Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования (ФГБОУ ВПО)
«Московский государственный технический университет им. Н.Э. Баумана»

На правах рукописи

УДК 004.7: 519.2

Колесников Александр Владимирович

Моделирование сетевого трафика и алгоритмы борьбы с перегрузками на основе методов нелинейной динамики и краткосрочного прогнозирования временных рядов

Специальность 05.13.15 – Вычислительные машины, комплексы и компьютерные сети

Диссертация на соискание ученой степени

кандидата технических наук

Научный руководитель:

доктор техн. наук

И.П. Иванов

Москва 2015

ОГЛАВЛЕНИЕ

Введение

Бурное развитие телекоммуникационных технологий, а также снижение стоимости передачи и обработки информации приводит к постоянному росту объема сетевого трафика. Ежегодный отчет компании Sandvine [1] говорит о среднем увеличении агрегированного трафика для Северной Америки на 30-40% ежегодно. На 2014 год он составляет 51,4 Гб в месяц на одного пользователя. При этом источником более 60% трафика служат мультимедиа приложения потокового аудио и видео. Как известно, трафик подобного рода наиболее чувствителен к задержкам в передаче пакетов [2]. Так, при передаче голосовых данных критичной становится задержка более 10 мс, для потокового видео задержка не должна превышать 100 мс. Таким образом, предъявляются повышенные требования к сетям передачи данных, алгоритмам управления трафиком и, в частности, к методам борьбы с перегрузками.

Задача управления трафиком является одной из ключевых при обеспечении качества обслуживания (QoS - Quality of Service) абонентов. С точки зрения топологии сети, управление трафиком включает в себя сетевое планирование и оптимизацию. Сетевое планирование – это процесс определения топологии сети и пропускной способности каналов связи с учетом предполагаемой нагрузки. Оптимизация предполагает управление распределением трафика в существующей сети [3].

Борьба с перегрузками в компьютерной сети – важная часть задачи обеспечения QoS и управления трафиком в частности. Эффективные алгоритмы борьбы с перегрузками позволяют повысить не только надежность, но и полезную пропускную способность сети. Перегрузка в компьютерной сети возникает тогда, когда объем передаваемой информации приближается к максимальной пропускной способности сети. Протокол IP для обмена данными не требует предварительной установки соединения, таким образом, сетевое устройство не может определить необходимый объем

ресурсов до получения первого сообщения, это может привести к тому, что входящий поток займет все ресурсы маршрутизатора. В первую очередь это связано с использованием буферов конечной длины в коммутирующем оборудовании. При заполнении этих буферов новые пакеты начинают отбрасываться узлами сети, что в свою очередь вызывает повторную передачу сообщения, и нагрузка растет лавинообразно. Причина такой ситуации кроется в том, что протоколы транспортного уровня модели OSI, например TCP, не имеют непосредственных средств определения состояния сети, делая выводы на основе диалога с оконечными системами. Получатель в рамках протокола TCP посылает отправителю подтверждение о приеме и объем данных, которые он готов принять. Таким образом, исходные данные для алгоритмов управления каналом связи получаются достаточно грубым методом [4].

Для нормального функционирования сети, необходимо поддержание пропускной способности на приемлемом уровне. Ряд методик на основе обратной связи позволяют управлять пропускной способностью в зависимости от нагрузки: противодавление, сдерживающий пакет, сигнализация о перегрузке. Способы обеспечения QoS без обратной связи, такие как принцип справедливого распределения ресурсов, алгоритмы управления очередями (в том числе RED), а также алгоритмы профилирования нагрузки, функционируют на уровне сетевого устройства. Вышеперечисленные подходы к обеспечению качества обслуживания разрабатывались и тестировались на пуассоновских моделях трафика и не учитывают самоподобной структуры телекоммуникационного трафика и распределения с тяжелым хвостом. Использование более адекватных моделей трафика может способствовать разработке эффективных протоколов передачи данных и методик обеспечения качества обслуживания абонентов. В частности, это касается методик прогнозирования трафика и состояния сети для борьбы с перегрузками.

В таком случае особенно важно не только разработать адекватный алгоритм прогноза, но и учесть природу трафика, протокол передачи, архитектуру сети, степень загруженности и характер нагрузки. В зависимости от перечисленных выше факторов статистические и динамические свойства трафика будут различаться.

Системы управления перегрузками в сети на основе прогнозируемой нагрузки показывают лучшие результаты по сравнению с системами, работающими с сетью в реальном времени. Стоит отметить, что в большинстве работ по разработке способов управления перегрузками в сети, не учитываются нелинейно-динамические свойства процесса передачи сообщений, хаотический характер трафика, а так же влияние трафика на аппаратные ресурсы серверов и сетевое оборудование. Между тем, выявление зависимостей между объемом трафика и состоянием оборудования может помочь при разработке способов прогноза перегрузок и, как следствие, повышения качества обслуживания абонентов сети. Поэтому актуальным является исследование реальной корпоративной сети, определение характера процесса передачи и обработки данных с последующей разработкой методик повышения качества обслуживания абонентов и управления перегрузками.

Применительно к телекоммуникационному трафику методы нелинейной динамики стали использоваться не так давно. Значительный вклад в развитие данного направления внесли такие исследователи, как P. Abry, S. Floyd, W.E. Leland, K. Park, M.S. Taqqu, W. Willinger [5, 6] и др. Среди отечественных исследователей особенно стоит отметить работы Шелухина О.И. [7, 8]. Так, в монографии [9], приводятся теоретические аспекты самоподобных случайных процессов, дается объяснение, почему трафик в современных телекоммуникационных системах следует считать фрактальным, а также рассматриваются математическая и программная реализация самоподобных математических моделей. Приводится анализ самоподобности LAN и WAN трафика с учетом особенностей протоколов транспортного и прикладного

уровней, рассматривается влияние самоподобия на оценку качества предоставления услуг абонентам.

Применительно к корпоративным сетям можно отметить такие специфические особенности трафика, как разнообразие передаваемой информации, ее высокая интенсивность и объем, закрытость и ограниченность методик анализа процессов в сетевых узлах не отечественных разработок. При этом наблюдается разрыв между реально наблюдаемыми результатами функционирования сетей от распространенных математических моделей источников информации, узлов сетей и трафика. Решению этой проблемы во многом посвящены работы Иванова И.П. [10,11,12] и других исследователей из ряда российских и зарубежных институтов и университетов. Вместе с тем, по-прежнему остается актуальной проблема рассмотрения вышеуказанных вопросов в корпоративных сетях на основе методов нелинейной динамики, что позволит осуществить внедрение и освоение новых информационных технологий для решения важных народно-хозяйственных задач во многих отраслях.

Цель работы. Целью работы являлась разработка методики борьбы с перегрузками в корпоративной сети на основе методов нелинейной динамики и краткосрочного прогнозирования поведения временных рядов, характеризующих сетевой трафик.

В рамках исследования был выделен нагруженный сервер корпоративной сети МГТУ им. Н.Э. Баумана, который в режиме реального времени осуществлял сбор и хранение данных о входящем и исходящем трафике, а также распределении аппаратных ресурсов (постоянной и оперативной памяти, ЦП, состоянии ОС).

Для достижения поставленной цели решены следующие **задачи исследования**:

1) Сбор и анализ реальных данных методами статистического и нелинейно-динамического анализа временных рядов; расчет параметров с целью

разработки адекватной имитационной модели сети и выявление степени персистентности исследуемых данных для оценки параметров прогноза.

2) Разработка имитационной модели сети средствами среды Matlab и Simulink, адекватно отражающей процессы, протекающие в реальной сети передачи данных.
3) Разработка методики прогнозирования динамических процессов в сети передачи данных на основе рассчитанных статистических и нелинейно-динамических параметров.

Методы исследования. Для решения поставленных задач в работе использованы методы статистической обработки и прогнозирования временных рядов, методы нелинейной (хаотической и фрактальной) динамики, имитационного моделирования.

Основные **положения, выносимые на защиту:**

1) Результаты статистического и нелинейно-динамического анализа суммарного трафика сервера корпоративной компьютерной сети.
2) Оценка и анализ влияния входящего и исходящего трафика на аппаратную нагрузку сервера сети.
3) Имитационная модель сети, количественно и качественно согласующаяся с реальной корпоративной сетью, в том числе по самоподобным свойствам.
4) Математическая модель сетевого трафика, оптимально подходящая для описания и прогнозирования исследуемых процессов передачи трафика.
5) Методика борьбы с перегрузками в сети на основе оценки нелинейно-динамических свойств трафика и краткосрочного прогнозирования.

Научная новизна данной работы заключается в следующем:

1) На основании количественного и качественного анализа временных рядов суммарного трафика и распределения аппаратных ресурсов корпоративной сети установлено, что исследуемые процессы

характеризуются высокой степенью самоподобия и наличием хаотических свойств.

2) В результате корреляционного анализа установлено, что объем входящего и исходящего сетевого трафика связан линейной зависимостью с нагрузкой на аппаратные ресурсы сервера; при этом в большей степени оказывается влияние на центральный процессор и оперативную память.
3) Установлено, что коэффициент использования сети может быть увеличен с помощью краткосрочного прогнозирования нагрузки и введения механизма обратной связи, учитывающего самоподобные и хаотические свойства сетевого трафика.

Научная значимость диссертации определяется следующими полученными в ней результатами:

1) Получены количественные и качественные оценки степени самоподобия и хаотичности агрегированного трафика сервера корпоративной сети.
2) Получены значения степени корреляции трафика и аппаратных ресурсов сервера.
3) Разработана модель, адекватно представляющая процесс передачи пакетов сообщений в корпоративной сети.
4) Разработана методика краткосрочного прогнозирования объема поступающего трафика с целью борьбы с перегрузками.

Достоверность и обоснованность научных результатов, полученных в данной работе, подтверждена адекватностью совокупности применяемых для исследования математических методов, длительностью и повторяемостью эксперимента, а также соответствием результатов имитационного моделирования, выдвигаемым положениям.

Практическая ценность работы. Предложенная в данной работе модель компьютерной сети представляет базовый функционал для имитационного моделирования сетевых процессов. Разработанная методика

прогнозирования нагрузки на основе динамических свойств трафика находит применение в реальных сетях передачи данных.

Личный вклад. Все основные научные положения и выводы, составляющие содержание данной работы, получены автором лично.

Апробация работы. Основные научные и практические результаты данной работы обсуждались и докладывались на международной конференции “Computer Data Analysis and Modeling: Theoretical and Applied Stochastics” (Minsk, 2013), II Международной научно-практической конференции «Теоретические и прикладные аспекты современной науки» (г. Белгород, 2014 г.), Всероссийской научно-технической конференции «Студенческая научная весна» (г. Москва, МГТУ им. Н.Э. Баумана, 2014), конференции «Телекоммуникационные и вычислительные системы» (г. Москва, МТУСИ, 2014 г.). Работа автора «Разработка метода управления перегрузками компьютерной сети на основе нелинейно-динамических свойств трафика» была отмечена Дипломом I степени на Конкурсе научно-исследовательских работ МГТУ им. Баумана с международным участием в 2014 г. Задействованные методики и установленные результаты использованы в учебном курсе «Методы моделирования фрактальных процессов в телекоммуникационных сетях» МГТУ им. Н.Э.Баумана.

Публикации. По теме диссертации опубликовано 7 работ, в том числе 3 статьи в изданиях, рекомендованных ВАК.

Структура и объем диссертации. Диссертационная работа содержит 147 страниц и состоит из введения, четырех глав, заключения, списка литературы и 2 приложений.

Во **введении** обоснована актуальность темы диссертации, показана научная новизна и практическая значимость результатов диссертации, а также перечислены основные положения, выносимые на защиту.

В первой главе приводится обзор и классификация существующих моделей трафика и подходов к реализации процесса поступления сообщений от источника. Приводится описание самоподобных моделей трафика. Рассматриваются основные методики борьбы с перегрузками в сетях передачи данных.

Во второй главе произведен сбор и статистический анализ трафика и процесса распределения аппаратных ресурсов на примере одного из физических серверов корпоративной сети МГТУ им. Н.Э. Баумана. Сделан вывод о наличии самоподобных свойств трафика корпоративной сети и возможности краткосрочного прогнозирования нагрузки.

В **третьей главе** приведены результаты сравнительного анализа алгоритмов краткосрочного прогнозирования временных рядов для выбора модели, оптимально описывающей исследуемые процессы. Было установлено, что модель ARFIMA(p,d,q) превосходит остальные рассмотренные по точности прогнозирования, что согласуется со спецификой модели и самоподобным характером исследуемых процессов.

В четвертой главе приводятся результаты моделирования передачи трафика в локальной сети с механизмом управления перегрузками с учетом кратковременного прогнозирования нагрузки. Средствами программного пакета Matlab с библиотекой Simulink разработана имитационная модель корпоративной сети, а также проведена оценка её адекватности. Предложена методика, позволяющая снизить потери пакетов с помощью обратной связи и краткосрочного прогнозирования нагрузки по схеме ARFIMA(p,d,q).

В заключении подведены итоги диссертации.

1 Модели трафика и методы борьбы с перегрузками в сетях передачи данных

1.1 Модели трафика

Как известно, постановку эксперимента и проведение исследования можно выполнить двумя противоположными подходами. В первом случае – задействовать реальное оборудование, исследовать процесс во времени. Во втором случае для проведения исследования можно выполнить численное моделирование эксперимента, задействовав вычислительные мощности компьютера и математические модели интересующего процесса. Зачастую совокупность этих двух подходов, их сравнительный анализ, позволяют получить достоверные результаты исследования. И если в источнике экспериментальных данных, полученных с помощью первого подхода, сомневаться не приходится, то во втором подходе очень важную роль при постановке эксперимента играет адекватность используемой математической модели процесса.

В случае компьютерного моделирования сети передачи данных (системы дискретных событий DES [14]) важно использовать подходящую модель сетевого трафика, используемую в численном эксперименте. Точность модели, то, насколько она отвечает параметрам реальной сети передачи данных, задает качество результатов эксперимента.

В простейшем случае трафик может быть представлен как процесс поступления дискретных сущностей (пакетов, сообщений, единичных сигналов и т.д.) и математически описан как точечный процесс, содержащий последовательность поступающих сущностей $X_1, X_2, \ldots, X_n \ldots$, где $X_0=0$. В данном случае точечным процессом может быть процесс подсчета поступающих сущностей и времени между их поступлением.

Смешанный трафик будет содержать больше одного элемента в поступающей сущности X_n. Для описания смешанного трафика используется

неотрицательная случайная последовательность $B_1, B_2, \ldots, B_n$, где $n=1,2,\ldots,\infty$, а B_n – число элементов в поступающей сущности [15].

Модели трафика могут быть классифицированы по характеру процесса поступления сущностей и программному обеспечению или приложению, осуществляющему передачу данных. По характеру процесса модели могут быть стационарными и нестационарными. Стационарные модели, в свою очередь, могут обладать краткосрочной и долгосрочной зависимостью. К моделям с краткосрочной зависимостью относятся классические регрессионные и модели, основанные на марковских процессах. Долгосрочной зависимостью отличаются фрактальные модели [16]. Также по приложению-источнику трафик может быть классифицирован как трафик web, peer-to-peer, потокового видео и т.д. [17].

1.1.1 Модели, основанные на процессах восстановления

Одними из первых разработанных моделей трафика были модели, основанные на теории восстановления. Вследствие своей простоты они нашли широкое применение в исследованиях первых сетей передачи данных. Интервалы времени между событиями в процессе восстановления являются положительными, независимыми и равномерно распределенными величинами. Процесс восстановления может быть определен с помощью процесса подсчета $\{N(t); t\geq 0\}$, где $N(t)$ – это число событий системы на интервале $(0;t]$. На каждом периоде наступления событий $S_n=X_1+\ldots+X_n$ с определенной вероятностью процесс начинается заново. То есть, если n-е событие наступает при $S_{n=}\tau$, тогда, начиная с $S_n=\tau$, j-я подпоследовательность периода наступления событий: $S_{n+j}-S_n=X_{n+1}+\ldots+X_{n+j}$. Таким образом, при $S_n = \tau$, $\{N(\tau+t)-N(\tau); t\geq 0\}$ – считающая функция процесса восстановления с независимыми, равномерно распределенными интервалами между событиями [18].

Процесс восстановления несложно использовать, однако он обладает существенным недостатком – функция автокорреляции ряда $\{X_n\}$ обращается

в ноль для всех ненулевых лагов, что не соответствует результатам исследования реального трафика. То есть анализ АКФ в таком случае говорит об отсутствии временной зависимости временного ряда. Более того, положительное значение автокорреляции ряда $\{X_n\}$ может объяснить наличие коротких вспышек сетевой активности [19]. Именно трафик переменного характера, с периодами повышенной активности, превалирует в компьютерных сетях, особенно широковещательных, поэтому модель, учитывающая автокорреллированность данных, будет ближе соответствовать реальной сети.

Модель на основе распределения Пуассона. Модель трафика на основе распределения Пуассона является одной из первых и наиболее часто используемых. Она применялась в основном для исследования телефонных сетей. Пуассоновский процесс – это частный случай процесса восстановления, в котором время поступления событий экспоненциально распределено с параметром λ: $P\{X_n \leq t\} = 1 - exp(-\lambda t)$. Распределение Пуассона применимо, когда трафик поступает от совокупности независимых источников, которые отвечают требованиям распределения. Среднее значение и дисперсия распределения Пуассона определяются параметром λ.

Графически распределение Пуассона может быть представлено в виде ограниченного биномиального распределения. Распределение обладает рядом математических свойств. Во-первых, суперпозиция независимых пуассоновских процессов дает новый пуассоновский процесс с распределением, равным сумме распределений исходных процессов. Во-вторых, свойство независимых приращений устраняет временные зависимости ряда. В-третьих, согласно теореме Пальма, пуассоновский процесс зачастую используется для моделирования совокупности независимых источников трафика [20]. Однако позже выяснилось, что агрегирование трафика не всегда приводит к распределению Пуассона.

Функция распределения вероятности для пуассоновского процесса:

$$F(t) = 1 - e^{-\lambda t}. \quad (1)$$

Функция плотности распределения:

$$f(t) = \lambda e^{-\lambda t}. \quad (2)$$

Самый простой способ для определения того, что процесс пуассоновский – графический. Для этого достаточно определить, что гистограмма времени наступления событий убывает по экспоненциальному закону.

Стоит отметить, что в случае, если модель трафика на основе распределения Пуассона зависит от времени, то есть λ не постоянная величина, то параметр распределения выражается как функция от времени $\lambda(t)$ [21].

Для моделирования глобальных сетей, в которых вклад в общую картину трафика одного абонента невелик, пользовательские сессии могут быть представлены, как пуассоновский процесс. Пуассоновский процесс подходит для моделирования TCP трафика на сессионном уровне модели OSI, когда сессии инициируются пользователями, то есть TELNET и FTP приложения [22].

Модель на основе распределения Бернулли. Модель трафика на основе распределения Бернулли – это дискретный аналог модели Пуассона. Вероятность p наступления события в любой промежуток времени не зависит от других событий. Для временного промежутка k соответствующее число наступления событий отвечает биномиальному распределению:

$$P\{N_k = n\} = \left(\frac{k}{n}\right) p^n (1-p)^{k-n}, \quad (3)$$

где n принимает значения от 0 до k.

Время между наступлением событий определяется параметром p с геометрическим распределением

$$P\{A_n = j\} = p(1-p)^j, \quad (4)$$

где j – положительное целое число.

Модель на основе фазового процесса восстановления. Одной из моделей, основанных на процессе восстановления, является модель трафика так называемого фазового типа. Фазовый процесс наступления событий может быть смоделирован как непрерывный во времени марковский процесс поглощения $C = \{C(t)\}_{t=0}^{\infty}$ в пространстве допустимых состояний $\{0,1,...,m\}$, где 0 соответствует состоянию поглощения, а все остальные состояния - переходные, и при этом процесс поглощения выполняется в течение конечного промежутка времени. Для определения X_n процесс C запускается с начальным распределением π. Когда происходит поглощение (то есть процесс входит в состояние 0), процесс останавливается. Прошедшее время будет соответствовать X_n, что приведет к вероятностной комбинации сумм экспонент. Затем процесс выполняется заново с начальным распределением π, и процедура повторяется независимо для получения X_{n+1}.

Использование модели на основе фазового процесса восстановления позволяет управлять характеристиками моделируемой сети, а также удовлетворяет условию аппроксимации распределений входящих сообщений [19].

1.1.2 Модели, основанные на марковских процессах

Модели трафика, основанные на марковских процессах, вводят зависимость между элементами в случайную последовательность в отличие от моделей на основе процессов восстановления. Считается, что вероятность перехода системы в следующее состояние S_{n+1} зависит только от S_n и не зависит от других состояний S_i, где $i<n$. Это приводит к положительной автокорреляции в $\{S_n\}$, что соответствует переменному характеру сетевого трафика, когда периоды повышенной активности следуют за периодами пониженной интенсивности передачи данных. В таких моделях число

состояний конечно. Чем оно выше, тем сильнее модель соответствует реальной сети передачи данных, однако это вызывает повышение сложности моделирования.

Полумарковские модели получаются, когда время между наступлением состояний подчиняется случайному распределению вероятностей. Если время между сменами состояний модели не учитывается вовсе, то процесс считается дискретной марковской цепью.

ON-OFF и IPP модели. ON-OFF модель широко применяется для моделирования сетей передачи голосовых данных [23]. Модель используется, когда необходимо учесть скейлинговый характер сетевого трафика. При этом допускается всего два состояния ON и OFF, а время перехода между состояниями распределено по экспоненциальному закону [24]. Для сети, в которой N статистически идентичных и независимых ON-OFF источников, каждый источник характеризуется L – средним числом пакетов, переданных за ON период, пиковым значением S и средним r. Равновесная вероятность источника в таком случае может быть рассчитана как $\gamma=r/S$.

В рамках IPP-модели трафика (Interrupted Poisson Process) сеть может находиться только в двух состояниях. В ON-состоянии сеть осуществляет передачу данных в соответствии с распределением Пуассона, в OFF-состоянии передача данных не осуществляется.

Модель на основе марковского процесса восстановления. В рамках модели на основе процесса восстановления для сети есть два состояния: S_1 и S_2. Амплитуда трафика в состоянии S_1 равна 0 и 1 – в состоянии S_2. Если средние интервалы времени перехода между состояниями принять равными d_1 и d_2 соответственно, то вероятность нахождения системы в S_1 равна $P_{S1}=d_1/(d_1+d_2)$, а в S_2 – соответственно $P_{S2}=d_2/(d_1+d_2)$. При этом суперпозиция независимых процессов восстановления обладает биномиальным распределением.

Модель на основе марковского модулированного пуассоновского процесса (ММПП). В силу простоты реализации, пуассоновский процесс – сам по себе достаточно привлекательный способ моделирования сетевого трафика. Однако, очевиден недостаток такого подхода из-за использования постоянной скорости потока λ. Если взять реальный трафик речевых данных, то скорость потока не будет одинаковой, так как сообщения будут начинаться и заканчиваться в случайные моменты времени.

Примем N голосовых сообщений за мультиплексированный поток, а каждое отдельное сообщение – независимый пуассоновский процесс. Таким образом, базовый процесс – пуассоновский со скоростью $\lambda(t)$. Скорость потока модулируется как $\lambda(t)=n(t)\lambda$, где $n(t)$ – это число активных в данный момент передач голосовых сообщений. В этом случае, $n(t)$ – это состояние непрерывной во времени цепи Маркова. ММПП сохраняет некоторые свойства отсутствия временной зависимости пуассоновского процесса и может быть проанализирована в рамках марковской теории.

Марковский модулированный пуассоновский процесс широко используется при моделировании трафика благодаря высокой гибкости в качественной настройке полученной модели, его также называют двойным стохастическим процессом.

Простой пример марковского модулированного пуассоновского процесса – это модель с двумя состояниями: активным, с соответствующим положительным параметром распределения Пуассона и выключенным состоянием, при котором параметр Пуассона равен нулю. Состояние ON в таком случае соответствует передачи звука, а состояние OFF соответствует тишине. Такая модель может быть улучшена агрегированием множества независимых источников, каждый из которых характеризуется ММПП с индивидуальным модулирующим марковским процессом.

Марковская модулированная жидкостная модель. Жидкостные модели определяют трафик как непрерывный поток с параметром,

определяющим скорость этого потока. Подобные модели особенно подходят в случаях, когда влияние отдельного пакета на сеть передачи данных незначительно. Жидкостные модели отличаются от традиционных точечных тем, что игнорируют дискретную природу пакетов данных [22]. Название модели исходит из аналогии с влиянием одной молекулы жидкости в трубе с водой.

Обработка жидкостных моделей достаточно проста и не требует высоких вычислительных мощностей [16]. Чаще всего используется марковская жидкостная модель, при этом текущее состоянии марковской цепи определяет скорость потока (трафика). При моделировании VBR видео используется марковская модулированная модель с постоянной скоростью, в которой состоянию S_k соответствует постоянная скорость λ_k. Входным переменным параметром сети на основе жидкостной модели является скорость потока, на фоне которой можно проводить исследование поведения отдельных узлов сети. Чаще всего жидкостные модели учитывают “ON-OFF” природу источников сообщений, при этом OFF-период соответствует отсутствию трафика, а в течение ON- периода сообщения поступают детерминировано с постоянной скоростью. Периоды не зависят друг от друга, а их распределение подчиняется экспоненциальному закону. Простота моделирования и аналитической трактовки делает модель достаточно популярной.

1.1.3 Авторегрессионные модели

В рамах авторегрессионных моделей следующая случайная величина в последовательности рассчитывается как явная функция от последовательности предыдущих значений. Другими словами, значение случайной величины X_n основано на наборе предыдущих значений $\{X_k\}$, где $k<n$. Авторегрессионная модель порядка p обозначается как AR(p), а случайная величина может быть выражена следующим образом:

$$X_k = r_1 X_{k-1} + r_2 X_{k-2} + \ldots + r_p X_{k-p} + W_k, \quad (5)$$

где W_k – случайная величина (белый шум), r_i – вещественные числа, X_t – коррелированные случайные величины.

Автокорреляционная функция процесса AR(p) представляет собой затухающую синусоиду. Дискретная авторегрессионная модель порядка p генерирует стационарную последовательность случайных величин с распределением вероятности и АКФ как у авторегрессионной модели порядка p.

Одним из способов настройки подобной модели является выбор размера исторической выборки для определения новых случайных значений переменной.

Авторегрессионная линейная модель. Авторегрессионные модели широко применяются при моделировании VBR видео трафика для разработки систем управления перегрузками в высокоскоростных сетях для передачи мультимедиа данных [25]. Подобная популярность исходит из характера видео данных, где на 1 секунду приходится до 30 кадров и, как следствие, отличия в этих кадрах чаще всего незначительны. Существенные изменения между двумя последующими кадрами видео ряда вносят изменения сцены, которые вызовут рост объема переданных данных. Таким образом, видеоряд в рамках одной сцены, без резких скачков в объеме трафика может быть смоделирован с помощью авторегрессионных моделей, а для резких переходов между кадрами можно задействовать марковские цепи.

В работе [25] видео трафик моделируется в рамках выражения:

$$X_n = Y_n + Z_n + V_n C_n, \quad (6)$$

где Y_n и Z_n – два независимых AR(1) процесса. Благодаря использованию одновременно двух процессов авторегрессии удается привести автокорреляционную функцию (АКФ) получившейся модели к виду,

соответствующему реальному видео трафику. Произведение V_nC_n – это состояние марковской цепи и независимой нормально распределенной случайной величины, которое вводится для учета скачкообразного роста нагрузки при смене сцен. Подобная модель применима для алгоритмов сжатия видео, при которых передаются только изменения при переходе между кадрами.

Дискретная авторегрессионная модель. Дискретная авторегрессионная модель порядка p генерирует стационарную последовательность дискретных случайных величин со случайным распределением вероятности и АКФ как у авторегрессионного процесса порядка p (AR(p)).

Дискретный авторегрессионный процесс первого порядка – это частный случай DAR(p) процесса, который определяется на основе двух последовательностей независимых случайных величин $\{V_n\}$ и $\{Y_n\}$. Случайная величина V_n принимает одно из двух значений, 0 или 1 с вероятностью $(1–\rho)$ и ρ соответственно. Случайная величина Y_n имеет дискретный набор состояний, а в матрице переходов $P\{Y_n=i\}=\pi(i)$. Случайная величина X_n формируется исходя из выражения DAR-процесса первого порядка:

$$X_n=V_nX_{n-1}+(1-V_n)Y_n. \quad (7)$$

DAR процесс первого порядка – это цепь Маркова с дискретным набором состояний S и матрицей переходов

$$P= \rho I + (1– \rho)Q, \quad (8)$$

где I – это единичная матрица, а Q – матрица с $Q_{ij} = \pi(j)$ для $i, j \in S$.

DAR-процесс первого порядка обладает АКФ авторегрессионного процесса первого порядка, а функция распределения вероятности – это функция от π.

DAR(1) процесс обладает меньшим, по сравнению с цепью Маркова, числом параметров, а оценка этих параметров не вызывает затруднений. Модель трафика на основе DAR процесса легко поддается аналитическому анализу, однако из-за экспоненциального затухания функции автокорреляции модель не применима к трафику с медленно затухающей АКФ.

Авторегрессионная модель скользящего среднего. Авторегрессионная модель скользящего среднего порядка (p,q) обозначается как ARMA(p,q) и принимает следующий вид:

$$X_t=c+\varphi_1 X_{t-1}+ \varphi_2 X_{t-2}+\ldots+ \varphi_p X_{t-p}+\varepsilon_t - \theta_1 \varepsilon_{t-1}- \theta_2 \varepsilon_{t-2}-\ldots- \theta_p \varepsilon_{t-p}. \quad (9)$$

Это эквивалентно следующей записи:

$$\varphi(B)X_t=c+\theta(B)\,\varepsilon_t, \quad (10)$$

где B – это оператор лага (оператор сдвига), такой что $X_{t-1}=BX_t$, а $\varphi(B)=(1-\varphi_1 B^1-\ldots-\varphi_p B^p)$, c – константа.

Это эквивалентно фильтрации белого шума ε_t линейным фильтром, инвариантным к сдвигу по времени, который имеет дробно-рациональную передаточную функцию с p полюсами и q нулями [7], т.е.:

$$H(z)=\frac{B_q(z)}{A_p(z)}=\frac{1-\sum_{k=0}^{q}\theta_k z^{-k}}{1-\sum_{k=1}^{p}\varphi_k z^{-k}}\,. \quad (11)$$

Автоковариация ARMA(p,q) процесса может быть получена произведением (11) и X_{t-k} с учетом математического ожидания и взаимной корреляции между ε_t и X_t. ARMA модели широко применяются для моделирования VBR трафика. В этом случае длительность видеокадра делится на равные m интервалы. Число ячеек n_i, i=0,..,m-1, во временном интервале моделируется с помощью ARMA процесса:

$$X_n=\varphi X_{n-m}+\sum_{i=0}^{m-1}\theta_i\varepsilon_{n-i}\,. \quad (12)$$

Так как видеоданные каждого кадра коррелированны между собой с переменным коэффициентом корреляции, то функция автокорреляции будет содержать пики на лагах, кратных *m*. В рассматриваемой модели AR-часть используется для эффекта повторной корреляции, а θ_k вводится для подбора корреляции для других задержек [7]. Параметрическая оценка ARMA моделей сложнее, чем для AR моделей, оценка θ_k требует решения множества нелинейных уравнений. Аналитический анализ также достаточно затруднителен.

Интегральная модель авторегрессии – скользящего среднего. Авторегрессионный интегральный процесс скользящего среднего порядка (*p*,*d*,*q*), обозначается как ARIMA(*p*,*d*,*q*) и строится на основе ARMA(*p*,*q*). То есть ARIMA(*p*,*d*,*q)* может быть интерпретирована как ARMA(*p*+*d*,*q*) – модель с *d* единичными корнями, остальные корни многочлена лежат за пределами единичной окружности. ARIMA(*p*,*d*,*q*) процесс может быть записан следующим образом:

$$\phi(B)\nabla^d X_t = \theta(B)\varepsilon_t, \tag{13}$$

где ∇ - оператор дифференцирования, такой что (X_t-X_{t-1}) = ∇X_t и ∇X_t=(*1*-*B*)X_t, а $\phi(B)$ – многочлен от *B*. ARIMA(*p*,*d*,*q*) процесс используется для моделирования нестационарных рядов, которые проявляют однородность в отличие от их локального уровня или тренда. Как правило, используют ARIMA с *d* равным 1 или 0: если *d=0,* то процесс обладает линейным трендом, а если *d*>1, то полиномиальным.

Модель расширения и преобразования выборки. Модель расширения и преобразования выборки (TES) – это нелинейная регрессионная модель, которая нацелена на воспроизведения предоставленной стационарной выборки с соблюдением маргинального распределения и структуры АКФ [26]. Другими словами, TES генерирует последовательность, управляемую случайным процессом, сглаженную и адаптированную под данное

маргинальное распределение. Качество моделирования трафика телекоммуникационных приложений сильно зависит от того, насколько смоделированный временной ряд отвечает основным характеристикам входной последовательности. Наиболее важных характеристик три: маргинальное распределение, структура АКФ, соответствие между графическим представлением исходных данных и смоделированного ряда. Общий алгоритм моделирования временного ряда представлен на рисунке 1.

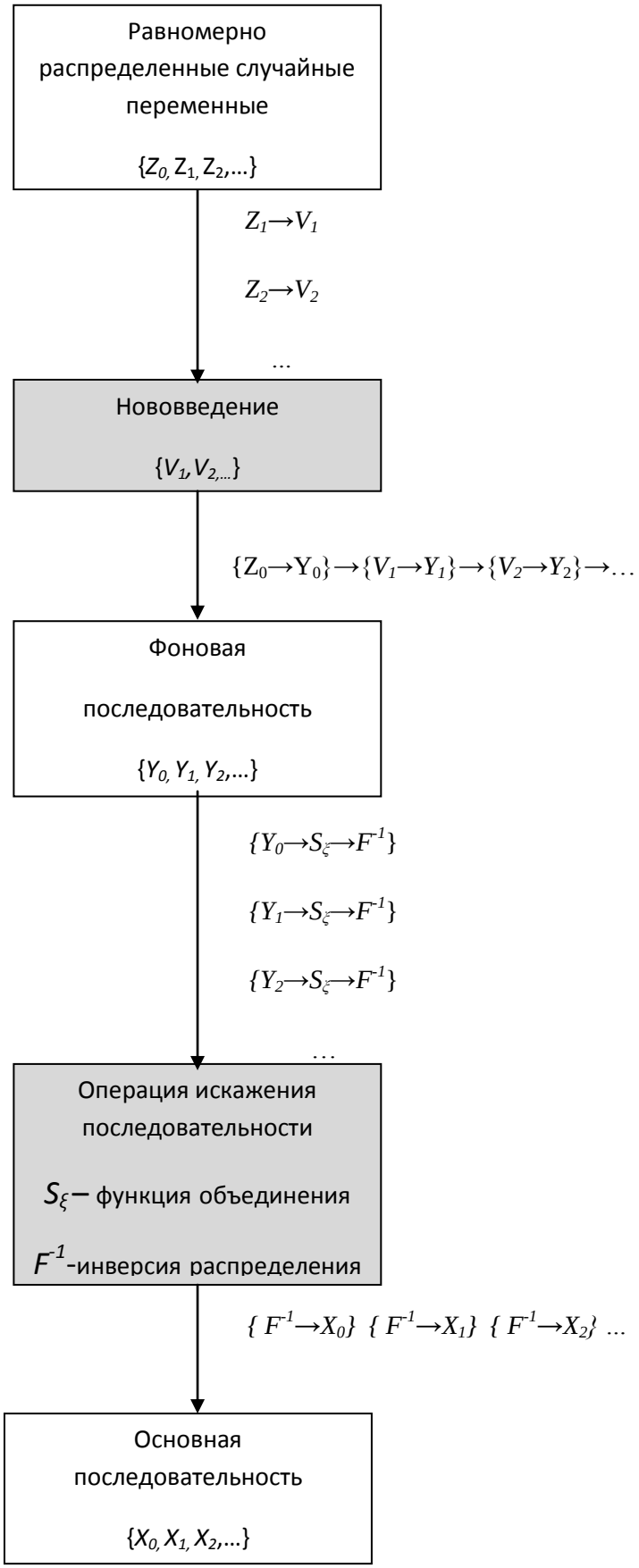


Рисунок 1. Алгоритм TES

На первом этапе происходит инициализация ряда $\{Z_i\} \in [0;1]$. После чего Z_i используется для выбора $V_i \in [-0.5;0.5]$ в соответствии с заданной плотностью. Величина V_i прибавляется к значению фоновой последовательности Y_{i-1} и формирует таким образом $Y_i = |Y_{i-1} + V_i|_{mod=1}$. В рамках алгоритма преобразование фоновой последовательности происходит дважды. Во-первых, расчеты по модулю 1 Y_i приводят к некоторому эффекту разрыва, если допустить, что значения Y_i могут быть достаточно близки к 1. Также небольшие значения V_{i+1} могут привести к близости Y_{i+1} нулю, то есть серьезным изменениям в фоновой последовательности. Подобные эффекты устраняются функцией объединения (оконной функцией) *S*, которая значения близкие к 0 и 1 устанавливает около 0:

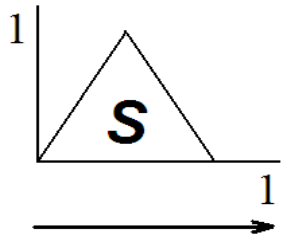


Рисунок 2. Функция объединения S

Второе преобразование заключается в инверсии гистограммы эмпирических данных, что приводит объединенную последовательность ближе к диапазону значений исходного ряда и устанавливает необходимое маргинальное распределение. Это последний этап моделирования эмпирического временного ряда, который генерирует основную последовательность. Отметим, что одна из основных особенностей TES модели заключается в возможности последующих расчетов лага АКФ искажения и плотности распределения нововведений.

1.1.4 Самоподобные модели

Пусть $\{X_t\}$ – стационарный в широком смысле стохастический процесс, обладающий стационарным средним $\mu=E[X_t]$, стационарной и конечной дисперсией $v=E[(X_t - \mu)^2]$ и стационарной АКФ $\gamma_k=E[(X_t - \mu)(X_{t+k} - \mu)]$, которая зависит только от k и не зависит от t. Отметим, что $v= \gamma_0$. Если принять $\{X_k\}$ при лаге k за ρ_k, то по определению $\rho_k= \gamma_k/ \gamma_0$.

Разделим исходный временной ряд $\{X_t\}$ на непересекающиеся последовательности длинной m и усредним их, чтобы получить последовательность $\{X_j^{(m)}\}$:

$$X_j^{(m)}=m^{-1}(X_{jm-m+1}+\ldots+X_{jm}), \tag{14}$$

где $X_j^{(m)}$ – это выборочное среднее последовательности $X_{jm-m+1}+\ldots+X_{jm}$. Обозначим v_m как дисперсию $\{X_j^{(m)}\}$:

$$v_m = E\left[\frac{1}{m}\left(X_{jm-m+1}+\ldots+X_{jm}\right)\right]^2\left[E\frac{1}{m}(X_{jm-m+1}+\ldots+X_{jm})\right]^2 =$$

$$= \frac{v}{m}+\frac{2}{m^2}\sum_{k=1}^{m}(m-k)\gamma_k =$$

$$= v\left[1+2\sum_{k=1}^{m}\left(1-\frac{k}{m}\right)\rho_k\right]m^{-1}. \tag{15}$$

В таком случае, если процесс является белым шумом, то $X_{jm-m+1}+\ldots+X_{jm}$ будут взаимно некоррелированными, $\rho_k=0$ для $k>0$ и $v_m=vm^{-1}$.

Для больших m можно усреднить:

$$v_m = v\left[2\sum_{k=1}^{m}\rho_k\right]m^{-1} \tag{16}$$

Рассмотрим случай, при котором $\rho_k \neq 0$ и $\sum_{k=-\infty}^{\infty} \rho_k < \infty$. Тогда дисперсия выборочного среднего будет асимптотически затухать и стремиться к нулю пропорционально m^{-1}, т.е.:

$$v_m \approx v c_\rho m^{-1}, \tag{17}$$

где c_ρ – константа.

Для большинства рассмотренных выше моделей, таких как ARMA или моделей на основе марковских процессов, средняя дисперсия выборки затухает именно в соответствии с (17).

В работе [5] для исследованного трафика указывается затухание средней дисперсии выборки даже медленнее, чем m^{-1}. Наиболее простым подходом в данном случае было бы принять затухание v_m пропорционально $m^{-\alpha}$ для некоторой $\alpha \in (0,1)$. Тогда $\sum_{k=1}^{m} \rho_k$ должна быть пропорциональна $m^{1-\alpha}$:

$$\sum_{k=1}^{m} \rho_k \approx C m^{1-\alpha}. \tag{18}$$

Так как $\alpha<1$, то $\sum_{k=-\infty}^{\infty} \rho_k \to \infty$. Таким образом АКФ затухает медленнее, потому что она не суммируется [24].

Процесс $\{X_t\}$ обладает краткосрочной зависимостью, если $\sum_k \rho_k < \infty$. Соответственно, v_m затухает асимптотически пропорционально m^{-1}, спектральная плотность мощности в нуле обращается в бесконечность, а усредненные последовательности $\{X_t^{(m)}\}$ стремятся к чистому шуму при $m \to \infty$. АКФ процессов, обладающих краткосрочной зависимостью, затухает экспоненциально.

Процесс $\{X_t\}$ обладает долговременной зависимостью, если $\sum_k \rho_k \to \infty$. Дисперсия среднего v_m затухает медленнее, чем m^{-1}. При этом долгосрочная

зависимость лишь определяет поведение АКФ при больших лагах, но не определяет автокорреляцию ряда для любого фиксированного конечного лага.

Процесс $\{X_t\}$ самоподобен, если $\rho_k^{(m)} = \rho_k$ для всех m и k, то есть структура АКФ сохраняется на разных временных масштабах. Стохастический самоподобный процесс сохраняет одинаковые статистические показатели на любом диапазоне масштабов и удовлетворяет следующему выражению:

$$\{X_{\alpha t}\} \overset{D}{=} \alpha^H \{X_t\}, \tag{19}$$

где $\overset{D}{=}$ означает равенство в распределении, а H называется параметром Херста [27].

Если процесс $\{X_t\}$ обладает стационарным инкрементом: $Y_t = X_t - X_{t-1}$, то АКФ:

$$\rho_k \to H(2H-1)k^{2H-2} \text{ при } k \to \infty, \tag{20}$$

где $H=1-\alpha/2$, а при $0<H<1$ и $H \neq 1/2$, $\sum_k \rho_k \to \infty$.

Фрактальная интегрированная модель авторегрессии – скользящего среднего. Фрактальная интегрированная модель авторегрессии скользящего среднего – ARFIMA(p,d,q) при $0<d<1/2$ – это пример модели стационарного процесса с долговременной зависимостью. Она представляет собой расширение ARIMA (p,d,q) модели и определяется следующим образом:

$$\phi(B)\nabla^d X_t = \theta(B)\varepsilon_t, \tag{21}$$

где $0<d<1/2$. Оператор $\nabla^d = (1-B)^d$ можно выразить через биномиальное разложение

$$(1-B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-1)^k B^k , \tag{22}$$

$$\binom{d}{k} = \frac{d!}{k!(d-k)!} = \frac{\Gamma(d+1)}{\Gamma(k+1)\Gamma(d-k+1)}, \tag{23}$$

где $\Gamma(x)$ – гамма-функция.

Так как гамма-функция обладает полюсами для отрицательных целых чисел. и биномиальные коэффициенты равны нулю, если $k>d$ и d – целое число, то для всех положительных целых чисел, только первые $d+1$ будут ненулевыми.

С помощью ARFIMA можно моделировать процессы с краткосрочной и долгосрочной зависимостью.

Модель фрактального броуновского движения. Броуновское движение это стохастический процесс $\{B_t\}$ для $t\geq 0$. Характерны следующие свойства:

- Инкремент (B_{t+t0}-B_{t0}) обладает нормальным распределением с нулевым средним и дисперсией $\sigma^2 t^{2H}$.
- Инкремент на двух непересекающихся интервалах $[t_1,t_2]$ и $[t_3,t_4]$, B_{t4}-B_{t3} и B_{t2}-B_{t1} – независимые случайные величины.
- $B_0=0$ и B_t непрерывна при $t=0$ [28].

Фрактальное броуновское движение $\{fB_t\}$ – это самоподобный гауссовский процесс с параметром самоподобия $0{,}5\leq H<1$. Фрактальное броуновское движение отличается от броуновского движения тем, что дисперсия инкремента равна $\sigma^2 t^{2H}$. Определим дисперсию инкремента

$$\sigma^2 = E\{(fB_t - fB_{t-1})^2\} = E\{(fB_1 - fB_0)^2\} = E\{fB_1{}^2\}, \tag{24}$$

тогда

$$E\{(fB_{t2} - fB_{t1})^2\} = E\{(fB_{t2-t1} - fB_0)^2\} = \sigma^2 (t2-t1)^{2H} \tag{25}$$

Также

$$E\{(fB_{t2} - fB_{t1})^2\} = E\{fB_{t2}^2\} + E\{fB_{t1}^2\} - 2E\{fB_{t2}\, fB_{t1}\} = \sigma^2 t 2^{2H} + \sigma^2 t 2^{2H} - 2\gamma(fB_{t1}, fB_{t2})$$

$$\gamma(fB_{t1}, fB_{t2}) = \frac{1}{2}\sigma^2 (t_2^{2H} - (t_2 - t_1)^{2H} + t_1^{2H}) \qquad (26)$$

Поэтому ковариация инкремента на двух непересекающихся интервалах:

$$\gamma(fB_{t4} - fB_{t3}, fB_{t2} - fB_{t1}) = \gamma(fB_{t4}, fB_{t2}) - \gamma(fB_{t4}, fB_{t1}) - \gamma(fB_{t3}, fB_{t2}) + \gamma(fB_{t3}, fB_{t1}) =$$

$$= \frac{\sigma^2}{2}(t_4 - t_1)^{2H} - (t_3 - t_1)^{2H} + (t_3 - t_2)^{2H} - (t_4 - t_2)^{2H}. \qquad (27)$$

Фрактальное броуновское движение $\{fB_t\}$ может быть получено из броуновского движения $\{B_t\}$ интегрированием:

$$fB_t = \int_0^t (t-u)^{H-0.5}\, dB(u), \qquad (28)$$

таким образом, взаимозависимость между инкрементами фрактального броуновского движения может быть бесконечной [29].

В случае дискретного процесса автокорреляция инкремента может быть рассчитана заменой t_1, t_2, t_3, t_4 на n, $n+1$, $n+k$, $n+k+1$ и делением на σ^2:

$$\rho_k = \frac{1}{2}[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}] \qquad (29)$$

Процесс инкрементирования в данном случае – это фрактальный гауссовский шум. Автокорреляция в последнем выражении обладает долговременной зависимостью, $\rho^k \sim k^{2H-2}$, $k \to \infty$. Один из примеров моделирования трафика с помощью фрактального гауссовского шума приведен в работе [26], где также указывается на сложность аналитического анализа распределения заполнения буфера. Поэтому приводится приблизительный анализ поведения хвостов распределений. Показано, что для больших значений H рост нагрузки на сеть требует значительного

увеличения объемов памяти. Вероятность потери пакета растет алгебраически с размером буфера, а не экспоненциально, как в случае с марковскими и ARMA моделями.

1.1.5 Модели трафика приложений

Очевидно, что процесс передачи данных в сети и характер самих данных определяется используемыми приложениями. Большинство приложений можно отнести к следующим типам: Web, e-mail, peer-to-peer файлообменники. потоковое мультимедиа. В работе [30] указывается, что более 40% всех данных, передаваемых по сети, относится к Web.

Вполне возможно применить рассмотренные выше подходы для моделирования трафика приложений, однако, логичнее использовать специальные модели для получения лучших результатов.

Web трафик. Во множестве работ [31,32] упоминается, что Web – трафик занимает большую часть всего трафика сети Internet, что неудивительно с учетом того, что Web-браузеры предоставляют дружелюбный интерфейс для работы с электронной почтой, передачи файлов, удаленной обработки данных, потокового медиа и др.

Одни из первых исследований Web-трафика на самоподобие указывают коэффициент Херста близкий к 0,8 [6]. Помимо характера самого трафика было указано, что Web клиенты могут быть представлены как ON-OFF источники сообщений, обладающие распределением с тяжелым хвостом. При этом самоподобие трафика возникает при агрегировании множества потоков данных с распределением с тяжелым хвостом. Плотность распределения вероятности, соответствующая PTX, должна обладать степенным характером. «Тяжесть» хвоста означает, что экстремально большие значения ряда обладают ненулевой вероятностью на гистограмме распределения. OFF – периоды могут быть либо активными, когда загрузка больших объемов данных не производится, но пользователь работает с Web-браузером, либо неактивными, когда пользователь не использует Web-

браузер. Активные OFF периоды могут быть описаны распределением Вейбулла, а неактивные – Парето распределением с тяжелым хвостом.

В работе [33] проводилось исследование 1900 клиентских Web-браузеров, для которых также была характерна ON/OFF модель. OFF-период описывался распределением Вейбулла и ему соответствовал период, во время которого пользователь просматривал загруженный контент. ON-период соответствовал загрузке страницы, однако процесс рассматривался более детально. Загрузка основного кода страницы и дополнительных объектов описывалась отдельными гамма-распределениями.

В работе [34] проводился суточный сбор трафика с роутеров корпоративной сети. Согласно исследованию, 42% всего потока данных принадлежало Web-трафику. Были рассмотрены плотности распределения числа серверов, с которыми обменивался данными каждый клиент, а также число клиентов, обслуживаемых одним сервером, объемы трафика, переданные каждым клиентом, объемы переданных серверами данных. Одним из заключений стал тот факт, что крайне сложно выделить некоторое типичное поведение отдельного клиента или сервера.

Трафик одноранговых сетей. Peer-to-peer (p2p) или трафик одноранговых сетей часто сравнивают с Web-трафиком по той причине, что одноранговые сети представляют собой полную противоположность клиент-серверной архитектуре. На данный момент не так много работ посвящено исследованию p2p трафика.

Исследование университетской сети проводилось в работе [35]. Большая часть передаваемых данных относилась к видео-форматам AVI и MPEG, а также MP3. При этом наибольшую нагрузку давали несколько компьютеров в сети, на которых были расположены файлы более 700 Мб, доступ к которым осуществлялся остальными членами сети. Таким образом, небольшое число узлов в сети ответственны за основную долю нагрузки.

В другой работе [36] осуществлялся сбор p2p трафика через роутеры корпоративной сети. Вновь было обнаружено, что наибольшую нагрузку в сети создает несколько компьютеров.

Трафик видеоданных. Характеристики трафика сжатых видеоданных могут сильно отличаться в зависимости от характера самого видеоряда и частоты смены кадров, а также алгоритмов кодирования записи. Для большинства алгоритмов кодирования записи характерна регистрация лишь изменений между последовательно идущими кадрами без дублирующегося описания неизменной сцены. Поэтому более динамичное видео вызывает больший объем переданных данных, а резкая смена сцен может быть отмечена всплесками трафика.

Большинство источников видеоданных характеризуются скоростью передачи кадров. А модели, как правило, учитывают такие статистические характеристики, как плотность распределения вероятности объема данных, приходящихся на 1 кадр, характер АКФ ряда кадров, среднее время между сменой сцен, плотность распределения вероятности длины передаваемых файлов.

Существует множество видео – кодеков и алгоритмов кодирования видеоданных, поэтому достаточно сложно найти единую универсальную модель трафика видео. Разным файлам будут соответствовать разные модели, однако некоторые принципы можно обобщить [37]:

- Для описания объема данных, приходящихся на один кадр, как правило, используется логнормальное, гамма или распределение Парето.

- АКФ нединамичного видео принимает экспоненциальный вид. АКФ для широковещательного видео обладает более сложной структурой с быстро убывающей зависимостью на маленьких лагах и долговременной зависимостью на больших значениях лага.

- Распределение длин сцен обычно соответствует распределению Парето или Вейбулла, реже – гамма-распределению.
- Объем данных, приходящийся на одну сцену, некоррелирован и соответствует распределению Вейбулла или гамма-распределению.
- Длина потоковой Web-передачи видео соответствует распределению Парето с тяжелым хвостом [38].

Рассмотрены различные модели трафика, как разработанные достаточно давно и легшие в основу современных алгоритмов управления трафиком, так и актуальные модели, учитывающие нелинейно – динамические свойства трафика, распределение с тяжелым хвостом и медленно убывающую зависимость. Далее рассмотрим реализованные в современных протоколах связи алгоритмы управления трафиком для последующей разработки усовершенствованной методики борьбы с перегрузками.

1.2 Борьба с перегрузками в сетях передачи данных

Большинство способов управления трафиком можно разделить на алгоритмы с обратной связью и без обратной связи. Методики без обратной связи не учитывают текущего состояния сети, принимая решения об отбрасывании пакетов и составляя расписания работы отдельных участков сети. В данном случае достаточно трудной является задача определения причины задержки или утери пакета, например, в беспроводных сетях это может быть связано не с перегрузками, а с искажением сигнала из-за помех. В качестве примера можно обозначить алгоритм маркерной корзины [4].

Алгоритмы с обратной связью учитывают текущее состояние сети, ведут мониторинг возникновения перегрузок, выполняют оповещения других участков сети и принимают меры по устранению перегрузок. Мониторинг сети заключается в регистрации следующих показателей:

- Процент пакетов, не принятых к обработке из-за переполнения буфера;
- средняя длина очереди;
- процент пакетов, переданных повторно из-за отсутствия подтверждения о получении;
- среднее (среднеквадратичное отклонение) времени задержки пакетов.

Увеличение значений обозначенных показателей говорит о нарастающей нагрузке либо о наличии перегрузки. Алгоритмы явной обратной связи могут быть бинарными, то есть прямо указывать на наличие перегрузки, либо предоставлять информацию о конкретном уровне нагрузки. Примером реализации для TCP/IP сетей служит Explicit Congestion Notification и ICMP Source Quench [39].

Существует ряд общих алгоритмов борьбы с перегрузками. Наиболее очевидный заключается в том, чтобы узел сети, обнаруживший перегрузку, отправил соответствующее сообщение источнику данных о необходимости снижения скорости передачи. Однако такой подход в ряде случае только усугубляет ситуацию, увеличивая нагрузку на сеть новыми сообщениями, которые возможно вовсе не будут доставлены до источника трафика. Другое решение заключается в том, чтобы зарезервировать в заголовке сообщения поле, которое будет заполняться узлами сети при значительном повышении нагрузки сети, такой подход реализуют протоколы Frame Relay [5], DCCP [6], а также протоколы ATM [7]. Еще один подход к борьбе с перегрузками заключается в том, чтобы маршрутизаторы периодически отправляли пробные сообщения с целью выявить нагруженные участки сети [39]. Классификация алгоритмов борьбы с перегрузками (Рисунок 1) приводится в работе [41].

1.2.1 Методы без обратной связи

Борьба с перегрузками без обратной связи не подразумевает прямое оповещение узлов сети о наличии перегрузки для принятия соответствующих мер. Вместо этого источник сообщений, приемник, либо сетевое устройство локально ведет мониторинг и управление процессом передачи трафика. Другими словами, методы управления трафиком без обратной связи в первую очередь служат цели не допущения перегрузки, нежели управления сетью после возникновения перегрузки.

Принцип справедливого распределения ресурсов. Маршрутизатор, или другое сетевое коммутирующее оборудование, может обрабатывать сообщения из разных подсетей, множества компьютеров и в конечном итоге распределенных приложений. Классическая схема коммутатора без реализации функции QoS может быть описана в рамках теории массового

обслуживания, элементом сети с приоритетом обслуживания FIFO. При такой схеме обработки пакетов может сложиться ситуация, когда одним

абонентом будет инициирован резкий всплеск трафика, что приведет к заполнению буфера коммутатора и последующему отбрасыванию пакетов других абонентов.

Рисунок 3. Классификация методов управления перегрузками

Во избежание монопольного захвата сетевого оборудования одним из источников трафика разработан ряд алгоритмов управления очередью в рамках принципа справедливого распределения ресурсов. Базовый алгоритм был сформулирован в работе [42]: допустимая пропускная способность каждого потока данных, проходящего через критический участок сети, не должна быть меньше допустимой пропускной способности других потоков,

разделяющих данный участок сети. Форумом IETF и ATM были обозначены следующие требования к справедливому разделению ресурсов между потоками:

- Очередность выделения ресурсов определяется очередностью поступления запросов;
- пользователю не выделяет больше ресурсов, чем им было запрошено;
- всем пользователям, запросы, на пропускную способность которых превысили допустимый объем, предоставляется одинаковая пропускная полоса.

Таким образом, каждый новый зарегистрированный поток получает полосу пропускания, равную пользователю с минимальными требованиями. При этом оставшиеся свободные ресурсы разделяются между потоками, требования которых превысили присвоенную минимальную величину. Если после процедуры разделения пропускной способности необходимо выделить ресурсы под еще один поток, то ограничиваются ресурсы потоков, превышающих минимальное значение, так как считается, что безопаснее ограничить более ресурсоемкий поток, чем поток и так использующий минимальную пропускную полосу [43]. Известной модификацией является взвешенный принцип справедливого распределения ресурсов. При этом каждому потоку устанавливается весовой коэффициент, в зависимости от которого неудовлетворенным потокам достается разное количество ресурсов.

Алгоритмы обслуживания пакетов в очередях. Ранее был представлен принцип справедливого распределения ресурсов коммутирующего устройства, то есть требование к формированию и обслуживанию очередей коммутатора. Рассмотрим далее алгоритмы, согласно которым происходит обслуживание пакетов, помещенных в очереди буфера коммутатора.

Наиболее простой алгоритм обработки пактов в очередях буфера заключается в присвоении приоритета каждой очереди на её обслуживание. В таком случае, если одна из очередей пуста, то следующим будет обработан пакет из очереди с низшим приоритетом. Очевидно, что при кажущейся простоте реализации подобный подход к обработке пакетов обладает существенным недостатком – возможна ситуация, при которой высокоприоритетный поток монополизирует ресурсы коммутатора. Таким образом, при разработке алгоритма обработки пакетов, содержащихся в очередях коммутатора, необходимо использовать алгоритмы сглаживания профиля нагрузки, которые будут рассмотрены позже, а также использовать справедливый принцип доступа обработчика пакетов к очередям сетевого устройства.

В работе [44] представлен алгоритм GPS (Generalized Processor Sharing) для доступа к очередям маршрутизатора в рамках принципа справедливого распределения ресурсов. Зарезервированная скорость для каждого потока или очереди в рамках GPS рассчитывается согласно выражению:

$$g_i = r\frac{r_i}{\sum_j r_j}, \tag{30}$$

где r – пропускная способность исходящего канала, r_i – минимальная пропускная способность канала, такая что $\sum_{i=1}^{N} r_i \le r$, N – число каналов, а $\sum_j r_j$ - сумма скоростей обслуживания потоков, не обслуживаемых в данный момент времени.

Алгоритм GPS обладает рядом свойств:

- Реализация GPS позволяет обеспечить постоянную скорость потока r_i. До тех пор, пока $r_i \le g_i$, потоку гарантирована скорость обслуживания, не зависящая от других потоков. При превышении скорости $\ge g_i$, очередь с необработанными пакетами очищается.

- Скорость доставки пакетов одного потока зависит только от длины его очереди и не зависит от других потоков.

- Гибкость системы обеспечивается переменным значением r_i, таким, что сумма r_i должна быть ниже пропускной способности исходящего канала.

- Позволяет обеспечить постоянную задержку для источников трафика, что особенно важно для потокового мультимедиа, видео, голосовых данных.

Кроме обеспечения постоянной скорости потока, алгоритм GPS рассчитывает для каждого поступившего пакета время окончания обслуживания. После завершения обработки текущего пакета, следующим берется тот пакет, который будет обслужен быстрее, при этом рассматриваются пакеты, стоящие первыми в очереди.

Следует отметить, что в GPS при расчете времени обслуживания пакета его размер принимается бесконечно малым [3]. Аппроксимирующие алгоритмы моделируют GPS сервер в непрерывном времени, как, например, алгоритм взвешенной справедливой очереди WFQ (Weighted Fair Queue). В случае WFQ каждому потоку отводится одна очередь и гарантируется некоторая скорость передачи в соответствии с весовым коэффициентом потока. В первую очередь обслуживаются очереди, не исчерпывающие предоставленные им полосы пропускания, при этом оставшаяся полоса делится между остальными потоками. Наиболее распространенная схема организации обслуживания буфера коммутатора заключается в определении одной приоритетной очереди, которая обслуживается до тех пор, пока все её пакеты не будут переданы, остальные очереди просматриваются последовательно в зависимости от их весов.

В общем случае функционирования WFQ на обслуживание передается пакет, время обслуживания которого, рассчитанное моделированием GPS

сервера, окажется минимальным. Подобный подход приводит к пачечности исходящего потока. Если на обслуживание в первую очередь буфера будут поступать пакеты, время обслуживания которых меньше, чем в других очередях, то пакеты первого буфера монополизируют ресурсы обработчика очередей. Модификацией алгоритма WFQ является алгоритм WWFQ (worst-case fair weighted fair queuing). Выбор пакета для обслуживания в алгоритме WWFQ происходит без учета очереди, из которой был взят последний обслуженный пакет. Таким образом, если одна из очередей содержит пакеты, время обслуживания которых минимально, то не происходит монопольного захвата ресурсов коммутатора одним потоком и трафик на выходе коммутатора получается более сглаженным.

Алгоритмы управления очередями. Ранее были рассмотрены алгоритмы управления трафиком без обратной связи на уровне источника сообщений, где под источником сообщения подразумевается выходной порт коммутирующего устройства. Далее будут рассмотрены алгоритмы управления трафиком на уровне приемника сообщений, то есть принципы фильтрации пакетов на входном буфере коммутатора для предотвращения перегрузки.

Наиболее простой алгоритм TailDrop был одним из первых разработанных, согласно TailDrop при переполнении буфера, либо заполнении до критического объема, новые пакеты просто отбрасывались. Увеличение размеров буфера при этом приводит к повышению времени обработки пакета, что приводит к уменьшению размера окна передачи для протокола TCP. Происходит так называемая синхронизация, когда источники сообщений, проходящие через нагруженный участок сети, уменьшают окно передачи, что приводит к простою ресурсов сети. После чего окно передачи вновь увеличивается и нагрузка вновь возрастает. Существует несколько вариаций TailDrop [45]. Алгоритм случайного отбрасывания пакета при переполнении буфера с определенной вероятностью отфильтровывает

пакеты из очереди при критичном повышении нагрузки. Алгоритм сброса начала очереди с определенной вероятностью отбрасывает пакеты, стоящие в начале очереди. Приведенные алгоритмы в некоторой степени позволяют бороться с пачечностью трафика и монополизацией ресурсов коммутатора, тем не менее, разработан ряд методик, лучше справляющихся с поставленной задачей [19].

Алгоритм Random Early Detection (RED) впервые был предложен в [46] и положил начало целому ряду работ и модификаций, которые будут рассмотрены далее. Оригинальный алгоритм предполагает вычисление вероятности фильтрации пакета таким образом, чтобы не нарушать принципа справедливого распределения ресурсов, монопольного захвата ресурсов коммутатора одним потоком и избежать синхронизации источников сообщений, рассмотренной ранее. При поступлении нового пакета в буфер коммутатора, RED рассчитывает средний размер буфера и сравнивает это значение с ранее установленной нижней и верхней границей. Если средняя длина очереди ниже минимального значения, то пакет отправляется в очередь. Если средняя длина очереди лежит в пределах нижней и верхней границы, то с некоторой вероятностью P пакет отбрасывается, а с вероятностью $(1-P)$ попадает в очередь. Если средняя длина очереди выше верхней границы длины, то пакет однозначно отбрасывается. Средний размер очереди рассчитывается методом экспоненциального взвешенного среднего значения предыдущих размеров очереди. Это делается для того, чтобы не учитывать кратковременные заполнения буфера. Весовой коэффициент не позволяет заметно реагировать на кратковременную нагрузку.

Как можно заметить, алгоритм установки пороговых значений длины буфера и весового коэффициента для расчета средней длины играет большую роль в работе RED, но не описывается явным образом. Так в [47] представлен алгоритм ARED. В ARED верхнее и нижнее пороговое значение длины очереди зависит от нагрузки. Если средняя длина очереди становится меньше

нижнего порогового значения, то вероятность сброса пакета также понижается, считается, что нагрузка невелика. Как только нагрузка на коммутирующее устройство повышается, растет и вероятность сброса пакета. При понижении интенсивности трафика после перегрузки ARED показывает лучшие результаты в сравнении с RED, так как адаптивный алгоритм снижает вероятность сброса пакетов при повышенной, но снижающейся нагрузке.

Разработан ряд модификаций алгоритма RED:

- WRED (Weighted RED) [48] использует возможность протоколов TCP/IP задать в заголовке каждого пакета приоритет обслуживания и в дальнейшем обрабатывать помеченные пакеты независимо друг от друга.
- RIO (RED In and Out) [49] предполагает существование двух типов пакетов – помеченные, для потока, параметры которого превышают установленные пороговые значения, и непомеченные.
- FRED (Flow RED) [50] разработан с учетом устойчивости разных типов трафика к задержкам и потреблению ими ресурсов сети для лучшего соблюдения принципа справедливого распределения ресурсов.
- SRED (Stabilized RED) [51] представляет собой модифицированную версию FRED алгоритма.
- RED-PD (RED with Preferential Dropping) производит активное управление очередями, создающими повышенную нагрузку.

Алгоритмы управления профилем нагрузки. Ранее были рассмотрены алгоритмы обеспечения QoS, согласно которым поступающие в маршрутизатор пакеты формируются в очереди, а также методики управления очередями в буфере коммутирующего устройства. Далее

рассмотрим методики управления входящим потоком пакетов. Исходя из того, что сетевой трафик обладает пульсирующим характером, то есть за периодами относительно низкой активности следует всплеск нагрузки, что особенно характерно для самоподобного трафика, коммутирующее устройство должно брать на себя реализации алгоритмов ограничения трафика (traffic policing) и сглаживания (traffic shaping). Для ограничения трафика и сглаживания применяется алгоритм «корзина маркеров» [39].

Traffic policing заключается в ограничении трафика от источника. То есть выставляется некоторое пороговое значение скорости передачи, превышение которого приводит к отбрасыванию либо маркированию пакетов. Алгоритм представлен на рисунке 4 и состоит из следующих этапов:

- Вычисляется размер всплеска (Committed Burst Size) – допустимый для передачи объем данных. Этот параметр определяет размер корзины маркеров. В случае если в алгоритме используется несколько корзин, то CBS задает размер основной.

- Для соединения вычисляется средняя скорость передачи пакетов (Committed Information Rate), которая определяет скорость поступления маркеров в корзину. Если корзина заполнена, то новые поступающие маркеры отбрасываются, таким образом, размер корзины остается постоянным.

- При поступлении пакета из корзины вынимается число маркеров равное размеру передаваемого сообщения, а сообщение передается далее. Если размер пакета превышает размер корзины, то он, либо отбрасывается либо помечается.

- Если при поступлении нового сообщения корзина маркеров пуста, то сообщение отбрасывается.

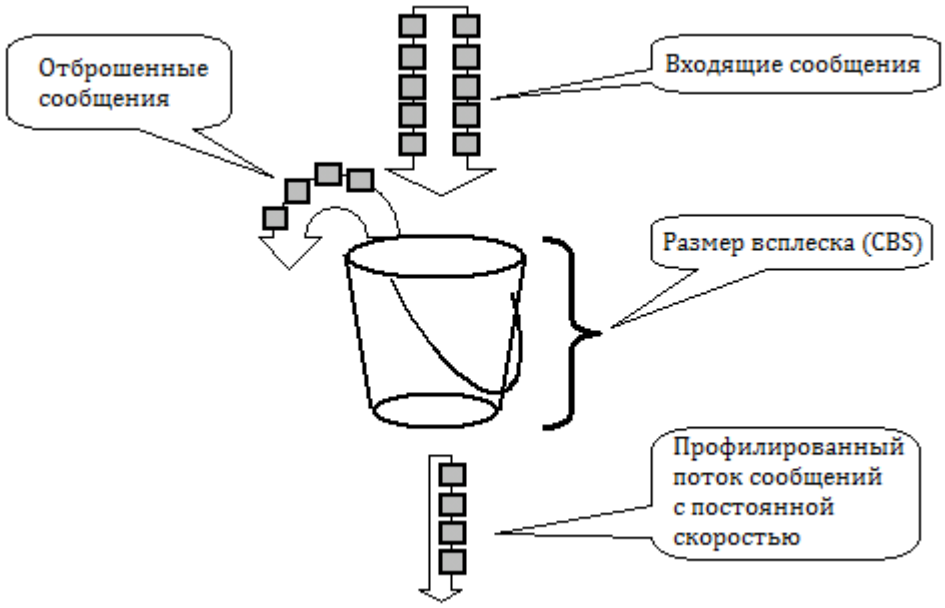


Рисунок 4. Классификация методов управления перегрузками

Модифицированный алгоритм маркерной корзины допускает три корзины и в зависимости от их опустошения, трафик маркируется, что в дальнейшем позволяет обрабатывать пакеты с приоритетом либо отбрасывать [52].

Traffic shaping (сглаживание) используется для сглаживания трафика, чаще исходящего. В случае превышения источником скорости передачи, поступающие пакеты не отбрасываются, а помещаются в очередь. Алгоритм представлен состоит из следующих этапов:

- На основе допустимой пропускной способности канала рассчитывается размер корзины маркеров и скорость поступления маркеров в корзину.
- При поступлении пакета из корзины берется количество маркеров равное длине сообщения и передается далее.
- При нехватке маркеров сообщение помещается в буфер для последующей передачи.

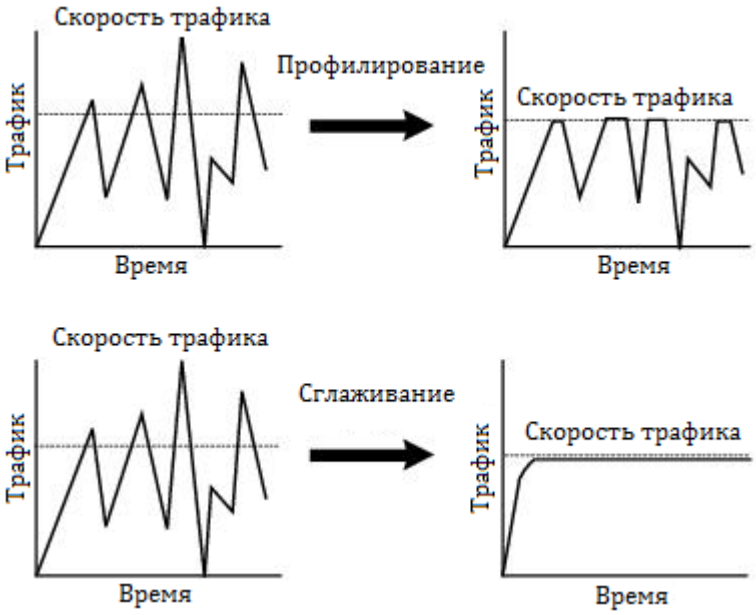


Рисунок 5. Сглаживание и профилирование трафика

Следует отметить, что сглаживание трафика накладывает задержки в передаче пакетов из-за буферизации. Как отмечалось ранее, различные типы трафика в разной степени чувствительны к задержкам и потерям пакетов. Использование дополнительного маркирования пакетов в traffic shaping позволяет задавать приоритет обслуживания для трафика, особенно чувствительного к задержкам.

Рассматривая методы борьбы с перегрузками и управления трафиком без обратной связи, фактически, были рассмотрены методики обеспечения QoS на уровне коммутирующего устройства, маршрутизатора или коммутатора. Таким образом, подобное сетевое устройство выполняет ряд функций в следующей последовательности:

- Входящий поток сообщений сглаживается с использованием алгоритмов формирования трафика, также на первом этапе возможна классификация трафика с помощью алгоритма маркерного ведра.

- Поступивший сглаженный поток пакетов подвергается воздействию алгоритмов управления очередями, определяется алгоритм помещения пакетов в буфер коммутатора.
- Согласно алгоритму обслуживания пакетов в очередях, сообщение из буфера поступает на выход коммутатора.
- Исходящий трафик подвергается сглаживанию в соответствии с заданными характеристиками исходящего канала.

1.2.2 Методы с обратной связью

Рассмотренные ранее алгоритмы управления трафиком и борьбы с перегрузками без обратной связи направлены скорее на предотвращение перегрузок. Обнаружение и управление перегрузками осуществляется алгоритмами с обратной связью, учитывающими, насколько это возможно, текущее состояние сети. Суть механизма обратной связи в данном случае заключается в оповещении перегруженным узлом других узлов сети, через которые следуют пакеты о необходимости временного ограничения скорости передачи данных для снижения нагрузки.

Согласно рисунку 1 обратная связь может быть явной и неявной. Явная ОС или явная сигнализация о перегрузке заключается в оповещении узлом о наличии перегрузки средствами протокола передачи данных. То есть либо отправкой сообщения о степени нагрузки либо добавлением в заголовок передаваемых пакетов бита, сигнализирующего о перегрузке.

Оповещение о растущей нагрузке может быть отправлено по направлению к источнику сообщения:

- Метод противодавления (backpressure) заключается в отправке нагруженным узлом специального сообщения по направлению к источнику нагрузки. Каждый узел сети, передающий далее такое

сообщение, ограничивает входящий поток, что приводит к снижению нагрузки.

- Загруженный узел может отправить источнику нагрузки сообщения (choke packet) о необходимости ограничения скорости передачи. Сообщение может быть сгенерировано не только коммутатором в ответ на каждый отвергнутый пакет из-за переполненного буфера, но и приемником помимо коммутатора. Методика реализуется в протоколе Internet Control Message Protocol (ICMP) сообщением Source Quench [53].

Оповещение о растущей нагрузке может быть отправлено по направлению к получателю сообщения, чтобы задействовать методики управления перегрузками протоколами более высокого уровня. Другими словами, управление или ограничение исходящей нагрузкой выполняется получателем.

Помимо сообщений с признаками перегрузки протокол передачи может задавать допустимую скорость передачи для узлов сети. В таком случае важно указать время отправки регулирующего сообщения иначе могут возникать незатухающие колебания нагрузки разных участков сети.

Также регулирующее сообщение может содержать максимально допустимый размер окна передачи, что составляет метод скользящего окна. В этом случае источник может передавать данные с любой скоростью, объем которых не превышает заданный кредит. Окно также называют кредитом. При повышении нагрузки окно передачи постепенно уменьшается, что позволяет избежать перегрузок.

Наконец, ОС может быть неявной, при которой принимающий узел по косвенным признакам принимает решение о степени загруженности сети. Такими признаками могут быть увеличение RTT (Round Trip Time – время между отправкой сообщения и приемом ответного), отсутствие подтверждения о приеме, то есть потеря пакетов, а так же дублирование

пакетов, что говорит о задержке, приведшей к повторной отправке пакета источником.

1.2.3 Управление трафиком и обеспечение QoS в TCP

Transmission Control Protocol (TCP – протокол управления передачей) первоначально определен в [54], разработан для надежной передачи данных через ненадежные каналы связи. Информация при передаче разбивается на пакеты, которые в свою очередь подвергаются сквозной нумерации, таким образом, получатель может контролировать получение пакетов и составлять исходное сообщение. Каждый передаваемый сегмент данных включает служебную информацию: SN – порядковый номер сегмента, AN – номер подтверждения, W – размер окна. После получения определенного объема данных получатель подтверждает принятые сообщения, в противном случае отправитель выполняет повторную передачу пакетов.

Задача управления перегрузками в рамках протокола TCP достаточно сложна. Это связано с тем, что TCP позволяет управлять потоком только на уровне получателя и отправителя, с помощью изменения окна передачи. Догадаться о перегрузках в сети в таком случае можно лишь по косвенным признакам: увеличении RTT и потерь. Также отдельные TCP потоки не обмениваются между собой информацией о требуемых ресурсах, скорости передачи и т.д.

Наиболее известные реализации протокола TCP, такие как Reno и Tahoe реализуют несколько общих стратегий по недопущению перегрузок в сети. Первая называется «медленным пуском» и состоит в следующем. На этапе установления соединения отправителю и получателю необходимо выбрать подходящий размер окна передачи. Для получателя отправной точкой в расчете размера окна становится размер буфера, то есть размер окна передачи не должен превышать размера буфера, чтобы не вызывать переполнения. Отправитель, опираясь на полученный размер буфера, должен рассчитать так называемое окно перегрузки. Для этого при каждой

последующей отправке сообщения размер окна удваивается до тех пор, пока он не достигнет размера буфера отправителя, либо пока один из пакетов не будет доставлен вовремя. Такой подход называется медленным или затяжным пуском, позволяющий достаточно быстро подобрать допустимую пропускную способность сети. В случае получения источником Source Quench сообщения, рассмотренного ранее, такая ситуация обрабатывается также, как и потеря пакета. При этом размер окна отправителя устанавливается равным максимальному сегменту передачи.

Другой общий для разных реализаций протокола TCP стратегией является учет изменчивости RTT при расчете времени повторной передачи:

$$T(i+1) = \mathrm{RTT}'(i+1) + k\Theta(i+1), \tag{31}$$

где $T(i+1)$ – таймаут повторной передачи, RTT’ – экспоненциально сглаженное значения RTT, Θ – среднее линейное отклонение RTT, а k - некий коэффициент, который согласно [55] равен 4 в большинстве современных реализаций протокола. Алгоритм Карна [56] и метод экспоненциального отката также управляют временем таймаута передачи.

Алгоритмы затяжного пуска, динамического изменения размера окна при перегрузке, быстрого восстановления и ограниченной передачи оперируют в первую очередь размером окна передачи для повышения полезной пропускной способности сети.

Выводы

1) Современные компьютерные сети отличаются от телефонных сетей, аналогии с которыми проводились достаточно долго. Вслед за изменением сетевой нагрузки, возникновению новых протоколов передачи данных и изменением характера самих данных претерпевали изменения и модели сетевого трафика, используемые в работах.

2) Ряд современных работ по изучению трафика посвящен исследованию самоподобных, фрактальных и хаотических свойств потока сообщений в современной компьютерной сети.

3) Большинство рассмотренных методик обеспечения QoS в сетях используют линейные алгоритмы, решая задачу управления трафиком, обнаружения и борьбы с перегрузками на основе только статистических свойств потока данных.

4) Является актуальной разработка методов управления трафиком и борьбы с перегрузками на основе нелинейно-динамических свойств потока данных и с учетом методов нелинейной динамики и теории хаоса.

На основе адекватной модели трафика возможно реализовать методику борьбы с перегрузками с учетом краткосрочного прогнозирования нагрузки в совокупности с рассмотренными алгоритмами формирования трафика или модификации RED, либо выполнять контроль источника сообщений на основе прогнозируемого состояния сети, задействовав широкий спектр методов обработки временных рядов.

2 Сбор и анализ экспериментальных данных

2.1 Описание экспериментальной среды

В работе производился мониторинг сервера корпоративной сети МГТУ им. Н.Э. Баумана [57]. Физическая машина поделена на несколько виртуальных с ОС Linux, каждая из которых используется под ряд задач, таких как СУБД, web-серверы, файл-серверы и т.д. Большая часть трафика передается по HTTP, интерфейс 100Мбит/с Ethernet. Параметры потоков исследуемого сервера представлены в таблице 1.

Таблица 1. – Параметры входных и выходных потоков информации сервера ЛВС

Сервер		λ, кадр/с	$L_{ср}$, байт
ns.bmstu.ru	in	260	104
	out	260	186
ftp.bmstu.ru	in	1527	105
	out	2070	1288
iptv.bmstu.ru	in	2	63
	out	306	1192
www.bmstu.ru	in	82	129
	out	98	1416
e-u.bmstu.ru	in	112	138
	out	129	1432
db.bmstu.ru	in	15	115
	out	18	619

В работе [58] указано следующее краткое описание серверов, подключенных к коммутаторам ядра университета:

- ns.bmstu.ru – сервер обеспечивающий выполнение службы доменных имен (DNS), транспортный протокол UDP;
- ftp.bmstu.ru – файловый сервер, обеспечивающий хранение файлов различного назначения, транспортный протокол TCP, протокол прикладного уровня FTP;
- iptv.bmstu.ru – сервер трансляции потокового видео, транспортный протокол UDP, протокол прикладного уровня RTP;

- www.bmstu.ru – web-сервер МГТУ им. Н. Э. Баумана, транспортный протокол TCP, протокол прикладного уровня HTTP;
- e-u.bmstu.ru – сервер портала информационной системы «Электронный университет» МГТУ им. Н. Э. Баумана, транспортный протокол TCP, протокол прикладного уровня HTTP;
- db.bmstu.ru – сервер баз данных, транспортный протокол TCP, протокол прикладного уровня SQL. [58]

Сеть университета построена с использованием технологии Ethernet на базе оборудования Cisco по топологии «звезда» с ядром в центре, от которого расходятся магистральные сегменты транспортной подсистемы до коммутаторов распределительного уровня, к портам которых подключаются непосредственно либо оконечные хосты (пользователи), либо ЛВС подразделений и кафедр.

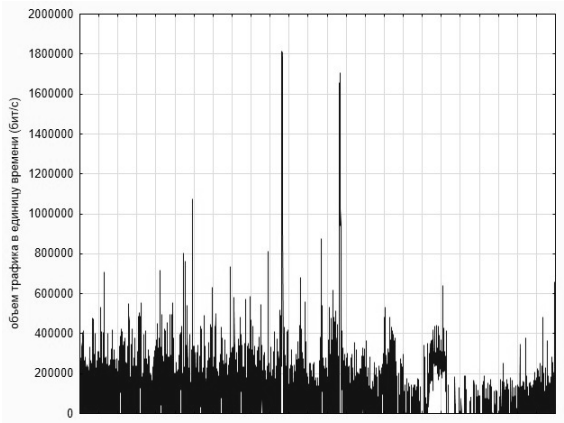
Общее число компьютеров ~7000, серверов ~150. Коммутаторов распределительного уровня ~115, коммутаторы Cisco серий 2950, 2960, 3560, 3750. Локальные сети подразделений Университета, либо сами пользователи подключаются к портам наиболее близко расположенных к ним коммутаторов, как правило, в пределах одного этажа или даже крыла и используют динамическую конфигурацию систем.

Каждый физический и виртуальный сервер сети университета использует для мониторинга Zabbix [59]. Zabbix – это клиент-серверное приложение, используемое для сбора, хранения и обработки информации о состоянии сети, сетевой нагрузке, а также состоянии операционной системы сервера в реальном времени. Приложение широко используется администраторами сети для мониторинга и своевременного оповещения о сбоях, перегрузках и отказах оборудования. Для дальнейшей обработки проводилось накопление данных следующих параметров:

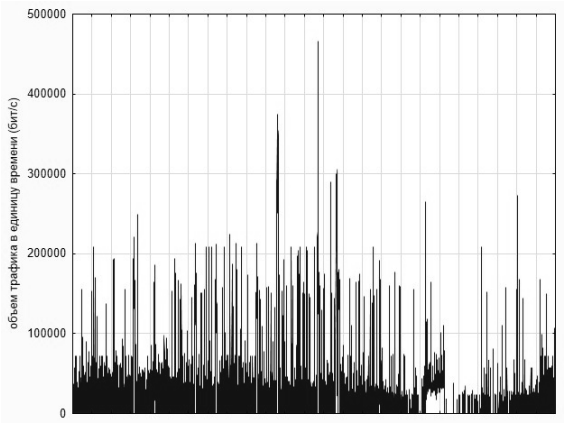
- объем кэшированной памяти;

- объем буферизированной памяти;
- время простоя процессора при операциях ввода/вывода;
- время процессора в режиме ожидания;
- время обработки процессором пользовательских процессов;
- время обработки процессором системных процессов;
- объем свободной памяти;
- входящий/исходящий трафик (бит/с);
- число процессов ОС;
- число процессов web-сервера Apache;
- суммарная загрузка процессора.

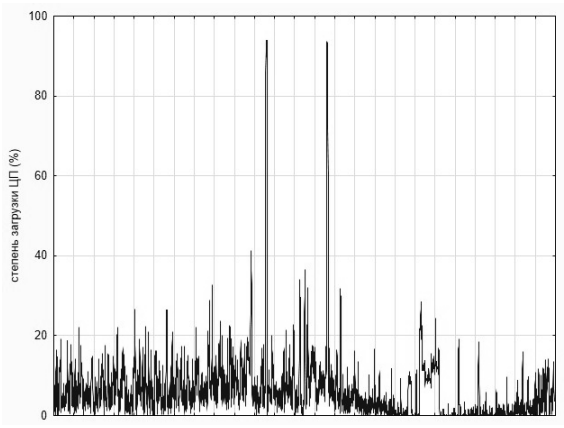
Данные снимались с различными временными промежутками для различных параметров, от 1 до 60 секунд в течение суток. Временная зависимость объема трафика в единицу времени (бит/с) приведена на рисунках 6(а, б) для входящего и исходящего трафика. На рисунках 6(в - е) показаны степень загрузки ЦПУ (в %), объем свободной памяти (бит), число процессов и объем кэшированной памяти.



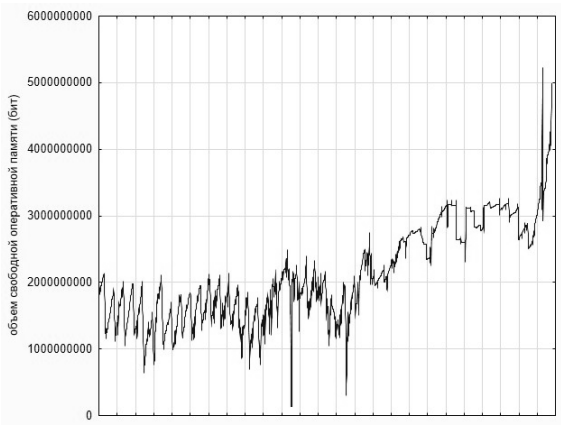
а



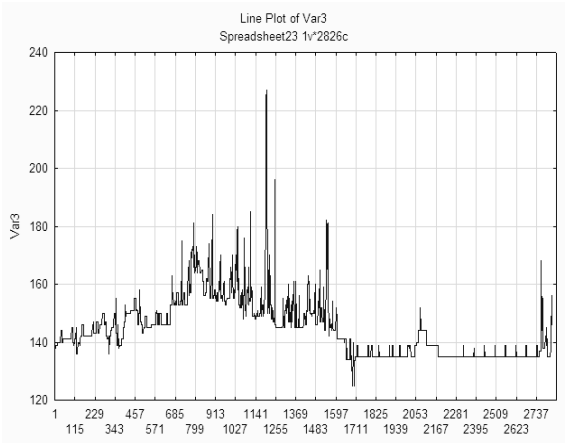
б



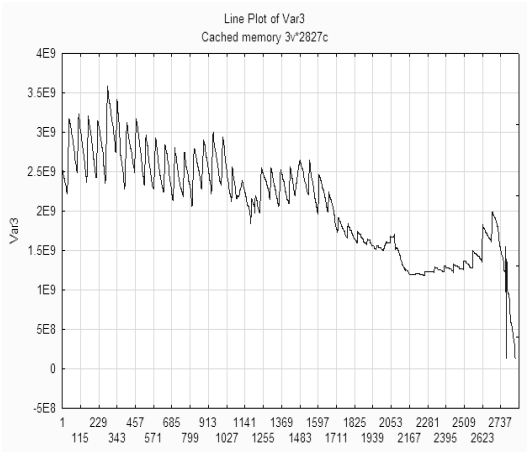
в



г



д



е

Рисунок 6. Временная зависимость объема трафика в единицу времени: (*а*) – входящий дневной трафик, (*б*) – исходящий дневной трафик, (в) - загрузка ЦПУ, (г) - объем свободной памяти, (д) - число процессов, (е) - Объем кэшированной памяти

Были накоплены данные широкого спектра параметров, характеризующих работу нагруженного сервера корпоративной сети. Для решения задачи борьбы с перегрузками, в частности, исследования характера реального потока сообщений, были задействованы только данные входящего и исходящего трафика. Данные по аппаратным характеристикам сервера использованы в решении задачи влияния кумулятивного сетевого трафика на ресурсы хоста.

2.2 Статистический анализ данных

Для выявления характерных особенностей исследуемых процессов необходимо провести детальный анализ собранных данных средствами корреляционного и регрессионного анализа для поиска корреляций, нахождения закона распределения рядов. Оценить плотность распределения и спектральную плотность процессов, определить скорость убывания зависимости значений ряда.

Некоторые не слишком популярные методы анализа данных, такие как расчет параметра Херста отличными от R/S методами, оценка степени хаотичности процессов, в частности показателя Ляпунова и энтропии, потребовали создания оригинального программного обеспечения. При этом большинство озвученных ранее задач по анализу данных выполнимы существующими программными пакетами.

Весьма популярный программный пакет Statistica [60] позволяет выполнять большинство известных алгоритмов статистического анализа данных, имеет удобный интерфейс с широкими возможностями по работе с графикой. Возможен расчет таких общих статистических показателей, как медиана, мода, среднее и стандартное отклонение, доверительные интервалы для среднего, асимметрия, эксцесс, гармоническое и геометрическое среднее. Доступно множество графиков, таких как гистограммы распределений, диаграммы рассеяния, вероятностные графики, и средств работы с графикой. Также в ПО доступен корреляционный, автокорреляционный и регрессионный анализ данных, подгонка распределения ряда. Отдельным блоком реализована возможность краткосрочного прогнозирования данных на основе популярных моделей AR, ARIMA. При такой обширной функциональности неизбежно возрастает сложность первичного вхождения в работу с ПО. Также в пакете нет встроенной среды для реализации пользовательских функций.

Оценка нелинейно – динамических свойств данных проводилась средствами нескольких программных пакетов. Расчет показателя Херста методом R/S – анализа и корреляционной размерности данных просто и наглядно можно выполнить с помощью ПО Fractan [61], также доступно построение графиков автокорреляции и 2D-3D фазовых пространств. Фазовые портреты данных и выделение 3D аттракторов возможно в программном обеспечении для нелинейного динамического анализа, разработанного в рамках работы [62]. Приложение также позволяет выполнять спектральный и вейвлет анализ данных с построением диаграмм.

Помимо представленных выше программных средств, работа с данными велась средствами ПО Matlab и Simulink [63]. Благодаря встроенной среде разработки пользовательских функций и средств автоматизации вычислений, большинство расчетов проводилось именно в Matlab, а моделирование сети коммутации пакетов в Simulink.

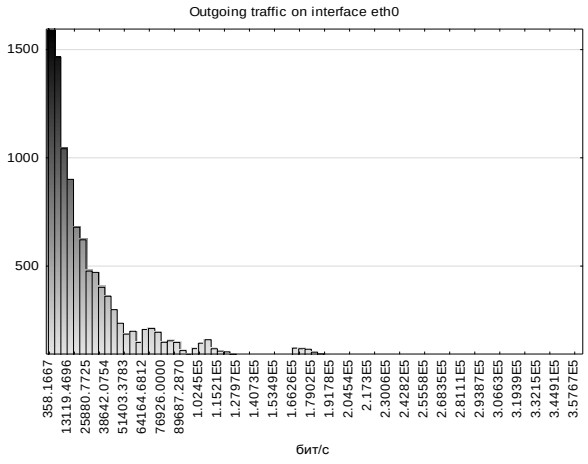
Анализ функции плотности распределения. Плотность распределения вероятностей случайной величины X – это первая производная от интегральной функции распределения вероятностей $F(x)$*:*

$$f(x) = \lim_{\Delta x \to 0} \frac{F(x+\Delta x) - F(x)}{\Delta x} = F'(x) = \frac{dF(x)}{dx}, \qquad (32)$$

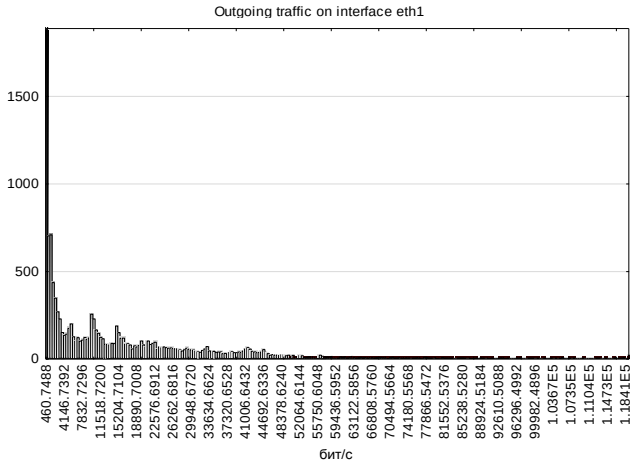
т.к. вероятность того, что случайная величина X содержится в промежутке $[x,x+\Delta x]$*:*

$$P(x < X < x + \Delta x) = F(x+\Delta x) - F(x) \approx dF(x) = f(x)dx\,. \qquad (33)$$

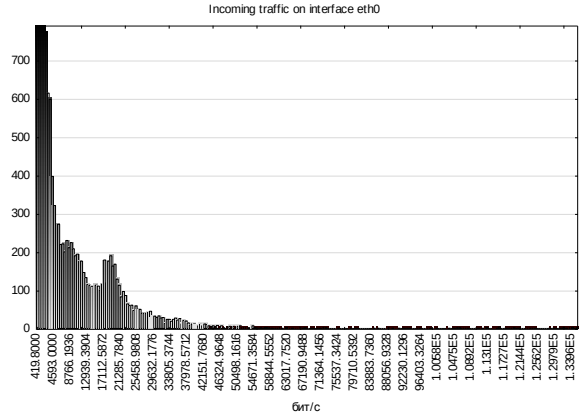
Графически плотность распределения ряда можно оценить по гистограмме частот появления участков данных [64].


Outgoing traffic on interface eth0
1500
1000
500
358.1667
13119.4696
25880.7725
38642.0754
51403.3783
64164.6812
76926.0000
89687.2870
1.0245E5
1.1521E5
1.2797E5
1.4073E5
1.5349E5
1.6626E5
1.7902E5
1.9178E5
2.0454E5
2.173E5
2.3006E5
2.4282E5
2.5558E5
2.6835E5
2.8111E5
2.9387E5
3.0663E5
3.1939E5
3.3215E5
3.4491E5
3.5767E5
бит/с

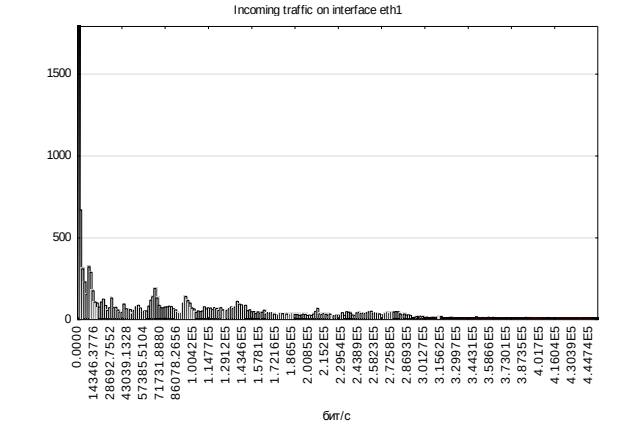
а


Outgoing traffic on interface eth1
1500
1000
500
0
460.7488
4146.7392
7832.7296
11518.7200
15204.7104
18890.7008
22576.6912
26262.6816
29948.6720
33634.6624
37320.6528
41006.6432
44692.6336
48378.6240
52064.6144
55750.6048
59436.5952
63122.5856
66808.5760
70494.5664
74180.5568
77866.5472
81552.5376
85238.5280
88924.5184
92610.5088
96296.4992
99982.4896
1.0367E5
1.0735E5
1.1104E5
1.1473E5
1.1841E5
бит/с

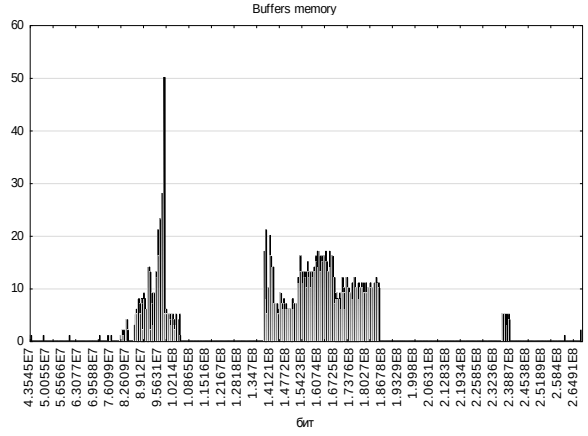
б


Incoming traffic on interface eth0
700
600
500
400
300
200
100
0
419.8000
4593.0000
8766.1936
12939.3904
17112.5872
21285.7840
25458.9808
29632.1776
33805.3744
37978.5712
42151.7680
46324.9648
50498.1616
54671.3584
58844.5552
63017.7520
67190.9488
71364.1456
75537.3424
79710.5392
83883.7360
88056.9328
92230.1296
96403.3264
1.0058E5
1.0475E5
1.0892E5
1.131E5
1.1727E5
1.2144E5
1.2562E5
1.2979E5
1.3396E5
бит/с

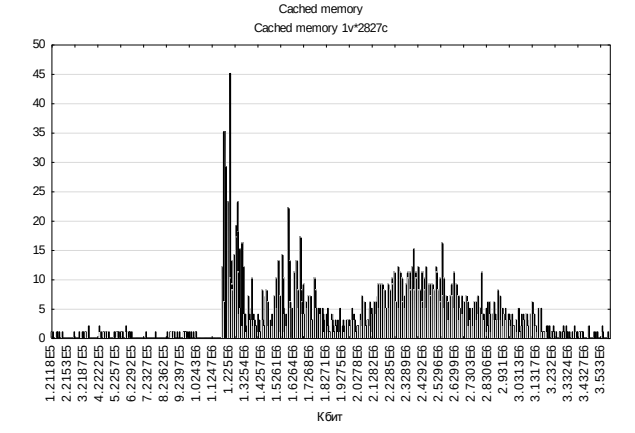
в


Incoming traffic on interface eth1
1500
1000
500
0
0.0000
14346.3776
28692.7552
43039.1328
57385.5104
71731.8880
86078.2656
1.0042E5
1.1477E5
1.2912E5
1.4346E5
1.5781E5
1.7216E5
1.865E5
2.0085E5
2.152E5
2.2954E5
2.4389E5
2.5823E5
2.7258E5
2.8693E5
3.0127E5
3.1562E5
3.2997E5
3.4431E5
3.5866E5
3.7301E5
3.8735E5
4.017E5
4.1604E5
4.3039E5
4.4474E5
бит/с

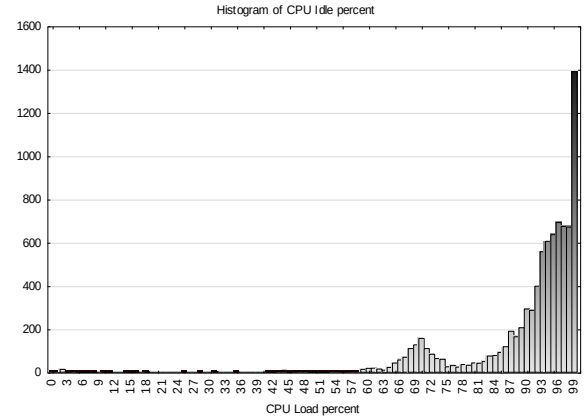
г

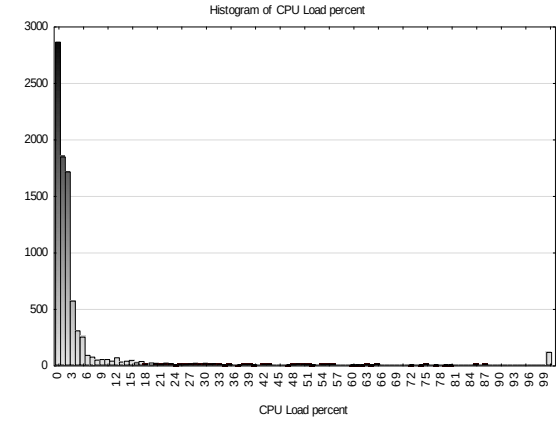

Buffers memory
60
50
40
30
20
10
0
4.3545E7
5.0055E7
5.6566E7
6.3077E7
6.9588E7
7.6099E7
8.2609E7
8.912E7
9.5631E7
1.0214E8
1.0865E8
1.1516E8
1.2167E8
1.2818E8
1.347E8
1.4121E8
1.4772E8
1.5423E8
1.6074E8
1.6725E8
1.7376E8
1.8027E8
1.8678E8
1.9329E8
1.998E8
2.0631E8
2.1283E8
2.1934E8
2.2585E8
2.3236E8
2.3887E8
2.4538E8
2.5189E8
2.584E8
2.6491E8
бит

д

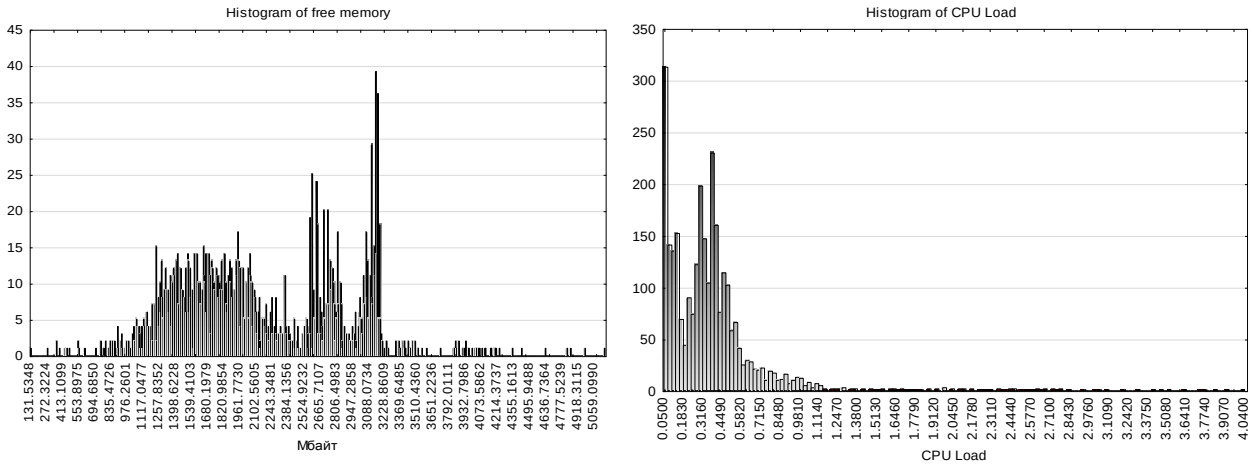

Cached memory
Cached memory 1v*2827c
50
45
40
35
30
25
20
15
10
5
0
1.2118E5
2.2153E5
3.2187E5
4.2222E5
5.2257E5
6.2292E5
7.2327E5
8.2362E5
9.2397E5
1.0243E6
1.1247E6
1.225E6
1.3254E6
1.4257E6
1.5261E6
1.6264E6
1.7268E6
1.8271E6
1.9275E6
2.0278E6
2.1282E6
2.2285E6
2.3289E6
2.4292E6
2.5296E6
2.6299E6
2.7303E6
2.8306E6
2.931E6
3.0313E6
3.1317E6
3.232E6
3.3324E6
3.4327E6
3.533E6
Кбит

е


Histogram of CPU Idle percent
1600
1400
1200
1000
800
600
400
200
0
0
3
6
9
12
15
18
21
24
27
30
33
36
39
42
45
48
51
54
57
60
63
66
69
72
75
78
81
84
87
90
93
96
99
CPU Load percent


Histogram of CPU Load percent
3000
2500
2000
1500
1000
500
0
0
3
6
9
12
15
18
21
24
27
30
33
36
39
42
45
48
51
54
57
60
63
66
69
72
75
78
81
84
87
90
93
96
99
CPU Load percent

ж *з*



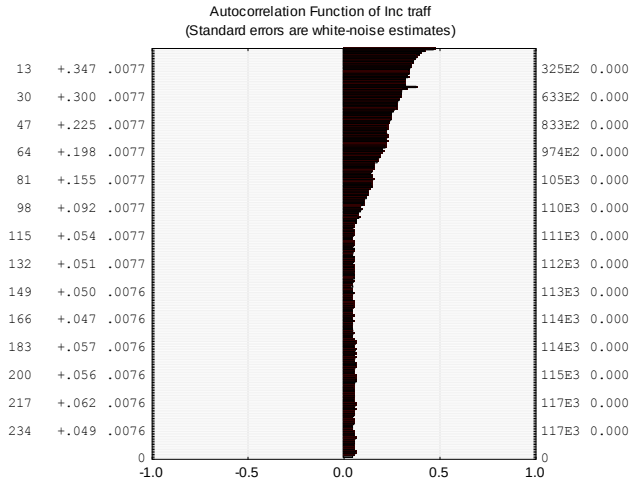
и *к*

Рисунок 7. Гистограммы собранных данных: (*а*) - гистограмма исходящего трафика (интерфейс Eth0), (*б*) - гистограмма исходящего трафика (интерфейс Eth1), (*в*) - гистограмма входящего трафика (интерфейс Eth0), (*г*) - гистограмма входящего трафика (интерфейс Eth0), (*д*) - гистограмма буферизированной памяти, (*е*) - гистограмма кэшированной памяти, (*ж*) - гистограмма времени ЦП в режиме ожидания, (*з*) - гистограмма времени ЦП в режиме ожидания ввода/вывода, (*и*) - гистограмма времени ЦП в режиме ожидания, (*к*) - гистограмма суммарной загрузки ЦП

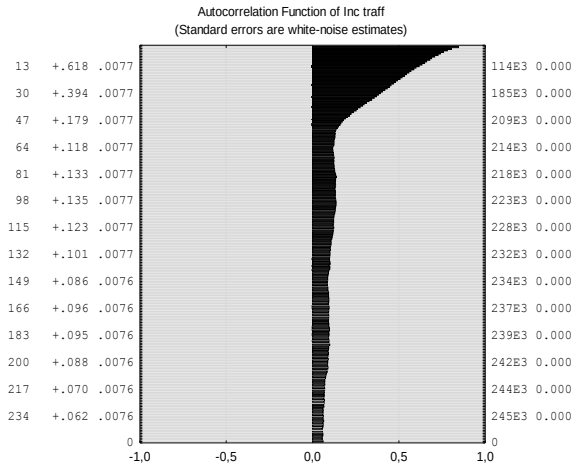
На основе гистограмм видно, что плотность распределения вероятности процессов передачи входящего и исходящего трафика имеет степенной характер и, соответственно подчиняются некоторому классу распределений с тяжелым хвостом [65] (Стьюдента или Парето). Что касается аппаратных ресурсов сервера, то можно сказать, что некоторая часть ряда может описываться степенной функцией плотности распределения вероятности, а часть ряда отвечает нормальному распределению.

Анализ функции автокорреляции. Рассмотрим временной ряд, состоящий из равноудаленных по времени значений (x_i) $i=1,...N$. Нас интересует выделение корреляций между значениями x_i и x_{i+s} для различных значений s. Как правило, в первую очередь из каждого значения вычитается среднее $\langle x \rangle = \frac{1}{N}\sum_{i=1}^{N} x_i$. Численно, величина корреляции между значениями $\tilde{x}$, отделенными на s шагов, определяется с помощью функции автоковариации $C(s) = \langle \tilde{x}_i, \tilde{x}_{i+s} \rangle$ или автокорреляции $C(s) / \langle \tilde{x}_i^2 \rangle$.

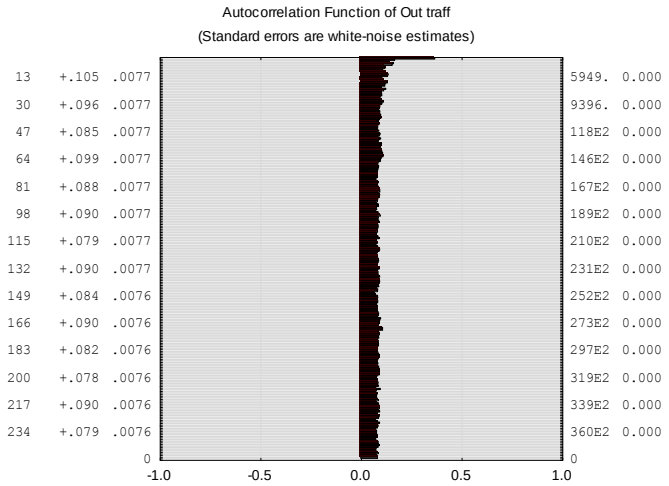
Если величина $C(s)$ убывает экспоненциально, то (x_i) обладает краткосрочной зависимостью, $C(s) \sim \exp(-s/t_x)$, а если $C(s)$ убывает по степенному закону $C(s) \propto s^{-\gamma}$ с показателем корреляции $0<\gamma<1$, то процесс обладает долгосрочной корреляцией. Нестационарность временного ряда в данном случае затрудняет поиск среднего значения.



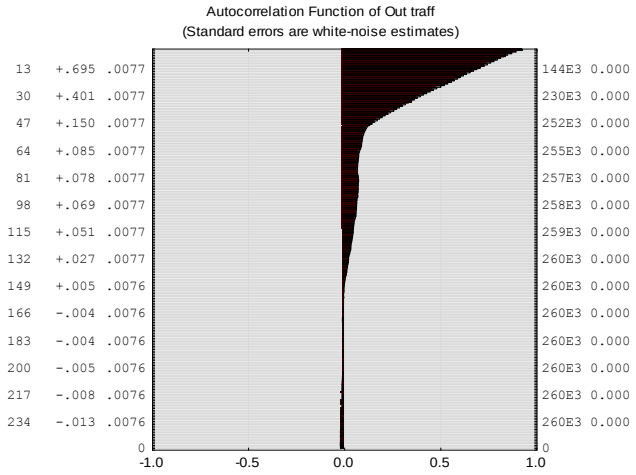
а



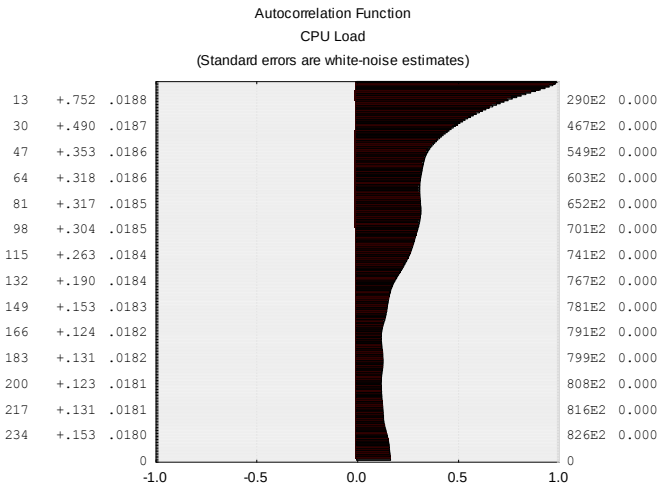
б



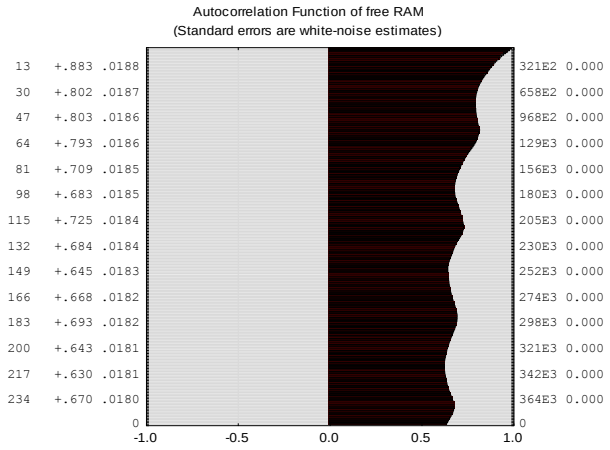
в



г



д



е

Рисунок 8. Графики АКФ собранных данных: (а) - АКФ входящего трафика на интерфейсе Eth0, (б) - АКФ входящего трафика на интерфейсе Eth1, (в) - АКФ исходящего трафика на интерфейсе Eth0, (г) - АКФ исходящего трафика на интерфейсе Eth1, (д) - АКФ суммарной загрузки ЦП, (е) - АКФ свободной оперативной памяти

Из графиков АКФ можно заключить, что процесс передачи трафика обладает медленно убывающей зависимостью. Также стоит отметить, что АКФ для процесса выделения памяти обладает ярко выраженной периодичностью. Очевидно, что в данном случае процесс в первую очередь характеризуется логикой работы операционной системы, что вносит сильную периодичную составляющую.

В целом можно заключить, что исследование АКФ сетевого трафика, а также большинства аппаратных характеристик, позволяет в дальнейшем работать с процессами с учетом убывающей зависимости.

Спектральный анализ. Спектральный анализ – это мощный инструмент обработки данных, в частности сглаживания и фильтрации, имеющий в своей основе различные интегральные преобразования. Спектром ряда $X(i)$ называют некоторую функцию $F(w)$, полученную в соответствии с определенным алгоритмом. Примерами спектров служат Фурье – преобразование, а также вейвлет – преобразование [66].

Преобразование Фурье превращает функцию в совокупность её частотных составляющих, то есть это интегральное преобразование, раскладывающее исходную функцию по базисным синусоидальным функциям:

$$F(w) = \int_{-\infty}^{\infty} y(x)\exp(-iwx)dx\,. \tag{34}$$

Дискретное преобразование Фурье в таком случае:

$$f(t) = \sum_{n=-\infty}^{\infty} C_n \exp[i\beta w_n t], \tag{35}$$

где $w_n = nw_o = n(2\pi/T)$ – круговая частота n-й гармонической составляющей, C_n – комплексная амплитуда n-й гармоники:

$$C_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t)\exp\{-iw_n t\}dt\,. \tag{36}$$

Совокупность C_n и есть спектр функции $f(t)$[66]. На рисунках далее представлены графики спектральной плотности от частоты для основных аппаратных характеристик севера, а так же сетевой нагрузки.

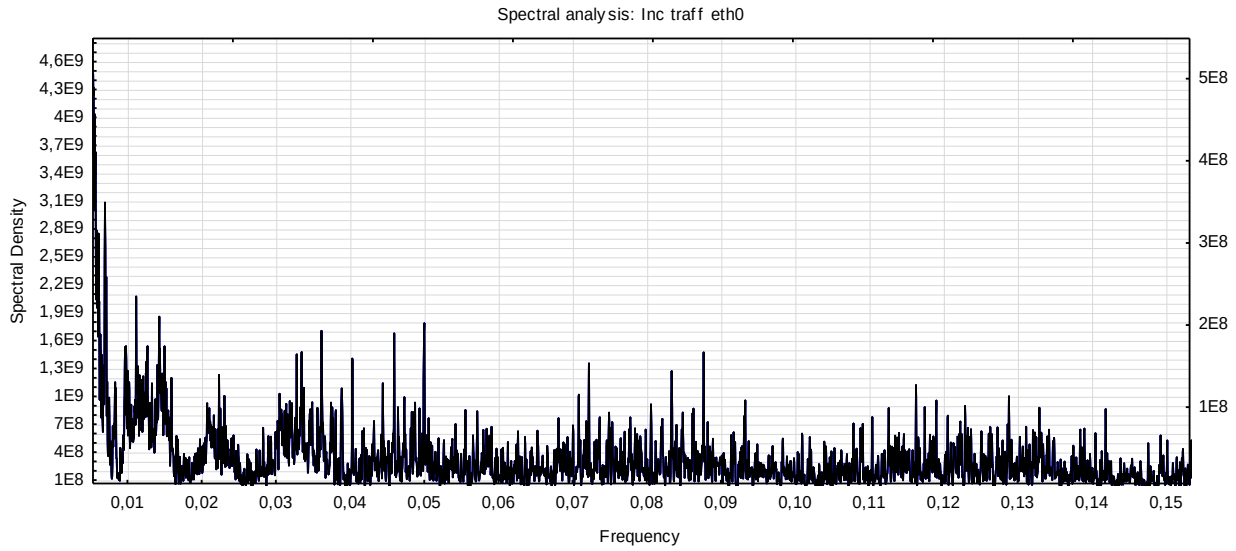


Рисунок 9. Спектр входящего трафика на интерфейсе Eth0

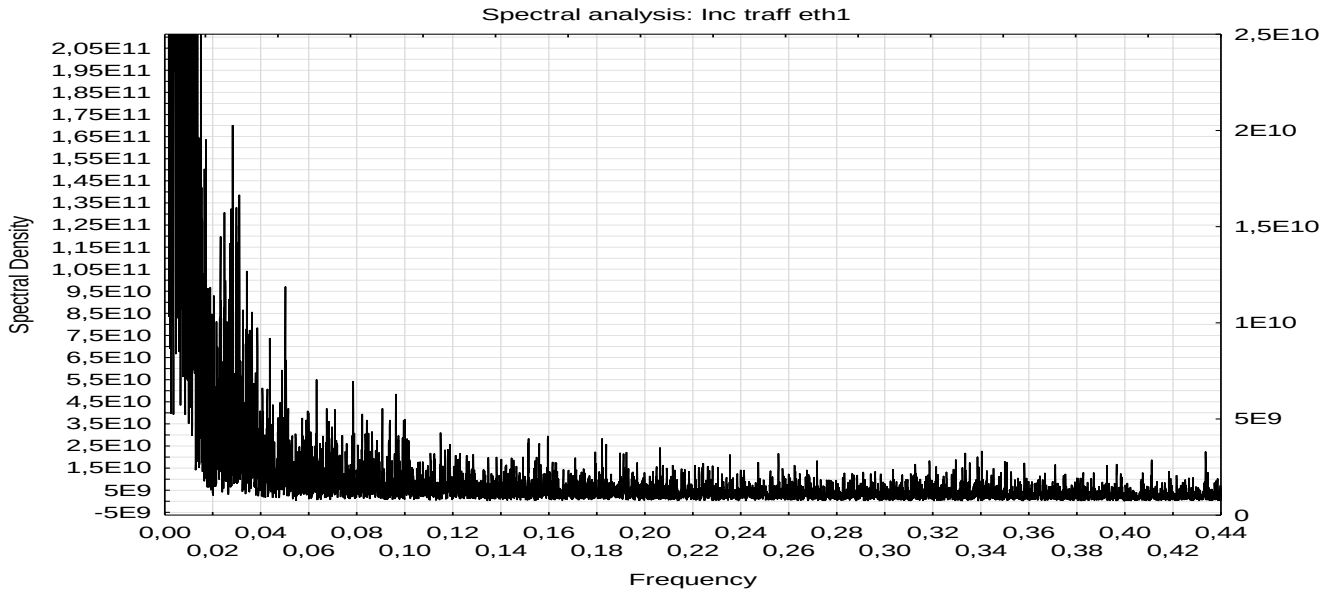


Рисунок 10. Спектр входящего трафика на интерфейсе Eth1

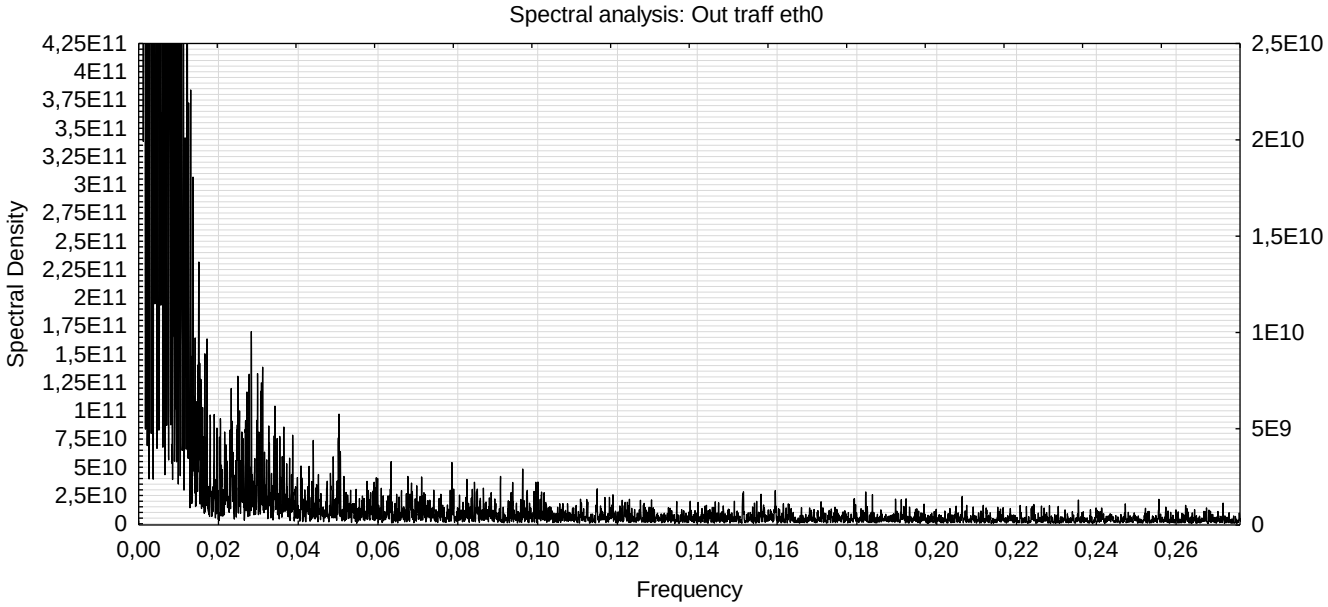


Рисунок 11. Спектр исходящего трафика на интерфейсе Eth0

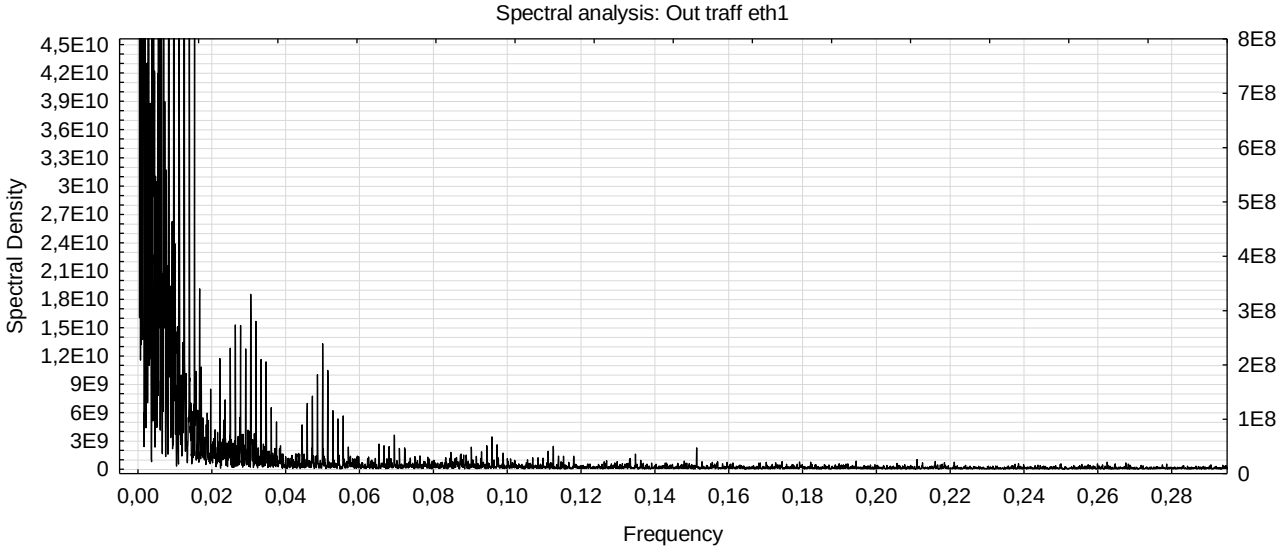


Рисунок 12. Спектр исходящего трафика на интерфейсе Eth1

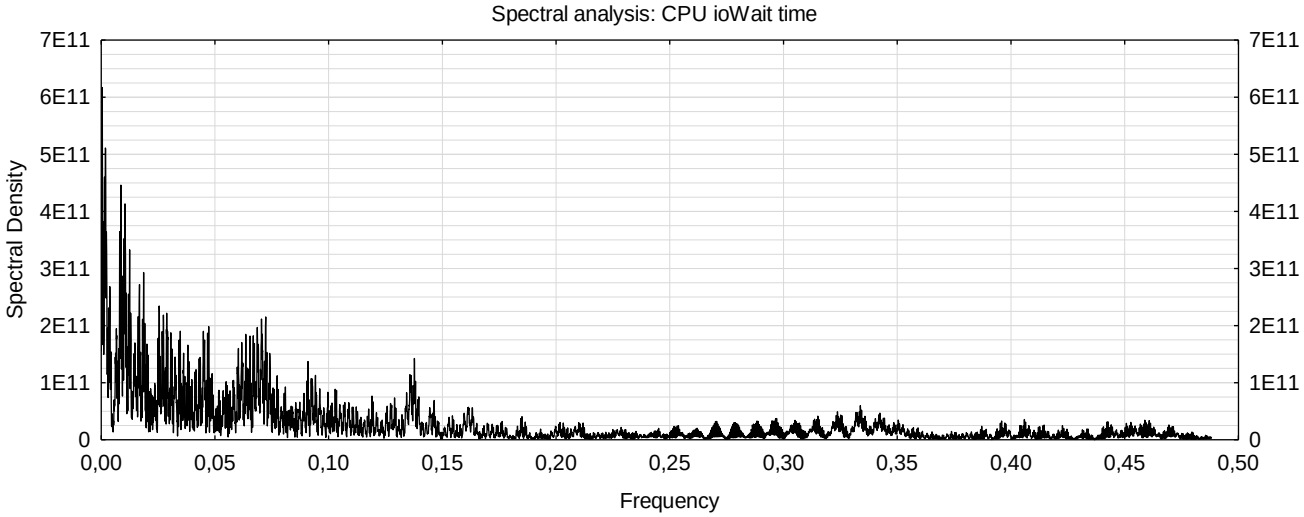


Рисунок 13. Спектр времени пребывания ЦПУ в режиме ожидания ввода/вывода

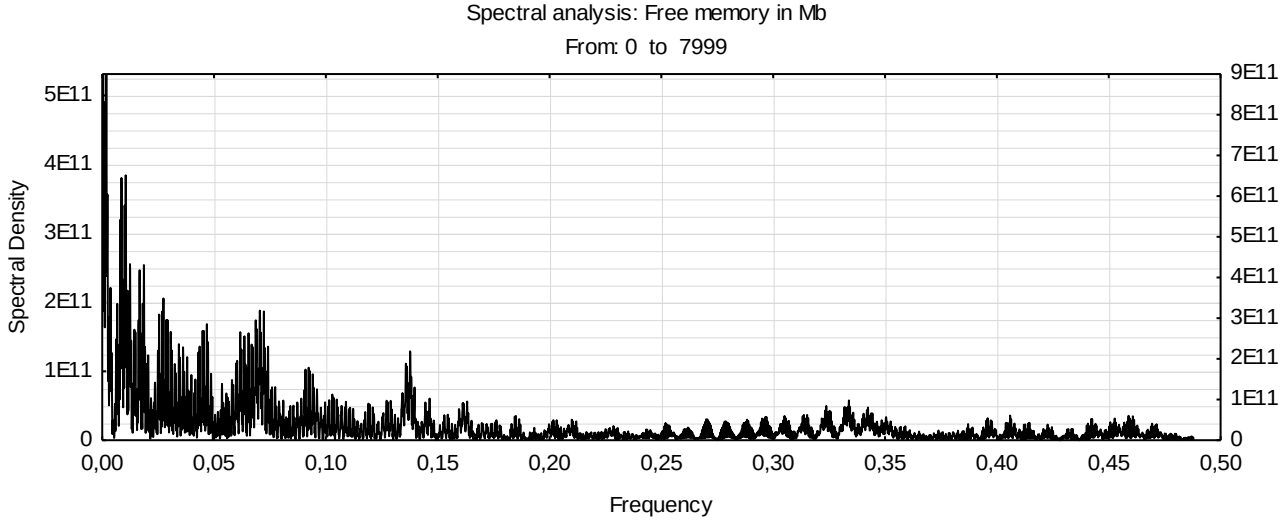


Рисунок 14. Спектр свободной оперативной памяти(Mb)

Согласно [65] для процесса с медленно убывающей зависимостью свойственен степенной характер убывания спектральной плотности, а также стремление к бесконечности при частотах близких к нулю. Подобное поведение наблюдается для всех исследуемых процессов, в большей степени для входящего и исходящего трафика.

Корреляционный и регрессионный анализ. Весьма интересной и малоизученной остается задача кумулятивного влияния сетевой нагрузки на распределение ресурсов сервера. В таблице 2 представлены значения парной корреляции между трафиком на разных интерфейсах сервера и аппаратной нагрузкой, темным цветом выделены наиболее значимые величины.

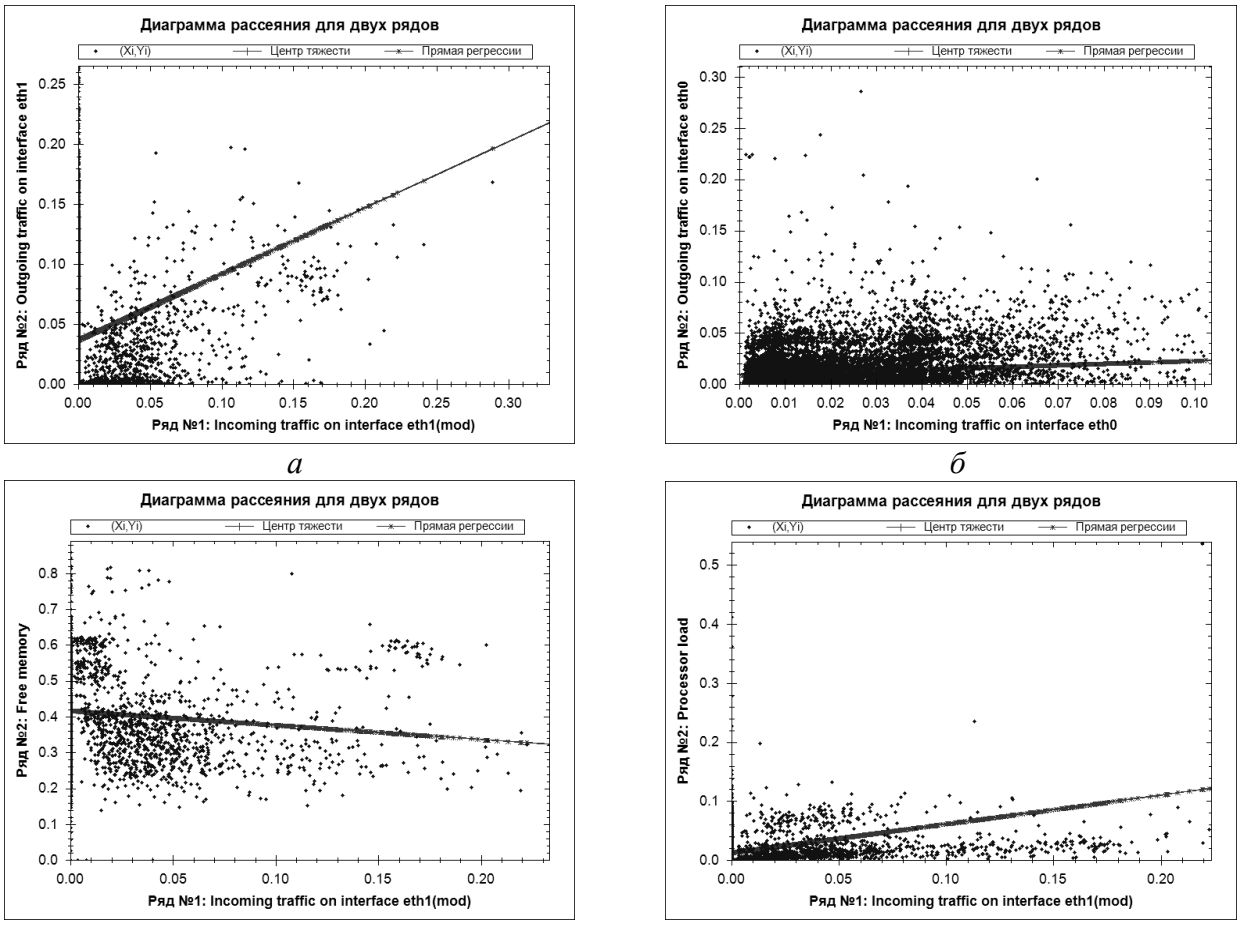
Таблица 2. Коэффициенты корреляции регистрируемых процессов

	Вход. трафик Eth0	**Вход. трафик Eth1**	**Исход. трафик Eth0**	**Исход. трафик Eth1**	**Сумм. загрузка ЦПУ**	**Объем своб. ОЗУ**	**Число польз. проц.**	**Время ожид. ЦПУ**
Вход. трафик Eth0	**1**	**0,02**	**0,23**	**0,01**	**0,17**	**-0,14**	**0,2**	**-0,12**
Вход. трафик Eth1	**0,02**	**1**	**0,05**	**0,2**	**0,49**	**-0,19**	**0,24**	**-0,17**
Исход. трафик Eth0	**0,23**	**0,05**	**1**	**0,19**	**0,12**	**-0,39**	**0,27**	**-0,1**
Исход. Трафик Eth1	**0,01**	**0,2**	**0,19**	**1**	**0,55**	**-0,28**	**0,33**	**-0,34**
Сумм. загрузка ЦПУ	**0,17**	**0,49**	**0,12**	**0,55**	**1**	**-0,24**	**0,39**	**-0,7**
Объем своб. ОЗУ	**-0,14**	**-0,19**	**-0,39**	**-0,28**	**-0,24**	**1**	**-0,33**	**0,35**
Число польз. проц.	**0,2**	**0,24**	**0,27**	**0,33**	**0,39**	**-0,33**	**1**	**-0,04**
Время ожид. ЦПУ	**-0,12**	**-0,17**	**-0,1**	**-0,34**	**-0,7**	**0,35**	**-0,04**	**1**

Система мониторинга Zabbix позволяет задавать разные периоды дискретизации для различных регистрируемых процессов. Например, данные сетевой нагрузки накапливаются и регистрируются каждую секунду, а данные по загрузке центрального процессора регистрируются с промежутком

в 5 секунд. Поэтому, ряды для регрессионного анализа, сначала приводились к одинаковым временным интервалам регистрации значений процессов, а после проводился расчет парной корреляции

В первую очередь стоит отметить, что все значения коэффициентов парной корреляции объяснимы аналитически. Так росту сетевой нагрузки соответствует рост нагрузки на аппаратные ресурсы сервера. В первую очередь это касается суммарной нагрузки на ЦПУ и выделения оперативной памяти. Также исходящий сетевой трафик сильнее входящего коррелирует с числом пользовательских процессов, что также несложно объяснить тем фактом, что источником исходящего трафика могут быть только локальные процессы сервера. Для оценки линейности корреляции были построены диаграммы рассеяния, а также прямые регрессии (Рисунок 15).



а *б* *в* *г*

Рисунок 15. Диаграммы рассеяния собранных данных: (*а*) - диаграмма рассеяния исх.-вх. трафика(Eth1), (*б*) - диаграмма рассеяния исх.-вх. трафик(Eth0), (*в*) - диаграмма рассеяния объем памяти - вх. трафик(Eth1), (*г*) - диаграмма рассеяния загрузка ЦП - вх. трафик(Eth2)

Большинство рассчитанных парных корреляций обладают вытянутой диаграммой рассеяния, что говорит о наличии линейной регрессии. Это подтверждает и прямая регрессии, которая весьма неплохо аппроксимирует диаграммы рассеяния.

2.3 Анализ нелинейно – динамических свойств данных

Применительно к телетрафику методы нелинейной динамики стали использоваться не так давно [67,68]. Среди отечественных исследователей особенно стоит отметить работы Шелухина О.И. Так, в [9] приводятся теоритические аспекты самоподобных случайных процессов, и дается объяснение, почему трафик в современных телекоммуникационных системах следует считать фрактальным, а также рассматриваются математическая и программная реализация самоподобных математических моделей. Приводится анализ самоподобности LAN и WAN трафика с учетом особенностей протоколов транспортного и прикладного уровней, рассматривается влияние самоподобия на оценку качества предоставления услуг абонентам.

В работе [7] приводится всесторонний анализ эффективности функционирования телекоммуникационных сетей в условиях мультифрактального характера трафика. Анализируются теоретические и практические аспекты мультифрактального анализа производительности глобальных и локальных сетей, спутниковых систем связи, систем подвижной связи, для различных инфокоммуникационных приложений: звуковых и видеосигналов, интернет - приложений и других информационных процессов. Все модели, задачи и решения показаны на множестве реальных примеров.

Расчет показателя Хёрста. В первую очередь при анализе долгосрочной устойчивости в поведении временного ряда в рамках теории случайных блужданий используется метод, предложенный инженером-гидротехником Г. Херстом [69,70]. Расчеты в рамках так называемого метода

нормированного размаха начинаются с разделения исходного временного ряда ($\tilde{x}_i$) на неперекрывающиеся сегменты V длины S, получая, таким образом, $N_s = \mathrm{int}(N/S)$ сегментов. На втором этапе для каждого сегмента $V=0,\ldots,N_s$-1 рассчитывается профиль:

$$Y_v(j) = \sum_{i=1}^{j} (\tilde{x}_{vs+i} - \langle \tilde{x}_{vs+i} \rangle_s) = \sum_{i=1}^{j} \tilde{x}_{vs+i} - \frac{j}{s} \sum_{i=1}^{s} \tilde{x}_{vs+i}. \qquad (37)$$

За счет вычитания локальных средних устраняются кусочно-постоянные тренды. На третьем этапе рассчитывается разность между минимальными и максимальными значениями ряда (размах) $R_v(s)$ и стандартное отклонение $S_v(s)$ для каждого сегмента данных:

$$R_v(s) = \max_{j=1}^{s} Y_v(j) - \min_{j=1}^{s} Y_v(j), \qquad S_v(s) = \sqrt{\frac{1}{s} \sum_{j=1}^{s} Y_v^2(j)}. \qquad (38)$$

Затем значение нормированного размаха усредняется по всем сегментам временного ряда для получения функции флуктуации $F(s)$:

$$F_{RS}(s) = \frac{1}{N} \sum_{v=0}^{N_{s-1}} \frac{R_v(s)}{S_v(s)} \sim s^H \quad \text{для s >> 1}, \qquad (39)$$

где H – показатель Хёрста. Также H можно выразить через β и γ, $2H \approx 1 + \beta = 2 - \gamma$.

Стоит отметить, что так как $0<\gamma<1$, то правая часть (39) ограничивается неравенствами $0.5<H<1$. Отношение не сохраняется в общем случае мультифрактальных данных. Величина $H<1/2$ говорит о долгосрочном некоррелированном характере ряда, $H>1/2$ указывает на долгосрочную положительную корреляцию. Для некоррелированных данных степень корреляции убывает быстрее, чем $1/s$, $H=1/2$ для больших значений s. По сравнению со спектральным анализом, метод нормированного размаха (R/S-анализ) позволяет добиться лучшего сглаживания, при этом требует меньших

вычислительных затрат, а так же позволяет работать с кусочно-постоянными трендами.

Как отмечалось выше, параметр Хёрста может быть мерой оценки долгосрочной зависимости временного ряда. Оценка параметра не только может помочь сделать заключение о самоподобии процесса, но и позволит в дальнейшем применить к нему ряд математических методов по прогнозированию фрактальных процессов [71]. Выполнение краткосрочных прогнозов нагрузки сервера может помочь в разработке методов аппаратной и программной оптимизации сети.

Параметр Хёрста H оценивался с помощью R/S-анализа выборки данных (Рисунок 16):

$$\frac{R}{S} = \left(\frac{N}{2}\right)^{H}, \tag{40}$$

где R – размах временного ряда, S – среднеквадратичное отклонение, N – объём выборки.

Для подтверждения результатов R/S анализа расчет параметра Херста производился также с помощью периодограммного анализа, при котором для самоподобного случайного процесса вычисляется периодограмма [72]:

$$I_N(w) = \frac{1}{2\pi N}\left|\sum_{k=1}^{N} X_k e^{jk\omega}\right|, \ \omega\epsilon[0,\pi]. \tag{41}$$

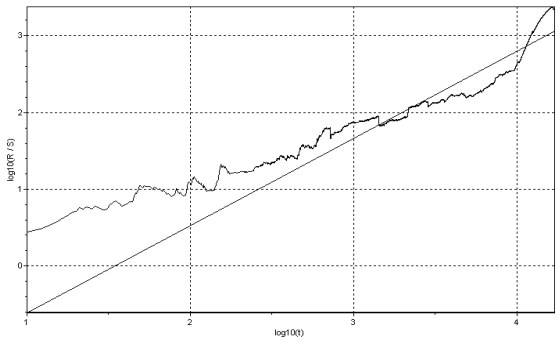
Коэффициент наклона прямой, образованной точками периодограммы в логарифмическом масштабе, будет соответствовать 1-2H. Наконец, оценка параметра Херста произведена методом агрегированных дисперсий [73].

Как и ожидалось, значение параметра указало на существование долгосрочной зависимости и самоподобия. Значения показателя Хёрста трафика, а также аппаратных процессов приведены в таблице 3.

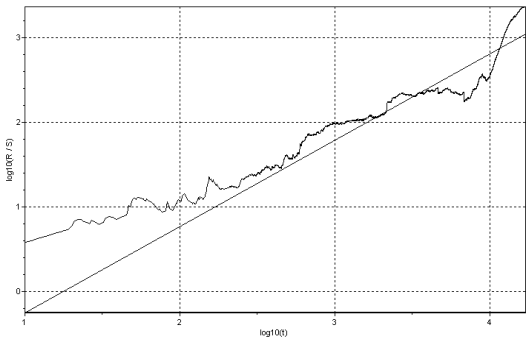
Таблица 3. Значения показателя Хёрста для сетевого трафика и основных аппаратных характеристик сервера

Характеристика	**Параметр Хёрста**		
	R/S	**Периодограммный анализ**	**Метод агрегированных дисперсий**
Объем буферизированной памяти (бит)	0,9656	0,9354	0,9984
Объем кэшированной памяти (бит)	0,9868	0,9687	0,9530
Время процессора в режиме ожидания (%)	0,9575	0,9145	0,7723
Время обработки процессором системных задач (%)	0,9903	0,9458	0,7677
Объем свободной памяти (бит)	0,9336	0,9254	0,9165
Входящий трафик (бит/с)	0,9775	0,9687	0,8655
Число процессов ОС	0,8835	0,8245	0,9259
Число запущенных процессов web – сервера	0,8343	0,8175	0,8615
Исходящий трафик (бит/с)	0,9712	0,9648	0,8511

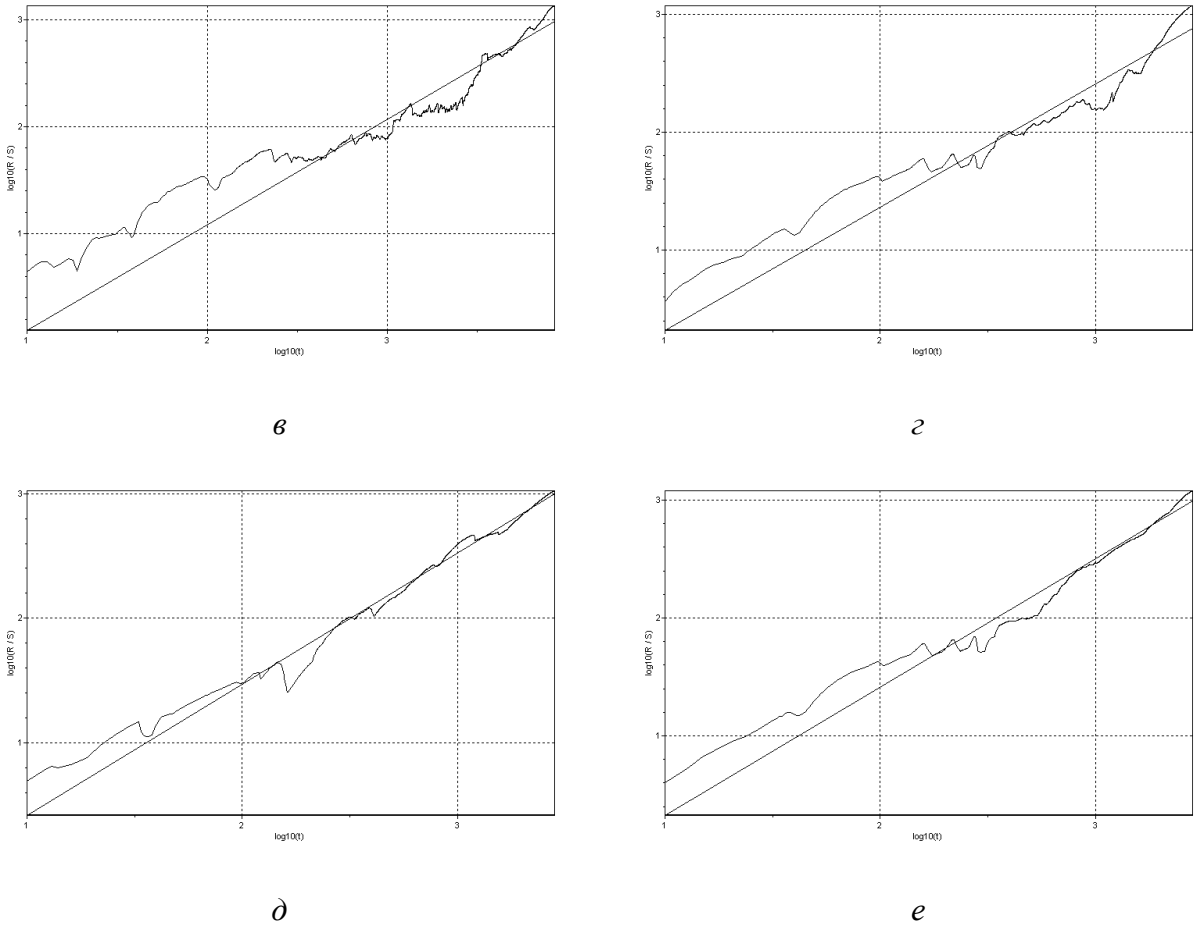
Была подтверждена гипотеза, указывающая на связь между параметром Хёрста и интенсивностью трафика (таблица 4). Так как в ряде работ [71] указывается на зависимость показателя Хёрста от числа отсчетов временного ряда, выборки брались на равных интервалах.



а



б



в *г*

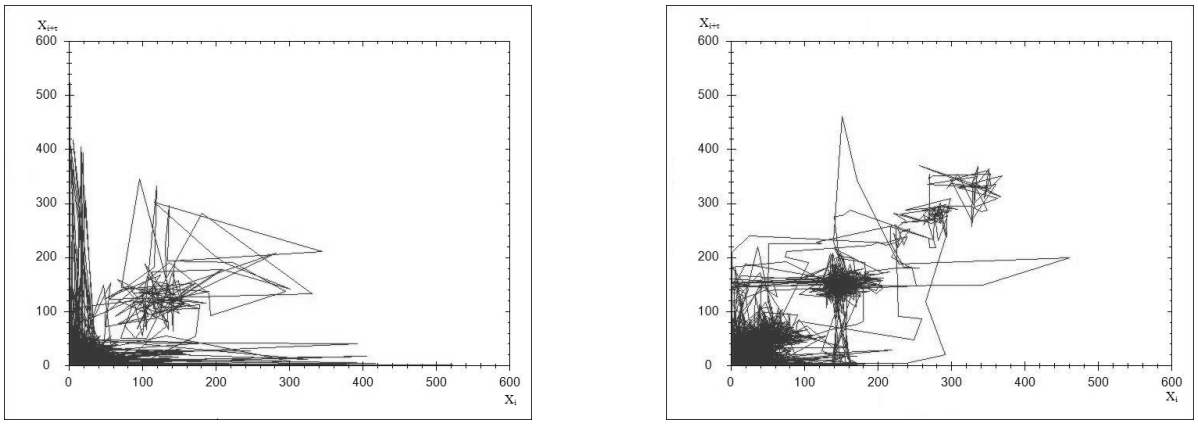
д *е*

Рисунок 16. R/S-диаграммы для входящего трафика (*а*), исходящего трафика (*б*), загрузки ЦПУ (*в*), свободного дискового пространства (*г*), числа процессов (*д*), кэшированной памяти (*е*)

Таблица 4. Зависимость показателя Хёрста от интенсивности трафика

	Значение параметра Хёрста			
	00:00 – 6:00	**06:00 – 12:00**	**12:00 – 18:00**	**18:00 – 24:00**
Входящий трафик (бит/с)	0,7142	0,6154	0,9836	0,9516
Исходящий трафик (бит/с)	0,8180	0,7385	0,9766	0,9112

Сетевой трафик как детерминированный хаос. Рассмотрим свойства процесса передачи данных по сети с позиций теории детерминированного хаоса [74]. Эта теория описывает поведение систем, чувствительных к начальным условиям. Выделение аттракторов на графике фазового пространства сигнализирует о фазовых переходных процессах, что также может позволить сделать прогноз поведения системы (Рисунок 17).



а *б*

Рисунок 17. Фазовые диаграммы входящего (*а*) и исходящего (*б*) сетевого трафика

В рамках исследования была произведена оценка экспоненты Ляпунова [75] как параметра, характеризующего хаотические системы:

$$\lambda_1(i) = \frac{1}{i\Delta t}\frac{1}{(M-i)}\sum_{j=1}^{M-i}\ln\frac{d_j(i)}{d_j(0)}, \tag{42}$$

где Δt – период выборки, $d_j(i)$ – расстояние между j-й парой ближайших соседей после i дискретных шагов, M – число восстановленных точек.

Для всех типов трафика старший показатель Ляпунова λ_1 принимает значение от 1,2 до 2,5, что в дальнейшем позволит работать с трафиком в рамках методов нелинейной динамики. Стоит отметить, что показатель Ляпунова для процессов распределения памяти сервера принимает значения ниже нуля, что говорит о высокой степени периодичности процессов, воспринимаемой как шум.

Величина корреляционной энтропии сигнала может служить мерой хаотичности процесса [76]. Для этого сначала необходимо получить корреляционный интеграл (число пар точек на расстоянии не больше r):

$$C(r) = \frac{1}{m(m-1)/2} \sum_{i=0}^{m-2} \sum_{j=i+1}^{m-1} \theta(r - p(x_i, x_j)), \quad (43)$$

где θ – функция Хэвисайда ($\theta(\alpha)=1$ при $\alpha \geq 0$, $\theta(\alpha)=0$ при $\alpha<0$), p – расстояние в n-мерном фазовом пространстве, m – число точек x_j на аттракторе. После чего рассматривается зависимость корреляционного интеграла от r и размерности фазового пространства n [77]*:*

$$C(r, n) \sim r^{D_2} \exp(-nK), \quad (44)$$

откуда получаем значение энтропии *K:*

$$K(r, n) = \ln \frac{C(r, n)}{C(r, n+1)}. \quad (45)$$

Для реализации входящего сетевого трафика на интерфейсе Eth1 максимальная корреляционная размерность принимает значение 7,24 при размерности фазового пространства равной 14 (Рисунок 18), а корреляционная энтропия равна 3,79 при размерности фазового пространства равной 3 (Рисунок 19). Графики корреляционной размерности и корреляционной энтропии реализации исходящего трафика представлены на рисунках 20 - 21.

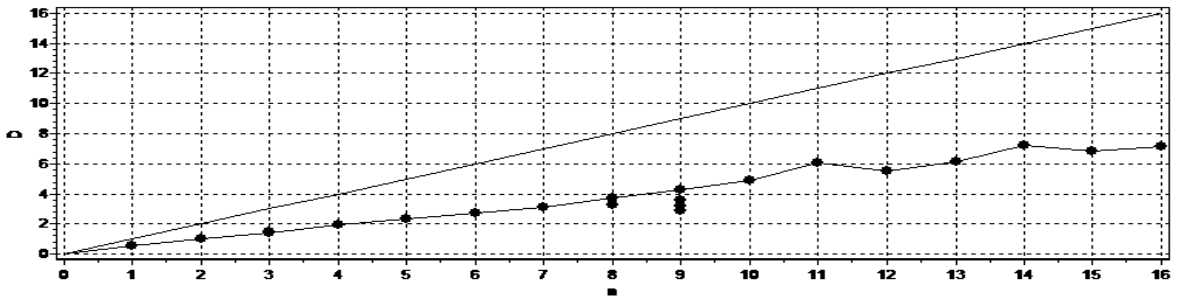


Рисунок 18. Зависимость корреляционной размерности от размера фазового пространства реализации входящего трафика на интерфейсе Eth1

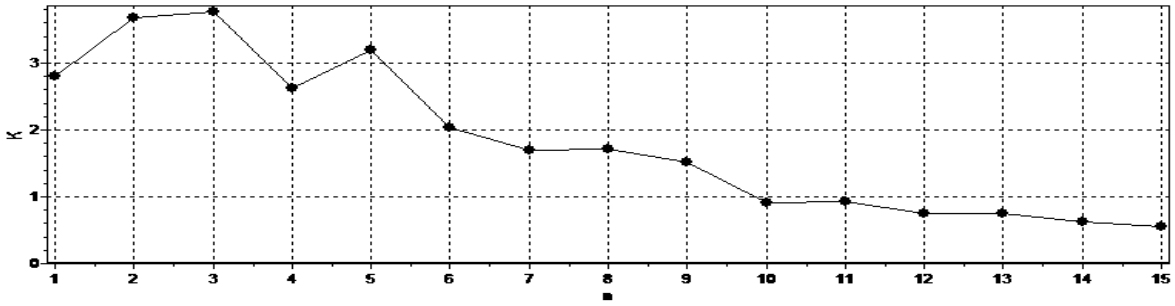


Рисунок 19. Зависимость корреляционной энтропии от размера фазового пространства реализации входящего трафика на интерфейсе Eth1

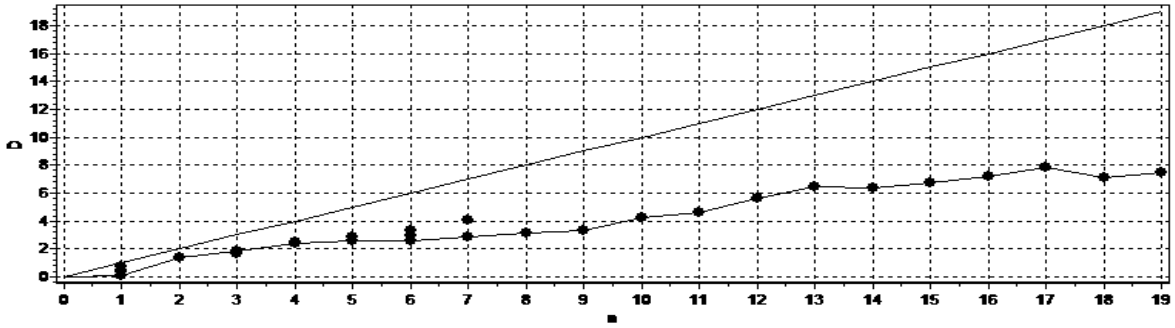


Рисунок 20. Зависимость корреляционной размерности от размера фазового пространства реализации исходящего трафика на интерфейсе Eth1 (D_{max}=7.86 при n=17)

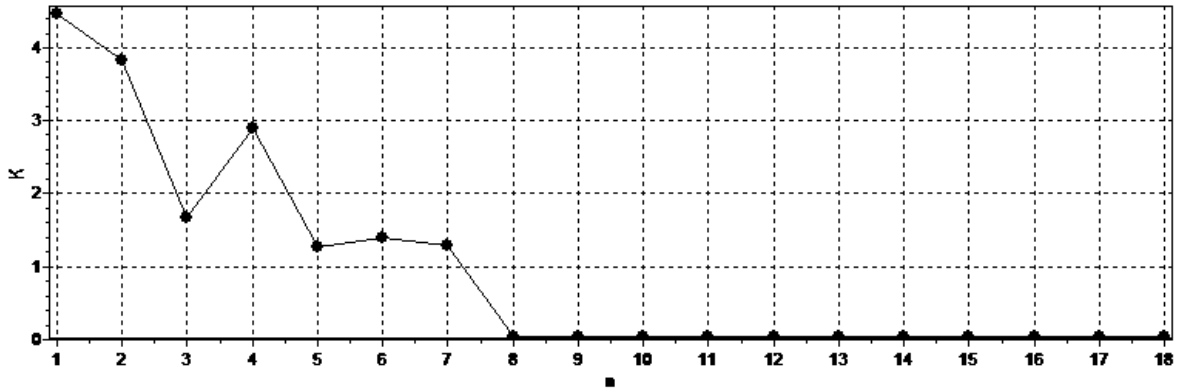


Рисунок 21. Зависимость корреляционной энтропии от размера фазового пространства реализации исходящего трафика на интерфейсе Eth1

Численное значение корреляционной энтропии не только служит показателем хаотичности процесса, но и показывает оптимальное время, на которое можно выполнить прогноз ряда [78].

Вейвлет-анализ. Преобразование Фурье – это очень полезный инструмент для анализа частотных компонентов временных рядов. Однако если использовать разложение Фурье на всем ряду, то нельзя сказать – в какой момент времени, какая частота превалирует. Оконное Фурье преобразование частично решает эту проблему путем использования некоторой выборки последовательных значений ряда определенной длины, таким образом, предоставляя данные о времени и частоте. В таком случае возникает следующая задача: величина окна ограничивает разрешение спектрограммы по частоте. Решением может служить вейвлет – преобразование, основанное на небольших функциях (вейвлетах), которые локальны по времени и частоте [79]. Вейвлеты имеют вид небольших волновых пакетов с нулевым средним, инвариантных к сдвигу и линейных к операциям масштабирования.

Вейвлет - преобразование может быть непрерывным (НВП) и дискретным (ДВП). Один из первых вейвлетов и вместе с тем наиболее простых являет вейвлет Хаара, родительская функция которого задается следующим образом:

$$\psi(x) = \begin{cases} 1, 0 \le x < 1/2 \\ -1, 1/2 \le x < 1 \\ 0, x \notin [0,1) \end{cases} \tag{46}$$

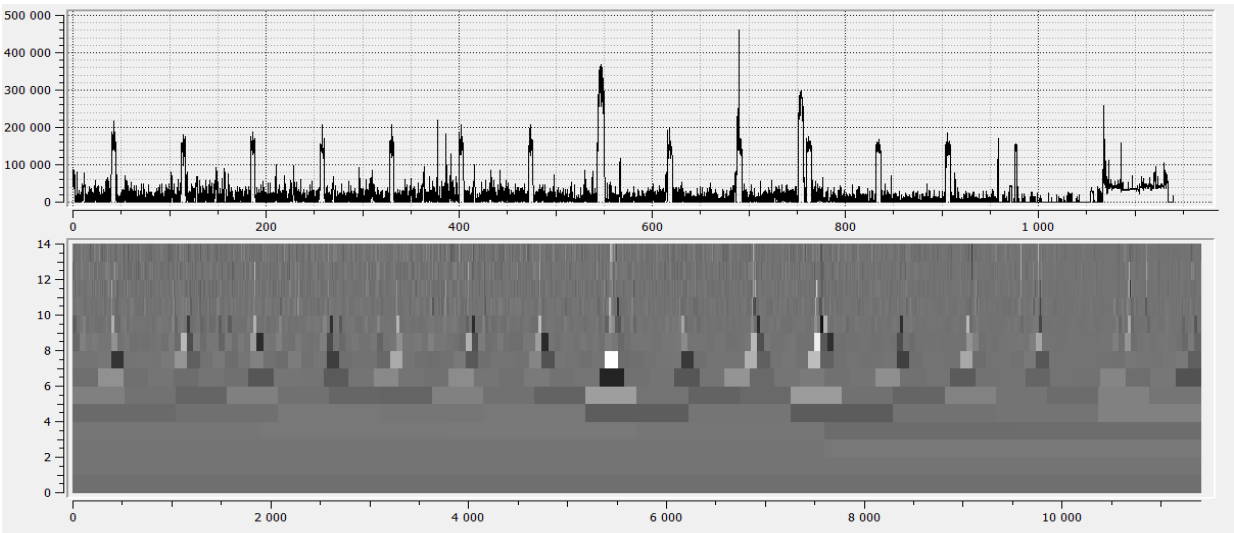


Рисунок 22. Вейвлет - диаграмма исходящего трафика на интерфейсе Eth0 (Хаар)

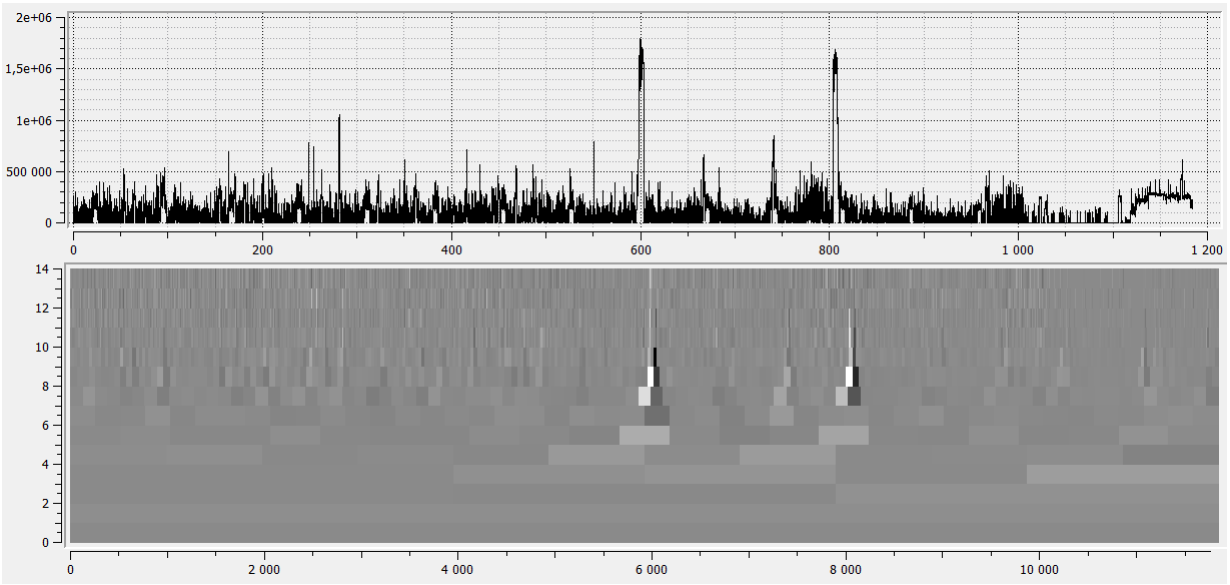


Рисунок 23. Вейвлет - диаграмма входящего трафика на интерфейсе Eth0 (Хаар)

Часто для непрерывного вейвлет – преобразования используются функции на основе производных функции Гаусса (Рисунки 24-25).

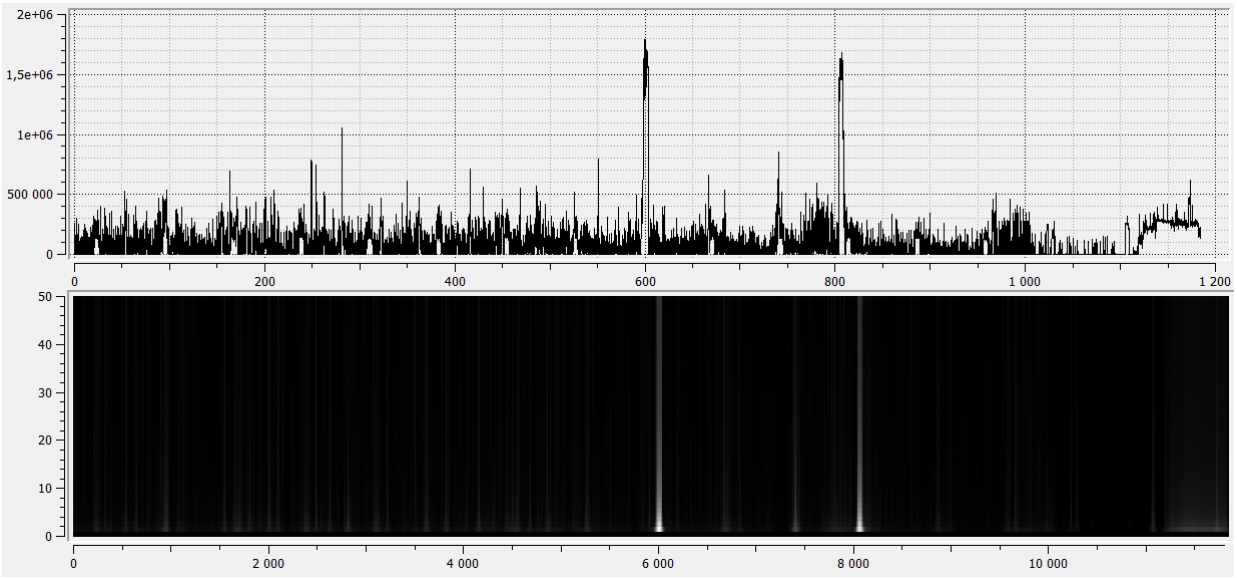


Рисунок 24. Вейвлет - диаграмма входящего трафика на интерфейсе Eth0 (Гаусс)

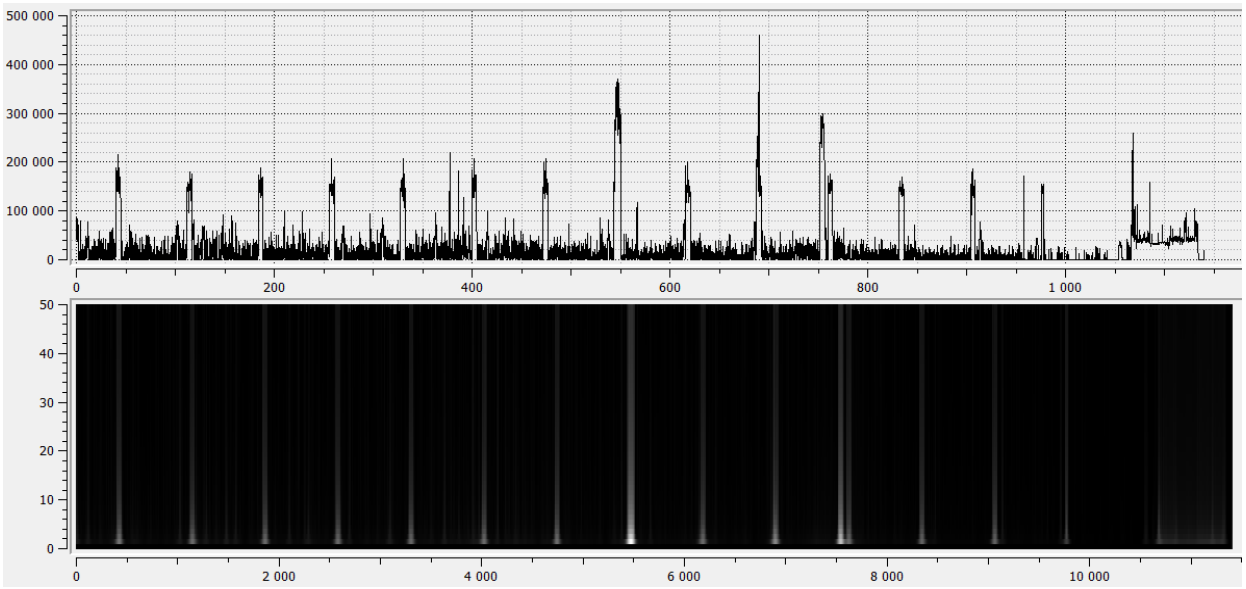


Рисунок 25. Вейвлет - диаграмма исходящего трафика на интерфейсе Eth0 (Гаусс)

На вейвлет диаграммах отчетливо видно, что всплескам трафика соответствуют, как правило, участки одинаковых частот. Также можно отметить, что повышенная сетевая нагрузка соответствует низким частотам диаграмм. Похожая ситуация возникает при обработке аппаратных данных сервера (Рисунки 26 - 29).

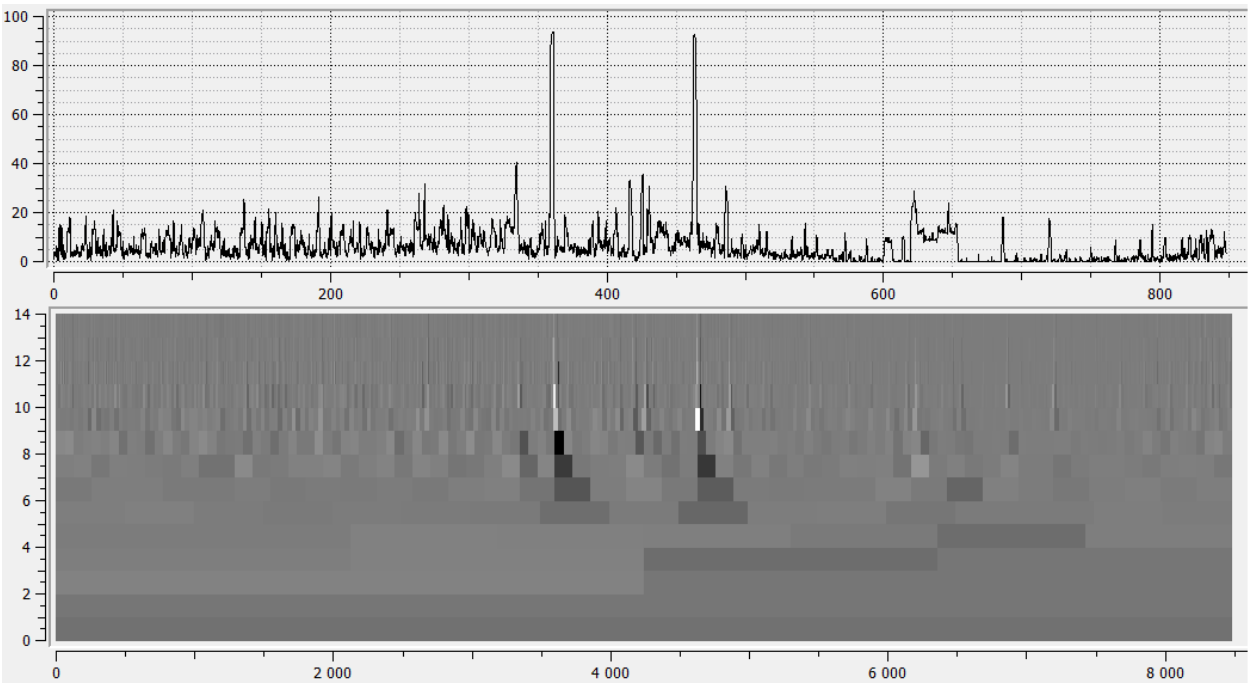


Рисунок 26. Вейвлет - диаграмма загрузки ЦП в режиме обработки пользовательских запросов (Хаар)

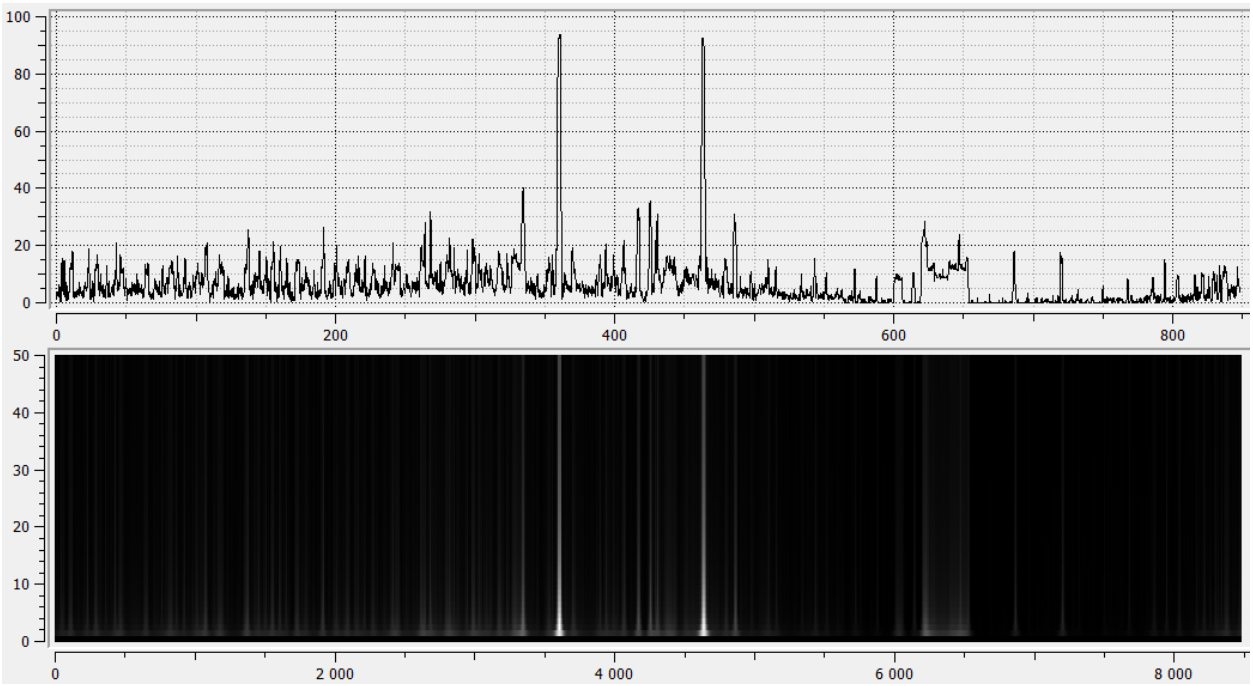


Рисунок 27. Вейвлет - диаграмма загрузки ЦП в режиме обработки пользовательских запросов (Гаусс)

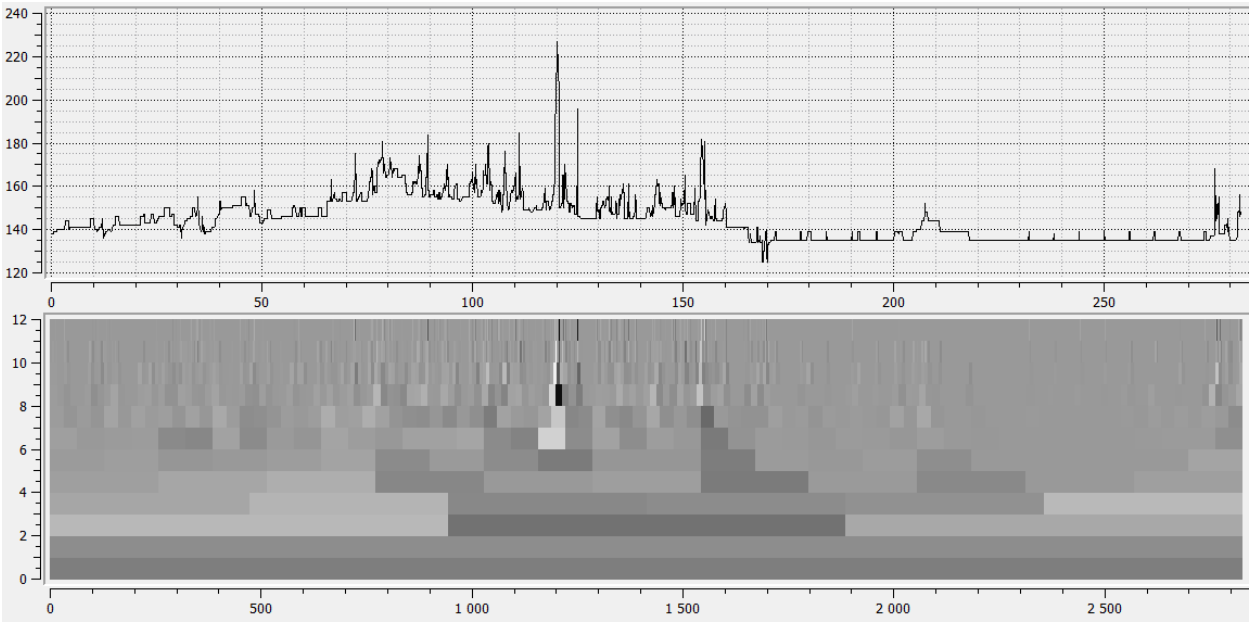


Рисунок 28. Вейвлет - диаграмма числа пользовательских процессов (Хаар)

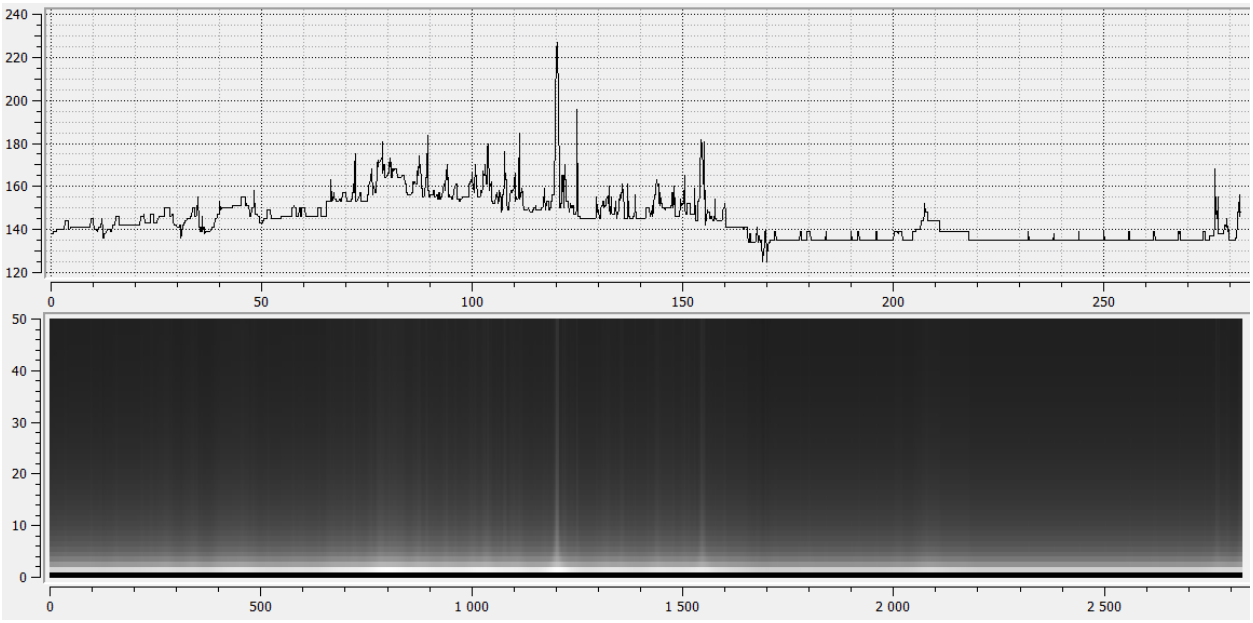


Рисунок 29. Вейвлет - диаграмма числа пользовательских процессов (Гаусс)

Выводы

1) Получены статистические и динамические характеристики процессов, протекающих в нагруженном сервере реальной корпоративной сети.

2) Анализ плотностей распределения и автокорреляционных функций говорит о присущей процессам долговременной зависимости, а также о наличии некоторого распределения с тяжелым хвостом, возможно, Стьюдента или Парето.

3) Результаты спектрального анализа указывают на наличие медленно убывающей зависимости.

4) Методами корреляционного и регрессионного анализа были выявлены основные аппаратные ресурсы сервера, наиболее чувствительные к повышению интенсивности трафика. Так, согласно рассчитанным коэффициентам корреляции, при увеличении сетевой нагрузки в первую очередь растет загрузка ЦПУ, увеличивается число процессов ОС и растет потребление оперативной памяти.

5) К собранным в ходе эксперимента данным были применены методы нелинейной динамики, рассчитаны показатель Херста и показатель Ляпунова.

6) С помощью метода нормированного размаха установлена мультифрактальная природа исследуемых процессов.

7) Значение показателя Ляпунова указывает на допустимость работы с данными в рамках методов нелинейной динамики и теории хаоса.

8) Построены фазовые диаграммы, на которых видно наличие аттракторов и, как следствие, фазовых переходов.

9) Расчет корреляционной энтропии позволил определить допустимый горизонт прогнозирования исследуемых процессов.

Для выбора максимально эффективной методики прогнозирования далее изложен сравнительный анализ известных моделей прогнозирования.

3 Сравнительный анализ методов прогнозирования в сетях передачи данных

3.1 Алгоритмы прогнозирования временных рядов

Задача прогнозирования новых значений временного ряда по исторической выборке не нова [80], однако исследования в данной области не теряют актуальности и постоянно возникают новые алгоритмы и методики прогнозирования [81]. Относительно недавно исследователи стали применять методы прогнозирования временных рядов по отношению к сетевому трафику данных для управления трафиком, обеспечения QoS и борьбы с перегрузками [82,83]. На данный момент работ в этой области немного, они редко опираются на данные по загруженности реальной корпоративной сети, а модели прогнозирования чаще авторегрессионные, реже - на основе нейросетей. К тому же алгоритмов прогнозирования временных рядов и их вариаций достаточно много, а исследование каждого из них достаточно трудоемкая задача. Таким образом, задача управления трафиком на основе краткосрочных прогнозов состояния сети остается весьма актуальной.

Формально задача прогнозирования временного ряда $X(t)=X(1)$, $X(2),\ldots,X(T)$ в момент времени T состоит в определении значений ряда $X(t)$ в моменты времени $T+1$, $T+2,\ldots,T+T'$, где T' – время упреждения или горизонт прогноза. Другими словами, определение будущего состояния процесса основывается на известных значениях ряда, а прогноз основывается на модели, отражающей функциональную зависимость между будущими и прошлыми значениями ряда. В работе [84] множество моделей прогнозирования подразделяется на статистические и структурные. К статистическим моделям относятся регрессионные и авторегрессионные, а также модели сглаживания. Модели прогнозирования на основе нейросетей, цепей Маркова и классификационных деревьев относятся к структурным моделям.

Алгоритмы сглаживания. Различные алгоритмы сглаживания временного ряда или расчета среднего значения могут использоваться в качестве простой и наглядной прогнозирующей модели [85]. Простейший алгоритм скользящего среднего заключается в вычислении следующего значения временного ряда на основании среднего за период:

$$X_{t+1} = (X_t + X_{t-1} + X_{t-2} + ... + X_{t-N-1}) / N, \quad (47)$$

где X_{t+1} – прогнозируемое значение ряда на момент t+1, X_{t-1} – значение ряда в момент t-1, N – период сглаживания. В данном случае важно подобрать подходящее значение N. К тому же логично предположить, что X_t значение ряда должно сильнее влиять на прогноз, чем $X_{t-N.}$ Алгоритм экспоненциального сглаживания учитывает этот недостаток через некоторый коэффициент:

$$X'_{t+1} = \alpha X_t + (1-\alpha) X'_t, \quad (48)$$

где $0<\alpha<1$ – весовой коэффициент, X_t – реальное значение ряда в момент t, а X'_t – прогнозное значение ряда. Таким образом, мы получаем взвешенную скользящую среднюю, в которой следующее прогнозируемое значение формируется из прогнозируемого и реального текущего значения ряда. При этом коэффициент подбирается в соответствии с характером ряда – чем сильнее меняются значения ряда, тем больше должен быть коэффициент, что быстрее «приспособиться» к большим изменениям. И чем выше шумовая составляющая ряда, тем меньше должно быть значение коэффициента, чтобы сглаживать флуктуации. Для решения задачи подбора коэффициента α разработан алгоритм адаптивного сглаживания.

Существует множество вариаций алгоритма адаптивного сглаживания, в самом простом случае коэффициент α подбирается следующим образом:

$$\alpha_{t+2} = \left| (X'_{t+1} - X_{t+1}) / X_{t+1} \right|. \quad (49)$$

При этом если значение коэффициента получается равным или более 1, то принимается *α=0.99*(*9*), а если равным 0, то *α=0.0*(*0*)*1*. Рассмотренные алгоритмы сглаживания временных рядов не учитывают наличие у ряда тренда и сезонной составляющей.

Алгоритм экспоненциального сглаживания с учетом тренда при сглаживании учитывает направление ряда:

$$L_t = \alpha X_t + (1-\alpha)(L_{t-1} + T_{t-1}) \tag{50}$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}, \tag{51}$$

где L –уровень тренда, а T – тренд, при этом $0<\alpha<1$ и $0<\beta<1$. Тогда прогнозируемое значение ряда на m шагов:

$$X_{t+m} = L_t + (T_t m). \tag{52}$$

С учетом сезонной составляющей ряда прогноз строится следующим образом [85]:

$$L_t = \alpha(X_t / SA_{t-c}) + (1-\alpha)(L_{t-1} + T_{t-1}) \tag{53}$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} \tag{54}$$

$$SA_t = \gamma(X_t / L_t) + (1-\gamma)(SA_{t-C}), \tag{55}$$

где L – уровень тренда, T-тренд, SA_t – сезонная составляющая ряда на момент времени t, C – размер цикла для сезонной составляющей, $0<\alpha<1$, $0<\beta<1$, $0<\gamma<1$. Тогда прогнозируемое значение ряда на m шагов вперед:

$$X_{t+m} = (L_t + (T_t m))SA_{t-C+m} \tag{56}$$

Несмотря на попытки не просто усреднить ряд скользящим окном, а учесть его сезонную, трендовую и циклическую составляющую, прогнозы на основе сглаживающих моделей отфильтровывают достаточно много информации о реальном процессе. Вместе с шумами устраняется информация, которая

могла бы быть использована при прогнозировании ряда. Рассмотренные далее регрессионные и авторегрессионные модели отличаются большей точностью, учитывают больше факторов, влияющих на распределение ряда.

Регрессионные и авторегрессионные алгоритмы. В статистической обработке данных регрессионный анализ применяется для определения зависимости между исходной зависимой переменной и внешними независимыми переменными, т.н. регрессорами. Простейшая регрессионная модель называется линейной и предполагает наличие единственного внешнего фактора, воздействующего на процесс с линейной связью:

$$X_t = \beta_0 + \beta_1 Y_t + \varepsilon_t, \tag{57}$$

где Y_i – независимая переменная, β_0 и β_1 – коэффициенты регрессии, ε_t – ошибка модели. В случае множественной регрессионной модели внешних факторов, влияющих на значения ряда – несколько:

$$X_t = \beta_0 + \beta_1 Y_t^1 + \beta_2 Y_t^2 + ... + \beta_n Y_t^n + \varepsilon_t \tag{58}$$

Зависимость между исходным процессом и регрессором не обязательно должна быть линейной, возможен случай, при котором зависимость описывается произвольной функцией.

Зачастую при исследовании процесса сложно выделить внешние воздействующие факторы, при этом регрессионный анализ основывается только на самих значениях ряда, такие модели называются авторегрессионными. Одна из наиболее известных авторегрессионных моделей – ARMA (Autoregressive Moving Average) – представляет собой комбинацию авторегрессионной модели и скользящего среднего. Авторегрессионная (AR(p)) модель может быть выражена следующим образом:

$$X_t = c + \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + ... + \varphi_p X_{t-p} + \varepsilon_t, \tag{59}$$

где X_t – текущее значение ряда, ε_t – случайная ошибка, $\varphi_i(i=1,2,\ldots,p)$ – весовые коэффициенты, а c – константа. В записи AR(p), p – это порядок модели. Немного видоизмененная запись представленной ранее модели скользящего среднего MA(q):

$$X_t = \mu + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \ , \tag{60}$$

где μ – среднее значение выборки, $\theta_j(j=1,2,\ldots,q)$ – параметры модели и q-порядок модели. Тогда ARMA(p,q) модель:

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} \tag{61}$$

ARMA модель работает с предположением, что ряд стационарен. Операция устранения тренда в лучшем случае может привести к стационарности ряда или хотя бы устранить сам тренд. Если в качестве исходных данных используется не сам временной ряд, а разность его компонентов, то в таком случае модель называется ARIMA(p,d,q) – Autoregression integrated moving average, где d – порядок разности между значениями ряда. На основе ARMA модели разработано множество других прогнозирующих моделей. FARIMA – одна из них, при которой операция детрендинга проводится с дробным порядком [86]. SARIMA – модель, в которой учтена сезонная составляющая ряда [87].

Алгоритмы на основе нейросетевых моделей. Нейронные сети пользуются большой популярностью в задаче прогнозирования временных рядов из-за возможности имитации процессов с множественным внешним влиянием. Эффективны в задачах классификации объектов и распознавания образов. Принципиально структура нейрона ANN похожа строением на человеческий нейрон и представлена на рисунке 30.

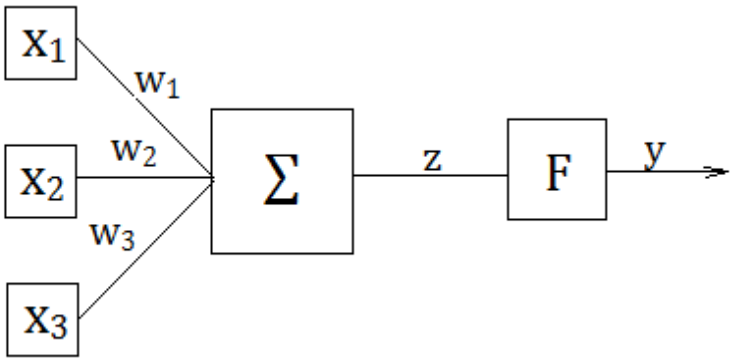


Рисунок 30. Принципиальная схема нейрона

На вход искусственного нейрона поступает вектор входных сигналов $X_1,X_2,X_3,\ldots,X_n$, для которых высчитывается среднее значение z, которое используется в активационной функции F для расчета выходного значения Y. Активационная функция может быть функцией единичного скачка, линейной функцией и т.д, например:

$$F(z)=\frac{1}{1+e^{-z}} \tag{62}$$

Пример ANN, состоящей из 3 выходных и 2 скрытых нейронов, представлен на рисунке 31. Входной слой служит для распределения входного вектора значений и не содержит нейронов. Отметим, что нейроны не все нейроны обязательно должны быть связаны между собой, то есть для некоторых связей w может быть нулевым. Также нейронная сеть может обладать обратной связью для перенастройки весов. Весовые коэффициенты связей назначаются в процессе обучения, а также могут пересчитываться в зависимости от результатов работы сети.

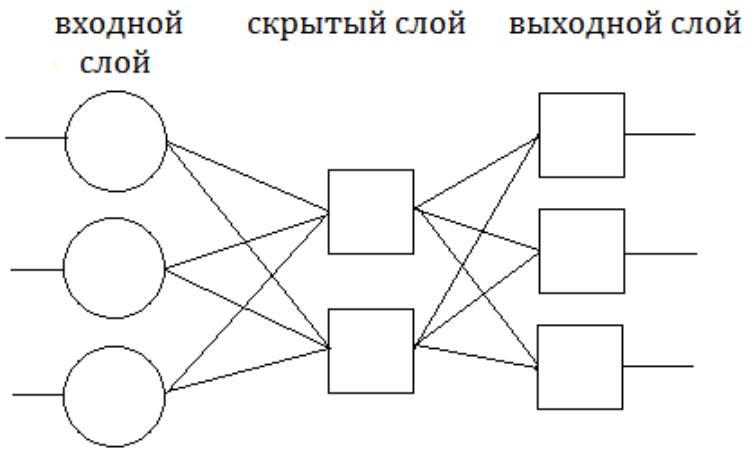


Рисунок 31. Структурная схема нейронной сети

Согласно [88] можно выделить основные особенности ANN(Artificial Neural Network):

- ANN могут моделировать нелинейные зависимости между входными и выходными данными.
- ANN обучается на входных данных, то есть данные определяют модель между входными и выходными данными.
- ANN могут обобщать входные данные, что не приводит к понижению эффективности работы сети при изменении характера данных.
- В отличие от статистических моделей, ANN вносить предположения о распределении входных данных.

В ходе использования ANN возникает ряд трудностей, которые согласно [89] выглядят следующим образом:

- Сложность подбора оптимальной комбинации параметров сети, таких как скорость обучения, число скрытых слоев, число скрытых нейронов на каждом слое.

- Сложность выбора алгоритма обучения и строгие требования к обучающей выборке.

- Обученная нейронная сеть служит некоторым «черным ящиком», то есть достаточно сложно выделить совокупность правил, по которым ANN принимает решения.

Генетические и эволюционные алгоритмы. Весьма нетривиальной является задача выбора подходящей модели прогнозирования, а также подбора параметров для линейной статистической модели или настройки нелинейной модели. Применение генетических алгоритмов может помочь в решении такой задачи.

Первые работы в данном направлении описывали использование генетических алгоритмов в подборе коэффициентов для регрессионных [90] и авторегрессионных моделей [91]. При этом популяция формировалась скользящим окном по исходному ряду данных, а коэффициенты регрессии служили вектором генов. После дальнейших скрещиваний и мутаций формировалась новая популяция, приспособленность которой определялась точность прогнозирования ряда.

В более современных работах [92] генетические алгоритмы используются для настройки искусственных нейронных сетей, то есть выбора архитектуры сети: числа входных нейронов, числа выходных нейронов, веса синапсов (в том числе нулевые, если необходимо убрать связь между некоторыми нейронами).

Алгоритм поиска оптимальной ANN для некоторого числового ряда следующий:

- Случайным образом генерируется исходная популяция (набор хромосом) нейронных сетей.

- Рассчитывается фенотип (архитектура ANN) и значение оптимизационной функции для каждой «особи» популяции.

- Применяются методы генетических алгоритмов, такие как элитизм, отбор удачных особей, кроссовер и мутация, для формирования новой популяции.
- Предыдущие два шага повторяются некоторое ограниченное число раз до наступления эффекта переобучения сети.

Методы локальной аппроксимации применяются для прогнозирования хаотических и квазипериодических рядов, то есть процессов, в которых может отсутствовать глобальная линейная составляющая [93]. То есть прогнозирование основывается на локальной подпоследовательности ряда, при этом модель все еще может оставаться линейной в рамках некоторой локальной выборки. Выборка же формируется не по временной близости значений ряда, а по близости в пространстве задержек.

Основная особенность метода заключается в том, что длина прогноза определяется не возможностями модели, а динамическими свойствами самого ряда. При этом методом локальной аппроксимации оценивается динамика ряда, а анализ основных закономерностей осуществляется с помощью других методов, например сингулярного спектрального анализа (SSA).

Алгоритм построения прогноза на один шаг вперед примерно следующий [94]:

- Построение матрица задержек и выбор локального представления.
- Определение числа соседей.
- Оценка параметров выбранной модели и построение прогноза в предположении, что X изменяется по тому же закону и с теми же параметрами, что и его соседи.

Прогнозирование значений ряда методом локальной аппроксимации может быть выполнено одним из следующих способов:

- Итеративный. При этом коэффициенты влияния отклонений в рамках каждой из матриц задержек и среднее значение прогнозов от каждого из соседей рассчитываются один раз, а спрогнозированный вектор используется как новый стартовый.
- Итеративный с пересчетом. Отличие от предыдущего заключается в пересчете всех параметров после получения прогноза и выборе новых соседей.
- Прямой. Заключается в том, что после вычисления прогнозного значения, оно не добавляется к исходным данным в дальнейших расчетах. Преимущество такого подхода заключается в том, что таким образом не происходит накопления ошибки прогноза.

Методы локальной аппроксимации имеют много общего с широко известными авторегрессионными методами прогнозирования рядов. Однако, за счет использования локально - кусочной линейной аппроксимации вместо глобальной линейной, методы ЛА позволяют учитывать квазипериодическую и хаотическую природу процесса, чего невозможно добиться с помощью авторегрессионных методов [95].

Способы оценки точности прогноза. Для определения степени адекватности модели прогнозирования необходимо использовать ряд методик по оценке точности прогноза, в частности – ошибки прогнозирования. Ошибка прогнозирования e_τ – это разница между реализацией ряда и прогнозным значением ряда:

$$e_\tau = X_{T+\tau} - X'_{T+\tau}, \tag{63}$$

где $X_{T+\tau}$ – реальное значение ряда, $X'_{T+\tau}$ – прогнозное значение ряда.

Расчет среднего значения ошибки прогнозирования (Mean Forecast Error) MFE – самый простой способ оценить точность прогнозирования:

$$MFE = \frac{1}{n}\sum_{t=1}^{n} e_t \tag{64}$$

Позволяет оценить среднее отклонение прогнозных значений от реализации ряда, а также знак или направление ошибки. При этом абсолютное значение MFE еще не означает высокой точности прогнозирования. К тому же MFE зависит от временного масштаба ряда и слабо учитывает экстремальные значения ошибки прогноза.

Абсолютная средняя ошибка прогнозирования MAE (Mean Absolute Error) [96]:

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |e_t| \tag{65}$$

Позволяет рассчитать среднее абсолютное отклонение прогноза от реальных значений ряда, оценить суммарную ошибку прогноза. В отличие от MFE, отрицательные и положительные по значению ошибки прогнозирования не компенсируют друг друга, при этом невозможно оценить направление ошибки. Также зависит от временного масштаба выборки и слабо учитывает экстремальные значения ошибок.

Средняя абсолютная ошибка в процентах MAPE (Mean Absolute Percentage Error):

$$MAPE = \frac{1}{n}\sum_{t=1}^{n} \left|\frac{e_t}{X_t}\right| \times 100 \tag{66}$$

Оценивает средний процент ошибки прогнозирования без учета знака. MAPE не отражает экстремальных значений ошибок, при этом противоположные по знаку ошибки не компенсируют друг друга.

Знаковая среднеквадратичная ошибка SMSE (Signed Mean Squared Error):

$$SMSE = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{e_t}{|e_t|}\right)e_t^2 \tag{67}$$

Показывает среднеквадратичное отклонение ошибки прогноза с учетом знака.

Нормальная среднеквадратичная ошибка (Normalized Mean Squared Error) NMSE:

$$NMSE = \frac{1}{\sigma^2 n}\sum_{t=1}^{n} e_t^2 \ , \tag{68}$$

$$\sigma^2 = \frac{1}{n-1}\sum_{t=1}^{n}\left(X_t - \overline{X}\right), \tag{69}$$

$$\overline{X} = \frac{1}{n}\sum_{t=1}^{n} X_t \ . \tag{70}$$

Достаточно эффективный способ оценки точности прогнозирования, при котором чем ниже значение NMSE, тем точнее модель.

Как правило, при оценке точности метода прогнозирования временного ряда, используется ряд метрик, позволяющих рассчитать абсолютное значение ошибки и её направленность.

Рассмотренные алгоритмы прогнозирования процессов являются общеизвестными и в задачах исследования временных рядов применяются достаточно давно. Однако для прогнозирования реального процесса, в силу его характера, некоторые модели подходят лучше других. При этом подбор подходящей модели, описывающей исследуемый процесс, является нетривиальной задачей, решение которой необходимо базировать на точных статистических и динамических данных о природе самого процесса.

Согласно известной методологии Бокса-Дженкинса [80], анализ и прогноз временных рядов можно разделить на три этапа:

- Идентификация модели, наиболее точно описывающей процесс:
    - проверка ряда на стационарность;

- анализ полной и частной АКФ.

- Проведение оценки модели и проверка её адекватности:
  - оценка параметров модели, рассчитанных на первом этапе.
  - проверка модели на соответствие исходным данным.

- Расчет прогнозных значений ряда.

Зачастую для нестационарных рядов свойство стационарности достигается операцией устранения тренда, то есть взятия разности между соседними элементами (Рисунок32).

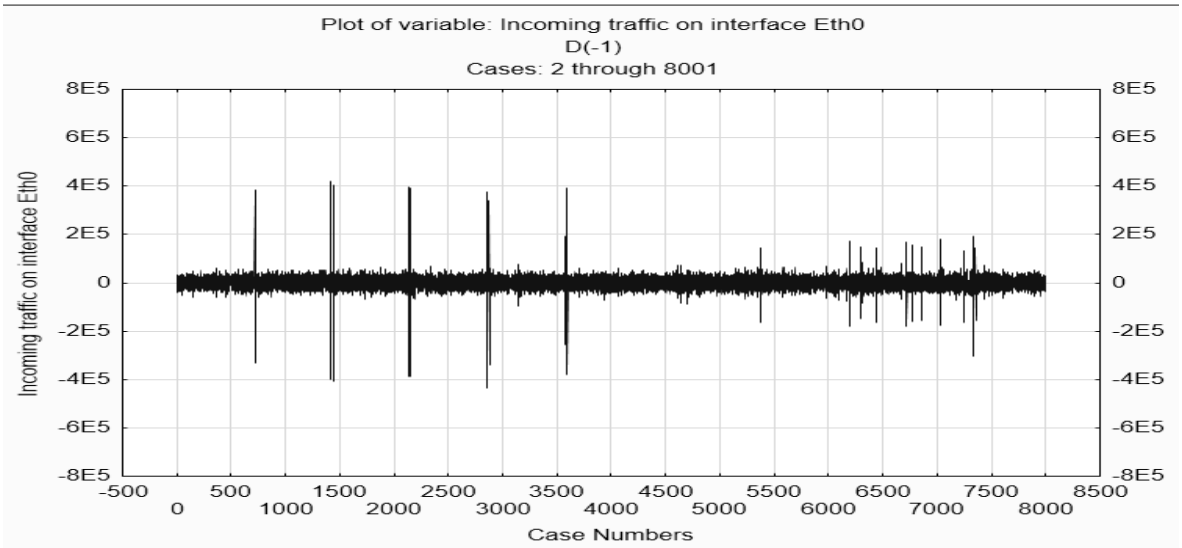


Рисунок 32. График интенсивности входящего трафика (Eth0) после операции устранения тренда

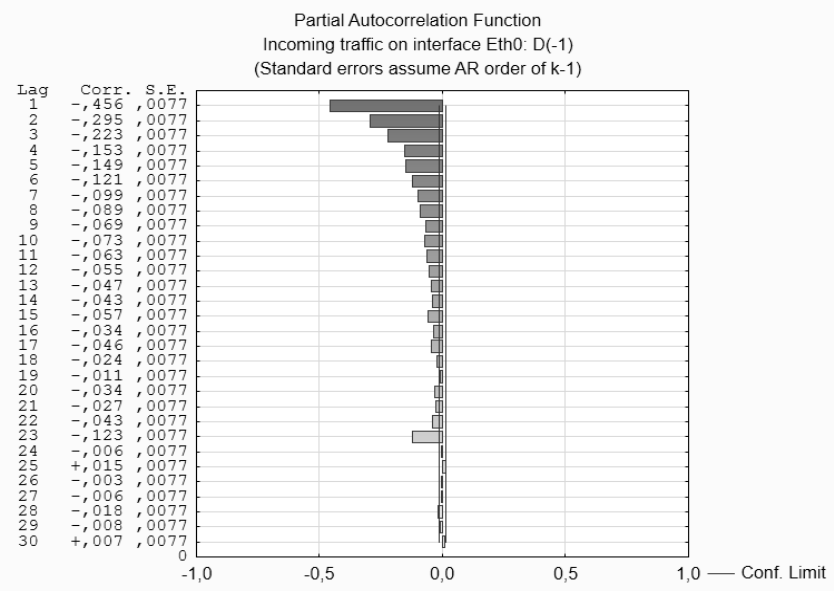


Рисунок 33. График ЧАКФ интенсивности входящего трафика (Eth0) после операции устранения тренда

Порядок модели авторегрессии (AR(p)) выбирается по графику ЧАКФ (Рисунок 33), как последний ненулевой элемент, в данном случае p=23, то есть модель AR(23). Порядок модели скользящего среднего (MA(q)) выбирается по графику АКФ (Рисунок 34), как последний ненулевой элемент, то есть q=24, то есть MA(24). Авторегрессионная проинтегрированная модель скользящего среднего в таком случае будет ARIMA(23,1,24) [97].

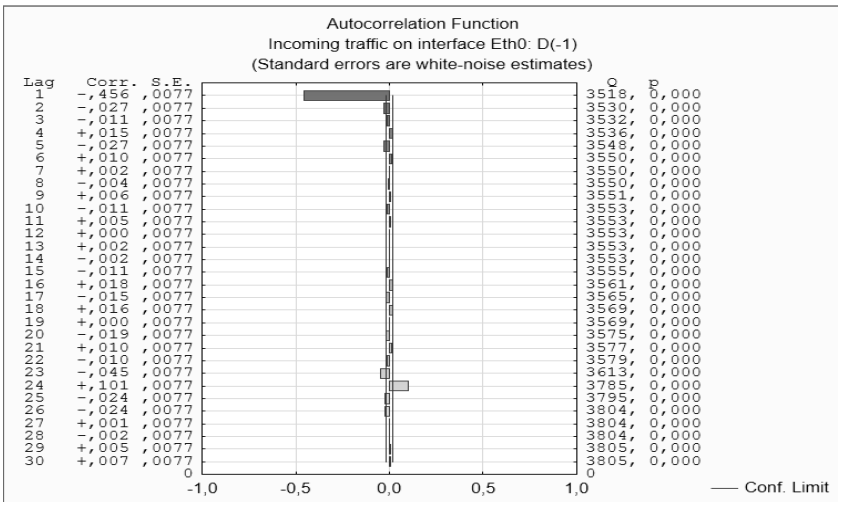


Рисунок 34. График ЧАКФ интенсивности входящего трафика (Eth0) после операции устранения тренда

В дальнейшем прогноз строился согласно изложенному алгоритму на один шаг вперед с целью выполнить анализ эффективности различных алгоритмов прогнозирования применительно к исследуемой области.

Стоит отметить, что оценка точности прогноза проводилась одним из наиболее простых способов из-за больших числовых значений данных и резких кратковременных повышений интенсивности исследуемых процессов, что при использовании других способов оценки прогноза приводит к необъективным количественным значениям.

3.2 Прогноз на основе AR(*p*) – модели

Расчеты проводились на собранных в ходе эксперимента данных, оценивалась точность прогноза, а также подбор значений параметра *p* модели AR(*p*) не только по АКФ, но и простым перебором значений порядка модели (Приложение 1.1). Графически результаты прогнозирования представлены на рисунках 35-38.

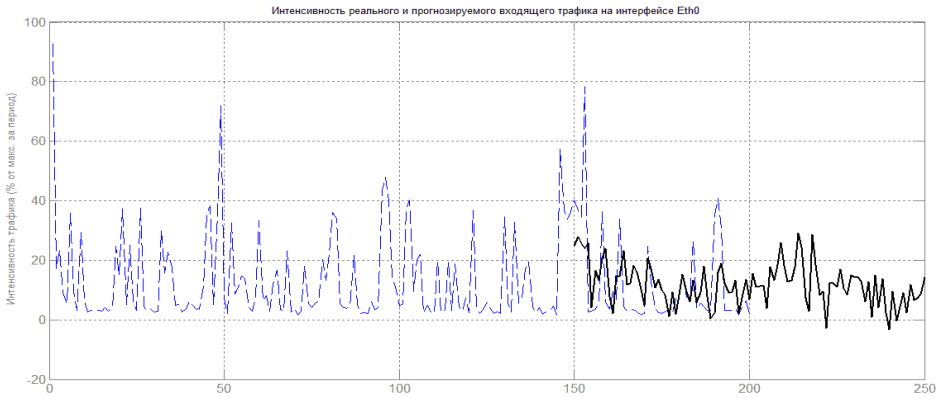


Рисунок 35. Интенсивность реального и прогнозируемого входящего трафика (Eth0)

Рисунок 36. Интенсивность реального и прогнозируемого входящего трафика (Eth1)

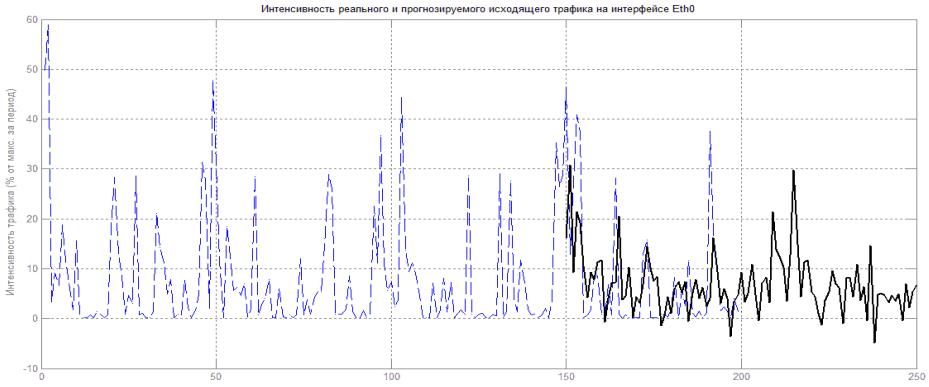


Рисунок 37. Интенсивность реального и прогнозируемого исходящего трафика (Eth0)

Рисунок 38. Интенсивность реального и прогнозируемого исходящего трафика (Eth1)

Ошибка прогноза рассчитывалась по известной формуле:

$$ME = \frac{1}{n}\sum_{t=1}^{n} e_t \,, \qquad (71)$$

где $e_\tau = X_{T+\tau} - X'_{T+\tau}$ - это ошибка прогнозирования. Для процессов передачи трафика MAE, как правило, принимала значения около 12%, рисунки 39-42.

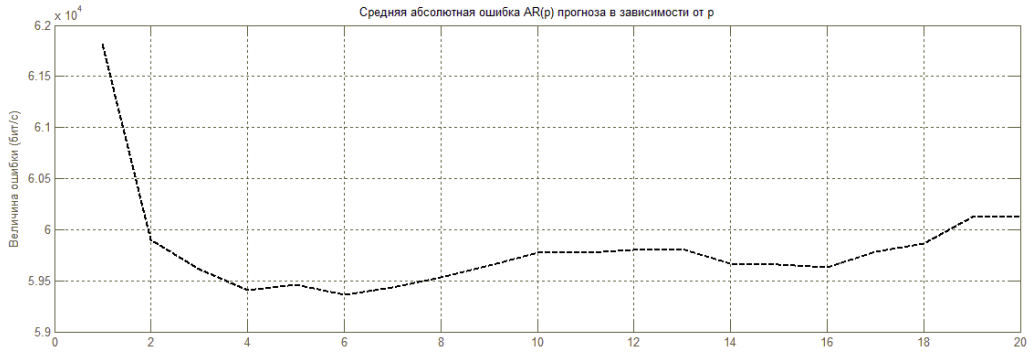


Рисунок 39. ME AR(p) прогноза в зависимости от p для входящего трафика (Eth1)

Рисунок 40. ME AR(p) прогноза в зависимости от p для входящего трафика (Eth0)

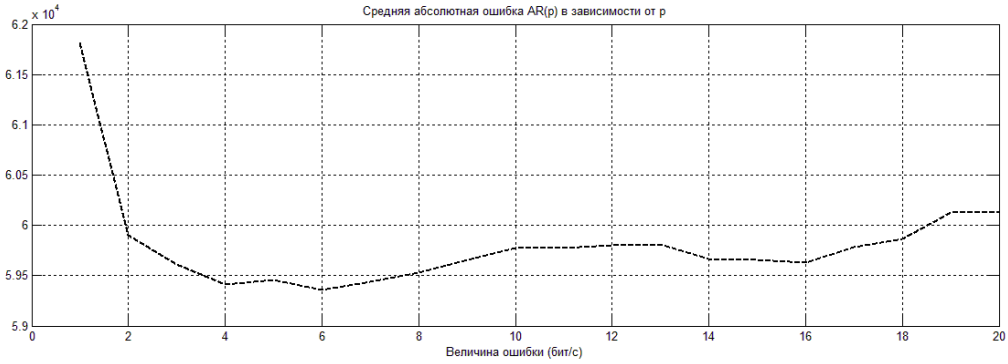


Рисунок 41. ME AR(p) прогноза в зависимости от p для исходящего трафика (Eth0)

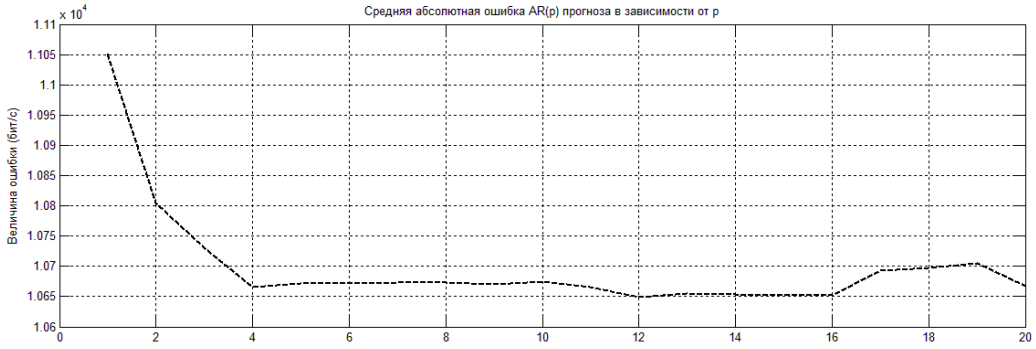


Рисунок 42. ME AR(p) прогноза в зависимости от p для исходящего трафика (Eth1)

Стоит заметить, что процессы распределения аппаратных ресурсов сервера отличаются большей линейностью и наличием периодических составляющих, что было отмечено еще при статистическом анализе. Для таких процессов AR(p) модель дает большую точность и качество прогнозирования. Например, ME для процесса загрузки ЦП составила всего 1-3% от реального значения.

Рисунок 43. Реальная загрузка ЦП и прогнозируемое значение

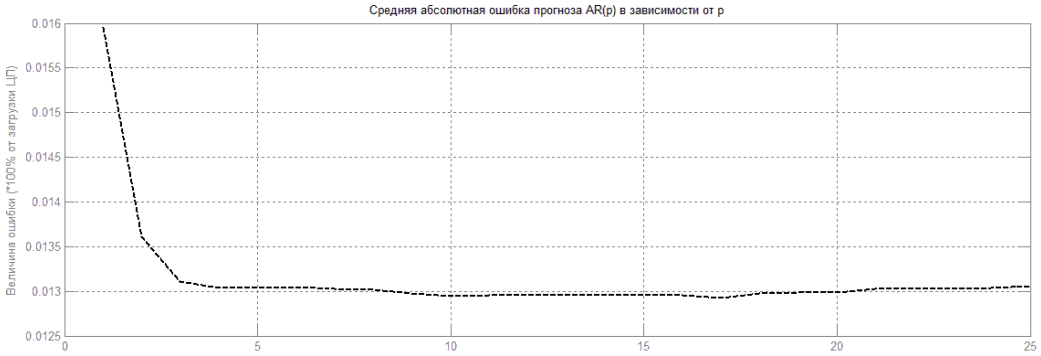


Рисунок 44. ME AR(p) прогноза в зависимости от p для загрузки ЦП

Таблица 5. Ошибка прогноза AR(p) модели

Прогнозируемый процесс	**Ошибка прогноза (ME)**
Входящий трафик Eth0	12,33%
Исходящий трафик Eth0	11,81%
Входящий трафик Eth1	12,26%
Исходящий трафик Eth1	11,98%
Загрузка ЦП	6,14%
Объем свободной памяти	5,97%

3.3 Прогноз на основе ARIMA(p,d,q) – модели

ARIMA(p,d,q) модель представляет собой совокупность авторегрессионной модели и скользящего среднего, где p – порядок AR(p) модели, q – порядок MA(q) модели, d – порядок разности между значениями ряда (для устранения тренда). ARIMA(p,d,q) модель может быть представлена следующим выражением:

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} \ . \qquad (72)$$

Для оценки подходящих значений p, d и q – порядков модели был осуществлен расчет ошибки прогнозирования на основе ARIMA(p,d,q), где p и q принимали значения от 1 до 10.

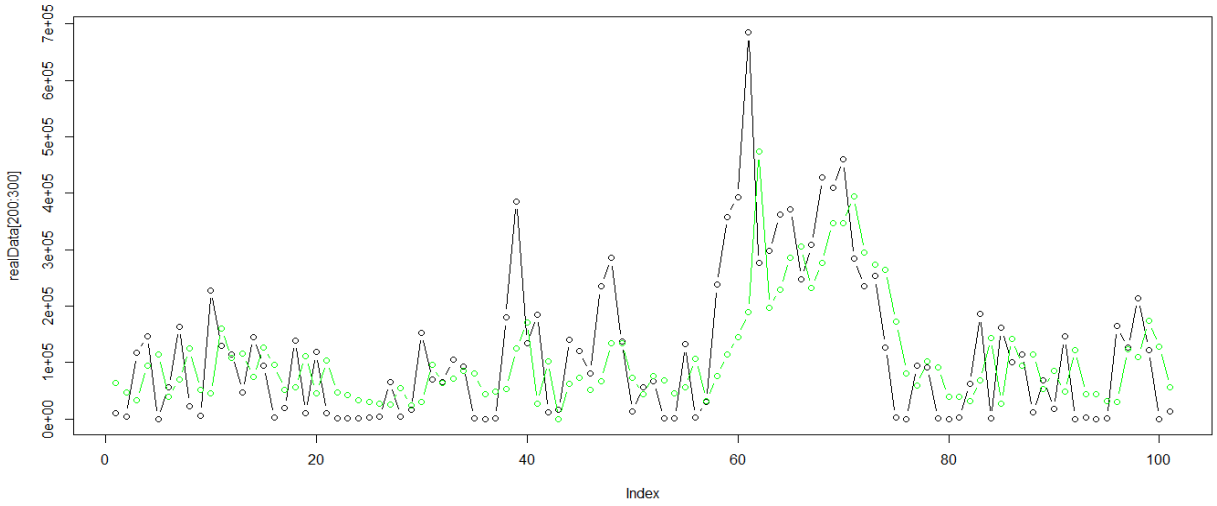


Рисунок 45. Интенсивность реального и прогнозируемого входящего трафика (Eth1)

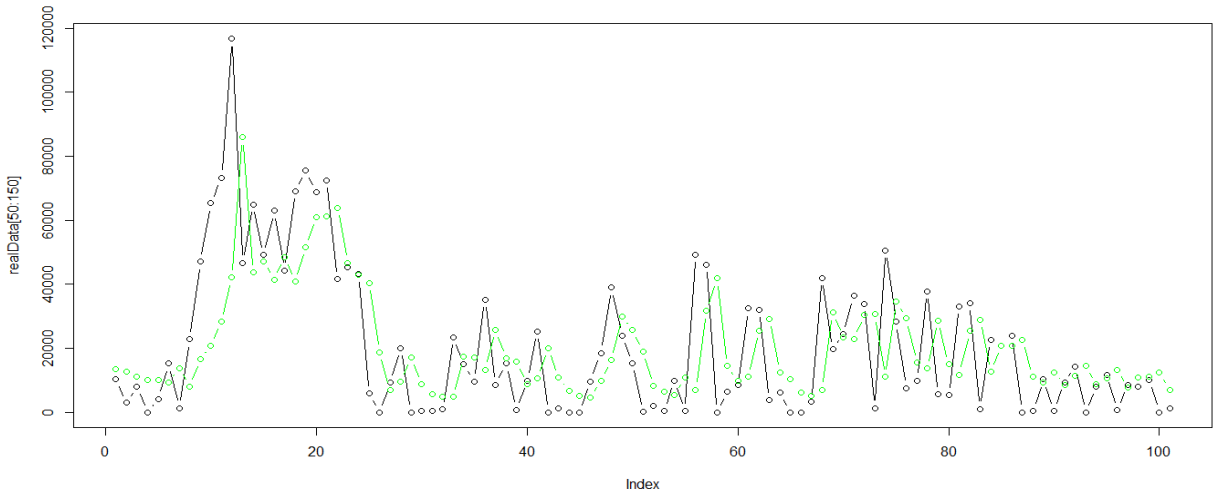


Рисунок 46. Интенсивность реального и прогнозируемого исходящего трафика (Eth1)

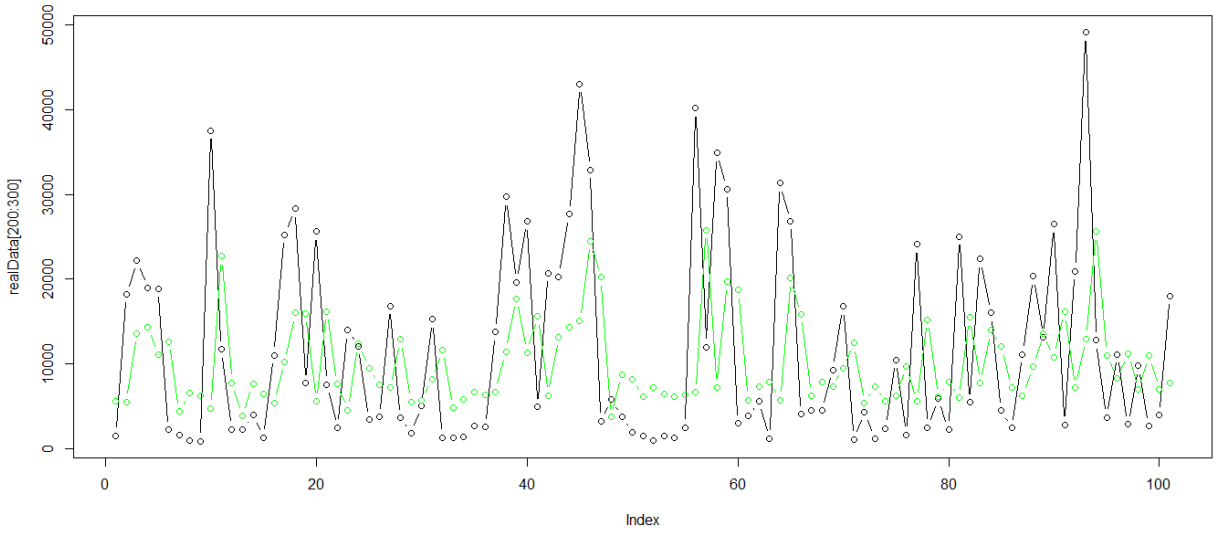


Рисунок 47. Интенсивность реального и прогнозируемого входящего трафика (Eth0)

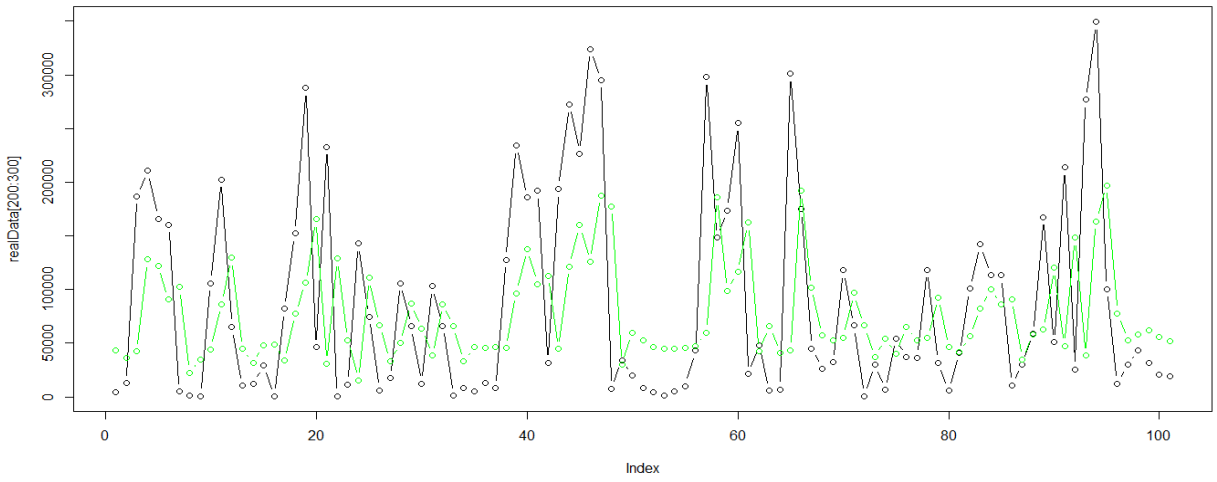


Рисунок 48. Интенсивность реального и прогнозируемого исходящего трафика (Eth0)

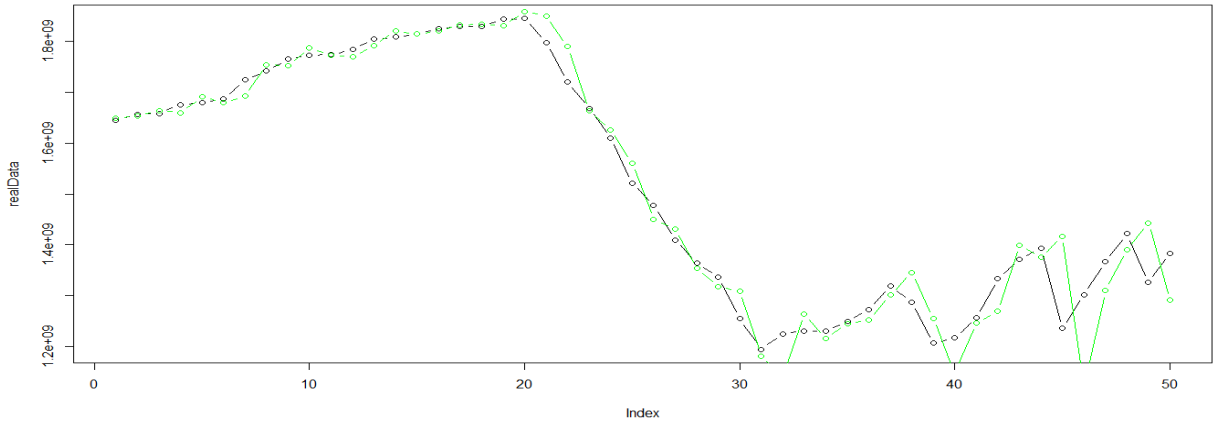


Рисунок 49. Реальный и прогнозируемый объем свободной памяти

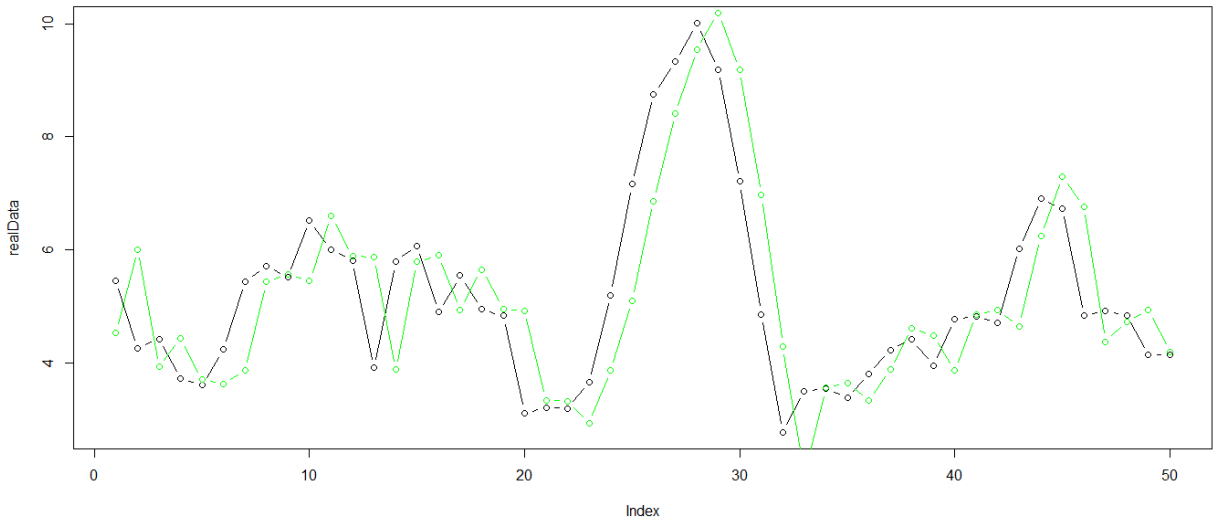


Рисунок 50. Реальная и прогнозируемая загрузка ЦП

Таблица 6. Ошибка прогноза ARIMA(p,d,q) модели

Прогнозируемый процесс	**Ошибка прогноза (MAE)**
Входящий трафик Eth0	10,56%
Исходящий трафик Eth0	9,92%
Входящий трафик Eth1	10,47%
Исходящий трафик Eth1	10,25%
Загрузка ЦП	5,94%
Объем свободной памяти	5,13%

3.4 Прогноз методом SSA («Гусеница»)

Алгоритм Singular Spectrum Analysis (SSA) или «Гусеница» - это относительно новый и мощный способ обработки временных рядов, который вобрал в себя элементы классических методов анализа, многомерной статистики, многомерной геометрии и динамических систем [98].

Метод применим для анализа любых процессов, в которых потенциально может быть обнаружена сложная структура, путем разложения исходного временного ряда на композицию небольшого числа независимых, интерпретируемых компонентов, таких как медленно изменяющиеся тренды, периодические составляющие и шумы [99].

На первом этапе при обработке ряда $X_1,...,X_N$ методом SSA выбирается окно длины L ($1<L<N$) и строится массив векторов X'_i:

$$X_i^{'} = (X_i,...,X_{i+L-1})^T, i = 1,2,...,K = N - L + 1, \quad (73)$$

из которого формируется матрица $\mathbf{X} = (X_{i+j-1})_{i,j=1}^{L,K} = [X_1^{'} : ... : X_K^{'}]$ размера $L*K$, которую часто называют матрицей траекторий. Это Ганкелева матрица, то есть на всех диагоналях, перпендикулярных главной, стоят одинаковые элементы (элементы $i+j=const$).

Столбцы X'_j матрицы $\mathbf{X}$ принимаются за вектора, лежащие в L-размерном пространстве R^L. Сингулярное разложение $\mathbf{XX^T}$ позволяет получить L сингулярных собственных чисел и векторов. Комбинация некоторого числа $l<L$ этих собственных векторов определяет l-размерное субпространство в R^L. Затем L-размерный набор данных $\{X_1,...,X_K\}$ проецируется на l-размерное субпространство с последующим усреднением по диагоналям, что дает некоторую Ганкелеву матрицу $\tilde{\mathbf{X}}$, которая является некоторым приближением матрицы $\mathbf{X}$. Ряд, полученный восстановлением $\tilde{\mathbf{X}}$ матрицы, удовлетворяет некоторому линейному рекуррентному

соотношению, которое может быть использовано для прогнозирования будущих значений [99].

Помимо прогнозирования базовый алгоритм SSA может быть использован для сглаживания данных, фильтрации, устранения шума, детрендинга, извлечения периодик. Существует ряд модификаций алгоритма SSA, например, для анализа стационарных рядов [100], Монте-Карло SSA [101], алгоритмы с модифицированной операцией сингулярного разложения и обработки многомерных рядов.

Применительно к прогнозированию трафика SSA показал лучший результат, нежели ARIMA и AR.

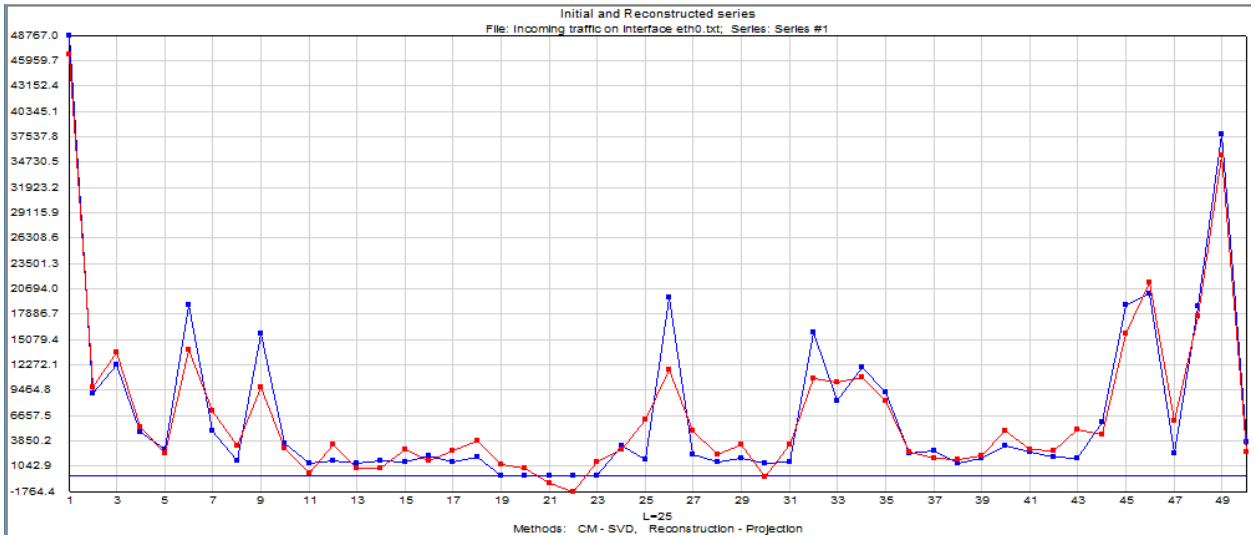


Рисунок 51. Интенсивность реального и прогнозируемого входящего трафика (Eth0)

Таблица 7. Ошибка прогноза SSA модели

Прогнозируемый процесс	**Ошибка прогноза (MAE)**
Входящий трафик Eth0	9,93%
Исходящий трафик Eth0	10,21%
Входящий трафик Eth1	10,16%
Исходящий трафик Eth1	9,98%
Загрузка ЦП	5,54%
Объем свободной памяти	4,97%

3.5 Прогноз на основе ARFIMA модели

Как известно, задача прогнозирования значений временного ряда основывается на используемой модели, описывающей процесс. Чем достовернее и точнее модель отражает процесс, тем ближе прогнозные значения к реальным данным. Ранее были приведены методы прогнозирования рядов на основе авторегрессии и скользящего среднего, а также упоминалось об открытии эффекта самоподобия, что в совокупности представляет собой модель ARFIMA (Autoregressive fractionally integrated moving average). Моделирование процессов с учетом долговременной зависимости рассмотрено в работах [102,103,104], задачи прогнозирования обширно рассмотрены в [105].

В данной работе ARFIMA модель оптимизировалась на исторической выборке на основе алгоритма Бройден-Флетчера-Гольдфарба-Шанно [106]. БФГШ использует приближение Тейлора целевой функции в окрестности d точки X [107]:

$$f(x+d) \approx q(d) = f(x) + d^T g(x) + \frac{1}{2} d^T H(x) d, \tag{74}$$

где g(x) – градиент, а H(x) – матрица Гессе [108].

После оптимизации модели производился прогноз на 1 шаг вперед, после чего модель вновь оптимизировалась, и рассчитывался новое прогнозное значение и т.д.

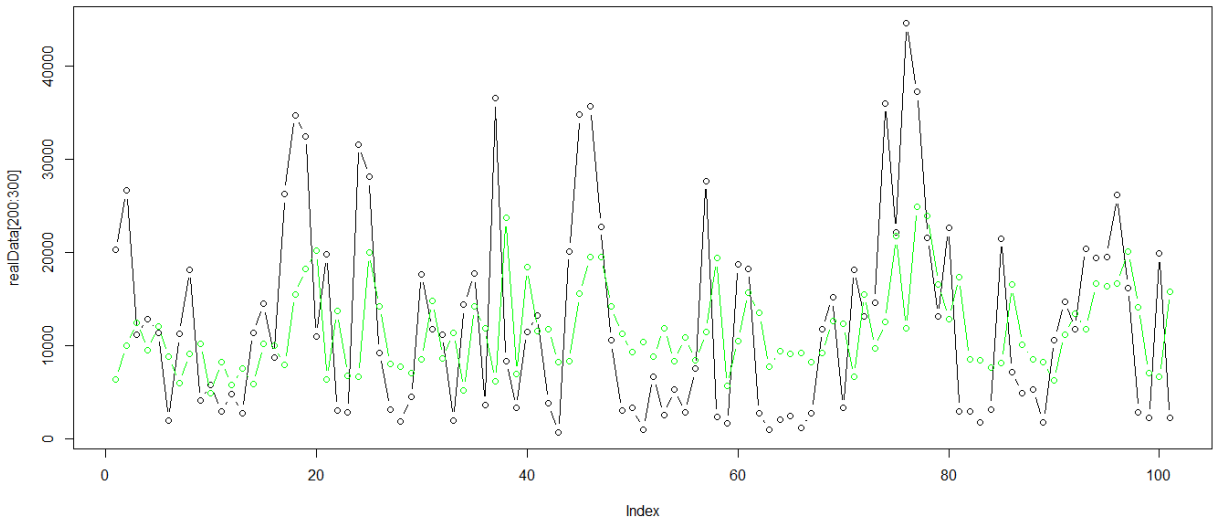


Рисунок 52. Интенсивность реального и прогнозируемого входящего трафика (Eth0)

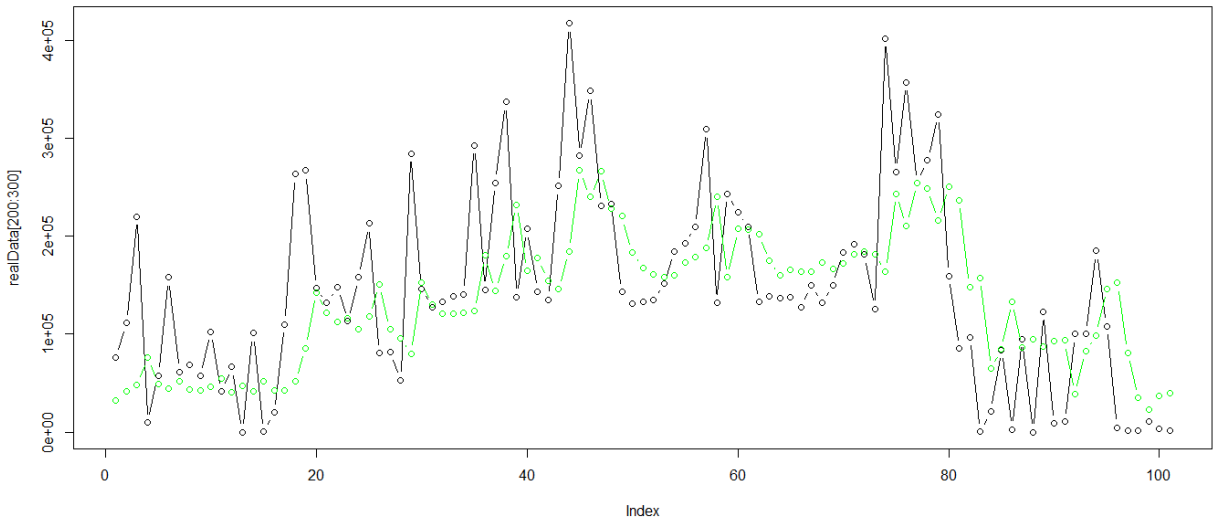


Рисунок 53. Интенсивность реального и прогнозируемого входящего трафика (Eth1)

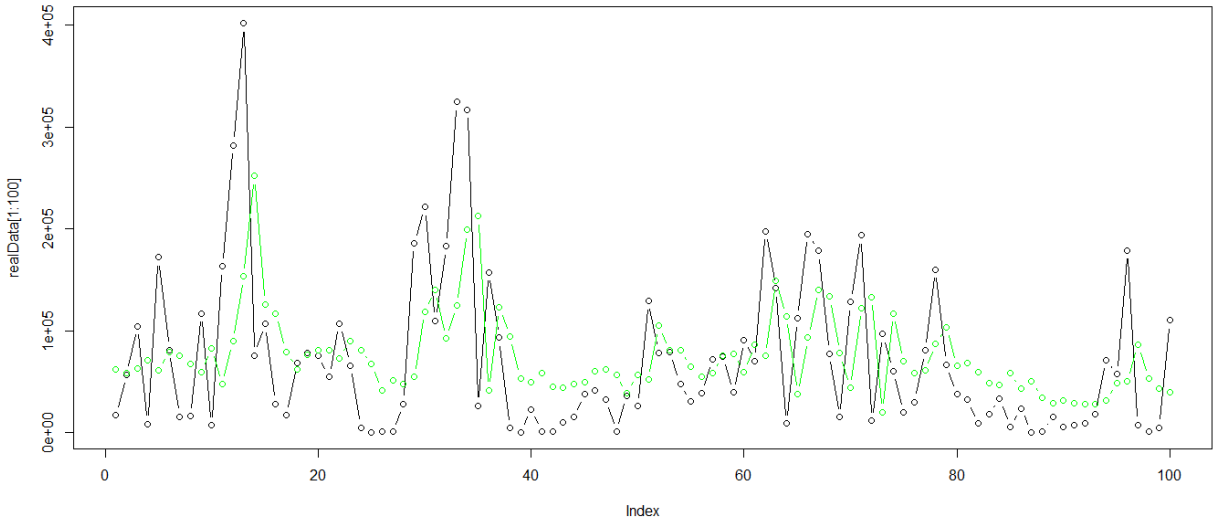


Рисунок 54. Интенсивность реального и прогнозируемого исходящего трафика (Eth0)

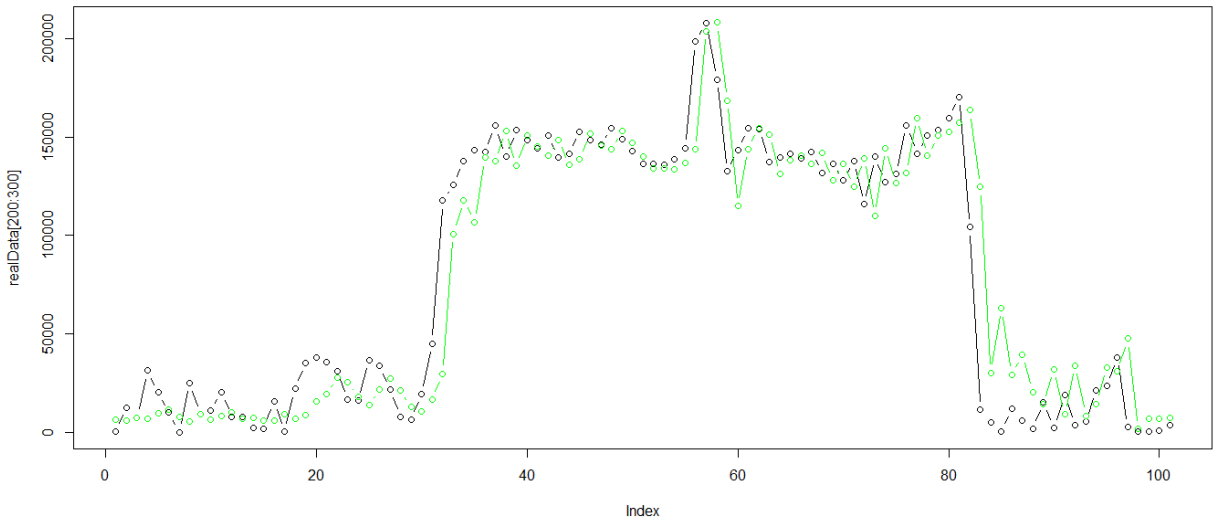


Рисунок 55. Интенсивность реального и прогнозируемого исходящего трафика (Eth1)

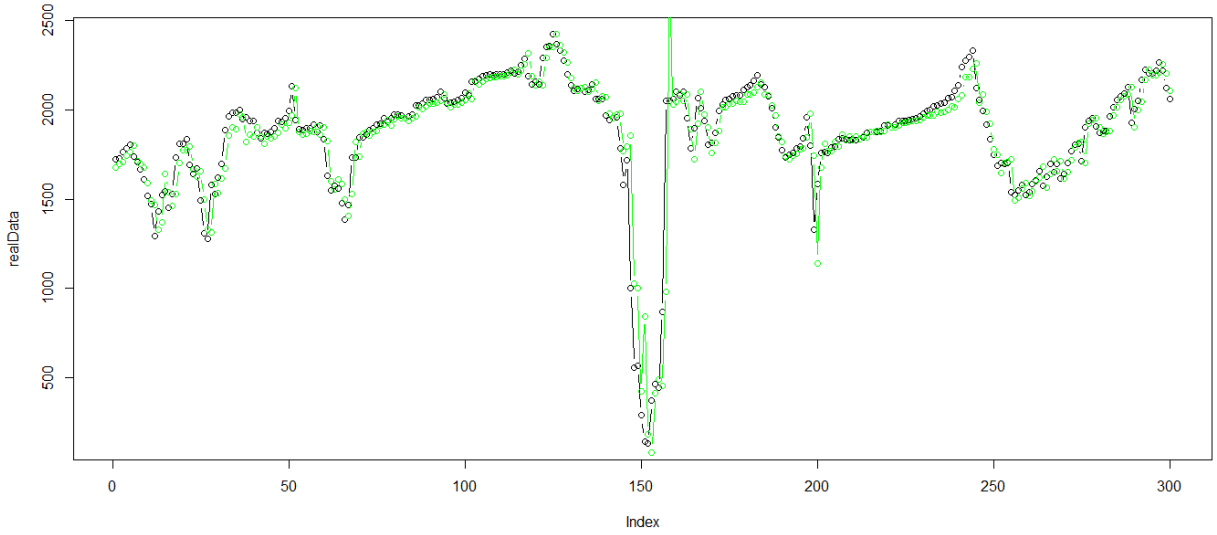


Рисунок 56. Реальный и прогнозируемый объем свободной памяти

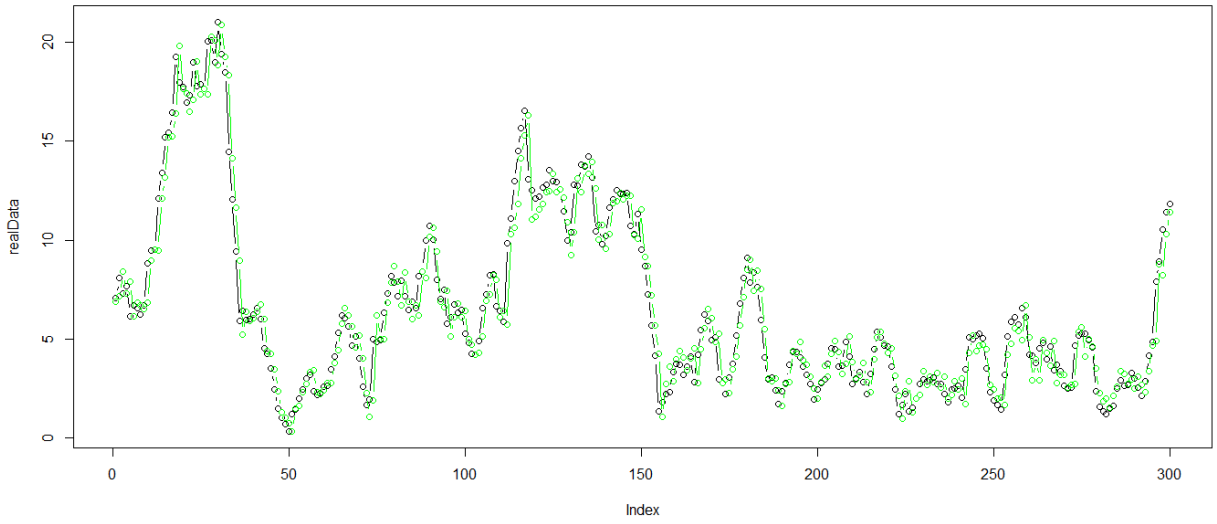


Рисунок 57. Реальная и прогнозируемая загрузка ЦП

Таблица 8. Ошибка прогноза ARFIMA(p,d,q) модели

Прогнозируемый процесс	**Ошибка прогноза (MAE)**
Входящий трафик Eth0	8,85%
Исходящий трафик Eth0	9,21%
Входящий трафик Eth1	8,46%
Исходящий трафик Eth1	8,95%
Загрузка ЦП	4,94%
Объем свободной памяти	4,57%

Выводы

1) Для выбора подходящего метода прогнозирования исследуемых процессов были рассмотрены наиболее известные модели временных рядов.

2) Была проведена оценка точности прогноза основных характеристик сервера и потока трафика (Рисунок 58).

3) Модель ARFIMA(p,d,q) превосходит остальные рассмотренные по точности прогнозирования, что согласуется со спецификой модели и самоподобным характером исследуемых процессов.

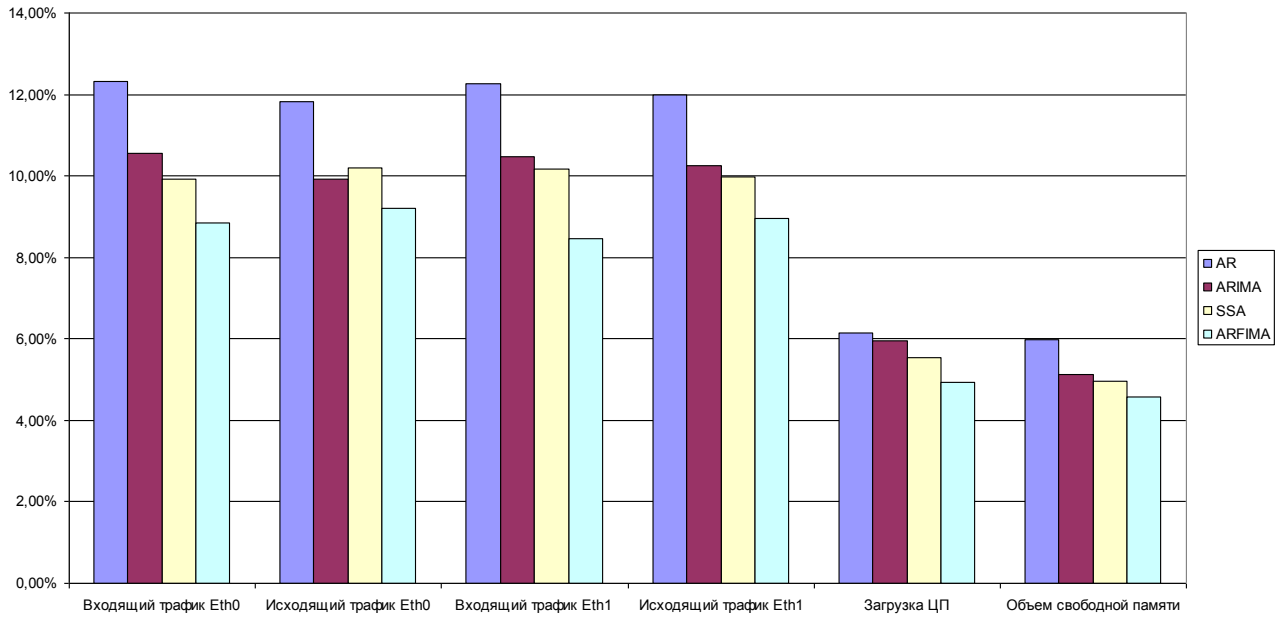


Рисунок 58. Ошибка прогнозирования для различных моделей

4 Разработка модели сети с краткосрочным прогнозированием нагрузки

В ряде случаев изучение реальной компьютерной сети и отдельных её узлов может быть затруднительно, сопряжено с большими временными и материальными затратами, а иногда и невозможно из соображений безопасности. В таком случае возникает необходимость создания компьютерной модели, отражающей реальные процессы с целью дальнейшего их исследования. Имитационное моделирование – известный способ изучения характеристик системы без проведения реального эксперимента, тем не менее, позволяющий делать выводы о свойствах протекающих процессов при условии адекватности модели. Принимая во внимание условия адекватности модели логично вести разработку последней с учетом всех известных характеристик процессов, протекающих в реальном объекте исследования.

4.1 Создание имитационной модели сети

Создание модели корпоративной сети проводилось средствами библиотеки SimEvents пакета MatLab. Принципиально модель представлена на рисунке 59. Это сеть с топологией типа «звезда» с переменным числом серверов, подключенных к коммутатору. Блок Bandwidth позволяет задать пропускную способность сети, блоки Comp1…CompN моделируют поведение серверов, то есть генерируют пакеты сообщений с переменной скоростью и разной длины. Топология «звезда» является наиболее популярной при организации сетей и соответствует сети МГТУ им. Баумана, данные с сервера которой были сняты ранее.

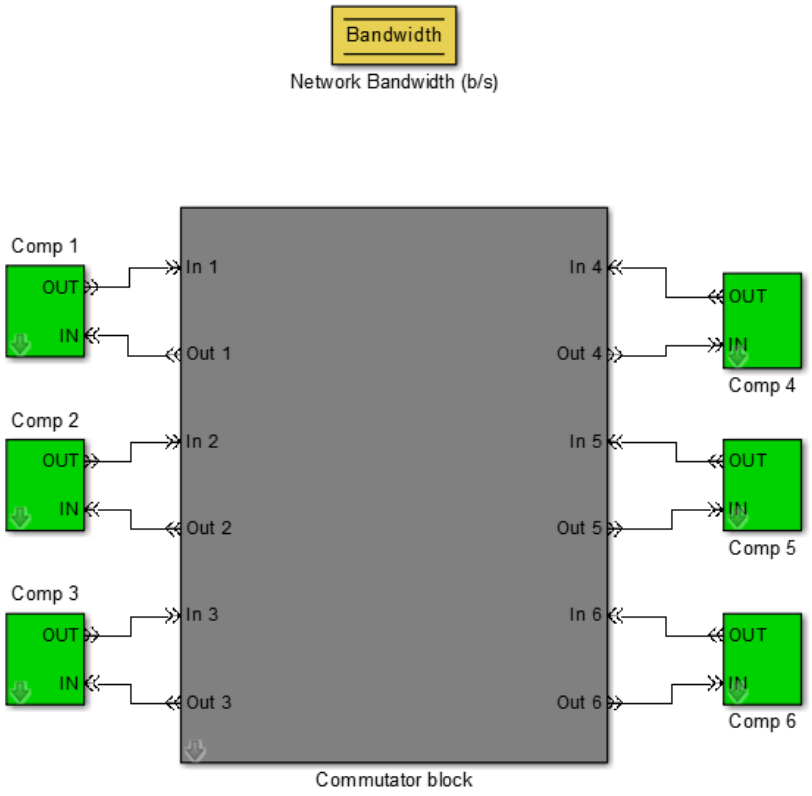


Рисунок 59. Общий вид модели

Роль источников сообщений, то есть подключенных к коммутатору серверов играет модель, представленная на рисунке 60. С заданной скоростью (пакетов/с) генерируются сообщения, длина которых лежит в определенном интервале – рисунок 61.

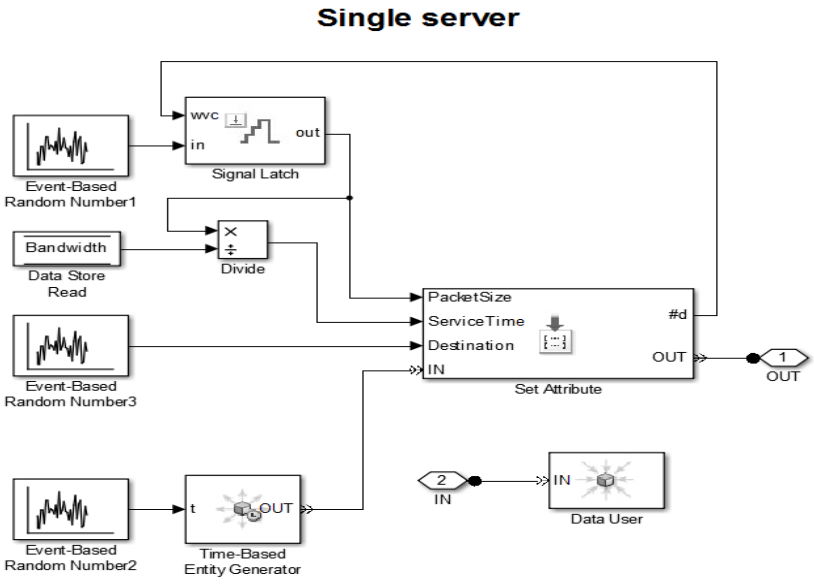


Рисунок 60. Модель сервера, источника пакетов данных

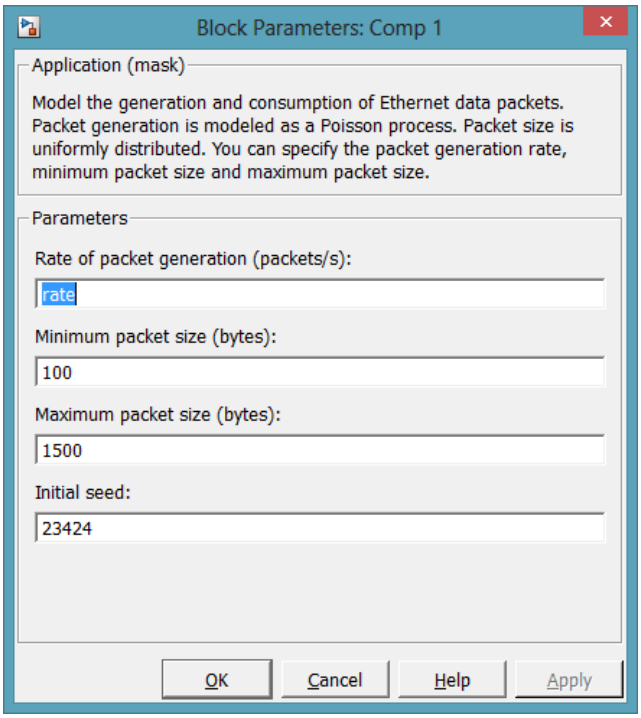


Рисунок 61. Основные характеристики источника пакетов сообщений

Каждому созданному пакету присваивается три параметра (Рисунок 62):

- PacketSize – длина сгенерированного пакета в байтах.
- ServiceTime – время обслуживания сообщения.
- Destination – адрес назначения пакета.

Подключение к модели коммутатора осуществляется через соединения IN и OUT.

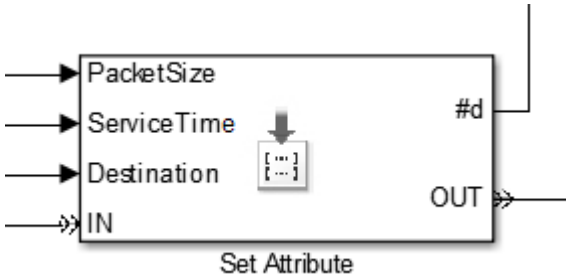


Рисунок 62. Блок Set Attribute модели источника сообщений

Модель коммутатора учитывает дисциплину обслуживания заявок FIFO с контролем переполнения буфера. Данные об объеме трафика в коммутаторе в каждый конкретный момент времени рассчитываются с

учетом длины сообщений в буфере и их количества, а затем передаются в среду MatLab для дальнейшей обработки.

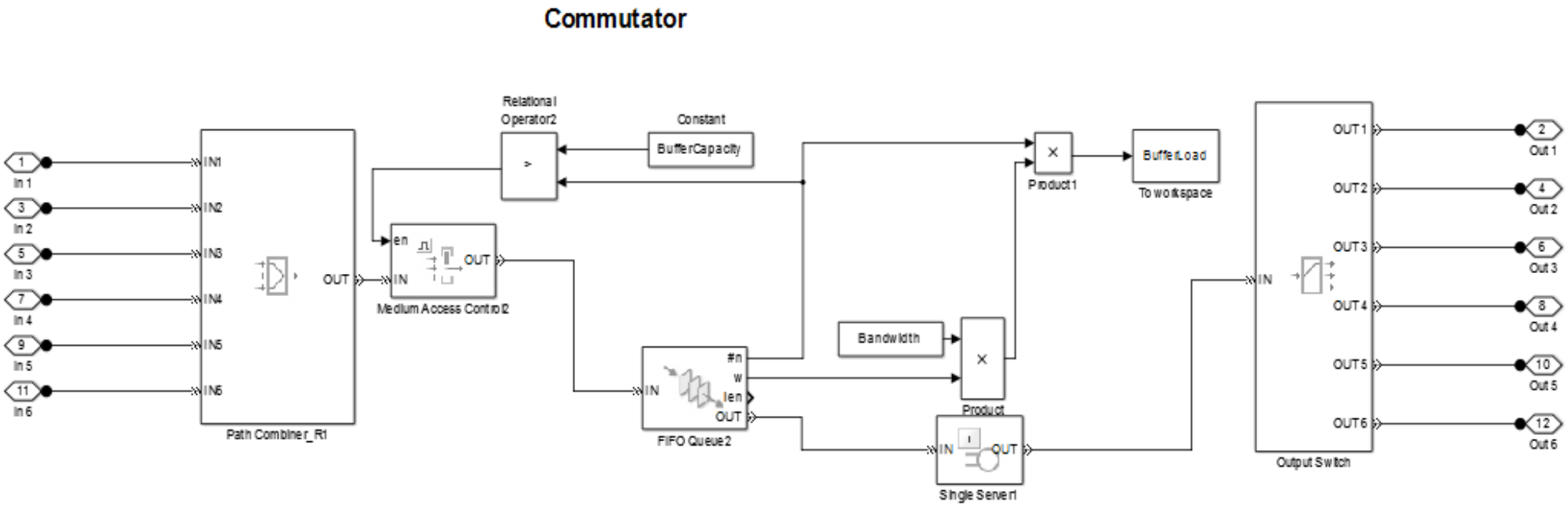


Рисунок 63. Имитационная модель коммутатора

В представленной модели основным исследуемым процессом будет загрузка буфера коммутатора и объем трафика в единицу времени. Для разработки методики борьбы с перегрузками необходима дальнейшая разработка модели, реализация алгоритмов TCP и сравнительный анализ применимости алгоритмов прогнозирования для повышения QoS.

4.2 Реализованные алгоритмы TCP обеспечения QoS

На сегодняшний день TCP – это основной протокол сети интернет, который разрабатывался с учетом ряда методик управления трафиком и предотвращению перегрузок в сети передачи данных, базируется на возможностях межсетевого IP протокола. Протокол каждому сообщению добавляет заголовок, структура которого представлена в таблице 9.

Таблица 9. Формат заголовка TCP-пакета

<table>
<tr><td colspan="2">0 3</td><td>9</td><td>15</td><td>23</td><td>31</td></tr>
<tr><td colspan="3">Порт источника</td><td colspan="3">Порт приемника</td></tr>
<tr><td colspan="6">Номер в последовательности</td></tr>
<tr><td colspan="6">Номер подтверждения</td></tr>
<tr><td>Смещение данных</td><td>Зарезервировано</td><td>URG;ACK;PSH; RST;SYN;FIN</td><td colspan="3">Размер окна</td></tr>
<tr><td colspan="3">Контрольная сумма</td><td colspan="3">Указатель</td></tr>
<tr><td colspan="5">Дополнительные данные заголовка</td><td>Данные выравнивания</td></tr>
</table>

Добиться реализации всех алгоритмов передачи протокола TCP – нетривиальная техническая задача. Поэтому на данном этапе ограничимся наиболее важными методиками TCP по управлению перегрузками. Реализуем динамический контроль над размером окна передачи. В TCP используется принцип «скользящего окна», заключающийся в том, что каждая сторона передачи может отправлять максимум столько байт, сколько было указано в поле «размер окна» заголовка пакета, подтверждающего получение предыдущего блока банных. Принцип «скользящего окна» обеспечивает опережающую подтверждение посылку данных [109]. Для управления окном

передачи реализуем алгоритм медленного старта, заключающийся в согласовании интенсивности передачи сообщений источником на основе размера буфера приемника. При этом размер окна передачи повышается постепенно до возникновения повторных передач.

Реализуем также таймер повторной передачи. Таймер отсчитывает период до повторной передачи пакета, подтверждение которого не пришло, обозначается как RTO (Retransmission Timeout) и рассчитывается динамически на основе временного интервала от момента посылки сообщения до получения подтверждения - RTT (Round Trip Time):

$$SRTT = k * SRTT + (1 - k) * RTT, \quad (75)$$

где SRTT (Smoothed RTT) – сглаженное значение RTT, k – сглаживающий коэффициент. Тогда RTO вычисляется согласно следующему выражению:

$$RTO = \min(U, \max(L, p * SRTT)), \quad (76)$$

где U – ограничение сверху на значение RTO, L – ограничение снизу на значение RTO, p – некоторый коэффициент. Если после уменьшения таймера до нуля и повторной передачи, подтверждение не приходит вновь, то RTO увеличивается и передача возобновляется.

4.3 Реализация модели сети с коммутацией пакетов с учетом алгоритмов TCP

Для реализации обозначенных выше алгоритмов необходимо внести ряд изменений в разработанную ранее модель. В первую очередь – добавить несколько дополнительных полей свойств в каждый генерируемый пакет (рисунок 73). Поле Ack имитирует служебный бит подтверждения заголовка TCP пакета. Поле CreationTime хранит время создания пакета для удобства расчета RTT и, соответственно, таймера повторной передачи. SequenceNumber служит для определения очередности пакетов. AcknowledgmentNumber используется приемником пакетов для того, чтобы указать отправителю номер следующего ожидаемого пакета. Состояние

таймеров содержится в блоках Data Store. После введения базовых алгоритмов TCP, число источников пакетов было увеличено, чтобы оставалась возможность моделировать перегрузки в сети, при этом оставляя должную адекватность моделей абонентов.

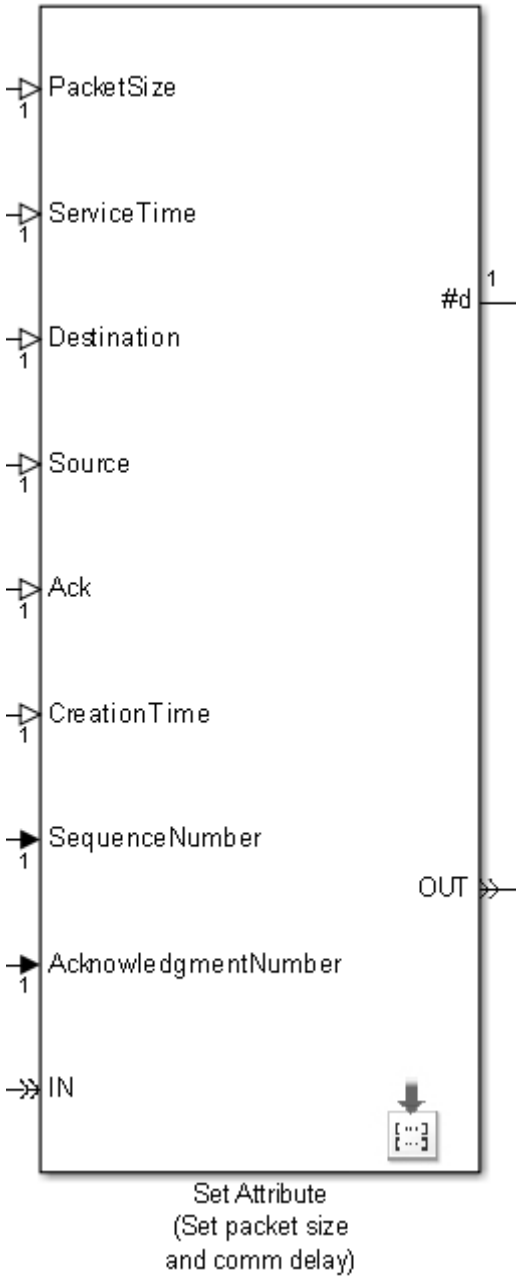


Рисунок 64. Блок присвоения параметров пакету для дальнейшей передачи

Через хранимые в блоках Data Store значения проводится расчет RTO и повторная отправка непринятых пакетов (рисунок 65).

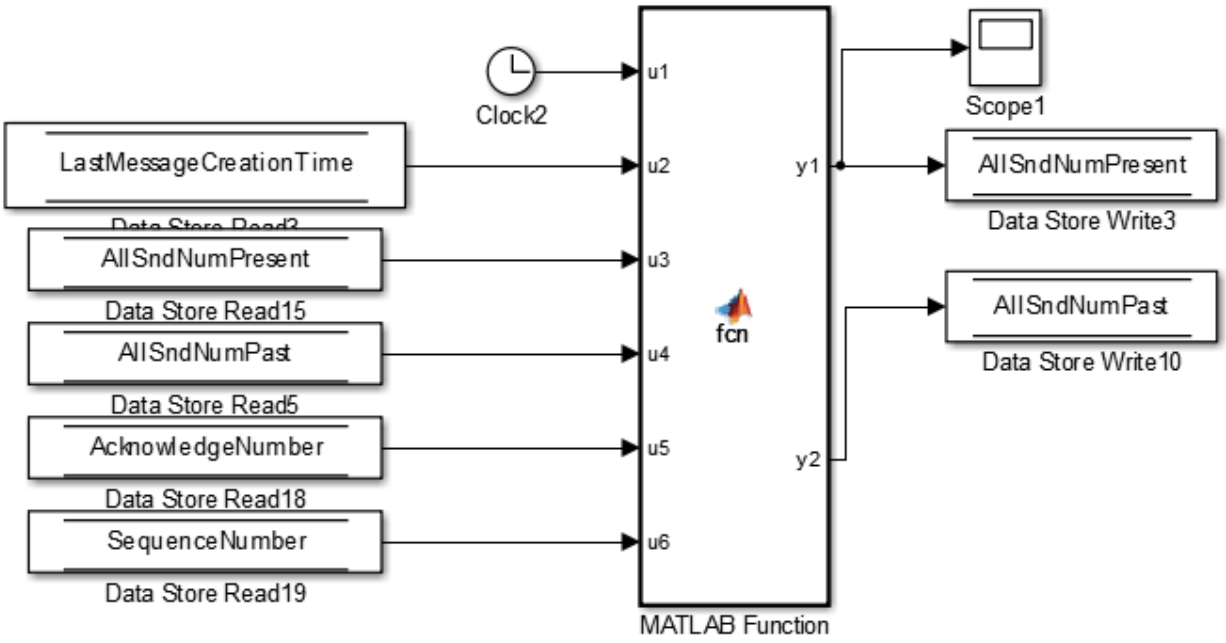
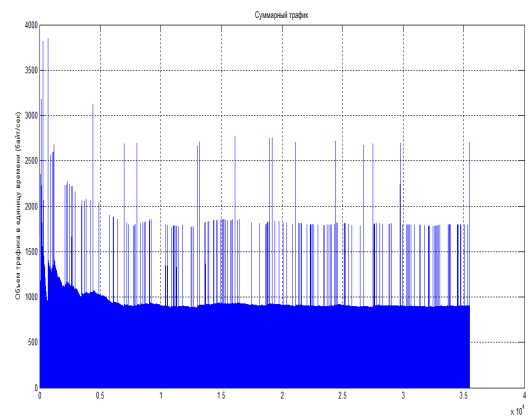


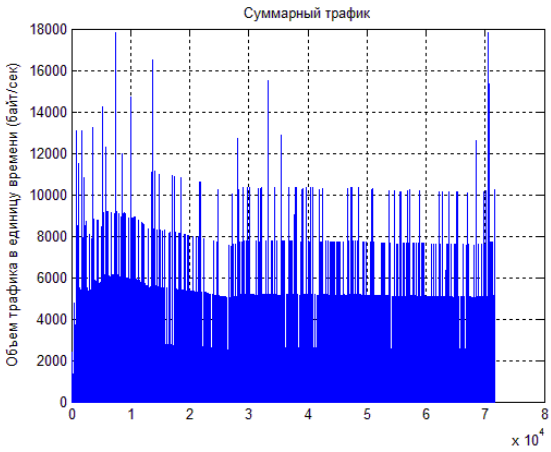
Рисунок 65. Блок расчета значений таймера RTO

4.4 Расчетные характеристики полученной модели

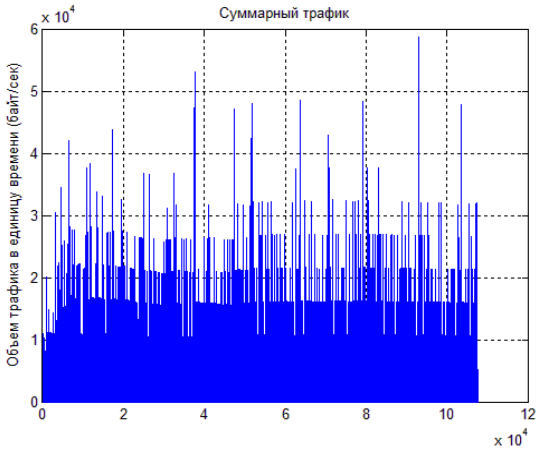
Для первичной оценки работоспособности модели было проведено моделирование функционирования сети в течение 30 секунд с переменной интенсивностью трафика. Для каждого источника сообщений установлены одинаковые параметры, то есть длина пакетов в диапазоне, соответствующем протоколу TCP/IP от 100 до 1500 байт и скорость генерации пакетов в 50, 100, 150 и 200 (пакетов/с) соответственно для рисунков 66а, 66б, 66в, 66г. На рисунке 67 представлена зависимость между скоростью генерирования пакетов и суммарным трафиком, проходящим через коммутатор. Вид графика объясняется конечным объемом буфера коммутатора, что ограничивает пропускную способность.



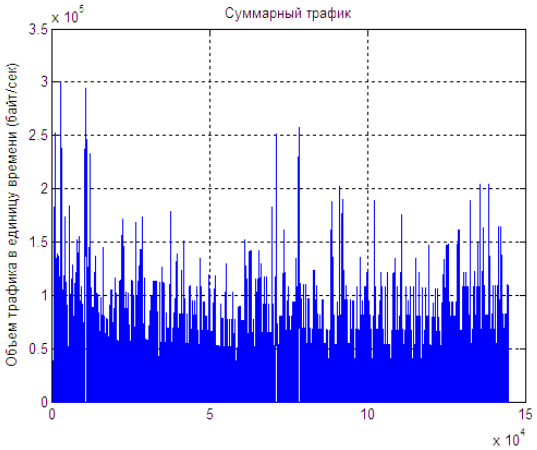
а



б



в



г

Рисунок 66. Суммарный трафик, проходящий через коммутатор в единицу времени при скорости генерации сообщений узлами в 50 пакетов в сек. (*а*), 100 пакетов в сек. (*б*), 150 пакетов в сек. (*в*), 200 пакетов в сек. (*г*)

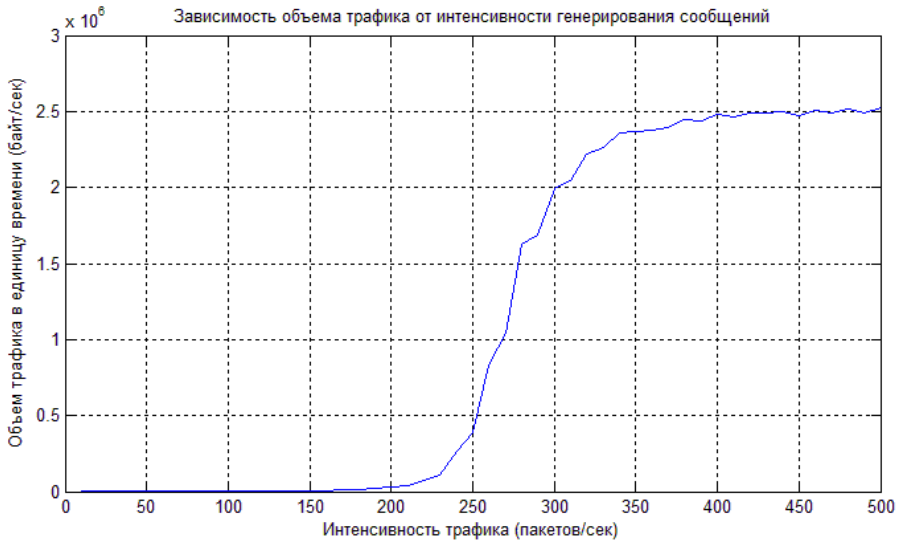


Рисунок 67. Зависимость интенсивности трафика от скорости генерирования сообщений.

В пользу медленно убывающей зависимости в процессе формирования трафика говорит медленно затухающий характер АКФ – рисунок 68.

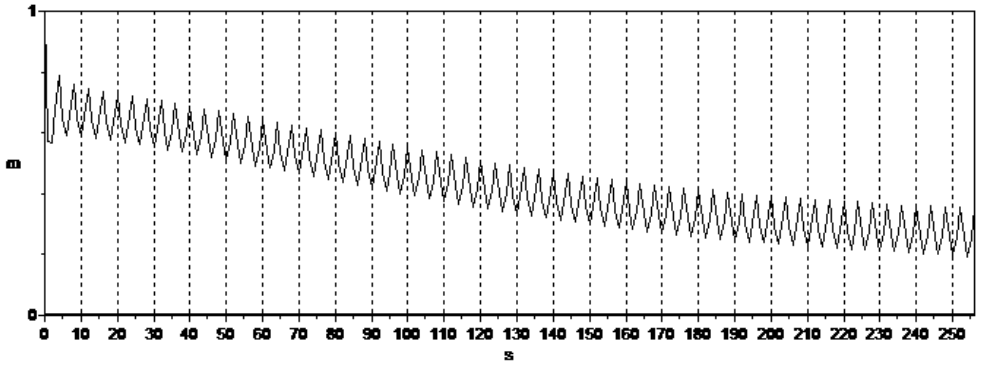


Рисунок 68. АКФ трафика

На рисунке 69 представлен график изменения показателя Херста от интенсивности трафика, результаты согласуются с исследованиями реального трафика корпоративной сети и указывают на прямую зависимость. При увеличении интенсивности трафика показатель Херста растет экспоненциально.

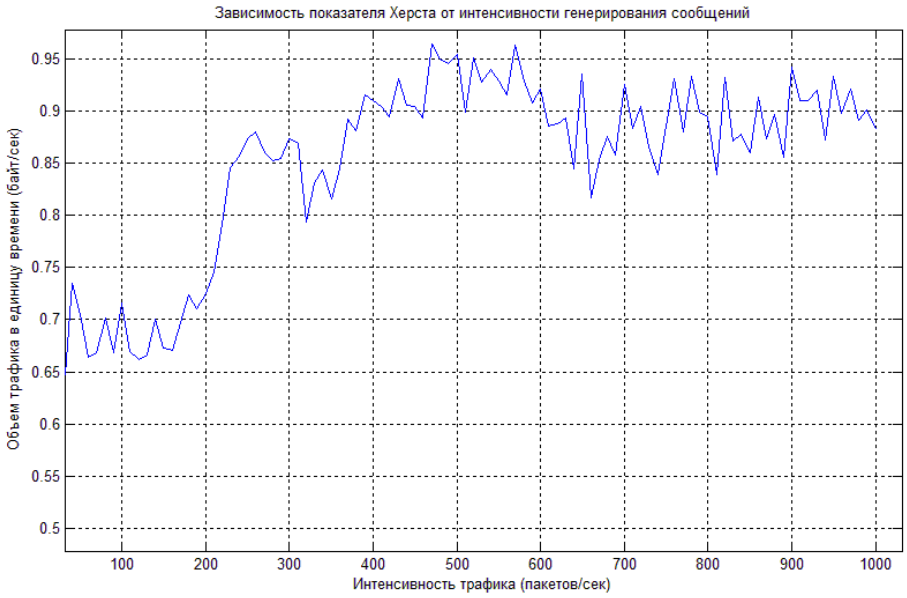


Рисунок 69. График зависимости показателя Херста от интенсивности трафика

Разработанная имитационная модель позволяет задавать необходимые параметры источников сообщений и коммутатора, таким образом, становится возможным достичь большего соответствия между реальной корпоративной сетью и моделью. В дальнейших расчетах использовались данные о состоянии сети МГТУ им. Баумана, средний размер пакета и число пакетов в секунду, указанные в работе [58] (см. таблица 1).

По результатам моделирования функция автокорреляции агрегированного трафика представлена на рисунке 70. Медленное затухание говорит о наличии долгосрочной зависимости.

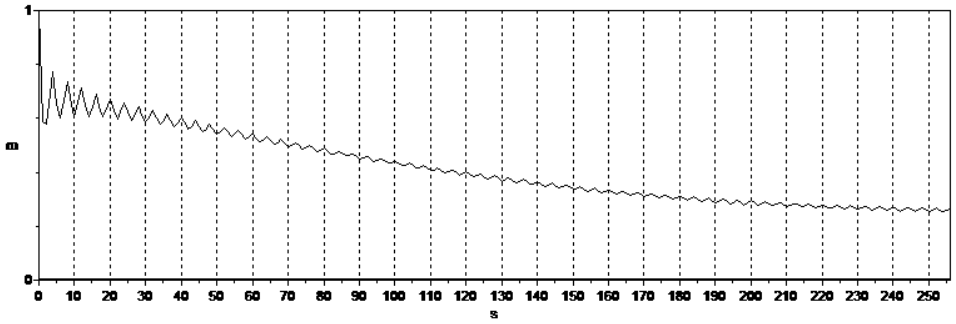


Рисунок 70. АКФ для модели с параметрами реальной сети

Для оценки самоподобия процесса был вычислен показатель Херста, который составил H=0.86, что говорит о самоподобии полученных данных.

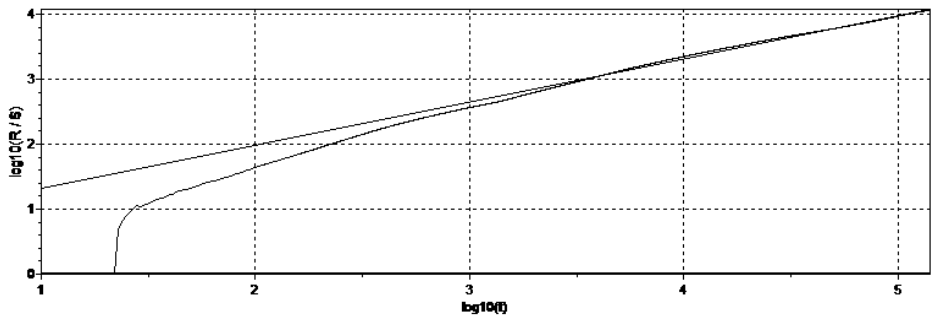


Рисунок 71. АКФ для модели с параметрами реальной сети

4.5 Оценка эффективности модели с использованием алгоритма прогнозирования

Ранее нами были рассмотрены различные алгоритмы прогнозирования рядов и выполнен сравнительный анализ применимости некоторых математических моделей к задаче прогнозирования исследуемых процессов. На основе реальных экспериментальных данных был сделан вывод об эффективности модели ARFIMA. Теперь необходимо определить, позволяет ли использование методики краткосрочного прогнозирования объема трафика достигнуть повышения полезной пропускной способности сети.

Как указывалось выше, полное моделирование сети передачи сообщений по протоколу TCP – весьма нетривиальная техническая задача, выходящая за рамки текущей работы, а разработка TCP с прогнозированием заслуживает отдельных глубоких исследований. На данном этапе ограничимся выводом о применимости ARFIMA в сетях с коммутацией пакетов и реализуем один из возможных алгоритмов управления перегрузками.

Добавим в разработанную модель обратную связь между коммутирующим устройством и источниками сообщений для управления размером окна передачи. Добавим в модель коммутатора функцию подсчета

числа уникальных источников пакетов. В результате используем следующее выражение для расчета окна передачи Win_i в момент времени i ($0<i<T$), T – полное время моделирования:

$$Win_i = Min[X_i/N, CW], i=0,1,\ldots,T, \quad (77)$$

где N – число уникальных источников пакетов, X_i – прогнозируемый доступный свободный объем буфера коммутатора, CW – размер окна передачи, рассчитанный согласно алгоритму медленного старта.

Теперь смоделируем работу сети при неизбежном возникновении перегрузки, оценим потери пакетов при передаче и полезную пропускную способность.

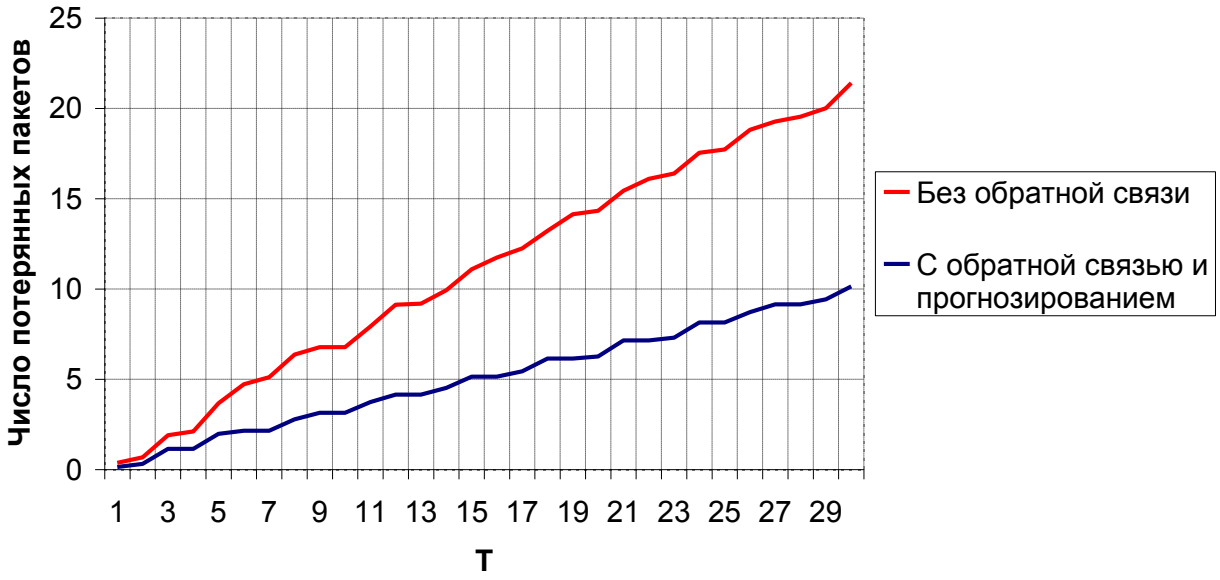


Рисунок 72. Зависимость потерь пакетов от времени

Применение механизма обратной связи, сообщающего источникам сообщений о загруженности буфера, позволяет снизить потери пакетов в среднем от 9% до 12%, результат варьируется в зависимости от параметров источника. Полезная пропускная способность рассчитывалась как отношение максимальной заданной пропускной способности (параметр модели, определяющий время передачи сообщения) к текущей, реальной пропускной способности. Применение обратной связи между коммутатором и источником сообщений с учетом прогнозирования позволило повысить

полезную пропускную способность на величину от 12% до 17%, при этом значение также зависит от параметров источников сообщений (Рисунок 73).

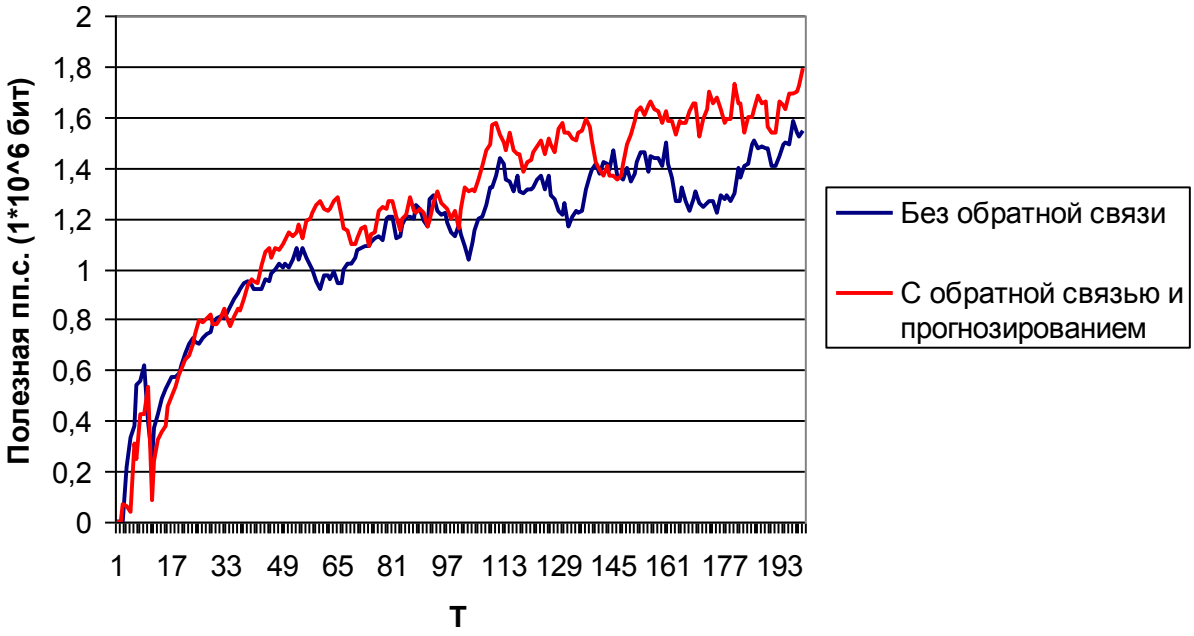


Рисунок 73. Зависимость полезной пропускной способности сети от времени

Выводы

1) Для оценки применимости алгоритмов прогнозирования в задаче управления перегрузками была разработана модель компьютерной сети с коммутацией пакетов.

2) Реализованы основные алгоритмы обеспечения QoS, заложенные в TCP.

3) Адекватность модели подтверждена соответствием её основных статистических характеристик реальному эксперименту. Результаты расчета нелинейно – динамических свойств смоделированного трафика также соответствуют реальному процессу передачи данных.

4) На уровне коммутирующего устройства реализована функция прогнозирования загруженности буфера и оповещения источников с последующим управлением окном передачи.

5) Сделан вывод о применимости методов прогнозирования в задаче управления перегрузками сети и приведена количественная оценка эффективности алгоритма.

Заключение

Основные результаты диссертационной работы следующие:

1) Выполнен сбор статистических данных входящего и исходящего трафика, а также процесса распределения аппаратных ресурсов одного из физических серверов корпоративной сети МГТУ им. Н. Э. Баумана; проведен статистический и динамический анализ собранных данных, выявлен самоподобный и хаотический характер процессов.

2) Определена качественная и количественная зависимость между суммарным объемом трафика, проходящего через сервер в сети и распределением аппаратных ресурсов узла; проведен корреляционный и регрессионный анализ временных рядов.

3) Разработана и протестирована имитационная модель сети в целом и отдельных процессов передачи данных, установлена адекватность модели; получена модифицированная имитационная модель компьютерной сети с учетом кратковременного прогнозирования (ARFIMA) загрузки буфера коммутатора.

4) Вычислительный эксперимент показал, что предложенная методика на основе механизма обратной связи, сообщающего источникам сообщений о загруженности буфера, позволяет снизить потери пакетов в среднем от 9% до 12%; применение обратной связи между коммутатором и источником сообщений с учетом прогнозирования позволяет повысить полезную пропускную способность на величину от 12% до 17% в зависимости от параметров источников сообщений.

Полученные результаты могут быть использованы при решении практических задач, возникающих при исследовании функционирования различных типов сетей передачи данных, а также для улучшения их технико-экономических и эксплуатационных характеристик. Разработанные модели и алгоритмы в перспективе могут быть распространены на решение задач

обнаружения аномалий и угроз в компьютерных сетях, в том числе в комбинации с методами сигнатурного анализа [8].

Список источников

1.Sandvine Global Internet Phenomena Report, 1H 2014 [Электронный ресурс]. - 2014. - Режим доступа: https://www.sandvine.com/downloads/general/global-internet-phenomena/2014/1h-2014-global-internet-phenomena-report.pdf.

2.Семенов, Ю.А. Протоколы Internet для электронной торговли [Электронный ресурс]. - ИТЭФ МФТИ. -2013. – Режим доступа: http://book.itep.ru/4/44/qos_lan.htm.

3.Кучерявый, Е.А. Управление трафиком и качество обслуживания в сети Интернет. - СПб.: Наука и Техника. - 2004.

4.Столлингс, В. Современные компьютерные сети. - СПб.: Питер. - 2003.

5.Land, W. On the self-similar nature of Ethernet traffic / W.Land, M.Taqqu, W.Willinger // IEEE/ACM Transactions on Networking. - 1994.

6.Crovella, M. Self-similarity in World Wide Web traffic: evidence and possible causes / M. Crovella, A. Bestavros // IEEE/ACM Transactions on Networking. - 1997.

7.Шелухин, О.И. Самоподобие и фракталы / О.И. Шелухин, А.В. Осин, С.М. Смольский. – ФИЗМАЛИТ, 2008.

8.Шелухин, О.И. Обнаружение вторжений в компьютерные сети (сетевые аномалии)/ О.И. Шелухин, Д.Ж. Сакалема, А.С Филинова.- М.: Горячая линия – Телеком, 2013.

9.Шелухин, О.И. Мультифракталы. Инфокоммуникационные приложения / О.И. Шелухин. - Горячая Линия-Телеком. - 2011.

10.Бойченко, М. К. Исследование характера трафика в магистральных сегментах ЛВС МГТУ им. Н. Э. Баумана / М. К. Бойченко, И. П. Иванов // М.: Вестник МГТУ. Приборостроение. - N 3. - 2009.

11.Иванов, И. П. Интегральная оценка состояния ресурсов пользовательского маршрута в корпоративной сети / И. П. Иванов // М.: Вестник МГТУ. Приборостроение. - N 2. -2010.

12.Иванов, И. П. Система адаптивного управления трафиком / И.П. Иванов, Л.И. Колобаев, В.А. Лохтуров // М.: Вестник МГТУ. Приборостроение. – N 2. - 2005.

13.Фомин, В.В. Статистический анализ IP и VoIP трафика / В.В. Фомин // Инфокоммуникационные технологии. - 2009.

14.Fishman, G.S. Principles of Discrete Event Simulation / G.S. Fishman. - 1978.

15.Frost, V. Traffic modeling for telecommunications networks / V. Frost, B. Melamed //IEEE Communications Magazine. - 1994.

16.Abdelnaser, A. Traffic Models in Broadband Telecommunication Networks / A. Abdelnaser // Department of Electrical Engineering. - 1996.

17.Chen, T.M. The Handbook of Computer Networks / T.M. Chen // Southern Methodist University. - 2007.

18.Discrete Stochastic Processes: MIT Open Course. - 2011.

19.Brandauer, C. Comparison of Tail Drop and Active Queue Management Performance for built-data and Web-like Internet Traffic / C. Brandauer, C. Diot. - 2001.

20.Пуассоновский процесс[Электронный ресурс]. – Режим доступа: http://www.intuit.ru/studies/courses/666/522/lecture/6754?page=2.

21.Клейнрок, Л. Теория массового обслуживания. Пер. с англ./ Л. Клейнрок. - М.: Машиностроение, 1979. - 432 с.

22.Chen, T. Network Traffic Modelling / T. M. Chen // Wiley. - 2007.

23.Hefles, H. A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance / H. Hefles, D. Lucantoni // IEEE Journal on Selected Areas in Communications. - 1986.

24.Abdelnaser, A. Traffic Models in Broadband Networks / A. Abdelnaser // IEEE Communications Magazine. - 1997.

25.Shim, C. Modeling and call admission control algorithm of variable bit rate video in ATM networks / C. Shim, I. Ryoo, J. Lee, S. Lee // IEEE journal on Selected Areas in Communications. - N 3. - 1993.

26.Reichl, P. A Generalized TES Model for Periodical Traffic / P. Reichl // IEEE International Conference. – 1998.

27.Samorodnitsky, G. Stable non-Gaussian random processes / G. Samorodnitsky and M.Taqqu // Chapman and Hall. - 1994.

28.Karlin, S. A first course in stochastic processes / S. Karlin, H. Taylor // Academic Press. - 1975.

29.Beran, J. Statistics for long-memory processes / J. Beran // Chapman and Hall. - 1994.

30.Fraleight, C. Packet-level traffic measurements from the Sprint IP backbone / C. Fraleight, S. Moon // IEEE Network. - 2003.

31.Meiss, M. On the lack of typical behavior in the global Web traffic network / M. Meiss, F. Menczer // 14th International World Wide Web Conf. - 2005.

32.Li, M. Characteristics of streaming media stored on the Web / M. Li, M. Claypool // ACM Transactions on Internet Technology. - 2005.

33.Choi, H-K. A behavioral model of Web traffic / H-K. Choi, J. Limb // In proc. of 7th International Conf. on Network Protocols (ICNP’99). - 1999.

34.Meiss, M. On the lack of typical behavior in the global Web traffic network / M. Meiss, F. Menczer, A. Vespignani // In proc. of 14th International World Wide Web Conf. - 2005.

35.Saroiu, S. An analysis of Internet content delivery systems / S. Saroiu, K. Gummadi, R. Dunn, S. Gribble, H. Levy // ACM SIGOPS Operating Systems Review. - 2002.

36.Sen, S. Analyzing peer-to-peer traffic across large networks / S. Sen, J. Wang // IEEE/ACM Transactions on Networking. - 2004.

37.Dai, M. Analysis and modeling of MPEG-4 and H.264 multilayer video traffic / M. Dai, D. Loguinov // In proc. of IEEE Infocom 2005. - 2005.

38.Li, M. Characteristics of streaming media stored on the Web / M. Li, M. Claypool, R. Kinicki, J. Nichols // ACM Transactions on Internet Technology. - 2005.

39.Таненбаум, Э. Компьютерные сети. - СПб.: Питер. - 2003.

40.Reichl, P. How to Model Complex Periodic Traffic with TES. / P. Reichl, M. Schuba, S. Hoff // UKPEW Ilkley. - West Yorkshire. - 1997.

41.Cui-Qing, Y. A Taxonomy for Congestion Control Algorithms in Packet Switching Networks / Y. Cui-Qing, Alapati V.S. Reddy // IEEE Network. - 1995.

42.Jaffrey, M. Bottleneck Flow Control / M. Jaffrey // IEEE Transactions On Communications. - N 7. - 1981.

43.Frank, K. Charging and rate control for elastic traffic / K. Frank // University of Cambridge. - 1998.

44.Abhay, K. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks / K. Abhay, Parekh, G. Robert Gallager // The Single-Node Case. - 1994.

45.Network Working Group. - Request for Comments: 2309. - 1998.

46.Sally, F. Random Early Detection Gateways for Congestion Avoidance / F. Sally, V. Jacobson // Lawrence Berkeley Laboratory. - 1993.

47.Feng, W. A self-Configuring RED Gateway / W. Feng, D.D. Kandlur, D. Saha. - 2000.

48.Lambadaris, I. Empirical Study of Buffer Management Scheme for DiffServ Assumed Forwading / I. Lambadaris, R. Makkar // PHB. - 2000.

49.Toica, I. Packet marking for Web traffic in networks with RIO routers / I. Toica, M. Mellia // Globecom. - 2001.

50. Flow RED (FRED): adapting RED for UDP traffic [Электронный ресурс]. – Режим доступа: http://www.mathcs.emory.edu/~cheung/Courses/558/Syllabus/08-RED/FRED.html.

51.Lakshman, T. SRED: Stabilized RED / T.V. Lakshman, T.J. Ott. - 1999.

52.Network Working Group, A single Rate Three Color Marker. - RFC 2697. - 1999.

53.Олифер, В.Г. Компьютерные сети / В.Г. Олифер, Н.А. Олифер. - СпБ.: Питер, 2010.

54.Network Working Group, A single Rate Three Color Marker. - RFC 793. - 1981.

55.Jacobson, V. Berkeley TCP Evolution from Tahoe to Reno / V. Jacobson // Proceedings of the Eighteen Internet Engineering Task Force. - 1990.

56.Karn, P. Improving Round-Trip Extimates in Reliable Transport Protocol / P. Karn. - 1991.

57.Basarab, M.A. University corporative network traffic analysis on the base of nonlinear dynamics methods / M.A. Basarab, I.P. Ivanov, A.V. Kolesnikov. // Science and Education. Electronic scientific and technical periodical. - N. 08. - DOI: 10.7463/0813.0587054. - 2013

58.Иванов, И.П. Математические модели, методы анализа и управления в корпоративных сетях: дис. … д-ра. техн. наук: 05.13.15 / Иванов Игорь Потапович. – М., 2010. – 249 с.

59.User manual Zabbix [Электронный ресурс]. – Режим доступа: http://www.zabbix.com/en/documentation.php.

60.Официальный сайт StatSoft [Электронный ресурс]. – Режим доступа: http://www.statsoft.ru.

61.Программа для вычисления корреляционной размерности и корреляционной энтропии по временному ряду данных [Электронный ресурс]. – Режим доступа: http://www.iki.rssi.ru/magbase/RESULT/APPENDIX/fractan.boom.ru/soft.htm

62.Коннова, Н. С. Цифровая обработка сигналов допплеровского датчика объемной скорости кровотока в условиях переходных процессов в

микроциркуляторном русле / Н.С. Коннова // Наука и образование. Электронное научно-техническое издание. - 2012.

63.Официальный сайт Matlab и Simulink [Электронный ресурс]. – Режим доступа: http://www.mathworks.com.

64.Елисеева, И.И. Эконометрика: учебник / И.И. Елисеева. - М.: Финансы и статистика, 2002.

65.Бельков, Д.В. Статистический анализ сетевого трафика / Д.В. Бельков, Е.Н. Едемская // Донецкий национальный технический университет. - 2011.

66.Бекман, И.Н. Метод частотного зондирования в методе проницаемости / И.Н. Бекман // МГУ им. Ломоносова. - 2008.

67.Land, W. On the self-similar nature of Ethernet traffic / W.Land, M.Taqqu, W.Willinger // IEEE/ACM Transactions on Networking. - 1994.

68.Bong, R.K. Modeling, analysis, and simulation of self-similar traffic using the fractal-shot-noise-driven poisson process / K. Bong Ryu, Steven B. Lowen. - 1995.

69.Hurst, H. Long-Term Storage: An Experimental Study / H. Hurst, R. Black // London: Constable. - 1965.

70.Hurst, H. E. Long-Term Storage Capacity of Reservoirs. Transactions of the American Society of Civil Engineering / H. E. Hurst. - N 116. - 1951.

71.Kirillov, D.S. Distribution of the Hurst Exponent of a Nonstationary Marked Time Series / D.S. Kirillov, O.V. Korob, N.A. Mitin, Yu.N. Orlov, R.V. Pleshakov // Keldysh Institute of Applied Mathematics. Preprints. - 2013. – N 11.

72.Теория телетрафика: учебное пособие. - Ульяновский Государственный Технический Университет. - 2006.

73.Cano, J.C. On the use calculation of the Hurst parameter with MPEG videos data traffic / J.C. Cano, P. Manzoni // Valencia. - 2000.

74.Schroeder, M. Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise. / M. Schroeder // NY: W.H. Freeman and Co. - 1991.

75.Rosenstein, M.T. A Practical Method for Calculating Largest Lyapunov Exponents from Small Data Sets / M.T. Rosenstein, J.J. Collins, C.J. De Luca // Physica D. - 1993.

76.Grassberger, P. Estimation of the Kolmogorov entropy from a chaotic signal / P. Grassberger, I. Procaccia // Phys. Rev. - 1983.

77.Сычев, В.В. Вычисление стохастических характеристик физиологических данных / В.В. Сычев // Пущино. - 1999.

78.Балгабекова, Л.О. Исследование корреляционной энтропии сетевого трафика / Л.О. Балгабекова // Технические науки /Электротехника и радиоэлектроника. - N 6.

79.Chun-Lin, L. A tutorial to the wavelet transforms / L. Chun-Lin. - 2010.

80.Бокс, Д. Анализ временных рядов, прогноз и управление / Д. Бокс, Г.М. Дженкинс. - М.: Мир, 1974.

81.Nozian, M. Short Term Load Forecasting Using Double Seasonal ARIMA model / M. Nozian, H.A. Maiza, I. Zuhaimy // Regional Conference on Statistical Sciences. - 2010.

82.Игнатенко, Е.Г. Методика краткосрочного прогнозирования трафика телекоммуникационных сетей / Е.Г. Игнатенко, И.В. Дегтяренко. - 2011.

83.Гребенников, А.В. Моделирование сетевого трафика и прогнозирование с помощью модели ARIMA / А.В. Гребенников, Ю.А. Крюков. - 2011.

84.Yang, J. Power System Short-term Load Forecasting: Thesis for Ph.d degree / Jingfei Yang M. Sc. - 2006.

85.John, T. Sales Forecasting Management: A Demand Management Approach / T. John, A. Moon Mark. - 2004.

86.Ocker, D. Stationary and Non-stationary FARIMA Models – Model Choice, Forecasting, Aggregation and Intervention / D. Ocker // University of Konstanz. – Germany. - 1999.

87.Фадеев, И.В. Авторегрессионные алгоритмы прогнозирования / И. В. Фадеев, Н.П. Ивкин // Машинное обучение и анализ данных. - 2011.

88.Eakins, S.G. Can value-based stock selection criteria yield superior risk-adjusted returns: an application of neural networks / S.G. Eakins, S.R. Stansell // Int. Rev. Finantial Analysis. - 2003.

89.Hussain, A.J. Financial time series prediction using polynomial pipelined neural networks / A.J. Hussain, A. Knowles // Lisboa PJG. - 2007.

90.Chai, C. Time Series Modelling and Forecasting using Genetic Algorithms, Proceedings of the First International Conference on Knowlenge-Based Intelligent Electronic Systems / C. Chai, C. Chuek. - 1995.

91.Cortez, P. Genetic and Evolutionary Algorithms for Time Series Forecasting / P. Cortez, M. Rocha // Engineering of Intelligent Systems. - 2001.

92.Peralta, J. Time series forecasting by evoling artificial neural networks using genetic algorithms and differential evolution / J. Peralta, L. Xiaodong. - 2010.

93.Franner, J.D. Physical Review Letter / J.D. Franner, J.J. Sidorowich. - 1987.

94.Лоскутов, А.Ю. Временные ряды: Анализ и прогноз / А.Ю. Лоскутов, О.Л. Котляров, Д.И. Журавлев // Физический факультет МГУ им. Ломоносова. - 2004.

95.Лоскутов, А.Ю. Применение метода локальной аппроксимации для проноза экономических показателей / А.Ю. Лоскутов, О.Л. Котляров, Д.И. Журавлев // Физический факультет МГУ им. Ломоносова. - 2003.

96.Cao, L.J. Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting / L.J. Cao // Francis E.H. Tay. - 2003.

97.Крюков, А.Ю. ARIMA-модель прогнозирования значений трафика / А.Ю. Крюков, Д.В. Чернягин // Информационные технологии и вычислительные системы. - 2001.

98.Elsner, J.B. Singular Spectral Analysis. A new tool in the time series analysis / J.B. Elsner, A.A. Tsonis // Plenum Press. - 1996.

99.Golyandina, N. Analysis of time series structure: SSA and related techniques / N. Golyandina, V. Nekrutkin, A. Zhigljavsky // Chapman & Hall/CRC. - 2001.

100.Vautard, R. Singular spectrum analysis in nonlinear dynamics, with applications to paleoclimatic time series / R. Vautard, M. Ghil // Physica D. - 1989.

101.Allen, M.R. Monte Carlo SSA: Detecting irregular oscillations in the presence of colored noise / M.R. Allen, L.A. Smith // Journal of Climate. – N 9. - 1996.

102.Robinson, P. Time series with strong dependence / P. Robinson // In C. A. Sims (Ed.), Advances in Econometrics, Sixth World Congress, Cambridge: Cambridge University Press. – 1994.

103.Baillie, R.T. Analysing inflation by the fractionally integrated ARFIMA-GARCH model / R.T. Baillie, C.-F. Chung, Tieslau // Journal of Applied Econometrics. - N 11. - 1996.

104.Beran, J. Statistical methods for data with long-range dependence / J. Beran // Statistical Science. - N 7. - 1992.

105.Beran, J. Statistics for Long-Memory Processes / J. Beran // London: Chapman and Hall. - 1994.

106.Broyden, G. The convergence of a class of double-rank minimization algorithms / G. Broyden // Journal of the Institute of Mathematics and Its Applications. - N 6. - 1970.

107.Fletcher, R. A new approach to variable metric algorithms / R. Fletcher // Computer Journal. – N 13. - 1970.

108.Shanno, F. Conditioning of quasi-Newton methods for function minimization / F. Shanno // Mathematics of Computation. - N 24. - 1970.

109.Федорук, В.Г. Протоколы сетевого взаимодействия TCP/IP [Электронный ресурс] / В.Г. Федорук. - Режим доступа: http://www.opennet.ru/docs/RUS/tcpip/.

Приложения

Листинг функций пакета MATLAB

Код MATLAB функции построения прогноза на основе AR(p) модели с выводом графика зависимости ошибки прогнозирования от порядка p.

```
function [ minRMSE ] = MyARfunc(IniArray,tuneSteps)
maxValue = max(IniArray);
s = size(IniArray);
arraySize = s(1);
normArray = zeros(1, arraySize);
transpArray = transpose(normArray);

for k = 1:arraySize
   transpArray(k) = (IniArray(k)/maxValue)*100;
end

MapeError = zeros(1, tuneSteps);
for i = 1:tuneSteps
    DAT = iddata(transpArray(1:arraySize),[],1);
    arModel=ar(DAT,i,'ls');
    YP=predict(DAT,arModel);
    MaperSum=0;
    data = DAT.OutputData;
    predictData = YP.OutputData;
    for k =1:arraySize
       temp = MaperSum;
```

```
        if(predictData(k) ~=0)
            MaperSum =((data(k)-predictData(k)) /data(k));
        end
        MaperSum =temp+MaperSum;
    end
    MAPE=(MaperSum/arraySize)*100;
    if (MAPE ~= 0)
        MapeError(i) =MAPE;
    end
    minRMSE = min(MapeError);
    if(minRMSE==MAPE)
        p=i;
    end
   dif =data-predictData;
   sumOfErrors(i) = sum(abs(dif));
   mae(i)=sumOfErrors(i)/arraySize;
end

subplot(3,1,1);
plot(Yres);
subplot(3,1,2);
plot(dif);
subplot(3,1,3);
plot(Yres);
end
```

Код MATLAB функции построения прогноза на основе ARIMA(p,d,q) модели с выводом графика зависимости ошибки прогнозирования (MAE) от порядка p.

```
function [ MAE ] = MyARfunc(IniArray,forecastSteps,tuneSteps,D,Q)
s = size(IniArray);
arraySize = s(1);
normArray = zeros(arraySize,1);
p = tuneSteps;
%avgValue = mean(IniArray);
for k = 1:arraySize
   %transpArray(k) = (IniArray(k)/avgValue);
   normArray(k) = IniArray(k);
end
for x=1:tuneSteps
        arModel=arima(x,D,Q);
        fittedArima = estimate(arModel,normArray);
        Ymdl1pred=zeros(length(arraySize),1);
       for i=p+1:length(normArray)-p;
          [Ymdl1pred(i)]=forecast(fittedArima,forecastSteps,'Y0',normArray(1:i,1));
       end;
       tY = transpose(Ymdl1pred);
       diff = zeros(size(tY),1);
       diffSum =0;
       for l=1:size(tY)
          diff(l)=abs(normArray(l)-tY(l));
```

```
        diffSum = diffSum + abs(diff(l))
    end
    MAE(x) =diffSum/l;
end
subplot(3,1,1);
plot(normArray);
subplot(3,1,2);
plot(Ymdl1pred);
subplot(3,1,3);
%plot(diff);
plot(MAE);
```

Листинг скриптов пакета R

Код R скрипта построения прогноза на основе ARIMA(p,d,q) модели.

```
remove(arimaNv);

library(forecast);

remove(newArrayArima);
remove(fit);
remove(ME);
remove(MAE);
remove(diffArray);

newArrayArima <-vector();

window <- 50;
initialIndex <- 500;
data<-CPU;
loopSize <- 50;

for (i in 1:loopSize)
{
        from<-initialIndex+i;
        to<-initialIndex+window+i;
```

```
        fit <- arima(data[from:to], order=c(2,1,1));
        arimaNv <- forecast(fit,h=1);
        newArrayArima[i]<-as.numeric(arimaNv$mean[1]);
}
from<-initialIndex+window+2;
to<-initialIndex+window+loopSize+1;

realData <-data[from:to];
diffArray=(realData-newArrayArima);
ME=mean(diffArray);

remove(diffArray);
diffArray=(realData-newArrayArima);
MAE=mean(diffArray)/mean(realData)*100;

remove(diffArray);
diffArray=((realData-newArrayArima)/realData)^2;
RMSE=sqrt(sum(diffArray)/length(diffArray));

plot(realData,type="b");lines(newArrayArima,col="green",type="b");
```

Код R скрипта построения прогноза на основе ARFIMA(p,d,q) модели.

```
remove(farimaNv);
library(arfima);
library(forecast);

remove(newArrayFarima);
remove(fit);
remove(ME);
remove(MAE);
remove(diffArray);

newArrayFarima <-vector();

window <- 50;
initialIndex <-1000;
data<-(CPU);
loopSize <- 300;

for (i in 1:loopSize)
{
        from<-initialIndex+i;
        to<-initialIndex+window+i;

        fit <- arfima(data[from:to],drange=c(0, 1));
```

```
        farimaNv <- forecast(fit,h=1);
        newArrayFarima[i]<-as.numeric(farimaNv$mean[1]);
}
from<-initialIndex+window+2;
to<-initialIndex+window+loopSize+1;

realData <-data[from:to];
diffArray=(realData-newArrayFarima);
ME=mean(diffArray);

remove(diffArray);
diffArray=abs(realData-newArrayFarima);
MAE=mean(diffArray)/mean(realData)*100;
remove(diffArray);
diffArray=((realData-newArrayFarima)/realData)^2;
RMSE=sqrt(sum(diffArray)/length(diffArray));
plot(realData,type="b");lines(newArrayFarima,col="green",type="b");
```