

Московский Энергетический Институт
(Технический университет)

на правах рукописи

Петров Виталий Валерьевич

**СТРУКТУРА ТЕЛЕТРАФИКА И АЛГОРИТМ ОБЕСПЕЧЕНИЯ
КАЧЕСТВА ОБСЛУЖИВАНИЯ ПРИ ВЛИЯНИИ ЭФФЕКТА
САМОПОДОБИЯ**

05.12.13 – “Системы, сети и устройства телекоммуникаций”

Диссертация на соискание ученой степени кандидата технических наук

Научный руководитель:
к.т.н., профессор
Е.А. Богатырев

Москва, 2004

Содержание

Перечень сокращений.....	5
Введение.....	7
Глава 1. Современное состояние и основные понятия теории самоподобного телетрафика.....	16
1.1 Понятие фрактальности.....	16
1.2 Самоподобный (фрактальный) телетрафик.....	18
1.2.1 Проблема самоподобного телетрафика.....	18
1.2.2 Определения самоподобного процесса.....	19
1.3 Основные свойства самоподобного трафика.....	22
1.3.1 Медленно, быстро убывающие зависимости, продолжительная память.....	25
1.3.2 Понятие коэффициента Херста.....	27
1.3.3 Понятие фрактальной размерности и ее связь с коэффициентом Хэрста.....	30
1.3.4 Распределения с “тяжелыми хвостами”.....	35
1.3.5 Аспекты теории нелинейной динамики.....	38
1.4 Постановка задачи обеспечения качества обслуживания (QoS) в условиях влияния эффекта самоподобия.....	43
1.5 Выводы по главе 1.....	51
Глава 2. Статистический анализ реализаций сетевого трафика.....	53
2.1 Описание реализаций сетевого трафика.....	53
2.1.1 Реализация сетевого трафика BC-Oct89Ext.TL.....	53
2.1.2 Реализация сетевого трафика LBL-PKT-5.TCP.....	56
2.1.3 Реализация сетевого трафика LBL-TCP-3.....	57
2.2 Формирование временных рядов.....	59
2.2.1 Процедура агрегирования.....	59
2.2.2 Тестовые реализации (хаос и белый шум).....	67
2.2.3 Логарифмированные реализации.....	68
2.3 Классический анализ.....	70

2.3.1	Плотности распределения.....	71
2.3.2	Автокорреляционные функции.....	73
2.3.3	Энергетические спектры.....	78
2.4	Исследование показателя Хэрста реализаций.....	80
2.5	Исследование сетевого трафика методами нелинейной динамики...	82
2.5.1	Концепция суррогатных данных.....	84
2.5.2	Идея реконструкции аттрактора.....	86
2.5.3	Ложные ближайшие соседи.....	86
2.5.4	Вычисление корреляционного интеграла.....	88
2.5.5	Проверка гипотезы о статистической независимости (BDS-тест).....	94
2.6	Эксперимент по сбору трафика в беспроводной сети.....	95
2.6.1	Постановка эксперимента по сбору трафика беспроводной сети.....	95
2.6.2	Характеристика реализаций.....	97
2.6.3	Особенности технологии IEEE 802.11b.....	98
2.6.2	Анализ результатов обработки трафика беспроводной сети.	101
2.7	Выводы по главе 2.....	106
Глава 3. Исследование возможностей прогнозирования самоподобного телетрафика.....		110
3.1	Предпосылки к прогнозированию самоподобного трафика.....	110
3.2	Задача динамического управления пропускной способностью канала с помощью прогнозирования. Оценки выигрыша.....	113
3.3	Анализ алгоритмов управления пропускной способностью канала.	118
3.3.1	Статическое задание пропускной способности.....	119
3.3.2	Динамическое распределение пропускной способности с простым предсказателем.....	121
3.3.3	Динамическое распределение пропускной способности с авторегрессионным предсказателем первого порядка.....	123
3.3.4	Динамическое распределение пропускной способности с авторегрессионным предсказателем второго порядка.....	125

3.3.5	Динамическое распределение пропускной способности с ARMA-предсказателем.....	126
3.3.6	Динамическое распределение пропускной способности с FARIMA-предсказателем.....	128
3.4	Сравнение алгоритмов динамического распределения пропускной способности и выбор метода прогнозирования.....	130
3.5	Выводы по главе 3.....	135
Глава 4. Метод обеспечения качества обслуживания в условиях самоподобного телетрафика.....		137
4.1	Принцип динамического управления пропускной способностью.....	137
4.2	Алгоритмы контроля и управления трафиком.....	141
4.2.1	Алгоритм полисинга на основе механизма “корзина маркеров”.....	141
4.2.2	Алгоритм шейпинга на основе механизма “корзина маркеров”.....	144
4.3	Разработка метода управления трафиком для работы в условиях влияния эффекта самоподобия.....	146
4.4	Моделирование механизма динамического управления пропускной способностью канала с использованием прогнозирования в среде ns-2.....	148
4.5	Анализ результатов моделирования механизма динамического управления пропускной способностью канала с использованием прогнозирования.....	151
4.6	Выводы по главе 4.....	163
Заключение.....		165
Список литературы.....		169
Приложение 1.....		176
Приложение 2.....		180
Акты о внедрении.....		198

Перечень сокращений

АКФ	- Автокорреляционная функция
АСШС	- Асимптотическое самоподобие в широком смысле
БУЗ	- Быстро убывающая зависимость
МУЗ	- Медленно убывающая зависимость
БД	- Броуновское движение
ПО	- Программное обеспечение
РЛХ	- Распределение с “легкими хвостами”
РТХ	- Распределение с “тяжелыми хвостами”
СКО	- Среднеквадратическое отклонение
ССШС	- Строгое самоподобие в широком смысле
СУС	- Самоподобие в узком смысле
ФБД	- Фрактальное броуновское движение
ФГШ	- Фрактальный гауссовский шум
ЦПТ	- Центральная предельная теорема
ASCII	- American standard code for information (американский стандартный код обмена информацией)
ATM	- Asynchronous transfer mode (асинхронный режим передачи)
AR	- Autoregressive (авторегрессия)
ARMA	- Autoregressive moving average (авторегрессионное скользящее среднее)
ARIMA	- Autoregressive integrated moving average (авторегрессионное проинтегрированное скользящее среднее)
CBR	- Constant bit rate

- (постоянная скорость передачи)
- CIR - Committed information rate
(согласованная скорость передачи информации)
- CRC - Cyclical redundancy check
(контроль с помощью циклического избыточного кода)
- CSMA/CA - Carrier Sense Multiple Access with Collision Avoidance
(множественный доступ с контролем несущей и предотвращением конфликтов)
- DSSS - Direct-Sequence Spread Spectrum
(расширение спектра с помощью прямой последовательности)
- FARIMA - Fractionally differenced autoregressive integrated moving average
(Фрактальное авторегрессионное проинтегрированное скользящее среднее)
- FNN - False Nearest Neighbors
(ложные ближайшие соседи)
- OSI - Open system interconnection
(взаимодействие открытых систем)
- QoS - Quality of Service
(качество обслуживания)
- VBR - Variable bit rate
(непостоянная скорость передачи)
- VC - Virtual channel
(виртуальный канал)

Введение

При проектировании, запуске и эксплуатации информационных телекоммуникационных сетей одной из основных проблем является задача обеспечения качества обслуживания (заданных уровней задержек, потерь и пр.) при обработке потока данных - трафика, являющегося следствием информационного обмена между системами.

До недавнего времени теоретическую базу для проектирования систем распределения информации обеспечивала теория телетрафика, которая является одной из ветвей теории массового обслуживания и появилась в результате работ А.К. Эрланга, Т. Энгсета, Г. О'Делла, К. Пальма, А.Я. Хинчина и др. Данная теория хорошо описывает процессы, происходящие в таких системах распределения информации, как телефонные сети, построенных по принципу коммутации каналов. Наиболее распространенной моделью потока вызовов (данных) в теории телетрафика является простейший поток (стационарный ординарный поток без последствия), также называемый стационарным пуассоновским потоком.

Настоящий период бурного развития высоких технологий привел к появлению и повсеместному распространению сетей с пакетной передачей данных, которые постепенно стали вытеснять системы с коммутацией каналов, но, по-прежнему, они проектировались на основе общих положений теории телетрафика.

Однако, в 1993 году группа американских исследователей W.Leland, M.Taqqu, W.Willinger и D.Wilson опубликовали результаты своей новой работы, которая в корне изменила существующие представления о процессах, происходящих в телекоммуникационных сетях с коммутацией пакетов. Эти исследователи изучили трафик в информационной сети корпорации Bellcore и обнаружили, что потоки в ней нельзя аппроксимировать простейшими и, как следствие, они уже имеют совершенно иную структуру, чем принято в классической теории телетрафика. В частности, было установлено, что трафик

такой сети обладает так называемым свойством “самоподобия”, т.е. выглядит качественно одинаково при почти любых масштабах временной оси, имеет память (последствие), а также характеризуется высокой пачечностью¹.

В результате теоретический расчет параметров системы распределения информации, предназначенной для обработки такого трафика, по классическим формулам дает некорректные и неоправданно оптимистические результаты. Более того, привычные алгоритмы обработки трафика, созданные для работы с простейшими потоками, оказываются недостаточно эффективными для потоков с самоподобием.

Таким образом, образовалась “проблема самоподобия телетрафика”, которой за последние 11 лет посвящено более тысячи работ и которая до сих пор не утратила своей актуальности. Среди зарубежных ученых, активно занимающихся этой проблемой, необходимо выделить уже упоминавшихся авторов, которым принадлежат наиболее фундаментальные труды в этом направлении, а также К. Park, В. Ryu, V. Paxson, R. Mondragon и др. Среди отечественных исследователей необходимо отметить работы В.И. Неймана, Б.С. Цыбакова, Н.Б. Лиханова, О.И. Шелухина, В.С. Заборовского, А.Я. Городецкого и др.

Несмотря на значительную популярность этой тематики и продолжительный (11 лет) период ее активного изучения, приходится констатировать, что до сих пор остается множество вопросов и нерешенных задач. Перечислим, на наш взгляд, основные из них:

- фактически отсутствует строгая теоретическая база, которая пришла бы на смену классической теории массового обслуживания при проектировании современных систем распределения информации с самоподобным трафиком;

- нет единой общепризнанной модели самоподобного трафика;
- не существует достоверной и признанной методики расчета

¹ Коэффициент пачечности (пачечность) для заданного потока соответствует отношению пиковой интенсивности процесса поступления заявок на обслуживание к его среднему значению.

параметров и показателей качества систем распределения информации при влиянии эффекта самоподобия;

- отсутствуют алгоритмы и механизмы, обеспечивающие качество обслуживания в условиях самоподобного трафика.

Настоящая диссертация посвящена решению последней из перечисленных, но далеко не последней по степени важности задаче.

Целью настоящей диссертационной работы является разработка алгоритма обеспечения качества обслуживания в системах распределения информации с самоподобным трафиком. Данный алгоритм должен обеспечить увеличение эффективности обработки самоподобного телетрафика с точки зрения улучшения таких показателей как задержки, потери пакетов, а также коэффициент использования системы. Для этого требуется провести анализ состояния проблемы и решить следующие основные задачи:

- Формирование основных идей и принципа функционирования алгоритма обеспечения качества обслуживания в условиях самоподобного трафика;

- Подготовка и проведение эксперимента по сбору трафика, а также выполнение статистического анализа реализаций трафика на предмет выявления его характерных особенностей, которые необходимо учитывать при разработке алгоритма обеспечения качества обслуживания в условиях самоподобного трафика;

- Анализ прогнозируемости сетевого трафика как базовой концепции разрабатываемого алгоритма;

- Разработка блок-схемы функционирования и принципов реализации в существующих системах механизма обеспечения качества обслуживания в условиях самоподобного трафика;

- Проведение испытаний (имитационное моделирование) и оценка эффективности разработанного алгоритма.

Методы исследования. Для решения перечисленных задач в работе использовались методы статистической обработки данных, теории нелинейных динамических систем, в том числе, хаотических (реконструкция динамической системы по ее реализации), регрессионного анализа временных рядов, имитационного моделирования.

Научная новизна. В диссертации получены следующие новые научные и практические результаты:

1. Разработан новый алгоритм обеспечения качества обслуживания в условиях самоподобного трафика, использующий прогнозирование.

2. Показано, что метод динамического распределения пропускной способности канала, основанный на прогнозировании, дает ощутимый выигрыш в уменьшении потерь и увеличении использования канала при самоподобном телетрафике по сравнению со статическим способом распределения при том же самом среднем значении пропускной способности.

3. Впервые аналитически доказана возможность прогнозирования самоподобного трафика.

4. Обнаружены регулярные периодические составляющие в агрегированном сетевом трафике.

5. Впервые показано, что трафик беспроводных сетей передачи данных также обладает свойством самоподобия. Показана актуальность проблемы самоподобия для современных телекоммуникационных сетей.

Практическая ценность работы и её реализация. Результаты, полученные в ходе выполнения настоящей диссертационной работы, могут быть использованы при разработке алгоритмов функционирования и программного обеспечения узлов телекоммуникационного оборудования с целью повышения качества обслуживания и эффективности обработки трафика, обладающего свойством самоподобия.

Разработанная в диссертации имитационная модель алгоритма динамического распределения пропускной способности с прогнозированием

используется в демонстрационной лабораторной работе по дисциплине “Методы и устройства цифровой обработки сигналов” на кафедре РПУ МЭИ (ТУ). Имеется соответствующий Акт об использовании.

Материалы данной работы вошли в НИОКР по теме: «Сопряжение периферийных земных станций спутниковой связи с абонентскими пунктами информационно-коммуникационной системы», целью которой являлось сокращение затрат на аренду частотно-энергетического ресурса спутника-ретранслятора “Ямал-200” при проектировании и построении Ведомственной Технологической Сети Спутниковой Связи для Министерства Российской Федерации по атомной энергии, о чем свидетельствует соответствующий Акт внедрения.

Апробация работы. Основные результаты диссертационной работы докладывались и обсуждались на IX и X международных научно-технических конференциях студентов и аспирантов “Радиоэлектроника, электротехника, энергетика” в 2003 и 2004 годах, научно-техническом семинаре кафедры РПУ МЭИ в 2003 году, 58-й научной сессии РНТОРЭС им. А.С. Попова в 2003 году и Международной конференции “Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN)” 2004 г.

По теме диссертации автором опубликовано 5 печатных работ.

В **первой главе** на основе анализа известных автору источников излагается современное состояние и главные достижения теории самоподобного телетрафика. Даются определения самоподобного процесса и обсуждаются его основные свойства. Подробно рассматриваются и взаимоотношляются в отношении телетрафика такие понятия как самоподобие, фрактальность и хаос, медленно и быстро убывающие зависимости, продолжительная память, коэффициент Хэрста и фрактальная размерность, распределения с “тяжелыми хвостами”, персистентность и антиперсистентность, до сих пор во многих работах изучаемые отдельно. Данное рассмотрение позволяет с более широких позиций подойти к проблеме

обеспечения качества обслуживания в системах распределения информации при наличии эффекта самоподобия трафика.

Представлены широко используемые алгоритмы управления интенсивностью трафика, такие как шейпинг и полисинг.

С одной стороны, поскольку шейпинг не допускает отбрасывания пакетов, то это делает его привлекательным для задач управления передачей информации реального времени (голос, реальное видео). С другой стороны, он вносит задержки, связанные с буферизацией, что отрицательно сказывается на характеристиках передаваемого трафика. Алгоритм полисинга в отношении высокопачечного трафика также проявляет себя далеко не с лучшей стороны: чтобы достичь приемлемых показателей потерь, необходимо значительно увеличить пропускную способность канала, снизив при этом утилизацию в канале.

Для устранения перечисленных выше недостатков алгоритмов шейпинга и полисинга предлагается реализовать новый алгоритм динамического распределения пропускной способности канала, использующий прогнозирование интенсивности сетевого трафика. Возможность осуществлять прогнозы возникает благодаря свойству продолжительной памяти процессов и теоретически должна обеспечить повышение коэффициента использования канала, качества обслуживания, а значит – и увеличение общей эффективности системы.

Автор полагает, что с помощью данного алгоритма удастся улучшить качество обслуживания для заданного потока, характеризующегося высокой пачечностью и продолжительной памятью, по сравнению со случаем, когда для его обработки используются полисинг или шейпинг.

Во **второй главе** производится подробный анализ реализаций сетевого трафика канального и транспортного уровней модели OSI. Описывается процедура агрегирования (приведения реализаций сетевого трафика к виду, удобному для анализа). Оцениваются основные статистические характеристики

временных рядов, соответствующих агрегированным реализациям трафика, такие как среднее и дисперсия.

Приводятся результаты расчетов плотностей распределения вероятности, автокорреляционных функций, энергетических спектров. С помощью решения задачи регрессии доказывается, что все исследуемые временные ряды обладают свойством длительной памяти.

Отличительной особенностью настоящей диссертационной работы является применение концепций исследования временных рядов, широко используемых теорией нелинейных динамических систем для анализа и идентификации режимов динамического хаоса. В частности, исследования сетевого трафика проводились методом “ближайших ложных соседей”, использовались так называемые “суррогатные” данные; проводились вычисления и анализ корреляционного интеграла и базирующейся на нем BDS-статистики.

В рамках данной диссертационной работы проведено оригинальное исследование особенностей структуры трафика современной беспроводной сети стандарта IEEE 802.11b.

В **третьей главе** рассматриваются свойства самоподобного сетевого трафика, которые обуславливают его прогнозируемость.

Аналитически доказывается принципиальная прогнозируемость временных рядов, обладающих свойством гиперболически убывающей автокорреляционной функции.

Производится постановка задачи динамического управления пропускной способностью канала с помощью прогнозирования.

Изучается и сравнивается эффективность применения различных вариантов управления (алгоритмов прогнозирования) пропускной способности канала на реализациях трафика.

Для вычисления выигрыша, получаемого от применения алгоритма динамического управления с прогнозированием, основные характеристики

эффективности оцениваются относительно аналогичных, полученных для случая с простым статическим заданием пропускной способности при условии, что средняя пропускная способность канала в обоих случаях - одна и та же. При этом по наибольшему выигрышу определяется наиболее подходящий алгоритм прогнозирования.

В **четвертой главе** рассматривается способ реализации метода обеспечения качества обслуживания путем динамического распределения пропускной способности канала с помощью прогнозирования в условиях самоподобия телетрафика.

Разработанный в настоящей диссертации алгоритм функционирования такой схемы базируется на популярном принципе “корзина маркеров” и основанных на нем методов шейпинга и полисинга, в которые внедряется модуль прогнозирования сетевого трафика и управления скоростью поступления маркеров в корзину. При этом не трафик выравнивается под заданный наперед профиль (как в алгоритмах шейпинга и полисинга), а напротив, пропускная способность системы подстраивается под профиль трафика, уменьшая при этом потери и увеличивая использование выделенных ресурсов.

Для проверки полученных в настоящей диссертационной работе результатов с помощью имитационного моделирования на ПЭВМ был поставлен эксперимент по анализу эффективности алгоритма динамического управления пропускной способностью с прогнозированием в условиях самоподобного телетрафика. Моделирование производилось в среде популярного сетевого эмулятора ns-2. Источником самоподобного трафика в данном эксперименте является одна из реализаций реального сетевого трафика, изучаемая в главах 2 и 3 настоящей диссертации.

Полученные в результате моделирования результаты подтверждают выводы, сделанные ранее в главе 3 диссертации, о безусловном повышении эффективности системы благодаря применению метода динамического

распределения пропускной способности с помощью прогнозирования. Величина полученного в результате имитационного моделирования выигрыша от применения метода динамического распределения пропускной способности соответствует оценкам, произведенным в главе 3 диссертации, что подтверждает корректность расчетов.

В Заключение сформулированы основные результаты работы.

Хочу выразить искреннюю благодарность своему научному руководителю к.т.н., проф. Е.А. Богатыреву (МЭИ, Россия) за координирование, помощь в написании и подготовке к защите настоящей диссертации, д.т.н., проф. С.М. Смольскому (МЭИ, Россия) за ценные рекомендации и внимание, проявленное к данной работе. Отдельное спасибо хочется высказать моим коллегам: Виктору Платову, в сотрудничестве с которым была выполнена работа по сбору и анализу трафика беспроводной сети, а также Дмитрию Соколову, который сделал ряд замечаний к первоначальному варианту работы.

Также выражаю особую признательность Н.Б. Лиханову (ИППИ РАН, Россия), М.В. Капранову (МЭИ, Россия), А.С. Дмитриеву (ИРЭ РАН, Россия), Е.А. Кучерявому (TUT, Финляндия), А. Осину (МГУ С, Россия), А. Саенко (Финляндия), В.И. Найденову (ИНВП РАН, Россия), I. Kaplan (США), D. Wischik (Англия), и др. за плодотворные дискуссии, поддержку и понимание.

Глава 1. Современное состояние и основные понятия теории самоподобного телетрафика

Цель настоящей главы – познакомить читателя с основными положениями теории самоподобного телетрафика, рассмотренными с точки зрения проблемы повышения качества обслуживания в современных системах распределения информации при влиянии эффекта самоподобия, а также поставить задачи, которые необходимо решить в данной диссертационной работе.

1.1 Понятие фрактальности

Блох больших кусают блошки,
Блошек тех – малютки-крошки,
Нет конца тем паразитам,
Как говорят, ad infinitum.

Дж. Свифт “Стихотворения”, II, 651 (1733)

Понятие *фрактал* было впервые введено Бенуа Мандельбротом в 1975 году. Слово образовано от латинского *fractus* – состоящий из фрагментов. С математической точки зрения фрактальный объект, прежде всего, обладает дробной (нецелой) размерностью.

Известно, что точка (с математической точки зрения) имеет размерность, равную нулю, отрезок прямой и окружность (характеризующиеся длиной) – единице, круг и сфера (характеризующиеся площадью) – двум, и т.д. Однако, что представляет собой множество объектов с размерностью, скажем, 1.5? Видимо, для его описания требуется нечто среднее между длиной и площадью. Еще раньше, в 1919 году, Ф. Хаусдорф привел первые примеры объектов, обладающих дробной размерностью (Канторово множество, кривая фон Коха и пр.).

Другое важное свойство, которым обладают почти все фракталы – свойство *самоподобия* (масштабная инвариантность). Оказывается, фрактал можно разбить на сколь угодно малые части так, что каждая часть окажется просто уменьшенной частью целого. Другими словами, если посмотреть на

фрактал в микроскоп, то мы увидим ту же самую картинку, что и без микроскопа! Примером природного фрактального объекта является представленный на рис. 1.1 лист папоротника¹.



Рис. 1.1. Пример фрактального объекта в природе – лист папоротника

Природа создавала фракталы на протяжении миллионов лет. Фактически большинство объектов в природе при более тщательном рассмотрении не являются кругами, квадратами или простыми линиями. Вместо этого они, по существу, – фракталы, и происхождение фракталов обычно связывается с уравнениями *хаоса*. Хаос и фрактальная красота представляют природу реальности. Одним из таких самоподобных (фрактальных) процессов является телетрафик. Поэтому применение аппарата теории нелинейных динамических процессов (в частности, теории хаоса) для исследования самоподобного телетрафика представляется также достаточно перспективным направлением и разумным развитием идей фрактального исследования трафика. В последнее время работ в данной области появляется все больше [16],[18],[19]. Заметим, что часто используемый в литературе термин *хаос* подразумевает под собой

¹ Конечно, природные фракталы, как правило, не являются строгими геометрическими фракталами в смысле точного соответствия частей объекта уменьшенной копии целого, однако в этом случае можно говорить о “приближенном” соответствии, подобии форм, структуры или характеристик при масштабировании.

словосочетание *детерминированный хаос*, чтобы подчеркнуть отличие такого процесса от чисто случайного, однако в разговорной речи слово “детерминированный” для краткости часто опускается. В этом смысле принцип детерминированности потенциально может играть значительную роль не только в сетевом трафике, но и во многих аналогичных процессах, кажущихся на первый взгляд случайными.

В отличие от детерминированных фракталов стохастические фрактальные объекты (процессы), как правило, описываются масштабной инвариантностью (самоподобием) статистических характеристик второго порядка (свойство неизменности коэффициента корреляции при масштабировании). Как раз с такими стохастическими фракталами мы столкнемся ниже при изучении характеристик сетевого трафика. В этой связи в литературе понятия фрактального и самоподобного телетрафика часто используются как синонимы.

1.2 Самоподобный (фрактальный) телетрафик

1.2.1 Проблема самоподобного телетрафика

Впервые о самоподобном телетрафике заговорили с момента его обнаружения в 1993 году группой ученых (W.Leland, M.Taqqu, W.Willinger и D.Wilson [1]), которые исследовали Ethernet-трафик в сети корпорации Bellcore и обнаружили, что он обладает свойством самоподобия, т. е. выглядит качественно одинаково при почти любых масштабах временной оси. При этом оказалось, что в условиях самоподобного трафика методы расчета современных компьютерных сетей (пропускной способности каналов, емкости буферов и пр.), основанные на пуассоновских моделях и формулах Эрланга, которые с успехом используются при проектировании телефонных сетей, дают неоправданно оптимистические решения и приводят к недооценке реальной нагрузки. Различие между компьютерной и телефонной сетями здесь следует понимать в следующем смысле: так уж исторически сложилось, что телефонные сети изначально строились по

принципу коммутации каналов. Характеристики трафика в данных сетях хорошо изучены, а также разработаны строгие методики расчетов. В основу компьютерных сетей, как правило, был положен принцип коммутации пакетов, а методики расчетов, возможно, вследствие некоторого отставания теоретической базы от бурно развивающихся технологий остались практически теми же, что и привело к возникновению “проблемы самоподобия”. Кроме того, в настоящее время все большее распространение получают способы передачи речевой информации по сетям с коммутацией пакетов VoIP, OKC№7 [14], трафик которых также является самоподобным.

1.2.2 Определения самоподобного процесса

Дадим определения строго и асимптотически самоподобных в широком смысле случайных процессов дискретного аргумента и укажем их связь с процессами, самоподобными в узком смысле, и с процессами с медленно убывающей зависимостью. Следует заметить, что теория самоподобного телетрафика проходит относительно раннюю стадию своего развития, по этой причине существуют некоторые различия в терминологии и даже в определениях.

Обозначения. Пусть $\mathbf{X}=(X_1, X_2, \dots)$ - полубесконечный отрезок стационарного в широком смысле случайного процесса дискретного аргумента (времени) $t \in \mathbb{N}^{\Delta}=\{1,2,\dots\}$. Обозначим через $\mu < \infty$ и $\sigma^2 < \infty$ среднее и дисперсию процесса \mathbf{X} соответственно, а через

$$r(k) \stackrel{\Delta}{=} \frac{\overline{(X_{t+k} - \mu)(X_t - \mu)}}{\sigma^2},$$

$$b(k) \stackrel{\Delta}{=} \sigma^2 r(k), \quad k \in \mathbb{Z}_+^{\Delta} = \{0,1,2,\dots\}$$

автокорреляционную функцию и автоковариацию процесса \mathbf{X} . Так как процесс \mathbf{X} – стационарный в широком смысле, среднее $M[X] = \mu$, дисперсия

$D[X] = \sigma^2 \equiv b(0)$, коэффициент корреляции $r(k)$ и автоковариация $b(k)$ не зависят от времени t и $r(k)=r(-k)$, $b(k)=b(-k)$.

Допустим, процесс X имеет автокорреляционную функцию следующего вида:

$$r(k) \sim k^{-\beta} L_1(k), k \rightarrow \infty, \quad (1.1)$$

где $0 < \beta < 1$ и L_1 – медленно меняющаяся на бесконечности функция, то есть

$$\lim_{t \rightarrow \infty} \frac{L_1(tx)}{L_1(t)} = 1 \text{ для всех } x > 0 \text{ (примерами медленно меняющейся функции}$$

могут служить $L_1(t)=\text{const}$, $L_1(t)=\log(t)$).

Обозначим через $X^{(m)} = (X_1^{(m)}, X_2^{(m)}, \dots)$ усредненный по блокам длины m процесс X , компоненты которого определяются равенством

$$X_t^{(m)} \triangleq \frac{1}{m} (X_{tm-m+1} + \dots + X_{tm}), m, t \in \mathbb{N}. \quad (1.2)$$

В дальнейшем изложении, следуя [9], будем называть такой ряд *агрегированным*. Обозначим через $r_m(k)$, $b_m(k)$ и $V_m=b_m(0)$ коэффициент корреляции, автоковариацию и дисперсию процесса $X^{(m)}$ соответственно. Приведем ниже определение строго самоподобного в широком смысле процесса.

Определение. Процесс X называется *строго самоподобным в широком смысле* [ССШС] (exactly second-order self-similar) с параметром $H=1-(\beta/2)$, $0 < \beta < 1$, если

$$r_m(k)=r(k), k \in \mathbb{Z}_+, m \in \{2,3,\dots\}, \quad (1.3)$$

т. е. ССШС процесс не меняет свой коэффициент корреляции после усреднения по блокам длины m [4]. Другими словами [1], X является ССШС, если агрегированный процесс $X^{(m)}$ неотличим от исходного процесса X , как минимум в отношении статистических характеристик второго порядка. Параметр H , называемый *коэффициент Хэрста* [Hurst parameter], имеет принципиальное значение в теории самоподобных процессов. Он является

индикатором степени самоподобия процесса, а также свидетельствует о наличии у него таких свойств как персистентность/антиперсистентность и продолжительная память. Более подробно мы рассмотрим этот параметр в разделе 1.3.2 настоящей главы.

Определение. Процесс X называется *асимптотически самоподобным в широком смысле* (АСШС) [second-order asymptotical self-similarity] [2] с параметром $H=1-(\beta/2)$, $0<\beta<1$, если

$$\lim_{m \rightarrow \infty} r_m(k) = g(k), k \in N, \quad (1.4)$$

где $g(k) = \frac{1}{2}[(k-1)^{2-\beta} - 2k^{2-\beta} + (k+1)^{2-\beta}]$ - коэффициент корреляции ССШС [2].

Смысл этого определения состоит в том, что X является АСШС процессом, если после усреднения по блокам длины m и при $m \rightarrow \infty$ он сходится к ССШС процессу.

Вместе с понятием ССШС существует понятие просто самоподобного процесса, которое для большего терминологического различия мы будем называть самоподобным в узком смысле процессом (СУС).

Определение. Процесс X называется *самоподобным в узком смысле* (СУС) [strictly self-similarity] с параметром $H=1-(\beta/2)$, $0<\beta<1$, если справедливо выражение

$$m^{1-H} X^{(m)} \stackrel{\bullet}{=} X, m \in N, \quad (1.5)$$

которое понимается в смысле равенства распределений. Связь между процессами ССШС и СУС аналогична связи между процессами, стационарными в широком и узком смыслах. Необходимое и достаточное условие того, что процесс называется ССШС, дано в [4].

Возвращаясь к работе [1], укажем, что исследователи отметили одинаковость распределений исходного и агрегированного процессов (при значительном интервале изменения m), измерили параметр H (параметр Хэрста) и обнаружили, что последний для сетевого трафика находится в интервале $(0.5, 1)$. На качественном уровне такой самоподобный трафик

имеет постоянный “взрывной” характер [burstiness], то есть обладает высокой пачечностью на многих масштабах временной оси. Напомним, что *коэффициент пачечности (пачечность)* для заданного процесса соответствует отношению пиковой интенсивности процесса поступления заявок на обслуживание к его среднему значению.

1.3 Основные свойства самоподобных процессов

Наиболее интересная черта самоподобных процессов – медленное убывание автокорреляционной функции агрегированного процесса $X^{(m)}$ при $m \rightarrow \infty$ в отличие от распространенных стохастических моделей, для которых выполняется (1.6):

$$r_m(k) \rightarrow 0, m \rightarrow \infty, k \in \mathbb{N}. \quad (1.6)$$

В 1984 году Сох показал, что задание АКФ в виде (1.1) или задание спектральной плотности в виде (1.17) (см. ниже) равносильно определению процесса со следующей дисперсией:

$$\sigma^2 \sim am^{-\beta}, m \rightarrow \infty, \quad (1.7)$$

где a – конечная положительная константа, не зависящая от m , и $0 < \beta < 1$.

С другой стороны, для процессов, удовлетворяющих (1.6), можно показать выполнение

$$\sigma^2 \sim bm^{-1}, m \rightarrow \infty, \quad (1.8)$$

где b – конечная положительная константа, не зависящая от m .

На качественном уровне в соответствии со свойствами (1.7) и (1.8) можно предположить, что самоподобный процесс при достаточно больших m выглядит менее сглаженным, более неравномерным, (т.е. обладает большей дисперсией) чем процесс, удовлетворяющий (1.6). Действительно, достаточно взглянуть на рис.1.2, чтобы в этом убедиться.

Следует заметить, что процесс X не может быть самоподобным в узком смысле, если его среднее значение не равно нулю. Действительно, в силу (1.5)

$$M[X] = m^{1-H} M[X^{(m)}]. \quad (1.9)$$

Можно показать, что в силу (1.2) $M[X^{(m)}] = M[X]$, тогда (1.9) выполняется только в случае $M[X] = 0$.

Поведение процесса X при агрегировании (1.2) имеет принципиальное значение, поскольку некоторые механизмы в теории систем распределения информации, например, распределение пропускной способности, часто рассматриваются применительно к агрегированным процессам. Чтобы представлять особенности, которые могут проявляться у агрегированных процессов, приведем простой пример [7].

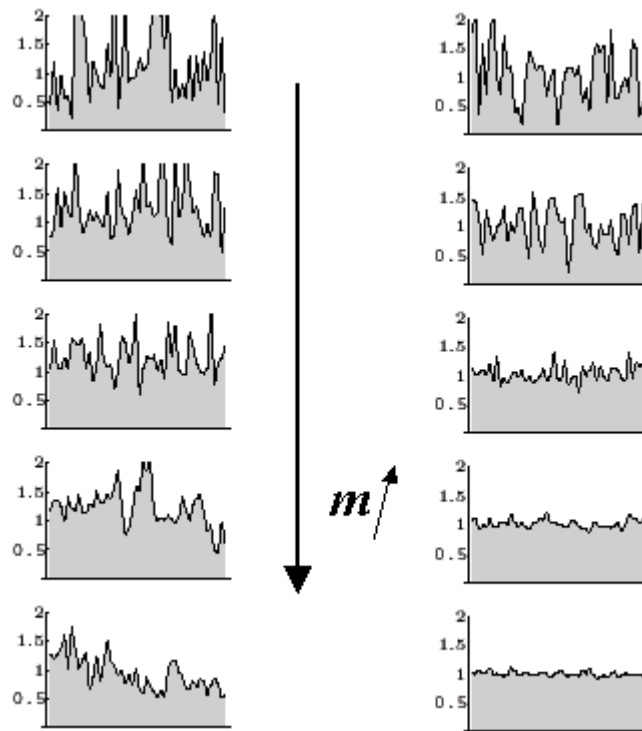


Рис. 1.2. Временные реализации реального сетевого (самоподобного) трафика (слева) и традиционной несамоподобной (Пуассоновской) модели телетрафика (справа) при различных масштабах временной оси (из D.Wischik. Implication of long-range dependence, 2001). Сверху вниз масштаб временной оси укрупняется

Пусть приращения ε (т.е. первые разности) временного ряда X являются случайными. Тогда первые m членов ряда можно записать:

$$X_1 = X_1,$$

$$X_2 = X_1 + \varepsilon_1,$$

$$X_3 = X_1 + \varepsilon_1 + \varepsilon_2,$$

$$\dots$$

$$X_m = X_1 + \varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_{m-1}$$

а второй набор из m членов – как

$$\left. \begin{aligned} X_{m+1} &= X_1 + \varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_{m-1} + \varepsilon_m \\ X_{m+2} &= X_1 + \varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_{m-1} + \varepsilon_m + \varepsilon_{m+1} \\ X_{m+3} &= X_1 + \varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_{m-1} + \varepsilon_m + \varepsilon_{m+1} + \varepsilon_{m+2} \\ &\dots \\ X_{2m} &= X_1 + \varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_{2m-1} \end{aligned} \right\} \quad (1.10)$$

Если вычислить разность между средним второго и первого наборов, то получим

$$d = (1/m) (\varepsilon_1 + 2\varepsilon_2 + 3\varepsilon_3 + \dots + (m-1)\varepsilon_{m-1} + m\varepsilon_m + (m-1)\varepsilon_{m+1} + \dots + \varepsilon_{2m-1}). \quad (1.11)$$

Заметим, что d представляет собой первую разность агрегированного ряда, определенного в соответствии с (1.2). Найдем дисперсию временного ряда d :

$$\begin{aligned} D[d] &= D[\varepsilon] \cdot \frac{1}{m^2} (1 + 2^2 + 3^2 + \dots + (m-1)^2 + m^2 + (m-1)^2 + \dots + 3^2 + 2^2 + 1) = \\ &= D[\varepsilon] \cdot \frac{1}{m^2} (2 \cdot \sum_{j=1}^{m-1} j^2 + m^2). \end{aligned} \quad (1.12)$$

Найдем ковариацию двух последовательных разностей агрегированного ряда. Для этого умножим выражение (1.11) на аналогичное выражение, сдвинутое во времени на m , и вычислим среднее:

$$b_m = D[\varepsilon] \cdot \frac{1}{m^2} \left(\sum_{j=1}^{m-1} j(j-1) + m(m-1) \right) = \frac{m^2 - 1}{6m} \cdot D[\varepsilon], \quad (1.13)$$

тогда корреляция $R(m)$ между последовательными разностями агрегированного ряда равна

$$R(m) = \frac{m^2 - 1}{2(2m^2 + 1)}. \quad (1.14)$$

На рис. 1.3. изображена зависимость (1.14) от m .

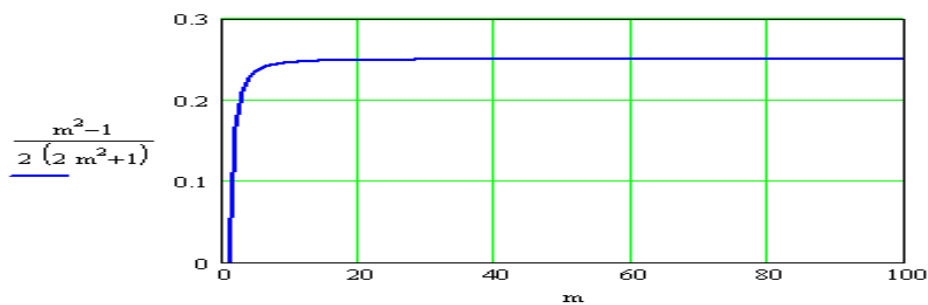


Рис. 1.3. Зависимость корреляции между соседними первыми разностями агрегированного ряда от значения m

Таким образом, несмотря на то, что первые разности исходного ряда независимы, первые разности агрегированного ряда не являются таковыми.

1.3.1 Медленно и быстро убывающие зависимости, продолжительная память

Понятие *медленно убывающей зависимости*¹ (автоковариации) имеет ключевое значение в теории самоподобных процессов и фактически описывает интересное в отношении прогнозирования свойство – продолжительную память [26], [28]. К сожалению, несмотря на долгие поиски, автору не встретилось строгого определения объекта с продолжительной памятью, однако на интуитивном уровне данное свойство можно объяснить следующим образом: будущее процесса определяется его прошлым, причем с убывающей степенью влияния по мере того, как прошлое удалено от настоящего. Таким образом, процесс с продолжительной памятью хорошо “помнит” свое недавнее прошлое, но как бы “постепенно забывает” свои давно минувшие состояния по мере продвижения времени в будущее.

Дадим несколько определений.

Определение. Говорят [1], [2], [4], что процесс X обладает *медленно убывающей зависимостью* (МУЗ)² [long-range dependence], если

¹ В литературе [13] можно встретить и другое обозначение этого понятия – “долговременная зависимость” (а также термин “кратковременная зависимость” в отношении такого понятия как “быстро убывающая зависимость”).

² Сокращение “МУЗ” введено Цыбаковым Б.С. в [2]. В этой же работе используется сокращение “БУЗ” для обозначения быстро убывающих зависимостей.

выполняется (1.1). Таким образом, процессы с МУЗ характеризуются автокорреляционной функцией, которая убывает гиперболически (по степенному закону) при увеличении временной задержки (лага) (см. рис. 1.4).

Более того, можно показать, что из (1.1) следует несуммируемость АКФ, то есть

$$\sum_k r(k) = \infty. \quad (1.15)$$

В отличие от процессов с МУЗ процессы с *быстро убывающей зависимостью* (БУЗ) [short-range dependence] обладают экспоненциально спадающей АКФ вида

$$r(k) \sim \rho^k, \quad k \rightarrow \infty, \quad 0 < \rho < 1 \quad (1.16)$$

и, как следствие, суммируемостью АКФ: $0 < \sum_k r(k) < \infty$.

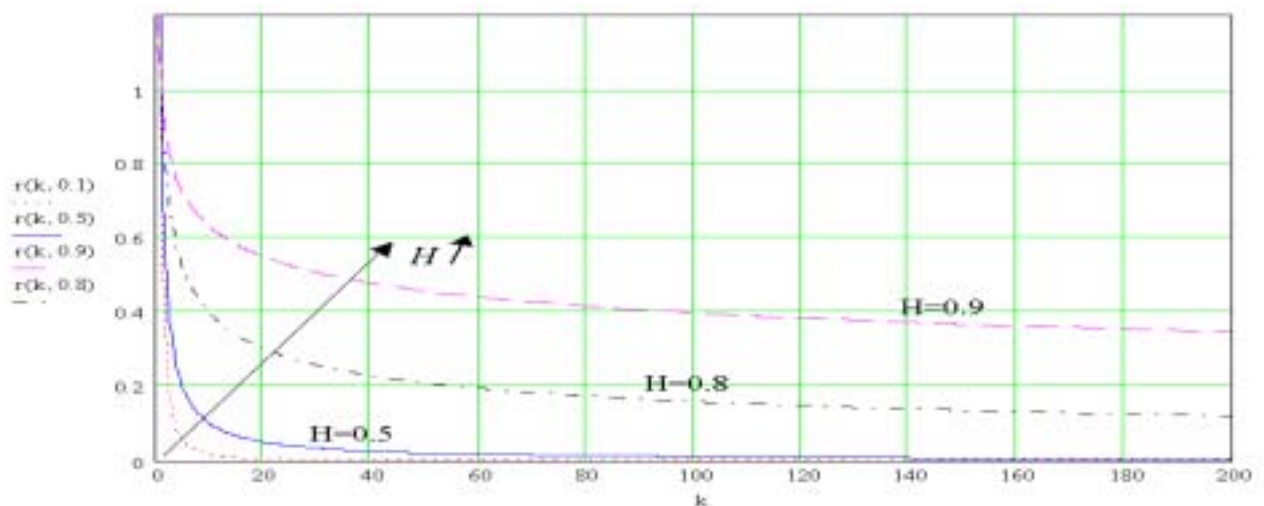


Рис. 1.4. Степенной закон убывания АКФ согласно (1.1) при различных значениях параметра $H=1-(\beta/2)$

В частотной области МУЗ отражается на характерном степенном законе поведения спектральной плотности рассматриваемого процесса. Действительно, эквивалентно (1.1) можно констатировать, что процесс X обладает МУЗ, если

$$f(\lambda) \sim \lambda^{\beta-1} L_2(\lambda), \quad \lambda \rightarrow 0, \quad 0 < \beta < 1. \quad (1.17)$$

Здесь L_2 – медленно изменяющаяся в нуле функция, $f(\lambda) = \sum_k r(k) e^{ik\lambda}$ – означает спектральную плотность. Таким образом, с точки зрения

спектрального анализа процесс с МУЗ (кроме того, что имеет не суммируемую АКФ) обладает спектральной плотностью с особенностью в нуле (т.е. спектральная плотность $f(\lambda)$ такого процесса стремится к бесконечности, по мере того как частота λ стремится к нулю). Такой процесс часто называют “фликкер-шумом”.

Важно отметить, что согласно приведенной в [4] теореме МУЗ влечет за собой АСШС.

1.3.2 Понятие коэффициента Хэрста

Одна старуха от чрезмерного любопытства вывалилась из окна, упала и разбилась. Из окна высунулась другая старуха и стала смотреть на разбившуюся, но, от чрезмерного любопытства, тоже вывалилась из окна, упала и разбилась. Потом из окна вывалилась третья старуха, потом четвертая, потом пятая...

Д. Хармс. “Вываливающиеся старухи”

Гарольд Эдвин Хэрст, будучи британским чиновником в Каире (1906 год), участвовал в гидротехнических проектах на реке Нил. Следует заметить, что на протяжении столетий ежегодные разливы Нила были основой сельского хозяйства многих известных цивилизаций Африки. Хорошее орошение означало хороший урожай, в то время как малая вода приводила к недороду и нехватке продовольствия. Кроме того, ежегодно в долине реки на всей затопляемой территории откладывается плодородный слой красного аллювия, который придает красноватый оттенок водам Нила. Просмотрев летописи за 800 лет на предмет разливов Нила, Хэрст обнаружил, что существовала тенденция, когда за годом хорошего наводнения следовал еще один плодородный год, и, наоборот, за годом малой воды следовал еще один “голодный” год¹. Другими словами, казалось, что появление голодных и плодородных лет – не случайно. Для

¹ Позже Мандельброт для описания подобных явлений использовал термин “эффект Иосифа” по аналогии с семью годами Египетского изобилия, перемежающихся с семью годами голода в библейской истории Иосифа.

подтверждения данного факта, Хэрст ввел коэффициент $0 < H < 1$, который в его честь сейчас называется коэффициентом (экспонентой) Хэрста. В случае независимости друг от друга уровней ежегодных разливов, логично было бы представить процесс разливов обычным броуновским движением с независимыми приращениями, которое было рассмотрено ранее, при этом коэффициент Хэрста $H=0.5$. Однако, как обнаружил Хэрст, для Нила - $H=0.7$.

Одним из способов вычисления коэффициента H является анализ так называемой R/S статистики (нормированного размаха). Обозначим ξ – ежегодный уровень воды в Ниле, тогда средний уровень воды за τ лет:

$$M[\xi] = \frac{1}{\tau} \sum_{i=1}^{\tau} \xi_i. \quad (1.18)$$

Получим новый (кумулятивный) ряд, представляющий сумму за время t ежегодных колебаний уровня Нила относительно среднего $M[\xi]$:

$$X(t, \tau) = \sum_{i=1}^t (\xi_i - M[\xi]), \quad 1 \leq t \leq \tau. \quad (1.19)$$

При этом диапазон между максимальным и минимальным значением $X(t, \tau)$ за время τ обозначается $R(\tau)$:

$$R(\tau) = \max(X(t, \tau)) - \min(X(t, \tau)), \quad 1 \leq t \leq \tau. \quad (1.20)$$

Тогда R/S статистика определяется безразмерным отношением диапазона $R(\tau)$ к стандартному отклонению S :

$$R/S = \frac{R(\tau)}{S(\tau)} = \frac{R(\tau)}{\sqrt{\frac{1}{\tau} \sum_{i=1}^{\tau} (\xi_i - M[\xi])^2}}. \quad (1.21)$$

Хэрст показал, что для многих природных явлений справедлива зависимость:

$$M\left[\frac{R(\tau)}{S(\tau)}\right] \sim c\tau^H, \quad \tau \rightarrow \infty, \quad (1.22)$$

где c – положительная константа, не зависящая от τ .

В частности, если приращения исследуемого временного ряда (1.20) независимы, т.е. ряд представляет собой БД с независимыми приращениями, то, как уже упоминалось ранее, коэффициент Хэрста в (1.22) $H=0.5$. Однако

Хэрст обнаружил для Нила $H \approx 0.7$, что подтверждало присутствие некоторой зависимости между последовательными отсчетами ξ_i и ξ_{i+1} !

Следует заметить, что в случае $0.5 < H < 1$ говорят о **персистентном** (поддерживающемся) поведении процесса, либо о том, что процесс обладает **длительной памятью**. Другими словами, если в течение некоторого времени в прошлом наблюдались положительные приращения процесса, то есть происходило увеличение, то и впредь в среднем будет происходить увеличение. Иначе говоря, вероятность того, что процесс на $i+1$ шаге отклоняется от среднего в том же направлении, что и на i шаге настолько велика, насколько параметр H близок к 1. Таким образом, персистентные стохастические процессы обнаруживают четко выраженные тенденции изменения при относительно малом “шуме”.

В случае $0 < H < 0.5$ говорят об **антиперсистентности** процесса. Здесь высокие значения процесса следуют за низкими, и наоборот. Другими словами, вероятность того, что на $i+1$ шаге процесс отклоняется от среднего в противоположном направлении (по отношению к отклонению на i шаге) настолько велика, насколько параметр H близок к 0.

При $H=0.5$ отклонения процесса от среднего являются действительно случайными и не зависят от предыдущих значений, что соответствует случаю БД.

Заметим, что именно свойство персистентности оправдывает применение для моделирования и предсказания самоподобных рядов AR (авторегрессионных) моделей вида

$$X_n = \varphi_0 + \sum_{r=1}^p \varphi_r \cdot X_{n-r} + \varepsilon_n, \quad (1.23)$$

где φ_i – константы, ε_n – некоррелированные случайные переменные (белый шум) с нулевым средним. Выражение (1.23) показывает, как, зная прошлое процесса, предсказать его будущее. В частности, получили широкое распространение такие авторегрессионные модели, как ARMA (процесс скользящего среднего), ARIMA (интегральный процесс скользящего

среднего) и FARIMA (фрактальный интегральный процесс скользящего среднего).

Свойство длительной памяти характерно для половодий Нила [12], сетевого трафика [1], процессов, происходящих на финансовых рынках [11], [15], и пр.

Соотношение (1.22) может использоваться для оценки коэффициента H по временному ряду. Для этого необходимо прологарифмировать обе части (1.22)

$$\log\left(M\left[\frac{R(\tau)}{S(\tau)}\right]\right) \sim H \log(\tau) + \log c, \quad \tau \rightarrow \infty \quad (1.24)$$

и построить график зависимости $\log\left(M\left[\frac{R(\tau)}{S(\tau)}\right]\right)$ от $\log(\tau)$. Наклон прямой, аппроксимирующей данную зависимость, и есть коэффициент Хэрста, характеризующий исследуемый временной ряд.

Для оценки параметра Хэрста временного ряда существует множество методов: R/S статистика, анализ графика изменения дисперсии (согласно (1.7)), оценка Виттла, вейвлет-анализ, анализ индекса дисперсии и пр. С ними можно познакомиться, например, в [13].

1.3.3 Понятие фрактальной размерности и ее связь с коэффициентом Хэрста

Никто не может изгнать нас из рая,
который создал нам Кантор.

Д. Гильберт

Как уже упоминалось, сетевой трафик относится к классу фрактальных процессов, обладающих свойством статистического самоподобия характеристик второго порядка, и, следовательно, может характеризоваться ***фрактальной размерностью***. Дадим строгое определение этому понятию, следуя [9].

Известно, что длина кривой линии L_0 в евклидовом пространстве определяется числом $N(\delta)$ отрезков длины δ прямой, необходимых для покрытия ее полностью. Мера L длины кривой можно ввести с помощью следующего соотношения:

$$L = \lim_{\delta \rightarrow 0} N(\delta) \cdot \delta = \frac{L_0}{\delta} \cdot \delta = L_0. \quad (1.25)$$

Как следует из (1.25), в пределе при уменьшении δ мера L становится равной длине кривой L_0 , и при этом ее значение не зависит от δ . Формально, множеству точек, соответствующих рассматриваемой кривой, можно сопоставить другой измерительный элемент, например квадрат со стороной δ . Площадь такого измерительного элемента равна δ^2 . Если для покрытия кривой требуется $N(\delta)$ квадратов, то результирующая мера (“площадь”) кривой будет равна

$$S = \lim_{\delta \rightarrow 0} N(\delta) \cdot \delta^2 = \lim_{\delta \rightarrow 0} \frac{L_0}{\delta} \cdot \delta^2 = \lim_{\delta \rightarrow 0} L_0 \delta. \quad (1.26)$$

При $\delta \rightarrow 0$ величина S стремится к нулю. Поэтому, с практической точки зрения, единственной представляющей интерес мерой кривой является ее длина L , а не “площадь” S .

Аналогичным образом, множеству точек, образующих площадь поверхности S_0 , можно сопоставить меру площади

$$S = \lim_{\delta \rightarrow 0} N(\delta) \cdot \delta^2 = \lim_{\delta \rightarrow 0} \frac{S_0}{\delta^2} \cdot \delta^2 = S_0. \quad (1.27)$$

С другой стороны, формально евклидовой поверхности S_0 можно также поставить в соответствие какую-либо длину кривой, покрывающую эту поверхность

$$L = \lim_{\delta \rightarrow 0} N(\delta) \cdot \delta = \lim_{\delta \rightarrow 0} \frac{S_0}{\delta^2} \cdot \delta = \lim_{\delta \rightarrow 0} \frac{S_0}{\delta}. \quad (1.28)$$

Однако, при $\delta \rightarrow 0$ такая мера поверхности расходится, т.е. для покрытия поверхности требуется кривая бесконечной длины.

Для общего случая меры можно записать

$$M = \lim_{\delta \rightarrow 0} N(\delta) \cdot \delta^\beta = \lim_{\delta \rightarrow 0} \frac{M_0}{\delta^D} \cdot \delta^\beta = \lim_{\delta \rightarrow 0} M_0 \cdot \delta^{\beta-D} \xrightarrow{\delta \rightarrow 0} \begin{cases} 0, \beta > D \\ \infty, \beta < D \end{cases}, \quad (1.29)$$

где β – размерность выбранной меры.

Величина $\beta=D$, при которой M меняется скачком, рассматривается как **размерность Хаусдорфа**. Если β – нецелое число, то соответствующая мера называется **фрактальной**.

Простейшим фрактальным и самоподобным объектом, который используется в качестве модели сложных процессов, является **множество Кантора (или Канторова пыль)**. Образующий элемент такого множества можно выбрать в виде единичного отрезка. Разделив этот отрезок на три части и отбросив среднюю часть, получим два отрезка длины $\zeta=1/3$. Применяя многократно такую процедуру после k -го этапа разбиения, получим $N=2^k$ отрезков, длиной $\delta=(1/3)^k$ (см. рис. 1.5).



Рис. 1.5. Тριάдное Канторово множество

Мера полученного в результате такой процедуры множества может быть вычислена с использованием (1.29):

$$M = \lim_{\delta \rightarrow 0} N(\delta) \cdot \delta^\beta = \lim_{\delta \rightarrow 0} \frac{M_0}{\delta^D} \cdot \delta^\beta = \lim_{\delta \rightarrow 0} M_0 \cdot \delta^{\beta-D} \xrightarrow{\delta \rightarrow 0} \begin{cases} 0, \beta > D \\ \infty, \beta < D \end{cases}; \quad (1.30)$$

$$M = \lim_{\delta \rightarrow 0} N(\delta) \cdot \delta^\beta \xrightarrow{\delta \rightarrow 0} \lim_{k \rightarrow \infty} 2^k (1/3)^{k \cdot \beta}. \quad (1.31)$$

Такая мера не расходится и не стремится к нулю, если $2^k(1/3)^{k\cdot\beta} = 1$, отсюда $D = \beta = \frac{\ln 2}{\ln 3} \approx 0.631$. Хотя топологическая размерность Канторова множества равна нулю, его размерность Хаусдорфа – дробная, следовательно, множество Кантора является фрактальным.

В теоретическом аспекте может быть интересным применение множества Кантора при моделировании сети с коммутацией пакетов как системы с потерями [9], [10].

Фрактальная размерность D может рассматриваться как мера неровности, изрезанности поверхности объекта при $D \in [n, n + 1)$ для поверхности в n -размерном пространстве, причем более неровные, “шероховатые” поверхности соответствуют более высоким значениям D . Эффект Хэрста характеризует длительную память, или персистентность процесса. В принципе, фрактальная размерность и коэффициент Хэрста могут рассматриваться независимо друг от друга: фрактальная размерность характеризует локальное свойство, а длительная память – глобальная характеристика всего процесса. Тем не менее, в научной литературе эти понятия тесно связаны, главным образом, благодаря фрактальному гауссовскому шуму и фрактальному броуновскому движению, которые проявляют оба свойства.

Итак, для самоподобного процесса локальные свойства отражаются на глобальных в соответствии с известным соотношением

$$D = n + 1 - H \quad (1.32)$$

между фрактальной размерностью и коэффициентом Хэрста для самоподобного объекта в n -мерном пространстве [6]. Для временного ряда $n=1$, и, соответственно,

$$D = 2 - H \quad (1.33)$$

Поэтому свойство медленно убывающей зависимости, ассоциирующейся с $0.5 < H < 1$, соответствует низкой фрактальной размерности. Случай “шероховатой” поверхности, обладающей высокой фрактальной

размерностью, соответствуют *антиперсистентному* процессу с $0 < H < 0.5$ (см. рис. 1.6).

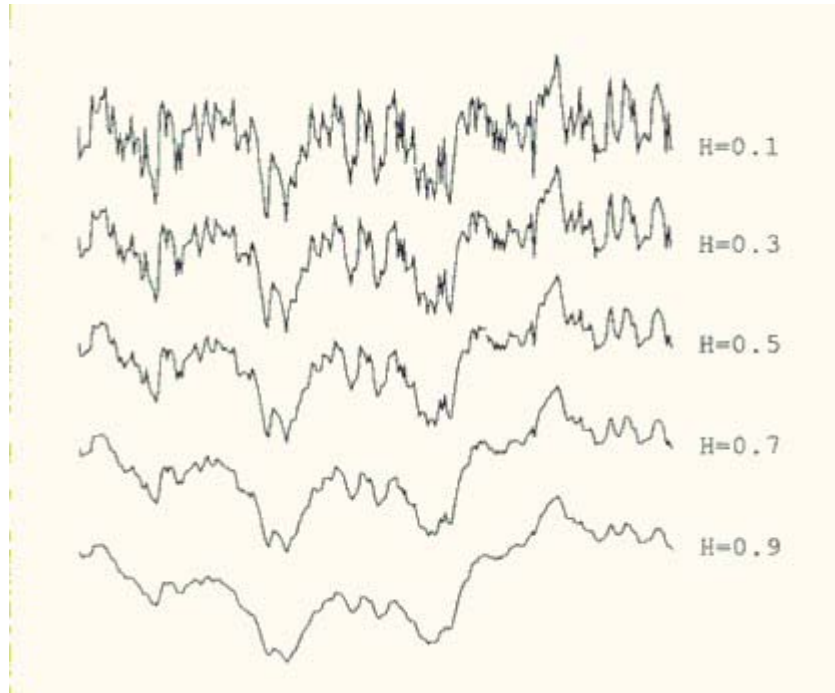


Рис. 1.6. Характерные реализации временной последовательности для различных значений параметра Хэрста (из “*Algorithms for random fractals*”, Dietmar Saupe, Chapter 2 of *The Science of Fractal Images* by Barnsley et al, Springer-Verlag, 1988)

Рассмотрим в качестве примера стационарный в широком смысле случайный процесс X с нулевым средним и дисперсией, равной единице. Поведение корреляционной функции данного процесса характеризует локальные свойства реализации, т.е. если выполняется соотношение

$$1 - r(k) \sim |k|^\alpha, \text{ при } k \rightarrow 0, \quad (1.34)$$

где $\alpha \in (0, 2]$, то реализация случайного процесса имеет фрактальную размерность

$$D = n + 1 - \frac{\alpha}{2}. \quad (1.35)$$

Если процесс X при этом обладает персистентностью (или МУЗ), то он характеризуется ранее рассмотренным поведением корреляционной функции

$$r(k) \sim |k|^{-\beta}, \text{ при } |k| \rightarrow \infty, \quad (1.36)$$

где $\beta \in (0, 2)$, с учетом взаимосвязи с коэффициентом Хэрста

$$H = 1 - \frac{\beta}{2} \text{ и } \beta = 2 - 2H. \quad (1.37)$$

Для фрактального гауссовского шума – классического стационарного самоподобного стохастического процесса, рассмотренного ранее, корреляционная функция которого удовлетворяет (1.38):

$$r(k) = \frac{1}{2} [|k+1|^{2H} - |2k|^{2H} + |k-1|^{2H}], \quad k \in \mathbb{N}, \quad (1.38)$$

можно показать, что

$$r(k) \sim H(2H-1) |k|^{-(2-2H)}, \quad |k| \rightarrow \infty \quad (1.39)$$

и

$$1 - r(k) \sim |k|^{2H}, \quad |k| \rightarrow 0. \quad (1.40)$$

Сравнивая (1.34) и (1.40), заключаем, что соотношение (1.35) выполняется при $n=1$.

Таким образом, предположение самоподобия процесса X (в условиях отождествления его с ФГШ) определяет взаимосвязь между фрактальной размерностью и коэффициентом Хэрста (или локальным и глобальным поведением) процесса. Вместе с тем, для несамоподобных процессов соотношение (1.35) может не выполняться и, следовательно, параметры H и D могут в таком случае меняться независимо [6].

1.3.4 Распределения с “тяжелыми хвостами”

Многочисленные измерения сетевого трафика показали, что он лучше всего описывается так называемыми распределениями с “тяжелыми хвостами” (РТХ). Для начала дадим несколько определений и рассмотрим наиболее характерные случаи.

Определение. Считается, что случайная переменная Z имеет **распределение с тяжелым хвостом**, если

$$P[Z > x] \sim c \cdot x^{-a}, \quad x \rightarrow \infty, \quad (1.41)$$

где $0 < a < 2$ и называется параметром формы, c – некоторая положительная константа. В отличие от распределений с легкими хвостами (РЛХ), такими как экспоненциальное или гауссовское, которые имеют экспоненциальное убывание хвоста, РТХ обладают хвостами, спадающими по степенному (гиперболическому) закону. При $0 < a < 2$ РТХ имеют бесконечную дисперсию, а при $0 < a \leq 1$ еще и обладают бесконечным средним. В случае сети особо интересным является случай $1 < a < 2$.

Среди РТХ наиболее часто используется распределение Парето с функцией распределения:

$$P[Z \leq x] = 1 - \left(\frac{b}{x}\right)^a, \quad b \leq x. \quad (1.42)$$

Основное свойство случайной величины, распределенной в соответствии с РТХ, состоит в том, что она проявляет высокую изменчивость. Иными словами, выборка из РТХ представляет собой большей частью относительно небольшие значения, однако также содержит и достаточное количество очень больших значений. Не удивительно, что при $a \rightarrow 1$ влияние тяжелого хвоста отражается на выборке спадом сходимости выборочного среднего к математическому ожиданию. Например, при размере выборки m выборочное среднее \overline{Z}_m Парето распределенной случайной величины может значительно отклоняться от математического ожидания, часто недооценивая его. Фактически модуль ошибки $|\overline{Z}_m - M[Z]|$ ведет себя, как $m^{(1/a)-1}$, и поэтому для значений a , близких к 1, следует заботиться о том, что выводы о сетевой производительности, связанные с ошибкой выборки, будут неверными.

Можно показать, что тяжелые хвосты распределения тесно связаны с понятием длительной памяти и МУЗ процесса. Рассмотрим предсказуемость некоторой случайной величины, обладающей РТХ.

Пусть Z – случайная величина, обладающая РТХ и интерпретируемая как время жизни (продолжительность) сетевого соединения (ТСР-сессии,

например). Предположим, что соединение было активно в течение некоторого времени $\tau > 0$. Для упрощения выкладок будем полагать время дискретным $t \in Z_+$, и введем индикаторную функцию $A : Z_+ \rightarrow \{0;1\}$ такую, что $A(t) = 1$ только в том случае, когда $Z(t) \geq t$. Тогда условная вероятность P того, что соединение, поддерживающее активность в течение времени $1 \leq t \leq \tau$, будет существовать в будущем, оценивается как

$$L(\tau) = P\{A(\tau + 1) = 1 \mid A(t) = 1\}, \quad (1.43)$$

которая также может быть представлена в виде

$$L(t) = 1 - \frac{P\{Z = \tau\}}{P\{Z \geq \tau\}}. \quad (1.44)$$

Вычислим для начала $L(t)$ для случая легких хвостов, в частности для распределения с экспоненциальными хвостами $P\{Z > x\} \sim c_1 e^{-c_2 x}$, где $c_1, c_2 > 0$ – константы. Второй член в (1.44) вычисляется как

$$\frac{P\{Z = \tau\}}{P\{Z \geq \tau\}} \sim \frac{c_1 e^{-c_2 \tau} - c_1 e^{-c_2(\tau+1)}}{c_1 e^{-c_2 \tau}} = 1 - e^{-c_2}. \quad (1.45)$$

Для больших τ получаем $L(\tau) \sim e^{-c_2}$. Поэтому в случае экспоненциально легких хвостов продолжительность активности соединения не оказывает влияния на прогноз. Для тяжелых хвостов аналогичные вычисления приводят к

$$\frac{P\{Z = \tau\}}{P\{Z \geq \tau\}} \sim \frac{c \cdot \tau^{-a} - c(\tau + 1)^{-a}}{c \cdot \tau^{-a}} = 1 - \left(\frac{\tau}{\tau + 1}\right)^a, \quad (1.46)$$

что означает

$$L(\tau) \rightarrow 1, \text{ при } \tau \rightarrow \infty. \quad (1.47)$$

Таким образом, чем больше период наблюдаемой активности соединения, тем выше вероятность, что соединение продолжит существовать в будущем, т.е. процесс обладает персистентностью. Можно обобщить данный результат на случай предсказания активности на $\delta > 1$ шагов в будущее, т.е.

$$L(\tau) = P\{A(\tau + s) = 1 \mid A(t) = 1\}, \quad (1.48)$$

где $1 \leq s \leq \delta$ и $1 \leq t \leq \tau$. Для легких хвостов получаем $L(\tau) \sim e^{-c_2 2^\delta}$, что качественно не меняет выводы для данного случая. Для тяжелых хвостов $L(\tau)$ ведет себя как $(1 + \delta/\tau)^{-a}$, и, следовательно, при достаточно больших τ ошибка предсказания может быть сколь угодно малой.

1.3.5 Аспекты теории нелинейной динамики

Я говорю вам: нужно носить в себе еще хаос, чтобы быть в состоянии родить танцующую звезду.

Ф. Ницше “Так говорил Заратустра”

Достаточно интересная в теоретическом аспекте работа была выполнена в 2000 г. (Andras Veres и Miklos Boda из Ericsson Research [16]), в которой с помощью имитационного моделирования на симуляторе ns-2 [79] показано, что трафик протокола TCP (использовалась версия TCP Tahoe) может быть как простым периодическим процессом, так и при некоторых условиях обладать сложным поведением, согласующимся с понятием ***детерминированного хаоса***.

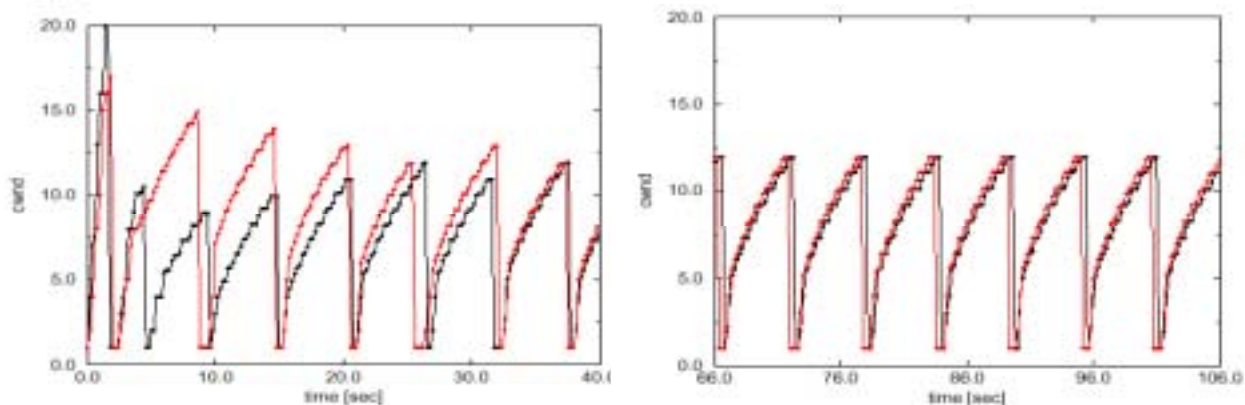
Приведем и обсудим основные результаты данной работы.

Простое периодическое поведение было исследовано для конфигурации сети, в которой две TCP сессии одновременно работают по одному и тому же соединению. Параметры соединения были выбраны следующие:

- скорость $C = 0.2$ Мбит/с;
- задержка $d = 10$ мс;
- размер буфера $B = 20$ пакетов.

Обе сессии запускаются одновременно и после короткого переходного процесса устанавливаются в периодические колебания, одно из которых всегда впереди другого, но оба следуют одним и тем же фазам: медленному старту, избежанию перегрузки и возврату. Это хорошо видно на рис.18.

Здесь представлен процесс изменения окна перегрузки (cwnd) TCP сессий во времени.



a) фаза переходного процесса;

b) после того, как закончился переходной процесс

Рис.1.7. Окна перегрузки двух TCP сессий, одновременно работающих по одному соединению

Этот график также обнаруживает, что обе сессии синхронизированы. Они двигаются вдоль “лестницы”: увеличивают свои cwnd один за другим, на вершине лестницы обнаруживаются потери и обе сессии уменьшают свои окна до одного пакета, далее этот процесс повторяется.

При изменении параметров соединения была снова получена периодичность, но более сложная (см. рис. 1.8).

В основе все еще лежат регулярные удары, но на больших масштабах времени существуют перемежающиеся части (сначала одна сессия выигрывает в скорости, а затем другая берет верх). Изменяя таким образом параметры, можно получить еще более сложные реализации.

Можно наблюдать эволюцию системы не только как функцию времени, но и как траекторию системы в фазовом пространстве, сделав соответствующие построения. Фазовое пространство – многомерное пространство, где каждое измерение соответствует определенной системной переменной, таким образом, каждая точка в фазовом пространстве соответствует уникальному состоянию системы. Если система периодична, то отвечающая ей траектория в фазовом пространстве будет петлей, и

наоборот – если эволюция системы может быть представлена петлей в фазовом пространстве, то система – периодична.

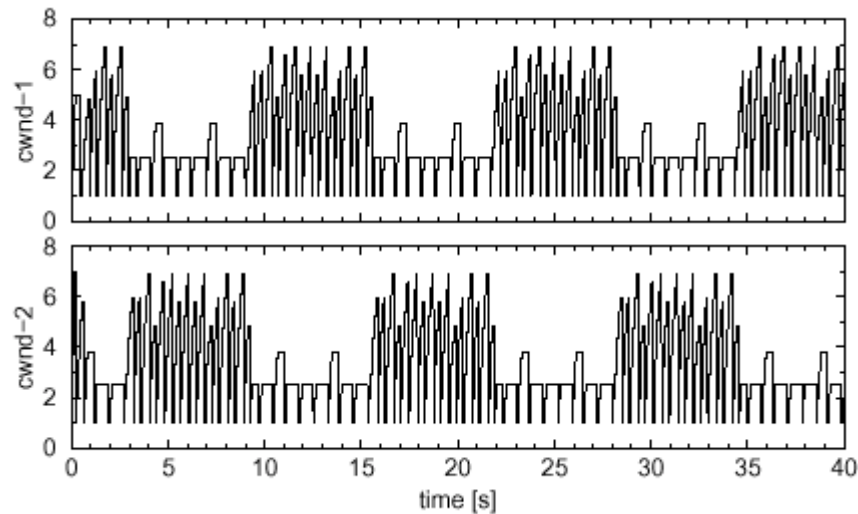


Рис.1.8. Окна перегрузки двух TCP сессий. $C=0.5$ Мбит/с, $d=10$ мс, $V=4$ пакета

На рис.1.9 представлены траектории на фазовой плоскости для простой и более сложной (перемежающейся) периодических систем, обсуждаемых выше. Обе системы представляются замкнутыми петлями, что есть признак периодичности, но уровень сложности этих петель весьма разный: простая система характеризуется простой петлей, в то время как “перемежающаяся” система имеет более сложную, но более симметричную траекторию.

Несмотря на различную сложность, обе траектории доказывают свою высокую устойчивость. Это означает, что не имеет значения, каким образом мы возбуждаем систему (например, случайными потерями пакетов или, изменяя начальные условия, например, запускаем вторую сессию TCP на несколько секунд позже) в конце концов, система вновь вернется к тем же регулярным траекториям. В теории нелинейных динамических систем эти траектории называются *аттракторами*. Аттрактор – предельное множество траекторий в фазовом пространстве системы, к которому стремятся все траектории из некоторой окрестности этого множества [32].

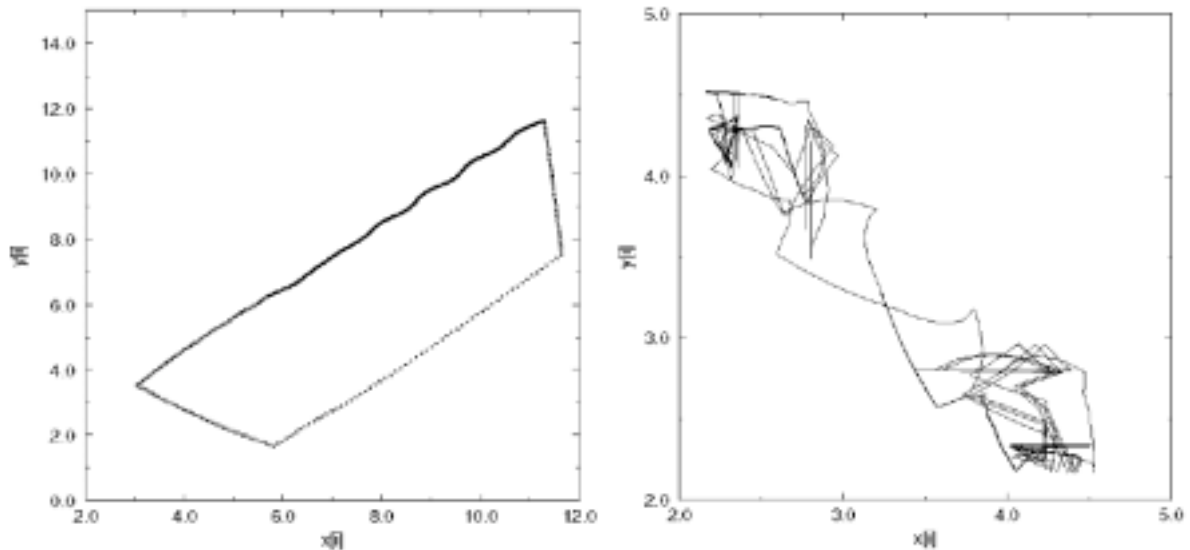


Рис.1.9. Периодические аттракторы двух TCP сессий

- а) $C=0.2$ Мбит/с, $d=10$ мс, $B=20$ пакетов, $n=100$ б) $C=0.5$ Мбит/с, $d=10$ мс, $B=4$ пакетов, $n=90$

Если это предельное множество есть устойчивое состояние равновесия, то аттрактор системы будет просто неподвижной особой точкой; если это устойчивое периодическое движение – аттрактором будет замкнутая кривая, называемая также предельным циклом. Режим детерминированного хаоса также характеризуется аттрактором, но траектория такого аттрактора непериодическая (не замыкается) и режим функционирования неустойчив (малые отклонения от начальных условий нарастают экспоненциально). С легкой руки математика Ф.Тakensа такие аттракторы стали называть **“странными”**.

Итак, для одного набора значений параметров (например, количество конкурирующих TCP сессий, скорость, размеры буферов, задержки передачи) поведение системы выглядит просто, для другого – очень сложно. Нетрудно подыскать параметры, для которых кажется, что система не повторит себя никогда. Конечно, так как система имеет конечную размерность, траектории всегда будут периодическими, но размер этого периода – очень большой. Аттрактор данной системы показан на рис. 1.10. В [16] показано, что размерность Хаусдорфа такого аттрактора больше 1, но меньше 2, а именно равна 1,61. Кроме того, для случая одновременной

работы 30 ТСП сессий была оценена экспонента Ляпунова $\lambda \approx 1.11$, означающая, что после внесения возмущения различие между системами нарастает со средней скоростью $e^{\lambda} \approx 3.03$ в секунду. Сочетание этих двух фактов (дробность размерности и положительность экспоненты Ляпунова) дает основание авторам говорить о “странности” аттрактора и, как следствие, о наличии в обсуждаемой системе признаков режима детерминированного хаоса.

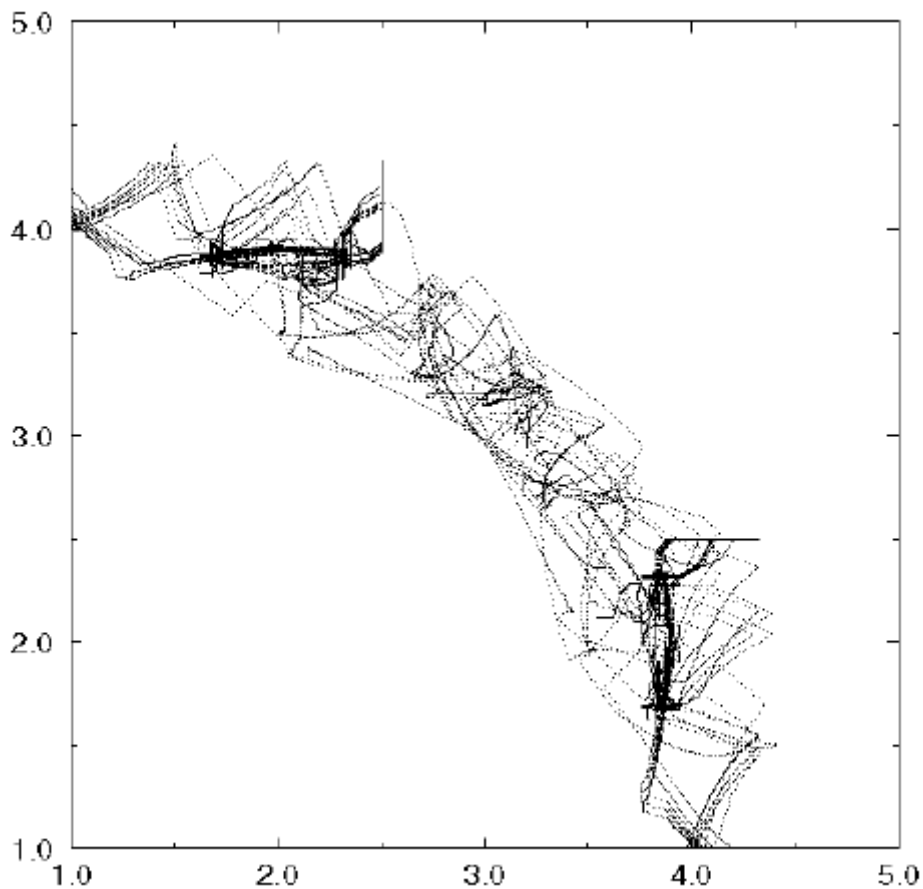


Рис.1.10. “Странный” аттрактор. $C=0.1$ Мбит/с, $d=10$ мс, $V=4$ пакета $n=300$

Заметим однако, что с точки зрения теории хаоса траектория странного аттрактора должна быть непериодической [32]. Поэтому, так как аттрактор, представленный на рис.1.10, является периодическим (несмотря на то, что период его достаточно большой – около 4 часов), будем называть его почти странным (т.е. “странным”). Так или иначе, пусть с некоторыми допущениями, но в данной работе был сделан очень важный шаг, а именно, показано, что система одновременно работающих по одному соединению

ТСР сессий при некоторых параметрах может войти в режим детерминированного хаоса и производить трафик, обладающий скрытым порядком, но который выглядит как абсолютно случайный процесс и моделировался ранее именно с привлечением теории случайных процессов. Здесь важно, что это сложный, похожий на случайный, но, тем не менее, детерминированный процесс. Задавая абсолютно точно начальные условия, мы можем повторять данный процесс сколь угодно раз, при этом траектории будут абсолютно одинаковыми. Однако сколь угодно малое отклонение от первоначальных условий – и траектории разбегаются, причем расстояние между ними во времени увеличивается по экспоненте. Но опять-таки, последующее состояние системы всегда полностью определяется ее прошлым.

С точки зрения механизмов управления трафиком присутствие режимов детерминированности (хаоса) в трафике означает теоретическую возможность его предсказания, в случае определения точной зависимости, конечно. Однако следует заметить, что данный подход к прогнозированию имеет следующий недостаток: поскольку физически невозможно установить абсолютно точные начальные условия для предсказательной функции, соответствующие данному трафику, то с увеличением интервала предсказания (времени, на которое предсказывается процесс) различие между реальным трафиком и предсказанным будет стремительно увеличиваться. Причем, как следует из теории нелинейных динамических систем, в том числе хаотических [32],[38], это локальное увеличение будет происходить в среднем по экспоненциальному закону.

1.4 Постановка задачи обеспечения качества обслуживания (QoS) в условиях влияния эффекта самоподобия

Для того чтобы представить себе особенности, возникающие в реальной сети вследствие эффекта самоподобия, рассмотрим механизм статистического мультиплексирования информационных потоков.

Алгоритм статистического мультиплексирования потоков широко используется в телекоммуникациях, поскольку позволяет экономно использовать пропускную способность магистральных каналов. Рассмотрим простейший пример передачи информации многих источников по одному магистральному каналу. В принципе, можно закрепить за каждым из источников определенную часть ресурсов магистрального канала (скажем, разделить источники по частоте). В этом случае каждый источник может использовать только ту часть ресурсов, которая ему отведена (см. рис. 1.11 слева). Другой способ передачи, называемый статистическим мультиплексированием, состоит в том, что потоки отдельных источников складываются (агрегируются) в магистральном канале с экономией пропускной способности dC (см. рис. 1.11 справа).

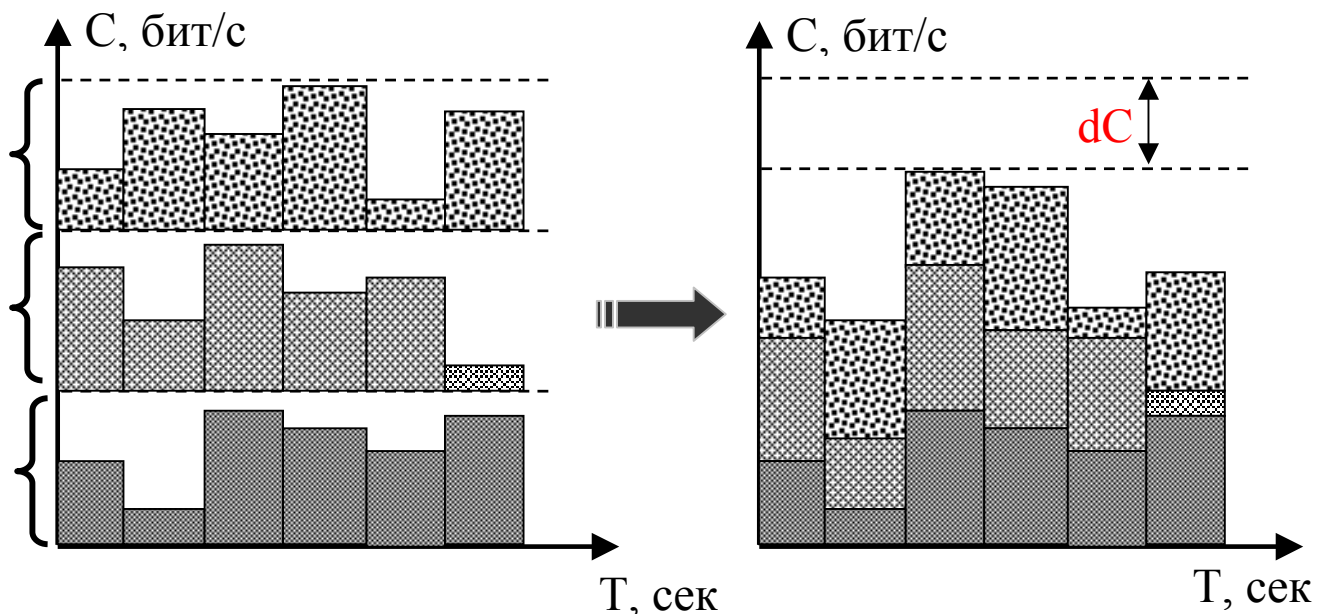


Рис. 1.11. К пояснению работы алгоритма статистического мультиплексирования. Слева показаны трафики трех отдельных источников при жестком разделении пропускной способности общего канала между ними. Справа – трафики тех же источников в магистральном канале при работе алгоритма статистического мультиплексирования. При этом достигается выигрыш в пропускной способности dC

Рассмотрим второй вариант более подробно. Допустим, имеются n отдельных (парциальных) источников. Пусть процессы (см. рис.1.12) $\xi_1(M[\xi_1], D[\xi_1]), \dots, \xi_n(M[\xi_n], D[\xi_n])$ имеют одинаковые средние $M[\xi_i] = \mu$ и

дисперсии $D[\xi_i] = \sigma^2$, тогда при условии независимости и одинаковом распределении $\xi_1, \xi_2, \dots, \xi_n$ коэффициент вариации результирующего процесса ξ_Σ в магистральном канале:

$$\text{cov}(\xi_\Sigma) = \frac{\sqrt{D[\xi_\Sigma]}}{M[\xi_\Sigma]} = \frac{\sqrt{D[\xi_1 + \xi_2 + \dots + \xi_n]}}{M[\xi_1 + \xi_2 + \dots + \xi_n]} = \frac{\sqrt{n \cdot D[\xi_1]}}{n \cdot M[\xi_1]} = \frac{\sigma}{\mu\sqrt{n}}. \quad (1.49)$$

Как видно из (1.49), коэффициент вариации представляет собой отношение среднеквадратического отклонения процесса к его среднему.

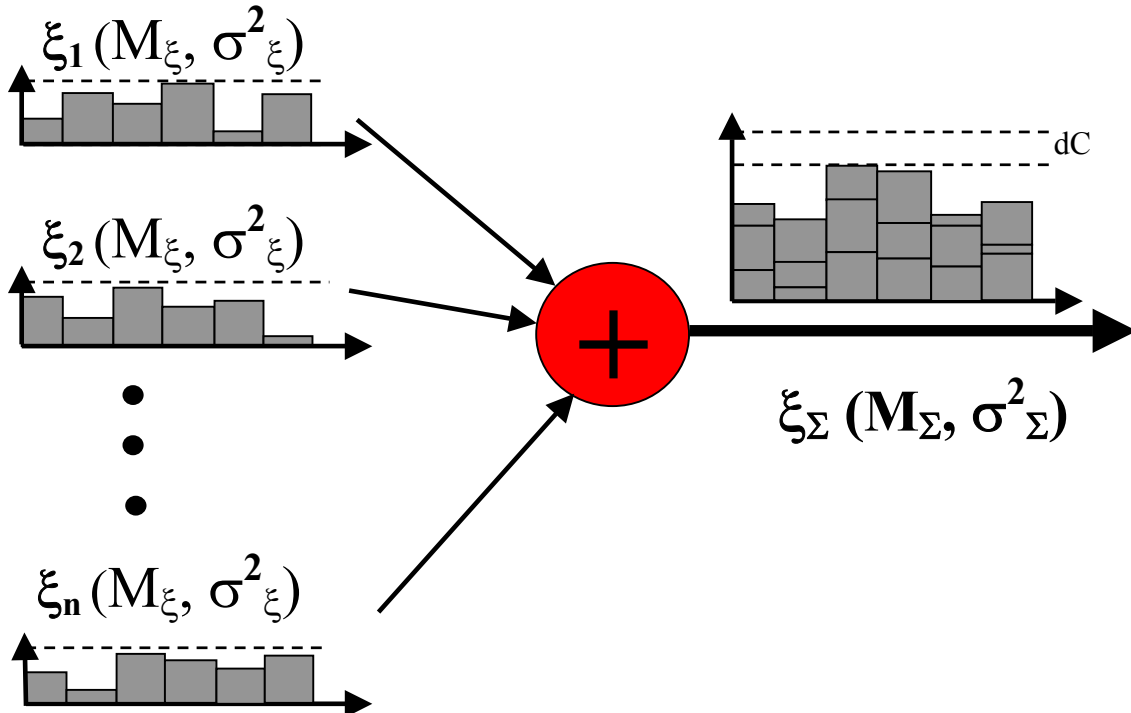


Рис.1.12. Схема статистического мультиплексирования $\xi_1(M[\xi_1], D[\xi_1]), \dots, \xi_n(M[\xi_n], D[\xi_n])$ с получением в магистральном канале процесса ξ_Σ

В данном случае коэффициент вариации отражает степень сглаживания результирующего процесса ξ_Σ при увеличении количества мультиплексируемых парциальных процессов. Эффект зрительного сглаживания процесса ξ_Σ при росте n достигается благодаря более быстрому росту среднего процесса ξ_Σ по сравнению с его среднеквадратическим отклонением (СКО). Принципиально, что среднее сравнивается с СКО, а не с дисперсией. Это делается для получения безразмерной оценки степени сглаженности (коэффициента вариации в данном случае). Более того, в

соответствии с центральной предельной теоремой (ЦПТ) при увеличении n должна также происходить нормализация процесса ξ_{Σ} . Однако на практике чаще всего ресурсы магистрального канала (пропускная способность, буферы) много меньше, чем суммарная потенциальная возможность мультиплексируемых процессов, что определяет эффективность системы [34].

Как результат, парциальные потоки при сложении в ограниченном буфере напимер теряют независимость.

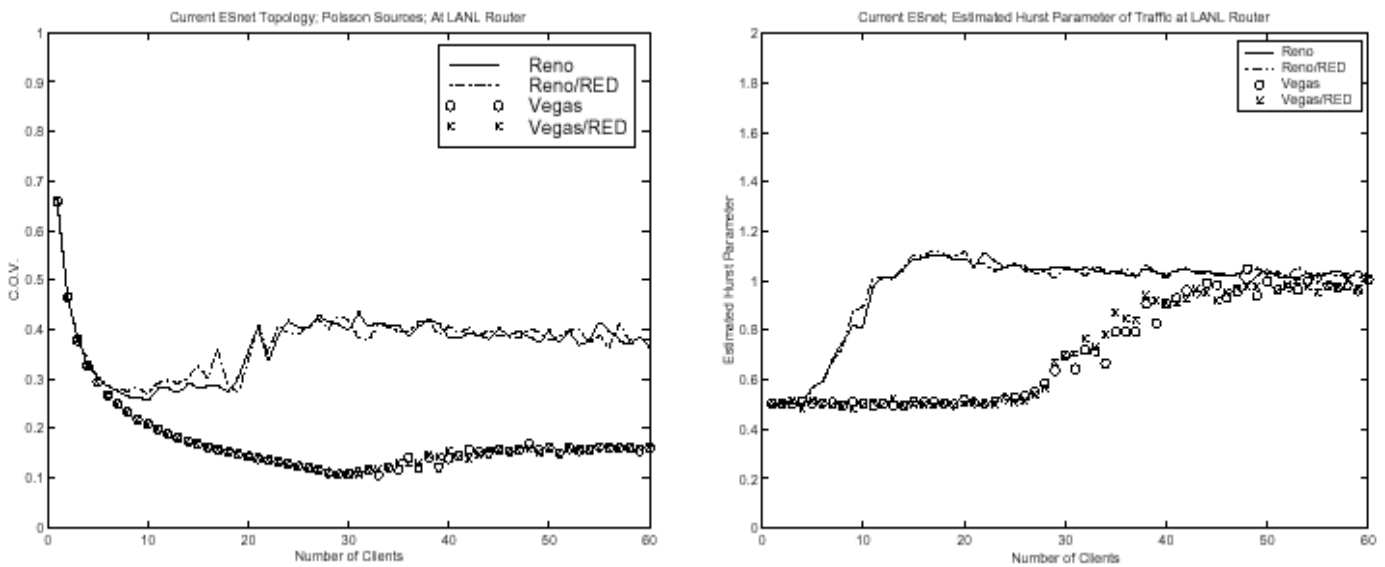


Рис. 1.13. Слева – зависимость коэффициента вариации магистрального потока от числа клиентов (мультиплексируемых потоков). Справа – зависимость величины показателя Хэрста от числа клиентов. В обоих случаях приведены графики для версий протокола TCP (Reno, Vegas), функционирующих в исследуемой сети

Эффекты, которые могут возникать при мультиплексировании зависимых потоков подробно рассмотрены в [8]. В частности, эксперимент показал, что уже при $n=10...15$ (и при работе широко распространенной реализации Reno протокола TCP) формула (1.49) дает погрешности, а при $n=60$ реально измеренный $cov(\xi_{\Sigma})$ больше рассчитанного по данной формуле на 300% (см. рис. 1.13)!

Примечательно, что степень зависимости парциальных потоков зависит от версии протокола сетевого уровня TCP, функционирующей в сети. Действительно (см. рис. 14), при работе версии Vegas коэффициент вариации

ξ_{Σ} уменьшается при увеличении n до 30 (в отличие от более распространенной версии Reno). Далее, видимо, мультиплексируемые потоки утрачивают независимость. Характерно, что при этом показатель Хэрста процесса ξ_{Σ} , увеличиваясь от 0.5 до 1 в точке $n=15$ для Reno и в точке $n=30$ для Vegas. Можно сделать вывод, что, по мере того как входные процессы теряют независимость, процесс на выходе становится все более персистентным. В результате агрегированный процесс не достигает расчетной (1.49) степени сглаживания, и алгоритм статистического мультиплексирования оказывается малоэффективным.

Типичный вид агрегированного сетевого трафика показан на рис. 1.14.

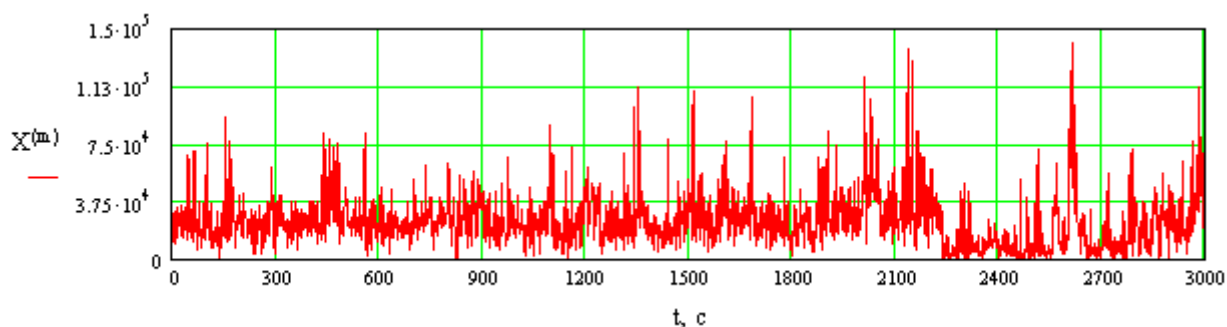
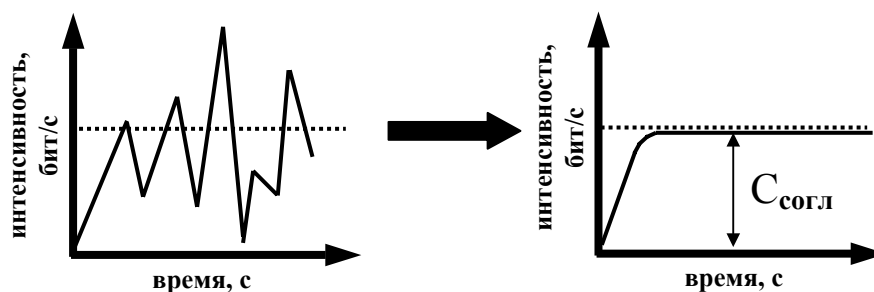


Рис. 1.14. Усредненная по блокам длины $m = 1$ сек. реализация lbl-pkt-5 [80]

Каждая точка на данном графике представляет собой количество байт информации, переданной в магистральном канале за интервал времени в 1 секунду. Длительность реализации составляет 3000 точек или 50 минут. Коэффициент Хэрста соответствует примерно 0.8. Как видно из рисунка, процесс имеет высокую изменчивость (поскольку подчиняется распределению с тяжелым хвостом) и его вряд ли можно назвать сглаженным. Как результат, для того чтобы передать такой трафик без потерь, пропускная способность канала должна соответствовать уровню пиковых выбросов, т.е. в данном случае быть не менее $1.4 \cdot 10^5$ Бит/с. Можно заметить, поскольку средний уровень трафика все-таки достаточно низкий, пропускная способность будет расходоваться неэффективно. Другими словами, коэффициент использования такого канала будет низким.

Для управления интенсивностью и сглаживания профиля передаваемого по сети трафика существуют два основных способа: шейпинг (Traffic Shaping) и полисинг (Traffic Policing) [24], [25]. Суть данных алгоритмов, как следует из рис. 1.15, заключается в следующем: шейпинг сглаживает трафик и пересылает его с постоянной интенсивностью (согласованной скоростью передачи $C_{\text{согл}}$) путем постановки в очередь (буферизации) пакетов, интенсивность передачи которых превысила заданное значение; механизм полисинга в свою очередь просто отбрасывает пакеты, интенсивность которых выше согласованной скорости передачи $C_{\text{согл}}$.

• **шейпинг**



• **полисинг**

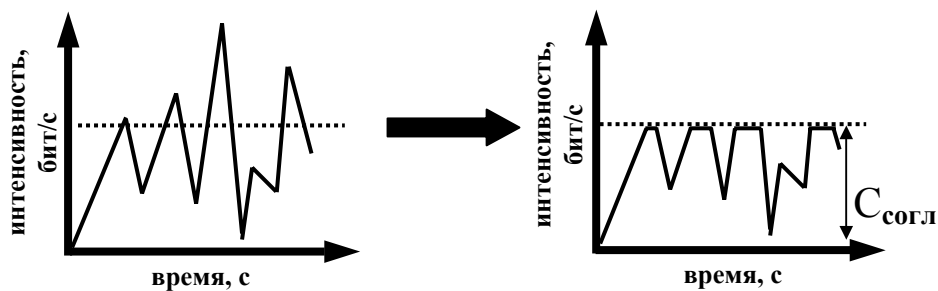


Рис. 1.15. Принципы функционирования механизмов управления интенсивностью трафика

С одной стороны, так как шейпинг не допускает отбрасывания пакетов, это делает его привлекательным для управления передачей информации реального времени (голос, реальное видео). С другой стороны он вносит задержки, связанные с буферизацией, что отрицательно сказывается на характеристиках передаваемого трафика. Алгоритм полисинга в отношении burstного трафика (т.е. трафика, обладающего высокой пачечностью) также

проявляет себя далеко не с лучшей стороны: чтобы достичь приемлемых показателей потерь, необходимо значительно увеличить пропускную способность канала, снизив при этом утилизацию (коэффициент использования) в канале.

С развитием теории самоподобного телетрафика все больше появляется работ по предсказанию интенсивности трафика [27], [29], [30], [31], [51], [52]. Возможность осуществлять прогнозы, как уже упоминалось выше, обязана свойству длительной памяти процессов и теоретически должна обеспечить повышение коэффициента использования канала и увеличение общей эффективности системы.

На рис. 1.16 иллюстрируется идея функционирования данного алгоритма. Очевидно, чем ближе прогнозируемая пропускная способность \hat{C} к профилю трафика, тем меньше потерь при пропуске такого трафика через канал с прогнозируемой пропускной способностью, тем меньше вносимые задержки пакетов, возникающие из-за буферизации и тем выше утилизация (использование) канала.

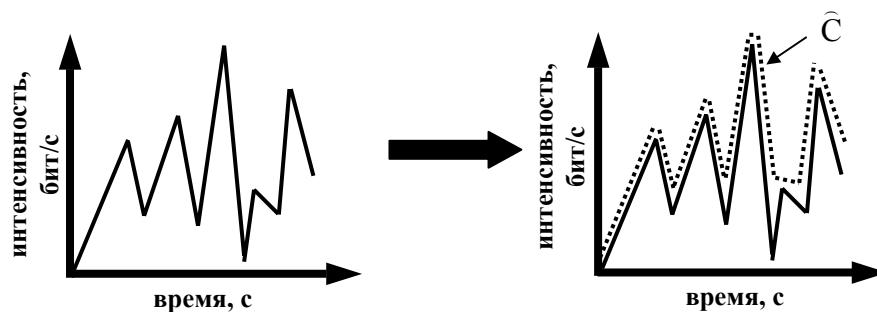


Рис.1.16. Вариант функционирования алгоритма динамического распределения пропускной способности с прогнозированием

Иначе говоря, пропускная способность канала меняется динамически, отслеживая профиль трафика. Это основное отличие предложенной нами схемы от статического способа задания пропускной способности с помощью полисинга и шейпинга.

В разделе 1.3.4 была показана принципиальная возможность предсказания будущего процесса, удовлетворяющего распределению с

тяжелыми хвостами. Более детальная информация о прогнозируемости трафика содержится в работе [23]. Там же представлена зависимость средней энтропии S трафика от длины интервала времени, на котором он исследуется. По данной зависимости теоретически можно оценить время, требующееся на наблюдение трафика, прежде чем сделать выводы о его будущем с приемлемой степенью достоверности.

Таким образом, основные задачи, которые необходимо решить в данной диссертации с целью увеличения эффективности алгоритмов обеспечения качества обслуживания при обработке сетевого трафика, обладающего самоподобными свойствами, сформулируем следующим образом:

- формирование основных идей и принципа функционирования алгоритма обеспечения качества обслуживания в условиях самоподобного трафика;
- подготовка и проведение эксперимента по измерению (снятию) трафика, а также выполнение статистического анализа реализаций трафика на предмет выявления его характерных особенностей (самоподобия, скрытых детерминированных компонент, периодичностей, тренда, режимов детерминированного хаоса и пр), которые необходимо учитывать при разработке алгоритма обеспечения качества обслуживания в условиях самоподобного телетрафика;
- анализ прогнозируемости сетевого трафика с помощью различных методов предсказания временных рядов. Оценка качества прогноза и выбор (либо формирование) наиболее подходящего предсказателя для алгоритма динамического распределения пропускной способности с использованием прогнозирования;
- разработка блок-схемы алгоритма функционирования и принципов реализации в существующих системах распределения информации механизма обеспечения качества обслуживания при влиянии эффекта самоподобия;

- проведение испытаний (имитационное моделирование на ПЭВМ) и оценка эффективности разработанного алгоритма при обработке самоподобного телетрафика.

1.5 Выводы по главе 1

1. В настоящей главе на основе известных автору источников изложено современное состояние и основные понятия теории самоподобного телетрафика.
2. Подробно рассмотрены и, по возможности, взаимосвязаны в отношении прогнозирования телетрафика такие понятия как фрактальность, коэффициент Хэрста, самоподобие, хаос, персистентность/антиперсистентность, продолжительная память, до сих пор во многих работах изучаемые отдельно друг от друга. Данный шаг позволяет с более широких позиций подойти к проблеме обеспечения качества обслуживания в системах распределения информации с присутствием эффекта самоподобия.
3. Показано, что предсказание будущих процессов с продолжительной памятью, к которым относятся самоподобные процессы, теоретически можно осуществить, зная их относительно недавнее прошлое. Применение концепций теории нелинейных динамических систем (в том числе теории хаотических систем) видится полезным только при построении краткосрочных прогнозов, поскольку невозможность задания начальных условий с бесконечной точностью в сочетании с экспоненциальной расходимостью соседних траекторий хаотической системы делает предсказание на длительное время вперед практически невозможным.
4. Сформулирована задача увеличения качества обслуживания в условиях самоподобного телетрафика при помощи прогнозирования. Показаны теоретически возможные превосходства алгоритма динамического распределения пропускной способности канала для самоподобного

телетрафика по сравнению с классической схемой статического задания пропускной способности: уменьшение потерь и увеличение утилизации (коэффициента использования) в канале.

5. Сформулированы основные задачи, которые необходимо решить для разработки окончательного и практически реализуемого в условиях самоподобного телетрафика алгоритма динамического распределения пропускной способности канала:

- подготовка и проведение эксперимента по измерению (снятию) трафика, а также выполнение статистического анализа реализаций трафика;
- анализ прогнозируемости сетевого трафика с помощью различных методов предсказания временных рядов;
- разработка блок-схемы алгоритма функционирования механизма обеспечения качества обслуживания при влиянии эффекта самоподобия;
- проведение испытаний (имитационное моделирование на ПЭВМ) и оценка эффективности разработанного алгоритма при обработке самоподобного телетрафика.

Глава 2. Статистический анализ реализаций сетевого трафика

Целью настоящей главы является статистический анализ и исследование структуры трафика в различных системах распределения информации на предмет выявления его характерных особенностей, которые необходимо учитывать при разработке алгоритма обеспечения качества обслуживания в условиях влияния эффекта самоподобия.

2.1 Описание реализаций сетевого трафика

Если достаточно долго пытаться данные, то,
в конце концов, Природа признается.

Рональд Коуз

Для дальнейшего анализа выберем известные реализации сетевого трафика, которые представлены в Internet traffic archive [80]. Ниже, в п. 2.6, приведены результаты оригинального эксперимента по сбору и исследованию трафика в беспроводной сети передачи данных.

2.1.1 Реализация сетевого трафика BC-Oct89Ext.TL

Первая из изучаемых нами реализаций была получена Will Leland и Dan Wilson. Она названа авторами BC-Oct89Ext.TL и представляет собой зависимость от времени размеров сетевых кадров протокола второго уровня модели OSI (Ethernet-кадров в данном случае), проходящих через вычислительную сеть Bellcore Morristown Research and Engineering facility в глобальную сеть Internet. Реализация исследуется в работах [1] и [36].

Данные представлены двумя колонками чисел в ASCII формате. Первый столбец представляет собой время в секундах. Второй – длину поля данных Ethernet-кадра в байтах (не включая преамбулу, заголовка и контрольной суммы CRC). Заметим, что существует как минимум четыре основных разновидности кадров Ethernet:

- Ethernet Type II;

- Ethernet 802.3;
- Ethernet 802.2;
- Ethernet SNA.

Поскольку авторы не уточняют спецификацию кадра, который использовался при записи трафика, напомним общий формат кадров Ethernet (см. рис. 2.1.).

Реализация VC-Oct89Ext.TL начинается 3 октября 1989 года (день недели – вторник) в 23 час. 44 мин. по восточноевропейскому времени, заканчивается спустя 122797.83 секунд (34,11 часов) и содержит 1 миллион записей об Ethernet-кадрах, проходящих в это время между сетью корпорации Bellcore и сетью Internet.

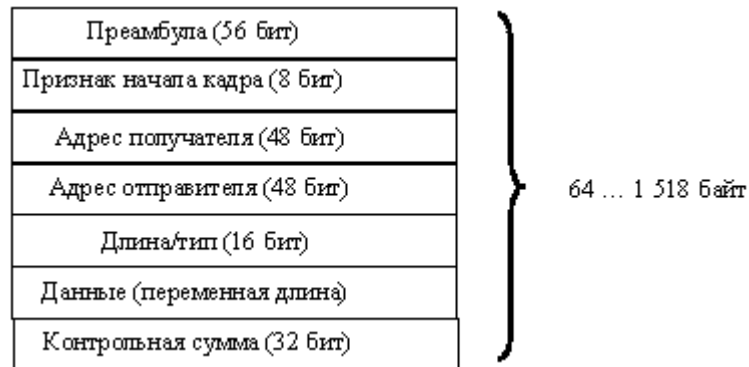
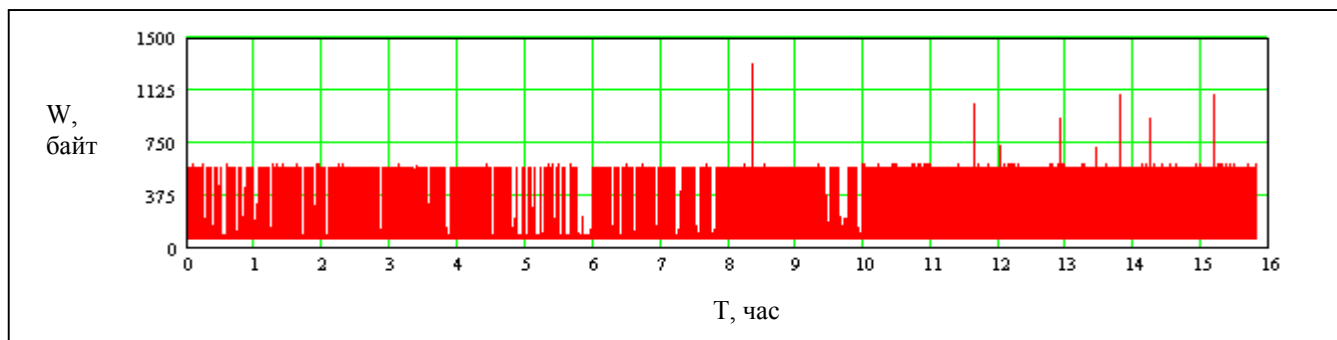
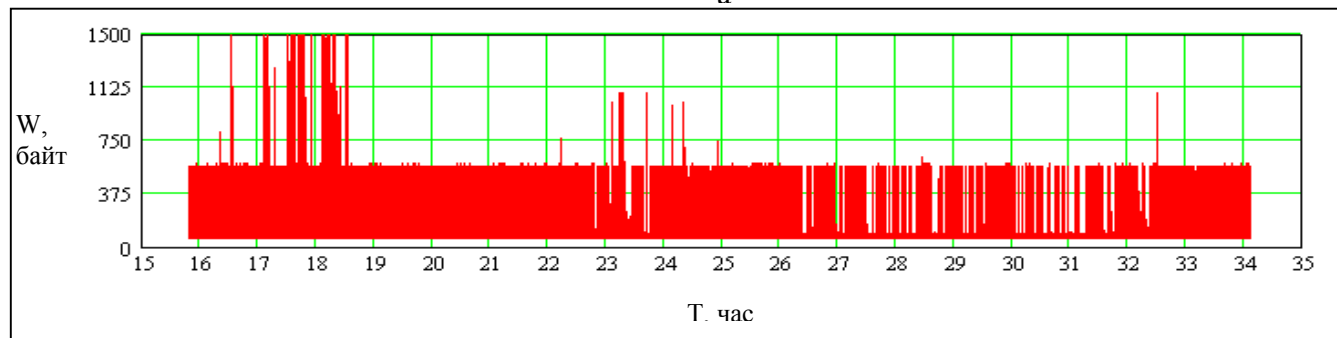


Рис. 2.1. Общий формат кадра Ethernet

Исследователи утверждают, что аппаратура измерения не отбрасывала каких-либо Ethernet-кадров, проходящих через нее. Тем не менее, испорченные пакеты (возникающие, например, при коллизиях) в реализацию не включались. Кроме того, 99.5% инкапсулированных в Ethernet-кадры пакетов относятся к протоколу IP. По данным первоисточника точность временных отсчетов составляет порядка 10 микросекунд. Заметим, что отсчеты не эквидистантны, а соответствуют моментам прихода кадров. Иначе говоря, для дальнейшего анализа реализацию необходимо будет привести к эквидистантной шкале по оси времени (это делается в разделе 2.2). График реализации VC-Oct89Ext.TL представлен на рис 2.2: здесь по оси абсцисс отложены временные отсчеты “Т” (в часах), а по оси ординат – длительности “W” (в байтах) Ethernet-кадров.



а



б

Рис. 2.2. Зависимость длин кадров реализации BC-Oct89Ext.TL от времени: а) первые 16 часов, б) оставшиеся 18 часов

Минимальная и максимальная длительность кадра в данной реализации составляет 64 байта и 1 515 байт соответственно. Для наглядности построим распределение длительностей кадров – гистограмму (см. рис. 2.3).

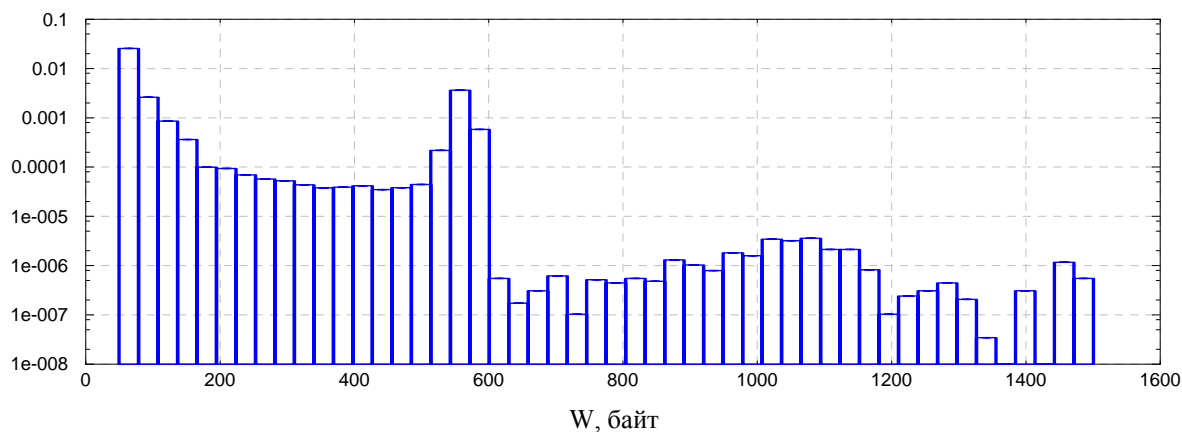


Рис. 2.3. Распределение длительностей Ethernet-кадров реализации BC-Oct89Ext.TL

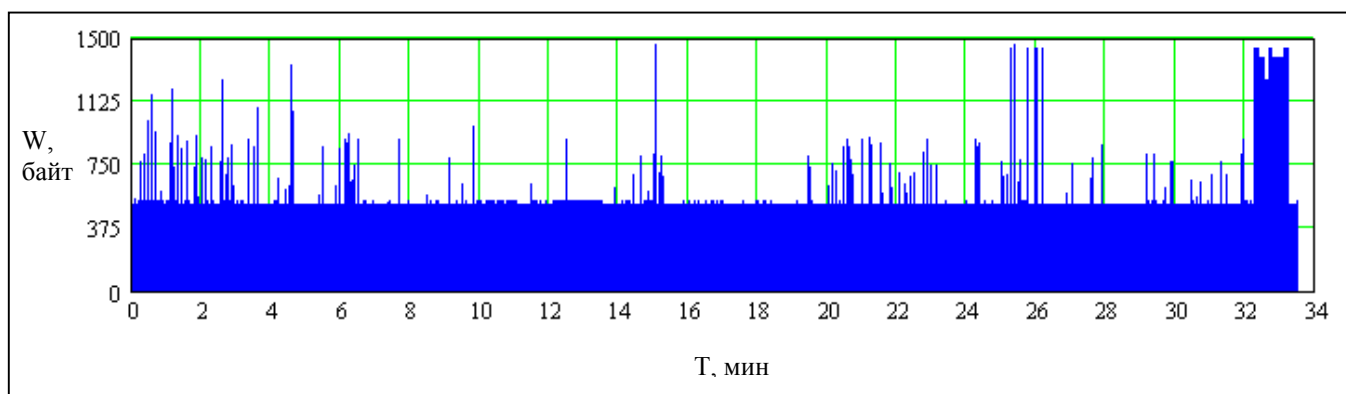
Визуальный анализ гистограммы показывает, что большинство кадров имеют длительности порядка 64 и 570 байт. Может оказаться интересным,

что частота появления кадров, длительность которых $W > 600$ байт, существенно ниже (на два порядка), чем частота появления кадров, обладающих $W < 600$ байт.

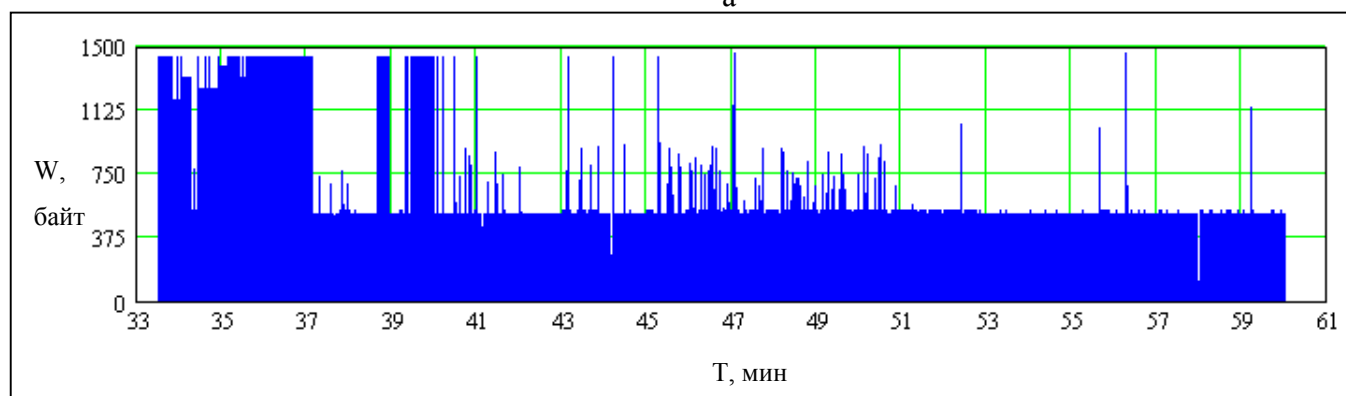
2.1.2 Реализация сетевого трафика LBL-PKT-5.TCP

Следующей реализацией, которую мы будем исследовать, будет реализация LBL-PKT-5.TCP, полученную Vern Paxson и изучаемую, например в [37]. Она представляет собой зависимость от времени размеров дэйтаграмм протокола четвертого уровня модели OSI (TCP в данном случае), проходящих между сетью Lawrence Berkeley Laboratory, расположенной в Berkeley, California, и глобальной сетью Internet.

Данные представлены шестью колонками чисел в ASCII формате, из которых нас будут интересовать только первый (время в секундах) и шестой столбцы (длительность дэйтаграмм в байтах).



а



б

Рис. 2.4. Зависимость длин дэйтаграмм реализации LBL-PKT-5.TCP от времени: а) первые 33 минуты, б) оставшиеся 27 минут

Реализация записывалась 28 января 1994 года (пятница) с 14 час. 00 мин. до 15. час. 00 мин (по Тихоокеанскому поясному времени) и содержит 677 846 отсчетов, сделанных через неодинаковые интервалы времени, точность которых – 1 микросекунда. График реализации LBL-РКТ-5.ТСР представлен на рис 2.4.

Минимальная и максимальная длительность дэйтаграмм в данной реализации составляет 0 байт (пакеты "ping-ack") и 1 460 байт соответственно. Распределение длительностей дэйтаграмм приводится на рис. 2.5.

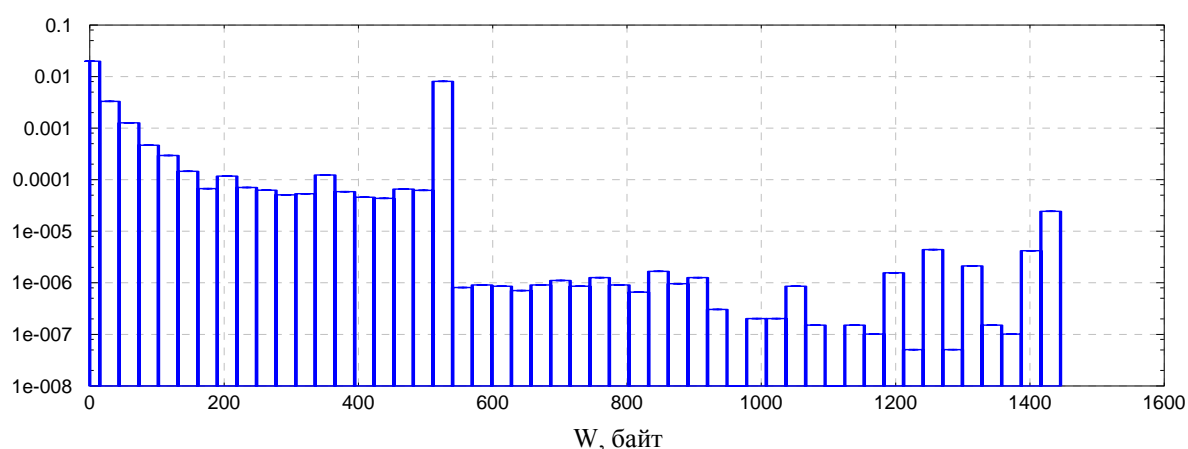


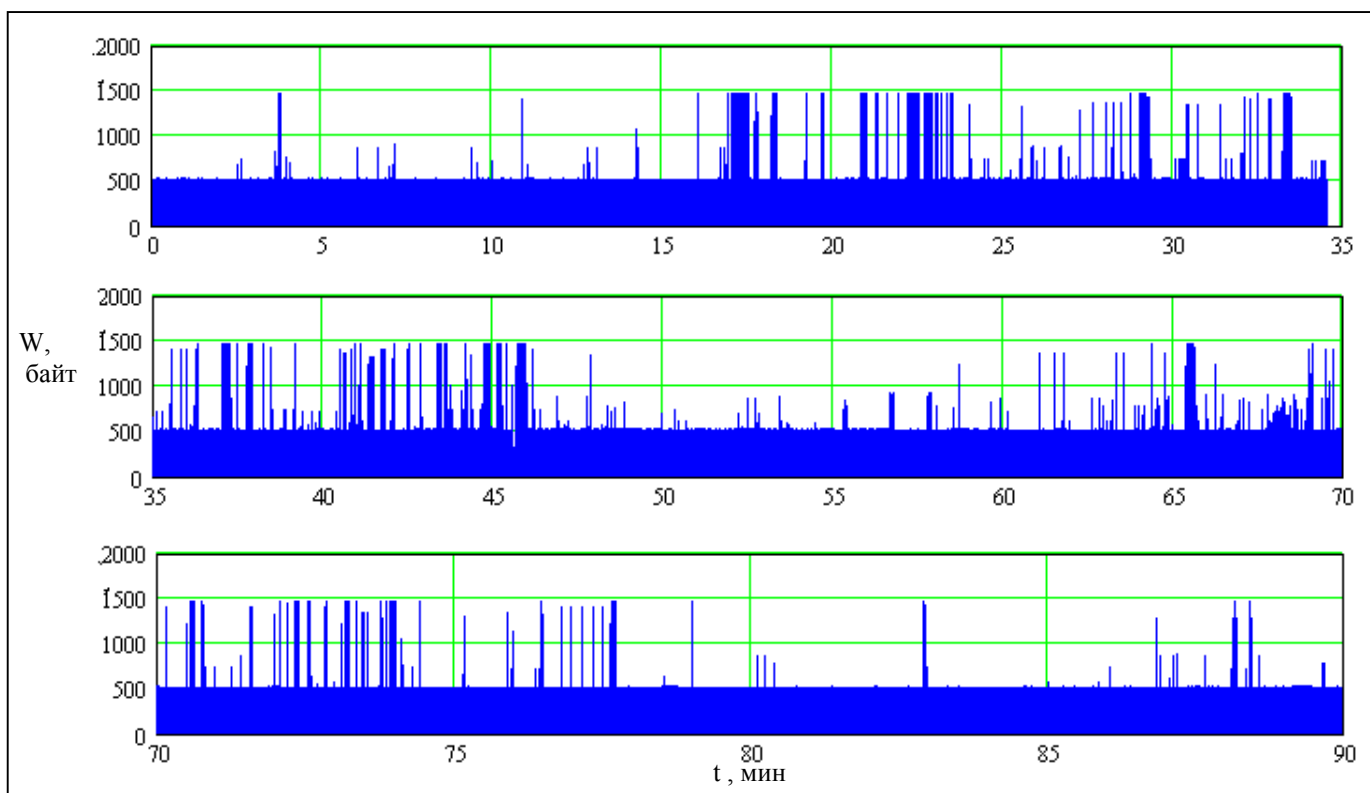
Рис. 2.5. Распределение длительностей дэйтаграмм реализации LBL-РКТ-5.ТСР

Изучая гистограмму реализации LBL-РКТ-5.ТСР, можно отметить, что большинство дэйтаграмм имеют длительности 0 байт и 512 байт. Кроме того, наблюдается характерный спад частот появления дэйтаграмм, длительность которых превышает значение 512 байт.

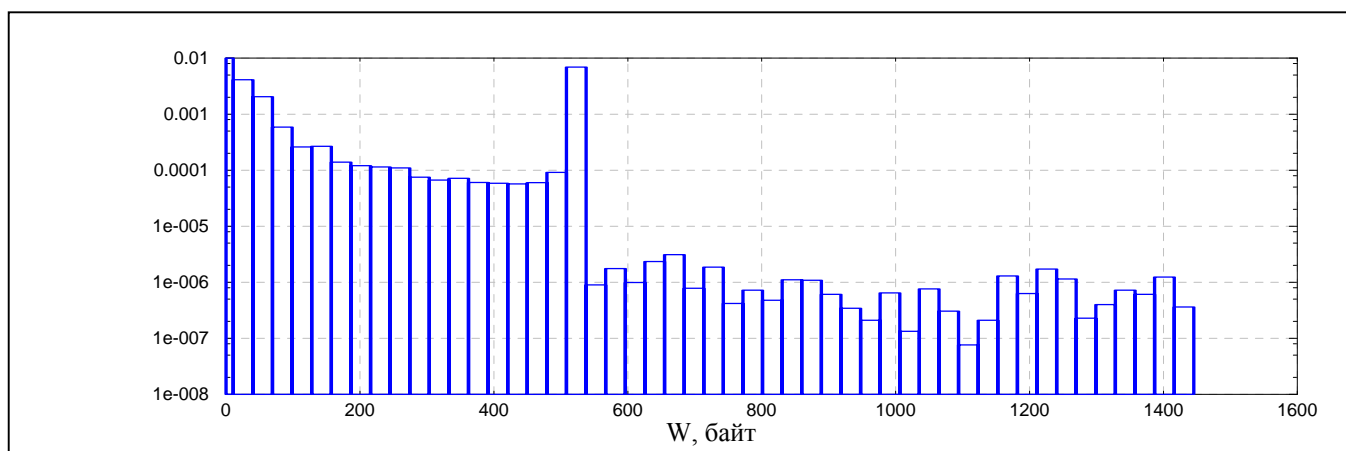
2.1.3 Реализация сетевого трафика LBL-ТСР-3

Возьмем еще одну реализацию LBL-ТСР-3. Эта реализация аналогична LBL-РКТ-5.ТСР и представляет собой запись ТСР-трафика между сетью Internet и локальной вычислительной сетью Lawrence Berkeley Laboratory. Трафик наблюдался Vern Paxson 20 января 1994 (четверг) с 14 час. 10 мин. до 16 час. 10 мин. Соответствующий временной ряд содержит информацию об 1.3 миллионе ТСР-дэйтаграмм и представлен, также как и предыдущий,

шестью столбцами чисел: временные отсчеты, хост-источник, хост-получатель, TCP-порт источника, TCP-порт получателя и количество байт в дэйтаграмме (ноль для "pure-ack" – пакетов). По информации первоисточника [1] временные отсчеты имеют точность – 1 микросекунда. Данная реализация изучается в [37]. График реализации и соответствующее распределение длительностей TCP-дэйтаграмм представлены на рис. 1.6.



а



а

Рис. 2.6. а) Зависимость количества байт от времени в реализации LBL-TCP-3
 б) Распределение длительностей дэйтаграмм реализации LBL-TCP-3

Сравнивая гистограмму реализации LBL-TCP-3 (рис. 2.6) с гистограммами рассмотренных выше реализаций (рис. 2.5 и рис. 2.3), можно заметить, что все они обладают схожими структурными особенностями. А именно, наблюдается плавное уменьшение частот появления дэйтаграмм, длительности которых находится в интервале 0 ...512 байт, далее следует характерный всплеск в районе 512 байт практически на 2 порядка, после чего - резкое уменьшение (на 4 порядка) частот появления дэйтаграмм, длительность которых >512 байт (> 600 для Ethernet-трафика).

Поскольку дэйтаграммы протокола TCP инкапсулируются в Ethernet-кадры, можно сделать вывод, что распределение длин Ethernet-кадров определяются распределением TCP-дэйтаграмм, а не наоборот. Интересно заметить, что распределение Ethernet-кадров наследует особенности распределения TCP-дэйтаграмм, несмотря на то, что они относятся к разным реализациям.

2.2 Формирование временных рядов

2.2.1 Процедура агрегирования

Как уже упоминалось ранее, процедура агрегирования необходима для приведения исходных реализации к эквидистантной (с постоянным шагом ΔT) шкале по оси времени. При этом величину ΔT будем называть уровнем агрегирования. Для этого ось времени разбивается на интервалы ΔT , каждому из которых ставится в соответствие отношение количества прошедшей за данный интервал времени информации (в байтах) к длительности интервала (в секундах).

Допустим $\xi = \{\xi(t_1), \xi(t_2), \dots, \xi(t_n)\}$ - исходная (не эквидистантная) реализация трафика, длительностью n отсчетов. При этом $|t_{i+1} - t_i| \neq \text{const}$, где $i \in 1, n$. На рис. 2.7 схематично показана процедура агрегирования.

В связи с тем, что отсчеты $\Delta T, 2\Delta T, \dots, N\Delta T$ чаще всего не совпадают с отсчетами t_i исходной реализации, примем за правило (при усреднении) в

качестве последнего элемента, принадлежащего некоторому интервалу ΔT брать ближайший отсчет ξ . Иначе говоря, к интервалу ΔT - $2\Delta T$ отнесем отсчеты $\xi(t_{h+1}), \xi(t_{h+2}), \dots, \xi(t_{f+1})$. Очевидно, чем меньше значения ошибок $|q(\Delta T)|, |q(2\Delta T)|, |q(3\Delta T)|, \dots, |q(N\Delta T)|$, тем меньше искажений вносится на данном этапе.

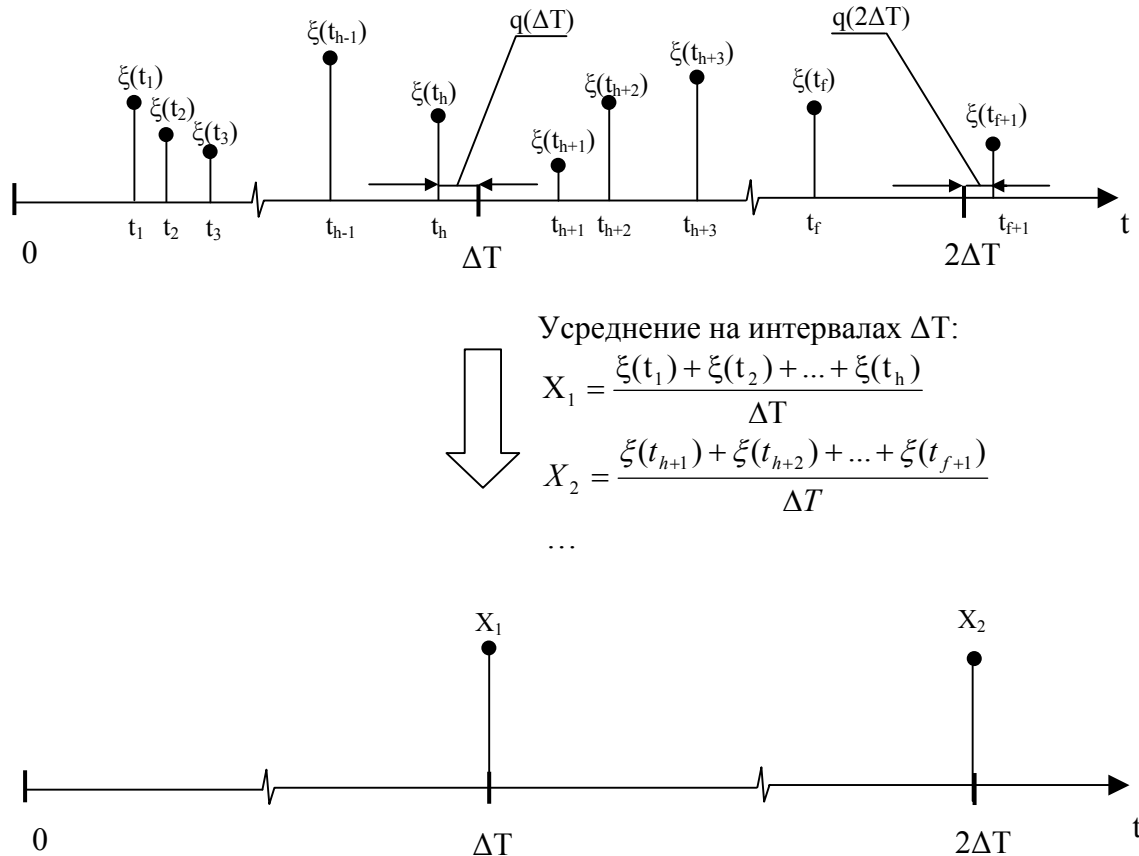
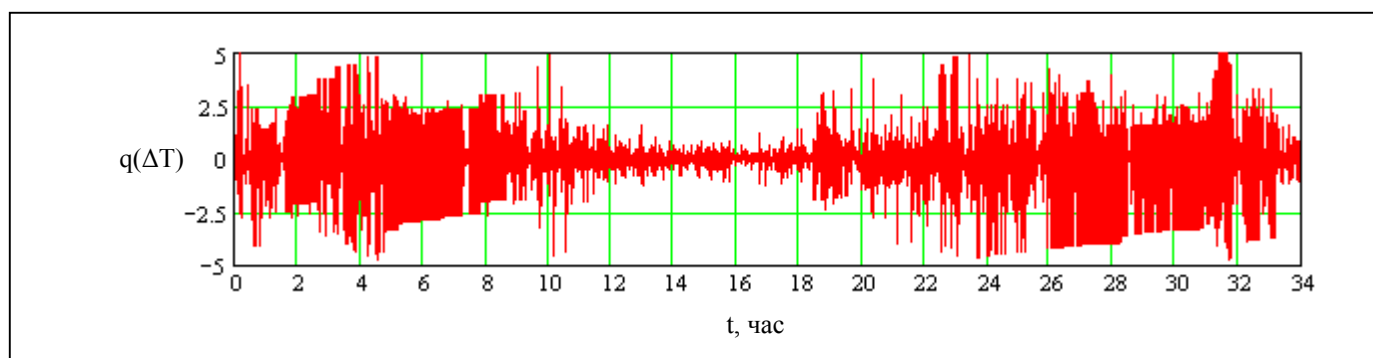
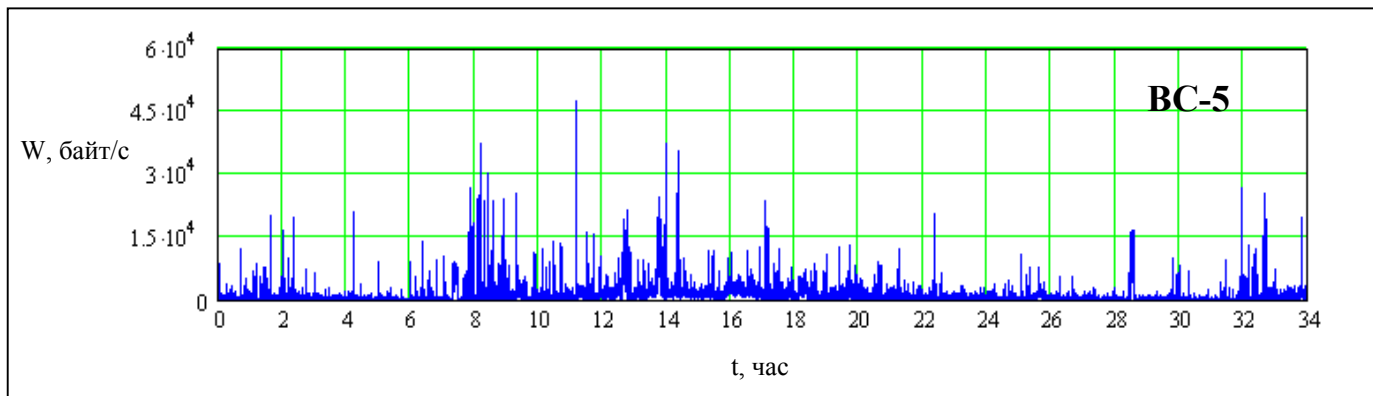


Рис. 2.7. К пояснению процедуры агрегирования реализации

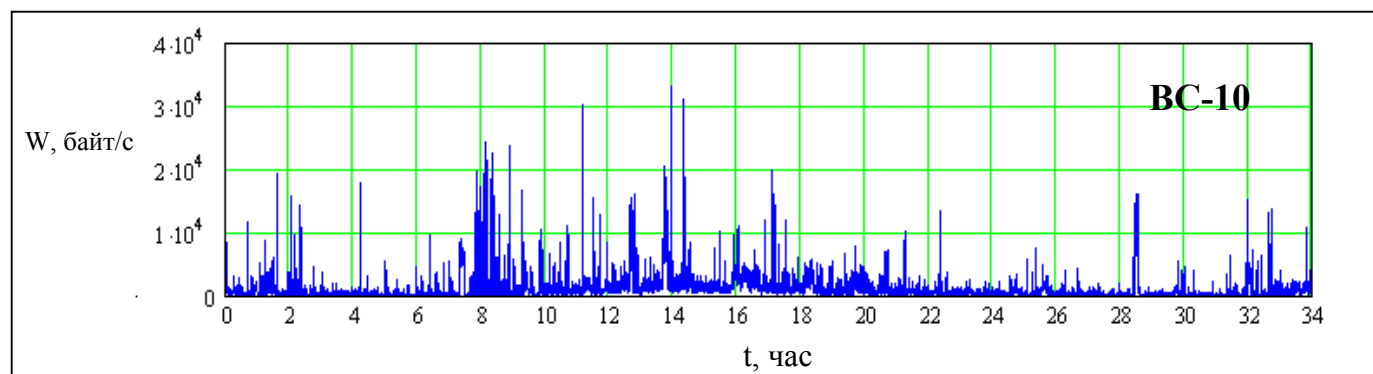
Таким образом, получается агрегированная эквидистантная реализация трафика $X = \{X(\Delta T), X(2\Delta T), \dots, X(N \cdot \Delta T)\}$, содержащая N элементов. Физический смысл каждого ее элемента – средняя скорость (байт/сек) на соответствующем интервале ΔT .

Произведем агрегирование реализации VC-Oct89Ext.TL с уровнями агрегирования $\Delta T=5$ с, $\Delta T=10$ с и $\Delta T=100$ с. В дальнейшем изложении будем обозначать получившиеся временные ряды VC-5, VC-10 и VC-100

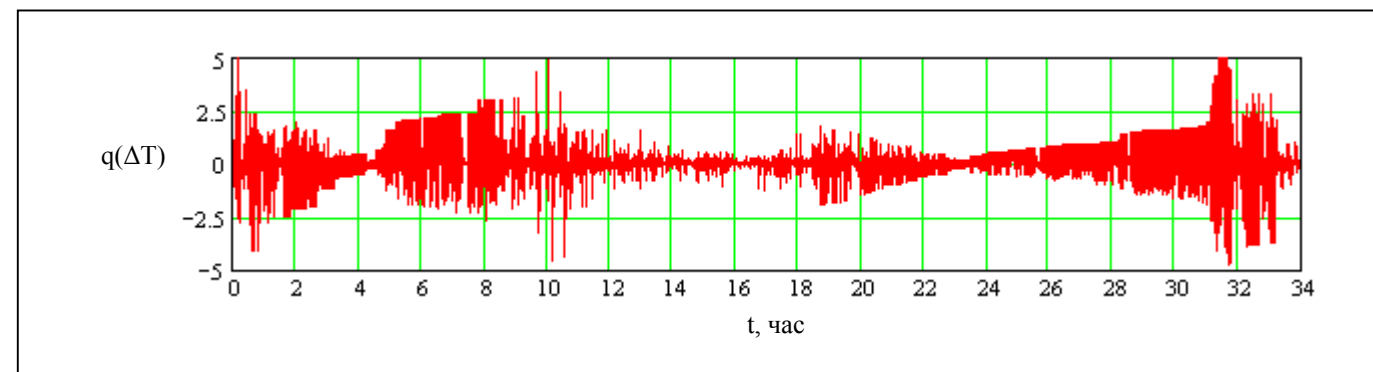
соответственно. На рис. 2.8 показан внешний вид рядов ВС-5, ВС-10 и ВС-100, а также значения ошибок $q(\Delta T)$, $q(2\Delta T)$, $q(3\Delta T), \dots, q(N\Delta T)$.



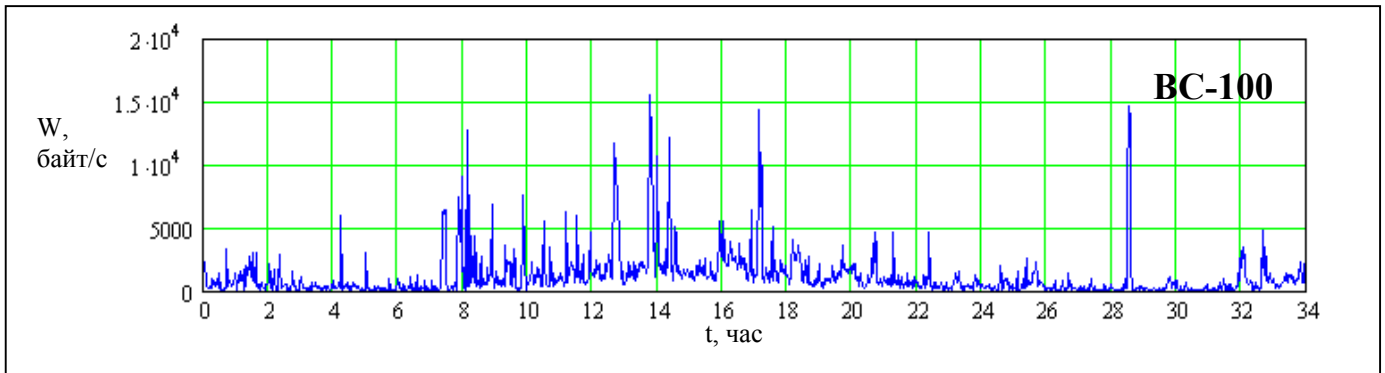
б



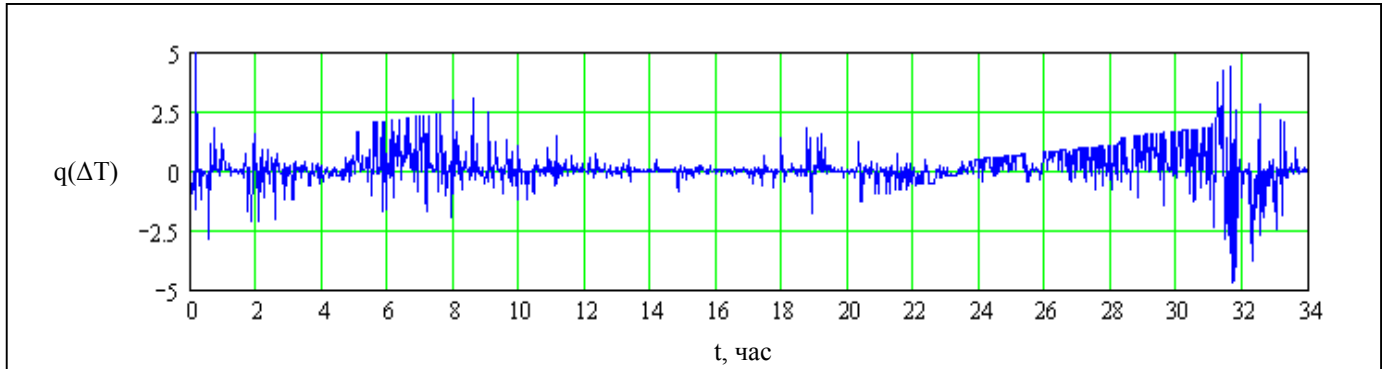
в



г



д



е

Рис. 2.8. а) Временной ряд BC-5;
 б) Ошибки $q(\Delta T)$, $q(2\Delta T)$, $q(3\Delta T)$, ..., $q(N\Delta T)$ при получении ряда BC-5;
 в) Временной ряд BC-10;
 г) Ошибки $q(\Delta T)$, $q(2\Delta T)$, $q(3\Delta T)$, ..., $q(N\Delta T)$ при получении ряда BC-10;
 д) Временной ряд BC-100;
 е) Ошибки $q(\Delta T)$, $q(2\Delta T)$, $q(3\Delta T)$, ..., $q(N\Delta T)$ при получении ряда BC-100

Еще одной характеристикой, интересной с точки зрения изучения реализации трафика, может служить зависимость $|t_{i+1} - t_i|$ от времени, где $i \in 1, n$, (см. рис. 2.7), которая служит наглядным отражением степени неэквиливантности исходной реализации (см. рис. 2.9).

Интересно, что в промежуток примерно с 11:00 до 18:00 наблюдаются минимальные значения $|t_{i+1} - t_i|$, которые Ethernet-кадры на данном интервале времени следуют друг за другом максимально плотно. Этот факт может объясняться высокой активностью использования сетевых ресурсов в указанное время суток, что, в свою очередь, скорее всего, связано с продолжительностью рабочего дня. Аналогично, можно отметить, что на данном интервале времени наблюдаются минимальные ошибки (см. рис. 2.8)

при агрегировании. Очевидно, последнее является следствием высокой плотности следования пакетов.

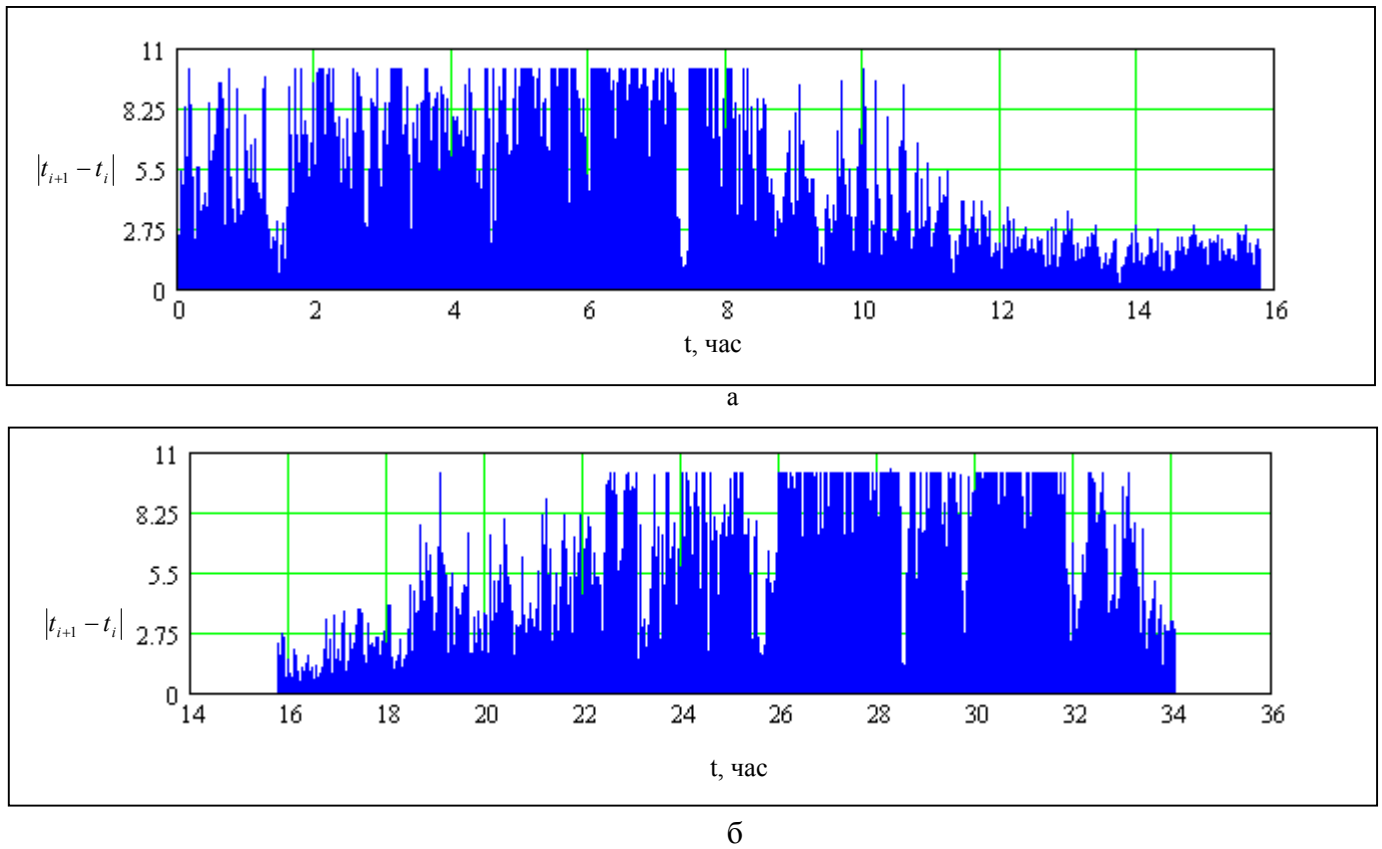
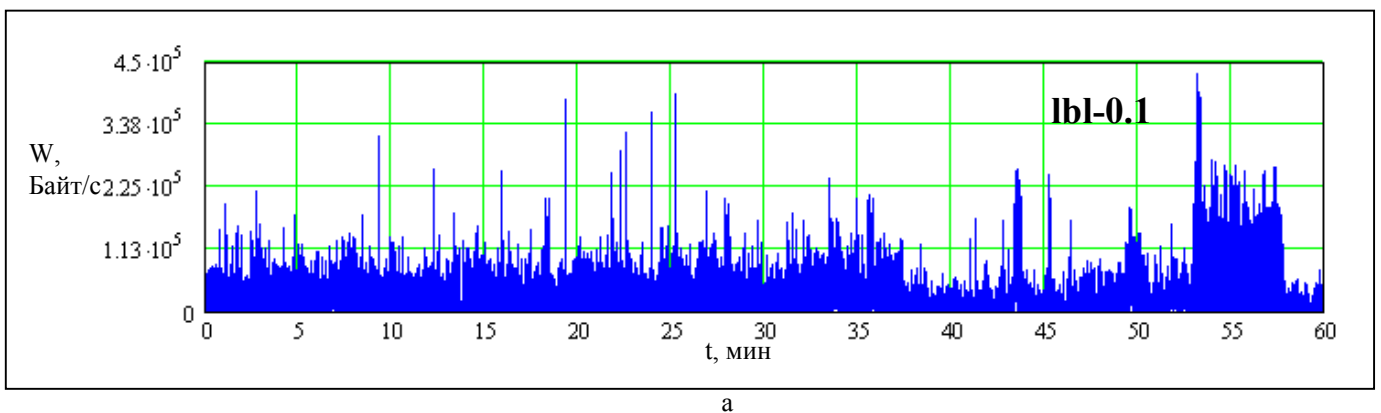
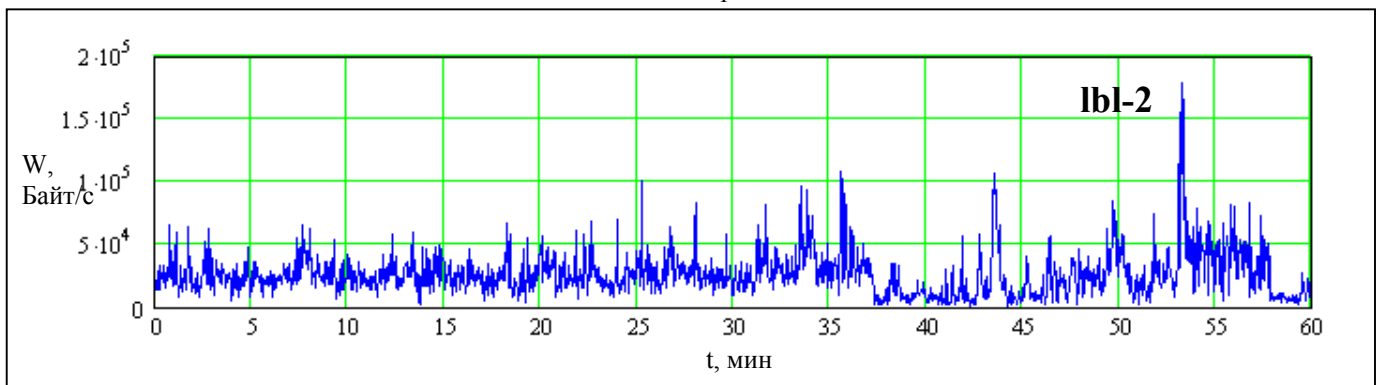
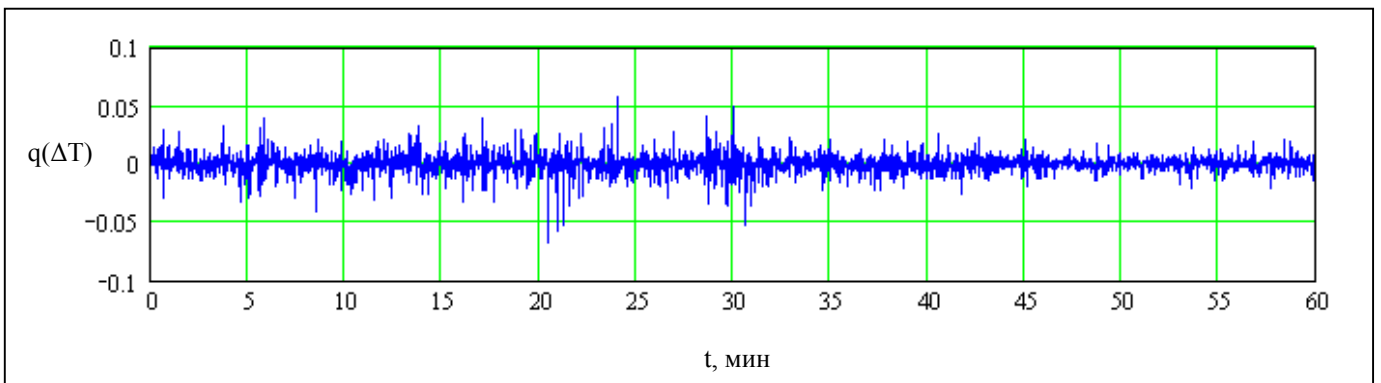
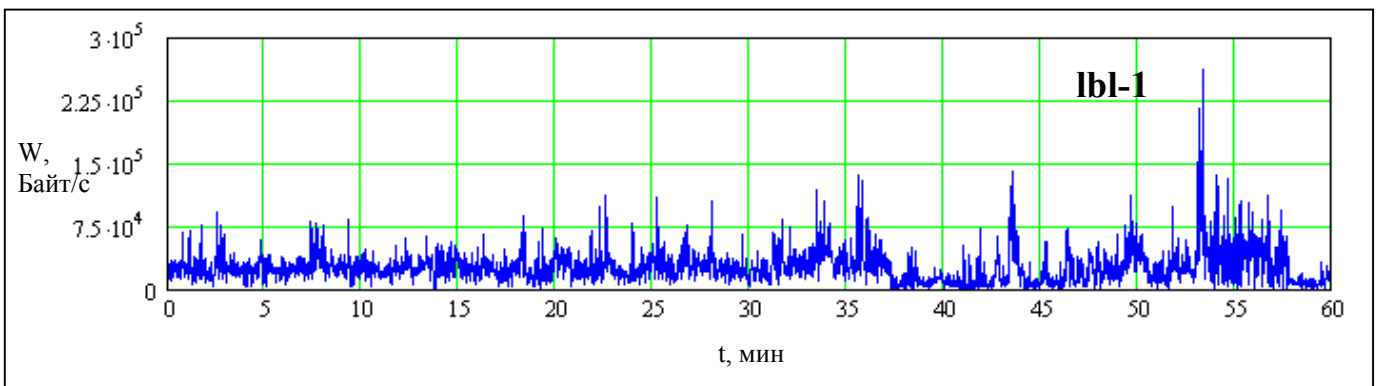
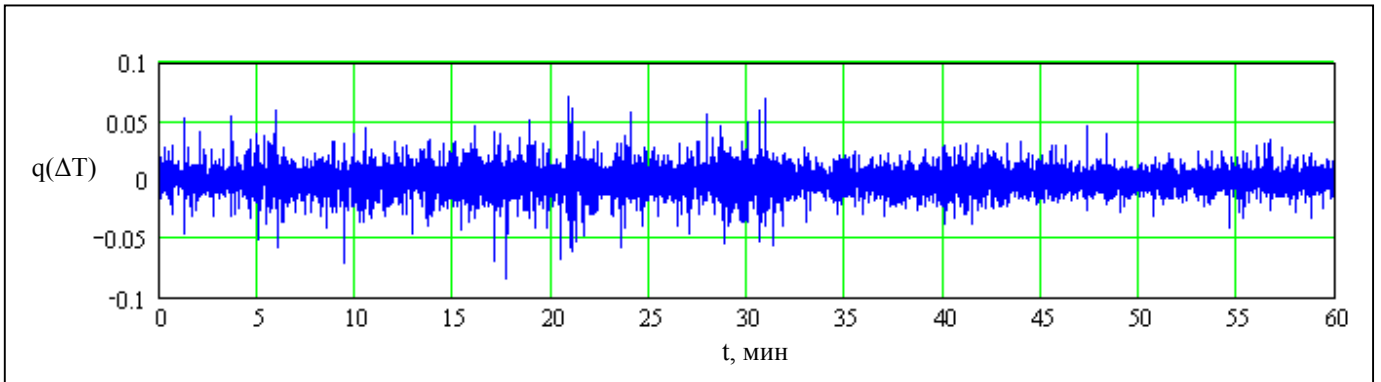


Рис. 2.9. Первые разности временных отсчетов исходной реализации BC-Oct89Ext.TL

Произведем агрегирование реализации LBL-РКТ-5.TCP с уровнями $\Delta T=0.1$ с, $\Delta T=1$ с и $\Delta T=2$ с. Получившиеся ряды назовем $lbl-0.1$, $lbl-1$, и $lbl-2$ соответственно (см. рис. 2.10).





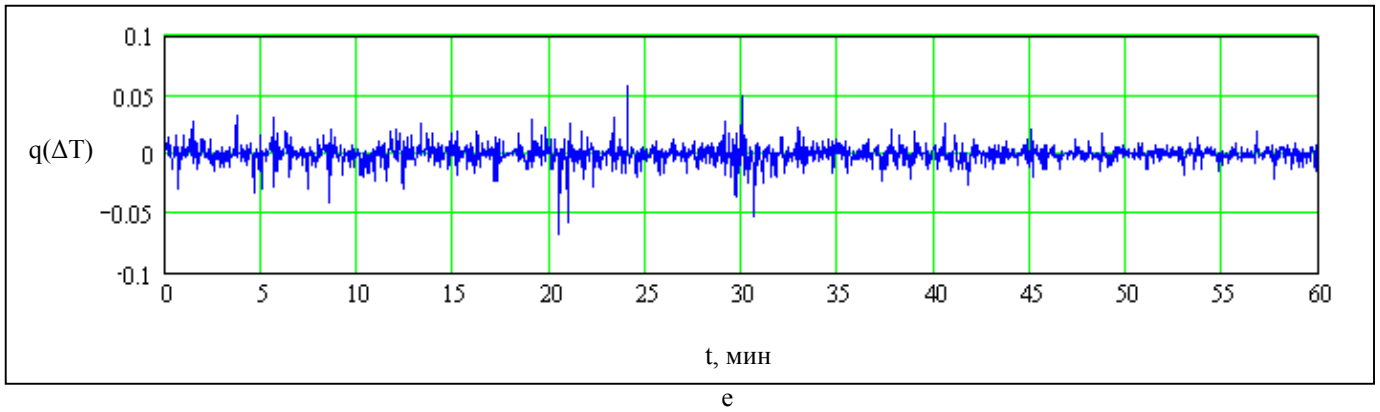


Рис. 2.10. а) Временной ряд $l_{bl-0.1}$;
 б) Ошибки $q(\Delta T), q(2\Delta T), q(3\Delta T), \dots, q(N\Delta T)$ при получении ряда $l_{bl-0.1}$;
 в) Временной ряд l_{bl-1} ;
 г) Ошибки $q(\Delta T), q(2\Delta T), q(3\Delta T), \dots, q(N\Delta T)$ при получении ряда l_{bl-1} ;
 д) Временной ряд l_{bl-2} ;
 е) Ошибки $q(\Delta T), q(2\Delta T), q(3\Delta T), \dots, q(N\Delta T)$ при получении ряда l_{bl-2}

Первые разности временных отсчетов исходной реализации LBL-РКТ-5.TCP приведены на рис. 2.11. В отличие от реализации BC-Oct89Ext.TL трафик LBL-РКТ-5.TCP (возможно благодаря более короткой длительности последнего) характеризуется более равномерным поступлением дэйтаграмм.

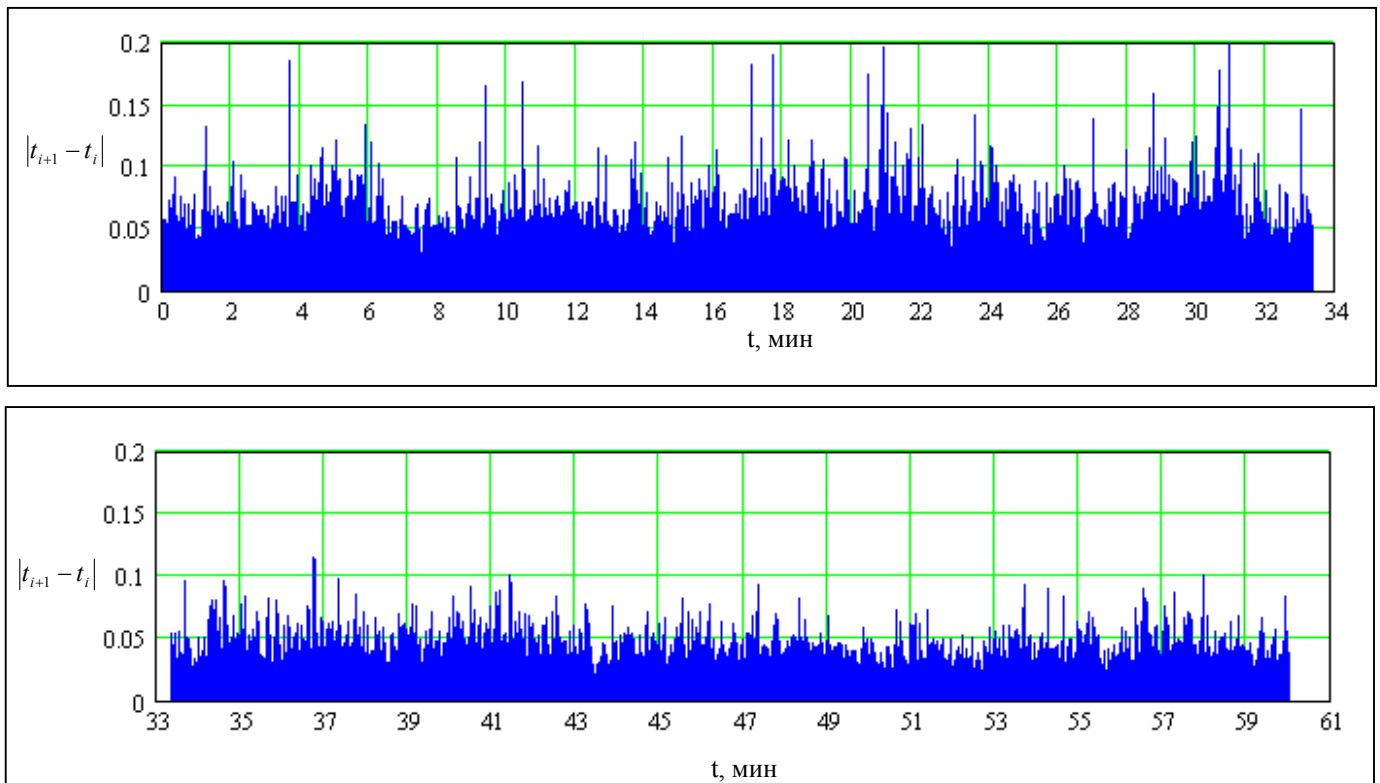


Рис. 2.11. Зависимость $|t_{i+1} - t_i|$ от времени, где $i \in 1, n$ для реализации LBL-РКТ-5.TCP

Агрегированная по уровням $\Delta T=0.1c$ и $\Delta T=1c$ и $\Delta T=2c$ реализация LBL-ТСП-3.ТСП представлена на рис. 2.12.

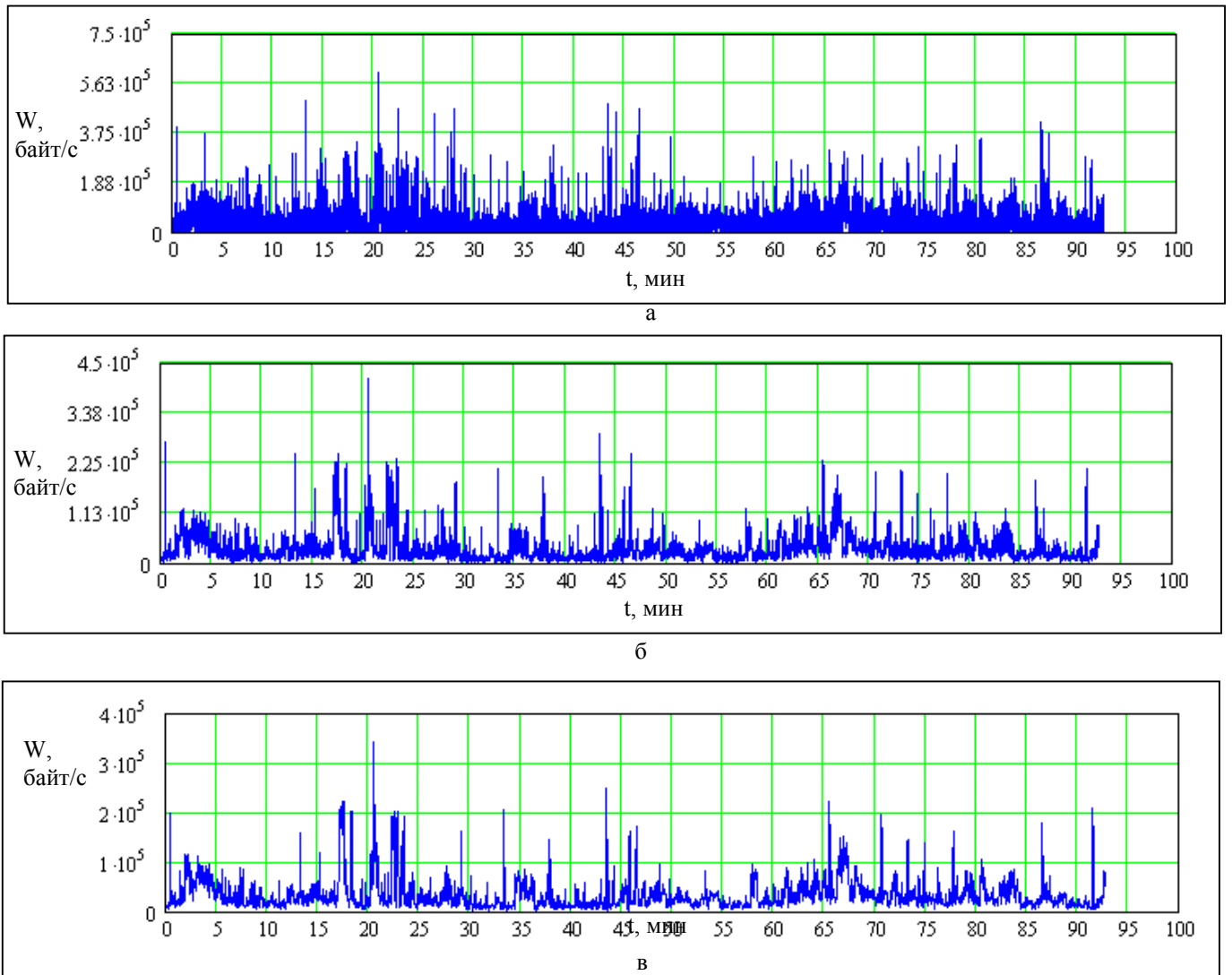
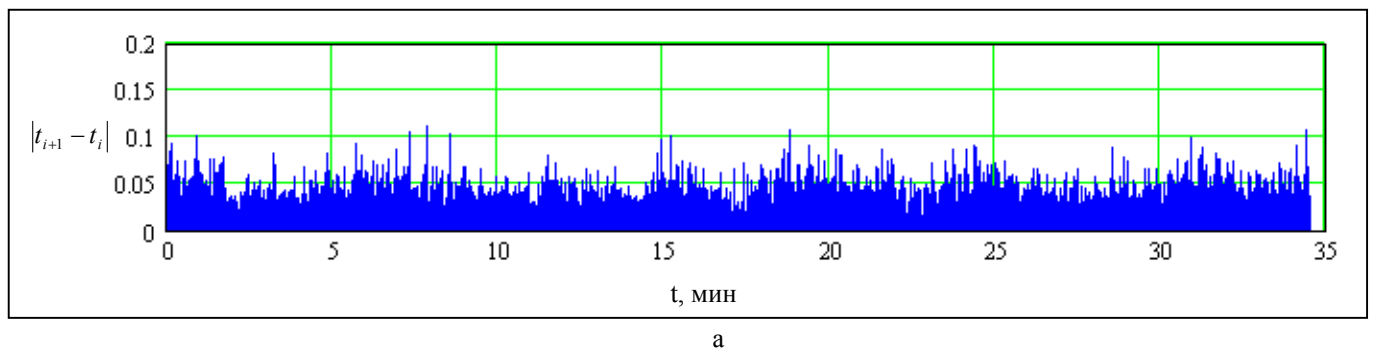


Рис. 2.12. Реализация LBL-ТСП-3.ТСП при различных уровнях усреднения: а) $\Delta T=0.1c$; б) $\Delta T=1c$; в) $\Delta T=2c$



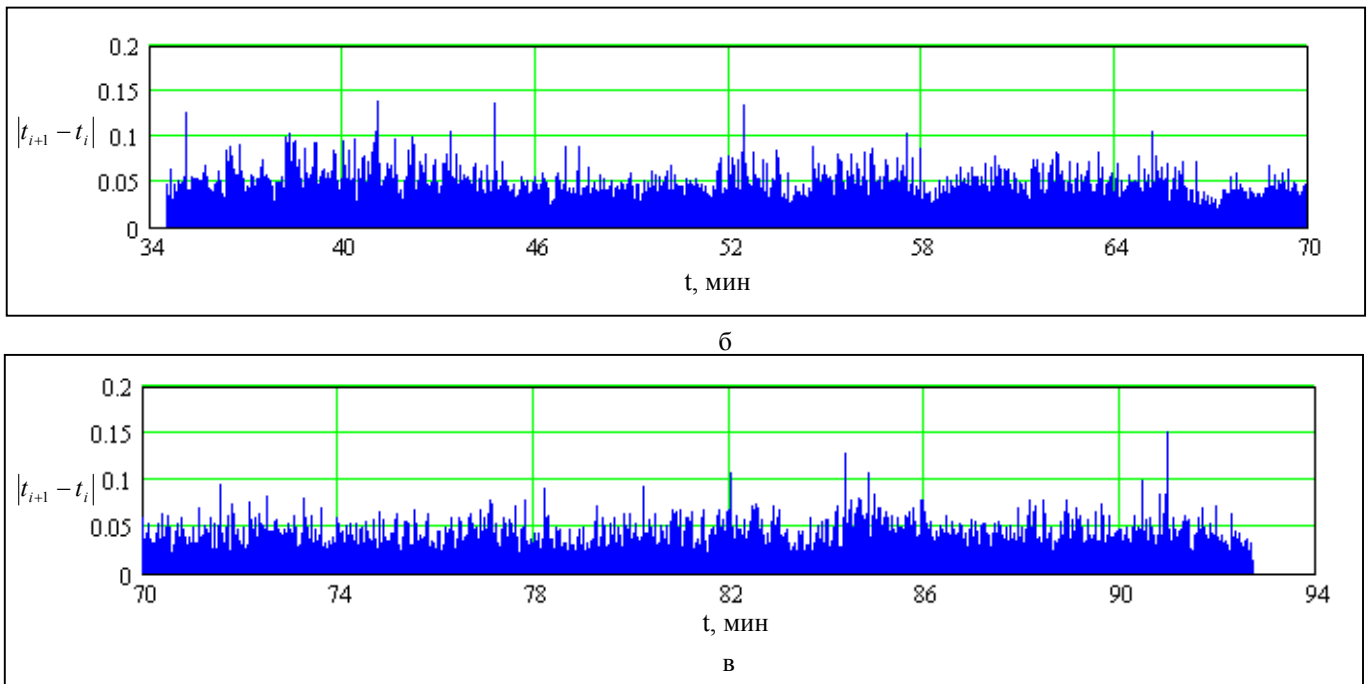


Рис. 2.13. Первые разности временных отсчетов реализации LBL-ТСП-3.ТСП

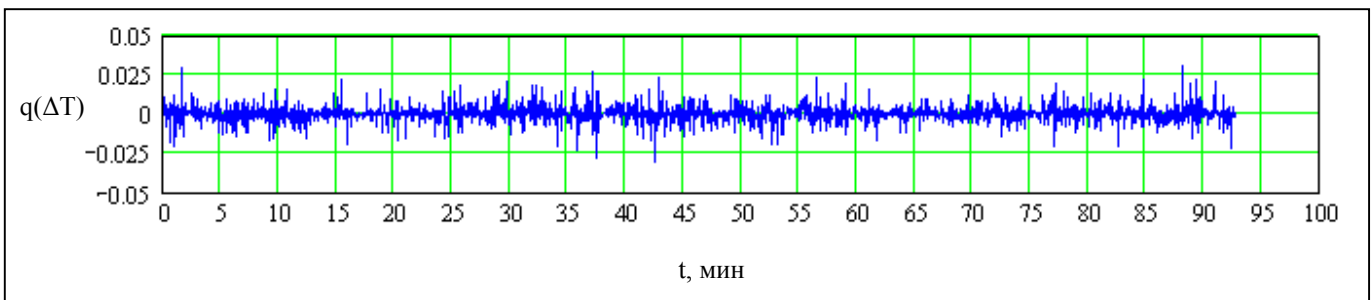


Рис. 2.14. Ошибки $q(\Delta T)$, $q(2\Delta T)$, $q(3\Delta T)$, ..., $q(N\Delta T)$ при получении ряда ТСП-2

Полученные временные ряды будут нами анализироваться далее.

2.2.2 Тестовые реализации (хаос и белый шум)

Для дальнейшего анализа в качестве тестовых рядов нам также понадобятся реализации чисто детерминированного и чисто случайного процессов. В качестве первой выберем часто используемую в литературе зависимость (далее - ряд L) одной из переменных хаотической системы Лоренца [38] (см. рис. 2.15):

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= rx - y - xz \\ \frac{dz}{dt} &= -bz + xy\end{aligned}$$

при “стандартных” параметрах $\sigma = 10$, $r = 28$, $b = 8/3$.

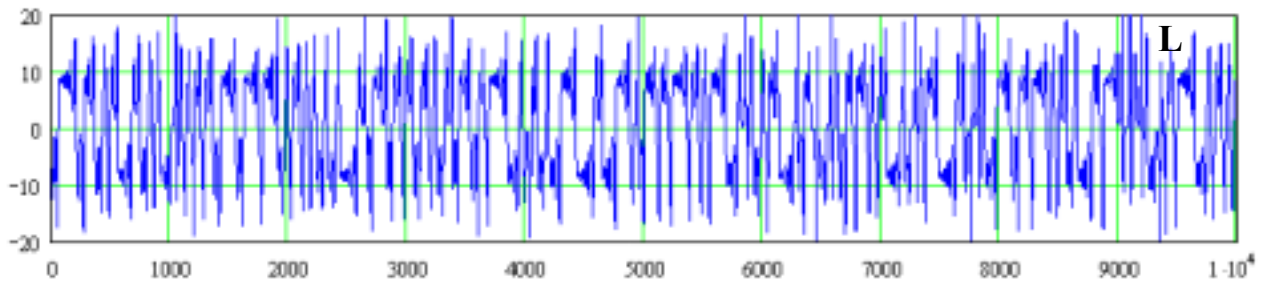


Рис. 2.15. Зависимость от времени одной из переменных хаотической системы Лоренца, полученная при помощи программы Fractan 4.4 [82]

В качестве чисто случайного процесса выберем реализацию (ряд N далее) классического белого гауссовского шума с нулевым средним и среднеквадратическим отклонением σ .

Пример такого процесса при $\sigma = 1000$ представлен на рис. 2.16.

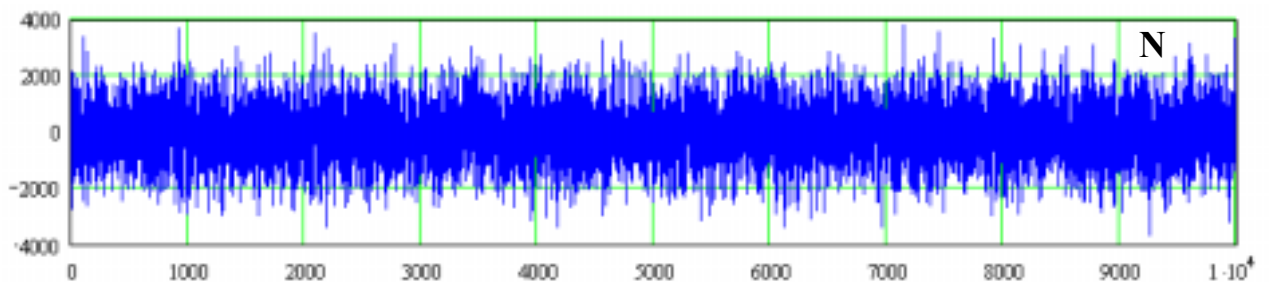


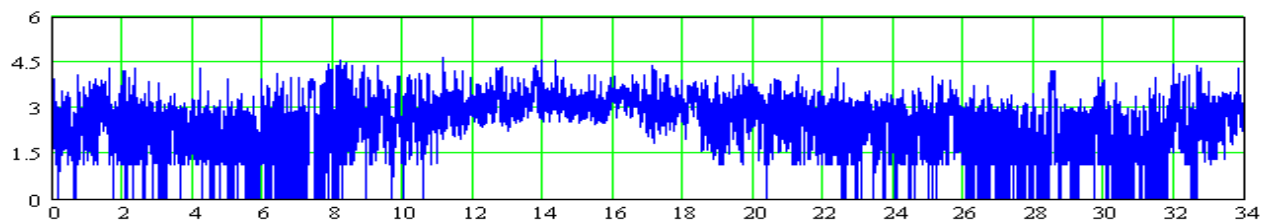
Рис. 2.16. Пример реализации белого гауссовского шума. Ряд получен при помощи программы Fractan 4.4 [82]

2.2.3 Логарифмированные реализации

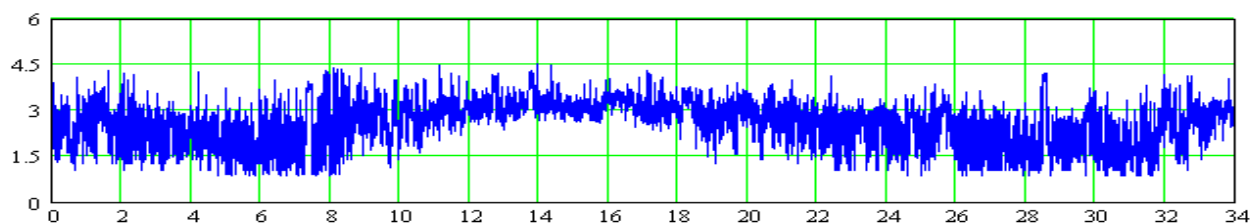
Построим также логарифмы эквидистантных рядов ВС-5, ВС-10, ВС-100, LBL-0.1, LBL-1, LBL-2, TCP-0.1, TCP-1, TCP-2 в соответствии с (2.1)

$$XL_i = \log(X_i + 1), i \in 1, N. \quad (2.1)$$

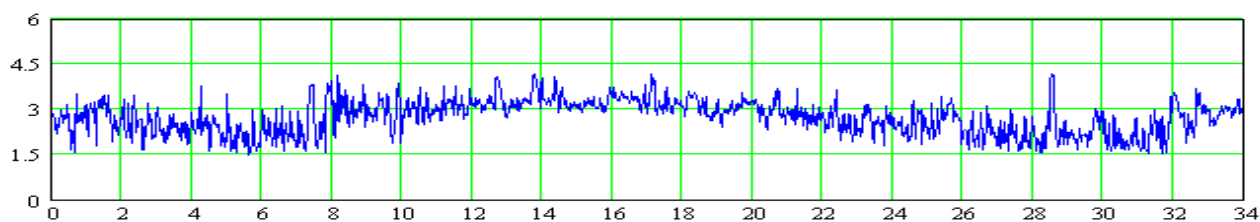
Полученные ряды назовем BC-5-LOG, BC-10-LOG, BC-100-LOG, LBL-0.1-LOG, LBL-1-LOG, LBL-2-LOG, TCP-0.1-LOG, TCP-1-LOG и TCP-2-LOG.



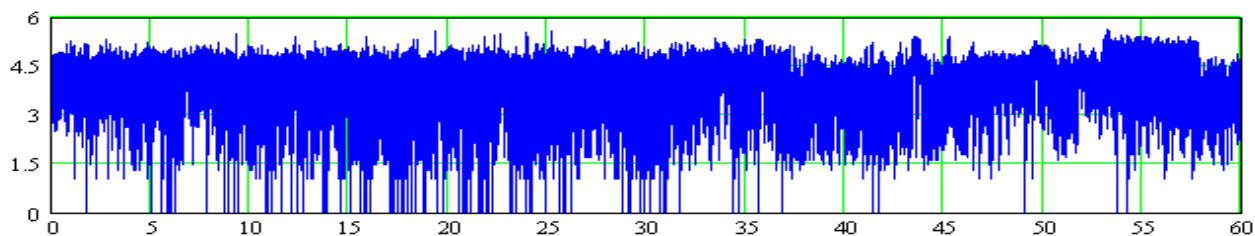
а



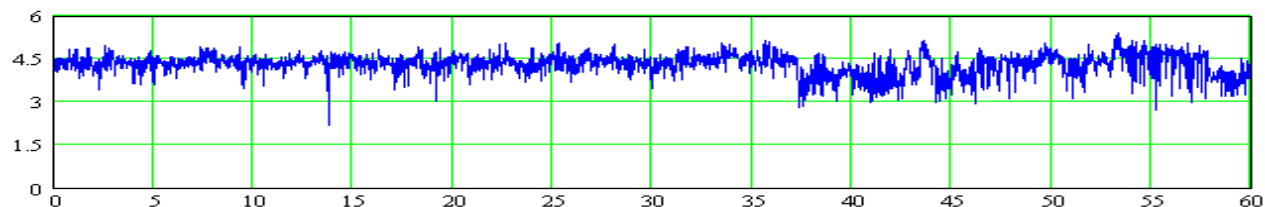
б



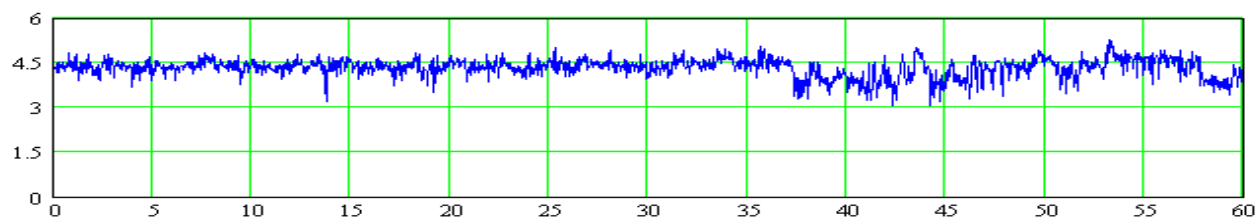
в



г



д



е

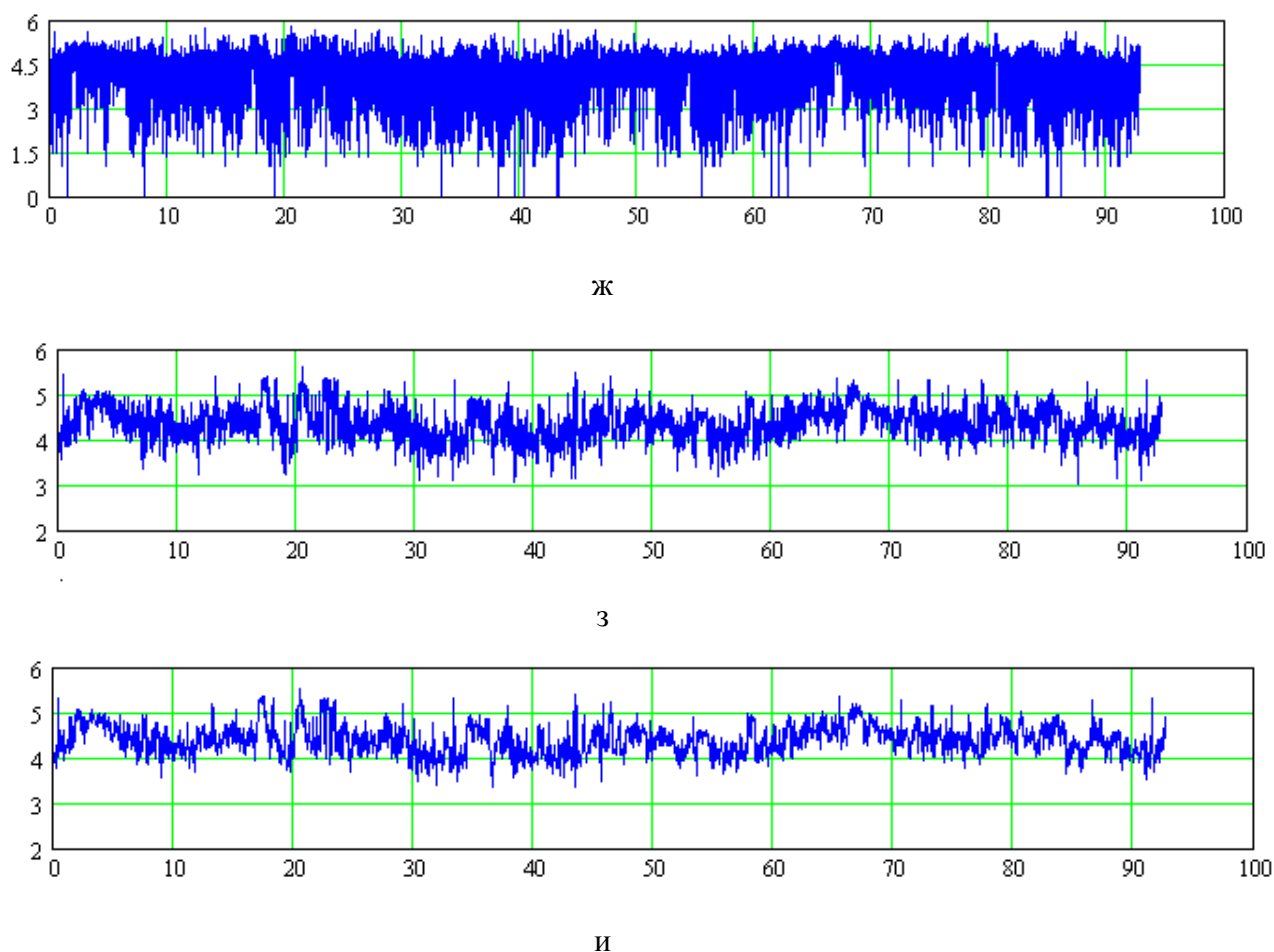


Рис. 2.17. Временные ряды
а) BC-5 LOG; б) BC-10-LOG; в) BC-100-LOG; г) LBL-0.1-LOG; д) LBL-1-LOG;
е) LBL-2-LOG, ж) TCP-0.1-LOG, з) TCP-1-LOG, и) TCP-2-LOG

2.3 Классический анализ

В таблице 2.1 приведены основные характеристики исследуемых реализаций, полученные в результате статистического анализа: среднеквадратическое отклонение σ , среднее m , количество N отсчетов (членов) ряда, а также максимально возможная корреляционная размерность $D_{\max} = 2 \lg(N)$, которую можно оценить при помощи корреляционного интеграла по ряду данной длины (см. ниже).

Табл. 2.1. Основные характеристики исследуемых реализаций

Реализация/ Характеристика	σ	m	N	D_{\max}
BC-5	2405	1 142	24 560	8.782
BC-10	2243	1 142	12 279	8.178
BC-100	1779	1 142	1 227	6.178

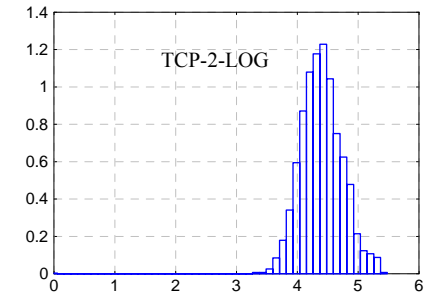
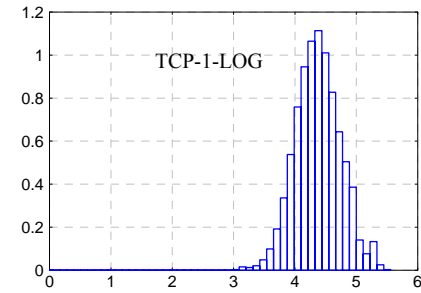
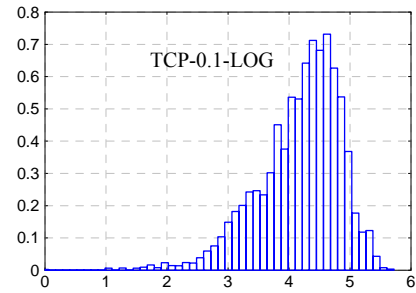
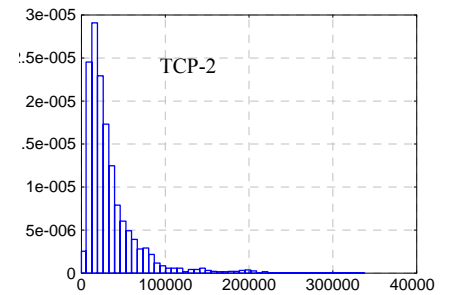
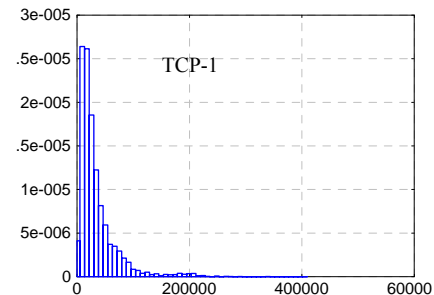
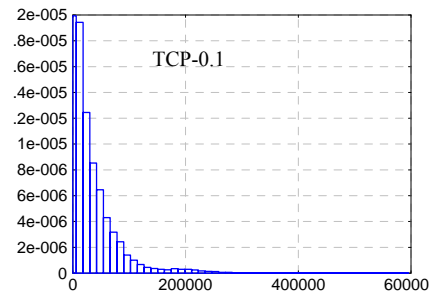
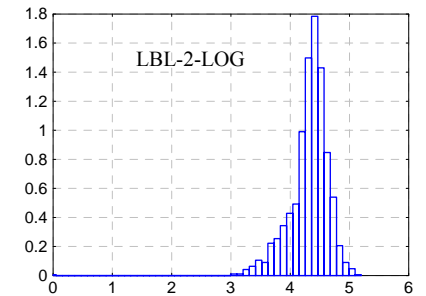
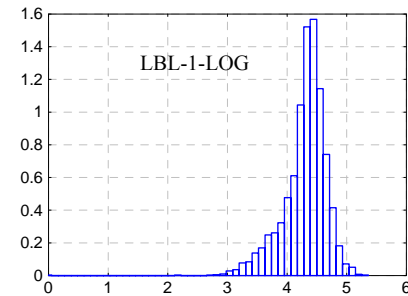
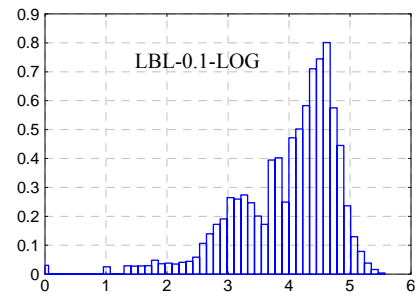
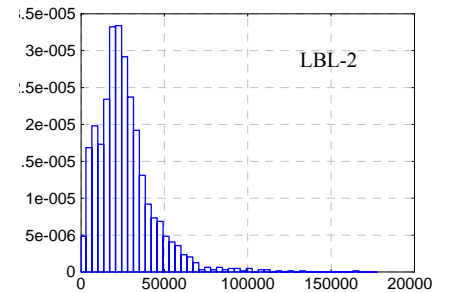
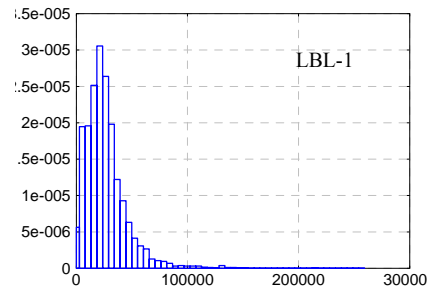
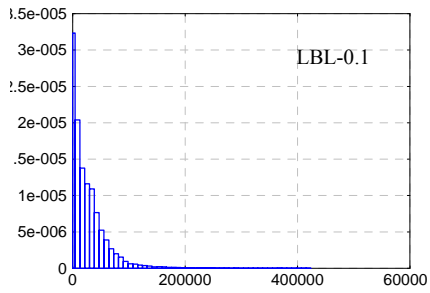
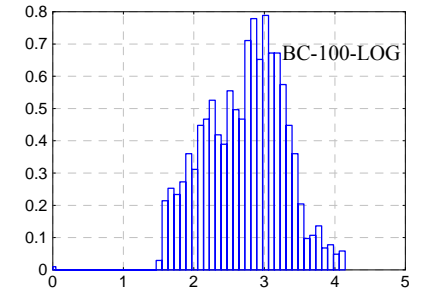
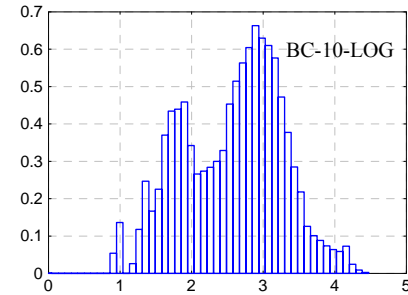
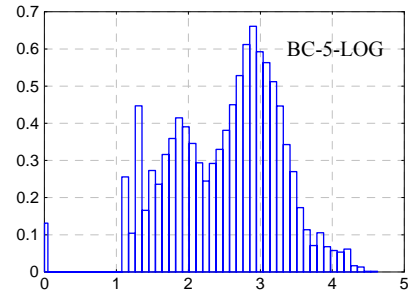
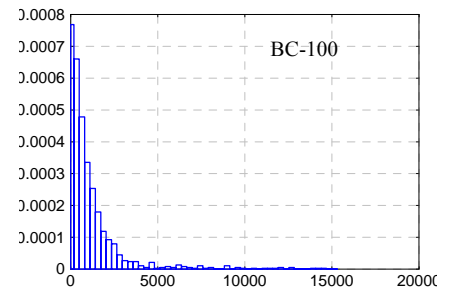
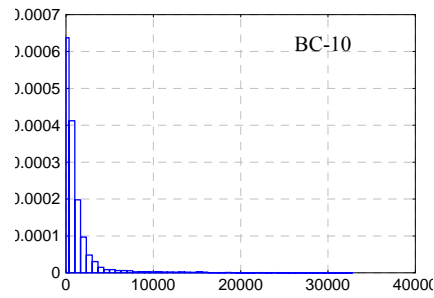
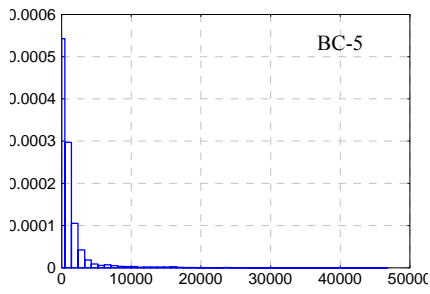
BC-5-LOG	0.775	2.508	24 560	8.782
BC-10-LOG	0.702	2 572	12 279	8.178
BC-100-LOG	0.565	2.721	1 227	6.178
BC-5-surr-001	2409	1 139	24 300	8.771
BC-10-LOG-surr-001	0.704	2.567	12 150	8.169
LBL-0.1	32 250	26 146	36 000	9.113
LBL-1	19 331	26 151	3 600	7.113
LBL-2	17 174	26 157	1 800	6.511
LBL-0.1-LOG	0.821	3.977	36 000	9.113
LBL-1-LOG	0.365	4.297	3 600	7.113
LBL-2-LOG	0.328	4.320	1 800	6.511
LBL-0.1-surr-001	32 520	26 639	34 992	9.088
LBL-0.1-LOG-surr-001	0.824	3.989	34 992	9.088
TCP-0.1	43 420	34 107	55 634	9.491
TCP-1	34 773	34 109	5 563	7.491
TCP-2	32 528	34 100	2 781	6.888
TCP-0.1-LOG	0.674	4.175	55 634	9.491
TCP-1-LOG	0.373	4.374	5 563	7.491
TCP-2-LOG	0.344	4.396	2 781	6.888
TCP-0.1-surr-001	43 474	34 092	55 296	9.485
TCP-0.1-LOG-surr-001	0.674	4.175	55 296	9.485
N	1 010	-7.892	10 000	8.000
L	8.450	0.528	10 000	8.000

Некоторые из перечисленных в табл.2.1 реализаций (с суффиксами “-surr-”) также будут нами рассматриваться ниже.

Изучая результаты измерений, можно заметить, что для нелогарифмированных реализаций (BC, LBL, TCP) при увеличении уровня агрегирования наблюдается уменьшение среднеквадратического отклонения на фоне практически неизменного среднего. Данный факт говорит о “сглаживании” реализации при увеличении уровня агрегирования.

2.3.1 Плотности распределения

В настоящем разделе производится анализ плотностей распределения агрегированных реализаций BC, LBL, TCP а также тестовых реализаций N и L. Оценка плотностей распределения производится на основании геометрического изображения (гистограмм) относительных частот появления предварительно заданных участков данных (см. рис. 2.18).



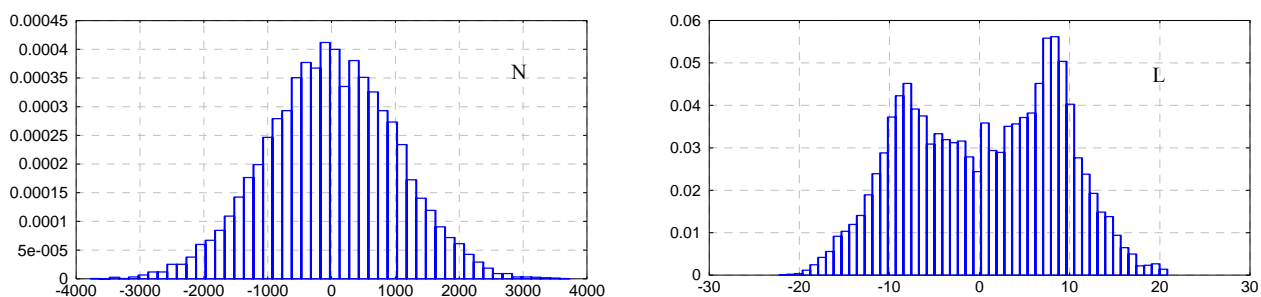


Рис. 2.18. Гистограммы изучаемых реализаций

Визуальный анализ гистограмм изучаемых временных рядов позволяет сделать следующие выводы:

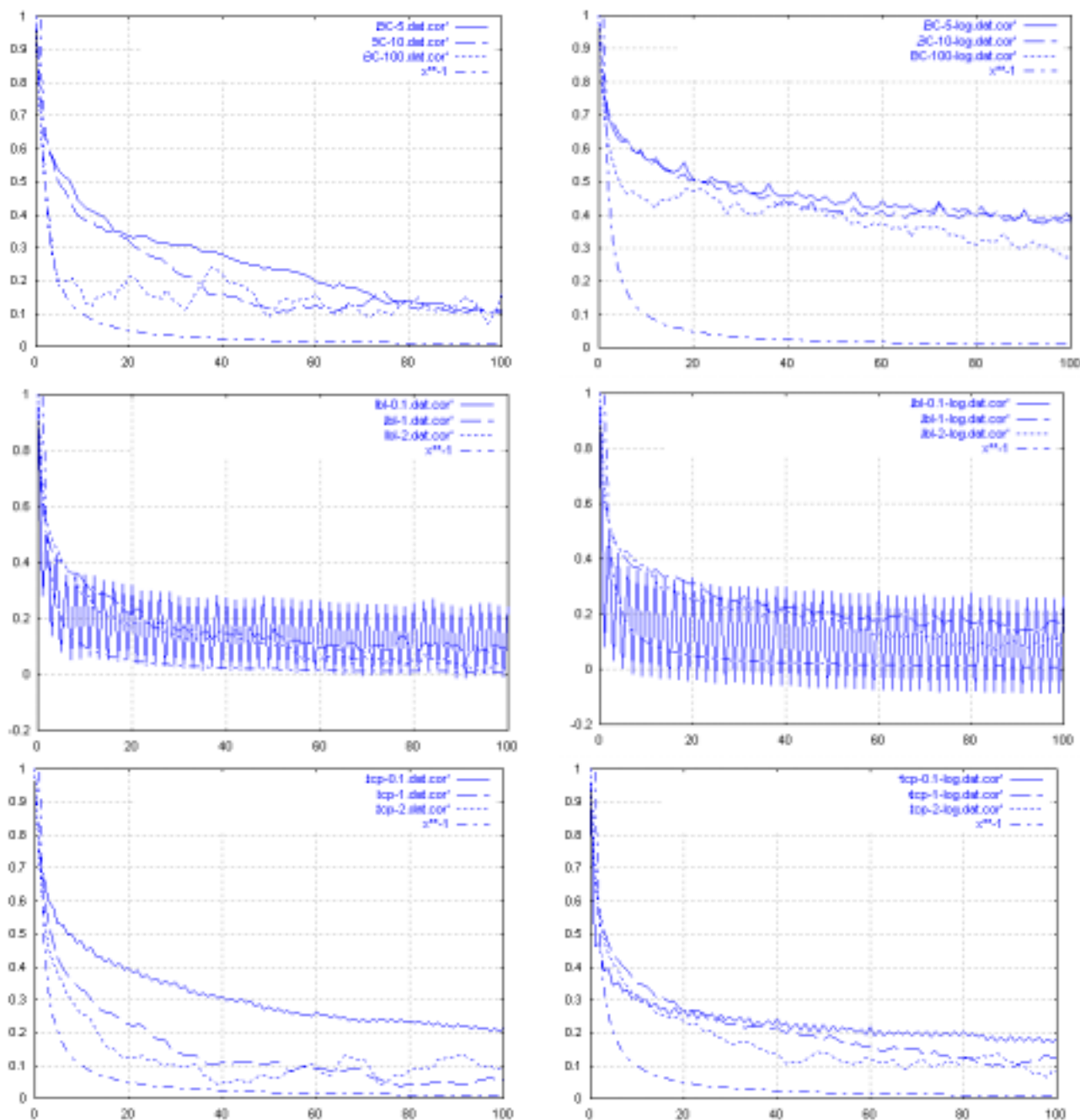
- логарифмирование рядов приводит к нормализации их распределений;
- ряды ВС-х, LBL-х, а также ТСР-х, скорее всего, подчиняются некоторому распределению с так называемым “тяжелым хвостом” (например, Парето);
- увеличение уровня агрегирования приводит к изменению параметров распределения. Более точный ответ может дать исследование распределений методом максимального правдоподобия;
- гистограмма хаотической реализации L имеет два характерных максимума (при значениях ~ -9 и ~ 9). Аналогичные максимумы наблюдаются в реализации ВС-10-LOG и (менее выраженные) в LBL-0.1-LOG.

2.3.2 Автокорреляционные функции

На рис. 2.19 приведены графики автокорреляционных функций исследуемых временных рядов. Вычисления производились по классической формуле

$$r(k) = \frac{1}{N - \tau} \cdot \frac{\sum_i (X_i - \bar{X})(X_{i+k} - \bar{X})}{\sigma^2(X)}, \quad (2.2)$$

где \bar{X} - выборочное среднее ряда X , $\sigma^2(X)$ - выборочная дисперсия ряда X , $k \in Z_+ = \{0,1,2,\dots\}$.



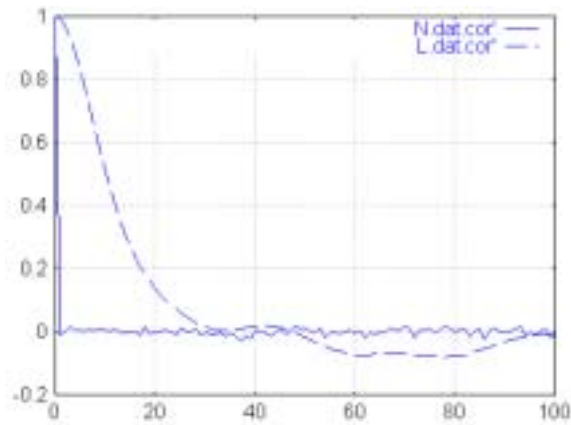


Рис. 2.19. Графики автокорреляционных функций $r(k)$

Заметим, что АКФ рядов LBL-0.1 и LBL-0.1-LOG имеют отчетливую периодическую структуру (с частотой ~ 5 Гц). Менее выраженные периодичности (пульсации) АКФ имеются в рядах BC-5-LOG и BC-10-LOG, а также в рядах TSP-0.1 и TSP-0.1-LOG. Можно предположить наличие соответствующих гармоник в спектрах данных рядов, в частности для рядов LBL-0.1, LBL-0.1-LOG, TSP-0.1 и TSP-0.1-LOG – выделяющейся гармоники с частотой 5 Гц.

Выясним, какие из исследуемых временных рядов обладают медленно убывающей зависимостью (МУЗ), а какие быстро убывающей зависимостью (БУЗ).

На каждом графике рис. 2.19 для сравнения также изображена зависимость $C(k) = k^{-\beta}$ при $\beta = 1$, соответствующая предельному случаю существования МУЗ.

Для того чтобы проверить гипотезу о присутствии МУЗ в некотором временном ряде, необходимо решить задачу регрессии, т. е. вычислить по экспериментальной АКФ методом наименьших квадратов параметры A_0 и β модели (2.3):

$$r(k) = A_0 \cdot k^{-\beta}, \text{ при } k \rightarrow \infty, \quad (2.3)$$

где $0 < \beta < 1$ и $A_0 = \text{const}$. Для оценки степени точности необходимо найти дисперсию $\sigma^2_{\text{МУЗ}}$ разности экспериментальной АКФ и полученной ее аппроксимации (2.3).

Одновременно проверяется гипотеза о присутствии БУЗ в этом же ряде. Для этого аналогичным образом вычисляются параметры B_0 и ρ модели (2.4):

$$r(k) = B_0 \cdot \rho^k, \text{ при } k \rightarrow \infty, \quad (2.4)$$

где $0 < \rho < 1$ и $B_0 = \text{const}$. Для оценки степени точности необходимо найти дисперсию $\sigma^2_{\text{БУЗ}}$ разности экспериментальной АКФ и полученной ее аппроксимации (2.4).

Вывод о присутствии МУЗ в ряде логично сделать, если $\sigma^2_{\text{МУЗ}}$ достаточно мало и значение коэффициента

$$LR = \frac{\sigma^2_{\text{МУЗ}}}{\sigma^2_{\text{БУЗ}}} \quad (2.5)$$

меньше 1. Напротив, будем говорить, что в исследуемом ряде присутствует БУЗ, если $LR > 1$ и значение $\sigma^2_{\text{БУЗ}}$ мало. Конечно, условие малости $\sigma^2_{\text{МУЗ}}$ или $\sigma^2_{\text{БУЗ}}$ достаточно условно и отражает лишь тот факт, что степень аппроксимации экспериментальной зависимости выражениями (2.3) или (2.4) нас удовлетворяет в плане корректности выбора модели. Выполнимость данного условия проще всего проконтролировать по графикам, на которых изображены одновременно экспериментальная АКФ и модели. Поскольку (2.3) и (2.4) справедливы для $k \rightarrow \infty$, хотя экспериментальные данные (АКФ) ограничены количеством отсчетов M , условимся, что регрессию будем осуществлять для $k = \frac{M}{10}, \dots, M$. То есть будем рассматривать асимптотику.

Заметим, что реализации LBL-0.1 и LBL-0.1-LOG не имеет смысла аппроксимировать моделями (2.3) или (2.4), поскольку они имеют сильно выраженную периодическую структуру. Результаты вычислений приведены в таблице 2.2.

Анализируя полученные результаты, отметим, что в большинстве изучаемых случаев коэффициент $LR < 1$, что подтверждает присутствие в них МУЗ. Исключения составляют ряды ВС-100, LBL-1, LBL-2, LBL-1-LOG и LBL-2-LOG (выделенные участки таблицы 2.2). Однако, рассматривая более внимательно параметры модели БУЗ согласно (2.4), можно заметить, что абсолютно для всех случаев $\rho \sim 1$. Другими словами модель (2.4) вырождается в (2.6):

$$r(k) \sim \text{const}, \quad (2.6)$$

что говорит об очень медленном убывании АКФ.

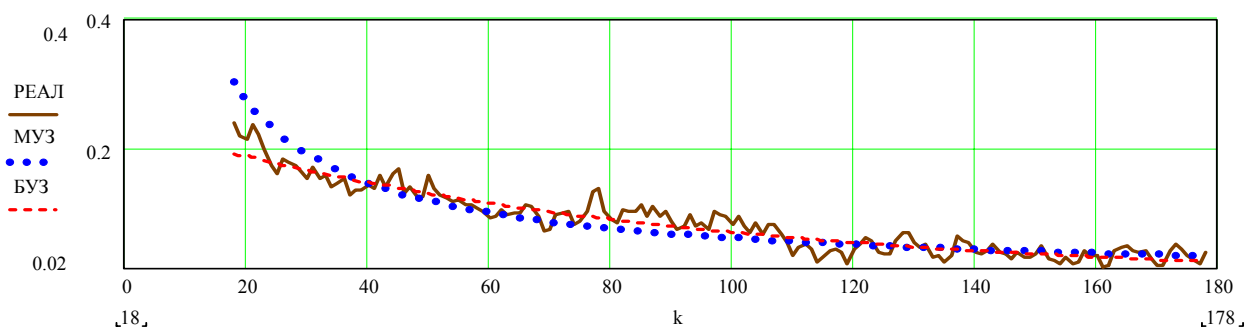


Рис. 2.20. Регрессионный анализ ряда LBL-1

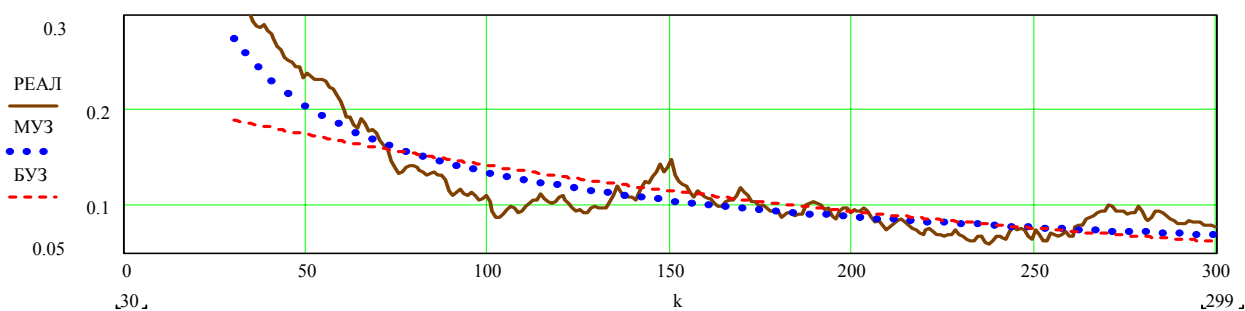


Рис. 2.21. Регрессионный анализ ряда BC-5

Для примера на рис. 2.20 и 2.21 даны результаты аппроксимации АКФ реализаций LBL-1 и BC-5 с помощью моделей (2.3) и (2.4).

Табл. 2.2. Результаты регрессионного анализа

BC-5, M=300,	BC-10, M=300,	BC-100, M=150,
--------------	---------------	----------------

$A_0 = 2.25, \beta = 0.611, \sigma_{\text{МУЗ}}^2 = 0.000411$ $B_0 = 0.214, \rho = 0.9959, \sigma_{\text{БУЗ}}^2 = 0.00116$ LR= 0.354404	$A_0 = 0.24, \beta = 0.162, \sigma_{\text{МУЗ}}^2 = 0.000435$ $B_0 = 0.126, \rho = 0.9990, \sigma_{\text{БУЗ}}^2 = 0.00051$ LR= 0.850971	$A_0 = 0.422, \beta = 0.280, \sigma_{\text{МУЗ}}^2 = 0.001214$ $B_0 = 0.180, \rho = 0.9958, \sigma_{\text{БУЗ}}^2 = 0.00122$ LR= 0.992989
BC-5-LOG, M=300, $A_0 = 0.766, \beta = 0.145, \sigma_{\text{МУЗ}}^2 = 0.000174$ $B_0 = 0.4398, \rho = 0.9990, \sigma_{\text{БУЗ}}^2 = 0.00037$ LR= 0.473328	BC-10-LOG, M=300, $A_0 = 0.511, \beta = 0.054, \sigma_{\text{МУЗ}}^2 = 0.000242$ $B_0 = 0.4131, \rho = 0.9997, \sigma_{\text{БУЗ}}^2 = 0.000298$ LR= 0.811935	BC-100-LOG, M=150, $A_0 = 1.433, \beta = 0.3440, \sigma_{\text{МУЗ}}^2 = 0.000681$ $B_0 = 0.5112, \rho = 0.9946, \sigma_{\text{БУЗ}}^2 = 0.000223$ LR= 3.056881
—————	LBL-1, M=180, $A_0 = 4.4887, \beta = 0.916, \sigma_{\text{МУЗ}}^2 = 0.00047$ $B_0 = 0.2405, \rho = 0.9883, \sigma_{\text{БУЗ}}^2 = 0.000247$ LR= 1.906777	LBL-2, M=91, $A_0 = 3.16996, \beta = 0.92473, \sigma_{\text{МУЗ}}^2 = 0.00063$ $B_0 = 0.30474, \rho = 0.9769, \sigma_{\text{БУЗ}}^2 = 0.000298$ LR= 2.117156
—————	LBL-1-LOG, M=250, $A_0 = 4.3239, \beta = 0.7679, \sigma_{\text{МУЗ}}^2 = 0.00056$ $B_0 = 0.9928, \rho = 0.9934, \sigma_{\text{БУЗ}}^2 = 0.000224$ LR= 2.503366	LBL-2-LOG, M=100, $A_0 = 2.24027, \beta = 0.69325, \sigma_{\text{МУЗ}}^2 = 0.00058$ $B_0 = 0.36159, \rho = 0.98428, \sigma_{\text{БУЗ}}^2 = 0.000264$ LR= 2.208149
TCP-0.1, M=300, $A_0 = 2.259, \beta = 0.523, \sigma_{\text{МУЗ}}^2 = 0.00012$ $B_0 = 0.3228, \rho = 0.9960, \sigma_{\text{БУЗ}}^2 = 0.00015$ LR= 0.80758	TCP-1, M=200, $A_0 = 0.2920, \beta = 0.2786, \sigma_{\text{МУЗ}}^2 = 0.00093$ $B_0 = 0.09944, \rho = 0.9982, \sigma_{\text{БУЗ}}^2 = 0.00120$ LR= 0.77632	TCP-2, M=89, $A_0 = 0.55824, \beta = 0.47581, \sigma_{\text{МУЗ}}^2 = 0.00112$ $B_0 = 0.14666, \rho = 0.9908, \sigma_{\text{БУЗ}}^2 = 0.001906$ LR= 0.585866
TCP-0.1-LOG, M=300, $A_0 = 0.7557, \beta = 0.314, \sigma_{\text{МУЗ}}^2 = 0.00007$ $B_0 = 0.2342, \rho = 0.9976, \sigma_{\text{БУЗ}}^2 = 0.00008$ LR= 0.89552	TCP-1-LOG, M=200, $A_0 = 1.3356, \beta = 0.5201, \sigma_{\text{МУЗ}}^2 = 0.00020$ $B_0 = 0.22649, \rho = 0.9945, \sigma_{\text{БУЗ}}^2 = 0.00055$ LR= 0.36068	TCP-2-LOG, M=100, $A_0 = 1.15918, \beta = 0.55224, \sigma_{\text{МУЗ}}^2 = 0.00027$ $B_0 = 0.256227, \rho = 0.9885, \sigma_{\text{БУЗ}}^2 = 0.000731$ LR= 0.36993

Таким образом, в результате приведенного анализа логично сделать вывод о присутствии МУЗ практически во всех исследуемых реализациях трафика.

2.3.3 Энергетические спектры

В настоящем разделе производится расчет энергетических спектров исследуемых временных реализаций. Известно, что в частотной области МУЗ отражается на характерном степенном законе поведения спектральной плотности рассматриваемого процесса.

Рассчитанные энергетические спектры реализаций представлены на рис.2.22.

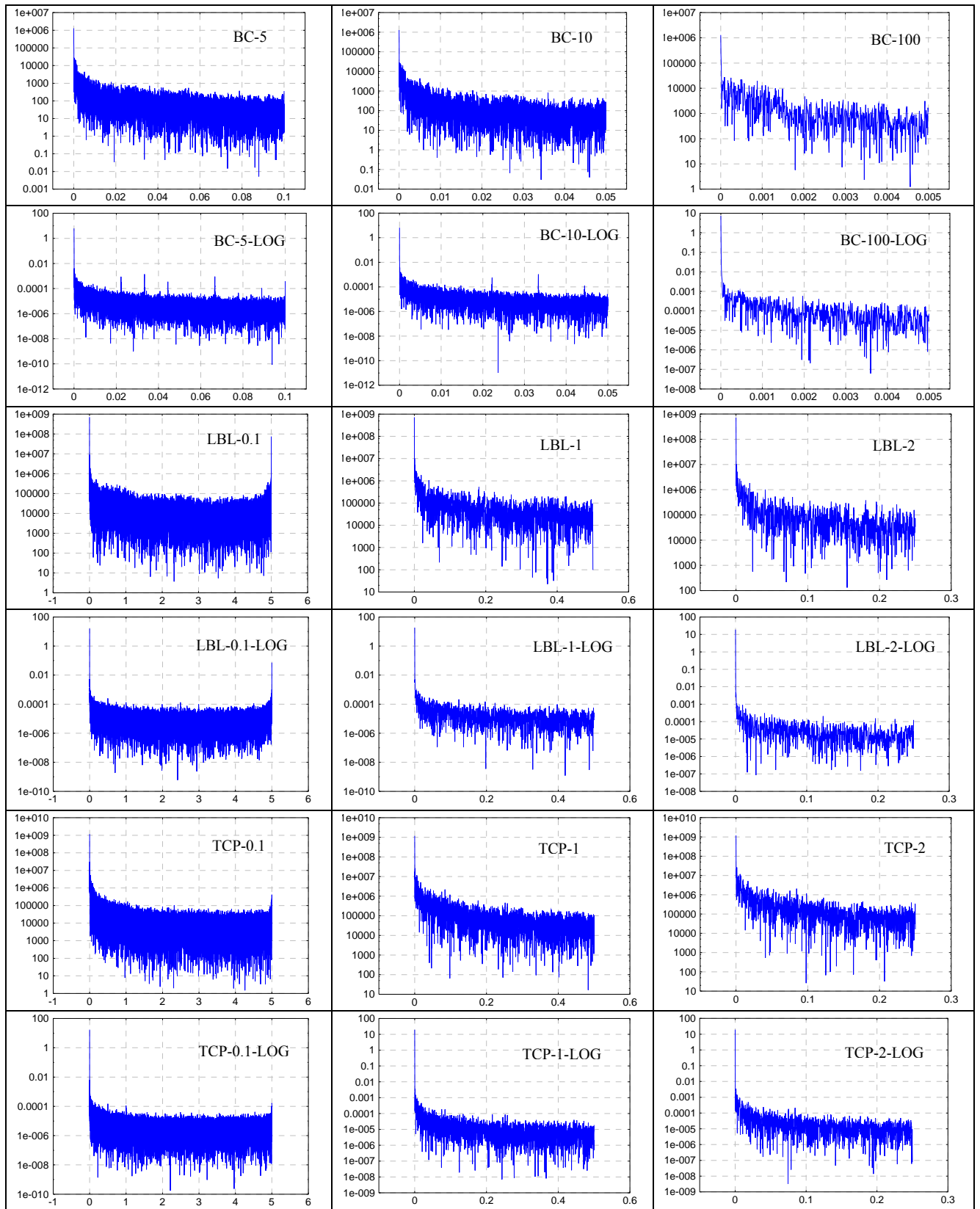


Рис. 2.22 . Энергетические спектры реализаций трафика. Здесь по оси абсцисс отложена частота λ (Гц), по оси ординат – спектр $W(\lambda)$ в логарифмическом масштабе

Анализируя полученные спектры можно заметить наличие сильной гармонической компоненты ~ 5 Гц в реализациях LBL-0.1, LBL-0.1-LOG. Данная гармоника также присутствует (хотя и менее существенна) в реализациях TCP-0.1 и TCP-0.1-LOG. При увеличении уровня агрегирования гармоника исчезает. Очевидно, периодичность АКФ реализаций LBL-0.1, LBL-0.1-LOG обусловлена именно присутствием в их спектрах гармоники 5 Гц. Также в реализациях BC-5-LOG и BC-10-LOG наблюдается несколько сильных гармонических компонент. Отметим, что в [41] и [42], при исследовании реализаций трафика получены схожие результаты и выявлены гармоники с частотами ~ 11.9 Гц и 1.25 Гц, присутствующие в wan-трафике. Тем не менее, пока непонятны причины появления подобных компонент в спектрах трафика.

Обнаруженное явление имеет принципиальное значение, поскольку выявляет присутствие регулярной детерминированной составляющей в агрегированном сетевом трафике, что может быть интересным при решении задач прогнозирования телетрафика.

Одновременно, можно отметить преобладание в полученных спектрах низкочастотных компонент, что согласуется с понятием фликкер-шума, о котором упоминалось ранее.

2.4 Исследование показателя Хэрста реализаций

Произведем оценку показателя Хэрста H реализаций. Напомним, что при $0.5 < H < 1$ говорят о **персистентном** (поддерживаемом) поведении процесса, либо о том, что процесс обладает длительной памятью. В случае $0 < H < 0.5$ говорят о **антиперсистентности** процесса. При $H = 0.5$ отклонения процесса от среднего являются действительно случайными и не зависят от предыдущих значений, что соответствует случаю БД.

Результаты наших измерений показателя Хэрста 7-ю методами приведены в табл. 2.3. Здесь через дробь приведены непосредственно сам

показатель H и соответствующая ему точность измерения. Точность определяется качеством аппроксимации точек вычисляемых статистик отрезком прямой либо вероятностью нахождения параметра H в заданном доверительном интервале в процентах (для методов Abru-Veitch Estimator и Whittle Estimator). Если точность невелика (заштрихованные области), то данную оценку H нельзя принимать во внимание.

Табл. 2.3. Оценка показателя H реализаций сетевого трафика

№	Реализация/ Метод	Коэффициент Хэрста						
		Aggregate Variance	R/S	Periodo- gram	Absolute Moments	Variance of Residuals	Abru- Veitch Estimator	Whittle Estimator
1	BC-5	0.842 / 98.93	0.770 / 99.36	0.955	0.534 / 55.75	1.128 / 99.94	1.007 / 95	0.949 / 95
2	BC-5-LOG	0.945 / 95.86	0.639 / 94.54	0.945	0.943 / 98.98	1.056 / 99.47	0.959 / 95	0.889 / 95
3	BC-10	0.837 / 98.95	0.745 / 99.08	0.920	0.479 / 56.61	1.111 / 99.58	1.037 / 95	0.951 / 95
4	BC-10-LOG	0.943 / 95.61	0.583 / 92.41	0.928	0.945 / 95.17	1.066 / 99.23	0.979 / 95	0.920 / 95
5	BC-100	0.695 / 73.95	0.720 / 97.36	0.653	0.116 / 65.67	1.114 / 99.65	0.986 / 95	0.904 / 95
6	BC-100-LOG	0.771 / 63.99	0.436 / 80.52	0.759	0.793 / 54.78	1.059 / 99.53	0.900 / 95	0.873 / 95
7	BC-5-surr-001	0.845 / 99.23	0.796 / 99.73	1.016	0.504 / 60.53	0.923 / 99.78	1.006 / 95	0.946 / 95
8	BC-10-LOG-surr-001	0.961 / 83.43	0.667 / 96.21	0.914	0.963 / 96.40	1.062 / 99.00	0.972 / 95	0.916 / 95
9	LBL-0.1	0.591 / 80.01	0.805 / 99.79	0.838	0.214 / 69.89	0.849 / 99.89	0.755 / 95	0.715 / 95
10	LBL-0.1-LOG	0.741 / 91.09	0.640 / 95.59	0.823	0.733 / 90.50	0.836 / 99.90	0.700 / 95	0.65 / 95
11	LBL-1	0.677 / 98.79	0.598 / 98.31	0.762	0.018 / 66.87	0.703 / 99.46	0.881 / 95	0.823 / 95
12	LBL-1-LOG	0.637 / 98.64	0.153 / 70.05	0.699	0.709 / 87.77	0.698 / 99.40	0.921 / 95	0.785 / 95
13	LBL-2	0.661 / 98.91	0.531 / 97.31	0.823	0.117 / 67.53	0.669 / 99.17	0.858 / 95	0.788 / 95
14	LBL-2-LOG	0.624 / 99.54	0.051 / 39.48	0.623	0.719 / 86.81	0.649 / 98.51	0.888 / 95	0.684 / 95
15	LBL-0.1-surr-001	0.746 / 90.50	0.803 / 99.69	0.787	0.355 / 63.2	0.837 / 99.84	0.773 / 95	0.709 / 95
16	LBL-0.1-LOG-surr-001	0.743 / 88.27	0.666 / 96.87	0.795	0.734 / 87.67	0.816 / 99.89	0.682 / 95	0.647 / 95
17	TCP-0.1	0.832 / 97.07	0.820 / 99.55	0.978	0.450 / 55.12	1.020 / 99.83	0.956 / 95	0.872 / 95
18	TCP-0.1-LOG	0.873 / 98.83	0.617 / 94.42	0.880	0.842 / 98.50	0.940 / 99.80	0.801 / 95	0.742 / 95
19	TCP-1	0.738 / 91.77	0.762 / 98.85	0.976	0.147 / 62.17	1.014 / 99.94	1.046 / 95	0.981 / 95
20	TCP-1-LOG	0.734 / 87.60	0.422 / 85.87	0.947	0.765 / 78.15	0.933 / 99.87	0.952 / 95	0.833 / 95
21	TCP-2	0.715 / 92.09	0.721 / 98.25	0.941	0.018 / 63.44	1.015 / 99.95	1.052 / 95	0.974 / 95
22	TCP-2-LOG	0.703 / 88.30	0.323 / 79.59	0.955	0.743 / 76.92	0.938 / 99.89	1.007 / 95	0.890 / 95
23	TCP-0.1-surr-001	0.845 / 97.93	0.809 / 99.72	0.977	0.455 / 55.86	0.940 / 99.97	0.946 / 95	0.860 / 95
24	TCP-0.1-LOG-surr-001	0.816 / 94.02	0.671 / 96.30	0.870	0.816 / 92.02	0.917 / 99.75	0.800 / 95	0.738 / 95
25	N	0.483 / 99.45	0.560 / 99.87	0.541	0.134 / 80.28	0.483 / 99.63	0.546 / 95	0.5 / 95
26	L	0.629 / 94.36	0.529 / 94.50	1.581	0.496 / 94.46	0.893 / 94.56	2.673 / 95	0.999 / 95
27	R	0.024 / 99.01	0.149 / 79.91	0.660	0.081 / 99.36	0.102 / 60.73	1.058 / 95	0.999 / 95

Строго говоря, оценка показателя Херста для хаотической реализации системы Лоренца, возможно, не имеет смысла, поскольку последняя не обладает соответствующим видом АКФ: $S(k) = k^{2H-2}$, где $0 < H < 1$. Тем не менее, представляет определенный практически интерес изучение взаимоотношений свойств персистентности и хаотичности. Например, в [40] изучается эффект аномального скэйлинга на графиках корреляционного интеграла при анализе фликкер-шума. Иными словами, показывается, что при определенных условиях фликкер-шум может быть неверно идентифицирован как хаотическая реализация.

Изучая результаты измерений показателя H , можно отметить, что для реализаций сетевого трафика $H > 0.5$, т. е. трафик относится к классу персистентных процессов. В то же время для белого шума практически все методы показали $H \sim 0.5$, что подтверждает правильность измерений. Следует подчеркнуть, что такое разнообразие методов оценки было выбрано неслучайно и преследовало цель получить более достоверные результаты. При этом можно заметить, например, что метод абсолютных моментов и (в меньшей степени) метод RS-статистики часто дают некорректные результаты.

2.5 Исследование реализаций методами теории нелинейной динамики

Нелинейная динамика – наука, изучающая структуры и свойства эволюционных процессов в нелинейных динамических системах [32]. Особенностью, присущей исключительно нелинейным системам, является возможность реализации в них множества различных режимов функционирования (детерминированного хаоса в том числе), которые зависят от начального состояния, параметров системы и внешних воздействий. Теория нелинейной динамики становится в последнее время все более востребованной при изучении временных рядов и предоставляет обширные методы для их исследования. Известно, что хаотические системы

обладают следующими основными свойствами: нелинейностью, детерминированностью, чувствительностью к начальным условиям. Кроме того, хаотическая реализация выглядит как некоторый стохастический процесс, и аттрактор хаотической системы часто является фрактальным. Если удастся обнаружить признаки детерминированного хаоса в трафике, мы получим новую модель трафика и новый алгоритм по его предсказанию благодаря детерминированной природе хаоса. Как уже отмечалось выше, в работе [16] с помощью имитационного моделирования на симуляторе ns-2 показано, что модель трафика протокола TCP может быть как простым периодическим процессом, так и при некоторых условиях обладать более сложным поведением, согласующимся с понятием *детерминированного хаоса*. Можно также отметить существование хаотических моделей сетевого трафика [43]. В них используются свойства хаотических отображений (chaotic maps) типа

$$\begin{aligned} x_{n+1} &= f_1(x_n), y_n = 0, (0 < x_n < d) \\ x_{n+1} &= f_2(x_n), y_n = 1, (d < x_n < 1) \end{aligned} \quad (2.6)$$

где $f_1(\cdot)$ и $f_2(\cdot)$ – некоторые функции, для которых выполняется условие чувствительности к начальным условиям. Кроме того, при построении модели предполагается, что источник трафика находится в активном или пассивном состоянии в зависимости от того, большее или меньшее некоторого предела d значение принимает величина x_n . Адекватность данной модели подтверждается присущими ей свойствами медленно убывающей зависимости и распределением с тяжелым хвостом. Для одного из наборов параметров хаотического отображения корреляционная размерность аттрактора оценена как 0.91.

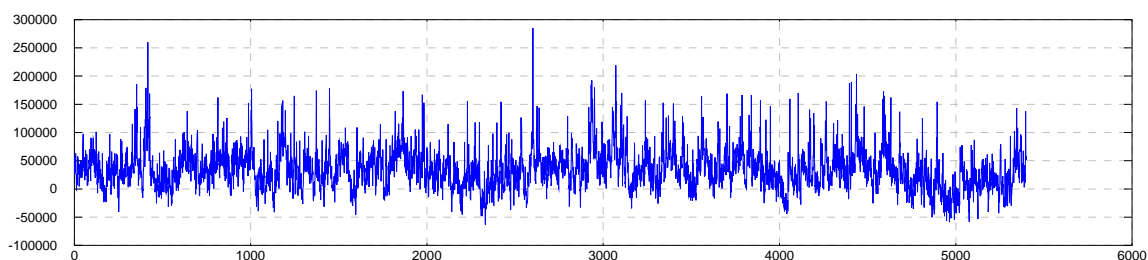
Изучая данные материалы можно предположить, что исследование сетевого трафика методами нелинейной динамики способно оказать пользу при изучении его характеристик, а также разработке и улучшению алгоритмов его обработки.

2.5.1 Концепция суррогатных данных

Одной из основных задач настоящего анализа является проверка гипотезы о том, является ли сетевой трафик линейно коррелированным шумом или детерминированным процессом. Для этой цели воспользуемся концепцией, так называемых “суррогатных данных”. Суть данного метода заключается в следующем: для исследуемой временной реализации процесса создается ансамбль (обычно пять или десять) “суррогатных” реализаций, являющихся случайными по своей природе, но обладающих точно такими же автокорреляционной функцией и дисперсией как у исходной реализации. На практике это достигается Фурье-преобразованием исходной реализации, изменением случайным образом фаз и обратным Фурье-преобразованием. Далее сравниваются результаты вычисления различных статистических характеристик (размерностей, корреляционных интегралов и пр.) для оригинального процесса и его ансамбля суррогатных реализаций. В случае если результаты значительно различаются, нулевая гипотеза о том, что исходная реализация является коррелированным шумом, может быть отклонена.

Для каждой из реализаций трафика составим по 5 суррогатных реализаций и будем использовать их в ходе дальнейшего анализа. Для примера на рис. 2.23 приведена одна из суррогатных реализаций для реализации ТСП-1, а также ее гистограмма и АКФ.

Можно заметить, что суррогатная реализация имеет такую же АКФ, что и у исходной, но отличное распределение, которое теперь более похоже на нормальное.



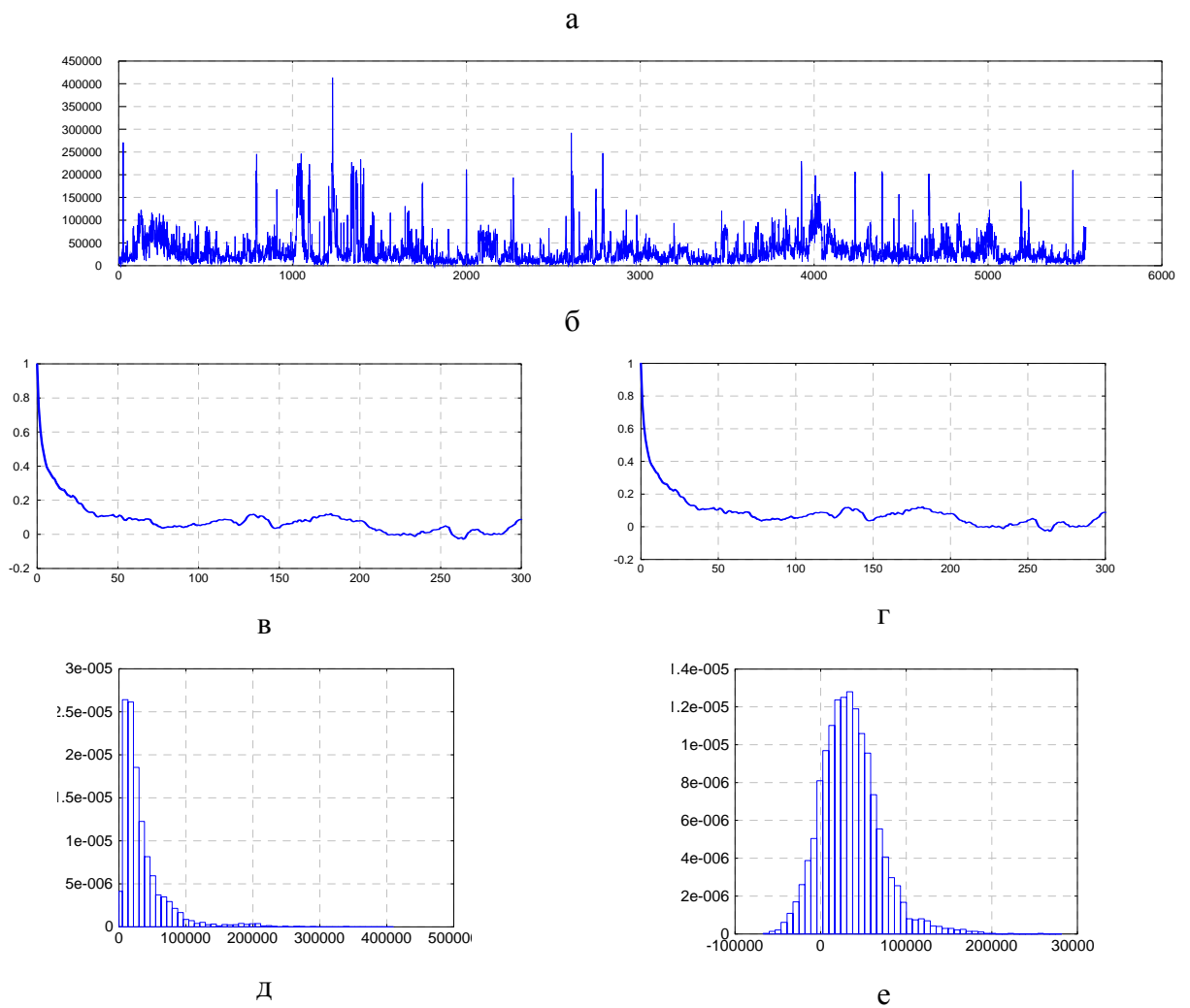


Рис. 2.23. а) Суррогатная реализация TCP-1-SURR-001;
 б) Исходная реализация TCP-1;
 в) АКФ исходной реализации TCP-1;
 г) АКФ суррогатной реализации TCP-1-SURR-001;
 д) Гистограмма исходной реализации TCP-1;
 е) Гистограмма суррогатной реализации TCP-1-SURR-001.

Произведенная ранее (см. табл. 2.3) оценка показателя Херста показывает, что суррогатные реализации (обозначены суффиксами “-surr-”) обладают примерно таким же уровнем персистентности, что и оригинальные реализации.

Далее мы будем использовать полученные суррогатные реализации при изучении характеристик реализации трафика.

2.5.2 Идея реконструкции аттрактора

Анализ сетевого трафика фактически сводится к задаче обработки временного ряда. Как уже отмечалось, теория нелинейной динамики, в свою очередь, предоставляет широкие возможности для изучения, идентификации и прогнозирования временных рядов, обладающих некоторыми специфическими свойствами. Одной из ключевых концепций теории нелинейной динамики является использование теоремы Такенса о погружении аттрактора в пространствах различных размерностей. Под аттрактором здесь понимается предельное множество траекторий в фазовом пространстве системы, к которому стремятся все траектории из некоторой окрестности этого множества.

Более того, существует методика, позволяющая восстановить параметры динамической системы по единственной реализации (временному ряду) процесса с помощью изучения траектории системы в m -мерном фазовом пространстве, координатами которого являются компоненты следующего вектора: $z_i^m = \{X_i, X_{i+\tau}, \dots, X_{i+(m+1)\tau}\}$, где τ – временной сдвиг.

Данная операция называется “*погружением аттрактора*” в пространство размерности m . Результатом успешного погружения является выявление определенных закономерностей в поведении траектории системы в пространстве данной размерности.

2.5.3 Ложные ближайшие соседи (FNN)

Метод False Nearest Neighbors (FNN) служит для определения минимальной приемлемой размерности пространства погружения. Его принцип понятен на интуитивном уровне. Допустим $z_i^{(m)}$ и $z_j^{(m)}$ - два близких соседа в реконструкции размерности m , а $z_i^{(m+1)}$ и $z_j^{(m+1)}$ соответствуют им в реконструкции $m+1$. Если мы имеем дело с

действительно близкими соседями, то они, как правило, близки в обеих реконструкциях. В то же время, если соседи, близкие в реконструкции m ($\|z_i^{(m)} - z_j^{(m)}\|$ - мало), становятся отдаленными в реконструкции $m+1$ ($\|z_i^{(m+1)} - z_j^{(m+1)}\|$ - велико), то они называются ложными близкими соседями (false nearest neighbors).

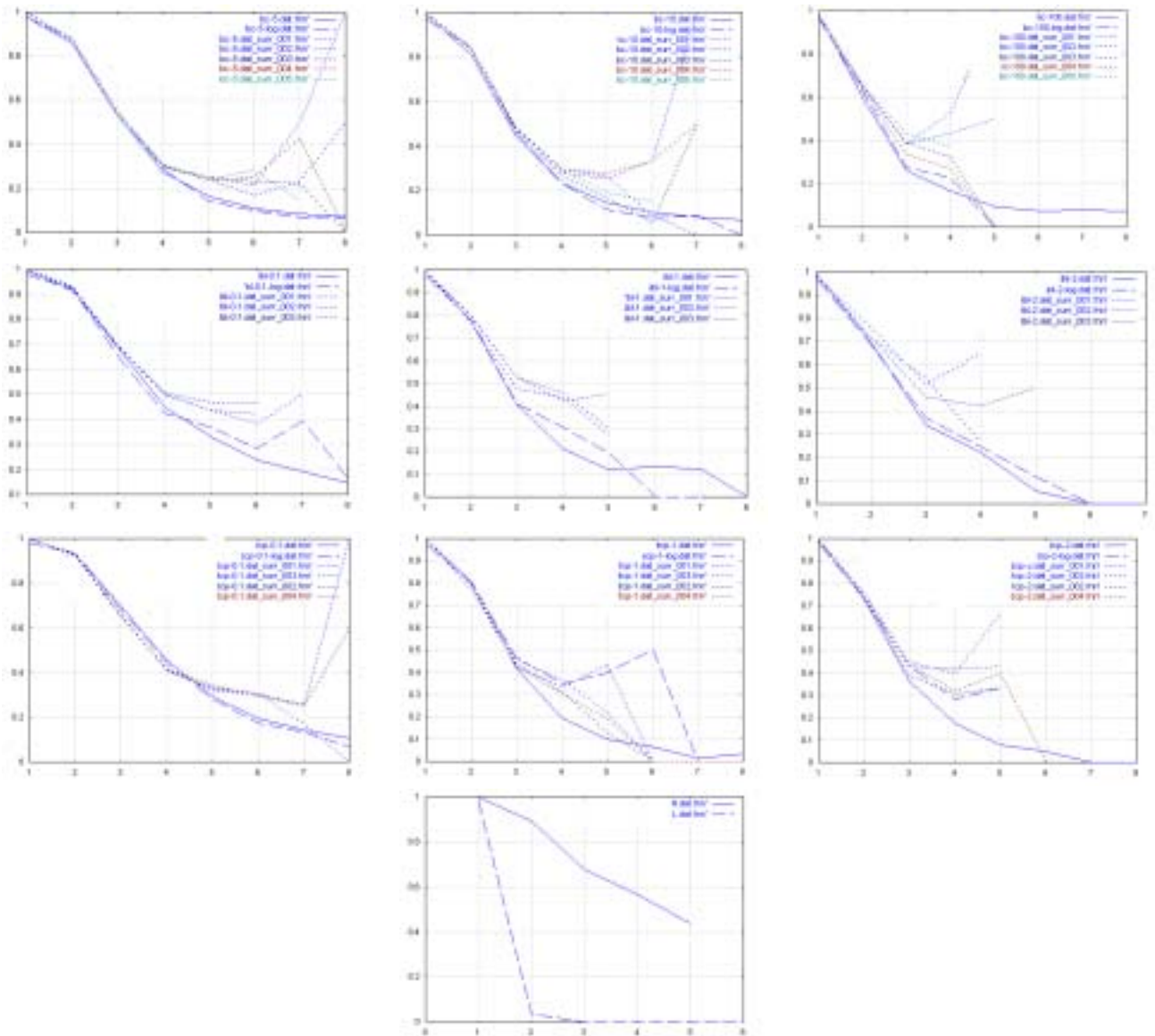


Рис. 2.24. Зависимости FNN (для реализаций трафика временной сдвиг $d=5$, для белого шума $d=1$)

Если теперь увеличивать m и оценивать количество FNN, то при достижении нужной размерности, при которой достигается правильная реконструкция, это количество резко уменьшается. Очевидно, что по

графику зависимости количества “ложных соседей” от размерности пространства погружения (по его минимуму или убыванию в нуль) можно сделать вывод о минимально возможной размерности фазового пространства. Построим графики FNN для исследуемых реализаций, суррогатных реализаций, а также для белого шума и детерминированного хаоса (в виде системы Лоренца).

Изучая зависимости FNN можно сделать вывод о том, что наиболее приемлемой размерностью реконструкции для реализаций ВС-5, LBL-0.1, TCP-0.1 является размерность ~ 8 , а для реализаций ВС-10, ВС-100, LBL-1, LBL-2, TCP-1 и TCP-2 размерность ~ 5 . При этом характерно, что зависимости количества ложных соседей для суррогатных реализаций заметно отличаются от аналогичных для исходных реализаций и больше похожи на зависимость FNN для белого шума (см. последний график). Данное замечание говорит в пользу того, что возможно свойства изучаемых реализаций трафика обусловлены не только корреляциями, свойственными $1/f^a$ – процессам.

В то же время графики FNN для оригинальных агрегированных реализаций трафика занимают некоторое промежуточное положение (по степени убывания) между чисто хаотическим процессом и число случайным процессом (белым шумом). Данное наблюдение может служить в пользу классификации сетевого трафика как некоторого детерминированного (возможно хаотического) процесса.

2.5.4 Вычисление корреляционного интеграла

Одним из основных инструментов, используемых для изучения фазового портрета системы служит понятие корреляционного интеграла, который позволяет оценить размерность D_2 аттрактора, погруженного в пространство размерности m . Вычисление размерности аттрактора по

временному ряду, в свою очередь, представляет интерес, поскольку позволяет оценить минимальное число существенных динамических переменных, необходимых для описания наблюдаемого процесса. Наиболее популярный алгоритм для расчета корреляционного интеграла был предложен Grassberger P., Procaccia I. [71] и основывается на соотношении (2.7) [39]

$$C(\varepsilon, m) = \frac{\text{число пар с } \|z_i - z_j\| < \varepsilon}{\text{общее число пар } z_i, z_j}. \quad (2.7)$$

Здесь z_i, z_j – векторы координат точек в фазовом пространстве размерности m . При этом справедливо соотношение

$$C(m, \varepsilon) \sim \varepsilon^{-D_2}, \quad (2.8)$$

позволяющее оценить размерность D_2 аттрактора по наклону наиболее линейного участка графика

$$\log C(\varepsilon, m) \cong -D_2 \log \varepsilon + \text{const}. \quad (2.9)$$

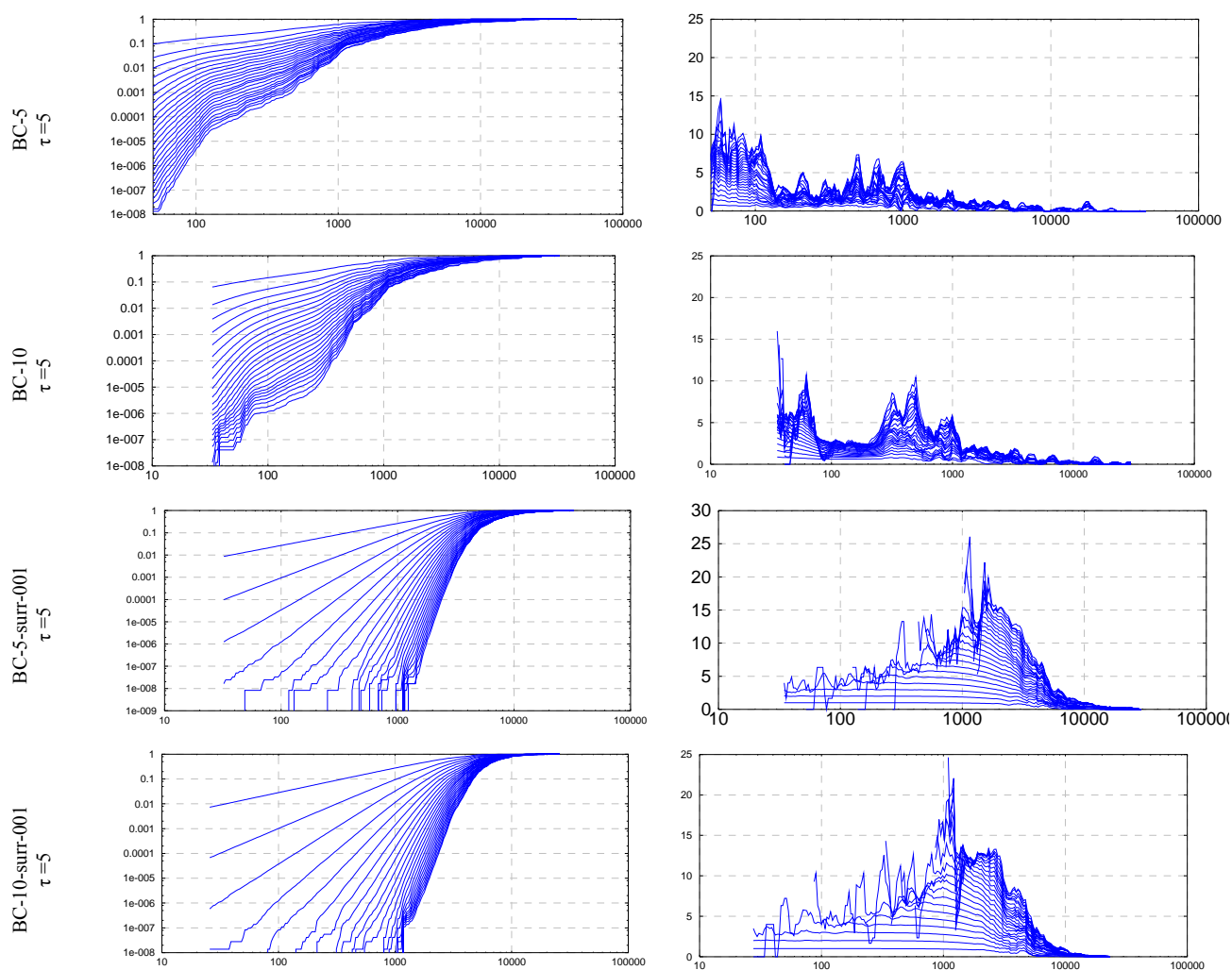
При увеличении m оценка D_2 должна стремиться к истинному значению корреляционной размерности аттрактора.

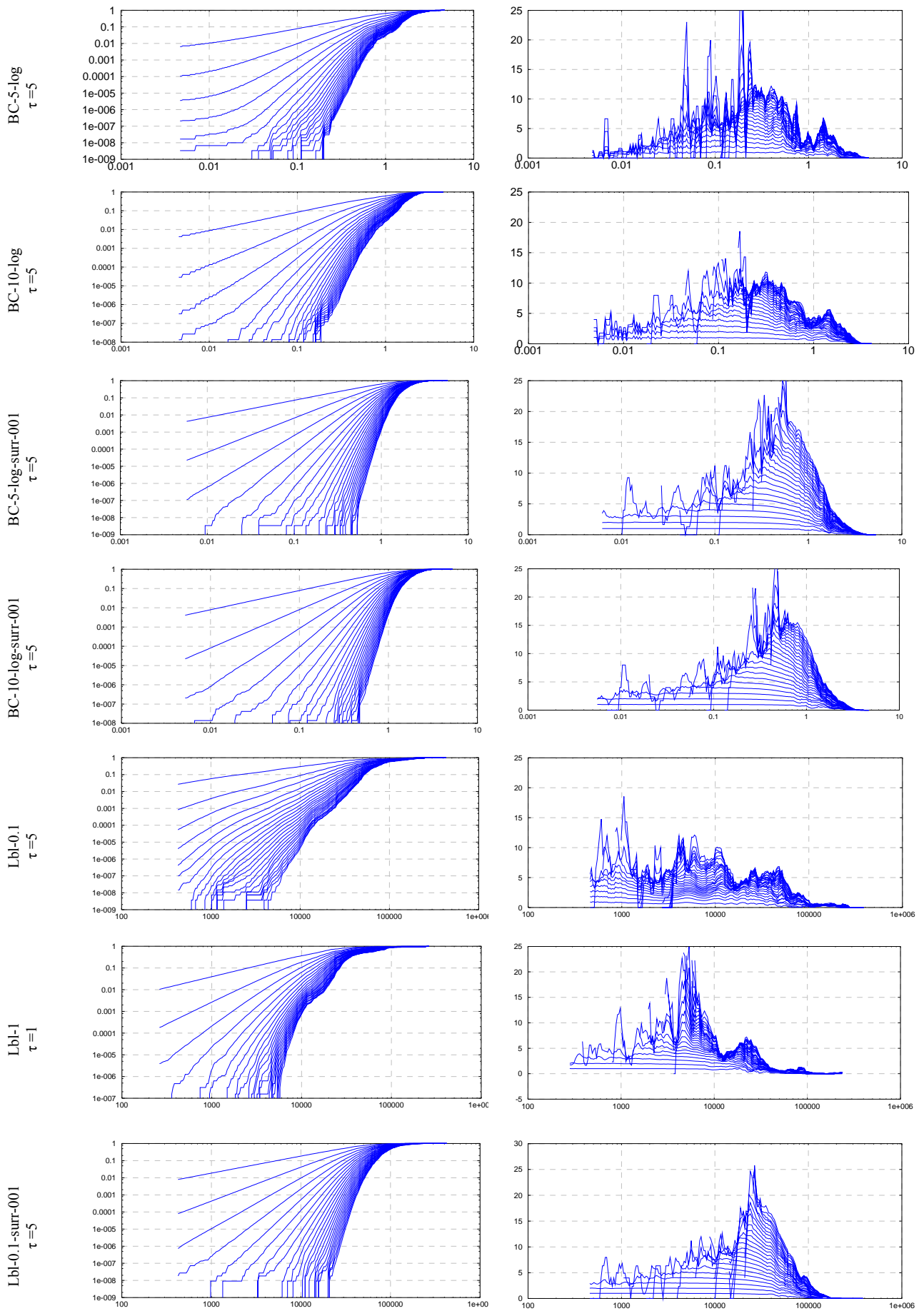
Если мы имеем дело с данными, которые по определению не должны давать достоверных результатов при такой обработке, то корреляционный интеграл вообще может не иметь линейного участка (хотя может и иметь), а его наклон обычно возрастает с увеличением m . В ряде случаев можно показать, что этот наклон будет линейным, т.е.

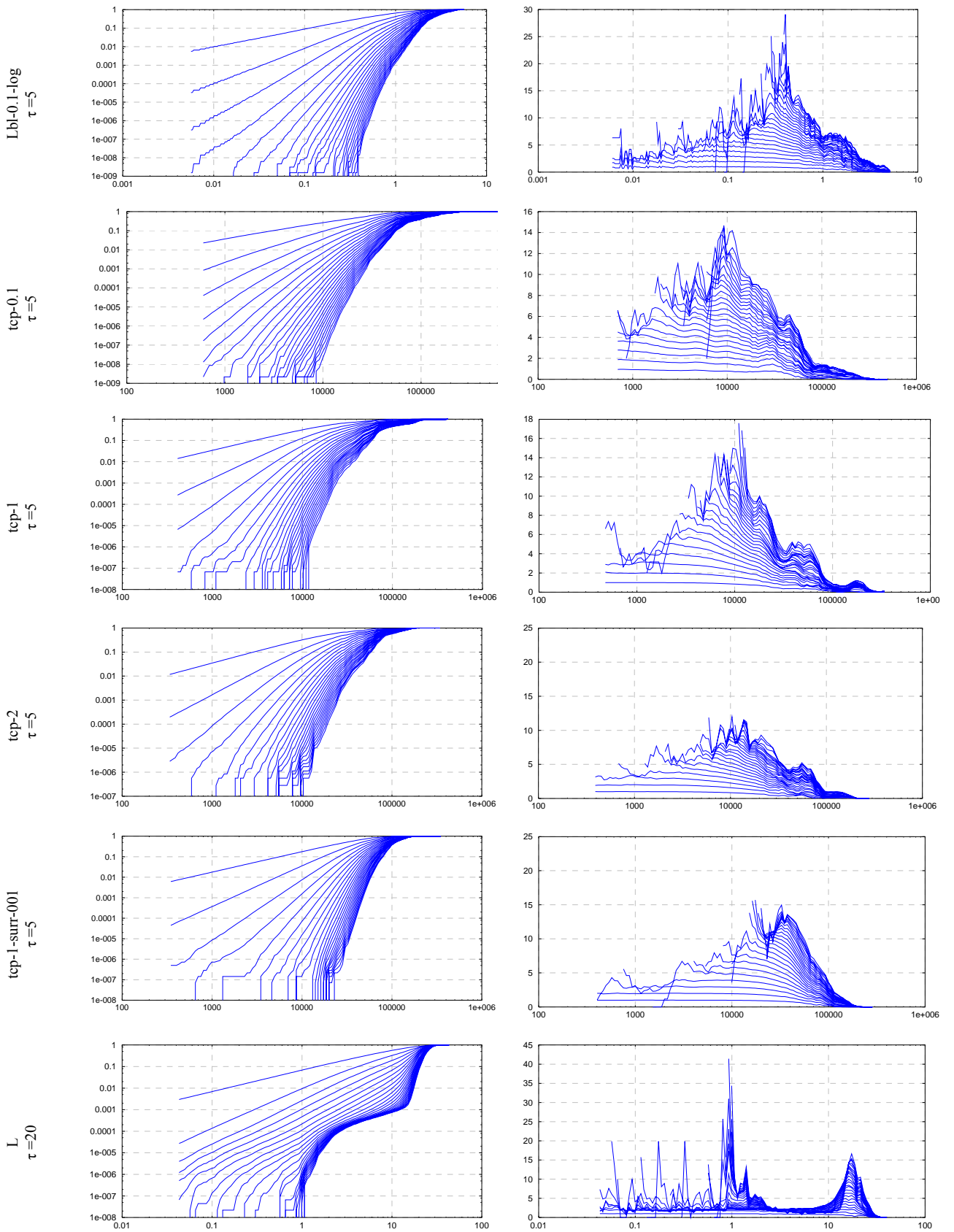
$$S(\varepsilon, m) \cong \varphi_0(\varepsilon)m + S_0(\varepsilon). \quad (2.10)$$

В других случаях, особенно при обработке данных ограниченной длины и небольших значениях m , зависимость чаще всего более сложная, однако в большинстве случаев она удовлетворительно аппроксимируется линейной зависимостью. Исследование зависимости наклона от m может помочь разрешению вопроса о том, позволяет ли реконструкция судить о свойствах аттрактора или нет (по самым разным причинам: недостаточно

данных, слишком сложная динамика, данные не детерминированные, а случайные и пр.). Специальное исследование [44] показало, что $\phi_0 = 0.05 \dots 0.1$ является некоторым пороговым значением. При меньших значениях методы нелинейной динамики обычно применимы, при больших – нет. Таким образом, с помощью корреляционного интеграла можно решить вопрос об эффективности алгоритмов нелинейной динамики при обработке данного временного ряда или же использовать статистические подходы или как-либо комбинировать динамические и статистические методы. Ниже приведем результаты расчета корреляционного интеграла для исследуемых временных рядов.







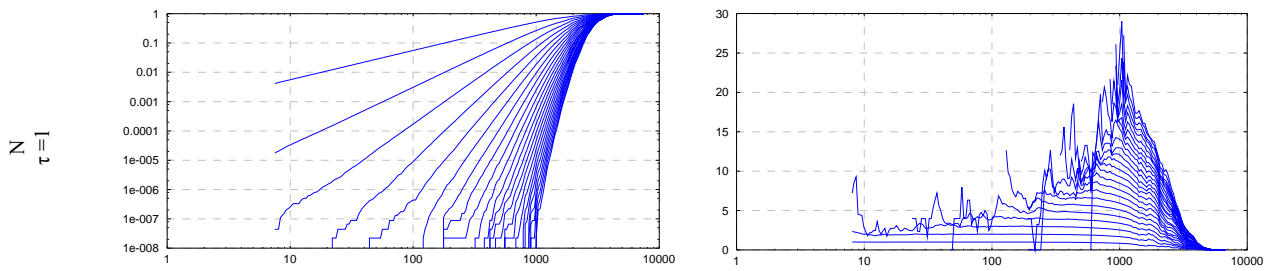


Рис.2.25. Слева: графики семейства зависимостей корреляционного интеграла. По оси ординат отложена величина $C(\epsilon)$, по оси абсцисс - ϵ . Параметром семейства является размерность вложения m . Сверху – вниз m увеличивается от 1 до 25. Справа – семейство зависимостей локальных наклонов D_2 корреляционного интеграла от ϵ . Параметром семейства является m (снизу вверх m увеличивается от 1 до 25)

Основные наблюдения, которые можно сделать, изучая зависимости корреляционного интеграла, состоят в следующем:

- для всех оригинальных временных реализаций трафика наблюдается характерный изгиб кривых корреляционного интеграла. Наиболее выражен изгиб у реализаций ВС-5 (на интервале 200-800 Байт/с), ВС-10 (на интервале 100-300 Байт/с), а также у хаотической реализации системы Лоренца;
- кривые корреляционных интегралов для суррогатных реализаций больше напоминают кривые корреляционного интеграла для белого шума, чем аналогичные для хаотической реализации. Это наблюдение отвергает гипотезу о классификации сетевого трафика как некоторого шумового процесса, обладающего коррелированной структурой. Другими словами, в трафике, возможно, присутствует некоторая детерминированная, однако достаточно слабая составляющая (не исключено, что с некоторой степенью хаотичности).
- если наблюдаемое насыщение наклона кривых корреляционного интеграла для реализации ВС-10 неслучайно, оно может свидетельствовать о размерности соответствующего аттрактора ~ 2.5 .

- для хаотической реализации системы Лоренца наблюдается характерное насыщение наклонов корреляционного интеграла ($D_2 \sim 2.5$), в то время как для случая белого шума насыщения вообще не происходит, что хорошо согласуется с имеющейся теоретической базой.

2.5.5 Проверка гипотезы о статистической независимости (BDS-тест)

Для проверки гипотезы о статистической независимости последовательных членов временного ряда, соответствующего сетевому трафику, воспользуемся так называемым BDS – тестом [45]. Данный тест основан на свойствах корреляционного интеграла и требует вычисления следующей статистической характеристики:

$$w_{m,n}(\varepsilon) = \sqrt{n-m+1} \frac{C(\varepsilon, m, n) - C^m(\varepsilon, 1, n)}{\sigma_{m,n}(\varepsilon)} \quad (2.11)$$

где n – объем анализируемого участка временного ряда, $C(\varepsilon, m, n)$ – корреляционный интеграл. Считается, что нулевая гипотеза о независимости и одинаковости распределения членов ряда отвергается при уровне значимости 5%, если $|w_{m,n}(\varepsilon)| < 1.96$ для рекомендованного диапазона изменения ε . Для вычислений воспользуемся программной реализацией W. Davis Dechert. Поскольку программа позволяет работать только с числом отсчетов, не превышающих 7 500, проанализируем с ее помощью первые 7 500 отсчетов исследуемых реализаций. Обычно рекомендуется использовать значения $\varepsilon = 1.5 \cdot \left(\frac{\sigma}{s}\right) \dots 2 \cdot \left(\frac{\sigma}{s}\right)$, где σ – среднеквадратическое отклонение, s – максимальный размах значений ряда. Некоторые результаты расчетов программы даны в приложении. Анализируя полученные численные результаты можно сделать вывод о том, что для хаотической реализации L, реализаций трафика BC-5, BC-10, LBL-0.1, LBL-1, TCP-0.1, TCP-1, их логарифмов, а также суррогатных реализаций гипотеза о статистической

независимости членов ряда отвергается для всех рекомендованных значений ε и значений размерности погружения $m=2..10$. В то же время для белого шума нулевую гипотезу нельзя отвергнуть, поскольку $|w_{m,n}(\varepsilon)| < 1.96$ для всех изученных наборов параметров. Ряды ВС-100, LBL-2, ТСП-2 не анализировались, поскольку они имеют относительно небольшое количество отсчетов.

2.6 Эксперимент по сбору трафика в беспроводной сети

Основные широкоизвестные реализации сетевого трафика, полученные Vern Paxson, Will Leland и Dan Wilson доступны в Internet Traffic Archive [80]. Однако данные реализации, в основном датируемые началом 90-х годов, благодаря появлению новых технологий, протоколов, приложений к настоящему времени могли несколько устареть. Кроме того, все большее распространение получают беспроводные телекоммуникационные технологии, данные о степени самоподобия трафика в которых пока отсутствуют.

В настоящем разделе описывается оригинальный эксперимент¹ по снятию реализаций трафика в беспроводной сети одного из крупных интернет-провайдеров масштаба города, а также приводятся результаты анализа структурных особенностей данного трафика.

2.6.1 Постановка эксперимента по сбору трафика в беспроводной сети

Все реализации трафика были получены в процессе мониторинга магистрального канала в беспроводной сети одного из крупных Интернет-провайдеров масштаба города (ЗАО "Виплайн", Воронеж). Логическая схема

¹ Эксперимент по сбору и анализу трафика беспроводной сети проводился совместно с В.В. Платовым.

организации сети отображена на рис. 2.26. На коммуникационном узле провайдера установлена шести-секторная антенна, к каждому сектору которой подключена точка доступа стандарта IEEE 802.11b.

Всего ко всем шести точкам доступа провайдера подключено порядка 40 беспроводных клиентов. Беспроводной клиент представляет собой точку доступа, аналогичную провайдерской, к которой по локальной сети Fast Ethernet подключен один или несколько хостов. Все точки доступа работают как прозрачные мосты для протоколов верхних уровней. Маршрутизацию всех соединений, а также контроль и управление пропускной способностью (шейпинг) осуществляет маршрутизатор. С помощью него каждому беспроводному клиенту предоставлен канал с пропускной способностью от 64 до 512 Кбит/с. Механизмы раннего обнаружения перегрузок, такие как RED или WRED не используются.

Для сбора трафика использовался персональный компьютер под управлением операционной системы Windows 2000 Professional с установленным на нем анализатором протоколов (сниффер) Ethereal, который обеспечивает точность определения времени получения пакета 10^{-6} сек.

Сниффер был подключен к сети таким образом, чтобы фиксировать трафик, проходящий через точку “1” и, одновременно через точку “2”.

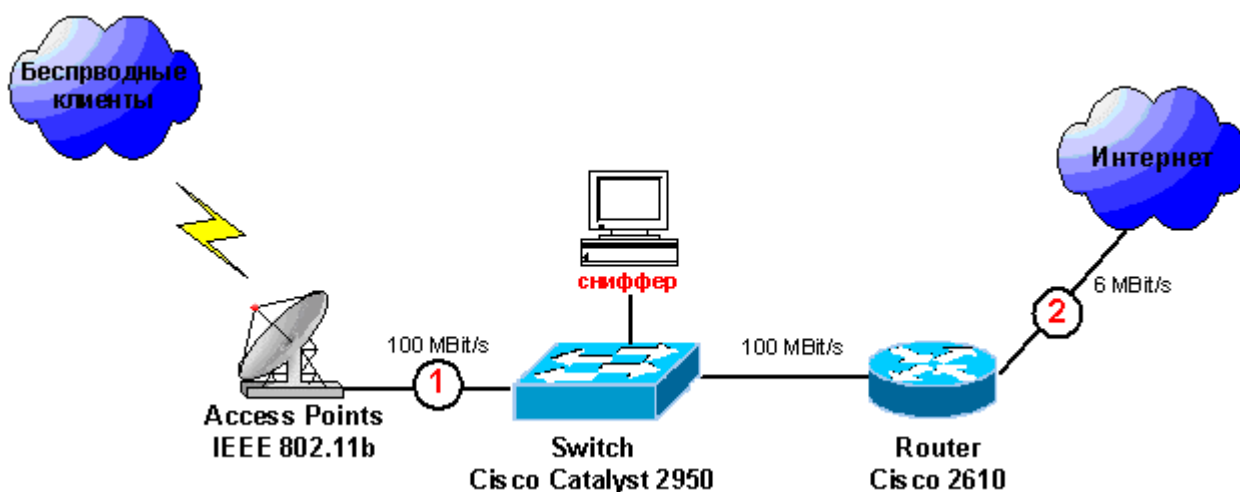


Рис. 2.26. Логическая схема организации эксперимента по снятию трафика

Обратим внимание, что через точку “1” проходит трафик обмена информацией между беспроводными клиентами, и вместе с ним, трафик обмена информацией между беспроводными клиентами и сетью Интернет. Через точку “2” проходит лишь последний из них.

Все зафиксированные пакеты записывались в файл формата tcpdump. Разделение общего трафика, собранного сниффером, на трафики, проходящие через точку “1”, точку “2”, входящие и исходящие производилось в ходе его дальнейшей обработки на основе адресной информации.

Всего в результате эксперимента было собрано более 12,7 миллионов пакетов. Из них 67% пришлось на протокол TCP.

2.6.2 Характеристика реализаций

Процедура записи трафика началась 18 марта 2004 года (четверг) в 10.00 и закончилась 18 марта в 17.00 по московскому времени. Общая длительность составила 7 часов. В результате эксперимента были получены следующие реализации трафика:

Weth04.dat – общий (входящий + исходящий) трафик уровня протокола Fast Ethernet, зафиксированный в точке “1”. Включает в себя трафик между клиентами, а также трафик обмена с сетью Internet. Содержит информацию о 12 712 076 Ethernet-кадрах.

Weth04.src – трафик уровня протокола Fast Ethernet, зафиксированный в точке “2”, с направлением от беспроводных клиентов в сеть Internet. Содержит информацию о 4 554 668 Ethernet-кадрах.

Weth04.dst – трафик уровня протокола Fast Ethernet, зафиксированный в точке “2”, с направлением из сети Internet к беспроводным клиентам. Содержит информацию о 5 586 556 Ethernet-кадрах.

Wtcp04.dat – общий (входящий + исходящий) трафик уровня протокола TCP, зафиксированный в точке “1”. Включает в себя трафик между клиентам,

а также трафик обмена с сетью Internet. Содержит информацию о 8 468 548 TCP-дейтаграммах.

Wtcp04.src – трафик уровня протокола TCP, зафиксированный в точке “2”, с направлением от беспроводных клиентов в сеть Internet.

Wtcp04.dst – трафик уровня протокола TCP, зафиксированный в точке “2”, с направлением из сети Internet к беспроводным клиентам.

Данные представлены двумя колонками чисел в ASCII-формате: первая колонка содержит временные метки (в сек.), а второй – общий размер Ethernet-кадра в байтах, либо размер поля данных ip-пакета в случае TCP-дейтаграммы. Заметим, что отсчеты не эквидистантны, а соответствуют моментам прихода пакетов. Для приведения реализаций трафика к эквидистантному виду по оси времени, была использована процедура агрегирования с уровнем агрегирования $dT = 0.05$ сек. Суть данной процедуры заключается в следующем: ось времени разбивается на равные интервалы dT , каждому из которых ставится в соответствие отношение количества прошедшей за данный интервал времени информации (в байтах) к длительности интервала (в секундах). При этом получаются соответствующие реализации Weth04_0.05.dat, Weth04_0.05.src, Weth04_0.05.dst, Wtcp04_0.05.dat, Wtcp04_0.05.src, Wtcp04_0.05.dst, однако смысл данных, представленных в последних столбцах реализаций, изменяется и теперь представляет собой интенсивность передачи информации (байт/сек).

2.6.3 Особенности технологии IEEE 802.11b

Стандарт RadioEthernet IEEE 802.11 определяет порядок организации беспроводных сетей на уровне управления доступом к среде (MAC-уровне) и физическом (PHY-уровне). Данная технология использует методы расширения спектра сигнала с помощью прямой последовательности (DSSS). Стандарт предусматривает работу на частоте 2.4 ГГц и обеспечивает скорость передачи данных до 11 Мбит/с.

Согласно спецификации IEEE 802.11b [70] в качестве алгоритма доступа к среде используется метод CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Протокол CSMA широко распространен и применяется, например, в локальных проводных сетях Ethernet (вариант CSMA/CD). Принцип его работы очень прост: станция, которая намерена начать передачу должна убедиться в том, что среда не занята. В противном случае передача данных будет отложена на более позднее время. Этот протокол хорошо работает в малозагруженных сетях. В сетях с довольно интенсивным обменом данными существенно возрастает вероятность возникновения коллизий. Коллизия – это одновременная передача данных несколькими станциями. В случае Ethernet-сетей возникновение коллизии определяют передающие станции (CD).

Однако, протокол CD (Collision Detection), хорошо работающий в проводных сетях, плохо подходит для сетей беспроводных по следующим причинам:

- использование механизма CD потребует использования полнодуплексных радиоустройств, способных одновременно передавать и принимать данные, что существенно увеличит цену оборудования;
- в случае беспроводной сети нельзя гарантировать того, что каждая станция будет иметь возможность взаимодействовать со всеми остальными станциями сети (что является главным условием механизма CD).

Для решения этих проблемы в стандарте IEEE 802.11b применен механизм предотвращения коллизий (Collision Avoidance) в сочетании с обязательными подтверждениями приема всех пакетов. Данный механизм работает следующим образом: станция перед передачей данных проверяет среду передачи, если она занята, то передача откладывается. Если среда свободна на требуемое время (называемое DIFS, Distributed Inter Frame Space), тогда

станция передает данные. Принимающая станция проверяет CRC у принятого пакета, и если ошибок не зафиксировано, то шлет пакет с подтверждением (АСК). Получение АСК говорит о том, что коллизии не произошло. Если передающая сторона не получит подтверждения, будет произведена повторная передача.

Для того, чтобы уменьшить вероятность коллизии двух станций, которые находятся вне зоны покрытия друг друга, стандартом определен механизм определения состояния виртуальной несущей (Virtual Carrier Sense): станция, желающая начать передачу данных, сначала передает короткий управляющий пакет, называемый RTS (Request To Send), в котором указывается источник, получатель и продолжительность основной передачи. Получатель (если среда передачи свободна) посылает ответный управляющий пакет, называемый CTS (Clear To Send), в котором содержится информация о длительности будущей передачи. Все станции, получившие либо RTS, либо CTS (либо и то, и другое), устанавливают свои Virtual Carrier Sense индикаторы (названные NAV, Network Allocation Vector) на указанное время.

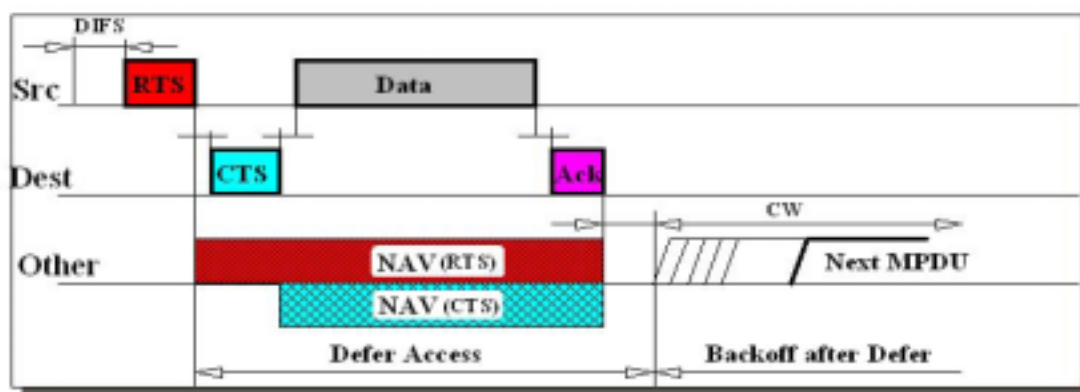


Рис. 2.27. Временная диаграмма функционирования алгоритма CSMA/CD

Станции будут использовать эту информацию вместе с физической проверкой среды передачи перед передачей данных. Временная диаграмма функционирования алгоритма CSMA/CD представлена на рис. 2.27.

2.6.4 Анализ результатов обработки трафика беспроводной сети

На рис.2.28 представлен график реализации трафика Weth04_0.05.dat, которая представляет собой агрегированный по уровню 0,05 сек временной ряд, соответствующий общему трафику протокола Fast Ethernet, зафиксированному в сети (в точке “1”).

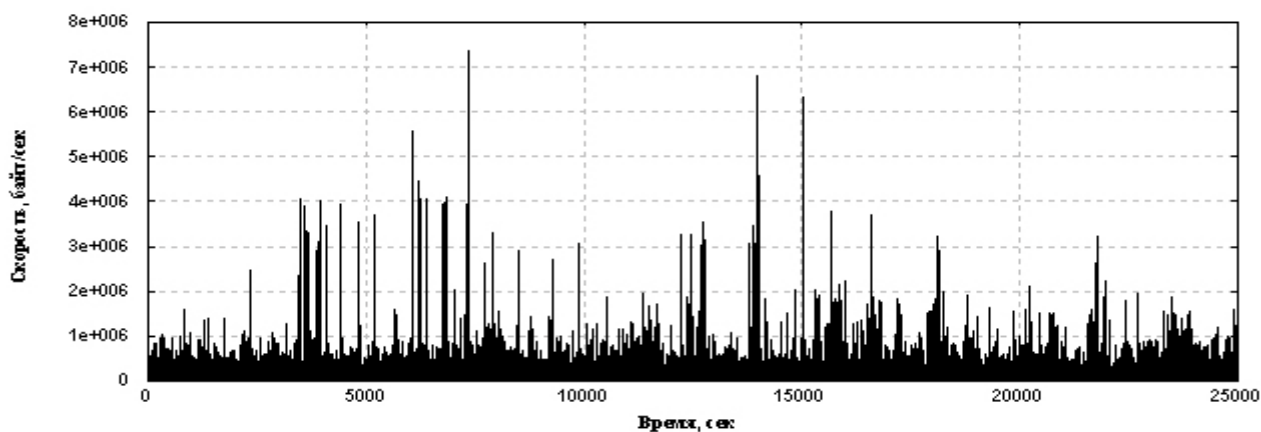


Рис. 2.28. Временной ряд Weth04_0.05.dat, соответствующий трафику (Ethernet) беспроводной сети

Напомним, что данная реализация включает в себя трафик обмена информацией беспроводных клиентов друг с другом, а также трафик обмена с сетью Internet.

В таблице 2.4 приведены основные характеристики данной реализации.

Анализируя визуально график, приведенный на рис. 2.28 можно заметить, что сетевой трафик является значительно неравномерным процессом: имеются выбросы достаточно сильной амплитуды, в то время как среднее значение трафика – относительно мало. Описывая такое поведение, часто говорят, что данный процесс подчиняется одному из законов распределения с “тяжелыми хвостами” (типа Парето) или имеет “берстную” структуру (высокую пачечность).

Табл.2.4. Основные характеристики реализации Weth04_0.05.dat

Параметр/Временной ряд	<i>Weth04_0.05.dat</i>
Уровень агрегирования	0,05 сек
Длительность реализации	504 010 отсчетов по 0,05 сек, 7 часов
Среднее	1.348998e+05 байт/сек ~ 1,079 Мбит/с

СКО	2.558526e+05 байт/сек ~ 2,046 Мбит/с
СКО/Среднее	1,89

В частности, наибольший выброс в данной реализации превышает 58 Мбит/с (см. рис.2.28) (при потенциально достижимой скорости 100 Мбит/с в данном канале согласно спецификации Fast Ethernet), однако среднее значение трафика соответствует ~1 Мбит/с (см. табл. 2.4).

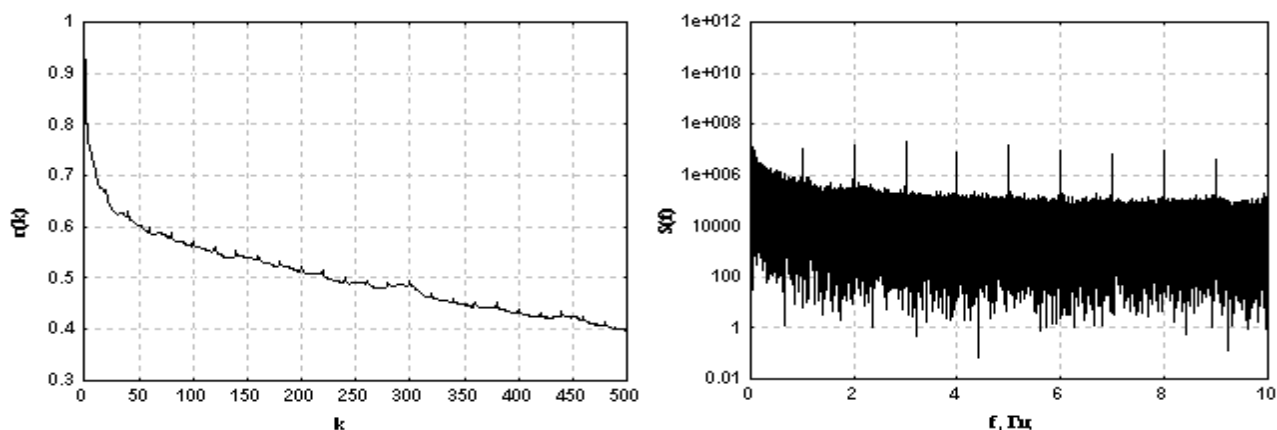


Рис. 2.29. а) Автокорреляционная функция (АКФ) временного ряда Weth04_0.05.dat
 б) Энергетический спектр временного ряда Weth04_0.05.dat

На рис. 2.29 можно видеть автокорреляционную функцию (а) и энергетический спектр (б) анализируемого процесса (в логарифмическом масштабе по оси ординат). Изучая АКФ, легко заметить достаточное слабое ее убывание: даже при сдвиге 500 значение АКФ остается выше 0,4. В то же время на графике АКФ отчетливо прослеживаются периодические пульсации. Присутствие значительных гармонических компонент также можно отследить по графику энергетического спектра. Очевидно наличие гармоник с частотами 1, 2, 3... Гц, которые не исчезают при изменении уровня агрегирования. Благодаря каким именно сетевым механизмам и протоколам возникают данные гармоники, сказать с уверенностью на данном этапе пока затруднительно. Отметим, однако, что аналогичные явления уже отмечались в [41], [42] (~11,9 Гц и 1,25 Гц), а также в [58] (~5 Гц).

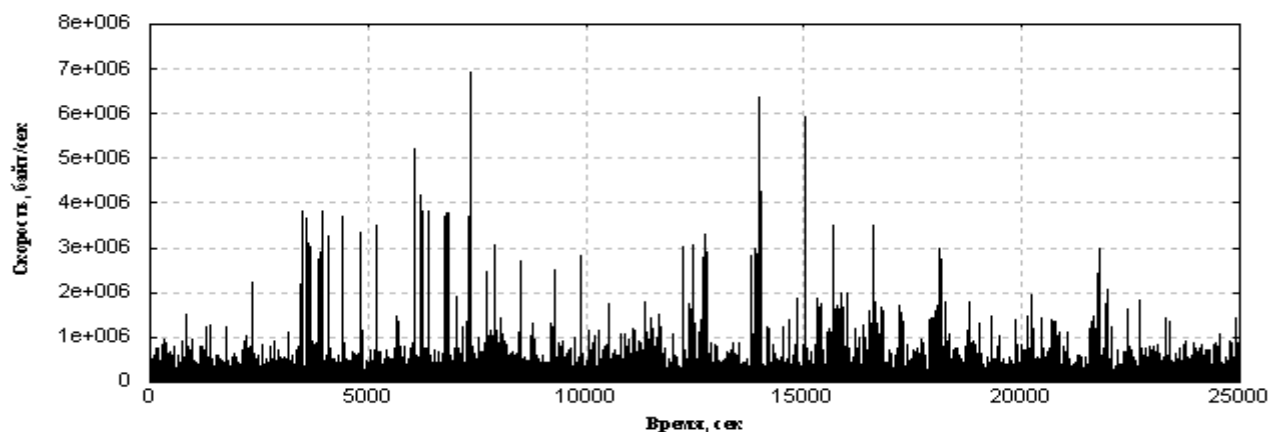


Рис. 2.30. Временной ряд *Wtcp04_0.05.dat*, соответствующий трафику (TCP) беспроводной сети

На рис. 2.30 представлен график реализации трафика *Wtcp04_0.05.dat*, представляющий собой агрегированный по уровню 0,05 сек временной ряд, соответствующий трафику протокола TCP, который был зафиксирован в точке “2” сети. Как можно видеть, профиль данного временного ряда практически полностью повторяет профиль временного ряда трафика *Weth04_0.05.dat*. Изучая автокорреляционные функции и энергетические спектры (см. рис. 2.31), также можно отметить присутствие выделяющихся гармонических компонент и слабое убывание АКФ данного вида трафика.

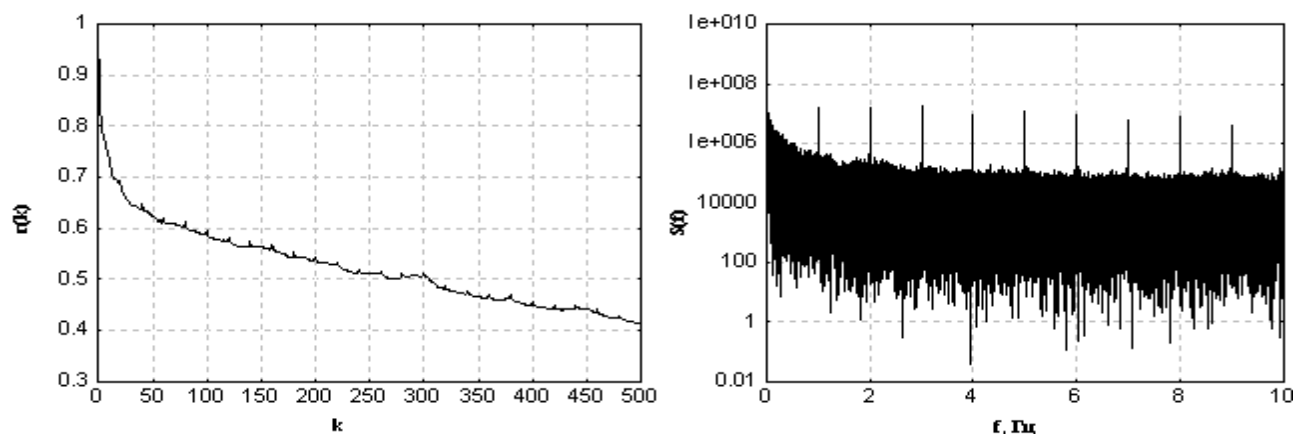


Рис. 2.31. а) Автокорреляционная функция временного ряда *Wtcp04_0.05.dat*.
 б) Энергетический спектр временного ряда *Wtcp04_0.05.dat*.

Таким образом, гармонические компоненты, профиль и самоподобная структура сетевого трафика проявляют себя уже на транспортном уровне, и,

по-видимому, определяются либо спецификой работы непосредственно самого протокола TCP, либо алгоритмами работы протоколов и механизмов верхних уровней модели OSI.

Табл.2.5. Участки временных реализаций

Номер участка	Положение на оси времени, секунд с начала реализации	Длительность
1	0,52 – 2257,58	45 143 отсчетов по 0,05 сек; 0,63 ч
2	3954,87 – 6142,53	43 753 отсчетов по 0,05 сек; 0,61 ч
3	8070,16 – 10108,64	40 770 отсчетов по 0,05 сек; 0,57 ч
4	12546,54 – 13933,84	27 746 отсчетов по 0,05 сек; 0,38 ч

Более детальный анализ полученных реализаций в настоящей работе производился с помощью изучения и сравнения характеристик временных рядов Weth04_0.05.dat, Weth04_0.05.src, Weth04_0.05.dst, Wtcp04_0.05.dat на одинаковых временных интервалах. Для этого во всех изучаемых реализациях были выделены участки 1, 2, 3 и 4, положение на оси времени и длительность которых отражены в табл. 2.5.

На каждом из участков для изучаемых реализаций трафика были оценены средние, среднеквадратические отклонения (СКО), экспоненты Хэрста Н (методами Виттла и Эбри-Вейча). Основные характеристики изучаемых реализаций приведены в табл. 2.6, 2.7, 2.8, 2.9.

Табл.2.6. Характеристики реализации Weth04_0.05.dat

Параметр	Участок 1	Участок 2	Участок 3	Участок 4
Среднее, байт/сек	8,981711e+04	1,227245e+05	1,182806e+05	2,878128e+05
СКО, байт/сек	9,694183e+04	1,898454e+05	1,245987e+05	6,336444e+05
СКО/Среднее	1,07	1,54	1,05	2,2
Параметр Хэрста (Whittle Estimator)	0,780 (95% Confidence Intervals [0,773-0,787])	0,964 (95% Confidence Intervals [0,957-0,972])	0,772 (95% Confidence Intervals [0,765-0,780])	0,978 (95% Confidence Intervals [0,967-0,988])
Параметр Хэрста (Abry-Veitch Estimator)	0,859 (95% Confidence Intervals [0,851-0,867])	1,045 (95% Confidence Intervals [1,037-1,054])	0,811 (95% Confidence Intervals [0,803-0,819])	0,953 (95% Confidence Intervals [0,941-0,965])

Табл.2.7. Характеристики реализации Weth04_0.05.dst

Параметр	Участок 1	Участок 2	Участок 3	Участок 4
Среднее,	4,313164e+04	6,309383e+04	6,765121e+04	5,603362e+04

байт/сек				
СКО, байт/сек	7,803364e+04	8,726733e+04	1,096809e+05	7,574341e+04
СКО/Среднее	1,8	1,3	1,6	1,35
Параметр Херста (Whittle Estimator)	0,817 (95% Confidence Intervals [0,809-0,824])	0,721 (95% Confidence Intervals [0,713-0,728])	0,796 (95% Confidence Intervals [0,788-0,803])	0,751 (95% Confidence Intervals [0,741-0,761])
Параметр Херста (Abry-Veitch Estimator)	0,884 (95% Confidence Intervals [0,876-0,892])	0,762 (95% Confidence Intervals [0,754-0,770])	0,835 (95% Confidence Intervals [0,827-0,843])	0,820 (95% Confidence Intervals [0,809-0,832])

Табл.2.8. Характеристики реализации Weth04_0.05.src

Параметр	Участок 1	Участок 2	Участок 3	Участок 4
Среднее, байт/сек	2,424995e+04	2,370111e+04	2,845095e+04	3,321249e+04
СКО, байт/сек	4,263671e+04	5,473894e+04	7,595552e+04	5,369175e+04
СКО/Среднее	1,7	2,3	2,6	1,6
Параметр Херста (Whittle Estimator)	0,801 (95% Confidence Intervals [0,794-0,808])	0,716 (95% Confidence Intervals [0,709-0,723])	0,858 (95% Confidence Intervals [0,851-0,865])	0,716 (95% Confidence Intervals [0,706-0,726])
Параметр Херста (Abry-Veitch Estimator)	0,856 (95% Confidence Intervals [0,848-0,864])	0,742 (95% Confidence Intervals [0,734-0,751])	0,925 (95% Confidence Intervals [0,917-0,933])	0,770 (95% Confidence Intervals [0,758-0,781])

Табл.2.9. Характеристики реализации Wtcp04_0.05.dat

Параметр	Участок 1	Участок 2	Участок 3	Участок 4
Среднее, байт/сек	5.227452e+04	8.656856e+04	8.085651e+04	2.287812e+05
СКО, байт/сек	7.978429e+04	1.761661e+05	1.092429e+05	5.932242e+05
СКО/Среднее	1,53	2,03	1,35	2,59
Параметр Херста (Whittle Estimator)	0,770 (95% Confidence Intervals [0,763-0,777])	0,973 (95% Confidence Intervals [0,966-0,980])	0,755 (95% Confidence Intervals [0,747-0,762])	0,983 (95% Confidence Intervals [0,973-0,993])
Параметр Херста (Abry-Veitch Estimator)	0,834 (95% Confidence Intervals [0,826-0,842])	1,057 (95% Confidence Intervals [1,049-1,065])	0,784 (95% Confidence Intervals [0,776-0,792])	0,967 (95% Confidence Intervals [0,955-0,978])

Анализируя результаты оценок основных параметров можно заметить, что трафик в канале Internet асимметричен: уровень среднего исходящего трафика (реализация Weth04_0.05.src) беспроводных клиентов меньше, чем уровень среднего входящего (реализация Weth04_0.05.dst). Отмеченное явление достаточно типично для данного случая.

Значение экспоненты Херста для всех рассмотренных случаев стабильно больше 0,5, что говорит о проявлении эффекта самоподобия в

изучаемых реализациях. Однако с течением времени значение экспоненты Хэрста трафика несколько изменяется. Это особенно отчетливо видно для реализации Wtcp04_0.05.dat: оно принимает значения от 0,770 до 0,967.

Таким образом, проведенное исследование подтверждает наличие самоподобных свойств в трафике современных телекоммуникационных сетей, использующих в том числе технологии беспроводного доступа IEEE 802.11b. Самоподобные свойства проявляют себя в трафике как на канальном (Fast Ethernet), так и на транспортном (TCP) уровнях. Как видно из сравнения рис. 2.30 и 2.28 структура трафика канального уровня практически полностью определяется трафиком транспортного уровня. В трафике канального и транспортного уровней обнаружены значительные гармонические составляющие с частотами 1, 2, 3... Гц. В этой связи при разработке адекватных математических моделей телетрафика следует обращать внимание на присутствие периодических компонент.

Реализации трафика, полученные и изучаемые в настоящей работе, можно найти на информационном портале “Self-Similarity h.u.”, посвященном проблеме самоподобного телетрафика по адресу www.teletraffic.ru.

2.7 Выводы по главе 2

На основании проведенного в настоящей главе статистического анализа и изучения структуры различных реализаций сетевого трафика сформулируем основные выводы:

1. Показано, что ряды ВС-х, LBL-х, а также TCP-х, скорее всего, подчиняются некоторому распределению с так называемым “тяжелым хвостом” (например, Парето или логнормальному). Об этом также свидетельствует тот факт, что логарифмирование рядов приводит к

нормализации их распределений (см. рис. 2.18). Увеличение уровня агрегирования приводит к изменению параметров “тяжелохвостого” распределения. Однако более точный ответ может дать исследование распределений методом максимального правдоподобия.

2. Результаты регрессионного анализа АКФ реализаций сетевого трафика подтверждают присутствие медленно убывающих зависимостей для всех исследуемых случаев.

3. Произведено измерение показателя Хэрста (H) семью методами: анализа дисперсии, нормированного размаха (R/S), периодограмм, абсолютных моментов, дисперсии остатков, Эбри-Вейча и Виттла. Обнаружено, что для всех реализаций сетевого трафика $H > 0.5$, то есть трафик относится к классу персистентных процессов. В то же время для белого шума практически все методы показали $H \sim 0.5$, что подтверждает правильность измерений (см. табл. 2.3). Использование разнообразных методов оценки показателя Хэрста преследовало цель получить более достоверные результаты. Усредненное значение H для сетевого трафика $H \sim 0.8$. Зависимости коэффициента H от уровня агрегирования (для рассматриваемых уровней) не выявлено.

4. Показано, что зависимости количества ложных соседей (FNN) для суррогатных реализаций заметно отличаются от аналогичных для исходных реализаций и больше похожи на зависимость FNN для белого шума (см. 2.24). Данное замечание говорит в пользу того, что, возможно, свойства изучаемых реализаций трафика обусловлены не только корреляциями, свойственными $1/f^a$ – процессам.

5. Обнаружено, что кривые FNN (рис. 2.24) для агрегированных реализаций трафика занимают некоторое промежуточное положение (по степени убывания количества ложных соседей) между чисто хаотическим процессом и число случайным процессом (белым шумом). Этот факт может

служить в пользу классификации сетевого трафика как некоторого детерминированного (возможно хаотичного) процесса.

6. Впервые обнаружено, что кривые корреляционного интеграла для изучаемых реализаций трафика имеют характерный изгиб. Последний наиболее выражен у реализаций ВС-5 (на интервале 200-800 Байт/с), ВС-10 (на интервале 100-300 Байт/с), а также у хаотической реализации системы Лоренца. В то же время показано, что кривые корреляционных интегралов для суррогатных реализаций больше напоминают кривые корреляционного интеграла для белого шума, чем аналогичные для хаотической реализации. Эти наблюдения отвергают гипотезу о классификации сетевого трафика как некоторого шумового процесса, обладающего коррелированной структурой. Другими словами, в трафике, возможно, присутствует некоторая детерминированная, однако достаточно слабая составляющая.

7. На основании анализа результатов вычисления BDS-статистики можно сделать вывод о том, что для хаотической реализации L , реализаций трафика ВС-5, ВС-10, LBL-0.1, LBL-1, TCP-0.1, TCP-1, их логарифмов, а также суррогатных реализаций гипотеза о статистической независимости членов ряда отвергается для всех рекомендованных значений ε и значений размерности погружения $m=2..10$. Из всех изучаемых случаев данная гипотеза не отвергается только для белого шума, что только подтверждает достоверность результатов анализа.

8. Подготовлен и выполнен оригинальный эксперимент по сбору и исследованию трафика беспроводной сети, подтверждающий наличие самоподобных свойств в трафике современных телекоммуникационных сетей, использующих технологии беспроводного доступа IEEE 802.11b в том числе. Полученные реализации трафика беспроводной сети размещены для публичного пользования в Интернет по адресу www.teletraffic.ru.

9. Обнаружено, что в агрегированном трафике канального и транспортного уровней присутствуют значительные гармонические

составляющие. В этой связи при разработке адекватных математических моделей телетрафика следует обращать внимание на присутствие периодических компонент. Кроме того, обнаруженное явление выявляет присутствие регулярной детерминированной составляющей в агрегированном сетевом трафике, что может быть интересным при решении задач прогнозирования телетрафика.

10. На основании анализа результатов данной главы сетевой трафик можно классифицировать как некоторый сложный, похожий на случайный, однако предсказуемый процесс. В этой связи механизм обеспечения качества обслуживания, предназначенный для работы в системах с самоподобным трафиком, может основываться на алгоритмах прогнозирования трафика.

Глава 3. Исследование возможностей прогнозирования самоподобного телетрафика

Целью данной главы является исследование возможности прогнозирования сетевого трафика с помощью различных алгоритмов предсказания и выбор наиболее подходящего из них для применения в разрабатываемом механизме обеспечения качества обслуживания.

3.1 Предпосылки к прогнозированию самоподобного сетевого трафика

Сетевой трафик, приведенный к эквидистантной форме по оси времени (с помощью процедуры агрегирования) представляет собой некоторый временной ряд (реализацию дискретного случайного процесса), мгновенные значения которого можно пытаться предсказывать с помощью существующих методик прогнозирования. Поясним принципиальные возможности осуществления такого прогноза.

Числовой характеристикой случайного процесса, напрямую связанной с его предсказуемостью является *интервал корреляции* τ_k , определяемый выражением

$$\tau_k = \int_0^{\infty} |r(\tau)| d\tau, \quad (3.1)$$

где $r(\tau)$ – нормированная функция корреляции случайного процесса. Если известно поведение реализации в прошлом, то возможен вероятностный прогноз случайного процесса на время порядка τ_k [77]. Однако прогнозирование процесса на большее время оказывается безрезультатным, поскольку далеко отстоящие друг от друга отсчеты являются статистически несвязанными (некоррелированными). Чтобы уяснить физический смысл понятия интервала корреляции достаточно взглянуть на рис. 3.1.

Анализируя его, можно заметить, что интервал корреляции численно равен площади под графиком нормированной корреляционной функции случайного процесса.

Вспомним, что все определения самоподобного процесса даются в условиях гиперболически убывающей корреляционной функции:

$$r(\tau) \sim \tau^{-\beta}, \tau \rightarrow \infty, 0 < \beta < 1. \quad (3.2)$$

Подставив (3.2) в (3.1), получим

$$\tau_k = \int_0^{+\infty} |\tau^{-\beta}| d\tau.$$

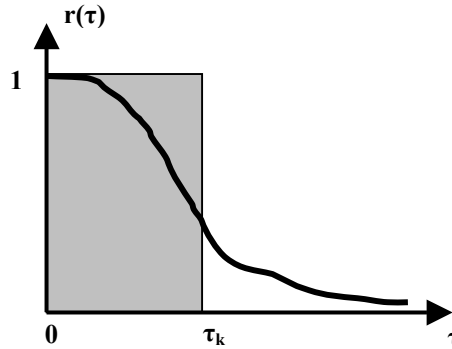


Рис. 3.1. К определению интервала корреляции (площадь под кривой равна площади заштрихованного прямоугольника)

Решая последнее выражение в общем случае для любого β :

1) Если $\beta \neq 1$, то для любого $R > 0$

$$\lim_{R \rightarrow +\infty} \int_0^R \frac{d\tau}{\tau^\beta} = \lim_{R \rightarrow +\infty} \left. \frac{\tau^{1-\beta}}{1-\beta} \right|_0^R = \lim_{R \rightarrow +\infty} \frac{R^{1-\beta}}{1-\beta} = \begin{cases} 0 & \text{при } \beta > 1 \\ \infty & \text{при } \beta < 1 \end{cases};$$

2) Если $\beta = 1$, то для любого $R > 0$

$$\lim_{R \rightarrow +\infty} \int_0^R \frac{d\tau}{\tau} = \lim_{R \rightarrow +\infty} \ln \tau \Big|_0^R = \infty.$$

Таким образом, для случая самоподобного процесса, характеризующегося $0 < \beta < 1$, имеем бесконечно протяженный интервал корреляции

$$\tau_k = \infty. \quad (3.3)$$

Данный результат обеспечивает собой теоретически принципиальную возможность прогнозирования самоподобного процесса. Конечно, в реальной ситуации (при изучении реального самоподобного сетевого трафика) сложно требовать строгого выполнения (3.2), однако даже в этом случае можно ожидать, что τ_k принимает достаточно большие значения.

Кроме того, вспомним, что с понятием “самоподобный сетевой трафик” тесно связаны такие понятия как *персистентность* и *длительная память*, которые отражают степень статистической связи между последовательными отсчетами процесса. Например, если в течение некоторого времени в прошлом наблюдались положительные приращения процесса, то есть происходило увеличение, то и впредь в среднем будет происходить увеличение. Иначе говоря, вероятность того, что процесс на $i+1$ шаге отклоняется от среднего в том же направлении, что и на i шаге настолько велика, насколько параметр Херста H близок к 1. Т.е. персистентные стохастические процессы обнаруживают четко выраженные тенденции изменения при относительно малом “шуме”.

Еще один аспект в пользу прогнозируемости сетевого трафика – наличие в распределении последних так называемых “тяжелых хвостов”. Как было показано выше (в главе 1), тяжелые хвосты также могут быть причиной хорошей предсказуемости.

Более того, при анализе реального сетевого трафика (глава 2), наряду с некоторой постоянной составляющей (математическим ожиданием процесса) и стохастической компонентой выявлены также регулярные (детерминированные) компоненты, соответствующие гармоническим колебаниям с определенными частотами. При построении прогностической модели трафика логично было бы учесть существование данных гармонических колебаний (в виде добавления некоторой периодической компоненты) и попытаться улучшить качество прогноза.

3.2 Задача динамического управления пропускной способностью канала с помощью прогнозирования. Оценки выигрыша

Как уже отмечалось выше, задача прогнозирования сетевого трафика сводится к задаче прогнозирования дискретного временного ряда. С точки зрения общей теории прогнозирования [53] временной ряд, соответствующий сетевому трафику можно представить в виде

$$x(k) = \chi(k) + \varepsilon(k), \quad (3.4)$$

где $\chi(k)$ - неслучайная компонента (включая в себя в общем случае тренд, циклические и сезонные колебания), $\varepsilon(k)$ - случайные (нерегулярные) колебания. Понятия “неслучайная компонента” и “случайная компонента” здесь введены достаточно условно. Например, в реальной ситуации случайная компонента $\varepsilon(k)$ может быть совершенно неслучайной по своей природе (относясь хотя бы к классу хаотических колебаний), но при всем при этом оставаться трудно-предсказуемой, в то же время так называемая “неслучайная компонента” $\chi(k)$ - в принципе может меняться произвольным образом. Однако если она меняется достаточно медленно, отражая относительно долговременные тенденции, то это позволяет нам достаточно точно предугадать ее последующие значения и отнести ее к разновидности “неслучайной компоненты”.

С этих позиций прогноз n -го значения временного ряда $x(k)$ складывается из прогностической оценки случайной и неслучайной компонент:

$$\hat{x}(n) = \hat{\chi}(n) + \hat{\varepsilon}(n), \quad (3.5)$$

поэтому качество прогноза ряда $x(k)$ тем выше, чем качественней оценки последних.

По нашему мнению, задачу прогнозирования сетевого трафика следует рассматривать не саму по себе, а в составе задачи динамического управления сетевыми ресурсами (пропускной способностью канала, в частности). Данное обобщение позволит не только сделать выводы относительно

прогнозируемости сетевого трафика, но и (что более важно) одновременно оценить выигрыш от применения алгоритма динамического управления по сравнению с классической схемой статического распределения ресурсов, что определяет целесообразность его использования.

В этой связи оценку $\widehat{C}(n)$ пропускной способности канала, для которого предназначен трафик $x(k)$, на n -ом шаге будем рассчитывать в виде суммы оценки трафика на n -ом шаге $\widehat{x}(n)$ (динамическая составляющая) и некоторого постоянного уровня bs (статически заданная составляющая пропускной способности):

$$\widehat{C}(n) = \widehat{x}(n) + bs = \widehat{\chi}(n) + \widehat{\varepsilon}(n) + bs. \quad (3.6)$$

Отметим, что, если в формуле (3.6) положить $\widehat{x}(n)=0$, мы придем к широко используемому в настоящее время методу статического задания пропускной способности со всеми его недостатками в отношении берстного самоподобного сетевого трафика. В то же время, если положить $bs=0$, то получим спрогнозированные значения непосредственно самого сетевого трафика $x(k)$.

На рис. 3.1 (а, б) показан прогнозируемый дискретный временной ряда $x(k)$ (сетевой трафик в эквидистантной форме), оцененные значения $\widehat{C}(k)$ пропускной способности канала, а также абсолютная ошибка $e(k)$, которая вычисляется по формуле

$$e(k) = x(k) - \widehat{C}(k) \quad (3.7)$$

и фактически является невязкой между исходным трафиком и рассчитанными значениями пропускной способности.

На практике обычно стремятся уменьшить абсолютную ошибку $e(k)$. В качестве величины, оценивающей качество алгоритма прогнозирования, часто пользуются так называемым отношением сигнал/шум:

$$SN = \frac{M[x(k)^2]}{M[e(k)^2]} = \frac{\sum x(k)^2}{\sum e(k)^2}, \quad (3.8)$$

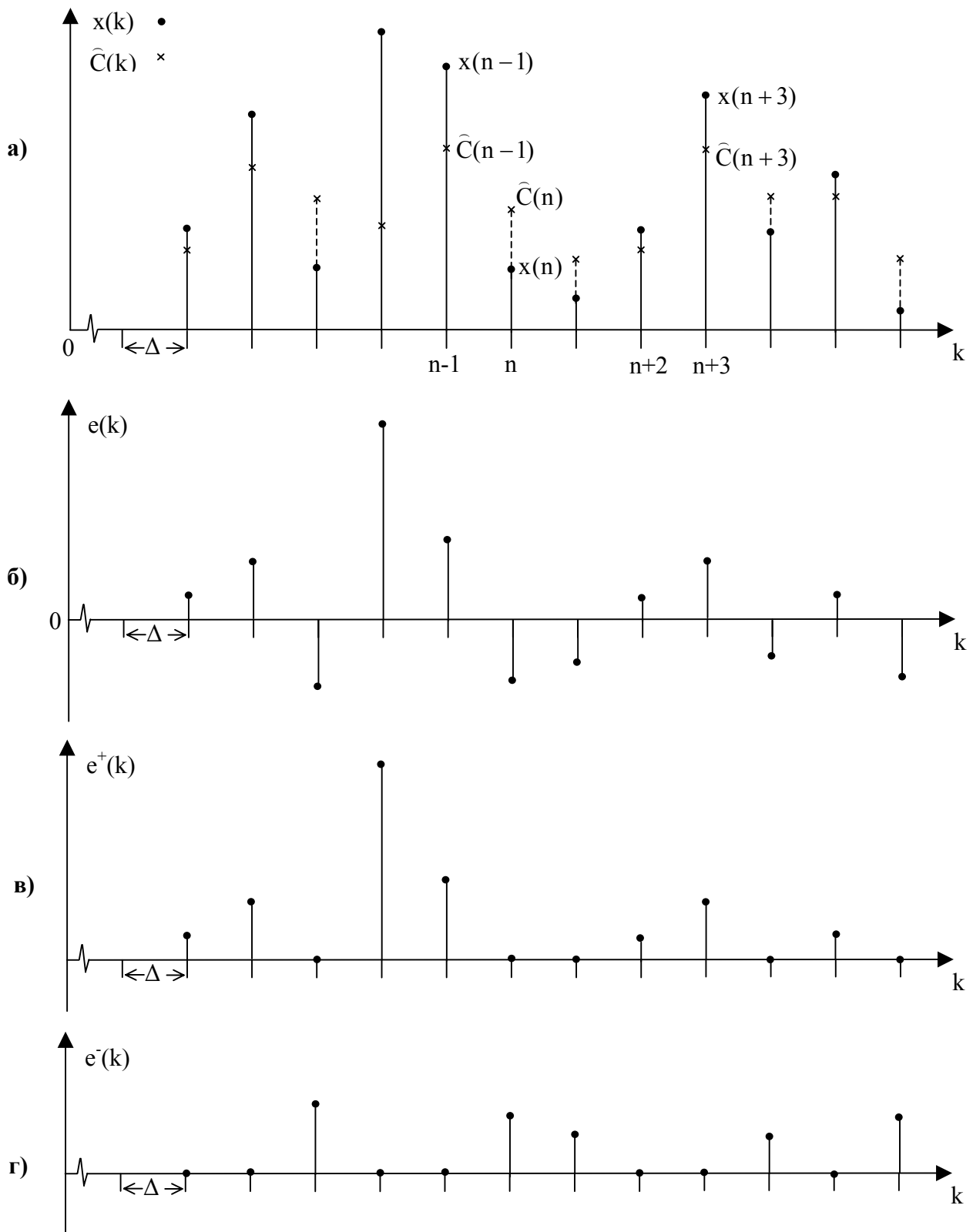


Рис.3.1 а) Прогнозируемый участок $x(k)$ и оцененные значения $\hat{C}(k)$,
 б) Абсолютная ошибка прогноза $e(k)$,
 в) Ошибка недооценки $e^+(k)$,
 г) Ошибка переоценки $e^-(k)$

либо различными его модификациями [52], [51] (обратному отношению, выражению в децибелах и пр.). Однако такая оценка, как легко заметить, зависит от среднего значения ряда $x(k)$, что не позволяет сравнивать качество прогноза рядов с различным средним между собой.

Поэтому в дальнейшем исследовании предлагается использовать несколько доработанную оценку

$$SNR^{-1} = \frac{M[e(k)^2]}{M[(x(k) - M[x(k)])^2]}, \quad (3.9)$$

которая не зависит от среднего значения прогнозируемого участка ряда и отражает степень улучшения прогноза трафика по сравнению с прогнозированием по среднему значению. Другими словами оценка $SNR^{-1} = 1$, если качество прогноза такое же, как в случае прогноза по среднему значению прогнозируемого участка ряда и $SNR^{-1} < 1$, если качество прогноза лучше. Благодаря такому свойству, оценка SNR^{-1} позволяет сравнивать качество прогноза различных временных рядов между собой.

Кроме того, наряду с абсолютной погрешностью прогноза $e(k)$ будем рассматривать отдельно ошибки недооценки ряда (см. рис.3.1, в):

$$e^+(k) = \begin{cases} e(k), & \text{если } e(k) \geq 0 \\ 0, & \text{если } e(k) < 0 \end{cases}, \quad (3.10)$$

которые с точки зрения управления пропускной способностью канала связаны с количеством потерянной информации, и ошибки переоценки ряда (см. рис.3.1, г):

$$e^-(k) = \begin{cases} |e(k)|, & \text{если } e(k) < 0 \\ 0, & \text{если } e(k) \geq 0 \end{cases}, \quad (3.11)$$

которые, в свою очередь, отражают количество недоиспользованных ресурсов.

В качестве оценок качества прогноза будем также рассматривать *коэффициент недооценки (коэффициент потерь)*:

$$D^+ = \frac{M[e^+(k)]}{M[x(k)]} = \frac{\sum e^+(k)}{\sum x(k)} \quad (3.12)$$

и коэффициент переоценки (коэффициент недоиспользования):

$$D^- = \frac{M[e^-(k)]}{M[x(k)]} = \frac{\sum e^-(k)}{\sum x(k)}. \quad (3.13)$$

Эти коэффициенты, хоть и зависят от среднего значения прогнозируемого участка ряда, однако имеют четкую физическую интерпретацию: при заданном способе оценки пропускной способности C канала коэффициент недооценки D^+ выражает отношение количества потерянной информации к общему количеству информации, которое нужно было обработать (пропустить через канал). Соответственно, коэффициент переоценки D^- отражает количество неиспользованной пропускной способности канала к общему количеству информации, которое нужно было пропустить через канал. Очевидно, чем ближе прогностические оценки $\hat{C}(k)$ к действительным значениям $x(k)$, тем ближе к нулю коэффициенты D^+ и D^- . Данные коэффициенты позволят нам для каждого из трафиков сравнить механизмы динамического выделения пропускной способности канала (основанные на различных методах прогнозирования) с классическим случаем статического задания пропускной способности.

Эксперимент по исследованию возможностей механизма динамического выделения (на основании прогноза) пропускной способности канала проведем с помощью следующего алгоритма:

- в исходном дискретном временном ряде $x(k)$, соответствующем трафику выделим так называемый *тренировочный участок* T фиксированной длины LT ;
- изучая характеристики ряда на данном тренировочном участке (оценивая параметры прогностической модели), формируем прогноз $\hat{C}(i)$ (на один шаг вперед) i -го значения ряда $x(k)$, следующего за концом тренировочного участка;
- фиксируем получившуюся абсолютную ошибку прогноза $e(i)$;

- сдвигаем тренировочный участок T длины LT на один шаг вперед, считая, что к наступившему моменту времени нам уже стало известно действительное значение только что спрогнозированного отсчета i ;
- осуществляем прогноз следующего значения;
- и.т.д.

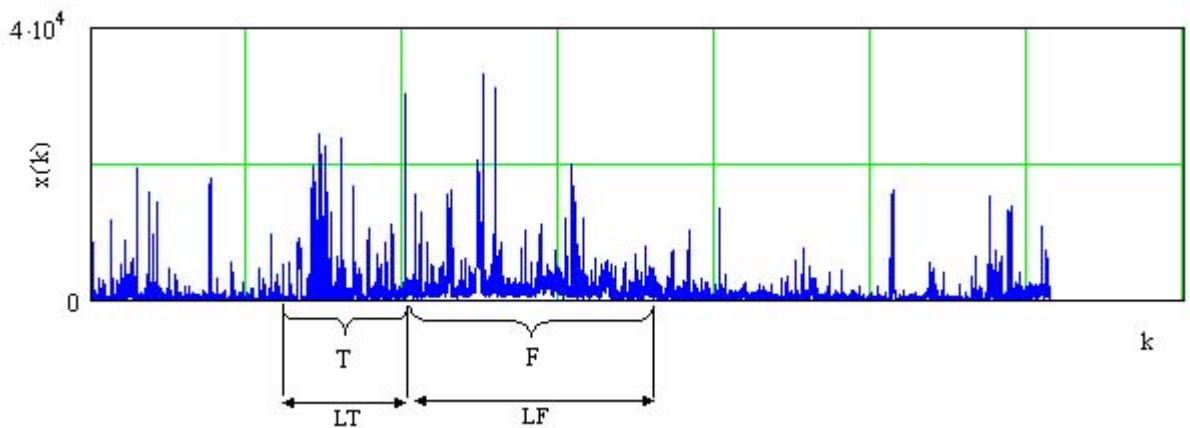


Рис.3.2. К пояснению алгоритма эксперимента по прогнозированию трафика

Примечание. Договоримся, что в момент прогнозирования отсчета n исходного ряда, соответствующий тренировочный ряд будем обозначать $T(n)$. В результате повторения такой процедуры LF раз мы получим прогностические оценки $\hat{C}(k)$ и ошибки прогноза $e(k)$ для прогнозируемого участка F длины LF исходного ряда (см. рис.3.2).

3.3. Анализ алгоритмов управления пропускной способностью канала

В данном разделе рассматриваются способы распределения пропускной способности канала: статический и динамические (с различными предсказательными алгоритмами). На основе проведенного статистического анализа и оценок характеристик, введенных в предыдущем разделе (SNR^{-1} , D^+ и D^-), изучается эффективность применения того или иного метода для случая реального трафика, обладающего самоподобием. Для обеспечения возможности сравнения результатов применения различных методов распределения

пропускной способности и алгоритмов прогнозирования все статистические исследования в данном разделе производятся на временном ряде ВС-10. Данный ряд представляет собой агрегированную по уровню 10 с временную реализацию сетевого трафика протокола Ethernet, которая подробно изучалась в главе 2 настоящей диссертации. В качестве прогнозируемого участка F выбран интервал между отсчетами 3000 и 6000, которые содержит 3000 отсчетов и характеризуется общей длительностью ~ 8.3 часа.

3.3.1 Статическое задание пропускной способности

Как было показано выше, статический способ задания пропускной способности канала основывается на соотношении (3.6) при условии $\hat{x}(n)=0$. То есть оценка пропускной способности в i -ый момент рассчитывается следующим образом:

$$\hat{C}(i) = bs. \quad (3.14)$$

Это классический случай, при котором сетевой администратор должен выбрать фиксированную пропускную способность для канала (которая напрямую связана со стоимостью канала), мирясь при этом с большими перегрузками и одновременно низкой утилизацией, что является нетривиальной задачей. Для того, чтобы лучше понять этот факт будем изменять значение bs и наблюдать за поведением оценок SNR^{-1} , D^+ и D^- для случая реального сетевого трафика.

Поскольку сетевой трафик по своей физической сути представляет собой неотрицательный ряд, то при $bs=0$ в качестве абсолютной ошибки $e(k)$ будет выступать непосредственно сам исходный ряд $x(k)$. При этом $SNR^{-1} > 1$, максимальны ошибки недооценки ($D^+=1$) и минимальны ошибки переоценки ($D^-=0$). При увеличении значения bs в момент, когда оно равно среднему прогнозируемого участка ($M[F]$), показатель SNR^{-1} достигает своего минимума ($SNR^{-1}=1$), а коэффициенты D^+ и D^- становятся равны. При дальнейшем увеличении bs коэффициент SNR^{-1} снова увеличивается, в то время как ошибки недооценки D^+ продолжают уменьшаться, а ошибки переоценки растут.

Характерно, что $SNR^{-1}=1$ при $bs = M[F]$. Согласно (3.6) это можно интерпретировать как

$$\widehat{C}(n) = \widehat{x}(n) + bs = M[F] + 0, \quad (3.15)$$

что как раз соответствует случаю прогнозирования ряда участка F ряда $x(k)$ по его среднему значению ($M[F]$).

Рассмотрим возможности данного метода на примере временного ряда BC10 (см. рис. 3.3).

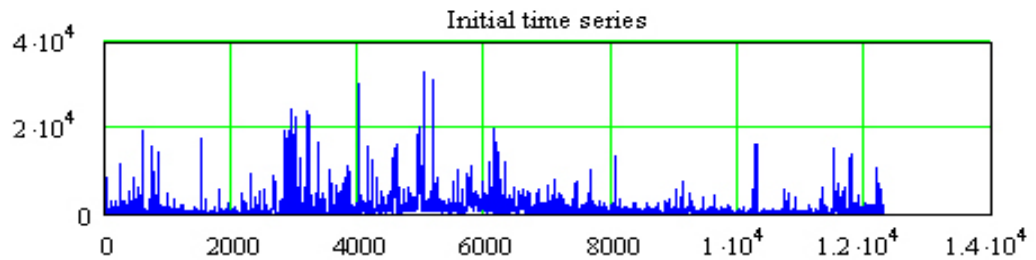


Рис. 3.3. Временной ряд BC10, соответствующий реальному трафику

Для определенности выберем в качестве прогнозируемого участка F участок между отсчетами 3000 и 6000. Длина LT тренировочного участка T в данном случае не имеет никакого значения.

На рис. 3.4 представлены результаты расчета оценок SNR^{-1} , D^+ и D^- для различных значений параметра bs . Для удобства по оси абсцисс отложена нормированная средняя пропускная способность канала в долях среднего $M[F]$ прогнозируемого участка F трафика:

$$bs_norm = \frac{M[\widehat{C}(n)] - M[F]}{M[F]}. \quad (3.16)$$

Для случая статического задания пропускной способности выражение (3.16) можно переписать в виде

$$bs_norm = \frac{bs - M[F]}{M[F]}.$$

Обратим также внимание, что случай $bs_norm = 0$ соответствует выбору в качестве пропускной способности канала среднего прогнозируемого участка F.

При этом, коэффициент недооценки $D^+ = 0.38$, что соответствует потерям 38% обрабатываемой информации.

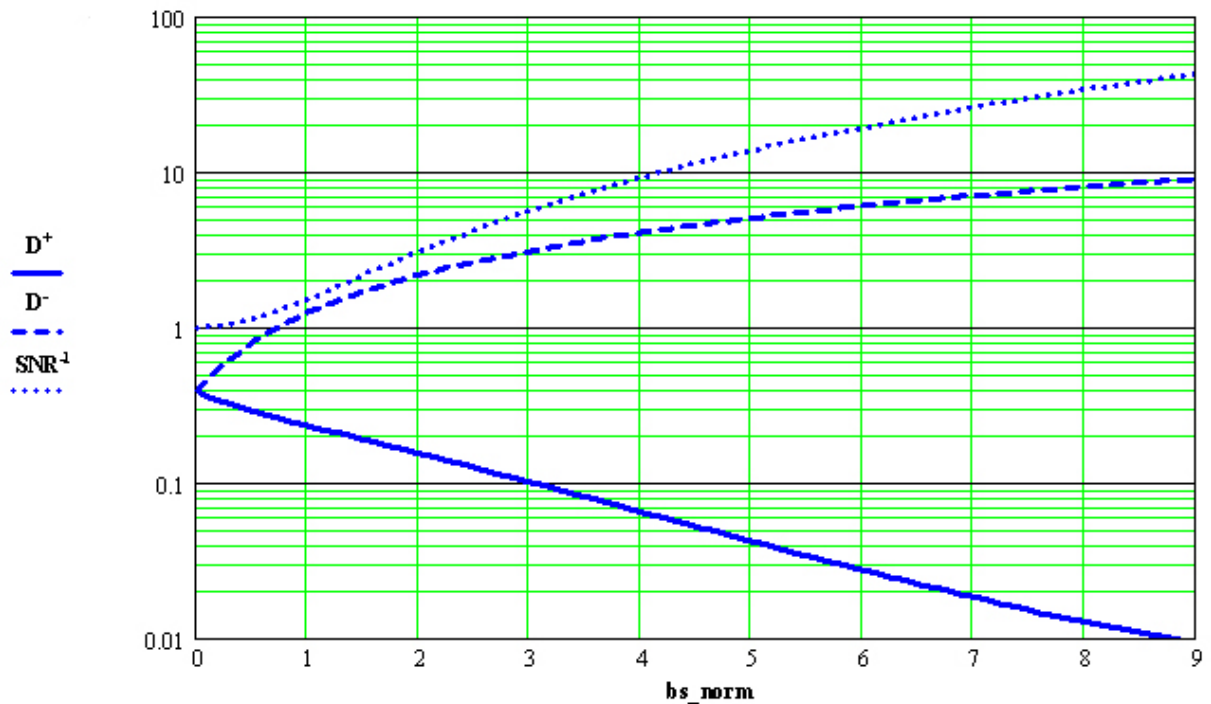


Рис. 3.4. Оценки статистических характеристик при статическом задании пропускной способности канала

В дальнейшем, мы будем пользоваться полученными кривыми для того, чтобы оценить выигрыш при использовании методов динамического распределения пропускной способности, сравнивая их между собой.

3.3.2 Динамическое распределение пропускной способности с простым предсказателем

В данном случае оценка пропускной способности канала в i -ый момент рассчитывается по формуле (3.6) в которой в качестве оценки неслучайной компоненты $\hat{\chi}(i)$ возьмем среднее тренировочного участка на i -ом шаге $M[T(i)]$, а в качестве оценки случайной компоненты выберем последнее известное значение усредненного тренировочного ряда T :

$$\hat{C}(i) = \hat{x}(i) + bs = \hat{\chi}(i) + \hat{\varepsilon}(i) + bs = M[T(i)] + (x(i-1) - M[T(i)]) + bs. \quad (3.17)$$

Нетрудно видеть, что (3.17) также может быть представлена в виде:

$$\widehat{C}(i) = x(i-1) + bs. \quad (3.18)$$

Следовательно, значение пропускной способности рассчитывается в виде суммы последнего известного отсчета исходного временного ряда, соответствующего агрегированному трафику и некоторого постоянного уровня bs . Это случай, так называемого простого (“наивного”) или naïve-предсказателя. Результаты расчета оценок SNR^{-1} , D^+ и D^- для ряда BC-10 представлены на рис. 3.5. Можно отметить улучшение показателей по сравнению со статическим заданием полосы. В частности, при $bs_norm=0$ коэффициент SNR^{-1} снизился с 1 до значения ~ 0.46 , а коэффициенты D^+ и D^- снизились со значения ~ 0.38 до ~ 0.21 . При этом убывание кривой, соответствующей коэффициенту D^+ , происходит заметно быстрее, а возрастание кривой, соответствующей коэффициенту D^- , происходит заметно медленнее по сравнению со случаем статического задания пропускной способности (см. рис. 3.4).

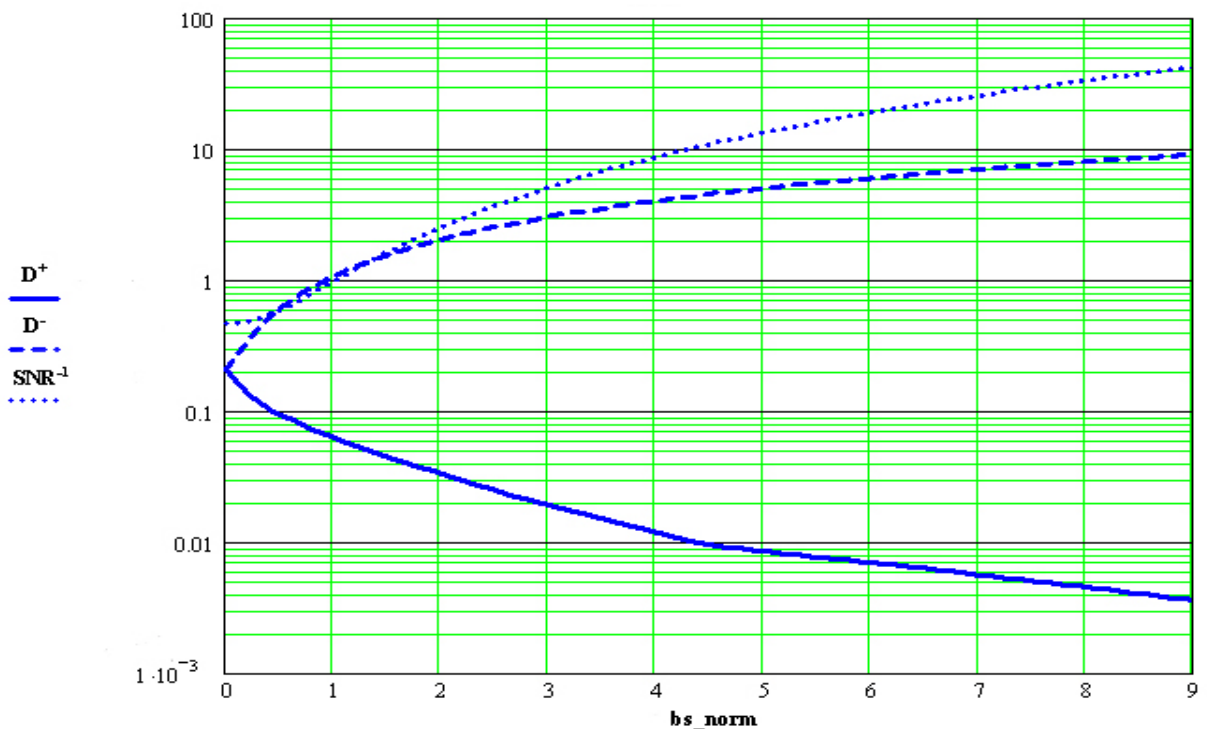


Рис. 3.5. Оценки статистических характеристик при динамическом задании пропускной способности канала с простым предсказателем

Другими словами, при тех же самых значениях средней пропускной способности канала (bs_norm) в случае динамического распределения

пропускной способности канала с простым предсказателем потерь информации будет значительно меньше, а использование канала на столько же больше. Что говорит о преимуществах данного алгоритма по сравнению со статическим заданием пропускной способности. Более детальный анализ сравнения полученных оценочных кривых будет проведен ниже.

3.3.3 Динамическое распределение пропускной способности с авторегрессионным предсказателем первого порядка

Одним из способов описания поведения и соответственно прогнозирования случайной компоненты $\varepsilon(k)$ (см. (3.6)) в теории временных рядов является подбор подходящей модели. Как правило, сначала задается общий вид модели, для которой методами статистического оценивания по имеющемуся в распоряжении временному ряду вычисляются необходимые параметры модели.

В качестве популярной модели используется общий линейный процесс (3.19), при котором анализируемый временной ряд $\varepsilon(k)$ получается в виде классической линейной модели множественной регрессии, когда в роли объясняющих переменных выступают значения самого временного ряда во все прошлые моменты времени:

$$\varepsilon(i) = \pi_1 \varepsilon(i-1) + \pi_2 \varepsilon(i-2) + \dots + \delta(i) = \sum_{k=1}^{\infty} \pi_k \varepsilon(i-k) + \delta(i), \quad (3.19)$$

где $\delta(i)$ означает временную последовательность, образующую белый шум. При этом весовые коэффициенты подбираются таким образом, чтобы обеспечить стационарность смоделированного ряда $\varepsilon(k)$.

Частным случаем (3.19) является модель авторегрессии 1-го порядка – AR(1) (марковский процесс). Она представляется в следующем виде:

$$\varepsilon(i) = \alpha \varepsilon(i-1) + \delta(i), \quad (3.20)$$

где α - некоторый числовой коэффициент, не превосходящий по модулю единицу.

Аппроксимация временного ряда $\varepsilon(k)$ с помощью модели (3.20), то есть поиск оценок коэффициента α и белого шума $\delta(k)$, осуществляется в условиях формулирования критерия минимизации необходимой характеристики. Наиболее популярный критерий – минимизация среднеквадратической ошибки $\sqrt{M[(\varepsilon(i) - \hat{\varepsilon}(i))^2]}$. Тогда в качестве оценки коэффициента α выбирается значение нормированной автокорреляционной функции процесса $\varepsilon(k)$ при сдвиге равном единице, т.е. $r(1)$. В этом случае оптимальным в смысле выбранного критерия предсказателем является выражение

$$\hat{\varepsilon}(i) = \hat{\alpha}\varepsilon(i - 1) = r(1)\varepsilon(i - 1).$$

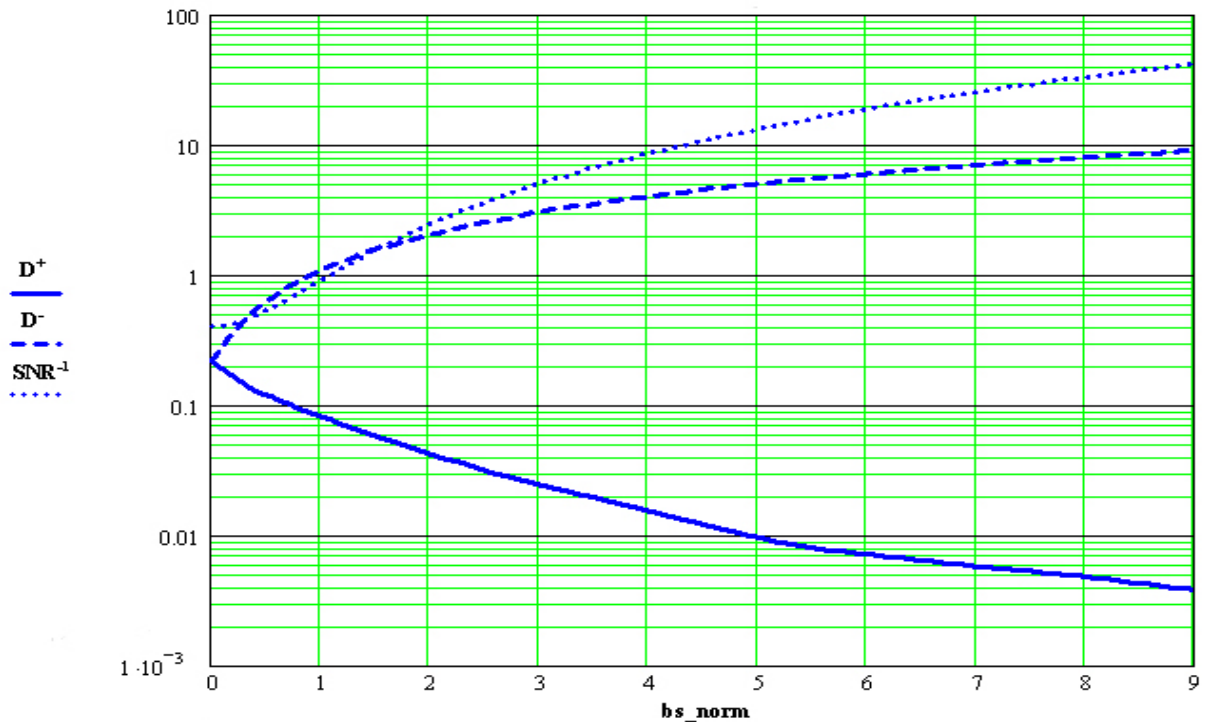


Рис. 3.6. Оценки статистических характеристик при динамическом задании пропускной способности канала с авторегрессионным предсказателем первого порядка

Таким образом, окончательное выражение для оценки пропускной способности канала с помощью авторегрессионного предсказателя 1-го порядка представляется в таком виде:

$$\hat{C}(i) = M[T(i)] + r(1) \cdot (x(i - 1) - M[T(i)]) + bs. \quad (3.21)$$

Результаты расчета оценок SNR^{-1} , D^+ и D^- для ряда BC-10 представлены на рис. 3.6. Можно отметить некоторое улучшение показателя SNR^{-1} по сравнению со случаем простого предсказателя. Однако правильнее ориентироваться на характеристики D^+ и D^- , которые имеют четкую физическую интерпретацию и непосредственно отражают процент потерь информации и использование канала. С точки зрения коэффициентов недооценки и переоценки авторегрессионный предсказатель первого порядка все же уступает простому предсказателю в эффективности прогнозирования.

Более детальное сравнение рассмотренных алгоритмов будет произведено в разделе 3.4.

3.3.4 Динамическое распределение пропускной способности с авторегрессионным предсказателем второго порядка

Авторегрессионная модель второго порядка, как и марковский процесс (3.20), является частным случаем общего линейного процесса (3.19) при равенстве нулю всех коэффициентов π_k кроме первых двух. Соответственно она может быть представлена в виде

$$\varepsilon(i) = \alpha_1 \varepsilon(i-1) + \alpha_2 \varepsilon(i-2) + \delta(i). \quad (3.22)$$

При минимизации среднего квадрата абсолютной ошибки $\varepsilon(i) - \hat{\varepsilon}(i)$ параметры α_1 и α_2 вычисляются на основании соотношений (3.23) и (3.24):

$$r(1) = \frac{\hat{\alpha}_1}{1 - \hat{\alpha}_2}, \quad (3.23)$$

$$r(2) = \hat{\alpha}_2 + \frac{\hat{\alpha}_1^2}{1 - \hat{\alpha}_2}. \quad (3.24)$$

В этом случае выражение для оценки пропускной способности канала с помощью авторегрессионного предсказателя 2-го порядка представляется в виде

$$\hat{C}(i) = M[T(i)] + \hat{\alpha}_1 \cdot (x(i-1) - M[T(i)]) + \hat{\alpha}_2 (x(i-2) - M[T(i)]) + bs. \quad (3.25)$$

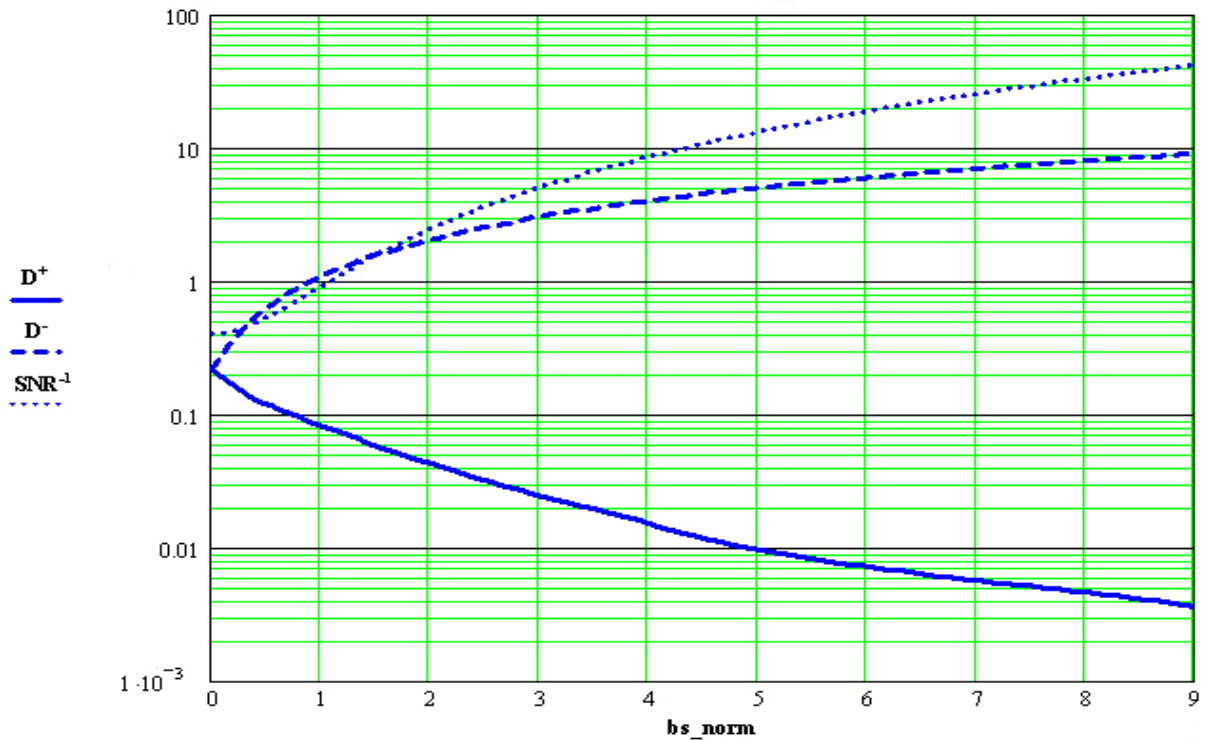


Рис. 3.7. Оценки статистических характеристик при динамическом задании пропускной способности канала с авторегрессионным предсказателем второго порядка

Результаты расчета оценочных статистик представлены на рис. 3.7 и фактически неотличимы от результатов, полученных с помощью соотношения (3.22) для случая авторегрессионного предсказателя первого порядка.

3.3.5 Динамическое распределение пропускной способности с ARMA-предсказателем

Для описания и прогнозирования поведения случайной компоненты $\varepsilon(k)$ можно использовать авторегрессионную модель скользящего среднего порядка (p,q) , обозначаемую как ARMA (p,q) [53]. В общем виде модель ARMA (p,q) для процесса $\varepsilon(i)$ записывается как

$$\alpha(B)\varepsilon(i) = \beta(B)\delta(i), \quad (3.26)$$

где $\delta(i)$ - временная последовательность, образующая белый шум, $\alpha(B)$, $\beta(B)$ - полиномы степень p и q соответственно:

$$\alpha(B) = 1 - \alpha(1)B^1 - \alpha(2)B^2 \dots - \alpha(p)B^p, \quad (3.27)$$

$$\beta(B) = 1 + \beta(1)B^1 + \beta(2)B^2 \dots + \beta(q)B^q,$$

и B - так называемый оператор сдвига, который определяется следующим образом:

$$B^j \varepsilon(i) = \varepsilon(i - j). \quad (3.28)$$

Автокорреляционная функция ARMA(p,q) – процесса убывает экспоненциально (см (1.16)), поэтому такая модель удачно описывает процессы с быстро убывающей зависимостью. Тем не менее, используем ARMA(p,q)-предсказатель для прогнозирования самоподобного трафика (обладающего медленно убывающей зависимостью) и сравним его эффективность с остальными.

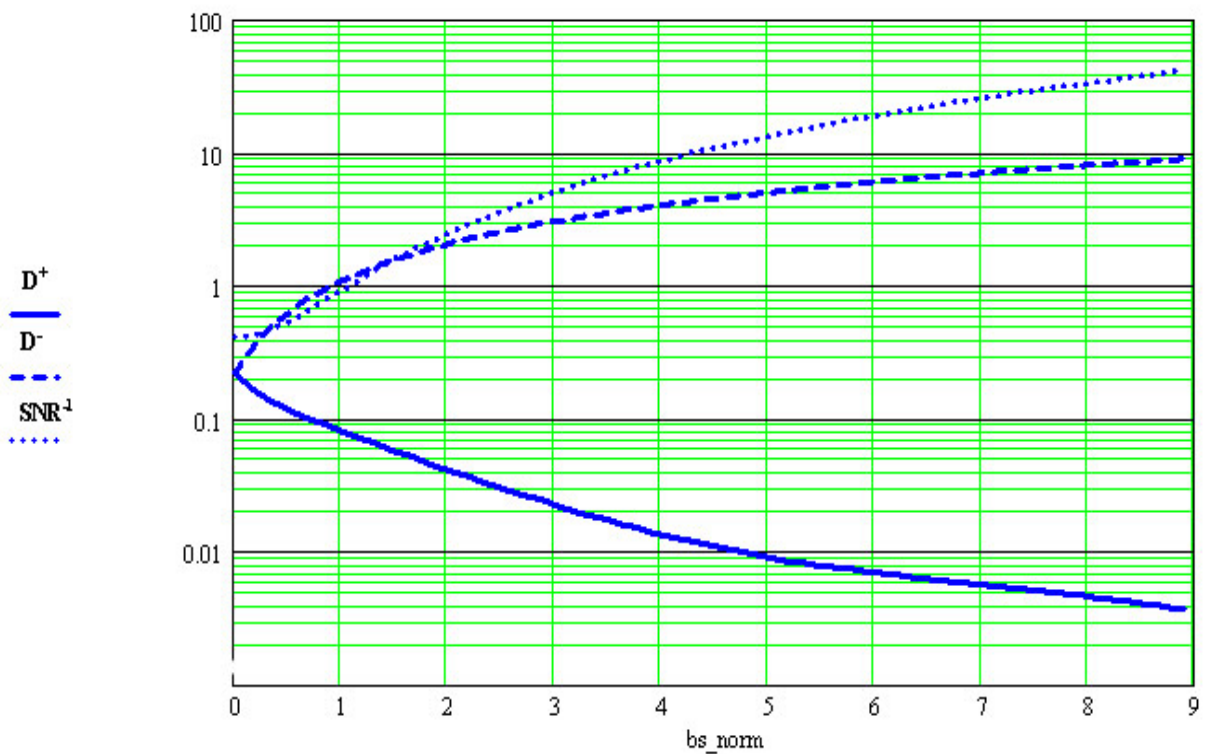


Рис. 3.8. Оценки статистических характеристик при динамическом задании пропускной способности канала с ARMA(1,1)-предсказателем

На рис. 3.8 и 3.9 представлены зависимости для основных изучаемых статистик в случае прогнозирования самоподобного трафика с помощью предсказателей ARMA(1,1) и ARMA(1,2) соответственно.

На основании сравнения приведенных графиков можно сделать вывод о том, что увеличение порядка ARMA-предсказателя фактически не сказывается на качестве прогнозирования самоподобного трафика. Аналогичные характеристики получаются для других параметров p и q модели ARMA.

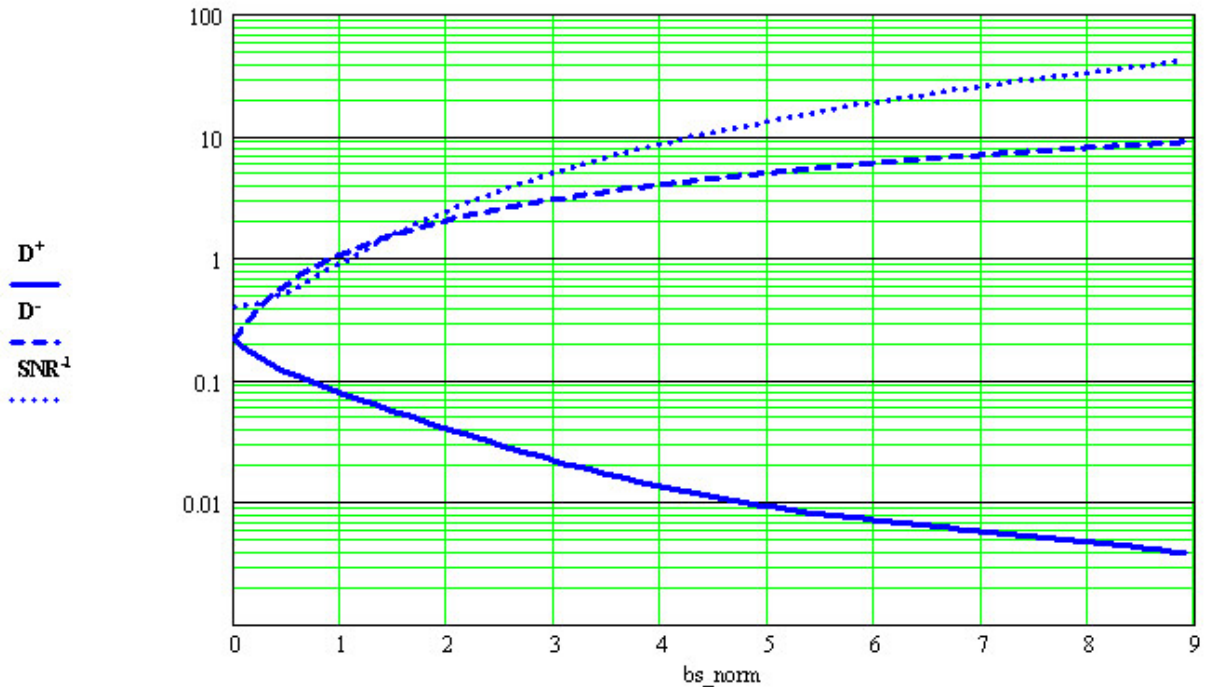


Рис. 3.9. Оценки статистических характеристик при динамическом задании пропускной способности канала с ARMA(1,2)-предсказателем

3.3.6 Динамическое распределение пропускной способности с FARIMA-предсказателем

Использование FARIMA (p,d,q) -процессов для моделирования и прогнозирования самоподобного трафика имеет особую актуальность, поскольку FARIMA (p,d,q) -процесс обладает медленно-убывающей АКФ при $0 < d < 0.5$, т.е. такой процесс обладает продолжительной памятью, как и самоподобный трафик [59].

Определение. Пусть $\varepsilon(k)$ - стационарный процесс, такой что

$$\alpha(B)(1 - B)^d \varepsilon(i) = \beta(B)\delta(i), \quad (3.29)$$

для некоторого $d \in (-0.5, 0.5)$. Тогда $\varepsilon(k)$ является FARIMA (p,d,q) -процессом.

Применение преимуществ FARIMA-моделей для прогнозирования сетевого трафика основывается на соотношении

$$\hat{C}(i) = \hat{x}(i) + bs = \hat{\chi}(i) + \hat{\varepsilon}(i) + bs = M[T(i)] + \hat{\varepsilon}(i) + bs, \quad (3.30)$$

где $\hat{\varepsilon}(k)$ вычисляется с помощью FARIMA (p,d,q)-предсказателя.

На рис. 3.10 представлена зависимость для основных изучаемых статистик в случае прогнозирования самоподобного трафика с помощью предсказателей FARIMA(1,d,1). При этом на рис. 3.11 показан процесс изменения оцениваемых значений параметра d этой модели.

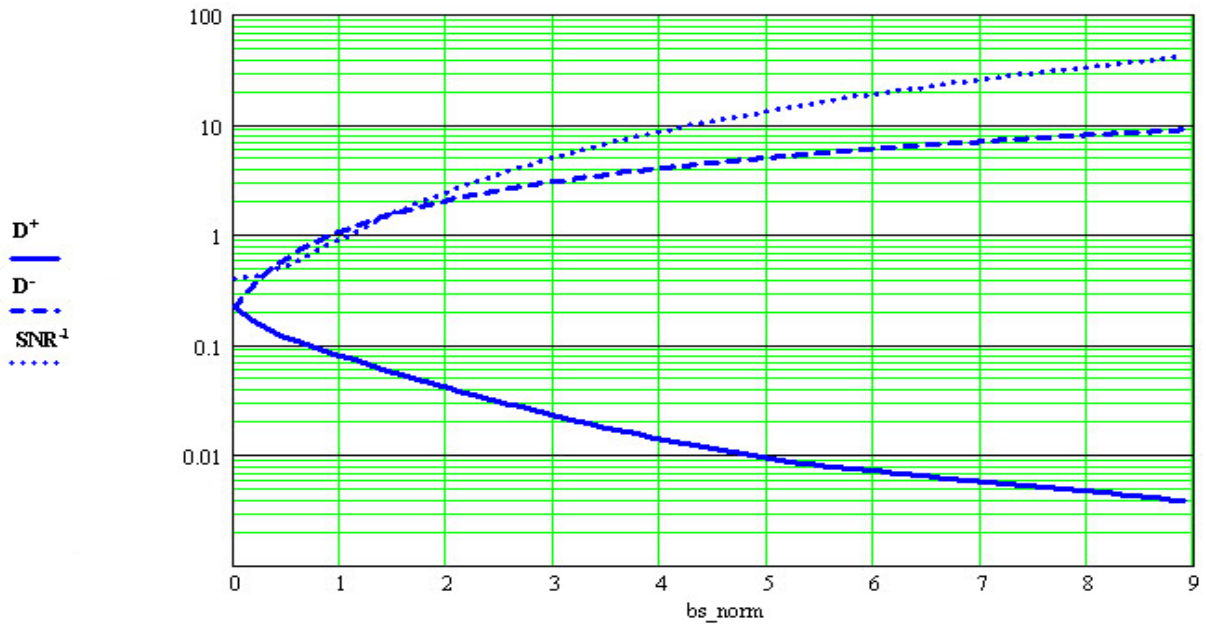


Рис. 3.10. Оценки статистических характеристик при динамическом задании пропускной способности канала с FARIMA(1,d, 1)-предсказателем

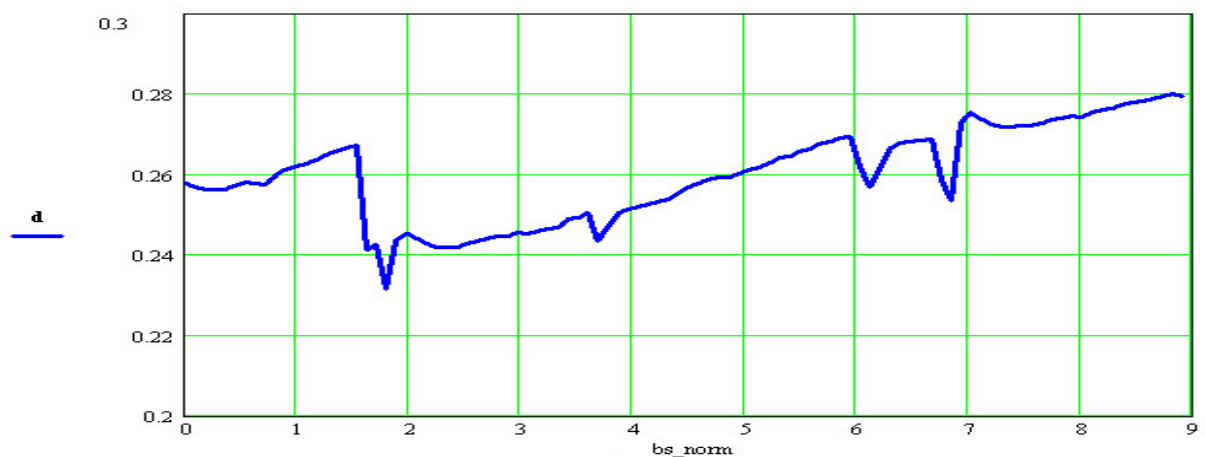


Рис. 3.11. График изменения параметра d в процессе прогнозирования самоподобного трафика FARIMA(1,d, 1)-предсказателем

Заметим, что параметр d связан с показателем Хэрста соотношением

$$d = H - 0.5. \quad (3.31)$$

Поскольку все оцененные значения параметра d находятся в интервале $0 < d < 0.5$, то можно утверждать, что прогнозирование велось с помощью модели, обладающей продолжительной памятью. Однако, анализируя график 3.10, отметим, что заметное улучшение характеристик системы по сравнению со случаями AR или ARMA-моделей отсутствует. Заметим, что также как и в случае ARMA и AR моделей, изменение порядка FARIMA-предсказателя (параметры p и q) практически не улучшает оцениваемые статистические характеристики SNR^{-1} , D^+ и D^- .

Листинг разработанной программы, которая реализует используемые методы прогнозирования (в пакете Ox) представлен в приложении 1.

3.4 Сравнение алгоритмов динамического распределения пропускной способности и выбор метода прогнозирования

Для того чтобы оценить выигрыш от использования динамического распределения пропускной способности по сравнению со статическим способом, а также для выбора наиболее подходящего способа прогнозирования рассмотрим оценки улучшения качества обслуживания:

$$AdvD_{forecast}^+(i) = D_{st}^+(i) - D_{forecast}^+(i), \quad (3.41)$$

$$AdvD_{forecast}^-(i) = D_{st}^-(i) - D_{forecast}^-(i), \quad (3.42)$$

$$AdvSNR_{forecast}^{-1}(i) = SNR_{st}^{-1}(i) - SNR_{forecast}^{-1}(i), \quad (3.43)$$

которые фактически представляют собой разности между соответствующими оценками для статического метода распределения и выбранного динамического метода изменения пропускной способности. При этом, чем выше значения оценок (3.41) или (3.42) или (3.43), тем выше значение выигрыша в качестве обслуживания для выбранного метода динамического распределения по отношению к методу статического задания пропускной способности.

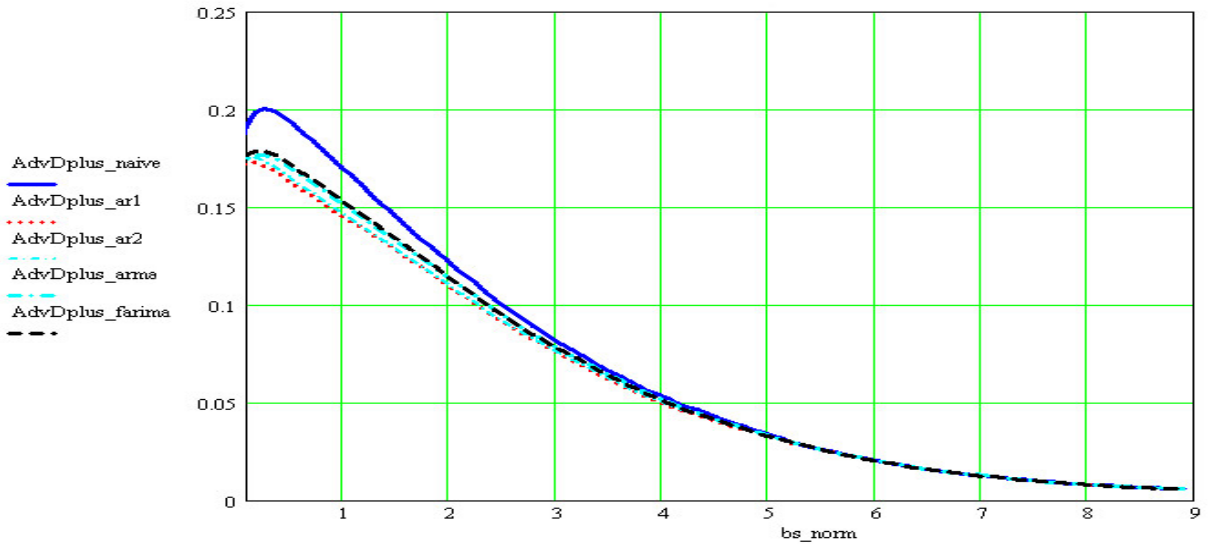


Рис. 3.12. Выигрыш в ошибке недооценки D^+ при динамическом задании пропускной способности с различными алгоритмами прогнозирования

Заметим, что выигрыши в коэффициентах недооценки (3.41) и переоценки (3.42) для рассматриваемых методов прогнозирования фактически совпадают, поэтому далее будем изучать лишь одно из соотношений, скажем (3.41). Результаты расчетов оценок по соотношениям (3.41) и (3.43) для различных методов прогнозирования ряда BC-10 приведены на рис. 3.12, и рис. 3.13.

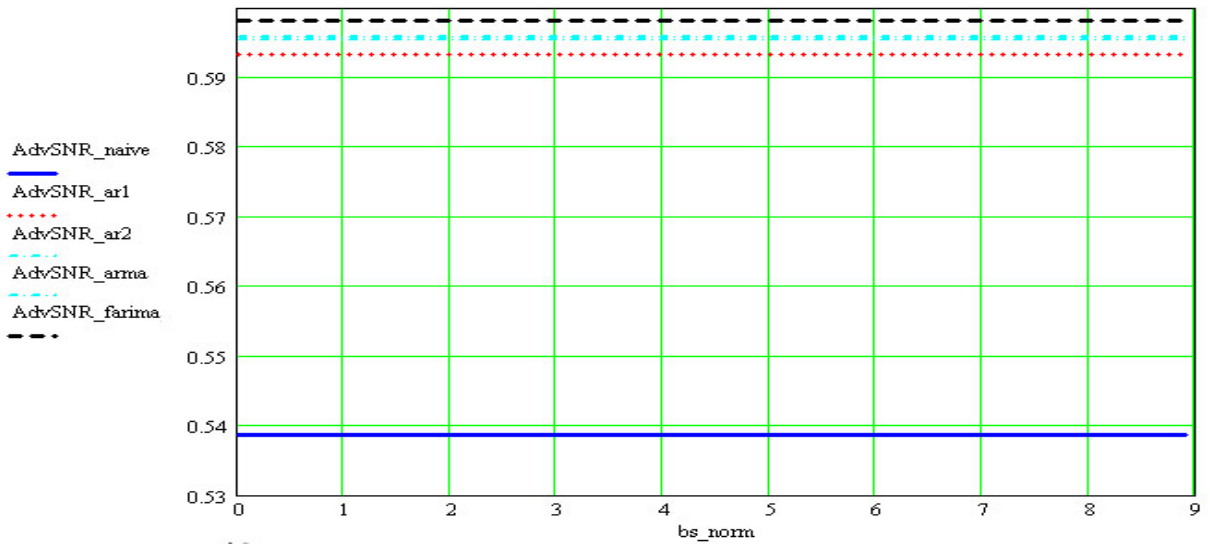


Рис. 3.13. Выигрыш в коэффициенте SNR^{-1} при динамическом задании пропускной способности с различными алгоритмами прогнозирования

Изучая приведенные графики можно отметить, что все оценки выигрыша (3.41), (3.42) и (3.43) больше нуля, что означает преимущества алгоритмов

динамического распределения полосы по сравнению со статическим способом. Обратим внимание на тот факт, что при увеличении bs выигрыш в использовании динамического способа уменьшается и стремится к нулю.

С точки зрения улучшения коэффициентов D^+ и D^- лучшие характеристики получаются при прогнозировании с помощью простого предсказателя. Например, судя по графику рис. 3.4 при статическом способе задания полосы в точке $bs_norm=1$ коэффициент недооценки $D_{st}^+ \sim 0.2433$. Это означает потерю 24.33% всей информации, переносимой трафиком $x(k)$. В то же время с помощью простого предсказателя (рис. 3.5) (при том же самом среднем значении полосы пропускания \hat{C} , то есть при $bs_norm=1$) потери недооценки удастся уменьшить до 6.4% ($D_{naive}^+ \sim 0.064$). Таким образом, выигрыш в D^+ и D^- при использовании простого предсказателя (в точке $bs_norm=1$) составил $\sim 18\%$ от всего объема информации (см. рис. 3.12). В то же время выигрыш в показателе SNR^{-1} для простого предсказателя соответствует $\sim 54\%$!

Оценим некоторые количественные показатели для конкретного случая. Как правило, для обеспечения удовлетворительного функционирования системы видеоконференцсвязи, например, считается, что величина потерь не должна превышать 1-2 % от всего объема транслируемой информации. Судя по графику на рис. 3.5, для алгоритма с простым предсказателем с целью достижения такой величины потерь достаточно ограничить пропускную способность канала уровнем, соответствующем значению $3 < bs_norm < 4.5$. Другими словами, пропускная способность канала должна быть равна среднему значению трафика (на изучаемом участке, т.е. на участке F), умноженному на коэффициент 3.. 4.5. В то же время, для достижения такого же эффекта при использовании статического задания пропускной способности, значение данного коэффициента возрастает значительно и соответствует 7..9 (см. рис. 3.4). Очевидно, данное обстоятельство увеличивает стоимость канала примерно в 2 раза.

Переходя далее к изучению рис. 3.12, можно сделать вывод о том, что выигрыш в статистических характеристиках D^+ и D^- при использовании алгоритма динамического распределения пропускной способности для рассмотренного случая видеоконференцсвязи (при $3 < bs_norm < 4.5$) составит от 4% до 8%.

Следует заметить, что видеоконференцсвязь является наиболее требовательным к величине потерь сервисом. В общем случае, при организации менее требовательных сервисов, величина выигрыша может быть больше, ~10% (см. рис. 3.12).

Выигрыш в статистике SNR^{-1} (которая, как будет показано ниже, отвечает за джиттер) составляет ~ 54% (см. рис.3.13). С другой стороны, для более сложных моделей эта величина несколько больше и составляет ~ 59%.

Вместе с тем с точки зрения улучшения соотношения SNR^{-1} , наоборот, лучшие показатели предоставляет (в порядке убывания выигрыша): FARIMA(p,d,q), ARMA(p,q), AR(2), AR(1) и простой предсказатели.

На первый взгляд может показаться странным, что усложнение предсказателя, т.е. переход от простого предсказателя к более сложным - ARMA и FARIMA-предсказателям, которые теоретически должны обеспечивать лучшее качество прогнозирования, на самом деле ухудшает характеристики потерь D^+ и использования D^- ресурсов канала. Однако вспомним, что при оценке параметров выбранной модели по исследуемой реализации процесса, как правило, стремятся уменьшить дисперсию соответствующей ошибки. Для этого случая существуют алгоритмы и разработаны методики эффективных, нересурсоемких процедур по оценки параметров таких моделей. В тоже время, критерий минимизации дисперсии ошибки не гарантирует минимизацию модуля ошибки, от которого зависят характеристики D^+ и D^- . Таким образом, переход от простого предсказателя к более сложному FARIMA-предсказателю улучшает прогноз в смысле уменьшения дисперсии ошибки, и, соответственно, в смысле статистики SNR^{-1} ,

которая с ней связана. Однако при этом ухудшаются характеристики потерь D^+ и степени использования канала D^- . Указанное обстоятельство хорошо отражено на графиках рис. 3.12 и рис.3.13. Следует заметить, что для решения данной проблемы необходимо при оценке параметров авторегрессионных моделей пользоваться, например, критерием минимизации модуля ошибки. Однако, в этом случае значительно усложняются алгоритмы и методики оценки параметров моделей, которые ко всему прочему редко встречаются в литературе (не смотря на долгие поиски, автору не удалось найти функционирующего алгоритма оценки параметров сложных авторегрессионных моделей с минимизацией модуля ошибки).

В данном случае, для окончательного выбора наиболее приемлемого предсказателя необходимо определиться в приоритетах и решить что лучше: иметь минимальный коэффициент SNR^{-1} , но сложную модель и не лучшие характеристики D^+ и D^- или иметь простую модель, лучшие характеристики D^+ и D^- , но при этом несколько проиграть в SNR^{-1} ? Вспомним, что статистики D^+ и D^- напрямую связаны с процентом потерь и степенью утилизации в канале. Показатель SNR^{-1} , в свою очередь, характеризует меру разброса ошибки прогнозирования относительно ее среднего значения. Чем меньше значение данного коэффициента, тем меньше изменяется величина ошибки, т.е. ошибка стабилизируется и стремится к своему среднему значению. Напротив, если ошибка изменяется значительно (случай большой дисперсии), то в процессе функционирования алгоритма, количество потерянной информации будет также сильно меняться во времени, что в свою очередь будет выглядеть как непостоянство качества услуги и, возможно, большой джиттер.

Заметим, однако, что даже значительное усложнение модели дает, тем не менее, достаточно небольшое улучшение в характеристике SNR^{-1} (а именно, выигрыш при этом увеличивается с 54% до 59%) на фоне также незначительного ухудшения характеристик D^+ и D^- (см. рис.3.12). Поэтому, с точки зрения простоты реализации, меньшей ресурсоемкости и

требовательности предсказателя, а также лучших показателях потерь и использования ресурсов канала (D^+ и D^-) для применения в алгоритме обеспечения качества обслуживания в системах распределения информации выберем простой предсказатель.

3.5 Выводы по главе 3

1. В настоящей главе нами изложены теоретические обоснования прогнозируемости самоподобных процессов, обладающих медленно убывающей АКФ. Вычислен интервал корреляции, а также указано, какие свойства самоподобного телетрафика могут влиять на его прогнозируемость.
2. Поставлена задача прогнозирования сетевого трафика в составе задачи динамического (адаптивного) распределения пропускной способности канала. Определен алгоритм проверки прогнозируемости и оценки качества прогноза. Наряду с классической оценкой сигнал/шум вводятся оценки: коэффициент потерь и коэффициент недоиспользования. Последние, по нашему мнению, применительно к изучаемой задаче имеют определенный физический смысл и позволяют адекватно оценить выигрыш от применения алгоритма динамического управления по сравнению со статическим заданием пропускной способности канала.
3. С помощью статистических исследований оценены характеристики потерь при обработке самоподобного трафика в системе со статическим заданием пропускной способности. При этом использовался трафик реальной сети.
4. Оценены характеристики потерь при обработке самоподобного сетевого трафика в системах с динамическим распределением пропускной способности, использующих различные алгоритмы прогнозирования: простой предсказатель, авторегрессионные первого и второго порядка, ARMA и FARIMA – предсказатели различных порядков.

5. На основании результатов сравнения оценок потерь в схемах динамического распределения пропускной способности (для различных алгоритмов предсказания) с аналогичными показателями, полученными для статического способа задания пропускной способности, а также сложности реализации той или иной схемы сделан выбор наиболее подходящего – простого предсказателя.

6. Предполагается, что применение простого предсказателя в алгоритме динамического управления пропускной способностью обеспечит уменьшение потерь и увеличение использования ресурсов системы примерно на 10%, а также улучшение показателя SNR^{-1} , влияющего на величину джиттера, примерно на 54%.

Глава 4. Метод обеспечения качества обслуживания в условиях влияния эффекта самоподобия

Цель главы: разработка алгоритма обеспечения качества обслуживания, предназначенного для работы в системах распределения информации, которые обрабатывают самоподобный трафик.

4.1 Принцип динамического управления пропускной способностью

Самоподобный телетрафик, как было показано выше, имеет отчетливую структуру VBR (variable bit rate) типа. То есть интенсивность передачи информации значительно изменяется во времени, в результате чего трафик выглядит бёрстным (имеет высокую пачечность). Как следствие, статическое задание пропускной способности канала (алгоритмы типа шейпинга и полисинга), предназначенного для данного вида трафика, приводит к большим потерям информации и низкой утилизации канала. Первое обстоятельство ухудшает характеристики качества обслуживания, а второе сказывается на неэффективности использования ресурсов канала.

Для улучшения характеристик обработки самоподобного трафика в настоящей работе предлагается использовать алгоритм динамического (адаптивного) управления пропускной способностью канала, основанный на прогнозировании самоподобного сетевого трафика. Принцип динамического распределения пропускной способности не является новым. В частности, о его возможностях относительно VBR трафика АТМ сетей (безотносительно к самоподобию) сказано в работах [52], [56], [60], [61]. Также существует некоторое количество работ, касающихся проблемы прогнозирования самоподобного телетрафика [51], [59], [67]. Однако до настоящего времени практически не существовало исследований, посвященных изучению возможностей адаптивного распределения пропускной способности канала на

основе прогнозирования применительно к самоподобному телетрафику. Данная глава посвящена этой проблеме.

Структурная схема системы с динамическим распределением пропускной способности канала представлена на рис. 4.1. С помощью предсказателя в данной схеме пропускная способность канала на выходе сетевого устройства устанавливается в соответствии с интенсивностью входного потока, отслеживая профиль трафика. Чем совершеннее предсказатель, тем выше эффективность схемы.

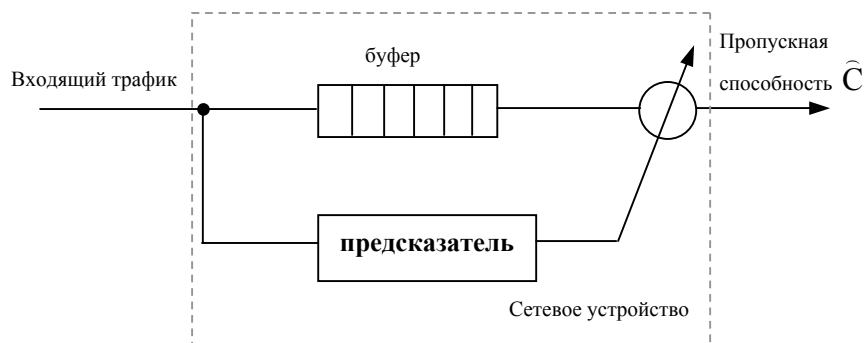


Рис. 4.1. Структурная схема системы с алгоритмом динамического распределения пропускной способности канала на основе прогнозирования

Обратим внимание, что существует вариант адаптивного управления пропускной способностью, не использующий прогнозирования. Структурная схема такой системы показана на рис. 4.2. В данной схеме входящий трафик искусственно задерживается на необходимое время в звене задержки.

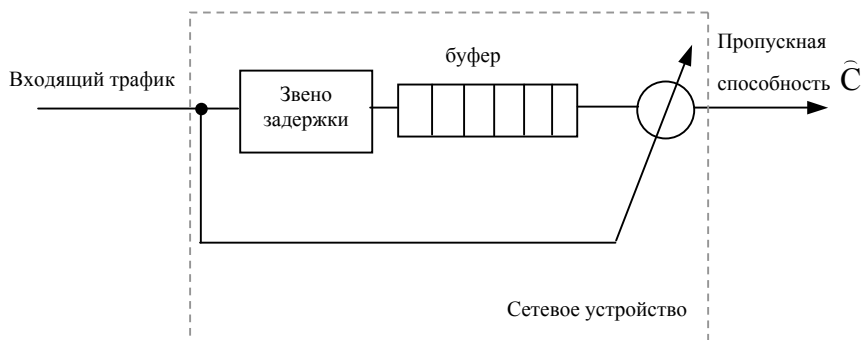


Рис. 4.2. Структурная схема системы с алгоритмом динамического распределения пропускной способности канала без прогнозирования

Время задержки определяется инерционностью механизмов регулирования пропускной способности. В результате удается достаточно точно оценить требуемую пропускную способность канала на выходе устройства. Однако данная схема имеет существенный недостаток – вносимые задержки. Более того, задержки будут вноситься на каждом транзитном сетевом узле, реализующем данный алгоритм. Это делает ее малопривлекательной с точки зрения систем передачи информации реального времени. Потому в настоящей работе рассматривается только схема с прогнозированием. В качестве возможного варианта применения схемы с прогнозированием рассмотрим простой пример. Имеется физический канал с потенциально достижимой пропускной способностью C_{Σ} . В этом физическом канале организовано два логических (виртуальных) канала VC (virtual channels) со статически заданными пропускными способностями C_1 и C_2 соответственно, причем $C_{\Sigma} = C_1 + C_2$ (см. рис. 4.3). В первом из них передается информация реального времени, чувствительная к задержкам и потерям (для системы видеоконференцсвязи, например), а во втором – второстепенная информация (данные www, ftp и пр. информация, не чувствительная к задержкам и потерям).

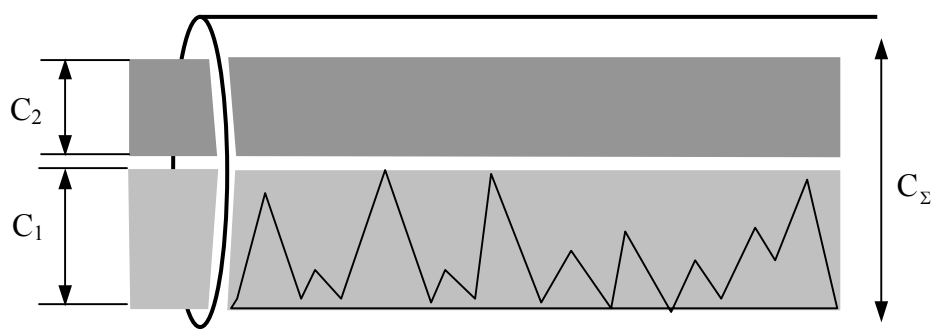


Рис. 4.3. Статическое распределение пропускной способности физического канала между двумя логическими

Возникает задача наиболее эффективного способа разделения общей пропускной способности C_{Σ} физического канала между двумя логическими. Поскольку в первом виртуальном канале передается критичная к потерям информация, необходимо увеличивать пропускную способность C_1 этого

канала. Однако при этом уменьшается его утилизация и доступная пропускная способность для второго канала, поскольку $C_2 = C_\Sigma - C_1$. В результате ресурсы физического канала расходуются неэффективно. Более того, чем выше пачечность трафика в первом канале, тем ниже эффективность системы.

Гораздо большие возможности в этом случае может предложить схема с прогнозированием. В этом случае с помощью предсказателя система заранее получает информацию о требованиях к пропускной способности канала с приоритетной информацией (первого канала) в некоторый момент времени t в ближайшем будущем.

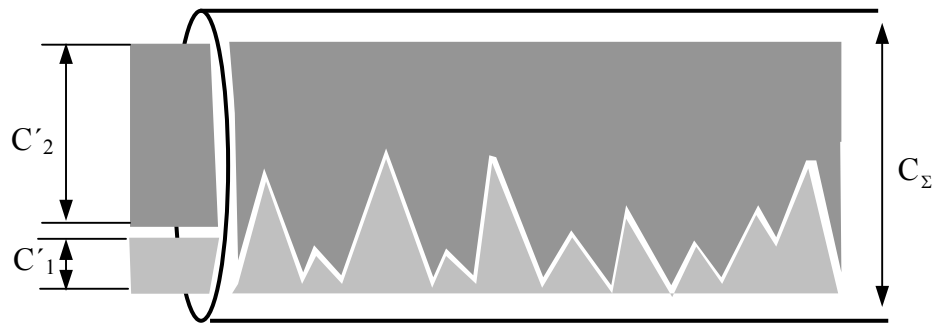


Рис. 4.4. Динамическое распределение пропускной способности физического канала между двумя логическими

Поэтому к наступлению момента времени t система выделяет для первого канала требуемый ресурс пропускной способности C'_1 , распределив при этом второму каналу всю оставшуюся доступную пропускную способность $C'_2 = C_\Sigma - C'_1$. Как можно видеть из рис.4.4 принцип динамического управления пропускной способностью в данном примере увеличивает утилизацию в первом канале и предоставляет больше ресурсов для второго канала.

4.2 Алгоритмы контроля и управления трафиком

Для осуществления функций контроля и управления интенсивностью трафика, а также обеспечения качества обслуживания (QoS) существуют специальные алгоритмы [57]. Как правило, они основываются на так называемом принципе “корзины маркеров” (Token Bucket) или его модификациях. Рассмотрим два режима функционирования этого алгоритма – полисинг (traffic-policing), при котором происходит сбрасывание неконформной нагрузки, а также шейпинг (traffic-shaping), буферизующий неконформные пакеты.

4.2.1 Алгоритм полисинга на основе механизма “корзина маркеров”

Алгоритм полисинга используется для измерения и управления интенсивностью трафика. При этом трафик приводится к заданному профилю с помощью отбрасывания неконформных пакетов. Профиль трафика в механизме “корзина маркеров” задается согласованным размером всплеска трафика V_c (бит) за определенный интервал времени T_c (сек). При этом интенсивность CIR (бит/с) генерирования маркеров¹ определяется как

$$CIR = V_c / T_c. \quad (4.1)$$

Пакеты трафика, удовлетворяющие заданному профилю, пропускаются на выход без какой либо задержки, а остальные – отбрасываются. Как показано на рис. 4.5, механизм Token Bucket включает в себя буфер для маркеров² (tokens) размером V_c .

Буфер пополняется маркерами с заданной интенсивностью, то есть V_c (бит) через каждый интервал времени T_c . Если буфер маркеров полон, но вновь генерируемый маркер теряется. Пакет размера L , поступающий на вход Token Bucket, транслируется на выход системы только в том случае, если в буфере

¹ Под термином “маркер” в данной работе понимается некоторая виртуальная величина, соответствующая объему информации 1 бит, пропускаемой на выход схемы “корзина маркеров”.

² Иногда термин token переводят на русский язык как “жетон”.

имеется L маркетров. При этом содержимое f буфера уменьшается на L . Иначе пакет отбрасывается. Алгоритм работы данного механизма представлен на рис. 4.6.

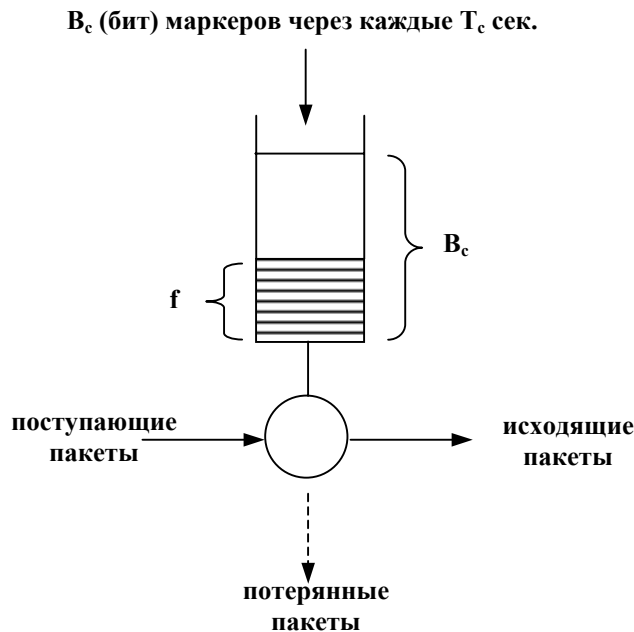


Рис. 4.5. Схема работы механизма “корзина маркетров” в режиме полисинга

Здесь слева показана логика работы классификатора пакетов на конформный и неконформные, а справа - алгоритм наполнения буфера маркетров. Следует заметить, что правая и левая части алгоритма выполняются одновременно. В связи с тем, что генерация маркетров происходит с постоянной интенсивностью CIR , можно утверждать, что минимальная интенсивность обработки поступающих пакетов равна интенсивности генерации маркетров. Превышение минимальной интенсивности обработки возможно в случае, когда в буфере накопилось некоторое количество маркетров.

На рис. 4.7 показаны график функционирования алгоритма “корзина маркетров” в режиме полисинга для некоторого неравномерного процесса поступлений пакетов. Обратим внимание, что на первом такте ($0 < t < T_c$) интенсивность I трафика превышает допустимую согласованную скорость CIR , поэтому еще задолго до конца интервала T_c доступные маркетры в буфере кончаются и поступающие вслед за этим пакеты считаются неконформными, вследствие чего отбрасываются. В момент времени T_c буфер маркетров вновь наполняется маркетрами в объеме V_c .

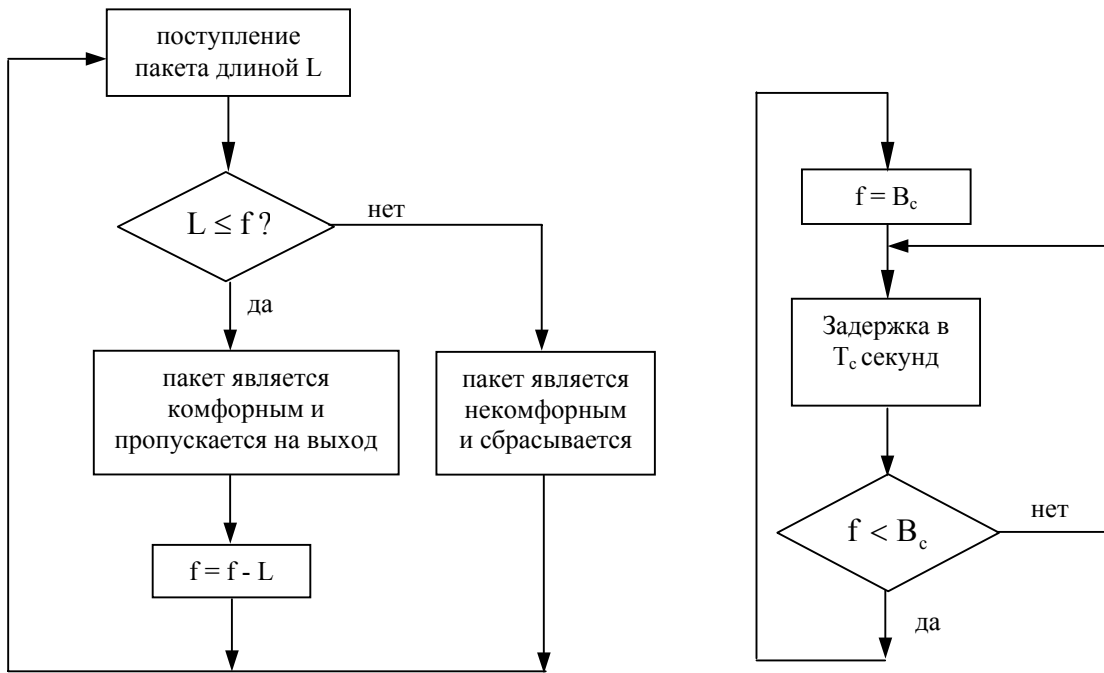


Рис. 4.6. Алгоритм работы механизма “корзина маркеров” в режиме полисинга

На втором такте ($T_c < t < 2T_c$) средняя интенсивность I трафика за интервал T_c меньше значения CIR . Поэтому в момент времени $2T_c$ маркеры полностью не израсходовались. Тем не менее, буфер дополняется маркерами до уровня B_c , после чего начинается следующий такт, и т.д.

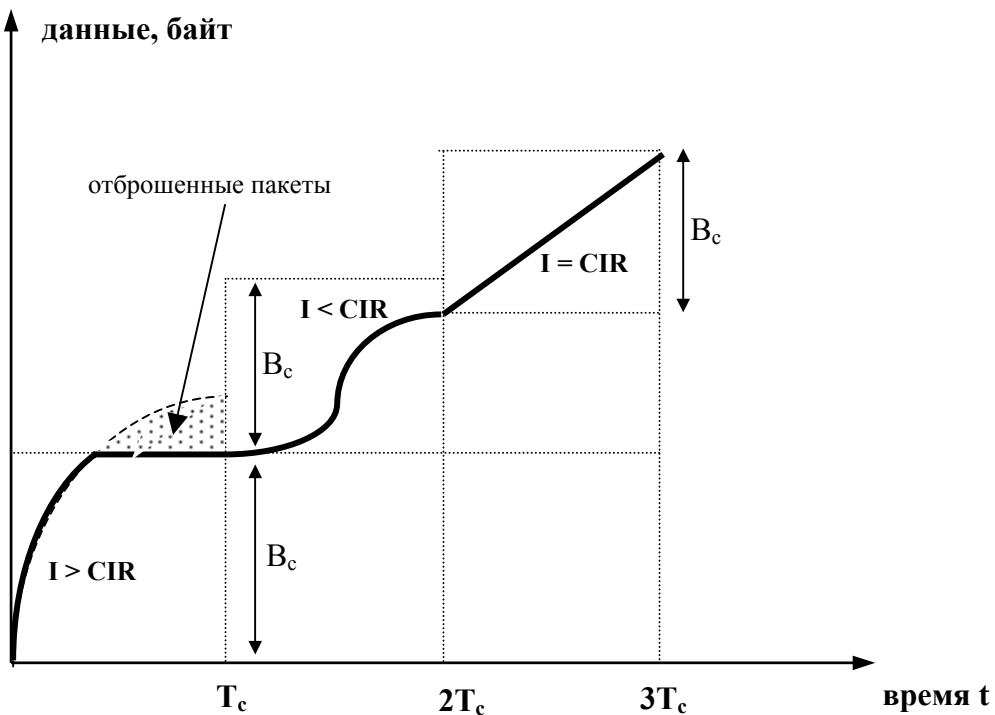


Рис. 4.7. График функционирования алгоритма “корзина маркеров” в режиме полисинга для неравномерного процесса поступлений пакетов

В результате на выходе данного алгоритма максимальный выброс трафика за время T_c не превышает значения V_c . При этом фактически реализуется схема статического задания пропускной способности, которая рассматривалась в главе 3. Действительно, значения сетевого трафика, приведенного к эквидистантной форме с помощью процедуры агрегирования по уровню ΔT , пропорциональны (с коэффициентом ΔT) значениям всплесков трафика за интервалы $T_c = \Delta T$. А значения статически задаваемой пропускной способности \hat{C} пропорциональны значению V_c .

4.2.2 Алгоритм шейпинга на основе механизма “корзина маркеров”

Еще одним популярным алгоритмом выравнивания трафика является алгоритм шейпинга (traffic-shaping). Он отличается от полисинга только наличием дополнительного буфера для входящего потока пакетов (см. рис. 4.8). При этом алгоритме пакеты, которые не соответствуют заданному профилю, не сбрасываются как в режиме полисинга, а буферизуются. Алгоритм данной схемы представлен на рис. 4.9.

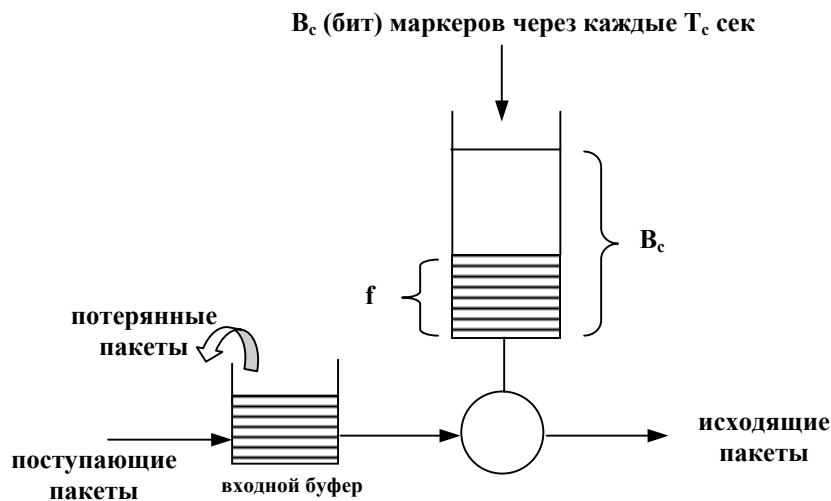


Рис. 4.8. Схема работы механизма “корзина маркеров” в режиме шейпинга

Буферизация неконформных пакетов приводит к сглаживанию профиля трафика на выходе такой схемы, что позволяет несколько уменьшить потери при его дальнейшей обработке.

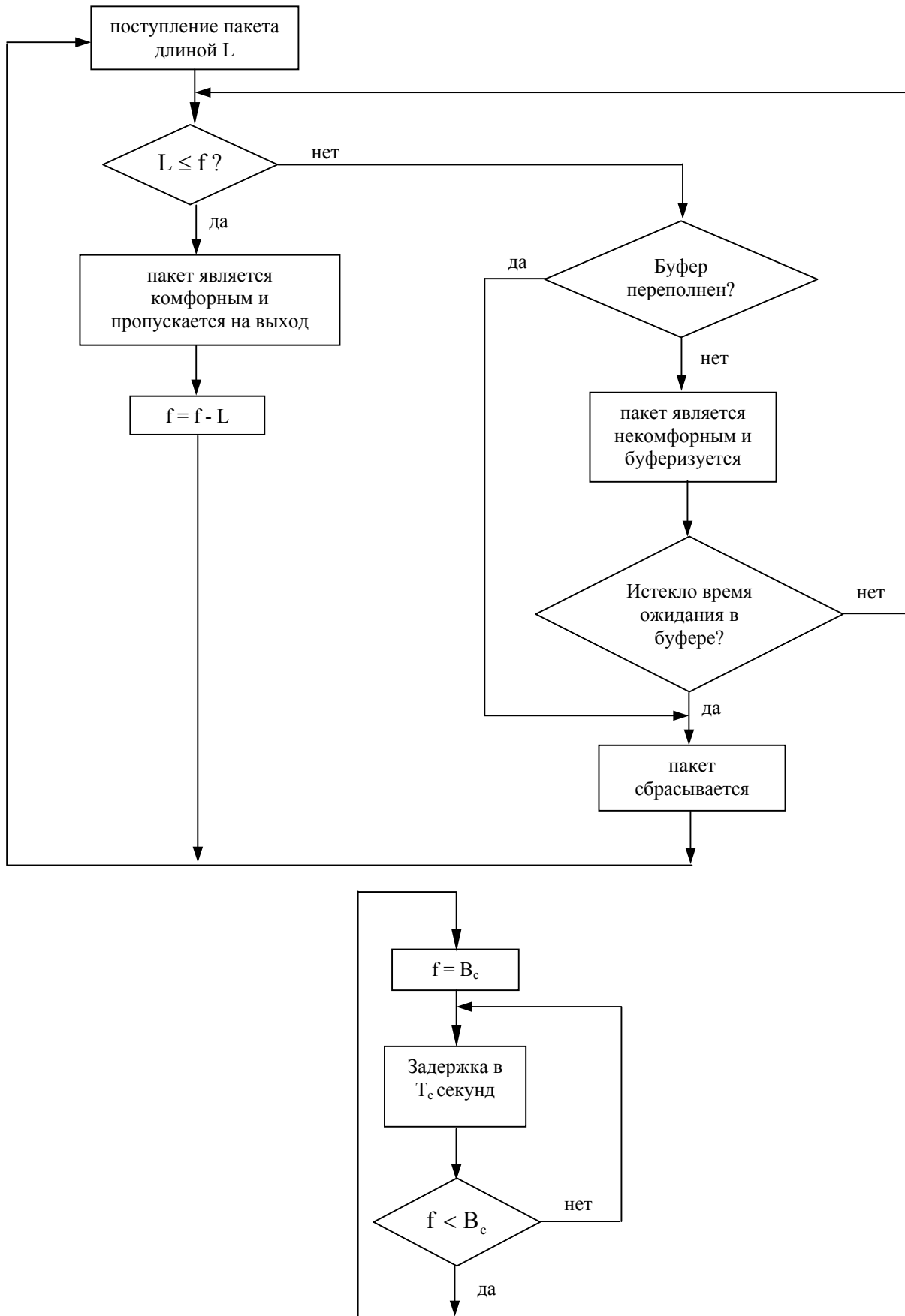


Рис. 4.9. Алгоритм механизма “корзина маркеров” в режиме шейпинга

Благодаря наличию буфера для пакетов данная схема наряду с уменьшением показателя потерь вносит задержки, что в некоторых случаях ограничивает ее применение для систем обработки информации реального времени. Если буфер заполнен полностью, то вновь прибывающие пакеты сбрасываются до тех пор, пока (в результате обработки накопленных пакетов) в буфере не станет достаточно места для их хранения.

Таким образом, процедуру шейпинга также можно классифицировать как метод статического задания пропускной способности, но несколько более гибкий, чем полисинг, поскольку он позволяет не сразу отбрасывать неконформные пакеты, а буферизует их и обрабатывает при первой возможности.

Отметим, что алгоритм шейпинга может быть эффективным для VBR-трафика с относительно невысоким уровнем пачечности. В случае берстного самоподобного трафика профиль трафика будет выравниваться за счет значительных вносимых задержек, которые, скорее всего, будут неприемлемы для большинства систем обработки информации реального времени.

4.3 Разработка метода управления трафиком для работы в условиях влияния эффекта самоподобия

В данном разделе предлагается новая разработка – алгоритм обеспечения качества обслуживания при влиянии эффекта самоподобия. Идея данного алгоритма состоит в модификации рассмотренных выше механизмов управления трафиком (полисинга и шейпинга, служащих для реализации функций обеспечения качества обслуживания (QoS)) с помощью введения дополнительного модуля, прогнозирующего на некоторое время вперед необходимое значение V_c в соответствии с изменением интенсивности поступающих пакетов. При этом не трафик выравнивается под заданный наперед профиль, а напротив, пропускная способность системы подстраивается

под профиль трафика, уменьшая при этом потери и увеличивая использование выделенных ресурсов.

Вариант функционирования алгоритма динамического управления пропускной способностью с прогнозированием, основанный на механизме “корзина маркеров”, изображен на рис. 4.10.

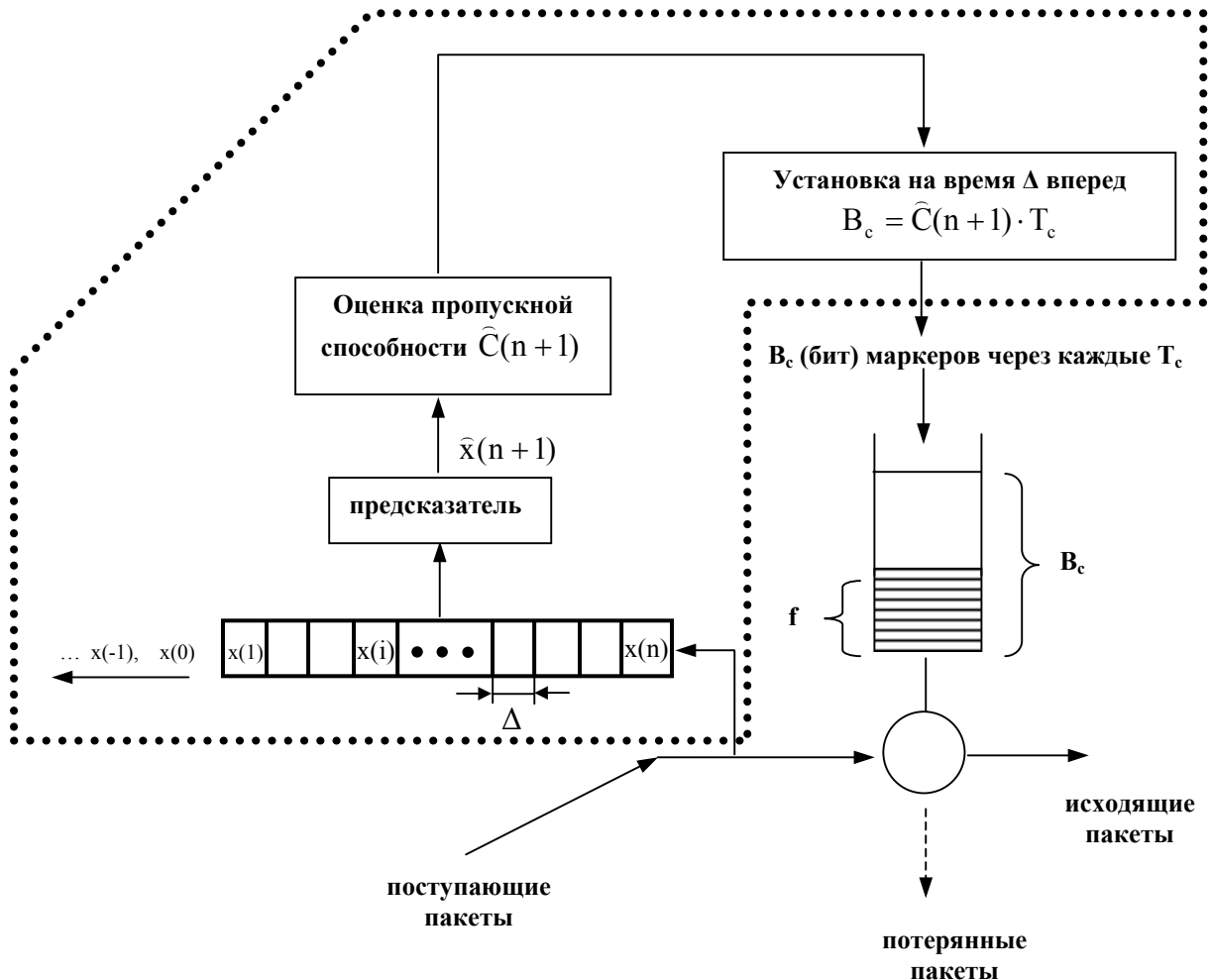


Рис. 4.10. Алгоритм механизма “корзина маркеров” в режиме динамического управления пропускной способностью с прогнозированием

Согласно данной схеме входящий поток пакетов одновременно с поступлением в систему попадает на схему агрегирования по временным интервалам Δ . После чего n последних отсчетов получившегося временного ряда поступают на предсказатель, который выдает прогностическую оценку следующего отсчета $\hat{x}(n+1)$ агрегированного ряда x на время Δ вперед. Исходя из требований к величине возможных потерь оценивается пропускная способность системы $\hat{C}(n+1)$. При этом, как показано на рис. 4.10, с помощью

обратной связи может учитываться статистика потерянных пакетов. В результате размер буфера жетонов устанавливается $V_c = \hat{C}(n + 1) \cdot T_c$ на время Δ вперед.

Таким образом, значение V_c (а вместе с ним и CIR) будет меняться каждый интервал Δ , отслеживая динамику изменения интенсивности трафика. Как было показано в главе 3, при той же самой (в среднем) пропускной способности системы, в режиме динамического управления пропускной способностью с прогнозированием удается достичь лучших показателей потерь и использования чем в случае реализации классического полисинга. С этих позиций предложенный алгоритм можно классифицировать как способ обеспечения качества обслуживания (QoS).

Аналогичным образом, с помощью введения прогнозирующего модуля можно модифицировать алгоритм шейпинга. При этом отброс пакетов из-за несовершенства предсказателя будет снижен за счет увеличения задержек. Алгоритм функционирования такой схемы фактически будет соответствовать приведенной на рис. 4.10, в которую дополнительно введен буфер для поступающих в систему пакетов.

4.4 Моделирование механизма динамического управления пропускной способностью канала с использованием прогнозирования в среде ns-2

Проверка пудинга состоит в том, что его съедают

Для проверки полученных в настоящей диссертационной работе результатов с помощью имитационного моделирования на ПЭВМ был поставлен эксперимент по анализу эффективности алгоритма динамического управления пропускной способностью с прогнозированием в условиях самоподобного телетрафика. Моделирование производилось в среде популярного сетевого эмулятора ns-2. Схема сценария представлена на рис. 4.11.

Источником самоподобного трафика в данном эксперименте является одна из реализаций реального сетевого трафика, изучаемая в главах 2 и 3 настоящей диссертации и подаваемая на узел N5. Посредством узла N5 полученный таким образом самоподобный поток упаковывается в UDP-пакеты и передается в сторону получателя N4. Кроме того, на схеме также имеется еще один (вспомогательный) источник трафика N0, генерирующий и транслирующий поток UDP-пакетов в направлении получателя N4.

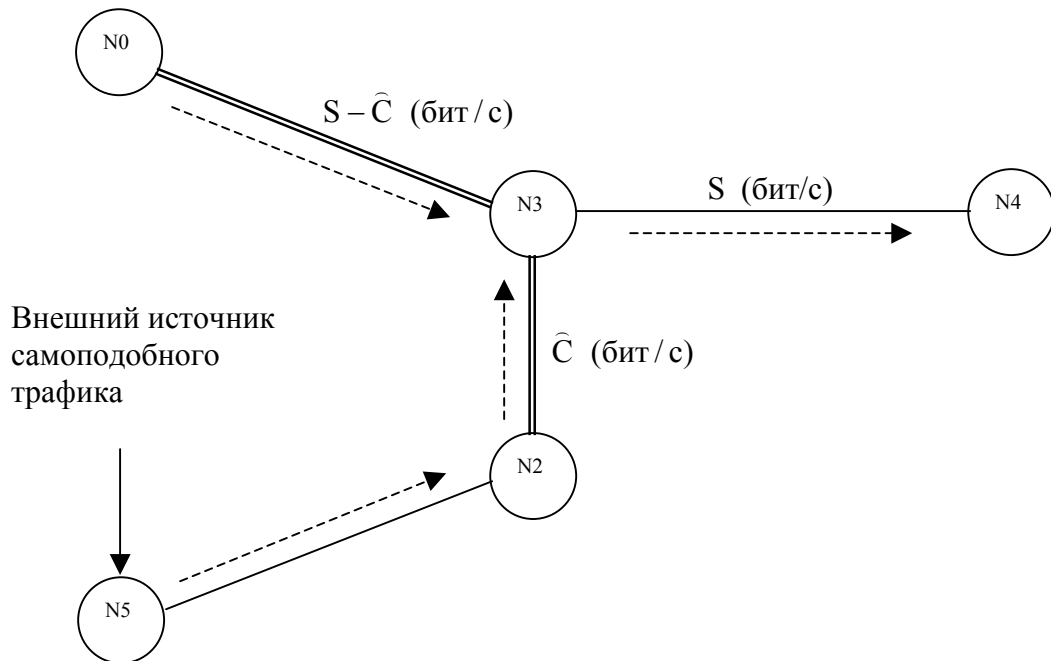


Рис. 4.11. Схема эксперимента по моделированию статического и динамического режимов управления пропускной способностью канала

В целях данного эксперимента для оценки потенциальных возможностей алгоритма выберем источник с постоянной интенсивностью (S бит/с) генерирования пакетов (так называемый, CBR-источник).

Трафики обоих источников (самоподобного и CBR) имеют в составе своего пути к получателю N4 один общий участок, одновременно являющийся “узким” местом сети – канал N3-N4 с пропускной способностью S бит/с.

В данном случае возникает задача эффективного разделения ресурсов канала N3-N4 (его пропускной способности) между трафиками обоих источников N5 и N0.

В процессе проведения эксперимента система рис. 4.11 изучается в двух режимах:

- в режиме статического разделения пропускной способности канала N3-N4 между самоподобным и CBR трафиками. При этом некоторый ресурс этой пропускной способности $C < S$ (бит/с) закрепляется постоянно за самоподобным трафиком, а оставшаяся пропускная способность $S - C$ (бит/с) выделяется под трафик CBR. Однако, как уже отмечалось выше, такой способ малоэффективен вследствие высокой пачечности трафика источника N5;

- в режиме динамического перераспределения пропускной способности канала N3-N4 между самоподобным и CBR трафиками при помощи прогнозирования. В этом случае на основе информации, полученной в процессе мониторинга интенсивности самоподобного трафика в канале N5-N2 в момент времени i , делается прогноз потребностей потока источника N5 в пропускной способности \hat{C} (бит/с) на участке N3-N4 в последующий интервал времени $i + \Delta$. На основании такой прогностической оценки самоподобному трафику в канале N3-N4 выделяется требуемый ресурс \hat{C} (бит/с), а CBR-трафику, соответственно оставшийся ресурс $S - \hat{C}$ (бит/с) пропускной способности на время Δ .

В обоих исследуемых режимах оценивались статистики D^+ , D^- , SNR^{-1} (см. главу 3), а также общее количество информации, переданное источником N0 и достигшее получателя N4. Для прогнозирования использовался алгоритм с простым предсказателем, как наименее ресурсоемкий и наиболее эффективный.

Моделирование, как уже было замечено, производилось в среде популярного сетевого эмулятора ns-2, который явился следствием работы многих научных организаций над проектом VINT (Virtual InterNetwork Testbed). Программное обеспечение ns-2 имеет открытый исходный код и распространяется бесплатно, без каких-либо ограничений на право использования, модификации и распространения. Ns-2 является объектно-ориентированным ПО, ядро которого реализовано на языке C++, а язык

сценариев (скриптов) Otcl используется в качестве интерпретатора. С описанием ПО ns-2 можно ознакомиться на официальном сайте проекта VINT [84].

В целях настоящей работы все необходимые программные модули реализованы на языке Otcl и использованием стандартных библиотек C++ для ns-2. Пакет программ ns-allinone-2.27 устанавливался на ПО Cygwin-2.427 для операционной системы Windows 2000. Всего использовалось три программных модуля: vega.tcl, nsrun.tcl и tp-sim.tcl. Поскольку данный эксперимент является оригинальным и, по нашему мнению, представляет определенный интерес для широкого круга исследователей (например, в части реализации его на ns-2), то в Приложении №2 приведены листинги разработанных программ, а также их краткое описание и методы работы с ними.

В результате работы программ мы получаем те же самые (или более полные) статистики, которые исследовали ранее в главе 3, и теперь можем их сравнить.

4.5 Анализ результатов моделирования механизма динамического управления пропускной способностью канала с использованием прогнозирования

Для того, чтобы проверить выводы, сделанные ранее в главе 3 об эффективности различных алгоритмов предсказания и механизме динамического распределения пропускной способности в целом, приведем и проанализируем результаты моделирования в среде ns-2 для той же реализации трафика, что использовалась нами ранее. После этого, для подтверждения общности суждений приведем аналогичные результаты моделирования и для других реализаций трафика.

В главе 3 была использована реализация самоподобного трафика VC-Oct89Ext.TL, приведенная к эквидистантному виду с помощью процедуры агрегирования по уровню $\Delta=10$ сек.

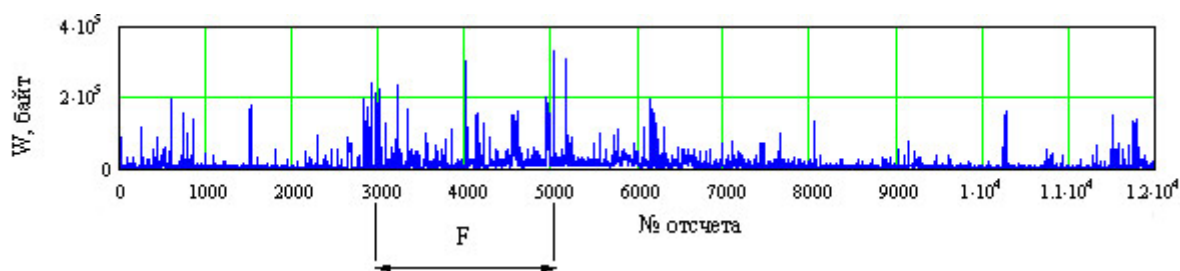


Рис. 4.12. Временной ряд ВС-10 и его исследуемый фрагмент F

При этом для изучения (прогнозирования) был выбран ее фрагмент F, заключенный между отсчетами 3000 ($3000 \cdot \Delta / 3600 = 8.3$ часа) и 6000 ($6000 \cdot \Delta / 3600 = 16.6$ часа), который характеризуется общей длительностью ~ 8.3 часа и, фактически, представляет собой реализацию трафика, в течение стандартного рабочего дня. Для настоящего эксперимента по моделированию (в целях экономии времени) возьмем более короткий участок между отсчетами 3000 и 5000, имеющий длительность 2000 отсчетов = 20000 сек (см. рис. 4.12).

Изменения, которые необходимо внести в программу `tp-sim.tcl` для проведения данного эксперимента и получения приведенных ниже результатов, отражены в табл. 4.1:

Табл. 4.1. Значения параметров в программе `tp-sim.tcl`

Название параметра	Значение параметра
<code>max_band2_3</code>	200000.0
<code>max_band0_3</code>	200000.0
<code>\$ns simplex-link \$n0 \$n3</code>	200000.0 1ms DropTail
<code>\$ns simplex-link \$n1 \$n2</code>	200000.0 1ms DropTail
<code>\$ns simplex-link \$n2 \$n3</code>	0.0 1ms DropTail
<code>\$ns simplex-link \$n3 \$n4</code>	200000.0 1ms DropTail
<code>\$ns simplex-link \$n5 \$n1</code>	200000.0 1ms DropTail
<code>\$cbr0 set rate_</code>	200000.0
<code>band3_4</code>	200000.0
<code>band0_3</code>	200000.0

При этом эксперимент начинается с введения в командной строке следующих инструкций:

ns nsrun.tcl 4 4 2 20000 10 0 0.9 - для статического метода;

ns nsrun.tcl 4 4 2 20000 10 1 0.9 - для динамического метода с простым предсказателем.

Для начала сравним статистики Dplus и Dplus_drops, отражающие коэффициент D^+ , но измеренные разными способами (Dplus оценивался так же, как в главе 3, а Dplus_drops – с помощью специального счетчика отброшенных пакетов, встроенного в ns-2). Поскольку нас больше интересует выигрыш от применения метода динамического управления пропускной способностью с прогнозированием, то сравним выигрыши для статистик Dplus и Dplus_drops, определяемых как разность:

$$\text{AdvDplus}_{\text{naive}}(\text{bs_norm}) = \text{Dplus}_{\text{st}}(\text{bs_norm}) - \text{Dplus}_{\text{naive}}(\text{bs_norm}) \quad (4.1)$$

для статистики Dplus, и как разность:

$$\begin{aligned} \text{AdvDplus_drops}_{\text{naive}}(\text{bs_norm}) &= \\ &= \text{Dplus_drops}_{\text{st}}(\text{bs_norm}) - \text{Dplus_drops}_{\text{naive}}(\text{bs_norm}) \end{aligned} \quad (4.2)$$

для статистики Dplus_drops. Соответствующие графики приведены на рис. 4.13.

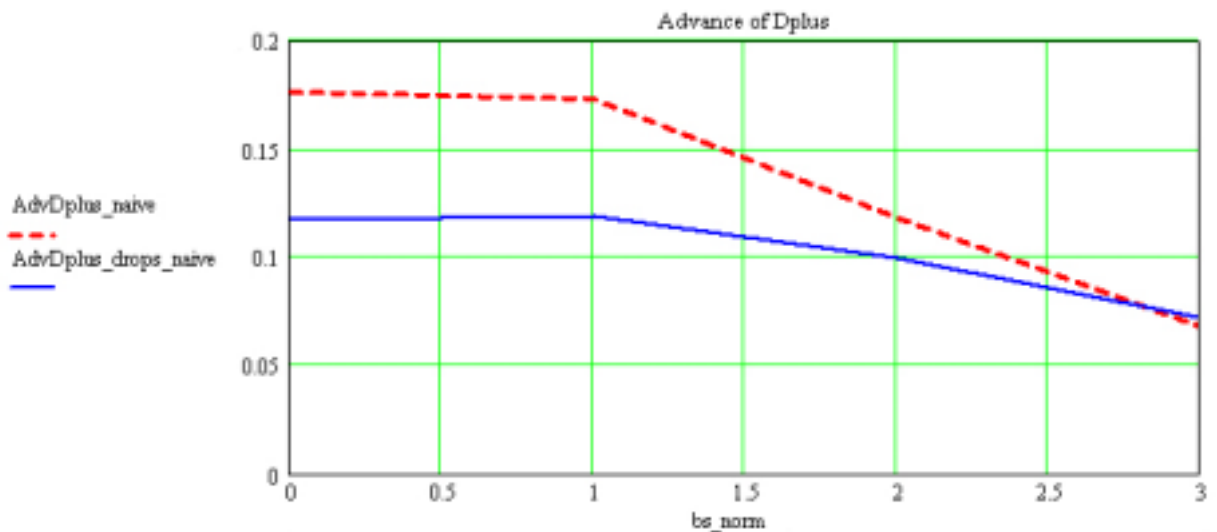


Рис. 4.13. Выигрыши в статистиках Dplus и Dplus_drops от применения метода динамического распределения пропускной способности с простым предсказание для ряда BC-10

Заметим, что выигрыш в статистике Dplus равен выигрышу в статистике Dminus. Как можно видеть из рис. 4.13, выигрыши в статистиках Dplus и Dplus_drops отличаются незначительно, особенно в рабочем диапазоне (при $bs_norm > 2$), характеризующимся приемлемым уровнем потерь (см. ниже). Поэтому для дальнейшего анализа выберем одну из этих статистик, например Dplus. Полученные в результате моделирования на ns-2 основные статистики для методов статического и динамического распределения пропускной способности приведены на рис. 4.14, 4.15, 4.16 соответственно.

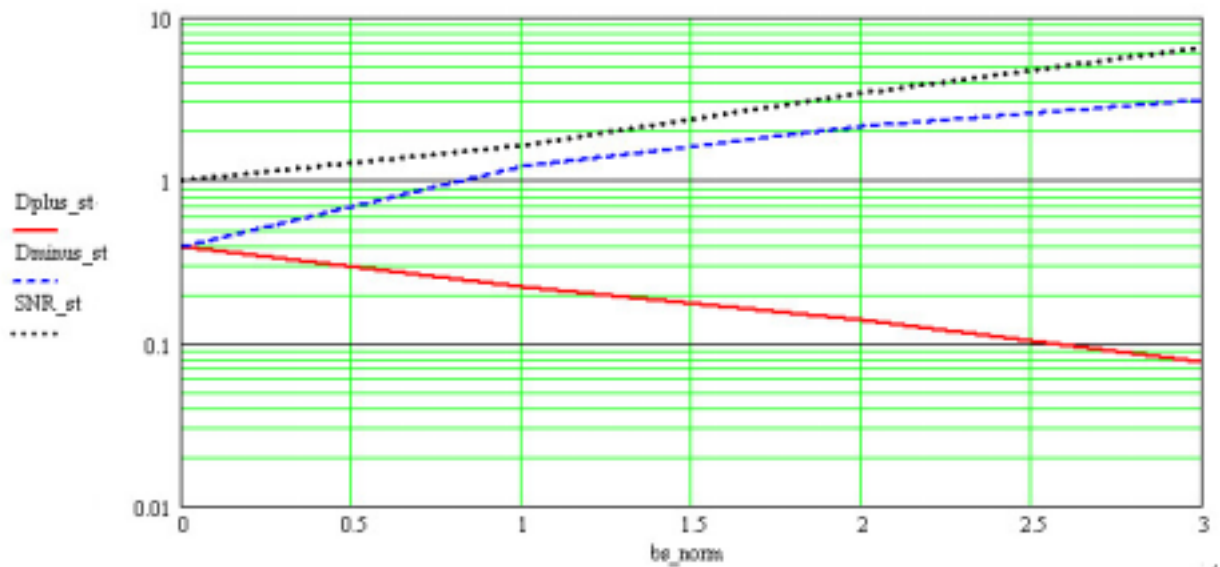


Рис. 4.14. Основные статистики, полученные в результате моделирования на ns-2 механизма статического распределения пропускной способности для ряда ВС-10

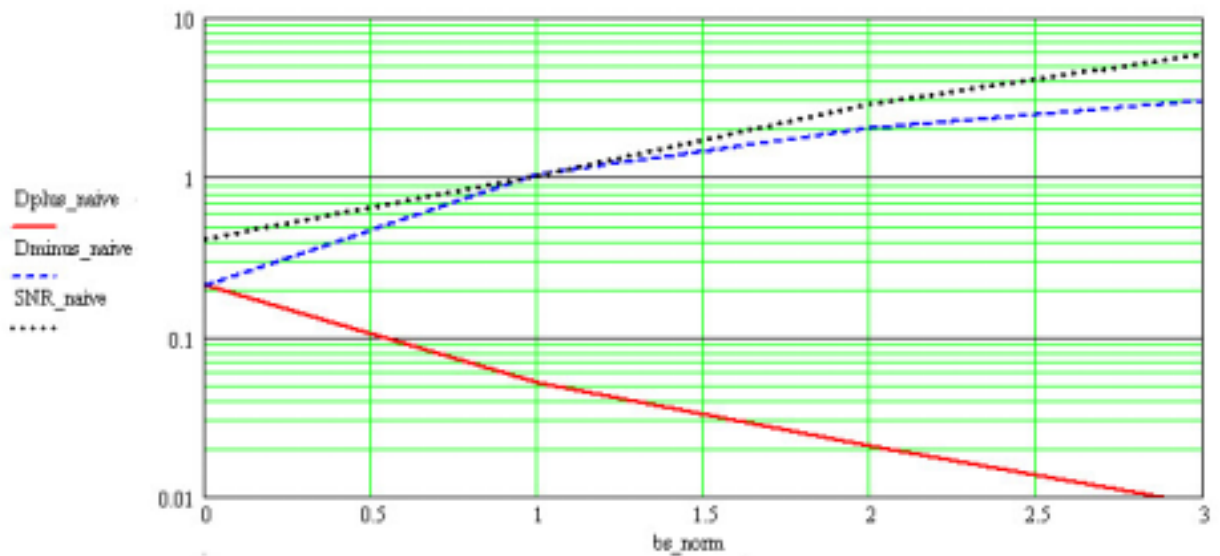


Рис. 4.15. Основные статистики, полученные в результате моделирования на ns-2 механизма динамического распределения пропускной способности с простым прогнозированием для ряда ВС-10

Сравнивая полученные зависимости с аналогичными, представленными ранее в главе 3 (см. рис. 3.4 и рис. 3.5), можно отметить, что результаты моделирования хорошо согласуются с теоретическими расчетами.

Выигрыш в статистике SNR^{-1} определяется аналогичным образом:

$$\text{AdvSNR}_{\text{naive}}(\text{bs_norm}) = \text{SNR}_{\text{st}}(\text{bs_norm}) - \text{SNR}_{\text{naive}}(\text{bs_norm}). \quad (4.3)$$

Заметим, что приведенные выше характеристики имеет смысл рассматривать, если для каждого значения bs_norm средняя пропускная способность канала N2-N3 не зависит от механизма распределения пропускной способности (статического или динамического с прогнозированием).

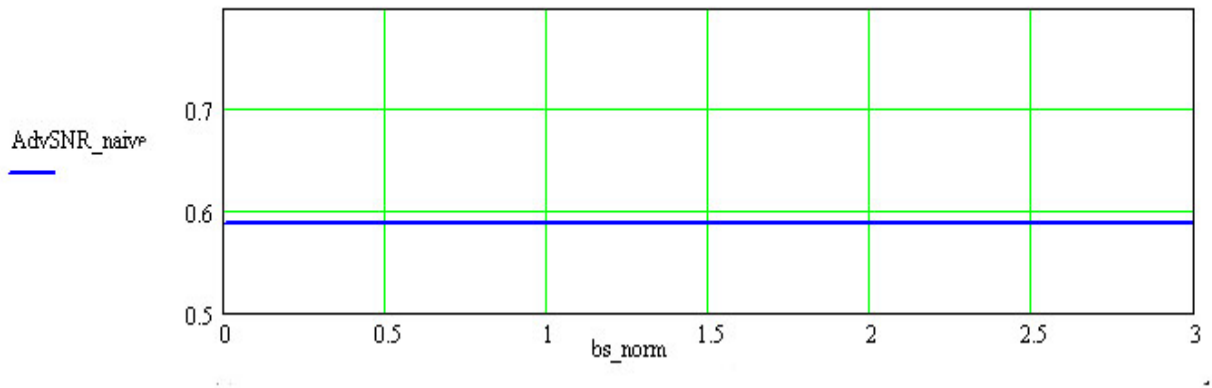


Рис. 4.16. Выигрыш в статистике SNR^{-1} от применения метода динамического распределения пропускной способности с простым прогнозированием для ряда BC-10

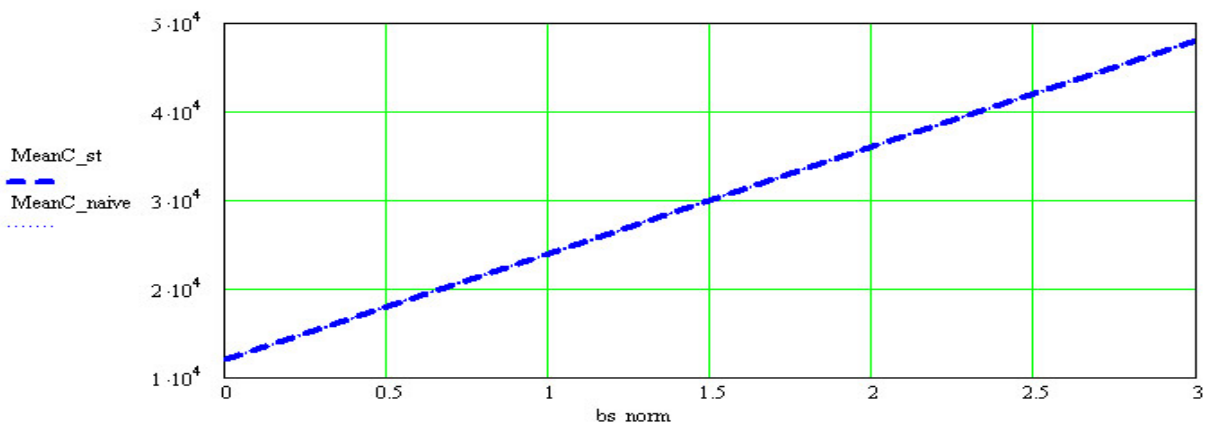


Рис. 4.17. Средняя пропускная способность (бит/с) канала N2-N3 для методов статического (MeanC_st) и динамического (MeanC_naive) распределения пропускной способности с простым прогнозированием для ряда BC-10

Проконтролируем неизменность среднего значения пропускной способности канала N2-N3 для рассматриваемых механизмов. Соответствующие графики приведены на рис. 4.17.

Как можно видеть, средняя пропускная способность этого канала не зависит от метода ее распределения и растет линейно с увеличением bs_norm . Последнее объясняется линейным ростом статической компоненты bs (см. (3.6) в процессе проведения эксперимента.

Проанализируем еще одну статистическую характеристику – количество информации, принятое узлом N4 от источника CBR-трафика за время эксперимента при различных методах распределения пропускной способности. Как замечалось ранее, при заданном значении bs_norm среднее значение пропускной способности канала N2-N3 для обоих методов управления остается постоянным. Следовательно, постоянным остается и связанная с ней величина – средняя пропускная способность канала N0-N3 (см рис.4.11). Тогда, располагая графиком на рис.4.17, можно оценить количество информации, принятое от источника N0 узлом N4. Например, для параметра $bs_norm = 1.5$ средняя пропускная способность канала N2-N3 равна примерно 30 000 бит/с. Значит, средняя пропускная способность канала N0-N3 равна $200\ 000 - 30\ 000 = 170\ 000$ бит/с, где 200 000 бит/с – пропускная способность канала N3-N4, задаваемая параметром $band3_4$ в табл. 4.1. В этом случае количество информации, принятое узлом N4 от источника CBR трафика за все время эксперимента (20 000 сек) составляет порядка $170\ 000 * 20\ 000 / 8 = 4.25 * 10^8$ байт. Проверим этот результат с помощью зависимостей, представленных на графике (рис. 4.19) и полученных с помощью встроенных в ns-2 счетчиков принятых пакетов (агента LossMonitor).

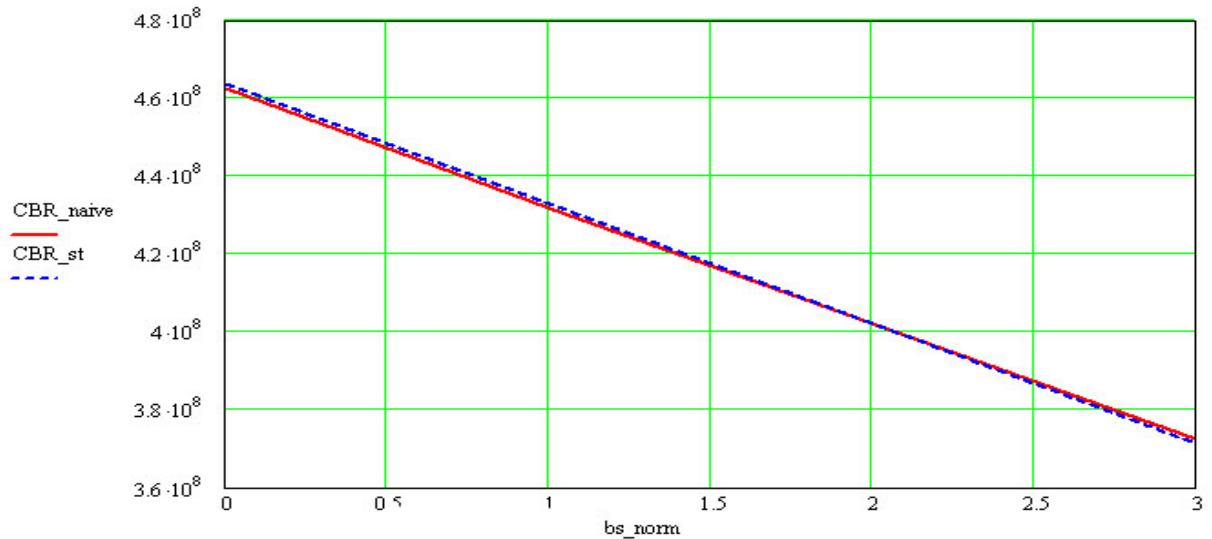


Рис. 4.18. Количество информации (байт), принятое на узле N4 от источника N0 для статического (CBR_st) и динамического (CBR_naive) методов распределения пропускной способности

при том же самом объеме информации, принятой узлом N4 от источника N0, потери при передаче самоподобного трафика от источника N5 заметно (~10 %) ниже, при использовании метода динамического распределения пропускной способности. При этом показатель SNR^{-1} , характеризующий джиттер, улучшатся на 59 %.

Кроме того, заметим, что метод динамического управления пропускной способности с помощью простого предсказания практически не требует вычислительной мощности процессора (по сравнению с FARIMA-предсказателем, например) и достаточно легко реализуем.

Теперь приведем результаты моделирования с использованием в качестве источника самоподобного трафика реализацию Weth04.dat, зафиксированную в беспроводной сети крупного интернет-провайдера. Свойства и характеристики данной реализации также исследовалась нами ранее в разделе 2.6 главы 2 настоящей диссертации.

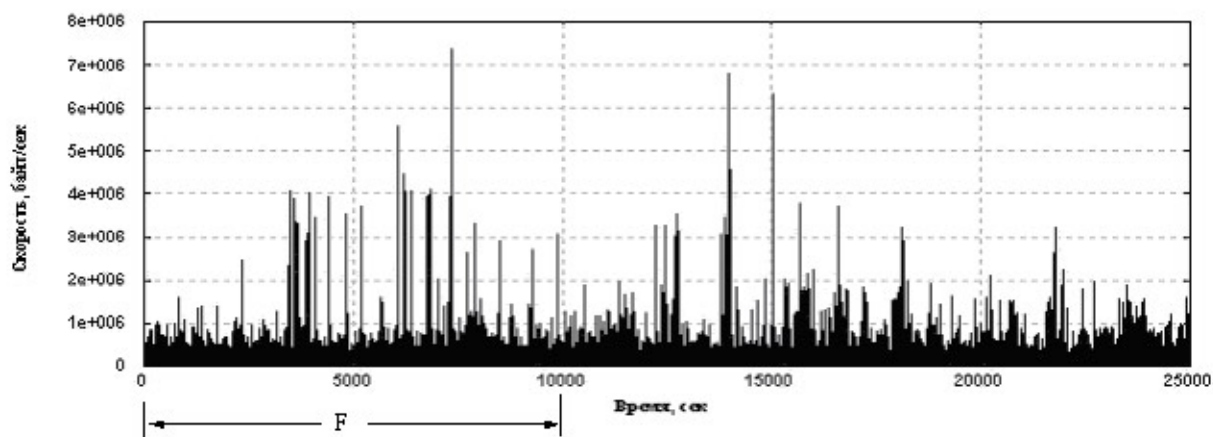


Рис. 4.19. Временной ряд Weth04_0.05.dat и исследуемый участок F

Выберем в качестве исследуемого участка, например, интервал между началом реализации и временной меткой 10000 секунд, продолжительностью ~ 2.778 часа (см. рис. 4.19).

Параметры программы `tr-sim.tcl` для проведения данного эксперимента и получения приведенных ниже результатов представлены в табл. 4.2:

Табл. 4.2. Значения параметров в программе `tr-sim.tcl`

Название параметра	Значение параметра
<code>max_band2_3</code>	10000000.0
<code>max_band0_3</code>	10000000.0
<code>\$ns simplex-link \$n0 \$n3</code>	10000000.0 1ms DropTail
<code>\$ns simplex-link \$n1 \$n2</code>	10000000.0 1ms DropTail
<code>\$ns simplex-link \$n2 \$n3</code>	0.0 1ms DropTail
<code>\$ns simplex-link \$n3 \$n4</code>	10000000.0 1ms DropTail
<code>\$ns simplex-link \$n5 \$n1</code>	10000000.0 1ms DropTail
<code>\$cbr0 set rate_</code>	10000000.0
<code>band3_4</code>	10000000.0
<code>band0_3</code>	10000000.0

При этом представим результаты моделирования уровня агрегирования 0.05 сек.

На рис. 4.20 изображены зависимости выигрышей в коэффициенте недооценки (коэффициенте потерь) при использовании метода динамического распределения пропускной способности с простым предсказанием для ряда Weth04.

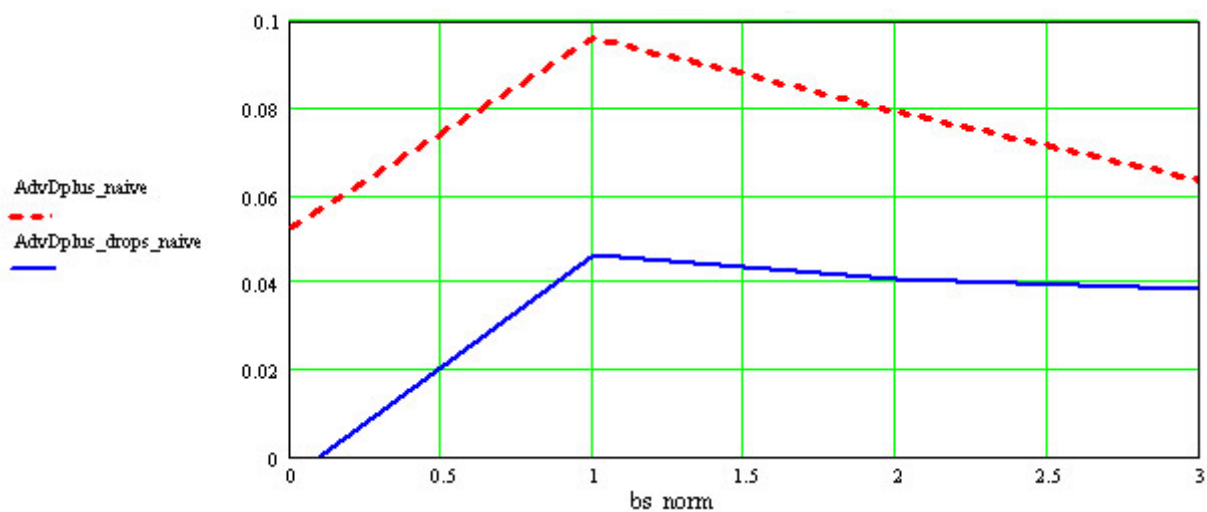


Рис. 4.20. Выигрыши в статистиках Dplus и Dplus_drops от применения метода динамического распределения пропускной способности с простым предсказание для ряда Weth04

Как можно видеть, выигрыш (в статистике Dplus) при среднем значении пропускной способности канала N2-N3, равной трем средним значениям интенсивности трафика на изучаемом участке (т.е. в точке $bs_norm=2$) составляет $\sim 8\%$. Это значение хорошо согласуется с результатами, рассмотренными ранее для ряда BC-10 ($\sim 10\%$). На рис. 4.21 и 4.22 приведены основные изучаемые статистики для случаев статического и динамического распределения пропускной способности соответственно. Далее, на рис. 4.23 представлена зависимость выигрыша в коэффициенте SNR^{-1} , определяемый в соответствии с соотношением (4.3). Величина выигрыша не зависит от значения bs_norm и составляет $\sim 58\%$.

Анализируя графики, представленные на рис 4.24, можно убедиться, что в ходе эксперимента среднее значение пропускной способности канала N2-N3 для заданного значения bs_norm сохраняется неизменным при переходе от статического метода к методу динамическому. Таким образом, показатели

эффективности (SNR^{-1} , D^+ и D^-) для этих методов можно сравнивать между собой для любого фиксированного значения bs_norm .

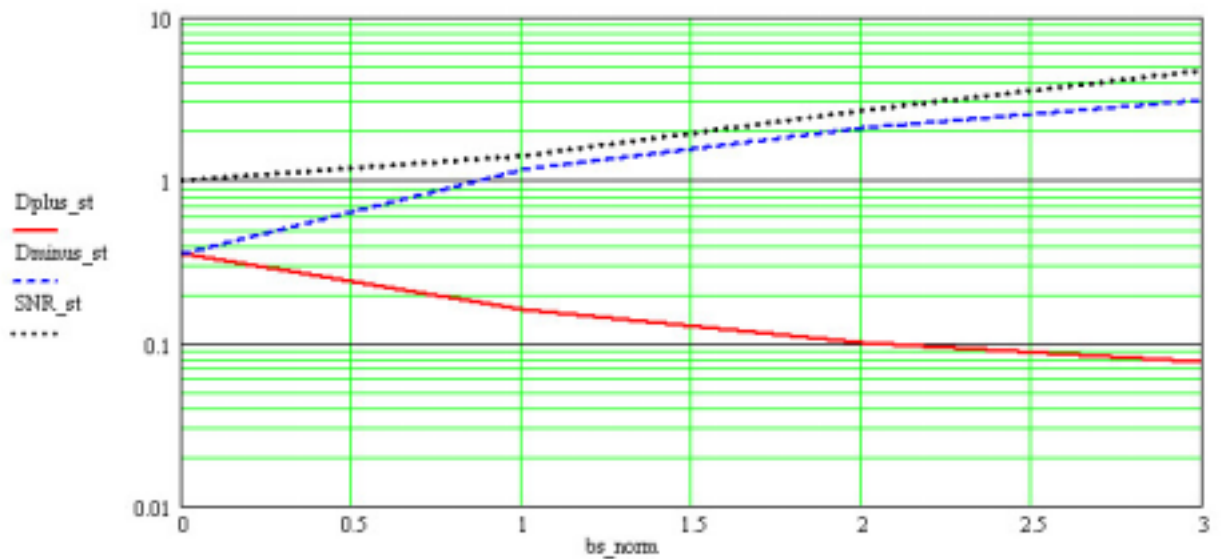


Рис. 4.21. Основные статистики, полученные в результате моделирования на ns-2 механизма статического распределения пропускной способности для ряда Weth04

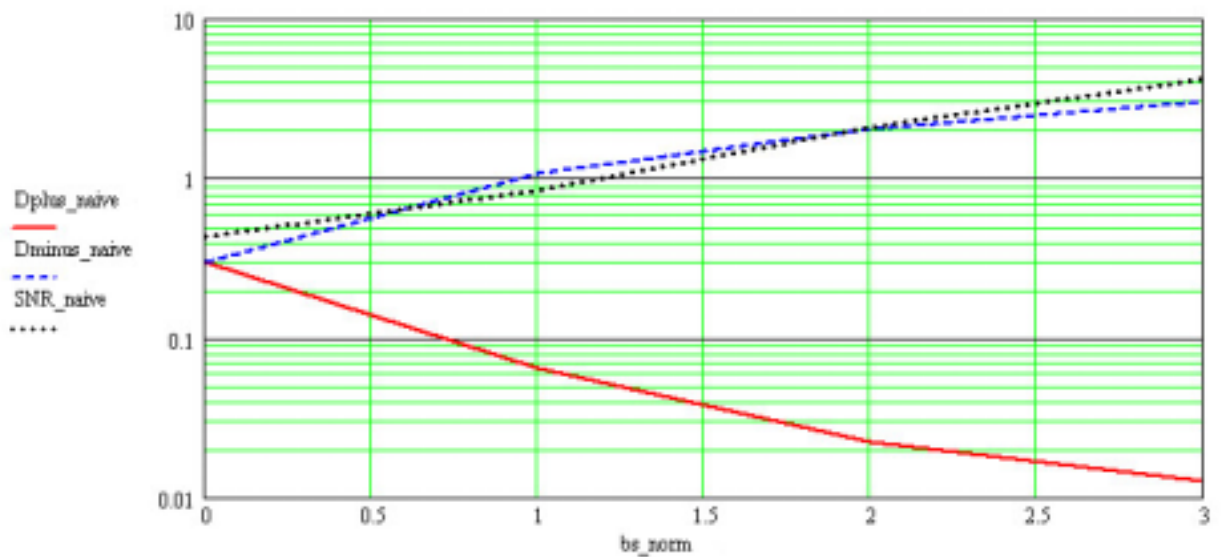


Рис. 4.22. Основные статистики, полученные в результате моделирования на ns-2 механизма динамического распределения пропускной способности с простым прогнозированием для ряда Weth04

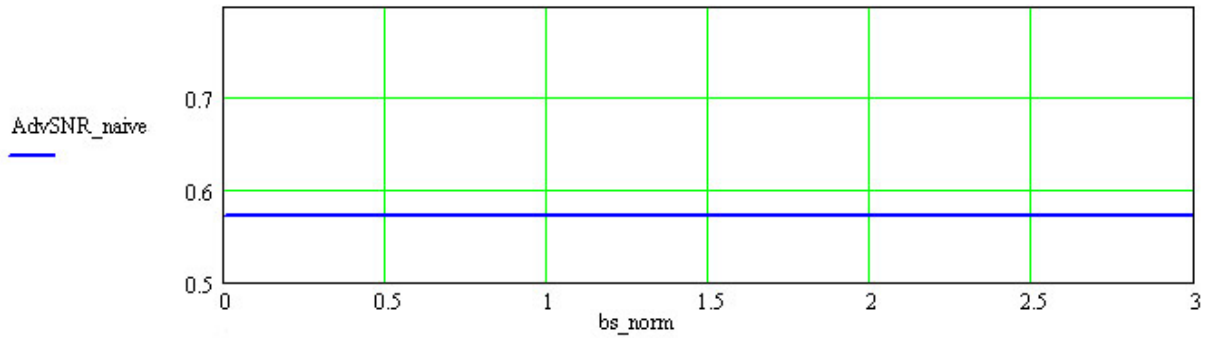


Рис. 4.23. Выигрыш в статистике SNR^{-1} от применения метода динамического распределения пропускной способности с простым прогнозированием для ряда Weth04

Более того, судя по графику, представленному на рис. 4.25, в случае применения метода динамического распределения пропускной способности не только удается достичь уменьшения потерь информации в канале N2-N3 на 8 %, но и сохранить (и даже несколько увеличить) количество информации, переданное в канале N0-N4. Параметр SNR^{-1} при этом улучшается на 58%.

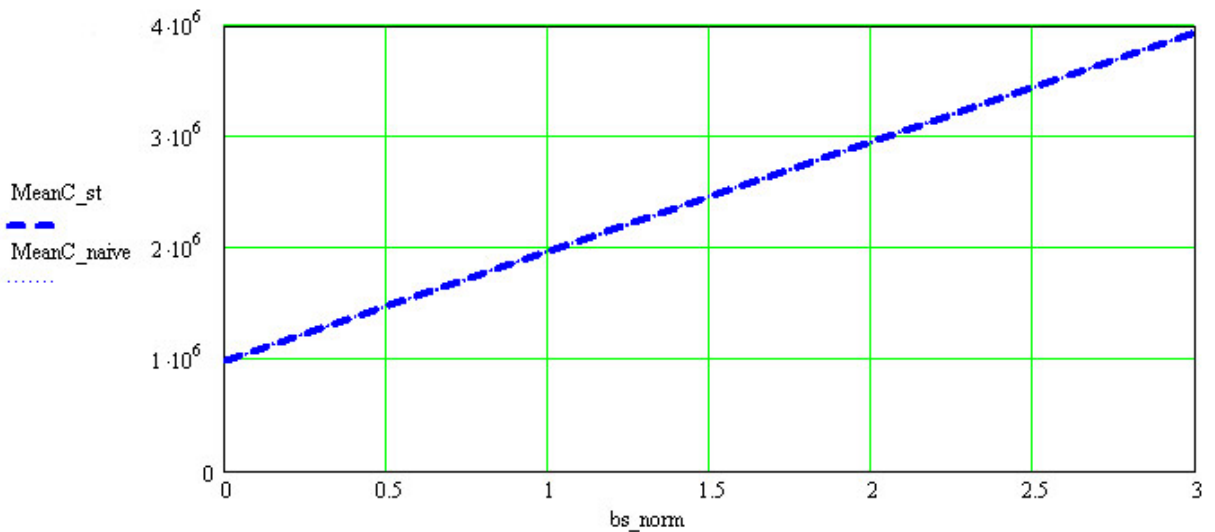


Рис. 4.24. Средняя пропускная способность (бит/с) канала N2-N3 для методов статического (MeanC_st) и динамического (MeanC_naive) распределения пропускной способности с простым прогнозированием для ряда Weth04

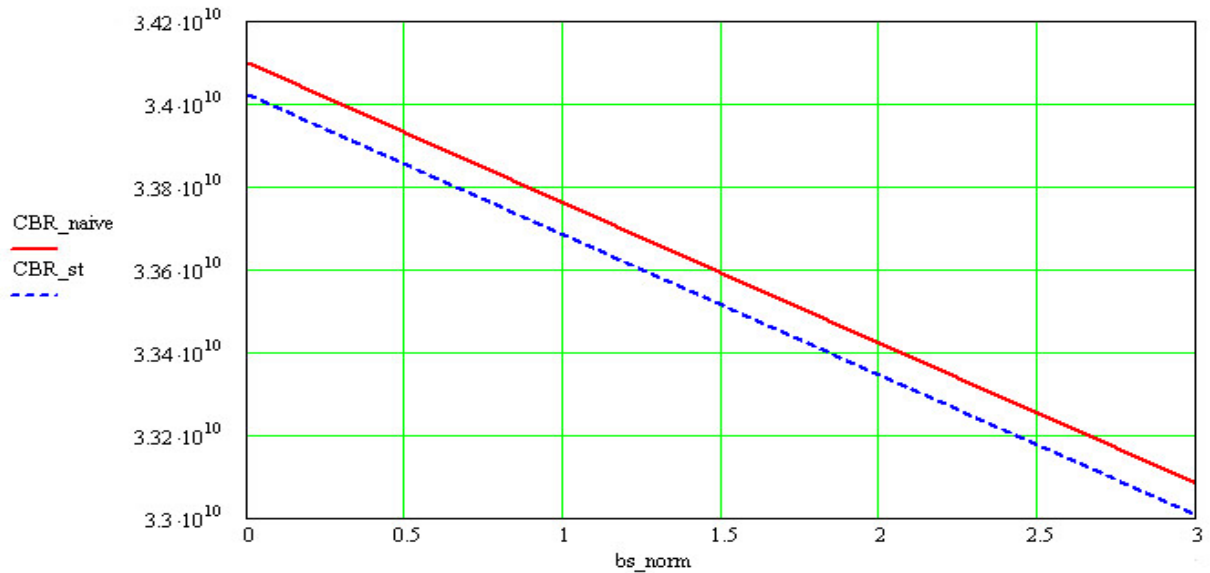


Рис. 4.25. Количество информации (байт), принятое на узле N4 от источника N0 для статического (CBR_st) и динамического (CBR_naive) методов распределения пропускной способности с простым прогнозированием для ряда Weth04

Полученные результаты подтверждают выводы, сделанные ранее в главе 3 диссертации, о безусловном повышении эффективности системы вследствие применения метода динамического распределения пропускной способности с помощью прогнозирования: при том же самом объеме информации, полученной узлом N4 от источника N0, потерь в самоподобном трафике заметно меньше (8-10 %) при использовании алгоритма динамического распределения пропускной способности с прогнозированием. При этом параметр SNR^{-1} , характеризующий джиттер, улучшается на 58%. Повышение общей эффективности системы обусловлено более эффективным распределением ее ресурсов.

Величина полученного в результате имитационного моделирования выигрыша от применения метода динамического распределения пропускной способности соответствует оценкам, произведенным в главе 3 диссертации, что подтверждает корректность расчетов.

В заключении необходимо отметить, что разработанный метод не является панацеей в борьбе с проблемой самоподобия. Область его применения ограничивается случаями, когда возможно (чаще всего это удастся сделать)

разделить трафик на две категории: приоритетный и неприоритетный, а также когда используется техника виртуальных каналов (в отличие от статистического мультиплексирования потоков). При этом если приоритетный трафик оказывается самоподобным и обладает высоким коэффициентом пачечности, то разработанный алгоритм будет эффективен. В противном случае его эффективность окажется не хуже эффективности метода статического распределения пропускной способности.

Полные распечатки всех программ приведены в Приложении 2.

4.6 Выводы по главе 4

1. Настоящая глава посвящена разработке алгоритма обеспечения качества обслуживания (QoS) в условиях влияния эффекта самоподобия с помощью схемы динамического (адаптивного) распределения пропускной способности на основе прогнозирования.
2. Представлены возможные структурные схемы систем динамического распределения пропускной способности с прогнозированием и без него. Показано, что схема, не использующая прогнозирование, вносит задержки и поэтому практически неприменима в системах передачи информации реального времени. Вместо схем со статическим способом задания пропускной способности для пачечного трафика предлагается использовать схемы с прогнозированием. Показан возможный вариант применения такой схемы в реальной ситуации.
3. Подробно изучены известные алгоритмы контроля и управления трафиком полисинга и шейпинга на основе механизма “корзины маркеров”, использующие метод статического задания пропускной способности.
4. На основе алгоритмов полисинга и шейпинга разработан новый алгоритм обеспечения качества обслуживания, реализующий принцип динамического распределения пропускной способности с помощью прогнозирования и предназначенный для работы с самоподобным трафиком.

5. В среде популярного сетевого эмулятора ns-2 разработан комплекс программ, имитирующий функционирование алгоритма динамического распределения пропускной способности с прогнозированием для случая реального трафика, обладающего самоподобными свойствами.

6. Экспериментально с помощью моделирования показано, что предлагаемый механизм в случае простого предсказателя дает заметный выигрыш (~ 8-10 %) в уменьшении потерь и увеличении эффективности использования канала при самоподобном трафике по сравнению со статическим способом распределения пропускной способности. При этом параметр SNR^{-1} , характеризующий джиттер, улучшается на 58%. Повышение общей эффективности системы обусловлено более эффективным распределением ее ресурсов.

Заключение

Настоящая диссертация посвящена исследованию свойств самоподобного телетрафика и разработке метода, позволяющего добиться более эффективного способа его обработки в системах распределения информации, чем существующие методы, связанные со статическим распределением пропускной способности соответствующих каналов.

Основным результатом проведенных в диссертационной работе теоретических и экспериментальных исследований является разработанный **метод обеспечения качества обслуживания, основанный на прогнозировании и предназначенный для работы с самоподобным трафиком**. Метод реализует принципы предоставления ресурсов “по требованию” и динамического распределения пропускной способности каналов телекоммуникационных сетей. При этом достигается выигрыш (~ **58%**) в статистике SNR^{-1} , характеризующей джиттер, а также в уменьшении потерь и увеличении использования ресурсов системы (~ **10%**) за счет более эффективного их распределения.

Для достижения этой цели в работе сделано следующее:

1. Подробно рассмотрено современное состояние и основные понятия теории самоподобного телетрафика. С точки зрения возможностей прогнозирования рассмотрены основные концепции, свойства и характеристики самоподобного телетрафика, такие как фрактальность, самоподобие, коэффициент Херста, хаос, продолжительная память и распределения с тяжелыми хвостами и др. Изложенный материал не является переводом рекомендаций, документов или научных статей и представляет собой результат анализа нескольких десятков научных работ.
2. Показано, что существующие методы управления трафиком (шейпинг и полисинг) оказываются малоэффективны при обработке самоподобного

телетрафика. В частности, для уменьшения потерь алгоритм полисинга требует увеличения пропускной способности канала, в результате чего уменьшается его использование (снижается утилизация), а алгоритм шейпинга вносит задержки, что может быть неприемлемо при обработке информации реального времени.

3. Подготовлен и выполнен оригинальный эксперимент по сбору и исследованию трафика беспроводной сети, подтверждающий наличие самоподобных свойств в трафике современных телекоммуникационных сетей, использующих технологии беспроводного доступа IEEE 802.11b в том числе. Полученные реализации трафика беспроводной сети представлены для публичного пользования в Интернет по адресу www.teletraffic.ru.
4. Представлены результаты статистического анализа реализаций сетевого трафика. Целью проведенного анализа являлось выявления характерных особенностей телетрафика, которые необходимо учитывать при разработке алгоритма обеспечения качества обслуживания в условиях влияния эффекта самоподобия. Здесь можно выделить следующие результаты:
 - результаты регрессионного анализа АКФ реализаций сетевого трафика подтверждают присутствие медленно убывающих зависимостей для всех исследуемых случаев;
 - произведено измерение показателя Хэрста (H) семью методами: анализа дисперсии, нормированного размаха (R/S), периодограмм, абсолютных моментов, дисперсии остатков, Эбри-Вейча и Виттла. Обнаружено, что для всех реализаций сетевого трафика $H > 0.5$, то есть трафик относится к классу персистентных процессов. Использование разнообразных методов оценки показателя Хэрста преследовало цель получить более достоверные результаты. Усредненное значение H для сетевого трафика $H \sim 0.8$. Зависимости

коэффициента H от уровня агрегирования (для рассматриваемых уровней) не выявлено;

- обнаружено, что в агрегированном трафике канального и транспортного уровней присутствуют значительные гармонические составляющие. В этой связи при разработке адекватных математических моделей телетрафика следует обращать внимание на присутствие периодических компонент;
- использование методов теории нелинейной динамики (изучение графиков FNN, корреляционных интегралов, анализа BDS-статистики) позволяет сделать выводы о присутствии в реализациях сетевого трафика некоторой детерминированной, однако достаточно слабой составляющей. Кроме того, показано, что гипотеза о статистической независимости членов ряда, характеризующего сетевой трафик отвергается.

5. Изложены теоретические обоснования прогнозируемости самоподобных процессов, обладающих медленно убывающей АКФ. Получено аналитическое выражения для интервала корреляции самоподобного процесса, а также указано, какие свойства самоподобного телетрафика могут влиять на его прогнозируемость.
6. Поставлена задача прогнозирования сетевого трафика в составе задачи динамического (адаптивного) распределения пропускной способности канала. Определен алгоритм проверки прогнозируемости и оценки качества прогноза. Наряду с классической оценкой сигнал/шум разрабатываются другие оценки: коэффициент потерь и коэффициент недоиспользования.
7. С помощью статистических исследований оценены характеристики потерь при обработке самоподобного сетевого трафика в системах с динамическим распределением пропускной способности, использующих различные алгоритмы прогнозирования: простой предсказатель,

авторегрессионные первого и второго порядка, ARMA и FARIMA – предсказатели различных порядков.

8. На основании результатов сравнения оценок потерь в схемах динамического распределения пропускной способности (для различных алгоритмов предсказания) с аналогичными показателями, полученными для статического способа задания пропускной способности, а также сложности реализации той или иной схемы, сделан выбор наиболее подходящего алгоритма – простого предсказателя.
9. В среде сетевого эмулятора ns-2 разработан комплекс программ, имитирующий функционирование алгоритма динамического распределения пропускной способности с прогнозированием для случая реального трафика, обладающего самоподобными свойствами.
10. Экспериментально с помощью моделирования показано, что предлагаемый механизм даже в случае простого предсказателя дает заметный выигрыш (~ 8-10 %) в уменьшении потерь и увеличении эффективности использования канала при самоподобном трафике по сравнению со статическим способом распределения пропускной способности. При этом параметр SNR^{-1} , характеризующий джиттер, улучшается на 58%. Повышение общей эффективности системы обусловлено более эффективным распределением ее ресурсов.

Список литературы

...Значит, нужные книги ты в детстве читал...

В. Высоцкий “Баллада о борьбе”

1. Leland W.E., Taqqu M.S., Willinger W., and Wilson D.V. On the self-similar nature of ethernet traffic // IEEE/ACM Transactions of Networking, 2(1), 1994. – P. 1-15.
2. Цыбаков Б.С. Модель телетрафика на основе самоподобного случайного процесса // Радиотехника. – 1999. – № 5. – С. 24-31.
3. В.И. Нейман. Новое направление в теории телетрафика // Электросвязь. – 1998. – № 7. – С. 27-30.
4. Tsybakov B.S., Georganas N.D. Self-similar processes in communications networks // IEEE Trans. Inform. Theory, vol. 44. – Sep.1998. – P. 1713-1725.
5. Tsybakov B.S., Georganas N.D. Self-similar traffic: upper bounds to buffer-overflow probability in an ATM queue // Proceedings of CCBR'97, the Canadian Conference on Broadband Research, Ottawa. – 1997. – P. 137-148.
6. Gneiting T., Schlather M. Stochastic models which separate fractal dimension and Hurst effect // NRCSE-TRS. – Sep. 20, 2001. – № 069.
7. Кендел М. Временные ряды: Пер. с англ. и предисл. Ю.П. Лукашина. – М: Финансы и статистика, 1981. – 199с.
8. Feng W., Tinnakornsriruphap P. The Failure of TCP in High-Performance Computational Grids // SC2000: High-Performance Network and Computing Conference, Dallas, TX. – November, 2000.
9. Заборовский В.С. Методы и средства исследования процессов в высокоскоростных компьютерных сетях: Диссертация на соискание ученой степени доктора технических наук – СПб., 1999 г.
10. Городецкий А.Я., Заборовский В.С., Информатика. Фрактальные процессы в компьютерных сетях / Учебное пособие. – СПб.: СПбГТУ, 2000.
11. Grabbe O. Chaos and Fractals in Financial Markets. www.aci.net/kalliste/.

12. Найденов В.И., Кожевникова И.А. Эффект Харста в геофизике // Природа. – 2000. – №1.
13. Шелухин О.И., Тенякшев А.М., Осин А.В. Фрактальные процессы в телекоммуникациях. Монография: Под ред. О.И. Шелухина. – М.: Радиотехника, 2003. – 480 с.
14. Криштофович А.Ю. Самоподобие трафика сети ОКС №7 // МКИССиТ, Санкт-Петербург, 2002 г.
15. Бершадский А.В. Статистическая модель рыночных событий. Электронный журнал “Исследовано в России”, 1476-1488, 2002 г.
<http://zhurnal.ape.relarn.ru/articles/2002/132.pdf>.
16. Veres A., Voda M. The Chaotic Nature of TCP Congestion Control // Proceedings of IEEE INFOCOM'2000, March 2000.
17. Veres A., Kenesi Zs., Molnar S., Vattay G. On the Propagation of Long-Range Dependence in the Internet // Proc. ACM SIGCOMM 2000. – Stockholm, Sweden, Sep. 2000.
18. Kugiumtzis D., Boudourides M. Chaotic Analysis of Internet Ping Data: Just a Random Generator? // SOEIS meeting at Bielefeld, March 27-28, 1998.
19. Петров В.В. Самоподобие в сетевом трафике // 58-я Научная сессия РНТОРЭС им. А.С. Попова: Сборник трудов. Том 2. – М., 14-15 мая 2003. – С. 126.
20. Петров В.В., Платов В.В. Исследование самоподобной структуры телетрафика беспроводной сети // Радиотехнические тетради. – 2004. – № 30. – С. 58 – 62.
21. Петров В.В., Богатырев Е.А. О самоподобном сетевом трафике // РАДИОЭЛЕКТРОНИКА, ЭЛЕКТРОТЕХНИКА И ЭНЕРГЕТИКА: Тез. докл. Девятой Междунар. научно-техн. конференции студентов и аспирантов. Том 1. – М: Издательство МЭИ, 4-5 марта 2003. – С. 53 – 54.
22. Петров В.В., Богатырев Е.А. Статистический анализ сетевого трафика // РАДИОЭЛЕКТРОНИКА, ЭЛЕКТРОТЕХНИКА И ЭНЕРГЕТИКА: Тез.

- докл. Десятой Междунар. научно-техн. конференции студентов и аспирантов. Том 1. – М: Издательство МЭИ, 2-3 марта 2004.
23. Park K., Willinger W. Self-Similar Network Traffic and Performance Evaluation. – John Wiley & Sons, 2000.
 24. Вегешна Ш. Качество обслуживания в сетях IP.: Пер. с англ. – М.: Изд. дом “Вильямс”, 2003. –368 с.
 25. Policing and Shaping Overview, QC: Cisco IOS Release 12.0 Quality of Service Solutions Configuration Guide. <http://www.cisco.com>.
 26. Östring S., Sirisena H. The Influence the Long-Range Dependence on Traffic Prediction // Proceedings of ICC'01. – Helsinki, June 2001.
 27. Miloucheva I., Müller E., Anzaloni A. A practical approach to forecast Quality of Service parameters considering outliers, 2003. http://www.ist-intermon.org/workshop/papers/06_03_end_arima_corrected.pdf.
 28. Beran J. Statistical Methods for Data with Long-Range Dependence // Statistical Science, Volume 7, Issue 4. –1992. – P. 404-416.
 29. Foag J., Wild T. Traffic Prediction Algorithm for a Speculative Network Processor // 17th Intl. Symposium for High Performance Computing Systems and Applications HPCS 2003. – Sherbrooke, May 2003.
 30. Trajkovic L., Neidhardt A. Internet traffic prediction // Centre for Systems Science, Simon Fraser University, Vol. 12, Issue 1. – Mar. 2000.
 31. Koucheryavy Y., Harju J. A novel approach for self-similar traffic prediction. / Proceedings of the St. Petersburg Regional International Teletraffic Seminar, St. Petersburg, Russia, January 29 – February 1. – 2002. – P. 172 – 179.
 32. Анищенко В.С. Знакомство с нелинейной динамикой // Лекции соросовского профессора. – Саратов, 2000.
 33. Шредер М. Фракталы, хаос, степенные законы. Миниатюры из бесконечного рая. – Москва-Ижевск, 2001.
 34. Уайндер С. Справочник по технологиям и средствам связи. – М.: “Мир”, 2000.

35. Gripenberg G., Norros I. On the prediction of fractal Brownian motion // Journal of Applied Probability, vol.33. –1996. – P. 400-410.
36. Fowler H. J., Leland W. E. Local Area Network Traffic Characteristics, with Implications for Broadband Network Congestion Management // IEEE JSAC, 9(7) . – September 1991. – P. 1139-1149.
37. Paxson V., Floyd S. Wide-Area Traffic: The Failure of Poisson Modeling // IEEE/ACM Transactions on Networking, 1995.
38. Кузнецов С.П. Динамический хаос (курс лекций). – М.: Изд-во Физико-математической литературы, 2001. – 296 с.
39. Малинецкий Г.Г., Потапов А.Б. Современные проблемы нелинейной динамики. М.: Едиториал УРСС, 2002. – 360с.
40. Theiler J. Some comments on the correlation dimension of $1/f^\alpha$ noise // Phys. Lett. A. 155. – 1991. – P. 480-493.
41. Olowoyeye G., Kim B., Chandra K. Modelling Spectral Features in TCP Traffic. Submitted to ITC'99, October 1998.
42. Chandra K., You C., Olowoyeye G. and Thompson C. Non-linear Time-Series Models of Ethernet Traffic // Submitted to INFOCOM'99, July 1998.
43. Erramilli A., Singh R.P., Pruthi P. An Application of Deterministic Chaotic Maps to Model Packet Traffic // Queueing Systems, 20(3) . –1995. – P.171-206.
44. Potapov A., Kurths J. Correlation integral as a tool for distinguishing between dynamics and statistics in time series data // Physics D. 120. –1998. – P.369-385.
45. Brock W. A., Dechert W. D., Scheinkman J.A. and LeBaron B. A test for independence based on correlation dimension // Econometric Reviews 15. – 1996. – P. 197-235.
46. Kennel M. B., Brown R. and Abarbanel H. D. I. Determining embedding dimension for phase-spase reconstruction using a geometrical construction // Phys. Rev. A 45. – 1992. – P. 3403.
47. Gao J.B., Cao Y., Lee J-M. Principal component analysys of $1/f^\alpha$ noise // Physics Letters A, 314. – 2003. – P. 392-400.

48. Hegger R., Kantz H., Schreiber T. Practical implementation of nonlinear time series methods: The TISEAN package, 1998.
49. Eckmann J.P., Kamphorst S. and Ruelle D. Recurrence plots of dynamical systems // Europhys. Lett. 4. – 1987. – P. 973-977.
50. Голяндина Н.Э., Некруткин В.В., Браулов К.А. Метод “Гусеница”-SSA: анализ временных рядов, 2002. http://www.gistatgroup.com/gus/ssa_an.pdf.
51. Ghaderi M. On the Relevance of Self-Similarity in Network Traffic Prediction, 2003. <http://www.cs.uwaterloo.ca/cs-archive/CS-2003/28/TR-CS-2003-28.pdf>.
52. Sadec N., Khotanzad A., Chen T. ATM Dynamic Bandwidth Allocation Using F-Arima Prediction Model. <http://snoopy.seas.smu.edu/papers/icccn03.pdf>.
53. Айвазян С.А. Прикладная статистика. Основы эконометрики: Учебник для вузов: В 2 т. 2-е изд. испр. - Т.2.: Айвазян С.А. Основы эконометрики. – М.: ЮНИТИ-ДАНА, 2001. – 432 с.
54. Dang T. D., Sonkoly B., Molnar S. Fractal Analysis and Modelling of VoIP Traffic // NETWORKS2004, Vienna, Austria, June 13-16, 2004.
55. Пономарев Д.Ю. Вероятностно-временные характеристики асинхронных систем обработки интегральной информации с учетом влияния свойства самоподобия. Диссертация на соискание ученой степени кандидата технических наук – Красноярск, 2002.
56. Chong S., Li S., Grosh J. Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real-Time VBR Video over ATM // IEEE Journal on Selected Areas in Communications, Vol.13, No 1. – January 1995. – P. 12-23.
57. Кучерявый Е.А. Управление трафиком и качество обслуживания в сети Интернет. – СПб.: Наука и Техника, 2004. – 336 с.
58. Petroff V. Self-Similar Network Traffic: From chaos and Fractals to Forecasting and QoS // NEW2AN. – St.Petersburg, 2004. – P. 110-118.
59. Traffic Modeling Based on FARIMA Models. Xue F., Liu J., Shu Y., Zhang L., Yang O.W.W // CCECE99 Proceed. – May 1999. – P. 162-167.

60. A High Speed Implementation of Adaptive Shaping for Dynamic Bandwidth Allocation. Brown C., Sirkay V., Uriona H., Seetharam S., Yousefi E., Petr D., Niehaus D., Frost V., Evans J., Minden G. // IEEE Communications Magazine, 1997.
61. Chiruvolu G., Sankar R., Ranganathan N. Adaptive VBR Video Traffic Management for Higher Utilization of ATM Networks // ACM SIGCOMM, Vol. 28, Issue 3. – July 1998. – P. 27-40.
62. Park K., Kim G., Crovella M. On the relationship between file sizes, transport protocols, and self-similar network traffic // Proceedings of the Fourth International Conference on Network Protocols (ICNP'96) . –1996. – P.171-180.
63. Заборовский В.С., Рязанов М.Г. Управление в компьютерных сетях: концепция сетевых процессоров // Демиург. – 1998. – №1, – С. 47-81.
64. Paxson V., Floyd S. Wide-Area Traffic: The Failure of Poisson Modeling // IEEE/ACM Transactions on Networking. – 3(3) . – 1995. – P. 226-244.
65. Fowler H.J., Leland W.E. Local area network traffic characteristic, with implications for broadband network congestion management // IEEE Journal on Selected Areas in Communications. vol. 9. – 1991. – P. 1139-1149.
66. Zhao H., Ansari N., Shi Y.Q. A Fast Non-linear Algorithm for Video Traffic Prediction // ITCC, 2002.
67. Chen B., Peng S., Wang K. Traffic Modeling, Prediction, and Congestion Control for High-Speed Networks: A Fuzzy AR Approach // IEEE Trans. On Fuzzy Systems Vol. 8. – 2000. – №5.
68. Лившиц Б.С., Пшеничников А.П., Харкевич А.Д. Теория телетрафика // М.: "СВЯЗЬ", Учебник для вузов.2-е изд., перераб. и доп., 1979. – 224 с.
69. Столлингс В. Современные компьютерные сети: Питер, 2-е изд. (пер. с англ. Леонтьева А.), 2003 г. – 784 с.
70. Brenner P. A Technical Tutorial on the IEEE 802.11 Protocol // Breezecom, 1997.

71. Grassberger P., Procaccia I. Characterization of strange attractors // Phys. Rev. Lett. 58. – 1983. – P. 2387–2389.
72. Уолрэнд Дж. Телекоммуникационные и компьютерные сети. Вводный курс. Москва: Постмаркет, 2001.- 480 с.
73. Ершов М.А., Кузнецов Н.А. Теоретические основы построения сети с интеграцией служб. – М.: ИППИ РАН, 1995.
74. Лагутин В.С., Степанов С.И. Телетрафик мультисервисных сетей связи. – М.: Радио и связь, 2000. – 320 с.
75. Гмурман В.Е. Теория вероятностей и математическая статистика. Учеб. пособие для вузов. – 8-е изд., стер. – М.: Высш. шк., 2002. – 479 с.
76. Вентцель Е.С., Овчаров Л.А. Теория случайных процессов и ее инженерные приложения. – Учеб. пособие для вузов. – 2-е изд., стер. – М.: Высш. шк. , 2002. – 383с.
77. Баскаков С.И. Радиотехнические цепи и сигналы.: Учеб. для вызов по спец. ”Радиотехника”. – 2-е изд. перераб. и доп. – М.: Высш. шк., 1998 – 448 с.
78. Кирьянов Д.В. Самоучитель MathCad 2001. – СПб.: БХВ-Петербург, 2002. – 544с.
79. Документация и программное обеспечение сетевого симулятора ns-2: <http://www-mash.CS.Berkeley.EDU/ns>.
80. Архив трафика: <http://ita.ee.lbl.gov>.
81. Пакет программ для анализа временных реализаций методами нелинейной динамики TISEAN v2.1:http://lists.mpipksdresden.mpg.de/~tisean/TISEAN_2.1.
82. Программа FRACTAN v4.4, предназначенная для фрактального анализа временных реализаций: <http://impb.psn.ru/~sychyov/soft.shtml>.
83. Selfis v0.1b – программа для анализа экспоненты Хэрста разработки Thomas Karagiannis: <http://www.cs.ucr.edu/~tkarag>.
84. Официальный сайт проекта VINT: <http://www.isi.edu/nsnam/vint/index.html>.

Листинг программы для пакета Ox, предназначенной для прогнозирования временных рядов с помощью различных предсказателей

```

#include <oxstd.h>
#include <oxfloat.h>
#import <packages/arfima/arfima>

main()
{
    decl arfima, j, fmin, fmax, tTRENDmin, tFARIMAMin, tmax, MeanC, D, DAR, switch2;
    decl mx, i, Trend, Dplus, SNR, Dminus, TraTrendArray, file, st, CN, ly, dly, ddly;
    decl actual, MF, alpha, np, C, e, eplus, eminus, MeanT, MeanEcorrect, switch1, mf;

// *****declare the conditions of experiment *****

    tTRENDmin=2000; // start of estimation period for TREND
    tFARIMAMin=2000; // start of estimation period for FARIMA
    tmax=2999; // end of estimation period for TREND&FARIMA
    fmin=3000; // start of forecasting period
    fmax=6000; // end of forecasting period
    np=100; // number of points for graph of Dplus, Dminus and SNR

    switch1=0; // if Switch1=1 than forecasting with help second differences will be
              // if Switch1=0 than ordinarily forecasting will be

    switch2=1; // if Switch2=1 than Farima(p, d, q) forecasting will be
              // if Switch2=0 than naive forecasting will be (in this case the state of
              // "switch1" does not matter)

// *****Presets *****

    actual=zeros(fmax-fmin,1); //precreate matrix of actual sample
// Trend=zeros(fmax-fmin,1); //precreate matrix of Trend
    C=zeros(fmax-fmin, np); //precreate matrix of forecasted sample
    CN=zeros(fmax-fmin, 1); //precreate matrix of forecasted sample (without correction)
    e=zeros(fmax-fmin, np); //precreate matrix of absolute errors
    eplus=zeros(fmax-fmin, np); //precreate matrix for errors of underestimation
    eminus=zeros(fmax-fmin, np); //precreate matrix of absolute overestimation
    MeanT=zeros(fmax-fmin,1); //precreate matrix of trend (mean of training sample)
    Dplus=zeros(np,3); //precreate matrix for Dplus
    Dminus=zeros(np,3); //precreate matrix for Dminus
    SNR=zeros(np,3); //precreate matrix for SNR
    st=zeros(np,2); //precreate matrix for st
    TraTrendArray=zeros(tmax-tTRENDmin,1); //precreate matrix for TraTrendArray
    MeanC=zeros(np,3); //precreate matrix for MeanC
    DAR=zeros(fmax-fmin, 1); //precreate matrix for parameter "d"

    format ("%#13.8g"); //set new default format for double

    arfima = new Arfima();
    arfima.LoadObs("BC-10.dat", 2, 10000, 0, 0, 0.1, FALSE); // start year, perid, frequence

    ly = arfima.GetVar("Var2"); // var "ly" has actual values
    dly = diff0(ly, 1, M_NAN); // the first diff of "ly"
    ddly = diff0(dly, 1, M_NAN); // the second diff of "ly"
    arfima.Append(dly, "Inflat", 0); // add the dly to the database
    arfima.Append(ddly, "DInflat", 0); // add the ddly to the database

    arfima.Info();
    mx=arfima->GetAll();
// print (mx);

// *****Estimate the mean of forecasted range*****

    for (i=0; i<fmax-fmin; ++i)

```



```

        {
            actual[i]=mx[fmin+i][1]; // write actual values to the "actual"
            print ("actual[i]=", actual[i]);
        }
        MF=meanc(actual);

        print("MF=", MF);

// *****Static component of bandwidth *****

        alpha=9*MF; // Dplus, Dminus, SNR are changing in range 0...alpha

        for (i=0; i<np; ++i)
        {
            st[i][0]=i*alpha/np; // column "0" is the static bandwidth
        }
        st[][1]=st[][0]/MF; // column "1" is the static bandwidth in part of MF

//        print ("st=", st);

// *****Formulate the arfima model *****

        if (switch1==1)
            arfima.Select(Y_VAR, { "DInflat", 0, 0 } ); //forecasting with help second differeces is planing
        else
            arfima.Select(Y_VAR, { "Var2", 0, 0 } ); //ordinarily forecasting is planing

        arfima.Deterministic(TRUE);
        arfima.Info();
        arfima.Select(X_VAR, { "Constant",0,0, "Trend",0,0} );
        arfima.ARMA(1,2);
//        arfima.FixD(0);

//        arfima.SetPrint(FALSE);

// *****Forecasting *****

        for (i=0; i<fmax-fmin; ++i)

        {
            for (j=tTRENDmin; j<tmax; ++j) // -|
            { // |
                TraTrendArray[j-tTRENDmin][0]=mx[j][1]; // |
            } // | Estimate the TREND
            Trend=meanc(TraTrendArray); // |
            print("Trend=", Trend); // -|

//        arfima.FixMean(Trend); // -| Treatment of the mean
//        arfima.UseSampleMean(); // -|

        arfima.SetSelSample(tFARIMAMin, 0, tmax, 0);

//        arfima.SetMethod(M_MAXMPLIK); // -|
//        arfima.SetMethod(M_NLS); // | Set the estimation method
//        arfima.SetMethod(M_NLS_STATIONARY); // |
//        arfima.SetMethod(M_MAXLIK); // -|

        arfima.Estimate(); // -| Estimate the coefficients of model

        decl lastobs=tmax;
        print ("lastobs= ", lastobs, "\n");
        print ("ly=",ly[lastobs][]);

        if (switch1==1)
        {
            mf = arfima.Forecast(1, ly[lastobs - 1 : lastobs][]); //forecasting one step ahead with help second differeces is planing
            print ("mf[][6]=", mf[][6]);
        }
    
```

```

    }
    else
    mf = arfima.Forecast(1);           //ordinarily forecasting one step ahead is planing

    print("mf=", mf);

    decl D=arfima.GetD();             // -|
    print ("d=", d);                 // | Create array of parameter "d"
    DAR[i]=D;                        // -|

    tFARIMAMin=tFARIMAMin+1;         // -|
    tTRENDmin=tTRENDmin+1;           // | Increment of indexes
    tmax=tmax+1;                     // -|

if (switch2==1)
    {
    if (switch1==1)
        {
        if (mf[6]>0)                 // -| forecasting with help second differeces is planing
            CN[i][0]=mf[6];         // | write forecasted values to the matrix CN
        else                          // |
            CN[i][0]=0;
        }
    else
        {
        if (mf[0]>0)                 // -| ordinarily forecasting is planing
            CN[i][0]=mf[0];         // | write forecasted values to the matrix CN
        else                          // |
            CN[i][0]=0;             // -|
        }
    }
else
    {
    print ("naivy", "\n");
    if (i==0)
        CN[i][0]=Trend;
    else
        CN[i][0]=actual[i-1];
    }

print ("CN[i]=", CN[i][0], "\n");

    e[i][0]=actual[i]-CN[i][0];      // write absolute errors to the matrix "e"
    print ("e=", e[i][0], "\n");

    if (e[i][0]>=0)
        eplus[i][0]=e[i][0];        // write plus errors to result
    else
        eplus[i][0]=0;

    if (e[i][0]<0)
        eminus[i][0]=e[i][0];       // write minus errors to result
    else
        eminus[i][0]=0;

    MeanT[i]=Trend;                  // write Trend to result
    }

// print ("CN without correction=", CN);

// *****To correct for mean of absolute errors *****

MeanEcorrect=meanc(e[0]);

for (j=0; j<np; j++)
    {
        for (i=0; i<fmax-fmin; ++i)
            {

```

```

if ((CN[i][0]+MeanEcorrect+st[j][0])>0) // -|
C[i][j]=(CN[i][0]+MeanEcorrect+st[j][0]); // |write forecasted values to result
else // |
C[i][j]=0; // -|

e[i][j]=actual[i]-C[i][j]; // write absolute errors to matrix "e"

if (e[i][j]>=0)
{eplus[i][j]=e[i][j]; // write plus errors to the matrix "eplus"
eminus[i][j]=0;}
else
{eplus[i][j]=0;
eminus[i][j]=e[i][j];} // write minus errors to the matrix "eminus"
}

// print ("C+correct+st", C[][j]);
// print ("meanc(C+correct+st)", meanc(C[][j]));
// print ("st[j][0]",st[j][0], "\n");
// print ("st[j][1]",st[j][1],"\n");
// print ("meanc(C+correct+st)/MF", (meanc(C[][j]))/MF,"\n");
//

// *****Formulate the Estimators *****

Dplus[j][0]=st[j][0];
Dplus[j][1]=(meanc(eplus[][j]))/MF;
Dplus[j][2]=st[j][1];

Dminus[j][0]=st[j][0];
Dminus[j][1]=fabs((meanc(eminus[][j]))/MF);
Dminus[j][2]=st[j][1];

SNR[j][0]=st[j][0];
SNR[j][1]=meanc(pow((e[][j]), 2))/meanc(pow((actual[]-MF),2));
SNR[j][2]=st[j][1];

MeanC[j][0]=st[j][0];
MeanC[j][1]=(meanc(C[][j]))/MF;
MeanC[j][2]=st[j][1];
}

// *****Print of Results (with corrections for mean of forecast)*****

// print("C=", C);
// print ("e=", e);
// print ("eplus=", eplus);
// print ("eminus=", eminus);
// print ("Dplus=", Dplus, "\n");
// print ("Dminus=", Dminus, "\n");
// print ("SNR=", SNR);
// print ("MF=", MF);
// print ("MeanC=", MeanC);

// *****Save results into files*****

savemat("Dplus_farima.mat", Dplus, 1);
savemat("Dminus_farima.mat", Dminus, 1);
savemat("SNR_farima.mat", SNR, 1);
savemat("MeanC_farima.mat", MeanC, 1);
savemat("D.mat", DAR, 1);

delete arfima;
}

```

**Описание и листинги программ для сетевого симулятора ns-2,
предназначенных для моделирования механизма динамического
распределения пропускной способности канала с прогнозированием**

Программа **vega.tcl** (ее листинг можно увидеть ниже), разработанная F.Vega, преобразует файл с реализацией сетевого трафика в вид, необходимый для его использования в качестве источника трафика в ns-2. Программа написана также на языке Otcl. В качестве исходного файла – используется текстовый файл, в котором в ASCII – формате записана реализация трафика в виде “временная метка (сек) размер пакета (байт)”, то есть:

```
0.516445 94
0.517346 60
0.520313 666
...
0.530747 74
0.532295 74
```

При этом данный файл должен находиться в той же директории, что и программа **vega.tcl** и называться **Weth04.dat** (для того варианта программы, который приведен в приложении). Тогда получившийся файл в “бинарном” виде, пригодный для использования в ns-2, будет иметь имя **bk.dat**. Модуль **vega.tcl** запускается обычным образом, как любой Otcl-сценарий, из командной строки.

Листинг программы vega.tcl:

```
#this script changes a file in text format to
#the binary format that NS-2 uses.
#felix vega
#jfx@007mundo.com
#
#the text file is
#
#time1 longpacket1
#time2 longpacket2
#time3 longpacket3
#time4 longpacket4
#
#
#the Binary file MUST be
```

```

#
#time1-time2 longpacket1
#time2-time3 longpacket2
#time3-time4 longpacket3
#time4-time5 longpacket4
#etc.... but in binary format
#the time MUST be in microseconds
#check the NS-2 manual for further information

#Text file = Weth04.tl
set original_file_name Weth04.tl

#Binary NS-2 file BC2.dat (destination file)
set trace_file_name bk.dat

set original_file_id [open $original_file_name r]
set trace_file_id [open $trace_file_name w]

set last_time 0
set k 0
fconfigure $trace_file_id -encoding binary
fconfigure $trace_file_id -translation binary

while {[eof $original_file_id] == 0} {
  gets $original_file_id current_line
  if {[string length $current_line] == 0 ||
      [string compare [string index $current_line 0] "#" ] == 0} {
    continue
  }
  scan $current_line "%g%u" next_time length
  set time [expr int(1000000*(($next_time-$last_time))]

  set last_time $next_time
  puts -nonewline $trace_file_id [binary format "I" $time $length]
  #puts $trace_file_id "$last_time $length"
  puts "$last_time $length"
}

close $original_file_id
close $trace_file_id

#-----
#end of script

```

Модуль **nsrun.tcl** разрабатывался нами специально для настоящей работы и служит для управления основной программой **tp-sim.tcl**, которую мы рассмотрим чуть ниже. Приведем текст программы и дадим некоторые пояснения. Строки, начинающиеся со знака “#”, являются комментариями.

Листинг программы **nsrun.tcl**:

```

# FILE: NSRUN.TCL
# AUTHOR: VITALY PETROFF (vpetroff@nm.ru)
# DATES: SEPTEMBER 2004
# NOTES: INVESTIGATION OF TRAFFIC PREDICTION ALGORITHM
#        SHELL SCRIPT FOR BASIC PROGRAM "tp-sim.tcl"
#=====

# VERIFICATION THE NUMBER OF ARGUMENTS
if {$argc != 7} {
  # Must get a 7 argument or program fails.
  puts "ERROR! ns called with wrong number of arguments!($argc)"
  puts ""
}

```

```

puts "argument 1 - max mean bandwidth for the experiment (in part of MF)"
puts "argument 2 - number points of graphs (number of iterations)"
puts "argument 3 - number of Pareto-sources"
puts "argument 4 - time of simulation, sec"
puts "argument 5 - sample time (aggregation time), sec"
puts "argument 6 - if switch1=0 then static bandwidth is used "
puts "                - if switch1=1 then naive prediction bandwidth is used"
puts "argument 7 - the hurst parameter for self-similar traffic (from n1): hurst=(3-shape)/2"
puts ""
puts "for example ns nsrun.tcl 3 10 30 10 0.1 1 0.9"

puts ""
puts ""
exit 1
} else {

```

Программа запускается, как и любой Otcl-сценарий, из командной строки и имеет 7 параметров:

1 – максимальное значение средней пропускной способности канала N2-N3 в долях от среднего (MF) прогнозируемого участка. Фактически это максимальное для данного эксперимента значение величины bs_norm (см. 3.16);

2 – количество значений, которое принимает величина bs_norm за все время эксперимента (то есть количество точек на графиках SNR^{-1} , D^+ и D^-);

3 – количество источников с распределением On-Off интервалов по закону Парето. В данной программе существует возможность использовать в качестве источника трафика не внешние данные, а генератор самоподобного трафика, основанный на мультиплексировании некоторого количества источников, периоды активности (On) и неактивности (Off), которых подчинены закону Парето. Данный источник хоть и вырабатывает самоподобный процесс, однако этот процесс вряд ли можно использовать в качестве адекватной модели реального трафика. В частности, он обладает значительно меньшей пачечностью. Поэтому в дальнейшем исследовании мы будем пользоваться реальным трафиком, а значение этого параметра для нас в данном случае не имеет значения;

4 - время моделирования в секундах. Обратим внимание, что этот параметр должен иметь значение не больше, чем продолжительность реализации трафика, которую мы используем в качестве источника. Например, для реализации VC-Oct89Ext.TL, а точнее ее фрагмента, который мы прогнозировали в главе 3 настоящей диссертации длительностью 3000 отсчетов

по 10 секунд каждый (для ВС-10), время моделирования не должно превышать 30000 секунд;

5 – уровень агрегирования Δ в секундах. Фактически эта величина соответствует параметру T_c “корзины маркеров”. Через каждый интервал Δ (секунд) происходит контроль интенсивности трафика за прошедший участок времени Δ , прогнозирование и установка пропускной способности каналов в следующем интервале времени Δ ;

6 – переключатель, соответствующий выбору между простым предсказателем и статическим заданием полосы;

7 – параметр, влияющий на степень самоподобия генератора трафика, базирующего на On-Off – источниках. Как уже отмечалось ранее, в данном эксперименте этот генератор не используется, поэтому значение данного параметра не принципиально.

```
# SET SOME PARAMETERS
set alpha [lindex $argv 0];      # alpha value/MF (max static bandwidth for the experiment)
set np [lindex $argv 1];        # np value - number of iterations (points of graphs)
set N [lindex $argv 2];         # number of Pareto-sources
set time_sim [lindex $argv 3];  # time of simulation
set time_agg [lindex $argv 4];  # sample time (aggregation time)
set switch1 [lindex $argv 5];   # if switch1=0 then static bandwidth is used
                                # if switch1=1 then naive prediction bandwidth is used
set hurst [lindex $argv 6];     # the hurst parameter for self-similar traffic (from n1): hurst=(3-shape)/2
```

Здесь выводится некоторая информация о выбранной конфигурации:

```
#OUTPUT SOME INFORMATION ABOUT SIMULATION
puts ""
puts "NOTION: file tp-sim.log contains log-information from execution basic program"
puts ""
puts "NOTION: output files: out_Dminus.tr, out_Dplus.tr, out_SNR.tr, out_Dplus_drops.tr, out_CBR.tr, out_MEANBand23.tr"
puts ""

if {$switch1==0} {
puts "   NOTION: ***static bandwidth is used****"
puts ""
}

if {$switch1==1} {
puts "NOTION: ***naive bandwidth prediction is used****"
puts ""
}
```

Для того чтобы сравнивать возможности алгоритма динамического управления с методом статического задания пропускных способностей каналов оценочные статистики SNR^{-1} , D^+ и D^- необходимо сопоставлять для одних и тех же (в

среднем) значений пропускных способностей, которые, в свою очередь удобно измерять в долях среднего исследуемого участка ряда. Таким образом, для начала необходимо оценить среднее MF этого участка. Кроме того, если при статическом методе среднее значение пропускной способности некоторого канала соответствует фактически заданной для него пропускной способности, то для случая простого предсказания это не так. В случае простого предсказания средняя пропускная способность не равна среднему прогнозируемому участку, как это может показаться на первый взгляд. Существует некоторое отклонение (ME) в ту или иную сторону. Поэтому следующий участок программы (его результаты записываются в файл tp-sim-correct.log) оценивает среднее значение (MF) исследуемого участка ряда и оценивает отклонение (ME_correction) среднего значения пропускной способности канала от MF при прогнозировании с помощью простого предсказателя (значения этих параметров записываются в файлы out_ME.tr и out_MF.tr):

```
#CORRECTION FOR ME VALUE
puts "...correction for ME procedure is running..."
puts ""

exec ns tp-sim.TCL $N 0.0 $time_sim $time_agg $switch1 $shurst 0.0 0.0 > tp-sim-correct.log
set f7 [open out_ME.tr r]; #open file for reading value ME
set f8 [open out_MF.tr r]; #open file for reading value MF
set f9 [open out2.tr r]; #open file for reading traffic rate 1-2
set f10 [open out_traffic_rate1_2.tr w]; #open file for writing traffic rate 1-2

set ME_correction [read $f7]
set MF [read $f8]
puts $f10 [read $f9]; #Write data from out2.tr to the out_traffic_rate1_2.tr

puts "ME_correction= : $ME_correction"

puts "MF= : $MF"
puts ""
```

Далее (с учетом поправки ME_correction) происходит непосредственно моделирование и оценка статистик: SNR^{-1} (SNR), D^+ (Dplus и Dplus_drops), а также D^- (Dminus), результаты которого записываются в файл tp-sim.log. Поправка ME_correction в программе tp-sim.tcl используется для вычисления статически задаваемой компоненты пропускной способности каналов.

```
#EXECUTION BASIC PROGRAM REQUIRED QUANTITY TIMES
puts "...basic program is running..."
puts ""
```



```

# CREATE OUTPUT FILES AND CLEAR THEIR CONTENT
set f3 [open out_Dplus.tr w];          #Dplus
set f4 [open out_Dminus.tr w];        #Dminus
set f5 [open out_SNR.tr w];           #SNR
set f6 [open out_Dplus_drops.tr w];   #Dplus_drops
set f7 [open out_CBR.tr w];           # ALL RECIEVED ON SINK0 CBR TRAFFIC, BYTES
set f8 [open out_MEANBand23.tr w];    # Mean of bandwidth on link n2-n3, Mbps

set beta [expr $alpha*$MF/$np]

```

Цикл вызова программы `tp-sim.tcl` и оценки необходимых статистик выполняется заданное (параметром № 2 к модулю `nsrun.tcl`) количество раз:

```

for {set j 0} {$j<$np} {incr j} {
exec ns tp-sim.tcl $N [expr $j*$beta] $time_sim $time_agg $switch1 $hurst $SME_correction $MF > tp-sim.log
puts "# iteration: [expr $j+1]          static bandwidth : [expr ($j*$beta)/$MF] of MF"
}

```

После этого происходит вызов ПО `nam`, которое служит для визуализации эксперимента (последней конфигурации, которая фиксируется после выполнения цикла, приведенного выше):

```

#STARTING NAM FOR LAST USING PARAMETERS
exec nam out.nam &
}

```

В результате мы имеем следующий набор файлов:

- `out_Dplus.tr` – содержит статистику D^+ ;
- `out_Dminus.tr` – содержит статистику D^- ;
- `out_SNR.tr` – содержит статистику SNR^{-1} ;
- `out_Dplus_drops.tr` – содержит статистику D^+ , собранную особым образом (см. ниже);
- `out_CBR.tr` – содержит количество информации, принятое узлом N_4 от узла N_0 за все время эксперимента;
- `out_MEANBand23.tr` – содержит измеренное среднее значение пропускной способности канала N_2-N_3 .

Каждый из файлов представляет собой данные в ASCII-формате в виде:

- 1 столбец – среднее значение пропускной способности канала N_2-N_3 (в абсолютных единицах) за минусом MF ;
- 2 столбец – значение исследуемого параметра;
- 3 столбец – среднее значение пропускной способности канала N_2-N_3 за минусом MF , в единицах MF (т. е. это фактически значение `bs_norm` (см. 3.16)).

Поскольку ns-2 позволяет считать отброшенные пакеты, то статистика D^+ была вычислена двумя способами: во-первых, тем, которым она вводилась в главе 3 (данные содержатся в файле out_Dplus.tr), и, во-вторых, с помощью встроенного в ns-2 счетчика отброшенных пакетов (данные содержатся в файле out_Dplus_drops.tr).

В результате мы получаем те же самые (или более полные) статистики, которые исследовали в главе 3, и теперь можем их сравнить.

Теперь подробно рассмотрим основные функциональные блоки программы **tp-sim.tcl**. Конфигурация сценария представлена на рис. П2.1. Источники N100...Nxxx являются составляющими генератора самоподобного трафика, основанного на принципе мультиплексирования On-Off-источников (в настоящей конфигурации выключены). Интенсивность трафика, поступающего из внешнего источника на узел N5, анализируется на участке N1-N2, после чего происходит прогнозирование и установка в предсказанное значение пропускной способности канала N2-N3, а также канала N0-N3.

Нетрудно видеть, что данная схема фактически реализует конфигурацию, описанную с помощью рис. 4.11.

Листинг программы **tp-sim.tcl**:

```
# FILE: TP-SIM.TCL
# AUTHOR: VITALY PETROFF (vpetroff@nm.ru)
# DATES: SEPTEMBER 2004
# NOTES: INVESTIGATION OF TRAFFIC PREDICTION ALGORITHM
#=====

set ns [new Simulator]

# VERIFICATION THE NUMBER OF ARGUMENTS
if {$argc != 8} {
    # Must get a 8 argument or program fails.
    puts "ERROR! ns called with wrong number of arguments!($argc)"
    puts ""
    puts "argument 1 - number of Pareto-sources"
    puts "argument 2 - static component of bandwidth, bps"
    puts "argument 3 - time of simulation"
    puts "argument 4 - sample time (aggregation time)"
    puts "argument 5 - if switch1=0 then static bandwidth is used"
    puts "          - if switch1=1 then naive prediction bandwidth is used"
    puts "argument 6 - the hurst parameter for self-similar traffic (from n1): hurst=(3-shape)/2"
    puts "argument 7 - ME value, bps (for correction procedure)"
    puts "argument 8 - MF value, bps (needed for static bandwidth allocation only)"
    puts ""
    puts "for example ns tp-sim.tcl 30 1000.0 10.0 0.1 1 0.9 0.0 0.0"
    puts ""
}
```

```
puts ""
exit 1
} else {
```

Данная программа также имеет возможность запускаться из командной строки с параметрами.

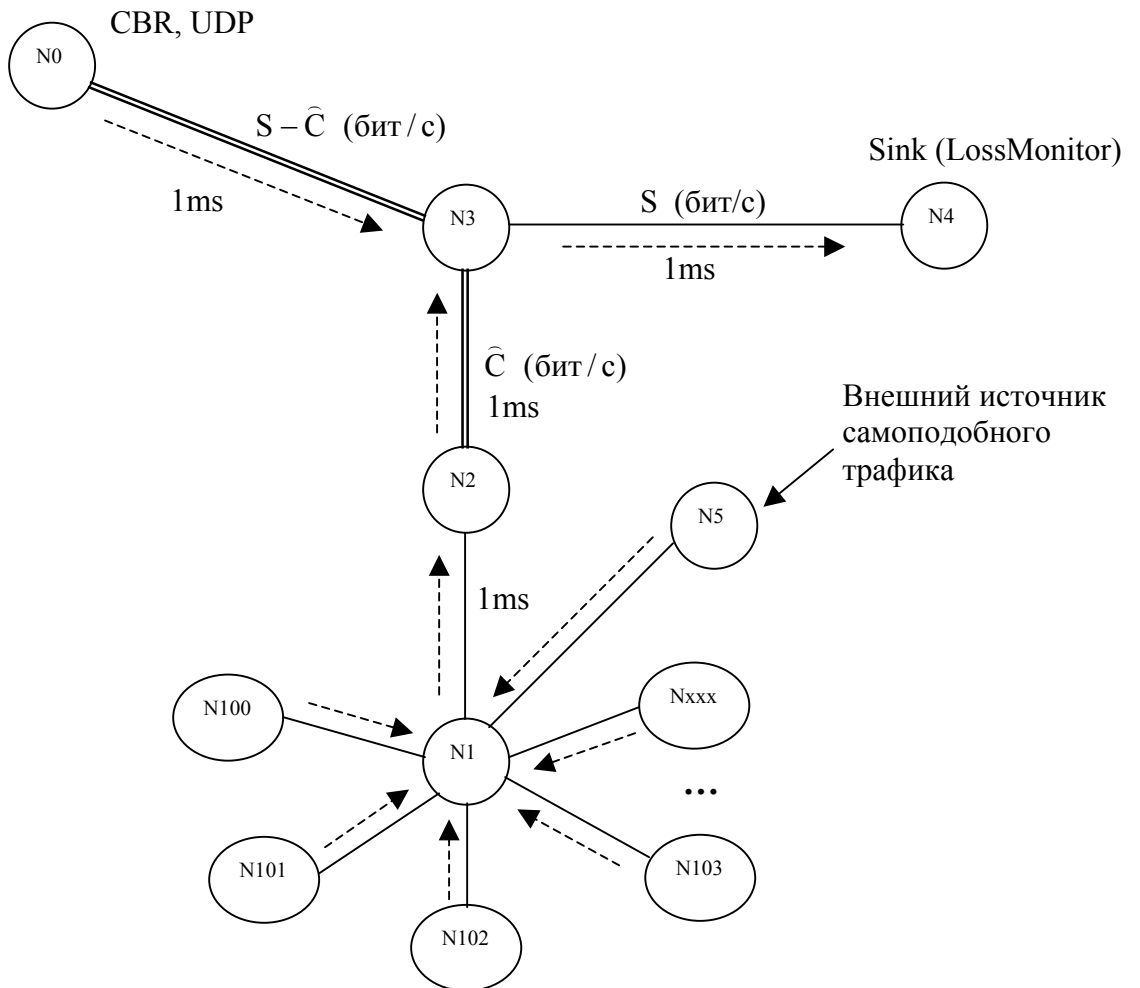


Рис. П2.1. Конфигурации сети, моделируемой в программе tp-sim.tcl

Эти параметры несколько отличаются от рассмотренных выше и передаются в данную программу из программного модуля nsrun.tcl. В частности,

1 – количество источников распределением On-Off интервалов по закону Парето (в данном случае неважно);

2 – статическая компонента bs пропускной способности канала. В дальнейшем статическая компонента будет скорректирована на значение поправки ME_correction;

3 – время моделирования;

4 – интервал агрегирования;

5 – переключатель режима: статическое задание, либо простое прогнозирование пропускной способности;

6 – параметр Херста для (On-Off)-генератора;

7 – значение поправки ME;

8 – среднее значение MF исследуемого участка трафика.

Два последних параметра предварительно вычисляются с помощью процедуры оценки в программе nsrun.tcl.

```
# SET SOME PARAMETERS
set N [index $argv 0];      # number of Pareto-sources
set st [index $argv 1];    # static component of bandwidth, bytes
set time_sim [index $argv 2]; # time of simulation
set time_agg [index $argv 3]; # sample time (aggregation time)
set switch1 [index $argv 4]; # if switch1=0 then static bandwidth is used
                             # if switch1=1 then naive prediction bandwidth is used
set hurst [index $argv 5]; # the hurst parameter for self-similar traffic (from n1): hurst=(3-shape)/2
set ME_correction [index $argv 6]; # ME value, bytes (for correction procedure)
set MF [index $argv 7];    # MF value, bytes (needed for static bandwidth allocation only)

set shape [expr 3-2*$hurst]; # hurst parameter is connected with shape-parameter from Pareto application
```

Создание 5 узлов:

```
#Create 5 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
```

Создание N узлов, предназначенных для On-Off-генератора:

```
#CREATE N TRANSMISSION NODES, 1 MUX NODE
for {set i 0} {$i<$N} {incr i} {
    set n([expr $i+100]) [$ns node]
}
```

Создание файла моделирования (out.tr) и файла (out.nam), необходимого для функционирования ПО nam (в данной конфигурации не используется в целях экономии времени):

```
#Open the Nam trace file
# set nf [open out.nam w]
# $ns namtrace-all $nf
```

```
#Open the Trace file
# set tf [open out.tr w]
# $ns trace-all $tf
```

Создание файлов для записи результатов эксперимента:

```
#OPEN OUTPUT FILES
set f1 [open out1.tr w+]
set f2 [open out2.tr w]; # Traffic on n1-n2
set f3 [open out_Dplus.tr a+]; # Dplus
set f4 [open out_Dminus.tr a+]; # Dminus
set f5 [open out_SNR.tr a+]; # SNR
set f6 [open out_Dplus_drops.tr a+]; # Dplus_drops
set f7 [open out_ME.tr w]; # Open file for writing value ME
set f8 [open out_MF.tr w]; # Open file for writing value MF
set f9 [open out_CBR.tr a+]; # ALL RECIEVED ON SINK0 CBR TRAFFIC, BYTES
set f10 [open out_MEANBand23.tr a+]; # Mean of bandwidth on link n2-n3, Mbps
```

Создание соединений и инициализация их основных параметров (пропускной способности и задержки):

```
#Create links

set max_band2_3 10000000.0; #The max value of bandwidth 2-3
set max_band0_3 10000000.0; #The max value of bandwidth 0-3

$ns simplex-link $n0 $n3 10000000.0 1ms DropTail
$ns simplex-link $n1 $n2 10000000.0 1ms DropTail
$ns simplex-link $n2 $n3 0.0 1ms DropTail
$ns simplex-link $n3 $n4 10000000.0 1ms DropTail
$ns simplex-link $n5 $n1 10000000.0 1ms DropTail

# set link [$ns simplex-link $n2 $n3 0 1ms DropTail]

# qrd [[ $ns link $n1 $n2] queue]
```

Установка объектов qmon12 и qmon23, позволяющих регистрировать пакеты, прошедшие и отброшенные на связях N1-N2 и N2-N3 соответственно:

```
set qmon12 [$ns monitor-queue $n1 $n2 ""]
set qmon23 [$ns monitor-queue $n2 $n3 ""]

for {set i 0} {$i<$N} {incr i} {
$ns simplex-link $n([expr $i+100]) $n1 1000000 1ms DropTail
}
```

Установка размеров буферов. Поскольку в настоящем эксперименте имитируется алгоритм динамического распределения пропускной способности с прогнозированием без буферизации (см. рис.4.10), буферы во всех каналах устанавливаются в минимально-возможное здесь значение – 2 пакета. Поскольку пропускная способность канала N3-N4 в каждый момент времени соответствует сумме пропускных способностей каналов N2-N3 и N0-N3, разделяясь между ними в нужных пропорциях, то в канале N3-N4 перегрузки

исключены, а, значит, практически исключены буферизация и отброс пакетов. Поэтому размер буфера в канале N3-N4 не имеет большого значения:

```
#Set the buffer size on the links
set buffer_size 2
$ns queue-limit $n0 $n3 $buffer_size
$ns queue-limit $n1 $n2 $buffer_size
$ns queue-limit $n2 $n3 $buffer_size
$ns queue-limit $n3 $n4 100
$ns queue-limit $n5 $n1 $buffer_size
for {set i 0} {$i<$N} {incr i} {
$ns queue-limit $n([expr $i+100]) $n1 $buffer_size
}
```

```
#Set the orientations of links
$ns simplex-link-op $n0 $n3 orient right-down
$ns simplex-link-op $n1 $n2 orient right-up
$ns simplex-link-op $n2 $n3 orient right-up
$ns simplex-link-op $n3 $n4 orient right
$ns simplex-link-op $n3 $n4 queuePos 0.5
$ns simplex-link-op $n0 $n3 queuePos 0.5
$ns simplex-link-op $n1 $n2 queuePos 0.5
$ns simplex-link-op $n2 $n3 queuePos 0.5
```

Создание агентов протокола UDP и ассоциирование их с узлами N0 и N5:

```
#Create a UDP agents
set udp0 [new Agent/UDP]
$udp0 set fid_ 1
$ns attach-agent $n0 $udp0

set udp5 [new Agent/UDP]
$udp5 set packetSize_ 1500
$udp5 set fid_ 2
$ns attach-agent $n5 $udp5
```

Создание агентов (типа LossMonitor), принимающих и нейтрализующих пакеты на узле N4. Создание виртуального соединения N0-N4:

```
#CREATE SINK-AGENTS (LOSSMONITOR)
set sink0 [new Agent/LossMonitor]
$ns attach-agent $n4 $sink0
$ns connect $udp0 $sink0

set sink1 [new Agent/LossMonitor]
$ns attach-agent $n4 $sink1
```

Создание генератора CBR- трафика, вырабатывающего пакеты размером 128 байт со скоростью 10 Мбит/с, и ассоциирование его с узлом N0:

```
#CREATE CBR TRAFFIC SOURCE AND ATTACH IT TO UDP0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 128
$cbr0 set rate_ 10000000.0
$cbr0 attach-agent $udp0
```

Подпрограмма создания (On-Off)-источников с распределением On-Off-интервалов по закону Парето:

```
proc attach-pareto-traffic { source node sink } {
global shape

#Get an instance of the simulator
set ns [Simulator instance]
```

```

$ns attach-agent $node $source

#Create an Pareto traffic agent and set its configuration parameters
set traffic [new Application/Traffic/Pareto]
$traffic set packetSize_ 128
$traffic set burst_time_ 50ms
$traffic set idle_time_ 50ms
$traffic set rate_ 10000.0
$traffic set shape_ $shape

# Attach traffic source to the traffic generator
$traffic attach-agent $source
#Connect the source and the sink
$ns connect $source $sink
return $traffic
}

```

Создание источника трафика, использующего внешнюю реализацию из бинарного файла `bk.dat`, и ассоциирование его с узлом N5. Создание виртуального соединения N5-N4:

```

#CREATE TRAFFIC SOURCE FROM FILE
Tracefile set debug_ 0
set ss_file [new Application/Traffic/Trace]
set trace [new Tracefile]
$trace filename bk.dat
$ss_file attach-tracefile $trace
$ss_file attach-agent $udp5
$ns connect $udp5 $sink1

```

Создание генератора самоподобного трафика на основе On-Off-источников, используя вышеописанную подпрограмму *attach-pareto-traffic*:

```

#CREATE TRAFFIC SOURCES
for {set i 0} {$i<$N} {incr i} {
set source([expr $i+100]) [new Agent/UDP]
$source([expr $i+100]) set fid_ 2
set traffic([expr $i+100]) [attach-pareto-traffic $source([expr $i+100]) $n([expr $i+100]) $sink1]
}

```

Установка цветов потоков (удобно при визуализации процессов в ПО nam):

```

#SET THE COLOR OF FLOWS
$ns color 1 blue
$ns color 2 red

```

Подпрограмма учета потерянных и принятых пакетов:

```

#PROCEDURE RECORD FOR CALCULATION THE TRAFFIC AND DROP RATES
proc record12 {} {
global sink0 sink1 qmon12 time_agg t_rate_1_2 now band2_3 band0_3 rate_udp1_sink rate_udp0_sink
global allbytes_n2 drop_rate_1_2 N allbytes_04

#Get an instance of the simulator
set ns [Simulator instance]

$qmon12 instvar bdrops_ bdepartures_

#Calculate the rate udp0 (in Bit/s) with help sink
set rate_udp1_sink [expr [$sink1 set bytes_]/$time_agg*8]

#Calculate the rate udp0 (in Bit/s) with help sink
set rate_udp0_sink [expr [$sink0 set bytes_]/$time_agg*8]

#Calculate the rate (in Bit/s) with queue-monitor
set t_rate_1_2 [expr $bdepartures_/$time_agg*8]
}

```

```

#Calculate the drop rate (in Bit/s) with queue-monitor
set drop_rate_1_2 [expr $bdrops_/ $time_agg*8]

#Calculate summ of all bytes received on n2
set allbytes_n2 [expr $allbytes_n2+$bdepartures_]

#Calculate the quantity bytes on the sink0
set allbytes_04 [expr $allbytes_04+$rate_udp0_sink*$time_agg/8]

#Reset the bytes_ values on the traffic sinks

$sink0 set bytes_ 0
$sink1 set bytes_ 0
$qmon12 set bdepartures_ 0
$qmon12 set bdrops_ 0
}

```

Еще одна подпрограмма вычисления принятых и отброшенных пакетов, а также вычисления основных статистик и вывода промежуточных результатов:

```

#PROCEDURE RECORD FOR CALCULATION THE TRAFFIC AND DROP RATES AND WRITING THEIRS TO THE OUTPUT FILE
proc record23 {} {
    global f1 f2 qmon23 time_agg rate_udp1_sink rate_udp0_sink drop_rate_1_2 drop_rate_2_3 band0_3 i time_sim
    global alldrops_23 now t_rate_1_2 t_rate_2_3 band2_3 allbytes_n2 Dplus_drops sub allbytes_04
    global band23_array band23_array e_array tr_array eminus_23_array eplus_23_array Dplus_drops

    $qmon23 instvar bdrops_ bdepartures_

    #Calculate the drop rate on link n2-n3 (in Bit/s) with queue-monitor
    set drop_rate_2_3 [expr $bdrops_/ $time_agg*8]

    #Calculate the rate on link n2-n3 (in Bit/s) with queue-monitor
    set t_rate_2_3 [expr $bdepartures_/ $time_agg*8]

    #Calculate summ of all drops on link n2-n3
    set alldrops_23 [expr $alldrops_23+$bdrops_]

#    #Calculate Dplus
    set Dplus_drops [expr $alldrops_23/$allbytes_n2]

    #Calculate errors of overestimation (eminus)
    if {[expr $t_rate_1_2-$band2_3]<0 & [expr $now!=0.0]} {
        set eminus_23 [expr abs($t_rate_1_2-$band2_3)]
    } else {
        set eminus_23 0.0
    }

    #Calculate errors of underestimation (eplus)
    if {[expr $t_rate_1_2-$band2_3]>0 & [expr $now!=0.0]} {
        set eplus_23 [expr $t_rate_1_2-$band2_3]
    } else {
        set eplus_23 0.0
    }

    #Calculation absolute error
    if {[expr $now!=0.0]} {
        set err [expr $t_rate_1_2-$band2_3]
    } else {
        set err 0.0
    }

    #Write array of eplus_23
    set eplus_23_array($i) $eplus_23

    #Write array of eminus_23
    set eminus_23_array($i) $eminus_23

    #Write array of errors
    set e_array($i) $err

    #Write array of t_rate_1_2
    set tr_array($i) $t_rate_1_2

    #Write array of band2_3

```



```

set band23_array($i) $band2_3

#Display the obtained data
puts "# Data from procedures RECORD12 & RECORD23:"
puts "# - now time, sec                               : $now"
puts "# - traffic rate udp1 on sink, bps               : $rate_udp1_sink"
puts "# - traffic rate udp0 on sink, bps               : $rate_udp0_sink"
puts "# - traffic rate udp1 on n2-n3, bps              : $t_rate_2_3"
puts "# - traffic rate udp1 on n1-n2, bps              : $t_rate_1_2"
puts "# - traffic drop rate udp1 on n1-n2, bps         : $drop_rate_1_2"
puts "# - traffic drop rate udp1 on n2-n3, bps         : $drop_rate_2_3"
puts "# - bandwidth n2-n3, bps                         : $band2_3"
puts "# - bandwidth n0-n3, bps                         : $band0_3"
puts "# - sum of all bytes, received on sink from cbr, bytes : $allbytes_04"
# puts "# - summ of drops on link n2-n3, bytes         : $alldrops_23"
# puts "# - summ of all bytes received on n2, bytes     : $allbytes_n2"
# puts "# - Dplus_drops                                 : $Dplus_drops"
# puts "# - eminus_23                                     : $eminus_23"
# puts "# - alleminus_23                                  : $alleminus_23"
# puts "# - Dminus                                         : $Dminus"
# puts "# - earray                                         : $e_array($i)"
# puts "# - tr_array                                       : $tr_array($i)"
puts ""

#Write traffic rates to the file
puts $f1 "$now      $t_rate_1_2      $rate_udp1_sink      $drop_rate_1_2      $drop_rate_2_3      $band2_3 "
puts $f2 "$now      $t_rate_1_2"

#Reset the bytes_ values on the traffic sinks
$qmon23 set bdrops_0
$mqmon23 set bdepartures_0

set i [expr $i+1]
}

#PROCEDURE FOR TUNING BANDWIDTH
proc verification {} {
    global rate_udp1_sink t_rate_2_3 t_rate_1_2 drop_rate_2_3

    #Display the obtained data
    puts "# Data from procedure VERIFICATION:"
    puts "# rate_udp1_sink-t_rate_2_3                    :[expr $rate_udp1_sink-$t_rate_2_3]"
    puts "# t_rate_1_2-drop_rate_2_3-t_rate_2_3            :[expr $t_rate_1_2-$drop_rate_2_3-$t_rate_2_3]"
    puts ""
}

```

Подпрограмма для вычисления требуемого значения пропускной способности канала N1-N2 на следующем интервале Δ (time_agg):

```

#PROCEDURE FOR PREDICTION TRAFFIC RATE
proc forecast {} {
    global t_rate_1_2 f_rate n2 n3 now st switch1 ME_correction MF

    #Get an instance of the simulator
    set ns [Simulator instance]

    if {$switch1==0} {
#
        #Static bandwidth

```

В случае статического задания пропускная способность равна сумме статически заданной компоненты st и среднего значения прогнозируемого участка ряда MF:

```

        puts "# ***static bandwidth is used****"
        puts ""
        set f_rate [expr $st+$MF]
    }

```

```

#         if {$switch1==1} {
#             #Naive prediction algorithm

```

В случае простого прогнозирования пропускная способность равна сумме статически заданной компоненты `st`, спрогнозированного значения интенсивности трафика `t_rate_1_2` и поправки `ME_correction`:

```

                puts "# ***naive bandwidth prediction is used****"
                puts ""
                if {$t_rate_1_2 > 0} {
                set f_rate [expr $t_rate_1_2+$st+$ME_correction]
                } else {
                set f_rate [expr 10000.0+$st+$ME_correction]
                }
            }

            if {$switch1!=1 & $switch1!=0} {
                puts "# error:      incorrect value of switch1!"
                puts ""
            }

            #Display the obtained data
            # puts "# Data from procedure FORECAST:"
            # puts "# - bandwidth n2-n3 for next step (f_rate), bps          : $f_rate"
            # puts ""
            # puts ""
        }
}

```

Подпрограмма непосредственной установки пропускных способностей связей N2-N3 и N0-N3 в вычисленные выше значения:

```

#PROCEDURE FOR TUNING BANDWIDTH
proc tune {} {
    global f_rate n1 n2 n3 n4 n0 now ns band3_4 t_rate band2_3 band0_3 time_agg max_band2_3 max_band0_3

    #Get the current time
    set now [$ns now]

    #Get an instance of the simulator
    set ns [Simulator instance]

    #Set the bandwidth
    if {$f_rate < $max_band2_3} {
        set band2_3 $f_rate
    } else {
        set band2_3 $max_band2_3
    }

    if {[expr $band3_4-$f_rate] > 0} {
        set band0_3 [expr 1.0*($band3_4-$f_rate)]
    } else {
        set band0_3 0
    }

    $ns at $now "$ns bandwidth $n2 $n3 $band2_3 simplex"
    $ns at $now "$ns bandwidth $n0 $n3 $band0_3 simplex"

    #Display the obtained data
    puts "# Data from procedure TUNE:"
    puts "# - bandwidth n2-n3 for next step, bps          : $band2_3"
    puts "# - bandwidth n0-n3 for next step, bps          : $band0_3"
    puts ""
    puts ""
    puts "-----"
    puts ""
    puts ""
}

```

Процедура вычисления окончательных статистик и формирования файлов с результатами:

```

#THE STATISTICS CALCULATION PROCEDURE
proc calc {} {
    global i e_array tr_array sub band23_array emin_23_array eplus_23_array f3 f4 f5 f6 f7 f8 f9 f10
    global allbytes_04 st Dplus_drops

    set summ_err_q 0.0
    set summ_err 0.0
    set summ_sub 0.0
    set summ_tr 0.0
    set mean_sub 0.0
    set summ_band23 0.0
    set allemin_23 0.0
    set alleplus_23 0.0

    #Calculate the M[e(k)**2]
    for {set j 0} {$j<=$i-1} {incr j} {
#       puts "# e_array(j)                :$e_array($j)"
#       puts "# pow(e_array(j),2)         :[expr pow($e_array($j),2)]"
        set summ_err_q [expr $summ_err_q+[expr pow($e_array($j),2)]]
        set ME2 [expr $summ_err_q/($i-1)]
    }
    puts "# - ME2                :$ME2"

    #Calculate the ME
    for {set j 0} {$j<=$i-1} {incr j} {
#       puts "# e_array(j)                :$e_array($j)"
        set summ_err [expr $summ_err+$e_array($j)]
    }
    set ME [expr $summ_err/($i-1)]
    puts $f7 "$ME"; #Write to file the current value of ME

    puts "# - mean of error (ME)    :$ME"

    #Calculate the MF - mean of t_rate_1_2
    for {set j 0} {$j<=$i-1} {incr j} {
#       puts "# tr_array(j)                :$tr_array($j)"
        set summ_tr [expr $summ_tr+$tr_array($j)]
    }
    set MF [expr $summ_tr/($i-1)]
    puts $f8 "$MF"; #Write to file the current value of MF

    puts "# - mean of t_rate_1_2 (MF) :$MF"

    #Calculate Dminus
    for {set j 0} {$j<=$i-1} {incr j} {
        set allemin_23 [expr $emin_23_array($j)+$allemin_23]
    }
    set Dminus [expr $allemin_23/($i-1)/$MF]
    puts "# - Dminus                :$Dminus"

    #Calculate Dplus
    for {set j 0} {$j<=$i-1} {incr j} {
        set alleplus_23 [expr $eplus_23_array($j)+$alleplus_23]
    }
    set Dplus [expr $alleplus_23/($i-1)/$MF]
    puts "# - Dplus                  :$Dplus"

    #Calculate the (x(k)-MF)**2
    for {set j 0} {$j<=$i-1} {incr j} {
#       set sub($j) [expr pow(($tr_array($j)-$MF),2)]
#       puts "# - sub                :$sub($j)"
    }

    #Calculate the mean(sub)
    for {set j 0} {$j<=$i-1} {incr j} {
        set summ_sub [expr $summ_sub+$sub($j)]
    }
    set mean_sub [expr $summ_sub/($i-1)]
#       puts "# - mean of sub                :$mean_sub"

    #Calculate the SNR
    set SNR [expr $ME2/$mean_sub]
}

```

```

puts "# - SNR          :$SNR"

#Calculate the mean of band2_3
for {set j 0} {$j<=$i-1} {incr j} {
set summ_band23 [expr $summ_band23+$band23_array($j)]
}
set mean_band23 [expr $summ_band23/($i-1)]
puts "# - mean of band23      :$mean_band23"

puts $f3 "$st $Dplus [expr $st/$MF]"
puts $f4 "$st $Dminus [expr $st/$MF]"
puts $f5 "$st $SNR [expr $st/$MF]"
puts $f6 "$st $Dplus_drops [expr $st/$MF]"
puts $f9 "$st $allbytes_04 [expr $st/$MF]"
puts $f10 "$st $mean_band23 [expr $st/$MF]"

}

```

Подпрограмма, управляющая последовательностью исполнения подпрограмм record12, record23, forecast, tune. Эта подпрограмма с помощью последней своей строки сама себя запускает через каждый интервал времени time_agg (то есть через Δ сек):

```

#PROCEDURE FOR SCHEDULING
proc schedule {} {
    global ns time_agg now

    #Get an instance of the simulator
    set ns [Simulator instance]
    #Get the current time
    set now [$ns now]

    #Re-schedule the procedure
    $ns at $now "record12"
    $ns at $now "record23"
#   $ns at $now "verification"
    $ns at $now "forecast"
    $ns at $now "tune"
    $ns at [expr $now+$time_agg] "schedule"
}

```

Процедура, завершающая выполнение программы tp-sim.tcl:

```

proc finish {} {
    global ns f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 f10
#   global nf tf

    $ns flush-trace
#   close $nf
#   close $tf
    close $f1
    close $f2
    close $f3
    close $f4
    close $f5
    close $f6
    close $f7
    close $f8
    close $f9
    close $f10

#   exec nam out.nam &
    exit 0
}

```

Процедура начальной инициализации некоторых параметров, необходимых для функционирования программы:

```

#START SETUP PROCEDURE
proc setup {} {
    global band2_3 band0_3 band3_4 alldrops_23 allbytes_n2 alleminus_23 i allbytes_04

    set band2_3 0.0
    set band3_4 10000000.0
    set band0_3 10000000.0
    set alldrops_23 1.0
    set allbytes_n2 1.0
    set alleminus_23 0.0
    set i 0
    set allbytes_04 0
}

```

Основной модуль программы, запускающий на старте и останавливающий по окончании все необходимые сервисы:

```

# MAIN PART
#starting:
    $ns at 0.0 "setup"
    $ns at 0.0 "$cbr0 start"
    $ns at 0.0 "$ss_file start"

```

On-Off- источники в данной конфигурации не включаются:

```

#   for {set i 0} {$i<$N} {incr i} {
#       $ns at 0.0 "traffice([expr $i+100]) start"
#   }
#   $ns at 0.0 "schedule"

#stopping:
    $ns at $time_sim "$cbr0 stop"
    $ns at $time_sim "$ss_file stop"
#   for {set i 0} {$i<$N} {incr i} {
#       $ns at $time_sim "traffice([expr $i+100]) stop"
#   }
    $ns at $time_sim "calc"
    $ns at $time_sim "finish"
    $ns run
}

```

УТВЕРЖДАЮ

Проректор по научной работе
ГОУВПО МЭИ (ТУ)
д.т.н., профессор



Серебрянников С. В.
Серебрянников С. В.
«15» декабря 2004 года

А К Т

использования в учебном процессе
результатов диссертационной работы В.В. Петрова
« Структура телетрафика и алгоритм обеспечения
качества обслуживания при влиянии эффекта самоподобия »

Настоящим актом подтверждается, что имитационная модель, разработанная в диссертационной работе В.В.Петрова « Структура телетрафика и алгоритм обеспечения качества обслуживания при влиянии эффекта самоподобия », представленной на соискание ученой степени кандидата технических наук, используется в демонстрационной лабораторной работе по дисциплине «Методы и устройства цифровой обработки сигналов», в которой показывается эффективность алгоритма динамического управления пропускной способностью с прогнозированием в условиях самоподобного телетрафика, имеющего место в современных компьютерных сетях.

Зам. Заведующий кафедрой РПУ МЭИ (ТУ)
д.т.н., профессор

Смольский С.М.



Федеральное государственное унитарное предприятие
«Ситуационно-Кризисный Центр
Министерства Российской Федерации по атомной энергии»

119017, г. Москва, ул. Большая Ордынка, д. 24/26
тел. (095) 933-60-40, факс (095) 933-60-41, e-mail skc@skc.ru

УТВЕРЖДАЮ

Первый заместитель директора
ФГУП «СКЦ Минатома России»



Э.Б.Наумов

2004 г.

Акт внедрения

результатов диссертационной работы Петрова Виталия Валерьевича

Настоящим актом подтверждается, что результаты диссертационной работы Петрова В.В. на тему **“Структура телеграфика и алгоритм обеспечения качества обслуживания при влиянии эффекта самоподобия”** использованы при проектировании и построении Ведомственной Технологической Сети Спутниковой Связи (ВТССС) для Министерства Российской Федерации по атомной энергии.

В частности, материалы данной работы вошли в научно-исследовательскую опытно-конструкторскую работу по теме: «Сопряжение периферийных земных станций спутниковой связи с абонентскими пунктами информационно-коммуникационной системы», целью которой являлось сокращение затрат на аренду частотно-энергетического ресурса спутника-ретранслятора “Ямал-200”.

В результате внедрения выработаны рекомендации и конкретные технологические решения по наиболее эффективному распределению сетевых ресурсов (частотно-энергетического ресурса спутника в том числе) и обеспечению качества обслуживания в ВТССС Минатома России.

Начальник службы эксплуатации ИКС

М.Е.Тимофеев

Заместитель начальника службы эксплуатации ИКС

О.Г.Подрезов