгоджується з властивостями звичайного Броунівського руху, що має незалежні прирости. Нескладно показати, що асимптотично виконується

$$r(k) \approx H(2H-1)k^{2H-2} \tag{2.34}$$

при $k \to \infty$.

Це означає повільно спадну залежність при ½ < H < 1. Отже, фрактальний гаусів шум є самоподібним процесом.

2.3.5. Сходження суперпозиції РВХ-процесів до самоподоби

Майже відразу після виявлення самоподібних властивостей пакетного трафіку було зроблене припущення про його причину, пов'язану з РВХ інтервалів активності користувачів мережі. Було зроблено спробу моделювання трафіку шляхом суперпозиції деякої кількості парето-подібних *ON*-/*OFF*- джерел. Кожне таке джерело перебуває в одному із двох станів: *ON*-, під час якого він активно передає пакети, і *OFF*-, протягом якого він не діє. Якщо припустити, що періоди активності є незалежними ідентично розподіленими випадковими величинами і що кожне джерело управляється однаковими розподілами, тоді можна визначити поведінку суперпозиції, або

мультиплексування, таких джерел. Дослідники промоделювали періоди *ON-/OFF-* за допомогою розподілів з нескінченною дисперсією, зокрема, використовуючи розподіл Парето з параметром a від 1 до 2. Як уже згадувалося, коли параметр a знаходиться в цьому діапазоні, випадкова змінна має кінцеве середнє значення і нескінченну дисперсію.

Виявилось, що суперпозиція великої кількості *ON-/OFF-* джерел, що підпорядковуються РВХ, дозволяє одержати самоподібний трафік з параметром Херста

$$H = (3 - \alpha)/2 \qquad (2.35)$$

Відзначимо, що при $1 < \alpha < 2$ ми одержимо $H \in (0{,}5;1)$, що є діапазоном самоподоби (з ПСЗ), що цікавить нас. Докладно метод, що ґрунтується на такому агрегуванні джерел з РВХ періодів активності, розглянутий нижче в розділі (3.3.2)

2.4. Фрактальна природа трафіку реальних телекомунікаційних мереж

2.4.1. «Всесвітня павутина»

В [12] повідомляється про дослідження трафіку *WorldWideWeb* («Всесвітньої павутини» або *WWW*), що включає більше півмільйона запитів до веб-документів. Дані були зібрані на 37 веб-браузерах, що працювали на робочих станціях факультету кібернетики Бостонського університету. Використана методологія була подібною до тієї, що застосовувалася в описаному раніше дослідженні мережі *Ethernet*. Отримані дані свідчать про те, що трафік, сформований веб-браузерами, є самоподібним. Подібні результати були опубліковані в [13].

Дослідники моделювали кожний веб-браузер як джерело *ON*/*OFF* і виявили, що дані дуже добре відповідають розподілу Парето. Для різних наборів вимірювань дослідники визначили відповідні значення параметра α розподілу Парето. Виявилося, що вони знаходяться у діапазоні від 1,16 до 1,5. Автори провели додаткові тести у спробі пояснити причини самоподібної поведінки. Вони розглядали обсяг веб-даних, переданих серверами браузерам, і виявили, що хвіст розподілу відповідав розподілу типу Парето. Автори висунули гіпотезу про те, що веб-трафік відображає випадковий вибір файлів для передачі. Зокрема, якщо користувачі обирали файли, слідуючи за посиланнями і не звертаючи уваги на розмір файлів, що завантажують, то обсяг переданих даних являв собою, по суті, випадкові вибірки з розподілу веб-файлів. Продовжуючи аналіз, дослідники виявили, що роміри файлів, доступні в Інтернеті, мають PBX. Це схоже на правду, тому що хоча безліч файлів в Інтернеті має невеликі розміри, там є також багато великих і дуже великих файлів, наприклад мультимедійних, які із поширенням широкосмугового доступу стають усе більш популярними в *WWW*.

2.4.2 Загальна канальна сигналізація №7

Повідомляється також про дослідження [14], присвячені трафіку керуючих сигналів цифрових телекомунікаційних мереж з використанням протоколу ЗКС№7, характерного для мереж *ISDN* та інших мереж. Дослідження охоплювали близько 170 мільйонів сигнальних повідомлень, зібраних на безлічі різних робочих мереж, керованих протоколом ЗКС№7. Протокол ЗКС№7 є сигнальним протоколом загального каналу, що використовується внутрішніми комутуючими вузлами телекомунікаційної мережі для обміну керуючими повідомленнями. Керуючі функції включають вибір маршрутів з'єднань, встановлення і розрив з'єднань, а також передачу повідомлень про стан трафіку та про помилки.

Дослідження чітко показали, що традиційні моделі, засновані на пуасонівському розподілі, не відповідали поведінці трафіку ЗКС№7. Як і в інших дослідженнях, у цій науковій праці виявилося, що самоподібні моделі трафіку забезпечують кращу відповідність. Більш того, завдяки доступності керуючих сигналів автори також змогли вивчити шаблон трафіку фактичних запитів і, зокрема, дослідити тривалість з'єднань. Як з'ясувалося, тривалість з'єднань найкраще описується PBX.

2.4.3. Використання протоколів TCP, FTP і Telnet

В [15] повідомляється про дослідження широкого спектру *TCP*-трафіку, а також про вивчення *FTP* і *Telnet*-трафіку, переданого по *TCP*-з'єднанням. Були зроблені наступні загальні висновки:

- ♦ Використовувані зазвичай пуасонівські моделі істотно недооцінюють нерівномірність *TCP*-трафіку в широкому діапазоні часових масштабів.
- ♦ Для інтерактивного *Telnet*-трафіку надходження з'єднань добре моделюються пуасонівським розподілом. Однак припущення про пуасонівський розподіл інтервалів часу між надходженнями пакетів істотно недооцінює нерівномірність трафіку.
- ♦ Для групового пересилання даних, що здійснюється протоколом *FTP*, структура трафіку, знову ж, помітно відрізняється від пуасонівського. Як і у випадку Telnet-даних, надходження *FTP-сеансів* добре відповідають пуасонівській моделі, але швидкість надходження даних по *FTP*-з'єднанням має самоподібні властивості. Крім того, розподіли кількості байтів у кожному сплеску мають важкі хвости.

В іншій роботі [16] повідомляється про використання *FTP-пакетів* для вимірювання затримки доставки пакетів в Інтернеті. Результат цього дослідження показує, що затримки доставки пакетів підпорядковуються

ПСЗ.

2.4.4. Цифровий відеотрафік з VBR

Аналіз відеотрафіку з кодуванням зі змінною бітовою швидкістю *VBR* (*Variable Bit Rate*) показав ті ж результати, що й для трафіку *Ethernet* й *WWW* – він також є фрактальним. Беран та ін. [17] проаналізували декілька мільйонів кадрів, що надійшли з різних серверів, і визначили, що ПСЗ є невід'ємною рисою *VBR*. Вони також вказали на розбіжності між вимірюваними даними і штучно сгенерованим *VBR*-трафіком за допомогою методів, описаних у літературі. Див. також [18], [19]

В [90] описується експеримент, що був проведений з двома годинами відео (використовувався фільм «Зоряні війни») Фільм був закодований за допомогою стандарту *JPEG*. Результат аналізу показав, що отриманий потік даних, що складається з кадрів змінної довжини, кожен з яких містив відеокадр, проявляє самоподібний характер, і що довжина кадру підпорядковується розподілу Парето, принаймні, хвосту цього розподілу. Автори показують, як висока мінливість пов'язана із зображенням у кадрі. У відео містяться сцени, у яких зображення змінюється дуже мало, сцени з невеликою кількістю руху, і сцени, у яких зображення змінюється дуже швидко. І, внаслідок природи алгоритму стискання відео, це призводить до РВХ у кодованому відеосигналі.

2.4.5. Інші

Також самоподібні властивості трафіку були виявлені в локальних мережах *Ethernet*, у мережах з асинхронним режимом передачі (*ATM*), про дослідження яких уже повідомлялося вище, а крім того, у різних бездротових мережах, у тому числі в стільникових телефонних, і численних інших.

2.5. Вплив самоподібного трафіку на продуктивність мереж

Масштабно-незалежний (самоподібний) трафік створює істотний негативний вплив на продуктивність мережі (зокрема, на роботу комутаційного устаткування через теорію черг). Він призводить до збільшення довжин черг, імовірностей втрати пакетів і повторних передач, часу затримки передачі даних [9,21,22], що в остаточному підсумку приводить до зменшення рівня використання мережних ресурсів при фіксованій якості обслуговування і тому до збільшення витрат. Як наслідок, вплив самоподоби також поширюється на такі задачі мережевого управління, як контроль за перевантаженнями, визначення мережевих ресурсів, керування доступом і виділення ресурсів, політику ціноутворення на послуги [23].

У найзагальнішому розумінні, самоподібний процес має високу мінливість (оскільки підпорядковується розподілу з важким хвостом) і його не можна назвати згладженим. У результаті, для того, щоб передати такий трафік із заданою якістю обслуговування, пропускна здатність каналу повинна відповідати рівню пікових викидів, а оскільки середній рівень трафіку є досить низьким, пропускна здатність буде витрачатися неефективно. Інакше кажучи, коефіцієнт використання такого каналу буде низьким. Те ж саме стосується і комутаційного устаткування, оскільки, як буде показано нижче, самоподоба трафіку призводить до довжин черг у буферах, що підпорядковуються PBX. Отже, обсяги буферів комутаторів також повинні бути досить великими для забезпечення пікових довжин черг.

Очевидно, що застосування традиційних (пуасонівських, заснованих на експонентних розподілах і незалежних надходженнях пакетів) моделей трафіку призводить до необґрунтовано оптимістичних висновків щодо продуктивності. Значно перебільшуються рівень ефективного використання частот, пропускна здатність, якість обслуговування, що приводить до

неадекватних результатів при розрахунку необхідних мережевих ресурсів (кількості каналів, смуги частот, параметрів устаткування, розмірів буферів і т.п.). Як вказується в [24], саме припущення про пуасонівський характер трафіку та відповідні результати аналізу черг призвели до того, що виробники *ATM*-комутаторів оснащували комутатори першого покоління занадто малими буферами (від 10 до 100 комірок). Коли ці комутатори пройшли польові випробування та піддалися реальному навантаженню, то кількість загублених комірок виявилася значно більше очікуваної.

В [25] вивчений зв'язок між самоподобою трафіку і продуктивністю мереж з протоколом *TCP*/*IP*. Показано, що зі збільшенням ступеня самоподоби (параметра Херста) відбувається різке погіршення продуктивності мережі, що виражається в збільшенні ймовірності втрати пакетів, рівня повторних передач і затримок у чергах. Аналогічні результати отримані в багатьох інших дослідженнях для різних мереж з різними протоколами передачі, архітектурами та конфігураціями, алгоритмами керування і т.п.

У дослідженнях, опублікованих в [26] і [27] ученими з лабораторії *Bellcore*, вивчалися дані, передані по мережах Ethernet і ISDN та складались з більш ніж 100 000 пакетів. Був побудований графік залежності фактичної затримки доставки пакета від коефіцієнта використання мережі, а також очікуваної затримки, розрахованої за допомогою загальноприйнятої теорії черг. На рис. 2.9 зображений такий графік для мережі *ISDN*. Як видно, узгодженість між фактичним і очікуваним часом знаходження у черзі дуже погана. Розрахунки, що ґрунтуються на пуасонівському аналізі черг, показують, що ефективна потужність сервера становить близько 80 %, тоді як у реальній мережі затримка починає різко зростати вже при коефіцієнті використання в діапазоні від 50 до 60 %. Подібні результати були отримані і для другого набору даних для мережі *Ethernet*.

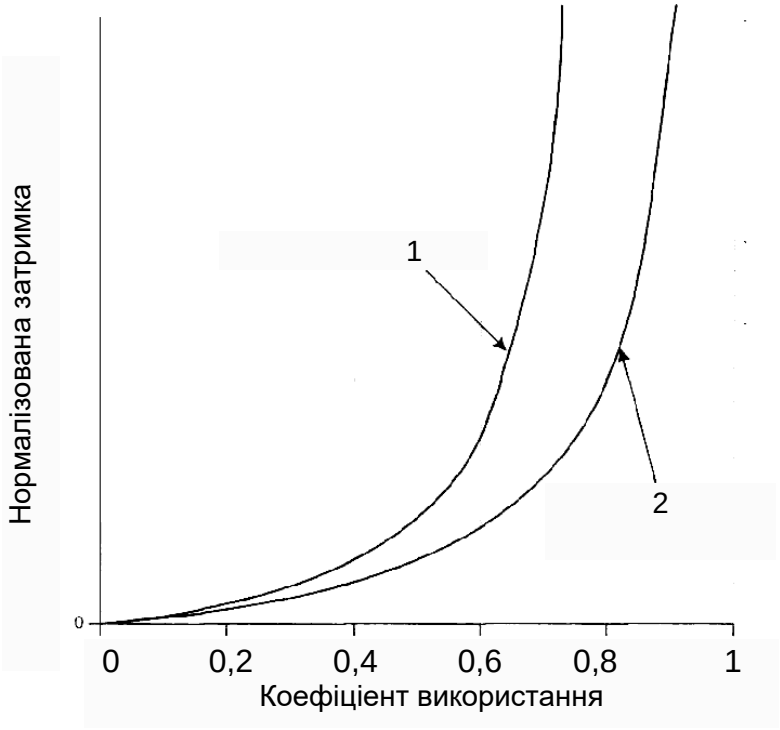


Рисунок 2.9. Середній час очікування: 1 - Реальний трафік; 2 - Модель теорії черг

В [28] проводилося подібне дослідження відповідності реальної продуктивності мереж *Ethernet* результатам, прогнозованим за Пуасонівською моделлю. На рис. 2.10, *а* в логарифмічному масштабі зображені залежності ймовірності втрати пакета (при переповненні буфера) від розмірів буфера при різних коефіцієнтах використання мережі ρ, побудовані на основі експериментальних даних. На рис. 2.10, *б* зображені аналогічні графіки результатів моделювання (з використанням пуасонівської моделі M/M/1). Як видно із графіків, пуасонівська модель недооцінює імовірність втрати пакетів на кілька порядків (при високому коефіцієнті використання).

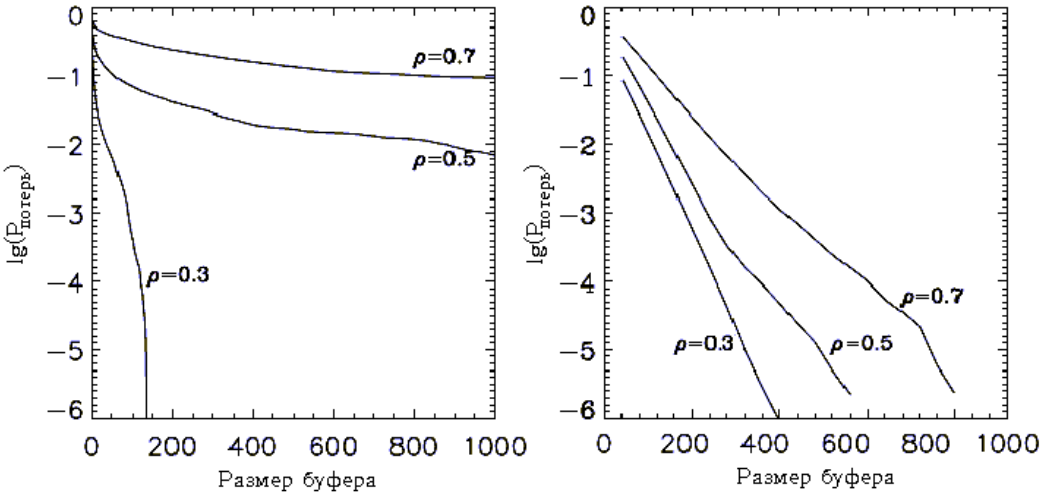


Рисунок 2.10. Втрати пакетів:
а - експериментальні дані; б - пуасонівська модель (1 - ρ = 0,3; 2 - ρ = 0,5; 3 - ρ = 0,7)

Дуже багато публікацій присвячено дослідженню впливу самоподібного трафіку на поведінку черг (і, отже, на роботу та параметри комутаційного устаткування), що є одним з визначальних факторів для якості обслуговування. В [29] продемонстрована очевидна розбіжність між показниками моделей обслуговування із самоподібними вхідними потоками і показниками, прогнозованими традиційними моделями пуасонівської теорії черг. Самоподібний трафік призводить до того, що довжини черг і час обслуговування в буферах підпорядковуються РВХ, як показали дослідження, засновані на експериментальних даних, зібраних у різних мережах (*Ethernet, ISDN, WWW*), опубліковані в [30].

Про аналогічні результати для мереж *ATM* повідомляється в [31]. Показано, що самоподібний трафік, що надходить на вхід *ATM*-комутатора, викликає РВХ тривалості зайнятості буфера. Автори також виявили, що при збільшенні розмірів буфера імовірність втрати комірки зменшується не експоненційно, як показує аналіз на базі Марковських моделей трафіку, а гіперболічно. В [32] продемонстровано, що імовірність переповнення буфера зменшується за степеневим законом при збільшенні його розмірів, набагато повільніше, ніж експоненційно. Це рівною мірою стосується комутаторів,

наприклад, для мереж *ATM*, ретрансляції кадрів в 100*BASE-T*, маршрутизаторів глобальних мереж, локальних мереж із загальним носієм, наприклад, Ethernet, і статистичних мультиплексорів.

Велику кількість робіт присвячено спробам розробити достовірні аналітичні моделі обслуговування потоків із самоподібною поведінкою. Розглянемо коротко статтю [11], що послужила поштовхом до появи інших робіт у цій області. Була розроблена модель навантаження на основі процесу ФБР і нескінченного буфера з експоненціальним часом обслуговування. Математичні викладення даної статті виходять за рамки нашого обговорення, однак основний результат є характерним для дослідження впливу самоподоби на продуктивність. При певних припущеннях залежність необхідного обсягу буфера q від середнього коефіцієнта використання p підпорядковується такому співвідношенню:

$$q = \frac{\rho^{1/2(1-H)}}{(1-\rho)^{H/1-H}}. \qquad (2.36)$$

При H = 1/2 ця формула спрощується до $q = \rho/(1-\rho)$, що є класичним результатом системи масового обслуговування з експоненціально розподіленими часовими інтервалами між надходженнями запитів й експоненціально розподіленою тривалістю обслуговування (M/M/1). Для системи з постійним часом обслуговування (M/D/1) розмір буфера визначається в такий спосіб:

$$q = \frac{\rho(2-\rho)}{2(1-\rho)}. \qquad (2.37)$$

На рис. 2.11 показані результати для $H = 0{,}9$ й $H = 0{,}75$ у порівнянні з моделями M/M/1 і M/D/1. Як можна бачити, при великих значеннях H потреби в розмірі буфера починають стрімко зростати вже при незначному коефіцієнті використання. Це створює очевидний вплив на проектування буферів.

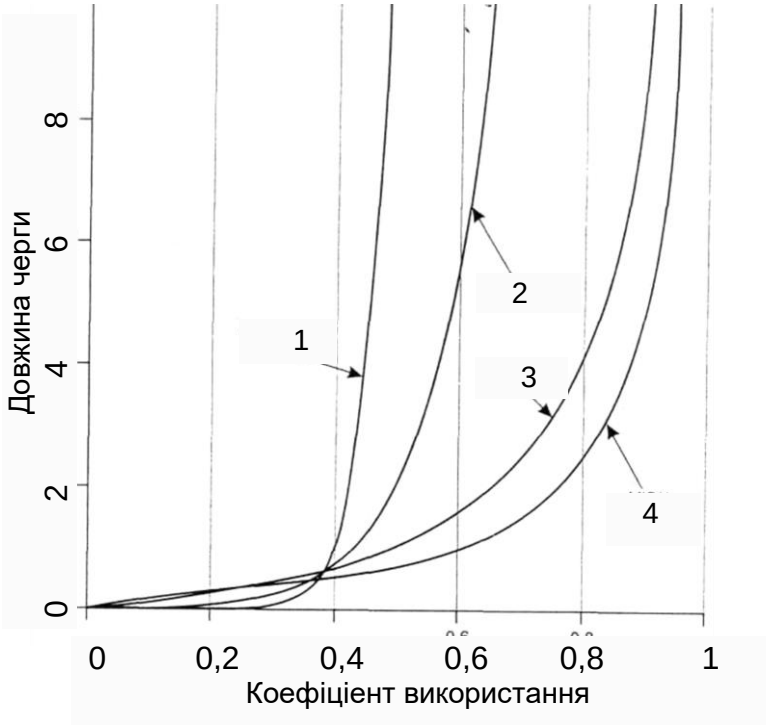


Рис. 2.11. Порівняння моделей черг

1 - самоподібна модель (H=0,9)

2 - самоподібна модель (H=0,75)

3 - модель M/M/1

4 - модель M/D/1

В [8] аналізувався вплив самоподібного трафіку на продуктивність мереж *Ethernet*. Одним з основних відкриттів було те, що збільшення навантаження на мережу призводить до збільшення степені самоподоби (зростання параметра Херста) трафіку. Цей аспект є дуже важливим, оскільки питання продуктивності здобуває особливого значення саме при високому навантаженні. Також показується, що при будь-якому мультиплексуванні самоподібних потоків тривалість затримок буде високою і потрібні буфери збільшеного розміру, тобто звичайне припущення, що мультиплексування великої кількості незалежних потоків дає в результаті пуасонівський процес,

виявляється неправильним. В [33] також вказується, що злиття, об'єднання потоків трафіку, наприклад, у статистичних мультиплексорах або ATM-комутаторах, призводить не до згладжування трафіку, а найчастіше навпаки, до збільшення ступеня самоподоби.

В [34] показано, що самоподоба може «розмножуватися», передаватися від одного потоку до іншого в сучасних комп'ютерних мережах із протоколом TCP завдяки механізму контролю перевантажень цього протоколу. Продемонстровано, що якщо з'єднання TCP використовує канал з малою пропускною здатністю разом із самоподібним потоком, то воно здобуває його кореляційну структуру. Авторами також показано, що якщо періоди перевантажень мають ПСЗ, то трафік, прийнятий кінцевим користувачем, також є ПСЗ.

У той же час, у ряді інших публікацій автори приходять до протилежних висновків: у деяких випадках вплив самоподоби на роботу мережі незначний і ним можна зневажити. Наприклад, в [35] відзначено, що великі значення параметра Херста не завжди призводять до збільшення довжин черг. В [36] показано, що в деяких системах імовірність переповнення буфера відносно нечутлива до самоподібних властивостей трафіку. В [37] представляється свідчення того, що при обчисленні розмірів буфера в мережах ATM самоподобою VBR-трафіку можна знехтувати. Таким чином, наявність самоподібних ефектів є істотною в одних мережевих конфігураціях і не створює значного впливу в інших [37]. Аналіз умов, за яких самоподоба є ключовим чинником впливу на продуктивність мережі є актуальною задачею, що вимагає подальшого вивчення. На даний час у цій області ведуться активні дослідження, які повинні привести до більш глибокого розуміння ефектів, що створюються самоподібним трафіком на різні процеси в мережі та на її продуктивність.

З іншого боку, якщо враховувати сильно корельований характер телетрафіку, це може призвести до більш ефективних механізмів керування трафіком, заснованих на прогнозуванні його інтенсивності, покладаючись на

спостереження трафіку протягом певного інтервалу часу [38,35]. Наприклад, в [39] указується, що так званий алгоритм доступної пропускної здатності для контролю за перевантаженнями *ABR* (*available bit-rate*), при розробці якого враховувалися властивості реального самоподібного телетрафіку, дозволяє при скінченному розмірі буфера досягти менших довжин черг і значно меншої імовірності втрати пакетів.

Можливість здійснювати прогнози, як уже згадувалося вище, зобов'язана властивості тривалої пам'яті процесів і теоретично повинна забезпечити підвищення коефіцієнта використання каналів і загальної ефективності системи. Докладно прогностичність трафіку розглядається в [40]. З розвитком напрямку самоподоби все більше з'являється робіт із пророкування інтенсивності трафіку [41,42,43] .

2.6. Обмеження самоподоби. Мультифрактали

Самоподібні процеси з ПСЗ є дуже привабливою моделлю для опису телекомунікаційного трафіку, оскільки вони математично визначені і уже досить повно вивчені й описані. Крім цього, їхня величезна перевага полягає в їх простоті й економності: всі їх властивості визначаються і управляються всього одним параметром, H. Однак, у цьому полягає і головний недолік самоподібної моделі. Дуже неправдоподібно, що така складна і різноманітна масштабно-інваріантна поведінка, виявлена у трафіку, може бути змодельована процесом з одним параметром. Самоподібна модель є дуже негнучкою в декількох відношеннях. По-перше, (2.10) повинен виконуватися для статистичних моментів усіх порядків (при $\forall q \in N$), або моменти вище другого порядку не враховуються взагалі (у випадку СШЗ). Однак емпіричні дані говорять про те, що моменти різних порядків можуть мати різні масштабні експоненти H_q, а деякі моменти можуть взагалі не

підпорядковуватись властивості масштабної інваріантності. По-друге, самоподоба має на увазі виконання (2.9) для всіх часових масштабів (коефіцієнтів масштабування) (при $\forall m \in (0;\infty)$), або поведінка процесу в малих масштабах часу не враховується (у випадку АСШЗ).

Подальші більш глибокі дослідження мережевого трафіку показали, що в той час, як його поводження у великих масштабах часу досить точно описується простими самоподібними моделями, у малих масштабах поведінка трафіку відрізняється набагато більшою складністю і меншою визначеністю, і не може бути адекватно змодельована за допомогою самоподібних процесів. Докладний аналіз мультифрактальних моделей трафіку, що описують його поведінку в малих масштабах часу, виходить за рамки даної роботи, однак зупинимося коротко на цьому питанні. Детально мультифрактали розглядаються в [44].

Як було показано раніше, основною причиною прояву телетрафіком самоподібних властивостей є активність кінцевих користувачів (що характеризується РВХ тривалостей з'єднання, розмірів файлів, що завантажують, періодів активності). Однак, на формування трафіку також істотно впливає сама мережа. Цей вплив може бути як безпосереднім (наприклад, вплив застосовуваних протоколів та алгоритмів передачі й обробки даних), так і результатом взаємодії з іншими потоками даних, що проходять по мережі. Трафік, сформований користувачами мережі, піддається трансформації, перерозподілу в процесі проходження різних вузлів і обробки на різних рівнях мережі (інкапсуляції протоколами, кожен з яких має свою динаміку). Зокрема, трафік може піддаватися різним затримкам при проходженні через проміжні вузли, при втраті пакетів і подальшій повторній передачі, через контроль вікон потоків у протоколі *TCP*, при реінкапсуляції, сегментуванні й повторній зборці пакетів у деяких інтерфейсах. Всі ці процеси пов'язані з часовими інтервалами в діапазоні від десятих часток до сотень мілісекунд, що набагато менше інтервалів, асоційованих з активністю користувачів. Таким чином, мережева динаміка не відбивається на

самоподібних властивостях трафіку в досить великих масштабах часу. Однак, у малих масштабах вона призводить до значної іррегулярності, нерівномірності в поведінці трафіку, що не враховується самоподібними (ПСЗ) моделями [45]. Існує характерний масштаб 1 с, що відокремлює монофрактальну поведінку трафіку (що вже стала класичною) від мультифрактальної [23].

Залишається ще багато роботи, щоб одержати ясне розуміння поведінки трафіку в малих масштабах часу, що характеризується набагато більшою мінливістю, більш складними і менш певними законами, а також необхідністю мати справу з негаусовими величинами і тому зі статистиками вище другого порядку.

Стаціонарний процес $\{X(t), t \in R\}$ має локальну регулярність Хьолдера з показником $h \in (0;1)$ у точці t_0, якщо виконується [8]

$$M\left|X(t_0+\delta)-X(t_0)\right|^2 \sim \sigma^2 |\delta|^{2h} \qquad \delta \to 0 \tag{2.38}$$

де σ^2 – дисперсія $X(t)$.

Автоковаріаційна функція $X(t)$ при $\delta \to 0$ має параболічний вигляд:

$$M\left|X(t)X(t+\delta)\right| \sim \sigma^2 \frac{1-C|\delta|^{2h}}{2C} \tag{2.39}$$

Локальна регулярність Хьолдера описує регулярність, неперервність стохастичного процесу шляхом локального порівняння зі степеневим законом спадання автоковаріації і тому тісно пов'язана з масштабною інваріантістю в малих масштабах часу [46]. Показник регулярності h, названий параметром (експонентою) Хьолдера, характеризує міру регулярності процесу: траєкторія процесу стає все більш мінливою, порізаною при зменшенні h від 1 до 0.

Процеси, для яких параметр Хьолдера постійний, не змінюється з часом, тобто $h(t) = const$, часто називаються монофрактальними процесами (самоподібні процеси). Нескладно показати, що для самоподібних процесів

(2.38) виконується в будь-який момент часу при $h = H$. Таким чином, параметр Хьолдера є локальним (асимптотичним при $m \to 0$) аналогом параметра Херста. На рис. 2.12 показана графічна інтерпретація локальної регулярності Хьолдера для $H = 0{,}25$ (в) і $H = 0{,}75$ (г), де в нижньому ряді 50 реалізацій процесу з показником Хьолдера $h = H$ об'єднані разом в одній випадковій точці; червоним кольором показана права частина (2.38). У верхньому ряді (а і б) для порівняння показані реалізації монофрактального процесу з відповідними параметрами Херста H

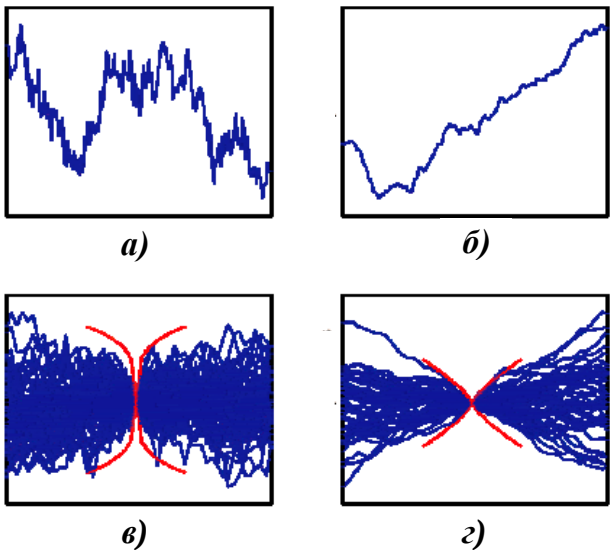


Рис. 2.12. Самоподоба та локальна регулярність Хьолдера

а - самоподібний процес (H=0,25)

б - самоподібний процес (H=0,75)

в - локальна регулярність Хьолдера (H=0,25)

г - локальна регулярність Хьолдера (H=0,75)

Однак для реального трафіку мереж передачі даних $h(t)$ безладно та випадково змінюється в часі, тобто $h(t)$ є сама по собі іррегулярною функцією часу. У цьому випадку процес X називають мультифрактальним, а флуктуації регулярності процесу описуються не функцією $h(t)$, а так званим мультифрактальним спектром $D(h)$. [46, 47]

Відзначимо, що фрактальні (самоподібні) моделі застосовні та використовуються не тільки в телекомунікаціях, але і у таких областях наукової діяльності, як кліматологія, економіка, біоінженерія, різні області фізики, обчислювальна техніка та ін. У біоінженерії, наприклад, фрактальний Броунівський рух використовується для моделювання розподілу плину крові в серці, легенях і нирках [48].

Питання для самоперевірки до розділу 2

1. Розподіли з важкими хвостами Парето.
2. Розподіли з важкими хвостами Вейбулла.
3. Розподіли з важкими хвостами Логарифмічно-нормальний.
4. Повільно спадна залежність самоподоби
5. Властивості самоподібних процесів: повільно спадна дисперсія.
6. Властивості самоподібних процесів: ефект Херста.
7. Метод оцінки параметра Херста. Метод R/S.
8. Метод оцінки параметра Херста. Оцінка Уіттла.
9. Фрактальний Броунівський рух.
10. Сходження суперпозиції PBX-процесів до самоподоби.
11. Фрактальна природа трафіку реальних телекомунікаційних мереж
12. Вплив самоподібного трафіку на продуктивність мереж
13. Обмеження самоподоби. Мультифрактали

3. Короткий огляд і класифікація методів

Після одержання таких численних емпіричних підтверджень того, що фрактальні (самоподібні) процеси істотно краще Пуасонівських відображають характеристики реального мережевого трафіку, такі як берстність і кореляції, стали особливо важливими розробка і застосування нових методів для генерування пакетного трафіку з метою моделювання.

З попередньої глави видно, що фрактальний Броунівський рух, самоподібний гаусовський процес, моделює деякі види трафіку даних краще і більш реалістично, ніж традиційні моделі. Оскільки отримано недостатньо теоретичних результататів щодо систем черг із процесом прибуття типу фрактального Броунівського руху, важливо знати, як цей процес можна симулювати. Таким чином, важливу роль грає вивчення моделювання, особливо для складних систем черг. Метою даної глави є опис ряду як точних, так і наближених методів моделювання фрактального Броунівського руху.

Оскільки моделювання можливо лише в дискретному часі, ми адаптували запис дискретного часу Y_0, Y_1, … для значень фрактального Броунівського руху на дискретні моменти часу 0, 1, ... Зі змодельованої одного разу вибірки фрактального Броунівського руху легко одержати реалізацію в інший еквідистантній решітці використовуючи властивість самоподоби.

Самі методи моделювання представлені в Додатку А, а програма генерації фрактального трафіку в Додатку Б.

Питання для самоперевірки до підрозділу 3

1. Точні методи моделювання. Метод Хоскінга.
2. Точні методи моделювання. Метод Холецького.

3. Точні методи моделювання. Метод Девіса й Харте.

4. Наближені методи моделювання.

5. Метод стохастичного подання фрактального Броунівського руху.

6. Агрегування процесів джерел (ON-/OFF- метод).

7. Метод випадкового зміщення проміжних точок і вдалий випадковий приріст.

8. Метод спектрального моделювання, метод Паксона і наближений циклічний метод

9. Спектральне моделювання

10. Метод Паксона

11. Наближений циклічний метод

12. Моделювання, засноване на вейвлетах

13. Метод хаотичних карт

14. Метод Декройсфонда та Лавода

15. Метод Джапарідзе і ван Зантена

ВИСНОВКИ

Моделювання самоподібного трафіку на даний час є однією з найбільш перспективних і молодих областей теорії сучасних телекомунікаційних мереж. У представленій роботі обґрунтована актуальність використання підходів самоподоби до моделювання мережного трафіка, досліджений вплив таких підходів на продуктивність і якість обслуговування різних сучасних пакетних мереж. Також систематизовані математичні моделі, що описують реальний трафік мереж з комутацією пакетів, і, як ми думаємо, вперше була представлена в наочному вигляді їхня класифікація. Досліджено та проаналізовано існуючі методи моделювання самоподібного трафіка з погляду їхньої точності і швидкості. Так, з погляду швидкості роботи найкращим з точних методів є метод Девіса і Харте. З наближених методів найбільш раціональним можна вважати метод випадкового зсуву середньої точки процесу (ВЗСТ), що є найбільш швидким із всіх розглянутих і забезпечує найкраще наближення за точністю самоподоби вибірки, що генерується до точних методів. Також, одним з найбільш перспективних апроксимуючих методів є наближений циклічний метод, одержаний модифікацією методу Девіса і Харте.

Проведений аналіз може слугувати гарною базою для продовження досліджень у цій області з метою відпрацювання представлених методів моделювання на практиці і подальшому їхньому використанні для розв’язання прикладних задач мережної інженерії.

Також даний посібник з поглибленим представленням моделювання самоподібного трафіку рекомендовано для використання при написанні магістерської дисертації.

Список використаної літератури

[1] A History of Engineering and Science in the Bell System, The Early Years (1875-1925) / Fagen M., ed. – Bell Telephone Laboratories, Incorporated, 1975, ISBN 0608080667.

[2] Bradley A. B. Traffic Models for Hybrid Satellite-Terrestrial Networks: Master's thesis, Number: CSHCN MS 2000-1. – University of Maryland, 1999. http://hdl.handle.net/1903/6176.

[3] Willinger W., Paxson V. Where Mathematics meets the Internet // Notices of the American Mathematical Society. – 1998. – Vol. 45.- № 8.- P. 961–970.

[4] Романов О.І. Телекомунікаційні мережі та управління: навчальний посібник. - К.: Київський університет», 2003. – 247с.

[5] Jiang H. Source-level IP packet bursts: causes and effects // Proceedings of the 3rd ACM SIGCOMM conference on internet measurements, 2003. http://dx.doi.org/10.1145/948205.948245.

[6] Ложковський А.Г. Нові методи теорії телетрафіка. – Одеса: ОНАЗ ім. О.С. Попова, 2018. – 80 с.

[7] TM8102 - Traffic Analysis of Communication Networks: academic year 2022/2023 // https://www.ntnu.edu/studies/courses/TM8102#tab=omEmnet.

[8] Leland W., Taqqu M., Willinger W., Wilson D. On the Self-Similar Nature of Ethernet Traffic (extended version) // IEEE/ACM Transactions on Networking. – 1994. – Vol. 2, № 1. – P. 1–15.

[9] Jeong H.-D. J. Modelling of self-similar teletraffic for simulation: PhD thesis. – University of Canterbury, July 2002. https://ir.canterbury.ac.nz/bitstream/handle/10092/5454/jeong_thesis.pdf?sequence=1.

[10] Mandelbrot B.B., van Ness J.W. Fractional Brownian motions, fractional noises and applications // SIAM Review. – 1968. – Vol. 10. – P. 422–437.

[11] Norros I. A storage model with self-similar input // Queueing Systems. – 1994. – Vol. 16. – P. 387–396.

[12] Crovela M., Bestavros A. Self-similarity in World-Wide Web traffic: evidence and possible causes // Proceedings of ACM Sigmetrics Conference on Measurement and Modelling of Computer Systems, May 1996.

[13] Arlitt M, Friedrich R., Jin T. Workload characterization of a Web proxy in a cable modem environment // ACM Performance Evaluation Review, September 1999.

[14] Duffy D., McIntosh A., Rosenstein M, Willinger W. Statistical Analysis of CSN/SS7 Traffic Data from Working CCS Subnetworks // IEEE Journal on Selected Areas in Communications. – April 1994.

[15] Giroux N., Ganti S. Quality of Service in ATM Networks. – Upper Saddle River, NJ: Prentice Hall 1999.

[16] Borela M., Brewster G. Measurement and Analyses of Long-Range Packet Dependent Behaviour of Internet Packet Delay // IEEE INFOCOM'98. – April 1998.

[17] Beran J., Sherman R., Taqqu M., Willinger W. Long-Range Dependence in Variable-Bit-Rate Video Traffic // IEEE Transactions on Communications. – 1995. – Vol. 43, 2. – P. 1566–1579.

[18] Ji C., Ma S., Tian X. Approximation Capability of Independent Wavelet Models to Heterogeneous Network Traffic // Proceedings of IEEE INFOCOM'99. – New York, NY, USA. – 1999. – P. 170–177.

[19] Ma S., Ji C. Modelling Video Traffic in the Wavelet Domain // Proceedings of IEEE INFOCOM'98. – San Francisco, CA, USA. – 1998. – P. 201–208.

[20] Garrett M., Willinger W., Analysis, Modelling, and Generation of Self-Similar VBR Video Traffic // Proceedings, SIGCOMM'94. – August 2004.

[21] Park K., Kim G., Crovella M. On the Relationship Between File Sizes, Transport Protocols, and Self-Similar Network Traffic // Proceedings of IEEE International Conference on Network Protocols. – 1996. – P. 171–180.

[22] Roberts J. Traffic Theory and the Internet // IEEE Communications Magazine. – 2001. – P. 94–99.

[23] Abry P., Baraniuk R., Flandrin P., Riedi R., Veitch D. Multiscale Nature of network Traffic // IEEE Signal Processing Magazine. – May 2002. – P. 28-46.

[24] Willinger W., Taqqu M., Wilson D. Self-similar traffic modelling for high-speed networks // ConneXions, November 1994.

[25] Park K., Kim G., Crovella M. On the Effect of Traffic Self-Similarity on Network Performance // Proceedings of SPIE International Conference on Performance and Control of Network Systems. – 1997. – P. 296–310.

[26] Erramilli A., Narayan O., Willinger W. Experimental Queuing Analysis with Long Range Dependent Packet Traffic // IEEE/ACM Transactions on Networking. – 1996. – Vol. *4*, №2. – P. 209–223.

[27] Erramilli A., Gordon J., Willinger W. Applications of Fractals in Engineering for Realistic Traffic Processes // Proceedings, International Telecommunications Conference (ITC-14). – Amsterdam: Elsevier, 1994. – Vol. 1a. – P. 35–44.

[28] Paxson V., Floyd S. Wide-Area Traffic: the Failure of Poisson Modeling // IEEE/ACM Transactions on Networking. – 1995. – Vol. 3, No. 3. – p. 226-244

[29] Duffield N. Queueing at Large Resources Driven by Long-Tailed $M=G=\infty$-Modulated Processes // Queueing Systems. – 1998. – 28. – P. 245–266.

[30] Dahl T., Willemain T. The Effect of Long-Memory Arrivals on Queue Performance // Operations Research Letters. – 2001. – 29. – P. 123–127.

[31] Georganas N. Self-Similar (“Fractal") Traffic in ATM Networks // Proceedings of the 2nd International Workshop on Advanced Teleservices and High-Speed Communication Architectures (IWACA'94). – Heidelberg, Germany, 1994. – P. 1–7.

[32] Tsybakov B., Georganas N. Self-Similar Traffic and Upper Bounds to Buffer-Overflow Probability in an ATM Queue // Performance Evaluation. – 1998. – 32. – P. 57–80.

[33] Fan Y., Georganas N. On Merging and Splitting of Self-Similar Traffic in High-Speed Networks // Proceedings of ICCC'95. – Seoul, Korea, 1995.

[34] Veres A., Kenesi Z., Molnár S., Vattay, G. On the Propagation of Long-Range Dependence in the Internet // Computer Communication Review, Proceedings of ACM SIGCOMM'00. – Stockholm, Sweden, 2000. – Vol. 30. № 4. – P. 243–254.

[35] Neidhardt A., Wang J. The Concept of Relevant Time Scales and its Application to Queuing Analysis of Self-Similar Traffic (or Is Hurst Naughty or Nice?) // Performance Evaluation Review, Proceedings of ACM SIGMETRICS'98. Madison, Wisconsin, USA, 1998. – P. 222–232.

[36] Grossglauser M., Bolot J.-C. On the Relevance of Long-Range Dependence in Network Traffic // IEEE ACM Transactions on Networking. – 1999. – *Vol. 7. №* 5. – P. 629–640.

[37] Ryu B., Lowen S. Point Process Approaches for Modelling and Analysis of Self-Similar Traffic, Part II – Applications // Proceedings, International Conference on Telecommunications Systems, Modelling, and Analysis. – March 1997.

[38] Jeong H.-D., McNickle D., Pawlikowski K. A Comparative Study of Three Self-Similar Teletraffic Generators // Proceedings of 13th European Simulation Multiconference, ESM'99. – Warsaw, Poland, 1999. – Vol. 1. – P. 356–362.

[39] Östring S., Sirisena H., Hudson I. Sensitivity of ABR Congestion Control Algorithms to Hurst Parameter Estimates // Proceedings of Networking 2000. – Paris, France, 2000. – P. 36–48.

[40] Park K., Willinger W. Self-Similar Network Traffic and Performance Evaluation. – John Wiley & Sons. – 2000.

[41] Östring S., Sirisena H. The Influence the Long-Range Dependence on Traffic Prediction // Proceedings of ICC'01. – Helsinki, June 2001.

[42] Miloucheva I., Műller E., Anzaloni A. A practical approach to forecast Quality of Service parameters considering outliers. – 2003.

[43] Trajkovic L., Neidhardt A. Internet traffic prediction // Centre for Systems Science, Simon Fraser University. – Mar. 2000. – Vol. 12, Issue 1.

[44] Rudolf H. Riedi. Multifractal Processes // Long Range Dependence: theory and applications. – Birkhauser. – 2002. – p. 625-715

[45] Kant K. On Aggregate Traffic Generation with Multifractal Properties // IEEE, Seamless Interconnection for Universal Services Global Telecommunications Conference. – 1999. – Vol. 2. – p. 1179-1183

[46] Falconer K. Fractal Geometry – Mathematical Foundations and Applications. – New York: Wiley, 1990 – P. 337.

[47] Riedi R. An improved multifractal formalism and self-similar measures // J. Math. Anal. and Appl. – 1995. – Vol. 189. – P. 462-490.

[48] Dieker T. Simulation of fractional Brownian motion. – University of Twente, Amsterdam, 2004. http://www.cwi.nl/~ton

[49] Hosking J.R.M. Modelling persistence in hydrological time series using fractional differencing // Water resources research. – 1984. – 20. – P. 1898–1908.

[50] Brockwell P.J., Davis R.A. Time series: Theory and methods – New York: Springer-Verlag. – 1987.

[51] Asmussen S. Stochastic simulation with a view towards stochastic processes // Notes of a Concentrated Advance Course at MaPhySto. – Aarhus, Denmark, February 1999.

[52] Davies R.B., Harte D.S. Tests for Hurst effect // Biometrika. – 1987. – No. 74 (1). – P. 95–102.

[53] Dietrich C.R., Newsam G.N. Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix // SIAM Journal Sci. Comput. – 1997. – 18. – P. 1088–1107.

[54] Wood A.T.A., Chan G. Simulation of stationary Gaussian processes in $[0; 1]^d$ // Journal of Computational and Graphical Statistics. – 1994. – 3. – P. 409–432.

[55] Coeurjolly J.F. Inférence statistique pour les mouvements browniens fractionnaires et multifractionnaires: PhD thesis. – Université Joseph Fourier. – Dec. 2000.

[56] Willinger W., Taqqu M., Sherman R., Wilson D. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level // IEEE/ACM Transactions on Networking. – 1997. – Vol. 5, № 1. – P. 71–86.

[57] Крилов В.В., Самохвалова С.С. Теорія телетрафіка та її застосування. 2005. – С. 288.

[58] Norros I., Mannersalo P., Wang J.L. Simulation of fractional Brownian motion with conditionalized random midpoint displacement // Advances in Performance Analysis. – 1999. – 2. – P. 77–101.

[59] Dieker A.B., Mandjes M. On spectral simulation of fractional Brownian motion // Probab. Engrg. Inform. Sci. – 2003. – 17. – P. 417–434.

[60] Paxson V. Fast, approximate synthesis of fractional Gaussian noise for generating self-similar network traffic // Computer Communication Review. – 1997. – 27. – P. 5–18.

[61] R. H. Riedi, V. J. Ribeiro, M. S. Crouse, and R. G. Baraniuk. Network Traffic Modeling using a Multifractal Wavelet Model // European Congress of Mathematics. – Jul, 2000.- p. 609-618

[62] Daubechies I. Ten lectures on wavelets, CBMS-NSF Regional Conference Series, SIAM, 1992.

[63] Abry P., Sellan F. The wavelet-based synthesis for fractional Brownian motion proposed by F. Sellan and Y. Meyer: remarks and fast implementation // Applied and computational harmonic analysis. – 1996. – 3. – P. 377–383.

[64] Feyel D., de La Pradelle A. On fractional Brownian processes // Potential Anal. – 1999. – 10. – P. 273–288.

[65] Erramilli A., Singh R.P., Pruthi P. Chaotic maps as models of packet traffic // Proc. 14th Int. Teletraffic Cong. – 6-10 June 1994, vol. 1, North-Holland Elsevier Science B.V., 1994, – P. 329–338.

[66] Decreusefond L., Lavaud N. Simulation of the fractional Brownian motion and application to the fluid queue // Proceedings of the ATNAC '96 conference. – 1996.

[67] Dzhaparidze K., van Zanten H. Optimality of an explicit series expansion of the fractional Brownian // Statistics and Probability Letters. – 2005. – vol. 74, issue 4. – p. 295-301

Додаток А. МЕТОДИ МОДЕЛЮВАННЯ САМОПОДІБНОГО ТРАФІКУ

1 Точні методи

1.1 Метод Хоскінга

Метод Хоскінга (*Hosking*) (також відомий як метод Дурбіна (*Durbin*) або Левінсона (*Levinson*)) – це алгоритм моделювання загального стаціонарного Гаусового процесу; тому, зосередимося на моделюванні фрактального Гаусового шуму $X_0, X_1, \ldots, X_n$. Відзначимо, що вибірка фрактального Броунівського руху $Y_0, Y_1, \ldots$ отримується із вибірки фрактального Гаусового шуму взяттям кумулятивних сум. Метод рекурсивно генерує X_{n+1} за даними $X_n, \ldots, X_0$. Він не використовує специфічні властивості ані фрактального Броунівського руху, ані фрактального Гаусового шуму; алгоритм може бути застосований до будь-якого стаціонарного Гаусового процесу. Ключовою ідеєю є те, що розподіл X_{n+1} при даному минулому процесу, може бути обчислений в явному вигляді.

Запишемо $\gamma(k)$ – функцію коваріації процесу (з нульовим середнім):

$$\gamma(k) := M\left|X_n X_{n+k}\right| \tag{3.1}$$

при $n, k = 0,1,2,\ldots$. Приймемо для зручності $\gamma(0) = 1$. До того ж, нехай $\Gamma(n) = \left(\gamma(i-j)\right)_{i,j=0,\ldots,n}$ – матриця коваріації і $c(n)$ – вектор з $(n+1)$ стовпцями і з елементами $c(n)_k = \gamma(k+1), \quad k = 0, \ldots n..$

Визначимо матрицю $F(n) = \left(1(i = n - j)\right)_{i,j=0,\ldots,n}$ розмірів $(n+1)\times(n-1)$, де $1(\cdot)$ – функція індикації. Відзначимо, що при множенні цієї матриці на вектор-рядок або множенні вектора-стовпця на матрицю цей вектор «транспонується».

Матриця $\Gamma(n+1)$ може бути розбита в такий спосіб:

$$\Gamma(n+1) = \begin{pmatrix} 1 & c(n)' \\ c(n) & \Gamma(n) \end{pmatrix} = \tag{3.2}$$

$$= \begin{pmatrix} \Gamma(n) & F(n)c(n) \\ c(n)'F(n) & 1 \end{pmatrix}, \tag{3.3}$$

де «штрих» означає транспонування.

Далі обчислюємо умовний розподіл X_{n+1} при даних $X_n,\ldots,X_0$. Фактично, ми показуємо, що цей розподіл є Гаусовим з математичним очікуванням μ_n і дисперсією σ_n^2, рівними:

$$\mu_n := c(n)'\Gamma(n)^{-1}\begin{pmatrix} X_n \\ \vdots \\ X_1 \\ X_0 \end{pmatrix}, \tag{3.4}$$

$$\sigma_n^2 := 1 - c(n)'\Gamma(n)^{-1}c(n). \tag{3.5}$$

Визначимо $d(n) = \Gamma(n)^{-1}c(n)$ для зручності. Нескладно перевірити за (3.2) і (3.3), що для транспонованої $\Gamma(n+1)$ виконується таке:

$$\Gamma(n+1)^{-1} = \frac{1}{\sigma_n^2}\begin{pmatrix} 1 & -d(n)' \\ -d(n) & \sigma_n^2\Gamma(n)^{-1} + d(n)d(n)' \end{pmatrix} = \tag{3.6}$$

$$= \frac{1}{\sigma_n^2}\begin{pmatrix} \sigma_n^2\Gamma(n)^{-1} + F(n)d(n)d(n)'F(n) & -F(n)d(n) \\ -d(n)'F(n) & 1 \end{pmatrix}, \tag{3.7}$$

З (3.6) легко зробити висновок, що для будь-якого $x \in R^{n+1}$ та $y \in R$

$$(y \quad x')\Gamma(n+1)^{-1}\begin{pmatrix} y \\ x \end{pmatrix} = \frac{\left(y - d(n)'x^2\right)}{\sigma_n^2} + x'\Gamma(n)^{-1}x \tag{3.8}$$

Звідси випливає, що розподіл X_{n+1} при даних $X_n,\ldots,X_0$ дійсно є Гаусовим з математичним очікуванням μ_n і дисперсією σ_n^2 .

Тепер, коли ми знаємо цей розподіл, необхідна вибірка отримується за допомогою генерування нормованої випадкової величини, розподіленої за нормальним законом і рекурсивним моделюванням X_{n+1} для n = 0, 1, … ... Звичайно, складність полягає в організації обчислень таким чином, щоб уникати транспонування матриці на кожному кроці. Алгоритм, запропонований Хоскінгом [49], рекурсивно обчислює $d(n)$. Алгоритм, що представлено тут злегка відрізняється, але він більш швидкий, ніж первісний.

Припустимо, що ми знаємо μ_n, σ_n^2 й $\tau_n = d(n)'F(n)c(n) = c(n')F(n)d(n)$. З (3.7) видно, що σ_n^2 задовольняє наступній рекурсії

$$\sigma_{n+1}^2 = \sigma_n^2 - \frac{\left(\gamma(n+2) - \tau_n\right)^2}{\sigma_n^2} \tag{3.9}$$

Рекурсія для $d(n+1) = \Gamma(n+1)^{-1}c(n+1)$ також отримана з (3.7):

$$d(n+1) = \begin{pmatrix} d(n) - \phi_n F(n)d(n) \\ \phi_n \end{pmatrix} \tag{3.10}$$

де $\phi_n = \dfrac{\gamma(n+2) - \tau_n}{\sigma_n^2}$.

Тому перші n елементів $(n+2)$-рядкового вектора $d(n+1)$ можна обчислити за $d(n)$ й ϕ_n. При відомому $d(n+1)$ нескладно обчислити $\mu_{n+1}, \sigma_{n+1}^2$ та τ_{n+1}. Ми починаємо рекурсію з $\mu_0 = \gamma(1)X_0$, $\sigma_0^2 = 1 - \gamma(1)^2$ й $\tau_0 = \gamma(1)^2$. Для одержання вибірки фрактального Броунівського руху $Y_0, Y_1, \ldots$ необхідно взяти кумулятивні суми отриманого ряду $X_0, X_1, \ldots, X_n$.

Підхід, тісно пов'язаний з алгоритмом Хоскінга – це алгоритм нововведень (innovations). Замість рекурсивного генерування X_{n+1} за даними $X_n,\ldots,X_0$ цей метод симулює нововведення X_{n+1} за даними $X_n-\mu_{n-1},\ldots,X_1-\mu_0,X_0$.. Перевага цього алгоритму в тому, що він може застосовуватися і для нестаціонарних процесів. Для більш докладної інформації див. [50].

Через обчислення рекурсій $d(n)$, μ_n й v_n, складність алгоритму є порядку N^2 при вибірці розміру N. Наприклад, обчислення $d(n)-\phi_n F(n)d(n)$ за даними ϕ_n й $d(n)$ вимагає порядку n послідовних обчислень. Обчислення цього для $n = 1, 2, \ldots, N$ вимагає порядку N^2 операцій.

Перевага методу Хоскінга в тому, що вибірки можуть генеруватися «на льоту». Це означає, що розмір вибірки не потрібно знати заздалегідь. Це потрібно, наприклад, коли симуляція повинна зупинитися в якийсь випадковий момент часу. Іншим достоїнством цього методу є його надзвичайна простота. Це використовується для вираження деяких ймовірностей на основі побічних продуктів алгоритму, як ϕ_n.

Коли потрібно більше однієї вибірки, наприклад при моделюванні з декількома «прогонами», немає потреби робити заново більшість обчислень. Однак через обчислення μ_n, що залежать від сгенерованих точок $X_0,\ldots,X_n$, кількість операцій залишається порядку N^2; часова складність не зменшується.

1.2 Метод Холецького

Не дивно, що метод Холецького (*Cholesky*) [51] використовує так зване розкладання Холецького матриці коваріації. Це означає, що матриця коваріації $\Gamma(n)$ може бути представлена як $L(n)L(n)'$, де $L(n)$ це $(n+1)\times(n+1)$ нижня

трикутна матриця. Позначивши елемент (i, j) матриці $L(n)$ як l_{ij} для $i,\ j = 0, \ldots n,$ назвемо матрицю $L(n)$ нижньою трикутною, якщо $l_{ij} = 0$ при $j > i$. Можна довести, що таке розкладання існує, якщо $\Gamma(n)$ є симетричною додатно визначеною матрицею.

На відміну від методу Хоскінга, цей метод може бути застосований до нестаціонарних гаусових процесів, але ми припускаємо стаціонарність із причин зручності.

Елементи $L(n)$ можна обчислити, помітивши, що елементи $(i; j)$ $L(n)L(n)'$ й $\Gamma(n)$ повинні бути рівні при $j < i$ (а також при $j > i$ через симетрію). Тобто

$$\gamma(i - j) = \sum_{k=0}^{j} l_{ik} l_{jk}, \qquad j \le i \tag{3.11}$$

Це рівняння зводиться до $\gamma(0) = l_{00}^2$ при $i = j = 0$. При $i = 1$ ми одержуємо два рівняння що визначають l_{10} і l_{11}:

$$\gamma(1) = l_{10} l_{00} \tag{3.12}$$

$$\gamma(0) = l_{10}^2 + l_{11}^2, \tag{3.13}$$

Оскільки зрозуміло, що l_{ij} не може залежати від n, $L(n+1)$ може бути отримана за допомогою додавання ряду (і певної кількості нулів, щоб зробити $L(n+1)$ нижньою трикутною), що визначається таким чином

$$l_{n+1,0} = \frac{\gamma(n+1)}{l_{00}} \tag{3.14}$$

$$l_{n+1,\,j} = \frac{1}{l_{jj}} \left(\gamma(n+1-j) - \sum_{k=0}^{j-1} l_{n+1,k} l_{jk} \right), \qquad 0 < j \le n \tag{3.15}$$

$$l_{n+1,n+1}^2 = \gamma(0) - \sum_{k=0}^{n} l_{n+1,k}^2 .$$

(3.16)

Із цих формул випливає, що $L(n)$ унікальна при додатковому обмеженні, що елементи на головній діагоналі строго додатні. Якщо $\Gamma(n+1)$ є додатно визначеною матрицею, невід'ємність $l_{n+1,n+1}^2$ гарантована, отже матриця $L(n+1)$ існує.

Позначимо як $V(n) = (V_i)_{i=0,\ldots,n}$ вектор з $(n+1)$ стовбцями незалежних однаково розподілених (н.о.р.) нормованих випадкових величин, розподілених за нормальним законом, і побудуємо $V(n+1)$ по $V(n)$ за допомогою додавання нормованої випадкової величини, розподіленої за нормальним законом. Ключова ідея – рекурсивно моделювати $X(n) = L(n)V(n)$. Для кожного $n \geq 0$, $X(n)$ має нульове середнє та матрицю коваріації

$$Cov(L(n)V(n)) = L(n)Cov(V(n))L(n)' = L(n)L(n)' = \Gamma(n) \qquad (3.17)$$

Підбиваючи підсумок, X_{n+1} можна досить легко змоделювати, як тільки обчислена $L(n+1)$:

$$X_{n+1} = \sum_{k=0}^{n+1} l_{n+1,k} V_k \qquad (3.18)$$

Як і у методі Хоскінга, не потрібно заздалегідь встановлювати період часу моделювання. Недолік методу в тому, що він повільний (порядку N^3 при N точках) і вимогливий відносно пам'яті, тому що матрицю $L(n)$, що збільшується з кожним кроком рекурсії, доводиться постійно тримати в пам'яті.

Якщо необхідно більше однієї вибірки, значний обчислювальний час можна зберегти, обчисливши $L(n)$ лише один раз. Фактично, тоді кожна додаткова вибірка вимагає порядку N^2 послідовних операцій.

Хоча це не очевидно при порівнянні алгоритмів, метод Хоскінга власне кажучи рівноцінний методу Холецкого (але швидше!) у розумінні, що метод Хоскінга неявно обчислює ту ж матрицю $L(n)$. Дійсно, у методі Хоскінга, X_{n+1} генерується за $\mu_n + \sigma_n V_{n+1}$ і μ_n залежить тільки від $X_n, \ldots, X_0$. Аналогічно, X_0 генерується за $\mu_{n-1} + \sigma_{n-1} V_n$, де μ_{n-1} залежить лише від $X_{n-1}, \ldots, X_0$. Інакше кажучи, X_{n+1} генерується за деякою лінійною комбінацією $n+1$ н.о.р. нормованих випадкових величин $h_{n+1,k}$, розподілених за нормальним законом:

$$X_{n+1} = \sum_{k=1}^{n+1} h_{n+1,k} V_k, \tag{3.19}$$

де ми знаємо, що $h_{n+1,n+1} = \sigma_n > 0$. Тепер зрозуміла аналогія з (3.18): так само, як і у методі Холецького, вибірка генерується множенням вектора з н.о.р. нормованими компонентами, розподіленими за нормальним законом на квадратну нижню трикутну матрицю.

1.3 Метод Девіса й Харте

Цей алгоритм був напочатку запропонований Девісом (Davies) і Харте (Harte) [52] і пізніше узагальнений Дітріхом (Dietrich) і Ньюсемом (Newsam) [53] і Вудом (Wood) і Чаном (Chan) [54]. Як і у двох попередніх методах, цей метод намагається добути “квадратний корінь” з матриці коваріації, у тому розумінні, що $\Gamma = GG'$ при деякій квадратній матриці G. Припустимо, що необхідна вибірка розміру N і що розмір матриці коваріації – це степінь двох, $N = 2^g$ при якомусь $g \in N$. Щоб ефективно обчислити такий квадратний корінь, Γ вбудовується в так називану циклічну матрицю коваріації C розміру $2N = 2^{g+1}$. Для більшої конкретності, визначимо C:

$$C=\begin{pmatrix} \gamma(0) & \gamma(1) & \cdots & \gamma(N-1) & 0 & \gamma(N-1) & \gamma(N-2) & \cdots & \gamma(2) & \gamma(1) \\ \gamma(1) & \gamma(0) & \cdots & \gamma(N-2) & \gamma(N-1) & 0 & \gamma(N-1) & \cdots & \gamma(3) & \gamma(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma(N-1) & \gamma(N-2) & \cdots & \gamma(0) & \gamma(1) & \gamma(2) & \gamma(3) & \cdots & \gamma(N-1) & 0 \\ 0 & \gamma(N-1) & \cdots & \gamma(1) & \gamma(0) & \gamma(1) & \gamma(2) & \cdots & \gamma(N-2) & \gamma(N-1) \\ \gamma(N-1) & 0 & \cdots & \gamma(2) & \gamma(1) & \gamma(0) & \gamma(1) & \cdots & \gamma(N-3) & \gamma(N-2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma(1) & \gamma(2) & \cdots & 0 & \gamma(N-1) & \gamma(N-2) & \gamma(N-3) & \cdots & \gamma(1) & \gamma(0) \end{pmatrix} \quad (3.20$$

Відзначимо, що i -тий ряд можна отримати переміщенням першого ряду на $i-1$ елементів вправо і вставкою елементів, що залишилися, з лівої сторони. Також відзначимо, що матриця симетрична і що верхній лівий кут – матриця Γ. Якщо $\gamma(k)$ є функцією коваріації фрактального Гаусового шуму і нульові елементи в матриці замінити на $\gamma(N)$, матриця є додатно визначеною. Заміна їх на нульові ніяк не впливає на одержувану вибірку фрактального Гаусового шуму, якщо λ_j невід'ємні (на практиці, це задовольняється для будь-яких прийнятних обсягів вибірки N).

Необхідно сказати, що циклічна матриця не обов'язково є додатно визначеною для загальних функцій автоковаріації, але в [53] описано, як бути в такій ситуації. Для деяких функцій автоковаріації навіть неможливо включити матрицю коваріації в додатно визначену матрицю. Автори пропонують у такому випадку використовувати наближений метод.

Якщо кількість необхідних точок – не степінь двійки, доведеться вставити більшу кількість нулів у перший ряд для одержання циклічної матриці, але це не змінює сутності методу.

Алгоритм використовує таку теорему. Будь-яка циклічна матриця C може бути розкладена як $C=Q\Lambda Q^*$, де Λ є діагональною матрицею власних значень C, а Q є унітарною матрицею, визначеної як

$$(Q)_{jk} = \frac{1}{\sqrt{2N}} \exp\left(-2\pi \hat{i} \frac{jk}{2N} \right) \quad \text{для} \quad j,k = 0,...,2N-1, \tag{3.21}$$

Тут і далі $\hat{i} = \sqrt{-1}$ – уявна одиниця.

Q^* означає матрицю, комплексно спряжену із транспонованою Q. Легко перевірити, що Q дійсно унітарна: $QQ^* = I$. Власні значення, що складають матрицю Λ, визначаються так:

$$\lambda_k = \sum_{j=0}^{2N-1} r_j \exp\left(2\pi \hat{i} \frac{jk}{2N} \right) \qquad \text{для} \quad k = 0,...,2N-1, \tag{3.22}$$

де r_j – $(j+1)$-й елемент першого ряду матриці C.

Оскільки матриця C є додатно визначеною і симетричною, отримані власні значення будуть додатні та дійсні. Таким чином, матриця із власними значеннями $\sqrt{\lambda_1},...,\sqrt{\lambda_{2N-1}}$ і з такими ж власними векторами як C, є також додатно визначеною і дійсною. Відзначимо, що оскільки $C = Q\Lambda Q^*$ і Q унітарна, матриця $S = Q\Lambda^{1/2}Q^*$ задовольняє умові $SS^* = SS' = C$. Таким чином, S має в точності бажану властивість. Однак S *не* є тією же матрицею, що була в методі Холецького, тому що S не є нижньою трикутною.

Щоб отримати вибірку процесу, нам потрібно знайти спосіб змоделювати $Q\Lambda^{1/2}Q^*V$, де V є вектором з н.о.р. нормованими елементами, розподіленими за нормальним законом. Це робиться в три послідовних етапи:

1. Обчислюються власні значення за (3.22), що ефективно проводиться за допомогою ШПФ. При даній комплексній послідовності $(\alpha_k)_{k=0}^{j-1}$, ШПФ є алгоритмом виконання перетворення Фур'є цієї послідовності, тобто

$$\sum_{k=0}^{j-1} \alpha_k \exp\left(2\pi \hat{i} \frac{nk}{j} \right) \tag{3.23}$$

для $n = 0,..., j-1$. Якщо j є степінем двох, кількість обчислень, необхідних для ШПФ, порядку $j\log(j)$; істотний виграш у швидкості в порівнянні з безпосереднім обчисленням порядку j^2.

2. Обчислюємо $W = Q^* V$. Одержання структури коваріації W приводить до наступної процедури симуляції:

Генеруємо дві нормовані нормальні випадкові величини W_0 й W_N;

Для $1 \le j < N$ генеруємо дві незалежні нормовані нормальні випадкові величини $V_j^{(1)}$ й $V_j^{(2)}$ і нехай

$$W_j = \frac{1}{\sqrt{2}}(V_j^{(1)} + \hat{i} V_j^{(2)}) \tag{3.24}$$

$$W_{2N-j} = \frac{1}{\sqrt{2}}(V_j^{(1)} - \hat{i} V_j^{(2)}) \tag{3.25}$$

Отриманий вектор W має такий же розподіл як $Q \cdot V$.

3. Обчислюємо $Z = Q\Lambda^{1/2} W$:

$$Z_k = \frac{1}{\sqrt{2N}} \sum_{j=0}^{2N-1} \sqrt{\lambda_j} W_j e^{-2\hat{i}\pi \frac{jk}{2N}} \tag{3.26}$$

Знов-таки, це обчислення краще зробити за допомогою ШПФ для максимальної швидкості. У підсумку, послідовність $(Z_k)_{k=0}^{2N-1}$ є перетворенням Фур'є

$$w_k = \begin{cases} \sqrt{\frac{\lambda_k}{2N}} V_k^{(1)} & k = 0; \\ \sqrt{\frac{\lambda_k}{4N}} \left(V_k^{(1)} + \hat{i} V_k^{(2)} \right) & k = 1,\ldots, N-1; \\ \sqrt{\frac{\lambda_k}{2N}} V_k^{(1)} & k = N; \\ \sqrt{\frac{\lambda_k}{4N}} \left(V_{2N-k}^{(1)} + \hat{i} V_{2N-k}^{(2)} \right) & k = N+1,\ldots, 2N-1 \end{cases} \tag{3.27}$$

Вибірка фрактального Гаусового шуму отримується взяттям N перших елементів Z. X є дійсною за будовою.

З (3.20) видно, що останні N елементів Z також мають бажану структуру коваріації. Отже в цьому методі, ми одержуємо другу вибірку "на шару". Однак, ці дві вибірки *не* можуть бути з'єднані, щоб одержати вдвічі більшу вибірку, оскільки структура кореляції *між* двома вибірками не відповідає фрактальному Гаусовому шуму. Оскільки ці дві вибірки крім того не є незалежними, ця друга вибірка найчастіше не використовується.

Головна перевага цього методу – швидкість. Точніше кажучи, кількість обчислень становить порядку $N \cdot \log(N)$ для вибірки довжиною N. Якщо потрібно більше реалізацій, власні значення необхідно обчислювати лише раз. Однак, обчислення на другому кроці повинні бути зроблені окремо для кожної реалізації. Через це, кількість обчислень залишається порядку $N \cdot \log(N)$.

2. Наближені методи

2.1. Метод стохастичного подання фрактального Броунівського руху

Як було показано (2.29), фрактальний Броунівський рух визначається стохастичним інтегралом відносно звичайного Броунівського руху. Природною ідеєю є наближення цього інтеграла сумами Риманового типу для апроксимованого моделювання процесу.

При наближенні (2.29) за допомогою сум перший інтеграл повинен бути урізаний, скажімо, до $-b$. Наближення $\widetilde{B}_H(n)$ при $n = 1,\ldots,N$ обчислюється як

$$\tilde{B}_H(n) = C_H\left(\sum_{k=-b}^{0}\left[(n-k)^{H-1/2} - (-k)^{H-1/2}\right]B_1(k) + \sum_{k=0}^{n}(n-k)^{H-1/2}B_2(k)\right), \quad (3.28)$$

де B_1 й B_2 – вектори, що складаються відповідно з $b+1$ й $N+1$ нормованих нормально розподілених величин, взаємно незалежних.

C – стала, не рівна константі в (2.29), оскільки ми розглядаємо нормований фрактальний Броунівський рух.

Наближення може бути зроблене точніше за допомогою збільшення параметра урізування b і вибору кращої сітки для сум Римана. В [55] запропоновано обрати $b = N^{3/2}$. Цей метод цікавий тільки з історичної точки зору і вважається поганим способом генерування фрактального Броунівського руху.

2. 2. Агрегування процесів джерел (*ON*-/*OFF*- метод)

Як вказувалося в розділі 2.3.5, походження самоподоби пояснюється на рівні джерел РВХ пакетних процесів (періодів активності) індивідуальних пар джерело-одержувач. Таким чином, можна здійснити їхнє моделювання в мікро-масштабі та агрегувати, щоб отримати самоподібний трафік. Цей метод різним чином описується в різних джерелах. Ми будемо слідувати [56].

Припустимо, є S н.о.р. джерел. Кожне джерело s має активні й неактивні періоди, які моделюються стаціонарними двійковими часовими рядами: $\{W^{(s)}(t): t \geq 0\}$. $W^{(s)}(t) = 1$ означає, що джерело пересилає пакет у момент часу t і в протилежному випадку $W^{(s)}(t) = 0$. Довжини активних ("*ON*-") періодів є н.о.р., неактивних періодів ("*OFF*-") є н.о. р., і довжини *ON*- і *OFF*- періодів незалежні. *OFF*- період слідує за *ON*- періодом і розподіли довжин *ON*- і *OFF*- періодів можуть бути різними.

Змінюючи масштаб часу в T разів, нехай

$$W_S(Tt) = \int_0^{Tt} \left(\sum_{s=1}^{S} W^{(S)}(u) \right) du \qquad (3.29)$$

буде агрегованою кумулятивною кількістю пакетів за інтервал $[0; T\,t]$.

Для спрощення, нехай розподіли *ON*- і *OFF*- періодів будуть РВХ зі спільним параметром $1 < \alpha < 2$. Нагадаємо, що *ON*- і *OFF*- періоди мають нескінченну дисперсію, оскільки $1 < \alpha < 2$.

Як зазначено в [56], що наступний ліміт відповідає $\{W_S(Tt) : 0 \le t < \infty\}$:

$$\lim_{T\to\infty} \lim_{S\to\infty} T^{-H} S^{-1/2} \left(W_S(Tt) - \frac{1}{2} TSt \right) = \sigma B_H(t) \qquad (3.30)$$

де $H = (3-\alpha)/2$,

$\sigma > 0$ – деяка константа,

$B_H(t)$ – фрактальний Броунівський рух з параметром Херста Н.

Потрібно обережно інтерпретувати цей асимптотичний результат, тому що (3.30) не виконується, коли порядок узяття лімітів змінений.

Згаданий результат може використовуватися для моделювання, шляхом агрегування великої кількості джерел з ON- і OFF- періодами, що підпорядковуються розподілу Парето (або будь-якому іншому РВХ). Це алгоритм складності порядку N, коли необхідна вибірка розміру N. Однак, кількість джерел, які доведеться агрегувати, щоб одержати прийнятний результат, може бути дуже великою, що може призвести до серйозного сповільнення. Більш того, досить складно визначити, скільки джерел необхідно для одержання прийнятних результатів. Фактично, через згадану проблему з перестановкою лімітів, доводиться дуже підозріло ставиться до отриманої вибірки.

Інший метод агрегації для моделювання процесів з ПСЗ використовує модель черги $M/G/\infty$, у якій заявки користувачів надходять згідно Пуасонівського процесу а інтервали часу їхнього обслуговування підпорядковуються розподілам з нескінченною дисперсією. Якщо X_t

позначає кількість заявок у системі в момент часу t, то $\{X_t : t \geq 0\}$ є асимптотично самоподібним [8]. Знову таки, необхідно визначати тривалість обчислень для одержання припустимої апроксимації.

2.3. Випадкове зміщення проміжних точок і вдалий випадковий приріст

Методи Хоскінга та Холецького рекурсивно генерують вибірку фрактального Гаусового шуму. Замість генерування X_{n+1} по всьому минулому ряду, генерування може бути прискорене за допомогою генерування X_{n+1} за останніми m згенерованими значеннями для якогось $m \geq 1$. Однак, при цьому зникають кореляції з більшим запізненням, що робить такий підхід не придатним для генерування процесів з ПСЗ.

Метод випадкового зміщення проміжних точок (ВЗПТ) [9] генерує аппроксимовано самоподібні вибірки в інтервалі [0;1]. Відзначимо, що ця вибірка може масштабуватися на інтервал будь-якої бажаної довжини, використовуючи властивість самоподоби.

Ідея, що стоїть за методом ВЗПТ полягає в рекурсивній видозміні інтервалу шляхом вставки деякого значення в середній точці, обчисленого за значеннями в крайніх так, що зберігається деяка ПСЗ. Інтервал між 0 і 1 послідовно ділиться навпіл, і до середніх точок кожного інтервалу додаються зміщення (масштабовані відповідним чином на кожній ітерації), що підпорядковуються нормальному розподілу.

На рис. 3.1 представлені перші три кроки процесу. Розглянемо алгоритм методу покроково.

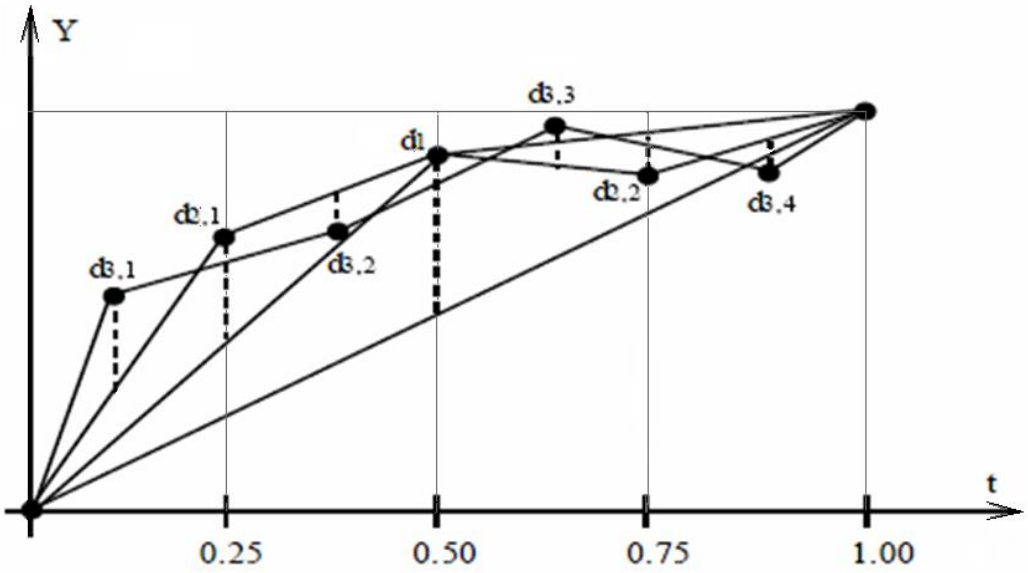


Рис. 3.1. Перші три кроки методу ВЗПТ

Крок 0. Нехай необхідно сконструювати реалізацію самоподібного процесу на інтервалі (0, 1) із заданою дисперсією σ_0^2 та параметром Херста H. Початковій точці присвоється значення $Y_0 = 0$, а кінцевій Y_1 – псевдовипадкове число з нормального розподілу з нульовим середнім і дисперсією $D[Y_1] = \sigma_0^2$. Тоді $D[Y_1 - Y_0] = \sigma_0^2$.

Крок 1. Значення $Y_{\frac{1}{2}}$ визначається як середнє між Y_0 й Y_1 з деяким зміщенням, тобто

$$Y_{\frac{1}{2}} = \frac{1}{2}(Y_0 + Y_1) + d_1 \qquad (3.31)$$

де d_1 – зміщення, випадкове число з нормального розподілу з нульовим середнім і дисперсією S_1^2.

Дисперсію S_1^2 обираємо так, щоб виконувалося $Var[Y_{t_2} - Y_{t_1}] = |t_2 - t_1|^{2H} \sigma_0^2$ для $0 \le t_1 \le t_2 \le 1$, тобто

$$Var\left[Y_{\frac{1}{2}} - Y_o\right] = \frac{1}{4} Var[Y_1 - Y_0] + S_1^2, \qquad (3.32)$$

$$(1/2)^{2H}\sigma_0^2 = 1/4\sigma_0^2 + S_1^2 \tag{3.33}$$

Тому

$$S_1^2 = \left(\tfrac{1}{2^1}\right)^{2H}(1-2^{2H-2})\sigma_0^2. \tag{3.34}$$

Крок 2. Розділимо два інтервали від 0 до $\frac{1}{2}$ й від $\frac{1}{2}$ до 1 навпіл. Зменшуємо коефіцієнт зміщення, і знову додаємо зміщення до середніх точок кожного інтервалу; $Y_{\frac{1}{4}}$ визначаємо як середнє $\frac{1}{2}(Y_0 + Y_{\frac{1}{2}})$ плюс зміщення $d_{2,1}$. Відповідно $Y_{\frac{3}{4}}$ визначається як

$$Y_{\frac{3}{4}} = \frac{1}{2}(Y_{\frac{1}{2}} + Y_1) + d_{2,2} \tag{3.35}$$

де $d_{2,2}$ – випадкове зміщення, обчислене аналогічно.

Аналогічно, $d_{2,*}$ повинне мати дисперсію S_2^2, таку що $Var\left[Y_{\frac{1}{4}} - Y_o\right] = \frac{1}{4}Var\left[Y_{\frac{1}{2}} - Y_0\right] + S_2^2$. $\left(\frac{1}{2^2}\right)^{2H}\sigma_0^2 = \frac{1}{4}\left(\frac{1}{2}\right)^{2H}\sigma_0^2 + S_2^2$. Тому

$$S_2^2 = \left(\tfrac{1}{2^2}\right)^{2H}(1-2^{2H-2})\sigma_0^2. \tag{3.36}$$

Крок 3. Продовжуємо ділення відрізків навпіл і аналогічно обчислюємо їхні середні точки:

$$\begin{aligned} Y_{\frac{1}{8}} &= \tfrac{1}{2}(Y_0 + Y_{\frac{1}{4}}) + d_{3,1}, \\ Y_{\frac{3}{8}} &= \tfrac{1}{2}(Y_{\frac{1}{4}} + Y_{\frac{1}{2}}) + d_{3,2}, \\ Y_{\frac{5}{8}} &= \tfrac{1}{2}(Y_{\frac{1}{2}} + Y_{\frac{3}{4}}) + d_{3,3}, \\ Y_{\frac{7}{8}} &= \tfrac{1}{2}(Y_{\frac{3}{4}} + Y_1) + d_{3,4}. \end{aligned} \tag{3.37}$$

Кожне зі зміщень генерується окремо і дисперсія на цьому кроці

$$S_3^2 = \left(\frac{1}{2^3}\right)^{2H}(1-2^{2H-2})\sigma_0^2. \quad (3.38)$$

Крок 4. Продовжуємо обчислення в середніх точках згідно тому ж алгоритму поки не досягнемо

$$2^k = N \quad (3.39)$$

де k – кількість зроблених ітерацій,

N – необхідна кількість точок вибірки.

Дисперсія зміщень на k-й ітерації становить

$$S_k^2 = \left(\frac{1}{2^k}\right)^{2H}(1-2^{2H-2})\sigma_0^2. \quad (3.40)$$

Після цього сформуємо підсумкову самоподібну послідовність $X_0, X_1, \ldots, X_N$, перейменувавши $X_i = Y_{i/2^k}$ для $i = 0,1,\ldots,N$.

Відмінність алгоритму вдалого випадкового приросту (ВВП) від ВЗПТ полягає в наступному. У ВЗПТ випадкові прирости додаються до точок тільки один раз, у процесі формування точки (значення кожної точки дорівнює "інтерпольоване значення + випадкове зміщення"). На відміну від нього, у ВВП випадковий приріст додається до всіх вже сформованих точок. Таким чином, кожна точка формується в результаті підсумовування декількох випадкових зміщень, залежно від кроку рекурсії. Відповідно дисперсія для випадкового зсуву k-го кроку алгоритму буде розраховуватися трохи інакше [57].

Крок 1. Аналогічний алгоритму ВЗПТ

Крок 2. Значення $Y_{\frac{1}{2}}$ визначається як середнє між Y_0 й Y_1

$$Y_{\frac{1}{2}} = \frac{1}{2}(Y_0 + Y_1)$$

Крок 3. Додаємо до всіх вже розрахованих точок, тобто $Y_0, Y_{\frac{1}{2}}, Y_1$ випадковий приріст з дисперсією S_1^2 обраною так, щоб виконувалося $Var\left[Y_{t_2} - Y_{t_1}\right] = \left|t_2 - t_1\right|^{2H} \sigma_0^2$ для будь-яких $0 \le t_1 \le t_2 \le 1,$. У той же час, відповідно до алгоритму обчислень, дисперсія буде складатися з дисперсії інтерпольованої точки і подвоєної дисперсії приросту S_1^2:

$$Var\left[Y_{\frac{1}{2}} - Y_o\right] = \frac{1}{4} Var\left[Y_1 - Y_0\right] + 2S_1^2$$

Маємо:

$$S_1^2 = \frac{1}{2}\left(\frac{1}{2^1}\right)^{2H} (1 - 2^{2H-2})\sigma_0^2.$$

Крок 4. Рекурсивно повторюємо кроки 2 й 3 для кожного з підінтервалів, отриманих на попередньому кроці до одержання значень всіх точок шуканої послідовності, тобто до досягнення $2^k = N$. Для обчислення дисперсії випадкового приросту S_k^2, що використовується на *k-м* кроці рекурсії, застосовуємо узагальнення результатів, отриманих на кроці 3:

$$S_k^2 = \frac{1}{2}\left(\frac{1}{2^k}\right)^{2H} (1 - 2^{2H-2})\sigma_0^2.$$

Перевірка, проведена для генерованих за допомогою цих алгоритмів випадкових послідовностей, показує, що метод ВЗПТ дає більш точну відповідність самоподобі, ніж ВВП.

Зрозуміло, що алгоритми, викладені вище, не підходять для генерування вибірки «на льоту», оскільки період часу моделювання повинен бути відомий заздалегідь. Однак, не обов'язково генерувати вибірку точно в такому порядку, як описано вище. Тобто, замість генерування кожного дозволу повністю перед переходом до кращого, ми можемо залишити деякі дозволи незакінченими. Детально це питання розглядається в [58].

Складність цього алгоритму порядку N, якщо необхідна вибірка розміру N. Відзначимо, що це швидше, ніж найшвидший точний метод.

Таким чином, алгоритм ВЗПТ надзвичайно цікавий для моделювання. Головне – впевнитися, чи розумний зроблений компроміс між швидкістю і точністю.

2.4. Спектральне моделювання, метод Паксона і наближений циклічний метод

Спектральне моделювання – це метод моделювання стаціонарного Гаусового процесу з використанням спектрального аналізу [59]. При цьому важливу роль грає Швидке Перетворення Фур'є (ШПФ). Спершу, розглянемо основні принципи спектрального моделювання. Основна ідея полягає в моделюванні процесу в частотній області і подальшому перетворенні отриманої послідовності в часову область. Незважаючи на те, що при використанні даного підходу неможливо одержати вибірку, що є у точності самоподібною, незабаром ми побачимо, що відповідність зростає при зростанні величини вибірки.

У ході опису стане ясно, що спектральне моделювання тісно пов'язане з методом Паксона (*Paxson*) і методом Девіса і Харта. Остаточна модифікація методу Девіса і Харта (що робить метод наближеним для прискорення моделювання) називається наближеним циклічним методом.

2.4.1. Спектральне моделювання

Однозначний спектральний аналіз стохастичних процесів можливий лише тоді, коли ці процеси стаціонарні. У цьому розділі ми припускаємо, що необхідна довжина N вибірки є степінем двох.

Спектральний аналіз стаціонарного дискретного за часом Гаусового процесу $X = \{X_n : n = 0, \ldots, N-1\}$ показує, що він може бути представлений на основі спектральної щільності як

$$X_n = \int_0^{\pi} \sqrt{\frac{f(\lambda)}{\pi}} \cos(n\lambda) dB_1(\lambda) - \int_0^{\pi} \sqrt{\frac{f(\lambda)}{\pi}} \sin(n\lambda) dB_2(\lambda), \qquad (3.41)$$

де рівність розуміється як рівність розподілів.

Два інтегратори є взаємно незалежними (звичайними) Броунівськими рухами.

Спектральне моделювання засноване на апроксимуванні виразу (3.41). При цьому підінтегральний вираз заміняється простою функцією в такий спосіб. Позначимо $\xi_n(\lambda) = \sqrt{f(\lambda)/\pi}\cos(n\lambda)$ і визначимо деяке ціле число ℓ. Після цього позначимо $t_k = \pi k/\ell$ для $k = 0,1\ldots,\ell-1$ і визначимо просту функцію $\xi_n^{(\ell)}(\lambda)$ на інтервалі $[0,\pi]$ для $0 \le n \le N-1$ як

$$\begin{aligned} \xi_n^{(\ell)}(\lambda) &= \sqrt{f(t_1)/\pi}\cos(nt_1)1_{\{0\}}(\lambda) + \\ &+ \sum_{k=0}^{\ell-1} \sqrt{f(t_{k+1})/\pi}\cos(nt_{k+1})1_{(t_k,t_{k+1}]}(\lambda). \end{aligned} \qquad (3.42)$$

Нехай $\theta_n^{(\ell)}(\lambda)$ буде визначена так само як і $\xi_n^{(\ell)}(\lambda)$, тільки із синусами замість косинусів. Перший інтеграл в (3.41) апроксимується $\int_0^{\pi} \xi_n^{(\ell)}(\lambda) dB_1(\lambda)$, що може бути обчисленим за допомогою наступного виразу [48]:

$$\int_0^{\pi} \phi(s) dB(s) = \sum_{i=0}^{\ell-1} \phi_i \left(B(t_{i+1}) - B(t_i)\right) \qquad (3.43)$$

де $\phi(s)$ – проста функція (як $\xi_n^{(\ell)}(\lambda)$ або $\theta_n^{(\ell)}(\lambda)$)

Оскільки така ж апроксимація може бути застосована і до другого інтеграла, ми приходимо до наступної апроксимації X_n:

$$\hat{X}_n^{(\ell)} = \sum_{k=0}^{\ell-1} \sqrt{f(t_{r+1})/\ell}\left[\cos(nt_{k+1})U_k^0 - \sin(nt_{k+1})U_k^{(1)}\right] \qquad (3.44)$$

де $U_k^{(i)}$ знову н.о.р. стандартні нормальні випадкові величини, при $k = 0,1,\ldots\ell-1$.

Два вектори U^0 й U^1 також повинні бути взаємно незалежними, оскільки незалежні B_1 і B_2.

Слід зазначити, що при $H > 1/2$ ми апроксимуємо спектральну щільність функціями, що не мають полюса в 0. Як уже вказувалося вище, цей полюс еквівалентний ПСЗ. Отже, ми апроксимуємо процес із ПСЗ процесом зі ШСЗ. Незважаючи на це, ми можемо одержати вибірку зі структурою коваріації, що дуже добре наближається до самоподібних процесів.

Для ефективного обчислення (3.44) може бути використане ШПФ. Для цієї мети визначимо послідовність $(a_k)_{k=0,\ldots,2\ell-1}$:

$$a_k = \begin{cases} 0 & k = 0; \\ \frac{1}{2}\left(U_{k-1}^{(0)} + jU_{k-1}^{(1)}\right)\sqrt{f(t_k)/\ell} & k = 1,\ldots,\ell-1; \\ U_{k-1}^{(0)}\sqrt{f(t_k)/\ell} & k = \ell; \\ \frac{1}{2}\left(U_{2\ell-k-1}^{(0)} - jU_{2\ell-k-1}^{(1)}\right)\sqrt{f(t_{2\ell-k})/\ell} & k = \ell+1,\ldots,2\ell-1 \end{cases} \qquad (3.45)$$

Використовуючи (3.23), можна перевірити, що перетворення Фур'є послідовності (a_k) є дійсним та відповідає (3.44).

Оскільки $\xi_n^{(t)}$ апроксимує ξ_n краще при великих ℓ, цікавим є питання, чи сходиться апроксимована вибірка до точної при $\ell \to \infty$. Однак існують різні типи сходження випадкових величин і процесів. Це питання докладно розглянуте в [48].

2.4.2. Метод Паксона

У роботі [60] Паксон запропонував більш наочний метод моделювання фрактального Гаусового шуму. Статистично вивчаючи отримані вибірки, він перевіряє, чи дійсно вони мають бажані властивості. На жаль, у публікації відсутнє вичерпне обґрунтування запропонованої процедури, і залишається не зрозумілим, чому отримані вибірки повинні бути (приблизно) Гаусовими.

Однак, використовуючи раніше описані формули, можна зробити обґрунтування більш точними. У методі Паксона апроксимована вибірка ФГШ отримується перетворенням Фур'є такої послідовності

$$b_k = \begin{cases} 0 & k = 0; \\ \sqrt{\dfrac{R_k f(t_k)}{N}} \exp(\hat{i}\Phi_k) & k = 1,\ldots,N/2; \\ b^*_{N-k} & k = N/2+1,\ldots,N-1 \end{cases} \qquad (3.46)$$

де R_k – незалежні експоненціально розподілені випадкові змінні із середнім, рівним 1; $k \geq 1$; «зірочка» позначає комплексне сполучення.

За винятком $\Phi_{N/2}$, рівного 0, Φ_k це незалежні, рівномірно розподілені випадкові змінні в діапазоні $[0, 2\pi]$, $k \geq 1$, також незалежні від R_k. Тоді t_k рівні $2\pi k/N$. Відзначимо, що отримана вибірка дійсна за побудовою.

Оскільки R_k експоненціально розподілені із середнім, рівним 1, $\sqrt{2R_k}\exp(\hat{i}\Phi_k)$, розподілені так само, як і $U_k^{(0)} + \hat{i}U_k^{(1)}$, де $U_k^{(0)}$ і $U_k^{(1)}$ – незалежні стандартні нормальні випадкові змінні. Цей факт також використовується в добре відомому алгоритмі Бокса-Мюллера для моделювання Гаусових випадкових змінних.

Порівняємо це зі спектральним методом моделювання при $\ell = N/2$. У цьому випадку вибірка отримується шляхом перетворення Фур'є послідовності

$$b'_k = \begin{cases} 0 & k = 0; \\ \sqrt{\dfrac{2f(t_k)}{2N}}(U_{k-1}^{(0)} + iU_{k-1}^{(1)}) & k = 1,\ldots,N/2-1; \\ \sqrt{\dfrac{2f(t_k)}{N}}U_{k-1}^{(0)} & k = N/2; \\ \sqrt{\dfrac{2f(t_k)}{2N}}(U_{N-k-1}^{(0)} - iU_{N-k-1}^{(1)}) & k = N/2+1,\ldots,N-1, \end{cases} \qquad (3.47)$$

де $U_k^{(i)}$ має своє звичайне значення.

Зараз стало очевидно, що існує зв'язок між спектральним методом і методом Паксона. Порівнявши b_k і b'_k, видно, що єдина різниця полягає в значенні $b_{N/2}$ (звичайно, індексування випадкових змінних $U^{(i)}$ відрізняється, але це неістотно). Однак, цей ефект зникне при «великих» N. Тому, точність методу Паксона зростає зі збільшенням N, і він стає навіть точним при $N \to \infty$ в розумінні, що кожна точка вибірки прямує до точки вибірки, отриманої за допомогою точного методу.

Зараз ми можемо аргументувати нераціональність пропозиції Паксона прирівняти $\Phi_{N/2}$ до нуля, оскільки це руйнує нормальність в $b_{N/2}$, що відбивається в *кожній* точці вибірки. Із цієї причини метод Паксона в деяких випадках генерує вибірки, точки яких мають «майже» нормальні розподіли. Тому визначимо

$$b_{N/2} = \sqrt{\frac{f(t_{N/2})}{2N}} U^{(0)}_{N/2} \tag{3.48}$$

Будемо називати це поліпшеним методом Паксона.

Цей метод швидше методу Девіса і Харта, оскільки він вимагає всього одного перетворення Фур'є послідовності довжиною N, замість двох перетворень послідовності довжиною $2N$, однак він усе ще порядку $N\log(N)$. Отже, метод Паксона приблизно в 4 рази швидше точного методу Девіса і Харта, за умови що N є великим. Тому він може бути гарною альтернативою точним методам, коли необхідні великі вибірки.

Вибірки ФБР, створені за вибірками ФГШ Паксона мають особливість: кінцева точка завжди 0. Цю властивість легко побачити, використовуючи визначення ШПФ (3.23):

$$\sum_{n=0}^{N-1}\sum_{k=0}^{N-1} b_k \exp\left(2\pi\hat{i}\,\frac{nk}{N}\right) = \sum_{k=0}^{N-1} b_k \sum_{n=0}^{N-1} \exp\left(2\pi\hat{i}\,\frac{nk}{N}\right) = N_{b_0} = 0\,. \tag{3.49}$$

Ця вкрай небажана властивість вибірок ФБР Паксона може розглядатися як фрактальний Броунівський мостовий ефект. Упізнавши у методі Паксона метод спектрального моделювання при $\ell = N/2$, можна поставити запитання, що можна сказати про спектральне моделювання при $\ell = N$. Наступний підрозділ показує, що тут існує зв'язок з методом Девіса і Харта.

2.4.3. Наближений циклічний метод

Метод спектрального моделювання пов'язаний із циклічним діагональним алгоритмом Девіса і Харта. У цьому алгоритмі, λ_k обчислюються за (3.23), що можна переписати як

$$\lambda_k = \sum_{j=0}^{N-1} \gamma(j) \exp\left(2\pi i \frac{jk}{2N}\right) + \sum_{j=1}^{N-1} \gamma(j) \exp\left(2\pi i \frac{(2N-j)k}{2N}\right) = \sum_{j=-N+1}^{N-1} \gamma(j) \exp\left(2\pi i \frac{jk}{2N}\right). \qquad (3.50)$$

Ввівши позначення $t_k = \pi k / N$, результат виразу (3.50) можна апроксимувати нескінченною сумою $\sum_{j=-\infty}^{\infty} \gamma(j) \exp(ijt_k)$ за умови «великих» N. Відзначимо, що це саме спектральна щільність (1.9). У такому випадку, λ_k можна апроксимувати при $k = 0, 1, \ldots, N$ через функцію $f(t_k)$, а при $k = n+1, \ldots, 2N-1$ через $f(t_{2N-k})$. Щоб уникнути проблем з полюсами, $f(0)$ повинна бути апроксимована кінцевим значенням. Оскільки обчислити λ_0 можна безпосередньо з (3.50), то вибір $f(0) = \lambda_0 = N^{2H} - (N-1)^{2H}$ цілком обґрунтований.

Замість використання точного λ_k, ми досліджуємо, що станеться, якщо в методі Девіса і Харта використати апроксимації $f(t_k)$. Застосовуємо ШПФ (див. 3.27)

$$c_k = \begin{cases} \sqrt{\dfrac{f(t_k)}{2N}} U_k^{(0)} & k = 0; \\ \sqrt{\dfrac{f(t_k)}{4N}} (U_k^{(0)} + \hat{i} U_k^{(1)}) & k = 1, \ldots, N-1; \\ \sqrt{\dfrac{f(t_k)}{2N}} U_k^{(0)} & k = N; \\ \sqrt{\dfrac{f(t_{2N-k})}{4N}} (U_{2N-k}^{(0)} - \hat{i} U_{2N-k}^{(1)}) & k = N+1, \ldots, 2N-1. \end{cases} \tag{3.51}$$

Перші N коефіцієнтів перетворення Фур'є є вибіркою апроксимованого ФГШ. Єдиною відмінністю від методу Девіса і Харта (який ґрунтується на циклічній матриці) є використання апроксимованих λ_k, і ми будемо називати його наближеним циклічним методом. Вхідні коефіцієнти c_k дуже схожі з (3.45) за умови $\ell = N$. Відмінність проявляється також у коефіцієнтах при k = 0 і $k = N$. Знову таки, ефект цих відмінностей зникає при зростанні розмірів вибірки N, що говорить про те, що метод асимптотично точний у розумінні, описаному раніше.

Більш того, зараз ми можемо оцінити, як працює метод спектрального моделювання при $\ell = N$ в спектральній області при скінченних N: можна підрахувати, яка «спектральна» помилка відбувається при використанні спектральної щільності (1.9) замість λ_k.

Апроксимуючи λ_k значенням спектральної щільності в t_k, кількість застосувань ШПФ послідовності довжиною $2N$ скорочується вдвічі. Замість цього, спектральна щільність обчислюється в $2N$ точках; тобто складність методу порядку N. Тому, метод спектральної симуляції при $\ell = 2N$ теоретично вдвічі швидше, ніж точний метод Девіса і Харта. Однак на практиці цей метод працює лише трохи швидше ніж точний при розумних довжинах вибірки.

2.5. Моделювання, засноване на вейвлетах

Підхід, тісно пов'язаний зі спектральним моделюванням – це моделювання, засноване на вейвлетах [61]. Як і у спектральному моделюванні, ідея полягає в тому, щоб генерувати процес, трансформована версія якого може розглядатися як вибірка, наближена до самоподібної. Однак, це не єдина подібність спектрального і вейвлетного моделювання. В обох випадках, частотна інформація зберігається в коефіцієнтах. Невеликий приклад роз'яснить розбіжності. Припустимо, що ми маємо вибірку лише з однією частотою (наприклад, масштабований синус або косинус). Його перетворення Фур'є дає один пік на цій конкретній частоті. Створимо нову вибірку вдвічі більших розмірів, таку, що перша її частина ідентична оригінальній вибірці, а друга частина - константа (наприклад, нуль). Перетворення Фур'є цієї нової вибірки знову дає пік на первісній частоті, однак у спектрі присутні й інші частоти в невеликих потужностях, що відповідають другій половині вибірки. Тому, незважаючи на те, що коефіцієнти в спектральному методі (тобто, перетворення Фур'є вибірки у часовій області) унікально визначають вибірку у часовій області, нетрансформовані коефіцієнти не дають інформації про знаходження цих частот у часі. До того ж пік у сигналі в часовій області поширюється на всю частотну область його перетворення Фур'є, тобто ще одним його недоліком є «глобальна» чутливість до «локальних» скачків і різких піків амплітуди сигналу. При аналізі ж сигналів непостійного, локального характеру, часто вигідно визначити кореляцію між часом і спектром сигналу. Тому виникає задача – як зв'язати особливості спектра та характер зміни сигналу в часі.

При дослідженні нестаціонарних сигналів, що не мають чіткого періодичного характеру, найбільш ефективним було б використання не тригонометричних, а деяких локалізованих у часі компактних базисів, коефіцієнти розкладання за якими зберігають інформацію про зміну параметрів сигналу, що апроксимується. Спочатку спроби побудови таких базисів функцій зводилися до сегментування сигналу на фрагменти (вікна) із застосуванням для них розкладання Фур'є. Відповідне перетворення – віконне

перетворення Фур'є - було запропоновано ще в 1946...1947 роках, але воно не знайшло широкого застосування.

Інженер-геофізик Дж. Морлет (*J. Morlet*) наприкінці 70-х років XX в. зіштовхнувся із проблемою аналізу сигналів сейсмодатчиків, які містили високочастотний компонент (сейсмічна активність) протягом короткого проміжку часу та низькочастотні складові (спокійний стан земної кори) – протягом тривалого періоду. Віконні перетворення дозволяли аналізувати або високі частоти, або низькочастотний компонент у вузькому інтервалі (вікні) часу, але не обидві складові відразу. Тому був запропонований метод аналізу, у якому для різних сигналів використовувалися часові вікна різної тривалості.

Віконні перетворення отримувалися в результаті стиснення-розтягнення і зсуву у часі однієї породжуючої (названної скейлінг-функцією – *scaling function, scalet*) функції – гаусіана (гаусів імпульс). Ці базисні функції були названі вейвлетами – (англійське слово *wavelet* – вейвлет – що є перекладом французького «*ondelette*», можна формально перекласти українською мовою як словосполучення «маленька хвиля», «невелике коливання», «компактна хвиля», «невеликі хвилі, що слідують одна за одною») – однакові, але рознесені у часі та що мають до того ж найрізноманітніші розтягнуті або стиснені копії.

На базі такого підходу виник цілий напрямок у теорії і техніці сигналів, що отримав назву вейвлет-аналіз. На відміну від традиційного перетворення Фур'є, вейвлет-аналіз має базис функцій, локалізованих і за часом t, і за частотою f. Тому сигнал можна аналізувати одночасно у фізичному (час, координата) і в частотному просторах. Вейвлет-аналіз являє собою ніби безперервний континіум локальних перетворень Фур'є з різними вікнами для кожної частоти. Власне кажучи вейвлет-аналіз сигналів є проміжним між повністю спектральним і повністю часовим поданнями і відбиває ідею багатомасштабного аналізу сигналів. Умовно спосіб багатомасштабного, мультирезолюційного (англійські слова – *multiscale* і *multiresolution*), полягає в тому, щоб подивитись на досліджуваний об'єкт (у цьому випадку сигнал)

спочатку за допомогою мікроскопа, потім – через лупу, потім подивитися, відійшовши на пару кроків, і, нарешті, оцінити здалеку. Дивлячись у бінокль, мікроскоп або кінокамеру і намагаючись уловити максимально чітке і велике зображення об'єктів, що цікавлять, у багатьох інших випадках спостерігач, не усвідомлюючи цього, проводить подібний аналіз.

Сигнал, представлений у системі базисних функцій вейвлета, вже можна записати у вигляді узагальненого ряду Фур'є. Якщо сигнал на досить великому інтервалі часу не має різких стрибків, то вейвлет розтягують по осі t і на ньому проводять аналіз; при цьому істотно заощаджуються ресурси обробки. При наявності різкого короткочасного викиду сигналу вейвлет стискується до інтервалу цього викиду, і аналіз проводиться на даному короткому інтервалі часу. У можливості використання різномасштабного аналізу й криється великий виграш у представленні сигналу за допомогою вейвлетів.

Коефіцієнти такого розкладання несуть найважливішу інформацію про еволюцію сигналу в частотній і часовій областях. Для кожного конкретного практичного завдання можна підібрати найбільш ефективний вейвлет.

Більше того, на практиці бажано мати більшу розподільну здатність за часом на високих частотах, ніж на низьких. Вейвлетні коефіцієнти мають цю властивість. Тому, замість моделювання перетворення Фур'є, можна моделювати вейвлетні коефіцієнти вибірки самоподібного процесу. Вибірку в часовій області можна потім отримати шляхом трансформування вейвлетних коефіцієнтів.

Перейдемо до опису методу, почавши з короткого опису основ вейвлетів і мультирезолюційного аналізу. Більш докладно вейвлети описані в [62].

Позначимо простір всіх квадратично-інтегрованих функцій (у змісті Лебезге) в R як $L^2(R)$. Елементи цих просторів найкраще розуміти як класи функцій. Дві функції в цьому просторі одного класу, якщо вони майже скрізь рівні, тобто рівні аж до зліченної кількості точок. Для прикладу, 0 позначає функції, майже скрізь рівні нулю. Скалярний добуток у цьому просторі визначається як

$$\langle f, g \rangle = \int_R f(t) g(t) dt, \tag{3.52}$$

для $f, g \in L^2(R)$,

і (псевдо-)норма як $\|f\|^2 = \langle f, f \rangle$, що розуміється як L^2-норма в R

Сім'я функцій $\{f_k : k \in K\}$ називається ортонормованою, якщо $\langle f_i, f_j \rangle = 1(i = j)$ для всіх $i, j \in K$. Вона називається базисом для деякої підмножини A з $L^2(R)$, якщо будь-який елемент $f \in A$ може бути унікально записаний як

$$f = \sum_{k \in K} \alpha_k f_k \tag{3.53}$$

при деяких α_k.

Якщо K нескінченна, але зліченна, наприклад, $K = Z$, то $f = \sum_{k \in Z} \alpha_k f_k$ є єдиною функцією, для якої виконується $\lim_{N \to \infty} \left\| f - \sum_{k=-N}^{N} \alpha_k f_k \right\| = 0$. У цьому розділі ортонормований базис буде відігравати дуже важливу роль.

Як було сказано вище, на відміну від спектрального моделювання, де використовується спектральний аналіз, при вейвлетному аналізі використовується аналіз зі змінною розподільною здатністю – мультирезолюційний аналіз (МРА) (*multiresolutional analysis*). Цей аналіз складається із сукупності вкладених підпросторів $\{V_j : j \in Z\}$ і функції ϕ. Кожен простір V_j є підпростором $L^2(R)$. Підпростори задовольняють наступним властивостям:

1. $V_j \subset V_{j-1}$ *для* $j \in Z$;
2. $\bigcap_{j \in Z} V_j = \{0\}$;

3. Будь-яка функція $f \in L^2(R)$ може бути апроксимована функціями із V_j із будь-якою точністю, тобто при заданій $f \in L^2(R)$ існує послідовність $\{g_\ell\} \subset \bigcup_{j \in Z} V_j$, така, що $\lim_{\ell \to -\infty} \|g_\ell - f\| = 0$

4. $x(t) \in V_j$ тоді і тільки тоді, коли $x(2^j t) \in V_0$;

5. $\phi \in V_0$ і $\{\phi(t-k) : k \in Z\}$ – ортонормований базис для V_0.

Функція ϕ в п'ятій властивості називається *масштабуючою функцією.*

Проведення мультирезолюційного аналізу сигналу x означає його успішне проектування на кожний з підпросторів V_j. Проектування x на V_j називається апроксимацією на рівні j. Можна перевірити, що $\{\phi_{j,k}(t) = 2^{-j/2}\phi(2^{-j}t - k) : k \in Z\}$ є ортогональним базисом для V_j. Оскільки $V_j \subset V_{j-1}$, апроксимація на рівні j грубіша, ніж на рівні j - 1. Ключова ідея МРА полягає в дослідженні втрат інформації, названих деталізації, при переході з однієї апроксимації на більш грубу. Ця деталізація може бути отримана відніманням двох сусідніх апроксимацій однієї з іншої, а може бути прямо обчислена, тому що V_j і $W_j = V_{j-1} \setminus V_j$ є ортогональними просторами (тобто будь-яка функція в V_j є ортогональною будь-якій функції з W_j). З теорії про простір скалярних добутків випливає, що деталізація рівня j знаходиться при проектуванні x на W_j. Більше того, теорія МРА показує, що існує функція ψ, називана *материнським вейвлетом (mother wavelet),* яку можна одержати узяттям похідної від ϕ, така що $\{\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k) : k \in Z\}$ є ортогональним базисом для W_j.

Щоб проілюструвати ідею МРА, множини V_1, V_0 *и* V_{-1} зображені на діаграмі Віна на рис. 3.2. Можливі елементи цих просторів також зображені. Затінена множина означає W_1; її елементи ортогональні до V_1.

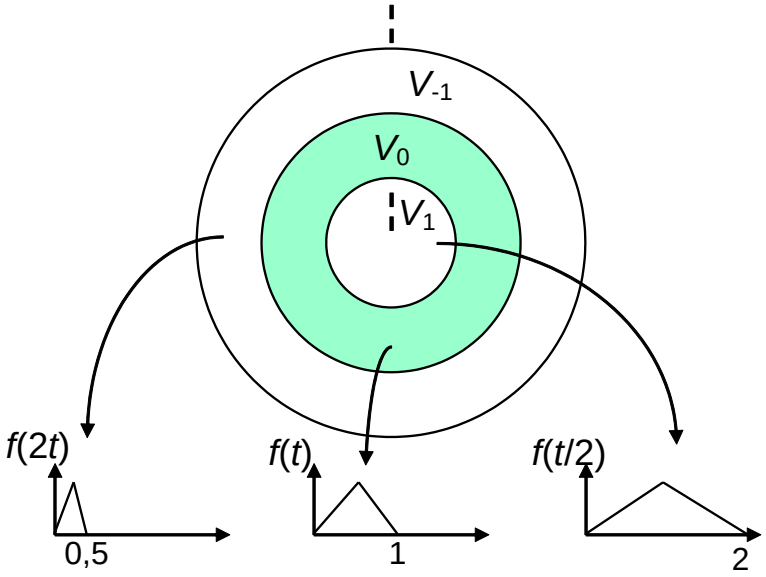


Рис. 3.2 Ілюстрація мультирезолюційного аналізу

Материнський вейвлет, сгенерований МРА, повинен задовольняти умові $\int \psi(t)dt = 0$ через деякі нюанси систематичності. Деякі вейвлети задовольняють навіть $\int t^k \psi(t)dt = 0$ при $k = 1, \ldots, \Phi$ (і $\int t^k \psi(t)dt \neq 0 \; при \; k = \Phi + 1, \ldots$). Число Φ називається кількістю зникаючих моментів (vanishing moments) вейвлета, і є однією з найбільш важливих властивостей вейвлетів на практиці.

Ряд «стандартних» МРА описаний в літературі. Найбільш відомий приклад – МРА, які генерують так звані Добечье- (*Daubechies*) сімейства вейвлетів.

При МРА, інформація в x переписується як сукупність деталізацій різних рівнів і апроксимації з низькою розподільною здатністю. Щоб це проілюструвати, запишемо $x \in L^2(R)$ як суму грубої апроксимації на рівні $\Im$ і деталізацій на всіх рівнях нижче або рівних $\Im$. Апроксимація записується як лінійна комбінація $\phi_{\Im,k}$, деталізація на рівні j записується як лінійна комбінація $\psi_{j,k}$:

$$x(t) = \sum_{k \in Z} a_x(\Im, k)\phi_{\Im,k}(t) + \sum_{j=-\infty}^{\Im} \sum_{k \in Z} d_x(j,k)\Psi_{j,k}(t) \qquad (3.54)$$

Коефіцієнти $a_x(\Im, k)$ й $d_x(j,k)$ визначаються скалярними добутками x і масштабуючої функції відповідного материнського вейвлета:

$$a_x(\Im, k) = \left\langle x, \phi_{\Im,k} \right\rangle \qquad (3.55)$$

$$d_x(j,k) = \left\langle x, \psi_{\Im,k} \right\rangle, \qquad (3.56)$$

для $j \le \Im$.

Існує ефективний алгоритм обчислення цих вейвлетних коефіцієнтів рекурсивно, однак ми в основному зацікавлені в *створенні* x, а не в розкладанні.

У випадку ФБР, $d_{B_H}(j,k)$ обчислюється як

$$d_{B_H}(j,k) = \left\langle B_H(t), \psi_{j,k}(t) \right\rangle = \int_R B_H(t)\psi_{j,k}(t)dt \qquad (3.57)$$

Опускаючи деталі, інтеграл визначається як межа (в L^2-нормі) апроксимування стохастичних інтегралів. Дисперсія $d_{B_H}(j,k)$ дорівнює $\sigma^2 2^{j(2H+1)}$, де $\sigma > 0$ – деяка константа. При $H > 1/2$ коваріації вейвлетних коефіцієнтів досить малі, але їхнє убування сильно залежить від вибору материнського вейвлета ψ. Відзначимо, що коефіцієнти (майже) незалежні якщо коваріації дуже близькі до 0. Наприклад, показано, що для найпростішого ортонормального вейвлетного базису, системи Хаара $(\Phi = 1)$, для якого

$$\Psi(t) = \begin{cases} +1 \text{ при } 0 \le t < 1/2 \\ -1 \text{ при } 1/2 \le t < 1 \\ 0 \text{ в інших випадках} \end{cases} \qquad (3.58)$$

припущення про незалежність коефіцієнтів сильно порушується. Проте, при використанні вейвлета з більш, ніж одним зникаючим моментом, коефіцієнти вейвлета «приблизно» незалежні.

Якщо ми хочемо використати наведені вище міркування для моделювання самоподібної вибірки, природно аналізувати $B_H(t)$ за допомогою МРА. Відзначимо, що фінітна вибірка ФБР із імовірністю 1 є елементом $L^2(R)$. Тому метод наближеного генерування може бути заснований на (3.54). Визначається процес

$$\widetilde{B}_H(t) = \lim_{\Im\to\infty} \sum_{j=-\infty}^{\Im} \sum_{k\in Z} \widetilde{d}_{B_H}(j,k) 2^{-j/2} \psi(2^{-j}t - k), \qquad (3.59)$$

де $\widetilde{d}_{B_H}(j,k)$ – незалежні Гаусові випадкові змінні з дисперсією $\sigma^2 2^{j(2H+1)}$.

Показано, що цей процес має спектральні властивості, подібні до властивостей ФБР. Отже, $\widetilde{B}_H$ – це ФБР-подібний процес, і його можна використати для апроксимації ФБР. Якість апроксимації покращується при збільшенні Φ. Однак швидкість методу часто дуже сильно залежить від кількості зникаючих моментів обраного вейвлета. Вейвлет Добечьє-10, для якого $\Phi \approx 2$ (оскільки обчислення Φ засноване на асимптотичних аргументах, точне значення невідомо) вважається гарним вибором.

У роботі [63] аргументовані два головних недоліки цього методу. Першим є те, що апроксимація на найгрубішому рівні прирівнюється до нуля, оскільки нескінченна сума в (3.59) повинна бути усічена. Їх обґрунтуванням є те, що груба апроксимація містить інформацію про низькочастотну поведінку вибірки, тоді як деталізації описують високочастотну поведінку. Забути про грубу апроксимацію означає забути про ПСЗ і самоподобу, що, саме собою, небажано. Більш того, вони аргументують, що використання незалежних вейвлетних коефіцієнтів з ортонормальними вейвлетами призводить до наближеної структури кореляції. Тому дуже можливо, що вейвлетні вибірки

не будуть мати бажаних статистичних властивостей ФБР. У літературі ці вибірки часто називається ФБР 2-го типу [64].

Є ще один аспект, про який необхідно подбати. Важлива операція алгоритму це згортання послідовностей. При згортанні двох послідовностей початкова і кінцева точки отриманої послідовності зіпсовані недостачею даних. Це так званий крайовий ефект. З цим борються за допомогою збільшення розмірів вибірки на деякий інтервал, і подальшого обрізання отриманої послідовності (по обидва боки!). Через це складно зробити теоретичні висновки про необхідну кількість часу.

2.6. Інші методи

Розроблено велику кількість інших методів для апроксимованого синтезу самоподібних процесів. У цьому розділі опишемо деякі з них і обговоримо їхні переваги та недоліки. Звичайно, цей список не претендує на повноту.

2.6.1. Метод хаотичних карт

Цей підхід згідно [65] є методом моделювання поведінки на рівні джерел. За даною початковою випадковою величиною x_0 і детермінованою функцією f створюється детермінована послідовність за рекурсією

$$x_{n+1} = f(x_n). \qquad (3.60)$$

Активні і неактивні періоди визначаються створеною послідовністю, наприклад, джерело активне в момент n, якщо $0 < x_n < 1/2$ Хаос (безладна або здавана випадкова поведінка) з'являється завдяки властивості так званої чутливої залежності до початкових умов. Це означає, що невелике відхилення

від початкових умов призводить до зовсім іншої послідовності. Фактично, абсолютна зміна *x* зростає експоненціально при збільшенні *n*.

Показано, що прості нелінійні хаотичні карти також відбивають деякі властивості реального телетрафіку. Однак, вони не підходять для прямої інтерпретації. Тому, кілька таких карт повинні бути створені в надії на одержання задовільних результатів симуляції самоподібного трафіку. Більше того, вихідна послідовність повинна бути агрегована, що призводить до деяких проблем як у випадку з методом агрегування процесів джерел (див. розділ 3.3.2).

2.6.2. Метод Декройсфонда та Лавода

Метод Декройсфонда (*Decreusefond*) і Лавода (*Lavaud*) [66] заснований на такому поданні фрактального Броунівського руху:

$$B_H(t) = \int_0^t K_H(t,s)dB(s) \qquad (3.61)$$

де $K_H(t, s)$ – деяка функція, B – звичайний Броунівський рух.

Припустимо, що нам потрібна вибірка фрактального Броунівського руху в інтервалі [0, 1], цей інтервал потім розбивається на N менших інтервалів рівної довжини. Нехай $t_n = n/N$ при $n = 0, 1,\ldots, N$...

Ми оцінюємо точку n вибірки як

$$B_H(t_n) = \sum_{i=0}^{n} \frac{1}{t_{i+1} - t_i} \int_{t_i}^{t_{i+1}} K_H(t_{n,s})ds\left(B(t_{t_{i+1}}) - B(t_{t_i})\right). \qquad (3.62)$$

Інтеграл $\int_{t_i}^{t_{i+1}} K_H(t_{n,s})ds$ може не апроксимуватися $K_H(t_n, t_i)$ або $K_H(t_n, t_{i+1})$, оскільки $K_H(t_n, t)$ не неперервна відносно t на інтервалі $[0, t_n]$.

Тому апроксимація може не сходитися до конкретної вибірки при великих N. В [66] автори описують подальший шлях для подолання цієї проблеми.

Хоча цей метод не був ретельно протестований і оцінений, головною його перевагою, зрозуміло, є те, що інтеграл в (3.62) може бути обчислений усього один раз, якщо потрібно більше однієї вибірки. Однак, кількість інтегралів, які потрібно обчислити – порядку N^2 при необхідній довжині вибірки N. Всі ці величини необхідно зберігати, що є недоліком методу. Оскільки існують навіть точні методи, що працюють набагато швидше цього, метод не дуже часто застосовується на практиці.

2.6.3. Метод Джапарідзе і ван Зантена

Метод Джапарідзе (*Dzhaparidze*) і ван Зантена (*van Zanten*) [67] використовує додатні нулі $x_1 < x_2 < \ldots$ функції Бесселя J_{-H} першого роду порядку $-H$ та $y_1 < y_2 < \ldots$ функції J_{1-H}. Звідси, у змісті рівності розподілів,

$$B_H(t) = \sum_{n=1}^{\infty} \frac{\sin x_n t}{x_n} X_n + \sum_{n=1}^{\infty} \frac{1-\cos y_n t}{y_n} Y_n, \qquad (3.63)$$

де $X_1, X_2, \ldots$ і $Y_1, Y_2, \ldots$– незалежні центровані Гаусові випадкові змінні. Їхні дисперсії виражаються як

$$D[X_n] = 2c_H^2 x_n^{-2H} J_{1-H}^{-2}(x_n), \qquad (3.64)$$

$$D[Y_n] = 2c_H^2 y_n^{-2H} J_{-H}^{-2}(y_n), \qquad (3.65)$$

де $c_H^2 = \pi^{-1}\Gamma(1+2H)\sin\pi H$.

Обидва ряди абсолютно та рівномірно сходяться в інтервалі $t \in [0;1]$ з імовірністю, рівною 1.

Цей метод особливо цікавий для моделювання, оскільки існують ефективні алгоритми обчислення нулів функції Бесселя. Більше того, ці нулі необхідно

обчислити всього один раз, незалежно від того, скільки вибірок необхідно генерувати. Природно, що ці ряди необхідно усікти на деякому рівні, і актуальним є питання рівня усікання.

Додаток Б . УТИЛІТИ ГЕНЕРАЦІЇ САМОПОДІБНОГО МЕРЕЖНОГО ТРАФІКУ

Утиліти генерації самоподібного мережного трафіку знаходяться в чотирьох заголовних файлах:

definition.h

random.h

source.h

generate.h

Дані утиліти дозволяють генерувати самоподібний трафік за допомогою об'єднання великої кількості джерел, що генерують *ON*- і *OFF*- періоди з розподілом Парето. Кожне джерело генерує пакети тільки одного розміру. Розподіл Парето забезпечується як у пачці пакетів (викидів), так і у проміжках (зазорах) між пачками пакетів.

Для пакетного трасування (програмна мітка часу, розмір пакета) може бути використаний наступний лістинг main-функції:

```
#include " generate.h"

void main(void)
{
/* Генератор здійснює об'єднання трафіку від 40 джерел і
генерування навантаження 0.3 на лінії зв'язку зі швидкістю 100
Мбіт/с (середня пропускна здатність становить 30 Мбіт/с)  */
      Generator Gen(100.0, 0.3, 40);
// генерування одного мільйона пакетів і вивід їх на STDOUT */
  Gen.OutputTraces( 1000000 );
}
```

Слід зазначити, що внаслідок обмеженої варіації розподілу Парето в межах 1 < alpha < 2, агреговане навантаження може значно змінюватися. Тому

може знадобитися велика кількість ітерацій для вибору одного навантаження, що найкраще відповідає заданому.

Якщо кількість пакетів не істотна, але важливим є досягнення середнього навантаження, то може бути використаний наступний лістинг main-функції:

```
#include " generate.h"

void main(void)
{
 DOUBLE load  = 0.3;       /* розрахункове навантаження */
 DOUBLE delta = 0.01;      /* припустиме розходження між
                              необхідним і одержуваним
навантаженням */
/* генерування не менш 100 млн. пакетів */
int32u max_packets = 100000000L;
    Generator Gen(100.0, load, 40);
    while( Gen.GetPackets() < max_packets && fabs( Gen.GetLoad()
- load) > delta )
        cout << Gen.GenerateTrace() << endl;
}
```

Коли Генератору приписуються ненульові значення навантаження і кількості джерел, то кожне джерело буде мати ті самі значення для преамбули, параметри форми (розподілу) пакетів і зазорів між ними. Якщо існує випадок, коли джерела будуть відрізнятися за цими параметрами, то може бути використаний такий код:

```
Generator Gen(100.0); // Немає навантаження, а джерело
формується
                      // додаванням індивідуальних джерел
//        packet size | preamble | min. gap | burst shape | gap
shape
Gen.AddSource( 64     ,     8    ,   1000   ,     1.4     ,
1.6  );
```

```
Gen.AddSource(512     ,    16    ,    4000    ,     1.7     ,
1.3  );
.....................................................................
.....
```

Preamble визначає мінімальний зазор між пакетами в серії (пачці) індивідуальних джерел. Щоб визначити значення преамбули для агрегованого трафіку використовується

```
Gen.Preamble = x;
```

Мінімальний зазор визначається як мінімальний інтервал між сплесками (пачками). Оскільки всі параметри в цьому випадку конфігуруються користувачем, то для Генератора не існує специфікації навантаження.

Приклад результату моделювання пакетів трафіку періодами *ON/OFF* з розподілами Парето і експоненціальним розподілом при різних часових масштабах представлений на рис. А.1. Трафік має навантаження 30% і генерувався шляхом об'єднання 40 джерел *ON/OFF*.

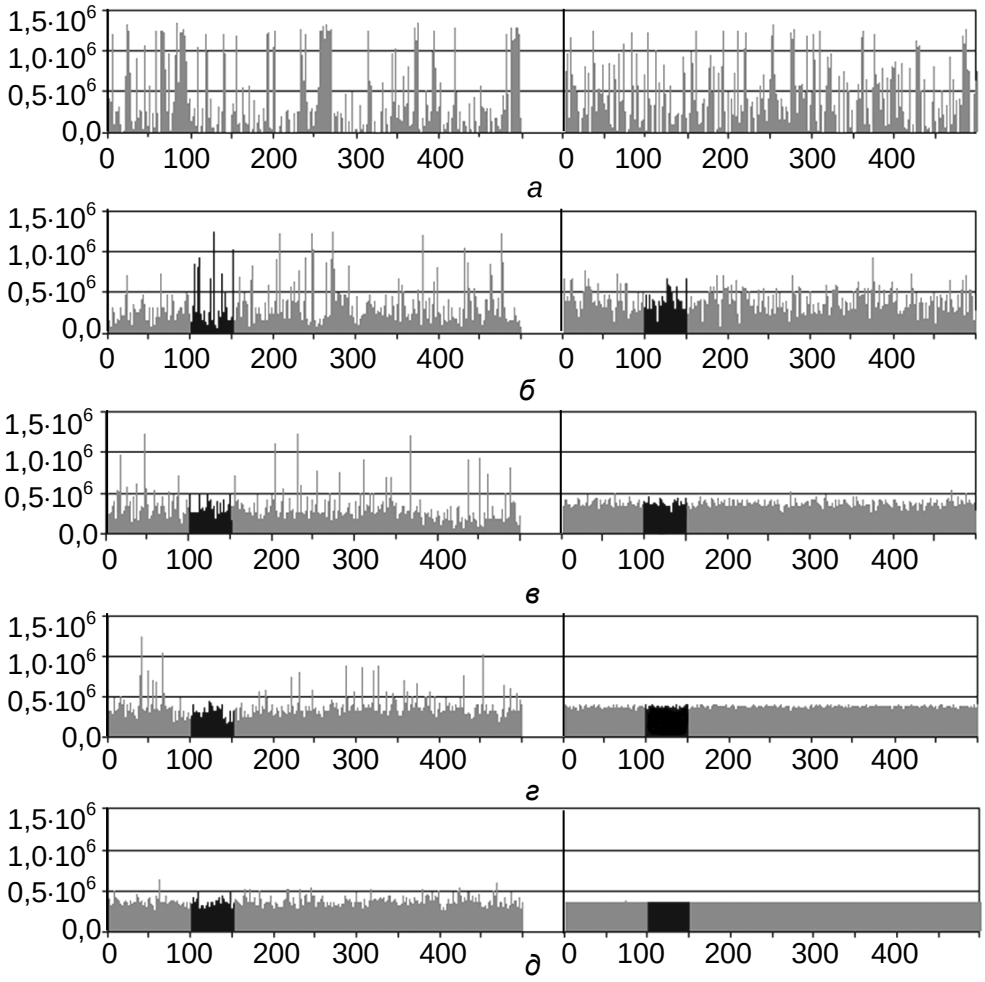


Рис. А. 1. Пакети трафіку, змодельовані за допомогою розподілів Парето (ліві графіки трафіку) і експонціального (праві графіки трафіку) у масштабах: *а* – 1 мс; *б* – 10 мс; *в* – 100 мс; *г* – 1 с; *д* – 10 с

Лістинги файлів:

definition.h

```
/*************************************************
 *        Contains type definitions and          *
 *              general macros                   *
 *************************************************/

#ifndef _TYPES_H_V001_INCLUDED_
#define _TYPES_H_V001_INCLUDED_

#ifndef NULL
#define NULL ((void*) 0L)
#endif

// Datatype abstractions

/* #if defined _USE_MSVC_SPECIFIC_TYPES_

    typedef unsigned __int8     int8u;
    typedef signed   __int8     int8s;

    typedef unsigned __int16    int16u;
    typedef signed   __int16    int16s;

    typedef unsigned __int32    int32u;
    typedef signed   __int32    int32s;

    typedef unsigned __int64    int64u;
    typedef signed   __int64    int64s;

    typedef unsigned __int8     BOOL;
    typedef unsigned __int8     CHAR;
    typedef unsigned __int8     BYTE;
    typedef unsigned int        WORD;

    typedef double              DOUBLE;

#else  */
/* use generic types */

    typedef unsigned char       int8u;
    typedef signed   char       int8s;

    typedef unsigned short int  int16u;
    typedef signed   short int  int16s;

    typedef unsigned long       int32u;
    typedef signed   long       int32s;

    typedef unsigned char       BOOL;
    typedef unsigned char       CHAR;
    typedef unsigned char       BYTE;
    typedef unsigned int        WORD;

    typedef double              DOUBLE;
```

```
//#endif

template < class T > inline T round( DOUBLE val )
                                        { return (T)( val + 0.5 ); }
template < class T > inline T MAX( T x, T y )
                                        { return x > y? x : y; }

const BOOL TRUE  = 1;
const BOOL FALSE = 0;

#endif // _TYPES_H_V001_INCLUDED_
```

---

random.h

```
/*********************************************************
 *     Contains routines for generating random values     *
 *********************************************************/

#ifndef _RAND_H_V001_INCLUDED_
#define _RAND_H_V001_INCLUDED_

#include <stdlib.h>
#include <time.h>
#include <math.h>

typedef double rnd_t;

// range of random values returned by rand()  [0 ... 2^15]
const int16u SMALL_RANGE  =  15;
// extended range of random values [0 ... 2^24]
const int16u LARGE_RANGE  =  24;

const int16u EXTENTION    = LARGE_RANGE - SMALL_RANGE;
const int16u EXTEN_MASK   = (0x0001 << (EXTENTION + 1)) - 1;

#define EXT_RANGE( X, Y )  ((((int32u)X) << EXTENTION ) | ( Y &
EXTEN_MASK ))

#if defined( RND_LARGE_RANGE )

    int32u _lrand(void)        { return EXT_RANGE( rand(), rand() ); }
    #define RAND               _lrand
    const int32u RND_MAXIMUM = EXT_RANGE( RAND_MAX, RAND_MAX );

#else

    #define RAND               rand
    const int32u RND_MAXIMUM = RAND_MAX;

#endif


const rnd_t SMALL_VAL = 0.5 / RND_MAXIMUM;
```

```
inline void   _seed(void)
                { srand( (unsigned)time( NULL ) );          }
inline rnd_t _uniform01(void)
                { return ((rnd_t)RAND()) / RND_MAXIMUM;     }
inline rnd_t _uniform_(rnd_t low, rnd_t hi)
                { return (hi - low) * _uniform01() + low;   }
inline rnd_t _uniform_non0(void)
                { return _uniform_( SMALL_VAL, 1.0 );       }
inline rnd_t _exponent_(void)
                { return -log( _uniform_non0() );           }
inline rnd_t _pareto_(rnd_t shape)
                { return pow( _uniform_non0(), -1.0/shape); }

#endif /* _RAND_H_V001_INCLUDED_ */
```

---

source.h

```
/************************************************
 *                                              *
 *          Contains declarations for           *
 *              class Source                    *
 *              class SourceLink                *
 *                                              *
 ************************************************/

#if !defined(_SOURCE_H_INCLUDED_)
#define _SOURCE_H_INCLUDED_

//#define RND_LARGE_RANGE

#include "definition.h"
#include "random.h"

typedef int16u   pct_size_t;
typedef DOUBLE   bytestamp_t;

#define MIN_BURST   1.0

// Choose one of the following lines
// Use Pareto distributed ON and OFF periods
inline DOUBLE rnd_val( DOUBLE shape )   { return _pareto_( shape ); }

/* Use exponentially distributed ON and OFF periods */
//inline DOUBLE rnd_val( DOUBLE shape )
//                    { return _exponent_() / (shape - 1.0) + 1.0; }

class Source
{
private:
    bytestamp_t  Elapsed;  // elapsed time (in byte transmission
times)
    pct_size_t   PctSize;  // packet size (bytes)
    pct_size_t   Preamble; // minimum inter-packet gap value (in
bytes)
    int32u       MinGap;   // minimum inter-burst gap value (in bytes)
```

```
    int16u       BurstSize;// number of packets remaining in current
burst
    DOUBLE       PctShape; // shape parameter for burst distribution
    DOUBLE       GapShape; // shape parameter for inter-burst gap
distribution

public:

    Source(pct_size_t pct_sz, pct_size_t preamble, int32u min_gap,
           DOUBLE pshape, DOUBLE gshape )
    {
        PctSize   = pct_sz;
        Preamble  = preamble;
        MinGap    = min_gap;
        PctShape  = pshape;
        GapShape  = gshape;

        Reset();
    }

    virtual ~Source()       {}

    //-----------------------------------------------------------------

    inline void Reset(void)
    {
        Elapsed   = 0.0;
        BurstSize = 0;
        ExtractPacket();
    }
    //-----------------------------------------------------------------

    inline bytestamp_t GetArrival(void) { return Elapsed; }
    inline pct_size_t  GetPctSize(void) { return PctSize; }


////////////////////////////////////////////////////////////////////////
//
  // FUNCTION:    void ExtractPacket(void)
  // DESCRIPTION: Generates new packet
  // NOTES:  If BurstSize > 0, the new packet assumed to immediately
follow
  //         the previous packet (with minimum inter-packet gap =
Preamble.)
  //         If BurstSize == 0, new burst size is generated from
Pareto
  //         distribution. Elapsed time is also incremented by a
  //         Pareto-distributed value to account for inter-burst gap.

////////////////////////////////////////////////////////////////////////
///
    inline void ExtractPacket(void)
    {
        if( BurstSize == 0 )
        {
            BurstSize = round<int16u>( rnd_val( PctShape ) * MIN_BURST
);
            Elapsed  += round<int32u>( rnd_val( GapShape ) * MinGap );
```

```
        }
        BurstSize--;
        Elapsed += ( PctSize + Preamble );
    }
    //-------------------------------------------------------
};


//-----------------------------------------------------------

class SourceLink: public Source
{
  private:
    SourceLink* pNext;

  public:
    SourceLink( pct_size_t pct_sz,
                pct_size_t preamble,
                int32u min_gap,
                DOUBLE pshape,
                DOUBLE gshape ):
                Source( pct_sz, preamble, min_gap, pshape, gshape )
                       { pNext = NULL; }
    virtual ~SourceLink() {}

    inline SourceLink* GetNext(void)
                                  { return pNext; }
    inline void        Insert(SourceLink* prv, SourceLink* nxt)
                                  { if(prv) prv->pNext = this; pNext =
nxt;}
    inline void        Remove(SourceLink* prv)
                                  { if(prv) prv->pNext = pNext; }
};
//-----------------------------------------------------------------

#endif // _SOURCE_H_INCLUDED_
```

---

generate.h

```
/*******************************************
 *    Declarations for class Trace and       *
 *              class Generator              *
 *******************************************/

#if !defined(_AGGREG_H_INCLUDED_)
#define _AGGREG_H_INCLUDED_

#include <iostream.h>
#include <iomanip.h>
#include "definition.h"
#include "source.h"

//  dafault values:
const int16s BYTE_SIZE  = 8;
const int16s PREAMBLE   = 8;
const int16s MIN_PACKET = 64;
```

```
const int16s MAX_PACKET = 1518;

const DOUBLE PCT_SHAPE  = 1.7;
const DOUBLE GAP_SHAPE  = 1.2;

const DOUBLE MIN_ALPHA  = 1.0;
const DOUBLE MAX_ALPHA  = 2.0;

typedef DOUBLE timestamp_t;

//---------------------------------------------------------------------

class Trace
{
public:
    timestamp_t TimeStamp;
    pct_size_t  PacketSize;

    Trace()                                 { TimeStamp = 0.0;
PacketSize = 0; }
    Trace(timestamp_t ts, pct_size_t ps ) { TimeStamp = ts; PacketSize
= ps; }
    ~Trace()                                {}
};

////////////////////////////////////////////////////////////////////
// FUNCTION:    ostream& operator<< ( ostream& os, const Trace& trc )
// DESCRIPTION: Insert operator for the class Trace
////////////////////////////////////////////////////////////////////
inline ostream& operator<< ( ostream& os, const Trace& trc )
{
    return ( os << trc.TimeStamp << "    " << trc.PacketSize );
}

// ---------------------------------------------------------------------
-
// ---------------------------------------------------------------------
-

class Generator
{
private:
    SourceLink* pFirst;       // pointer to the source of the next
packet
    timestamp_t ByteTime;     // Byte transmission time
    bytestamp_t TotalBytes;   // Total bytes generated
    int32s      TotalPackets; // Total packets generated
    bytestamp_t Elapsed;      // Elapsed time (expressed in ByteTime
units)
                              //          since the beginning of the
trace


////////////////////////////////////////////////////////////////
    // FUNCTION:    void InsertInOrder( SourceLink* pSrc )
    // DESCRIPTION: Inserts Source of packet into linked list of
Sources
    //              ordered by packet arrival times
```

```
////////////////////////////////////////////////////////////////
    void InsertInOrder( SourceLink* pSrc )
    {
        SourceLink *pPrv = NULL, *pNxt  = pFirst;

        for( ; pNxt && pSrc->GetArrival() > pNxt->GetArrival();
                                                pNxt = pNxt-
>GetNext() )
            pPrv = pNxt;

        pSrc->Insert(pPrv, pNxt);
        if( !pPrv ) pFirst = pSrc;
    }

    //-----------------------------------------------------------------
--

public:
    int16u      Preamble;   // minimum interpacket gap
    int16u      MinPacket;  // minimum packet size
    int16u      MaxPacket;  // maximum packet size


////////////////////////////////////////////////////////////////////
/
    // FUNCTION:    Generator( DOUBLE line_rate_Mbps, DOUBLE load =
0.0,
    //                         int16s sources = 0 )
    // DESCRIPTION: Constructor
    // NOTES:       line_rate_Mbps - rate of generated trace
    //              load           - desired line load (bandwidth
utilization)
    //                               of the generated trace
    //              sources        - number of sources to aggregate

////////////////////////////////////////////////////////////////
    Generator( DOUBLE line_rate_Mbps, DOUBLE load = 0.0, int16s
sources = 0 )
    {
        _seed();

        pFirst    = NULL;
        Preamble  = PREAMBLE;
        MinPacket = MIN_PACKET;
        MaxPacket = MAX_PACKET;
        // byte transmission time (sec)
        ByteTime  = BYTE_SIZE / ( line_rate_Mbps * 1.0E6);

        AddSources( load, sources, PCT_SHAPE, GAP_SHAPE );
    }

    ////////////////////////////////////////////
    // FUNCTION:    ~Generator()
    // DESCRIPTION: Destructor
    ////////////////////////////////////////////
    virtual ~Generator()        { RemoveSources(); }
```

```
    ////////////////////////////////////////////////////////////////
    // FUNCTION:    void Reset( void )
    // DESCRIPTION: resets all statistic variables and elapsed time
    // NOTES:       Reset should be called to generate multiple
    //              traces with same set of sources
    ////////////////////////////////////////////////////////////////
    void Reset( void )
    {
        TotalBytes   = 0.0;
        Elapsed      = 0.0;
        TotalPackets = 0;
        for( SourceLink* p = pFirst; p; p = p->GetNext() )
            p->Reset();
    }

    //----------------------------------------------------------------
---

    inline int32s      GetPackets(void)  { return TotalPackets; }
    inline bytestamp_t GetBytes(void)    { return TotalBytes;   }
    inline bytestamp_t GetTime(void)     { return Elapsed;      }
    inline DOUBLE      GetLoad(void)     { return TotalBytes /
Elapsed; }

 ////////////////////////////////////////////////////////////////////
 // FUNCTION:    void AddSource( pct_sz, preamble, min_gap, pshape,
gshape )
 // DESCRIPTION: Adds new source to the Generator.
 // NOTES:       OFF periods are Pareto-distributed with alpha =
gshape
 //              and mimimum gap size of (coef * packet_size).
 //              ON periods consist of bursts of packets with
 //              Pareto-distributed number of packets per burst
(minimum = 1)
 //              and alpha = pshape.
 ////////////////////////////////////////////////////////////////////
    void AddSource(pct_size_t pct_sz, pct_size_t preamble,
                   int32u min_gap, DOUBLE pshape, DOUBLE gshape )
    {
        InsertInOrder( new SourceLink( pct_sz, preamble,
                       min_gap, pshape, gshape ) );
    }


////////////////////////////////////////////////////////////////////
    // FUNCTION:    AddSources( DOUBLE load, int16s sources,
    //                          DOUBLE pshape, DOUBLE gshape )
    ////////////////////////////////////////////////////////////////
    void AddSources( DOUBLE load, int16s sources,
                     DOUBLE on_shape, DOUBLE off_shape )
    {
        pct_size_t  packet_size;
        DOUBLE on_coef, off_coef, coef;

        if( ( load > 0.0 && load <= 1.0 )  &&
            ( on_shape  > MIN_ALPHA && on_shape  < MAX_ALPHA ) &&
            ( off_shape > MIN_ALPHA && off_shape < MAX_ALPHA ) )
        {
```

```
            on_coef  = on_shape  / ( on_shape - 1.0 );
            off_coef = off_shape / ( off_shape - 1.0 );
            coef     = (sources / load - 1.0) * ( on_coef / off_coef
);

     /* coef - coefficient used to calculate minimum inter-burst
interval
      *        such that aggregated traffic from all sources would
produce
      *        the desired link load.
      *    (1)  LOAD = SOURCES * ( MEAN_ON / MEAN_ON + MEAN_OFF );
      *    (2)  MEAN_ON   = MIN_ON  * on_shape  / (on_shape - 1)  =
      *                                                    MIN_ON  *
on_coef;
      *    (3)  MEAN_OFF  = MIN_OFF * off_shape / (off_shape - 1) =
      *                                                  MIN_OFF *
off_coef;
      *    (4)  MIN_OFF   = MIN_ON * ( SOURCES / LOAD - 1) *
      *                                                ( on_coef /
off_coef );
      *        MIN_OFF   = MIN_ON * COEF
      *    (5)  COEF      = ( SOURCES / LOAD - 1) * ( on_coef /
off_coef )
      *
      *        Due to infinite variance of Pareto distribution with
      *        [1 < alpha < 2], the aggregated load of generated trace
      *        may fluctuate considerably.  Multiple iterations may be
      *        needed to choose the one with load closest to the
specified.
      */
            while( sources-- > 0 )
            {
                // Every source has a constant packet size from
uniform
                // distribution [MinPacket ... MaxPacket]
                //
                packet_size = round<int16u>(_uniform_(MinPacket,
MaxPacket));
                AddSource(packet_size, Preamble, (int32u)(coef *
packet_size),
                          on_shape, off_shape);
            }
        }
        Reset();
    }

    ////////////////////////////////////////
    // FUNCTION:    RemoveSource( void )
    // DESCRIPTION: Removes first source
    ////////////////////////////////////////
    inline void RemoveSource( void )
    {
        if( pFirst )
        {
            SourceLink* pSrc = pFirst;
            pFirst = pFirst->GetNext();
            delete pSrc;
        }
```

```
    }

    ////////////////////////////////////////
    // FUNCTION:    RemoveSources( void )
    // DESCRIPTION: Removes all sources
    ////////////////////////////////////////
    void RemoveSources( void )  { while( pFirst ) RemoveSource(); }

    ////////////////////////////////////////////////////////////////
    // FUNCTION:    Trace GenerateTrace( void )
    // DESCRIPTION: Return the next packet from the aggregated traffic
    ////////////////////////////////////////////////////////////////
    Trace GenerateTrace( void )
    {
        SourceLink* pNextSrc = pFirst;
        // The source at the head of the linked list has a packet with
        //  earliest arrival time that has not been sent yet.
        //
        pFirst  = pFirst->GetNext();    // remove head of the linked
list

        Trace Trc(0.0, pNextSrc->GetPctSize());

   // Elapsed + PacketSize + Preamble  -- earliest time the packet can
be sent
   // pFirst->GetArrival() -- packet arrival time
   //
        Elapsed = MAX( pNextSrc->GetArrival(),
                       Elapsed + Trc.PacketSize + Preamble );
        Trc.TimeStamp = Elapsed * ByteTime; //get timestamp from the
bytestamp
        TotalBytes   += Trc.PacketSize;
        TotalPackets++;

        pNextSrc->ExtractPacket();    // receive new packet
        InsertInOrder( pNextSrc );    // place the source in the list
                                      // in order of packet arrival
        return Trc;
    }

    ////////////////////////////////////////////////////////////
    // FUNCTION:    void OutputTraces( int32s packets )
    // DESCRIPTION: Outputs 'packets' packets to 'cout'
    // stream, then outputs statistics
    ////////////////////////////////////////////////////////////
    void OutputTraces( int32s packets )
    {
        Reset();

        cout.setf(ios::fixed);
        cout.precision(8);

        while( packets-- > 0 )
            cout << GenerateTrace() << endl;

        ShowStatistic();
    }
```

```
    ////////////////////////////////////////////////////////
    // FUNCTION:    void ShowStatistic(void)
    // DESCRIPTION: Outputs statistic to the 'clog' stream
    ////////////////////////////////////////////////////////
    void ShowStatistic(void)
    {
        clog << "\n Total packets: " << TotalPackets
             << "\n Total bytes:   " << TotalBytes
             << "\n Total seconds: " << Elapsed * ByteTime
             << "\n Link load:     " << TotalBytes / Elapsed << endl;
    }

    //--------------------------------------------------------
};

#endif // _AGGREG_H_INCLUDED_
```