Федеральное государственное автономное образовательное учреждение
высшего образования «Уральский федеральный университет имени первого
Президента России Б.Н. Ельцина»
Институт радиоэлектроники и информационных технологий-РТФ
Кафедра радиоэлектроники информационных систем

На правах рукописи

Божалкин Даниил Александрович

Математическое и алгоритмическое обеспечение для анализа характеристик информационных потоков в магистральных интернет-каналах

Специальность 05.13.01 – Системный анализ, управление и обработка информации (информатизация и связь)

Диссертация на соискание ученой степени кандидата технических наук

Научный руководитель:
д.т.н., проф. Поршнев
Сергей Владимирович

Екатеринбург – 2020

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Актуальность темы исследования и степени ее разработанности

Сегодня компьютерные сети (КС) широко используются для передачи различной информации, контроля и управления различными сервисами в реальном времени, просмотра телепередач, онлайн покупок и т.д. В связи с увеличением новых классов телекоммуникационных устройств и соответствующих сервисов быстрыми темпами увеличиваются объемы информации, передаваемой через сеть Интернет (Интернет-трафик). Например, по данным Cisco Visual Networking Index (наглядные показатели Сети) объемы передаваемого Интернет-трафика увеличились со 100 Г/сутки в 1992 г. до 16 000 Гб/с в 2014 г. [84]. При этом существенно усложнилась собственно структура передаваемой информации, которая создается и используется многочисленными пользователями персональных компьютеров, смартфонов, планшетов, телевизоров, бытовой техникой (Интернет вещей) и др.

В этой ситуации закономерно возрастают требования к гибкости и масштабируемости современных КС свойства, которых оказываются существенно отличными от свойств КС с классической архитектурой (по сути, статических). Например, традиционные архитектуры/дизайны КС оказываются неэффективными в динамических средах. При этом классические подходы, ориентированные на распределенное управление устройствами традиционных КС (например, виртуальные сети (VLAN)), не соответствуют современному уровню развития виртуализации серверов и систем хранения данных, а также требованиям крупного бизнеса и сервис-провайдеров (например, AT&T, Verizon, Google, Facebook, Microsoft и др.). Сложившаяся ситуация в телекоммуникационной отрасли подтверждается, том числе, данными, содержащимися в аналитическом отчете за 2016 г. компании KPMG [10], где введены понятия «разрушающий трафик» (т.е. трафик таких объемов, с которыми не справляется используемое сетевое оборудование) и «разрушительные технологии», которые создают разрушающий трафик (виртуальная реальность, облачные сервисы, искусственный интеллект, анализ данных в реальном времени и др.).

Для эффективного решения проблем «разрушающего трафика», вопросов проектирования оборудования отвечающего потребностям современных КС, а так же проектирования КС нового поколения, в том числе и виртуальных программно-конфигурируемых сетей – SDN (Software-defined Networking), необходимо понимать особенности информационных потоков, предаваемых в современных КС, механизмы их взаимодействия друг с другом и влияния на загрузку канала.

Анализ состояния современной теории телетрафика показывает, что имеет место определенный разрыв между сегодня современным уровнем развития телекоммуникационных технологий и математическими описаниями информационных процессов в КС, который пытаются восполнить большим количеством результатов, проведенных

экспериментальных исследований особенностей информационных потоков в КС, в особенности, высокоскоростных магистральных Интернет-каналах (см., работы О.И. Шелухина [100,101], В.В. Петрова [75,76], Н.Г. Треногина [96], Е.В. Никуличева [105], M. Soysal, K. Fukuda, W. Leland, W. Willinger, D. Wilson и др.)

Однако объективный анализ этих работ показал, что проводимы экспериментальные исследования, зачастую, имеют бессистемный характер. Это проявляется: в отсутствии общепринятой методики исследований Интернет-трафика; их направленности не на проверку, но на подтверждение тех или иных популярных математических моделей Интернет-трафика (в первую очередь, самоподобных), кроме того:

– при проведении исследований используются устаревшие дампы Интернет-трафика, полученные в 90-х годах XX в. [75], на основе анализа которых далее идентифицируются параметры моделей Интернет-трафика, уже утративших свою актуальность [69,57] (в том числе: искусственный трафик, передаваемый в тех или иных модельных локальных вычислительных сетях (ЛВС) [101,64,96], трафик, синтезированный с помощью соответствующих программных инструментов, в которых реализованы классические математические модели: систем массового обслуживания (СМО) (SPSS [22]), on-off-модель [68], жидкостной модели (ЖМ) [60] и гибридной жидкостной модели (ГЖМ) [61]), а также программных генераторов трафика (NS-2, NS-3 [41]), свойства которых, как очевидно, существенного отличаются от свойств реальных информационных процессов, протекающих в корпоративных ЛВС или провайдерских магистралях);

– зачастую проводится раздельное изучение свойств информационных потоков, созданных различными типами источников Интернет-трафика, без учета взаимодействия данных потоков друг на друга (см., например [58]), что не позволяет достичь полного представления о процессах, протекающих в КС;

– в большинстве случаях анализ собранной экспериментальной информации проводится в соответствие со следующей схемой: выбор, зачастую субъективный, той или иной известной математической модели Интеренет-трафика и далее идентификация ее параметров, без проверки адекватности выбранной математической модели изучаемым информационным процессам, что подтверждает анализ многочисленных работ, посвященных исследованию самоподобных свойств Интерент-трафиках, число, по нашим оценка, сегодня уже привысило тысячу;

– не обсуждаются используемые технологии получения Интернет-трафика и инструменты его анализа;

– отсутствуют общепринятые методики и общедоступные инструменты для анализа сетевого трафика.

Отметим, что Интернет-трафик является сложным многомерным объектом, который можно изучать в различных измерениях (например, число и размеры передаваемых пакетов, объемы передаваемой информации и т.д.). В этой связи, очевидно, что первым и неотъемлемым этапом количественного анализа данных дампов является этап семантического анализа (парсинг) дампов Интернет-трафика (pcap-файлов), на котором из данных файлов извлекается необходимая количественная информация, используемая далее для получения количественных оценок характеристик трафика.

Таким образом, проведение экспериментальных исследований свойств информационных потоков в высокоскоростных магистральных Интернет-каналах на основе системного подхода является актуальным.

Объект исследования: информационные потоки в высокоскоростных магистральных Интернет-каналах.

Предмет исследования: свойства информационных потоков в высокоскоростных магистральных Интернет-каналах.

Цель работы: разработка и применение математического и алгоритмического обеспечения для анализа характеристик информационных потоков в высокоскоростных магистральных Интернет-каналах.

Для достижения поставленной цели были поставлены и решены следующие **основные задачи исследования**:

1. Анализ методов исследования информационных потоков в КС сетей с точки зрения их применимости для исследования трафика в высокоскоростных магистральных Интернет-каналах.
2. Разработка программного инструмента, обеспечивающего автоматическое извлечение информации из pcap-файлов в выбранном измерении.
3. Разработка методики анализа первичной информации, извлеченной в соответствующем измерении из pcap-файлов, обеспечивающей получение количественных характеристики информационных потоков, переданных в магистральном высокоскоростном Интернет-канале.
4. Изучение особенностей информационных потоков в магистральном Интернет-канале, создаваемых выбранными классами пользователей («Слоны», «Мулы», «Мыши»), и их взаимного влияния друг на друга.

Научная новизна полученных результатов

В диссертации получены следующие новые научные результаты:

1. Проведен анализ известных подходов к изучению Интернет-трафика, передаваемого в высокоскоростных магистральных Интернет-каналах, и выявлены их недостатки.

2. Разработано математическое и алгоритмическое обеспечение, а также соответствующее программное обеспечение (ПО) для анализа характеристик информационных потоков в высокоскоростных магистральных Интернет-каналах.

3. Предложена методика анализа Интернет-трафика и доказана ее работоспособность при исследовании информационных потоков в высокоскоростном магистральном Интернет-канале.

4. Проведено исследование взаимного влияния информационных потоков, создаваемых в магистральном Интернет-канале выбранными классами пользователей («Слоны», «Мулы», «Мыши»), и доказано, что связи между объемами информации, переданной в магистральном Интернет-канале каждым из выбранных классов пользователей, описываются детерминированными линейными моделями.

5. Предложен алгоритм управления загрузкой канала передачи информационных потоков позволяющая, за счет отслеживания глобального показателя Херста накопленных сумм случайных последовательностей объема переданной информации класса «Мыши» и соответствующего проактивного ограничения скоростей потоков класса «Слоны» и «Мыши», минимизировать кол-во сбросов скользящего окна для каждого потока, обеспечивая тем самым использование пропускной способности канала близкой к максимальной.

Теоретическая и практическая значимость работы

1. Создан программно-аппаратный комплекс, обеспечивающий анализ дампов Интернет-трафика, размещенных в pcap-файлов, адаптированный для использования на суперкомпьютере «Уран» Института математики и механики им. академика Н.Н. Красовского УрО РАН.

2. Проведен анализ дампов Интернет-трафика, зарегистрированных в магистральном Интернет-канала, проложенным между США и Японией, результаты которого подтвердили работоспособность разработанной методики анализа Интернет-трафика и созданного программно-аппаратного комплекса.

3. Предложен механизм балансировки объемов передаваемой информации, каждым из выделенных классов пользователей («Слоны», «Мулы», «Мыши»), устанавливающий скорость передачи информации для каждого класса пользователей, исходя из значений показателей Херста накопленных сумм зависимостей «мгновенного» числа переданных пакетов, «мгновенного» объема переданной информации, «мгновенного» объема информации, переданной одним пакетом, от времени. (Здесь и далее под «мгновенными» значения понимаются значения соответствующих параметров, подсчитанные в течение заданного временного интервала.)

Методология и методы исследования

В работе проведено экспериментальное исследование свойств информационных потоков в высокоскоростных магистральных Интернет-каналов на основе системного подхода с использованием методов математической статистики, генетических алгоритмов и методов анализа временных рядов.

Достоверность полученных результатов

Достоверность полученных результатов, научных положений и выводов, изложенных в диссертации, подтверждается использованием адекватных методов анализа первичной информации и выбранных количественных показателей процесса передачи данных в высокоскоростных магистральных интернет каналах и согласованностью полученных результатов с моделью OSI (Open Systems Interconnection – Модель взаимодействия открытых систем), технологией Ethernet (технология организации сетей) на базе которых построено большинство современных сетей, а также с результатами математического моделирования фрактального броуновского движения.

Научные положения, выносимые на защиту

1. Для количественного описания информационных потоков в высокоскоростных магистральных Интернет-каналах целесообразно использовать зависимости «мгновенного» числа переданных пакетов (N_i), «мгновенного» объема переданной информации (V_i), «мгновенного» объема информации, переданной одним пакетом $(\tilde{V}_i)$, от времени.

2. Зависимости N_i, V_i, $\tilde{V}_i$, представляющие собой случайные последовательности (СП), члены которых есть случайные величины с ограниченной областью рассеяния, вопреки устоявшимся представлениям, не являются самоподобными СП.

3. СП, вычисленные в соответствии с известными алгоритмами генерации ФБД, в которых вместо нормально распределенных случайных величин используются случайные величины ограниченной областью рассеяния, являются самоподобными.

4. Накопленные первые суммы ВР N_i, V_i, $\tilde{V}_i$ являются самоподобными СП.

Апробация работы

Материалы работы докладывались на следующих научных конференциях: Международной IEEE Сибирской конференция по управлению и связи SIBCON-2015, Омск, 21-23 мая 2015 г.; Международной Крымской конференции «СВЧ-техника и телекоммуникационные технологии» (КрыМи-Ко'2015), Севастополь, 6-12 сентября 2015; Международной IEEE конференции AICT, Ростов-на-Дону, 14-16 октября 2015 г.; Международная научная конференция «Современные методы прикладной математики, теории управления и компьютерных технологий», Воронеж, 20-26 сентября 2016 г; Международной научной

конференции «Динамика систем, механизмов и машин» Dynamics 2016, Омск 15-17 Ноября 2016г.; Международной IEEE конференции AICT, Москва, 20-22 сентября 2017г.

Публикации по теме диссертации

По теме диссертации опубликовано 14 работ, отражающих основные положения исследования, среди которых 5 статей в журналах, рекомендованных ВАК РФ и Аттестационным советом УрФУ, 4 статьи в изданиях, входящих в международные реферативные базы данных и системы цитирования, 2 свидетельства о государственной регистрации программы для ЭВМ (Приложения Е, Ж) и 3 статьи в других рецензируемых научных изданиях. В работах, опубликованных в соавторстве, лично соискателю принадлежат: разработка технологии семантического анализа дампа трафика информационных потоков в компьютерных сетях, исследование самоподобия трафика в высокоскоростной КС.

Структура и объем диссертации

Диссертация состоит из введения, четырех глав, заключения, списка литературы из 105 наименований, 7 приложений, содержит 42 рисунка и 55 таблиц. Основной текст работы составляет 143 страницы, общий объем – 198 страниц.

ГЛАВА 1. Анализ состояния предметной области. Постановка задач исследования

Большинство современных КС можно условно отнести к одному из следующих классов (рис. 1.1.):

1. Локальные вычислительные сети коммерческих и образовательных учреждений.
2. Сети уровня провайдера: последняя миля, городская сеть.
3. Магистральные каналы, глобальные вычислительные сети.

Однако также существует большое множество виртуальных, анонимных и др. смешанных КС, которые могут быть, одновременно, отнесены к двум или более из перечисленных выше классов КС.

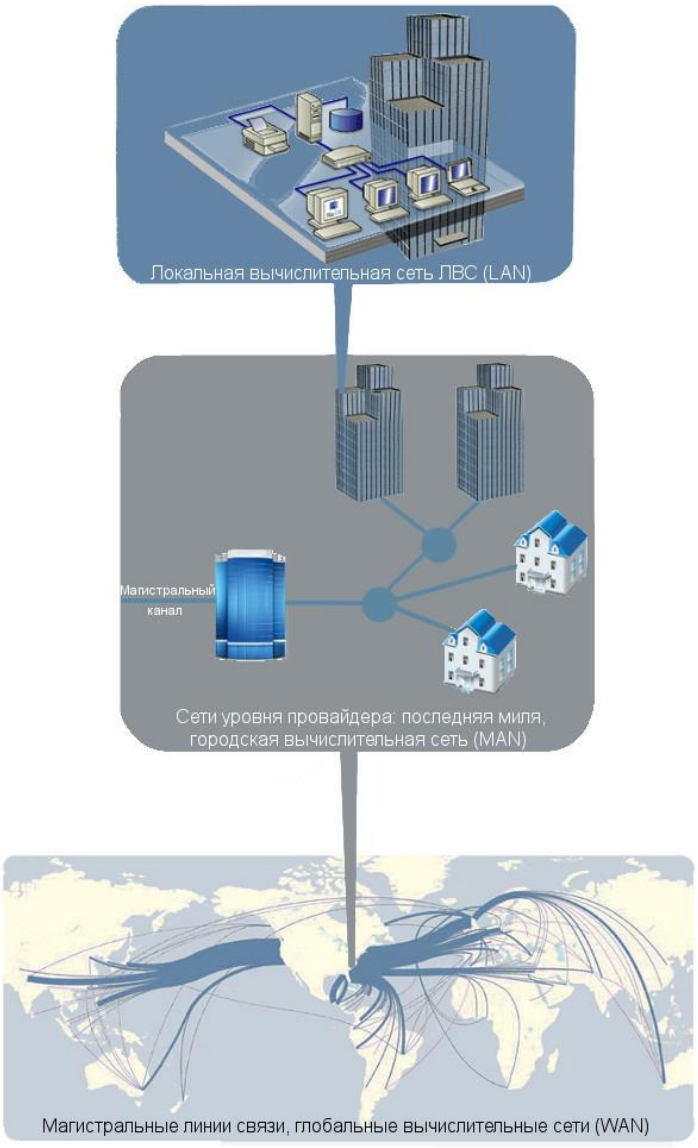


Рис. 1.1. Основные классы КС

1.1. Модели и протоколы, регламентирующие передачу данных в компьютерных сетях

Для обеспечения взаимодействия КС различных уровней разработан используется ряд коммуникационных моделей, обсуждаемых далее в настоящем разделе.

1.1.1. Эталонная модель OSI

Методологической основой сетевых телекоммуникаций является модель Взаимодействия открытых систем (Open System Interconnection – OSI), разработанная Международной организацией по стандартизации (International Organization of Standardization – ISO) [18]. Модель OSI, разработанная в 1984 г., представляет набор стандартов, которые обеспечивают совместимость и эффективное взаимодействие различных сетевых технологий и сетевого оборудования. В модели OSI выделены 7 уровней рассмотрения процессов передачи информации (см. таблицу 1.1).

Таблица 1.1

Уровни модели OSI и их назначение

Уровень (layer)	*Назначение*	*Единицы информации*
7. Прикладной (application)	Доступ к сетевым службам	Объем данных
6. Представительский (presentation)	Представление и шифрование данных	Поток
5. Сеансовый (session)	Управление сеансом связи	Сеансы
4. Транспортный (transport)	Прямая связь между конечными пунктами и надежность	Сегменты / Дейтаграммы
3. Сетевой (network)	Определение маршрута и логическая адресация	Пакеты
2. Канальный (data link)	Физическая адресация	Кадры
7. Физический (physical)	Работа со средой передачи, сигналами и двоичными данными	Биты

Из таблицы 1.1 видна основная особенность описания информационных потоков в КС – на разных уровнях модели OSI используются различные единицы измерения информации. Соответственно, при описании процессов передачи информации на каждом из уровней рассмотрения информационных потоков в КС используются собственные физические механизмы их описания. Отмеченная особенность описания процессов передачи информации в КС свидетельствует о том, что с научной точки зрения трафик в КС относится к категории «система», а его исследование к задачам системного анализа. Для обоснованного выбора уровня рассмотрения информационных потоков в высокоскоростных магистральных Интернет-каналах рассмотрим каждый из уровней модели OSI более подробно.

Уровни эталонной модели OSI

Уровень 7 (уровень приложений)

Уровень приложений – наиболее близкий к пользователю уровень модели OSI. На данном уровне обслуживаются прикладные программы пользователей, находящихся вне

пределов модели OSI, поэтому какие-либо услуги на данном уровне модели OSI не предоставляются. При этом, однако, идентифицируются источники и приемники передаваемой информации (партнеры), устанавливается их доступность для связи, синхронизируются совместно работающие прикладные программы, а также устанавливается договоренность о процедурах восстановления после обнаружения и устранения ошибок и контроля целостности данных. Также на данном уровне определяется степень достаточности ресурсов для осуществления предполагаемого обмена информацией между партнерами.

Уровень 6 (уровень представлений)

На уровне представлений обеспечивается согласование синтаксиса передачи данных с уровнем приложений другой системы. При необходимости на данном уровне форматы данных приложений конвертируются в более общие форматы представления информации.

Уровень 5 (сеансовый)

На сеансовом уровне обеспечивается установление, управление и завершение сеансов взаимодействия приложений. Сеансы состоят из диалога между двумя или более партнерами уровня представлений, в ходе которого реализуется управление обменом информации и обеспечивается синхронизация диалога между партнерами. На этом уровне также формируются отчеты об особых ситуациях, возникающих на сеансовом уровне, а также на уровнях приложений и представлений.

Уровень 4 (транспортный)

На транспортном уровне данные сегментируются и повторно собираются в единый поток. Здесь решаются вопросы надежной транспортировки данных в КС со сложной топологией. На транспортном уровне функционируют механизмы установки, поддержания и упорядоченного завершения действия виртуальных каналов, обнаружения и устранения неисправностей транспортировки, а также управления информационным потоком с целью предотвращения перегрузи одной системы данными, передаваемыми другой системой.

Уровень 3 (сетевой)

На сетевом уровне функционируют механизмы, обеспечивающие выбор маршрута между пользователями, которые могут находиться в разных подсетях КС и их соединение.

Уровень 2 (канальный)

На канальном уровне решаются вопросы физической адресации, топологии сети, дисциплины в канале связи, уведомления об ошибках, упорядоченной доставки кадров, а также вопросы управления потоком данных. Этим обеспечивается надежный транзит данных через физический канал.

Уровень 1 (физический)

На физическом уровне процесс передачи информации описывается в терминах теории электрических цепей (уровни напряжений, временные параметры изменения напряжений, скорости физической передачи данных, максимальные расстояния передачи информации, физические разъемы и т.д.). На физическом уровне задаются электротехнические, механические, процедурные и функциональные характеристики активизации, поддержания и деактивации физического канала между конечными системами.

Отметим, что на практике при выборе архитектуры сети, сетевого оборудования, его настройке и модернизации используется такая информация как физическая адресация, маршруты, проверка доставки и др. [17], которая в соответствие с моделью OSI носится к канальному и сетевому уровням.

Одним из основных протоколов, регламентирующим правила передачи информации в КС, является стек протоколов TCP/IP. На практике производители аппаратного и программного обеспечения для описания и моделирования КС используют как уровни модели OSI, так и стек протоколов TCP/IP. В этой связи целесообразно рассмотреть, как соотносятся уровни рассмотрения информационных потоков в соответствие с моделью OSI и стеком протоколов TCP/IP.

1.1.2. Стек протоколов TCP/IP

Стек протоколов TCP/IP регламентирует процесс передачи информации между конечными точками. В нем определены формат данных, правила адресации, маршрутизации и процесса обмена. Стек TCP/IP включает в себя множество коммуникационных протоколов, важнейшими из которых являются TCP и IP. Функции стека протокола TCP/IP, как и в эталонной модели OSI, разделены на несколько уровней (рис. 1.2).

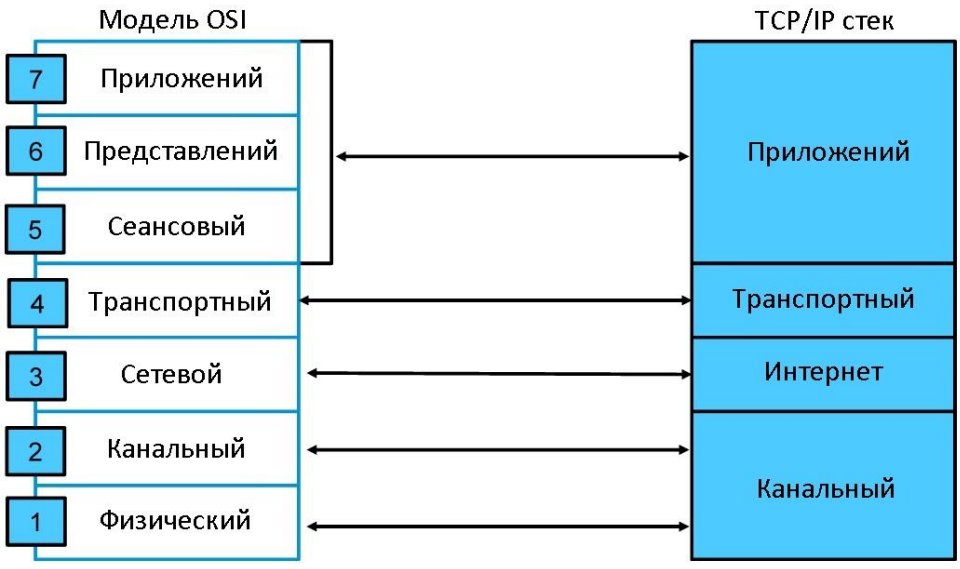


Рис. 1.2. Соответствие между уровнями модели OSI и стеком TCP/IP

К канальному уровню протокола TCP/IP отнесены процессы, которые реализуются на двух нижних уровнях модели OSI – канальном и физическом. На канальном уровне протокола TCP/IP задаются физические характеристики соединения и контролируется доступ к передаваемым данным и формат их передачи.

На Интернет-уровне решается задача маршрутизации данных от источника до места назначения за счет идентификации пакетов, в том числе переданных удаленными хостами, а также перемещения данных между канальным и транспортным уровнями, фрагментации и сборки пакетов данных.

Ядром стека протоколов TCP/IP является транспортный уровень. На данном уровне предоставляются услуги связи прикладным процессам, которые запущены на сетевых хостах.

На прикладном уровне рассматриваются приложения, обеспечивающие передачу файлов, устранение неполадок КС и доступа пользователей к сети Интернет, а также реализуется поддержка Интерфейса Программирования Приложений (Application Program Interface – API), который обеспечивает доступ к КС программам, разработанным для различных операционных систем.

С научной точки зрения основной интерес представляют сетевой и канальный уровень модели OSI, соответственно, транспортный и интернет уровни TCP/IP стека. На данных уровнях работают TCP, UDP и IP протоколы, которые являются основой функционирования вычислительных сетей. Данные между уровнями TCP/IP стека (также, как и между уровнями модели OSI) передаются методом инкапсуляции.

1.1.3. Инкапсуляция

Данные, передаваемые прикладными программами в соответствие с протоколом TCP, прежде чем они превратятся в поток битов, передаваемых в линию, проходят через сверху вниз все уровни протокольного стека (рис 1.3).

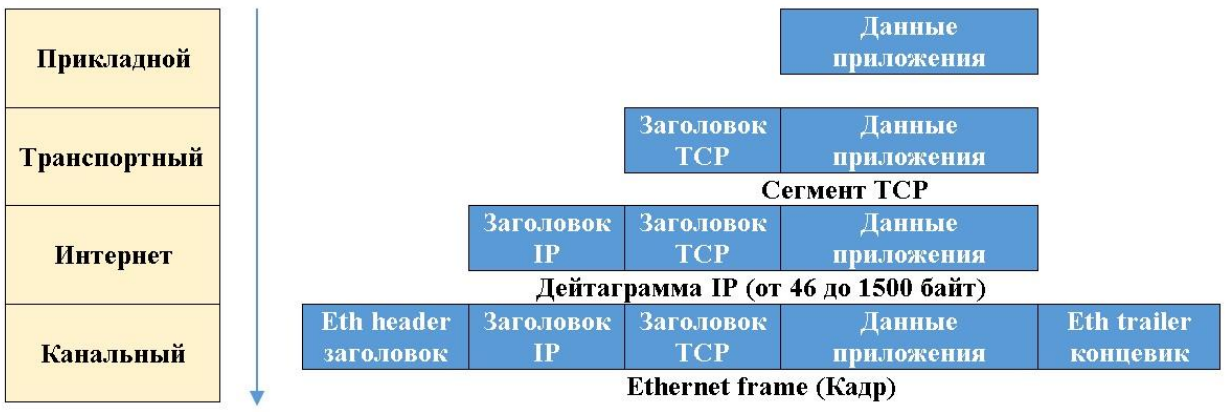


Рис. 1.3. Схема процесса инкапсуляции данных

Из рис. 1.3 видно, что на каждом слое к информации, полученной с верхнего уровня, добавляется дополнительная информация в виде заголовка (header), а на канальном уровне

– дополнительного блока (концевик, trailer). Затем сегмент TCP (TCP segment), состоящий из заголовока TCP и данных приложения, передается IP-модулю. Порция данных, которую IP отдает драйверу интерфейса, называется IP дейтаграммой (IP datagram). Пачка битов, передаваемых по кабелю Ethernet, образует кадр (frame).

Длина кадра в локальной сети Ethernet находится в диапазоне от 46 до 1500 байтов, что обусловлено физическими ограничениями на размер поля данных.

Передача данных в соответствие с протоколом UDP происходит аналогично. Здесь с транспортного уровня на Интернет-уровень передается UDP-дейтаграмма, состоящая из 8-ми байтного UDP-заголовка и данных приложения. Отметим, что подобным образом организована передача данных между рассматриваемыми уровнями и для других протоколов стека протоколов TCP/IP, например, ICMP и IGMP. В этой связи в IP заголовке предусмотрено 8-ми битовое поле (protocol), в котором указывается код протокола, по которому осуществлена передача данных с транспортного уровня на Интернет-уровень: ICMP – 1, IGMP – 2, TCP – 6, UDP – 17.

Аналогично, несколько различных приложений могут одновременно использовать протокол TCP (или UDP). В этой связи в TCP- и UDP-заголовках, добавляемых к данным на транспортном уровне, предусмотрены специальные поля (16-разрядный номер порта (port numbers)), в которое записывается условный код, позволяющий идентифицировать приложение-источник данных и приложение-получатель данных.

На канальном уровне к данным Интернет-уровня добавляется Ethernet-заголовок, в котором имеется 16-разрядное поле (type), предназначенное для идентификации типа протокола, использованного для передачи данных между обсуждаемыми уровнями (IP, ARP или RARP).

Так как протоколы TCP, UDP и IP являются сегодня основными протоколами, в соответствие с которыми реализуется передачи данных в КС, далее эти протоколы рассматриваются более подробно.

1.1.4. Протокол TCP

Протокол TCP, являющийся надежной потоковой службой, соответствует транспортным уровням стека TCP/IP и эталонной модели OSI. По сути данный протокол – это независимый протокол общего назначения, который можно адаптировать для использования с любыми средствами доставки [4].

Единицей передачи данных между двумя хостами в протоколе TCP является сегмент. Сегменты обеспечивают установление соединений, передачу данных, отправку сигналов подтверждения приема, анонсирование размеров окон передачи данных и закрытие соединения. Поскольку протокол TCP является дуплексным (т.е. потоки данных

могут одновременно передаваться в двух направлениях) сигналы подтверждения, посланные от хоста А к хосту Б, передаются в тех же сегментах, что и потоки данных от хоста А к хосту Б, несмотря на то, что сигналы подтверждения приема относятся к потокам данных, текущих от хоста Б к хосту А. Формат TCP сегмента представлен на рисунке 1.4:

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="16">Номер порта отправителя</td><td colspan="16">Номер порта получателя</td></tr>
<tr><td colspan="32">Порядковый номер</td></tr>
<tr><td colspan="32">Номер сигнала подтверждения</td></tr>
<tr><td colspan="4">Длина заголовка</td><td colspan="6">Резерв</td><td colspan="6">Код сегмента</td><td colspan="16">Размер окна</td></tr>
<tr><td colspan="16">Контрольная сумма</td><td colspan="16">Указатель срочных данных</td></tr>
<tr><td colspan="24">Параметры протокола TCP (при наличии)</td><td colspan="8">Выравнивание</td></tr>
<tr><td colspan="32">Область данных</td></tr>
</table>

Рис. 1.4. Формат TCP сегмента

Из рис. 1.4. видно, что каждый сегмент состоит из двух частей – заголовка и блока данных. В заголовке, называемом TCP-заголовком, находятся идентификационные данные и управляющая информация. В первых двух полях заголовка располагаются номера TCP-портов отправителя и получателя, которые идентифицируют прикладные программы по обе стороны соединения. В поле «Порядковый номер» заносится текущее положение текущего сегмента в потоке данных отправителя. В поле «Номер сигнала подтверждения» указывается номер октета, который отправитель ожидает в дальнейшем получить обратно. Значение поля «Порядковый номер» относится к потоку данных, направленному в том же направлении, что и передаваемый сегмент, номер сигнала подтверждения – к потоку данных, направление которого противоположно направлению передаваемого сегмента.

В поле «Длина заголовка» указывается длина заголовка сегмента, выраженное в блоках длиной 32 бита. Данное поле включено в заголовок, поскольку поле параметров протокола имеет переменную длину, зависящую от того, какие параметры включены в сегмент. Таким образом, размер TCP-заголовка зависит от того, какие параметры в него включены. Поле «Резерв» (см. рисунок 1.4) размером 6 битов зарезервировано для использования в будущих стандартах протокола.

Поскольку в протоколе TCP сегменты используются и для передачи данных, и для передачи сигналов подтверждения их приема, а также для передачи запросов на установку и закрытие соединения, в TCP-заголовок включено специальное 6-битовое поле «Код сегмента», которое определяет его формат и содержимое. Значения битов, записанных в данное поле, определяют правила интерпретации других полей TCP-заголовка (табл. 1.2).

Таблица 1.2

Значение битов кода сегмента TCP-заголовка

Название бита (слева направо)	*Значение, если бит установлен в 1*
URG	В заголовке присутствует указатель срочных данных
ACK	В заголовке указано поле подтверждения приема
PSH	В данном сегменте указан запрос на немедленную отправку данных (push)
RST	Сброс соединения
SYN	Синхронизация порядковых номеров
FIN	Отправитель достиг конца потока данных

В поле «Размер окна» отправляемого сегмента по протоколу TCP размещается количество октетов, которые получатель может принять (т.е. указывают размер своего приемного буфера). В это поле помещается 16-битовое беззнаковое целое число, имеющее стандартный сетевой порядок следования байтов. Анонсирование окна является примером дуплексной передачи данных, поскольку оно выполняется для всех сегментов, включая сегменты, в которых передаются данные, и сегменты с сигналами подтверждения приема.

Для установки соединения в протоколе TCP используется трехэтапный метод квитирования (three-way handshake). Пример простейшего процесса квитирование представлен на рисунке 1.5.

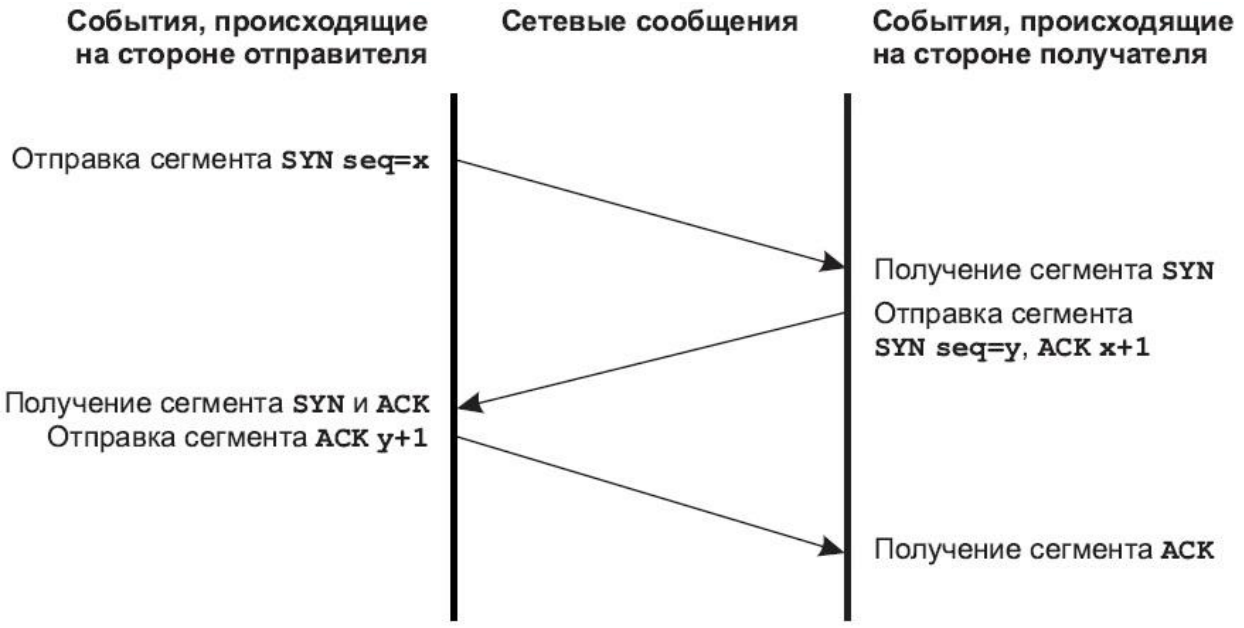


Рис. 1.5. Процесс квитирования протокола TCP

Из рисунка 1.5 видно, что первый сегмент, посылаемый в процессе квитировании отправителем, в поле кода сегмента его заголовка установлен бит SYN. В заголовке второго сегмента, отправляемого получателем, установлены два бита: SYN и ACK. Для отправителя это означает, что с получателем успешно установлено соединение и он готов

к получению данных. Далее отправитель передает получателю сегмент ACK, который подтверждает, что обе стороны уведомлены об установке соединения.

Трехэтапный метод квитирования выполняет две важные функции. Во-первых, он гарантирует, что обе стороны соединения готовы к приему данных и о готовности одной стороны знает другая сторона. Во-вторых, с его помощью выполняется процесс согласования начального значения поля «Порядковый номер». Во время квитирования обе стороны соединения обмениваются начальными порядковыми номерами и ожидают подтверждения их приема. Порядковые номера используются для идентификации потоков данных, посылаемых каждой из сторон открытого соединения. Они выбираются сторонами самостоятельно (обычно случайным образом) во время открытия соединения, однако, они никогда не могут начинаться с одного и того же значения. Это позволяет не допустить совпадения номеров октетов, указываемых в сигналах подтверждения приема, и номеров, используемых в заголовках при передаче сегментов данных.

Отметим, что в протоколе TCP невозможна передача данных вместе с начальным порядковым номером непосредственно в сегментах квитирования. При возникновении подобных случаев модуль протокола TCP блокирует данные до завершения процесса квитирования, а по его завершению доставляет ожидающему их приложению.

Метод скользящего окна

Метод скользящего окна в сравнении с описанным выше вариантом квитирования является более сложной технологией, в которой реализовано подтверждение приема с повторной передачей. Он позволяет эффективно использовать полосу пропускания – максимально возможную скорость передачи информации в КС, соединяющей компьютер пользователя с Internet через оператора услуг связи (количество бит данных, которое можно передать по каналу в каждую секунду; Кбит/с (1024 бита в секунду), Мбит/с, Гбит/с) [8], поскольку отправитель может послать несколько пакетов сразу, не дожидаясь подтверждения приема каждого пакета. Суть метода состоит в том, что реализуется режим передачи пакетов, называемый «методом скользящего окна». При использовании данного метода выбирается некоторое окно фиксированного размера, которое в процессе передачи сдвигается вдоль пронумерованной последовательности пакетов, предназначенных для передачи от источника к приемнику (рис.1.6).

Рис. 1.6. К объяснению метода «скользящего окна» (размер окна – 8 пакетов)

Источник передает в КС пакеты №№ 1–8. После получения подтверждения о приеме пакета № 1 скользящее окно сдвигается на один пакет вправо и в сеть отправляется пакет № 9 (рисунок 1.6. б). Если для какого-либо пакета, переданного при данном положении окна, не будет получен сигнал подтверждения его получения пакета, то выполняется повторная передача данного пакета. Пакет считается непринятым (unacknowledged), если он был послан в КС, но подтверждение о его приеме не было получено. Формально количество непринятых пакетов не может превышать размеров окна и, на практике, оказывается относительно небольшим целым числом.

Иллюстрация метода скользящего окна на примере окна, состоящего из трех пакетов, представлена на рисунке 1.7.

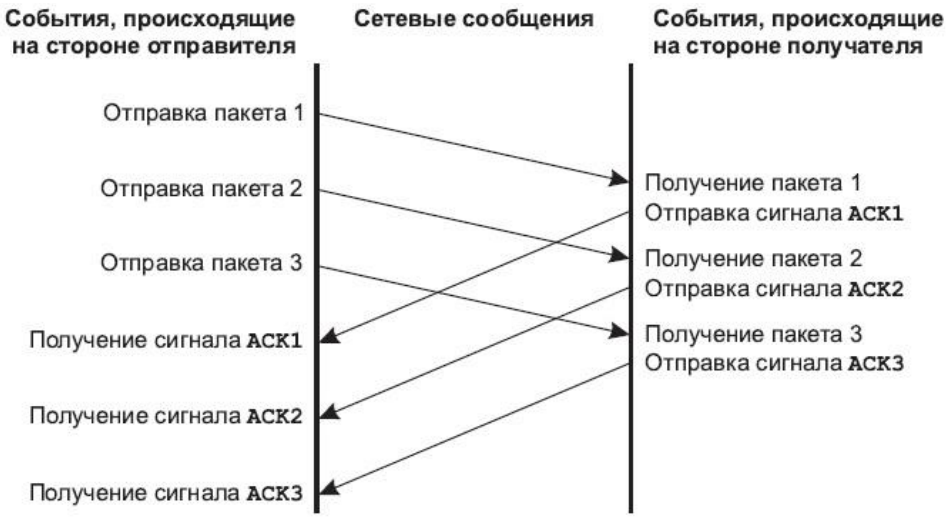


Рис. 1.7. Иллюстрация метода скользящего окна

Из рисунка 1.7 видно, что отправитель успевает послать все три пакет в КС до того, как будет получено хотя бы одно сообщение о подтверждении их приема. Таким образом, при правильном выборе скользящего окна удается полностью исключить простои КС и достичь большей производительности в сравнении с методом подтверждения приема и повторной передачей.

1.1.5. Протокол UDP

Протокол UDP соответствует третьему уровню стека TCP/IP и четвертому уровню эталонной модели OSI. Заголовок протокола пользовательских дейтаграмм (User Datagram Protocol, UDP) значительно проще, чем у протокола TCP (рис. 1.8).

Рис. 1.8. Структура UDP Сегмента

Из рисунка 1.8 видно, что он содержит только четыре поля: «Номера порта передатчика», «Номер порта приемника», «Длина дейтограммы», «Контрольная сумма». Так протокол UDP не гарантирует доставки всех пакетов, в данном протоколе контроль передачи данных решается на нижних уровнях.

1.1.6. Протокол IP

Протокол IP, называемый межсетевым или Internet протоколом соответствует второму уровню TCP/IP стека и третьему уровню модели OSI. В данном протоколе используется ненадежный механизм доставки, не требующий установки соединения с получателем. Данные, передаваемые по протоколу IP (дейтаграммы IP), представляют собой инкапсулированные TCP и UDP пакеты (рис. 1.9).

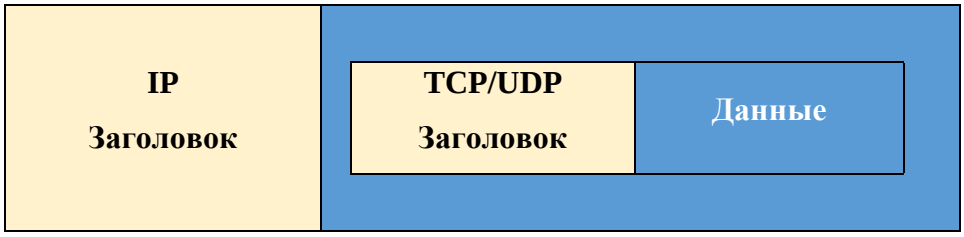


Рис. 1.9. Дейтаграмма IP

Заголовок IP протокола состоит из идентификатора, протокола верхнего уровня, IP адреса отправителя и получателя и других опций (рис. 1.10).

Рис. 1.10. Заголовок IP протокола

Ненадежность передачи данных по протоколу IP с физической точки зрения означает, что по пути следования пакет может быть утерян, продублирован, задержан или доставлен с нарушением порядка следования. При этом служба доставки не контролирует вышеперечисленные проблемы и не сообщает о них отправителю и получателю. При этом каждый пакет считается независимым от остальных, поэтому соответствующая служба доставки не требует установки соединения. Отметим, что пакеты из одних и тех же последовательностей пакетов, передаваемых отправителем и получателем, могут проходить по разным маршрутам.

Для решения проблемы потери пакетов, передаваемых по протоколу IP в данном протоколе:

1. определен базовый элемент передачи данных в КС на основе протокола TCP/IP, что, в свою очередь, обеспечило возможность четкого определения протоколе IP формата передаваемых данных;

2. реализована функция маршрутизации, которая обеспечивает возможность выбора пути, по которому будут посылаться данные;

3. в протокол включен набор правил, регламентирующих способы обработки пакетов узлами сети и маршрутизаторами, а также условия, при которых должны генерироваться сообщения об ошибке и удаляться пакеты.

Таким образом, проведенный выше анализ механизмов передачи данных в КС позволяет сделать вывод о высокой сложности задачи описания свойств информационных потоков в КС. Данный вывод также подтверждается отсутствием сегодня адекватных математических моделей современного Интернет-трафика и соответствующих программных инструментов, с помощью которых на этапе проектирования архитектуры КС и выбора сетевого оборудования можно получать количественные оценки характеристики [90]. В этой связи изучение свойств информационных потоков в современных КС представляет интерес, как с научной, так и с практической точек зрения.

1.2. Dump трафика, как носитель информации о параметрах информационных потоков в КС

Дамп (от. англ. dump – сваливать) сетевого трафика представляет собой текстовый файл, который можно рассматривать как «снимок данных» прошедших через сеть или сетевое устройство за определенный промежуток времени. Сегодня стандартом записи потока сетевого трафика, де-факто, является формат .pcap (Packet Capture). С данным форматом работает большинство сетевых анализаторов, в частности, такие известные программные инструменты как tcpdump [44] и wireshark [49], которые предназначены для работы с трафиком (запись, чтение дампов, визуализация и др.) передаваемым в КС. Также

следует отметить, что для работы с pcap-файлами используются библиотеки libpcap и winpcap [30], работающие под управлением ОС Unix и ОС Windows, соответственно. Данные библиотеки поддерживаются их разработчиками, что обеспечивает работоспособность инструментов, предоставляемых данными библиотека, при работе с современным сетевым оборудованием (например, в версии 2.4 была добавлена поддержка записи времени с точностью до одной наносекунды).

Рассмотрим структуру pcap-файла, представленную на рисунке 1.11.

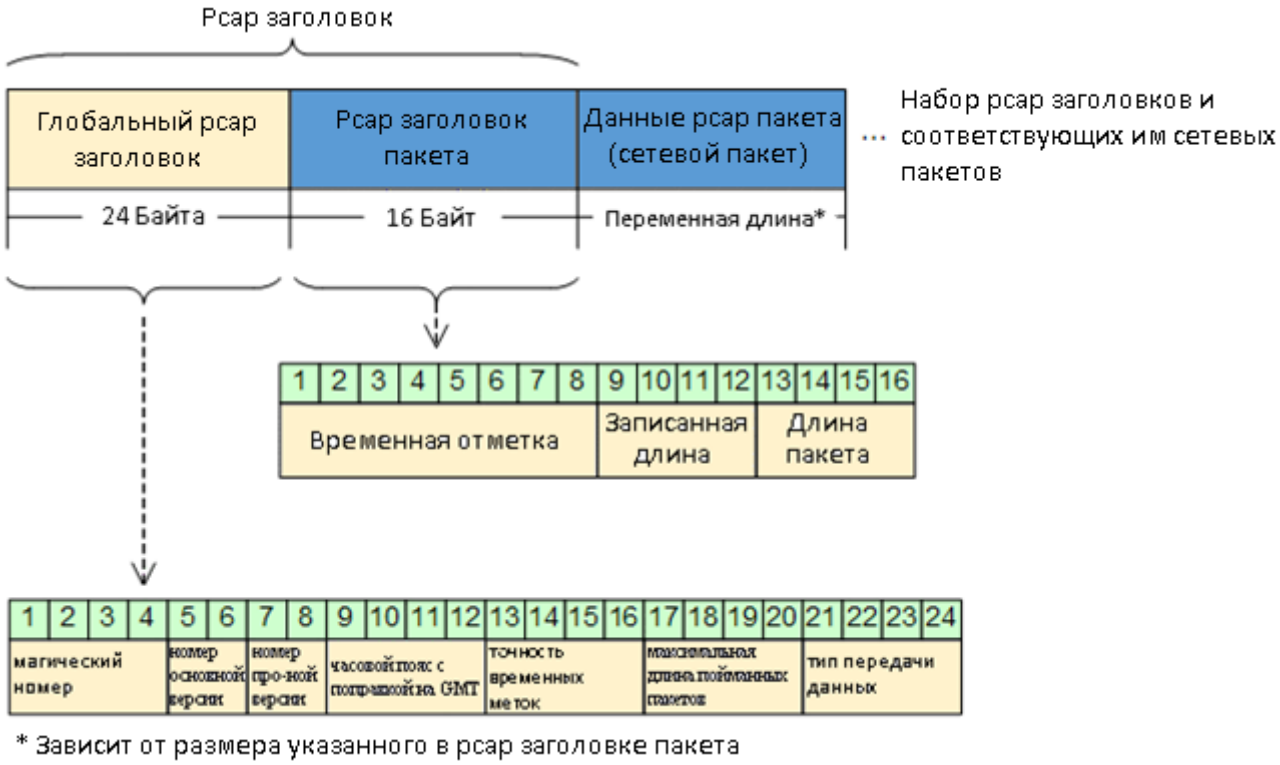


Рис. 1.11. Структура pcap-файла

В начале pcap-файла размещается заголовок, состоящий из двух частей: глобального pcap-заголовка и pcap-заголовка пакета. Глобальный pcap-заголовок содержит следующую информацию.

– magic_number (магическое число) – число, используемое для определения формата файла и порядка следования байтов. (Например, если программа сетевого анализатора, захватывающая трафик в файл дампа, имеет в данной переменной 0xa1b2c3d4 в оригинальном формате и соответствующем порядке байтов, то программа сетевого анализатора, читающая этот файл дампа, будет считывать последовательность байт аналогичную записанной последовательности – 0xa1b2c3d4. Если последовательность магического числа обратная – 0xd4c3b2a1, приложение, читающее файл дампа, знает, что считывается измененное значение магического числа 0xd4c3b2a1, а потому необходимо поменять местами соответствующие поля.)

– version_major, version_minor (основная, промежуточная версии) – номера соответствующих версий формата текущего файла.

– thiszone (часовой пояс) – время коррекции в секундах между средним временем по Гринвичу (GMT) или всемирным координированным временем (UTC) и временем текущего часового пояса для временных меток заголовка очередного пакета. (Например, если временные метки зафиксированы по GMT или UTC, то часовой пояс будет равняться нулю. Если временные метки зафиксированы для центральноевропейского времени (Амстердам, Берлин и др.), которое составляет GMT + 1:00, часовой пояс должен иметь значение – 3600. На практике всегда используются отметки времени в формате GMT, и часовой пояс всегда равен 0.)

– sigfigs (точность временных меток) – параметр, задающий точность, с которой будут определятся временные метки.

– snaplen (максимальная длина пойманных пакетов) – величина, определяющая, сколько октетов каждого пакета необходимо захватывать (обычно используется значение, гарантирующее захват всех октетов для любого пакета, равное 65 535).

– network (тип способа передачи данных) – параметр, выбираемый в соответствии с типом заголовка канального уровня и типом заголовка для каждого пакета (например, 802.11, 802.11 с различной уточняющей информацией, PPP, Token Ring, FDDI и т.д.).

После глобального pcap-заголовка следует набор, чередующихся pcap-заголовков пакета и соответствующих данных сетевых пакетов. Pcap-заголовок пакета содержит в себе информацию о времени записи пакета, записанной и фактической длинах сетевого пакета. Данные сетевого пакета в зависимости от выбранной длины записи представляют собой информацию о проходящем через сеть пакете, в том числе адрес отправителя и получателей на всех уровнях модели OSI, порты, передаваемые данные и т.д. В том случае, когда длина записи данных сетевого пакета не ограничена, pcap-файл содержит в себе исчерпывающую информацию о данных, передаваемых в КС. Если в данный временной интервал не было получено ни одного пакета, в поле данных pcap-пакета записывается ноль. Формат записи захваченных пакетов зависит от протокола и используемой технологии передачи данных (Ethernet, ATM и т.д.).

Очень важным оказывается вопросом обеспечения анонимности пользовательских данных, переданных в КС во время снятия дампа, который, как правило, решается стороной, предоставляющей для анализа pcap-файлы.

1.3. Программные инструменты для работы с pcap-файлами и методы анализа дампа

1.3.1. Программные инструменты для работы с pcap-файлами

Наиболее известным программным инструментом, позволяющим извлекать из pcap-файлов интересующие исследователя характеристики информационных потоков в КС, является Wireshark [49], программные реализации которого разработаны как для операционных систем (ОС) семейства Windows, так и ОС семейства UNIX, а также tcpdump[44], разработанный для ОС семейства UNIX.

Обсуждаемые программные инструменты обеспечивают извлечение в выбранном измерении требуемой информации из pcap-файлов данных. Однако они не предоставляют пользователю возможности проведения количественного анализа извлеченной информации. Так же необходимо отметить, что Wireshark и tcpdump имеют значительные ограничения на размер анализируемого дампа. В результаты проведенного нами эксперимента показали, что на компьютере с объемом оперативной памяти 4Гб можно обработать pcap-файл размером не более 200 Мбайт. (Отметим, что размеры реальных pcap-файлов, содержащих дампы трафика в высокоскоростном магистральном Интернет-канале, могут составлять 1 Гб и более (см., например, [47]). Следовательно, отношение объема анализируемого дампа к объему оперативной памяти, необходимой для его обработки, только при открытии pcap-файла составляет 1:20. При этом понятно, что при необходимости провести какие-либо операции с данными, требования к объему оперативной памяти значительно увеличатся. Кроме того, для извлечения информации из дампов и ее анализа помимо значительных объемов оперативной памяти, так же предъявляются не менее значительные требования к вычислительным мощностям процессора/процессоров ЭВМ.

В связи с выявленными ограничениями существующих инструментов анализа pcap-файлов, понятно, их использование при проведении исследования исследований реальных информационных потоков, передаваемых в высокоскоростных КС, невозможно. В этой связи было принято решение разработать собственный программный инструмент, позволяющий не только извлекать в выбранном разрезе информацию из pcap-файлов, но и проводить ее количественный анализ.

1.3.2. Методы анализа трафика

Анализ трафика после извлечения в выбранном измерении информации из pcap-файла осуществляется в два этапа. Сначала вычисляются те или иные количественные характеристики изучаемых информационных потоков. Затем проводится анализ данных характеристик и классификация трафика. Признаки, используемые для анализа, извлекаются из содержания пакета, его заголовка или потока данных – однонаправленного потока пакетов между указанной парой IP-адресов.

Анализ трафика по содержимому пакета

Данный подход к изучению Интернет-трафика основан на исследовании трафика на различных уровнях, в том числе, прикладном уровне. В данном случае содержимое пакета исследуется последовательно (побитно), для того чтобы найти конкретные потоки битов, обладающие свойствами выбранного соответствующего протокола. Если такой битовый поток найден, пакет может быть корректно отмечен. На практике методы исследования содержимого пакета обычно используются для обнаружения P2P трафика и обнаружения аномальных состояний КС [77,56]).

Анализ содержимого пакетов, несмотря на относительно высокую точность его результатов, имеет большое число недостатков. Прежде всего, высокая с вычислительной точки зрения сложность процедуры поиска битов, требующая большого объема дискового пространства для хранения полученной информации. Поиск определенных свойств трафика в содержимом пакета требует перехвата или зеркалирования трафика. Данный метод может оказаться неэффективным для анализа потоков, создаваемых новыми приложениями (tor, p2p, torrents и т.д.), свойства которых отличаются от известных ранее (например, используемые порты и их кол-во, логика взаимодействия клиента и сервера) свойств потоков, или в случае, когда трафик передается в туннеле. Он также оказывается абсолютно бесполезным, если используется шифрованное соединение точка-точка. При этом возникает важная проблема, связанная с необходимостью выполнения требований Закона о защите информации [99]. В этой связи нат практеике применение данного метода предваряется предварительной подготовкой трафика с целью анонимизации (замена реальных ip-адресов, очистка содержимого пакетов и т.д.), что, в свою очередь, требует дополнительных временных затрат и вычислительных ресурсов.

Анализ трафика без исследования содержания пакета

Рассмотрим методы анализа pcap-файлов, в которых не проводится анализ содержимого пакета, в том числе методы, основанные на анализе информации, извлекаемой из заголовков транспортного и сетевого уровней (портов) [17], а также оценивании характеристик транспортного потока оценивает.

Анализ pcap-файлов на основе портов

Данный метод основан на 16-битных номерах портов транспортного уровня, которые используются серверами для идентификации процесса, создавшего данный информационный поток. Напомним, что значения номеров портов регламентируются «Уполномоченной организацией по распределению нумерации в сети Интернет» (Internet Assigned Numbers Authority – IANA).

На ранних стадиях исследования информационных потоков в КС основывались на том, что многие службы передачи данных работают на «известных» стандартных транспортных портах из списка зарегистрированных портов IANA. При этом

использовалась идентификация по портам транспортного уровня, особенно, для P2P трафика. Однако для других типов Интернет-трафика данный метод теряет свою эффективность, так как установить взаимно-однозначное соответствие между службой и транспортным портом оказывается невозможным. Это обусловлено тем, что некоторые протоколы, используемые для передачи данных в КС, не зарегистрированы в IANA (4915–65535), кроме того IANA определяет порты не для всех приложений (например, порт TCP 5555 неофициально используется 3-мя приложениями: Freeciv, HP Data Protector и SAP), а некоторые порты определены неоднозначно (например, порт UDP 2049 официально определен для протоколов nfs и shilp). Кроме этого, стандартные службы, реализующие передачу данных в КС, для того чтобы обходить политики или ограничения контроля доступа операционной системы могут работать на нестандартных портах (например, 82UDP порт официально используется протоколом XFER, а неофициально проектом Tor для целей управления).

Также необходимо отметить, что некоторые приложения, например, передающие P2P трафик, намерено не используют стандартный набор портов или меняют их в динамическом режиме, чтобы избежать обнаружения. Таким образом, анализ на основе портов может давать ложные результаты или не дать результатов вообще, если используется неклассифицированный порт. По известным оценкам точность данного метода анализа составляет от 50% до 70% [43].

Метод анализа по потоку

Отмеченные выше недостатки метода анализа трафика по портам определи необходимость разработки методов идентификации трафика, в которых не используется информация о транспортных портах, но используется информация о переданных потоках. В работе под понятием «поток» будем понимать двунаправленное соединение отправитель-получатель. При этом каждый переданный поток данных характеризуется четко определенным набором признаков, которые принимают различные значения в зависимости от соответствующего класса трафика.

Признаки потоков извлекаются, в первую очередь, из IP заголовков, а также из TCP-данных (общее количество передаваемых данных, размер TCP сегмента и др.). Поток можно собирать средствами роутера. На основе вышеприведённой информации можно сделать вывод о том, что Анализ на основе потока является более гибким для высокоскоростных КС, чем проверка содержимого пакетов. Необходимо отметить, что анализ потока возможен только после того как будут переданы все пакеты.

Для выявления типов КС, представляющих наибольший интерес для исследователей особенностей трафика, а также соответствующих методов и инструментов анализа

характеристик информационных потоков проводится анализ работ, посвященных исследованию трафика в КС, а также полученных при этом полученных результатов.

1.4. Анализ современных работ по исследованию трафика

Рассмотрим современные подходы к изучению трафика различных типов КС (LAN, WAN, MAN и др.) на разных уровнях стека протоколов TCP/IP (сетевой, канальный, приложения и т.д).

1.4.1. Уровень локальной вычислительной сети

Исследование трафика локальной сети университета

Интернет-трафик, переданный в локальной КС университета [94], исследовал В.В. Стешенко (Астраханский государственный технический университет (АГТУ)). Цель исследования состояла в подтверждении изначально принятой гипотезы о самоподобии [67] данного трафика.

Напомним, что степень самоподобия оценивается по значению показатель Херста (Hurst) H, в зависимости от значений которого выделяют три типа случайных процессов:

1. $0 \leq H \leq 0{,}5$ – случайный процесс является антиперсистентным или эргодическим рядом, который не обладает самоподобием;
2. $H = 0{,}5$ – полностью случайный ряд, аналогичный случайным смещениям частицы при классическом броуновском движении;
3. $H > 0{,}5$ – персистентный (самоподдерживающийся) процесс, который обладает длительной памятью и является самоподобным.

В проведенных исследованиях использовался суточный дамп Интернет-трафика университета.

При вычислении показателя Херста экспериментального трафика были использованы методы, представленные в табл. 1.3.

Таблица 1.3

Значения параметра Херста

Метод	*Значение H*
Aggregate Variance	0.862
R/S	0.822
Periodogram	0.933
Absolute Moments	0.098
Variance of Residuals	13.25
Abry-Veitch Estimator	1.031
Whittle Estimator	0.97

Из таблицы 1.3 видно, что оценки значений показателя Херста, полученные различными методами, оказались существенно отличными друг от друга. Однако автор сделал вывод о том, что экспериментальный трафик, сгенерированный Интернет-

пользователями локальной КС АГТУ является самоподобным. Данный вывод, с нашей точки зрения, представляется не вполне обоснованным.

1.4.2. Анализ трафика клиент-серверной информационной системы

Свойства сетевого трафика в клиент-серверной ИС исследовали Н.Г. Треногин и Д.Е. Соколов [96]. Цель исследований также состояла в подтверждении фрактальных (самоподобных) свойств трафика в клиент-серверных ИС различной архитектуры – от «классических» двухзвенных до многоуровневых с WEB-доступом и терминальных.

Данные для проведения эксперимента были получены перехватом кадров на FastEthernet-интерфейсе сервера СУБД с помощью программы tcpdump. При этом в предположении дуплексности канала передачи данных, рассматривался трафик одного направления, исходящий по отношению к рабочим станциям. В качестве сервера СУБД использовался сервер Oracle8i; приложение (биллинговая система оператора связи) было реализовано по классической двухзвенной клиент-серверной схеме, т.е. сетевое взаимодействие реализовывалось через протокол TNS / SQL .net поверх TCP.

Рассмотренный срез рассматривался как «свободный трафик» – трафик, свойства которого полностью определяются свойствами соответствующего источника трафика и не зависят от состояния КС. Так как скорость передачи данных в рассматриваемом канале (100 Мбит/с FastEthernet) обеспечивала передачу значительно больших объемов, нежели реально измеренные объемы суммарного трафика, но сторонняя нагрузка в день исследований пренебрежимо мала. Параметры среза трафика приведены в таблице 1.4.

Таблица 1.4

Параметры среза трафика

Параметр	*Значение*
Выходной файл	*.td
Длительность, *ч*	6.093
Число пакетов	688108
Интенсивность λ , c^{-1}	31.368
Средний объем пакета, *байт*	193.2
Рабочих мест	25
Параметр Херста *H* (IDC)	0.729
Параметр Херста *H* (автокорреляция)	0.724

Авторы сделали вывод о том, что сетевой трафик реальных SQL-серверных приложений является самоподобным, а его свойства оказывают значительное влияние на вероятностно-временные характеристики ИС.

1.4.3. Исследование сетевого трафика с целью повышения безопасности компьютерных систем и сетей

Сетевой трафик также исследовался И.М. Ажмухамедовым и А.Н. Марьенковым [52]. Цель исследований состояла в разработке метода выявления аномалий сетевого трафика, обусловленного Интернет-атаками.

Для обнаружения аномалий трафика использовалась модель, основанная на среднем значении и среднеквадратичном отклонении параметров сетевого трафика. При этом локальные (текущие) характеристики потока пакетов сравнивались с глобальными (усредненными), а в качестве статистических показателей использовались выборочное среднее значение, выборочная дисперсия и критерий согласия хи-квадрат.

Блок-схема алгоритма, использованного для выявления аномалий сетевого трафика, схема которого представлена на рис. 1.12.

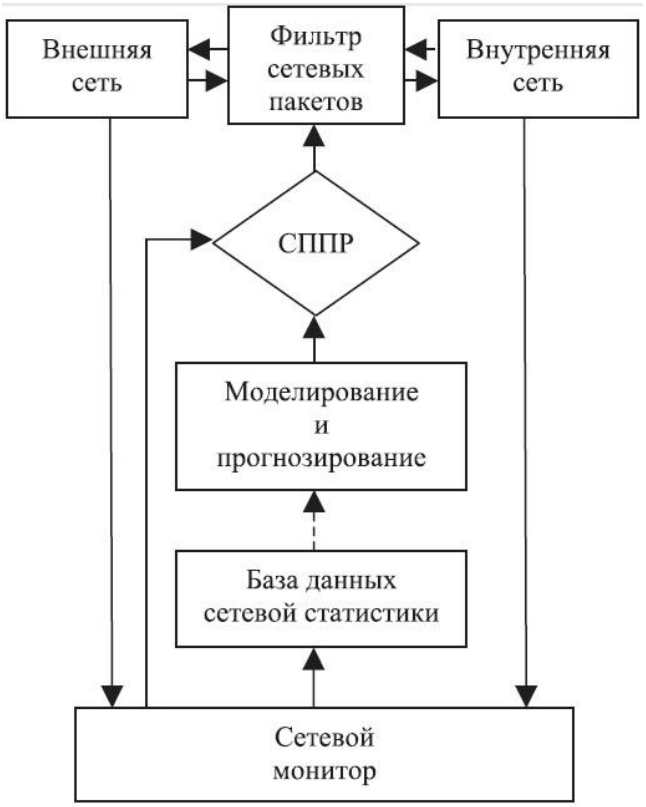


Рис. 1.12. Блок-схема алгоритма, использованного для выявления аномалий сетевого трафика

В блоке «Сетевой монитор» (рис. 1.12) осуществлялся перехват как исходящего, так и входящего информационных потоков на втором уровне коммуникационной модели обмена OSI. Информация, необходимая для дальнейшей работы, извлекалась из найденных заголовков IP-пакетов и сохранялась в «Базе данных сетевой статистики» вместе с датой и временем прихода кадра.

Далее на основе накопленной статистики с помощью метода циклического анализа и метода Хольта моделировалось поведение КС и прогнозировался объем сетевого трафика.

В том случае, если локальные характеристики существенно отличались от глобальных, принималось решение об аномальном поведении потока пакетов, и возникновении потенциальных угроз сбоев в работе сетевого оборудования, ПО или нарушениям политики конфиденциальности.

1.4.4. Анализ трафика с целью выявления возникновения аномальных состояний КС

Интернет-трафик, передаваемый в КС Интернет-провайдера городского уровня, с целью разработки методики выявления аномальных состояний КС по интегральным характеристикам трафика изучался Э.В. Афонцевым и С.В. Поршневым [55]. Авторы провели сравнительный анализ работоспособности ряда математических методов (статистический анализ, спектральный анализ, вейвлет-анализ, методы нелинейной динамики, методы моделирования временных рядов) в задаче выявления сетевых аномалий по реализациям реального Интернет-трафика, выявили интегральные характеристики, значения которых являются признаками возникновения аномальных состояний КС, и дали научно-обоснованные рекомендации по выбору математических методов, используемых в рассматриваемой задаче, для различных источников диагностируемой аномалии.

В ходе проведенных исследований была экспериментально установлена связь между значениями масштабных коэффициентов вейвлет-преобразований зависимостей числа пакетов, переданных в течение 100 мс, от времени входящего и исходящего потоков Интернет-трафика, а также предложен диагностический критерий аномальности сетевого трафика, основанный на вычислении значений функции кросскорреляции вейвлет-коэффициентов зависимостей числа пакетов, переданных в течение 100 мс, от времени. Также отметим, что для изученных зависимостей также были получены оценки значений показателей Херста, которые как для нормального, так и аномального трафиков оказались несколько больше 0.5. Однако отличия оценок данных показателей для каждого из типов изученного трафика оказалась статистически незначимыми.

На основе полученных результатов, авторы предложили комплексную методику выявления аномальных состояний КС, основанную на параллельном использовании корреляционного анализа, который обеспечивает эффективное детектирование сканирования, вирусной активности и атаки на отказ в обслуживании и методов моделирования временных рядов, позволяющих детектировать нарушения производительности КС (рис. 1.13).

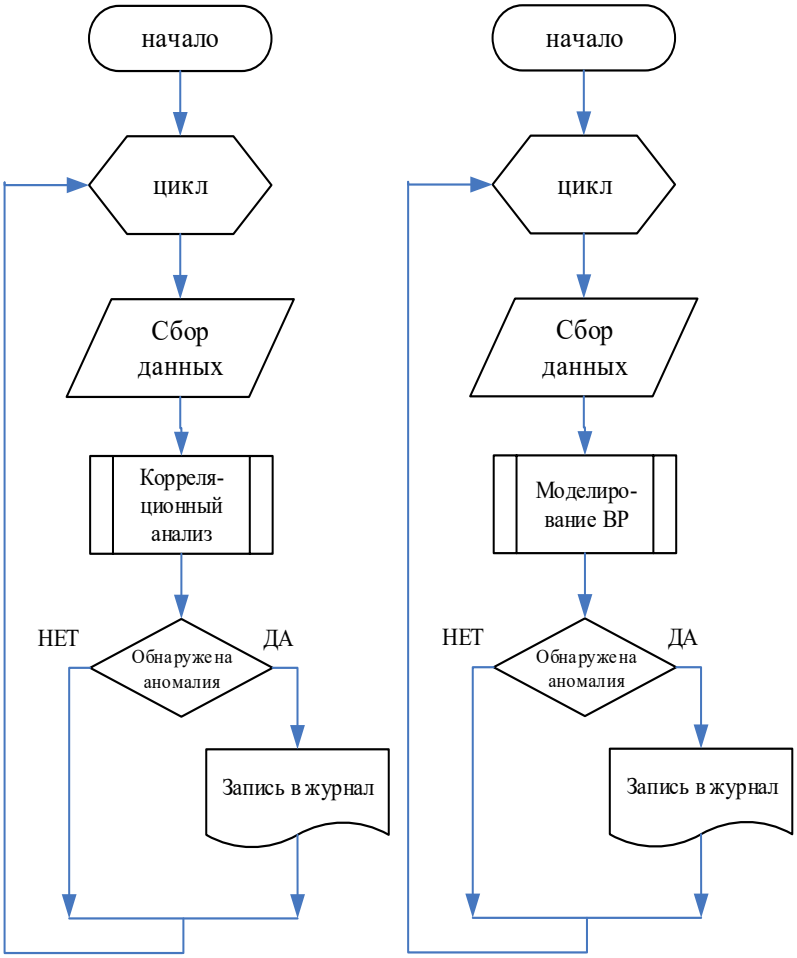


Рис. 1.13. Алгоритм реализации комплексной методики детектирования аномалий

Разработанная комплексная методика выявления аномалий Интернет-трафика, основанная на комбинировании методов корреляционного анализа параметров потоков трафика и моделирования временных рядов Хольта-Винтерса, позволяет выявлять наиболее значимые для операторов связи аномальные состояния трафика по интегральным характеристикам.

Также вопросы, связанные с разработкой модели трафика для выявления аномальных состояний сети и противодействия сетевым вторжениям, исследовались в Гальцевым А.А. и Суховым А.М. [59]. Они провели обзор существующих моделей трафика и методов выявления аномальных состояний КС, модифицировали модель магистрального Инернет-трафика, предложенную Чади Баракатаном в 2002 г. [104], предложили методы, позволяющие проводить системный анализ Интернет-трафика с целью выявления аномальных состояний сети и предотвращения несанкционированных вторжений.

Доработка модели Чади Баракатана состояла в использовании вместо среднего значения суммарной скорости (бит/сек) и ее дисперсии (бит^2/сек^2) функции $B(t)$, описывающей изменение загрузки канала во времени, и функции $N(t)$, описывающей зависимость числа активных потоков от времени, что позволило увеличить скорость сбора и обработки данных о состоянии сети.

Для определения границ рабочей области на плоскости состояний КС авторы использовали уравнение

$$B(t) = b \cdot (N(t) \pm a \cdot A(\varepsilon) \cdot \sqrt{N(t)}),$$

где

$B(t)$ – загрузка канала

$N(t)$ – число активных потоков

$A(\varepsilon)$ – нормальная квантильная функция

b – средняя скорость потока

a – постоянная величина, зависящая от технических характеристик КС.

Пример плоскости состояний КС представлен на рисунке 1.14.

Рис. 1.14. Область состояний сети

Полученная модель была протестирована на академических научно образовательной сетях FREEnet и HEAnet, а так же на сети ЗАО «СамараТелеком». Результаты статистических тестов для сети FREEnet приведены в таблице 1.5

Таблица 1.5

Статистические тесты. Сеть FREEnet

n	*Границы интервала*	$N_{ср}$, *поток*	$B_{ср}$, *Мбит/с*	*σ(B), Мбит/с*	*b, бит/с*	*a*	χ^2 *для* a_x*=0,95*	*Критерий Пирсона*	*Коэффициент корреляции*
1	15000-20000	17489	113,15	23,12	6784	16,3 ± 3,1	–	–	0,7
2	20000-25000	23260	126,05	21,41	5682		–	–	
3	25000-30000	27145	141,63	37,91	5471		–	–	
4	30000-40000	34361	150,70	26,85	4599		3,49(9,49)	+	
5	40000-50000	44751	155,93	27,61	3655		–	–	
6	50000-60000	54591	165,18	18,53	3173		3,45(7,81)	+	
7	>60000	64862	191,98	20,39	3104		0,5(9,49)	+	

Полученные для $\sigma(B)$ и $\sqrt{N_i}$ коэффициенты корреляции позволили сделать вывод о высокой корреляции между данными полученными в результате эксперимента и теоретической моделью.

Далее авторы провели экспериментальную проверку модели, предложенной для обнаружения различных аномальных состояний, и в частности, сетевых атак двух типов: сканирование портов и DDoS-атаки.

В результате анализа данных, собранных вовремя сканирования сети было выявлено резкое увеличение числа активных потоков при практически неизменном количестве передаваемого трафика. Анализ эксперимента по моделированию DDoS- атаки утилитой LOIC так же показал резкое увеличение числа потоков наряду с увеличением передаваемого трафика. В результате было определено предельное количество потоков (см. рис 1.15) для одного клиента, при увеличении которого трафик клиента определяется как враждебный и блокируется.

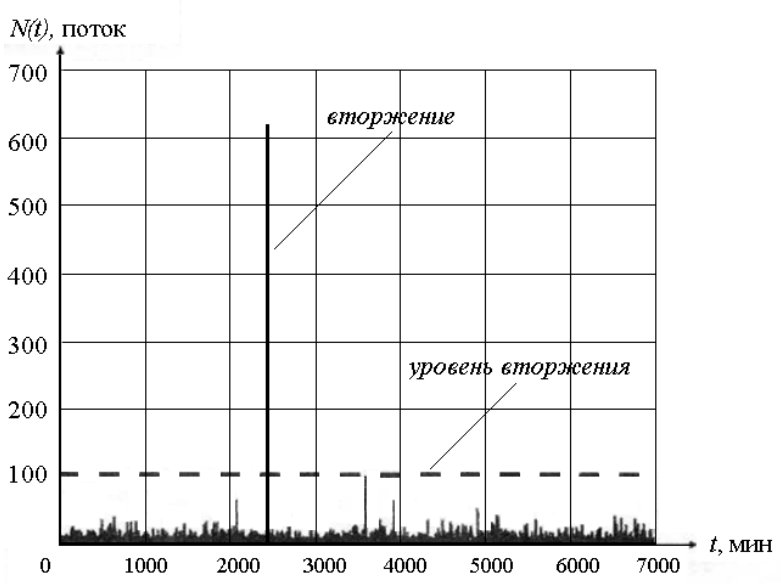


Рис. 1.15. максимальное число потоков для одного IP-адреса

На основании полученных данных, авторами был разработан метод определения IP-адресов, осуществляющих сетевые атаки двух типов: сканирование портов и DDoS-атаки, который заключается в многократном росте активных потоков, генерируемых IP-адресом, предложена методика определения уровня отсечения подозрительных IP-адресов по числу активных потоков. Результатом работы стал комплекс программно-аппаратных средств, который реализует защищенное информационное пространство для размещения Интернет-ресурсов, запущенный в эксплуатацию в одном из сегментов сети СГАУ. Функциональная схема комплекса приведена на рис. 1.16

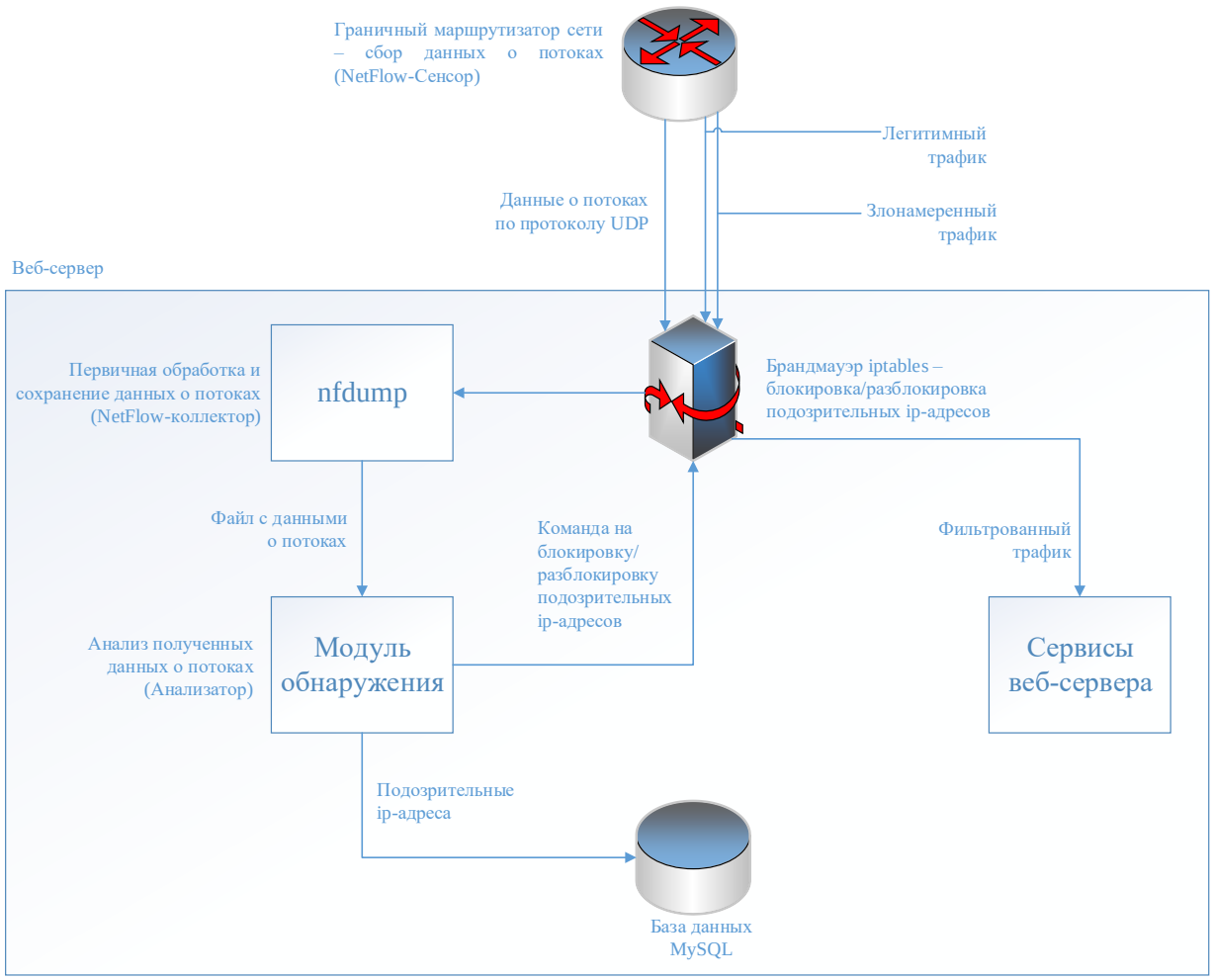


Рис. 1.16. Функциональная схема программно-аппаратного комплекс

1.4.5. Анализ и моделирование трафика в высокопроизводительных КС

А.С. Родионов, С.Д. Белов и др. исследовали входящие и исходящие потоки IP-трафика Института вычислительной математики и математической геофизики (ИВМиМГ) СО РАН (г. Новосибирск), зарегистрированные на 5-ти минутных временных интервалах за период более полутора лет [54].

Для сбора статистики использовалась ЭВМ следующей конфигурации: два процессора (Xeon 3.20GHz), 4 Гб оперативной памяти, 100 Мбит/с Ethernet-интерфейс в качестве системного интерфейса и два Ethernet-интерфейса, работающих на скорости 1 Гбит/с). На сервере была установлена операционная система OpenBSD, а для сбора интегральной статистики трафика использовалась программа CNUPM, обеспечивающая минимальную загрузку процессора и, следовательно, максимальную его производительность, а также параллельное использование других специализированных коллекторов трафика, которые анализировали не только сетевые атрибуты пакета, но и содержащиеся в пакетах данные (payloads). В качестве дополнительных программ-коллекторов, анализирующих передаваемые данные, применялась программа URLSNARF (выделяла из анализируемого потока характерные запросы BitTorrent сеансов),

являющаяся компонентой пакета DSNIFF, и система SNORT (фиксировала сигнатуры, характерные для протокола EDonkey). Исследуемый поток передавался с центрального коммутатора СПД СО РАН на основе технологии мониторирующих span-портов.

Проведенный статистический и содержательный анализа данных позволил получить информацию о структуре трафика, распределении потребителей и источников трафика внутри КС ИВМиМГ. На основе анализа используемых портов был сделан ряд интересных выводов о протоколах, использующихся в сети в соответствии с протоколом RFC 4340 DCCP (Datagram Congestion Control Protocol), который регламентирует соответствие между протоколами и портами. Например, о том, что соотношение входящего и исходящего трафиков составляет 75 % и 25 %, соответственно, от общего объема внешнего трафика института; летом объемы трафика снижаются и составляют до 75% от объемов весеннего периода; 80% трафика суммарно потребляют 5% хостов, менее 1% трафика – хостов 60%. По источникам трафика ситуация практически не меняется: на 4% хостов приходится 80% исходящего трафика, 70% хостов – менее 1%.

Основу внешнего трафика института составляли WEB-протоколы: http (порт 80), http-alt (порт 591), https (порт 443) и proxy (порты 3128, 8008, 8080 и др.). Доля протокола http (порт 80) в этом семействе составил порядка 95 %.

Анализа зависимостей переданных объемов суммарного внешнего трафика от времени обнаружил в них суточные, недельные и сезонные составляющие. Также в изученных зависимостях были обнаружены случайные выбросы (всплески) и сделан обоснованный вывод о необходимости их учета при построении математических моделей данного трафика. Полученные результаты были положены в основу статистических алгоритмов Интернет-трафика КС ИВМиМГ. По результатам проведенных исследований их авторы сделали вывод о том, что такие характеристики как плотности распределения, корреляции и фрактальные размерности временных рядов ВР, сгенерированных в соответствие с предложенными алгоритмами, оказываются достаточно близкими к аналогичным характеристикам реального трафика. Однако они не дали каких-либо рекомендаций по использованию полученных результатов для оптимизации настройки сетевого оборудования.

1.4.6. Анализ трафика с помощью метода машинного обучения

Исследователи из Турецкого академического центра сетей и информации (Мурат Сойсал) и факультета электроники и электротехники Ближневосточного технического университета Турции (Ису Гуран Шмидт) провели сравнительный анализ результатов, полученных в задаче классификации трафика по типам создающих его источников с помощью методов машинного обучения [23].

При этом они использовали трафик, собиравшийся на маршрутизаторе, через который получает доступ в Интернет Национальная Сетевая Академия Турции, насчитывающая около 90 тыс. человек персонала и более 2 миллионов студентов (рис. 1.17).

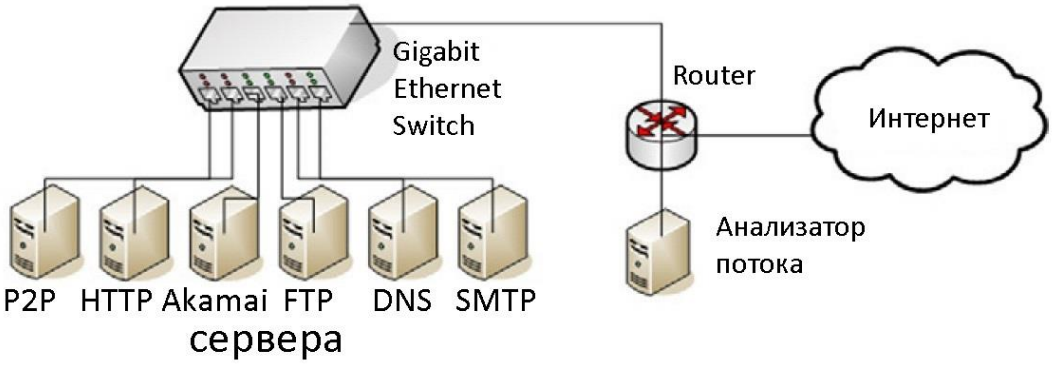


Рис.1.17. Топология сети Национальной Сетевой Академии Турции

Эксперимент проводился в 2008 г. В течение первой недели собиралась обучающая выборка трафика (PB), а в течение второй недели – тестовая (TS). Для классификации трафика они применили методы машинного обучения, в которых используются различные алгоритмические процедуры для создания классификаторов, группирующих данные в соответствующие классы на основе значений их признаков. Классификационные классы были выбраны, исходя их набора приложений, значения портов которых, регламентируется функцией управления пространством IP-адресов (Internet Assigned Numbers Authority – IANA), заведомо работавших в сети во время регистрации трафика. Всего было выделено 5 классов трафика: точка-точка (P2P), web (HTTP), доставки контента (Akamai), bulk (FTP), сервисный (DNS) и почта (SMTP). Для классификации трафика по классам потоков были использованы следующие алгоритмы машинного обучения: Бейсовские сети (BN), C4.5 дерево решений (DT) и Многослойный персептрон (DT). Эксперименты проводились с использованием программного обеспечения WEKA.

Машинные алгоритмы обучались на обучающей выборке (PB), а затем классифицировали обучающую (PB) и тестовую (TS) выборки. Результаты проведенных экспериментов представлены в таблице 1.6.

Таблица 1.6

Точность классификации трафика различными алгоритмами машинного обучения

Изучаемый дамп трафика	*Эталонные потоки*			*10% неверных портов*			*20% неверных портов*		
	BN	DT	MLP	BN	DT	MLP	BN	DT	MLP
PB	99.98	99.95	99.86	99.79	99.98	99.17	99.68	100.00	99.40
TS	99.69	99.95	88.03	98.16	97.92	86.49	77.30	64.35	58.88

Из таблицы 1.5 видно, что при изменении 10% портов в обеих выборках, точность классификации тестовой выборки снизилась на 2–12 %, а при увеличении количества

измененных портов в обеих выборках до 20% точность классификации тестовой выборки снизилась на 12–41%. При этом точность классификации обучающей выборки при этом заметно не изменилась и оставалась в пределах 0,2%. Таким образом, основным фактором, определяющим точность классификации трафика методами машинного обучения, оказывается точность извлечения первичной информации из дампов Интернет-трафика.

Консолидированная информация о современных подходах к изучению трафика уровня локальной КС разных типов во всевозможных разрезах, решающих разнородные задачи отражена в таблице 1.7.

Таблица 1.7

Работы по исследованию свойств трафика в локальных КС

Авторы/ Работа	*Источник трафика*	*Тип трафика*	*Изучаемые свойства трафика/ Уровень модели OSI, на котором проводились исследования*	*Используемые программные инструменты*	*Результаты*
Турецкий академический центр сетей и информации[1] Факультет электроники и электротехники Ближневосточного технического университета Турции[2] Исследователи: Мурат С.[1], Шмидт И.[2] [23]	Реальный: LAN	Весь	Классификация трафика с помощью методов машинного обучения/ Транспортный, Сетевой	SNORT для обнаружения вторжений, Netspec для генерации данных, flow tools для сбора трафика	Получены оценки эффективности алгоритмов машинного обучения для классификации информационных потоков в КС и даны рекомендации по их использованию.
Харьковский национальный университет радиоэлектроники. Исследователи: Карпухин А.В., Кириченко Л.О., Грицив Д.И., Ткаченко А.А. [63]	Реальный: LAN	Весь	Фрактальная размерность, показатель Херста, показатель Ляпунова / Транспортный	Wireshark для сбора информации, Tisean для анализа временных рядов методами, основанными на теории нелинейных динамических систем	Сделан вывод о том, что изученный Интернет-трафик обладает самоподобными свойствами. При этом значения показатель Ляпунова и показатель Херста оказываются зависящими от состояния изучаемой КС.
Институт ядерной физики СО РАН,[2 3 4 5] Институт вычислительной математики и математическое геофизики СО РАН,[6] Институт вычислительных технологий СО РАН Исследователи: Белов С.Д.[1], Ломакин С.В.[2], Огородников В.А.[3], Пригарин	Реальный: LAN	Весь	Фрактальная размерность, показатель Херста, показатель Ляпунова / Транспортный	Системы сбора статистики CNUPM, URLSNARF, SNORT	Получены оценки параметров квазигауссовской модели в предположении о том, что процесс передачи данных в КС является стационарным и эргодическим.

Авторы/ Работа	*Источник трафика*	*Тип трафика*	*Изучаемые свойства трафика/ Уровень модели OSI, на котором проводились исследования*	*Используемые программные инструменты*	*Результаты*
С.М.[4], Родионов А.С.[5], Чубаров Л.Б.[6]. [54]					
Астраханский государственный технический университет Исследователь: Стешенко В.В. [94]	Реальный: LAN	-	Показатель Херста/ Транспортный	Не указаны	Сделан вывод о том, что экспериментальный трафик, сгенерированный пользователями ЛВС АГТУ, является самоподобным.
Сибирский государственный университет телекоммуникаций и информатики Исследователи: Треногин Н.Г., Соколов Д.Е. [96]	Реальный: LAN	Однонаправленный трафик сервера баз данных	Оценка коэффициента Херста выборочная дисперсия и критерий согласия хи-квадрат / Транспортный.	Не указаны	Сделан вывод о том, что экспериментальный трафик между клиентом и сервером баз данных является самоподобным. Предложена имитационная модель данного типа трафика на основе модели стационарной СМО.
ООО «Лаборатория сетевых технологий». Исследователь: Перышкин С.В. [73]	Реальный: LAN	Весь	Распределения числа сессий, исходящих портов, сетевых адресов, средний размер исходящего трафика, коэффициент Херста, автокорреляция, энергетический спектр, стационарность/ Транспортный и сетевой	RapidMiner	Сделан вывод о том, что в общем потоке трафика по диапазонам значений ключевых параметров и их сочетаний можно выделить устойчивые фрагменты, обладающие высокой степенью стационарности
Уральский государственный технический университет — УПИ	Реальный: LAN	Весь	Анализ работоспособности ряда математических методов в задаче выявления сетевых	Мониторинг каналов - snmpget из пакета net-snmp, мониторинг инфраструктуры - Remote	Разработана комплексная методика выявления аномалий Интернет-трафика, основанная на комбинировании методов

Авторы/ Работа	*Источник трафика*	*Тип трафика*	*Изучаемые свойства трафика/ Уровень модели OSI, на котором проводились исследования*	*Используемые программные инструменты*	*Результаты*
имени первого Президента России Б. Н. Ельцина Исследователи Э.В. Афонцевым и С.В. Поршневым [55]			аномалий по реализациям реального Интернет-трафика	Monitoring (RMON), для сбора данных - tcpdump, анализ данных - Matlab	корреляционного анализа параметров потоков трафика и моделирования временных рядов Хольта-Винтерса, позволяющая выявлять наиболее значимые для операторов связи аномальные состояния трафика.
Самарский государственный аэрокосмический институт Исследователи А.А. Гальцев, А.М Сухов [59]	Реальный/Моделируемый:LAN/Магистральный	Весь	Проведен обзор существующих моделей трафика и методов по выявлению аномальных состояний сети, исследовали системные связей трафика на уровне агрегированных состояний сети.	SNORT для обнаружения вторжений	Результатом работы стал комплекс программно-аппаратных средств, который реализует защищенное информационное пространство для размещения Интернет-ресурсов.

1.5. Анализ результатов исследований Интернет-трафика в низкоскоростных сетях (КС провайдера, последняя миля)

1.5.1. Анализ Интернет-трафика в КС Интернет-провайдера

Сравнительный анализ методов расчета параметров трафика (интенсивности обслуживания, интенсивности поступления пакетов, среднего времени ожидания пакетов в очереди, длины очереди и коэффициент загрузки) для узла КС Интернет-провайдера был проведен И.А. Меркуловой [70]. Автором были использованы данные реальной КС Интернет-провайдера компании ООО «Опти-Телеком», полученные за один рабочий день. Проведенные исследования позволили, в том числе, оценить общее количество принятой пользователями информации, скорость передачи данных, полосы пропускания, предоставленные каждому из пользователей, а также рассчитать следующие показатели: длительности временных интервалов между поступлениями пакетов, понимаемой в указанном в разделе 1.4.1 смысле, среднее время обработки пакетов и коэффициент загрузки сети, показавший, что КС загружена лишь на 2% при использовании полосы пропускания 2 Мбит/с и на 64%, если полоса пропускания составляет 64 Кбит/с.

На основе результатов имитационного моделирования данной КС были получены оценки среднего времени ожидания пакетов в очереди при загрузке сети $\rho = 0.02$, $\rho = 0.64$, составившие 0.0029 мс и 7.29 мс, соответственно, и сделан обоснованный вывод о том, что при двухпроцентной загруженности сети очередь пакетов на маршрутизаторе практически отсутствует, а при загрузке 64 % пользователи получают данные с небольшими задержками (средняя длина очереди составляет 1 пакет), которые можно уменьшить путем увеличения общей полосы пропускания. Оценка среднего времени ожидания для 4 пользователей с интенсивностью обслуживания $\mu = 5$ пакетов/мс составила 0.98 мс, а оценка средней длина очереди – 3 пакета.

Проведенные исследования позвали сформулировать рекомендации по выбору минимально допустимой ширины полосы пропускания, выделенной провайдером, обеспечивающей предоставления заявленного качества требуемого качества обслуживания пользователей. Также была выявлена ожидаемая тенденция роста среднего времени ожидания и средней длины очереди при увеличении коэффициента загрузки сети.

По утверждению автора полученные им результаты могут быть использованы при проектировании и эксплуатации КС Интернет-провайдеров, при выборе размера памяти буфера для исключения перегрузок в сети и гарантированной скорости трафика, а также полезны при внедрении нового коммутационного оборудования. Однако, принимая во внимание, что информационные потоки в современных КС не могут быть в рамках модели

случайных величин с пуассоновским распределением, можно поставить справедливость данного утверждения И.А. Меркуловой по сомнение.

1.5.2. Анализ нагрузки, создаваемой абонентами ADSL при безлимитном доступе в сеть Интернет

Анализ нагрузки, создаваемой абонентами широкополосного Интернет-канала с целью количественного обоснования, конструкторских решений, принимаемых на этапе проектирования, был проведен Д.В. Агеевевым и Д.В. Евлаш [50]. Они использовали данные о трафике, созданном группой абонентов Интернет-провайдера, подключенных к сети Интернет по технологии широкополосного доступа ADSL, которые работали по безлимитным тарифным планам. Общее количество абонентов составляло 1338 чел., из них 538 человек были подключены в соответствие с тарифным планом «256 кб/с» (40.2 %); 82 абонента (6.1 %) – «512 кб/с»; 36 абонентов (2.8 %) – «1024 кб/с»; 682 абонента (50.9 %) – «2048». Сбор данных осуществлялся на протяжении месяца. При этом фиксировался как входящий, так и исходящий трафик каждого из абонентов.

Результаты исследований показали, что средняя загруженность каналов связи и время, необходимое для приема или передачи месячного объема данных, при максимальной скорости доступа не зависит от скорости доступа абонента к сети Интернет.

Также были получены количественные оценки временных параметров загруженности каналов связи: средняя загруженность каналов связи для входящего трафика – 0.038%, а для исходящего – 0.015%, время необходимое для приема/передачи месячного объема данных, на максимальной скорости доступа составляет 1.15 суток для входящего трафика и 0.44 суток для исходящего трафика.

Полученные временные параметры авторы рекомендовали использовать для оценки величины нагрузки, создаваемой абонентами в процессе проектирования широкополосных сетей абонентского доступа в Интернет, вместо параметра «интенсивность поступления информации в сеть».

В связи с тем, что КС данного уровня не являются предметом нашего исследования, однако, представляют интерес с точки зрения оцениваемых характеристик трафика и используемых для этого инструментов далее приведена сводная таблица 1.8 с указанием исследований, выполненных в данном направлении, и полученные при этом результаты.

Таблица 1.8

Работы по исследованию свойств трафика в низкоскоростных КС

Авторы/ Работа	*Источник трафика*	*Тип трафика*	*Изучаемые свойства трафика/ Уровень OSI на котором проводились исследования*	*Используемые программные инструменты*	*Результаты*
Московский энергетический институт Исследователи: В.В. Платов и В.В. Петров [30]	Реальный: беспроводная сеть провайдера (MAN)	Весь	Оценка коэффициента Херста / Транспортный.	Tcpdump для сбора статистики	На основании полученных значений параметра Херста делается вывод о том, что трафик беспроводных сетей является самоподобным.
Харьковский национальный университет радиоэлектроники. Исследователи: Агеев Д.В., Евлаш Д.В. [50]	Реальный: сеть провайдера (MAN)	Весь	Количественные характеристики загруженности КС/ Транспортный.	Не указано	Сделан вывод о том, что средняя загруженность каналов связи и время, необходимое для приема или передачи месячного объема данных, при максимальной скорости доступа не зависит от скорости доступа абонента к сети Интернет.
Харьковский национальный университет радиоэлектроники1 Запорожский национальный технический университет Исследователи: Агеев Д.В.1, Игнатенко А.А.1, Копылев А.Н.2 [31]	Моделирование On/Off	Весь	Получены оценки параметров потоков, представляющие собой результат агрегации потоков, описываемых моделью фрактального броуновского движения, потоки, представляющие собой классическое броуновское блуждание, а также потоки, создаваемые On/Off источниками, / Транспортный.		Предложен метод выбора параметров мультисервисных КС, обеспечивающие заданные технические характеристики.

1.6. Анализ результатов исследования Интернет-трафика в высокоскоростных магистральных КС

1.6.1. Исследование самоподобной структуры Интернет-трафика в беспроводной КС

Исследование особенностей Интернет-трафика, передаваемого в магистральном канале беспроводной КС одного из крупных городских Интернет-провайдеров (ЗАО «Виплайн», Воронеж), было проведено В.В. Платовым и В.В. Петровым [76]. Для этого на коммуникационном узле провайдера устанавливалась шестисекторная антенна, к каждому сектору которой была подключена точка доступа (Access Point) стандарта IEEE 802.11b.

Всего к шести точкам доступа провайдера были подключены около 40 беспроводных клиентов. С физической точки зрения беспроводной клиент представлял собой точку доступа аналогичную провайдерской, к которой по локальной сети Fast Ethernet был подключен один или несколько хостов. Все точки доступа работали как прозрачные мосты для протоколов верхних уровней. Маршрутизацию всех соединений, а также контроль и управление пропускной способностью (Traffic Shaping) осуществлял маршрутизатор, который предоставлял каждому из беспроводных клиентов собственные полосы пропускания в диапазоне 64 до 512 Кбит/с.

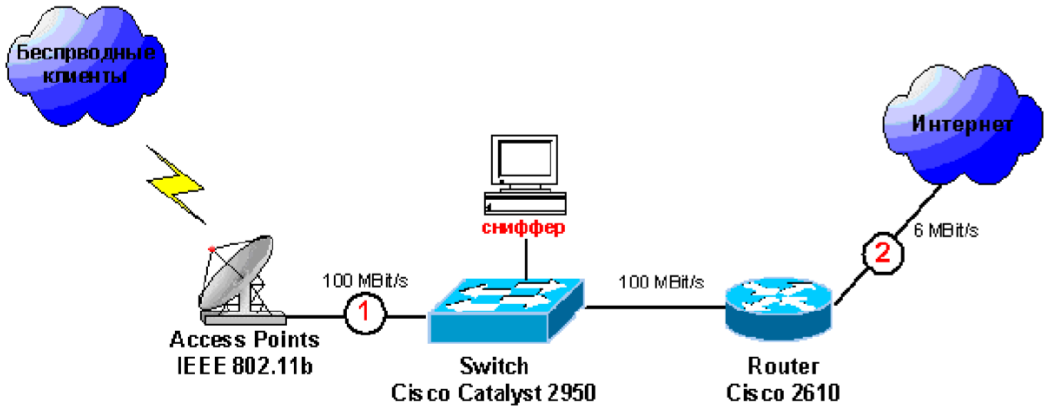


Рис. 1.18. Логическая схема организации эксперимента по снятию трафика

Реализации дампов трафика, зарегистрированные в общей сложности в течение 7 часов, были разделены на четыре части. Длительности временных интервалов, на которых они были изучены свойства Интернет-трафика, представлены в таблице 1.9.

Таблица 1.9

Участки временных реализаций

Номер участка	*Положение на оси времени, секунд с начала реализации*	*Длительность*
1	0,52 - 2257,58	45 143 отсчета по 0,05 сек; 0,63 ч
2	3954,87 - 6142,53	43 753 отсчета 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

Далее на каждом из выбранных участков на последовательных временных интервалах (см. таблицу 1.8) были вычислены значения переданных объёмов данных (зависимости «мгновенных значений» переданных объёмов данных, отнесенные к длительности последовательных временных интервалов, от времени – $V_{k,i} = f(t_i)$, k – номер части дампа трафика, $k = \overline{1,4}$, i – номер последовательного временного интервала). Далее для каждой из выбранных частей дампа Интернет-трафика были оценены среднее $\bar{V}_k$, среднеквадратическое отклонения (СКО) $\mathrm{std}(V_k)$ и параметра Херста H соответствие с методами Виттла (МВ) и Эбри-Вейча (МЭВ)) временных рядов $V_{k,i}$. Основные характеристики Интернет-трафика приведены в таблице 1.10.

Таблица 1.10

Параметры интернет-трафика беспроводной КС

Параметр	*Участок 1*	*Участок 2*	*Участок 3*	*Участок 4*
$\bar{V}_k$, байт/сек	$8{,}98 \cdot 10^4$	$1{,}23 \cdot 10^5$	$1{,}18 \cdot 10^5$	$2{,}88 \cdot 10^5$
$std(V_k)$, байт/сек	$9{,}69 \cdot 10^4$	$1{,}90 \cdot 10^5$	$1{,}25 \cdot 10^5$	$6{,}34 \cdot 10^5$
$std(V_k)/\bar{V}_k$	1,07	1,54	1,05	2,2
H (МВ)	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])
H (МЭВ)	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])

В связи с тем, что для всех изученных зависимостей значение *H* было больше 0.5, авторы сделали вывод о наличии самоподобных свойств в трафике современных телекоммуникационных сетей, использующих, в том числе, технологии беспроводного доступа IEEE 802.11b. Далее было проведено сравнение параметров изученного Интернет-трафика беспроводной КС с аналогичными параметрами проводных КС. На основе результатов сравнения авторы сделали вывод о том, что существенные отличия между значениями показателей Херста в проводных и беспроводных КС отсутствуют, несмотря на различные принципы функционирования канального и физического уровней.

1.6.2. Анализ результатов исследования трафика в магистральном Интернет-канале

Исследователи из Лионской высшей школы физики (Пьер Боргнат, Гуилами Деваэль, Патрис Абри) и из Токийского национального института информатики (Кенсуке Фукуда) провели анализ дампов трафика в магистральном Интернет-канале, проложенном через Тихий океан между США и Японией, за период с 2001 по 2007 гг. [5]. Исследуемые дампы Интернет-трафика (MAWI-дампы [32]) регистрировались ежедневно в течение 15 минут с 14:00 до 14:15 (стандартное японское время, UTC+9). Для обеспечения конфиденциальности в MAWI дампах реальные IP-адреса, первоначальное содержимое предаваемых пакетов, а также вся личная информация заменены на случайные данные. Однако вносимые при этом изменения не меняют собственно структуры трафика (размеров пакетов, направления передачи пакетов, протоколов передачи и т.д.) и, следовательно, его свойств.

Цель исследования состояла в описании динамики изменения относительных объемов различных типов Интернет-трафика, переданных в магистральном канале в течение 7 лет, за выбранный период времени, обусловленной как расширением полосы

пропускания канала, которая в 2007 г. была увеличена со 150 Мбит/с до 1 Гбит/с, так и изменением собственно структуры Интернет-трафика, и долгосрочном прогнозирования их дальнейшей эволюции на основе метода эскизов [21].

Зависимости относительных объемов различных типов Интернет-трафика, переданных в магистральном канале в течение одного месяца по различным протоколам, представлены на рисунке 1.16 снизу-вверх объемы данных переданных по протоколам: Ping, DNS, общие службы, уязвимости MS, Sasser, HTTP, бродкасты, предполагаемый P2P, обнаруженный P2P, другие TCP/UDP, GRE (US2Jp) или INLSP (Jp2US), соответственно.

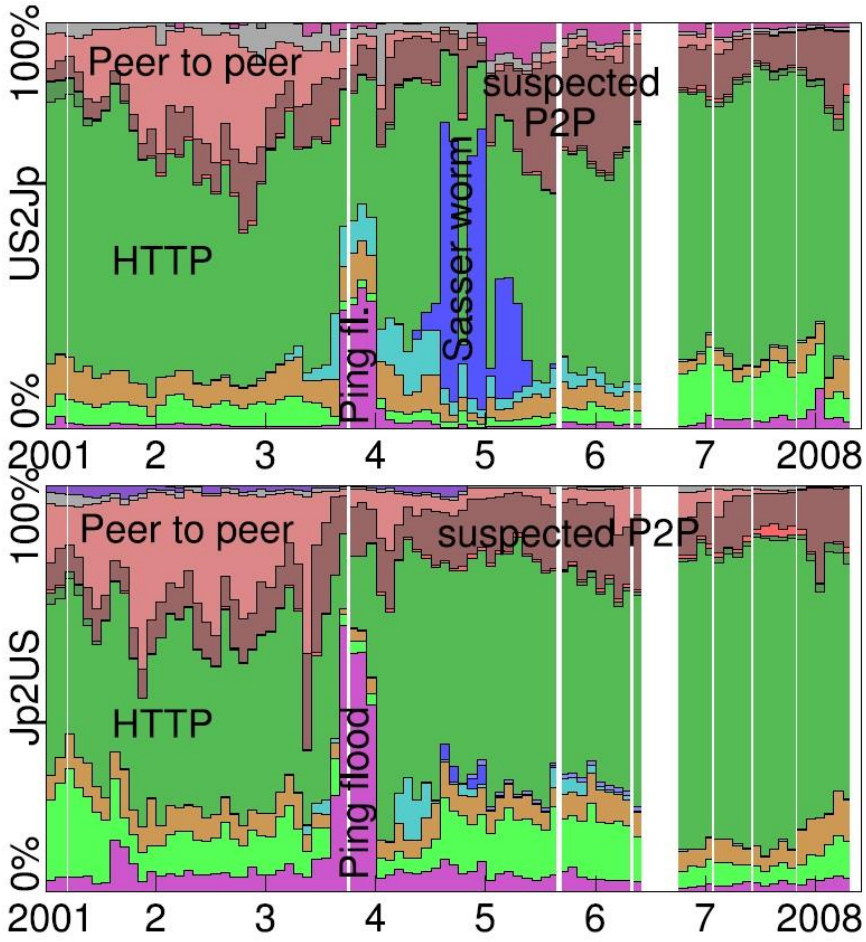


Рис.1.19. График зависимостей относительных объемов Интернет-трафика магистрального Интернет-канала, переданного по каждому из выбранных протоколов в период с 2001 по 2008 гг. [27]

Из рис. 1.19 видно, что около 90% объема изученного трафика было передано по протоколам TCP и UDP, около 5% – по протоколу ICMP (Ping). Отметим, что также было обнаружено, что некоторая часть Интернет-трафика была передана по протоколу туннелирования GRE (около 5% трафика из Японии в США) и протоколу безопасности Интернет INSLP (около 1%).

Отметим, что в обсуждаемом исследовании изучались агрегированные на больших временных интервалах (месяцы и годы) характеристики Интернет-трафика, что позволило

описать динамику долгосрочных макроуровневых изменений трафика, в то время как обоснованного выбора технических характеристик сетевого оборудования необходимо понимать особенности Интерент-трафика на временных интервалах, существенно меньшей длительности.

Консолидированная информация о современных подходах к изучению трафика на уровне магистральных вычислительных сетей разных типов во всевозможных разрезах, решающих разнородные задачи отражена в таблице 1.11.

Таблица 1.11

Работы по исследованию самоподобных свойств сетевого трафика магистральных интернет-каналов

Авторы/ Работа	*Источник трафика*	*Тип трафика*	*Изучаемые св-ва трафика/ Уровень OSI на котором проводились исследования*	*Используемые программные инструменты*	*Результаты*
Сибирский Федеральный Университет. Исследователи: Симаков Д.В., Кучин А.А. [92]	Реальный: магистральный	Весь	Плотности распределения вероятностей интенсивности трафика, оценка коэффициента Херста, вычисление стандартного отклонения и математического ожидания. / Уровень модели OSI: Транспортный.	Wireshark для сбора статистики и анализа данных	Сделан вывод о том, что агрегированный трафик магистрального канала сети Интернет обладает высокой дисперсией и степенью самоподобия и характеризуется распределением интенсивности, обладающим «тяжелым хвостом».
Мичиганский Университет[1], Университет Джорджии[2] Исследователи: Стилиан С[1], Челву П[2] [42]	Реальный: магистральный	Весь	Оценка параметра Херста с помощью вейвлет-спектра. Моделирование временных рядов методом дробного Гауссовского шума. / Уровень модели OSI: Транспортный.	Не указано. Судя по графикам, для анализа использовался MATLAB	Вывод о том, что вейвлет-спектр может быть использован для получения относительно достоверных оценок параметра Херста стационарного трафика.
Лионская высшая школа физики[1] Токийский национальный институт информатики[2] Исследователи: Боргнат П.[1], Деваэль Г.[1], Абри П.[1], Фукуда К.[2] [5]	Реальный: магистральный	Весь	Поиск девиаций в статистических параметрах трафика/ Уровень модели OSI: Транспортный.	Не указано.	Выделены девиации в статистических параметрах трафика за 7 лет на основе чего сформировали обучающую выборку для эскизного метода машинного обучения прогнозирующею возникновение возможных отклонений поведения трафика

1.7. Анализ результатов исследования трафика в смежных уровнях сети

1.7.1. Анализ трафика в анонимной сети Tor

Абделбери Шабане, Пьер Манилс и Мохамед Али Каафар из Французского Института науки и техники INRIA провели исследование анонимной сети Tor [6]. Tor – коммутируемый сервис анонимного соединения с низкой задержкой, представляющий собой наложенную сеть так называемых луковичных роутеров, обеспечивающих анонимизации TCP приложений (web, p2p, secure shell и др.).

Авторы собирали Интернет-трафик с 6 выходных Tor серверов, 2 из которых были установлены в США, 2 в Европе (Франция, Германия), 2 в Азии (Япония, Тайвань). Общий период сбора трафика составил 23 дня. При полосе пропуская в 100 Кб/с объем трафика составлял около 20 Гб в день, а общий объем изученных данных составил 2.6 ТБ. Для анализа Интернет-трафика был использован программный инструмент OpenDPI (Deep packet inspection).

Было изучено, какие протоколы используются в сети Tor. Для анализа учитывались только те потоки, в которых был успешно передан хотя бы один пакет, т.е. были отброшены все потоки меньше четырех пакетов (трехэтапное квитирование TCP и один пакет данных). Было замечено, что огромное количество соединений не в состоянии добраться до места назначения (передается один syn пакет или только таймауты возвращаются после процедуры рукопожатия). Такие неудачные попытки соединения составляют около 40% от всего трафика, передаваемого через исследуемый выходной сервер. Вероятнее этот трафик генерируется BitTorrent клиентами, пытающимися взаимодействовать с клиентами, которые больше не доступны. Так как данный трафик не представляет конкретных приложений, он не был учтен в исследовании, остальной представлен в таблице 1.12.

Таблица 1.12

Результаты анализа Интернет-трафика в сети Tor

Протокол	*Пакеты (млн.)*	*Объем и относительный объем переданной информации по данному протоколу*	*Потоки (тыс.)*
HTTP	185.7 (34.31%)	136 GB (36.44%)	4735 (68.57%)
BitTorrent (чистый)	136.8 (25.27%)	93 GB (24.92%)	320.5 (4.64%)
SSL	28.5 (5.26%)	20 GB (5.37%)	126 (1.83%)
Другие P2P/обмен файлами	5.7 (1.07%)	4.4 GB (1.17%)	15 (0.22%)
Ненадежные (ftp, telnet, email и др.)	1.3 (0.26%)	1.2 GB (0.32%)	6 (0.09%)
Система мгновенных сообщений	6.5 (1.22%)	972 MB (0.26%)	119 (1.72%)
Известные (другие общепризнанные протоколы)	18.2 (3.37%)	22.6 GB (6.04%)	1173 (16.99%)

Протокол	*Пакеты (млн.)*	*Объем и относительный объем переданной информации по данному протоколу*	*Потоки (тыс.)*
«Неизвестные»	158 (29.21%)	95 GB (25.47%)	410 (5.94%)
Итого	541.5	373.6 GB	6905

Из таблицы 1.12 видно, что сеть Tor в основном используется для посещения Интернет-страниц, так называемого серфинга, (предположительно для целей анонимизации или доступа к ресурсам, запрещенным в месте нахождения пользователя), что достаточно ожидаемо. Далее по объемам передаваемой информации следуют различные протоколы прямого обмена контентом между пользователями (Torrent, P2P и др.). Оставшаяся достаточна значимая долю объемов Интернет-трафика была передана по различным шифрованным протоколам.

Для получения информации о географическом местоположении клиентов (таблица 1.13), а также о передаваемой ими информации была имитирована классическая атака «человек посередине». Взлом происходил на исходящих узлах, на которые трафик проходил в открытом виде. Взлом заключался в изменении первого адреса в списке адресов клиентов, отдаваемых Текером на адрес контролируемого BitTorent клиента. При получении последующих пакетов во время процедуры BitTorrent-рукопожатия определялось, использует ли обманутый клиент шифрование или нет. Если клиент использовал шифрование только для связи с Трекером, то контролируемый BitTorent-клиент имел возможность видеть его публичный IP адрес, если для соединения между клиентами тоже использовалось шифрование, то публичный адресом был адрес выходного сервера.

Таблица 1.13

Территориальное распределение клиентов Tor (топ 7)

Страны	*Доля*	*Накопительный итог*
Германия	14.7%	14.7%
США	12.8%	27.5%
Польша	11.08%	38.58%
Румыния	7.7%	46.28%
Российская Федерация	7.3%	53.58%
Китай	5.8%	59.38%
Франция	4.3%	63.68%
Другие	36.32%	100 %

Из таблицы 1.13 видно, что наиболее активными пользователями анонимной сети Tor являются жители Германии, США и Польши. Однако использовать эти результаты для обоснования выбора характеристик оказывается невозможным.

Консолидированная информация о современных подходах к изучению трафика на смежных уровнях вычислительных сетей разных типов во всевозможных разрезах, решающих разнородные задачи отражена в таблице 1.14.

Таблица 1.14

Работы по исследованию самоподобных свойств трафика вычислительных сетей смежного типа

Авторы/ Работа	*Источник трафика*	*Тип трафика*	*Изучаемые св-ва трафика/ Уровень OSI на котором проводились исследования*	*Проверка самоподобия / распределения первых разностей*	*Результаты*
Поволжский гос. Университет Телекоммуникаций и Информатики. Исследователь: ст. преподаватель Буранова М.А [58]	Реальный: ПК - сервер	Мультимедийный	Плотности распределения вероятностей, оценка параметра Херста посредством R/S-анализа/ Транспортный	Вычисление параметра Херста / Не проводилась	Вывод о том, что мультимедийный трафик в сети Интернет обладает свойствами самоподобия (интервалы между пакетами, последовательность длин пакетов и число пакетов в единицу времени)
Исследователь: Шелухин О.И. [100]	Моделирование ФГШ	-	Оценка параметра Херста посредствам: Изменения дисперсии, R/S статистики, Периодограммного метода, метода Виттла/ Транспортный	Вычисление параметра Херста / Не проводилась	Вывод о том, что основной причиной самоподобия телекоммуникационного трафика является объединение множества отдельных, хотя и сильно изменчивых ON/OFF-источников.
Московский технический университет связи и информатики Исследователь: Шмелев И.В. [102]	Нет информации	Однонаправленный входящий поток	Рассмотрена математическая модель трафика, представляющего собой сумму двух независимых самоподобных процессов с различными значениями параметра Херста/ Транспортный	Вычисление параметра Херста для мат. моделей трафика / Не проводилась	Предлагается использовать фрактальное броуновское движение с различными значениями параметра Херста для описания отклонения интенсивности трафика
Харьковский национальный университет радиоэлектроники1 Запорожский национальный технический университет2 Исследователи: Агеев Д.В.1, Игнатенко А.А.1, Копылев А.Н.2 [51]	Моделирование On/Off	Весь	Рассмотрен агрегированный поток для случая объединения потоков, описываемых такими моделями как: фрактальный Броуновский трафик, потоки от индивидуальных источников, потоки без эффекта самоподобия/ Транспортный	Использовался опыт предыдущих авторов/ Не проводилась	Предложен метод синтеза мультисервисных сетей в качестве математических моделей потоков на различных участках мультисервисной телекоммуникационной сети с использованием модели самоподобных процессов
Французский Институт науки и техники INRIA Исследователи: Шабане А., Манилс П. и Али М [6]	Реальный: Распределённые WAN сети	Весь	Геолокационное распределение клиентских хостов, отношение различных типов переданного трафика относительно приложений/ Транспортный	Вычисление параметра Херста / Не проводилась	Благодаря проведенным исследованиям удалось выяснить, кто и для каких целей использует Tor сети. Также было обнаружено, что многие пользователи используют Tor как бесплатный шифрованный прокси-сервер, устанавливая количество узлов равным единице.
Астраханский государственный технический университет Исследователи: Ажмухамедов И.М., Марьенков А.Н. [53]	Результаты исследований других авторов	-	Были использованы результаты исследований трафика других авторов.	Не проводились	На основе результатов исследований о самоподобии трафика полученных другими авторами была разработана методика отслеживания угроз сетевой безопасности

Таким образом, проведенный анализ работ по исследованию трафика КС позволяет сделать вывод о том, что сравнение результатов независимых исследований свойств Интернет-трафика и оценка их адекватности встречает принципиальные трудности, обусловленные тем, что, как правило, выбор измерений, в которых изучались свойства информационных потоков в КС, как правило, не обосновывался (с нашей точки зрения данная ситуация является следствием отсутствия единой методики подобных исследований), а кроме использовались:

– реализации дампов трафика, недоступные широкому кругу исследователей;

– различные программные инструменты, которые в ряде публикаций не указываются.

Необходимо также отметить, что спорной оказывается изначальная цель данных исследований, состоявшая в изучении самоподобных свойств информационных потоков в КС (точнее, оценивании соответствующих параметров фрактального броуновского движения [32,85,50 и др. работы приведённые выше в ходе анализа]), но не системного изучения собственно свойств данного трафика. При этом вопрос об адекватности данной математической модели исследуемому объекту, обладающему, как показано выше, свойствами сложной системы, авторами не изучался.

Таким образом, проведенный анализ состояния предметной области показал, что для объективного всестороннего рассмотрения исследования свойств информационных потоков реального трафика в магистральном Интернет-канале необходимо разработать математическое и программное обеспечение для анализа трафика, передаваемого в высокоскоростных магистральных каналах передачи данных.

1.8. Постановка задачи исследования

Проведенный анализ современного состояния области исследования показал.

1. Информационные потоки, передаваемые в современных КС (трафик), являются объектом активного изучения большого количества исследователей. При этом во многих работах рассматриваются искусственно смоделированные случаи (определенные типы трафика (web, video и т.д.), однонаправленные потоки, трафик модели или сервера и т.д.), что не позволяет получить достоверного представления о реальных процессах, протекающих в вычислительных сетях. В этой связи, особый интерес, как с научной, так и практической точек зрения, представляет собой реальный трафик вычислительных сетей.

2. Наибольший интерес для изучения представляют магистральные высокоскоростные Интернет-каналы, так как они:

– являются основой связи сетей крупнейших провайдеров телекоммуникационных услуг и сети Интернет, в целом;

– предоставляют полный спектр сервисов каждому из типов пользователей;

– стабильная работа магистральных Интернет-каналов является критичным фактором, как для поставщиков каналов связи, так и для поставщиков сервисов, использующих данные каналы.

3. В настоящее время отсутствуют универсальные программные инструменты и общепризнанные методики для анализа Интернет-трафика, что затрудняет верификацию результатов экспериментальных исследований, проведенных различными авторами

4. В подавляющем большинстве работ по исследованию особенностей Интернет-трафика в КС отсутствует интерпретация полученных результатов с точки зрения с точки зрения выбора технических характеристик сетевого оборудования и его настройки.

Таким образом, проведенный анализ состояния предметной области показал, что несмотря на многочисленные исследования свойств информационных потоков в КС, цель исследования, состоящая в разработке научно обоснованных программных инструментов, также проведении самостоятельных исследований Интернет-трафика и их трактовка с точки зрения выбора технических характеристик и режимов работы сетевого оборудования является актуальной.

Для достижения поставленной цели требуется решить следующие задачи:

1. Разработать математическое и программное обеспечение для анализа трафика, передаваемого в высокоскоростных магистральных каналах передачи данных.

2. Провести самостоятельные исследования свойств информационных потоков реального трафика в магистральном Интернет-канале.

3. Дать интерпретацию результатов исследования свойств информационных потоков реального трафика в магистральном Интернет-канале.

ГЛАВА 2. Разработка математического и программного обеспечения для анализа трафика, передаваемого в высокоскоростных магистральных каналах передачи данных

2.1. Анализ особенностей объекта исследования

В качестве объекта исследований были выбраны ежедневные пятнадцатиминутные дампы трафика магистрального канала между США и Японией, находящиеся в общедоступном хранилище – MAWI (Measurement and Analysis on the WIDE Internet) [27]. Здесь находится как актуальная информация о текущем состоянии Интернет-канала, так и архивные данные, начиная с 1999 г. Для сохранения конфиденциальности информации в дампе обезличены сведения о конкретных пользователях. Кроме того, содержание пакетов заменено на сгенерированные случайным образом данные, которые, однако, хранят в себе всю необходимую для исследований техническую информацию (номера портов, протоколы, размер пакета и т.д.). Задача дальнейшего анализа дампа Интернет-трафика относится к задачам полного синтаксического анализа (парсингу) текстовых файлов.

Перечень исследованных в работе дампов приведен в таблице 2.1.

Таблица 2.1

Перечень исследованных в работе дампов

Дата создания файла	*Имя файла*	*Размер файла*
27.10.14	201410271400.dump	9249.41MB
28.10.14	201410281400.dump	8983.29MB
29.10.14	201410291400.dump	8435.98MB
30.10.14	201410301400.dump	8892.00MB
31.10.14	201410311400.dump	8407.92MB
01.11.14	201411011400.dump	7153.41MB
02.11.14	201411021400.dump	6542.95MB

2.2. Разработка технология обработки дампа Интернет-трафика

Необходимо отметить, что сегодня существует большое количество различных сетевых анализаторов Интернет-трафика (, обеспечивающих создание и просмотр дампов сетевого трафик tcpdump [44] и wireshark [43]). Однако данные инструменты не позволяют проводить полноценный, с математической точки зрения, анализ свойств трафика, потому что у них отсутствует возможность непосредственной передачи информации, извлеченной из дампа, в какой-либо математический пакет для ее последующей обработки.

Основная причина, обусловившая описанную выше ситуацию, состоит в том, что размеры дампов, содержащих исходную информацию о трафике, оказываются столь большими (несколько сотен МБ и более), что стандартные средства для работы с данными файлами и анализа их содержимого оказываются неработоспособными. В сложившейся

ситуации необходима разработка соответствующей технологии, обеспечивающей возможность полноценной, как синтактической, так и математической обработки дампа.

2.2.1. Обоснование выбора программного обеспечения для обработки дампов Интерент-трафика

Разработку технологии работы с дампом целесообразно начать с выбора программного инструмента, в котором реализованы математические методы обработки данных. Перечень изученных программных продуктов, которые потенциально можно использовать для обработки дампа трафика, представлен в табл. 2.2.

Таблица 2.2

Сравнительная таблица средств математической обработки данных

Критерии сравнения	*Наименования программных пакетов*			
	Scilab	*Octave*	*Sage*	*MATLAB*
Возможность построения графиков	+	Использует дополнительные программы: Gnuplot и Grace	+	+
Возможность имитационного моделирования	+	–	–	+
Функционал	Большое количество встроенных математических функций	Большое количество встроенных математических функций	Ограниченное количество встроенных математических готовых	Большое количество встроенных функций математических и пакетов расширений
Максимально возможный размер системы линейных алгебраических уравнений	256×256	2048×2048	1024×1024	2048×2048
Возможность компилирования других языков	C, C++, Fortran	C, C++, Fortran	Phyton	C, C++, Visual C, Fortran
Удобство установки и настройки	Не требует дополнительных настроек после установки	Необходима дополнительная настройка для запуска графического интерфейса	Необходима дополнительная настройка шрифтов для корректного отображения в графическом окне	Не требует дополнительных настроек после установки

Критерии сравнения	*Наименования программных пакетов*			
	Scilab	*Octave*	*Sage*	*MATLAB*
Совместимость с операционными системами	Linux, MacOS X, Windows	Linux, UNIX, Cygwin, Windows, Android	Linux, Unix, Windows	Microsoft Windows, Mac OS, Linux, Unix
Разработчик	Scilab Enterprises	John W. Eaton	Уильям Стей	The MathWorks
Техническая поддержка	Нерегулярные обновления. Получение информации на форумах и в сообществах.	Нерегулярные обновления. Получение информации на форумах и в сообществах.	Нерегулярные обновления. Получение информации на форумах и в сообществах.	Регулярные обновления. Получение информации онлайн по телефону, а также на форумах и в сообществах.
Стоимость	Бесплатно. Лицензия GNU GPL	Бесплатно. Лицензия CeCILL (совместима с GNU GPL v.2)	Бесплатно. Лицензия GNU GPL	1500 рублей для студентов. Лицензия проприетарная

Из таблицы 2.2 видно, что рассмотренные программные продукты, в целом, являются достаточно сходными по реализованному в них функционалу. При этом пакеты Scilab [38], Octave [15], Sage [37] являются свободно распространяемыми программными продуктами. Однако у данных пакетов отсутствует техническая поддержка, кроме того существуют проблемы совместимости с некоторыми языковыми таблицами (кодировками), меньшее количество доступных расширений, дополнений и реализованных в них математических функций [28, 29]. Пакет MATLAB [24] не имеет недостатков реализации, перечисленных у конкурентов, но является платным коммерческим продуктом. Однако для студентов и аспирантов плат за использование пакета MATLAB, работающего под управлением операционных систем (ОС) семейства Windows является вполне приемлемой (45$), а соответствующая версия пакета, работающая под управлением ОС семейства UNIX, является бесплатной. При этом MATLAB является инструментом, который сегодня активно используется как членами научного сообщества, так и профессионалами различных сферах человеческой деятельности. В этой связи в качестве базового программного средства, используемого для анализа дампа Интернет-трафика в высокоскоростном магистральном канале, был выбран пакет MATLAB.

2.2.2. Обоснование выбора операционной системы для анализа дампов Интернет-трафика

Важность выбора операционной системы обусловлена тем, что работа с дампом сетевого трафика является достаточно ресурсоемкой задачей, поэтому здесь целесообразно использовать такую ОС, которая потребляет на собственные нужды минимум вычислительных ресурсов.

Проведенный анализ известных ОС показал, что ОС семейства Windows (7, 8, 2008 server, 2012 server) [93] предъявляют следующие минимальные требования к вычислительным ресурсам:

– процессор с тактовой частотой 1 ГГц или выше;

– 1 ГБ (для 32-разрядной системы) или 2 ГБ (для 64-разрядной системы) оперативной памяти (ОЗУ);

– 16 ГБ (для 32-разрядной системы) или 20 ГБ (для 64-разрядной системы) пространства на жестком диске;

соответственно, Unix-подобные ОС операционные системы (Ubuntu, BSD, OS X, Red hat и т.д.) [97]:

– процессор с тактовой частотой 0,8 ГГц или выше;

– 512 МБ оперативной памяти (ОЗУ);

– 5 ГБ пространства на жестком диске.

В связи с тем, что требования Unix-подобных ОС к вычислительным ресурсам оказывается ниже, чем у ОС семейства Windows, было выбрано последнее семейство ОС. Основные характеристики наиболее распространенных Unix-подобных ОС представлены в табл. 2.3.

Таблица 2.3

Сравнительная таблица Unix-подобных операционных систем

Критерии сравнения	*Операционные системы*			
	Ubuntu	*FreeBSD*	*OS X*	*Red hat*
Техническая поддержка (регулярные обновления)	+	+	+	+
Поддерживаемое оборудование	Оборудование с x32/x64 архитектурой	Оборудование с x32/x64 архитектурой	Работает только на оборудование разработчика ОС	Оборудование с x32/x64 архитектурой
Поддержка программных пакетов выбранных нами для исследования	+	-	+	+
Стоимость	Бесплатно. Лицензия GNU GPL	Бесплатно. Лицензия BSD	Бесплатно (Текущая версия - Maveric). Лицензия проприетарная	От 1500 рублей. Лицензия проприетарная

Из таблицы 2.3 видно, что наиболее подходящей ОС по выбранным параметрам является ОС Ubuntu Server [48,46]. В этой связи было принято решение разрабатывать далее анализатор дампа, фунционирующих под управлением данной ОС.

2.2.3. Обоснование выбора самостоятельной реализации программного обеспечения для лингвистического анализа дампа трафика

Для обоснования выбора подхода, использованного при разработке программного инструмента средства для парсинга дампа, был проведен сравнительный анализ возможных подходов на основании критериев, приведенных в таблице 2.4.

Таблица 2.4

Сравнительная таблица средств парсинга дампа

Критерии сравнения	*Подходы к реализации лингвистического анализа*			
	Написание парсера самостоятельно	*Импорт дампа вручную*	*Traceplay*	*Использование готового списка параметров*
Открытый исходный код	+	-	-	+
Возможности распознавания	Неограниченные возможности по извлечению параметров протоколов любых уровней.	Только заранее выбранные параметры для определенных протоколов	Ограниченное число параметров	Возможности по извлечению параметров протоколов любых уровней ограничены размером списка параметров
Поддержка ОС Ubuntu	+	+	+	+
Трудозатраты на реализацию	Высокие	Отсутствуют (используется стандартная функции импорта данных MATLAB)	Низкие (реализуется единственная mex-функция MATLAB)	Низкие (реализуется единственная mex-функция MATLAB)
Количество операций, которые необходимо совершить вручную для переноса данных в среду MATLAB	Зависит от реализации	1.Подготовка дампа для импорта (перевод в текстовый формат) 2.Импорт дампа 3.Правка некорректно импортированных данных	Формируется запрос из командной строки MATLAB	Формируется запрос из командной строки MATLAB
Быстродействие	Зависит от реализации	Очень медленно, так как для экспорта в рабочее пространство MATLAB требуется предварительная подготовка данных и корректировка возможных ошибок	Наиболее быстрый способ, так как данные попадают в среду MATLAB напрямую	Наиболее быстрый способ, так как данные попадают в среду MATLAB напрямую

Критерии сравнения	*Подходы к реализации лингвистического анализа*			
	Написание парсера самостоя-тельно	*Импорт дампа вручную*	*Traceplay*	*Использование готового списка параметров*
Стоимость	Бесплатно	Бесплатно	Бесплатно в образова-тельных це-лях. Лицензия проприетар-ная	Бесплатно. Ли-цензия GNU GPL

Из таблицы 2.4 видно, что точки зрения выбранных критериев наиболее оптимальным оказывается вариант реализации парсера в виде самостоятельно реализованной mex-функции MATLAB (mex-функции, использующиеся в MATLAB для вызова подпрограмм, написанных на языках C или Фортран, аналогично функциям, включенныхэ в ядро пакета MATLAB). При этом в качестве готовых списков параметров можно использовать фильтры, реализованные в программных инструментах tcpdump или Wireshar, так как списки параметров фильтров данных программных продуктов покрывают большинство возможных запросов к файлам дампов Инитернет-трафика. Примеры фильтров (DisplayFilters) Wireshark и их описания представлены в таблице 2.5.

Таблица 2.5

Примеры фильтров представления (Display Filters) программы Wireshark

Фильтр для отображения	*Описание*
eth.addr	MAC адрес отправителя или получателя
eth.src	MAC-адрес отправителя
eth.dst	MAC-адрес получателя
tcp.port	TCP порт получателя или отправителя
tcp.dstport	TCP порт получателя
tcp.srcport	TCP порт отправителя
frame.time	Время прохождения пакета через интерфейс
frame.len	Размер пакета

В связи с тем, что информация о списке фильтров (DisplayFilters) Wireshark и технологиях их использования в открытых источниках представлена более подробно было принято решение использовать данный программный инструмент для реализации MATLAB mex-функции matshark.

Структурная схема работы реализованной mex-функции, представлена на рис. 2.1.

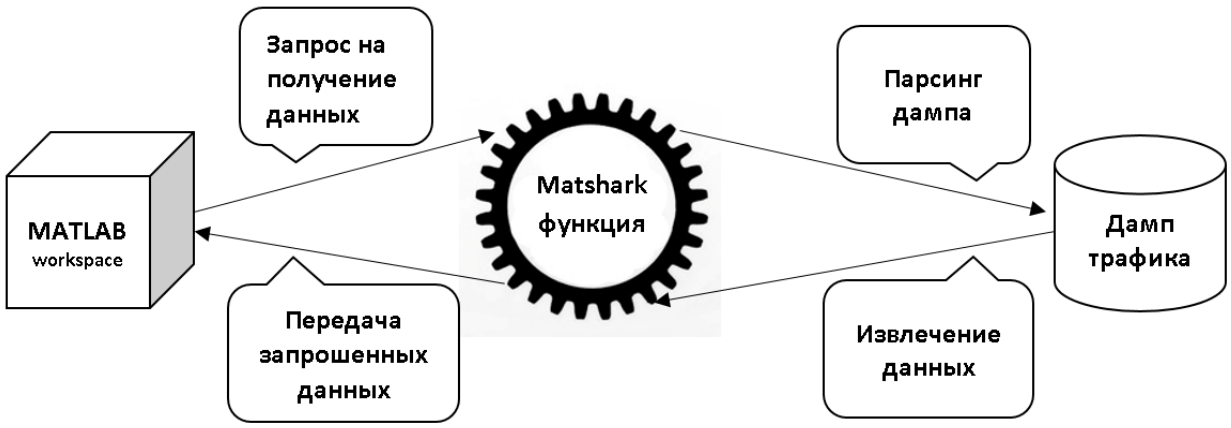


Рис. 2.1. Организация импорта данных из дампа в среду обработки MATLAB

Блок-схема алгоритма, реализованного в данной mex-функции, представлена на рис. 2.2

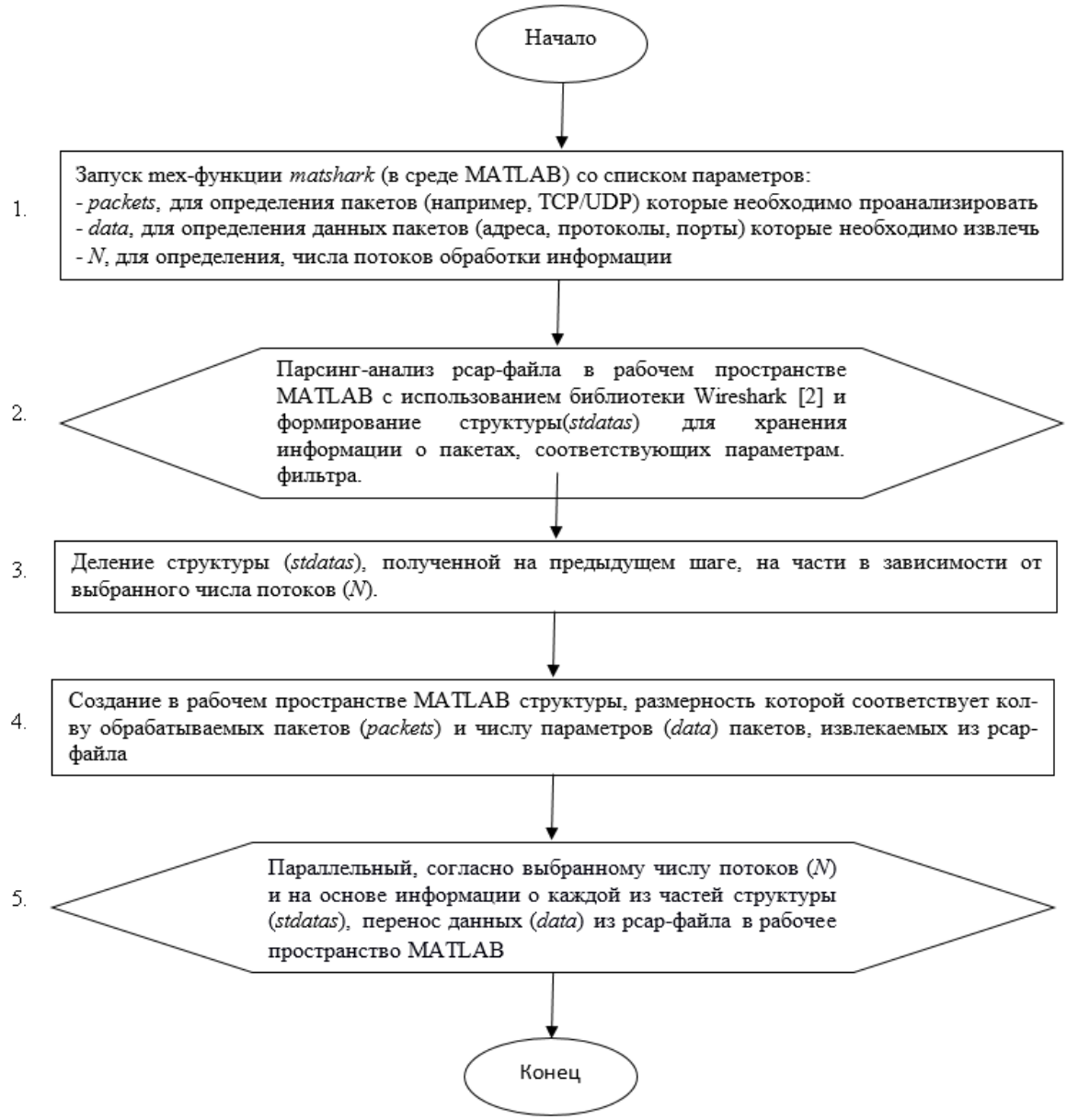


Рис. 2.2. Блок-схема алгоритма, реализованного в mex-функции matshark

Из рисунка 2.2 видно, что данный алгоритм реализуется следующей последовательностью действий.

1. Из среды MATLAB, вызывается функция matshark с указанием списка параметров.

2. Функция matshark при помощи библиотеки Wireshark [2] проводит парсинг-анализ pcap-файла и формирует структуру, для хранения информации о, подходящих под заданные параметры, пакетах.

3. Структура, полученная на предыдущем шаге, делится на части в зависимости от выбранного числа потоков (N).

4. В рабочем пространстве MATLAB создается структура размерность которой соответствует количеству пакетов (packets), которые необходимо обработать, и числу параметров (data) пакетов, которые необходимо извлекать.

5. Основываясь на информации о каждой из частей структуры (stdatas) данные (data) параллельно (согласно выбранному числу потоков N) переносятся из PCAP файла в среду MATLAB.

Данная mex-функция, листинг которой приведен в Приложении А, имеет следующий синтаксис ее вызова:

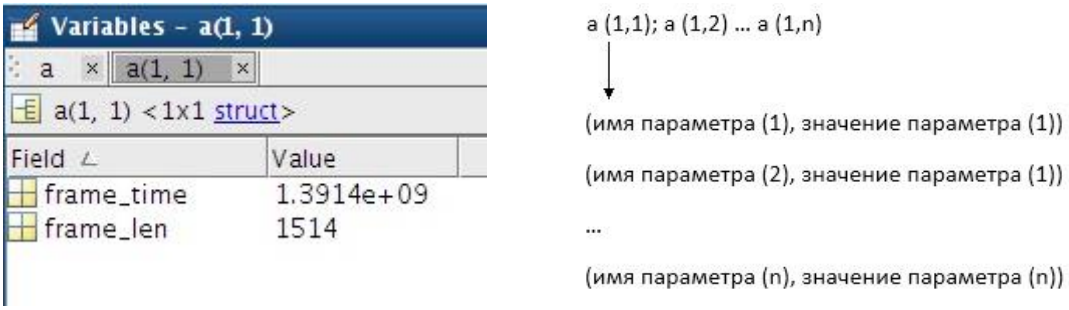
a = matshark('Имя файла', {'Извлекаемый параметр1 в формате Wireshark фильтра', 'Извлекаемый параметр2 в формате Wireshark фильтра',…}).

Отметим, что при вызове мена извлекаемых параметров должны совпадать с форматом соответствующего фильтра программы Wireshark. Далее информация, извлеченная из дампа, передается функцией matshark в виде упорядоченной структуры (массива двумерных ячеек, первое поле, которых содержит название параметра, извлеченного из дампа, вторая ячейка – значение данного параметра) непосредственно в рабочее пространство MATLAB. Пример, иллюстрирующий состояние рабочего пространства MATLAB после выполнения команды:

a = matshark('MAW102072014.pcap', {'frame.time', 'frame.len'})

представлен на рисунке 2.3.

Здесь в качестве объекта исследования был использован дамп трафика магистрального интернет-канала от 07.02.2014 [47] из хранилища MAWI. Размер файла, содержащего 15-минутный снимок дампа трафика магистрального канала, составлял 4,72 Гб.



а) б)

Рис.2.3. Данные из файла дампа, переданные в рабочее пространство MATLAB, функцией matshark: а) состояние рабочего пространства MATLAB после окончания работы функции matshark в MATLAB; б) структура массива двумерных ячеек

Из рисунка 2.3 видно, что каждая ячейка структуры содержит время прибытия соответствующего пакета (переменная frame_time) и его размер (переменная frame_len). Время в файле формата .pcap указано в секундах, прошедших с полуночи по UTC 01.01.1970 до времени поступления пакета.

Наличие данных о времени прибытия пакета и его размере позволяет вычислить, например, объемы информации, переданной за выбранный временной интервал (зависимость «мгновенных» объемов переданной информации от времени). каждую. Пример подобной зависимости представлен на рис. 2.4.

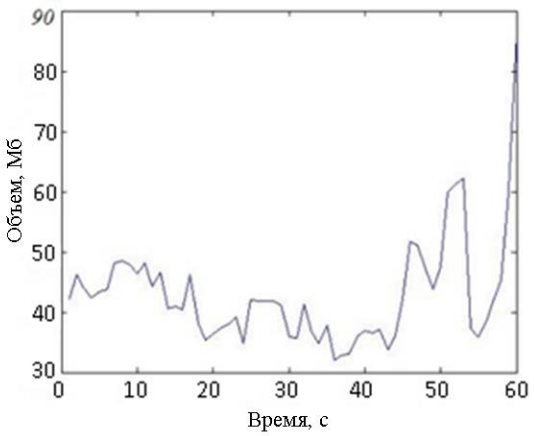


Рис. 2.4. Зависимость объема информации, переданной в течение одной минуты, от времени

Отметим, что анализ дампа следует проводить, начиная со второй секунды и заканчивая предпоследней секундой, так как информация о первой и последней секунде может оказаться неполной, и, следовательно, невозможно достоверно определить объём передаваемых в этот момент данных.

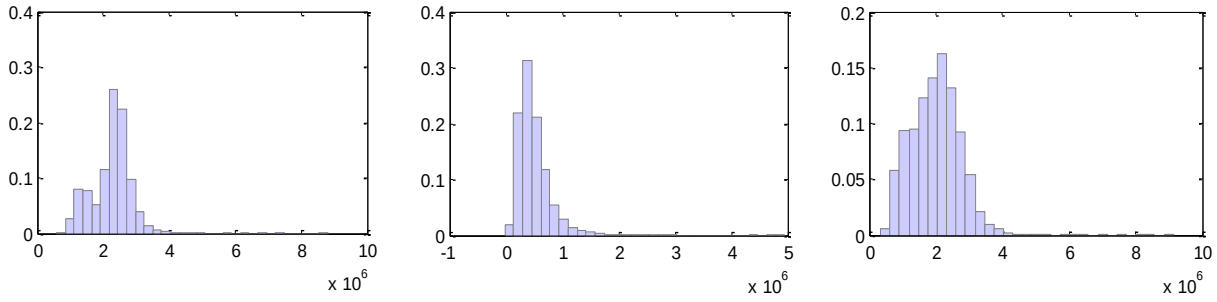
2.2.4. Результаты экспериментальной проверки работоспособности разработанной технологии семантического анализа дампов Интетрнет-трафика

Для проверки работоспособности предложенной технологии работы с дампом Интернет-трафика на основе результатов проведенного анализа существующих

программных инструментов для математической обработки данных, лингвистического анализа, сетевых анализаторов и операционных систем был выбран следующий перечень программ и аппаратных компонентов:

1. Операционная система – Ubuntu Server 12.04.4 LTS.
2. Инструмент математического анализа – MATLAB 2013b.
3. Сетевой анализатор – Wireshark 1.10.05.
4. ЭВМ следующей конфигурации:
   - 4.1. процессор Intel Xeon X5670 2.93GHz (2ядра);
   - 4.2. 16 Гб оперативной памяти;
   - 4.3. жесткий диск объёмом 500 Гб.

В качестве исходной информации использовались файлы с дампом трафика, размещенные на сайте [27]. Из них помощью mex-функции пакета MATLAB matshark, разработанной авторами, выбиралась необходимая для целей дальнейшего анализа информация. Далее в пакете MATLAB рассчитывались выбранные количественные характеристики исследуемых информационных потоков и далее проводился их количественный анализ. Примеры вычисленных количественных характеристик Интернет-трафика из MAWI-архива, зарегистрированного 27.10.14 (файл 201410271400.dump), представлены на рис. 2.5.



а б в

Рис. 2.5. Гистограммы распределений «мгновенных» объёмов данных, переданных 27.10.14 потоками различных размеров за 0,1 с.: а – потоками объемом до 0,3 Мбайт, б – потоками объемом до от 0,3 до 10 Мбайт, в – потоками объемом более 10 Мбайт

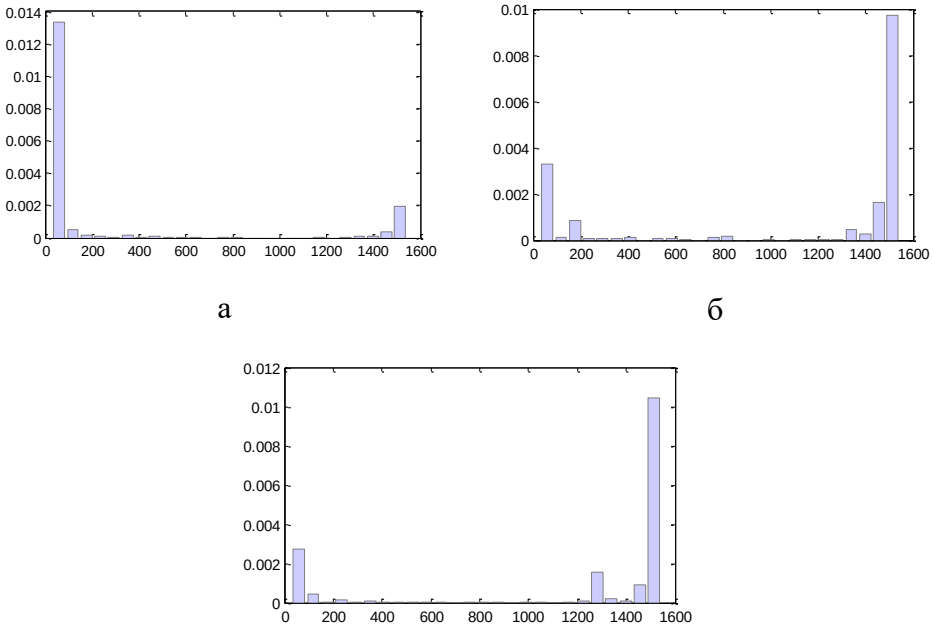


Рис. 2.6. Гистограммы распределений размеров пакетов, переданных потоками различных размеров в течении 15 минут: а – до 0,3 Мбайт, б – от 0,3 до 10 Мбайт, в – более 10 Мбайт (28.10.14)

Также были получены оценки временных затрат, необходимых для извлечения из дампа Интернет-трафика, информации о выбранных параметрах (табл. 2.6).

Таблица 2.6

Время обработки дампа в зависимости от его размера и количества извлекаемых параметров

Размер дампа	*Количество извлекаемых параметров*	*Время обработки, мин*
200 Мб	2	2–5
400 Мб	2	5–10
600 Мб	2	10–15
1 Гб	2	25–30
1 Гб	4	55–65
2 Гб	2	55–65

Анализ данных, представленных в таблице показал, что зависимость времени обработки от размера дампа и количества извлекаемых параметров может быть аппроксимирована функцией вида

$$y = a_1 \cdot x_1 + a_2 \cdot x_2 + a_3$$

где y – время обработки, мин; x_1 – размер дампа, Мб; x_2 – количество извлекаемых параметров приведены в таблице 2.7, результаты анализа качества аппроксимации – в таблице 2.8.

Таблица 2.7

Коэффициенты множественной линейной регрессии зависимости времени обработки от размера дампа и числа извлекаемых параметров

Коэффициенты	*Значения*
a_3	–37,597
a_1	0,033
a_2	17,202

Таблица 2.8

Статистика линейной регрессии

Множественный *R*	0,999205595
R-квадрат	0,998411821
Нормированный *R*-квадрат	0,997353035
Стандартная ошибка	1,396399239
Наблюдения	6

Из таблицы 2.8 видно, что коэффициент корреляции (множественный *R*), оказывается равным 0,999, что свидетельствует о тесной связи рассматриваемых признаков и надежности найденного уравнения регрессии.

2.3. Реализация программного обеспечения для обработки данных, извлекаемых из дампа Интернет-трафика

2.3.1. О выборе групп пользователей, создающих информационные потоки

В работе, следуя [32], мы использовали модель генераторов потоков трафика (поток трафика – совокупность данных, переданная приложением, инициировавшим данный поток, например, запрос клиента к серверу, как в прямом, так и в обратном направлениях), в которой совокупность генераторов трафика разделена на следующие классы:

1. Класс «Слоны», к котором отнесены потоки, передающие данные объемами более 10 Мбайт (P2P, торренты, скачивание больших файлов), время жизни таких потоков велико.

2. Класс «Мулы», к которому отнесены потоки, передающие данные объемами от 0,3 до 10 Мбайт (просмотр видеороликов, прослушивание музыки, скачивание файлов небольших размеров).

3. Класс «Мыши», к которому отнесены потоки, передающие данные объемами менее 0,3 Мбайта (ICQ-сообщения, просмотр WEB-страниц и т.д.).

2.3.2. Разработка алгоритма идентификации потоков и его программной реализации

Анализ дампа Интернет-трафика, цель которого состояла в извлечение его количественных показателей, осуществлялся в два этапа:

Этап 1. Семантический анализ файлов [82](Приложение Е), содержащих дамп Интернет-трафика, обеспечивающий извлечение количественных показателей Интернет-трафика, удовлетворяющих заданным условиям, и их автоматическую передачу в рабочее пространство пакета MATLAB с помощью технологии, описанной в разделе 2.2. (Отметим, что при проведении семантического анализа дампов Интернет-трафика в распоряжении автора имелся компьютер с объемом оперативной памяти (ОЗУ) 16 Гбайт. Как показала практика, ОЗУ данного объема обеспечивало размещение в рабочем пространстве MATLAB и обработку не более 400 Мбайт данных, что соответствует объему информации, извлекаемой из дампа Интернет-трафика, зарегистрированного на временном интервале длительностью около 2 мин. При превышении указанного объема данных программа из-за заполнения ОЗУ начинала работать нестабильно. В этой связи исходный файл дампа Интернет-трафика делился на части, каждая из которых обрабатывалась по отдельности.)

Этап 2. Вычисление параметров потоков, которым принадлежат соответствующие пакеты [80](Приложение Ж).

При разработке алгоритма идентификации потоков было принято во внимание, что:

1. поток состоит из пакетов, передаваемых приложением, относящемуся к одному из выбранных в разделе 2.3.1. классов пользователей, в прямом и обратном направлении;
2. суммарный объем трафика равняется суммарному объему пакетов данного потока, переданных в обоих направлениях.

Для отнесения конкретного пакета к соответствующему потоку и вычислению объема передаваемой в потоке информации, как очевидно, из дампа Интернет-трафика достаточно извлечь следующие параметры:

– время прохождения пакета через узел записи дампа;
– размер пакета (включая передаваемые данные);
– IP-адрес отправителя и получателя пакета;
– порт отправителя и получателя пакета;
– тип протокола.

Действительно, в соответствие с алгоритмом работы [7] сетевого (IP-адреса) и транспортного (порты) уровней модели OSI (open systems interconnection), пакеты будут принадлежать одному потоку, если они идентичны в части адреса отправителя и получателя, а также портов отправителя и получателя. При этом необходимо учитывать, что поток содержит пакеты двух типов: запросы к серверу и ответы от него (рис.2.7).

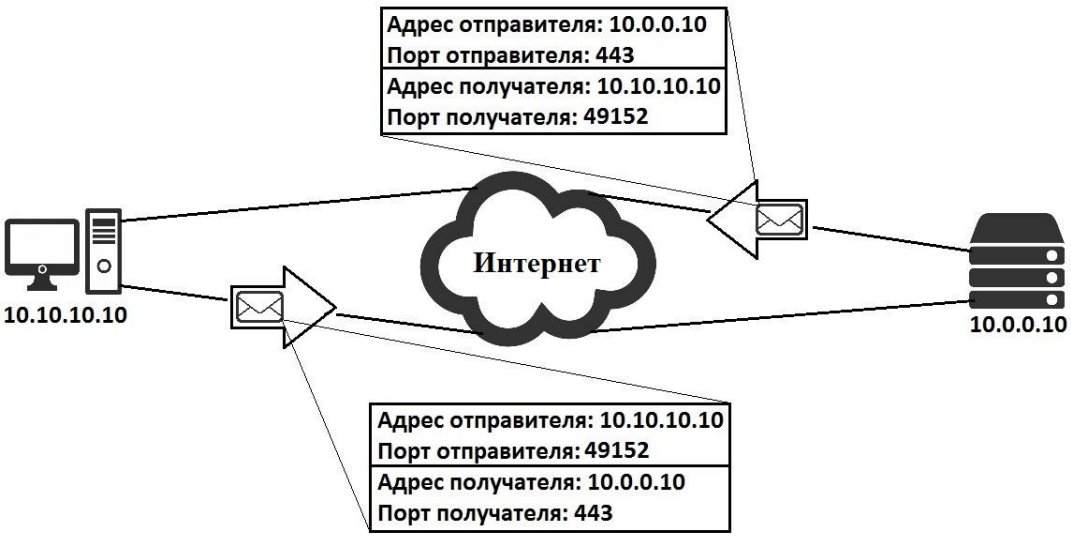


Рис. 2.7. Схема передачи пакетов одного потока

Пример отправки от клиента (10.10.10.10) к серверу (10.0.0.10) и ответа от сервера клиенту пакетов одного потока приведен на рис. 2.7. Здесь все пакеты, являющиеся запросами клиента (порт отправителя 443) к серверу (порт получателя 49152), а также ответы сервера (порт отправителя 49152) клиенту (порт получателя 443) принадлежат одному и тому же потоку.

Так как один и тот же клиент может создавать несколько параллельных сессий (потоков), то порт получателя клиента может быть одинаковым для нескольких сессий. При этом порт отправителя является уникальным. Его значения назначаются динамически. Они находятся в диапазоне от 49151 до 65535, что позволяет достоверно определять к какому потоку принадлежит пакет. При этом порты получателя, как правило, являются общеизвестными (80-http, 53-dns) — это порты в диапазоне от 0 до 1023 или порты, зарегистрированные от 1024 до 49151. Назначение портов контролируется администрацией адресного пространства Интернет IANA (Internet Assigned Numbers Authority) [19]. Объем данных, передаваемых потоком, можно вычислить, сложив объемы данных, передаваемых каждым пакетом этого потока (рис. 2.8 действие 9).

Программная реализация алгоритма классификации информационных потоков в MATLAB представлена в Приложении Д. Блок-схема последовательности действий, реализующих методику обработки дампа трафика, включающую в себя сематический анализ дампов Интернет-трафика, и последующую идентификацию потоков, представлена на рис. 2.8.

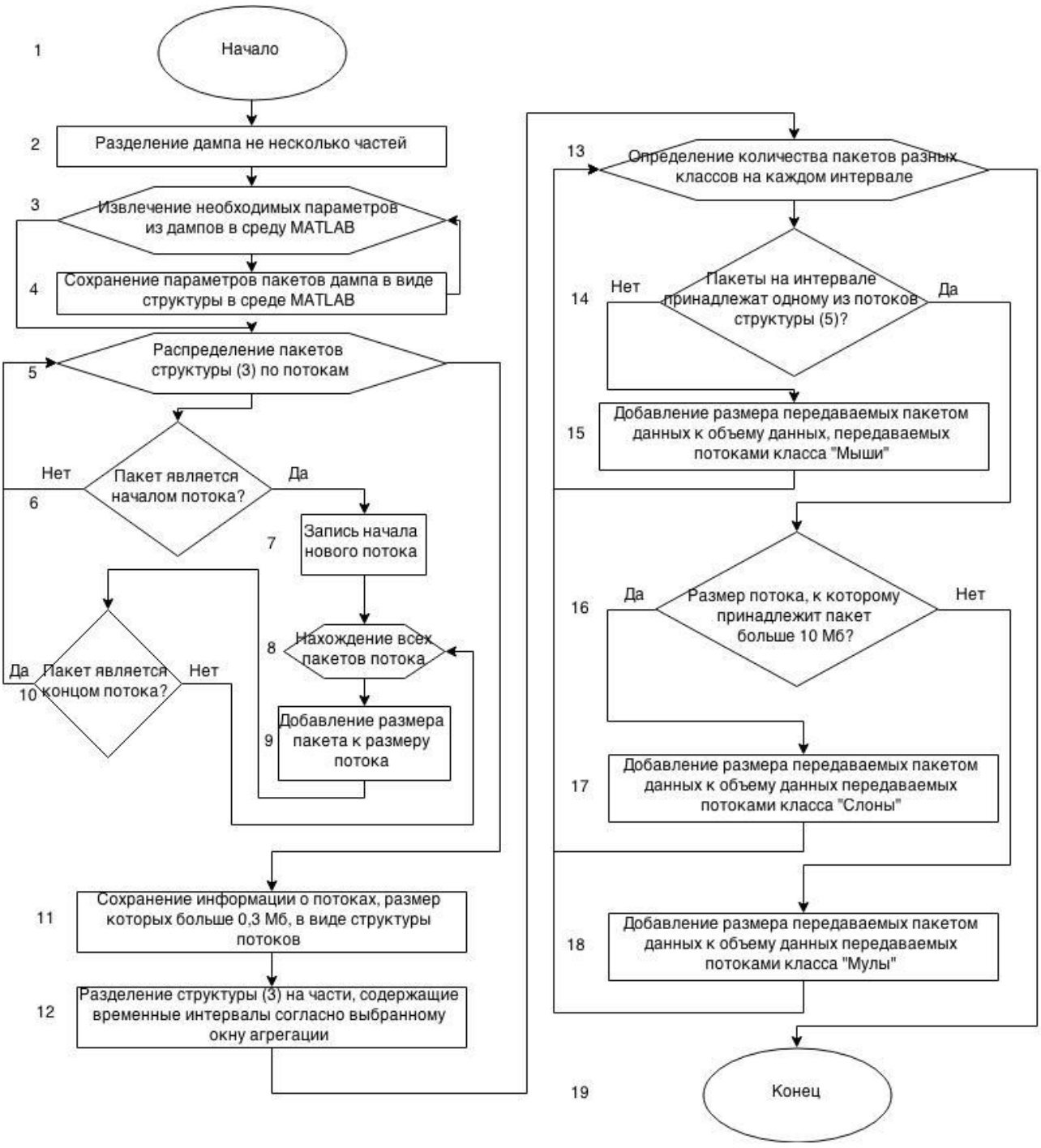


Рис. 2.8. Блок-схема методики обработки дампа Интернет-трафика

2.3.3. Оптимизация алгоритма идентификации потоков Интернет-трафика

Анализ дампов Интернет-трафика показал, что общее число потоков, создаваемых «Мулами» и «Слонами» оказывается существенно меньшим, чем число потоков (см. главу 4), создаваемых «Мышами». В этой связи, понятно, что можно уменьшить время анализа дампа Интернет-трафика, если сначала разделить информацию, имеющуюся в дампе Интернет-трафика, выделив из нее информацию, относящуюся к потокам, создаваемым «Мышами», размеры потоков которых не превышают 0,3 Мбайта. Пакеты, отправляемые и получаемые «Мышами», можно определить по протоколу (например, ICMP – протокол межсетевых управляющих сообщений) или порту (например, DNS – система доменных имен).

Тогда информация, оставшаяся в дампе Интернет-трафика, будет относиться только к потокам, созданных «Мулами» и «Слонами». Описанный выше подход, как показал опыт его использования на практике, позволил сократить время обработки дампа Интернет-трафика на почти на треть.

Также был предложен еще один подход, позволивший оптимизировать алгоритм идентификации потоков, основанный на том, что большинство потоков состоит не из двух, но, как минимум, из нескольких пакетов. В этой связи целесообразно определять начало и конец анализируемого потока, но не искать по всему файлу дампа Интернет-трафика каждый из принадлежащих данному потоку пакетов, в котором может содержаться информация, вообще говоря, о миллионах пакетов. Для определения времени начала и времени конца потока оказывается достаточным проверить не встречаются ли пакеты, относящиеся к текущему потоку, на временном интервале равном 4 с (максимальное время ожидания команды ping windows) до и после данного пакета. Если пакеты, соответствующие данному потоку, не встречаются в течении четырех секунд до регистрации текущего пакета, данный пакет считается началом потока, если в течении четырех секунд после его регистрации – концом потока. Во всех остальных случаях текущий пакет является частью потока. Интервал, определяющий начало и конец потока, равный 4 секундам, позволяет свести к минимуму разрывы одного потока на несколько, так как взят с заведомым запасом (среднее время ожидания (Round-trip time) в дампе не превышает 50 мс). Данный подход позволил, как показала практика, в десятки раз сократить время обработки дампа Интернет-трафика в сравнении с полным прогоном. Корректное определение потоков, в том случае, когда поток продолжается после перерыва более 4000 мс (время ожидания некоторых приложений может превышать этот интервал), можно обеспечить сравнением (алгоритм аналогичный сравнению пакетов) идентифицированных потоков друг с другом. При выявлении идентичных потоков можно они считаются одним потоком, имевшим длительную задержку.

2.3.4. Аппаратные средства технологии работы с дампом Интернет-трафика

Опыт обработки ежедневных пятнадцатиминутных дампов трафика, передаваемого в магистральном канале между США и Японией (средний объем пятнадцатиминутного дампа архива MAWI составляет около 10 ГБ), находящиеся в общедоступном хранилище – MAWI (Measurement and Analysis on the WIDE Internet) [27] показал, что их анализ требует значительных временных затрат. Например, в том случае, когда на одноядерном процессоре с тактовой частотой 3 ГГц исходный pcap-файл делился на файлы размером по 100 МБ, проводился независимый семантический анализ каждого из файлов меньшего объема и далее выбранные в ходе парсера количественные показатели передавались в рабочее пространство MATLAB, среднее время обработки пяти дампов Интернет-трафика размером

100 МБ составило около 9 часов. Таким образом, примерное время обработки всего пятнадцатиминутного дампа на данном компьютере составит примерно 960 часов.

В этой связи была высказана гипотеза о целесообразности переноса описанной выше технологии обработки дампа Интернет-трафика на суперкомпьютер. Действительно, априори, можно ожидать, что использование многоядерных процессоров, число ядер которых которых выбирается в зависимости от размера обрабатываемых данных, и кластеров в сочетании с технологиями параллельных вычислений[1], в том числе и суперкомпьютеров позволит сократить время обработки в десятки и сотни раз.

Для подтверждения данной гипотезы была создана программная реализация алгоритма идентификации потоков, основанная на использовании технологии параллельных вычислений.

2.4. Реализация механизмов параллельных вычислений для алгоритма идентификации потоков Интернет-трафика

Так как обработка дампа и последующие вычислительные операции производятся из среды MATLAB и его средствами для реализации возможности многопоточной обработки дампа были изучены с параллельных вычислений, реализованные в пакете MATLAB.

2.4.1. Анализ программных инструментов MATLAB, поддерживающих технологии параллельных вычислений

Пакет MATLAB предоставляет пользователю следующие средства программные инструменты для реализации параллельных вычислений:

1. Функции, реализующие параллельные алгоритмы для решения задач линейной алгебры, вычисления преобразования Фурье, операции с матрицами и др. для выполнения которых одновременно используются несколько центральных процессоров [26]. Данные функции реализуют режим параллельных вычислений автоматически без участия пользователя, используя для этого локальные ресурсы систем с общей памятью.

2. Пакет расширения (Toolbox) MATLAB Parallel Computing Toolbox, обеспечивающий пользователя средствами настройки и контроля процесса распараллеливания вычислений. Данный инструмент предназначен для написания параллельных алгоритмов (циклов, распределенных массивов) и организации распределенных вычислений. При совместном использовании с Toolbox Distributed Computing Server он позволяет использовать не только локальные, но и распределенные

[1] Под технологией параллельных вычислений мы понимаем совокупность программных и аппаратных средств, позволяющих организовать набор взаимодействующих вычислительных процессов, работающих одновременно.

вычислительные ресурсы, работает на серверах с общей памятью и графических ускорителях.

3. Пакет Distributed Computing Server, являющийся наиболее мощным инструментом реализации параллельных вычислений. Данный пакет представляет собой серверную частью системы организации распределенных вычислений, которая функционирует как на выделенном кластере MATLAB, так и на разделяемом кластере под управлением системы запуска задач. Он используется для запуска задач MATLAB на кластере с распределенной памятью. При этом на узлах кластера запускается один или несколько процессов MATLAB, которые обмениваются данными между собой с помощью библиотеки функций MPI (Message passing interface).

Возможны различные способы организации одновременного взаимодействия между разными уровнями параллельных вычислений в пакете MATLAB. Например, в кластере можно одновременно использовать несколько вычислительных процессов MATLAB, использующих многопоточные функции. При использовании встроенной многопоточности MATLAB или средств Toolbox Parallel Computing Toolbox возможно использование только локальных ресурсов компьютера. При этом существует программное ограничение на максимальное число процессоров/ядер, используемых в параллельных вычислениях – не более 12.

Отметим, что увеличение числа процессоров/ядер, как правило, увеличивает скорость выполнения задачи в арифметической прогрессии. При использовании одной ЭВМ с 12 ядрами, тактовая частота которых равна 3.0 ГГц, время обработки дампа составит около 80 часов. При необходимости обработки большого количества дампов сетевого трафика, понятно, что данный результат также следует признать неудовлетворительным. В тоже время можно обойти ограничение на число центральных процессоров можно, используя вычислительный кластер, на котором установлен пакет Distributed Computing Server

Однако использование кластера с Distributed Computing Server оказалась невозможной из-за высокой стоимости программного и аппаратного обеспечения и текущей политикой его продаж (только для юридических лиц). В связи с этим необходимо были рассмотрены альтернативные проекты, позволяющие получить доступ к аналогичным программным и аппаратным ресурсам и требующие при этом существенно меньших финансовых затрат.

2.4.2. Обзор кластеров MATLAB

1. Облако Mathworks Cloud, разработанное компанией Mathworks [9]. Данный сервис предназначен, в первую очередь, для мобильных устройств и использовать его можно только из специализированной версии MATLAB Mobile. Данный сервис обладает

существенными ограничениями: отсутствует возможность использования редактора MATLAB. Однако объем передаваемых при этом данных не может превышать 500 Мб, поэтому данный инструмент не подходит для обработки дампов.

2. Вычислительное облако Amazon EC2, являющееся совместным проектом Mathworks и Amazon [35]. Для его использования на сервере EC2 должен быть запущен кластер Distributed Computing Server, а на клиентском компьютере может использоваться Parallel Computing Toolbox. Однако в данный момент проект находится на начальной стадии развития и в России недоступен.

3. Облако «Red Cloud» на основе использования Distributed Computing Server, доступ к которому предоставляет Корнельский университет [34]. К недостаткам данного сервиса следует отнести обязательное наличие клиента MATLAB с Parallel Computing Toolbox на компьютере клиента для выполнения задач на кластере.

4. Российские суперкомпьютерные вычислительные системы с терафлопсной производительностью, например: «Ломоносов» МГУ им. Ломоносова, «МВС-10П» межведомственного Суперкомпьютерного Центра Российской академии наук, «Лобачевский» НГУ им. Лобачевского и др.[96]. Однако только единицы из них имеют программное обеспечение MATLAB, предоставляемое в аренду вместе с вычислительными ресурсами, в том числе, суперкомпьютеры «Торнадо ЮУрГУ» Южно-Уральского государственного университета и «Уран» Института математики и механики Уральского отделения Российской академии наук.

Оба этих супервычислителя входят в первую десятку [96] рейтинга суперкомпьютеров России, обладают достаточными ресурсами для обеспечения удовлетворительного времени (не более одного рабочего дня) обработки дампов, предоставляют комфортные условия взаимодействия, не требующие наличия программного обеспечения MATLAB на клиентской машине. Поскольку серверная лицензия «Торнадо ЮУрГУ» позволяет использовать не более 32 [71] ядер, было принято решение использовать вычислитель «Уран», лицензия которого позволяет использовать сотни ядер [72].

2.4.3. Облачный кластер MATLAB суперкомпьютера «Уран»

Суперкомпьютер «Уран» ИММ УрО РАН собран на базе Blade серверов фирмы Hewlett-Packard. Он состоит из 204-х вычислительных узлов, установленных в модулях с высокой плотностью упаковки. Вычислительные узлы оснащены Intel Xeon, работающими на частотах 2.2-3 ГГц, 16–200 Гб оперативной памяти и графическими ускорителями NVIDIA Tesla и предоставляет пользователям для работы следующие вычислительные средства:

- Количество ядер: 1864 CPU Xeon (3.0ГГц) и 352 GPU Tesla.

- Оперативная память: 6976 ГБ.
- Объем памяти системы хранения: 10 ТБ.
- Коммуникативная среда: Infiniband, GiEthernet.

Это обеспечивает пиковую производительность 216.56 Tflop/s и 105.36 Tflop/s на тесте Linpack.

Базовое программное обеспечение вычислительного кластера включает в себя:

- Операционная система Linux;
- Система запуска задач Slurm;
- Языки программирования C, C++, Fortran;
- Компиляторы Intel, GNU, PGI;
- Библиотека Math Kernel Library (MKL) Intel;
- Реализации MPI: OpenMPI и MVAPICH2;
- Пакеты MATLAB, ANSYS CFX Academic Research.

Способы взаимодействия пользователя с СуперЭВМ представлены в таблице 2.9.

Таблица 2.9

Способы взаимодействия с вычислительным кластером

Тип доступа	*Программное обеспечение*
Командная строка	PuTTY
Графический интерфейс	WinSCP
Обмен файлами	Nomachine NX MobaXterm

2.4.4. Особенности подготовки кластера для запуска программ

Для подключения к вычислительному кластеру для каждого пользователю создавалась учетная запись (логин и пароль). В зависимости от заявки пользователя задавалось время для выполнения задач, режим выделения ресурсов (динамический или монопольный), а также выделялись вычислительные ресурсы: количество доступных процессорных ядер, размер оперативной памяти и дискового пространства.

В процессе запуска программы на кластере были обнаружены проблемы, связанные с использованием дополнительных компонентов: mex-функции [83] и библиотеки libwireshark.so. Это потребовало выполнение дополнительных настроек вычислительного кластера, позволивших MATLAB «увидеть» необходимые компоненты: в домашнем каталоге была создана директория (home/username/lib) для размещения дополнительных библиотек и функций. В частности, для указания каталога, в котором находятся библиотеки, потребовалось скопировать стартовый shell-скрипт MATLAB.matlab7rc.sh в домашнюю директорию и добавить в него переменную LDPATH_PREFIX="home/username/lib". Это

обеспечило автоматическое добавление каталога с необходимыми библиотеками при запуске MATLAB для каждого нового пользователя.

2.4.5. Запуск программы обработки дампа на кластере

Оказалось, что наиболее удобный способ запуска программ MATLAB на кластере – использование графического интерфейса, предоставляемого программой MobaXterm, который эмулирует программу MATLAB, как будто она запущена на клиентском компьютере.

Для запуска программы matshark в многопоточном режиме были использованы специальные конструкции языка parfor и spmd [13], что предполагает открытие пула MATLAB, равного необходимому числу процессоров (Matlab workers или labs) на кластере. При этом пул можно задавать как в явном, так и не явном виде.

Для разгрузки клиентской ЭВМ и обеспечения возможности интерактивной работы на ней использовалась команда *batch*, позволяющая запускать программы асинхронно, в пакетном режиме, разгружая клиент MATLAB. Команда запуска программы вида program_name.m выглядят следующим образом:

- job = batch('program_name') (запуск программы в однопроцессорном варианте).
- job = batch('program_name','matlabpool',100) (запуск программы в режиме использования 101-го процессора: 1 – для программы program_name.m и 100 – для пула).

Для выполнения программ создается объект Job (работа). Максимальное время и ресурсы для выполнения программы, указанной в объекте Job, определяет планировщик на кластере «Уран» (тип планировщика generic). По завершении работы результаты сохраняются в домашней папке пользователя с указанием номера работы.

Результаты проведенной проверки технологии анализа дампа [83] показали, что среднее время обработки одного дампа размером около 10ГБ составляло 7-8 часов, что примерно в 120–140 раз быстрее обработки дампа на одном процессоре. В течение двух суток, выделенных в ИММ УрО РАН для использования суперкомпьютера (120 процессорных ядер, 2Гб оперативной памяти на каждое ядро), было обработано 7 15-ти минутных дампов интернет-трафика, передаваемого в магистральном интернет-канале. Это подтверждает работоспособность и эффективность использованной технологии обработки данных дампов сетевого трафика, которая позволяет значительно ускорить процесс проведения экспериментов по изучению свойств трафика и дает возможность проведения детальных исследований особенностей информационных потоков в магистральных интернет-каналах.

2.5. Выводы по главе

1. Разработана методика работы с трафиком вычислительных сетей, позволяющая получать достоверные результаты при работе с различными видами трафика в различных

условиях с оптимальным использованием вычислительных ресурсов.

2. Создан программно-аппаратный комплекс, позволяющий в автоматизированном режиме получать необходимые для исследования данные. Программная реализация была адаптированная для использования суперкомпьютера «Уран» ИММ УрО РАН.

3. Работоспособность разработанной методики анализа трафика и программно-аппаратного комплекса подтверждена результатами анализа дампов Интернет-трафика, зарегистрированных в магистральном Интернет-канала, соединяющим США и Японию.

ГЛАВА 3. ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ ИНФОРМАЦИОННЫХ ПОТОКОВ В МАГИСТРАЛЬНОМ ИНТЕРНЕТ-КАНАЛЕ

Задача анализа дампа Интернет-трафика, как обосновано в Главе 2, декомпозируется на две последовательно выполняемые задачи:

1) задачу синтаксического анализа (парсинг) текстовых файлов, содержащих дампа Интернет-трафика, и извлечения из них количественной информации, характеризующей информационные потоки на изучаемом временном интервале;

2) анализ в пакете MATLAB количественной информации, извлеченной из дампа Интернет-трафика.

В настоящем разделе обсуждаются результаты анализа дампов Интернет-трафика, зарегистрированных в течение 7 дней с 27.10.2014 по 02.11.2014, каждый из которых представляет из себя 15-минутный дамп (см. раздел 2.1.). Перечень исследованных в работе файлов дампов Интернет-трафика представлен в таблице 3.1.

Таблица 3.1

Перечень исследованных в работе дампов

Дата создания файла	*Имя файла*	*Размер файла*
27.10.14	201410271400.dump	9249.41MB
28.10.14	201410281400.dump	8983.29MB
29.10.14	201410291400.dump	8435.98MB
30.10.14	201410301400.dump	8892.00MB
31.10.14	201410311400.dump	8407.92MB
01.11.14	201411011400.dump	7153.41MB
02.11.14	201411021400.dump	6542.95MB

3.1. Обоснование выбора количественных характеристики изучаемого магистрального Интернет-канала

Основной задачей КС является предоставление пользователю услуг по обработке, хранению и передаче данных. Основные характеристики КС, прямо или косвенно влияющие на качество обслуживания ее пользователей, а также экспертные оценки степени влияния данных характеристик на качество обслуживания представлены в таблице 3.2.

Таблица 3.2

Характеристики КС и степень их влияния на качество операционных возможностей сети

№	*Характеристика КС*	*Описание*	*Степень влияния на качество функционирования сети*
1	Производительность сети	Суммарная производительность всех вычислительных систем, поддерживающих инфраструктуру сети, характеризующая вычислительную мощность всей сети.	Высокая

№	*Характеристика КС*	*Описание*	*Степень влияния на качество функционирования сети*
2	Пропускная способность	Объем данных передаваемых сетью в единицу времени.	Высокая
3	Время реакции	Интервал времени между возникновением запроса клиента к какой-либо услуге сети и моментом получения ответа на данный запрос.	Высокая
4	Надежность	Набор факторов характеризующий следующие характеристики сети: доступность, достоверность, защищенность и отказоустойчивость.	Средняя
5	Прозрачность	Степень простоты работы пользователя в сети	Средняя
6	Управляемость	Возможность контроля состояния, разрешения возникающих проблем и анализа качества функционирования сети	Низкая
7	Расширяемость	Степень легкости замены или добавления/удаления отдельных элементов сети.	Отсутствует
8	Масштабируемость	Возможность расширения сети в широких пределах без заметного ухудшения качества функционирования сети.	Отсутствует
9	Совместимость/ интегрируемость	Возможность включать в сеть различное программное, техническое обеспечение.	Отсутствует
10	Стоимость обработки данных	Целесообразность построения и использования сети.	Отсутствует

Из таблицы 3.2 видно, что основными характеристиками, определяющими качество обслуживания пользователей КС, оказываются: производительность, пропускная способность и время реакции КС. Данные характеристики напрямую связаны с объемами передаваемой в КС информации, носителями которой являются пакеты передачи данных. В этой связи были изучены особенности информационных потоков в магистральном Интернет-канале в разрезе количества пакетов, переданных через канал, их размеров и объемов переданных данных.

3.2. Перенос данных из дампа Интернет-трафика в рабочее пространство MATLAB

На первом этап исследования осуществлялся состоял перенос выбранного дампа в рабочее пространство пакета MATLAB. Схема, иллюстрирующая данный процесс, представлена на рис. 3.1.

Рис. 3.1. Схема процесс переноса данных из pcap-файла в рабочее пространство пакета MATLAB

Из рис. 3.1 видно, что исходный pcap-файл в зависимости от размера оперативной памяти ЭВМ с помощью утилиты tcpdump, обеспечивающую также перехват и анализ сетевого трафика, разделялся на *n* файлов. При этом для устойчивой работы используемого вычислительного устройства размер каждого из этих файлов, как показала практика, не должен был превышать 1/30 объема оперативной памяти. Для реализации описанного деления pcap-файла на части осуществлялся вызов утилиты tcpdump с указанием следующих входных параметров данной утилиты:

tcpdump -r old_file -w new_files -C 400,

здесь tcpdump – вызов команды tcpdump;

-r – опция, предписывающая чтение исходного файла;

old_file – имя исходного файла;

-w – опция, предписывающая запись в новый файл;

new_files – имя нового файла (в зависимости от количества файлов к исходному имени добавляется префикс с его номером);

-C 400 – опция, задающая размер новых файлов в мегабайтах;

Отметим, что файл, содержащий последнюю часть дампа, может иметь разме, меньшее, чем задано опцией -C 400. Например, если выполнить деление дампа размером в 1000 Мбайт на части по 400 Мбайт, то в результате выполнения команды, представленной выше, будут получены 2 файла объема 400 Мбайт и один файл объемом 200 Мбайт.

Перенос данных в рабочее пространство MATLAB для последующего анализа осуществлялся в соответствие с методиками, описанными в разделах 2.2.3, 2.3.2 и 2.4.2, с помощью разработанной нами функции matshark. При этом использовался следующий синтаксис вызова:

a = matshark ('N', 'filename.sequence', {'frame.time', 'frame.len', 'ip.src', 'ip.dst', 'tcp.srcport', 'tcp.dstport', 'udp.srcport', 'udp.dstport', 'ip.proto'})

здесь N – число потоков обработки дамп файла;

filename.sequence – имя части дамп файла с указанием префикса;

frame.time – время прохождения пакета через узел записи дампа;

frame.len – размер пакета (включая передаваемые данные);

ip.src, ip.dst – IP-адрес отправителя и получателя пакета соответственно;

tcp.srcport/tcp.dstport и udp.srcport/udp.dstport. – порты отправителя и получателя пакета;

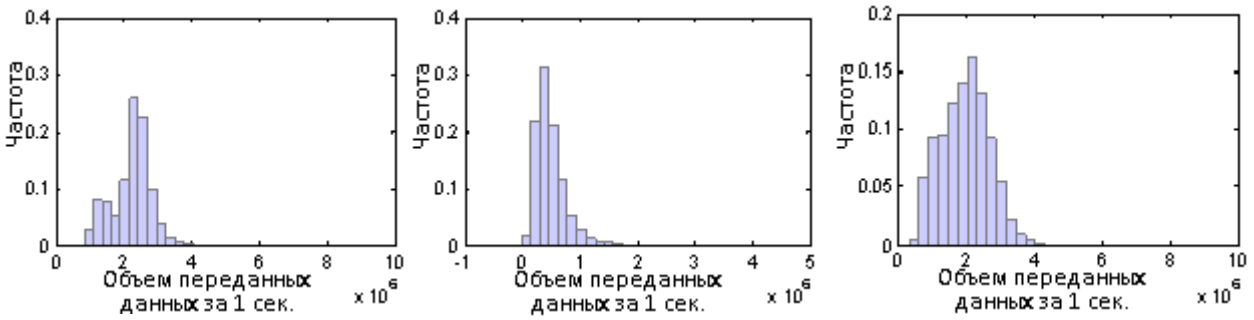
ip.proto – тип протокола.

При переносе данных из pcap-файлов в рабочее пространство MATLAB данные структурируются по выбранным классам пользователей («Слоны», «Мулы» и «Мыши»). Это позволяет далее исследовать особенности информационных потоков, создаваемых данными классами пользователей.

3.3. Выбор методов аппроксимации распределений информационных потоков магистрального канала

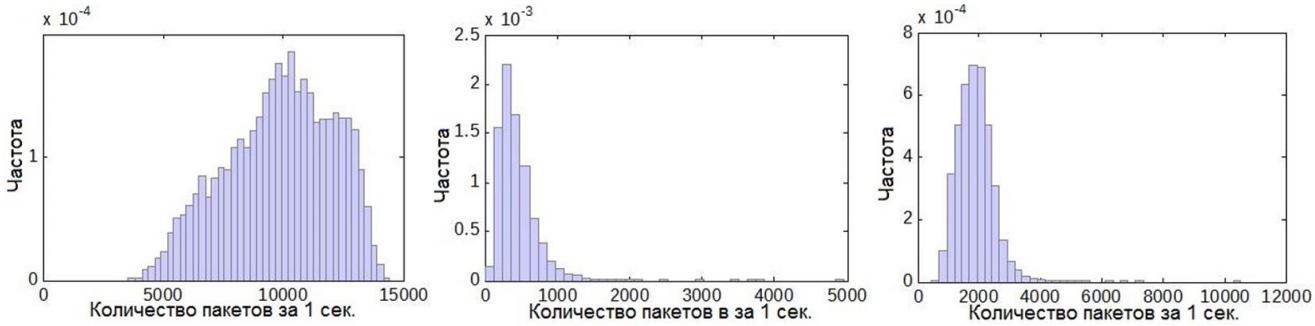
В ходе исследований были изучены дампы Интернет-трафика, зарегистрированные в одно и тоже время на временных интервалах длительностью 15 минут в течение 7 дней с

27.10.2014 по 02.11.2014 (см. раздел 2.1.). Для каждого из выбранных классов потоков («Мыши», «Мулы», «Слоны») были вычислены значения зависимостей объемов данных, переданных в течение выбранного временного интервала (окна агрегации), от времени – случайные последовательности (СП) $V_i^{\langle\tau\rangle}$, τ – размер окна агрегации, $i=1,\mathrm{int}\left(15\cdot 60/\tau\right)$ – порядковый номер окна агрегации, а также агрегированные значения зависимостей числа пакетов, переданных данным классом пользователей, от времени – СП $N_i^{\langle\tau\rangle}$. Типичные гистограммы СП $V_i^{\langle 0.1\rangle}$, $N_i^{\langle 0.1\rangle}$ представлены на рис. 3.2 и 3.3.



а б в

Рис. 3.2. День №1 (27.10.14, файл 201410271400.dump). Распределение СП $V_i^{\langle 0.1\rangle}$: а – «Мыши», б – «Мулы», в – «Слоны»



а б в

Рис. 3.3. День № 4 (30.10.14, файл 201410301400.dump). Распределение СП $N_i^{\{0.1\}}$: а – «Мыши», б – «Мулы», в – «Слоны»

Из рис. 3.2 видно, что СП $V_i^{\langle 0.1\rangle}$ относятся к классу случайных последовательностей с ограниченной областью рассеяния: у «Мышей» – $[0.8;8.0]\cdot 10^6$ байт, у «Мулов» – $[0,3.0]\cdot 10^6$ байт, у «Слонов» – $[0;8.0]\cdot 10^6$ байт. Из рис. 3.3 видно, что анализируемые СП $N_i^{\langle 1.0\rangle}$ относятся к классу СП с ограниченной областью рассеяния: у «Мышей» – [4800;14900]; у «Мулов» – [0;4000]; у «Слонов» – [800;10000].

В этой связи для аппроксимации плотностей распределений изучаемых случайных последовательностей был использован метод мнимых источников, [88], в соответствие с которым плотность распределения вычисляется по формуле:

$$f_{LAD}\left(x;\mu,\sigma,l\right)=A\left[\varphi\left(x;\mu,\sigma,l\right)+\sum_{n=0}^{\infty}\varphi_{2n+1}^{\pm}\left(x;\mu,\sigma,l\right)+\sum_{n=1}^{\infty}\varphi_{2n}^{\pm}\left(x;\mu,\sigma,l\right)\right], \quad (3.1)$$

где A – нормировочный коэффициент, определяемый из условия

$$\int_{a}^{b} f_{LAD}\left(\xi;\mu,\sigma,l\right)d\xi=1, \quad (3.2)$$

$$\varphi\left(x;\mu,\sigma,l\right)=\exp\left[-\left(x-\mu\right)^{2}/2\sigma^{2}\right],$$

$$\varphi_{2n+1}^{\pm}\left(x;\mu,\sigma,l\right)=\exp\left[-\left(x-x_{2n+1}^{\pm}\right)^{2}/2\sigma^{2}\right],$$

$$\varphi_{2n}^{\pm}\left(x;\mu,\sigma,l\right)=\exp\left[-\left(x-x_{2n}^{\pm}\right)^{2}/2\sigma^{2}\right],$$

$x_{2n}^{\pm}=\pm 4nl+\mu$, $x_{2n+1}^{\pm}=\pm\left(4n+2\right)l-\mu$ – координаты центров фиктивных источников;

l – размер области рассеяния,

μ – математическое ожидание случайной величины при отсутствии области ограничения,

σ – СКО случайной величины при отсутствии области ограничения.

Здесь значения параметров μ и σ являются в зависимости от выбранной меры близости эмпирической и расчетной ПР решением той или иной системы нелинейных уравнений, которое может быть найдено с помощью известных численных методов решения систем нелинейных уравнений и генетических алгоритмов [87] и метода аппроксимации Розенблатта-Парзена [31,36]. При практическом использовании методов мнимых источников, описанных в Приложении Б, и аппроксимации Розенблатта-Парзена, описанной в Приложении В, была использована свободно распространяемая программная библиотека, представленная в Приложении Г, на языке пакета MATLAB ES&RP, разработанная С.В. Поршневым и А.С. Копосовым [91]. В данной библиотеке для решения системы нелинейных уравнений, возникающей в методе мнимых источников, используются генетические алгоритмы. Подробное описание методики использования данной библиотеки в задаче аппроксимации плотностей распределений случайных последовательностей приведено в [66].

3.4. Анализ результатов исследования дампа трафика в разрезе объемов данных, передаваемых потоками

Рассмотрим результаты вычисления параметров распределений СП $V_i^{\langle 0.1\rangle}$, $V_i^{\langle 1.0\rangle}$, аппроксимированных с помощью метода мнимых источников, представленные в табл. 3.3–3.5.

Таблица 3.3

Математическое ожидание ПР СП $V_i^{\langle 0.1\rangle}$, $V_i^{\langle 1.0\rangle}$ зозданных «Мышами», «Мулами» и «Слонами» в течение каждого из 7 дней

τ	$\bar{\mu}$	$\frac{\lvert\mu_i-\bar{\mu}\rvert}{\bar{\mu}}\cdot 100\%$							Класс пользователя
		i=1	i=2	i=3	i=4	i=5	i=6	i=7	
0,1	2444194,95	2,40	17,25	4,57	4,96	11,17	8,02	9,99	«Мыши»
1	24441716,74	2,49	17,23	4,46	4,99	11,31	8,30	9,55	
0,1	706542,71	51,48	77,74	21,53	19,94	65,29	48,16	57,25	«Мулы»
1	7434381,93	45,97	68,95	17,26	15,37	35,08	57,25	55,21	
0,1	3449532,25	43,09	8,01	10,35	23,87	13,80	42,10	36,32	«Слоны»
1	34470949,62	43,41	8,07	10,15	23,11	13,95	42,03	36,36	

Таблица 3.4

Среднеквадратичное отклонение ПР СП $V_i^{\langle 0.1\rangle}$, $V_i^{\langle 1.0\rangle}$, созданных «Мышами», «Мулами» и «Слонами» в течение каждого из 7 дней

τ	$\bar{\sigma}$	$\frac{\lvert\sigma_i-\bar{\sigma}\rvert}{\bar{\sigma}}\cdot 100\%$							Класс пользователя
		i=1	i=2	i=3	i=4	i=5	i=6	i=7	
0,1	555543,79	3,30	1,05	10,83	9,89	19,85	3,52	32,70	«Мыши»
1	4084729,38	3,76	6,02	21,23	17,96	33,50	15,17	59,77	
0,1	467893,92	8,26	2,63	7,50	3,56	28,93	14,24	14,53	«Мулы»
1	3394353,41	2,25	1,35	7,68	9,54	16,44	8,09	6,41	
0,1	933262,70	11,61	4,31	15,78	58,04	13,45	20,67	42,36	«Слоны»
1	7414378,04	1,16	5,67	14,87	68,46	9,83	13,33	43,26	

Данные, представленные в табл. 3.3–3.4, позволяют оценить вариативность параметров распределений изученных СП в каждый из семи дней. (Здесь $\bar{\mu},\bar{\sigma}$ средние по семи дням значения параметров распределения.)

Из табл. 3.3–3.4 видно, что изученные СП вне зависимости от размера окна агрегации τ в разные дни принадлежат различным генеральным совокупностям, отличающимся друг от друга значениями параметров μ,σ. При этом максимальные вариации параметров значений $\mu_i,\sigma_i, i=\overline{1,7}$ относительно значений $\bar{\mu},\bar{\sigma}$ составили:

1. у «Мышей»: $\Delta\mu_{0.1}=17,25\%$, $\Delta\sigma_{0.1}=32,70\%$; $\Delta\mu_{1.0}=17,23\%$, $\Delta\sigma_{1.0}=59,77\%$;
2. у «Мулов»: $\Delta\mu_{0.1}=77,74\%$, $\Delta\sigma_{0.1}=28,93\%$; $\Delta\mu_{1.0}=68,95\%$, $\Delta\sigma_{1.0}=16,44\%$;
3. у «Слонов»: $\Delta\mu_{0.1}=43,09\%$, $\Delta\sigma_{0.1}=58,04\%$; $\Delta\mu_{1.0}=43,41\%$, $\Delta\sigma_{1.0}=68,46\%$,

для размеров окон агрегации 0.1 и 1.0 с, соответственно (здесь значение нижнего индекса соответствует размеру окна агрегации τ).

Таким образом, вне зависимости от размера окна агрегации τ наиболее стабильными значения параметров распределения μ,σ оказываются у класса пользователей «Мыши», наименее стабильными – у класса пользователей «Мулы».

Значения $\bar{\mu}_{1.0}, \bar{\sigma}_{1.0}$ и $\bar{\mu}_{0.1}, \bar{\sigma}_{0.1}$ оказываются связанными между собой следующими соотношениями:

1. «Мыши»: $\bar{\mu}_{1,0} \approx 9,99\bar{\mu}_{0,1}, \bar{\sigma}_{1,0} \approx 7.35\bar{\sigma}_{0,1}$;
2. «Мулы»: $\bar{\mu}_{1,0} \approx 10,52\bar{\mu}_{0,1}, \bar{\sigma}_{1,0} \approx 7.25\bar{\sigma}_{0,1}$;
3. «Слоны»: $\bar{\mu}_{1,0} \approx 9,99\bar{\mu}_{0,1}, \bar{\sigma}_{1,0} \approx 7.94\bar{\sigma}_{0,1}$.

Из приведенных соотношений видно, что использованная процедура агрегации размера потоков не является масштабно-инвариантным преобразованием по параметру σ, а потому статистические свойства СП, содержащих значения объемов переданной информации каждым из выбранных классов пользователей в течении данного временного оказываются зависящими от размера окна агрегации τ. Данное обстоятельство следует принимать во внимание при статистическом моделировании информационных потоков в КС.

Рассмотрим оценки параметров $\bar{\mu}_i$, $\bar{\sigma}_i$, $i = \overline{1,7}$ плотностей распределений СП $V_i^{\langle 0.1 \rangle}$, $V_i^{\langle 1.0 \rangle}$, полученные для каждого из 7 дней, а также относительные отклонения независимых оценок данных параметров $\mu_{i,k}$, $\sigma_{i,k}$, полученные по двум и четырем частям исходной выборки, от $\bar{\mu}_i$, $\bar{\sigma}_i$ (табл. 3.5).

Таблица 3.5

Оценки параметров распределений СП $V_i^{\langle 0.1\rangle}$, $V_i^{\langle 1.0\rangle}$

i	τ	$\bar{\mu}_i \cdot 10^5$	$\frac{\lvert\mu_{i,k}-\bar{\mu}_i\rvert}{\bar{\mu}_i}$, %		$\frac{\lvert\mu_{i,k}-\bar{\mu}_i\rvert}{\bar{\mu}_i}$, %				$\bar{\sigma}_i \cdot 10^5$	$\frac{\lvert\sigma_{i,k}-\bar{\sigma}_i\rvert}{\bar{\sigma}_i}$, %		$\frac{\lvert\sigma_{i,k}-\bar{\sigma}_i\rvert}{\bar{\sigma}_i}$, %				Класс пользо-вателя
		k=1	k=1	k=2	k=1	k=2	k=3	k=4	k=1	k=1	k=2	k=1	k=2	k=3	k=4	
1	0,1	25,03	5,93	5,79	3,96	15,90	4,71	6,72	5,74	0,13	6,61	14,77	6,14	0,37	15,47	«Мыши»
		10,70	2,92	2,64	2,76	4,35	0,82	4,39	5,07	8,98	23,47	3,63	18,48	58,15	11,45	«Мулы»
		49,36	1,11	0,95	0,81	3,19	5,45	7,27	10,42	11,69	12,29	17,40	8,79	26,24	14,90	«Слоны»
	1,0	250,50	5,69	6,01	4,28	15,61	5,07	7,02	42,38	3,27	17,43	25,04	9,21	0,14	30,81	«Мыши»
		108,52	2,45	2,40	0,07	5,50	9,59	4,57	33,18	7,41	8,56	3,98	11,56	21,88	11,26	«Мулы»
		494,36	0,88	1,20	1,08	2,90	4,39	7,40	73,28	8,81	9,98	14,47	4,87	20,27	12,67	«Слоны»
2	0,1	28,66	4,64	5,06	9,22	0,61	6,23	3,75	5,61	11,30	19,52	2,06	13,69	21,68	18,02	«Мыши»
		12,56	7,99	7,86	10,11	5,20	2,56	13,51	4,80	2,18	7,07	1,81	3,59	6,46	7,31	«Мулы»
		37,26	3,91	3,69	0,10	7,81	8,93	1,39	8,93	5,76	7,26	6,05	3,71	12,38	8,37	«Слоны»
	1,0	286,52	4,46	5,21	9,29	11,38	6,49	4,06	38,39	15,06	25,79	11,78	83,48	31,95	20,43	«Мыши»
		125,60	8,08	7,44	9,38	5,30	2,39	12,99	34,40	5,74	13,32	5,62	5,88	14,82	13,67	«Мулы»
		372,54	3,68	4,04	0,98	7,52	8,88	1,04	69,94	7,63	9,86	11,76	5,47	22,77	7,96	«Слоны»
3	0,1	25,56	3,07	3,11	2,56	9,18	0,35	5,83	4,95	2,48	4,95	8,62	14,12	3,37	8,07	«Мыши»
		8,59	1,44	1,48	0,45	2,43	0,36	8,64	4,33	21,96	11,27	26,75	6,18	0,86	5,77	«Мулы»
		30,93	8,60	8,83	16,20	1,21	3,36	14,12	7,86	5,95	5,01	5,52	15,16	3,67	10,71	«Слоны»
	1,0	255,32	3,11	3,08	2,73	8,94	0,28	5,59	32,18	4,47	6,56	4,21	18,36	2,82	19,23	«Мыши»
		87,17	2,16	2,77	7,60	0,23	1,09	8,15	31,34	13,49	15,28	61,28	1,57	8,81	18,29	«Мулы»
		309,73	8,39	8,53	16,16	1,46	2,60	13,75	63,11	10,26	8,10	12,64	22,16	0,08	17,00	«Слоны»
4	0,1	25,65	2,71	2,58	7,54	2,20	1,76	3,28	5,01	2,68	3,65	4,19	6,74	1,36	4,32	«Мыши»
		8,47	7,93	3,68	9,84	9,81	9,20	17,91	4,51	0,07	13,12	2,84	8,49	0,89	10,62	«Мулы»
		42,73	16,87	16,58	21,66	12,53	22,66	10,26	14,75	18,35	7,36	26,70	11,10	9,07	8,18	«Слоны»
	1,0	256,61	2,53	2,81	7,47	2,32	2,05	3,44	33,51	5,25	7,72	7,07	11,34	5,24	11,57	«Мыши»
		85,77	12,29	6,67	9,68	37,00	5,02	18,24	30,71	17,55	0,45	14,38	60,61	22,49	8,30	«Мулы»
		424,36	16,33	17,50	20,97	11,37	23,37	10,72	124,90	31,04	10,87	42,77	23,11	12,53	13,01	«Слоны»
5	0,1	21,71	1,54	1,22	1,55	1,61	0,50	2,84	4,45	7,26	6,29	6,08	8,93	5,78	9,16	«Мыши»

i	τ	$\bar{\mu}_i \cdot 10^5$	$\frac{\lvert\mu_{i,k} - \bar{\mu}_i\rvert}{\bar{\mu}_i}, \%$		$\frac{\lvert\mu_{i,k} - \bar{\mu}_i\rvert}{\bar{\mu}_i}, \%$				$\bar{\sigma}_i \cdot 10^5$	$\frac{\lvert\sigma_{i,k} - \bar{\sigma}_i\rvert}{\bar{\sigma}_i}, \%$		$\frac{\lvert\sigma_{i,k} - \bar{\sigma}_i\rvert}{\bar{\sigma}_i}, \%$				Класс пользо-вателя
		k=1	k=1	k=2	k=1	k=2	k=3	k=4	k=1	k=1	k=2	k=1	k=2	k=3	k=4	
		2,45	49,12	37,06	10,67	14,47	28,98	1,91	6,03	11,40	3,50	6,19	5,62	17,21	24,76	«Мулы»
		39,26	5,14	4,94	4,34	14,55	3,47	15,00	10,59	6,11	3,23	25,80	4,08	2,28	0,06	«Слоны»
	1,0	216,78	1,48	1,52	2,11	1,27	3,10	2,65	27,16	14,52	10,38	2,30	15,34	39,80	12,95	«Мыши»
		48,27	20,47	35,87	19,65	23,66	9,31	45,22	39,52	28,95	24,82	27,08	35,49	39,90	49,39	«Мулы»
		392,79	5,10	5,20	4,49	14,05	3,21	14,03	81,44	7,86	0,65	26,13	12,98	5,67	9,57	«Слоны»
6	0,1	22,48	1,50	1,21	3,23	0,28	1,36	1,06	5,75	12,43	12,41	30,69	6,93	11,53	16,14	«Мыши»
		3,66	15,77	1,38	19,35	3,03	25,43	69,81	4,01	17,02	6,76	22,20	31,43	1,79	28,35	«Мулы»
		19,97	19,16	17,71	28,34	11,29	20,80	13,89	7,40	4,09	19,87	12,41	7,83	23,88	15,58	«Слоны»
	1,0	224,12	1,54	1,45	2,46	0,07	1,67	2,75	47,04	19,74	19,00	40,61	5,06	17,62	14,29	«Мыши»
		31,78	22,69	46,50	38,42	29,09	45,17	0,41	36,69	9,57	40,38	4,91	54,27	49,65	9,43	«Мулы»
		199,83	20,39	17,32	28,14	10,77	20,83	13,20	64,26	8,48	29,87	13,63	15,15	31,40	28,80	«Слоны»
7	0,1	22,00	8,57	9,73	7,54	9,75	15,45	55,32	7,37	17,85	9,34	16,79	21,78	14,46	21,27	«Мыши»
		3,02	9,17	37,69	39,95	53,86	44,42	17,11	4,00	12,29	5,39	13,14	19,56	7,27	4,09	«Мулы»
		21,97	1,30	1,37	0,77	3,14	5,23	8,34	5,38	1,54	1,09	12,75	13,21	2,10	6,03	«Слоны»
	1,0	221,06	7,85	9,73	6,73	9,13	14,09	41,78	65,26	23,88	12,40	21,04	23,90	20,84	15,77	«Мыши»
		33,30	6,56	17,45	40,23	29,61	15,63	9,86	31,77	1,47	9,55	30,95	10,80	4,87	17,66	«Мулы»
		219,36	1,21	1,34	0,98	3,27	4,91	7,91	42,07	1,05	3,74	15,50	18,68	2,11	4,15	«Слоны»

Из табл. 3.5 видно, что максимальные отклонения значений параметров $\Delta\mu_{\tau,k} = \max\limits_{i}\left(\frac{\left|\mu_{\tau,i,k} - \overline{\mu}_{\tau,i}\right|}{\overline{\mu}_{\tau,i}}\right)$, $\Delta\sigma_{\tau,k} = \max\limits_{i}\left(\frac{\left|\sigma_{\tau,i,k} - \overline{\sigma}_{\tau,i}\right|}{\overline{\sigma}_{\tau,i}}\right)$ для выбранных классов потоков составили:

1. «Мыши»:

$$\Delta\mu_{0,1\,k=1,2} = 9,73\%, \quad \Delta\sigma_{0,1\,k=1,2} = 19,52\%;$$

$$\Delta\mu_{0,1\,k=\overline{1,4}} = 55,32\%, \quad \Delta\sigma_{0,1\,k=\overline{1,4}} = 30,69\%;$$

$$\Delta\mu_{1,0\,k=1,2} = 9,72\%, \quad \Delta\sigma_{1,0\,k=1,2} = 25,79\%;$$

$$\Delta\mu_{1,0\,k=\overline{1,4}} = 41,78\%, \quad \Delta\sigma_{1,0\,k=\overline{1,4}} = 83,48\%;$$

2. «Мулы»:

$$\Delta\mu_{0,1\,k=1,2} = 49,12\%, \quad \Delta\sigma_{0,1\,k=1,2} = 23,47\%;$$

$$\Delta\mu_{0,1\,k=\overline{1,4}} = 69,81\%, \quad \Delta\sigma_{0,1\,k=\overline{1,4}} = 58,15\%;$$

$$\Delta\mu_{1,0\,k=1,2} = 46,50\%, \quad \Delta\sigma_{1,0\,k=1,2} = 40,38\%;$$

$$\Delta\mu_{1,0\,k=\overline{1,4}} = 45,22\%, \quad \Delta\sigma_{1,0\,k=\overline{1,4}} = 61,28\%;$$

3. «Слоны»:

$$\Delta\mu_{0,1\,k=1,2} = 19,16\%, \quad \Delta\sigma_{0,1\,k=1,2} = 19,87\%;$$

$$\Delta\mu_{0,1\,k=\overline{1,4}} = 28,34\%, \quad \Delta\sigma_{0,1\,k=\overline{1,4}} = 26,70\%;$$

$$\Delta\mu_{1,0\,k=1,2} = 20,39\%, \quad \Delta\sigma_{1,0\,k=1,2} = 31,04\%;$$

$$\Delta\mu_{1,0\,k=\overline{1,4}} = 28,14\%, \quad \Delta\sigma_{1,0\,k=\overline{1,4}} = 42,77\%.$$

Приведенные выше результаты, подтверждают, что наиболее вариативным, оказывается параметр μ у «Мулов», даже при его оценке по СП $V_i^{\langle 0.1\rangle}$, $V_i^{\langle 1.0\rangle}$, зарегистрированным в один и тот же день.

Анализ результатов вычисления параметров распределения СП $V_i^{\langle 0.1\rangle}$, $V_i^{\langle 1.0\rangle}$, позволяет сделать следующие выводы.

1. СП $V_i^{\langle 0.1\rangle}$, $V_i^{\langle 1.0\rangle}$ вне зависимости от размера окна агрегации являются СП с ограниченными областями рассеяния.

2. Для аппроксимации ПР изученных СП $V_i^{\langle 0.1\rangle}$, $V_i^{\langle 1.0\rangle}$ целесообразно использовать обобщенный закон распределения нормальной с ограниченной областью рассеяния.

3. Оценки параметров ПР СП $V_i^{\langle 0.1\rangle}$, $V_i^{\langle 1.0\rangle}$ не являются стационарными величинами, но могут меняться как день ото дня, так и в течение одного дня.

4. Наиболее стабильными параметры ПР изученных СП $V_i^{\langle 0.1\rangle}$, $V_i^{\langle 1.0\rangle}$, как в течение одного дня, так и недели, оказываются у потоков, создаваемых классом пользователей «Мыши».

5. Наименее стабильные параметры ПР изученных СП $V_i^{\langle 0.1\rangle}$, $V_i^{\langle 1.0\rangle}$, как в течение одного дня, так и в течение недели, имеют потоки классов «Мулы» и «Слоны», у которых отклонения параметра μ от среднего значения в течение недели могут достигать 60% и 40%, соответственно.

3.5. Анализ результатов исследования дампа Интерент-трафика в разрезе количества пакетов

Рассмотрим оценки параметров распределений СП $N_i^{\langle 0,1\rangle}$, $N_i^{\langle 1,0\rangle}$, содержащих агрегированные значения количества пакетов, переданных в течение выбранного временного интервала в каждый из 7 дней, представленные в табл. 3.6–3.7.

Таблица 3.6

Параметры распределений СП $N_i^{\langle 0,1\rangle}$, $N_i^{\langle 1,0\rangle}$

τ	$\bar{\mu}$	$\frac{\lvert\mu_i-\bar{\mu}\rvert}{\bar{\mu}}\cdot 100\%$							$\bar{\sigma}$	$\frac{\lvert\sigma_i-\bar{\sigma}\rvert}{\bar{\sigma}}\cdot 100\%$							Класс пользователей
		i=1	i=2	i=3	i=4	i=5	i=6	i=7		i=1	i=2	i=3	i=4	i=5	i=6	i=7	
0,1	9813	1,60	5,92	2,59	1,30	0,09	1,07	9,19	1772	26,72	27,91	2,02	9,71	4,66	36,05	25,64	«Мыши»
1	97901	1,33	6,04	2,36	1,54	0,25	0,61	8,98	8992	18,03	26,44	12,28	7,04	0,75	17,34	33,12	
0,1	666	54,05	68,70	45,11	8,58	53,94	66,41	56,09	434	21,31	8,17	16,74	2,32	24,67	4,61	14,13	«Мулы»
1	7409	42,41	51,60	30,23	1,96	27,68	47,77	50,75	2915	21,53	1,14	2,13	0,82	4,50	17,14	7,07	
0,1	3004	34,42	4,62	6,22	24,45	16,55	38,10	35,74	722	10,50	0,25	17,63	48,54	14,48	20,30	35,84	«Слоны»
1	30029	34,51	4,66	6,20	24,40	16,53	38,15	35,74	5663	1,45	0,88	18,46	54,38	13,35	13,09	33,85	

Данные, представленные в табл. 3.6, позволяют оценить вариативность параметров распределений изученных СП $N_i^{\langle 0,1\rangle}$, $N_i^{\langle 1,0\rangle}$ в каждый из семи дней. (Здесь $\bar{\mu},\bar{\sigma}$ средние по семи дням значения параметров распределения СП $N_i^{\langle 0,1\rangle}$, $N_i^{\langle 1,0\rangle}$.)

Из табл. 3.6 видно, что изученные СП $N_i^{\langle 0,1\rangle}$, $N_i^{\langle 1,0\rangle}$ вне зависимости от размера окна агрегации τ в разные дни принадлежат различным генеральным совокупностям, отличающимся друг от друга значениями параметров μ,σ. При этом максимальные вариации параметров значений $\mu_i,\sigma_i,i=\overline{1,7}$ относительно значений $\bar{\mu},\bar{\sigma}$ составили:

1. у «Мышей»: $\Delta\mu_{0.1}=9,19\%,\quad \Delta\sigma_{0.1}=30,05\%;\quad \Delta\mu_{1.0}=8,98\%,\quad \Delta\sigma_{1.0}=31,20\%;$

2. у «Мулов»: $\Delta\mu_{0.1} = 68,70\%,\ \Delta\sigma_{0.1} = 24,67\%;\quad \Delta\mu_{1.0} = 51,60\%,\quad \Delta\sigma_{1.0} = 21,53\%;$

3. у «Слонов»: $\Delta\mu_{0.1} = 38,10\%,\ \Delta\sigma_{0.1} = 48,54\%;\ \Delta\mu_{1.0} = 38,15\%,\ \Delta\sigma_{1.0} = 54,38\%.$

для размеров окон агрегации 0.1 и 1.0 с, соответственно (здесь значение нижнего индекса соответствует размеру окна агрегации τ).

Таким образом, вне зависимости от размера окна агрегации τ, наиболее стабильными значения параметров распределения μ, σ оказываются у класса потоков «Мыши», наименее стабильными – у класса потоков «Мулы».

Значения $\bar{\mu}_{1.0}, \bar{\sigma}_{1.0}$ и $\bar{\mu}_{0.1}, \bar{\sigma}_{0.1}$ оказались связанными между собой следующими соотношениями:

1. «Мыши»: $\bar{\mu}_{1,0} \approx 9,98\bar{\mu}_{0,1}, \bar{\sigma}_{1,0} \approx 5.01\bar{\sigma}_{0,1};$

2. «Мулы»: $\bar{\mu}_{1,0} \approx 11,12\bar{\mu}_{0,1}, \bar{\sigma}_{1,0} \approx 6.72\bar{\sigma}_{0,1};$

3. «Слоны»: $\bar{\mu}_{1,0} \approx 10,00\bar{\mu}_{0,1}, \bar{\sigma}_{1,0} \approx 7.8\bar{\sigma}_{0,1}.$

Из приведенных соотношений видно, что использованная процедура агрегации числа переданных пакетов не является масштабно-инвариантным преобразованием по параметру σ, а потому статистические свойства СП СП $N_i^{\langle 0,1\rangle}$, $N_i^{\langle 1,0\rangle}$ оказываются зависящими от размера окна агрегации τ. Данное обстоятельство следует принимать во внимание при статистическом моделировании информационных потоков в компьютерных сетях.

Рассмотрим оценки параметров $\bar{\mu}_i,\ \bar{\sigma}_i, i = \overline{1,7}$ ПР СП $N_i^{\langle 0.1\rangle}$, $N_i^{\langle 1.0\rangle}$, полученные для каждого из 7 дней, а также относительные отклонения независимых оценок данных параметров $\mu_{i,k},\ \sigma_{i,k}$, полученные по двум и четырем частям исходной выборки, от $\bar{\mu}_i,\ \bar{\sigma}_i$ (табл. 3.7).

Таблица 3.7

Параметры ПР СП $N_i^{\langle 0.1\rangle}$, $N_i^{\langle 1.0\rangle}$

i	τ	$\bar{\mu}_i \cdot 10^3$	$\frac{\lvert\mu_{i,k}-\bar{\mu}_i\rvert}{\bar{\mu}_i}$, %		$\frac{\lvert\mu_{i,k}-\bar{\mu}_i\rvert}{\bar{\mu}_i}$, %				$\bar{\sigma}_i \cdot 10^2$	$\frac{\lvert\sigma_{i,k}-\bar{\sigma}_i\rvert}{\bar{\sigma}_i}$, %		$\frac{\lvert\sigma_{i,k}-\bar{\sigma}_i\rvert}{\bar{\sigma}_i}$, %				Класс пользо-вателей
		k=1	k=1	k=2	k=1	k=2	k=3	k=4	k=1	k=1	k=2	k=1	k=2	k=3	k=4	
1	0,1	96,56	1,40	1,48	3,28	0,45	1,69	4,69	12,99	0,22	0,29	4,37	5,38	1,06	3,14	«Мыши»
		10,26	1,95	1,59	5,28	8,62	11,28	7,50	5,27	4,29	10,80	6,33	27,30	37,42	27,81	«Мулы»
		40,39	1,34	1,43	0,35	2,24	2,82	5,82	7,98	7,32	5,69	10,84	1,18	13,54	6,95	«Слоны»
	1	965,98	1,43	1,41	3,32	0,44	1,69	4,48	73,70	3,71	0,83	9,42	5,07	12,03	4,21	«Мыши»
		105,50	5,82	1,98	2,87	11,42	14,32	9,92	35,43	14,53	2,21	20,99	15,71	10,87	25,70	«Мулы»
		403,91	1,40	1,57	0,32	2,28	2,15	5,85	55,81	0,91	1,27	1,98	6,19	5,94	5,40	«Слоны»
2	0,1	103,94	0,06	0,08	0,01	0,06	0,96	0,74	12,78	6,76	6,94	7,16	7,00	7,12	8,62	«Мыши»
		11,24	7,54	8,39	11,04	4,41	1,13	15,13	3,99	1,99	4,26	2,34	1,17	3,43	10,00	«Мулы»
		31,43	1,41	1,57	1,14	4,05	6,75	3,64	7,24	5,73	7,39	10,01	1,29	12,52	7,55	«Слоны»
	1	1038,16	0,08	0,17	0,15	0,10	0,98	1,10	66,14	2,30	6,10	5,31	1,46	6,03	14,06	«Мыши»
		112,31	7,53	7,83	11,45	4,19	0,96	15,10	29,49	0,18	8,54	3,05	0,29	9,37	15,83	«Мулы»
		314,27	1,34	1,77	2,41	3,71	6,74	3,28	56,14	10,69	6,64	28,91	3,51	19,10	4,46	«Слоны»
3	0,1	100,67	0,67	0,83	0,40	1,67	0,82	0,39	17,36	6,49	1,66	3,12	11,23	2,67	4,54	«Мыши»
		9,67	2,79	2,45	1,39	6,47	0,19	4,12	3,62	0,69	0,76	15,07	10,70	3,39	1,81	«Мулы»
		28,18	2,77	2,87	8,05	2,01	0,91	6,60	5,95	1,37	2,30	5,19	5,42	1,25	5,02	«Слоны»
	1	1002,11	0,46	0,46	0,63	1,65	0,92	0,05	78,87	3,85	0,55	5,07	8,55	10,10	7,21	«Мыши»
		96,48	2,59	2,11	0,09	6,47	0,58	4,02	28,53	2,25	1,81	25,39	18,33	5,57	3,71	«Мулы»
		281,68	2,82	2,91	7,89	1,86	0,66	6,57	46,18	0,82	3,64	2,50	4,18	1,06	7,61	«Слоны»
4	0,1	99,40	4,86	4,06	8,55	0,93	3,66	4,39	16,00	6,83	5,68	6,10	3,37	13,61	4,33	«Мыши»
		7,24	5,64	7,02	6,01	6,74	2,89	15,31	4,24	6,31	17,47	19,65	29,99	26,18	6,85	«Мулы»
		37,39	12,88	12,78	19,79	5,64	16,75	8,59	10,72	12,10	10,04	27,19	4,21	12,88	8,24	«Слоны»
	1	994,11	4,19	4,40	7,52	0,74	3,67	5,08	96,25	12,65	12,40	28,15	10,23	27,88	2,92	«Мыши»
		75,54	9,74	1,93	8,57	13,99	6,70	11,74	28,91	24,63	11,73	16,14	61,60	30,55	2,31	«Мулы»
		373,55	12,85	12,73	19,95	5,68	17,15	8,05	87,43	17,54	13,25	40,66	13,69	16,18	10,87	«Слоны»
5	0,1	98,04	0,08	0,74	0,84	0,12	0,28	2,17	18,55	12,63	0,34	6,90	11,57	3,47	9,92	«Мыши»
		3,07	78,84	3,87	109,30	38,25	46,95	32,17	5,41	25,25	0,15	27,85	24,98	41,87	8,76	«Мулы»
		35,02	3,58	3,63	3,62	10,67	3,98	11,73	8,26	7,03	5,41	23,61	1,72	0,20	5,96	«Слоны»

i	τ	$\bar{\mu}_i \cdot 10^3$	$\frac{\lvert\mu_{i,k} - \bar{\mu}_i\rvert}{\bar{\mu}_i}, \%$		$\frac{\lvert\mu_{i,k} - \bar{\mu}_i\rvert}{\bar{\mu}_i}, \%$				$\bar{\sigma}_i \cdot 10^2$	$\frac{\lvert\sigma_{i,k} - \bar{\sigma}_i\rvert}{\bar{\sigma}_i}, \%$		$\frac{\lvert\sigma_{i,k} - \bar{\sigma}_i\rvert}{\bar{\sigma}_i}, \%$				Класс пользо-вателей
		k=1	k=1	k=2	k=1	k=2	k=3	k=4	k=1	k=1	k=2	k=1	k=2	k=3	k=4	
	1	976,60	1,08	1,06	0,94	1,18	0,39	1,74	89,24	5,48	5,49	8,88	1,81	6,71	0,61	«Мыши»
		53,58	12,06	31,77	25,30	4,18	11,21	23,94	30,47	14,06	24,76	12,10	20,10	37,36	45,99	«Мулы»
		349,94	3,59	3,64	3,32	10,82	4,00	11,20	64,19	10,50	4,83	28,74	11,34	8,82	1,42	«Слоны»
6	0,1	99,18	0,15	0,03	0,50	0,85	0,14	0,34	24,11	0,67	1,38	0,33	1,37	0,81	0,45	«Мыши»
		2,24	44,10	44,73	79,60	63,23	61,85	14,33	4,14	1,95	26,24	7,47	42,20	41,19	9,82	«Мулы»
		18,60	11,68	11,16	15,74	8,71	14,93	7,37	5,75	3,87	12,93	3,34	8,55	13,84	13,25	«Слоны»
	1	984,99	0,17	0,20	0,36	0,79	0,41	0,10	105,51	0,98	1,37	6,51	3,07	2,77	2,92	«Мыши»
		38,69	3,09	5,75	19,14	1,70	0,72	6,14	24,16	22,56	13,66	35,54	35,04	38,53	6,34	«Мулы»
		185,72	12,56	11,29	15,54	9,28	15,04	7,45	49,22	11,58	21,27	2,12	13,80	19,14	25,75	«Слоны»
7	0,1	89,11	3,42	3,31	6,22	0,69	3,56	10,00	22,27	3,70	1,51	2,97	4,43	2,43	2,31	«Мыши»
		2,93	6,04	16,00	54,03	40,65	6,20	16,42	3,73	8,15	9,15	32,22	0,58	14,09	15,47	«Мулы»
		19,31	1,49	1,52	0,40	2,53	5,59	9,06	4,63	0,12	0,58	10,73	11,32	1,34	6,67	«Слоны»
	1	891,11	3,42	3,41	6,19	0,65	3,13	10,04	119,70	9,22	2,36	10,72	13,02	0,73	18,41	«Мыши»
		36,49	0,73	7,85	23,25	27,85	12,37	7,38	27,09	5,37	6,31	31,56	2,95	11,54	18,25	«Мулы»
		192,96	1,44	1,80	1,69	2,68	3,95	9,43	37,47	3,58	2,99	4,21	14,28	11,03	1,79	«Слоны»

Из таблицы 3.7 видно, что максимальные значения параметров $\Delta\mu_{\tau,k} = \max_i\left(\frac{\left|\mu_{\tau,i,k} - \overline{\mu}_{\tau,i}\right|}{\overline{\mu}_{\tau,i}}\right)$, $\Delta\sigma_{\tau,k} = \max_i\left(\frac{\left|\sigma_{\tau,i,k} - \overline{\sigma}_{\tau,i}\right|}{\overline{\sigma}_{\tau,i}}\right)$ для выбранных классов потоков равны:

1. «Мыши»:

$$\Delta\mu_{0,1\,k=1,2} = 4,86\%, \quad \Delta\sigma_{0,1\,k=1,2} = 12,63\%;$$
$$\Delta\mu_{0,1\,k=1,4} = 10,00\%, \quad \Delta\sigma_{0,1\,k=1,4} = 13,61\%;$$
$$\Delta\mu_{1,0\,k=\overline{1,2}} = 4,40\%, \quad \Delta\sigma_{1,0\,k=\overline{1,2}} = 12,65\%;$$
$$\Delta\mu_{1,0\,k=\overline{1,4}} = 10,04\%, \quad \Delta\sigma_{1,0\,k=\overline{1,4}} = 27,88\%;$$

2. «Мулы»:

$$\Delta\mu_{0,1\,k=1,2} = 78,84\%, \quad \Delta\sigma_{0,1\,k=1,2} = 24,00\%;$$
$$\Delta\mu_{0,1\,k=1,4} = 109,30\%, \quad \Delta\sigma_{0,1\,k=1,4} = 42,20\%;$$
$$\Delta\mu_{1,0\,k=\overline{1,2}} = 31,77\%, \quad \Delta\sigma_{1,0\,k=\overline{1,2}} = 24,76\%;$$
$$\Delta\mu_{1,0\,k=\overline{1,4}} = 27,85\%, \quad \Delta\sigma_{1,0\,k=\overline{1,4}} = 61,60\%;$$

3. «Слоны»:

$$\Delta\mu_{0,1\,k=1,2} = 11,68\%, \quad \Delta\sigma_{0,1\,k=1,2} = 13,93\%;$$
$$\Delta\mu_{0,1\,k=1,4} = 19,79\%, \quad \Delta\sigma_{0,1\,k=1,4} = 27,19\%;$$
$$\Delta\mu_{1,0\,k=\overline{1,2}} = 12,85\%, \quad \Delta\sigma_{1,0\,k=\overline{1,2}} = 21,27\%;$$
$$\Delta\mu_{1,0\,k=\overline{1,4}} = 19,95\%, \quad \Delta\sigma_{1,0\,k=\overline{1,4}} = 40,66\%.$$

Полученные результаты, подтверждают, что параметр μ наиболее вариативным оказывается у «Мулов», даже при его оценке по СП $N_i^{\langle 0,1\rangle}$, $N_i^{\langle 1,0\rangle}$, зарегистрированной в один и тот же день. Увеличение количества разбиений исходной СП (соответственно, уменьшения длительности временного интервала τ) для потока класса «Мулы» позволяет получить менее вариативную оценку параметра μ (31,70% вместо 78,84%), при практически неизменяющейся вариативности параметра σ (24,00% и 24,74%). Напротив, для потоков классов «Мыши» и «Слоны» увеличение числа разбиений исходной СП приводит к уменьшению вариативности оценок параметра μ с 4,86% до 10,00% и с 11,68% до 19,79%, соответственно. При этом у «Мышей» практически неизменной остается вариативность оценки параметра σ (12,63% и 11,57%), а у «Слонов» она изменяется в примерно в два раза (13,93% и 27,19%).

Анализ полученных результатов вычисления параметров ПР СП $N_i^{\langle 0,1\rangle}$, $N_i^{\langle 1,0\rangle}$, созданных каждым из классов пользователей позволяет сделать следующие выводы:

1. Число СП $N_i^{\langle 0,1\rangle}$, $N_i^{\langle 1,0\rangle}$, созданным каждым из классов пользователей вне зависимости от размера окна агрегации являются СП с ограниченными областями рассеяния.

2. Оценки параметров ПР СП $N_i^{\langle 0,1\rangle}$, $N_i^{\langle 1,0\rangle}$ $_{N_i}$ не являются стационарными величинами, но могут меняться по дням, так и в течение одного дня.

3. Наиболее стабильные параметры ПР СП $N_i^{\langle 0,1\rangle}$, $N_i^{\langle 1,0\rangle}$, как в течение одного дня, так и недели, оказываются у класса потоков «Мыши».

4. Наименее стабильные параметры ПР СП $N_i^{\langle 0,1\rangle}$, $N_i^{\langle 1,0\rangle}$, как в течение одного дня, так и в течение недели, имеют потоки классов «Мулы» и «Слоны», у которых отклонения параметра μ от среднего значения в течение недели могут достигать 60% и 40%, соответственно.

3.6. Анализ результатов исследования стационарности информационных потоков в магистральном интернет-канале

В связи с тем, что результаты, описанные в разделах 3.3, 3.4, оказались противоречащими результату формального применения теста Дики-Фуллера к ВР $N_i^{\langle\tau\rangle}, V_i^{\langle\tau\rangle}$, в соответствие с которым данные ряды следует считать стационарными, был проведен более подробный анализ статистических свойств СП $N_i^{\langle 0,1\rangle}, N_i^{\langle 1,0\rangle}$.

Для проведения исследования стационарности ПР изучаемых СП $N_i^{\langle 0,1\rangle}, N_i^{\langle 1,0\rangle}$ была использована методика, реализующаяся следующей последовательностью действий.

1. Для каждой СП $N_i^{\langle 0,1\rangle}, N_i^{\langle 1,0\rangle}$ вычислялись ПР в соответствие с методом мнимых источников [88] и аппроксимации Розенблатта-Парзена.

2. Каждая СП $N_i^{\langle 0,1\rangle}, N_i^{\langle 1,0\rangle}$ делилась на 2 части и для каждой из полученных СП2,1, СП2,2 выполнялся п.1.

3. Каждая СП $N_i^{\langle 0,1\rangle}, N_i^{\langle 1,0\rangle}$ делилась на 4 части и для каждого из полученных СП4,1–СП4,4 выполнялся п.1.

Для каждого класса потоков в каждый из дней были получены по 7 ПР СП $N_i^{\langle\tau\rangle}$ для окон агрегации размером 0,1 и 1,0 с. Примеры ПР ВР $N_i^{\langle\tau\rangle}$ для окон агрегации 0,1, и 1,0 с. представлены на рис. 3.4 и 3.5.

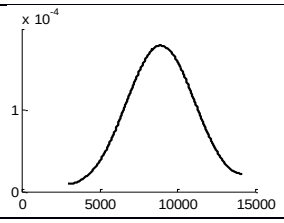
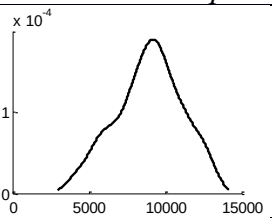
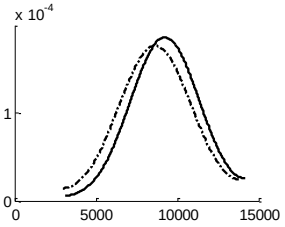
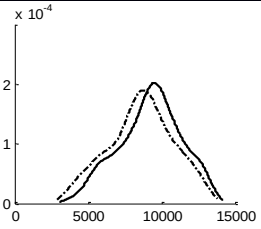
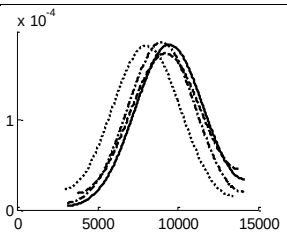
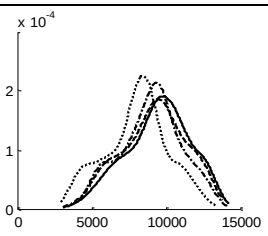
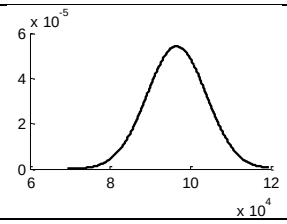
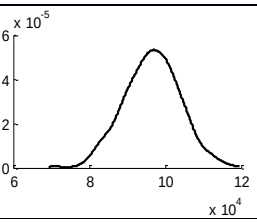
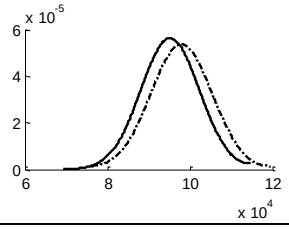
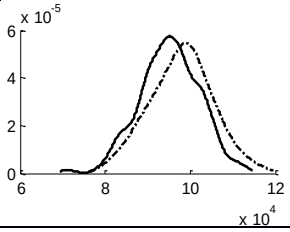
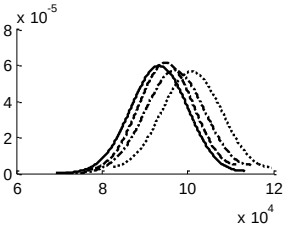
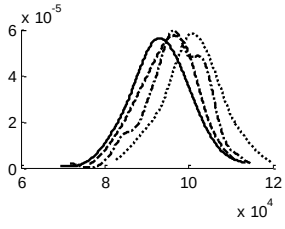
	Метод мнимых источников	*Аппроксимация Розенблатта-Парзена*
$ВР_{1,}$ 02.11.14 τ=1,0 с		
$ВР_{2,1}$, $ВР_{2,2,}$ 02.11.14 τ=1,0 с		
$ВР_{4,1}$ -$ВР_{4,4}$ 02.11.14 τ=1,0 с		
$ВР_{1,}$ 27.10.14 τ=0,1 с		
$ВР_{2,1}$, $ВР_{2,2}$ 27.10.14 τ=0,1 с		
$ВР_{4,1}$ – $ВР_{4,4,}$ 27.10.14 τ=0,1 с		

Рис. 3.4. ПР СП $N_i^{\langle\tau\rangle}$ (класс «Мыши»)

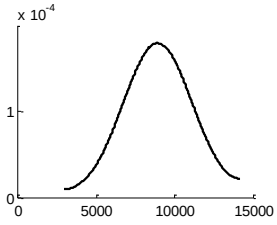
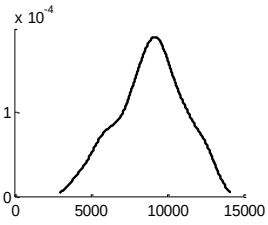
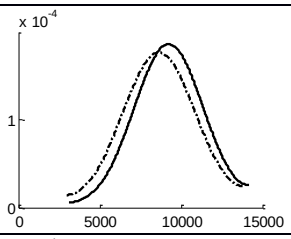
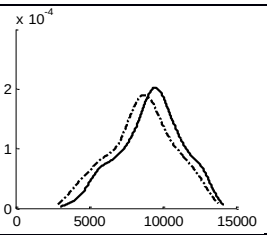
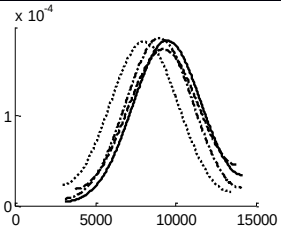
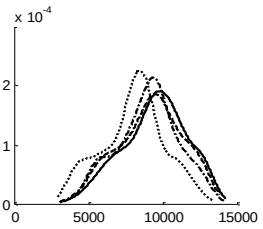
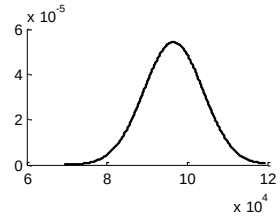
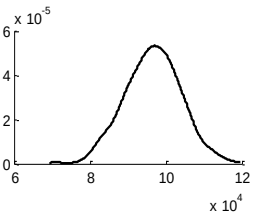
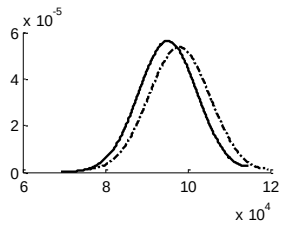
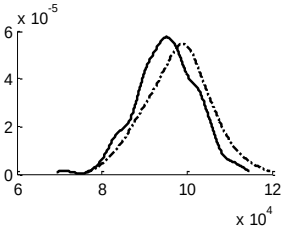
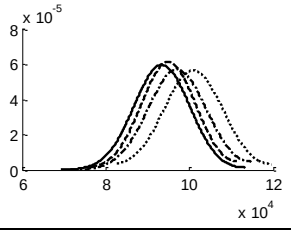
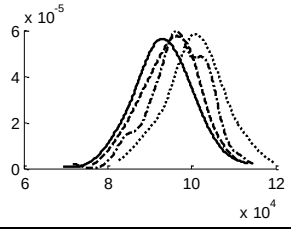
	Метод мнимых источников	*Аппроксимация Розенблатта-Парзена*
BP_1, 02.11.14		
τ=1 с		
$BP_{2,1}$, $BP_{2,2}$, 02.11.14		
τ=1 с		
$BP_{4,1} – BP_{4,4}$, 02.11.14		
τ=1 с		
BP_1, 27.10.14		
τ= 0.1 с		
$BP_{2,1}$, $BP_{2,2}$, 27.10.14		
τ= 0.1 с		
$BP_{4,1} – BP_{4,4}$, 27.10.14		
τ= 0.1 с		

Рис. 3.5. ПР СП $V_i^{\langle\tau\rangle}$ (класс «Мыши»)

Из рис. 3.4 и 3.5 видно, что параметры ПР, оцениваемые по всей СП $N_i^{\langle 0,1\rangle}, N_i^{\langle 1,0\rangle}$ и ее частям оказываются отличными друг от друга. Это, с нашей точки зрения, свидетельствует, вообще говоря, о его нестационарности.

Далее были получены оценки соответствия найденных аппроксимаций ПР СП $N_i^{\langle\tau\rangle}$ анализируемых СП действительной ПР. Для этого был использован критерий типа Колмогорова-Смирнова. При этом нулевая гипотеза формулировалась следующим образом:

найденная аппроксимация ПР соответствует ПР изучаемой СП $N_i^{\langle\tau\rangle}$. Оказалось, что для аппроксимации Розенблатта-Парзена данная гипотеза подтвердилась в 100% случаев. Данный результат вполне ожидаем, принимая во внимание, что аппроксимация Розенблатта-Парзена относится к непараметрическим методам аппроксимации ПР СП. Обобщенные результаты оценки адекватности аппроксимации ПР изученных СП СП $N_i^{\langle\tau\rangle}$ с помощью метода мнимых источников для 7 дней (27.10.14–02.11.14, имя файла дампа соответствует дате в формате: «год день время (всегда 1400).dump») и окон агрегации 0,1 и 1,0 с. представлены в табл.3.8 и 3.9.

Таблица 3.8

Процент выборок, для которых гипотеза о соответствии найденной аппроксимации ПР с помощью метода мнимых источников истинной ПР СП $N_i^{\langle 0.1\rangle}$ подтвердилась

Выборка	*«Мыши»*	*«Мулы»*	*«Слоны»*	τ
целая	0,0%	0,0%	0,0%	0,1 c/sec
	100,0%	42,9%	85,7%	1,0 c/sec
1/2	21,4%	0,0%	0,0%	0,1 c/sec
	100,0%	85,7%	85,7%	1,0 c/sec
1/4	21,4%	0,0%	17,8%	0,1 c/sec
	100,0%	96,4%	96,4%	1,0 c/sec

Таблица 3.9

Процент выборок, для которых гипотеза о соответствии найденной аппроксимации ПР с помощью метода мнимых источников ПР СП $N_i^{\langle 1,0\rangle}$

Выборка	*«Мыши»*	*«Мулы»*	*«Слоны»*	τ
целая	0,0%	0,0%	0,0%	0,1 c/sec
	42,9%	42,9%	28,6%	1,0 c/sec
1/2	0,0%	0,0%	0,0%	0,1 c/sec
	71,4%	85,7%	71,4%	1,0 c/sec
1/4	21,4%	0,0%	17,8%	0,1 c/sec
	71,4%	96,4%	92,9%	1,0 c/sec

Из табл. 3.8 и 3.9 видно, что при малых длительностях окна агрегации аппроксимация ПР с помощью мнимых источников оказывается неудовлетворительной. Данный результат, с нашей точки зрения, объясняется тем, что, как видно из рис. 3.4, ПР ВР, порожденных «Мышами», фактически образовано суперпозицией двух независимых друг от друг распределений. Методы расщепления подобных распределений для случая, когда анализируемая выборка представляет собой смешанное распределение, рассмотрены в [79]. При больших значениях окна агрегации, аппроксимацию ПР изученных СП $N_i^{\langle 0,1\rangle}, N_i^{\langle 1,0\rangle}$ можно считать удовлетворительной.

Результаты изучения статистических характеристик СП $N_i^{\langle 0,1\rangle}, N_i^{\langle 1,0\rangle}$ позволяют сделать следующие выводы.

1. При использовании окна агрегации 0,1 с. гипотеза о возможности аппроксимации ПР изученных СП $N_i^{\langle 0,1\rangle}, N_i^{\langle 1,0\rangle}$ (объёмы данных и кол-во пакетов) с помощью метода мнимых источников в большинстве случаев отвергается критерием типа Колмогорова-Смирнова.

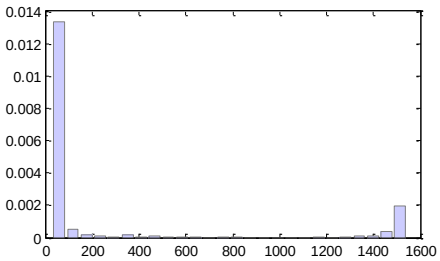
2. При использовании окна агрегации 0,1 с аппроксимации ПР СП $N_i^{\langle 0,1\rangle}, N_i^{\langle 1,0\rangle}$, построенные для частей ВР (объёмы данных и кол-во пакетов) как с помощью метода мнимых источников, так и с помощью аппроксимации Розенблатта-Парзена, изученных ВР оказываются отличными друг от друга, что свидетельствует о нестационарности СП $N_i^{\langle 0,1\rangle}, N_i^{\langle 1,0\rangle}$.

3. При использовании окна агрегации 1,0 с. для аппроксимации ПР методом мнимых источников изученных СП $N_i^{\langle 0,1\rangle}, N_i^{\langle 1,0\rangle}$ обеспечивается удовлетворительное качество аппроксимации.

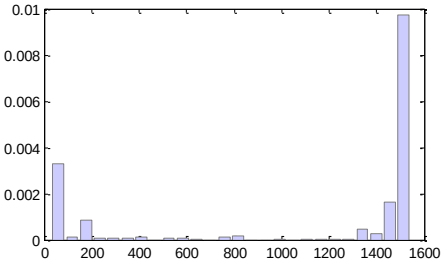
При использовании окна агрегации 1,0 с для аппроксимации ПР методом мнимых источников изученных СП $N_i^{\langle 0,1\rangle}, N_i^{\langle 1,0\rangle}$ на различных временных интервалах параметры аппроксимирующих функций оказываются отличными друг от друга, что свидетельствует о том, что даже для больших значений окон агрегации Интернет-трафик является нестационарным процессом.

3.7. Анализ результатов исследования дампа Интернет-трафика в разрезе размера пакетов

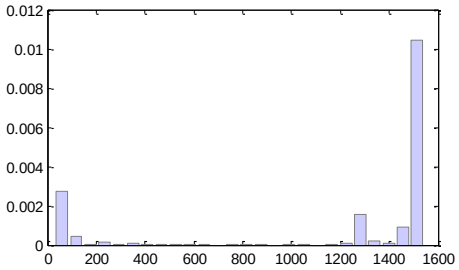
Типичные гистограммы СП $Vp_i^{\langle 1,0\rangle}$, содержащей значения размеров пакетов, переданных 28.10.14 (201410281400.dump) в течении 15 минут, для каждого из выделенных классов пользователей представлены на рис. 3.6.



«Мыши»



«Мулы»



«Слоны»

Рис. 3.6. Гистограммы СП $Vp_i^{\langle\tau\rangle}$ размеров пакетов, переданных в течении 15 минут 28.10.14: а – «Мыши», б – «Мулы», в – «Слоны» (201410281400.dump)

Из рис. 3.6 видно, что размеры пакетов изучаемого дампа варьируются в диапазоне [60; 1514] байт. Полученный результат вполне соответствует стандарту Ethernet II, включенный в 1997 г. в стандарт 802.3x [16], согласно которому на данный момент инкапсулируется подавляющее большинство пакетов Ethernet сетей. Действительно, в соответствие с данным стандартом кадр состоит из трех основных частей: заголовка (header) размером 14 байт (заголовок содержит: MAC-адрес получателя (Destination MAC Address); MAC-адрес отправителя (Source MAC Address); идентификатор протокола третьего уровня (IPv4, IPv6 и др.) (EtherType)); данных (data) – пакет третьего уровня размером от 46 до 1500 байт и циклический избыточный код (CRC), используемый для проверки целостности данных размером 4 байта (рис. 3.7). Таким образом, длина кадра может варьироваться от 64 до 1518 байт. (При этом кадры меньшего размера, как правило, являются результатом непредвиденных сбоев при передаче данных или ошибок в программном и аппаратном обеспечении.) В соответствие с алгоритмом записи данных в архив MAWI [27], последние 4 байта (CRC) каждого пакета считаются неинформативными, а потому обрезаются сетевым оборудованием и не записываются в дамп, поэтому максимальный и минимальный размер кадра в дампе составляют 60 и 1514 байт, соответственно.

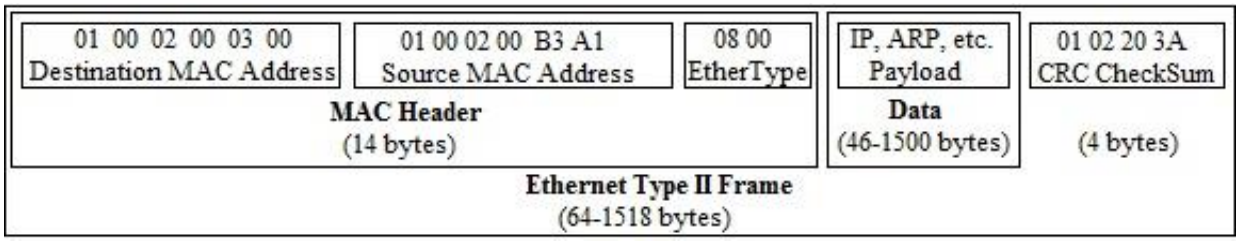


Рис. 3.7. Формат кадра Ethernet II [9]

Также необходимо отметить, что в потоках, создаваемых пользователями, относящимися к классу «Мыши», используются в основном пакеты небольших размеров: 60–200 байт (рис. 3.6а), в то время как в потоках, создаваемых пользователями, относящимися к классам «Мулы» и «Слоны», напротив, – пакеты больших размеров. Данный результат понятен, так как небольшие пакеты используются для инициализации соединения и передачи

служебной информации, а большие пакеты, имеющие максимально допустимый размер ~1500 байт, – для передачи данных.

Далее была проверена стабильность статистических свойств распределений СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$ по размерам пакетов для выделенных классов пользователей. При этом для аппроксимации ПР исследуемых СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$ использовалась аппроксимация Розенблатта-Парзена [36,31,86]. В качестве количественных показателей, характеризующих ПР исследуемых СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$, были использованы квантили, соответствующие уровням доверительной вероятности 0,25;0,5;0,75;0,95.

Результаты вычисления выбранных параметров ПР исследуемых СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$ представлены в табл. 3.10–3.12.

Таблица 3.10

Параметры ПР исследуемых СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$

i	$\bar{M}_{25}$	$\frac{\lvert M_{25,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$	$\bar{M}_{50}$	$\frac{\lvert M_{50,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$	$\bar{M}_{75}$	$\frac{\lvert M_{75,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$	$\bar{M}_{95}$	$\frac{\lvert M_{95,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$
1	81,29	18,14	108,29	29,15	196,99	47,77	1404,64	4,53
2		19,42		32,61		18,66		3,90
3		10,99		18,41		35,96		5,56
4		17,63		27,24		9,07		3,39
5		18,14		30,50		54,41		5,04
6		24,79		42,01		40,07		4,38
7		14,56		23,78		70,33		18,04

Таблица 3.11

Параметры ПР исследуемых СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$

i	$\bar{M}_{25}$	$\frac{\lvert M_{25,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$	$\bar{M}_{50}$	$\frac{\lvert M_{50,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$	$\bar{M}_{75}$	$\frac{\lvert M_{75,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$	$\bar{M}_{95}$	$\frac{\lvert M_{95,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$
1	309,36	40,89	1392,38	0,43	1462,80	0,07	1503,10	0,00
2		6,11		2,21		0,87		0,19
3		53,11		2,70		0,53		0,10
4		23,03		2,63		0,97		0,19
5		23,03		0,92		0,33		0,10
6		124,08		0,75		0,07		0,00
7		9,87		2,39		1,12		0,19

Таблица 3.12

Параметры ПР исследуемых СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$

i	$\bar{M}_{25}$	$\frac{\lvert M_{25,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$	$\bar{M}_{50}$	$\frac{\lvert M_{50,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$	$\bar{M}_{75}$	$\frac{\lvert M_{75,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$	$\bar{M}_{95}$	$\frac{\lvert M_{95,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$
1	666,01	93,60	1387,19	2,49	1459,27	0,81	1502,47	0,14
2		80,71		2,38		1,11		0,24
3		57,48		1,08		0,38		0,06

i	$\bar{M}_{25}$	$\frac{\lvert M_{25,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$	$\bar{M}_{50}$	$\frac{\lvert M_{50,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$	$\bar{M}_{75}$	$\frac{\lvert M_{75,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$	$\bar{M}_{95}$	$\frac{\lvert M_{95,i}-\bar{M}\rvert}{\bar{M}}\cdot 100\%$
4		35,87		0,49		0,11		0,04
5		48,75		3,22		1,61		0,33
6		51,80		4,01		1,58		0,35
7		19,59		3,49		1,68		0,35

(Здесь $\bar{M}$ – среднее по 7 дням (27.10.14 – 02.11.14) значения соответствующего квантиля распределения.)

Из табл. 3.10–3.12 видно, что у изученных случайных последовательностей значения $M_{25}, M_{50}, M_{75}, M_{95}$ отличаются друг от друга. Следовательно, данные случайные последовательности принадлежат различным генеральным совокупностям, отличающимся друг от друга. При этом максимальные вариации параметров значений $M_{25,i}, M_{50,i}, M_{75,i}, M_{95,i}, i=\overline{1,7}$ относительно значений $\bar{M}_{25}, \bar{M}_{50}, \bar{M}_{75}, \bar{M}_{95}$. составили:

1. У «Мышей»: $\Delta M_{25}=24,79\%, \Delta M_{50}=42,01\%, \Delta M_{75}=70,33\%,\ \Delta M_{95}=18,04\%$;
2. У «Мулов»: $\Delta M_{25}=124,08\%, \Delta M_{50}=2,70\%, \Delta M_{75}=1,12\%,\ \Delta M_{95}=0,19\%$;
3. У «Слонов»: $\Delta M_{25}=93,60\%, \Delta M_{50}=4,01\%, \Delta M_{75}=1,68\%,\ \Delta M_{95}=0,35\%$.

Таким образом, можно сделать следующие выводы.

1. У «Мышей» наиболее стабильным параметром оказывается M_{95}. Остальные параметры изменяются в диапазоне достигающим 70%.

2. У «Слонов» и «Мулов» наиболее стабильными параметрами оказываются M_{50}, M_{75}, M_{95}. (Диапазон их изменений не превышает 4%.) Наименее стабильным оказывается параметр M_{25}. (Диапазон его изменения достигает 124%.)

3. Можно считать, что распределения информационных потоков, создаваемых «Мулами» и «Слонами» в течение одной недели близки к стационарным распределениям.

Для проведения исследования стационарности ПР исследуемых СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$ в течение одного дня была использована методика, реализующаяся следующей последовательностью действий.

1. Для каждой СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$ вычислялись аппроксимация Розенблатта-Парзена плотности распределения.

2. Каждая СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$ делилась на 2 части и для каждой из полученных частей выполнялся п.1.

3. Каждая последовательность СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$ делилась на 4 части и для каждой из полученных частей выполнялся п.1.

4. Для каждой ФР, полученной в пп. 1–3 вычислялись квантили $M_{25}, M_{50}, M_{75}, M_{95}$.

В итоге для каждого класса пользователей в каждый из дней наблюдений были вычислены 7 ПР СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$. Примеры функций ПР СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$ представлены на рис. 3.8.

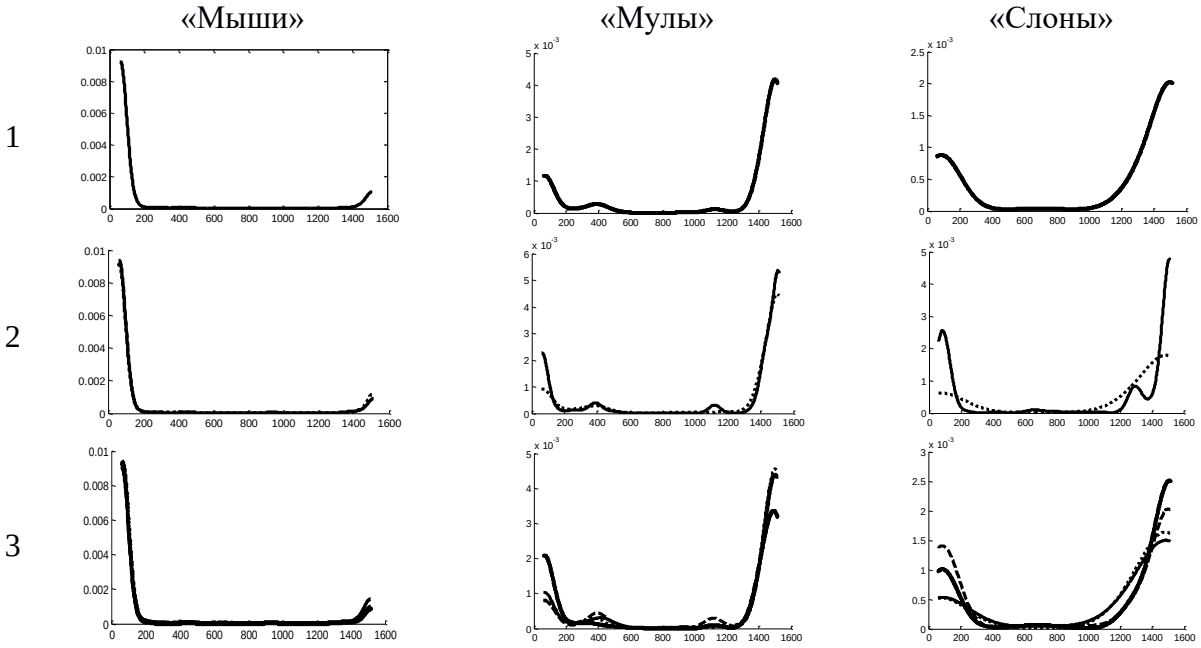


Рис. 3.8. Аппроксимация Розенблатта-Парзена плотностей распределений: 1 – полная выборка; 2 – 0.5 выборки; 3 – 0.25 выборки (27.10.14, файл 201427101400.dump)

Результаты вычисления выбранных параметров распределений СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$, переданных в течение каждого из 7 дней (27.10.14 – 02.11.14), представлены в табл. 3.13–3.15.

Таблица 3.13

Параметры распределений СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$, переданных «Мышами» в течение каждого из 7 дней (27.10.14–02.11.14)

i	$\bar{M}_{25,i}$	k	$\frac{\lvert M_{25,i,k}-\bar{M}_{25,i}\rvert}{\bar{M}_{25,i}}$, %	k	$\frac{\lvert M_{25,i,k}-\bar{M}_{25,i}\rvert}{\bar{M}_{25,i}}$, %	$\bar{M}_{50,i}$	k	$\frac{\lvert M_{50,i,k}-\bar{M}_{50,i}\rvert}{\bar{M}_{50,i}}$, %	k	$\frac{\lvert M_{50,i,k}-\bar{M}_{50,i}\rvert}{\bar{M}_{50,i}}$, %	$\bar{M}_{75,i}$	k	$\frac{\lvert M_{75,i,k}-\bar{M}_{75,i}\rvert}{\bar{M}_{75,i}}$, %	k	$\frac{\lvert M_{75,i,k}-\bar{M}_{75,i}\rvert}{\bar{M}_{75,i}}$, %	$\bar{M}_{95,i}$	k	$\frac{\lvert M_{95,i,k}-\bar{M}_{95,i}\rvert}{\bar{M}_{95,i}}$, %	k	$\frac{\lvert M_{95,i,k}-\bar{M}_{95,i}\rvert}{\bar{M}_{95,i}}$, %
1	66,54	1	50,26	1	50,26	76,72	1	94,76	1	96,65	102,89	1	183,71	1	130,01	1468,20	1	2,08	1	2,48
				2	89,59				2	180,04				2	683,95				2	4,26
		2	43,70	3	126,73		2	79,60	3	236,90		2	113,05	3	325,02		2	1,68	3	8,62
				4	63,37				4	117,50				4	162,51				4	2,28
2	97,08	1	25,46	1	7,49	143,61	1	37,46	1	11,14	233,75	1	39,81	1	24,26	1459,48	1	2,19	1	0,40
				2	4,49				2	7,09				2	5,60				2	0,60
		2	13,48	3	23,96		2	18,22	3	34,42		2	17,42	3	32,97		2	1,89	3	4,18
				4	34,45				4	50,62				4	57,23				4	2,79
3	72,36	1	26,12	1	24,11	88,35	1	47,72	1	42,79	126,16	1	57,63	1	49,56	1482,74	1	1,96	1	1,67
				2	46,22				2	80,64				2	103,73				2	3,53
		2	4,02	3	28,13		2	9,87	3	51,02		2	25,36	3	72,61		2	0,20	3	1,67
				4	16,08				4	31,27				4	68,00				4	0,78
4	95,62	1	9,12	1	3,04	137,79	1	13,72	1	6,33	214,85	1	17,60	1	6,77	1452,21	1	0,40	1	0,80
				2	1,52				2	1,06				2	4,06				2	0,40
		2	15,21	3	16,73		2	24,27	3	26,38		2	31,13	3	36,54		2	1,00	3	0,90
				4	33,45				4	50,65				4	59,55				4	2,60
5	66,54	1	2,19	1	30,59	75,27	1	3,86	1	57,95	89,81	1	8,10	1	90,67	1475,47	1	0,20	1	1,08
				2	32,78				2	61,82				2	97,14				2	2,37
		2	39,33	3	34,96		2	77,27	3	67,61		2	126,28	3	116,57		2	2,37	3	1,58
				4	56,81				4	108,18				4	165,14				4	4,24
6	101,44	1	1,43	1	31,53	153,78	1	1,89	1	44,44	275,92	1	7,90	1	42,68	1343,16	1	1,95	1	0,32
				2	11,47				2	16,07				2	14,76				2	5,85
		2	11,47	3	30,10		2	17,02	3	45,38		2	30,04	3	73,25		2	5,63	3	8,12
				4	14,33				4	20,80				4	28,98				4	3,14
7	69,45	1	29,31	1	50,25	82,54	1	59,90	1	100,41	335,53	1	199,77	1	191,10	1151,23	1	10,10	1	13,39
				2	52,34				2	102,17				2	194,57				2	9,73
		2	18,84	3	508,74		2	33,47	3	852,63		2	45,93	3	242,67		2	22,61	3	25,13
				4	23,03				4	40,52				4	50,27				4	22,73

Таблица 3.14

Параметры СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$, переданных «Мулами» в течение каждого из 7 дней (27.10.14 – 02.11.14)

i	$\bar{M}_{25,i}$	k	$\frac{\lvert M_{25,i,k}-\bar{M}_{25,i}\rvert}{\bar{M}_{25,i}}$, %	k	$\frac{\lvert M_{25,i,k}-\bar{M}_{25,i}\rvert}{\bar{M}_{25,i}}$, %	$\bar{M}'_{50,i}$	k	$\frac{\lvert M_{50,i,k}-\bar{M}_{50,i}\rvert}{\bar{M}_{50,i}}$, %	k	$\frac{\lvert M_{50,i,k}-\bar{M}_{50,i}\rvert}{\bar{M}_{50,i}}$, %	$\bar{M}'_{75,i}$	k	$\frac{\lvert M_{75,i,k}-\bar{M}_{75,i}\rvert}{\bar{M}_{75,i}}$, %	k	$\frac{\lvert M_{75,i,k}-\bar{M}_{75,i}\rvert}{\bar{M}_{75,i}}$, %	$\bar{M}'_{95,i}$	k	$\frac{\lvert M_{95,i,k}-\bar{M}_{95,i}\rvert}{\bar{M}_{95,i}}$, %	k	$\frac{\lvert M_{95,i,k}-\bar{M}_{95,i}\rvert}{\bar{M}_{95,i}}$, %
1	182,86	1	17,49	1	0,80	1398,41	1	0,52	1	6,24	1463,84	1	0,40	1	1,89	1503,10	1	0,10	1	0,39
				2	6,36				2	0,62				2	0,20				2	0,10
		2	30,21	3	5,57		2	1,25	3	0,31		2	0,60	3	0,10		2	0,10	3	0,00
				4	148,69				4	3,22				4	1,49				4	0,29
2	328,26	1	19,49	1	86,37	1423,13	1	1,12	1	4,80	1475,47	1	0,49	1	2,07	1506,00	1	0,10	1	0,39
				2	19,93				2	2,76				2	1,18				2	0,29
		2	3,54	3	12,85		2	0,82	3	3,17		2	0,39	3	1,38		2	0,10	3	0,29
				4	51,82				4	1,12				4	0,59				4	0,10
3	145,06	1	1,00	1	21,05	1354,79	1	3,22	1	2,47	1455,11	1	0,80	1	1,20	1501,64	1	0,19	1	0,19
				2	30,07				2	7,62				2	2,70				2	0,48
		2	9,02	3	191,45		2	8,37	3	41,96		2	1,80	3	20,98		2	0,29	3	4,07
				4	25,06				4	44,97				4	4,90				4	0,87
4	238,12	1	12,82	1	23,81	1428,94	1	1,42	1	2,95	1476,92	1	0,59	1	1,28	1506,00	1	0,10	1	0,29
				2	17,10				2	3,46				2	1,38				2	0,29
		2	48,85	3	20,15		2	0,61	3	3,87		2	0,30	3	1,67		2	0,10	3	0,29
				4	405,46				4	1,22				4	0,69				4	0,10
5	238,12	1	6,11	1	24,43	1379,51	1	7,17	1	6,64	1458,02	1	2,69	1	1,89	1501,64	1	0,48	1	0,29
				2	57,40				2	9,28				2	3,89				2	0,68
		2	79,99	3	78,16		2	1,69	3	1,37		2	0,90	3	0,80		2	0,10	3	0,10
				4	93,43				4	4,53				4	2,09				4	0,39
6	693,22	1	17,83	1	45,93	1402,77	1	0,62	1	2,28	1463,84	1	0,20	1	0,89	1503,10	1	0,00	1	0,19
				2	52,86				2	2,38				2	0,89				2	0,19
		2	19,51	3	15,52		2	0,10	3	3,73		2	0,10	3	1,59		2	0,00	3	0,29
				4	56,21				4	1,24				4	0,60				4	0,10
7	339,90	1	121,49	1	6,42	1359,15	1	2,25	1	2,78	1446,39	1	0,90	1	1,11	1500,19	1	0,10	1	0,19
				2	214,32				2	1,71				2	0,50				2	0,10
		2	14,54	3	100,53		2	2,46	3	0,64		2	1,01	3	0,20		2	0,19	3	0,00
				4	24,38				4	42,15				4	20,51				4	3,97

Таблица 3.15

Параметры СП $Vp_i^{\langle 0,1\rangle}$, $Vp_i^{\langle 1,0\rangle}$, переданных «Слонами» в течение каждого из 7 дней (27.10.14–02.11.14)

i	$\bar{M}_{25,i}$	k	$\frac{\lvert M_{25,i,k}-\bar{M}_{25,i}\rvert}{\bar{M}_{25,i}}$, %	k	$\frac{\lvert M_{25,i,k}-\bar{M}_{25,i}\rvert}{\bar{M}_{25,i}}$, %	$\bar{M}_{50,i}$	k	$\frac{\lvert M_{50,i,k}-\bar{M}_{50,i}\rvert}{\bar{M}_{50,i}}$, %	k	$\frac{\lvert M_{50,i,k}-\bar{M}_{50,i}\rvert}{\bar{M}_{50,i}}$, %	$\bar{M}_{75,i}$	k	$\frac{\lvert M_{75,i,k}-\bar{M}_{75,i}\rvert}{\bar{M}_{75,i}}$, %	k	$\frac{\lvert M_{75,i,k}-\bar{M}_{75,i}\rvert}{\bar{M}_{75,i}}$, %	$\bar{M}_{95,i}$	k	$\frac{\lvert M_{95,i,k}-\bar{M}_{95,i}\rvert}{\bar{M}_{95,i}}$, %	k	$\frac{\lvert M_{95,i,k}-\bar{M}_{95,i}\rvert}{\bar{M}_{95,i}}$, %
1	1289,36	1	1,13	1	6,88	1421,67	1	0,51	1	3,38	1471,11	1	0,30	1	1,48	1504,55	1	0,10	1	0,29
				2	1,01				2	0,00				2	0,00				2	0,00
		2	1,92	3	2,82		2	1,02	3	2,45		2	0,40	3	1,09		2	0,10	3	0,19
				4	8,80				4	2,86				4	1,28				4	0,19
2	1203,57	1	1,57	1	4,11	1420,22	1	2,35	1	5,84	1475,47	1	1,28	1	2,96	1506,00	1	0,29	1	0,58
				2	1,33				2	3,38				2	1,68				2	0,39
		2	22,71	3	32,50		2	2,15	3	6,86		2	0,99	3	3,15		2	0,19	3	0,68
				4	9,42				4	2,76				4	1,38				4	0,29
3	283,19	1	30,29	1	25,67	1372,24	1	0,53	1	1,17	1453,66	1	0,30	1	0,20	1501,64	1	0,10	1	0,10
				2	13,35				2	3,50				2	1,10				2	0,19
		2	232,07	3	39,53		2	0,64	3	1,80		2	0,50	3	0,80		2	0,10	3	0,19
				4	313,71				4	1,06				4	0,80				4	0,19
4	427,14	1	55,15	1	1,02	1394,05	1	0,00	1	43,60	1460,93	1	0,30	1	21,30	1503,10	1	0,10	1	4,16
				2	54,47				2	1,56				2	0,40				2	0,10
		2	190,29	3	199,48		2	0,83	3	0,00		2	0,50	3	0,20		2	0,10	3	0,10
				4	164,08				4	4,07				4	1,89				4	0,39
5	341,35	1	17,46	1	54,52	1431,85	1	1,32	1	4,26	1482,74	1	0,69	1	2,06	1507,46	1	0,19	1	0,39
				2	12,78				2	2,64				2	1,27				2	0,29
		2	19,17	3	69,86		2	3,35	3	5,89		2	1,57	3	2,94		2	0,29	3	0,58
				4	15,33				4	6,40				4	2,94				4	0,58
6	320,99	1	59,34	1	43,49	1331,52	1	0,22	1	3,28	1436,21	1	2,73	1	0,30	1497,28	1	0,58	1	0,00
				2	3,62				2	1,97				2	1,01				2	0,29
		2	200,21	3	127,74		2	0,66	3	3,06		2	0,51	3	1,52		2	0,10	3	0,29
				4	226,03				4	1,42				4	1,01				4	0,19
7	796,45	1	2,92	1	16,25	1338,79	1	0,22	1	0,54	1434,76	1	0,10	1	0,30	1497,28	1	0,00	1	0,10
				2	9,68				2	1,63				2	0,71				2	0,10
		2	2,92	3	45,46		2	1,95	3	0,54		2	0,81	3	0,41		2	0,10	3	0,10
				4	23,55				4	4,13				4	1,82				4	0,29

Из рис. 3.8 и табл. 3.13–3.15 видно, что ФР оцениваемые по всей выборке и его частям, оказываются незначительно отличающимися друг от друга. Таким образом, в рамках одного дня СП $Vp_i^{\langle 0,1 \rangle}$, $Vp_i^{\langle 1,0 \rangle}$, создаваемые каждым из классов пользователей можно считать стационарным.

3.8. Выводы по главе

Результаты исследования ПР информационных потоков, создаваемых тремя классами пользователей, по размерам пакетов сделать следующие выводы:

1. В информационных потоках, создаваемых, классом «Мыши» преобладающими являются пакеты размером не более 200 байт.
2. В информационных потоках, создаваемых, классами «Мулы» и «Слоны» преобладающими являются пакеты размером ~1500 байт.
3. Информационные потоки, создаваемые каждым из классов пользователей, на пятнадцати минутных интервалах имеют стационарные во времени распределения по размеру передаваемых пакетов.
4. Наиболее стабильные во времени параметры распределения информационных потоков по размеру пакетов в течение недели оказываются у класса пользователей «Слоны» и «Мулы», наименее стабильные – у пользователей класса «Мыши».

ГЛАВА 4. ОЦЕНКА АДЕКВАТНОСТИ САМОПОДОБНЫХ МОДЕЛЕЙ ИНТЕРНЕТ-ТРАФИКА ПО ЭКСПЕРИМЕНТАЛЬНЫМ РЕЗУЛЬТАТАМ

Результаты исследований приведённые в предыдущей главе показали, что использованная процедура агрегации СП $N_i^{\{\tau\}}, V_i^{\{\tau\}}$ не является масштабно-инвариантной по параметру σ, а потому статистические свойства изученных СП, оказываются зависящими от размера окна агрегации τ. Данный результат, в свою очередь, позволяет поставить под сомнение правомерность использования для описания изученных СП самоподобных моделей и определяет необходимость анализа значений показателей Херста СП N_i, V_i, результаты которого представлены в следующем разделе.

4.1. Анализ показателей Херста накопленных сумм временных рядов N_i, V_i

Оценка показателя Херста СП $N_i^{\{\tau\}}, V_i^{\{\tau\}}$ осуществлялась в соответствие с алгоритмом HCALC [67], предусматривающим получение оценки значения показателя Херста по зависимости накопленной дисперсии приращений обобщенного броуновского движения от длины приращения. Листинг m-функции, возвращающей значение показателя Херста ВР, вычисленного в соответствие с алгоритмом HCALC, приведен в [89].

В ходе проведенных исследований были вычислены в каждый из дней наблюдений (27.10.14–02.11.14) для каждого класса СП $N_i^{\{\tau\}}, V_i^{\{\tau\}}$ значения показателя Херста по полной последовательности («глобальное» значение показателя Херста) для ранее выбранных значений окон агрегации 0,1 и 1,0 с, а также усредненные по ансамблю, содержащему оценки показателя Херста, вычисленные на 9-ти непересекающихся частях СП (табл. 4.1).

При этом использовались следующие параметры алгоритма HCALC:

– длина приращения $P_{\max} = 20$;

– число отчетов, по которым осуществляется вычисление накопленной дисперсии, $L = 9000$ для полного ВР и $L = 1000$ для частей ВР.

– координата вектора, начиная с которой вычисляется накопленная дисперсия, $N_{start} = 1$ для полного ВР и $N_{start} = \overline{1, 8001}$ с шагом в тысячу для частей ВР.

Также были вычислены средние за весь период наблюдений значения глобальных и усредненных значений показателей Херста СП $N_i^{\{\tau\}}, V_i^{\{\tau\}}$ (табл. 4.2).

Таблица 4.1

Показатели Херста СП $N_i^{\{\tau\}}, V_i^{\{\tau\}}$ в каждый из дней наблюдений (27.10.14–02.11.14, окна агрегации 0,1 и 1 сек)

День	*Класс*	*СП* N_i				*СП* V_i			
		«Глобальный» показатель Херста		*Усредненный показатель Херста*		*«Глобальный» показатель Херста*		*Усредненный показатель Херста*	
		0,1 *с*	1,0 *с*	0,1 *с*	1,0 *с*	0,1 *с*	1,0 *сек*	0,1 *сек*	1,0 *сек*
1	«Мыши»	0,645	0,903	0,636	0,896	0,926	0,980	0,911	0,973
	«Мулы»	0,871	0,956	0,862	0,953	0,852	0,948	0,840	0,944
	«Слоны»	0,923	0,975	0,902	0,967	0,936	0,981	0,915	0,974
2	«Мыши»	0,719	0,933	0,695	0,917	0,957	0,986	0,944	0,978
	«Мулы»	0,893	0,964	0,872	0,957	0,850	0,948	0,834	0,942
	«Слоны»	0,912	0,973	0,899	0,967	0,907	0,974	0,893	0,967
3	«Мыши»	0,732	0,933	0,704	0,918	0,864	0,962	0,835	0,950
	«Мулы»	0,856	0,962	0,848	0,956	0,810	0,945	0,814	0,943
	«Слоны»	0,834	0,949	0,821	0,940	0,825	0,946	0,818	0,939
4	«Мыши»	0,672	0,919	0,661	0,911	0,828	0,949	0,801	0,942
	«Мулы»	0,873	0,966	0,858	0,958	0,856	0,960	0,843	0,953
	«Слоны»	0,883	0,967	0,867	0,961	0,879	0,967	0,864	0,961
5	«Мыши»	0,632	0,904	0,620	0,897	0,824	0,954	0,793	0,941
	«Мулы»	0,883	0,959	0,871	0,954	0,860	0,952	0,839	0,944
	«Слоны»	0,873	0,959	0,863	0,953	0,893	0,968	0,868	0,956
6	«Мыши»	0,766	0,953	0,724	0,935	0,811	0,944	0,775	0,928
	«Мулы»	0,885	0,965	0,864	0,951	0,852	0,956	0,835	0,944
	«Слоны»	0,909	0,976	0,860	0,956	0,917	0,979	0,865	0,958
7	«Мыши»	0,667	0,921	0,646	0,908	0,784	0,938	0,758	0,923
	«Мулы»	0,850	0,960	0,843	0,952	0,823	0,948	0,812	0,940
	«Слоны»	0,874	0,963	0,858	0,953	0,873	0,964	0,862	0,956

Таблица 4.2

«Глобальные» значения показателя Херста за 7 дней (27.10.14–02.11.14) для окон агрегации 0,1 и 1 сек

Класс	*«Глобальное» значение показателя Херста*			
	СП $N_i^{\{\tau\}}$		*СП* $V_i^{\{\tau\}}$	
	0,1 *с*	1,0 *с*	0,1 *с*	1,0 *с*
«Мыши»	0,690±0,049	0,902±0,062	0,856±0,064	0,959±0,181
«Мулы»	0,873±0,016	0,951±0,029	0,843±0,019	0,951±0,005
«Слоны»	0,887±0,031	0,958±0,024	0,890±0,036	0,968±0,012

Из таблиц 4.1 и 4.2 видно, что показатель Херста СП $N_i^{\{\tau\}}, V_i^{\{\tau\}}$ оказывается некоторой случайной величиной, изменяющейся как в течение недели (27.10.14–02.11.14), так и в течение одного дня. При этом его значения оказываются существенно зависящими от размера окна агрегации, что также является свидетельством, подтверждающим обоснованность отклонения гипотезы о самоподобии СП $N_i^{\{\tau\}}, V_i^{\{\tau\}}$.

Напомним, следуя [67], что фрактальным броуновским движением (ФБД) называется гауссовский процесс $X(t)$ с параметром Херста $H(0<H<1)$, обладающий следующими свойствами:

1. $X(0) = 0$ и функция $X(t)$ почти всегда непрерывна.
2. Случайная величина

$$\Delta X = X(t_2) - X(t_1)$$

имеет нормальное гауссовское распределение с нулевым математическим ожиданием и дисперсией $\sigma^2(t_2 - t_1)^{2H}$, где $t_2 > t_1$, σ – положительная константа.

В этой связи для объяснения причин обнаруженных случайных изменений значений показателя Херста были изучены статистических свойства первых разностей СП $N_i^{\{\tau\}}, V_i^{\{\tau\}}$.

4.2. Статистические свойства первых разностей временных рядов N_i, V_i

Всего были изучены статистические свойства первых разностей 4200 частичных последовательностей, выбираемых из СП $N_i^{\{\tau\}}, V_i^{\{\tau\}}$ (3 класса потоков × 2 случайные последовательности (объем информации и кол-во пакетов) × 7 дней (27.10.14–02.11.14) × 100 окон агрегации (от 0,1 до 10 с) = 4200). Для каждой частичной СП были вычислены первые разности и для них проверялась статистическая гипотеза о нормальности их распределений с помощью критерия типа Колмогорова-Смирнова и критерия χ^2. Далее была составлена бинарная таблица, в которой при принятии статистической гипотезы о нормальности закона распределения частичной СП устанавливалось значение равное 1 и 0 в противоположном случае. Затем по ней были вычислены относительные доли данного типа частичных последовательностей, для которых гипотеза о нормальности распределения первых разностей была принята. Зависимости относительной доли СП, у которых первые разности имеют нормальный закон распределения $D_\tau^{[N]}, D_\tau^{[V]}$, от размера окна агрегации представлены на рис. 4.1 и рис. 4.2.

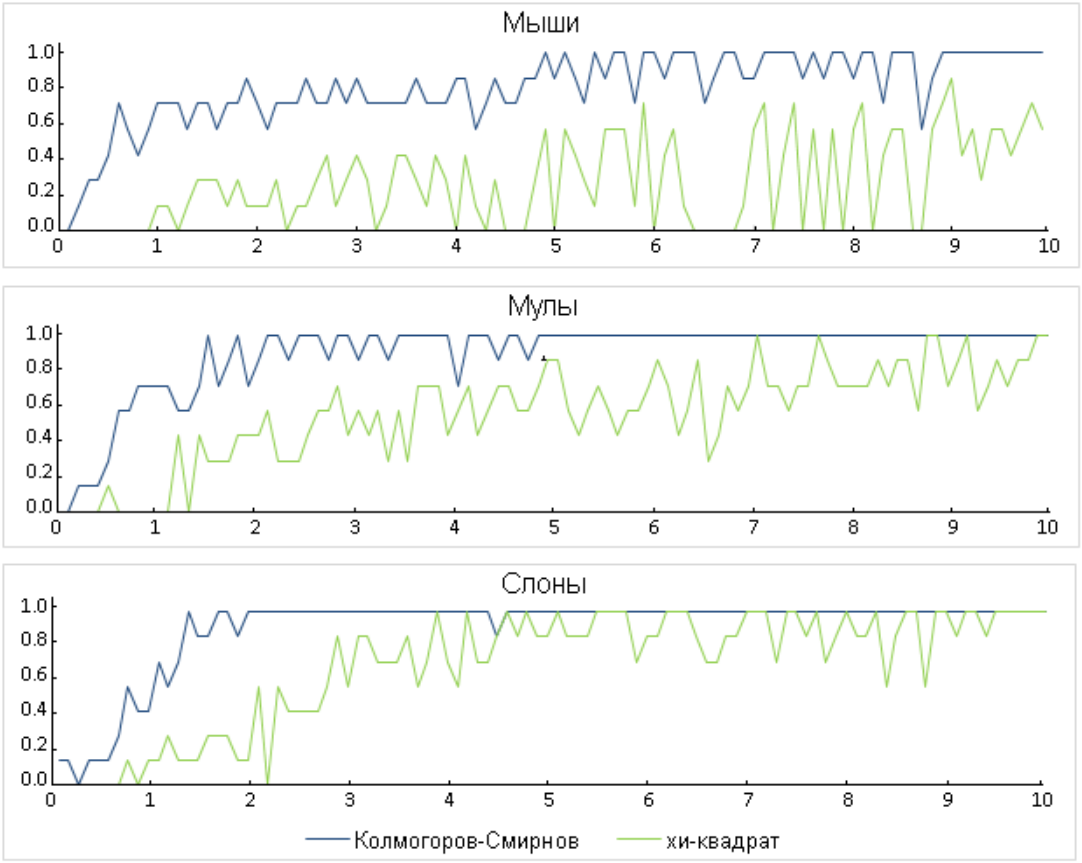


Рис. 4.1. Зависимости относительной доли $D_{\tau}^{[V]}$ СП, у которых первые разности имеют нормальный закон распределения, от размера окна агрегации (0,1–10 с)

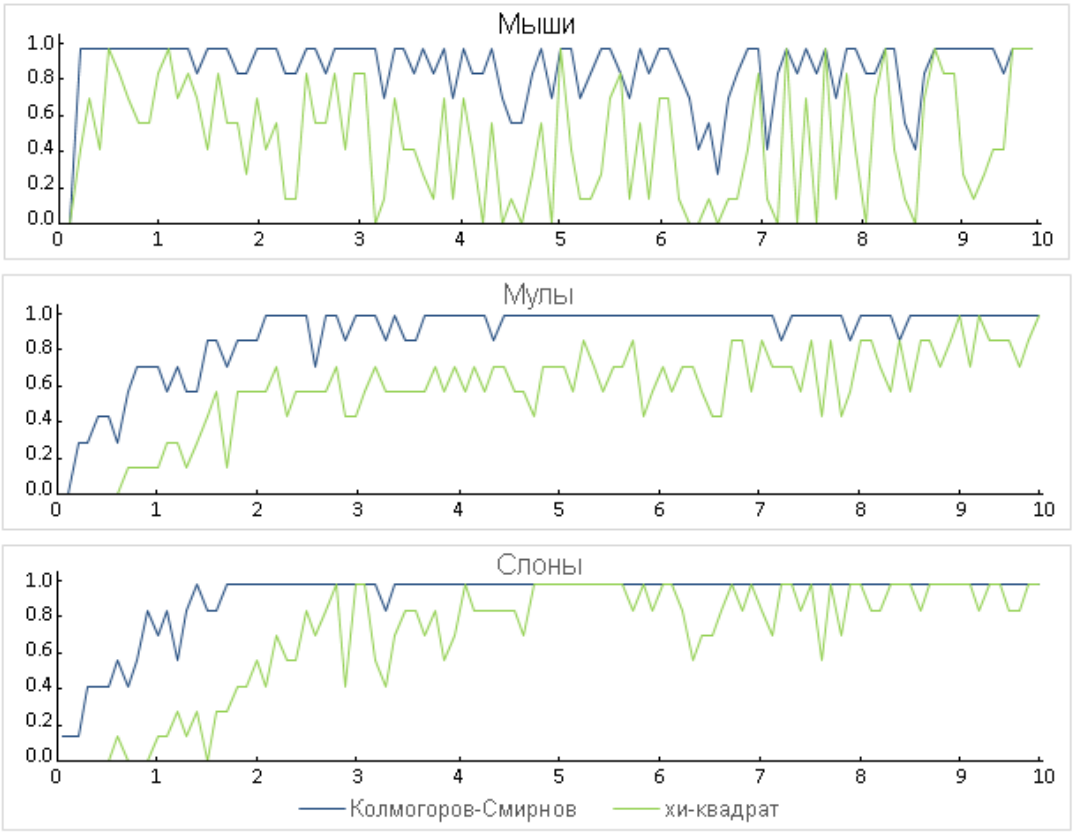


Рис. 4.2. Зависимости относительной доли $D_{\tau}^{[N]}$ СП, у которых первые разности имеют нормальный закон распределения, от размера окна агрегации (0,1–10 с)

Из рис. 4.1 и рис. 4.2 видно, что:

– При применении критерия типа Колмогорова-Смирнова и критерия χ^2 зависимости относительных долей $D_\tau^{[N]}$, $D_\tau^{[V]}$ оказываются отличными для различных окон агрегации.

– Гипотеза о нормальном распределении первых разностей изученных частичных СП отвергается, что подтверждает отсутствие самоподобия у изученных зависимостей.

– Наибольшей вариативностью, безотносительно критерия и окна агрегации, обладают зависимости относительных долей $D_\tau^{[N]}$, $D_\tau^{[V]}$ класса «Мыши». Это согласуется с физическими представлениями о формировании сессий информационного обмена данного класса, которые образованы пакетами небольших размеров, имеют короткий период существования, спонтанно появляются и исчезают. Суммарный объем данных, передаваемый такими сессиями, может значительно отличаться в различные моменты времени.

– Классы «Слоны» и «Мулы» имеют меньшую вариативность зависимости относительных долей $D_\tau^{[N]}$, $D_\tau^{[V]}$для окон агрегации от 5 сек. и выше. С точки зрения физического смысла это можно объяснить особенностями формирования сессий информационного обмена данных классов. Так как такие сессии образованы пакетами максимального размера, имеют продолжительный период существования и к ним применяются различные технологии шейпинга (ограничение скорости со стороны провайдера, в связи с ограниченной пропускной способностью канала), суммарные объемы данных, передаваемые этими потоками в различные моменты времени, отличаются незначительно.

– В соответствии с критерием χ^2, безотносительно окна агрегации, зависимости относительных долей $D_\tau^{[N]}$, $D_\tau^{[V]}$ с нормальным законом распределения оказываются отличным от единицы, что подтверждает неправомерность использования самоподобных моделей.

Проведено исследование особенностей информационных потоков, проходящих через магистральный Интернет-канал, на основе анализа случайных последовательностей, содержащих значения числа пакетов, переданных выделенными типами информационных потоков в течение выбранного временного интервала $\left(N_i^{\langle\tau\rangle}\right)$, а также объемов переданной информации данным типом информационных потоков в течение выбранного временного интервала $\left(V_i^{\{\tau\}}\right)$.

Полученные оценки значений показателя Херста СП $N_i^{\{\tau\}}, V_i^{\{\tau\}}$ и их первых разностей, позволяют сделать обоснованный вывод о несоответствии изученных процессов модели фрактального броуновского движения.

Анализ плотностей распределений и выборочных плотностей СП $N_i^{\{\tau\}}, V_i^{\{\tau\}}$ показал, что для их описания можно использовать модель случайных броуновских блужданий в ограниченной области рассеяния. Напомним, что в рамках данной модели плотность распределения представляется линейной комбинацией функций гауссовского распределения, сдвинутых друг относительно друга. Данный результат позволяет выдвинуть гипотезу о том, что свойствами ФБД могут обладать накопленные суммы СП $N_i^{\{\tau\}}, V_i^{\{\tau\}}$. Для подтверждения справедливости данной гипотезы или ее отклонения необходимо, в свою очередь, проверить гипотезу о возможности генерации фрактального броуновского движения в случае использования случайных величин с ограниченной областью рассеяния.

4.3. Исследование свойств ФБД с ограниченной областью рассеяния

Модель ФБД широко используется для описания временных рядов в различных областях науки, техники и экономики (см., например, [98,74]). Напомним, следуя [67], что ФБД с параметром H, $0 < H < 1$ (показатель Херста), называется гауссовский процесс $X(t)$, если он обладает следующими свойствами:

1. $X(0) = 0$ и функция $X(t)$ почти всегда непрерывна.
2. Свойство гауссовости приращений: случайная величина $\Delta X = X(t_2) - X(t_1)$ имеет НЗР с нулевым математическим ожиданием и дисперсией $\sigma^2(t_2 - t_1)^{2H}$, где $t_2 > t_1$, σ – положительная константа такая, что

$$P(\Delta X < x) = \frac{1}{\sqrt{2\pi}\sigma(t_2 - t_1)^H} \int_{-\infty}^{x} \exp\left(-\frac{1}{2}\left(\frac{u}{\sigma(t_2 - t_1)^H}\right)\right) du,$$

из которого следует закон дисперсии для ФБД:

$$E\left[\left(X(t_2) - X(t_1)\right)^2\right] = \sigma^2 \left|t_2 - t_1\right|^{2H}. \qquad (4.1)$$

Из (4.1) следует известное выражение:

$$E\left[\left(X(t) - X(0)\right)\left(X(t + \Delta t) - X(t)\right)\right] = \frac{1}{2}\sigma^2\left[(t + \Delta t)^{2H} - t^{2H} - \Delta t^{2H}\right], \qquad (4.2)$$

используемое в алгоритмах генерации ФБД.

Из (4.1) видно, что ФБД с параметром $H = 1/2$ совпадает с классическим броуновским движением.

Известен ряд алгоритмов, используемых для генерации ФБД, из которых наиболее известными в настоящее время являются алгоритмы, представленные на рис. 4.3 и 4.4.

4.3.1. Алгоритм срединного смещения

Назначение:

Вычисление ФБД в соответствие с методом половинного деления. Используется рекурсивная процедура DIVIDE1.

Вход:

H – показатель Херста;

MaxLevel – максимальное число рекурсий;

Scale – масштабирующий множитель.

Выход:

X – вектор длины $2^{MaxLevel}+1$, содержащий значения ФБД.

Инициализация:

$N = 2^{MaxLevel}$

$Level = 1$

$i_0 = 1$

$i_2 = N+1$

$X(i) = 0, i = 0,1,\ldots,N+1$

Шаги:

$Ratio = 1/2^H$

$Std = Scale \cdot Ratio$

$X(1) = Scale \cdot randn(0,1)$

$X(N+1) = Scale \cdot randn(0,1)$

($randn(0,1)$ – функция, возвращающая случайное число, сгенерированное в соответствие с нормальным законом распределения с нулевым средним и единичной дисперсией).

$X = DIVIDE1(X, Ratio, Std, i_0, i_1, Level, MaxLevel)$

Процедура *DIVIDE1*

Вход (параметры, передаваемые с верхнего уровня):

X, *Ratio*, *Std*, i_0, i_1 *Level*, *MaxLevel*.

Выход:

X (с измененными значениями).

Шаги:

$i_1 = (i_0 + i_2)/2$

$X(i_1) = 0.5 \cdot (X(i_0) + X(i_2)) + Std \cdot randn(0,1)$

if $Level < MaxLevel$

$StdMin = Std \cdot Ratio$

$X = DIVIDE1(X, Ratio, StdMid, i_0, i_1, Level+1, MaxLevel)$

$X = DIVIDE1(X, Ratio, StdMid, i_1, i_2, Level+1, MaxLevel)$

end if

Рис. 4.3. Описание алгоритма срединного смещения на псевдокоде

4.3.2. Алгоритм генерации ФБД с помощью Фурье-фильтрации (АФФ)

Вход:

H – показатель Херста;

Level – параметр, определяющий длину вектора, содержащего значения ФБД.

Выход:

X – вектор, содержащий значения ФБД.

Инициализация:

$N = 2^{Level}$

$i = \sqrt{-1}$

Шаги:

$\hat{X}(0) = randn(0,1)$

($randn(0,1)$ – функция, возвращающая случайное число, сгенерированное в соответствие с нормальным законом распределения с нулевым средним и единичной дисперсией).

for $j = 1$ to $N/2 - 1$

$\hat{X}(j) = randn(0,1)\, e^{2\pi i \cdot rand(0,1)} / j^{(H+0.5)}$

($rand(0,1)$ – функция, возвращающая случайное число, сгенерированное в соответствие с равномерным законом распределения на интервале [0,1]).

end for

$\hat{X}(N/2) = randn(0,1)\, e^{2\pi i \cdot rand(0,1)} / (N/2)^{(H+0.5)}$

for $j = N/2+1$ to $N - 1$

$\hat{X}(j) = randn(0,1)\, e^{2\pi i \cdot rand(0,1)} / j^{(H+0.5)}$

($rand(0,1)$ – функция, возвращающая случайное число, сгенерированное в соответствие с равномерным законом распределения на интервале [0,1]).

end for

$X = ОДПФ(\hat{X})$

($ОДПФ$ – обратное дискретное преобразование Фурье)

Рис. 4.4. Описание алгоритма генерации ФБД с помощью Фурье фильтрации на псевдокоде

Отметим, что свойства СП, сгенерированных в соответствие с алгоритмом срединного смещения (АСС) (рис. 4.3) для случая H = 1/2, оказываются близкими к свойствам классического броуновского движения. Для случаев $H \neq 1/2$, напротив, СП, сгенерированные в соответствие с данным алгоритмом, не обладают основными свойствами ФБД: дисперсия не удовлетворяет (4.1), а приращения оказываются нестационарными [4]. Однако АСС широко использовался при моделировании естественных ландшафтов, например, горных массивов [12].

Анализ алгоритма генерации ФБД с помощью Фурье-фильтрации (АФФ), псевдокод которого представлен на рис. 4.4, показывает, что он основан на использовании известного свойства спектральной плотности мощности

$$S_x(f) = \frac{1}{T}\left|\int_0^T X(t)e^{-2\pi ft}dt\right|^2$$

ФБД $X(t)$, заданного на интервале $[0;T]$:

$$S_x(f) \infty \frac{1}{f^{2H+1}}.$$

Также известны алгоритмы, основанные на использовании (4.2), в соответствие с которыми во временной области вычисляется детерминированная последовательность Z_k, $k = \overline{0, K-1}$, члены которой удовлетворяют (4.2). Далее вычисляется дискретное преобразование Фурье данной детерминированной последовательности $S_n = \sum_{k=0}^{N-1} Z_k e^{-2\pi ikn/N}$, $i = \sqrt{-1}$. Затем действительная часть каждой спектральной гармоники $\mathrm{Re}(S_n)$ умножается на случайное число вида $randn(0,1) + i \cdot randn(0,1)$, здесь $randn(0,1)$ – функция, возвращающая случайное число, сгенерированное в соответствие с нормальным законом распределения с нулевым средним и единичной дисперсией:

$$\tilde{S}_n = \mathrm{Re}\left(S_n\right) \cdot \left(randn(0,1) + i \cdot randn(0,1)\right).$$

Применение к преобразованному спектру $\tilde{S}_n$ обратного преобразования Фурье позволяет получить значения, так называемого фрактального броуновского шума (ФБШ):

$$z_k = \mathrm{Re}\left(\sum_{n=0}^{N-1} \tilde{S}_n e^{2\pi ikn/N}\right),$$

накопленная сумма $\sum_{k=0}^{m-1} z_k$, $m = \overline{0, N-2}$ которого принимается в качестве реализации ФБД X_m (Листинг функции на языке пакета MATLAB, возвращающей вектор, содержащий вычисленные в соответствие с данным алгоритмом значения ФБД, см. [25].)

Еще один вариант алгоритма генерации ФБД на основе детерминированной последовательности, члены которой удовлетворяют (4.2), предложен в [62]. Здесь каждый из членов детерминированной последовательности умножается случайное число, извлекаемое из выборки со стандартным нормальным распределением, что позволяет, по мнению авторов данного алгоритма, получить ФБД, накопленная сумма которого принимается в качестве ФБД.

Приведенные выше краткие описания алгоритмов генерации ФБД отнюдь не исчерпывают всего перечня известных алгоритмов ФБД. Однако позволяют выделить общий подход, состоящий в использовании случайных величин, генерируемых в соответствие с нормальным законом распределения. При этом возникает закономерный вопрос: возможно ли сгенерировать ФБД с заданным показателем Херста, если использовать случайные числа с законом распределения, отличным от нормального?

Отметим, что данный вопрос, представляет не только академический, но и практический интерес. Например, известны результаты систематических исследований дампов Интернет-трафика [27], передаваемого в высокоскоростном магистральном Интернет-канале [83,85], которые опубликованы в [84]. Результаты дальнейших исследований показали, что распределения СП, содержащих значения накопленных сумм зависимостей количества переданных пакетов и объемов переданной информации в течение выбранного временного интервала в магистральном Интернет-канале от времени, описываются соответствующими законами распределения случайной величины с ограниченной областью рассеяния, а их накопленные суммы имеют свойства, аналогичные свойствам ФБД [81]. Однако для подтверждения правомерности отнесения изученных СП к ФБД, требуется проверка гипотезы о том, что случайные числа с ограниченной областью рассеяния могут порождать ФБД.

4.3.3. Плотность распределения случайных распределений с ограниченной областью рассеяния

Свойства случайных чисел с ограниченной областью рассеяния изучались А. Эйнштейном и М. Смолуховским в работах по теории броуновского движения [103]. Они показали, что математическая модель плотности распределения (ПР) может быть построена как распределение конечного состояния некоторого случайного процесса без последействия с ограниченной областью рассеяния [$x_{\min}$; $x_{\max}$]. ПР данного случайного процесса в системе координат, начало которой находится в середине области рассеяний, относительно границ области рассеяния в точках с координатами $x_{\min,\max} = \mp l$ (рис. 4.5), вычисляется по формуле:

$$f_{LAD}(x;x_0,\sigma,l) = A\left[\phi(x;x_0,\sigma,l) + \sum_{g=0}^{\infty}\phi_{2g+1}^{\pm}(x;x_0,\sigma,l) + \sum_{g=1}^{\infty}\phi_{2g}^{\pm}(x;x_0,\sigma,l)\right], \quad (4.3)$$

где A – нормировочный коэффициент, определяемый из условия:

$$\int_a^b f_{LAD}(\xi;x_0,\sigma,l)d\xi = 1, \quad (4.4)$$

$$\phi(x;x_0,\sigma,l) = \exp\left[-(x-x_0)^2/2\sigma^2\right],$$

$$\phi_{2g+1}^{\pm}(x;x_0,\sigma,l) = \exp\left[-(x-x_{2g+1}^{\pm})^2/2\sigma^2\right],$$

$$\phi^{\pm}_{2g}(x; x_0, \sigma, l) = \exp\left[-(x - x^{\pm}_{2g})^2 / 2\sigma^2\right].$$

Здесь $\phi^{\pm}_{2g+1}, \phi^{\pm}_{2g}$ (см. рис. 4.5) вычисляются по формулам:

$$\phi^{\pm}_{2g} = \pm 4gl + x_0, \quad \phi^{\pm}_{2g+1} = \pm(4g+2)l - x_0, \tag{4.5}$$

где $g = 0,1,\ldots$

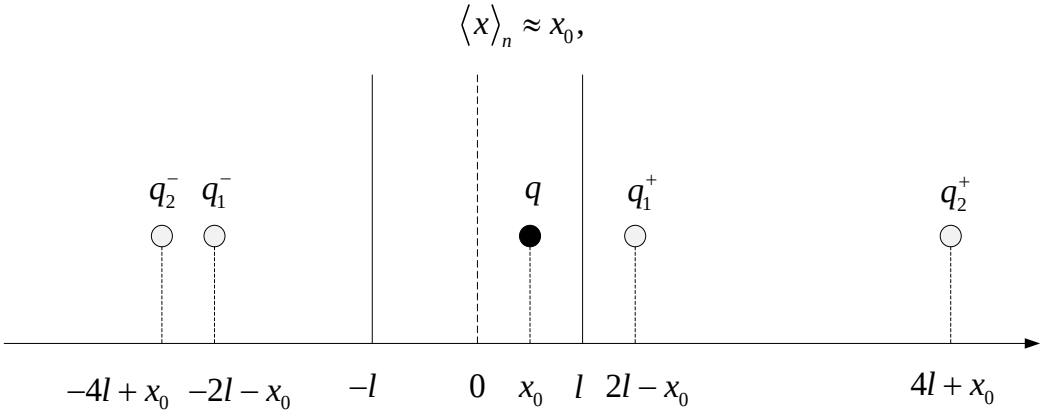


Рис. 4.5. Линейная комбинация функций распределений

Анализ свойств функции $f_{LAD}(x;0,\sigma,l)$, вычисляемой в соответствие с (4.3) показывает, что при $x_0 \equiv 0$ функция $f_{LAD}(x;0,\sigma,l)$ оказывается симметричной относительно прямой $x = 0$, а центр области рассеяния совпадает с математическим ожиданием, медианной и модой; при $l \geq 2.5\sigma$ влияние фиктивных источников уменьшается и рассеяние приближается к нормальному закону; по мере уменьшения размера области рассеяния влияние фиктивных источников возрастает и при $l \leq 0.7\sigma$ распределение выглядит как равномерное. Если положение источника не совпадает с центром области рассеяния, то $f_{LAD}(x;x_0,\sigma,l)$ также порождает семейство распределений с аналогичными свойствами, однако, несимметричными относительно прямой $x = 0$.

4.3.4. Анализ фрактальных свойств случайных последовательностей, генерируемых с использованием случайных чисел с ограниченной областью рассеяния

Для генерации случайных последовательностей был использован алгоритм генерации ФБД на основе Фурье-фильтрации (см. рис. 4.5), в котором вместо случайных чисел с нормальным законом распределения были использованы случайные числа, генерируемые в соответствие с распределением (4.3).

Для оценки показателя Херста были использованы методы, описанные в [39,40,11,20], которые реализованы в пакете MATLAB в виде функции wfbmesti.m, а также метод накопленной дисперсии, программная реализация которого приведена в [89]. (Далее оценки показателя Херста, полученные данными методами, будем обозначать H_1, H_2, H_3, H_4, соответственно).

На первом этапе исследований была проведена оценка точности вычисления показателей Херста, получаемых с помощью каждого из выбранных методов. Здесь для генерации ФБД использовалась функция пакета MATLAB wfbm.m, представляющая собой программную реализацию алгоритма, основанного на использовании биортогональных вейвлетов [1,14] (далее Метод № 1), а также разработанная авторами в пакете MATLAB функция, реализующая АФФ (далее Метод № 2).

Средние значения показателей Херста, вычисленные по ансамблю оценок показателей Херста для 100 независимых реализаций ФБД (длина реализации – 2^{15} значений), сгенерированных с помощью Методов № 1, № 2 представлены в таблицах 4.3 и 4.4, соответственно. Отметим, что использование двух различных алгоритмов генерации ФБД обусловлено тем, что, фактически, сравнивается не столько точность оценки показателя Херста реализаций ФБД сколько точность оценки результата последовательного использования алгоритма генерации ФБД и алгоритма оценки показателя Херста.

Таблица 4.3

Оценки значений показателей Херста ФБД, генерируемого с помощью Метода № 1

H	H_1	H_2	H_3	H_4
0.1	0.171±0.007	0.171±0.006	–0,027±0,013	0,157±0,008
0.2	0,240±0,008	0,240±0,007	0,099±0,012	0,229±0,009
0.3	0.321±0.009	0.320±0.009	0.219±0.011	0.313±0,008
0.4	0.408±0.008	0.408±0.008	0.335±0.012	0.403±0.010
0.5	0.499±0.008	0.499±0.008	0.450±0.013	0.499±0.009
0.6	0.596±0.006	0.596±0.007	0.564±0.012	0.598±0.010
0.7	0.696±0.007	0.696±0.008	0.671±0.014	0.697±0.011
0.8	0.794±0.006	0.795±0.008	0.774±0.014	0.791±0.011
0.9	0.894±0.006	0.894±0.008	0.864±0.015	0.872±0.013
0.95	0.945±0.006	0.945±0.010	0.902±0.014	0.912±0.012
0.99	0.984±0.007	0.984±0.010	0.925±0.014	0.931±0.012

Таблица 4.4

Оценки значений показателей Херста ФБД, генерируемого с помощью Метода № 2

H	H_1	H_2	H_3	H_4
0.1	0.101±0.009	0.100±0.007	–0.105±0.012	0.099±0.007
0.2	0.200±0.007	0.200±0.007	0.072±0.012	0.199±0.009
0.3	0.300±0.008	0.300±0.008	0.208±0.013	0.298±0.009
0.4	0.401±0.007	0.402±0.007	0.334±0.014	0.399±0.010
0.5	0.502±0.007	0.503±0.008	0.449±0.013	0.500±0.010
0.6	0.599±0.007	0.598±0.008	0.565±0.012	0.600±0.009
0.7	0.701±0.007	0.701±0.007	0.674±0.012	0.698±0.011
0.8	0.800±0.007	0.799±0.008	0.782±0.017	0.790±0.012
0.9	0.900±0.006	0.900±0.009	0.894±0.035	0.871±0.014
0.95	0.950±0.006	0.949±0.009	0.944±0.039	0.902±0.011
0.99	0.991±0.005	0.991±0.008	0.979±0.029	0.928±0.013

Из табл. 4.3–4.4 видно:

1. В целом, оценки показателя Херста реализаций ФБД H_1, H_2, H_4 для обоих методов генерации ФБД согласуются друг с другом.

2. Оценка значения показателя Херста реализаций ФБД H_3 оказывается неудовлетворительной для $H \in [0.1; 0.5]$ для обоих методов генерации ФБД.

3. Оценки значений показателя Херста реализаций ФБД, генерируемых в соответствие с методом № 1, для заданного значения показателя Херста $H \in [0.1; 0.3]$, составляют 70%, 20% и 6% при H = 0.1, 0.2, 0.3, соответственно.

4. Отличие оценок показателей H_1, H_2, H_4 Херста ФБД, генерируемого с помощью метода № 2, от H оказывается существенно меньше, что позволяет сделать вывод о том, что качество реализаций ФБД, генерируемых для $H \in [0.1; 0.3]$ в соответствие с методом № 2, оказывается выше, чем при использовании метода № 1.

5. Значения показателя Херста, оцениваемые с помощью метода накопленной дисперсии (H_4), при $H \in [0.8; 1]$ оказываются ниже заданного значения показателя Херста H. Обнаруженное отличие увеличивается по мере приближения значения заданного значения показателя Херста H к единице.

Средние значения показателей Херста, вычисленные по ансамблю оценок показателей Херста для 100 независимых реализаций ФБД, сгенерированных с помощью Метода № 2, в котором использовались случайные числа с ограниченной областью рассеяния, представлены в табл. 4.5–4.7, соответственно. ПР использованных случайных чисел представлены на рис. 4.6

Таблица 4.5

Оценки значений показателей Херста ФБД, генерируемого в соответствие с методом № 2 и использованием случайных чисел с ограниченной областью рассеяния (ПР – зависимость № 1 на рис. 4.6)

H	H_1	H_2	H_3	H_4
0.1	0.099±0.009	0.100±0.007	–0.102±0.012	0.107±0.008
0.2	0.199 ±0.008	0.199 ±0.007	0.073 ±0.011	0.203 ±0.008
0.3	0.300 ±0.008	0.300 ±0.007	0.210 ±0.011	0.301 ±0.009
0.4	0.399±0.007	0.399±0.007	0.334±0.012	0.399±0.010
0.5	0.500±0.008	0.500±0.009	0.451±0.013	0.498±0.009
0.6	0.599 ±0.007	0.599 ±0.009	0.566 ±0.014	0.600 ±0.010
0.7	0.700 ±0.008	0.700 ±0.009	0.673 ±0.014	0.696 ±0.010
0.8	0.800 ±0.006	0.800 ±0.008	0.786 ±0.018	0.790 ±0.012
0.9	0.900 ±0.006	0.900 ±0.008	0.893 ±0.031	0.870 ±0.015
0.95	0.951 ±0.007	0.951 ±0.008	0.942 ±0.038	0.903 ±0.014
0.99	0.990 ±0.006	0.989 ±0.009	0.985 ±0.024	0.924 ±0.014

Таблица 4.6

Оценки значений показателей Херста ФБД, генерируемого в соответствие с методом № 2 и использованием случайных чисел с ограниченной областью рассеяния (ПР – зависимость № 2 на рис. 4.6)

H	H_1	H_2	H_3	H_4
0.1	0.100 ±0.009	0.099 ±0.007	–0.101 ±0.012	0.187 ±0.006
0.2	0.201 ±0.008	0.200 ±0.007	0.073 ±0.012	0.251 ±0.007
0.3	0.301±0.007	0.300±0.007	0.210±0.012	0.324±0.009
0.4	0.401±0.008	0.401±0.008	0.333±0.012	0.405±0.010
0.5	0.500±0.008	0.499±0.008	0.452±0.014	0.498±0.009
0.6	0.600±0.007	0.600±0.008	0.564±0.012	0.601±0.008
0.7	0.702±0.007	0.701±0.008	0.679±0.014	0.699±0.011
0.8	0.803±0.007	0.800±0.009	0.793±0.025	0.787±0.012
0.9	0.905±0.006	0.900±0.008	0.910±0.039	0.854±0.012
0.95	0.957±0.005	0.950±0.008	0.954±0.037	0.884±0.012
0.99	0.999±0.006	0.991±0.009	0.982±0.026	0.903±0.010

Таблица 4.7

Оценки значений показателей Херста ФБД, генерируемого в соответствие с методом № 2 и использованием случайных чисел с ограниченной областью рассеяния (ПР – зависимость № 3 на рис. 4.6)

H	H_1	H_2	H_3	H_4
0.1	0.100±0.007	0.100±0.006	–0.104±0.012	0.099±0.007
0.2	0.200±0.007	0.200±0.007	0.071±0.011	0.200±0.008
0.3	0.301±0.008	0.300±0.007	0.210±0.014	0.299±0.009
0.4	0.399±0.007	0.399±0.007	0.334±0.012	0.399±0.008
0.5	0.500±0.007	0.500±0.007	0.451±0.013	0.500±0.009
0.6	0.600±0.007	0.599±0.007	0.563±0.013	0.598±0.009
0.7	0.699±0.007	0.699±0.008	0.674±0.013	0.697±0.012
0.8	0.801±0.006	0.801±0.008	0.782±0.016	0.791±0.013
0.9	0.900±0.007	0.900±0.009	0.893±0.031	0.873±0.014
0.95	0.950±0.006	0.949±0.008	0.949±0.035	0.906±0.014
0.99	0.990±0.006	0.990±0.0010	0.988±0.023	0.929±0.012

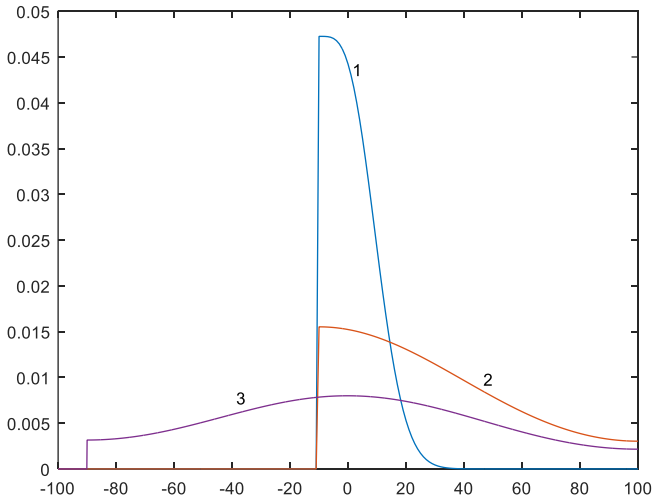


Рис. 4.6. Плотности случайных распределений, использовавшиеся для генерации случайных чисел реализацию ФБД в соответствие с алгоритмом Фурье-фильтрации

Из таблиц 4.5–4.7 видно, что свойства показателя Херста изученных случайных последовательностей, сгенерированных в соответствие с АФФ, в котором использовались случайные числа, генерируемые соответствие с плотностями распределений, представленных на рис. 4.6, оказываются аналогичными ранее изученным реализациям ФБД (см. табл. 4.3–4.4). Таким образом, СП, у которой ПР первых разностей описывается функцией, аналогичной закону распределения случайной величины с ограниченной областью рассеяния (4.1), может обладать свойством самоподобия.

Проведенное исследование свойств реализаций, генерируемых в соответствие с алгоритмом Фурье-фильтрации, позволило сделать вывод о том, что невыполнение требования определения [3] о НЗР первых разностей ФБД не является достаточным условием для отклонения гипотезы о том, что изучаемая СП является реализацией ФБД.

Данный результат подтверждает гипотезу, сформулированную в [81], о том, что накопленные суммы случайных последовательностей, содержащих значения зависимостей количества переданных пакетов, а также значений объемов переданной информации в течение выбранного временного интервала в магистральном Интернет-канале от времени, обладают свойствами, аналогичными свойствам ФБД.

Таким образом, разработанная методика анализа информационных потоков дампов Интерент-трафика в магистральном Интернет-канале позволила обнаружить следующие свойства сетевого трафика:

1. Размер потока каждого из классов вне зависимости от размера окна агрегации являются СП с ограниченными областями рассеяния. Число пакетов каждого из классов вне зависимости от размера окна агрегации являются случайными последовательностями с ограниченными областями рассеяния.

2. Оценки параметров распределения не являются стационарными величинами, но могут меняться по дням, так и в течение одного дня. Наиболее стабильные параметры ПР, как в течение одного дня, так и недели, оказываются у класса потоков «Мыши». Наименее стабильные параметры распределений, как в течение одного дня, так и в течение недели, имеют потоки классов «Мулы» и «Слоны», у которых отклонения параметра μ от среднего значения в течение недели могут достигать 60% и 40%, соответственно.

3. При использовании окна агрегации 0,1 с. аппроксимации ПР, построенные для частей ВР (N_i, V_i) как с помощью метода мнимых источников, так и с помощью аппроксимации Розенблатта-Парзена, изученных СП оказываются отличными друг от друга, что свидетельствует о нестационарности данных СП. При использовании окна

агрегации 1,0 с. для аппроксимации ПР изученных ВР методом мнимых источников на различных временных интервалах параметры аппроксимирующих функций оказываются отличными друг от друга, что свидетельствует о том, что даже для больших значений окон агрегации интернет-трафик является нестационарным процессом.

4. Информационные потоки, создаваемые каждым из классов пользователей, на пятнадцати минутных интервалах имеют стационарные во времени распределения по размеру передаваемых пакетов. Наиболее стабильные во времени параметры распределения информационных потоков по размеру пакетов в течение недели оказываются у класса пользователей «Слоны» и «Мулы», наименее стабильные – у пользователей класса «Мыши».

5. Полученные оценки значений показателя Херста случайных последовательностей $N_i^{\{\tau\}}, V_i^{\{\tau\}}$ и их первых разностей, позволяют сделать обоснованный вывод о несоответствии изученных процессов модели фрактального броуновского движения. Анализ плотностей распределений и выборочных плотностей случайных последовательностей $N_i^{\{\tau\}}, V_i^{\{\tau\}}$ показал, что для их описания можно использовать модель случайных броуновских блужданий в ограниченной области рассеяния.

6. СП, у которой плотность распределения первых разностей описывается функцией, аналогичной закону распределения случайной величины с ограниченной областью рассеяния (4.1), может обладать свойством самоподобия. Накопленные суммы случайных последовательностей, содержащих значения зависимостей количества переданных пакетов, а также значений объемов переданной информации в течение выбранного временного интервала в магистральном Интернет-канале от времени, обладают свойствами, аналогичными свойствам ФБД.

4.3.5. Исследование взаимного влияния информационных потоков в магистральном Интернет-канале, создаваемых различными группами пользователей друг на друга

Для выявления возможных взаимосвязей информационных потоков, передаваемых в магистральном Интернет-канале, были использованы показатели Херста накопленных сумм СП N_i, V_i и $\tilde{V}_i$ – СП, состоящей из отношений соответствующих членов СП N_i, V_i. Физический смысл СП $\tilde{V}_i$ – средний объем переданной информации, переданной одним пакетом.

В ходе проведенных исследований проверялась гипотеза о том, что взаимосвязь информационных потоков, передаваемых в магистральном Интернет-канале, проявляется в

наличии зависимостей показателей Херста каждого из классов пользователей как от интегральных характеристик (вычисляемых по всему пятнадцатиминутному дампу Интернет-трафика информационного потока, создаваемого данным классом пользователей: объема переданной информации V, числа переданных пакетов N, объема переданной информации одним пакетом $\tilde{V}$, так и информационных характеристик потоков, создаваемых другими классами пользователей, то есть:

$$
\begin{aligned}
H_V^{\{Mice\}} &= F_{V1}\left(V^{\{Mice\}}, V^{\{Mules\}}, V^{\{Elefants\}}\right),\\
H_V^{\{Mules\}} &= F_{V2}\left(V^{\{Mice\}}, V^{\{Mules\}}, V^{\{Elefants\}}\right),\\
H_V^{\{Elefants\}} &= F_{V3}\left(V^{\{Mice\}}, V^{\{Mules\}}, V^{\{Elefants\}}\right),\\
H_N^{\{Mice\}} &= F_{N1}\left(N^{\{Mice\}}, N^{\{Mules\}}, N^{\{Elefants\}}\right),\\
H_N^{\{Mules\}} &= F_{N2}\left(N^{\{Mice\}}, N^{\{Mules\}}, N^{\{Elefants\}}\right),\\
H_N^{\{Elefants\}} &= F_{N3}\left(N^{\{Mice\}}, N^{\{Mules\}}, N^{\{Elefants\}}\right),\\
H_V^{\{Mice\}} &= F_{V4}\left(\tilde{V}^{\{Mice\}}, \tilde{V}^{\{Mules\}}, \tilde{V}^{\{Elefants\}}\right),\\
H_V^{\{Mules\}} &= F_{V5}\left(\tilde{V}^{\{Mice\}}, \tilde{V}^{\{Mules\}}, \tilde{V}^{\{Elefants\}}\right),\\
H_V^{\{Elefants\}} &= F_{V6}\left(\tilde{V}^{\{Mice\}}, \tilde{V}^{\{Mules\}}, \tilde{V}^{\{Elefants\}}\right),\\
H_N^{\{Mice\}} &= F_{N4}\left(\tilde{V}^{\{Mice\}}, \tilde{V}^{\{Mules\}}, \tilde{V}^{\{Elefants\}}\right),\\
H_N^{\{Mules\}} &= F_{N5}\left(\tilde{V}^{\{Mice\}}, \tilde{V}^{\{Mules\}}, \tilde{V}^{\{Elefants\}}\right),\\
H_N^{\{Elefants\}} &= F_{N6}\left(\tilde{V}^{\{Mice\}}, \tilde{V}^{\{Mules\}}, \tilde{V}^{\{Elefants\}}\right),
\end{aligned}
\qquad (3)
$$

Исходные данные, использовавшиеся в проведенном нами исследовании, представлены в таблицах 4.8–4.10.

Таблица 4.8

Исходные данные для идентификации функций F_{V1}, F_{V2}, F_{V3}

День	Класс пользователей	Объем переданной информации	$H_V^{\{0.1\}}$	$H_V^{\{1.0\}}$	$\bar{H}_V^{\{0.1\}}$	$\bar{H}_V^{\{1.0\}}$
1	Мыши	30996017241	0,926	0,98	0,911	0,973
	Мулы	8118004515	0,852	0,948	0,840	0,944
	Слоны	37638020936	0,936	0,981	0,915	0,974
2	Мыши	28494531094	0,957	0,986	0,944	0,978
	Мулы	7462853381	0,8500	0,948	0,834	0,942
	Слоны	34600502042	0,907	0,974	0,893	0,967
3	Мыши	22566213503	0,864	0,962	0,835	0,950
	Мулы	6769864050	0,810	0,945	0,814	0,943
	Слоны	29336077554	0,825	0,946	0,818	0,939
4	Мыши	26623867248	0,828	0,949	0,801	0,942
	Мулы	7987160174	0,856	0,960	0,843	0,953
	Слоны	34611027422	0,879	0,967	0,864	0,961

День	Класс пользователей	Объем переданной информации	$H_V^{\{0.1\}}$	$H_V^{\{1.0\}}$	$\bar{H}_V^{\{0.1\}}$	$\bar{H}_V^{\{1.0\}}$
5	Мыши	24346132026	0,824	0,954	0,793	0,941
	Мулы	6376367911	0,860	0,952	0,839	0,944
	Слоны	29563160317	0,893	0,968	0,868	0,956
6	Мыши	18022343530	0,811	0,944	0,775	0,928
	Мулы	4095987165	0,852	0,956	0,835	0,944
	Слоны	20479935829	0,917	0,979	0,865	0,958
7	Мыши	18424280953	0,784	0,938	0,758	0,923
	Мулы	4187336580	0,823	0,948	0,812	0,940
	Слоны	20936682901	0,873	0,964	0,862	0,956

Таблица 4.9

Исходные данные для идентификации функций F_{N1}, F_{N2}, F_{N3}

День	Класс пользователей	Число переданных пакетов	$H_N^{\{0.1\}}$	$H_N^{\{1.0\}}$	$\bar{H}_N^{\{0.1\}}$	$\bar{H}_N^{\{1.0\}}$
1	Мыши	100880771	0,645	0,903	0,636	0,896
	Мулы	7964271	0,871	0,956	0,862	0,953
	Слоны	21238057	0,923	0,975	0,902	0,967
2	Мыши	97548380	0,719	0,933	0,695	0,917
	Мулы	10545770	0,893	0,964	0,872	0,957
	Слоны	23727984	0,912	0,973	0,899	0,967
3	Мыши	94385082	0,732	0,933	0,704	0,918
	Мулы	7451453	0,856	0,962	0,848	0,956
	Слоны	22354361	0,834	0,949	0,821	0,940
4	Мыши	98676762	0,672	0,919	0,661	0,911
	Мулы	7790270	0,873	0,966	0,858	0,958
	Слоны	23370812	0,883	0,967	0,867	0,961
5	Мыши	89766552	0,632	0,904	0,62	0,897
	Мулы	9974061	0,883	0,959	0,871	0,954
	Слоны	24935153	0,873	0,959	0,863	0,953
6	Мыши	87350886	0,766	0,953	0,724	0,935
	Мулы	4367544	0,885	0,965	0,864	0,951
	Слоны	17470177	0,909	0,976	0,86	0,956
7	Мыши	81001426	0,667	0,921	0,646	0,908
	Мулы	4050071	0,850	0,960	0,843	0,952
	Слоны	16200285	0,874	0,963	0,858	0,953

Таблица 4.10

Исходные данные для идентификации функций F_{V4}, F_{V5}, F_{V6}

День	Класс пользователей	Объем информации, приходящейся на один пакет	$H_V^{\{0.1\}}$	$H_V^{\{1.0\}}$	$\bar{H}_V^{\{0.1\}}$	$\bar{H}_V^{\{1.0\}}$
1	Мыши	307253967	0,926	0,980	0,911	0,973
	Мулы	1019302844	0,852	0,948	0,840	0,944
	Слоны	1772196991	0,936	0,981	0,915	0,974
2	Мыши	292106654	0,957	0,986	0,944	0,978
	Мулы	707663144	0,850	0,948	0,834	0,942
	Слоны	1458214964	0,907	0,974	0,893	0,967
3	Мыши	239086653	0,864	0,962	0,835	0,95
	Мулы	908529283	0,810	0,945	0,814	0,943
	Слоны	1312320075	0,825	0,946	0,818	0,939
4	Мыши	269808883	0,828	0,949	0,801	0,942
	Мулы	1025273757	0,856	0,960	0,843	0,953
	Слоны	1480950983	0,879	0,967	0,864	0,961

День	Класс пользователей	Объем информации, приходящейся на один пакет	$H_V^{\{0.1\}}$	$H_V^{\{1.0\}}$	$\bar{H}_V^{\{0.1\}}$	$\bar{H}_V^{\{1.0\}}$
5	Мыши	271216073	0,824	0,954	0,793	0,941
	Мулы	639295030	0,860	0,952	0,839	0,944
	Слоны	1185601692	0,893	0,968	0,868	0,956
6	Мыши	206321243	0,811	0,944	0,775	0,928
	Мулы	937823835	0,852	0,956	0,835	0,944
	Слоны	1172279794	0,917	0,979	0,865	0,958
7	Мыши	227456253	0,784	0,938	0,758	0,923
	Мулы	1033892060	0,823	0,948	0,812	0,940
	Слоны	1292365075	0,873	0,964	0,862	0,956

Таблица 4.11

Исходные данные для идентификации функций F_{N4}, F_{N5}, F_{N6}

День	Класс пользователей	Объем информации, приходящейся на один пакет данного класса	$H_N^{\{0.1\}}$	$H_N^{\{1.0\}}$	$\bar{H}_N^{\{0.1\}}$	$\bar{H}_N^{\{1.0\}}$
1	Мыши	100880771	0,645	0,903	0,636	0,896
	Мулы	7964271	0,871	0,956	0,862	0,953
	Слоны	21238057	0,923	0,975	0,902	0,967
2	Мыши	97548380	0,719	0,933	0,695	0,917
	Мулы	10545770	0,893	0,964	0,872	0,957
	Слоны	23727984	0,912	0,973	0,899	0,967
3	Мыши	94385082	0,732	0,933	0,704	0,918
	Мулы	7451453	0,856	0,962	0,848	0,956
	Слоны	22354361	0,834	0,949	0,821	0,940
4	Мыши	98676762	0,672	0,919	0,661	0,911
	Мулы	7790270	0,873	0,966	0,858	0,958
	Слоны	23370812	0,883	0,967	0,867	0,961
5	Мыши	89766552	0,632	0,904	0,620	0,897
	Мулы	9974061	0,883	0,959	0,871	0,954
	Слоны	24935153	0,873	0,959	0,863	0,953
6	Мыши	87350886	0,766	0,953	0,724	0,935
	Мулы	4367544	0,885	0,965	0,864	0,951
	Слоны	17470177	0,909	0,976	0,86	0,956
7	Мыши	81001426	0,667	0,921	0,646	0,908
	Мулы	4050071	0,850	0,960	0,843	0,952
	Слоны	16200285	0,874	0,963	0,858	0,953

На первом этапе были построены многомерные линейные модели зависимостей (3) вида

$$z = a_1 x + a_2 y + a_3 z,$$

коэффициенты которых при соответствующих переменных есть корреляции между показателям Херста накопленных сумм СП N_i, V_i, представленных в таблице 5, и количеством переданных пакетов (СП N_i), объемами переданных данных (СП V_i), средними объемами

переданных данных на пакет (СП $\tilde{V}_i$). Выбор данного вида модели обусловлен очевидными физическими соображения, состоящими в том, что значение показателя Херста при $x = y = z \equiv 0,$ то есть отсутствии информационных потоков, должно равняться 0. Коэффициенты моделей представлены в таблицах 4.12–4.17.

Таблица 4.12

Параметры линейных регрессионных моделей, описывающих зависимости между значениями показателей Херста накопленных сумм СП V_i и объемами переданной информации каждым из рассматриваемых классов пользователей[2]

Модель	R^2	R^2_{adj}	$a_{V^{\{Mise\}}}$	$S_V^{\{Mise\}}$	$a_{V^{\{Mule\}}}$	$S_V^{\{Mules\}}$
$H_V^{\{0.1\}\{Mise\}}$	0,986	0,980	5,995	2,076	–9,509	7,731
$H_V^{\{1.0\}\{Mise\}}$	0,976	0,967	7,085	2,994	–12,148	11,151
$\bar{H}_V^{\{0.1\}\{Mise\}}$	0,988	0,983	5,797	1,888	–9,128	7,031
$\bar{H}_V^{\{1.0\}\{Mise\}}$	0,977	0,968	6,912	2,902	–11,657	10,810
$H_V^{\{0.1\}\{Mules\}}$	0,975	0,965	6,454	2,701	–11,534	10,060
$H_V^{\{1.0\}\{Mules\}}$	0,972	0,960	7,123	3,257	–12,449	12,132
$\bar{H}_V^{\{0.1\}\{Mules\}}$	0,975	0,964	6,191	2,700	–10,735	10,057
$\tilde{H}_V^{\{1.0\}\{Mules\}}$	0,972	0,961	6,990	3,214	–12,049	11,972
$H_V^{\{0.1\}\{Elefant\}}$	0,979	0,971	7,531	2,606	–14,849	9,707
$H_V^{\{1.0\}\{Elefant\}}$	0,974	0,964	7,563	3,172	–13,826	11,815
$\bar{H}_V^{\{0.1\}\{Elefant\}}$	0,980	0,973	7,049	2,480	–13,341	9,237
$\bar{H}_V^{\{1.0\}\{Elefant\}}$	0,975	0,965	7,366	3,099	–13,224	11,543

Таблица 4.13

Параметры линейных регрессионных моделей, описывающих зависимости между значениями показателей Херста накопленных сумм СП N_i и объемами переданной информации каждым из рассматриваемых классов пользователей

Модель	R^2	R^2_{adj}	$a_{V^{\{Mise\}}}$	$S_V^{\{Mise\}}$	$a_{V^{\{Mule\}}}$	$S_V^{\{Mules\}}$
$H_N^{\{0.1\}\{Mise\}}$	0,958	0,941	5,307	2,888	–9,602	10,758
$H_N^{\{1.0\}\{Mise\}}$	0,969	0,956	7,071	3,335	–12,687	12,423
$\bar{H}_N^{\{0.1\}\{Mise\}}$	0,962	0,947	4,980	2,655	–8,670	9,889
$\bar{H}_N^{\{1.0\}\{Mise\}}$	0,969	0,957	6,923	3,250	–12,304	12,107

[2] Здесь и далее в таблица 13–17: R^2 – коэффициент детерминации модели; R^2_{adj} – скорректированный на объем выборки коэффициент детерминации; *a* – оценки коэффициентов линейной регрессии; S – стандартные ошибки коэффициентов линейной регрессии.

Модель	R^2	R^2_{adj}	$a_{V^{\{Mise\}}}$	$S_V^{\{Mise\}}$	$a_{V^{\{Mule\}}}$	$S_V^{\{Mules\}}$
$H_N^{\{0.1\}\{Mules\}}$	0,974	0,964	6,663	2,855	–11,877	10,634
$H_N^{\{1.0\}\{Mules\}}$	0,971	0,960	7,193	3,305	–12,552	12,311
$\bar{H}_N^{\{0.1\}\{Mules\}}$	0,974	0,964	6,523	2,815	–11,552	10,484
$\bar{H}_N^{\{1.0\}\{Mules\}}$	0,972	0,961	7,089	3,241	–12,265	12,070
$H_N^{\{0.1\}\{Elefant\}}$	0,978	0,970	7,306	2,655	–14,058	9,889
$H_N^{\{1.0\}\{Elefant\}}$	0,974	0,963	7,471	3,198	–13,518	11,911
$\bar{H}_N^{\{0.1\}\{Elefant\}}$	0,980	0,972	6,894	2,507	–12,793	9,338
$\bar{H}_N^{\{1.0\}\{Elefant\}}$	0,974	0,964	7,292	3,118	–12,978	11,615

Таблица 4.14

Параметры линейных регрессионных моделей, описывающих зависимости между значениями показателей Херста накопленных сумм СП V_i и числом переданных пакетов каждым из рассматриваемых классов пользователей

Модель	R^2	R^2_{adj}	$a_{N^{\{Mise\}}}$	$S_N^{\{Mise\}}$	$a_{N^{\{Mule\}}}$	$S_N^{\{Mules\}}$	$a_{N^{\{Elefant\}}}$	$S_N^{\{Elefant\}}$
$H_V^{\{0.1\}\{Mise\}}$	0,998	0,997	1,157	0,245	1,892	1,823	–1,688	1,575
$H_V^{\{1.0\}\{Mise\}}$	0,997	0,995	0,916	0,336	–2,553	2,498	1,389	2,159
$\bar{H}_V^{\{0.1\}\{Mise\}}$	0,998	0,997	1,205	0,233	2,734	1,731	–2,311	1,496
$\bar{H}_V^{\{1.0\}\{Mise\}}$	0,998	0,996	0,921	0,316	–2,275	2,350	1,220	2,031
$H_V^{\{0.1\}\{Mules\}}$	0,997	0,995	0,709	0,321	–3,113	2,388	1,946	2,063
$H_V^{\{1.0\}\{Mules\}}$	0,997	0,995	0,778	0,346	–4,293	2,572	2,562	2,223
$\bar{H}_V^{\{0.1\}\{Mules\}}$	0,998	0,996	0,700	0,286	–3,198	2,126	1,957	1,837
$\tilde{H}_V^{\{1.0\}\{Mules\}}$	0,997	0,995	0,776	0,334	–4,158	2,483	2,492	2,146
$H_V^{\{0.1\}\{Elefant\}}$	0,996	0,993	1,010	0,396	–1,888	2,945	0,426	2,545
$H_V^{\{1.0\}\{Elefant\}}$	0,997	0,994	0,896	0,375	–3,611	2,787	1,890	2,409
$\bar{H}_V^{\{0.1\}\{Elefant\}}$	0,996	0,994	0,977	0,356	–1,558	2,649	0,357	2,289
$\bar{H}_V^{\{1.0\}\{Elefant\}}$	0,997	0,995	0,893	0,360	–3,391	2,674	1,779	2,311

Таблица 4.15

Параметры линейных регрессионных моделей, описывающих зависимости между значениями показателей Херста накопленных сумм СП N_i и числом переданных пакетов каждым из рассматриваемых классов пользователей

Модель	R^2	R^2_{adj}	$a_{N^{\{Mise\}}}$	$S_N^{\{Mise\}}$	$a_{N^{\{Mule\}}}$	$S_N^{\{Mules\}}$	$a_{N^{\{Elefant\}}}$	$S_N^{\{Elefant\}}$
$H_N^{\{0.1\}\{Mise\}}$	0,994	0,989	0,686	0,383	–3,476	2,847	1,458	2,460
$H_N^{\{1.0\}\{Mise\}}$	0,997	0,994	0,803	0,370	–4,321	2,748	2,331	2,375

Модель	R^2	R^2_{adj}	$a_{N^{\{Mise\}}}$	$S_N^{\{Mise\}}$	$a_{N^{\{Mule\}}}$	$S_N^{\{Mules\}}$	$a_{N^{\{Elefant\}}}$	$S_N^{\{Elefant\}}$
$\bar{H}_N^{\{0.1\}\{Mise\}}$	0,995	0,992	0,653	0,319	–3,108	2,368	1,377	2,047
$\bar{H}_N^{\{1.0\}\{Mise\}}$	0,997	0,995	0,780	0,350	–4,249	2,603	2,354	2,250
$H_N^{\{0.1\}\{Mules\}}$	0,997	0,994	0,741	0,339	–3,098	2,518	1,939	2,176
$H_N^{\{1.0\}\{Mules\}}$	0,997	0,995	0,797	0,356	–4,237	2,649	2,508	2,289
$\bar{H}_N^{\{0.1\}\{Mules\}}$	0,997	0,995	0,721	0,328	–3,178	2,435	1,993	2,104
$\bar{H}_N^{\{1.0\}\{Mules\}}$	0,997	0,995	0,791	0,350	–4,060	2,598	2,436	2,246
$H_N^{\{0.1\}\{Elefant\}}$	0,997	0,994	1,020	0,361	–1,913	2,685	0,375	2,320
$H_N^{\{1.0\}\{Elefant\}}$	0,997	0,995	0,897	0,365	–3,670	2,711	1,893	2,343
$\bar{H}_N^{\{0.1\}\{Elefant\}}$	0,997	0,994	0,948	0,346	–1,587	2,570	0,482	2,221
$\bar{H}_N^{\{1.0\}\{Elefant\}}$	0,997	0,995	0,876	0,356	–3,462	2,645	1,871	2,286

Таблица 4.16

Параметры линейных регрессионных моделей, описывающих зависимости между значениями показателей Херста накопленных сумм СП V_i и объемом информации, приходящейся на один пакет, переданного каждым из рассматриваемых классов пользователей

Модель	R^2	R^2_{adj}	$a_{\tilde{V}^{\{Mise\}}}$	$S_{\tilde{V}}^{\{Mise\}}$	$a_{\tilde{V}^{\{Mule\}}}$	$S_{\tilde{V}}^{\{Mules\}}$	$a_{\tilde{V}^{\{Elefant\}}}$	$S_{\tilde{V}}^{\{Elefant\}}$
$H_V^{\{0.1\}\{Mise\}}$	0,995	0,992	3,508	1,486	0,328	0,226	–0,252	0,377
$H_V^{\{1.0\}\{Mise\}}$	0,997	0,995	5,156	1,320	0,724	0,201	–0,744	0,335
$\bar{H}_V^{\{0.1\}\{Mise\}}$	0,995	0,992	3,146	1,444	0,253	0,220	–0,154	0,366
$\bar{H}_V^{\{1.0\}\{Mise\}}$	0,997	0,995	5,005	1,268	0,693	0,193	–0,703	0,322
$H_V^{\{0.1\}\{Mules\}}$	0,998	0,996	4,979	1,035	0,706	0,157	–0,782	0,262
$H_V^{\{1.0\}\{Mules\}}$	0,998	0,996	5,520	1,173	0,854	0,178	–0,902	0,297
$\bar{H}_V^{\{0.1\}\{Mules\}}$	0,998	0,996	4,770	1,005	0,697	0,153	–0,746	0,255
$\tilde{H}_V^{\{1.0\}\{Mules\}}$	0,998	0,996	5,422	1,152	0,835	0,175	–0,876	0,292
$H_V^{\{0.1\}\{Elefant\}}$	0,996	0,993	4,625	1,381	0,666	0,210	–0,657	0,350
$H_V^{\{1.0\}\{Elefant\}}$	0,997	0,995	5,420	1,269	0,824	0,193	–0,851	0,322
$\bar{H}_V^{\{0.1\}\{Elefant\}}$	0,997	0,996	4,425	1,109	0,621	0,169	–0,604	0,281
$\bar{H}_V^{\{1.0\}\{Elefant\}}$	0,998	0,996	5,288	1,178	0,798	0,179	–0,816	0,299

Таблица 4.17

Параметры линейных регрессионных моделей, описывающих зависимости между значениями показателей Херста накопленных сумм СП N_i и объемом информации, приходящейся на один пакет, переданным каждым из рассматриваемых классов пользователей

Модель	R^2	R^2_{adj}	$a_{\tilde{V}^{\{Mise\}}}$	$S_{\tilde{V}}^{\{Mise\}}$	$a_{\tilde{V}^{\{Mule\}}}$	$S_{\tilde{V}}^{\{Mules\}}$	$a_{\tilde{V}^{\{Elefant\}}}$	$S_{\tilde{V}}^{\{Elefant\}}$

$H_N^{\{0.1\}\{Mise\}}$	0,988	0,980	3,729	1,900	0,689	0,289	–0,649	0,482
$H_N^{\{1.0\}\{Mise\}}$	0,996	0,993	5,307	1,507	0,863	0,229	-0,888	0,382
$\bar{H}_N^{\{0.1\}\{Mise\}}$	0,991	0,984	3,570	1,636	0,635	0,249	–0,599	0,415
$\bar{H}_N^{\{1.0\}\{Mise\}}$	0,996	0,994	5,241	1,395	0,844	0,212	–0,872	0,354
$H_N^{\{0.1\}\{Mules\}}$	0,997	0,994	5,187	1,250	0,731	0,190	–0,816	0,317
$H_N^{\{1.0\}\{Mules\}}$	0,997	0,995	5,562	1,248	0,859	0,190	–0,906	0,317
$\bar{H}_N^{\{0.1\}\{Mules\}}$	0,997	0,995	5,073	1,154	0,723	0,176	–0,799	0,293
$\bar{H}_N^{\{1.0\}\{Mules\}}$	0,998	0,996	5,494	1,202	0,837	0,183	–0,884	0,305
$H_N^{\{0.1\}\{Elefant\}}$	0,997	0,994	4,481	1,332	0,669	0,203	–0,633	0,338
$H_N^{\{1.0\}\{Elefant\}}$	0,997	0,995	5,354	1,275	0,828	0,194	–0,843	0,323
$\bar{H}_N^{\{0.1\}\{Elefant\}}$	0,998	0,996	4,500	1,101	0,624	0,167	–0,622	0,279
$\bar{H}_N^{\{1.0\}\{Elefant\}}$	0,998	0,996	5,322	1,188	0,802	0,181	–0,827	0,301

Из таблиц 4.12–4.17 видно:

1. Переменная, соответствующая объему информации, переданной классом «Слоны» отсутствует в регрессионных моделях, описывающих зависимости между значениями показателей Херста накопленных сумм СП V_i и объемами переданной информации каждым из рассматриваемых классов пользователей, а также в регрессионных моделях, описывающую зависимости показателя Херста накопленных сумм СП N_i от объемов переданной информации каждым из классов пользователей, что свидетельствует о наличии близкой к функциональной зависимости между объемами переданной информации классами «Мыши», «Мулы» и «Слоны».

2. Среднее значение скорректированного коэффициента детерминации R^2_{adj} = 0,964, что свидетельствует о статистической значимости построенных регрессионных моделей, а также о наличии сильной зависимости между показателями Херста накопленных сумм изученных СП и объемами переданной пользователями информации.

Для объяснения причины, обусловившей отсутствие в регрессионных моделях, описывающих зависимости между значениями показателей Херста накопленных сумм СП V_i, N_i и объемами переданной информации каждым из рассматриваемых классов пользователей, переменных, соответствующих «Слонам», были изучены корреляционные связи между объемами информации, переданной «Слонами», и объемами информации, переданной «Мышами» и «Мулами»(см. таблицу 4.18).

Таблица 4.18

Объем информации, переданный каждым из классов пользователей в каждый из 7 дней

День	*Объем информации, переданной классом «Мыши»*	*Объем информации, переданной классом «Мулы»*	*Объем информации, переданной классом «Слоны»*
1	30996017241	8118004515	37638020936
2	28494531094	7462853381	34600502042
3	22566213503	6769864050	29336077554
4	26623867248	7987160174	34611027422
5	24346132026	6376367911	29563160317
6	18022343530	4095987165	20479935829
7	18424280953	4187336580	20936682901

Анализ данных, представленных в таблице 18, показал, что между значениями объемов переданной информации каждым из классов пользователей имеет место линейная зависимость вида:

$$V^{\{Elefants\}} = 0,625 \cdot V^{\{Mice\}} + 2,25 \cdot V^{\{Mules\}} \quad (4)$$

с коэффициентом детерминации R^2 равным 1, чем объясняется исключение одной из переменных из регрессионных моделей, параметры которых представлены в таблицах 12, 13. С физической точки зрения (4) является следствием «закона сохранения общего объема переданной информации» в исследованном высокоскоростном магистральном канале, являющегося суммой объемов переданной информации каждым из выбранных классов пользователей, который, как очевидно, не может превосходить некоторого максимального объема, определяющегося техническими характеристиками канала передачи. Полученный результат свидетельствует о возможности разработки механизмов балансировки объемов передаваемой информации для повышения качества обслуживания пользователей каналом (проактивного контроля загрузки канала), например, за счет управления скоростью передачи информации каждого из выделенных классов пользователей в зависимости от текущих значений характеристик информационных потоков, вычисляемых в реальном режиме времени. Для этого, принимая во внимание полученные подтверждения существования зависимостей показателей Херста накопленных сумм временных рядов $N_i, V_i, \tilde{V}_i$, от объемов переданной информации каждым из классов пользователей, а также соответствующих количеств переданных пакетов, представляется перспективным использование этих показателей при решении рассматриваемой задачи.

Блок-схема одного из возможных алгоритмов проактивного управления загрузкой канала передачи информационных потоков с целью обеспечения его максимально полной утилизации, основанного на использования глобального показателя Херста накопленных сумм СП V_i, порожденных активностью пользователей класса «Мыши», который оказывается более чувствительным к локальным изменениям показателей балансируемых

информационных потоков, представлена на рис. 4.7. Данный алгоритм обеспечивает минимизацию кол-ва сбросов скользящего окна за счет сравнения глобального показателя Херста накопленных сумм СП V_i, порожденных активностью пользователей класса «Мыши», с известным пороговым значением выбранного показателя, при превышении которого соответствующим образом регулируется скорость потоков класса «Слоны» и «Мулы». Пороговое значение глобального показателя Херста накопленных сумм СП V_i, значение скорости передачи данных для каждого из выбранных классов пользователей для каждой конкретной сети, а так же длительность временных интервалов должны определяться экспериментально. Эффективность в данном случае будет равна проценту недоутилизации сети ввиду сброса скользящего окна и может достигать до 25%. Так как логика работы скользящего окна является стандартом для сетей, использующих TCP/IP модель передачи данных (наиболее распространенный на данный момент стандарт) она может быть применена на любых сетях использующих данную модель передачи данных.

Отметим, что данный алгоритм можно реализовать в реальном режиме времени. Действительно, в проведенном исследовании проводилась обработка файлов с 15 минутными дампами Интернет-трафика, содержащего только на IP и TCP уровнях значения более 300 показателей, в то время как для идентификации информационных потоков, создаваемых каждым из классов пользователей, оказывается достаточным использовать, как показано в [10], всего 7 параметров: IP-адрес отправителя и получателя пакета, порты отправителя и получателя пакета, размер пакета. При этом подавляющая часть времени было затрачено на поиск в исходных файлах, содержащих с дампы Интернет-трафика, необходимых параметров, в то время как время выполнения собственно вычислительных операций составляло порядка нескольких миллисекунд.

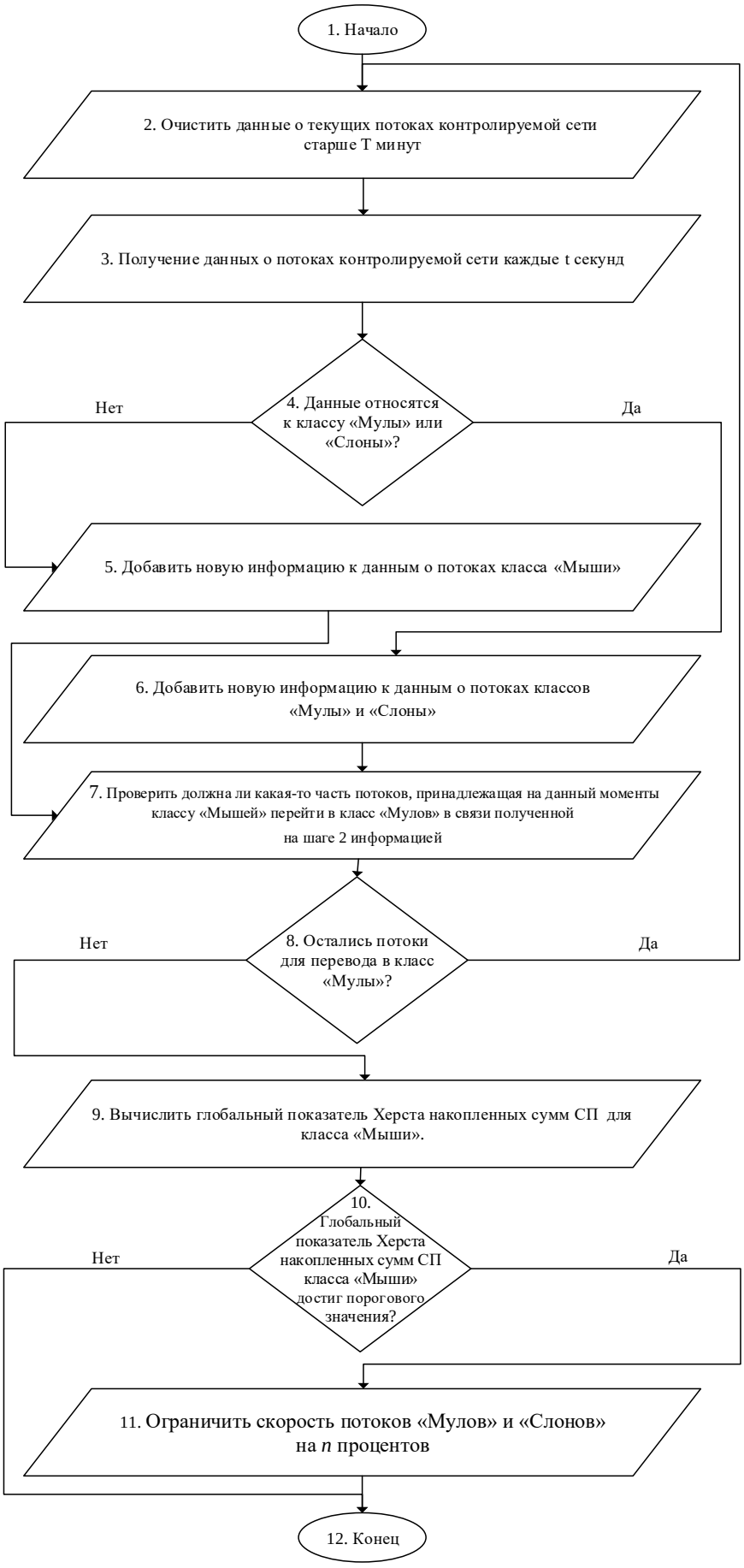


Рис. 4.7. Блок-схема алгоритма проактивного контроля нагрузки канала передачи информационных потоков

4.4. Выводы по главе

На основе анализа 15-ти минутных дампов Интернет-трафика, находящихся в свободном доступе в архиве MAWI [26], проведено исследование влияния информационных потоков, создаваемых в магистральном Интернет-канале пользователями, относящимися в зависимости от объемов информации, переданной инициированными ими потоками, к классам «Мыши», «Мулы» или «Слоны».

В соответствие с авторской методикой для каждого из выбранных классов потоков получены случайные последовательности (СП) N_i, V_i, $\tilde{V}_i$, содержащие упорядоченные во времени значения числа пакетов и объемов информации, переданных в течение заданного временного интервала, а также среднего объема переданной информации, приходящейся на один пакет.

Продемонстрировано, что вопреки устоявшимся представлениям, фрактальными (самоподобными) свойствами обладают не СП N_i, V_i, $\tilde{V}_i$, но их накопленные суммы, и получены соответствующие оценки их показателей Херста. Построены регрессионные модели, описывающие зависимости показателей Херста накопленных сумм СП N_i, V_i, $\tilde{V}_i$, от объемов информации и числа пакетов, переданных каждым из выбранных классов пользователей, по которым можно оценивать оценивать взаимное влияние информационных потоков, создаваемых «Мышами», «Мулами» и «Слонами», друг на друга.

Установлено, что значениями объемов информации, переданной в течении 15 минут каждым из классов пользователей, связаны детерминированными линейными зависимостями. Это свидетельствует о возможности разработки механизмов балансировки объемов передаваемой информации, призванных повысить качество обслуживания пользователей данным каналом, на основе управления скоростью передачи информации каждого из выделенных классов пользователей в зависимости от текущих значений показателей Херста накопленных сумм СП N_i, V_i, $\tilde{V}_i$ и приведен пример подобного алгоритма.

Заключение

Все поставленные в начале исследования задачи были решены. В частности, было разработано математическое и программное обеспечение для анализа трафика, передаваемого в высокоскоростных магистральных каналах передачи данных, что позволило получить следующие новые результаты:

1. Проведен анализ методов исследования информационных потоков в КС сетей с точки зрения их применимости для исследования трафика в высокоскоростных магистральных Интернет-каналах, и соответствующих программных инструментов, результаты, которого подтвердили необходимость разработки специализированного математического и алгоритмического программного обеспечения, обеспечивающего автоматическое извлечение информации из pcap-файлов в выбранном измерении.

2. Создан программно-аппаратный комплекс, обеспечивающий извлечение в автоматизированном режиме из дампов трафика, передаваемого в высокоскоростном магистральном Интернет-канале, количественные его количественные характеристики, адаптированный для суперкомпьютера «Уран» ИММ УрО РАН.

3. Разработана методики анализа первичной информации, извлекаемой из pcap-файлов, обеспечивающая получение количественных характеристики информационных потоков, переданных в магистральном высокоскоростном Интернет-канале.

4. Изучены в соответствие с авторской методикой особенности информационных потоков в магистральном Интернет-канале, создаваемых выбранными классами пользователей («Слоны», «Мулы», «Мыши»), и их взаимное влияния друг на друга, в том числе, установлено, что:

4.1. в информационных потоках, создаваемых, классом «Мыши» преобладающими являются пакеты размером не более 200 байт, в информационных потоках, создаваемых, классами «Мулы» и «Слоны» преобладающими являются пакеты размером 1500 байт;

4.2. наиболее стабильными во времени параметры распределения информационных потоков по размеру пакетов в течение недели оказываются у класса пользователей «Слоны» и «Мулы», наименее стабильные у пользователей класса «Мыши»;

4.3. временные ряды Ni, Vi, содержащие упорядоченные во времени значения числа пакетов и объемов информации, переданных в течение заданного временного интервала, а также среднего объема переданной информации, приходящейся на один пакет. представляют собой случайные последовательности с ограниченной областью рассеяния, накопленные суммы которых обладают свойством самоподобия;

4.4. существует детерминированная линейная связь между объемами информации, переданной в течении 15 минут каждым из классов пользователей, что свидетельствует о возможности создания на основе управления скоростью передачи информации каждого из выделенных классов пользователей механизмов балансировки объемов передаваемой информации, призванных повысить качество обслуживания пользователей данным каналом, и предложена структурная схема подобного алгоритма.

Перспективы дальнейшей разработки темы исследования заключаются в поиске возможностей совершенствования предложенного в данной работе алгоритма управления высокоскоростными магистральными Интернет-каналами с целью повышения эффективности и качества их работы.

СПИСОК СОКРАЩЕНИЙ

ААФ –алгоритм генерации фрактального броуновского движения, на основе Фурье-фильтрации

ВР – временной ряд

ГА – генетический алгоритм

ГЖМ – гибридная жидкостная модель

ЖМ – жидкостная модель

КС – компьютерная сеть

НР – нормальное распределение

ПР – плотность распределения

СНР – стандартное нормальное распределение

СП – случайная последовательность

ФБД – фрактальное броуновское движение

ФР – функция распределения

ADSL – Asymmetric Digital Subscriber Line (Асимметричная цифровая абонентская линия)

API – Application Program Interface (программный интерфейс приложения)

ICMP – Internet Control Message Protocol (протокол межсетевых управляющих сообщений)

IGMP – Group Management Protocol (протокол управления группами Интернета)

IP – Internet Protocol (сетевая модель передачи данных, представленных в цифровом виде)

ISO – International Organization of Standardization (Международная организация по стандартизации)

OSI – Open System Interconnection (модель Взаимодействия открытых систем)

SDN – Software-defined Networking (программно-конфигурируемая сеть)

TCP – Transmission Control Protocol (протокол управления передачей данных в Интернет)

TCP/IP – Transmission Control Protocol (протокол управления передачей данных в Интернет)/ Internet Protocol (сетевая модель передачи данных, представленных в цифровом виде)

UDP – User Datagram Protocol (протокол пользовательских датаграмм)

VLAN – Virtual Local Area Network (виртуальная локальная компьютерная сеть)

Список литературы

1. Abry P., F. Sellan. The wavelet-based synthesis for the fractional Brownian motion proposed by F. Sellan and Y. Meyer: Remarks and fast implementation // Appl. and Comp. Harmonic Anal. – 1996. – № 3(4). – P. 377–383.
2. Armenb/sharktools [Электронный ресурс] // GitHub - Build software better, together. URL: https://github.com/armenb/sharktools (дата обращения: 13.03.2014).
3. Benoit B. Mandelbrot, John W. Van Ness. Fractional Noises and Applications // SIAM Review. – 1968. – Vol. 10, №. 4. – P. 422-437.
4. Benoit B. Mandelbrot. Comment on computer rendering of fractal stochastic models // Communications of the ACM. – 1982. – Vol. 25, № 8. – P. 581–583.
5. Borgnat P. et al. Seven years and one day: Sketching the evolution of internet traffic //INFOCOM 2009, IEEE. – IEEE, 2009. – С. 711-719.
6. Chaabane A., Manils P., Kaafar M. A. Digging into anonymous traffic: A deep analysis of the tor anonymizing network //Network and System Security (NSS), 2010 4th International Conference on. – IEEE, 2010. – С. 167-174.
7. Cisco Systems. Interconnecting Cisco networking devices // Cisco press. Vol. 1, 2013. 278 p.
8. Computerworld Россия [Электронный ресурс] URL: https://www.osp.ru/cw/2000/38/7298/ (Дата обращения: 28.06.2017)
9. Connecting to the Cloud [Электронный ресурс] // Mathworks – MATLAB and Simulink for technical computing. URL: http://www.mathworks.com/mobile/connect-to-cloud.html (дата обращения: 10.02.2015).
10. Disruptive technologies barometer: Telecommunications sector [Электронный ресурс]. URL: https://assets.kpmg.com/content/dam/kpmg/xx/pdf/2016/12/disruptive-technologies-barometer-telecom-report.pdf (дата обращения: 01.12.2017).
11. Flandrin P. Wavelet analysis and synthesis of fractional Brownian motion // IEEE Trans. on Inf. Th., 38. – 1992. – P. 910–917.
12. Foumer A., Fussel D., Carpenter L. Computer rendering of stochastic models // Communications of the ACM. – 1982. – Vol. 25, № 6. – P. 371–384.
13. Functions in Parallel Computing Toolbox [Электронный ресурс] // Mathworks – MATLAB and Simulink for technical computing. URL: http://www.mathworks.com/help/distcomp/functionlist.html (дата обращения: 10.02.2015).
14. Generators of long-range dependence processes: a survey / J.-M. Bardet, G. Lang, G. Oppenheim, A. Philippe, S. Stoev, M.S. Taqqu // Theory and applications of long-range dependence. – 2003. – P. 579–623

15. GNU Octave [Электронный ресурс] // Операционная система GNU. 1998-2013. URL: https://www.gnu.org/software/octave (дата обращения: 13.03.2014).

16. IEEE STANDARTS ASSOCIATION [Электронный ресурс]. URL: http://standards.ieee.org/findstds/standard/802.3x-1997.html (дата обращения: 01.12.2017).

17. IEEE STANDARTS ASSOCIATION [Электронный ресурс]. URL: https://standards.ieee.org/findstds/standard/802-2014.html (дата обращения: 01.12.2017).

18. International Organization for Standardization [Электронный ресурс]. URL: https://www.iso.org (дата обращения: 01.12.2017).

19. Internet assigned numbers authority: [Электронный ресурс]. URL: http://www.iana.org/protocols#P (дата обращения: 01.02.2015).

20. Istas J., Lang G. Quadratic variations and estimation of the local Hölder index of a Gaussian process // Ann. Inst. Poincaré, 33. – 1994. – P. 407–436.

21. Krishnamurthy B. et al. Sketch-based change detection: methods, evaluation, and applications //Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement. – ACM, 2003. – C. 234-247.

22. Lihua N., Xiaorong C., Qian H. ARIMA model for traffic flow prediction based on wavelet analysis // Information Science and Engineering (ICISE), 2010 2nd International Conference on. — IEEE, 2010. — P.1028-1031

23. M. Soysal, E.G. Schmidt: Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison. Performance Evaluation Elsevier Journal, Vol. 67, pp. 451-467 (2010).

24. Mathworks – MATLAB and Simulink for Technical Computing [Электронный ресурс]. 1994-2014. URL: www.mathworks.com (дата обращения: 13.03.2014).

25. Mathworks – MATLAB and Simulink for Technical Computing [Электронный ресурс]. 1994-2014. URL: https://www.mathworks.com/matlabcentral/ fileexchange/29686-multifractal-model-of-asset-returns--mmar-/content/ffGn.m (дата обращения: 01.07.2016).

26. MATLAB и многоядерность. Работа в MATLAB на многоядерных и многопроцессорных компьютерах [Электронный ресурс] // MATLAB и Simulink центр компетенций компании Mathworks. URL: http://matlab.ru/solutions/tech-calc/parallel-computing/matlab-multicore (дата обращения: 10.02.2015).

27. MAWI Working Group Traffic Archive [Электронный ресурс]. URL: http://mawi.wide.ad.jp/mawi (дата обращения: 01.07.2016).

28. Neeraj Sharma and Matthias K. Gobbert. A comparative evaluation of MATLAB, octave, freemat, and scilab for research and teaching // University of Maryland.–Baltimore County.– 2010.

29. Neeraj Sharma. A comparative study of several numerical computational packages. M.S. thesis // University of Maryland.–Baltimore County.–2010.

30. Network media specific capturing [Электронный ресурс] // Wireshark. URL: http://wiki.wireshark.org/CaptureSetup/NetworkMedia (дата обращения: 13.03.2014).

31. Parzen E. On estimation of a probability density function and mode //The annals of mathematical statistics. – 1962. – Т. 33. – №. 3. – С. 1065-1076.

32. Pellicer-Lostao C., Morato D., Popez-Ruiz R. Modelling user's activity in a real-world complex network // International Journal of Computer Mathematics. Bristol: Taylor & Francis. Vol. 85, 2008. P. 1287– 1298.

33. Porshnev S. V., Bozhalkin D. A. The study of self-similarity of the traffic transmitted in the backbone Internet channel //Dynamics of Systems, Mechanisms and Machines (Dynamics), 2016. – IEEE, 2016. – С. 1-7.

34. Red Cloud [Электронный ресурс] // Cornell University Center for Advanced Computing. URL: https://www.cac.cornell.edu/RedCloud/default.aspx (дата обращения: 10.02.2015).

35. Register for Access to MATLAB Distributed Computing Server on the Cloud [Электронный ресурс] // Mathworks – MATLAB and Simulink for technical computing. URL: http://www.mathworks.com/programs/mdcs-cloud.html (дата обращения: 03.02.2015).

36. Rosenblatt M. et al. Remarks on some nonparametric estimates of a density function //The Annals of Mathematical Statistics. – 1956. – Т. 27. – №. 3. – С. 832-837.

37. Sage: Open Source Mathematics Software [Электронный ресурс]. URL: www.sagemath.org (дата обращения: 13.03.2014).

38. Scilab - Open source software for numerical computation [Электронный ресурс]. URL: www.scilab.org (дата обращения: 13.03.2014).

39. Self-similarity and long-range dependence through the wavelet lens / P. Abry, P. Flandrin, M.S. Taqqu, D. Veitch // Theory and applications of long-range dependence. – 2003. – P. 527–556.

40. Semi-parametric estimation of the long-range dependence parameter: a survey / J.-M. Bardet, G. Lang, G. Oppenheim, A. Philippe, S. Stoev, M.S. Taqqu // Theory and applications of long-range dependence. – 2003. – P. 557–577.

41. Shi J. et al. There is a will, there is a way: a new mechanism for traffic control based on VTL and VANET // High Assurance Systems Engineering (HASE), 2015 IEEE 16th International Symposium on. — IEEE, 2015. — P.240-246.

42. Stoev S. et al. On the wavelet spectrum diagnostic for Hurst parameter estimation in the analysis of Internet traffic //Computer Networks. – 2005. – Т. 48. – №. 3. – С. 423-445.

43. Szabó G. Methods for efficient classification of network traffic. Thesis, Budapest University of Technology and Economics, p.10 (2010).

44. Tcdump/Libcap public repository [Электронный ресурс]. 2010-2014. URL: http://www.tcpdump.org (дата обращения: 13.03.2014).

45. The Internet Engineering Task Force | Transmission Control Protocol [Электронный ресурс]. URL: https://tools.ietf.org/html/rfc793 (дата обращения: 04.12.2017).

46. The world's most popular free OS | Ubuntu [Электронный ресурс]. URL: www.ubuntu.com (дата обращения: 13.03.2014).

47. Traffic Trace Page [Электронный ресурс]. URL: http://mawi.nezu.wide.ad.jp/mawi/samplepoint-F/2014/201401071400.dump.gz (дата обращения: 13.03.2014).

48. WebCite query result [Электронный ресурс] // WebCite. URL: http://www.webcitation.org/64vYth5tU (дата обращения: 13.03.2014).

49. Wireshark [Электронный ресурс]. URL: www.wireshark.org (дата обращения: 13.03.2014).

50. Агеев Д. В. Статистический анализ нагрузки создаваемой абонентами ADSL при безлимитном доступе в сеть интернет / Д. В. Агеев, Д. В. Евлаш // «Вісник ДУІКТ». Т. 8, № 1. – 2010. – С. 38–43.

51. Агеев, Д. В. Методика определения параметров потоков на разных участках мультисервисной телекоммуникационной сети с учетом эффекта самоподобия [Текст] / Д. В. Агеев, А. А. Игнатенко, А. Н. Копылев // Проблемы телекоммуникаций. - Харьков. - 2011. - № 3 (5).- С. 18 - 37.

52. Ажмухамедов И. М., Марьенков А. Н. Повышение безопасности компьютерных систем и сетей на основе анализа сетевого трафика // Инфокоммуникационные технологии . – 2010. – Т . 8, No 3. – С . 106–172.

53. Ажмухамедов И. М., Марьенков А. Н., "Обеспечение информационной безопасности компьютерных сетей на основе анализа сетевого трафика", Вестн. Астрахан. гос. техн. ун-та. Сер. управление, вычисл. техн. информ., 2011, № 1, 137–141.

54. Анализ и моделирование трафика в высокопроизводительных компьютерных сетях // С.Д.Белов, С.В.Ломакин, В.А.Огородников, С.М.Пригарин, А.С.Родионов, Л.Б.Чубаров // Вестник НГУ. Серия: Информатика. - 2008. - Т.6, вып. 2. - С.41-49.

55. Афонцев Э. В. Разработка методики выявления аномалий трафика в магистральных интернет-каналах : дис. – Екатеринбург : Афонцев Эдуард Вячеславович, 2007.

56. Афонцев Э. В., Поршнев С. В. Детектирование аномалий интернет-трафика на основе вычисления коэффициента корреляции //Научные труды X отчетной конференции

молодых ученых ГОУ ВПО УГТУ-УПИ. Ч. 1.—Екатеринбург, 2006. – Уральский государственный технический университет-УПИ, 2006. – С. 173-175.

57. Бахарева. Н.Ф. Анализ и расчет непуассоновских моделей трафика в сетях ЭВМ / Н.Ф. Бахарева, И.В. Карташевский, В.Н. Тарасов// Инфокоммуникационные технологии. — 2009. — №4. — С. 61-66.

58. Буранова М. А. Исследование статистических характеристик самоподобного телекоммуникационного трафика //Инфокоммуникационные технологии. – 2012. – Т. 10. – №. 4. – С. 35-40.

59. Гальцев, Алексей Анатольевич. Системный анализ трафика для выявления аномальных состояний сети : автореферат диссертации на соискание ученой степени кандидата технических наук : 05.13.01 / А. А. Гальцев ; Самар. гос. аэрокосм. ун-т им. С. П. Королева. - Самара, 2013. - 19 с.

60. Гребенкин М.К. Влияние активности пользователей сети Интернет на свойства мультисервисного трафика / М.К. Гребенкин, С.В. Поршнев// Научно-технические ведомости СПбГПУ, Сер.: Информатика. Телекоммуникации. Управление. — 2011. — № 1 (115). — С. 7-12.

61. Гребенкин М.К., Поршнев С.В. Гибридная жидкостная модель магистрального Интернет-канала. — LAP LAMBERT Academic Publishing, 2012. — 172 с.

62. Закревская Н.С., Ковалевский А.П. Алгоритм идентификации фрактального броуновкого движения по разности оценок // Сборник научных трудов НГТУ. - 2004. – № 2(36). – С. 29 – 36.

63. Карпухин А. В. и др. Применение методов нелинейной динамики и фрактального анализа для оценивания работы инфокоммуникационных систем с протоколом TCP //Cloud of science. – 2014. – Т. 1. – №. 2.

64. Киреева Н.В., Чупахина Л.Р. Частный случай исследования параметров трафика сети для определения законов распределения времени передачи пакетов // Международный журнал прикладных и фундаментальных исследований. — 2015. — № 5. — С. 395-397.

65. Копосов А.С. О выборе математических моделей распределений ограниченных случайных последовательностей [Электр.] / С.В. Поршнев, А.С. Копосов // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета (Научный журнал КубГАУ) [Электронный ресурс]. – Краснодар: КубГАУ, 2012. – №10(84). – Режим доступа: http://ej.kubagro.ru/2012/10/pdf/53.pdf.

66. Копосов Александр Сергеевич. Математическое и алгоритмическое обеспечение для обработки случайных данных с ограниченной областью рассеяния: диссертация ...

кандидата Технических наук: 05.13.01 / Копосов Александр Сергеевич;[Место защиты: ФГБОУ ВО Сибирский государственный университет телекоммуникаций и информатики], 2017

67. Кроновер Р.М. Фракталы и хаос: в динамических системах. Основы теории. — М: Постмаркет, 2000. — 352 с.

68. Кучук Г.А., Можаєв О.О. Прогнозирование трафика для управления перегрузками интегри-рованной телекоммуникационной сети // Радиоэлектронные и компьютерные системы. — 2006. — № 6 (18). — С. 261-271

69. Лемешко А.В. Тензорная модель многопутевой маршрутизации агрегированных потоков с резервированием сетевых ресурсов, представленная в пространстве с кривизной // СПб.: Наука и техника, 2004. — № 4(40). — С. 12-18.

70. Меркулова И.А. Исследование и разработка методов анализа трафика Интернет-провайдера: дис. ... канд. тех. наук. ГОУВПО ПГАТИ, Самара, 2007.

71. Пакет Math Works MATLAB [Электронный ресурс] // Лаборатория суперкомпьютерного моделирования ЮУрГУ. URL: http://supercomputer.susu.ac.ru/users/simulation/matlab/ (дата обращения: 10.02.2015).

72. Параллельный MATLAB. Общая информация [Электронный ресурс] // Параллельные вычисления в УрО РАН. URL: http://parallel.uran.ru/node/119#com-lic (дата обращения: 10.02.2015).

73. Перышкин С. В. Выделение устойчивых фрагментов сетевого трафика и оценка их стационарности для системы анализа поведения сетей //Вопросы защиты информации. – 2010. – №. 3. – С. 42-49.

74. Петерс Э. Хаос и порядок на рынках капитала. Новый аналитический взгляд на циклы, цены и изменчивость рынка. – М.: Мир, 2000.

75. Петров В.В. Структура телетрафика и алгоритм обеспечения качества обслуживания при влиянии эффекта самоподобия: диссертация канд. тех. наук. — М. 2004. — С. 4-8.

76. Петров В.В., Платов В.В. Исследование самоподобной структуры телетрафика беспроводной сети // Радиотехнические тетради. — 2004. — № 30.-С. 58-62.

77. Поршнев С. В., Афонцев Э. В. Спектральные свойства аномального Интернет-трафика //Информационные технологии. – 2006. – №. 12. – С. 66-69.

78. Поршнев С. В., Божалкин Д. А. О возможности использования случайных величин с ограниченной областью рассеяния для генерации фрактального броуновского движения //Информатика и системы управления. – 2017. – №. 1. – С. 23-32.

79. Поршнев С.В. Теория и алгоритмы аппроксимации эмпирических зависимостей и распределений / Е. В. Овечкина, В.Е. Каплан // –Екатеринбург: УрО РАН, 2006. –166 с.

80. Поршнев С.В., Божалкин Д.А. Анализатор-классификатор информационных потоков дампов трафика компьютерных сетей // Свидетельство о государственной регистрации программы для ЭВМ № 2015661799 (Заявка № 2015618601. Дата поступления 17 сентября 2015 г. Дата государственной регистрации в Реестре программ для ЭВМ 09 ноября 2015 г.)

81. Поршнев С.В., Божалкин Д.А. К вопросу о самоподобии трафика, передаваемого в магистральном Интернет-канале // Фундаментальные исследования. – 2016. – № 2(32). – С. 301–310.

82. Поршнев С.В., Божалкин Д.А. Семантический анализатор дампов трафика информационных потоков в компьютерных сетях // Свидетельство о государственной регистрации программы для ЭВМ № 2015611426 (Заявка № 2014662922. Дата поступления 12 декабря 2014 г. Дата государственной регистрации в Реестре программ для ЭВМ 29 января 2015 г.)

83. Поршнев С.В., Божалкин Д.А. Технология семантического анализа дампа трафика информационных потоков в компьютерных сетях // Информационные технологии. 2014. №11. С. 12-19.

84. Поршнев С.В., Божалкин Д.А., Копосов А.С. Исследование особенностей потоков сетевого трафика в магистральном интернет-канале // Электросвязь. — 2016. — №2. — С. 16–23.

85. Поршнев С.В., Божалкин Д.А., Копосов А.С. Опыт использования суперкомпьютера для обработки дампов сетевого трафика магистрального интернет-канала // Информационные технологии. – 2016. – №1. – С. 42-47.

86. Поршнев С.В., Копосов А.С. Использование аппроксимации Розенблатта–Парзена для восстановления функции распределения непрерывной случайной величины с ограниченным одномодальным законом распределения // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета (Научный журнал КубГАУ) [Электронный ресурс]. Краснодар: КубГАУ, 2013. №08(092).

87. Поршнев С.В., Копосов А.С. Методика использования генетических алгоритмов в задаче оценки параметров распределений с ограниченной областью рассеяния // Современные проблемы науки и образования. 2014. № 4. С. 168.

88. Поршнев С.В., Копосов А.С. О выборе математических моделей распределений ограниченных случайных последовательностей// Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. 2012. №10(84). С. 298-312.

89. Поршнев С.В., Степаненко В.А., Калмыков А.А., Владимиров В.А., Дядькова С.Н. Диагностика газоперекачивающих агрегатов на основе анализа технологи ческой информации. — Екатеринбург: УрО РАН, 2007. — 203 с.

90. Поршнев, С. В. Математические модели информационных потоков в высокоскоростных магистральных интернет-каналах [Текст] : учеб. пособие для вузов / С. В. Поршнев. - Москва : Горячая линия-Телеком, 2016. - 232 с..

91. Свидетельство о государственной регистрации программы для ЭВМ № 2016614275 (Заявка № 2016611747. Дата поступления 02 марта 2016 г. Дата государственной регистрации в Реестре программ для ЭВМ 20 апреля 2016 г.).

92. Симаков Д. В., Кучин А. А. Анализ статистических характеристик Интернет трафика в магистральном канале //T-Comm-Телекоммуникации и Транспорт. – 2015. – Т. 9. – №. 5.

93. Системные требования ОС Windows 7 – Microsoft Windows [Электронный ресурс] // Microsoft Windows. URL: http://windows.microsoft.com/ru-ru/windows7/products/system-requirements (дата обращения: 13.03.2014).

94. Стешенко В.В. Исследование Интернет-трафика пользователей корпоративной вычислительной сети Астраханского государственного технического университета // Вестник АсГТУ. Астрахань, 2008.– С. 130-132.

95. Текущий рейтинг. 21-ая редакция от 23.09.2014 [Электронный ресурс] // TOP 50 | Суперкомпьютеры. URL: http://top50.supercomputers.ru/?page=rating (дата обращения: 10.02.2015).

96. Треногин Н.Г., Соколов Д.Е. Фрактальные свойства сетевого трафика в клиент-серверной информационной системе // Вестник НИИ СУВПТ/ Новосибирск, Изд. СибГУТИ, 2003 – С. 163-172.

97. Установка Ubuntu Linux: инструкция для пользователя Windows [Электронный ресурс] // Русскоязычная документация по Ubuntu. URL: http://help.ubuntu.ru/wiki/ubuntu_install (дата обращения: 13.03.2014).

98. Федер Е. Фракталы. – М.: Мир, 1991.

99. Федеральный закон от 27.07.2006 N 149-ФЗ (ред. от 25.11.2017) «Об информации, информационных технологиях и о защите информации».

100.Шелухин О. И. Причины самоподобия телетрафика и методы оценки показателя Херста //Электротехнические и информационные комплексы и системы. – 2007. – Т. 3. – №. 1.

101.Шелухин О.И., Савелов А.В. Имитационное моделирование аномалий трафика в локальной компьютерной сети // T-Comm - Телекоммуникации и Транспорт. — 2013. — Т. 7, № 10. — С. 103-110

102.Шмелев И. В. Исследование и разработка метода оперативного управления мультисервисной сетью для потока трафика с фрактальными свойствами : дис. – М. : [Моск. техн. ун-т связи и информатики], 2005.

103.Эйнштейн А., Смолуховский М. Брауновское движение: сб. статей. – Ленинград: ОНТИ - Главная редакция общетехнической литературы, 1936.

104.Barakat C. et al. A flow-based model for internet backbone traffic //Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment. – ACM, 2002. – С. 35-47.

105. Karpukhin A., Griciv D., Nikulchev E. Simulation of chaotic phenomena in infocommunication systems with the TCP protocol //Journal of Theoretical & Applied Information Technology. – 2018. – Т. 96. – №. 15.

ПРИЛОЖЕНИЕ А. Программа для ЭВМ «Семантический анализатор дампов трафика информационных потоков в компьютерных сетях»

Исходный текст программы

1.1 sharktools_core.c

```
//
//  File: matshark_threads.c
//
//  Abstract: This software is intended for use for research purposes
//            to analyze modern large backbone with ability to use CPU
//            based parallel computing
//
//  Version: <1.0>
//
//  Copyright (c) 2014 Daniil Bozhalkin. All rights reserved.
//
////////////////////////////////////////////////////////////////////////

#include <stdio.h>
#include<stdlib.h>
#include <string.h>
#include <errno.h>

#include <assert.h>

#include <glib.h>
#include <glib/gthread.h>
#include <glib/gprintf.h>


/* заголовки wireshark */
#define WS_VAR_IMPORT extern
#include <config.h>
#include <file.h>
#include <epan/epan.h>
#include <epan/tap.h>
#include <epan/proto.h>
#include <epan/dfilter/dfilter.h>
#include <epan/epan_dissect.h>
#include <epan/filesystem.h>

#if WIRESHARK_0_99_5
#include <epan/emem.h>
#include <register.h>
#define ep_alloc_array0(type,num) (type*)ep_alloc0(sizeof(type)*(num))
#endif

#include <register.h>
#include <epan/plugins.h>

/* для wireshark старше версии 1.0 необходим get_credential_info(), который
находится в
 * privileges.h
 */

#if WIRESHARK_1_0_0
#include <epan/privileges.h>
#elif (WIRESHARK_1_2_0 || WIRESHARK_1_4_0)
#include <wsutil/privileges.h>
#endif
```

```
#include "sharktools_core.h"

#include "sharktools_add_decode_as.h"

#if (WIRESHARK_0_99_5 || WIRESHARK_1_0_0 || WIRESHARK_1_2_0)
#include "sharktools_epan.h"
#include "sharktools_frame_data.h"
#include "sharktools_cfile.h"
#endif

struct timespec ts;

#ifndef DEBUG
#define DEBUG 1
#endif

#if DEBUG
#define dprintf(args...) printf(args)
#else
#define dprintf(args...) ((void)0)
#endif

#include "sharktools_stdata.h"

typedef struct
{
  st_data_t *stdata;
  epan_dissect_t *edt;
} stdata_edt_tuple_t;

/* Аргументы используемые всеми потоками */
typedef struct thread_arguments {
  gulong nfields;
  st_data_t stdata;
  int ipacket;
} thread_arguments;

/* Функция обратного вызова потока */
void process_packet_callback(gpointer arguments, gpointer callbacks);

static const gchar* get_node_field_value_as_string(field_info* fi, epan_dis-
sect_t* edt);
void proto_tree_get_fields(st_data_t* stdata, epan_dissect_t *edt);
static void proto_tree_get_node_field_values(proto_node *node, gpointer
data);
static const gchar* get_field_hex_value2(GSList* src_list, field_info *fi);
static const guint8 *get_field_data(GSList *src_list, field_info *fi);

gboolean process_packet(capture_file *cf, gint64 offset, st_data_t *stdata);

extern char sharktools_errmsg[2048];

#define errmsg sharktools_errmsg

long verbose = 1;
static guint32 cum_bytes = 0;

static nstime_t first_ts;
static nstime_t prev_dis_ts;
static nstime_t prev_cap_ts;

static const char *cf_open_error_message(int err, gchar *err_info, int
file_type)
```

```
{
  const char *errmsg;
  static char errmsg_errno[1024+1];

  if (err < 0)
    {

      switch (err)
      {

      case WTAP_ERR_NOT_REGULAR_FILE:
        errmsg = "The file \"%s\" is a \"special file\" or socket or other
non-regular file.";
        break;

      case WTAP_ERR_FILE_UNKNOWN_FORMAT:
        /* При открытии файла для чтения. */
        errmsg = "The file \"%s\" isn't a capture file in a format Sharktools
understands.";
        break;

      case WTAP_ERR_UNSUPPORTED:
        /* При открытии файла для чтения. */
        g_snprintf(errmsg_errno, sizeof(errmsg_errno),
                  "The file \"%%s\" isn't a capture file in a format
Sharktools understands.\n"
                  "(%s)", err_info);
        g_free(err_info);
        errmsg = errmsg_errno;
        break;

      case WTAP_ERR_CANT_WRITE_TO_PIPE:
        /* При открытии для записи. */
        g_snprintf(errmsg_errno, sizeof(errmsg_errno),
                  "The file \"%%s\" is a pipe, and %s capture files can't be "
                  "written to a pipe.", wtap_file_type_string(file_type));
        errmsg = errmsg_errno;
        break;

      case WTAP_ERR_UNSUPPORTED_FILE_TYPE:
        /* При открытии для записи. */
        errmsg = "Sharktools doesn't support writing capture files in that
format.";
        break;

      case WTAP_ERR_UNSUPPORTED_ENCAP:
        g_snprintf(errmsg_errno, sizeof(errmsg_errno),
                  "The file \"%%s\" is a capture for a network type that
Sharktools doesn't support.\n"
                  "(%s)", err_info);
        g_free(err_info);
        errmsg = errmsg_errno;
        break;

      case WTAP_ERR_ENCAP_PER_PACKET_UNSUPPORTED:
        errmsg = "The file \"%s\" is a capture for a network type that
Sharktools doesn't support.";
        break;

      case WTAP_ERR_BAD_RECORD:
        /* При открытии для чтения. */
        g_snprintf(errmsg_errno, sizeof(errmsg_errno),
                  "The file \"%%s\" appears to be damaged or corrupt.\n"
                  "(%s)", err_info);
```

```
        g_free(err_info);
        errmsg = errmsg_errno;
        break;

      case WTAP_ERR_CANT_OPEN:
        errmsg = "The file \"%s\" could not be opened for some unknown rea-
son.";
        break;

      case WTAP_ERR_SHORT_READ:
        errmsg = "The file \"%s\" appears to have been cut short"
          " in the middle of a packet or other data.";
        break;

      case WTAP_ERR_SHORT_WRITE:
        errmsg = "A full header couldn't be written to the file \"%s\".";
        break;

      default:
        g_snprintf(errmsg_errno, sizeof(errmsg_errno),
                  "The file \"%%s\" could not be opened: %s.",
                  wtap_strerror(err));
        errmsg = errmsg_errno;
        break;
      }
    }
  else
        errmsg = file_open_error_message(err, FALSE);
  return errmsg;
}

cf_status_t cf_open(capture_file *cf, const char *fname, gboolean is_temp-
file, int *err)
{
  wtap       *wth;
  gchar       *err_info;

  dprintf("%s: fname =  %s\n", __FUNCTION__, fname);
  dprintf("%s: is_tempfile = %d  err = %p\n", __FUNCTION__, is_tempfile,
err);
  wth = wtap_open_offline(fname, err, &err_info, FALSE);
  dprintf("wth = %p\n", wth);
  if (wth == NULL)
    goto fail;

  /* Успешное открытие. */

#if WIRESHARK_1_4_0
  /* Очистить все структуры данных используемых для анализа. */
  cleanup_dissection();
#endif

  /* Инициализировать все структуры данных используемых для анализа. */
  init_dissection();

  cf->wth = wth;
  cf->f_datalen = 0;

  /* Задать имя файла для выбора фильтра потока */
  cf->filename = g_strdup(fname);

  /* Определить является ли файл постоянным или временным. */
  cf->is_tempfile = is_tempfile;
```

```
  /* Если это временный файл пометить его как не сохраняемый. */
  cf->user_saved = !is_tempfile;

  cf->cd_t      = wtap_file_type(cf->wth);
  cf->count     = 0;
  cf->drops_known = FALSE;
  cf->drops     = 0;
  cf->snap      = wtap_snapshot_length(cf->wth);
  if (cf->snap == 0)
    {
      cf->has_snap = FALSE;
      cf->snap = WTAP_MAX_PACKET_SIZE;
    }
  else
    cf->has_snap = TRUE;
  nstime_set_zero(&cf->elapsed_time);
  nstime_set_unset(&first_ts);
  nstime_set_unset(&prev_dis_ts);
  nstime_set_unset(&prev_cap_ts);

  dprintf("%s: exiting\n", __FUNCTION__);

  return CF_OK;

fail:
  g_snprintf(sharktools_errmsg, sizeof(sharktools_errmsg),
             cf_open_error_message(*err, err_info, cf->cd_t), fname);
  return CF_ERROR;
}

/* Ошибки открытия/создания. */

static void open_failure_message(const char *filename, int err, gboolean
for_writing)
{
  dprintf("sharktools: open error");
  dprintf("\n");
}

/* Общие ошибки. */

static void
failure_message(const char *msg_format, va_list ap)
{
  dprintf("sharktools: %s", __FUNCTION__);
  vfprintf(stderr, msg_format, ap);
  dprintf("\n");
}

/* Ошибки чтения. */
static void
read_failure_message(const char *filename, int err)
{
  dprintf("An error occurred while reading from the file \"%s\": %s.",
          filename, strerror(err));
}

#if (WIRESHARK_1_2_0 || WIRESHARK_1_4_0)
/* Ошибки записи. */
static void
write_failure_message(const char *filename, int err)
{
  dprintf("An error occurred while writing to the file \"%s\": %s.",
          filename, strerror(err));
```

```
}
#endif

static void stdata_init(st_data_t* stdata, gulong nfields)
{
  gsize i;

  stdata->fields = g_ptr_array_new();

  /* Таблица поиска индекса из строки аббревиатуры . */
  stdata->field_indicies = g_hash_table_new(g_str_hash, g_str_equal);


  /* Буфер для хранения типа и значения каждого пакета */
  stdata->field_values_str = g_new(const gchar*, nfields);

  stdata->field_values_native = g_new(fvalue_t*, nfields);
  for(i = 0; i < nfields; i++)
    {
      stdata->field_values_native[i] = g_new(fvalue_t, 1);
    }

  stdata->field_types = g_new(gulong, nfields);

}

/* Очистка stdata путем освобождения её членов в порядке обратном заполнению
*/
static void stdata_cleanup(st_data_t* stdata)
{
  gsize i;

  g_assert(stdata);

  g_free(stdata->field_types);

  for(i = 0; i < stdata->fields->len; i++)
    {
      g_free(stdata->field_values_native[i]);
    }
  g_free(stdata->field_values_native);

  g_free(stdata->field_values_str);

  if(NULL != stdata->field_indicies)
    {
      /* Ключи хранятся в stdata->fields, значения являются целыми числами */
      g_hash_table_destroy(stdata->field_indicies);
    }

  for(i = 0; i < stdata->fields->len; ++i)
    {
      gchar* field = g_ptr_array_index(stdata->fields,i);
      g_free(field);
    }

  g_ptr_array_free(stdata->fields, TRUE);


}

/*
 * Функция добавляет поля <fields> в структуру данных stdata путем прямого
копирования
```

```
 * строк и настройки хеш-таблицы field_indicies.
 */
static void stdata_add_fields(st_data_t* stdata, const gchar** fields, gsize
nfields)
{
  gsize i;

  g_assert(stdata);
  g_assert(fields);

  for(i = 0; i < nfields; i++)
    {
      dprintf("adding outputfield: %s\n", fields[i]);

      gchar* field_copy;

      field_copy = g_strdup(fields[i]);

      g_ptr_array_add(stdata->fields, field_copy);

      g_hash_table_insert(stdata->field_indicies, field_copy, (gulong *)(i));
    }


#if 0
  for(i = 0; i < stdata->fields->len; i++)
    {
      gchar* field = g_ptr_array_index(stdata->fields, i);
      g_hash_table_insert(stdata->field_indicies, field, (gulong *)(i));
    }
#endif

}

static const guint8 *get_field_data(GSList *src_list, field_info *fi)
{
  GSList *src_le;
  data_source *src;
  tvbuff_t *src_tvb;
  gint length, tvbuff_length;

  for (src_le = src_list; src_le != NULL; src_le = src_le->next)
    {
      src = src_le->data;
      src_tvb = src->tvb;
      if (fi->ds_tvb == src_tvb)
      {

        tvbuff_length = tvb_length_remaining(src_tvb,
                                       fi->start);
        if (tvbuff_length < 0)
          {
            return NULL;
          }
        length = fi->length;
        if (length > tvbuff_length)
          length = tvbuff_length;
        return tvb_get_ptr(src_tvb, fi->start, length);
      }
    }
  g_assert_not_reached();
  return NULL;
}
```

```
GTree *G_native_types;

static inline gboolean is_native_type(gulong type)
{
  gboolean ret;

  if(G_native_types == NULL)
    {
      ret = FALSE;
    }
  else
    {
      if(g_tree_lookup(G_native_types, (gpointer)type) == NULL)
        {
          ret = FALSE;
        }
      else
        {
          ret = TRUE;
        }
    }

  dprintf("%s: type = %ld, ret = %d\n", __FUNCTION__, type, ret);
  return ret;
}

/*
 * Функция вызывается для каждого узла дерева диссектора(анализатора) для
каждого
 * пакета. Определяет является ли узел одним из ключей (например
'freme.len'). Ищет узлы
 * хранения ключа PITEM_FINFO(node)->hfinfo->abbrev. Если искомый ключ обна-
ружен,
 * копирует исходное значение или строковое представление и переключается на
дочерние
 * узлы.
 */
static void proto_tree_get_node_field_values(proto_node *node, gpointer data)
{
  stdata_edt_tuple_t *args;
  field_info*fi;
  gpointer field_index;
  gpointer orig_key;
  gboolean key_found;

  args = data;
  fi = PITEM_FINFO(node);

  dprintf("fi->hfinfo->abbrev = %s\n", fi->hfinfo->abbrev);

  key_found = g_hash_table_lookup_extended(args->stdata->field_indicies,
                                           fi->hfinfo->abbrev,
                                           &orig_key,
                                           &field_index);

  const gchar* val_str;
  gulong actual_index = (gulong)(field_index);

  if(key_found)
    {
      gulong type = fi->hfinfo->type;

      if(type == FT_STRING)
        {
```

```
          dprintf("found a string!\n");
          dprintf("string is: %s\n", (char*)fvalue_get(&(fi->value)));
          dprintf("string as gnfvas: %s\n",
get_node_field_value_as_string(fi, args->edt));
        }

      if(type == FT_NONE)
        {
        args->stdata->field_values_str[actual_index] = NULL;
        args->stdata->field_values_native[actual_index] = 0;
        args->stdata->field_types[actual_index] = FT_NONE;
        }
      else if(is_native_type(type) == TRUE)
        {
          // Сохранять тип если возможно, не конвертировать в строку
        args->stdata->field_values_str[actual_index] = NULL;
          memcpy(args->stdata->field_values_native[actual_index], &(fi-
>value), sizeof(fvalue_t));
        args->stdata->field_types[actual_index] = type;
        }
      else
        {
          // Сохранять строку если не удается сохранить тип
          val_str = get_node_field_value_as_string(fi, args->edt);
        args->stdata->field_values_str[actual_index] = val_str;
        args->stdata->field_types[actual_index] = type;
        }
    }

  if (node->first_child != NULL)
    {
      proto_tree_children_foreach(node,
                           proto_tree_get_node_field_values, args);
    }
}

/* Возвращает строку ep_alloced или статическую константу */
static const gchar* get_node_field_value_as_string(field_info* fi, epan_dis-
sect_t* edt)
{
  if (fi->hfinfo->id == hf_text_only)
    {
      if (fi->rep)
      {
        return fi->rep->representation;
      }
      else
      {
        return get_field_hex_value2(edt->pi.data_src, fi);
      }
    }
#if 0
  /* Интерпретированные данные т.е.,"Data" протокол, используются вместо про-
токола */
  else if (fi->hfinfo->id == proto_data)
    {
      return get_field_hex_value2(edt->pi.data_src, fi);
    }
#endif
  /* Нормальные протоколы и поля */
  else
    {
      gchar       *dfilter_string;
      gint        chop_len;
```

```
      switch (fi->hfinfo->type)
      {
      case FT_PROTOCOL:
        /* Все детали протокола. */
        if (fi->rep)
          {
            return fi->rep->representation;
          }
        else
          {
            /* Аббревиатура протокола */
            return fi->hfinfo->abbrev;;
          }
      case FT_NONE:
        return NULL;
      default:

        dfilter_string = proto_construct_match_selected_string(fi,
                                                  edt);
        if (dfilter_string != NULL)
          {
            chop_len = strlen(fi->hfinfo->abbrev) + 4;

            /* Удалить двойные кавычки. */
            if (dfilter_string[strlen(dfilter_string)-1] == '"')
            {
              dfilter_string[strlen(dfilter_string)-1] = '\0';
              chop_len++;
            }

            return &(dfilter_string[chop_len]);
          }
        else
          {
            return get_field_hex_value2(edt->pi.data_src, fi);
          }
      }
    }
}
/* Переводим значения в hex-формат.
static const gchar* get_field_hex_value2(GSList* src_list, field_info *fi)
{
  const guint8 *pd;

  if (fi->length > tvb_length_remaining(fi->ds_tvb, fi->start))
    {
      return "field length invalid!";
    }

  pd = get_field_data(src_list, fi);

  if (pd)
    {
      int i;
      gchar* buffer;
      gchar* p;
      int len;
      const int chars_per_byte = 2;

      len = chars_per_byte * fi->length;
      buffer = ep_alloc_array(gchar, len + 1);
      buffer[len] = '\0';
      p = buffer;
```

```
      /* Простой hex-дамп */
      for (i = 0 ; i < fi->length; i++)
      {
        g_snprintf(p, len, "%02x", pd[i]);
        p += chars_per_byte;
        len -= chars_per_byte;
      }
      return buffer;
    }
  else
    {
      return NULL;
    }
}

/*
 * Функция получает значения всех полей пакета путем создания структуры дан-
ных для
 * передачи в proto_tree_children_foreach()находящийся в libwireshark, кото-
рый
 * последовательно анализирует дерево диссектора.
 *
 * @param stdata Sharktools структура данных,
 * @param edt дерево диссектора
 */
void proto_tree_get_fields(st_data_t* stdata, epan_dissect_t *edt)
{
  g_assert(stdata);
  g_assert(edt);

  stdata_edt_tuple_t arg;
  arg.stdata = stdata;
  arg.edt = edt;

  proto_tree_children_foreach(edt->tree,
                              proto_tree_get_node_field_values,
                          &arg);
}

/*
 * Проверка соответствия display filters (фильтров отображения) wireshark.
При
 * обнаружении соответствия вызываем функцию proto_tree_get_fields() для чте-
ния
 * необходимых полей в stdata.
 */
gboolean process_packet(capture_file *cf, gint64 offset, st_data_t *stdata)
{
  frame_data fdata;
  epan_dissect_t edt;
  gboolean passed;

  const struct wtap_pkthdr *whdr = wtap_phdr(cf->wth);
  union wtap_pseudo_header *pseudo_header = wtap_pseudoheader(cf->wth);
  const guchar *pd = wtap_buf_ptr(cf->wth);

  /*
   * Считает пакет.
   * Кадр диссектора используется для оперделения frame.number
   */
  cf->count++;

  /*
   * Инициализация дерева дисектора
```

```
   */
  epan_dissect_init(&edt, TRUE, TRUE);

  frame_data_init(&fdata, cf->count, whdr, offset, cum_bytes);

  frame_data_set_before_dissect(&fdata, &cf->elapsed_time,
                                &first_ts, &prev_dis_ts, &prev_cap_ts);

  passed = TRUE;

  if(cf->rfcode)
    {
      epan_dissect_prime_dfilter(&edt, cf->rfcode);
    }

  tap_queue_init(&edt);

  /*
   * Обработка данного пактеа диссектором
   */
  epan_dissect_run(&edt, pseudo_header, pd, &fdata, NULL);

  tap_push_tapped_queue(&edt);

  // Фильтр чтения
  if(cf->rfcode)
    {
      passed = dfilter_apply_edt(cf->rfcode, &edt);
    }
  else
    {
      passed = TRUE;
    }

  if(passed)
    {
      frame_data_set_after_dissect(&fdata, &cum_bytes, &prev_dis_ts);

      if(stdata != NULL)
        proto_tree_get_fields(stdata, &edt);
    }

  epan_dissect_cleanup(&edt);
  frame_data_cleanup(&fdata);

  return passed;
}

/*
 * Предотвращение ошибок динамического вызова в тех случаях, когда диссектор
wireshark
 * (т.е. библиотека) не может динамически вызвать libwireshark and libglib-
2.0.so и
 * необходима функция для их вызова.
 */
int sharktools_preload_libs(void)
{

#define LIBWIRESHARK "libwireshark.so"
#define LIBGLIB "libglib-2.0.so"

  int ret = 0;

  GModule* handle;
```

```
  handle = g_module_open(LIBWIRESHARK, 0);
  if(!handle)
    {
      dprintf("%s", g_module_error());
      ret = 1;
    }

  GModule *handle2;
  handle2 = g_module_open(LIBGLIB, 0);
  if(!handle2)
    {
      dprintf("%s", g_module_error());
      ret = 1;
    }
  return ret;
}

/*
 * Запись значений GTree FT_* которые можно преобразовать в native-тип
 * для того, чтобы избежать излишней обработки при преобразовании в строки
 */
void sharktools_register_native_types(GTree *_native_types)
{
  G_native_types = _native_types;
}

GCompareFunc sharktools_gulong_cmp(gconstpointer a, gconstpointer b)
{
  gulong x1;
  gulong x2;
  x1 = (gulong)a;
  x2 = (gulong)b;

  return (gpointer)(x1 - x2);
}

/*
 * Инициализация Sharktools
 *
 * Эта функция последовательно инициализирует Wireshark и связанные с ним ме-
ханизмы.
 */
int sharktools_init(void)
{
  if(strcmp(epan_get_version(), VERSION))
    {
      printf("ERROR: sharktools was compiled using version %s of lib-
wireshark.\n", VERSION);
      printf("However, the libwireshark installed on this system is version
%s.\n", epan_get_version());
      printf("Please recompile sharktools with headers from libwireshark ver-
sion %s,\n", epan_get_version());
      printf("or modify LD_LIBRARY_PATH to point to version %s of lib-
wireshark.\n", VERSION);
      printf("Consult sharktools' README file for more information on using a
different version of libwireshark.\n");
      return -1;
    }

  sharktools_preload_libs();

#if (WIRESHARK_1_0_0 || WIRESHARK_1_2_0 || WIRESHARK_1_4_0)
  /*
   * Получить сведения об учетных данных для последующего использования.
```

```
   */
  get_credential_info();
#endif

  dprintf("%s: initializing...\n", __FUNCTION__);

#if WIRESHARK_0_99_5
  epan_init(register_all_protocols, register_all_protocol_handoffs,
            failure_message, open_failure_message, read_failure_message);
#elif WIRESHARK_1_0_0
  epan_init(register_all_protocols, register_all_protocol_handoffs, NULL,
NULL,
            failure_message, open_failure_message, read_failure_message);
#elif (WIRESHARK_1_2_0 || WIRESHARK_1_4_0)
  epan_init(register_all_protocols, register_all_protocol_handoffs, NULL,
NULL,
            failure_message, open_failure_message, read_failure_message,
write_failure_message);
#endif

  G_native_types = NULL;

  dprintf("%s: initialized.\n", __FUNCTION__);

  return 0;
}


int sharktools_cleanup(void)
{
  dprintf("%s: called\n", __FUNCTION__);

  epan_cleanup();
  return 0;
}

/* Добавить строку decode_as в механизм wireshark */
long sharktools_add_decode_as(char *s)
{
  dprintf("%s: called\n", __FUNCTION__);

  return add_decode_as(s);
}

/* Удалить строку decode_as из механизма wireshark */
long sharktools_remove_decode_as(char *s)
{
  dprintf("%s: called\n", __FUNCTION__);

  return remove_decode_as(s);
}

/*
 * Возвращаем количество пакетов в <filename> для прохождения через фильтр
<dfilter>.
 * Это необходимо для оптимизации работы памяти в MATLAB.
 */
glong sharktools_count(char *filename, char *dfilter)
{
  capture_file cfile;
  gchar *cf_name = NULL;
  dfilter_t *rfcode = NULL;
  glong count = 0;
```

```
  dprintf("%s: entering...\n", __FUNCTION__);

  dprintf("%s: dfilter: %s\n", __FUNCTION__, dfilter);
  if(!dfilter_compile(dfilter, &rfcode))
    {
      sprintf(errmsg, "%s", dfilter_error_msg);
      printf("errmsg");
      if(rfcode)
        dfilter_free(rfcode);
      return -1;
    }

  cap_file_init(&cfile);

  cf_name = filename;

  // Открыть pcap файл
  int err;
  if(cf_open(&cfile, cf_name, FALSE, &err) != CF_OK)
    {
        if(rfcode)
        dfilter_free(rfcode);
      return -1;
    }

  dprintf("%s: opened file\n", __FUNCTION__);

  cfile.rfcode = rfcode;


  gchar        *err_info;
  gint64       data_offset;

  // Читать и обрабатывать каждый пакет по одному
  while(wtap_read(cfile.wth, &err, &err_info, &data_offset))
    {
      gboolean passed = TRUE;

      // Обрабатывать пакеты только в члучае если указан фильтр
      if(dfilter != NULL && *dfilter != '\0')
        {
          // Передача NULL в st_data_t означает просто подсчет пакетов
          passed = process_packet(&cfile, data_offset, NULL);
        }

      if(passed)
      {
          count++;
        }
    }

  if(rfcode)
    dfilter_free(rfcode);
  wtap_close(cfile.wth);
  cfile.wth = NULL;

  return count;
}

long sharktools_read_data(char *filename, char *dfilter, GArray *offsets,
GArray *stdatas, gulong nfields, const gchar **fields)
{
  capture_file cfile;
  gchar *cf_name = NULL;
```

```
  dfilter_t *rfcode = NULL;

  dprintf("%s: entering...\n", __FUNCTION__);

  dprintf("%s: dfilter: %s\n", __FUNCTION__, dfilter);
  if(!dfilter_compile(dfilter, &rfcode))
    {
      sprintf(errmsg, "%s", dfilter_error_msg);
      printf("errmsg");
      if(rfcode)
        dfilter_free(rfcode);
      return -1;
    }

  cap_file_init(&cfile);

  cf_name = filename;


  int err;
  if(cf_open(&cfile, cf_name, FALSE, &err) != CF_OK)
    {
      if(rfcode)
        dfilter_free(rfcode);
      return -1;
    }

  dprintf("%s: opened file\n", __FUNCTION__);

  cfile.rfcode = rfcode;

  gchar        *err_info;
  gint64       data_offset;

  gsize i;
  int ipacket = 0;

    while(wtap_read(cfile.wth, &err, &err_info, &data_offset))
    {
      gboolean passed = TRUE;

      st_data_t stdata;

      stdata_init(&stdata, nfields);

      stdata_add_fields(&stdata, fields, nfields);

      dprintf("stdata.fields->len = %d\n", stdata.fields->len);

      dprintf("stdata.field_values_str = %lX\n", (glong)stdata.field_val-
ues_str);
      dprintf("stdata.field_types = %lX\n", (glong)stdata.field_types);

      for(i = 0; i < nfields; i++)
      {
        stdata.field_values_str[i] = 0;
        stdata.field_types[i] = FT_NONE;
      }


      if(dfilter != NULL && *dfilter != '\0')
        {
          // Передаем данные пакета в st_data_t struct
          passed = process_packet(&cfile, data_offset, &stdata);
```

```
        }

      if(passed)
      {
        g_array_append_val(offsets, ipacket);

        // Добавить данные в структуру
        g_array_append_val(stdatas, stdata);
        }

      ipacket++;
    }

  if(rfcode)
    dfilter_free(rfcode);
  wtap_close(cfile.wth);
  cfile.wth = NULL;

  return 0;
}

/*
 * Обработка выбранного файла с помощью фильтров отображения (display fil-
ters).
 * Обращение к функции обратного вызова <cb> для изменение структуры данных
 * запрошенной области
 *
 * @param nthreads число потоков
 * @param filename имя pcap файла
 * @param nfields положительное целое число, описываюзее количество полей
 * @param fields массив строк
 * @return 0 при успехе, если нет - ошибка.
 */
glong sharktools_get_cb_threads(int nthreads, gchar *filename, gulong
nfields, const gchar **fields,
                          gchar *dfilterorig, GArray *offsets, GArray *stdatas,
sharktools_callbacks *cb)
{

  capture_file cfile;
  gchar *cf_name = NULL;
  char *dfilter;
  dfilter_t *rfcode = NULL;

  dprintf("%s: entering...\n", __FUNCTION__);

  dprintf("%s: dfilterorig: %s\n", __FUNCTION__, dfilterorig);

  dfilter = strdup(dfilterorig);

  dprintf("%s: dfilter: %s\n", __FUNCTION__, dfilter);

  if(!dfilter_compile(dfilter, &rfcode))
    {
      sprintf(errmsg, "%s", dfilter_error_msg);
      printf("errmsg");
      if(rfcode)
        dfilter_free(rfcode);
      return -1;
    }

  cap_file_init(&cfile);

  cf_name = filename;
```

```
  int err;
  if(cf_open(&cfile, cf_name, FALSE, &err) != CF_OK)
    {
      if(rfcode)
        dfilter_free(rfcode);
      return -1;
    }

  dprintf("%s: opened file\n", __FUNCTION__);

  cfile.rfcode = rfcode;

  gchar        *err_info;
  gint64       data_offset;

  int ipacket = 0;
  int ioffset = 0;

  g_thread_init(NULL);

  /*
   * Создание пула потоков.
   * process_packet_callback вызывается каждым потоком
   * cb переменная общая для всех потоков
   */
  GThreadPool *thread_pool;
  thread_pool = g_thread_pool_new((GFunc)process_packet_callback, (gpointer
*)cb, nthreads, TRUE, NULL);

  thread_arguments *args = NULL;

  gboolean count_only;

  // Чтение данных пакета из файла
  int npackets = stdatas->len;

  int i;
  for(i = 0; i < npackets; i++)
    {
      /* Аргументы и потоки */
      args = malloc(sizeof(thread_arguments));
      args->nfields = nfields;
      args->ipacket = i;
      args->stdata = g_array_index(stdatas, st_data_t, i);

      // Объявление и запуск нового потока
      g_thread_pool_push(thread_pool, (gpointer)args, NULL);
    }

  // Завершение потоков
  while (g_thread_pool_unprocessed(thread_pool))
    nanosleep(&ts, NULL);

  g_thread_pool_free(thread_pool, TRUE, TRUE);

  if(rfcode)
    dfilter_free(rfcode);
  wtap_close(cfile.wth);
  cfile.wth = NULL;

  dprintf("%s: ...leaving.\n", __FUNCTION__);

  return 0;
```

```
}

void process_packet_callback(gpointer arguments, gpointer callbacks)
{
  thread_arguments *args = (thread_arguments*)arguments;
  sharktools_callbacks *cb = (sharktools_callbacks*)callbacks;

  static GMutex *mutex = NULL;
  if (!mutex)
    mutex = g_mutex_new();
  g_mutex_lock(mutex);

  gpointer row = (gpointer)args->ipacket;

  int i;
  for(i = 0; i < args->nfields; i++)
    {
      gpointer key;
      key = cb->keys[i];

      dprintf("key = %p\n", key);

      dprintf("values[%ld] = %p\n", i, args->stdata.field_values_str[i]);
      dprintf("types[%ld] = %ld\n", i, args->stdata.field_types[i]);

  args->stdata.field_values_str[i], args->stdata.field_types[i]);

      cb->row_set(cb, row, key,
              args->stdata.field_types[i],
              args->stdata.field_values_native[i],
              args->stdata.field_values_str[i]
              );

    }

  cb->row_add(cb, row);

  g_mutex_unlock(mutex);

  nanosleep(&ts, NULL);

}

glong sharktools_get_cb(gchar *filename, gulong nfields, const gchar
**fields,
                        gchar *dfilterorig, sharktools_callbacks *cb)
{
  gsize i;
  capture_file cfile;
  gchar *cf_name = NULL;
  char *dfilter;
  dfilter_t *rfcode = NULL;

  // Создать структуру stdata в стеке
  st_data_t stdata;

  dprintf("%s: entering...\n", __FUNCTION__);

  dprintf("%s: dfilterorig: %s\n", __FUNCTION__, dfilterorig);

  dfilter = strdup(dfilterorig);

  dprintf("%s: dfilter: %s\n", __FUNCTION__, dfilter);
```

```
if(!dfilter_compile(dfilter, &rfcode))
  {
    sprintf(errmsg, "%s", dfilter_error_msg);
    printf("errmsg");
    if(rfcode)
      dfilter_free(rfcode);
    return -1;
  }

  cap_file_init(&cfile);

cf_name = filename;

int err;
if(cf_open(&cfile, cf_name, FALSE, &err) != CF_OK)
  {
    if(rfcode)
      dfilter_free(rfcode);
    return -1;
  }

dprintf("nfields = %ld\n", nfields);

stdata_init(&stdata, nfields);

stdata_add_fields(&stdata, fields, nfields);

dprintf("stdata.fields->len = %d\n", stdata.fields->len);

dprintf("stdata.field_values_str = %lX\n", (glong)stdata.field_values_str);
dprintf("stdata.field_types = %lX\n", (glong)stdata.field_types);

dprintf("%s: opened file\n", __FUNCTION__);

cfile.rfcode = rfcode;

gchar        *err_info;
gint64       data_offset;

  while(wtap_read(cfile.wth, &err, &err_info, &data_offset))
  {
    dprintf("******************************\n");

    for(i = 0; i < nfields; i++)
      {
        stdata.field_values_str[i] = 0;
        stdata.field_types[i] = FT_NONE;
      }

    gboolean passed = FALSE;

    passed = process_packet(&cfile, data_offset, &stdata);

    if(passed)
    {
        gpointer row = cb->row_new(cb);

      for(i = 0; i < nfields; i++)
        {
            gpointer key;
            key = cb->keys[i];

            dprintf("key = %p\n", key);
```

```
            dprintf("values[%ld] = %p\n", i, stdata.field_values_str[i]);
            dprintf("types[%ld] = %ld\n", i, stdata.field_types[i]);

            i, stdata.field_values_str[i], stdata.field_types[i]);
              cb->row_set(cb, row, key,
                          stdata.field_types[i],
                          stdata.field_values_native[i],
                          stdata.field_values_str[i]
                          );
            }

          cb->row_add(cb, row);
        }
    }

  if(rfcode)
    dfilter_free(rfcode);
  wtap_close(cfile.wth);
  cfile.wth = NULL;

  stdata_cleanup(&stdata);

  dprintf("%s: ...leaving.\n", __FUNCTION__);

  return 0;
}
```

1.2 matshark_threads.c

```
//
//  File: matshark_threads.c
//
//  Abstract: This software is intended for use for research purposes
//            to analyze modern large backbone with ability to use CPU
//            based parallel computing
//
//  Version: <1.0>
//
//  Copyright (c) 2014 Daniil Bozhalkin. All rights reserved.
//
////////////////////////////////////////////////////////////////////////

#include <mex.h>
#include "matrix.h"
#include <sys/types.h>
#include <string.h>
#include <strings.h>
#include <glib.h>
#include "sharktools_core.h"
#include "sharktools_stdata.h"

#define BUFSIZE 1024

#define DEBUG 0
#if DEBUG
#define dprintf(args...) printf(args)
#else
#define dprintf(args...) ((void)0)
#endif

extern char sharktools_errmsg[2048];

#if DEBUG==0
static void log_func_ignore (const gchar *log_domain, GLogLevelFlags
log_level,
                        const gchar *message, gpointer user_data)
{
}
#endif

static int initialized = FALSE;

gpointer cb_row_new(sharktools_callbacks *cb)
{
  dprintf("%s: entering\n", __FUNCTION__);
  dprintf("%s: leaving\n", __FUNCTION__);

  return (gpointer)(cb->count);

}

gpointer cb_row_set(sharktools_callbacks *cb, gpointer row, gpointer key,
gulong type, fvalue_t *val_native, const gchar *val_string)
{
  dprintf("%s: entering\n", __FUNCTION__);

  long li = (long)row;
  long *lj = (long*)key;

  mxArray *mx = (mxArray *)cb->root;
  int i = (int)li;
```

```
  int j = (int)*lj;

  dprintf("mx = %p ; i = %d ; j = %X\n", mx, i, j);

  mxArray *obj = NULL;

  static unsigned long tmp_unsigned_long;
  static unsigned long long tmp_unsigned_long_long;
  static long tmp_long;
  static double tmp_double;
  static nstime_t *tmp_timestamp;

  int ndim;
  int *dims;

  /*
   * MATLAB не делает различия для следуюзих типов данных double, boolean,
unsigned
   * integers и signed integers.  MATLAB использует 64-битные числа с плавающ
ей точкой
   * (mxREALs) для их представления. Поэтому необходимо конвертировать типы
данных
   * Wireshark в native-типы данных C, затем преобразовать в double перед пе
редачей в
   * MATLAB.
   */

  switch(type)
    {
    case FT_NONE:  /* используется для текстовых меток без значения */
      /* Создание "[]" в MATLAB: */
      dprintf("NONE\n");
      ndim = 0;
      dims = NULL;
      obj = mxCreateNumericArray(ndim, dims, mxUINT32_CLASS, mxREAL);
      break;

    case FT_BOOLEAN:     /* TRUE и FALSE берутся из <glib.h> */
      /* Wireshark считает boolean как uintegers. */
    case FT_UINT8:
    case FT_UINT16:
    case FT_UINT24:      /* на самом деле UINT32, но отображается как 3 hex-
знака при FD_HEX*/
    case FT_UINT32:
      tmp_unsigned_long = fvalue_get_uinteger(val_native);
      tmp_double = tmp_unsigned_long; /* конвертация из long в double */
      obj = mxCreateDoubleMatrix(1, 1, mxREAL);
      *mxGetPr(obj) = tmp_double;
      break;

    case FT_INT8:
    case FT_INT16:
    case FT_INT24:
    case FT_INT32:
      tmp_long = fvalue_get_sinteger(val_native);
      tmp_double = tmp_long; /* конвертация из long в double */
      obj = mxCreateDoubleMatrix(1, 1, mxREAL);
      *mxGetPr(obj) = tmp_double;
      break;

    case FT_INT64:
      /* Wireshark не делает разницы между INT64 и UINT64 */
    case FT_UINT64:
      tmp_unsigned_long_long = fvalue_get_integer64(val_native);
```

```
      tmp_double = tmp_unsigned_long_long; /* конвертация из uint64 в double
*/
      obj = mxCreateDoubleMatrix(1, 1, mxREAL);
      *mxGetPr(obj) = tmp_double;
      break;

    case FT_FLOAT:
    case FT_DOUBLE:
      tmp_double = fvalue_get_floating(val_native);
      obj = mxCreateDoubleMatrix(1, 1, mxREAL);
      *mxGetPr(obj) = tmp_double;
      break;

    case FT_ABSOLUTE_TIME:
    case FT_RELATIVE_TIME:
      tmp_timestamp = fvalue_get(val_native);
      /* Используя функцию в $wireshark/epan/nstime.c для конфертации
timestamp в float */
      tmp_double = nstime_to_sec(tmp_timestamp);
      obj = mxCreateDoubleMatrix(1, 1, mxREAL);
      *mxGetPr(obj) = tmp_double;
      break;

#if 0
#endif
    default:
      obj = mxCreateString(val_string);
      break;
    }

  mxSetFieldByNumber(mx, i, j, obj);

  dprintf("%s: leaving\n", __FUNCTION__);

  return NULL;
}

gpointer cb_row_add(sharktools_callbacks *cb, gpointer row)
{
  dprintf("%s: entering\n", __FUNCTION__);
  int rownum;
  rownum = (long)row;

  cb->count++;

  if(cb->count != rownum)
    {
      dprintf("cb->count = %d, rownum = %d\n", cb->count, rownum);
    }

  dprintf("%s: leaving\n", __FUNCTION__);

  return NULL;
}

void mexFunction(int nlhs, mxArray *plhs[],
             int nrhs, const mxArray *prhs[]) {
  int m, i;
  mxArray *tmp;
  int nthreads;
  char *filename;
  char **fieldnames;
  char **mx_fieldnames;
  int nfields;
```

```
  char *dfilter;
  char *decode_as = NULL;

  int ret;

#if DEBUG==0
  GLogLevelFlags       log_flags;

  log_flags =
    G_LOG_LEVEL_ERROR|
    G_LOG_LEVEL_CRITICAL|
    G_LOG_LEVEL_WARNING|
    G_LOG_LEVEL_MESSAGE|
    G_LOG_LEVEL_INFO|
    G_LOG_LEVEL_DEBUG|
    G_LOG_FLAG_FATAL|G_LOG_FLAG_RECURSION;


  g_log_set_handler(NULL,
                log_flags,
                log_func_ignore, NULL );

  g_log_set_handler ("GLib", G_LOG_LEVEL_MASK | G_LOG_FLAG_FATAL,
                 log_func_ignore, NULL);

#if 0
  g_log_set_handler(LOG_DOMAIN_CAPTURE_CHILD,
                    log_flags,
                    log_func_ignore, NULL );
#endif
#endif

  dprintf("%s: entering...\n", __FUNCTION__);

  /* Регистрация всех диссекторов(только один раз) */
  if(!initialized)
    {
      sharktools_init();

      /*
       * Создать бинарное дерево с типами данных Wireshark которые могут быть
       * использованы в качестве типов данных MATLAB
       */
      GTree *native_types = g_tree_new((GCompareFunc)sharktools_gulong_cmp);

      gsize i;

      gulong native_type_array[] = { FT_BOOLEAN,
                                     FT_UINT8,
                                     FT_UINT16,
                                     FT_UINT24,
                                     FT_UINT32,
                                     FT_INT8,
                                     FT_INT16,
                                     FT_INT24,
                                     FT_INT32,
                                     FT_INT64,
                                     FT_UINT64,
                                     FT_FLOAT,
                                     FT_DOUBLE,
                                     FT_ABSOLUTE_TIME,
                                     FT_RELATIVE_TIME};
```

```
      gulong native_type_array_size = (sizeof(native_type_array)/sizeof(na-
tive_type_array[0]));

      /* Используем только ключи, а не их значения */
      gulong dummy_value = 1;

      for(i = 0; i < native_type_array_size; i++)
        {
          g_tree_insert(native_types, (gpointer)native_type_array[i],
(gpointer)dummy_value);
        }

      dprintf("native_types height = %d\n", g_tree_height(native_types));

      /* Записываем native_types с помощью механизма sharktools*/
      sharktools_register_native_types(native_types);

      initialized = TRUE;
    }

  if(nrhs < 4)
    mexErrMsgTxt("Must provide number of threads, filename, cell array of
fieldnames, and display filter");

  if(mxIsChar(prhs[1]) != 1)
    mexErrMsgTxt("2nd arg (filename) must be a string");

  if(mxIsCell(prhs[2]) != 1)
    mexErrMsgTxt("3rd arg (fields) must be a cellarray of strings");

  if(mxIsChar(prhs[3]) != 1)
    mexErrMsgTxt("4th arg (filter) must be a string");

  if(nrhs == 5)
    {
      if(mxIsChar(prhs[4]) != 1)
        mexErrMsgTxt("5th arg (decode_as rule) must be a string");

      decode_as = mxCalloc(BUFSIZE, sizeof(char));
      if(mxGetString(prhs[4], decode_as, BUFSIZE))
        mexErrMsgTxt("error getting decode_as string");

      dprintf("decode_as = %s\n", decode_as);

      sharktools_add_decode_as(decode_as);
    }

  nthreads = mxCalloc(1, sizeof(int));
  nthreads = (int)(mxGetScalar(prhs[0]));
  dprintf("number of threads = %d\n", nthreads);

  filename = mxCalloc(BUFSIZE, sizeof(char));
  if(mxGetString(prhs[1], filename, BUFSIZE))
    mexErrMsgTxt("error getting pcap filename string");

  dprintf("pcap filename = %s\n", filename);

  /* Получить количество полей */
  nfields = mxGetNumberOfElements(prhs[2]);
  dprintf("nfields = %d\n", nfields);

  fieldnames = g_new(char*, nfields);
  mx_fieldnames = mxCalloc(nfields, sizeof(char*));
```

```
  for(i = 0; i < nfields; i++)
    {
      mxArray *tmp;
      tmp = mxGetCell(prhs[2], i);

      /*
       * Получить копию имен полей для sharkcore
       */
      fieldnames[i] = g_new(char, BUFSIZE);

      if(mxGetString(tmp, fieldnames[i], BUFSIZE))
      mexErrMsgTxt("error getting a field string");

      /*
       * Получить копию имен полей для Matlab
       */
      mx_fieldnames[i] = mxCalloc(BUFSIZE, sizeof(char));

      if(mxGetString(tmp, mx_fieldnames[i], BUFSIZE))
      mexErrMsgTxt("error getting a field string");

      /*
       * В копии имен полей MATLAB, заменить "." на "_" (В MATLAB, '.'
       * недопустимый символ в имени поля)
       */
      char *s;
      s = mx_fieldnames[i];
      char *c = NULL;
      while((c = strchr(s, '.')))
      {
        *c = '_';
      }
    }

  dfilter = mxCalloc(BUFSIZE, sizeof(char));
  if(mxGetString(prhs[3], dfilter, BUFSIZE))
    mexErrMsgTxt("error getting filter string");

  dprintf("filter = %s\n", dfilter);

  /*
   * Matlab не имеет эффективного механизма для динамического расширения
структуры
   * данных. Используем sharktools_count() для прохода pcap файла и подсчета
числа
   * записей которое необходимо прописать при создании объекта и возвращаем
его
   * интерпретатору MATLAB.
   */

  GArray *offsets = g_array_new(FALSE, TRUE, sizeof(int));
  GArray *stdatas = g_array_new(FALSE, TRUE, sizeof(st_data_t));

  int status;
  status = sharktools_read_data(filename, dfilter, offsets, stdatas, nfields,
(const gchar**)fieldnames);

  int count;
  count = stdatas->len;
  dprintf("count = %d\n", count);

  if(count < 0)
    {
      mexErrMsgTxt(sharktools_errmsg);
```

```
    }

  m = 1;
  mxArray *mx;
  mx = mxCreateStructMatrix(m, count, nfields, (const char**) mx_fieldnames);

  /*
   * Создаем список ключей
   */
  int **keys = g_new(int*, nfields);
  for(i = 0; i < nfields; i++)
    {
      keys[i] = g_new(int, 1);
      *keys[i] = i;
    }

  /*
   * Строим cb "object" с переменными состояния и функцией обратного вызова
   */
  sharktools_callbacks cb;
  cb.root = (gpointer)mx;
  cb.keys = (gpointer *)keys;
  cb.count = 0;
  cb.row_new = cb_row_new;
  cb.row_set = cb_row_set;
  cb.row_add = cb_row_add;

  ret = sharktools_get_cb_threads(nthreads, filename, nfields, (const
gchar**)fieldnames, dfilter, offsets, stdatas, &cb);

  g_array_free(offsets, FALSE);
  g_array_free(stdatas, FALSE);

  if(!mexIsLocked())
    {
      /* Блокируем функцию, чтобы статические переменные не были очищены в
MATLAB */
      mexLock();
    }

  if(decode_as)
    {
      sharktools_remove_decode_as(decode_as);
    }

  /*
   * Очистить память
   */
  for(i = 0; i < nfields; i++)
    {
      g_free(keys[i]);
      g_free(fieldnames[i]);
    }

  g_free(keys);
  g_free(fieldnames);

  if(ret)
    {
      mexErrMsgTxt(sharktools_errmsg);
    }

  plhs[0] = mx;
}
```

1.3 sharktools_stdata.h

```
//
//  File: matshark_threads.c
//
//  Abstract: This software is intended for use for research purposes
//            to analyze modern large backbone with ability to use CPU
//            based parallel computing
//
//  Version: <1.0>
//
//  Copyright (c) 2014 Daniil Bozhalkin. All rights reserved.
//
////////////////////////////////////////////////////////////////////////

#ifndef SHARKTOOLS_STDATA
#define SHARKTOOLS_STDATA

#include <glib.h>

/*
 * Эта структура содержит специфические данные Sharktools которые которые
проходят через
 * систему обратного вызова libwireshark.
 *
 * Fields и field_indices расчитываются один раз при прогоне sharktools, а
field_values
 * and field_types обновляются при обработке каждого пакета
 */
typedef struct
{
  /*
   * 'fields' содержит упорядоченный список запрашиваемых ключей
('frame.number' or
   * 'ip.len')
   */
  GPtrArray* fields;

  /*
   * 'field_indicies' содержит хеш(ключ), который определяет порядковый номер
ключа
   * (порядок важен, так как разные пакеты имеют разное время обработки)
   */
  GHashTable* field_indicies;

  /*
   * 'field_values_str' содержит упорядоченный список значений (строк) найде-
ных для
   * конкретного пакета.
   */
  const gchar** field_values_str;

  fvalue_t **field_values_native;

  /*
   * 'field_types' содержит упорядоченный список типов данных
   * для каждого соответствующего значения в 'field_values_str'
   */
  gulong *field_types;
} st_data_t;

#endif
```

ПРИЛОЖЕНИЕ Б. Распределение случайных последовательностей с ограниченной областью рассеяния

НР случайной величины имеет неограниченную область рассеяния. В тоже время случайные величины, анализ которых приходится проводить в естествознании, технике и экономике, имеют конечную область рассеяния. (Далее будем называть случайные числа данного типа – случайными последовательностями с ограниченной областью рассеяния (СПООР).) Априори, понятно, что ФР и ПР СПООР, будут отличаться от аналогичных величин НР (1.3).

В качестве примеров СПООР можно привести плотности углей, изменяющиеся в диапазоне от ρ_{min} до некоторого максимального значения ρ_{max}, время безотказной работы группы однотипных приборов, изменяющегося в диапазоне от некоторого минимального значения T_{min} до некоторого максимального T_{max} и т.д.

Наиболее очевидной физической моделью СПООР служат песочные часы, в которых песок из «точечного» источника высыпается на горизонтальную плоскость, ограниченную непроницаемыми абсолютно упругими вертикальными стенками. При достаточном удалении стенок от источника, понятно, что они не будут оказывать влияния на формирование кучи песка, поэтому ее форма будет симметричной. В противоположном случае песчинки будут отражаться от стенок, что приведет к отличию формы кучи от «нормальной».

Анализ работ, посвященных построению моделей ФР и ПР СПООР, показывает, что существует два альтернативных подхода. Первый подход основан на описании ФР и ПР с помощью УНР. Во втором подходе, базирующемся на работах А. Эйнштейна и М. Смолуховского по теории броуновского движения, модель ФР и ПР СПООР строится как распределение конечного состояния некоторого случайного процесса без последействия с ограниченной областью рассеяния. Отметим, что сегодня в подавляющем большинстве учебников по теории надежности технических систем используется первый подход, как правило, без каких-либо обоснований подобного выбора. В этой связи сравнение обоих подходов к построению математических моделей ФР и ПР СПООР с точки зрения адекватности изучаемым физическим процессам является актуальной.

ПР СПООР в рассматриваемом случае можно найти, проведя аналогию между рассматриваемой случайной величиной и одномерным броуновским движением частицы в ограниченной области. Особенности статистических свойств данного движения были изучены М. Смолуховским.

Рассмотрим, решение обсуждаемой задачи для области рассеяния, ограниченной с правой стороны отражающей стенкой, расположенной в точке $x_{\max}$. Искомая вероятность нахождения броуновской частицы в точке с координатой *m* в рассматриваемом случае вычисляется по формуле:

$$P_n(m) \sim \frac{1}{\sqrt{2\pi n}} \left\{ e^{-\frac{m^2}{2n}} + e^{-\frac{(2x_{\max}-m)^2}{2n}} \right\}, \qquad (0.1)$$

где $x_{\max}$ – координата отражающей стенки, расположенной справа от источника, n – количество случайных толчков.

Из (1.10) видно, что вероятность нахождения броуновской частицы в произвольной точке *m*, может быть интерпретирована, как сложение интенсивностей двух источников (действительного и фиктивного), расположенных в точках $x_1 = 0,\ x_2 = 2x_{\max} - m,$ соответственно.

Рассуждая аналогично, можно показать, что для броуновского движения с ограниченной областью рассеяния слева в точке $x_{\min}$ вероятность нахождения броуновской частицы в точке с координатой *m* в данном случае вычисляется по формуле:

$$P_n(m) \sim \frac{1}{\sqrt{2\pi n}} \left\{ e^{-\frac{m^2}{2n}} + e^{-\frac{(m-2x_{\min})^2}{2n}} \right\}, \qquad (0.2)$$

где $x_{\min}$ – координата отражающей стенки, расположенной слева от источника, n – количество случайных толчков.

Из формулы (1.11) видно, что вероятность нахождения броуновской частицы в произвольной точке *m*, может быть интерпретирована, как сложение интенсивностей двух источников (действительного и фиктивного), расположенных в точках $x_1 = 0,\ x_2 = m - 2x_{\min},$ соответственно.

В связи с тем, что при практическом использовании модели броуновского движения в ограниченной области одним из основных оказывается вопрос о вычислении координат точек расположения фиктивных источников, рассмотрим его более подробно. Выберем систему координат с началом в середине отрезка области рассеяния (рисунок 1).

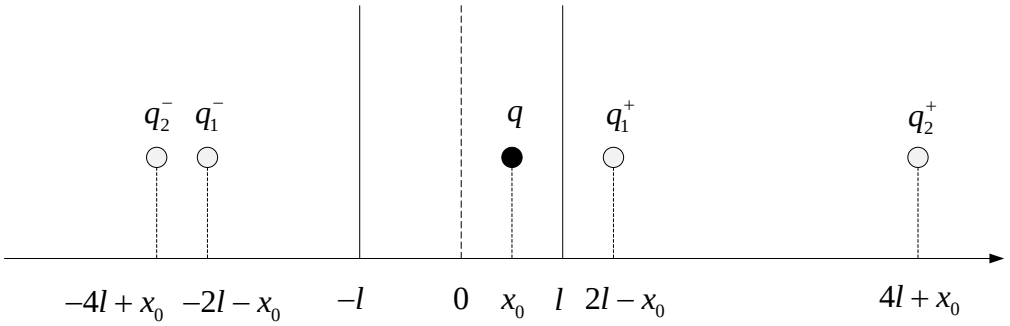


Рис. 1. К вычислению координат фиктивных источников

Из рисунка 1 видно, что наличие двух отражающих поверхностей порождает бесконечную систему фиктивных источников. Действительно, фиктивный источник q_g^+, полученный отражением относительно плоскости $x = +l$ и находящийся справа от нее, в свою очередь отражается относительно плоскости $x = -l$, формируя фиктивный источник $q_{g+1}^- = q_g^+$. И аналогично для фиктивного источника q_g^-. Дополнив эти соотношения правилами преобразования координат фиктивных источников $q_g^\pm$, получаем:

$$\begin{aligned} q_{g+1}^+ &= q_g^-, \quad x_{g+1}^+ - l = -\left(x_g^- - l\right), \\ q_{g+1}^- &= q_g^+, \quad x_{g+1}^- + l = -\left(x_g^+ + l\right). \end{aligned} \tag{0.3}$$

Принимая во внимание «начальные условия» $q_0^+ = q_0^- = q, \quad x_0^+ = x_0^- = x_0$, из (1.12) получаем:

$$x_{2g}^\pm = \pm 4gl + x_0, \quad x_{2g+1}^\pm = \pm\left(4g + 2\right)l - x_0, \tag{0.4}$$

где $g = 0,1,\ldots$, по которым можно вычислить координаты любого из мнимых источников.

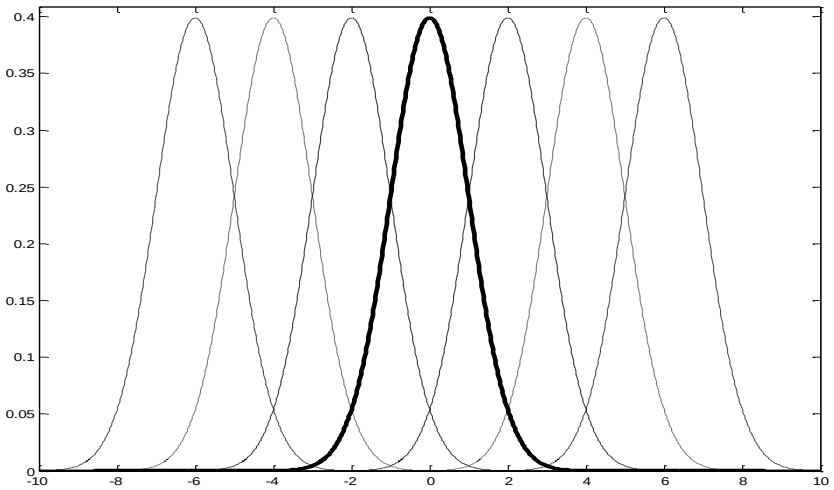


Рис. 2. ПР реального и фиктивных источников. $l = 1,\ x_0 = 0,\ x_1^+ = 2,\ x_1^- = -2,\ x_2^+ = 4,$
$x_2^- = -4,\ x_3^+ = 6,\ x_3^- = -6$

Таким образом, используя раздельное суммирование по четным и нечетным индексам, можно записать следующее выражение для ПР СПООР:

$$f_{LAD}\left(x;x_0,\sigma,l\right)=A\left[\varphi\left(x;x_0,\sigma,l\right)+\sum_{g=0}^{\infty}\varphi^{\pm}_{2g+1}\left(x;x_0,\sigma,l\right)+\sum_{g=1}^{\infty}\varphi^{\pm}_{2g}\left(x;x_0,\sigma,l\right)\right], \quad (0.5)$$

где A – нормировочный коэффициент, определяемый из условия

$$\int_a^b f_{LAD}\left(\xi;x_0,\sigma,l\right)d\xi=1, \quad (0.6)$$

$$\varphi\left(x;x_0,\sigma,l\right)=\exp\left[-\left(x-x_0\right)^2/2\sigma^2\right],$$

$$\varphi^{\pm}_{2g+1}\left(x;x_0,\sigma,l\right)=\exp\left[-\left(x-x^{\pm}_{2g+1}\right)^2/2\sigma^2\right],$$

$$\varphi^{\pm}_{2g}\left(x;x_0,\sigma,l\right)=\exp\left[-\left(x-x^{\pm}_{2g}\right)^2/2\sigma^2\right],$$

здесь $x^{\pm}_{2g+1}, x^{\pm}_{2g}$ вычисляются в соответствие с (0.4).

Соответственно, ФР СПООР вычисляется по формуле

$$F_{LAD}\left(x;x_0,\sigma,l\right)=\int_a^x f_{LAD}\left(\xi;x_0,\sigma,l\right)d\xi. \quad (0.7)$$

Необходимо отметить, что плотность распределения нормальной случайной величины (0.5), вообще говоря, формируется бесконечной системой фиктивных источников. Однако для расчетов на ЭВМ достаточно некоторого конечного числа фиктивных источников. На рисунке 3 представлена зависимость

$$\varepsilon\left(N,\mu,\sigma,x_{\min},x_{\max}\right)=\ln\left[\max\left(\left|f_{LAD}\left(x;\mu,\sigma,x_{\min},x_{\max},N\right)-\right.\right.\right.$$
$$\left.\left.\left.-f_{LAD}\left(x;\mu,\sigma,x_{\min},x_{\max},100\right)\right|\right)\right],$$

вычисленная для следующих значений параметров: $\mu=0,\sigma=0,x_{\min}=-5,x_{\max}=5.$

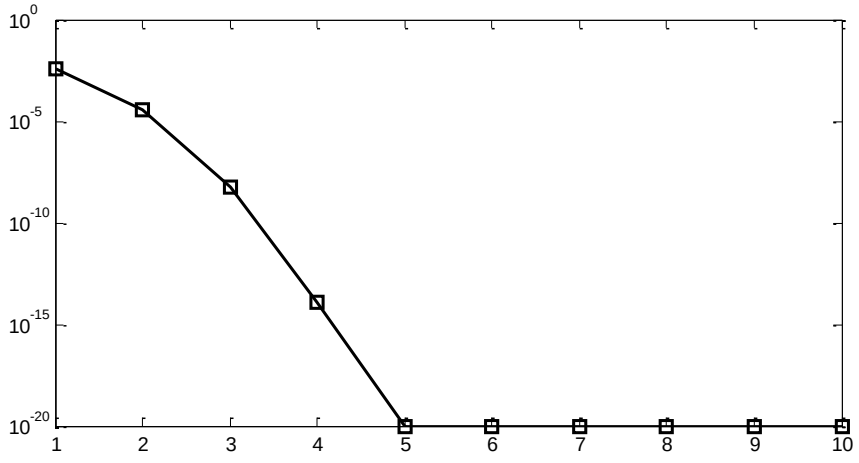


Рис. 3. График зависимости $\varepsilon\left(N,0,10,-5,5\right)$

Как видно из рисунка 3, достаточная точность вычисления плотности вероятности в соответствие с (0.5) достигается при $N = 5$ пар мнимых источников (соответственно, число мнимых источников – $2N = 10$), а потому указанную величину можно не оценивать с помощью ГА.

Характеристики рассматриваемого распределения и соответствующие формулы для их вычисления представлены в таблице 1.

Таблица 1.

Характеристики ПР СПООР

Название характеристики	*Формула или числовое значение характеристики*
Обозначение	$LAD\left(x_0, \sigma, x_{\min}, x_{\max}\right)$
Параметры	$\mu, \sigma, x_{\min}, x_{\max}$
Плотность распределения	$f_{LAD}\left(x; x_0, \sigma, x_{\min}, x_{\max}\right) = A\Big[\varphi\left(x; x_0, \sigma, x_{\min}, x_{\max}\right) + \sum_{g=0}^{\infty} \varphi^{\pm}_{2g+1}\left(x; x_0, \sigma, x_{\min}, x_{\max}\right) + \sum_{g=1}^{\infty} \varphi^{\pm}_{2g}\left(x; x_0, \sigma, x_{\min}, x_{\max}\right)\Big]$, где A – нормировочный коэффициент, определяемый из условия $\int_{x_{\min}}^{x_{\max}} f_{LAD}\left(\xi; x_0, \sigma, x_{\min}, x_{\max}\right) d\xi = 1$, $\varphi\left(x; x_0, \sigma, x_{\min}, x_{\max}\right) = \exp\left[-\left(x-\mu\right)^2 / 2\sigma^2\right]$, $\varphi^{\pm}_{2g+1}\left(x; x_0, \sigma, x_{\min}, x_{\max}\right) = \exp\left[-\left(x - x^{\pm}_{2g+1}\right)^2 / 2\sigma^2\right]$, $\varphi^{\pm}_{2g}\left(x; x_0, \sigma, x_{\min}, x_{\max}\right) = \exp\left[-\left(x - x^{\pm}_{2g}\right)^2 / 2\sigma^2\right]$, $x^{\pm}_{2g} = \pm 4g\left(x_{\max} - x_{\min}\right) + \mu$, $x^{\pm}_{2g+1} = \pm\left(4g+2\right)\left(x_{\max} - x_{\min}\right) - \mu$, $x_{\min} < x_{\max}$ – границы интервала рассеяния
Функция распределения	$F_{LAD}\left(x; x_0, \sigma, x_{\min}, x_{\max}\right) = \int_{x_{\min}}^{x} f_{LAD}\left(\xi; x_0, \sigma, x_{\min}, x_{\max}\right) d\xi.$

Отметим, что, формально, ФР, вычисляемая в соответствие с (0.7), является 3х-параметрическим (оно зависит от следующих параметров: x_0 – положение центра рассеяния, σ – СКО при отсутствии ограничения, l – размаха области рассеяния). Однако если принять во внимание, что

$$l = x_{\max} - x_{\min},$$

где $x_{\min}, x_{\max}$ – координаты, левой и правой границ области рассеяния, знание значений которых оказывается весьма важным в практических приложениях, распределение (0.7) следует отнести к классу 4-х-параметрических распределений.

Анализ свойств функций $f_{LAD}(x;0,\sigma,l)$, $F_{LAD}(x;0,\sigma,l)$, представленных на рисунках 4-6 показывает, что:

– при $x_2 = 30,$ функция $f_{LAD}(x;0,\sigma,l),$ оказывается симметричной относительно прямой $x = 0,$ а центр рассеяния совпадает с математическим ожиданием, медианной и модой;

– при $x_0 \equiv 0$ функция $F_{LAD}(x;0,\sigma,l)$ оказывается антисимметричной относительно прямых $x = 0,\ y = 0.5;$

– при $l \geq 2.5\sigma$ влияние фиктивных источников уменьшается и рассеяние приближается к нормальному закону;

– по мере уменьшения размера области рассеяния влияние фиктивных источников возрастает и при $l \leq 0.7\sigma$ распределение выглядит как равномерное.

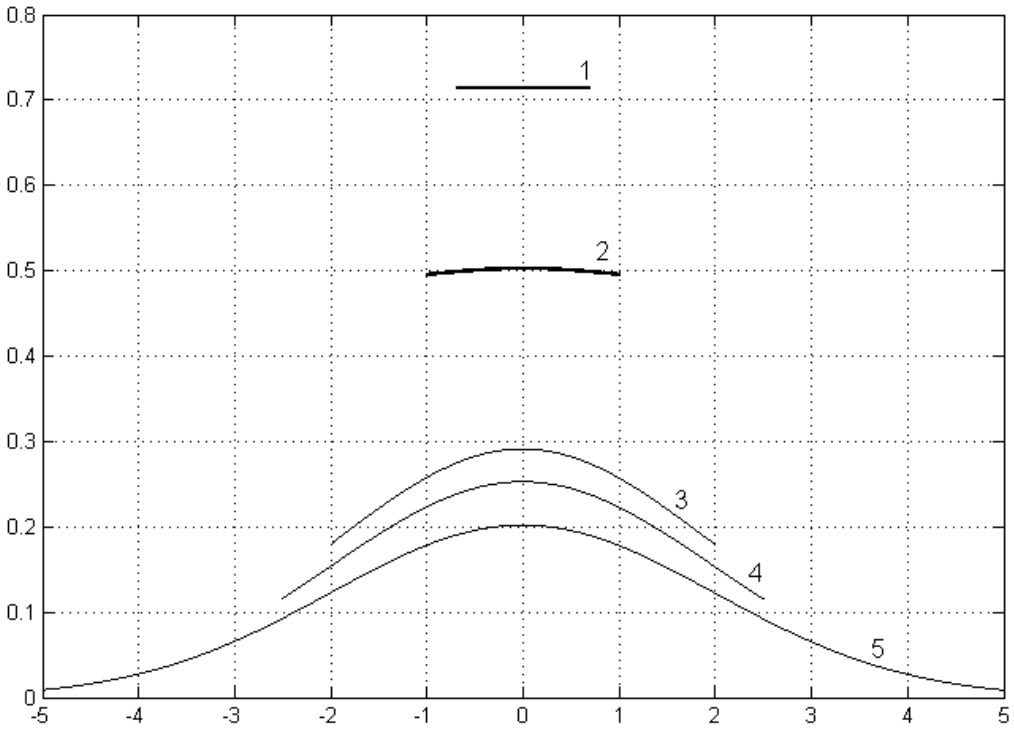


Рис. 4. СПООР: график функции $f_{LAD}(x;x_0,\sigma,l)$:

$1 - x_0 = 0, 2l = 0.7\sigma, x \in [-l,l];\ 2 - x_0 = 0, 2l = 1.0\sigma, x \in [-l,l];\ 3 - x_0 = 0, 2l = 2.0\sigma, x \in [-l,l];$

$4 - x_0 = 0, 2l = 2.5\sigma, x \in [-l,l];\ 5 - x_0 = 0, 2l = 5.0\sigma, x \in [-l,l]$

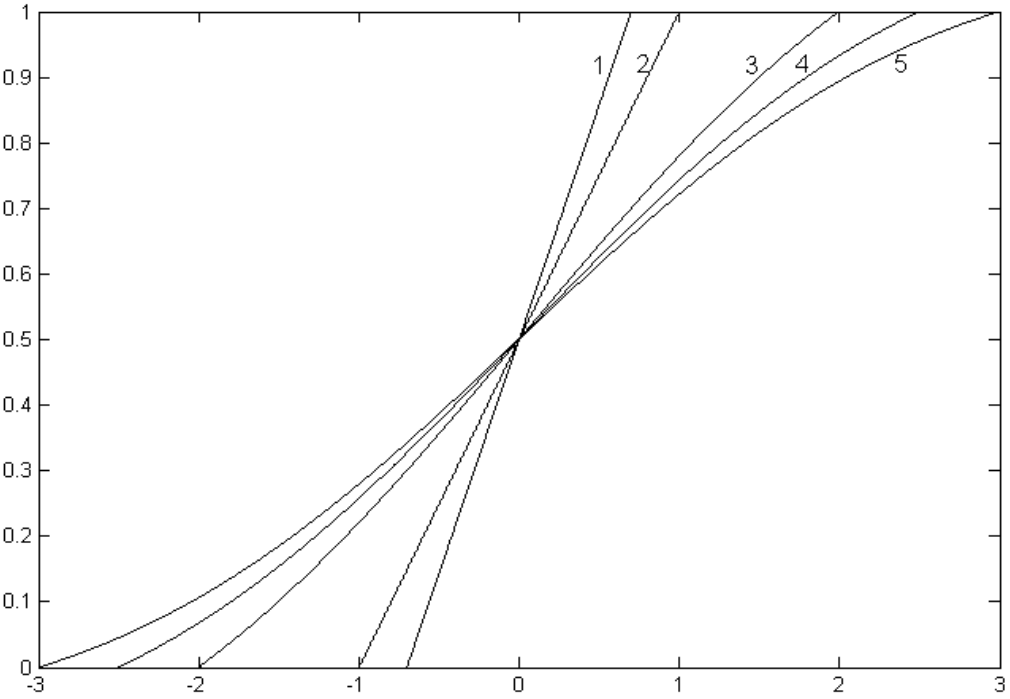


Рис.5. СПООР: график функции $F_{LAD}\left(x;x_0,\sigma,l\right)$:

$1-\ x_0=0,2l=0.7\sigma,x\in\left[-l,l\right];\ 2-\ x_0=0,2l=1.0\sigma,x\in\left[-l,l\right];\ 3-\ x_0=0,2l=2.0\sigma,x\in\left[-l,l\right];$

$4-\ x_0=0,2l=2.5\sigma,x\in\left[-l,l\right];\ 5-\ x_0=0,2l=5.0\sigma,x\in\left[-l,l\right]$

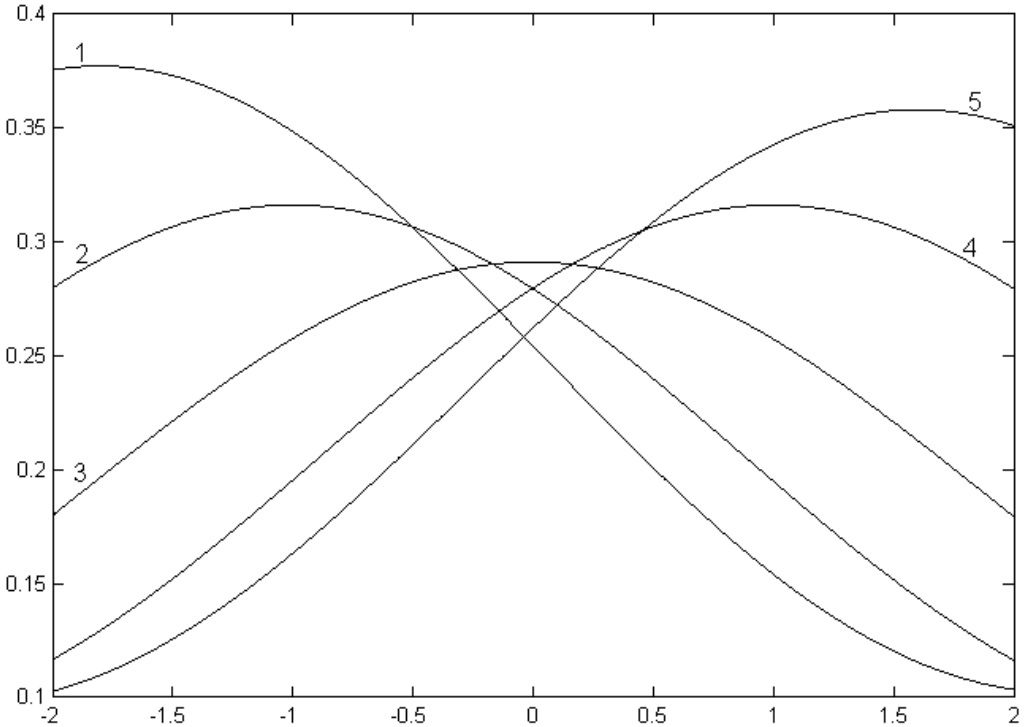


Рис. 6. СПООР: график функции $f_{LAD}\left(x;x_0,\sigma,l/\sigma\right)$:

$1-\ x_0=-0.9\sigma,2l=2\sigma;\ 2-\ x_0=-0.5\sigma,2l=2\sigma;\ 3-\ x_0=0,2l=2.0\sigma;\ 4-\ x_0=0.5\sigma,2l=2.0\sigma;$

$5-\ x_0=0.9\sigma,2l=2.0\sigma$

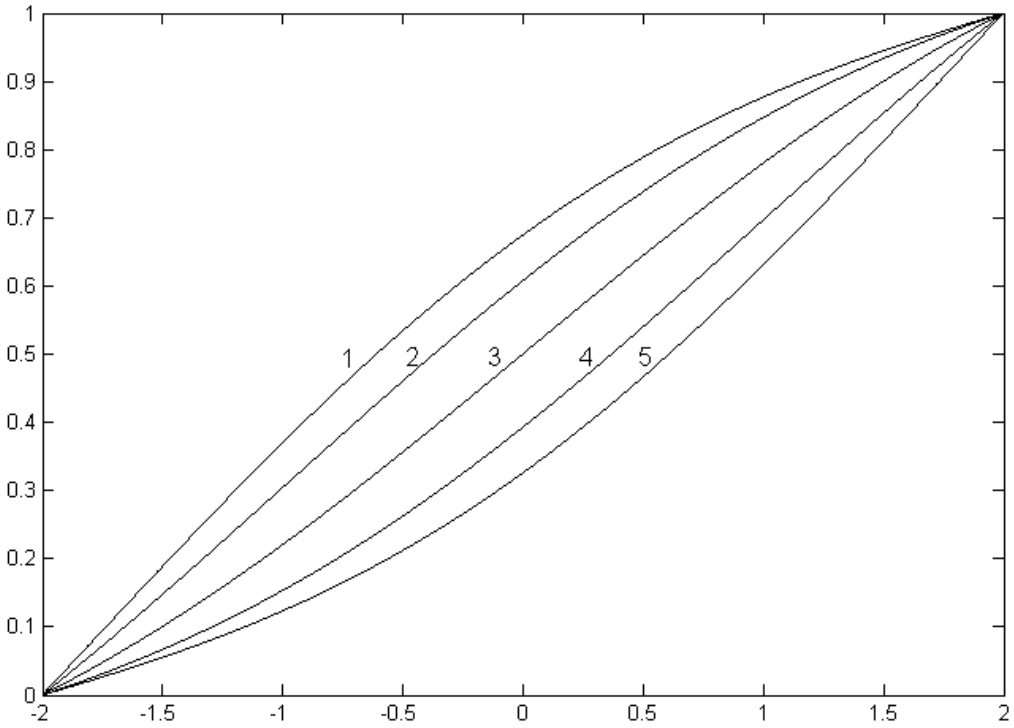


Рис. 7. СПООР: график функции $F_{LAD}\left(x;x_0,\sigma,l\right)$:

1 – $x_0=-0.9\sigma, 2l=2\sigma$; 2 – $x_0=-0.5\sigma, 2l=2\sigma$; 3 – $x_0=0, 2l=2.0\sigma$;

4 – $x_0=0.5\sigma, 2l=2.0\sigma$; 5 – $x_0=0.9\sigma, 2l=2.0\sigma$

При $l\in\left[0.7,2.5\right]\sigma$ распределение (0.5), (0.7), порождает семейство распределений, изменяющихся по своей форме от НР до равномерного распределения.

Анализ свойств функций $f_{LAD}\left(x;x_0,\sigma,2\sigma\right)$, $F_{LAD}\left(x;x_0,\sigma,2\sigma\right)$, представленных на рисунках 8-10, позволяет сделать следующие выводы:

– при $x_0\neq 0$ плотность вероятности $f_{LAD}\left(x;0,\sigma,l\right)$, вычисляемая в соответствие с (0.5), и функция распределения $F_{LAD}\left(x;0,\sigma,l\right)$ вычисляемая в соответствие с (0.7), оказываются асимметричными;

– коэффициент асимметрии изменяется в интервале $\left[-1,1\right]$, он равен: −1 при $x_0=-1$, 0 при $x_0=0$, −1 при $x_0=1$.

Квантиль функции нормального распределения с ограниченной областью рассеяния есть функция, зависящая от доверительной вероятности:

$$x_p^{\langle LAD\rangle}=\arg\left(F_{LAD}\left(x;x_0,\sigma,l\right)=p\right)=Q\left(p;x_0,\sigma,l\right)$$

Зависимости $Q(\alpha, x_0, \sigma, l)$ при заданных значениях α, x_0, σ от длины области рассеяния l представлены на рисунках 8−10.

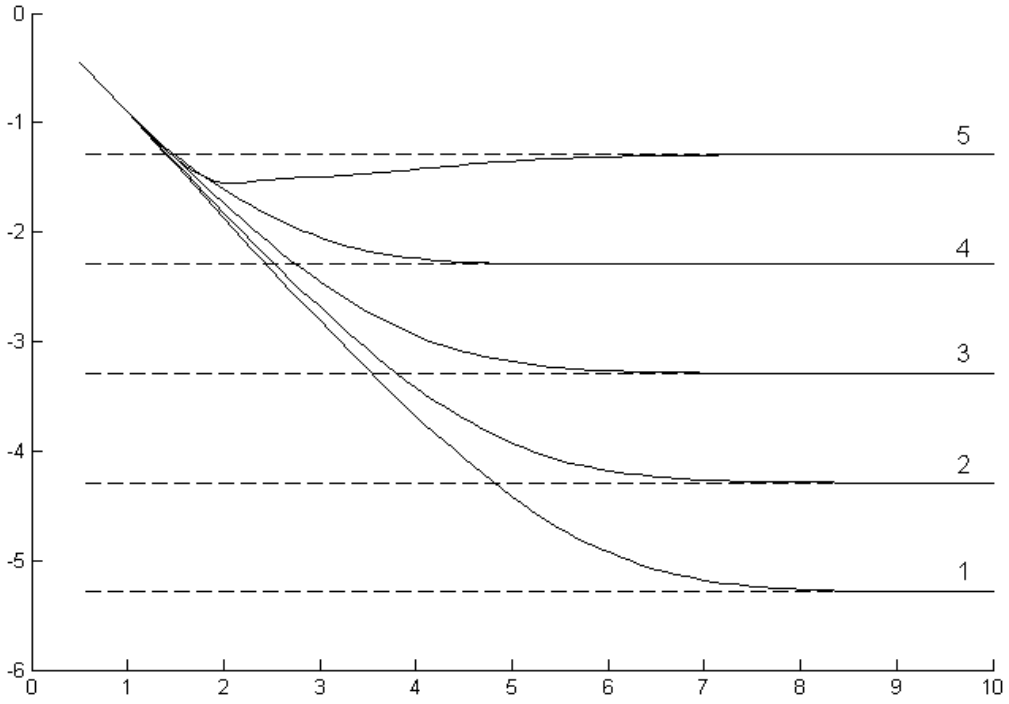


Рис. 8. СПООР: график функции $Q(0.05; x_0, \sigma, l/\sigma),\ l \in [0.5, 10]\sigma$

$1 - x_0 = -\sigma,\ 2 - x_0 = -0.5\sigma,\ 3 - x_0 = 0,\ 4 - x_0 = 0.5\sigma,\ 5 - x_0 = \sigma$

(пунктирные линии − соответствующие квантили равномерного распределения с неограниченной областью рассеяния)

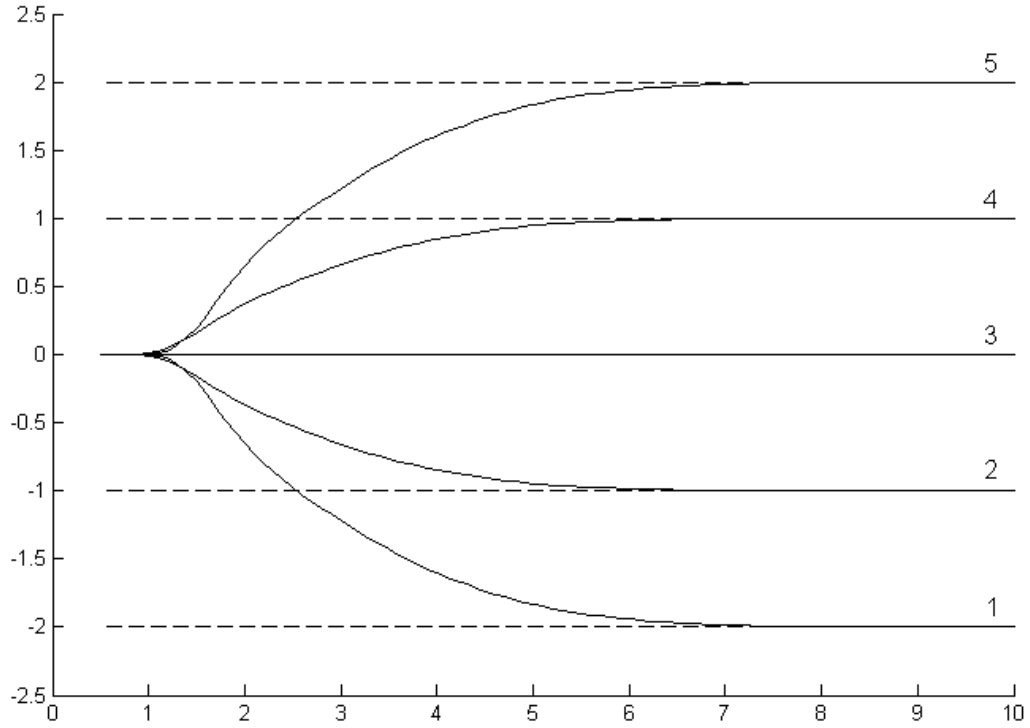


Рис. 9. СПООР: график функции $Q\left(0.5; x_0, \sigma, l/\sigma\right),\ l \in \left[0.5,10\right]\sigma$

1– $x_0 = -\sigma$, 2 – $x_0 = -0.5\sigma$, 3 – $x_0 = 0$, 4 – $x_0 = 0.5\sigma$, 5 – $x_0 = \sigma$

(пунктирные линии − соответствующие квантили равномерного распределения с неограниченной областью рассеяния)

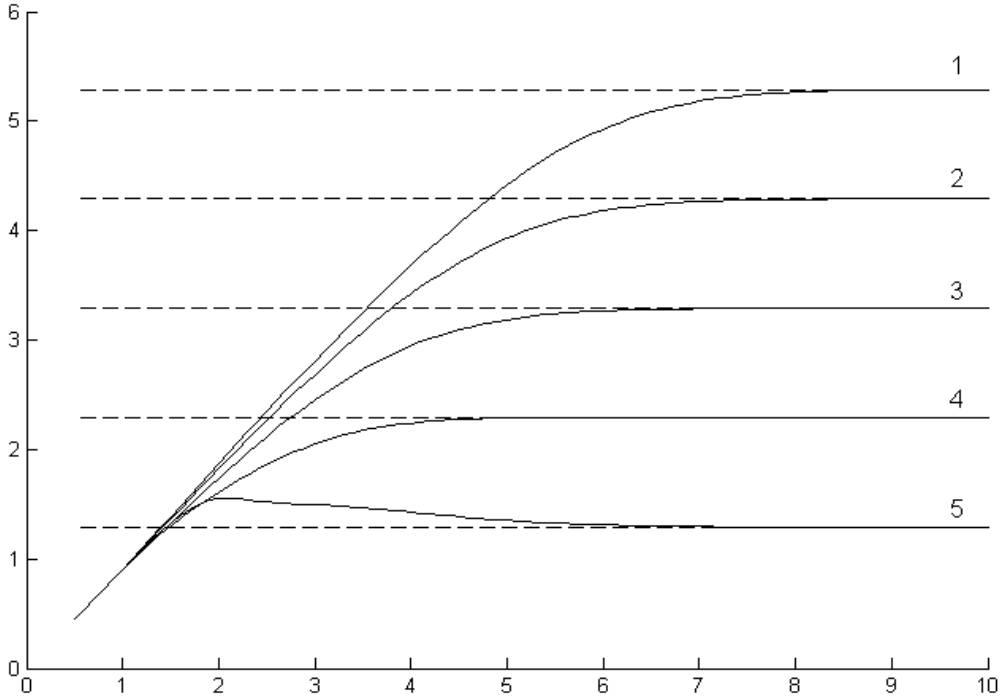


Рис. 10. СПООР: график функции $Q\left(0.95; x_0, \sigma, l/\sigma\right),\ l \in \left[0.5,10\right]\sigma$

1– $x_0 = -\sigma$, 2 – $x_0 = -0.5\sigma$, 3 – $x_0 = 0$, 4 – $x_0 = 0.5\sigma$, 5 – $x_0 = \sigma$

(пунктирные линии − соответствующие квантили равномерного распределения с неограниченной областью рассеяния)

Из рисунков 8−10 видно, что значения ФР СПООР с ограниченной областью рассеяния оказываются, существенно зависящими от параметров распределения. Например, при нахождении источника в центре области рассеяния $\left(x_0 = 0\right)$, значения $x_{0.05}^{\langle LAD \rangle}$, $x_{0.95}^{\langle LAD \rangle}$ совпадают с соответствующими значениями квантилей нормального закона распределения при размере области рассеяния $P_n\left(m, x_{\max}\right) = P_n\left(x = m\right) + P_n\left(x = 2x_{\max} - m\right)$. Данный результат следует принимать во внимание, при анализе функций распределения случайных данных, имеющих ограниченную область рассеяния.

ПРИЛОЖЕНИЕ В. Непараметрический подход, аппроксимация Розенблатта-Парзена

В основе непараметрической статистики лежит подход, позволяющий получать адаптивные оценки эмпирических распределений в виде некоторых функционалов, независящих от вида неизвестного априорного распределения. Для восстановления неизвестной ФР в непараметрической статистике разработан целый ряд методов и алгоритмов: метод гистограмм, «гребенка», метод ближайших соседей, метод разложения по базисным функциям, аппроксимация Розенблатта-Парзена и ряд других. Работоспособность методов непараметрической статистики и целесообразность их применения при анализе экспериментальных данных подтверждается результатами, полученными различными исследователями.

Напомним, что метод Розенблатта-Парзена восстановления (аппроксимации) ФР экспериментальной выборки основан на предположении о том, что ФР оценивается локально в каждой точке x_i с помощью элементов обучающей выборки из некоторой окрестности x_i. При этом общая ФР $F(y)$ есть сумма локальных функций

$$F(y)=\frac{1}{Ns}\sum_{i=1}^{Ns}K\left(\frac{y-x_i}{h}\right), \qquad (0.8)$$

где $K(t)$ – ядерная функция, удовлетворяющая следующим условиям:

а) $K(t)$ – монотонно неубывающая функция, область значений которой принадлежит интервалу $[0,1]$;

б) $K(t)=1-K(t)$ – функция, симметричная относительно 0;

в) $h\to 0$ при $Ns\to\infty$;

h – параметр «размытости», определяющий гладкость получаемой оценки.

В качестве ядерных функций $k(y)$, традиционно, используют функции, представленные в таблице 2.

Таблица 2.

Функции, используемые в качестве ядерных функций

№	*Ядро*	*Формула*
1	Нормальное	$k(t)=\frac{1}{\sqrt{2\pi}}e^{-\frac{t^2}{2}}$
2	Лапласа	$k(t)=\frac{1}{2}e^{-\lvert t\rvert}$

№	*Ядро*	*Формула*
3	Фишера	$k(t)=\frac{1}{2\pi}\left(\frac{\sin\left(\frac{t}{2}\right)}{\frac{t}{2}}\right), \left\|\frac{t}{2}\right\| \le \pi$
4	Коши	$k(t)=\frac{1}{\pi}\left(\frac{1}{1+t^2}\right)$
5	Логистическое	$k(t)=\frac{e^{-t}}{\left(1+e^{-t}\right)^2}$
6	Епанечникова	$k(t)=\frac{3\cdot\left(1-\frac{t^2}{5}\right)}{4\sqrt{5}}, \quad \|t\| \le \sqrt{5}$
7	Равномерное	$k(t)=\frac{1}{2}, \quad \|t\| \le 1$
8	Треугольное	$k(t)=1-\|t\|, \quad \|t\| \le 1$
9	Квадратичное	$k(t)=\frac{3\cdot\left(1-t^2\right)}{4}, \quad \|t\| \le 1$

Соответственно, плотности вероятности $f(y)$ вычисляется по формуле

$$f(y)=\frac{1}{Ns\cdot h}\sum_{i=1}^{Ns}k\left(\frac{y-x_i}{h}\right), \tag{0.9}$$

где $k(y)=\frac{d}{dy}K(y).$

Данные оценки предложены Розенблаттом и исследованы Парзеном.

В рассматриваемом методе качество аппроксимации зависит от вида ядра $k(t)$ и от значения параметра размытости h (рисунок 11).

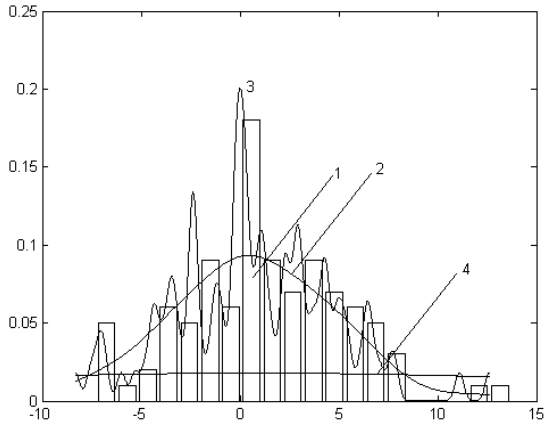


Рисунок 11. ПР случайной последовательности $x_i, i=\overline{1,100}$, сгенерированной в соответствие с НР $N(1,4)$: 1 – гистограмма случайной последовательности; 2 – нормальное ядро,

$h = h_1^*$, где h_1^* пока не определено; 3 – нормальное ядро, $h < h_1^*$ $h = h_1^*$, где h_1^* пока не определено; 4 – нормальное ядро, $h > h_1^*$ $h < h_1^*$, где h_1^* пока не определено

Оптимальные значения ядерной функции и параметра находятся из условия достижения функционалом

$$J = \int \ln k(t) \cdot f(t) dt \tag{0.10}$$

максимального значения, которое, как очевидно, выполняется при $k(t) = f(t).$

При анализе экспериментальных данных задача поиска оптимального значения параметра «размытости» рекомендовано находить для каждой из ядерных функций, представленных в таблице 2, оптимальное значения $h_m^*, m = \overline{1,9}$, как решение задачи

$$h_m{}^* = \arg\max \left\{ \frac{1}{Ns} \sum_{i=1}^{Ns} \ln \left[\frac{1}{(Ns-1) \cdot h_m} \sum_{j \neq i}^{Ns-1} k_m \left(\frac{x_i - x_j}{h_m} \right) \right] \right\}, \tag{0.11}$$

и далее выбирать из них ту ядерную функцию $k_m(y)$, для которой значение функции

$$\varphi(h_m) = \frac{1}{Ns} \sum_{i=1}^{Ns} \ln \left[\frac{1}{(Ns-1) \cdot h_m} \sum_{j \neq i}^{Ns-1} k_m \left(\frac{x_i - x_j}{h_m} \right) \right] \tag{0.12}$$

в точке $h_m = h_m^*$ будет наибольшим.

Из (0.11) видно, что нахождение оптимального значения параметра размытости h для каждой из базисных функций сводится к решению сложного нелинейного уравнения

$$\frac{\partial \varphi(h)}{\partial h} = \frac{\partial}{\partial h} \sum_{i=1}^{Ns} \ln \left[\frac{1}{(Ns-1) \cdot h} \sum_{j \neq i}^{Ns-1} k_m \left(\frac{x_i - x_j}{h} \right) \right] = 0, \tag{0.13}$$

найти которое оказывается возможным только численно.

Таким образом, в общем случае решение описанных выше задач Error! Reference source not found., (0.8), (0.11) приводит к решению системы нелинейных уравнений. Как известно, для системы нелинейных уравнений не существует универсального детерминированного метода решения, поэтому применяют численные методы, например, итерационный метод Ньютона, метод градиентного спуска, симплекс-метод. Однако, известно, что для итерационных методов сходимость к истинному решению очень сильно зависит от начального приближения. В этой связи представляются перспективными эвристические методы случайного поиска, результативность которых, как утверждается, не

зависит от начального приближения и позволяет найти оптимальное решение при любых начальных условиях. Одним из таких методов являются ГА.

ПРИЛОЖЕНИЕ Г. Библиотеки программных реализаций метода мнимых источников и аппроксимации Розенблатта-Парзена

1. Описание функции solveDoublePeak1.m, аппроксимации Розенблатта-Парзена

function [D1, D2, alpha, y] = solveDoublePeak1(cx, zD1, zD2, zalpha, fPR)

Функция, возвращающая

D1 – структуру, содержащую вычисленные параметры первой составляющей распределения

D2 - структуру, содержащую вычисленные параметры второй составляющей распределения

alpha – вычисленное значение параметра alpha

y – вектор значений аппроксимации плотности распределения

Функция применяется для первого шага итерационного алгоритма

Входные параметры

cx – значения аргумента для вычисления плотности распределения

zD1 – структура, содержащая вычисленные параметры первой составляющей распределения на предыдущем этапе

zD2 – структура, содержащая вычисленные параметры второй составляющей распределения на предыдущем этапе

fPR – значения аппроксимации Розенблатта-Парзена

zalpha – вычисленное значение параметра alpha на предыдущем этапе

fPR – значения аппроксимации Розенблатта-Парзена

2. Описание функции doublePeakLADinv2.m, метод мнимых источников.

function z=doublePeakLADinv2(p,...
Mu1,Sigma1,a1,b1,...
Mu2,Sigma2,a2,b2,...
alpha, N)

Функция, возвращающая

z – значение квантиля функции распределения двумодального распределения

Входные параметры

p – вектор значений вероятности

Mu1, Sigma1, a1, b1 – параметры первого распределения

Mu2, Sigma2, a2, b2 – параметры второго распределения

alpha – коэффициент первого распределения

N – количество мнимых источников

ПРИЛОЖЕНИЕ Д. Программа для ЭВМ «Анализатор-классификатор информационных потоков дампов трафика компьютерных сетей»

Исходный текст программы

1.1 streamfinder.m

```
//
//  File: streamfinder.m
//
//  Abstract: This software is intended for use for research purposes
//            to analyze modern large backbone with ability to use CPU
//            based parallel computing
//
//  Version: <1.0>
//
//  Copyright (c) 2015 Porshnev S.V., Bozhalkin D.A. All rights reserved.
//
////////////////////////////////////////////////////////////////////////

clc; clear; close all;

% Число обрабатываемых файлов дампа
nof=input('Enter number of files: ');

% Имена файлов дампа
for y=1:nof
    dsp=['Enter file name number ',num2str(y),': '];
    m(y).fname=input(dsp,'s');
    m(y).Slon=1;
    m(y).Mul=1;
    m(y).Mouse=1;
end

% Границы среднего потока
maxflowsize=input('Enter max flow size: ');
minflowsize=input('Enter min flow size: ');

% Число параллельных процессов
cpunum=input('Enter number of cores: ');
parpool (cpunum);

% Фиксируем время выполнения задания
tic

% Поиск потоков размером больше минимального значения среднего потока
for q=1:size((m),2)
```

```
% Загрузка файла содержащего структуру
a = load(m(q).fname);
M=1;

% Подготовка структуры для последующей обработки
parfor i=1:size((a),2);
    a(i).frame_time=fix(a(i).frame_time);
    if a(s).ip_src
    else
        a(s).ip_src='0';
    end

    if a(s).ip_dst
    else
        a(s).ip_dst='0';
    end

    if a(s).tcp_srcport
    else
        a(s).tcp_srcport=0;
    end

    if a(s).tcp_dstport
    else
        a(s).tcp_dstport=0;
    end

    if a(s).udp_srcport
    else
        a(s).udp_srcport=0;
    end

    if a(s).udp_dstport
    else
        a(s).udp_dstport=0;
    end

    if a(i).ip_proto==1
       M=[M,i];
    end
end

% Удаление ICMP-пакетов для ускорения обработки
```

```
    a(M)=[];
    A=a;
   % Определяем принадлежность пакетов к потокам
   parfor i=1:size((a),2);
        N=0; F=0;
        for j=i-1:-1:1
                if F<=10000
                    if strcmp (a(i).ip_src,a(j).ip_src) && strcmp
(a(i).ip_dst,a(j).ip_dst) || strcmp (a(i).ip_src,a(j).ip_dst) && strcmp
(a(i).ip_dst,a(j).ip_src);
                        if a(i).tcp_srcport==a(j).tcp_srcport  ||
a(i).tcp_srcport==a(j).tcp_dstport || a(i).udp_srcport==a(j).udp_srcport  ||
a(i).udp_srcport==a(j).udp_dstport;
                        N=1;
                        break
                        end
                    end
                else
                    N=1
                    break
                end
            F=F+1;
        end
        if N==0
                for y=i:size((a),2);
                    if N<=10000
                        if strcmp (a(i).ip_src,a(y).ip_src) && strcmp
(a(i).ip_dst,a(y).ip_dst) || strcmp (a(i).ip_src,a(y).ip_dst) && strcmp
(a(i).ip_dst,a(y).ip_src);
                            if a(i).tcp_srcport==a(y).tcp_srcport  ||
a(i).tcp_srcport==a(y).tcp_dstport || a(i).udp_srcport==a(y).udp_srcport  ||
a(i).udp_srcport==a(y).udp_dstport;
                            A(i).frame_len=A(i).frame_len+a(y).frame_len;
                            N=1
                            end
                        end
                    else
                        break
                    end
                    N=N+1
                end
        end
    end
```

```
   % Поиск потоков больше заданной длины
    D=1;
    parfor i=1:size((A),2)
        if A(1,i).frame_len<minflowsize
            D=[D,i]
        end
    end
    D(1)=[];
    A(D)=[];
    % Сохранение потоков больше заданной длины
    save(m(q).fname,'A');
end

% Объединение результатов обработки нескольких дампов
load(m(1,1).fname, 'A');
B=A;

for q=2:size((m),2)
    load (m(1,q).fname, 'A');
    B=[B,A];
end

b=B(1);

for j=1:size((B),2)
    N=0;
    for i=1:size((b),2)
       if strcmp (b(i).ip_src,B(j).ip_src) && strcmp
(b(i).ip_dst,B(j).ip_dst) || strcmp (b(i).ip_src,B(j).ip_dst) && strcmp
(b(i).ip_dst,B(j).ip_src);
           if b(i).tcp_srcport==B(j).tcp_srcport  ||
b(i).tcp_srcport==B(j).tcp_dstport || b(i).udp_srcport==B(j).udp_srcport  ||
b(i).udp_srcport==B(j).udp_dstport;
           b(i).frame_len=b(i).frame_len+B(j).frame_len;
           N=1;
           end
       end
    end
    if N==0
       b=[b,B(j)];
       n=size((b),2);
    end
```

```
end

% Поиск числа пакетов для каждого класса потоков для окна агрегации равного 1
миллисекунда
A=b; Slon=1; Mul=1; Mouse=1;

parfor q=1:size((m),2)
    a = load(m(q).fname);

    % Подготавливаем структуру для последующей обработки
    for s=1:size((a),2);
        if a(s).ip_src
        else
            a(s).ip_src='0';
        end

        if a(s).ip_dst
        else
            a(s).ip_dst='0';
        end

        if a(s).tcp_srcport
        else
            a(s).tcp_srcport=0;
        end

        if a(s).tcp_dstport
        else
            a(s).tcp_dstport=0;
        end

        if a(s).udp_srcport
        else
            a(s).udp_srcport=0;
        end

        if a(s).udp_dstport
        else
            a(s).udp_dstport=0;
        end
    end

      % Разбиваем дамп на отрезки по 1 миллисекунде
```

```
I=1;S=1;i=1;
while i~=size((a),2)
    if a(i).frame_time-a(I).frame_time<0.01;
    i=i+1;
    else
        I=i;
        S=[S,I-1];
    end
end

% Массивы равные длине дампа в миллисекундах
slon=zeros(1,size((S),2)-1);
mul=zeros(1,size((S),2)-1);
mouse=zeros(1,size((S),2)-1);

% Определение числа пакетов для каждого класса потоков на интервале 1 милли-
секунда
d=0;
for i=1:(size((S),2)-1)
    for I=S(i):S(i+1)
        for l=1:size((A),2);
            if strcmp (a(I).ip_src,A(l).ip_src) && strcmp
(a(I).ip_dst,A(l).ip_dst) || strcmp (a(I).ip_src,A(l).ip_dst) && strcmp
(a(I).ip_dst,A(l).ip_src);
                if a(I).tcp_srcport==A(l).tcp_srcport  ||
a(I).tcp_srcport==A(l).tcp_dstport || a(I).udp_srcport==A(l).udp_srcport
|| a(I).udp_srcport==A(l).udp_dstport;
                    if A(l).frame_len>=maxflowsize
                       slon=[slon,a(I).frame_len]
                       d=[d,I]
                       break;
                    else
                       mul=[mul,a(I).frame_len]
                       d=[d,I]
                       break;
                    end
                end
            end
        end
    end
end
   d(1)=[]
   aa=a
```

```
          aa(d)=[]
          mouse=[aa.frame_len]

      % Запись результатов для каждого окна агрегации
      m(q).Slon=[Slon,slon];
      m(q).Mul=[Mul,mul];
      m(q).Mouse=[Mouse,mouse];
end

for i=1:size((m),2)
    m(i).Slon(1)=[];
    Slon=[Slon,m(i).Slon];
    m(i).Mul(1)=[];
    Mul=[Mul,m(i).Mul];
    m(i).Mouse(1)=[];
    Mouse=[Mouse,m(i).Mouse];
end

Slon(1)=[];
Mul(1)=[];
Mouse(1)=[];

% Сохранение результатов обработки дампа
save('numder_of_packets','Slon', 'Mul', 'Mouse');

% Поиск объема переданных данных для каждого класса потоков для окна агрега-
ции равного 1 миллисекунда
A=b; Slon=1; Mul=1; Mouse=1;

parfor q=1:size((m),2)
    a = load(m(q).fname);

    % Подготавливаем структуру для последующей обработки
    for s=1:size((a),2);
        if a(s).ip_src
        else
            a(s).ip_src='0';
        end
        if a(s).ip_dst
        else
            a(s).ip_dst='0';
        end
        if a(s).tcp_srcport
```

```
    else
        a(s).tcp_srcport=0;
    end
          if a(s).tcp_dstport
    else
        a(s).tcp_dstport=0;
    end
    if a(s).udp_srcport
    else
        a(s).udp_srcport=0;
    end
    if a(s).udp_dstport
    else
        a(s).udp_dstport=0;
    end
end

  % Разбиваем дамп на отрезки по 1 миллисекунде
  I=1; S=1; i=1;
  while i~=size((a),2)
      if a(i).frame_time-a(I).frame_time<0.01;
      i=i+1;
      else
          I=i;
          S=[S,I-1];
      end
  end

  % Массивы равные длине дампа в миллисекундах
  slon=zeros(1,size((S),2)-1);
  mul=zeros(1,size((S),2)-1);
  mouse=zeros(1,size((S),2)-1);

  % Определение объема переданной информации для каждого класса потоков на
  интервале 1 миллисекунда
  for i=1:(size((S),2)-1)
      for I=S(i):S(i+1)
          for l=1:size((A),2);
              if strcmp (a(I).ip_src,A(l).ip_src) && strcmp
  (a(I).ip_dst,A(l).ip_dst) || strcmp (a(I).ip_src,A(l).ip_dst) && strcmp
  (a(I).ip_dst,A(l).ip_src);
```

```
                    if a(I).tcp_srcport==A(l).tcp_srcport  ||
      a(I).tcp_srcport==A(l).tcp_dstport || a(I).udp_srcport==A(l).udp_srcport
      || a(I).udp_srcport==A(l).udp_dstport;
                        if A(l).frame_len>=maxflowsize
                           slon(i)=slon(i)+a(I).frame_len;
                           break;
                        else
                            mul(i)=mul(i)+a(I).frame_len;
                            break;
                        end
                    end
                end
            end
        end
        tmp=0;
        for II=S(i):S(i+1)
            tmp=tmp+a(II).frame_len;
        end
        mouse(i)=tmp-mul(i)-slon(i);
      end
% Запись результатов для каждого окна агрегации
      m(q).Slon=[Slon,slon];
      m(q).Mul=[Mul,mul];
      m(q).Mouse=[Mouse,mouse];
end
for i=1:size((m),2)
    m(i).Slon(1)=[];
    Slon=[Slon,m(i).Slon];
    m(i).Mul(1)=[];
    Mul=[Mul,m(i).Mul];
    m(i).Mouse(1)=[];
    Mouse=[Mouse,m(i).Mouse];
end

Slon(1)=[];
Mul(1)=[];
Mouse(1)=[];

% Сохранение результатов обработки дампа
save('size_of_pakets','Slon', 'Mul', 'Mouse');
% Вывод времени выполнения задания и закрытие пула процессоров
toc
matlabpool close
```

ПРИЛОЖЕНИЕ Е. Свидетельство о регистрации программы для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2015611426

«Семантический анализатор дампов трафика информационных потоков в компьютерных сетях»

Правообладатель: ***Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Уральский федеральный университет имени первого Президента России Б.Н.Ельцина» (RU)***

Авторы: ***Поршнев Сергей Владимирович (RU), Божалкин Даниил Александрович (RU)***

Заявка № **2014662922**

Дата поступления **12 декабря 2014 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***29 января 2015 г.***

Врио руководителя Федеральной службы по интеллектуальной собственности

Л.Л. Кирий

ПРИЛОЖЕНИЕ Ж. Свидетельство о регистрации программы для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2015661799

«Анализатор-классификатор информационных потоков дампов трафика компьютерных сетей»

Правообладатель: ***Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Уральский федеральный университет имени первого Президента России Б.Н.Ельцина» (RU)***

Авторы: ***Поршнев Сергей Владимирович (RU), Божалкин Даниил Александрович (RU)***

Заявка № **2015618601**

Дата поступления **17 сентября 2015 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***09 ноября 2015 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев