Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Кафедра «Автоматическая электросвязь»
Специальность 6М071900 «Радиотехника, электроника и телекоммуникации»

ДОПУЩЕН К ЗАЩИТЕ
Зав. кафедрой
Чежимбаева К.С.
«____» январь 2014 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ
пояснительная записка

на тему: Исследование протоколов сигнализации сети передачи данных

Выполнил магистрант гр. ССССп–12–1 _____ Наушин Д. А.
(подпись) (Ф.И.О.)

Руководитель профессор, доктор технических наук _____ Данилина Г.П.
(ученая степень, звание) (подпись) (Ф.И.О.)

Рецензент
(ученая степень, звание) (подпись) (Ф.И.О.)

Нормаконтроль старший преподаватель _____ Абиров Д.А.
(ученая степень, звание) (подпись) (Ф.И.О.)

Вычислительная техника профессор, кандидат технических наук _____ Туманбаева К.Х.
(ученая степень, звание) (подпись) (Ф.И.О.)

Алматы, 2014

Аңдатпа

Берілген жұмыста SIP сигнализация протоколының сигналдық трафигін зерттеу қарастырылған. Бұл жұмыста SIP барлық өзітектестік қасиетінің бар екенін көрсететін, SIP протокол трафигінін статикалық анализы және жиынтығы келтірілген.Трафикті қысқа мерзімді болжау ескеріле отырып,SIP сигналды жуктемесін болжаудың тиімді тәсілі таңдалған.Сонымен қатар SIP сигнализация желісінде артық жуктемелерді басқарудың жаңа механизмі ойлап шығарылды

Аннотация

В данной работе рассмотрены вопросы исследования сигнального трафика протокола сигнализации SIP. В работе проведен сбор и статистический анализ трафика протокола SIP, показавший наличие в нем всех основных свойств самоподобия , выбран эффективный метод прогнозирования сигнальной нагрузки SIP ,а также разработан новый механизм управления перегрузками в сети сигнализации SIP, учитывающий кратковременный прогноз трафика.

Введение

Вектор развития услуг связи за последнее десятилетие имел четкую направленность в сторону повсеместного использования сетей на базе протокола IP в качестве транспорта для передачи речевых сообщений. Развивались и сами услуги связи, и теперь голосовой вызов - лишь одна из многочисленных услуг, предоставляемых операторами связи. Появилось множество различных технологий построения сетей фиксированной, мобильной и конвергентной связи на базе концепции передачи голоса поверх IP - VoIP.

Протоколы сигнализации в современных сетях связи эволюционировали наряду с технологиями построения этих сетей. Первый стандартизированный протокол сигнализации в сетях VoIP - H.323, заимствовал основные свои элементы у протоколов сигнализации традиционных телефонных сетей связи. Однако в скором времени протокол инициации сеансов SIP (Session Initiation Protocol), благодаря широкой поддержке производителей, был принят на вооружение ведущими стандартизирующими организациями и на сегодняшний день является основным протоколом сигнализации VoIP.

Одновременно с развитием технологий менялись и основные проблемы при проектировании и эксплуатации сетей связи. Одной из первых проблем был расчет необходимой полосы пропускания в сети для пропуска заданной нагрузки. Однако в связи с бурным развитием IP-сетей и ростом пропускной способности каналов связи выделение достаточной полосы пропускания перестало быть насущной проблемой. Другой немаловажной проблемой стало обеспечение должного качества обслуживания QoS (Quality of Service) в гетерогенной среде передачи критичной к параметрам QoS медиа информации и менее критичного, но более непредсказуемого, трафика данных. Изначально при расчете параметров узлов сети и требований к ним со стороны QoS применялись классические постулаты теории телетрафика. Однако вскоре было обнаружено, что трафик медиа данных, так же как и трафик данных, обладает долгой "памятью" и, следовательно, необходимо использовать новые методики расчета.

Одной из малоизученных проблем современных сетей связи является управление перегрузками (УП) в сети сигнализации. Однако в большинстве исследований трафик собирался в лабораторной сети и предложенные в результате исследований решения носили «косметический» характер. Поэтому актуальным представляется исследование трафика протокола сигнализации SIP, собранного на действующей мультисервисной сети, на предмет выявления в нем характерных свойств для дальнейшего их применения с целью управления перегрузками.

Целью диссертации является сбор и статистический анализ данных о величине трафика протокола SIP для выявления в нем характерных свойств, выбор эффективного метода прогнозирования сигнальной нагрузки SIP и

разработка улучшенного метода борьбы с перегрузками в сетях с учетом выявленных особенностей сигнального трафика.

Для достижения поставленной цели решены следующие задачи исследования:

1) Выполнен сбор статистических данных сигнального трафика протокола SIP в мультисервисной сети крупного оператора связи.

2) Проведен количественный и качественный анализ собранных статистических данных о трафике SIP с целью выявления его характерных свойств.

3) С учетом выявленных свойств реального трафика протокола SIP выбран наиболее эффективный метод его прогнозирования.

4) Разработан улучшенный метод борьбы с перегрузками в сети SIP, учитывающий статистические свойства сигнального трафика.

5) Произведено сравнение разработанного метода с существующими методами борьбы с перегрузками в сети SIP и оценен выигрыш и относительная стоимость его практической реализации.

Научная новизна данной работы заключается в следующем:

1) Количественный и качественный анализ собранных статистических данных о трафике сигнального протокола SIP в крупной мультисервисной сети связи свидетельствует о достаточно сильной степени самоподобия исследуемого случайного процесса поступления сигнальных сообщений на серверы сети.

2) Учет самоподобных свойств сигнального трафика протокола SIP позволяет повысить качество анализа и синтеза мультисервисных сетей связи.

3) Новый метод борьбы с перегрузками в сетях STP позволяет обеспечить более высокое качество работы сети по сравнению с существующим методом за счет учета статистических свойств сигнального трафика.

Предложенный в данной работе улучшенный алгоритм борьбы с перегрузками позволяет устранить недостатки существующего метода «503», существенно повысить устойчивость сетей SIP и значительно сократить задержки установления соединений в них. Результаты статистического анализа трафика SIP, предложенная методика выбора эффективного метода прогноза, а также новый метод борьбы с перегрузками в сетях SIP могут быть использованы на реальных сетях, а также в учебных целях.

1 Анализ исследований трафика IP-коммуникаций и постановка задачи исследования

1.1 Понятие IP-коммуникаций

Начало новой эпохи в телекоммуникациях многие связывают с появлением IP-телефонии. Со второй половины 90-х годов XX века эта технология приобретает все большую популярность, постепенно расширяет свои границы и сейчас это уже не просто услуга для голосового соединения двух абонентов. Она включает в себя видео вызовы, многоточечные конференции разного типа (видео, аудио, web), передачу сообщений, документов, контроль доступности абонентов, роуминг, а также интегрируется со средствами связи, не работающими в режиме реального времени (электронная почта, SMS, факс).

Помимо расширения набора предоставляемых услуг, технология эволюционировала в различных направлениях. В начале 2000х годов появляется концепция сетей следующего поколения NGN (Next Generation Networks). Основной областью применения для данной технологии явился рынок фиксированной телефонии. В мобильной среде появляется технология IMS, которая является основой построения сетей четвертого поколения, стандартизируемых организациями 3GPP (3rd Generation Partnership Project) и 3GPP2. Одновременно в среде корпоративной телефонии на смену традиционным управленческим АТС приходит, так называемые «унифицированные средства связи» (Unified Communication), известные в литературе под аббревиатурой UC.

В связи с указанными изменениями название «IP-телефония» перестало точно отражать суть предоставляемых услуг и принцип построения сетей. Объединяет все приведенные выше технологии и услуги то, что все они базируются на протоколе сетевого уровня IP и все тем или иным образом служат для общения абонентов сети. Поэтому для упрощения предлагается в дальнейшем, в контексте данной работы, называть все вышеперечисленные технологии и все услуги, предоставляемые ими, термином «IP- коммуникации».

1.2 Протоколы IP-коммуникаций

1.2.1 Сигнальная и пользовательская информация

При оказании абоненту той или иной услуги в сетях IP-коммуникаций информация, передаваемая в сети и служащая для организации предоставления услуги, называется сигнальной. В случае голосового соединения двух абонентов сигнальная информация - это набор сообщений, которыми обмениваются узлы сети для установления, поддержания и разрушения соединения. Непосредственно речевые сообщения в процессе разговора двух абонентов представляют собой пример пользовательской информации. Также пользовательской информацией может являться сканированной изображение (например, при передаче факса), текстовое сообщение (например, при предоставлении услуг мгновенных сообщений IM, передаче коротких сообщений SMS) или видео изображение (например, в случае видеотелефонного звонка или видеоконференции).

Кроме приведенных выше явных отличий трафика сигнальной и пользовательской информации, можно выделить еще несколько особенностей, которые делают эти два типа трафика отличными друг от друга:

- требования к качеству обслуживания (QoS - Quality Of Service);
- маршрут распространения трафика по IP-сети;
- объем передаваемого трафика;
- влияние изменений параметров QoS на качество, воспринимаемое пользователями.

К тому же для передачи сигнальной и пользовательской информации по IP-сети используются различные протоколы верхних уровней модели OSI, многие из которых обладают своими специфическими характеристиками.

1.2.2 Протоколы передачи пользовательской информации

Среди протоколов передачи пользовательской информации для передачи голосовой и видеоинформации основным, используемым на данный день, является протокол реального времени RTP (Real-Time Protocol) [86]. Как следует из его названия, данный протокол используется для передачи пользовательской информации в реальном масштабе времени. В качестве транспортного протокола обычно используется протокол передачи пользовательских датаграмм UDP (User Datagram Protocol) [87] без установления соединения. Протокол RTP маркирует все последовательно передаваемые пакеты порядковым номером и временным штампом. Таким

образом, приемная сторона может точно определить количество потерянных пакетов, а так задержку и дисперсию. Многие особенности данного протокола основаны на специфических требованиях QoS.

Так голосовой трафик очень чувствителен к задержке, джиттеру (колебаниям задержек) и потерям пакетов в IP-сети. Максимально допустимая задержка определяется задержкой передачи по IP-сети, а также величиной задержки в буферах транзитных и оконечных устройств, и, в соответствии с [80], не должна превышать 150 мс для лучшего класса качества.

Джиттер в трафике протокола RTP появляется, когда пакеты одного голосового соединения передаются по различным маршрутам. В случае, если промежуточные звенья в IP-сети вносят различную задержку в передаваемые

RTP пакеты, приемная сторона должна иметь возможность ее компенсировать. Для этого организуется джиттер-буфер, размер которого больше или равен размеру одного RTP пакета. Размер джиттер-буфера определяет размер дисперсии, которую он сможет компенсировать. Однако слишком большой размер буфера может привести к дополнительной задержке и ухудшению восприятия принимаемой информации. Поэтому джиттер, вносимый транспортной IP-сетью, стараются поддержать на максимально низком уровне.

Другой параметр качества обслуживания речевых сообщений - потери пакетов. В зависимости от используемого алгоритма кодирования речи, RTP трафик не очень чувствителен к небольшим потерям. Так в случае использования кодека 0.711 потерянный пакет можно спокойно подменить принятым до этого. Поскольку обычно размер речевого пакета (кадра) составляет 20 мс, абонент, скорее всего, не заметит подмены. Однако в случае использования алгоритмов с нелинейным кодированием 0.729, 0.723, 0.726 и др. возможности такого маскирования нет и любые потери пакетов негативно сказываются на качестве воспринимаемой речевой информации.

Маршрут передачи RTP пакетов по IP-сети обычно выбирается как можно короче. Это делается для минимизации задержки передачи и появления джиттера. В простейшем случае весь RTP трафик будет передаваться напрямую между двумя абонентами.

Объем передаваемого пользовательского трафика определяется типом используемого речевого кодека. Например, для кодека 0.711 интенсивность одностороннего потока с учетом всех заголовков составляет порядка 85 кбит/с.

Таким образом, на примере голосовой связи через IP-сети, видно, что трафик протокола RTP является чувствительным к различным параметрам QoS. Любые ухудшения этих параметров сразу же отразятся на качестве предоставляемых услуг. Поэтому поддержание этих параметров на приемлемом уровне является самой популярной задачей для теории и практики.

1.2.3 Протоколы передачи сигнальной информации

Протоколы передачи сигнальной информации более разнообразны по своему составу, чем протоколы передачи пользовательской информации. Многие из сигнальных протоколов различаются по области применения и по типу выполняемых функций. Помимо традиционных протоколов установления соединения, таких как H.323 и SIP, существует протоколы управления медиа шлюзами MGCP и H.248/MEGACO, а также протоколы передачи сигнальной информации стека протоколов ОКС№7 по IP-сети - SIGTRAN, SIP-T, SIP-I. Помимо этого существует ряд более экзотических протоколов с более узким кругом применения, например IAX, SCCP и др.

Самым первым протоколом (а точнее стеком протоколов), нашедшим свое применение на коммерческих сетях операторов связи, является H.323. Первая спецификация протокола увидела свет в 1996 году. Популярность протокола обусловлена по большей части тем, что часть его базируется на протоколе традиционных сетей ISDN - Q.931. Таким образом, H.323 легко интегрируется с существующими сетями, использующими сигнализацию ISDN ОКС№7/Q.93I. К тому же в первые несколько лет, когда технология передачи голоса по IP-сетям набирала популярность, данный протокол являлся, по сути, единственным зрелым протоколом, удовлетворяющим условиям надежности и стабильности, предъявляемым операторами связи.

В 1999 году организация IETF выпустила стандарт, описывающий первую версию протокола инициации сеансов связи SIP. При разработке данного протокола за основу был взят протокол HTTP. Таким образом, SIP меньше походил на традиционные протоколы установления соединения. Однако отличительная гибкость и масштабируемость, присущая протоколу SIP, а также широкая поддержка производителей и стандартизирующих организаций привели к его повсеместному внедрению. На данный момент действующей является вторая спецификация протокола SIP [70], выпущенная в 2002 году, а сам протокол является де-факто стандартом для современных сетей связи.

Протоколы управления шлюзами появились в результате эволюции концепции шлюза протокола H.323. Вместо использования единого элемента предлагалось выделить контроллер шлюза MGC (Media Gateway Controller), выполняющего всю логику управления соединениями и сам медиа шлюз MG (Media Gateway), представляющий собой набор TDM/IP портов. Для управления шлюзами контроллер MGC может использовать либо протокол MGCP, предназначенный для абонентских шлюзов небольшой емкости, или H.248 совместно с SIGTRAN для крупных транковых шлюзов. Стек протокол SIGTRAN был специально разработан для транспортировки сигнальных сообщений стека ОКС№7 по IP-сети. Все протоколы управления шлюзами имеют узкую область применения и не предназначены для непосредственного управления IP-телефонами.

Протоколы сигнальной информации гораздо менее требовательны к параметрам QoS. Задержка передачи сигнальных сообщений складывается из задержки передачи по каналам связи (от сервера к серверу), а также из задержки обработки сообщений этими серверами. Несмотря на то, что сообщение может проходить несколько промежуточных серверов, задержка, вносимая ими, при условии отсутствия перегрузок в сети, составляет доли секунд и может не приниматься в расчет. Однако в случае наличия нескольких перегруженных транзитных элементов на пути сообщений задержка возрастает значительно, а встроенные механизмы ретрансляции некоторых протоколов могут привести к еще большей загруженности элементов сети.

Влияние джиттера на трафик сигнализации незначительно. В рамках отдельного вызова большинство сообщений передается только после подтверждения приема предыдущего. Некоторые протоколы (H323, SIP (опционально)) используют TCP в качестве протокола транспортного уровня, который гарантирует доставку всех пакетов в исходной последовательности. Однако даже в случае использования на транспортном уровне протокола UDP, если сообщения, не требующие подтверждения, были приняты в разной последовательности, это не приведет к серьезным нарушениям.

Единичные потери сообщений никак не влияют на воспринимаемое качество услуг. Протокол H.323 базируется на протоколе транспортного уровня TCP, который гарантирует доставку сообщений, ретранслируя все потерянные сегменты. Протокол SIP, несмотря на использование протокола UDP, имеет встроенные механизмы ретрансляции сообщений. Однако перегрузки каналов связи могут привести к значительному возрастанию потерь, которые в свою очередь приведут к увеличению времени установления соединения.

Маршрут передачи сигнальной информации в большинстве случаев отличается от маршрута передачи пользовательской информации. Сигнальный трафик может пройти несколько промежуточных серверов, прежде чем достигнет вызываемого абонента. Но, как было сказано ранее, при отсутствии перегрузок на этих серверах, задержка, вносимая ими незначительна.

Объем трафика сигнальных протоколов в сравнении с объемом трафика пользовательской информацией минимален и часто может не приниматься в расчет.

Несмотря на менее строгие требования к параметрам QoS, сигнальный трафик, тем не менее, является очень чувствительным к перегрузкам в сети. Задержка установления соединения является основным параметров, характеризующим качество услуг, выполняемых сигнальным протоколам, а перегрузки сетевых элементов приводят к увеличению задержки передачи и сбросам сообщений. Поэтому в области исследования сигнального трафика в пакетных сетях одной из самых популярных задач является борьба с перегрузками.

1.3 Уровни исследования трафика IP-коммуникаций

В связи с большим разнообразием типов трафика в современных пакетных сетях и методов их анализа, необходимо провести систематизацию имеющихся теоретических исследований с целью выявления наиболее общих подходов к математическому описанию трафика IP-коммуникации, а также обозначить области для дальнейших исследований. Существует два основных подхода к исследованию трафика IP-коммуникаций:

- на уровне вызовов;
- на уровне пакетов.

При использовании первого подхода весь трафик в пакетной сети рассматривается как поток отдельных вызовов, поступающих на исследуемую систему. Такой подход является классическим подходом к исследованию телефонных систем. При этом не разделяют трафик сигнализации и пользовательских данных. Причиной этому является то, что в традиционных телефонных сетях сигнализация передавалась внутри речевого канала и каждый канал сигнализации обслуживал единственный речевой канал. С появлением общеканальной сигнализации ОКС№7 [21, 22], однако, этот подход остался неизменным, несмотря на то, что общий канал сигнализации обслуживает целую группу (иногда несколько тысяч) речевых каналов. Совместное исследование трафика сигнализации и медиаданных значительно упрощает задачу исследователя. Однако такой подход предполагает, что каждый поступающий вызов создает одинаковую нагрузку на исследуемую систему. В действительности каждая отдельная услуга может оказывать различную нагрузку на реализующие ее элементы. Некоторые услуги требуют передачи нескольких десятков сообщений (конференции), могут использовать большую полосу пропускания (широкополосное видео), а также могут оказываться без установления соединения как такового (передача сообщений).

В случае исследования трафика на уровне вызовов задача исследователей сводится к определению того, насколько трафик IP- коммуникаций отличается от традиционного телефонного трафика и насколько эти отличия (если таковые имеются) изменяют основные параметры, применяемые при расчете и проектировании сетей IP- коммуникаций (например, характер законов распределения интенсивностей поступления вызовов и распределения длительностей обслуживания вызовов в системе).

Второй подход основывается на том факте, что в отличие от традиционной телефонии в сетях IP-коммуникаций передача любых сообщений осуществляется с помощью технологии коммутации пакетов, что накладывает свои особенности на исследуемые характеристики (изменение нагрузки во времени, размер буферов узлов сети, длины очередей в этих буферах и т.д.). Для исследования трафика IP-коммуникаций на уровне пакетов необходимо произвести его декомпозицию для упрощения и конкретизации целей и объектов исследования.

Весь трафик IP-коммуникаций на уровне пакетов можно разделить на две основные составляющие:

- трафик сигнальной информации - трафик сигнальных сообщений, передаваемых для установления, модификации (изменения) и разрушения сеанса связи в пакетной сети;
- трафик пользовательской информации - трафик передачи голосовых сообщений, видео сообщений и данных пользователей.

Как отмечалось ранее, каждый из этих типов трафика использует свои протоколы передачи и имеет различные требования к качеству обслуживания.

Раздельное исследование трафика сигнализации и пользовательской информации позволяет более точно подобрать математическую модель для данного типа трафика. Трафик медиаданных для каждого отдельного вызова представляет собой непрерывную последовательность пакетов в обоих направлениях, в то время как трафик сигнализации - асинхронный диалог, состоящий из небольших сообщений, передаваемых, в общем случае, в начале и конце соединения. Таким образом, очевидно, что такая декомпозиция имеет смысл, и результаты исследования каждого типа трафика должны рассматриваться раздельно.

1.4 Исследования трафика IP-коммуникаций на уровне вызовов

Как указывалось выше, задача исследования трафика на уровне вызовов сводится к определению двух его основных характеристик:

- вероятностному закону распределения интенсивностей вызовов, поступающих на исследуемую систему;
- вероятностному закону распределения длительностей этих вызовов.

В ряде работ [8, 9, 15, 26, 34] показано, что интенсивность поступления вызовов в исследуемую систему имеет экспоненциальное распределение, а автокорреляционная функция показывает, что поступившие вызовы взаимно независимы, поэтому делается вывод, что интенсивность поступления вызовов достаточно хорошо может описываться классической Пуассоновской моделью.

Распределение длительностей между вызовами в большинстве работ оценивается как степенное [7, 8, 10, 14, 15, 26, 34, 35], подчиняющееся закону Парето [8, 14], лог-нормальному распределению [10, 35] и другим. Различие полученных законов распределения, скорее всего, объясняется различными размерами исследуемых сетей (в [8] сеть состояла из 4 устройств, в [35] измерения проводились на нескольких действующих сетях), длительностью сбора статистики и характером нагрузки.

В целом большинство результатов исследователей сходятся к одному: распределение интенсивностей вызовов хорошо описывается Пуассоновской моделью, в то время как распределение длительностей вызовов лучше

описывается степенными распределениями, а не экспоненциальными, как это полагалось ранее. Конкретный вид степенного распределения зависит от масштаба и структуры сети.

1.5 Исследования трафика на уровне пакетов

Как упоминалось выше для исследования трафика IP-коммуникаций на уровне пакетов целесообразнее рассматривать отдельно сигнальный трафик и трафик пользовательской информации (трафик медиаданных), так как в пакетных сетях эти два вида трафика могут передаваться по разным маршрутам, иметь разные законы распределения временных параметров и обрабатываться различными узлами сети.

1.5.1 Анализ исследований трафика передачи медиаданных

Трафик медиаданных в общем случае может состоять из нескольких типов трафика:

- речевой (голосовой) трафик;
- видеотрафик;
- трафик обмена мгновенными сообщениями (IM-трафик);
- трафик данных (web, факсы и др.).

Последние два типа представляют мало интереса для исследования, поскольку являются протоколами отложенного (не реального) времени, объем передаваемых данных относительно невелик (IM-трафик, факсы) или требования к QoS достаточно низкие (web). Например, исследование IM-трафика выполнено в работе [14], однако, в совокупности с речевым трафиком.

Значительное число работ посвящено исследованию потоков видеотрафика с переменной скоростью VBR (Variable Bit Rate) [36-38, 29, 42]. Основными результатами этих работ являются:

- определено наличие эффекта самоподобия в видеотрафике;
- определено, что распределение длин пакетов, содержащих закодированную информацию, подчиняется закону Парето;
- подтверждено, что долговременная зависимость является неотъемлемой частью видеотрафика и присутствует в нем в независимости от типа используемого видеокодека.

Однако большинство работ посвящено исследованию речевого трафика реального времени [8-14, 26, 29, 45, 61]. В рекомендации P.59 [7] международного союза электросвязи ITU-T описан сигнал, по своим статистическим характеристикам напоминающий человеческую речь. Этот сигнал отражает такие основные особенности человеческой речи, как периоды

активности одного или обоих говорящих (ON-периоды) и периоды, когда один или оба участника разговора молчат (OFF-периоды). Распределение длительностей таких периодов как предполагалось ранее, является экспоненциальным. Однако, как показали проведенные эксперименты, такое предположение действительно не всегда. В различных случаях исследователи доказали, что распределение длительности периодов ON/OFF подчиняется распределению Парето [8, 12, 45], Гамма- распределению [9], распределению Вейбула [9], лог-нормальному распределению [10] и другим. Такие отличия могут быть обусловлены различными размерами исследуемых сетей, различными объемами трафика, индивидуальными особенностями говорящих, механизмами определения голосовой активности VAD (Voice Active Detection) и др.

В целом большинство работ сходятся на нескольких основных выводах о распределениях длительностей периодов ON/OFF:

- распределение является скорее степенным с "тяжелым хвостом", а не экспоненциальным, как это предполагалось ранее;
- законы распределения длительностей периодов ON и OFF могут отличаться [9, 11, 13];
- обычно периоды OFF более «тяжелохвостые», чем периоды ON, что объясняется особенностью человеческой речи [11, 13];
- выбор конкретного закона распределения должен осуществляться в каждом случае индивидуально, так как он зависит от многих факторов;
- закон распределения не зависит от типа используемого кодека [8, 13].

Долговременная зависимость в распределениях длительности ON/OFF периодов накладывает свои особенности при проектировании сетей и расчете характеристик устройств. Требования к размеру буферов и длине очередей для трафика, обладающего самоподобными свойствами, гораздо выше, чем у трафика, основанного на пуассоновском процессе. Это объясняется сильной автокорреляцией самоподобного трафика, а также наличием "тяжелых хвостов" в распределении длительности разговорных периодов. Таким образом, можно сказать, что трафик медиаданных обладает сильными самоподобными свойствами. Однако большое разнообразие особенностей, влияющих на выбор конкретного распределения, свидетельствует о слишком сложной структуре трафика и делает задачу расчета характеристик сети индивидуальной в каждом конкретном случае.

1.5.2 Анализ исследований сигнального трафика

Сигнальный трафик исследовался значительно реже, чем медиатрафик, однако по важности он ни в чем не уступает последнему. Перегрузки на отдельных узлах обработки сигнальных сообщений или во всей сети в целом могут привести к задержке или даже к невозможности установить соединение.

Качество большинства телекоммуникационных услуг и пользовательских приложений в существенной мере определяется качеством функционирования сети сигнализации [21]. Отсюда можно сделать вывод, что сигнальный трафик не менее важен при проектировании и обслуживании сети, чем медиатрафик.

Большая часть исследований сигнального трафика связана с сигнализацией ОКС№7. Несмотря на то, что данная сигнализация в «чистом» виде не используется в IP-сетях, некоторые ее особенности делают похожей на различные типы сигнализаций IP-коммуникаций:

- все сообщения сигнализации ОКС №7 передаются в виде пакетов и сообщений, а сама «наложенная» общеканальная сеть может рассматриваться как сеть с коммутацией пакетов;
- процесс обмена сигнальными сообщениями на различных уровнях стека ОКС№7 очень похож на соответствующие процессы в различных системах сигнализации IP-коммуникаций (SIP, H.323);
- в некоторых случаях требуется передача сообщений протокола ОКС№7 по сетям IP, для чего существуют несколько транспортных протоколов (SIGTRAN, SIP-T).

Наиболее существенные выводы относительно характера трафик сети ОКС№7 на уровне сообщений приведены в [15]:

- автокорреляция процесса имеет форму медленно убывающей зависимости;
- дисперсия самого процесса и процесса, образованного от исходного путем усреднения по времени, убывает медленнее величины, обратной интервалу усреднения;
- очень важной является изучение изменения скорости поступления на исследуемую систему сообщений на небольших масштабах времени (3-10 секунд);
- «тяжелохвостое» распределение длительности вызовов сохраняет свою форму в периоды большой и малой нагрузки, в различных масштабах времени;
- задержки при обработке в узлах сигнализации, повторные передачи сообщений по истечении тайм-аута, сбои в маршрутизации и нестандартные процессы обмена сообщениями оказывают большое влияние на нагрузки в сети и ее характеристики.

В целом большинство выводов этой работы, говорит о том, что на уровне пакетов трафик ОКС№7 обладает долговременной зависимостью. Аналогичное исследование было проведено в [16, 17], однако объектом исследования являлся трафик ОКС№7 не только фиксированных, но и мобильных сетей между двумя центрами коммутации мобильной связи MSC (Mobile Switching Center). Результаты анализа аналогичны сделанным в [15], однако в последнем случае автор использовал метод авторегрессионного проинтегрированного скользящего среднего (АРПСС), позволяющий делать прогноз трафика. В результате, отклонения спрогнозированного трафика 0КС№7 от реального в час пик не превышали 7,14 % в фиксированной сети и 7,83 % - в мобильной.

Ряд работ посвящен разработке методов расчета сигнальной нагрузки сети ОКС№7. Простейшие методы расчета [43, 44] учитывают лишь некоторые параметры сети, такие как число каналов, обслуживаемых звеном сигнализации, среднее число/длина сигнальных единиц для удачных/неудачных вызовов и другие. В дальнейшем данные подходы были дополнены методиками расчета, позволяющими учитывать свойство мобильности абонентов [18], неравенство сигнальной нагрузки в прямом и обратном направлении [23], тип запрашиваемых интеллектуальных услуг на базе технологии CAMEL (Customized Application for Mobile Enhanced Logic) [19], а также нагрузку от передачи сообщений SMS (Short Message Service) [18].

Как уже говорилось выше, существует несколько способов передачи сигнализации ОКС№7 по IP сетям [39-41]. Основными из них являются использование стека протоколов SIGTRAN [39] и протокол SIP-T [40]. Стек протоколов SIGTRAN использует протокол SCTP в качестве протокола транспортного уровня. Для SCTP была разработана математическая модель его функционирования, применяемая далее для расчета задержки передачи сообщений SIGTRAN [20]. Данная модель была доработана в [23] для учета формирования пакета SCTP по таймеру.

Еще одним способом передачи сигнализации ОКС№7 по IP-сетям является протокол SIP-T (SIP for telephones). Сообщения ОКС передаются путем инкапсулирования в поле протокола SDP соответствующих сообщений SIP. В [24] для анализа качественных показателей работы протокола SIP-T рассчитывались размер очереди сигнальных сообщений (размер буфера), средняя задержка в очереди и ее вариация. Для этого в качестве модели очереди использовалась система массового обслуживания (СМО) типа M/G/1 с приоритетным обслуживанием. В результате сравнения теоретических и экспериментальных данных был сделан вывод о достаточной точности разработанных моделей и о возможности их использования на этапе планирования и проектирования транзитных сетей операторского класса, использующих протокол SIP-T.

Как уже упоминалось ранее, одним из самых популярных протоколов в сетях IP-коммуникаций является SIP. Стоит отметить, что протокол SIP имеет огромное число расширений, позволяющих не только установить простое голосовое или видео соединение между двумя абонентами, но и собрать многоточечную конференцию, передавать мгновенные сообщения IM, контролировать доступность абонента (Presence), регистрировать несколько абонентских терминалов (фиксированных или мобильных) с одним номером и множество других. Поскольку наборы услуг, реализованные в разных сетях SIP, могут отличаться, это может оказать существенное влияние на профиль трафика SIP.

Одним из популярных методов исследования трафика сети SIP является симулирование работы данного протокола в различных средах (например, NS-2, OMNeT, NetSim и др.) с помощью машины с конечным числом состояний FSM (Finite State Machine). В общем случае модель работы протокола SIP

может описываться «закрытой сетью». Закрытая сеть представляет собой несколько узлов, каждый из которых соответствует определенному состоянию SIP-сессии в процессе установления или разрушения простейшего голосового соединения, например, ожиданию предварительного или окончательного ответа, ожиданию подтверждения получения сообщения (не учитываются состояния регистрации, аутентификации, пересылки сообщений, размножения сообщений (sip- forking) и др.). Каждому переходу из одного состояния в другое соответствует определенная вероятность, основывающаяся на данных, получаемых из модели транспортной IP-сети. С помощью симуляции могут быть проанализированы различные характеристики сети SIP. Например, в[25] был получен упрощенный (из-за реализации не всех возможных состояний сессии SIP и некоторых допущений относительно транспортной сети) метод определения среднего числа отказов вызовов, получаемого из вероятности сброса вызова. К тому же полученная модель была расширена до модели симуляции работы SIP в беспроводных сетях, путем введения дополнительной вероятности хорошего (плохого) приема сигнала. Также с помощью симуляции может быть оценена задержка установления соединения из-за ретрансляции сообщений по таймеру [46].

Другим способом исследования трафика SIP является анализ реального трафика, собранного с действующей сети. Однако работ, исследовавших реальный сигнальный трафик протокола SIP, достаточно мало, так же, как и ценных результатов. Так в работе [26] на примере двух сетей SIP различного масштаба (от 2 до 57 пользователей) показано, что на уровне пакетов SIP-трафик достаточно точно описывается экспоненциальным распределением. Также установлено, что параметры QoS сигнального трафика (задержка установления соединения, количество несостоявшихся соединений и др.) испытывают незначительное ухудшение, если трафик данных, передаваемый в той же среде, что и телефонный, не превышает 80% (по данным [45] - 70%) от общей пропускной способности сети. В работе [155] был сделан вывод о принадлежности трафика SIP к тяжелохвостым распределениям. Однако большая величина интервалов анализа (несколько сот минут) делает эти выводы малопригодными для применения на действующих сетях, где время реакции сетевых устройств должно быть сокращено до минимума. Одной из интересных задач при исследовании трафика SIP является обеспечение безопасности работы сети. В связи с повсеместным развитием Интернета, обеспечение безопасности сетевой инфраструктуры становится одной из приоритетных задач сервис провайдеров. Каждый человек, получив доступ в сеть, может просканировать сетевые пространства операторов связи на предмет наличия открытых портов. Протокол SIP использует стандартный порт UDP 5060, поэтому злоумышленнику не составить труда за короткое время определить все устройства в сети, поддерживающие протокол SIP. К тому же одной из особенностей сканирования является то, что в сообщении-ответе на сканирующий запрос, многие устройства по умолчанию включают заголовок, указывающий не только производителя устройства, но и версию программного

обеспечения. Злоумышленник может обладать сведениями об уязвимостях данной версии программного обеспечения и попытаться их эксплуатировать. Помимо этого существует целый ряд так называемых man-in-the-middle атак, в которых злоумышленник вклинивается в сигнальный тракт передачи сообщений и может изменять или пассивно записывать весь обмен сигнальной информацией. Своевременное обнаружение этих атак существенно повысит доступность, надежность и стойкость работы сетей SIP. Помимо стандартных решений на основе пограничных контроллеров сессий SBC (Session Border Controller) необходимы дополнительные методы обнаружения атак [81]. В работе [82] разработан метод обнаружения DoS атак на основе особенностей протокола SIP. Уязвимости в процессе аутентификации в сетях IMS, работающих на базе протокола SIP, изучались авторами [83]. На основе статистического анализа трафика SIP были разработаны методы обнаружения искаженных сообщений [84] и спам-атак [85].

Другой очень важной задачей при исследовании трафика SIP, является защита сети от перегрузок [48, 4б]. Именно перегрузки узлов сети SIP могут привести к задержке установления сеанса связи из-за сброса пакетов и их ретрансляции. В [48] рассматриваются перегрузки в сети, вызванные одновременной попыткой множества SIP-терминалов произвести регистрацию на сервере. С помощью специально разработанной имитационной модели было показано, что увеличение паузы между ретрансляцией повторных запросов на регистрацию, как способ борьбы с перегруженностью сервера, менее эффективно, чем наращивание производительности оборудования. Результаты имитационного моделирования показали, что применение порогов в буфере обработки сообщений SIP для предотвращения перегрузок позволяет сократить время установления сеанса связи и вероятность разрушения вызова почти в 10 раз [48].

Несмотря на свою важность, проблема перегрузок в сети сигнализации SIP оказалась одной из самых малоизученных. Поскольку перегрузки в сети сигнализации оказывают влияние на основной параметр качества обслуживания - задержку установления соединений, необходимо понимать, что является причинами перегрузки и как с ними бороться. В следующем разделе более подробно рассмотрен существующий метод борьбы с перегрузками и его недостатки.

1.6 Перегрузки в сети SIP и борьба с ними

1.6.1 Перегрузки в сети SIP

Как и любой другой сетевой элемент, сервер SIP может находиться в состоянии перегрузки, когда он получает сообщений больше, чем он сам в

состоянии обслужить. Во время периодов перегрузки пропускная способность сервера значительно падает. Из-за необходимости постоянно принимать на обслуживании новые вызовы у сервера может в итоге не остаться ресурсов для обслуживания текущих вызовов.

Пропускная способность сервера определяется мощностью его центрального процессора/процессоров (ЦП), объемом оперативной памяти, скоростью сетевых интерфейсов, объемом дисковой памяти и другими ресурсами. Как правило, в случае перегрузки быстрее всего расходуются ресурсы ЦП, так как остальные ресурсы обычно планируются таким образом, чтобы вероятность перегрузки в них была минимальной. Сразу оговоримся, что под перегрузкой сервера SIP будем считать отсутствие ресурсов, необходимых для программной реализации протокола SIP на данном сервере (то есть стека протокола SIP).

Существуют также случаи, когда SIP-сервер может отказать обслуживать вызов, даже если еще не все ресурсы израсходованы. Например, в случае если шлюз с сетью TDM израсходовал все свои TDM ресурсы (отбой с ошибкой 488 (Not Acceptable Here)) или когда сервер регистрации потерял связь с базой данных по абонентам (отбой с ошибкой 500 (Server Error)).

Основные причины перегрузок в сети SIP [74]:

1. Плохо спланированные аппаратные ресурсы. Любая аппаратная платформа для SIP-сервера, как и для любого другого сервера, должна быть спланирована таким образом, чтоб обеспечить предоставление определенного набора услуг определенному количеству абонентов.

2. Неисправности зависимых компонентов. SIP-сервер зависит от нескольких сторонних компонентов - абонентские базы данных, DNS серверы, ENUM серверы и другие. Неисправность любого из них может привести к значительной потере пропускной способности сервера SIP.

3. Неисправности составных компонентов. SIP-сервер может являться кластером и состоять из нескольких серверов. Обычно при правильно спланированном кластере неисправность одного сервера не обязательно приводит к неисправности всего кластера. Однако при некоторых условиях все серверы в кластере будут неспособны обслужить вызов (испорченная запись в таблице базы данных, отсутствие необходимого файла аудио- подсказки и др.).

4. Лавинный рестарт. Одна из основных причин перегрузок. Она возникает, когда, например, в результате выключения/включения электропитания множество устройств пытаются одновременно зарегистрироваться на SIP сервере. Также эта проблема может возникнуть из-за временной неисправности промежуточного прокси-сервера.

5. Значительный поток вызовов в сети. В результате телевизионного голосования или рекламы абоненты могут одновременно начать звонить на определенные номера, что может, при неправильно спланированной сети, привести к перегрузке сетевого элемента.

6. Атака Denial of Service (DoS). В данном аспекте рассматривается злонамеренная атака. Атакующий с целью принесения вреда оператору может

посылать большие объемы трафика из одного или нескольких источников в сети SIP.

Сама по себе проблема перегрузки на одном отдельном SIP-сервере может привести к перегрузкам на других сетевых элементах. На данный момент существует и работает на всех сетях, использующих протокол SIP, механизм, встроенный в сам протокол. Подробнее он рассмотрен ниже.

1.6.2 Недостатки существующего метода борьбы перегрузками в сети SIP

Изначально в спецификации стандарта SIP [70] был определен метод борьбы с перегрузками с помощью сообщений № 503 "Service Unavailable" (условно назовем его "метод 503"). Когда SIP-сервер не в состоянии обработать вызов, он может послать сообщение 503 Service Unavailable. Он также может включить заголовок Retry-After с указанием временного промежутка, говорящего нижестоящему элементу сети о необходимости подождать указанное время. Основная идея данного метода - это перебросить нагрузку с одного сервера на другие на время, за которое сервер должен выйти из состояния перегрузки.

Однако в процессе эксплуатации у данного метода был выявлен целый ряд недостатков [74]:

1. Усиление нагрузки. В случае перегрузки множества компонентов в сети данный метод может привести к еще большей перегрузке. Метод "503" может быть приемлемым в случае перегрузки одного единственного сетевого элемента, но в случае перегрузки множества элементов этот метод приводит к значительному увеличению количества передаваемых сообщений, и соответственно к еще большей нагрузке на серверы. Например, если пара транзитных SIP-серверов находятся в состоянии перегрузки, то отказ в обслуживании одним из них приведет к дополнительной нагрузке на второй, который в свою очередь сам находится в состоянии перегрузки.
2. Недоиспользование ресурсов. В большинстве реализаций протокола SIP получение сообщения 503 означает, что в состоянии перегрузки находится весь сервер, определяемый доменным именем. Хотя в действительности за одним доменным именем может находиться несколько серверов и только один из них перегружен.
3. Проблема неустойчивости трафика. Использования таймера приостановки трафика в заголовке Retry-After приводит к резкому всплеску трафика SIP по истечении данного таймера, что потенциально может привести к очередной перегрузке сервера.
4. Неоднозначность применения. В некоторых стандартах, описывающих взаимодействие сети SIP с сетью ОКС№7, сообщение 503 используется для отображения соответствующего кода отбоя в сети ОКС. Такая

реализация протокола приводит к неоднозначному интерпретированию данной ошибки.

1.7 Выводы и постановка задачи исследования

Термин IP-коммуникации включает в себя весь спектр технологий для предоставления услуг связи пользователям сети на базе протокола IP. Долгая эволюция IP-коммуникаций привела к огромному разнообразию протоколов, а также методов исследования и проектирований современных телекоммуникационных сетей.

В главе предложена классификация методов исследования трафика IP-коммуникаций. На первом этапе классификации весь трафик предлагается рассматривать либо на уровне вызовов, либо на уровне пакетов. Первый подход, хотя и является наиболее простым и удобным, не учитывает, однако, многих особенностей трафика IP-коммуникаций. Различные услуги требуют передачи различного количества сообщений. К тому же есть ряд услуг, которые могут оказываться без установления соединения (передача сообщений, обновление статуса, оповещение о голосовой почте и пр.). Поэтому более точным является исследование трафика на уровне пакетов.

Классификация методов исследования на уровне пакетов предполагает отдельное рассмотрение трафика сигнальной информации и пользовательского трафика. В главе показано, что эти два типа трафика IP- коммуникаций существенно отличаются друг от друга. Помимо того, что они передаются по разным маршрутам и могут проходить через разные транзитные устройства в сети, они обладают отличными требованиями к качеству обслуживания.

Благодаря высоким требованиям QoS, а также их непосредственному влиянию на качество услуг, воспринимаемое пользователями, трафик пользовательской информации представляет наибольший интерес для различных исследований. Основным направлением многих исследований явилось определение модели трафика для уточнения методов расчета, применяемых в современных телекоммуникационных сетях. В большинстве работ авторы сходятся во мнении, что трафик медиаданных проявляет сильные самоподобные свойства, однако выбор конкретной модели зависит от множества параметров и индивидуален в каждом конкретном случае

В то же время сигнальный трафик IP-коммуникаций является наименее изученным. В главе приведен краткий обзор исследований трафика ОКС№7, SIGTRAN, SIP-T и SIP. Показано, что трафик ОКС№7 обладает сильными самоподобными свойствами, что потенциально может быть использовано для его прогнозирования. Для протоколов SIGTRAN и SIP-T, были указаны имеющиеся математические модели расчетов задержек и размеров очередей и буферов.

Протокол SIP является стандартом для современных телекоммуникационных сетей. В настоящее время существует несколько основных направлений исследований трафика данного протокола. Одним из них является метод имитационного моделирования, позволяющий рассчитывать различные характеристики сети SIP.

Одним из самых интересных и востребованных направлений исследований трафика SIP являются методы борьбы с перегрузками в сети сигнализации. Ряд работ на данных момент представили различные модификации существующего метода борьбы с перегрузками. Однако у данного метода были выявлено значительные недостатки, без устранения которых все предложенные модификации являются всего лишь временными "косметическими" решениями.

Одним из существенных недостатков большинства исследований является то, что объектом исследования зачастую являлся искусственно сгенерированный трафик протокола SIP, который, естественно, далеко не обязательно обладает теми же свойствами, что и настоящий трафик. К тому же часть работ, исследовавших трафик с действующих сетей, рассматривали сети небольшого масштаба (до нескольких десятков абонентов), в то время как задача обеспечения отказоустойчивости и борьбы с перегрузками наиболее остро стоит в сетях с большими объемами трафика, состоящими из большого количества элементов.

Из всего вышеперечисленного можно сформулировать следующие задачи исследования в диссертации:

1. Выполнить сбор статистических данных реального сигнального трафика протокола SIP в действующей сети крупного оператора связи.
2. Провести количественный и качественный анализ собранных статистических данных о трафике SIP с целью выявления его характерных свойств.
3. С учетом выявленных свойств реального трафика протокола SIP выбрать наиболее эффективный метод его прогнозирования.
4. Разработать улучшенный алгоритм борьбы с перегрузками в сети SIP, учитывающий статистические свойства сигнального трафика.

Сравнить разработанный алгоритм с существующими методами и оценить выигрыш и относительную стоимость его практической реализации.

2 Статистический анализ сигнального трафика протокола SIP

2.1 Исходные данные

Исходные данные для анализа собирались на сети одного из крупнейших казахстанских операторов IP-телефонии. Объект, на котором собирались данные, в архитектуре протокола SIP представляет собой Full State SIP Proxy/Registrar/Redirect, то есть SIP-прокси сервер, участвующий во всех фазах установления/разрушения вызова (голос, видео, факс), сервер регистрации и сервер переадресации. Также данный объект реализует различные дополнительные виды обслуживания (ДВО) как традиционные для телефонной сети (удержание вызова, переадресация, ожидание вызова и др.), так и специфические для сети SIP (3-х сторонняя конференция, регистрация одного номера за несколькими устройствами, обратный вызов занятого абонента, передачи сообщений и др.). Среди абонентов, зарегистрированных на сервере, присутствуют как абоненты делового (бизнес) сектора, так и домашние абоненты. В качестве абонентских устройств используются как обычные аналоговые телефоны, так и цифровые телефоны с функцией передачи видео и сообщений. Все это делает сигнальный трафик весьма разнообразным и по своей структуре не похожим на сигнализацию в традиционных сетях связи.

Полученные данные представляют собой временные метки прихода различных сообщений SIP (INVITE, NOTIFY, OPTION и др.), взятые из трассировки, сделанной с помощью программы tcpdump. Точность временных отчетов - до 10^6 секунды. Данные на сети собирались в течение недели 24 часа в сутки. В итоге было собрано около 5 миллионов временных отметок. На рис. 2.1-2.3 представлены графики зависимости количества сообщений протокола SIP от времени за различные периоды наблюдения: недельный (рис. 2.1), суточный (рис 2.2.) и 6-ти часовой (рис. 2.3.).

В данной работе будет рассматриваться трафик в максимальном масштабе времени до нескольких десятков секунд. Данный масштаб времени

является особенно интересным, так как специфические свойства статистических характеристик протокола SIP проявляются именно на таких интервалах, а в масштабах дня или недели трафик SIP выглядят почти также, как и трафик традиционной телефонии. К тому же методы борьбы с перегрузками, основанные на самоподобии трафика на небольших масштабах времени, являются наиболее эффективными, так как позволяют быстро реагировать на изменения нагрузки в сети [88].

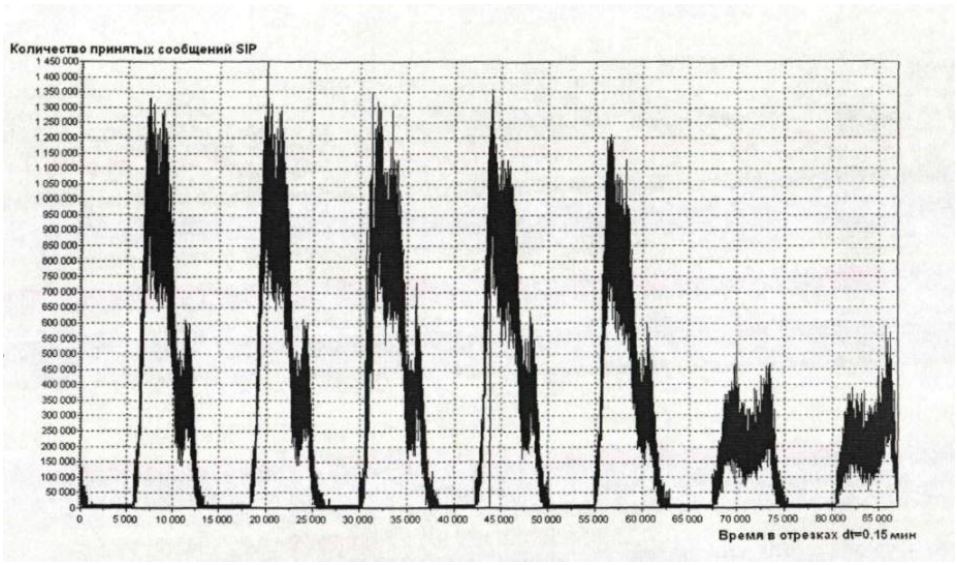


Рисунок 2.1 - Изменение количества сигнальных сообщений протокола SIP от времени за недельный период наблюдения

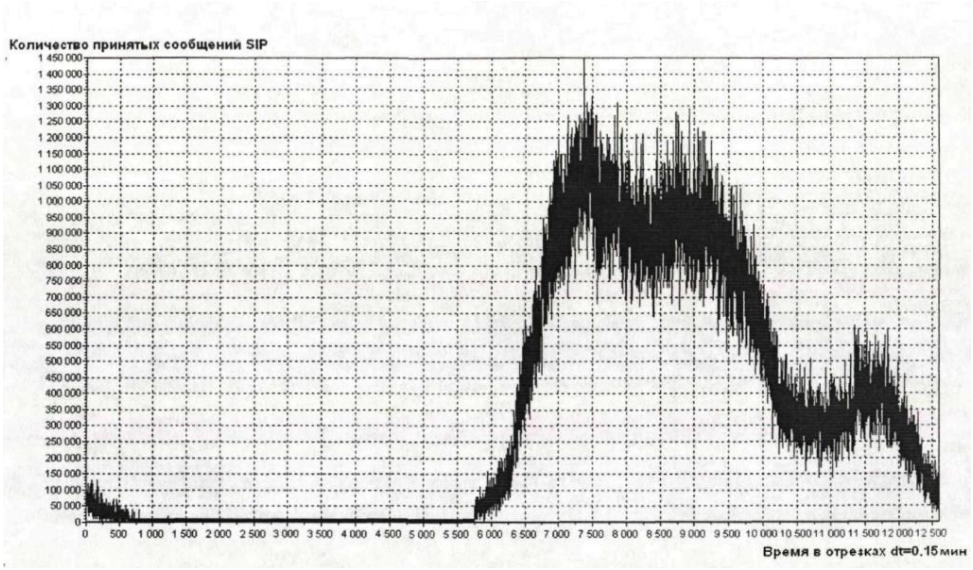


Рисунок 2.2 - Изменение количества сигнальных сообщений протокола SIP от времени за суточный период наблюдения

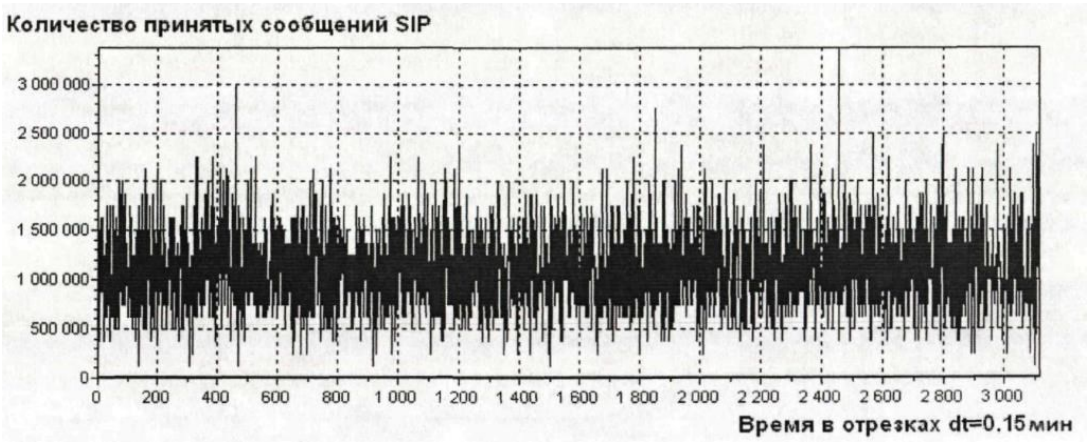


Рисунок 2.3 - Изменение количества сигнальных сообщений протокола SIP от времени за 6-ти часовой период наблюдения

2.2 Проверка наличия основных свойств самоподобных процессов

2.2.1 Определения дискретных во времени самоподобных процессов

Большинство свойств самоподобных процессов определяется особенностями их статистических характеристик (математическое ожидание, дисперсия, коэффициент корреляции). Поэтому необходимо для начала дать определения самоподобным процессам в широком и в узком смысле, определить их основные статические характеристики, а также их специфические особенности.

Допустим, имеется исходный временной ряд $X(t) = (X_1 + X_2 + ...)$ для всех $t \in N=\{1,2,...\}$. Процесс $X(t)$ - стационарный в широком смысле случайный процесс с математическим ожиданием $\mu = E[X(t)]$ и дисперсией

$\sigma^2 = E[X(t)- \mu)^2]$. То есть математическое ожидание такого процесса постоянно, а автоковариация: $\gamma(t,k)= E[(X(t)- \mu)-(X(t + \kappa) - \mu))]$ удовлетворяет условию:

$$\gamma(t,t+k)= \gamma(t,k),$$

где $E[]$- означает операцию усреднения (математическое ожидание). Из условия стационарности также следует, что коэффициент корреляции

$r(k)= \gamma(k)/ \sigma^2$ и автоковариация не зависят от времени.

Для рассмотрения масштабной инвариантности введем агрегированный процесс $X(t)\text{-}X^{(m)}$ с уровнем агрегации равным m.

$X^{(m)}(i)=1/m \sum_{t=m(i-1)+1}^{m1} X(t)$, где m,t €N={1,2,…} (2.1)

Данный процесс получается из исходного путем усреднения по неперекрывающимся блокам размера т и представляет собой, по сути, менее детализированную копию исходного процесса. Через μ_m, $\sigma_m^2 = \gamma_m(0)$ и γ_m обозначим соответственно мат ожидание, дисперсию и автоковариацию агрегированного процесса.

Определение: Процесс называется самоподобным в узком смысле с параметром *H=(0<H<1)*, если

$$X(t)^d = m^{1-H} X^{(m)}(t) \qquad (2.2)$$

То есть процесс $X(t)$ и нормализованный коэффициентом m^{1-H} процесс $X^{(m)}$ должны иметь одинаковые плотности распределения.

Определение: Процесс называется строго самоподобным в широком смысле с параметром *H=(1/2<H<1)*, если его автоковариация имеет вид:

$$\gamma(k) = \sigma^2/2((k+1)^{2H}-2k^{2H}+(k-1)^{2H}) \qquad (2.3)$$

Для всех k≥ 1. В данном случае:

$$\gamma(k) = \gamma(k)_m \qquad (2.4)$$

Для любых $m \geq 1$.

Определение: Процесс $X(t)$ называется асимптотически самоподобным в широком смысле с параметром *H=(1/2<H<1)*, если

$$\lim_{m\to\infty} \gamma^{(m)}(k) = \frac{\sigma^2}{2} \cdot \left((k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\right) \qquad (2.5)$$

Таким образом, формула (4) принимает вид:

$$\lim_{m\to\infty} \gamma^{(m)}(k) = \gamma(k) \qquad (2.6)$$

Форма автоковариации (2.3) и (2.5) не случайна и подразумевает наличие долговременной зависимости. Подробные описания вывода (2.3) приведены в [49, 50].

Зная, что $\sigma^2(aX)=a^2\,\sigma^2(X)$, можно показать, что

$$\sigma^2_m = \sigma^2/m^{\beta} \qquad (2.7)$$

где $0<\beta<1$; $\beta = 2-2H$

Используемый в определениях параметр H - это так называемый параметр Херста, показывающей степень самоподобия. Более детально этот параметр будет рассмотрен далее.

2.2.2 Обработка исходных данных

Для исследования наличия признаков самопободия в исходном временном ряде нам потребуется произвести его агрегирование, то есть приведение его к ряду с постоянным шагом т по шкале времени, где т является уровнем агрегации. Процедура происходит следующим образом: исходный ряд X разбивается на интервалы времени длительностью m. Каждый отчет нового агрегированного ряда будет являться отношением количества пришедших за данный интервал сообщений к длительности интервала т. В качестве исходного ряда возьмем ряд представленный на рис. 2.3., далее его будем обозначать как X. Уровни агрегации m = 10, 20 и 40 секунд. Агрегированные ряды $\boldsymbol{X^{(m)}}$ показаны на рисунках 2.4, 2.5 и 2.6 для соответствующих m.

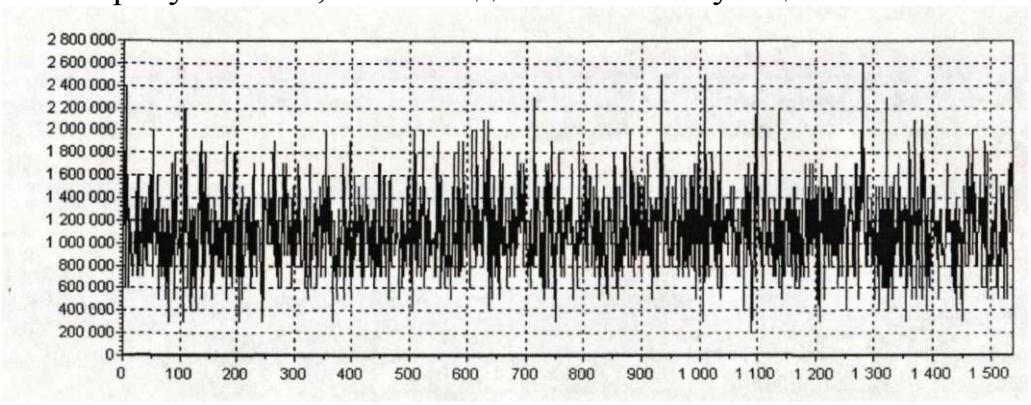


Рисунок 2.4 - Ряд $\boldsymbol{X^{(m)}}$ для $\boldsymbol{m}$ = 10 секунд

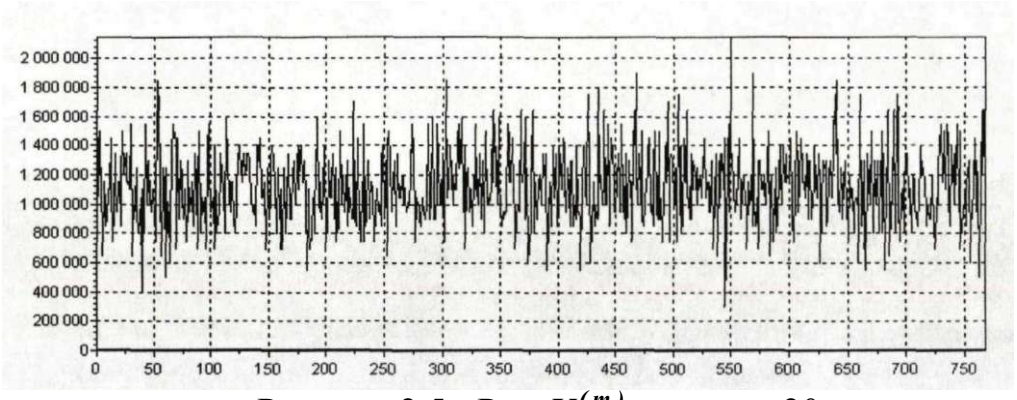


Рисунок 2.5 - Ряд $\boldsymbol{X^{(m)}}$ для $\boldsymbol{m}$ = 20 секунд

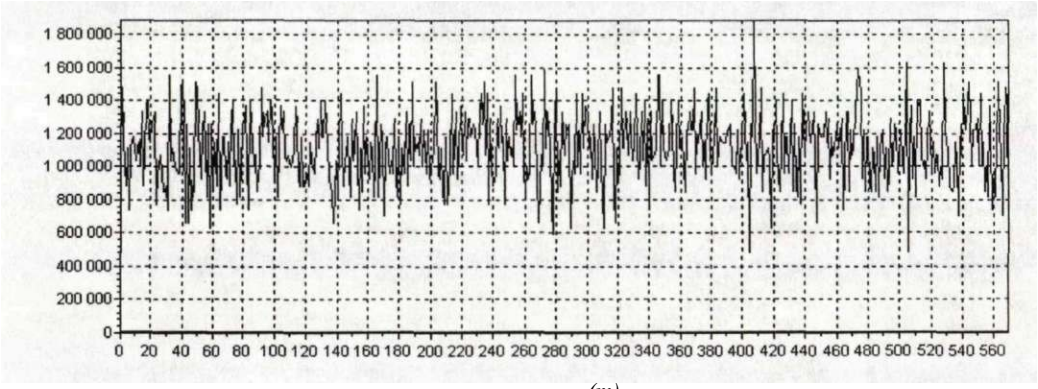


Рисунок 2.6 - Ряд $X^{(m)}$ для m = 40 секунд

Полученные ряды в дальнейшем будут использоваться для проверки наличия эффекта самоподобия в исследуемом трафике протокола SIP.

2.2.3 Анализ автокорреляционных функций

Долговременная зависимость или медленно убывающая зависимость (далее эти термины будут использоваться взаимозаменяемо) процесса ***X(t)*** проявляется, когда его автокорреляционная функция $r(k)=\gamma(k)/\sigma^2$ убывает гиперболически (по степенному закону) так, чтобы:

$$\sum_{k=\infty}^{\infty} r^{(k)} = \infty \qquad (2.8)$$

Автокорреляционная функции (АКФ) *r(к)* для процессов с долговременной (медленно убывающей) зависимостью LRD (Long Range Dependence) имеет вид:

$$r(k) \sim c_1 \cdot k^{-\beta}, k \to \infty \qquad (2.9)$$

В то время как для процессов с быстро убывающей зависимостью SRD (Short Range Dependence) та же функция имеет вид.

$$r(k) \sim c_2^{\,k}, k \to \infty \qquad (2\ 10)$$

где c_1, c_2 - некоторые положительные константы, $0<\beta<1$, $\beta = 2-2H$ [53, 88].

Примеры долговременных зависимостей для разных значений параметра Херста H приведены на рис. 2.7.

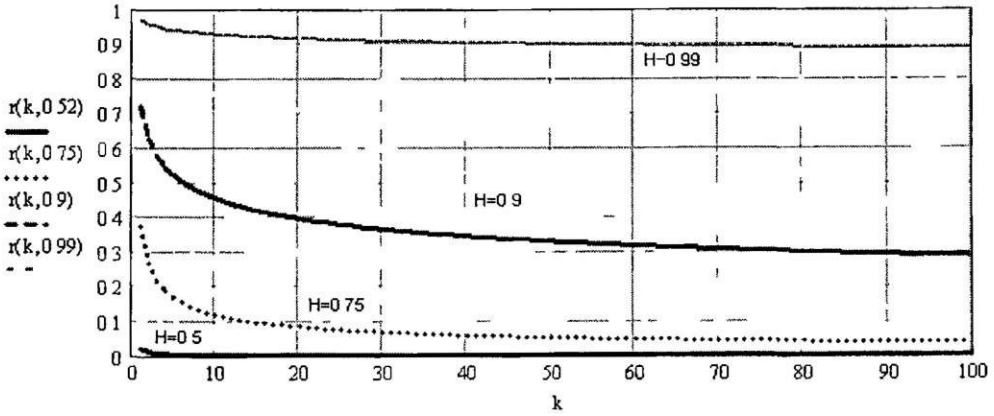


Рисунок 2.7 - Пример поведения автокорреляционной функции *r(к)* для различных *H=1-β/2*

Для обработанных временных рядов построим графики автокорреляционных функций. Для сравнения на каждом рисунке изобразим еще функции быстро и медленно убывающих зависимостей (SRD и LRD соответственно). Следует также заметить, что, не смотря на то, что в исследуемых временных рядах количество отчетов конечно и предельное условие несуммирования АКФ (2.8) не может выполняться, наличие или отсутствие долговременной зависимости нас интересует только в исследуемых временных рамках.

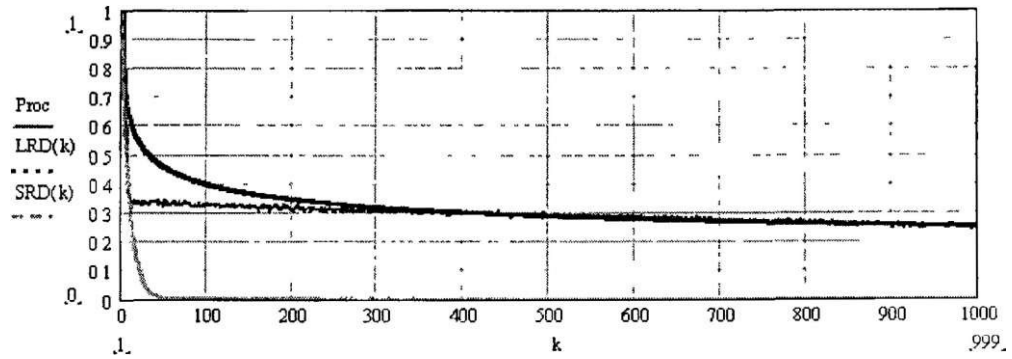


Рисунок 2.8 - Автокорреляционная функция исходного ряда и зависимостей LRD и SRD с параметрами $c_1 = c_2 = 1, \beta = 0.2$

Из рисунка видно, что $r(k)$ исходного ряда практически совпадает с АКФ LRD, а при параметрах с, = 0.37 и β = 0.04 они полностью совпадают (рис. 2.9).

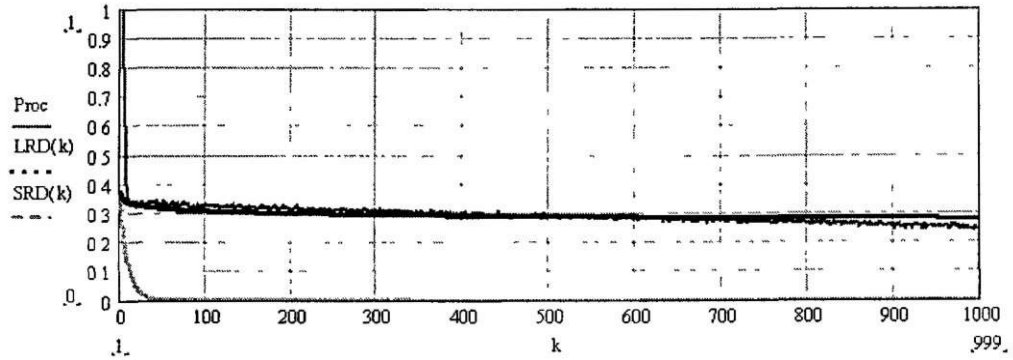


Рисунок 2.9 - Автокорреляционная функция исходного ряда и зависимостей LRD и SRD при $c_1 = c_2 = 0.37, \beta = 0.04$

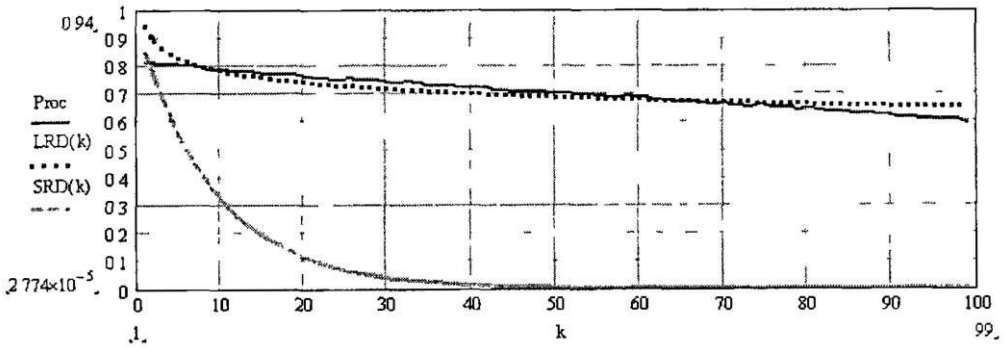


Рисунок 2.10 - Автокорреляционная функция ряда $X^{(10)}$ и зависимостей LRD и SRD при $c_1 = c_2 = 0.94, \beta = 0.08$

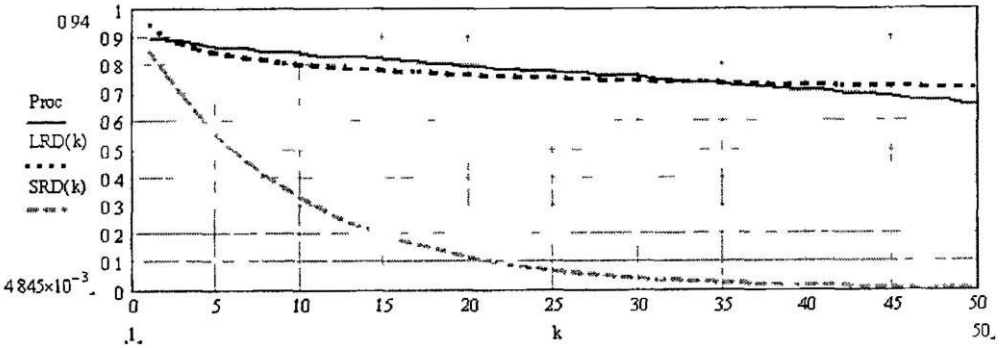


Рисунок 2.11 - Автокорреляционная функция ряда $X^{(20)}$ и зависимостей LRD и SRD при $c_1 = c_2 = 0.9, \beta = 0.05$

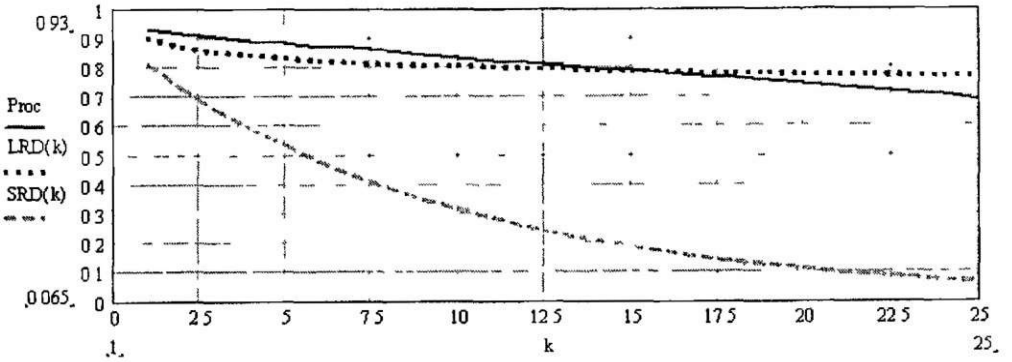


Рисунок 2.12 - Автокорреляционная функция ряда $X^{(40)}$ зависимостей LRD и SRD при $c_1 = c_2 = 0.9$, $\beta = 0.05$

Из приведенных выше графиков видно, что АКФ процессов убывает гиперболически, а не экспоненциально, как у SRD, и практически совпадает с АКФ медленно убывающей зависимости при определенных значениях коэффициентов c_1, c_2 и β.

2.2.4 Анализ спектральных плотностей

Эквивалентное определение медленно убывающей зависимости может быть дано в частотной области. Если спектральная плотность

$$F(\upsilon) = \frac{1}{2 \cdot \pi} \sum_{k=-\infty}^{\infty} r(k) \cdot e^{i k \upsilon} \qquad (2.11)$$

удовлетворяет условию:

$$F(\upsilon) \sim c|\upsilon|^{\beta-1}, \qquad (2.12)$$

где c - некоторая положительная константа, то можно сказать, что исходный процесс X(t) обладает долговременной зависимостью. Спектральная плотность ***F(v)*** в данном случае обладает одной особенностью. В области низких частот ***F(v)*** неограниченно возрастает, стремясь к бесконечности при стремлении частоты к нулю. Данный эффект называется ***1/f-*** шумом.

Следует также отметить, что наличие эффекта самоподобия не обязательно влечет за собой медленно убывающую зависимость АКФ, верно и обратное утверждение. Однако в случае асимптотически самоподобных процессов и ограничений $1/2 < H < 1$ можно утверждать, что процесс обладает долговременной зависимостью и наоборот [51].

Для обработанных временных рядов построим графики спектральной плотности. На всех нижерасположенных рисунках $S(\nu)$ - СП исследуемого процесса, $F(\nu)$ - СП самоподобного процесса ($F(\nu) \sim c|\nu|^{\beta-1}$), с параметром $\beta=0.2$.

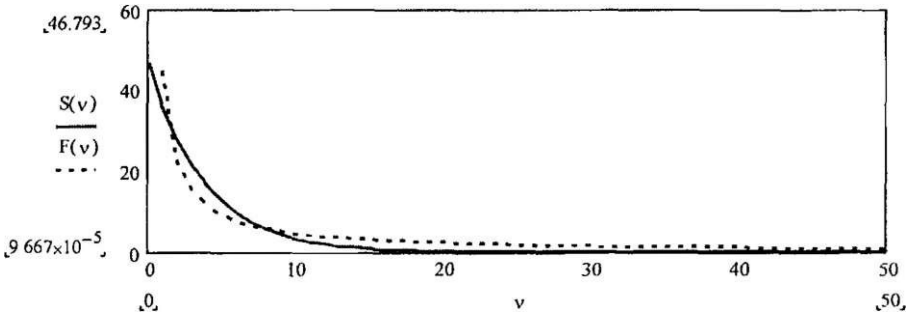


Рисунок 2.13 - Спектральная плотность исходного ряда

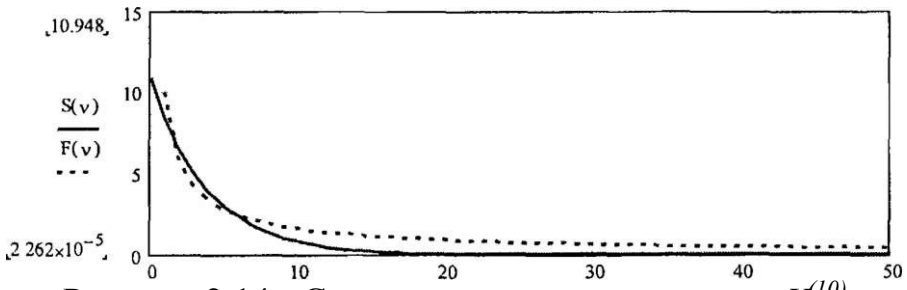


Рисунок 2.14 - Спектральная плотность ряда $X^{(10)}$

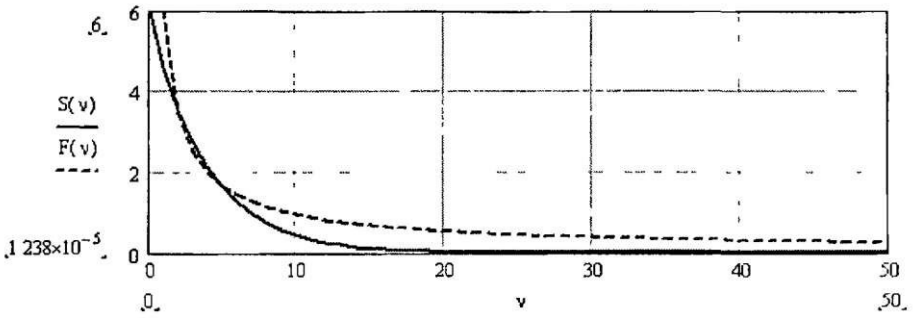


Рисунок 2.15 - Спектральная плотность ряда $X^{(20)}$

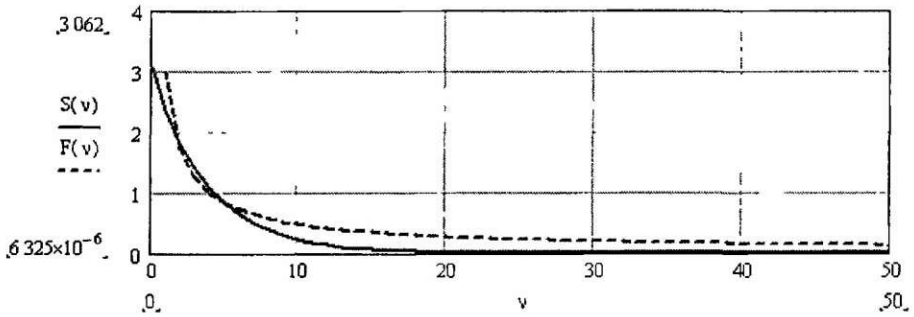


Рисунок 2.16 - Спектральная плотность ряда $X^{(40)}$

Из рисунков видно, что СП исследуемого процесса - *S(**ν**)* практически совпадает с СП самоподобного процесса *F(**ν**).* Также из рисунков видно, что, не смотря на то, что условие возрастания СП до бесконечности при стремлении частоты к нулю не выполняется, так как сумма $\sum_k r(k)$не к бесконечна и ограничена количеством значений АКФ, используемых при вычислении, СП исследуемого процесса асимптотически возрастает на участке $10 > \boldsymbol{\nu} > 0$ Гц.

2.2.5 Анализ плотностей распределений

Говорят, что переменная Z имеет «тяжелохвостое» распределение, если

$$P(Z>x) \sim c\ x^{-a},\ x \to \infty, \qquad (2.13)$$

где *c* - некоторая положительная константа или некоторая медленно изменяющаяся на бесконечности функция, *a* - параметр формы хвоста. Данное распределение, в отличие от «легкохвостых» распределений, имеющих экспоненциальное, быстрое убывание хвоста, имеет медленное, гиперболическое убывание «хвоста». Одной из особенностей «тяжелохвостых» распределения является бесконечная дисперсия для $0 < a < 2$, а при $0 < a < 1$ такие распределения имеют бесконечное математическое ожидание. В случае исследования распределения сетевого трафика наиболее интересным является

случай $1 < a < 2$. Одним из самых часто используемых «тяжелохвостых» распределений является распределение Парето, так как в [2, 4] было показано, что агрегированный трафик от многих ON/OFF источников имеет такое распределение. Надо отметить, что наличие в распределении тяжелого хвоста не является необходимым условием наличия в нем же самоподобных свойств, однако многие самоподобные процессы имеют такие распределения.

Анализ плотностей распределения будем проводить путем построения гистограмм, определяющих частоту появления определенного количества сообщений, для всех исследуемых временных рядов.

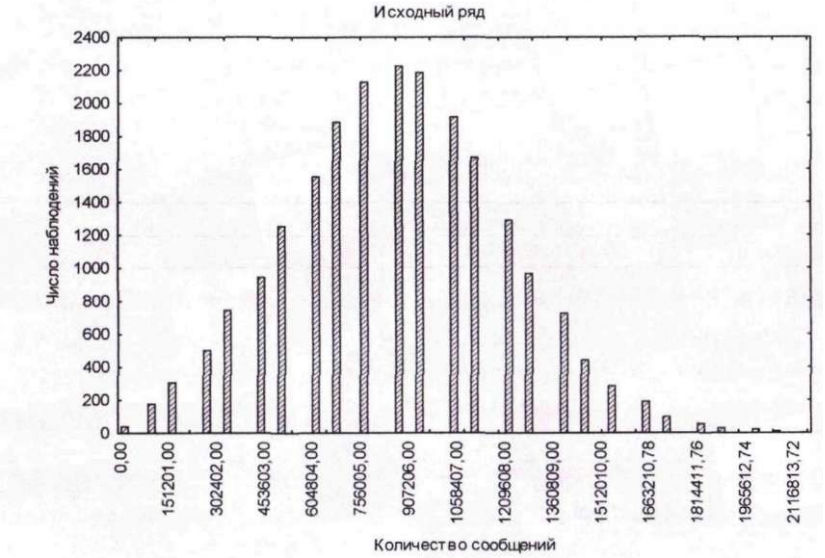


Рисунок 2.17 - Гистограмма для исходного временного ряда

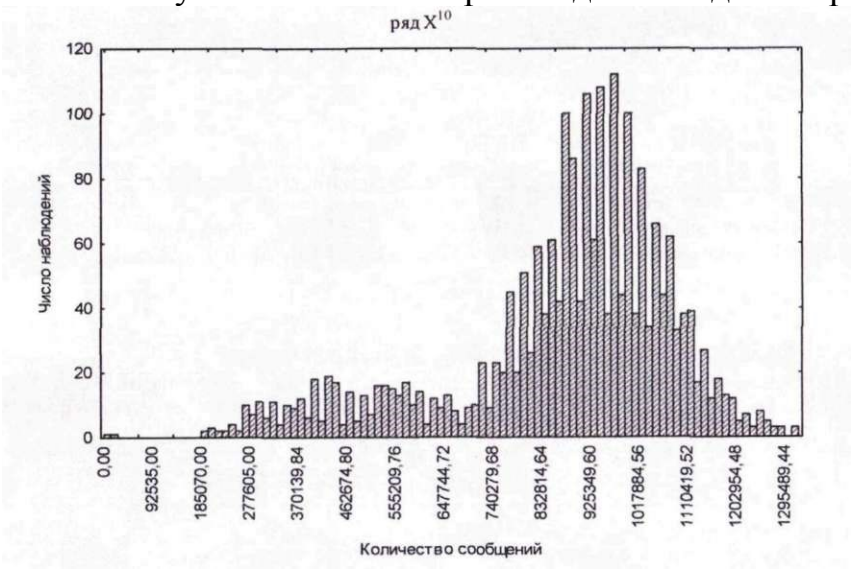


Рисунок 2.18 - Гистограмма для временного ряда $X^{(10)}$

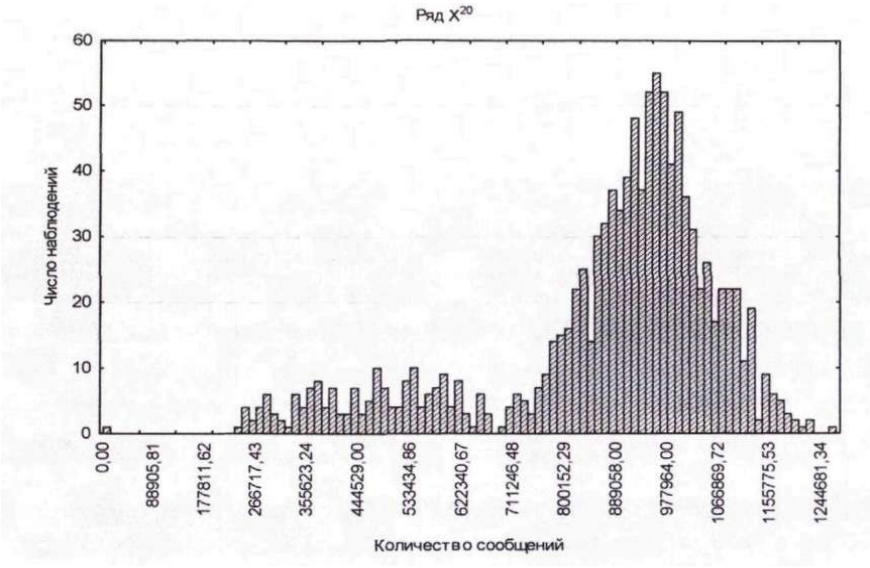


Рисунок 2.19 - Гистограмма для временного ряда $X^{(20)}$

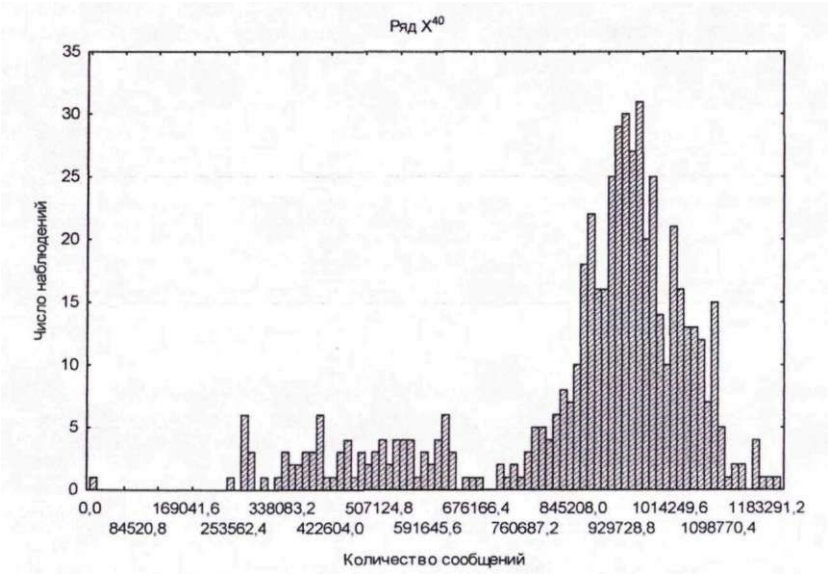


Рисунок 2.20 - Гистограмма для временного ряда $X^{(40)}$

Из приведенных рисунков видно, что с увеличением уровня агрегирования дисперсия ряда уменьшается, в то время как среднее значение

остается практически неизменным. Из визуального анализа гистограммы исходного ряда видно, что «хвосты» распределения убывают медленно, гиперболически, что позволяет сделать предположение, что исходный ряд имеет «тяжелохвостое» распределение.

2.2.6 Анализ дисперсии

Как уже отмечалось выше, дисперсия агрегированного процесса убывает медленнее, чем величина, обратная выборке агрегации (для достаточно больших значений m).

$$\sigma^2 \sim 1/m^{\beta}, \qquad (2.14)$$

Наличие медленно убывающей дисперсии легко протестировать. Достаточно нанести на log-log график зависимость дисперсии от величины т. В результате должна получиться прямая с отрицательным наклоном, меньшим единицы в широком диапазоне m.

Одним из важных свойств медленно убывающей дисперсии является то, что в случае классических статистических тестов, например вычисление доверительных интервалов, обычная мера среднеквадратического отклонения σ является ошибочной на величину, стремящуюся к бесконечности, с возрастанием размера выборки [52].

На рисунке 2.21 изображены графики зависимости логарифмов дисперсии исследуемого процесса (Var) и быстро убывающего процесса (Vr(m)) от логарифма m.

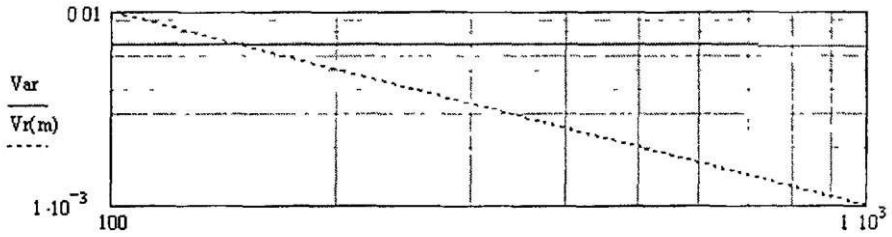


Рисунок 2.21 – Зависимость дисперсии исследуемого временного ряда от величины *m* на log-log графике

Из рисунка видно, что дисперсия исследуемого процесса убывает значительно медленнее, чем дисперсия кратковременно зависимого процесса, из чего можно сделать вывод, что исследуемый процесс имеет долговременную зависимость.

2.3 Параметр Херста и его оценка

Параметр Херста может являться мерой самоподобия процесса. Значение параметра $0.5 < H < 1$ определяют степень самоподобия. Чем ближе параметр H к 1, тем больше процесс самоподобен, то есть тем больше вероятность того, что если процесс возрастал/убывал в предыдущие промежутки времени, то он будет продолжать рост/убывание и дальше. В случае $H = 0.5$ можно говорить о полном отсутствии самоподобия, то есть приращения процесса на предыдущих

шагах никак не повлияют на приращения в последующих шагах. В случае, если значения параметра лежат в пределах $0<H<0.5$, то вероятность того, что на следующем шаге процесс отклонится в сторону, противоположную той, в которую он отклонялся на предыдущем, тем выше, чем ближе параметр H к нулю.

2.3.1 R/S статистика

Отношение *R/*S было введено Гарольдом Херстом при изучении разливов реки Нил и было названо также нормированным размахом. Для заданного набора наблюдений $\{X_n, n \in N = \{1,2,...\}\}$ вводились следующие понятия:

- выборочное среднее $X(n)=1/n \sum_{i=1}^{n} Xi$;
- выборочная дисперсия $S^2(n)=1/n\sum_{i=1}^{n}[Xi - X]^2$;
- размах $R(n)= max(0,1,\Delta_1, \Delta_2... \Delta_n)-min(0,1,\Delta_1, \Delta_2... \Delta_n)$

где $\Delta_k=\sum_{i=1}^{k} Xi - kX, k=1,2...n$

Соответственно отношение *R/S* имеет следующий вид:

$$\frac{R(n)}{S(n)} = \frac{max\Delta n - min\Delta n}{\sqrt{\sum_{i=1}^{n}[Xi-X]}} \quad (2.15)$$

Гарольдом Херстом было показано, что для многих природных явлений данное отношение принимает вид:

$$E\left[\frac{R(n)}{S(n)}\right] \sim c\, n^H, n \to \infty \quad (2.16)$$

где *c* - некоторая положительная константа, не зависящая от *n.*

Чтобы получить оценку параметра *H* надо, прологарифмировав обе части (2.16):

$$\log_E\left[\frac{R(n)}{S(n)}\right] \sim H \log(n) + \log(c) \quad (2.17)$$

построить график зависимости : $\log_E\left[\frac{R(n)}{S(n)}\right]$ от log(n) и, используя метод наименьших квадратов, подобрать прямую линию, наклон которой и будет равен параметру *H* (рис.2.22).

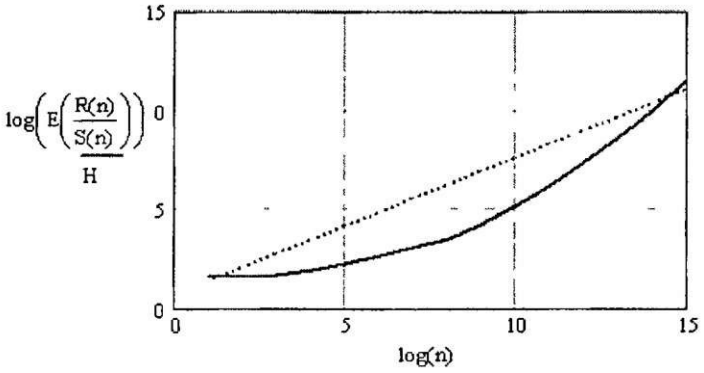


Рисунок 2.22 - Оценка параметра Хёрста *H* методом *R/S* статистики

Надо заметить, что данный метод дает лишь грубую оценку параметра Херста и может применяться лишь для оценки наличия свойства самоподобия в исследуемом процессе.

2.3.2 Дисперсионный анализ

Данный метод основан на свойстве дисперсии самоподобных процессов, описанном выше. В соответствии с ним дисперсия агрегированного процесса может быть вычислена следующим образом:

$\sigma^2_m = \sigma^2/m^\beta$

где $0 < \beta < 1;\ \beta = 2 - 2H$.

Прологарифмировав обе части равенства, получим:

$$\log(\sigma 2) = \log(\sigma 2) - \beta\log(m) \qquad (2.18)$$

Предполагая, что $\log(\sigma^2)$ - константа, не зависящая от *m*, можно найти значение $(-\beta)$ как наклон прямой, найденной по методу наименьших квадратов из точек, представляющих собой график зависимости $\log(\sigma^2_m)$ от log(m) (см. рис. 2.21). Зная оценку β можно найти и значение $H = t - \beta/2$.

Стоит отметить, что данный метод, также как и *R/S* метод дает очень грубую оценку параметра *H* и может быть использован только для оценки наличия самоподобия в исследуемом процессе.

2.3.3 Периодограммный метод

Оценка параметра Херста периодограммным методом основывается на оценке спектральной плотности исследуемого процесса. Периодограмма или

функция интенсивности $I_N(w)$ может быть найдена как преобразование Фурье за период N:

$$I_N(w)=1/2\pi N|\sum_{k=1}^{N} X_k e^{ikw}|^2 \quad (2.19)$$

где w - частота, $X_к$ - исследуемый временной ряд, N - длина временного ряда.

Принимая во внимание особенность поведения самоподобных процессов в частотной области, где частота близка к нулю, спектральную плотность можно представить следующим образом:

$$I_N(w)\sim|w|^{1-2H}, w\to 0 \quad (2.20)$$

Далее надо нанести на график зависимость $\log[I_N(w)]$ от $\log(w)$ для низших 10% значений (в частотной области) $I_N(w)$ и аппроксимировать полученные точки прямой, наклон которой будет равен $1-2H$(рис. 2.22).

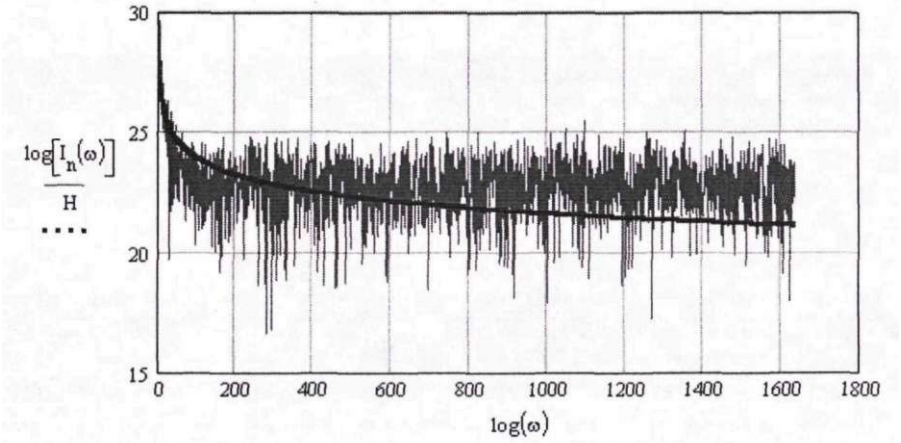


Рисунок 2.23 - Оценка параметра Хёрста H периодограммным методом

2.3.4 Оценка Виттла

Важной особенностью данного, не графического метода, является то, что он предполагает, что исследуемый процесс - самоподобен, но с неизвестным параметром H , и дает оценку этого параметра с определенной точностью. В данном методе используется спектральная плотность $S(w,H)$ известной модели самоподобного процесса, *fbm* - фрактального броуновского процесса. Для оценки параметра H используется, так называемое, выражение Виттла:

$$\int_{-x}^{x} \frac{I(w)}{S(w,H)} dw \quad (2.21)$$

Параметр Херста оценивается путем подбора его значения, которое минимизирует указанное выше выражение Виттла.

Существуют также множество других методов оценки параметра Херста, например, метод Эрби-Витча, метод абсолютных моментов, вейвлет анализ и метод дисперсии остатков.

2.3.5 Оценка параметра Херста

Все методы оценки параметра Херста, кроме оценки Эрби-Витча и Виттла, дают лишь приблизительную его оценку, так как все являются графическими и основываются на принципе аппроксимации, что может вносить значительные искажения в результаты. Поэтому, для этих методов даже в случае достаточно большой величины коэффициента корреляции, между исследуемым и аппроксимируемым процессами, нельзя утверждать, что параметр Хёрста посчитан правильно. Эти методы могут лишь позволить предположить, есть ли в анализируемых процессах долговременная зависимость.

Были получены оценки параметра Хёрста H исследуемых временных рядов большинством из указанных выше способов. Результаты оценок сведены в табл. 2.1. В среднем оценка параметра Хёрста для всех исследуемых рядов находится в пределах $0.6< H <0.8$, что позволяет сделать вывод о том, что исследуемый трафик действительно является самоподобным, то есть обладает долгой памятью.

Ряд данных	Методы оценки параметра Хёрста:

	R/S статистика	Дисперсионный анализ	Периодограмный метод	Метод абсолютных моментов	Метод дисперсии остатков	Метод Эрби-Витча	Метод Виттла
Исходный ряд *X*	0.729 94.45%	0.960 77.82%	0.658 23.63%	0.414 47.22%	0.973 95.86 %	0.567 [0.555-0.578]	0,678 [0,668-0,688]
Агрегированный ряд $X^{(10)}$,	0.543 88.01%	0.946 57.90%	1 71.00%	0.115 52.77%	196.52 %	0.637 [0.599-0.674]	0,857 [0,828-0,886]
Агрегированный ряд $X^{(20)}$	0.436 84.49%	0.941 56.20%	1 79.56%	0.052 54.92%	1 98.06%	0.677 [0.620-0.734]	0,931 [0,890-0,973]
Агрегированный ряд $X^{(40)}$	0.298 78.05%	0.932 56.45%	1 84.99%	0.278 57.40%	1 97.16%	0.766 [0.674-0.858]	0,991 [0,932-1,050]

Т а б л и ц а 2.1 - Результаты оценки параметра Хёрста различными методами

Для *R/S* статистики, дисперсионного анализа, периодограммного метода, метода абсолютных моментов и метода дисперсии остатков указаны в каждой ячейке таблицы оценка параметра Хёрста и коэффициент корреляции (в %), а для методов Арби-Витча и Виттла- среднее значение оценки параметра Хёрста и 95 % доверительный интервал.

2.4 ВЫВОДЫ

В данной главе был произведен статистический анализ трафика сигнального протокола SIP на предмет выявления в нем свойств самоподобия. Было проанализировано несколько сот тысяч пакетов в период •наибольшей нагрузки на сети. Для анализа было произведено агрегирование исходного временного ряда на трех уровнях - 10, 20 и 40 секунд. В результате было показано, что данный трафик обладает большинством из свойств самоподобного трафика: медленно убывающей автокорреляционной функцией, спектральной плотностью, асимптотически стремящейся к бесконечности, в области низких частот, тяжелохвостой плотностью распределения вероятности, медленно убывающей дисперсией. Получена оценка параметра Хёрста, характеризующего степень самоподобия трафика, которая находится в

пределах $0.6<H<0.8$, что свидетельствует о достаточно сильной степени самоподобия.

Таким образом, на основе анализа параметра Херста, а также большинства самоподобных свойств, можно утверждать, что исследуемый сигнальный трафик протокола SIP является асимптотически (благодаря не бесконечной АКФ) самоподобным в широком смысле. Поскольку одним из основных свойств самоподобных процессов является долговременная зависимость, полученные выводы в дальнейшем могут быть использованы для прогнозирования трафика. Полученный прогноз может быть использован для целей борьбы с перегрузками в сетях протокола SIP.

Прогнозирование самоподобного трафика протокола сигнализации SIP

3.1 Монофрактальные и мультифрактальные процессы

Наличие эффекта самоподобия в сетевом трафике после появления работы [1] считалось неоспоримым фактом. Однако почти сразу возник вопрос - достаточно ли точно самоподобные (монофрактальные) процессы описывают поведение сетевого трафика или для этого нужно использовать более широкий класс процессов - мультифрактальные процессы.

Изначально теория мультифракталов использовалась для изучения распределения дробных размерностей случайных или неслучайных последовательностей. В особенности мультифрактальный анализ подходит для изучения локальных флуктуации в этих последовательностях, так как соответствующая фрактальная размерность определяла степень "неровности" исследуемого участка в определенных зафиксированных областях. Поведение исследуемого процесса в малых масштабах времени обычно представляло интерес для исследований в области прикладной физики, например в области теории турбулентности. Однако для других областей, как например изучение сетевого трафика, больший интерес представляет поведение трафика на больших масштабах времени, так как локальные изменения сильно подвержены состоянию сети (количеству пользователей, типам приложений, протоколам, топологии). Для моделирования сетевого трафика более важным является исследование общих характеристик и моделей поведения, чтобы сделать возможным применения моделей для широкого класса различных сетей.

В результате исследований [63], целью которых было установить, могут ли самоподобные процессы достаточно точно описать сетевой трафик, было установлено, что совершенно не обязательно для этого использовать более общую модель мультифрактальных процессов. Однако в случае, если некоторые из признаков самоподобия не были найдены в исследуемом трафике, допустимо применять обобщенную модель мультифрактальных процессов. В большинстве же случаев статистические самоподобные процессы адекватно описывают трафик.

Поскольку, как было показано в предыдущей главе, трафик протокола SIP обладает всеми основными свойствами самоподобных процессов, в дальнейшем для исследования предлагается использовать модели монофрактальных процессов.

3.2 Классификации моделей самоподобных процессов в соответствии с задачей исследования

Основным недостатком теории самоподобных процессов, в отличие от классических подходов к расчету и анализу сетевого трафика, является отсутствие какой-то более или менее стройной теории, позволяющей достаточно точно описывать системы распределения трафика с самоподобной нагрузкой. Это и объясняет ограниченность ее применения при проектировании и обслуживании сети. Также отсутствует определенность в выборе модели для исследования такого трафика. Так для разных типов трафика и для разных задач исследователи используют большой спектр моделей трафика. Однако не всегда одна и та же модель пригодна для разного типа задач.

В [54] приведена классификация моделей в соответствии с поставленной задачей исследования. Показано, что существует два основных класса задач - задачи прогнозирования изменения трафика на определенный период и задачи оценки основных характеристик качества обслуживания трафика в пакетной сети (задержка пакетов, вариация этой задержки и доля потерянных пакетов). Для первого класса задач могут использоваться модели временных рядов - различные модели авторегрессии. Для второго типа задач могут использоваться так называемые псевдо-самоподобные модели трафика (ПСМТ) [64].

Появление ПСМТ объясняется, с одной стороны относительной сложностью фрактальных авторегресионных процессов, наиболее точно описывающих самоподобные процессы, с другой стороны желанием использовать уже наработанную базу приемов и инструментов, разработанных при использовании классического подхода к анализу трафика.

ПСМТ включают в себя весь спектр процессов, модулированных марковским - MAP (Markovian Arrival Process), MMPP (Markov Modulated Poisson Process), PH (Phase Type), MMBP (Markov Modulated Bernoulli Process) или более общий BMAP (Branch Markovian Arrival Process). Из данных процессов одним из самых широко используемых является MMPP [54, 64, 65, 66], что объясняется тем, что он хорошо описывается с физической точки зрения (пуассоновский процесс с несколькими уровнями интенсивности), а также тем, что он хорошо описывает многие реальные процессы.

Однако у ПСМТ есть ряд существенных недостатков. В [65] было показано, что каждая выбранная модель должна быть тщательно исследована на предмет того, что можно ей описать, а что нет. В частности показано, что ПСМТ не может адекватно отражать поведение дисперсии самоподобного трафика. Также одной из основных проблем всех моделей такого типа является невозможность обобщения модели до всех возможных сценариев, а не только тех, на которых она проверялась.

Для целей прогнозирования трафика могут быть использованы различные временные ряды. Поскольку выбор наиболее эффективного метода прогнозирования трафика протокола SIP является одной из задач,

поставленных в первой главе данной диссертации, далее будет рассмотрено несколько наиболее популярных методов, а также будет произведено их сравнение на основе собранных данных.

3.3 Методы прогнозирования сетевого трафика

Существует ряд моделей временных рядов, которые могут быть потенциально использованы для целей прогнозирования сетевого трафика. С точки зрения самоподобных процессов их можно разделить на две категории - модели с короткой памятью и модели с длинной памятью. Помимо этого есть, так называемые, нетрадиционные модели, которые могут быть использованы для прогнозирования рядов с долговременной зависимостью. Ниже рассмотрим основные модели, которые будут использоваться в дальнейшем для прогнозирования трафика протокола SIP.

3.3.1 Модели с короткой памятью

Среди моделей первой категории можно выделить модель авторегрессии (AR - autoregression), модель авторегрессии скользящего среднего (ARMA autoregression moving average), модель авторегрессии проинтегрированного скользящего среднего (ARIMA - autoregression integrated moving average).

Регрессионные модели позволяют определить неизвестный элемент ряда, зная набор предыдущих значений за период, называемый временным окном, и используя скользящее среднее «белого шума».

Для дальнейшего описания моделей введем оператор задержки B такой, что $BX_t = X_{t-1}$, или для более общего случая $B^s Xt = X_{t-s}$. Допустим также, что Δ будет являться оператором разности, то есть $\Delta X_t = X_t - X_{t-1}$ и, соответственно, $\Delta^d = (1-B)^d$, что также можно определить биномиальным выражением:

$$(1-B)^d = \sum_k \binom{d}{k} (-1)^k B^k, \qquad (3.1)$$

где $\binom{d}{k} = \frac{d!}{k!(d-k)!}$;

Определим также полиномы $\phi(B)$ и $\theta(B)$

$\phi(B) = (1 - \phi_1 B - A - \phi_p B^p)$;

$\theta(B) = (1 - \theta_1 B - A - \theta_q B^q)$;

1) Авторегрессионная модель порядка p, обозначается как AR(p) и имеет следующий вид:

$\phi(B)X_t=\varepsilon_t$ (3.2)

где ε_t - «белый шум» (независимая случайная переменная с нулевым средним и дисперсией σ^2_ε). В этой модели переменная X_t вычисляется из предыдущих значений самой себя:

$X_t= \phi_1 X_{t-1}+A + \phi_p X_{t-p}+\varepsilon_t,$ *(3.3)*

AR модели могут использоваться для моделирования стационарных временных рядов (временных рядов с нулевым средним).

Предсказатель, прогнозирующий следующее значение ряда на основе модели *AR*(1), имеет следующий вид:

$X_{t+1}= \phi_1 X_t+ \varepsilon_t,$ (3.4)

2) Авторегрессионная модель со скользящим средним ARMA(p,q) имеет следующий вид:

$\phi(B)X_t=\theta(B)\varepsilon_t,$ *(3.5)*

или:

$X_t= \phi_1 X_{t-1} +A+ \phi_p X_{t-p}+\varepsilon_t -A- \theta_p \varepsilon_{t-p},$ *(3.6)*

Следуем заметить, что $\theta_p \varepsilon_{t-p}$ является скользящим средним в данной модели. Данная модель обладает большой гибкостью при моделировании временных рядов, однако не может использоваться для моделирования нестационарных временных рядов.

3) Авторегрессионная интегрированная модель со скользящим средним ARIMA$_{(p,d,q)}$ является доработанной моделью ARMA(p,q), ARIMA$_{(p,d,q}$) имеет следующую форму:

$\phi(B)\Delta^d X_t=\theta(B)\varepsilon_t$ *(3.7)*

Метод ARIMA может использоваться для моделирования нестационарных процессов. Элементы ряда X_t могут быть найдены из следующего выражения:

$X_t=(1+B+B^2+A)^d \phi^{-1}(B)\ \theta(B)\varepsilon_t,$ (3.8)

В последнем выражении X_t - это регрессионная сумма (интеграл) бесконечного числа переменных шума.

Предсказатель, прогнозирующий следующее значение ряда на основе модели ARIMA (0,1,1) имеет следующий вид:

$X_{t+1}=X_t+ \varepsilon_t- \theta_1 \varepsilon_t$ (3.9)

3.3.2 Модели с длинной памятью

Модели с длинной памятью лучше всего подходят для описания самоподобного трафика. Среди этих моделей можно выделить модель фрактальной авторегрессии проинтегрированного скользящего среднего (FARIMA - fractal autoregression integrated moving average), модель фрактального броуновского движения (fBm - fractal Brownian motion), процесс,

описывающие приращения fBm - фрактальный гауссовский шум (fGn - fractal Gaussian noise) [57]. В контексте данной работы ограничимся рассмотрением самой распространенной моделью FARIMA.

Модель FARIMA является естественным расширением модели ARIMA, в которой предполагается, что коэффициент d может принимать

действительные, а не только целые значения. Процесс X_t является стационарным FARIMA процессом, если:

$$\phi(B)\Delta^d X_t=\theta(B)\varepsilon_t, \qquad (3.10)$$

где d- действительное число ($-0.5 < d < 0.5$).

Соотношение между параметром Херста и значением d имеет следующий вид:

$$H=d+1/2, \qquad (3.11)$$

Таким образом, можно говорить, что X, это процесс с долгой памятью, если ($0 < d < 0.5$) и с короткой памятью - если $d=0$).

Простейшая форма процесса FARIMA (0,d,0) имеет следующий вид:

$$\Delta^d X_t=\varepsilon_t, \qquad (3.12)$$

Предсказатель, прогнозирующий следующее значение ряда на основе модели FARIMA, имеет следующий вид:

$$X_{t+1}=-\sum_{j=1}^{\infty}\pi_j X_{t-j+1}, \qquad (3.13)$$

где коэффициенты π_j рассчитываются из следующего выражения:

$$\sum_{j=0}^{\infty} B^j\pi_j=\phi(B)\,\theta^{-1}(B)(1-B)^d$$

3.3.3 Предсказатель MMSE

Помимо моделей прогнозирования есть еще и так называемые простые предсказатели, которые также используются для целей прогнозирования сетевого трафика. Они не основываются ни на каких моделях и не делают никаких предположений о прогнозируемом трафике, а позволяют только предсказать дальнейшее поведение реализации случайного процесса, например, вычисляют оценку математического ожидания реализации в точке предсказания. Одним из наиболее часто используемых является предсказатель минимума среднего квадрата ошибки MMSE (Minimum Mean Square Error).

Для объяснения его работы рассмотрим линейный стохастический процесс X_t и допустим, что следующее значение X_t может быть представлено линейной комбинацией его предыдущих значений:

$$X_{t+1}=w_m X_t+A+w_1 X_{t-m+1}, \quad (3.14)$$

где m - порядок регрессии.

Эквивалентное выражение в виде матриц может быть записано в следующем виде:

$$\boldsymbol{X_{t+1}=WX_1^t}, \quad (3.15)$$

Как видно из этой формулы, это тот же случай, что и для регрессионных моделей, рассмотренных выше. В задачах прогнозирования трафика зачастую ничего неизвестно о его структуре, однако можно определить весовые коэффициенты w_m.

Спрогнозированное значение временного ряда может быть записано в виде:

$$\boldsymbol{X_{t+1}=WX^t}, \quad (3.16)$$

где W - является вектором оцененных значений весовых коэффициентов.

Предсказатель MMSE получает вектор оцененных значений, основываясь на критерии минимизации ошибки прогноза

$$e_t=X_{t+1}-X_{t+1}, \quad (3.17)$$

и ее дисперсии:

$$E[e_t] = E[(X_{t+1}-X_{t+1})^2] \quad (3.18)$$

Фактически задача минимизации сводится к вычислению вектора W по следующей формуле:

$$W =\Gamma G^{-1} \quad (3.19)$$

где G - автокорреляционная матрица вида:

$$G=\begin{bmatrix} p & A & p \\ M & 0 & M \\ p & A & p \end{bmatrix}$$

Г- автокорреляционный вектор вида:

$\Gamma = [p_m\ A\ p_1]$

Значения автокорреляций можно определить по следующей формуле:

$p_k=\frac{1}{m}\sum_{t=k+1}^{m} X_t X_{t-k}$,

где m - порядок предсказателя MMSE.

Преимуществом использования предсказателя MMSE является отсутствие необходимости знать что-либо о структуре прогнозируемого трафика, следовательно, его можно использовать в качестве предсказателя в реальном масштабе времени. Другим достоинством данного метода является относительная простота реализации [89].

3.4 Методы оценки качества прогнозирования

3.4.1 Аналитические методы оценки

В связи с большим разнообразием методов прогноза встает нетривиальная задача выбора оптимального предсказателя. Оптимальность его надо определять исходя из нескольких критериев [57]:

1. **Точность:** это самый важный критерий при выборе предсказателя, так как целью предсказания является максимально точное моделирование будущего.
2. **Простота:** чтобы предсказание велось в реальном масштабе времени необходима определенная степень простоты предсказателя. Простота играет существенную роль, так как отражает эффективность реализации предсказателя. Чем проще математический аппарат предсказателя, тем меньшую нагрузку он создает на вычислительные ресурсы сервера, реализующего его алгоритм.
3. **Работа в реальном времени:** большинство исследований по моделированию трафика проводились в относительном масштабе времени. На самом же деле для целей оперативного управления сетью необходим моментальный прогноз будущего. Прогноз делается исходя из предположения, что сначала о трафике ничего неизвестно и необходимо предсказывать некоторые его параметры "на лету".
4. **Адаптивность:** хороший предсказатель должен уметь адаптироваться к изменяющемуся трафику. С течением времени для анализа становятся доступны все большие данные о трафике. Предсказатель должен уметь использовать эти данные для корректировки и улучшения качества предсказания.

Такие критерии, как адаптивность и возможность работы в реальном времени, можно считать обязательными, так как это требуется для последующей реализации этих механизмов для целей управления трафиком в телекоммуникационном оборудовании.

Некоторые методы прогнозирования могут быть использованы только для off-line прогнозирования (прогноз в относительном масштабе времени) и выступать только в качестве образца или относительного уровня для сравнения on-line предсказателей (предсказатели в реальном масштабе времени). К таким методам можно отнести модели с длиной памятью. Для их применения

требуется оценка параметра *d,* который определяется из параметра Херста *H (d=H*-0,5). В то же время задача оценки этого параметра, как было показано в главе 2, далеко нетривиальна и основывается на множестве методов (часть из которых являются графическими), каждый из которых может дать сильно отличающиеся оценки значения этого параметра. Так что переложить задачу оценки параметра *H* на компьютер или каким- либо образом автоматизировать ее представляется очень сложным. Для простоты далее будем использовать предсказатель FARIMA, как относительный уровень для сравнения других предсказателей между собой.

Другие два критерия, простота и точность, можно считать двумя противоположными критериями. Самые точные методы прогноза зачастую бывают несоизмеримо сложными в реализации, что делает их малопривлекательными для разработчиков программного обеспечения. С другой стороны методы, показывающие не самый точный прогноз, могут быть значительно проще внедрены и реализованы на оборудовании. Отсюда можно сделать вывод, что основной вопрос при выборе предсказателя по вышеперечисленным критериям будет заключаться в выборе оптимального соотношения простота/точность.

3.4.2 Численные методы оценки

Помимо приведенных выше аналитических критериев необходимы также численные методы оценки и сравнения предсказателей между собой. Для этого существуют множество различных оценок. Основными методами оценок точности прогноза являются [72]:

- средний квадрат ошибки MSE (Mean Squared Error);
- обратное отношение сигнал/шум и коэффициент детерминации;
- отклонение и коэффициенты переоценки и недооценки;
- средняя абсолютная ошибка в процентах.

Все эти оценки основываются на ошибке прогнозирования. Ошибка прогнозирования представляет собой разность между действительным и спрогнозированным значением временного ряда:

$$e=|X-X|, \qquad (3.20)$$

Однако, используя абсолютную ошибку, никогда не оценивают качество прогнозирования, так как сумма абсолютных значений может быть равной нулю. Все последующие методы основываются на различных вариантах ее использования.

3.4.2.1 Средний квадрат ошибки

Одной из самых часто используемых оценок точности прогноза является средний квадрат ошибки MSE (Mean Square Error):

$$MSE=\Sigma e^2/n, \quad (3.21)$$

где n - количество наблюдений.

Ошибка возведена в квадрат для того, что положительные и отрицательные ее значения не уравновешивали друг друга. Данная оценка не имеет никакого физического смысла и малоинформативна, поэтому редко употребляется в чистом виде. Однако очевидно, что чем ближе MSE к нулю, тем точнее предсказатель. Также данная оценка используется в методе прогнозирования MMSE.

3.4.2.2 Коэффициент детерминации и отношение сигнал/шум

Другой способ оценки использует обратное отношение сигнал/шум:

$$SNR^{-1}=\frac{\Sigma e2}{\Sigma X2} \quad (3.22)$$

Чем меньше получается данное соотношение, тем лучше предсказатель. Однако такой способ оценки сильно зависит от среднего значения ряда X. Поэтому в [29] было предложено использовать доработанную оценку, исправляющую данный недостаток.

$$SNR^{-1}=\frac{M(e2)}{M(X2)}=\frac{M(X-X)2}{M(X-M(X)2)} \quad (3.23)$$

Данная оценка позволяет сравнивать качество прогноза различных предсказателей между собой. Она показывает, насколько используемый предсказатель улучшает прогноз по сравнению с прогнозированием по среднему значению. В случае, если $SNR^{-1}_{ср}=1$, то качество прогноза такое же, как и качество прогноза по среднему значению. Однако чем меньше этот коэффициент, тем качество прогноза лучше.

Коэффициент детерминации R^2 имеет прямо противоположный физический смысл. Чем ближе этот коэффициент к 1, тем точнее прогноз, чем ближе он к 0, тем прогноз хуже:

$$R^2=1-\frac{MSE}{Var(X)}=1-\frac{\Sigma(X-X)2}{\Sigma(X-M(X))2}=1-SNR^{-1}_{ср}, \quad (3.24)$$

Обе оценки могут взаимозаменяемо использоваться для оценки ряда. Они также достаточно точно отражают качество выбранного метода прогноза.

3.4.2.3 Смещения и коэффициенты переоценки и недооценки

Одним их важных критериев оценки являются смещения. Фактически это абсолютные значения ошибки, разделенные на положительные и отрицательные.

$$e^{+}=\begin{cases} e, \text{если } e>0 \\ 0, \text{если } e\le 0 \end{cases}, \quad (3.25)$$

$$e^{-}=\begin{cases} |e|, \text{если } e<0 \\ 0, \text{если } e\ge 0 \end{cases}, \quad (3.26)$$

Физический смысл смещения в смысле использования прогнозирования для резервирования сетевых ресурсов, например полосы пропускания, достаточно легко представить. Положительные смещения e^{+} - это случай, когда прогноз оказался меньше, чем действительная полоса, занятая трафиком ($X > \hat{X}$) и, фактически, показывает количество потерянных сообщений, то есть отражает недооценку. Отрицательные смещения e^{-} – это случай, когда прогноз оказался больше, чем действительная полоса, занятая трафиком ($X < \hat{X}$), то есть отражает переоценку.

Однако использование смещения в чистом виде недостаточно информативно, так как при больших значениях прогнозируемого ряда ошибки также будут иметь очень большое значение. Поэтому предлагается .использовать относительное значение этих смещений, или, по-другому, коэффициенты переоценки и недооценки:

$$K^{+}=\frac{\Sigma e+}{\Sigma X} \quad (3.27)$$

$$K^{-}=\frac{\Sigma e-}{\Sigma X}, \quad (3.27)$$

Применение данных коэффициентов в качестве критериев оценки очень важно, так как зачастую смещения в какую-либо одну сторону гораздо менее болезненны, чем в другую. Так, например, недоиспользование полосы пропускания (переоценка) трафика менее опасно, чем перегрузка (недооценка), когда реальный трафик оказался больше, чем был спрогнозирован предсказателем и часть трафика должна быть поставлена в очередь на узле, обрабатывающем его, а впоследствии, возможно, сброшена.

3.4.2.4 Средняя абсолютная ошибка в процентах

Средняя абсолютная ошибка в процентах MAPE (Mean Percentage Absolute Error) довольно простая оценка, которая отражает насколько, в среднем, ошибается выбранный метод прогнозирования:

$$\text{MAPE}=\frac{\Sigma|e|/X}{n}\ 100\%, \qquad (3.29)$$

где n - количество наблюдений. Очевидно, что чем меньше этот показатель, тем модель прогнозирования лучше.

3.5 Сравнение различных методов прогнозирования

Сравнение методов прогнозирования будем проводить для временного ряда, исследованного в главе 2. Напомним, что этот временной ряд представляет собой количество сообщений протокола SIP, принятых в единицу времени. Каждое сообщение представляет собой сообщение-запрос, или в терминологии протокола SIP, метод. Это объясняется тем, что именно методы создают нагрузку на узлы обработки сообщений SIP, а сообщения- ответы являются всего лишь их следствием.

Одним из важных моментов при прогнозировании является выбор размера данных или тренировочного участка, на основе которого будет осуществляться моделирование. Как было сказано ранее, модели авторегрессии относятся к типу моделей с короткой памятью, а FARIMA и другие модели - к типу моделей с длинной памятью. Соответственно оптимальный размер тренировочного участка у этих двух типов моделей будет отличаться. В [57] доказано, что для моделей второго класса оптимальным размером тренировочного участка является 100 отсчетов ряда, в то время как для моделей первого класса - достаточно 20 отсчетов. Однако следует учитывать, что увеличение длины тренировочного участка приводит к увеличению нагрузки процессора, производящего вычисления, что противоречит одному из сформулированных выше критериев оценки методов предсказания.

Следовательно, учитывая последнее утверждение, а также тот факт, что модели второго класса не могут использоваться в качестве on-line предсказателя, можно все дальнейшие исследования проводить с тренировочным участком длины в 20 отсчетов исходного ряда.

Исследуемый временной ряд состоит из 100 элементов, 20 из которых будут использоваться в качестве "истории" для прогнозирования, так что фактический прогноз будет осуществляться для элементов с 21-го по 100-й.

Предполагается, что история будет обновляться с каждым отчетом. Все параметры оценки считаются согласно формулам, приведенным выше.

В качестве основных методов прогнозирования выберем следующие:

- из моделей с короткой памятью выберем AR(1) (параметр авторегрессии AR=1) и ARIMA(0,1,1) (параметры интегрируемости и скользящего среднего d= MA= 1, параметр AR=0) как самую простую и самую "сложную" (в вычислительном плане) модель;
- из моделей с длинной памятью - FARIMA (1,d,1) (параметр интегрируемости *d* для данного трафика равен 0,2), как модель, которая лучше всего подходит для прогнозирования самоподобного трафика [57];
- метод прогнозирования MMSE;
- метод статического прогнозирования по среднему значению ряда.

Для каждого из методов построим прогноз исследуемого ряда на 80 отчетов. Ниже приведены рисунки, на которых изображены исходный ряд и прогнозируемые ряды при различных методах.

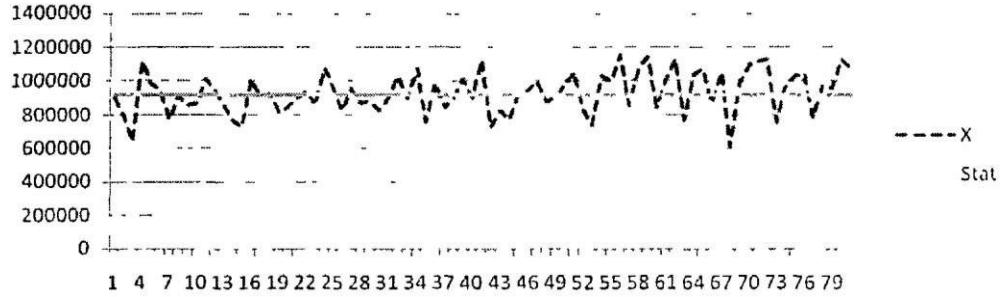


Рис. 3.1 - Исходный ряд X и ряд, спрогнозированный по среднему значению Stat

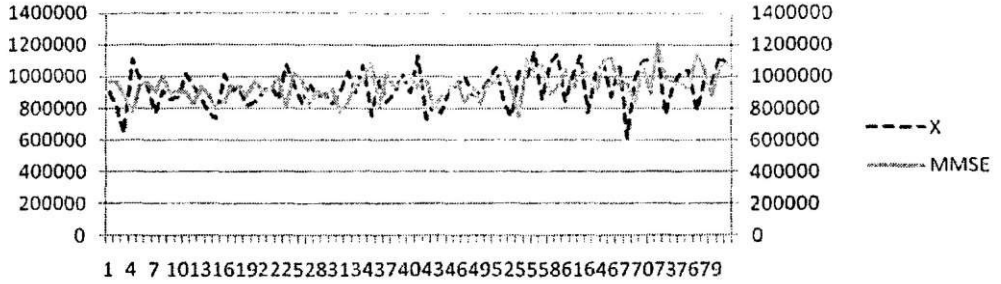


Рис. 3.2 - Исходный ряд X и ряд, спрогнозированный по методу MMSE

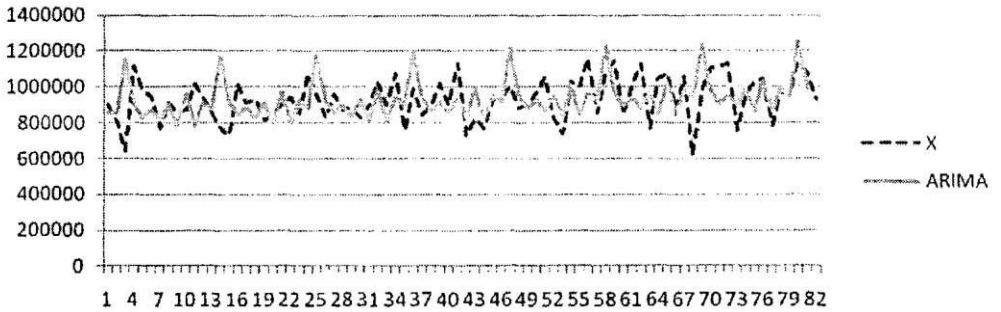


Рис. 3.3 - Исходный ряд X и ряд, спрогнозированный по методу ARIMA(0,1,1)

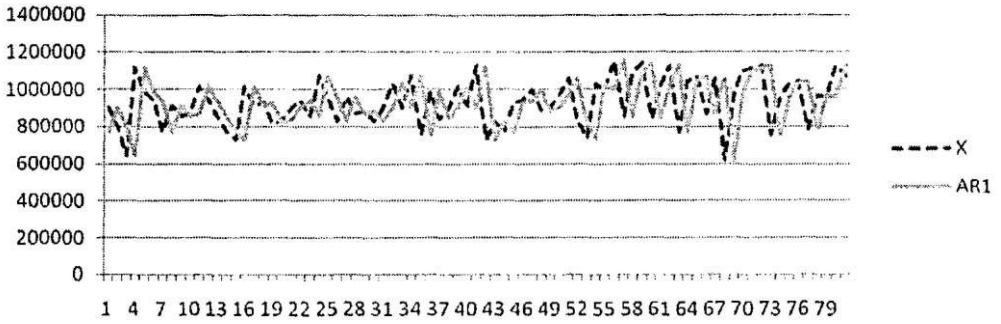


Рис. 3.4 - Исходный ряд X и ряд, спрогнозированный по методу AR(1)

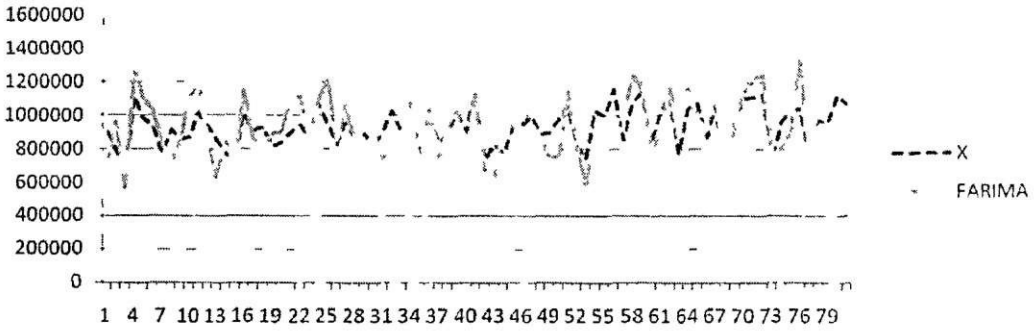


Рис. 3.5 - Исходный ряд X и ряд, спрогнозированный по методу FARIMA(l,d,l)

Далее, на основании ошибки прогноза $e = X - \hat{X}$ вычислим значения основных критериев оценки качества прогнозирования - коэффициентов детерминации, переоценки, недооценки и среднюю абсолютную ошибку. Результаты вычисления сведены в табл. 3.1.

Табл. 3.1 - Сравнение различных методов предсказания

Параметр	**Метод прогнозирования**				
	Stat	**AR(1)**	**ARIMA(0,1,1)**	**MMSE**	**FARIMA**
R^2	**0**	**0,477269**	**0,668002**	**0,661927**	**0,783652**
K^+	**0,247713**	**0,073457**	**0,061635**	**0,048101**	**0,050021**
K^-	**0,261019**	**0,077433**	**0,061871**	**0,033086**	**0,031384**
MAPE,%	**46,2873**	**15,9114**	**14,2973**	**13,6644**	**13,4085**

Проанализируем каждый показатель в отдельности.

3.5.1 Сравнение коэффициентов детерминации

Коэффициент SNR прямо пропорционален дисперсии ошибки. Чем ближе он к 1, тем больше дисперсия, чем ближе к 0 - дисперсия ошибки меньше.

Связанный с ним коэффициент детерминации R^2 показывает, насколько выбранный метод прогнозирования отличается от прогнозирования по среднему значению. И чем ближе этот коэффициент к 0, тем ближе качество прогноза к прогнозу по среднему, а чем ближе к 1 - тем прогноз лучше. Из этого очевидно, что у метода Stat коэффициент R^2 равен 0.

Следующим по точности прогноза является метод AR(1). Методы ARIMA и MMSE различаются на доли процента и опережают метод AR(1) на 19%. Однако они уступают методу FARIMA на 12 %.

Наибольший коэффициент, равный 0,783652, имеет метод FARIMA. Это подтверждает сделанное ранее утверждение о том, что метод FARIMA является самым точным для описания и прогнозирования самоподобных процессов. Однако, учитывая, что метод FARIMA не может быть использован для прогнозирования в реальном времени, самыми оптимальными с точки зрения величины коэффициента детерминации являются методы ARIMA и MMSE.

3.5.2 Сравнение смещений

Если коэффициент SNR является мерой дисперсии, то коэффициенты переоценки и недооценки являются мерой абсолютной ошибки. Определенный ранее физический смысл определяет важность каждого коэффициента. Очевидно, что K^+, отражающий доля потерянных пакетов, является более важным критерием оценки, нежели коэффициент K^-, показывающий долю "лишнего" трафика в прогнозе.

Напомним, что мы имеем дело с трафиком протокола SIP, который сильно зависит от задержек и потерь пакетов в сети. Время установления соединения является важным параметром, который регламентируется оператором связи. Задержки и потери пакетов могут значительно увеличить это время, а, могут даже привести к невозможности установления соединения.

В настоящее время отсутствуют нормы на величину задержки, джиттера и потерь пакетов для сообщений SIP. Поэтому необходимо оценить, насколько существенны данные потери.

Для начала стоит отметить, что исследуемый трафик протокола SIP представляет собой поток сообщений-запросов. За редким исключением, каждое сообщение-запрос должно быть подтверждено получателем с помощью

сообщения-ответа. В случае потери сообщения-запроса, оно ретранслируется отправителем. Например, если рассматривать метод INVITE, то в случае потерь, он будет ретранслироваться каждый раз по истечении определенного таймера:

$$A = 2^n \ T_t, \qquad (3.30)$$

где T_1 - таймер, отражающий примерное время передачи сообщения от источника к получателю и обратно, численно равный 500 мс (в соответствии с [70]), а n — 0,1,2,...,6 - счетчик потерь, увеличивается каждый раз при ретрансляции сообщения. Только в случае, если все 7 сообщений потеряны, соединение не будет установлено.

Учитывая, что вероятность потери единичного сообщения при использовании MMSE (метод с наименьшим коэффициентом $К^+$) составляет 5 %, можно определить вероятность того, что соединение не будет установлено, которая равняется вероятности того, что будут потеряны все 7 ретранслируемых сообщений:

$$P_{MMSE}=p_1 \cdot p_2 \cdot A \cdot p_t=0{,}05^7=7{,}8 \cdot 10^{-10} \approx 0$$

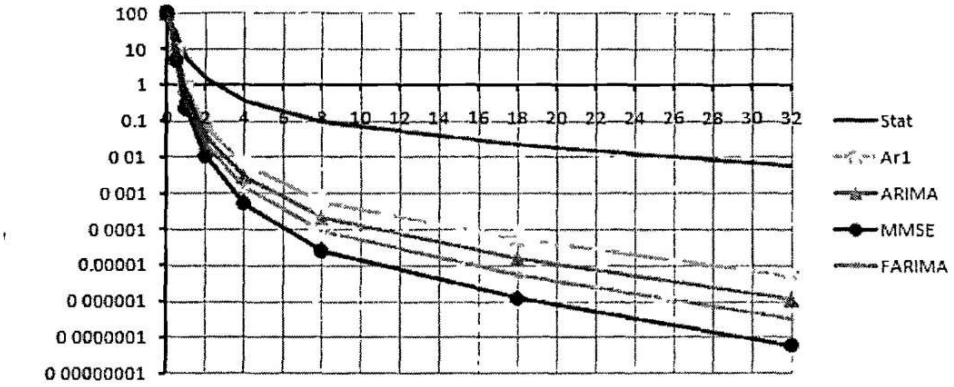
Это значит, что при использовании метода прогнозирования MMSE, несмотря на то, что может значительно увеличиться время установления соединения - вплоть до 32 секунд, все вызовы буду установлены. В то же время при использовании метода статического задания полосы пропускания (метод с наибольшим коэффициентом) вероятность неудачного вызова равна:

$$P_{STAT} = 0{,}25^7 = 6 \cdot 10^{-5} \approx 0$$

Отсюда можно сделать вывод, что потери, отражаемые в коэффициенте недооценки $К^+$, несущественны при оценке вероятности невозможности установления соединения и могут отразиться лишь на увеличении длительности установления вызова.

Как было сказано ранее, операторы обычно регламентируют длительность установления вызова. Помимо этого существуют международные стандарты, устанавливающие ограничения на этот параметр [69, 71]. Поэтому необходимо знать с какой вероятностью может увеличиваться длительность установления соединения при использовании различных предсказателей.

На рис. 3.6 показана зависимость вероятности задержки установления соединения от длительности установления соединения для различных методов прогнозирования. Максимальное значение - 32 секунды, равняется времени, по истечении которого вызов будет сброшен. Ось ординат представляет собой логарифмированную шкалу значений вероятности в процентах.



Рис, 3,6 - Вероятность задержки установления соединения

Из рис. 3.6 видно, что практически у всех методов вероятность того, что соединение будет устанавливаться дольше 2 секунды, практически равна 0. Это полностью удовлетворяет требованиям [69, 71], в которых максимальное значение длительности установления соединения колеблется в пределах от 3 до 8 секунд.

Несмотря на самый большой коэффициент детерминации, метод FARIMA показал худший результат при сравнении показателей недооценки. Из рис. 3.6 видно, что самым "быстрым" (с точки зрения задержки установления соединения) методом предсказания является MMSE.

Оценка по коэффициенту переоценки K^- методы FARIMA и MMSE являются наиболее оптимальными. В случае, если эти методы прогнозирования будут использоваться для управления ресурсами сети, они будут завышать оценку всего лишь на 3% от действительного значения. Для сравнения, метод статического задания будет переоценивать трафик на 26%.

3.5.3 Сравнение средних абсолютных ошибок

Интерпретация оценки MAPE вполне очевидна. Она показывает, насколько в среднем ошибается данный метод прогнозирования. С точки зрения этой оценки самыми оптимальными являются методы FARIMA и MMSE, при этом последний уступает первому всего лишь на 0,2%. Методы AR1 и ARIMA незначительно проигрывают первым двум (1-2%), в то же время метод прогнозирования по среднему значительно уступает своим конкурентам.

3.6 Выводы

Из результатов сравнения численных оценок различных методов прогнозирования видно, что абсолютно все рассмотренные методы превосходят по качеству метод прогнозирования по среднему значению по всем

использованным критериям. Самыми оптимальными с точки зрения большинства критериев являются методы MMSE и FARIMA. Несмотря на лучшие показатели коэффициента детерминации, метод FARIMA уступает методу MMSE в коэффициенте недооценки. К тому же, как было отмечено выше, с точки зрения аналитических критериев, метод FARIMA не может ,быть использован в качестве on-line предсказателя, а одним из главных достоинств метода MMSE является то, что он позволяет строить прогноз на основании истории о трафике в режиме реального времени.

Метод MMSE также удовлетворяет всем остальным аналитическим критериям. Он обладает высокой точностью и адаптивностью, что подтверждается численными критериями; он достаточно прост в реализации [89], что позволит реализовывать его в устройствах без увеличения их вычислительных ресурсов.

От других методов прогнозирования, использованных при сравнении, его отличает одно - у метода MMSE нет модели, на которой он бы основывался, а это значит, что он не может быть использован для генерирования искусственного трафика. Для таких целей наиболее подходит модель FARIMA. Однако, несмотря на это, метод MMSE является наиболее оптимальным методом для прогнозирования самоподобного трафика протокола SIP.

Разработка улучшенного метода борьбы с перегрузками в сети протокола SIP

4.1 Требования к методу борьбы с перегрузками

В первой главе диссертации были сформулированы основные недостатки существующего метода борьбы с перегрузками в сети на базе протокола SIP. Для устранения этих недостатков, а также с учетом новых возникших задач, были сформулированы требования к разрабатываемым механизмам управления перегрузками [74]:

- механизм управления перегрузками (УП) должен стараться поддержать максимальную пропускную способность сервера, но не допускать его перегрузки;
- в случае перегрузки одного элемента в сети SIP механизм должен стараться минимизировать отрицательное влияние переброса нагрузки на другие серверы;
- настройка работы данного механизма на серверах должна быть сведена до минимума или совсем исключена;
- серверы, реализующие данный механизм, должны иметь возможность работать с серверами, его не реализующими; чем больше элементов поддерживают данный механизм, тем больше должна быть пропускная способность сети;
- механизм должен эффективно работать в сети с потенциально опасными элементами (например, Интернет);
- сигнализация перегрузки должна однозначно означать только перегрузку, двусмысленность сигнализации недопустима;
- механизм должен иметь возможность регулировать интенсивность поступающей нагрузки, а не просто запрещать/разрешать трафик от вышестоящего элемента сети; механизм не должен допускать перераспределения нагрузки на другие перегруженные серверы;
- одновременно он не должен ограничивать перераспределение трафика на незагруженные серверы;
- механизм должен поддерживать работу с неограниченным множеством вышестоящих серверов;
- механизм должен работать между серверами в разных сетевых доменах (SIP сетях);
- механизм не должен накладывать ограничения на существующие механизмы приоритезации и маршрутизации вызовов внутри SIP;

I сервера;

- механизм должен явно сигнализировать о том, когда вышестоящему элементу сети следует повторно посылать сообщение (в

особенности это касается сообщений протокола SIP при использовании протокола TCP на транспортном уровне);

- механизм должен нормально функционировать в случаях, если другой SIP сервер не в состоянии обмениваться сообщениями (из-за перегрузки или сетевого сбоя);
- механизм должен стараться минимизировать количество передаваемой сигнальной информации;
- механизм не обязательно должен защищать от злонамеренных I DoS или DDoS атак;
- механизм должен явно указывать к чему относится перегрузка - к определенному IP адресу, доменному имени или S1P-URI;
- механизм должен учитывать, какие сообщения более приоритетны в обслуживании на основании важности с точки зрения предоставления сервиса (обслуживание сообщений уже установленных вызовов важнее, чем новых);
- в сети, где не все устройства реализуют данный механизм, не должно возникать диспропорций распределения нагрузки между элементами;
- механизм должен обеспечивать стабильность сети; в случае изменения общей нагрузки на сеть, при прочих равных условиях, эта нагрузка должна равномерно распределяться между элементами сети;
- должна быть возможность отключать использование данного механизма для конкретных направлений (SIP серверов) для целей безопасности;
- должна обеспечиваться работа механизма в схемах, состоящих из балансировщика нагрузки и нескольких серверов, на которых он эту нагрузку распределяет.

4.2 Описание обобщенной модели УП

С учетом вышеизложенных требований была разработана обобщенная модель реализации механизма УП [75]. Модель описывает два взаимодействующих сервера - сервер-отправитель и сервер-получатель. Механизм УП направлен на защиту сервера-получателя. Для этой цели между двумя серверами реализована обратная связь (ОС). В каждом из серверов выделены реализующие данную модель компоненты, представленные в виде выполняемых ими функций. Компоненты серверов могут быть сгруппированы в подсистемы по типу выполняемых задач, подсистема SIP отвечает за обработку сообщений протокола SIP, присутствует в каждом SIP сервере. Каждая функция передает определенные результаты своего выполнения на вход другой функции. Подробное описание компонентов и их взаимодействия изложено ниже.

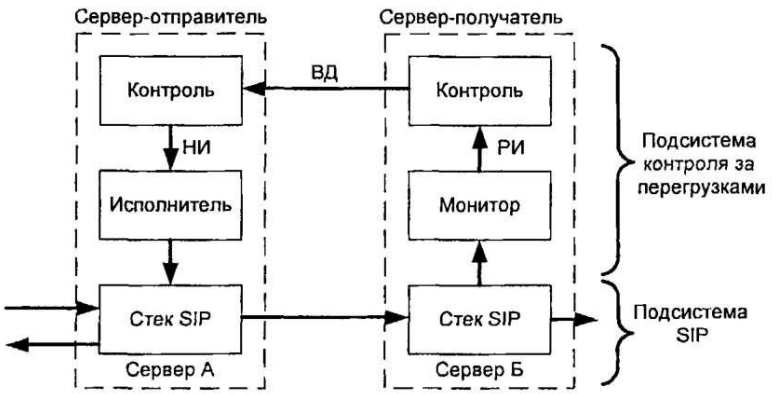


Рис. 4.1 - Обобщенная модель УП

Основными компонентами обобщенной модели УП являются:

1. *Стек SIP* - программная реализация протокола SIP. Функция, отвечающая за обработку вызовов. Является объектом защиты от перегрузок.

2. *Монитор* - функция, ответственная за измерение текущей загрузки стека SIP на приемной стороне. Результаты измерения (РИ) в виде отчетов передаются функции контроля.

3. *Контроль* - функция, реализующая алгоритм защиты от перегрузок. Используя полученные отчеты (РИ) как входную функцию, определяет наступление состояния перегрузки и необходимые изменения (НИ) нагрузки для оптимальной загрузки стека SIP сервера-получателя. Полученные выходные данные (ВД) передаются аналогичной функции контроля сервера-отправителя.

4. *Исполнитель* - функция, реализующая алгоритм действий, описанных в необходимых изменениях (НИ), путем передачи инструкций стеку SIP исходящего сервера А. Например, исполнитель может сообщить •требуемую сервером Б интенсивность сообщений. Исполнитель также должен определять поведение сервера при превышении порогов, определяемых НИ (кэшировать/сбрасывать/перенаправлять вызовы).

Тип выходных данных определяется алгоритмом управления перегрузками (например, алгоритм УП по потерям, интенсивности и др.) и другими параметрами. Выходные данные позволяются серверу А подстроить интенсивность передаваемого трафика в соответствии с требованиями сервера-получателя.

На рис 4.1 изображены две функции контроля - на передающей и на приемной стороне. Однако не обязательно присутствие обеих, так как необходимые изменения могут высчитываться и на стороне получателя и передаваться напрямую исполнителю сервера А.

Функция монитора должна иметь возможность различать несколько источников информации и иметь возможность сообщать результаты измерения РИ для каждого конкретного сервера-отправителя. В то же время функция исполнителя должна иметь возможность принимать НИ от нескольких

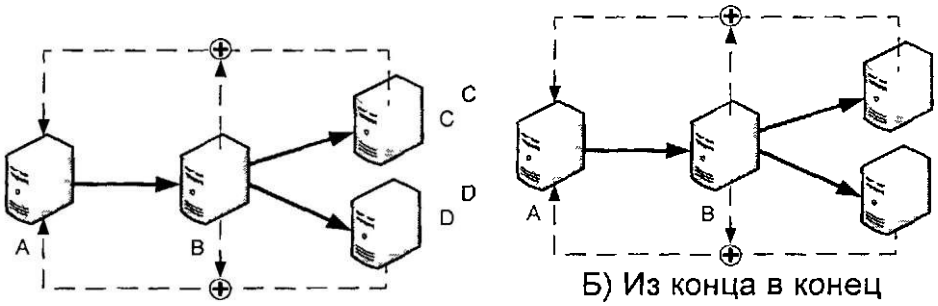
серверов-получателей и корректировать работу стека SIP соответственно для каждого направления.

Перед тем, как приступить к более подробному описанию функций подсистемы УП, необходимо рассмотреть несколько аспектов работы данной модели. Это важно для того, чтобы понять насколько данная модель вписывается в существующие топологии сетей SIP.

4.2.1 Способы взаимодействия сервера-отправителя и сервера-Получателя

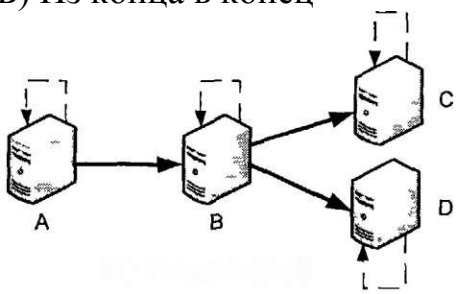
Обычно сообщение SIP проходит несколько серверов, прежде чем достигнет своего конечного получателя. Следовательно, возникает проблема определения оптимального места размещения компонентов подсистемы УП, в особенности Монитора и Исполнителя. В протоколе SIP определяется несколько способов взаимодействия между сетевыми элементами. В соответствии с ними основными способами работы элементов УП могут быть:

- пошаговый
- "из конца в конец"
- локальный



А) Пошаговый

Б) Из конца в конец



В) Локальный

Рис. 4.2 - Способы взаимодействия серверов в сети SIP

Идея пошагового способа взаимодействия заключается в организации связей между серверами, непосредственно обменивающимися трафиком. Например, Исполнитель расположен на сервере, стоящем сразу перед сервером, на котором располагается Монитор. На рис. 4.2 изображены 3 звена контроля - A-B, B-C, B-D. Информация, полученная от соседнего сервера, не передается другим серверам. Однако набор необходимых изменений влияет на его собственные результаты измерения, которые, в свою очередь, влияют на выходные данные, передаваемые следующему серверу. Пошаговый способ взаимодействия очень прост в реализации и легко масштабируется на сети различных масштабов. Одним из преимуществ является отсутствие необходимости общаться с далеко удаленными серверами для уменьшения нагрузки, что может быть нежелательно при наличии нескольких промежуточных доменов. Также при данном способе взаимодействия нет необходимости передавать и обрабатывать большие объемы информации. Недостатком данного метода является необходимость реализации компонентов УП на каждом из двух соседних серверов на пути следования сообщения SIP. То есть в случае, если на сервере B не реализованы компоненты УП, то возможности осуществлять управление перегрузками между A и D нет.

Способ "из конца в конец" использует звено обратной связи, проходящее через весь маршрут передачи сообщения SIP. Выходные данные от всех промежуточных элементов агрегируются на источнике и он использует их для соответствующих НИ. Одним из недостатков данного способа является недоиспользование ресурсов, которое может возникнуть в случае перегрузки одного из промежуточных серверов. Так, например, при перегрузке сервера D, сервер A уменьшает всю нагрузку к серверу B, поскольку априори не знает, какие именно вызовы попадут на D, а какие на C, что приводит к недоиспользованию ресурсов сервера C. Другим недостатком данного метода является необходимость обрабатывать большое количество информации одним сервером (сервер A). Данный способ может применяться в случае, если путь прохождения трафика на определенном участке заранее известен отправителю.

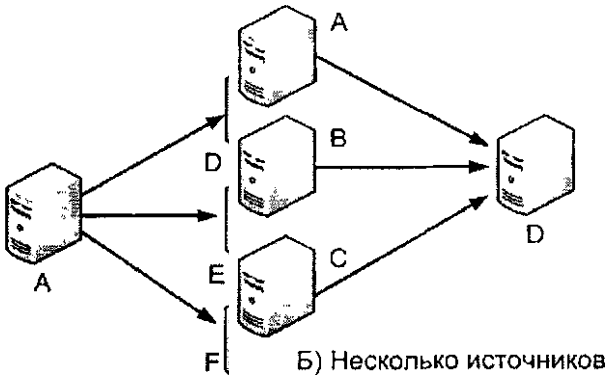
В случае локального способа взаимодействия Монитор и Исполнитель находятся на одном сервере. Основная идея данного способа основывается на предположении о том, что серверу проще сбросить вызов, а не обрабатывать его. Например, проще сбросить вызов, инициированный входящим сообщением INVITE, и не получать больше повторно посланных сообщений. Данный способ может использоваться совместно с другими способами и является дополнительной степенью защиты.

4.2.2 Базовые топологии сети SIP

Ниже описаны основные топологии, возможные в сети SIP. Каждая из них подразумевает свои особенности при борьбе с перегрузками.

Действительные топологии, встречающиеся в реальной практике, являются комбинациями нижеописанных топологий.

Балансировщик нагрузки (рис. 4.3а). В данной схеме задача сервера А не допустить перегрузки серверов О, Е и Р. В случае, если один из серверов ШЕЛ[7] перегружен, сервер А может перенаправить нагрузку на другие серверы. Также, если А может однозначно определить перегрузку на Б/Е/Р, тогда он может сообщить о перегрузке своему предыдущему серверу.



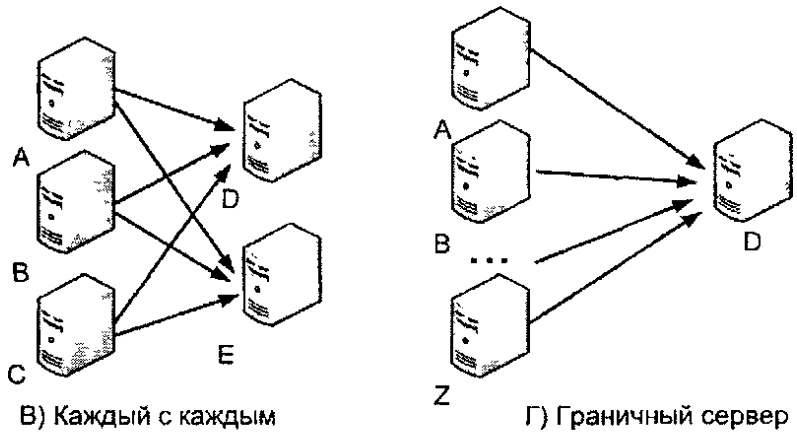


Рис. 4.3 - Базовые топологии в сети SIP

Несколько источников (рис. 4.3б). Сервер D получает трафик от нескольких источников A/B/C. Каждый из источников может создавать различную нагрузку. В случае перегрузки сервера D, он должен решить - кому из серверов и насколько следует изменить нагрузку.

"Каяедый с каждым" (рис. 4.3в). Данная топология является комбинацией двух предыдущих. Для эффективного контроля в такой топологии каждый сервер должен уметь дифференцировать загрузку, порождаемую каждым из своих соседей, и соответственно извещать их об этом.

Граничный сервер (рис. 4.3г). В данной топологии серверы A-Z обычно представляют собой абонентские терминалы. Естественно, они не могут уменьшить интенсивность трафика. В данном случае действенным является способ уменьшения загрузки сервера путем сброса части вызов. Это может быть сделано с помощью сообщения "503", с использованием таймера "Retry-

After". Это поможет при обычной интенсивности вызовов, однако не может предотвратить перегрузку в случае значительного увеличения потока вызовов (например, в случае DoS-атаки). Требования по управлению перегрузками у граничного сервера отличаются от требований остальных топологий и могут потребовать разработку отдельного механизма.

4.2.3 Приоритезация

В реальных условиях перегрузки зачастую возникают в короткие промежутки времени и обусловлены появлением новых источников трафика. В состоянии перегрузки сервер-получатель может распределять свои ресурсы между всеми серверами-отправителями одинаково. Однако в реальной практике такая ситуация крайне нежелательна. Рассмотрим ее на примере ниже.

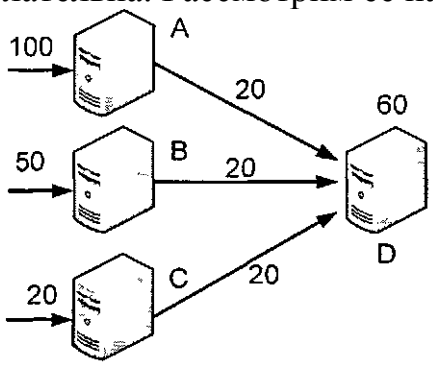


Рис. 4.4 - Ограничение нагрузки без приоритезации

Допустим, что в рассматриваемом примере контролируемый сервер D (рис. 4.4) имеет пропускную способность 60 вызовов в секунду. Нагрузку он получает с трех других серверов A/B/C. В момент перегрузки входящая нагрузка на серверы A/B/C будет равна 100/50/20 вызовов в секунду (выз/с) соответственно. В отсутствии приоритезации сервер D ограничит нагрузку от своих вышестоящих соседей до 20 вызовов, то есть разделит свои ресурсы поровну между A/B/C. Однако в результате этого серверам A и B придется отбросить нагрузку в 80 и 30 выз/с соответственно, в то время как серверу C не придется сбрасывать вызовы. Для устранения данного недостатка были введены два основных критерия приоритезации: по провайдерам и по пользователям.

Приоритезация по провайдерам означает, что все провайдеры получат одинаковые ресурсы сервера. Каждый провайдер может быть представлен как одним, так и несколькими SIP серверами. Равная приоритезация по пользователям означает, что каждый пользователь будет иметь одинаковую вероятность установить вызов в случае перегрузки. В большинстве случае приоритезации по пользователям является наиболее приемлемой, например, в

случае перегрузки при реализации услуги ТВ-голосования. Иначе пользователи одного провайдера получат большую возможность выиграть/проголосовать.

В случае, если заранее известны все вышестоящие источники, реализовать приоритезацию по провайдерам достаточно просто, так как серверу-получателю достаточно разделять свои ресурсы одинаково между ними.

Обеспечение приоритезации по пользователям означает, что сервер будет разделять свои ресурсы пропорционально между получателями в соответствии с входящей нагрузкой. Для обеспечения приоритезации по пользователям предлагается ввести в схему петлю прямой связи (рис. 4.5).

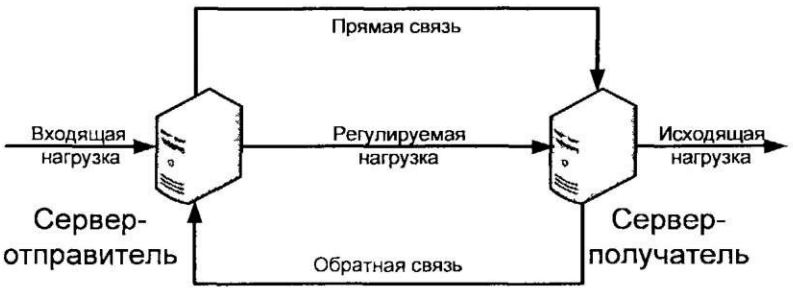
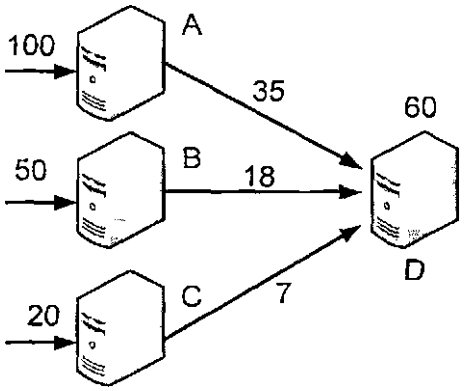


Рис. 4.5 - Управление приоритезацией с помощью прямой связи

Прямая связь используется для передачи информации об измеренной интенсивности входящей нагрузки. Получив эти данные, сервер-получатель сможет выделять свои ресурсы пропорционально нагрузке, поступающей на серверы-отправители. Так, при введении прямой связи распределенная нагрузка для примера, изображенного на рис. 4.4, будет выглядеть так, как показано на рис. 4.6.



Рис, 4.6 - Ограничение нагрузки с приоритезацией по пользователям

В данном случае каждый из серверов А, В и С получает количество ресурсов сервера Б, пропорциональное своей входящей нагрузке.

4.2.4 Метрики измерения качества управления перегрузками

Качество механизма УП может быть определено несколькими метриками. Основным показателем качества механизма УП является полезная пропускная способность сервера. В идеале БГР сервер должен использовать всю свою мощность во время перегрузки. То есть, если его производительность равна X вызовам в секунду, то механизм УП должен обеспечивать работу сервера именно при такой нагрузке, даже если потенциальная нагрузка на сервер намного больше.

Другим параметром качества является задержка, вносимая сервером в обмен сигнальными сообщениями Б1Р при установлении вызовов. Она должна находится на приемлемом уровне на протяжении всего периода перегрузки. В состоянии перегрузки сервер должен отвечать на поступающие сообщения почти с той же скоростью, что и в отсутствии перегрузки.

Быстрота реакции и стабильность также являются важными критериями качества. Механизм УП должен максимально быстро реагировать на скачки нагрузки как вверх, так и вниз. Также он не должен приводить к колебаниям поступающей нагрузки.

Есть несколько дополнительных критериев для измерения качества работы механизма УП:

1. *Приоритезация.* Определяет, как и каким образом осуществляется приоритезация вызовов на сервере.
2. *Самоограничение.* Сервер, удовлетворяющий данному критерию, должен уметь использовать локальный механизм контроля и реагировать на перегрузки стека SIP.
3. *Изменение топологии.* Механизм должен уметь быстро адаптироваться к изменениям в топологии сети SIP.
4. *Точки мониторинга.* Определяет, сколько и какие именно данные нужны Монитору для осуществления контроля.
5. *Точность контроля.* Определяет, насколько точно сможет механизм изменить поступающую на него нагрузку.

4.2.5 Алгоритм работы функции монитора

Функция монитора - первый этап в предлагаемом механизме УП. Основная ее задача - измерение текущей загрузки сервера-получателя и формирование на выходе данных, которые впоследствии будут использованы функцией Контроля. В задаче измерения нагрузки очень важным моментом является выбор минимальной единицы управления. Разные типы SIP-сообщений имеют различный вес с точки зрения УП. Например, в случае новой сессии принятие нового сообщения INVITE повлечет за собой обмен еще как минимум шестью сообщениями. Поэтому за единицу управления удобнее всего

принимать SIP-сессию. УП на основании сессий требует соответствующих метрик, таких как, например, интенсивность сессий. Для того, чтобы эти метрики получить, необходим полный контроль над сообщениями, то есть надо следить за первым и последним сообщениями в пределах одной сессии. Например, для расчета метрики сервер может считать, сколько сессий началось и сколько завершилось за определенный интервал времени. В случае сигнализации SIP - сессия начинается с сообщения INVITE и заканчивается обычно сообщением BYE. Однако, благодаря особенностям протокола SIP, сообщение BYE может пройти другим путем, нежели сообщение INVITE. Кроме того, существуют другие типы сообщений-запросов, которые могут проходить через сервер. По этим причинам расчет метрик, основанных на сессиях, является очень сложным и практически невыполнимым.

Поэтому в [77] предлагается использовать альтернативный подход при подсчете сессий - контроль начатых сессий. В нормальных условиях поступивший запрос на вызов в итоге приводит к обслуживанию вызова. Например, сервер может считать количество принятых сообщений INVITE $N_{INVITE}^{принятых}$ за период T_m. Тогда значение скорости обслуживания сессий определяется выражением:

$$\mu = N_{INVITE}^{принятых} / T_m$$

Еще одним важным параметрам является текущее количество вызовов, обслуживаемых сервером. Необходимо заметить, что это количество не равно количеству сообщений INVITE, пришедших на сервер, так как в очереди на обслуживание стеком протокола SIP могут находиться и другие сообщения (не INVITE) от уже установленных сессий, их количество обозначим через N_{NOINV} . Формула для расчета текущего количества обслуживаемых вызовов имеет вид:

$$N_{общ} = N_{INVITE} + N_{NOINV}/(L_{ВЫЗ}-1), \qquad (4.1)$$

где N_{INVITE}, N_{NOINV} - текущее количество сообщений в очереди на обслуживание стеком SIP;

$L_{ВЫЗ}$ - средняя длина вызова, определяемая средним количеством сообщений за один вызов.

Таким образом, параметр N_{INVITE} равен количеству новых вызовов, а отношение $N_{NOINV}/(L_{ВЫЗ}-1)$ - количеству вызовов, уже обслуживаемых сервером. Параметр $L_{ВЫЗ}$ рассчитывается на основании количества вызовов, обслуженных сервером $N_{INVITE}^{обслуженных}$ протяжении интервала измерения:

$$L_{ВЫЗ} = N_{INVITE}^{обслуженных} / N_{INVITE}^{принятых}$$

Используя рассмотренный в главе 3 метод MMSE, сервер-получатель может оценить поступающую нагрузку и построить ее прогноз на ближайшее

будущее. Результаты прогноза могут использоваться для более точного определения величины нагрузки и корректировки выходных данных функции Монитора. Обозначим прогнозированное количество сообщения INVITE следующим образом:

$$N_{пр}=f(N_{INVITE}^{\text{принятых}})$$

То есть прогноз определяется как функция от текущего количества принятых сообщений INVITE. Таким образом, с учетом прогноза формула 4.1 принимает вид:

$$N_{общ}= N_{INVITE}+ N_{NOINV}/(L_{ВЫЗ}-1)+N_{ПР}, \qquad (4.2)$$

Этот параметр и будет являться результатом работы функции Монитора. Эти результаты передаются следующей функции.

4.2.6 Механизмы уп функции контроля

Как было указано ранее, Контроль - функция, которая использует результаты измерений Монитора и определяет наступление состояния перегрузки и необходимые изменения (НИ) нагрузки для оптимальной загрузки стека Б1Р сервера-получателя. Существует три основных механизма работы данной функции. Ниже приведены несколько примеров реализации основных механизмов [77]. Каждый из примеров, обладает входным параметром, к которому он привязывается. Это может быть как загрузка процессора, так и длина очереди сообщений стека протокола Б1Р, однако и то и другое находится в прямой зависимости от описанного в предыдущем пункте параметра $Ы_{об1ч}$. Поэтому можно считать, что других входных параметров для данной функции не требуется.

4.2.6.1. Механизм УП по абсолютному значению скорости

Основная идея данного механизма - ограничение скорости запросов от вышестоящего сервера. В случае перегрузки БГР-сервер сигнализирует об этом своего вышестоящего соседа, разрешая ему передавать максимум X запросов в секунду. Каждый из вышестоящих соседей, в соответствии с критериями приоритезации, может иметь свои ограничения по скорости.

Ограничения по скорости для каждого из вышестоящих серверов определяются на основании специального алгоритма (например, на основе процессорной нагрузки). Сервер должен следить за всеми своими соседями и

быть готовым изменять ограничения по скорости для каждого из них в случае, если появляется новый сосед или старый перестанет передавать трафик. При этом сервер-получатель должен учитывать входящую нагрузку каждого из вышестоящих серверов.

В случае равного распределения ресурсов между вышестоящими серверами, процесс ограничения выглядит следующим образом. Сначала, каждому вышестоящему соседу назначается порог интенсивности *X/N,* где *N* - общее количество соседей, а X - максимальная производительность SIP-сервера в запросах в секунду. Затем в процессе работы контролируемый SIP-сервер может уменьшить ограничение для своего соседа, в случае если

интенсивность трафика от него больше *X/N,* или увеличить, если интенсивность меньше *X/N.* Если сумма интенсивностей от всех соседей не превышает собственную производительность сервера, то со временем профиль нагрузки стабилизируется и сервер начинает работать под оптимальной загрузкой. В данном случае сервер должен обязательно следить за интенсивностью нагрузки от всех своих соседей.

Примером реализации алгоритма УП по абсолютному значению скорости может являться механизм *rate-abs* (от англ. rate absolute). Основная его идея состоит в определении, что задержка пребывания сообщений в очереди на обслуживание стеком SIP не превышает определенного допустимого порога (D_H). Обозначим период контроля T_c, в течение которого сервер-получатель измеряет общее количество обслуживаемых им сессий $N_{общ}$, и скорость поступления **новых** сессий μ. По истечении этого периода он передает серверу-отправителю допустимый порог интенсивности:

$$\lambda^{k+1}=\mu^{k}(1-(d^{k}_{q}-D_B)/T_c), \qquad (4.3)$$

где μ^k - интенсивность нагрузки от сервера-отправителя в конце *k*-го периода контроля;

d^k_q - примерная оценка длины очереди, определяемая как:

$$d^{k}_{q}=N_{общ}/\ \mu^{k}$$

4.2.6.2. Механизм УП по относительному значению скорости

Данный механизм подразумевает, что серверы-отправители ограничат интенсивность трафика на X процентов по просьбе сервера-получателя. То есть X процентов трафика будет либо перенаправлено, либо отклонено.

Достоинством данного механизма является то, что нет необходимости контролировать трафик от каждого отдельного соседа. Достаточно лишь

контролировать общую загрузку системы. Каждый раз новый процент потерь X_t определяется на основании текущего процента потерь (X_{t-1}), текущей загрузки системы L_{t-1} и требуемой загрузки системы L_t:

$$X_t=f(X_{t-1}, L_{t-1}, L_t)$$

Например, если загрузка системы приближается к 90 %, а текущий процент потерь, сообщаемый паре серверов-отправителей, составляет 50%, тогда сервер может решить увеличить допустимые потери до 55 %, тем самым добиться уменьшения общей загрузки до 80 %.

Данный механизм должен достаточно быстро реагировать на изменения. То есть, если период измерения загрузки системы $T€(t_1,t_2)$ и сервер установил процент потерь X=10% в момент t_1, а действительная нагрузка увеличилась на 20%, то к моменту t_2 нагрузка увеличиться на 10 процентов. Таким образом, данный механизм не сможет обеспечить защиту от перегрузок в случае скачков нагрузки.

В качестве примера реализации данного алгоритма рассмотрим механизм *rate-occ* (от англ. rate occupancy). Данный механизм основывается на загрузке процессора. Загрузка определяется процентом времени, потраченным процессором на обработку SIP сообщений, за фиксированный временной промежуток. В конце каждого периода контроля T_c текущая загрузка процессора $\boldsymbol{p_к}$ сравнивается с допустимой $\boldsymbol{p_B}$. Сервер-получатель передает отправителю параметр f^{k+l}, коэффициент изменения нагрузки относительно текущей интенсивности:

$$f^{k+l}=\begin{cases} fmin, если\ \text{ф}f < f \\ 1, если\ \text{ф}f > 1 \\ \text{ф}f, \text{во всех других случаях} \end{cases}, \qquad (4.4)$$

где f_{min} - нижний порог коэффициента изменения нагрузки, который служит целью не допустить полного прекращения трафика от сервера-отправителя, обычно очень маленькое число (0.01;0.02). Параметр f^k - значение коэффициента изменения нагрузки в начале интервала контроля T_c. Множитель коэффициента изменения f^k-$\phi^к$ определяется по формуле:

$$\boldsymbol{\phi^к=min(p_B/p_k;\ \phi_{max})},$$

где $\boldsymbol{\phi_{max}=max(\phi^{k-1};\phi^{k+1})}$ максимальное значение множителя за два предыдущих интервала. Значение $\boldsymbol{\phi_{max}}$ на первых двух шагах может быть любым.

Таким образом, коэффициент изменения нагрузки /* принимает свои значения в интервале (0.01;1) тем самым регулируя интенсивность трафика сервера-отправителя от минимального значения ($0.01\cdot\lambda^k$) до максимального значения (λ^k).

4.2.6.3. Оконный механизм УП

Основная идея данного механизма заключается в том, что сервер-отправитель может послать только определенное количество сообщений до того, как получит хотя бы одно сообщение от получателя. Каждый сервер-отправитель имеет собственное окно на каждое направление и соответственно ограничивает количество посылаемых сообщений. Счетчик сообщений окна увеличивается с каждым посланным запросом и уменьшается с каждым полученным ответом.

Ключевым моментом данного механизма является размер окна. У каждого отправителя этот размер сначала устанавливается по умолчанию и изменяется в соответствии с запросом сервера-получателя. В случае уменьшения окна до нуля, сервер перестает посылать сообщения до тех пор, пока не будет получено достаточное количество сообщений-подтверждений.

Оконный механизм похож на механизм ограничения по скорости в том, что защищаемому серверу необходимо контролировать всех своих вышестоящих соседей. Однако, в отличие от механизма ограничения по скорости, данный механизм является самоограничивающим и в нормальной ситуации не приведет к переполнению буферов. Передача сообщений отправителем ограничивается подтверждениями получателя. Переполнение буфера может случиться, если возникает большое количество новых источников за короткий промежуток времени. В такой ситуации механизм управления по скорости не предотвратит перегрузку, в то время как оконный механизм не позволит отправителю передавать сообщения, если он не получает на них ответа.

Размер окна может быть специально выставлен в 0. В таком состоянии отправитель должен будет специально информирован, чтобы опять начать передачу. Однако получатель не может послать ни одного ответного сообщения, при отсутствии запросов. Для этого получатель должен использовать какой-нибудь другой механизм (например, посылка сообщения OPTIONS).

Рассмотрим несколько примеров реализации оконного алгоритма.

- Алгоритм дискретного размера окна

Основная идея алгоритма дискретного размера окна, *win-disc* (от англ. window-discreet), заключается в измерении допустимого размера окна за дискретные промежутки времени T_c. По истечении этого промежутка сервер-получатель производит оценку количества новый сессий, которых он может принять в следующем интервале T_c, удерживая при этом собственную загрузку и размер очередей в SIP-стеке на приемлемом уровне. Допустим, что сервер-получатель информирует N своих вышестоящих соседей в конце *к-го* интервала контроля T_c о допустимом размере окна . Формула расчета размера окна выглядит следующим образом:

$$w_i^k = \left(\mu^k T_c + \mu^k D_B - N_{общ}^k\right) \cdot a_i^k \qquad (4.5)$$

В данной формуле параметр $\mu^к$ - интенсивность поступления сессий в конце k-го интервала, $D_в$ - разрешенный размер длины очереди в стеке SIP. Слагаемое $\mu^к\ T_c$ - означает среднее количество сессий, которое сервер может обработать, $\mu^к\ D_в$ - среднее количество сессий, которые могут быть поставлены в очередь, $N^к_{общ}$ - измеренное количество сессий (формула 4.2) в конце *k-го* шага. Соответственно множитель в скобках представляет собой общее максимальное количество сессий, которое может принять сервер-получатель. Для того, чтобы получить индивидуальные значения для каждого сервера-отправителя введен весовой коэффициент a^k_t,обозначающий вес каждого из серверов-отправителей в конце *k-го* шага.

Сумма всех весов $\sum_t a^k_t = 1$.

Начальное значение окна может иметь любое значение, большее нуля. Например, зная примерно расчетное значение интенсивности поступающих сессий $\mu_{расч}$, размер первого окна может быть вычислен так:

$w_0 = \mu_{расч}\ T_c$.

- Алгоритм непрерывного размера окна

Второй оконный алгоритм, *win-cont* (от англ. window-continious), является непрерывным во времени. В отличие от основанного на временных интервалах алгоритма *win-disc,* данный алгоритм основан на событиях. Он постоянно изменяет размер окна, как только сервер имеет возможность принять другое количество сессий. Для этого измеряется максимальное количество сессий, которое может обслужить сервер $N^{max}_{сообщений} = \mu \cdot D_B$, где D_B - опять же максимально допустимый размер очереди, а μ - мгновенное значение интенсивности. В каждый момент времени вычисляется разница между текущим $N_{общ}$ и максимальным $N^{max}_{сообщений}$ количеством сессий. Эта измеренная разница (размер окна) сообщается после каждого проведенного измерения текущего количества сессий. Измерение параметра $N_{общ}$, в зависимости от количества требований, может проводиться либо после приема каждого сообщения, либо только после приема сообщений INVITE. Также в зависимости от требований размер окна может сообщаться только при превышении значением окна определенного порога (например, при пороге = 2 сессиям размер окна будет передаваться, только если он равен 2 или более сообщениям). Текущий размер окна определяется по формуле:

$$w_i = \mu \cdot D_B \cdot a, \qquad (4.6)$$

- Алгоритм самоуправляющегося размера окна

Автономный оконный алгоритм *win-auto* (от англ. window-autonomous) автоматически подстраивает размер окна таким образом, что число

поступивших сессий никогда не должно превысить число обслуженных сессий. С каждым ретранслированным сообщением INVITE размер окна уменьшается на единицу, а с каждым новым сообщением INVITE - увеличивается на единицу. Таким образом, в случае перегрузки сервер- получатель будет уменьшать размер окна до тех пор, пока не сможет нормально и быстро (до истечения таймера ретрансляции) обслуживать получаемые сообщения.

В зависимости от состояния размер окна определяется выражением:

- $w_t^0 = W_0 > 0$;
- $w_t^t = w_t^t$ -1 для каждого ретранслированного сообщения INVITE;
- $w_t^t = w_t^t$ +1 после обработки нового сообщения INVITE.

4.2.7 Алгоритмы реализации ограничений исполнителем

Исполнитель может использовать несколько способов ограничения интенсивности источника нагрузки [76]:

1. **Процентное регулирование.** Данный способ обычно используется совместно с механизмом управления по скорости. В случае УП по абсолютному значению скорости при необходимости ограничить скорость до λ^t и при входящей интенсивности λ , сервер-отправитель должен ограничить $(1-\lambda^t/\lambda)$ процентов запросов. При УП по относительному значению скорости процент ограниченного трафика заранее известен.
2. **Leaky bucket («Дырявое ведро») и Token bucket («Ведро с жетонами»).** Два классических способа ограничения скорости, используемые в оборудовании передачи данных. Оба способа ограничения скорости могут использоваться в механизмах УП по скорости. В случае УП по абсолютному значению скорости, например, информация, поступающая от сервера- получателя (ВД) может без изменения использоваться как входная информация для данных схем.
3. **Автоматическое разряжение вызовов.** Данная технология очень часто используется в сетях телекоммуникаций и является более консервативным механизмом ограничения скорости, чем техники Leaky/Token bucket (то же самое при глубине «ведер» = 0). По принятии первого вызова запускается таймер разряжения, до истечения которого все остальные поступившие вызовы сбрасываются. После истечения этого таймера принимается следующий вызов и сразу после его приема таймер перезапускается. Данный механизм не допускает всплесков нагрузки на выходе и генерирует нагрузку строго определенной интенсивности.
4. **Оконное регулирование.** В данном случае сервер-отправитель следит за размером окна, определенным сервером-получателем. Вызов будет отправлен только в случае, если окно еще не полностью заполнилось.

Вызов, отброшенный данными способами, может быть либо поставлен в очередь, либо сброшен моментально, либо переадресован. Для того чтобы сглаживать локальные всплески трафика необходимо кэширование вызовов,

либо использование механизмов с «ведрами». Размер кэш памяти или «ведра» должны быть выбраны таким образом, чтобы не вносить существенную задержку во время установления вызова.

4.3 Реализация уп в протоколе SIP

Для общего случая предположим, что направление трафика - одностороннее, от серверов-отправителей до сервера-получателя. Исходя из этого, на практике может существовать два основных метода реализации УП в протоколе SIP [76]:

- через параметр заголовка Via;
- через пакет событий.

Использование параметра заголовка Via дает следующие преимущества:

- заголовок Via достаточно маленький и не создает серьезной нагрузки на процессор;
- передача ВД может вестись очень часто, с каждым сообщением-ответом;
- с данным заголовком ВД будут передаваться всем нужным серверам базовыми средствами протокола; нет необходимости специально отслеживать статус своих соседей;
- ВД не передаются неактивным серверам и автоматически начинают передаваться новым серверам-отправителям;
- SIP-сервер может контролировать, кому именно передавать информацию об УП, а кому нет.

Использование пакета событий подразумевает, что серверы- отправители подписываются на соответствующее событие и впоследствии долучают уведомления о ВД в форме сообщений NOTIFY. Данный подход имеет следующие достоинства:

- информация ВД доставляется отдельно от основной сигнализации SIP - это позволяет отделить модули УП от основного стека SIP, чтобы не допустить перегрузки;
- в данном методе получатель будет продолжать передавать обновления даже неактивным соседям, после устранения перегрузки получатель может сам уведомить их об этом;
- SIP сервер посылает сообщения NOTIFY всем подписанным серверам только тогда, когда это необходимо; чтобы не потерять эти сообщения необходимо применить дополнительные механизмы защиты на время перегрузки.

Однако ряд недостатков делает метод пакета событий малопригодным для использования на реальной сети:

- использование данного метода создает дополнительную сигнальную нагрузку на сеть;

- подсистеме УП необходимо поддерживать подписки со всеми своими соседями. Новая подписка должна быть осуществлена до посылки первого сообщения; серверы могут подписываться в момент первичной загрузки, однако для этого потребуется отдельная защита от лавинного рестарта;
- в связи с тем, что каждое обновление передается в отдельном сообщении, данный метод не подходит для частых обновлений.

В связи с указанными недостатками реализации механизма УП через пакет событий, далее будем рассматривать лишь реализацию механизма с помощью параметров заголовка Via. Ниже рассмотрим пример реализации данного механизма.

4.3.1 Реализация метода УП с помощью параметра заголовка Via

В качестве примера реализации улучшенного метода УП предлагается использовать новый параметр 'oc' (overload control - управление перегрузками) в заголовке Via сообщений SIP. Помимо самого параметра 'oc' предлагается ввести три других вспомогательных параметра - 'oc_accept', 'oc_validity' и 'oc_income'.

4.3.1.1 Параметр 'oc_accept'

Любой сервер, поддерживающий данный стандарт, должен добавлять этот параметр в заголовок Via при передаче запроса следующему соседу. Это позволит получателю понять, что его вышестоящий сосед поддерживает данный метод УП. Сервер должен удалять этот параметр из последнего заголовка Via во всех ответных сообщениях, передаваемых назад.

4.3.1.2 Создание параметров 'oc' и 'oc_validity'

SIP сервер может сообщать своим вышестоящим соседям о перегрузках с помощью данного параметра заголовка Via. Этот параметр должен вставляться в самый верхний (при наличии нескольких заголовков) заголовок Via (заголовок, принадлежащий вышестоящему соседу) и не должен добавлять в любые другие заголовки Via. Самый верхний заголовок определяется после того, как сервер удалил свой собственный заголовок Via из полученного сообщения.

Поскольку этот заголовок удаляется каждым сервером на пути следования ответа, параметр 'oc' будет действовать только между одной парой

соседних серверов и не будет передаваться дальше. Таким образом, данный механизм является пошаговым с точки зрения способов взаимодействия 81P-серверов.

Данный параметр может использоваться во всех типах ответов включая промежуточные, успешные и неуспешные.

Помимо параметра 'oc' сервер должен добавить параметр 'oc_validity', который определяет время в миллисекундах, в течение которого будут действительны ограничения, налагаемые параметром 'oc'.

В некоторых ситуациях параметры 'oc' могут передаваться дальше по пути следования ответного сообщения, когда это обосновано с точки зрения топологии сети. Эти параметры могут передаваться только в ответных сообщениях, так как используются как звено обратной связи.

4.3.1.3 Определение значений параметра 'oc'

Значения параметра определяются работающим алгоритмом УП. В соответствии с механизмом УП он может принимать численные значения, означающие процент потерь, желаемую скорость или размер окна.

4.3.1.4 Обработка параметра'oc'

Сервер должен сохранять значение параметра 'oc' вместе с адресом сервера, от которого он его получил и значением параметра 'oc_validity '. Каждый раз при получении нового параметра 'oc' старое значение перезаписывается и таймер, определяемый параметром 'oc_validity', перезапускается. В случае, если значение таймера становится равным 0, значение 'oc' удаляется.

4.3.1.5 Использование значения параметра 'oc'

Каждый раз при наличии готового к отправке сообщения запроса, SIP-сервер должен проверить значении параметра 'oc' для этого сервера. Если оно еще в силе, он должен применить соответствующий механизм ограничения интенсивности нагрузки.

4.3.2 Самоограничение

В случае, если вышестоящий сервер перестал отвечать на запросы, серверу-отправителю необходимо прекратить отправку сообщений и изредка посылать пробные сообщения на перегруженный сервер. Как только сервер-получатель ответит на запрос - сервер-отправитель может восстановить нормальную интенсивность передачи сообщений. Данный механизм позволит не допустить перегрузки сервера, который и так не может ответить ни на один запрос.

4.3.3 Механизм приоритезации

В большинстве случаев приоритезация определяется каждым конкретным сервером индивидуально. Для реализации приоритезации по пользователям петлю прямой связи можно реализовывать с помощью параметра 'oc_income'. Однако в данном случае параметр необходимо вставлять в заголовки Via сообщений-запросов, посылаемых от серверов- отправителей. Значения параметра 'ocjncome' могут означать, например, скорость поступления запросов /л, на г'-й сервер-отправитель. Значение параметра 'oc_income' должно храниться сервером-получателем для каждого из i серверов-отправителей отдельно. Параметр 'oc_income', полученный в сообщении-ответе, должен игнорироваться.

4.3.4 Отбой запросов

Сервер-получатель должен отказывать в обслуживании всем новым запросам, передавая сообщение 503. В случае, если загрузка сервера дошла

до 100 процентов, сервер должен использовать ответы 503, даже если он сигнализирует о своей перегрузки с помощью параметра 'oc'.

Синтаксис

oc=”oc”[0-100]

oc_validity=” oc_validity”[мс]

oc-accept=” oc-accept”

oc_income=[0-1000]

4.3.5 Обратная совместимость

Новый механизм должен работать в сетях, состоящих не только из серверов, поддерживающих его. В частности пошаговый механизм работает только между парой серверов, которые его поддерживают. При этом остальные серверы на пути следования сообщений могут его не поддерживать. Соответственно чем больше таких серверов в сети, тем сеть более защищена от перегрузок. Поскольку из множества вышестоящих соседей не все могут поддерживать данный механизм УП, поэтому нагрузку от таких серверов следует регулировать с помощью передачи сообщения 503, если это возможно, иначе просто сбрасывать эти сообщения.

4.4.1 Сравнение различных алгоритмов УП между собой

Напомним, что существуют три основных критерия оценки качества работы механизмов УП - это полезная пропускная способность сервера, задержка и быстрота реакции. Также есть несколько второстепенных критериев - способность осуществления приоритезации, адаптивность к изменению топологии сети, точность контроля и другие.

Среди рассмотренных алгоритмов *win-auto* является самым простым для протокола SIP, так как не требует никаких входных данных. С другой стороны алгоритм *rate-occ* меньше всего использует возможности протокола SIP, так как основывается исключительно на загрузке процессора. Другие алгоритмы необходимо искусственно привязывать к протоколу SIP, используя входные параметры.

В [77] показано, что алгоритмы, учитывающие длину очереди сообщений ('*win-disc, win-cont* и *rate-abs),* показывают гораздо лучшие результаты, чем алгоритмы, учитывающие загрузку процессора (*rate-occ*) в плане использования процессорного времени, то есть в случае перегрузки процессор сервера-получателя полностью загружен (полезная пропускная способность сервера максимальна) и каждое полученное сообщение гарантированно обслуживается. К тому же механизмы, учитывающие длину очереди, имеют большую точность, нежели алгоритм *rate-occ,* который регулирует интенсивность УП исходя из загрузки ЦПУ. Таким образом, с точки зрения критерия максимизации полезной пропускной способности сервера под постоянной нагрузкой и точности настройки, предпочтительными являются все алгоритмы, кроме *rate-occ.*

Регулировка в каждом алгоритме может быть дискретной, основанной на временных интервалах *(win-disc* и *rate-abs, rate-occ)* и непрерывной,

основанной на событиях (*win-cont* и *win-auto*). Обычно регулировка, основанная на событиях, имеет меньше параметров регулировки и подстройки и является более точной. Однако при достаточно небольших интервалах управления T_c, разница между двумя способами управления становится незначительной.

Было выявлено, что все алгоритмы, кроме *win-auto,* хорошо адаптируются к изменениям нагрузки. Относительно критерия приоритезации, в особенности приоритезации по пользователям, лучшим является алгоритм *rate-occ,* так как в нем этот метод реализуется по умолчанию. Также все другие алгоритмы (кроме *win-auto*) в случае введения прямой связи могут обеспечить равнодоступность по пользователям. Тем не менее, по простоте реализации алгоритм *win-auto* значительно превосходит другие алгоритмы, так как не требует никаких входных параметров.

4.4.2 Сравнение эффективности различных методов УП

В предыдущем пункте было произведено сравнение различных алгоритмов УП между собой. Несмотря на то, что однозначно нельзя сказать какой из алгоритмов самый эффективный, можно сделать вывод, что оконные алгоритмы, в частности алгоритм *win-cont,* удовлетворяет всем критериям. Он быстро адаптируется к изменениям нагрузки, эффективно использует пропускную способность сервера, прост в реализации благодаря небольшому количеству входных параметров, с помощью прямой связи реализует критерий приоритезации, является более точным в плане регулировки входящей нагрузки.

Сравним теперь эффективность работы трех различных методов УП: - 503 - существующий метод УП (с помощью передачи сообщений "503")

способности сервера соответствует 72-м успешно обслуженным вызовам в одну секунду с учетом того, что время установления соединения (от момента передачи сообщения INVITE до приема сообщения АСК) не превышает 10 секунд. Нагрузка, создаваемая серверами-отправителями, также нормализуется относительно максимальной пропускной способности сервера таким образом, что одна условная единица нагрузки соответствует интенсивности трафика равной 72 вызова в секунду. На протяжении всего эксперимента нагрузка возрастала с постоянной скоростью от 0 до 10 условных единиц. При этом были сделаны следующие допущения:

- продолжительность вызова распределена экспоненциально и ее среднее значение равно 30 секундам;
- серверы-отправители не имеют ограничений по производительности (не допускается их собственная перегрузка).

Все эти допущения сделаны для упрощения тестовой схемы и при этом они не влияют на результаты моделирования. Результаты сравнения трех выбранных методов приведены на рис. 4.8

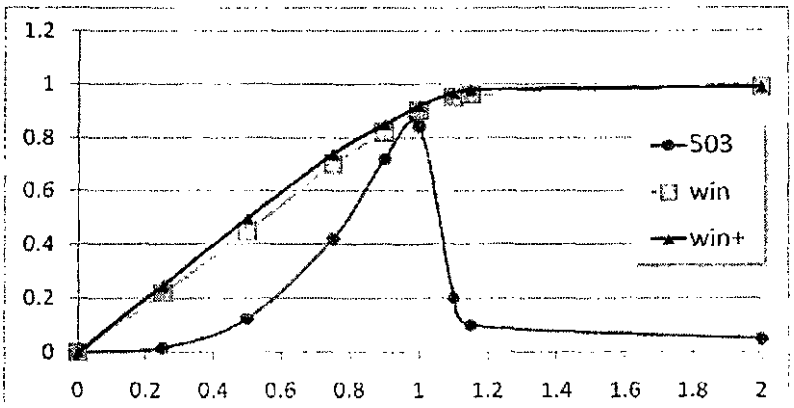


Рис. 4.8 - Результаты тестирования

На рисунке 4.8 ось абсцисс соответствует поступающей на сервер нагрузке, нормированной таким образом, что единица шкалы равна нагрузке, создаваемой трафиком с интенсивностью 72 вызова в секунду. Ось ординат

отражает полезную пропускную способность сервера, притом что 1 - это максимальная загрузка сервера.

Из приведенного рисунка видно, что при использовании метода 503 полезная пропускная способность сервера при увеличении входящей нагрузки сначала растет, а потом при достижении максимального значения загрузки сервера резко падает. Это резкое падение является коллапсом сервера, когда все его ресурсы уходят на отбой вновь поступающих вызовов. При этом у сервера нет возможности обслужить даже существующие вызовы. Его полезная пропускная способность почти равна нулю.

С другой стороны методы *win* и *win+* позволяют поддерживать максимальную загрузку процессора на протяжении всего тестирования. Благодаря оконному механизму вышестоящие серверы по информации от сервера-получателя ограничивают интенсивность исходящего трафика таким образом, что суммарная интенсивность трафика от всех серверов равнялась 72 вызовам в секунду.

Однако метод *win+* опережает своего ближайшего конкурента по одному важному критерию - времени установления соединения. При пересечении порога максимальной загрузки сервера, метод *win,* благодаря присущей ему инерционности, приводит к сбросу лишних сообщений и, следовательно, к последующий их ретрансляции.

В результате эксперимента было выявлено, что при использовании метода *win* количество ретранслированных сообщений на один вызов находится в основном пределах от 1 до 4 (рис. 4.9). В свою очередь 4 ретранслированных сообщения означают, что задержка установления соединения увеличится на $\wedge\ 2^{n-1} \bullet At = 7{,}5$ секунды *(n* – количество ретранслированных сообщений, *At* =500л/с - таймер, определенный в стандарте [70]), что при среднем времени установления соединения 7,5 секунд [8,9] приводит к суммарным 15 секундам установления соединения. К

тому же в реальной сети в период максимальной загрузки этот порог может достигаться много раз на протяжении нескольких часов, что в итоге приведет к значительному ухудшению качества предоставляемых сетью услуг вследствие увеличения задержки установления соединения.

4.5 Выводы

Существующий метод борьбы с перегрузками в сети SIP обладает целым рядом недостатков. В данной главе описан новый метод борьбы с перегрузками, разработанный с целью устранить недостатки существующего метода. В основе его лежит механизм обратной связи, с помощью которого серверы-получатели могут контролировать поступающую на них нагрузку. Предложенная обобщенная модель управления перегрузками может быть использована в сетях SIP с различной топологией и различными требованиями по приоритезации трафика. Модель состоит из нескольких основных компонентов, реализующих определенную функцию.

Одним из основных преимуществ от применения нового метода является устранение эффекта коллапса сервера, когда в случае перегрузки все ресурсы уходят на отбой вызовов и сервер не в состоянии обслуживать даже текущие вызовы. Такое поведение присуще существующему методу борьбы с перегрузками и может привести от перегрузки в одном узле к полной неработоспособности сети SIP. При использовании нового метода контролируемый сервер даже в случае перегрузки продолжает обслуживать максимально возможное для него количество вызовов.

Однако, в случае достижения сервером максимальной загрузки, неизбежны дополнительные задержки установления соединения. Для устранения данного недостатка можно использовать модифицированный метод УП, учитывающий самоподобные свойства сигнального трафика протокола SIP. Эти особенности позволяют строить кратковременный прогноз интенсивности на основе исторических данных. Модифицированный метод УП использует предложенный в главе 3 предсказатель MMSE. Результаты работы предсказателя MMSE используются для корректировки входных данных функции мониторинга нового метода УП. Таким образом, модифицированный метод дает 80 % выигрыш в уменьшении задержки

Заключение

В данной магистерской диссертации было произведено исследование свойств сигнального трафика протокола SIP и разработан новый метод управления перегрузками в сети сигнализации SIP.

В результате систематизации существующих работ в области исследования трафика IP-коммуникаций было выявлено два основных подхода. Первый подход рассматривал весь трафик как поток вызовов на исследуемую систему. Было показано, что данный подход обладает рядом недостатков, которые приводят к некорректной оценке производительности системы. Второй подход основывается на том факте, что все сообщения передаются в виде пакетов. При использовании этого подхода трафик медиаданных и трафик сигнализации рассматриваются отдельно. При исследовании трафика медиаданных много раз выявлялись его самоподобные свойства. В дальнейшем эти свойства были использованы для оптимизации механизмов обеспечения QoS для этого типа трафика. Трафик сигнализации исследовался значительно реже и существующие на данный момент исследования обладают рядом недостатков, ограничивающих сферу применения предлагаемых решений.

К тому же было показано, что существующие методы управления перегрузками в сети SIP обладают рядом серьезных недостатков. Для устранения этих недостатков в данной работе было произведено следующее:

1. Был произведен сбор статистических данных трафика сигнального протокола SIP на действующей сети крупного оператора связи;
2. Проведен анализ трафика, показавший наличие в нем всех основных свойств самоподобия;
3. Оценка параметра Херста, характеризующего степень самоподобия трафика, показала, что его значения находятся в пределах $0.6 < H < 0.8$;
4. Проведено компьютерное моделирование работы четырех различных методов прогнозирования;
5. На основе восьми критериев был выбран оптимальный метод, позволяющий делать кратковременный прогноз трафика SIP;
6. Разработан новый механизм управления перегрузками в сети сигнализации SIP, учитывающий кратковременный прогноз трафика;
7. Проведено компьютерное моделирование, показывающее преимущество разработанного механизма.

Показано, что разработанный механизм управления перегрузками не только позволяет устранить все основные недостатки существующего метода, но и позволяет сократить время установления соединения на 80% по сравнению с механизмом, не учитывающим кратковременный прогноз.

Список литературы

1. Leland, W.E. On the self-similar nature of ethernet taffic (extended version) / W.E.Leland, M.S.Taqqu, W.Willinger, D.V.Wilson // IEEE/ACM Transactions of Networking - 1993. - Vol.2 №1. - P. 1-15.
2. Casilari, E. Modelling of Voice Traffic Over IP Networks/ E. Casilari, H. Montes, F. Sandoval// IEICE - Transactions on Information and Systems. - 2007. - Iss. 12. - P. 2886-2896.
3. Jiang, W. Analysis of On-Off Patterns in VoIP and Their Effcct on Voice Traffic Aggregation/ W. Jiang, H. Schulzrinne// Ninth International Conference on Computer Communications and Networks. - 2000. - P. 82-87.
4. Susanto, H. Examining Self-Similarity Network Traffic intervals [Электронный документ]./ H. Susanto, Byung-Guk Kim. - Режим доступа: www.eecs.tufts.edu/~hsusan0a/ predictlnterval.pdf - 08.12.2008
5. Сейфетдинов, Р. Р. Исследование и разработка методов анализа нагрузки сети ОКС №7 в сетях мобильной связи второго и третьего поколения: дис. ... канд. техн. наук / Р. Р. Сейфетдинов - ПГАТИ, 2006.
6. Росляков, А. В. Оценка сигнальной нагрузки на сеть ОКС №7 от интеллектуальных услуг на базе технологии CAMEL/ А. В. Росляков, А. В. Титов // Мобильные телекоммуникации. - 2007. - № 6-7. - с. 42-51.
7. Росляков, А. В. ОКС №7: Архитектура протоколы применение / А. В. Росляков // - М.: Эко-Трендз, 2008
8. De Marco, G. A Technique to Analyse Session Initiation Protocol Traffic / G. De Marco, G. Iacovoni// 11th ICPD. - 2005. - Vol.2. - P. 595 - 599.
9. He, Q. Analyzing the Characteristics of VoIP Traffic [Электронный документ] / Q. He. - Режим доступа: Iibrary2.usask.ca/theses/available/etd-07132007-120004/unrestricted/thesis.pdf - 08.12.2008
10. Летников, А. И. Разработка модели для анализа показателей качества функционирования сигнализации по протоколу SIP / А . И . Летников, В. А. Наумов// Электросвязь. - 2007. - №7. - С. 44 - 47.
11. Neame, T. Characterisation and modeling of Internet traffic streams [Электронный документ] / T. Neame. - Режим доступа: www.ee.unimelb.edu.au/multimedia/research/ cubin_TimNeame_Thesis.pdf 08.12.2008.
12. Шелухин, О. И. Фрактальные процессы в телекоммуникациях / О. И. Шелухин, А. М. Тенякшев, А. В. Осин - М.: Радиотехника, 2003. - 480 с.
13. Duffy, D. E. Analyzing telecommunications traffic data from working common channel signaling subnetworks / D. E. Duffy, A. A. Mcintosh, M. Rosenstein, W. Willinger // Interface Foundation of North America. - 1993. - Vol. 25.-P. 156-165.
14. Vemuri, A. Session Initiation Protocol for Telephones (SIP-T) / A. Vemuri, J. Peterson // IETF RFC 3372. - 2002.

15. Bearer independent call control protocol / ITU-T Recommendation Q.1901. - 2000.

16. Beran, J. Long-Range Dependence in Variable-Bit Rate Video Traffic / J. Beran, R. Sherman, M.S. Taqqu, W. Willinger //IEEE Transactions on Communications - 1995. - Vol. 43, № 2/3/4.

17. Росляков, А. В. Анализ статистических параметров нагрузки звена ОКС №7 / А. В. Росляков, С. В. Канарейкин // Электросвязь. - 2006. - №7.

18. Галкин, А. М. Анализ характеристик сетей NGN с учетом свойств самоподобия трафика / А. М. Галкин, О. А. Симонина, Г. Г. Яновский // Электросвязь. - 2007. - №12.

19. Ohta, M. Overload Control in a SIP Signaling Network / M. Ohta// ICISP apos - 2006. - P. 11-11.

20. Kang, H. J. SIP-based VoIP Traffic Behavior Profiling and Its Applications / H. J. Kang, Z.-L. Zhang, S. Ranjan, A. Nucci // MineNet - 2007. - P. 39-44.

21. Иевлева, Т. В. Обнаружение и предотвращение перегрузок оборудования Softswitch при регистрации SIP-телефонов / Т. В. Иевлева, С. В. Журавлев // Электросвязь - 2007. - №12.

22. Samorodnitsky, G. Stable Non Gaussian Random Processes: Stochastic Model swith Infinite Variance [Электронный документ] / G. Samorodnitsky , M. Taqqu. - Режим доступа: http://math.bu.edu/people/murad/stable-expanded.html - 08.12.2008

23. Park, K. SelfSimilar Network Traffic: An Overview [Электронный документ]/ K. Park, W. Willinger. - Режим доступа:www.cs.purdue.edu/nsl/intro-ss-chap.pdf - 08.12.2008

24. Schroeder, M. Fractal, Chaos, Power Laws / Manfred Schroeder - New York: W. H. Freeman, 1991.

56. Xue, F. Traffic Modeling Based on FAR1MA Models Similarity [Электронный документ]/ F. Xue, J. Liu, Y. Shu, L. Zhang. - Режим доступа: il.tju.edu.cn/publications/2001/Jc_2001.pdf - 08.12.2008.

57. Урьев, Г. А. Исследование фрактальных свойств потоков трафика реального времени и оценка их влияния на характеристики обслуживания телекоммуникационных: автореф. дис. ... канд. тех. наук: 05.12.13: защищена 22.03.07 / Г. А. Урьев; МГУС. - Москва, 2007. - 23 с.

58. Зюльков, И. А. Самоподобные свойства трафика систем с повторными вызовами / И. А. Зюльков // Вестник ВГУ. — 2002. - №1

59. Nogueira, A. Modelling Self-similar traffic through Markov Modulated Poisson Process over multiple time scales [Электронный документ]/ A. Nogueira, P. Salvador, R. Valadas, A. Pacheco. - Режим доступа: www.av.it.pt/~rv/Papers/hsnmc03.pdf- 08.12.2008.

60. Rosenberg, J. SIP: Session Initiation Protocol / J. Rosenberg, H. Schulzrinne, G. Camarillo etal. //IETF RFC 3261.-2002.

61. Кашин, М. М. Исследование свойств сигнального трафика протокола SIP/ М. М. Кашин, А. В. Росляков // T-Comm - Телекоммуникации и Транспорт. - 2009. - №5. - С. 26-29.

62. Кашин, М. М. Методы борьбы с перегрузками в сети SIP / М. М. Кашин // Инфокоммуникационные технологии. - №1, т. 9. - 2011. - С. 67-70.

63. Кашин, М. М. Метод борьбы с перегрузками в сети SIP на основе статистического анализа сигнального трафика / М. М. Кашин // Инфокоммуникационные технологии. - №2, т. 10. - 2011 - С. 65-69.

64. Кашин, М. М. Исследование характера сигнального трафика IP-коммуникаций / М. М. Кашин, А. В. Росляков // Технологии и средства связи. - 2009. - №2.-С. 18-19.

65. Кашин, М. М. Обеспечение качества обслуживания в сетях NGN / М. М. Кашин, А. В. Росляков // VII Международная конференция «Актуальные проблемы современной науки». - Самара, 2006. - С. 40-42.

66. Кашин, М. М. Анализ принципов построения и применения на сетях NGN гибких коммутаторов (softswitch) / М. М. Кашин, А. В. Росляков // XIII юбилейная Российск. научн. конф. проф.-препод, состава, научн. сотрудн. и аспир., ПГАТИ. - Самара, 2006. - С. 46-47.

67. Кашин, М. М. Методика расчета поступающей нагрузки в сетях следующего поколения NGN / М. М. Кашин, А. В. Росляков // VII Междунар. науч.-техн. конф. «Проблемы техники и технологии телекоммуникаций». - Самара, 2006. - С. 157-159.

68. Кашин, М. М. Модель QoS для сетей IMS / Кашин М. М., А. В. Росляков // Труды 3-го Междунар. форума «Актуальные проблемы современной науки». - Самара, 2007. - С. 39-40.

69. Кашин, М. М. Метод оценки параметров трафика сети NGN / Кашин М. М., А. В. Росляков // XIV Российск. научн. конф. проф.-преп. состава, научн. сотрудн. и аспир. ПГАТИ. - Самара, 2007. - С. 44.

70. Кашин, М. М. Статистический анализ сигнального трафика протокола SIP / Кашин М. М., А. В. Росляков // XV Российск. научн. конф. профессор.-препод. состава, научн. сотруд.. и аспир. ПГАТИ. - Самара, 2008. -С. 85-86.

71. Кашин, М. М. Задача исследования сигнального трафика в сетях IP-телефонии / М. М. Кашин // IX Междунар. конф. «Проблемы техники и технологии телекоммуникаций». - Самара, 2008. - С. 83-84.

72. Кашин, М. М. Статистический анализ сигнального трафика протокола SIP/ М. М. Кашин // Доклады 10-й Междунар. конф. «Цифровая обработка сигналов и ее применение» (DSPA). - Москва, 2008. - С. 235-238.

73. Кашин, М. М. Выбор метода прогнозирования сетевого трафика / М. М. Кашин, А. В. Росляков // Труды XVI Российск. научн. конф. профессор-препод, состава ПГУТИ. - Самара, 2009. - С. 103-104.

74. Кашин, М. М. Выбор метода прогнозирования сетевого трафика протокола SIP / М. М. Кашин // X Междунар. конф. «Проблемы техники и технологии телекоммуникаций». - Самара, 2009. — С. 103-104.

75. Кашин, М. М. Модифицированный метод управления перегрузками в сети SIP / М. М. Кашин, А. В. Росляков // XI Междунар. науч.-техн. конф. «Проблемы техники и технологии телекоммуникаций». - Уфа, 2010.-С. 124-126.

76. Кашин, М. М. Методы борьбы с перегрузками в сети сигнализации SIP / М. М. Кашин, А. В. Росляков // Труды XVII Российской научн. конф. профессор.-препод. состава, научн. сотруд. и аспир. ПГУТИ. - Самара, 2010. - С. 69-70.

77. Кашин, М. М. Модифицированный метод управления перегрузками в сети SIP / М. М. Кашин, А. В. Росляков // XVIII Российская научн. конф. профессор.-препод. состава, научн. сотруд. и аспир. ПГУТИ. - Самара, 2011. - С. 75.

78. Internet protocol // IETF RFC 791. - 1981.

79. Roach, A. Session Initiation Protocol (SIP)-Specific Event Notification / A. B. Roach // IETF RFC 3265. - 2002.

119 Jennings, C. Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks / C. Jennings, J. Peterson, M. Watson // IETF RFC 3325. - 2002.

120. Willis, D. Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts/ D. Willis, B. Hoeneisen // IETF RFC 3327. - 2002.

121. Rosenberg, J. An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing / J. Rosenberg, H. Schulzrinne // IETF RFC 3581. - 2003.

122. Карташевский, И. В. Исследование и разработка методов анализа непуассоновских моделей трафика мультисервисных сетей: автореф. дис. ... канд. тех. наук:05.12.13: защищена 24.12.2010 / И. В. Карташевский; ПГУТИ. - Самара, 2010,- 16.