

РЕАЛИЗАЦИЯ МЕТОДИКИ ОЦЕНКИ ОПЕРАТОРСКИХ СЕТЕЙ НА СООТВЕТСТВИЕ РЕКОМЕНДАЦИЯМ ITU-T Y.1540, Y.1541

С. А. Владимиров^{1*}, И. С. Алексеев¹, А. С. Воронов¹

¹ СПбГУТ, Санкт-Петербург, 193232, Российская Федерация
* Адрес для переписки: vlsa.vlcom@yandex.ru

Аннотация

Оказание услуг операторами связи на сетях абонентского доступа требует непрерывного мониторинга качества таких услуг. При увеличении объемов предлагаемых и оказываемых услуг, особенно с ростом абонентской базы операторские сети оказываются перед фактом того, что обеспечиваемое на сети качество не соответствует заявленному и поддерживаемому при начале эксплуатации сети. В некоторых случаях даже хорошо развитая техническая поддержка оператора связи не может определить «узкое место» в сети передачи данных при сбоях отдельных видов оказываемых услуг, связанных со спецификой применения на сети транспортных протоколов. Программный инструмент, разработанный авторами и предлагаемый к использованию на сетях, позволит не только быстрее находить «узкие места», но и оценивать сети с точки зрения соответствия рекомендациям ITU-T Y.1540, Y.1541. **Предмет исследования.** Фрагменты сетей передачи данных, включая сети абонентского доступа и сети агрегации услуг. **Метод.** Раздельная оценка сетей по протоколам TCP и UDP с использованием разработанных авторами программ.

Основные результаты. Разработаны программы оценки сетей передачи данных. Проведено пробное тестирование гетерогенных операторских сетей. **Практическая значимость.** Разработанное программное обеспечение позволяет использовать его в качестве основы для построения программно-аппаратных комплексов мониторинга сетей и услуг у операторов связи.

Ключевые слова

качество обслуживания (QoS), операторские сети, рекомендации ITU-T Y.1540 и Y.1541, время доставки пакета, вариация задержки доставки пакета.

Информация о статье

УДК 004.738, 621.391

Язык статьи – русский.

Поступила в редакцию 15.06.18, принятая к печати 03.09.18.

Ссылка для цитирования: Владимиров С. А., Алексеев И. С., Воронов А. С. Реализация методики оценки операторских сетей на соответствие рекомендациям ITU-T Y.1540, Y.1541 // Информационные технологии и телекоммуникации. 2018. Том 6. № 3. С. 52–64.

IMPLEMENTATION OF OPERATOR NETWORKS EVALUATION METHODOLOGY FOR COMPLIANCE WITH ITU-T RECOMMENDATIONS Y.1540, Y.1541

S. Vladimirov^{1*}, I. Alekseev¹, A. Voronov¹

¹ SPbSUT, St. Petersburg, 193232, Russian Federation

* Corresponding author: vlsa.vlcom@yandex.ru

Abstract—The provision of services by telecom operators on subscriber access networks requires continuous monitoring of the quality of such services. With the increase in the volume of offered and provided services, especially with the growth of the subscriber base, the operator networks are faced with the fact that the quality provided on the network does not correspond to the previously declared. In some cases, even well-developed technical support of the telecommunications operator can not determine the "bottleneck" in the data network in case of failures of certain types of provided services, related to the specific application of the network transport protocols. The software tool developed by the authors and offered for use on networks will allow not only to quickly find bottlenecks, but also to evaluate networks in terms of compliance with ITU-T recommendations Y.1540, Y.1541. **Research subject.** Fragments of data transmission networks, including subscriber access networks and service aggregation networks. **Method.** Separate evaluation of networks using TCP and UDP protocols using the programs developed by the authors. **Core results.** Programs for evaluating data networks have been developed. Testing of heterogeneous operator networks has been carried out. **Practical relevance.** The developed software makes it possible to use it as a basis for constructing software and hardware systems for monitoring networks and services from telecom operators.

Keywords—Quality of Service (QoS), operator networks, ITU-T recommendations Y.1540 and Y.1541, packet delivery time, packet delivery delay variation.

Article info

Article in Russian.

Received 15.06.18, accepted 03.09.18.

For citation: Vladimirov S., Alekseev I., Voronov A.: Implementation of Operator Networks Evaluation Methodology for Compliance with ITU-T Recommendations Y.1540, Y.1541 // Telecom IT. 2018. Vol. 6. Iss. 3. pp. 52–64 (in Russian).

Введение

В настоящее время наблюдается значительный рост количества абонентов разных операторских сетей. Практически многие пользователи сети Интернет уже не представляют себе существование в отрыве от подключения к сети. При этом некоторые абоненты периодически сталкиваются в ходе своей работы с некачественной работой сети оператора. Это говорит о том, что передаваемый операторами трафик хоть и имеет объявленные гарантии по качеству обслуживания (QoS) [1, 2], но не имеет должного инструментального контроля, и часто объявленные требования этих гарантий занижаются в связи с ростом абонентской базы и нагрузки на сеть. С учетом того, что виды трафика в зависимости

от подключенных пользователем услуг, бывают разными, и некоторые из них являются более важными с точки зрения очередности доставки [3], возникает проблема качества операторской сети: в случае перегрузок сети по трафику более приоритетные в плане доставки пакеты могут не достигать своего места назначения и быть отброшенными или приходить с большой задержкой, когда актуальность этого пакета потеряна полностью.

Протокол сетевого уровня – IP – не гарантирует, что все переданные пакеты вовремя достигнут определенного узла сети. Часть пакетов может быть повреждена, часть – повторно продублирована на передаче, часть – отброшена. Ко всему этому IP-пакеты могут доставляться разными путями, и создаются нарушения последовательности пакетов по порядку в точке прибытия. IP заботится лишь о целостности данных, вычисляя для каждого пакета контрольную сумму, которая записывается в заголовок.

Вышеперечисленные проблемы – это то, что способен ощутить на себе пользователь сетевых приложений, в особенности тех, что работают с трафиком реального времени – например, если говорить об IP-телефонии [3], то это искашение речи, наличие пропадания звука или возникновение эха, и как следствие – неразборчивость и ухудшение качества речи. Для пользователей IPTV – выпадение строк и фрагментов, искашение и замирание картинки, пропадание целых полей и кадров изображения [3, 4].

Одной из причин возникновения таких проблем может являться сложная иерархия сети оператора на уровне доступа. Если представить ситуацию, когда количество присоединяемых пользователей больше количества портов на коммутаторе, то решить такой вопрос можно подключением дополнительного коммутатора по принципу свободного порта. И через некоторое время может возникнуть такой же прецедент, решением которого будет абсолютно такой же способ. Таким образом, возникает дополнительный уровень подсети доступа, который будет располагаться от уровня распределения через несколько коммутаторов. Это приведёт к тому, что пользователь из этой подсети будет «рассматриваться» после всех тех, кто подключен к коммутаторам, стоящим выше него по уровню. В периоды большой нагрузки в сети такой абонент, пользующийся, например, услугой IP-телефонии, набрав номер, по истечении определенного времени может не дождаться ответа. Однако пользователи вышестоящих подсетей могут успешно пользоваться услугами, являющимися менее важными, тем самым создавая очередь в узлах сети.

Для того чтобы наиболее важные сетевые приложения и службы могли функционировать во время перегрузок, участок сети, по которому передаются пакеты с информацией, должен иметь возможность дифференцированного обслуживания трафика (QoS) [1, 2]. QoS представляет собой набор нормирующих параметров, которым должны пользоваться операторы для поддержки ресурсов своей сети при транспортировке потока данных. Такой подход позволяет за счет качественной диспетчеризации трафика менее важных сетевых приложений правильно функционировать службам, критичным ко времени доставки.

Целью настоящей методики является получение возможности оценки сети оператора и определение численных значений параметров QoS конкретной сети с присвоением ей определенного класса качества согласно рекомендациям

Y.1540¹ и Y.1541². Для её реализации поставлена задача создания программного обеспечения для проведения тестирования операторских сетей на соответствие требованиям классов качества по рекомендациям Y.1540 и Y.1541.

Требование рекомендаций ITU-T Y.1540 и ITU-T Y.1541

Используя требования рекомендации ITU-T Y.1540, рассмотрим сетевые характеристики, наиболее влияющие на сквозное качество обслуживания абонентов операторских сетей:

- производительность сети;
- надежность сети (сетевых элементов);
- задержка;
- джиттер (вариация задержки);
- потери пакетов.

Подразумевая под производительностью сети эффективную скорость передачи за единицу времени, рекомендация Y.1540 не указывает значений параметра скорости передачи данных, но упоминает, что они могут быть получены методом, описанным в рекомендации Y.1221.

Операторы связи рассматривают надежность сети как свойство сохранять свою функциональность в заданных условиях эксплуатации, которая зависит от коэффициента готовности и определяется отношением времени простоя и суммарного времени наблюдения за объектом. Из этого следует вывод, что в идеале при нулевом времени между отказами данный коэффициент будет равен единице, но на практике, разумеется, такого встретить не удается. По этому параметру каждый оператор связи использует свою собственную систему показателей, используя собственный опыт эксплуатации построенной сети.

Анализируя параметры доставки IP-пакетов с учетом этапов сеанса связи, таких как установление соединения, передача данных и закрытие соединения, следует отметить, что рекомендация подразумевает только передачу данных без установления соединения, следовательно, первый и последний этап не рассматривается. Однако и с учетом этого есть все предпосылки оценивать параметры качества обслуживания, используя нормирующие значения.

Параметры качества обслуживания для оценки операторских сетей

Задержкой доставки IP-пакета $IPTD$ является время, проходящее от момента отправки пакета до достижения его места назначения для всех пакетов – с ошибками и без, рассчитываемое по формуле (1)³.

$$IPTD = t_2 - t_1, \quad (1)$$

где t_1 – время отправки IP-пакета из точки А, t_2 – время прибытия пакета в точку В.

¹ ITU-T Y.1540 IP Packet Transfer and Availability Performance Parameters: ITU-T SG12. Series Y: Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities. ITU, 2016. URL: <https://www.itu.int/rec/T-REC-Y.1540-201607-I>

² ITU-T Y.1541 Network performance objectives for IP-based services: ITU-T SG12. Series Y: Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities. ITU, 2011. URL: <https://www.itu.int/rec/T-REC-Y.1541-201112-I/en>

³ ITU-T Y.1540 IP Packet Transfer and Availability Performance Parameters: ITU-T SG12.

Средняя задержка доставки пакета находится как среднее арифметическое значение некоторого количества успешно переданных пакетов. При повышенной нагрузке в сети в узлах могут создаваться очереди, которые прямо пропорционально влияют на задержку доставки пакетов.

В случае, когда величина задержки превысит определенную норму, пакеты с информацией будут автоматически отброшены. Такое явление негативно скаживается при передаче трафика реального времени.

Параметр вариации задержки IP-пакета $IPDV$ находится по формуле (2) как разность абсолютной величины задержки конкретного пакета и эталонной величиной задержки, которая находится как абсолютное значение задержки доставки первого IP-пакета между данными сетевыми точками⁴.

$$IPDV_i = IPTD_i - d_{1,2}, \quad (2)$$

где $d_{1,2}$ – эталонная задержка доставки пакета, которое определяется как абсолютное значение задержки первого пакета для тех же самых точек сети.

Из-за нерегулярного поступления пакетов получателю, например, в IP-телефонии, могут и будут возникать искажения звука, и речь станет неразборчивой.

Коэффициент $IPLR$ вычисляется по формуле (3) как отношение количества потерянных пакетов к сумме потерянных и успешно принятых пакетов из определенного набора отправленных⁵.

$$IPLR = L / (A + L), \quad (3)$$

где L – суммарное количество потерянных пакетов из определенного набора отправленных, A – количество принятых пакетов из того же самого набора отправленных.

Если пакет был потерян, то получатель может отправить отправителю запрос с повторной отправкой. Например, в системах VoIP [3] потерянные пакеты будут вести к значительному искажению речи.

Коэффициент ошибок IP-пакетов $IPER$ находится по формуле (4) как отношение суммарного числа принятых пакетов с ошибками E , к сумме успешно принятых пакетов M пакетов, принятых с ошибками⁶.

$$IPER = E / (M + E), \quad (4)$$

В рекомендации ITU-T Y.1541 представлены численные значения параметров, обозначенных рекомендацией ITU-T Y.1540, в зависимости от класса качества обслуживания (табл. 1). Каждый класс является гарантией качества, который предоставляется оператором каждому пользователю. Пользователь же имеет возможность выбора, подходящего для него тарифа. Значения параметров сформированы на основании реальных измерений сетевых характеристик – длины и числа пакетов, временного интервала наблюдения и т. д.

⁴ ITU-T Y.1540 IP Packet Transfer and Availability Performance Parameters: ITU-T SG12.

⁵ Там же.

⁶ Там же.

Таблица 1.

Нормы для характеристик IP-сетей
с распределением по классам качества обслуживания⁷

Сетевые характеристики	Классы QoS					
	0	1	2	3	4	5
Задержка доставки пакета IP, <i>IPTD</i>	100 мс	400 мс	100 мс	400 мс	1 с	Н
Вариация задержки пакета IP, <i>IPDV</i>	50 мс	50 мс	Н	Н	Н	Н
Коэффициент потери пакетов IP, <i>IPLR</i>	$1 \cdot 10^{-3}$	Н				
Коэффициент ошибок пакетов IP, <i>IPER</i>	$1 \cdot 10^{-4}$	Н				

Некоторые классы QoS имеют ненормированные параметры, обозначенные как «Н». Такой случай предполагает, что значения этих параметров никак не контролируются классами данной рекомендации и могут быть проигнорированы⁸. Однако операторы связи вправе сами устанавливать значения неопределенных параметров.

Особенности разработки программного обеспечения

Учитывая назначение и особенности взаимодействия протоколов транспортного уровня стека TCP/IP, разработчиками принято решение при проведении оценок операторских сетей раздельно проводить такие оценки для протоколов TCP [5] и UDP [6]. Для этого разработаны две базовые программы с интерфейсом командной строки NETSegmentTCP и NETSegmentUDP, учитывающие специфики протоколов, которые будут на выходе представлять данные, пригодные для проведения оценок сквозного контроля качества сети передачи данных в соответствии с рекомендациями ITU-T Y.1540 и Y.1541. Обе программы написаны на языке C++ с применением стандарта C++11, и могут быть скомпилированы как под ОС Windows, так и под Unix-системы.

NETSegmentTCP – программа для тестирования и оценки параметров качества обслуживания на участках сети с использованием TCP-соединений.

NETSegmentUDP – программа для тестирования и оценки параметров качества обслуживания на участках сети с использованием UDP-протокола.

При запуске любой из программ, пользователю необходимо выбрать, в каком режиме для этого хоста она будет работать – в качестве сервера или клиента.

При запуске программы в качестве клиента, пользователь указывает IP-адрес или доменное имя сервера, после чего инициализируется сокет и на сервер отправляется соответствующий протоколу пакет, информирующий его об установлении соединения. Если после первой попытки присоединиться не удаётся, то клиент будет повторять свои попытки до тех пор, пока подключение не произойдет, или пользователь не закроет программу. Если соединение успешно установлено, то в окне терминала клиента отображается оперативная информация о соединении с хостом-сервером: IP адрес сервера, его порт, время пинга.

⁷ ITU-T Y.1541 Network performance objectives for IP-based services: ITU-T SG12.

⁸ Там же.

При запуске программы в качестве сервера, создается и инициализируется «слушающий» сокет [7], который ожидает получение пакетов от других программ-партнеров, запущенных на других хостах сети в качестве клиента. При получении пакета от клиента, информирующего о попытке установить соединение, сервер записывает адрес этого хоста в массив соединений для дальнейшего использования, и отправляет ответный пакет, информирующий клиента о том, что соединение успешно установлено, сервер готов, и работа по тестированию будет продолжена. В окне терминала сервера выводится оперативная информация обо всех хостах, записанных в массиве соединений сервера: IP-адрес, порт и время пинга.

Оценка качества обслуживания сети инициируется на сервере. Пользователь со стороны сервера выбирает, с каким клиентским хостом необходимо выполнить передачу пакетов для оценки качества соединения. Сервер и клиент обмениваются пакетами разного размера для оценки времени задержки передачи пакета, вариации задержки, а также потерь пакетов.

Для реализации задачи при написании кода было решено использовать объектно-ориентированный подход [8]. Подготовлены прототипы, а затем и выполнена реализация следующих классов: `Socket`, `Server`, `Client`, `Connection`, `Packet`, `Measures`. Так как клиентская и серверная часть очень похожи во многих аспектах, их рабочий код собран в одну программу, что позволяет в дальнейшем использовать её в качестве сервера или клиента в зависимости от проводимой оценки сети и расположения хоста. Для обеспечения кроссплатформенности, используются директивы условной компиляции (`ifdef`, `ifndef`), а также подготовлены и отложены вспомогательные макросы и функции для реализации одного функционала с учётом различных аппаратных платформ [7, 8].

Во избежание лишних задержек в связи с использованием ввода-вывода консоли принято решение перенести обработку ввода-вывода в отдельный ИО-поток (от англ. *Input-Output*), в котором для организации и обеспечения безопасного обмена информацией между потоками используются возможности стандартной библиотеки `std` стандарта C++11 – `std::thread` и `std::mutex`.

Дадим краткое описание классов, используемых в программе:

- Класс `Socket` используется в качестве родительского класса для `Client` и `Server` и реализует некоторые общие функции этих классов, такие как: инициализация сокета, хранение данных о потоке ввода-вывода, обновление данных вывода.

- Класс `Server` используется при запуске программы в качестве сервера. Он реализует обработку подключений клиентов, поддержание соединения с ними, хранение данных о подключенных клиентах. Также он отвечает за создание и контроль отдельного потока для ввода-вывода данных, за выполнение оценки качества соединения с хостом-клиентом. Важно, чтобы на хосте, где запущена серверная часть программы, был прощен через NAT соответствующий порт, как минимум до сети, в которой находится клиентская часть. В идеальном случае, IP-адрес сервера должен быть «белым».

- Класс `Client` используется только при запуске программы в качестве клиента. Выполняет почти те же функции, что и `Server`, но только для одного соединения – с хостом-сервером. У класса `Client` нет массива объектов `Connection`, так как соединение у клиента может быть только одно – с сервером. Для запуска

программы в режиме клиента не требуется иметь «белый» IP-адрес или проброшенные через NAT порты.

- Класс Connection хранит информацию об отдельном соединении: дескриптор сокета, адрес хоста, состояние соединения, время последнего обмена данными. Данный класс следит за временем последнего обмена и отправляет пакет для тестирования соединения. Также он отвечает за обработку входящих и исходящих данных, преобразуя данные из «сырого» вида в экземпляр класса Packet и наоборот соответственно. Экземпляры класса создаются методами классов Server и Client при установлении соединения и удаляются при разрыве соединения.

- Класс Packet служит для хранения данных о конкретном пакете данных, переданных от одного хоста другому. От этого класса наследуются несколько других классов, необходимых для работы программы (*PacketPing*, *PacketSys*, *PacketData*).

- После окончания оценки участка сети сервером отправляется специальный пакет об окончании данного процесса на другую сторону соединения. Все подсчитанные результаты записываются в созданный объект класса Measures.

Рассмотрим некоторые особенности работы программы для TCP-протокола. По своей структуре протокол TCP [5] является потоковым протоколом, ориентированным на соединение. Разрабатываемая авторами программа подразумевает отправку данных фиксированных размеров для возможности снятия статистики и проведения оценок. Исходя из этого, данные необходимо было «пакетировать». Для этого применено разделение пакетов – каждый пакет был отделен от «соседних» 2-мя байтами, как показано на рис. 1.

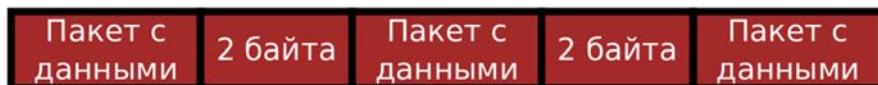


Рис. 1. Пакетирование данных для TCP-протокола

Вторая особенность: данные доставляются получателю в виде потока байтов, в котором нет понятий «сообщения» или «границы сообщения» [5]. В этом отношении чтение данных по протоколу TCP похоже на чтение из последовательного порта – заранее неизвестно, сколько байтов будет возвращено после обращения к функции чтения. Объединение в один пакет нескольких мелких сообщений реализуется протоколом и называется алгоритмом Нейгла [9]. Программа предусматривает отправку данных конкретных размеров и без буферизации, поэтому следует отключить данный алгоритм, что и производится в ходе работы программы. Каждый пакет данных начинается с 1 байта, содержащего тип пакета. Это позволяет принимающему хосту создавать экземпляр соответствующего класса пакета и заполнять его данными соответственно.

Протокол TCP является протоколом с гарантированной доставкой и встроенным механизмом дублирования пакетов, что не позволяет отследить потери пакетов [5]. Коэффициент ошибок IP-пакетов так же не отслеживается, так как программа не предусматривает работу с содержимым пакетов. Таким образом,

по рекомендациям Y.1541 оцениваются только те параметры, которые имеют смысл для TCP-соединений.

Алгоритм работы программ приведен на рис. 2.

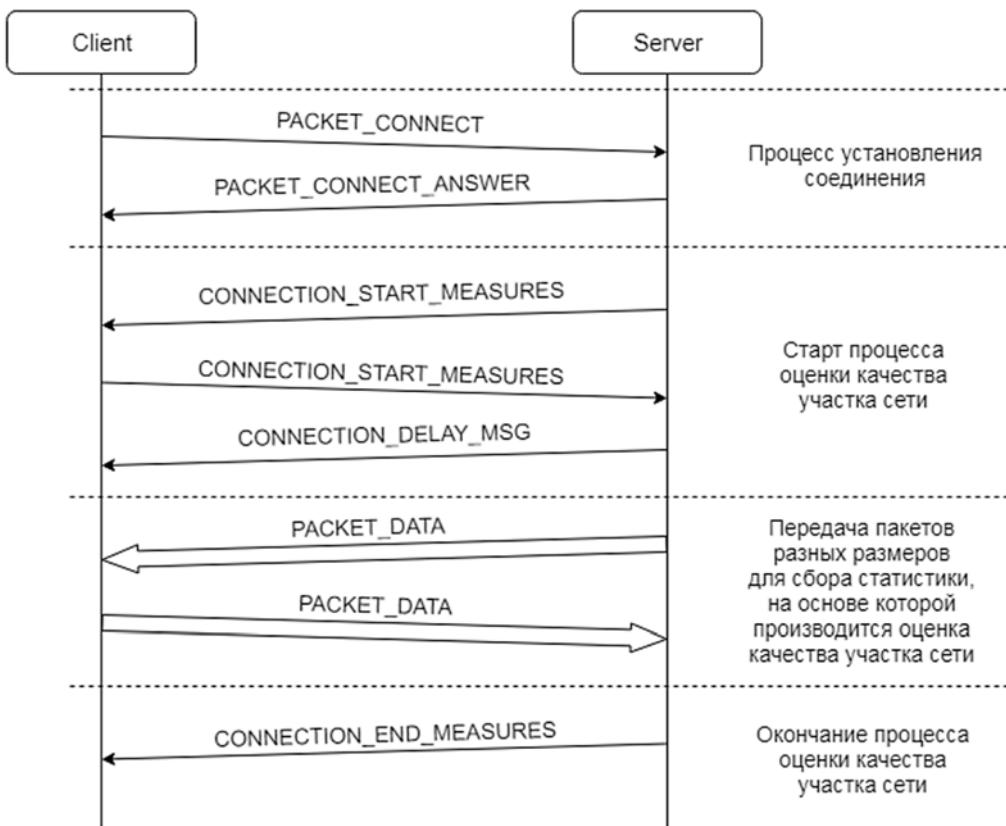


Рис. 2. Алгоритм работы программ NETSegmentTCP и NETSegmentUDP

Оценка качества соединения производится следующим образом: после установления соединения с клиентами пользователь на серверной стороне выбирает то соединение, оценку которого необходимо произвести, путём нажатия соответствующей клавиши. Класс Server обрабатывает это нажатие и посыпает специально сформированный пакет выбранному хосту, сигнализирующий о начале процесса оценки соединения. Клиент, приняв этот пакет, отправляет ответный пакет о том, что он готов произвести оценку качества. Затем сервер начинает отправку пакетов разных размеров, контролируя время отправки. Клиент, получив пакет, отправляет его обратно. Приняв ответный пакет, сервер записывает время приёма. Таким образом, сервер получает время прохождения пакета к хосту и обратно. Исходя из того, что выполнить синхронизацию таймеров хостов достаточно затруднительно, произвести измерение задержки передачи пакета в одном направлении только программными средствами не представляется возможным. Поэтому для проведения оценки параметра программа принимает половину времени, затраченного на передачу пакета в двух направлениях, за время задержки передачи пакета в одном направлении. Вариация задержки оценивается по разнице между задержкой передачи первого пакета и задержкой передачи N -го пакета.

Оценка производится по среднему значению параметров для каждого размера пакета в отдельности. По окончании оценки, результаты записываются в CSV-файл. Присвоение класса качества обслуживания осуществляется относительно максимальных значений времени, вариации задержки передачи и потерь пакетов в оцениваемой сети передачи данных.

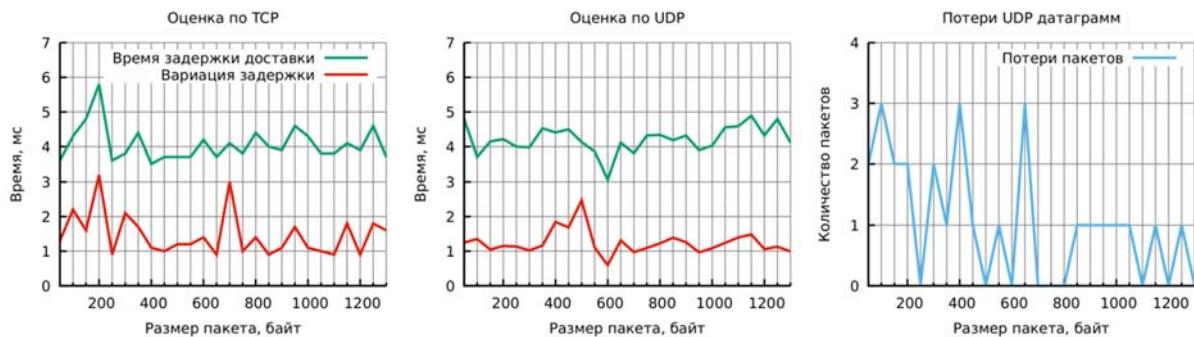
Проведение пробных оценок сетей передачи данных

Для тестирования разработанного программного обеспечения были использованы хосты, подключенные к сетям разных провайдеров и находящиеся в разных городах и областях.

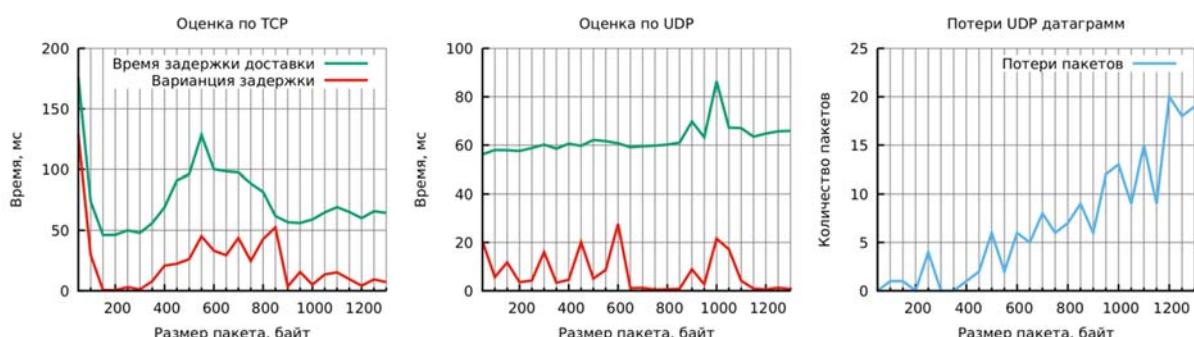
В качестве сервера был использован хост, находящийся в д. Новое Девяткино, Ленинградская область. Хост подключен к сети оператора SkyNet по технологии FTTB с использованием выделенного IP-адреса. На пограничном маршрутизаторе сети выполнен проброс портов с использованием NAT, что позволило программам-клиентам подключаться к серверу по известному IP-адресу или доменному имени хоста.

Для проверки разработанных программ в качестве оцениваемых сетей были взяты три сети с различной технологией абонентского доступа:

1. FTTB. Хост клиента размещен в С.-Петербурге на пр. Большевиков. Результаты тестирования приведены на рис. 3.



1. ADSL. Хост клиента размещен в п. Поля Новгородской области. Результаты тестирования приведены на рис. 4.



2. PON. Хост клиента размещен в г. Великие Луки Псковской области. Результаты тестирования приведены на рис. 5.

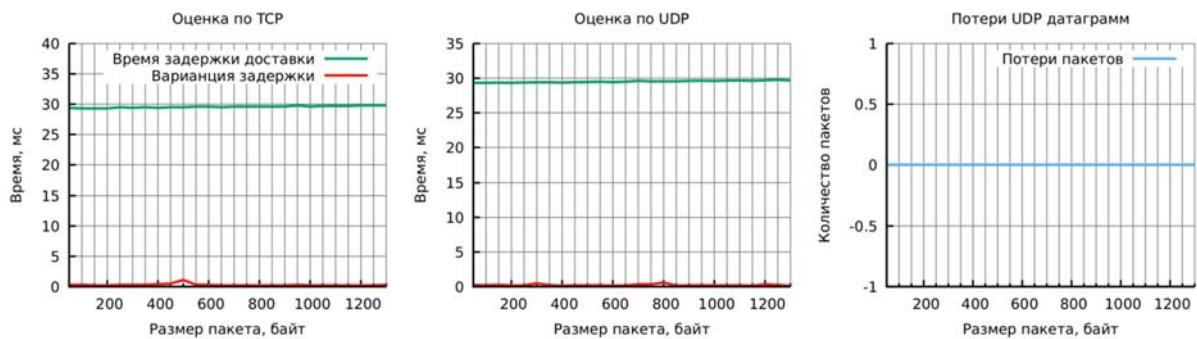


Рис. 5. Результаты тестирования для хоста клиента, подключенного по технологии PON (г. Великие Луки Псковской обл.)

Для этих сетей были получены различные результаты, что подтверждает существенную зависимость параметров качества обслуживания сети от применяемой технологии построения сети. Для каждой выбранной сети была проведена оценка QoS, которая представлена в табл. 2.

Таблица 2.

Оценка QoS для выбранных при тестировании сетей

Адрес	Технология доступа	Класс QoS по TCP	Класс QoS по UDP
Санкт-Петербург, пр. Большевиков	FTTB	0	5
п. Пола Новгородской обл.	ADSL	3	5
г. Великие Луки Псковской обл.	PON	0	0

Заключение

В ходе проведенной работы авторами был разработан и протестирован программный продукт для оценки качества сетей передачи данных и использования его в качестве основы создания программно-аппаратных комплексов для следующих задач:

1. Техническая поддержка доступа на сети оператора связи – предоставление гарантий качества обслуживания абонентов.
2. Тестирование абонентских услуг на качество и выявление узких мест сети оператора.
3. Приемно-сдаточные испытания сетей
4. Предпроектные исследования существующих сетей оператора с точки зрения развития и модернизации.
5. Непрерывная система мониторинга операторских сетей на соответствие заявленным классам QoS.

Литература

1. Яновский Г. Г. Качество обслуживания в IP-сетях // Вестник связи. 2008. № 1. С. 65–74 .
2. Кучеряный Е. А. Управление трафиком и качество обслуживания в сети Интернет. СПб.: Наука и Техника, 2004. 336 с.
3. Гольдштейн Б. С., Пинчук А. В., Суховицкий А. Л. IP-телефония. М.: Радио и связь, 2001. 336 с.
4. Пинчук А. В., Соколов Н. А. Triple-Play Services: аспекты реализации // Вестник связи. 2005. № 6.
5. Postel, J. RFC 793 Transmission Control Protocol. Darpa Internet Program. Protocol Specification. USC/Information Sciences Institute, 1981. URL: <https://tools.ietf.org/html/rfc793>.
6. Postel, J. RFC 768 User Datagram Protocol. USC/Information Sciences Institute, 1981. URL: <https://tools.ietf.org/html/rfc768>
7. Савич У. Программирование на C++. СПб.: Питер, 2004. 781 с.
8. Stroustrup, B. The C++ Programming Language. Addison-Wesley, 2013. 1346 p.
9. Nagle, J. RFC 896 Congestion Control in IP/TCP Internetworks. Ford Aerospace and Communications Corporation, 1984. URL: <https://tools.ietf.org/html/rfc896>

References

1. Yanovsky, G. Quality of Service in IP Networks // Vestnik svyazi. 2008. No. 1. pp. 65–74.
2. Koucheryavy, Y. Traffic Management and Quality of Service on the Internet. SPb: Nauka i Tekhnika, 2004. 336 p.
3. Goldstein, B., Pinchuk, A., Sukhovitskiy, A. Voice over IP. M.: Radio i svyaz', 2001. 336 p.
4. Pinchuk, A., Sokolov, N. Triple-Play Services: Aspects of Implementation // Vestnik svyazi. 2005. No. 6.
5. Postel, J. RFC 793 Transmission Control Protocol. Darpa Internet Program. Protocol Specification. USC/Information Sciences Institute, 1981. URL: <https://tools.ietf.org/html/rfc793>
6. Postel, J. RFC 768 User Datagram Protocol. USC/Information Sciences Institute, 1981. URL: <https://tools.ietf.org/html/rfc768>
7. Savitch, W. Problem Solving with C++. Addison Wesley, 2002.
8. Stroustrup, B. The C++ Programming Language. Addison-Wesley, 2013. 1346 p.
9. Nagle, J. RFC 896 Congestion Control in IP/TCP Internetworks. Ford Aerospace and Communications Corporation, 1984. URL: <https://tools.ietf.org/html/rfc896>

Владимиров Сергей Александрович – старший преподаватель, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, vladimirov.opds@gmail.com

Алексеев Иван Сергеевич – студент, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, 17aliva17@mail.ru

Воронов Алексей С. – студент, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, voron295@gmail.com

Vladimirov Sergey – Senior Lecturer, SPbSUT, St. Petersburg, 193232, Russian Federation, vladimirov.opds@gmail.com

Alekseev Ivan – Student, SPbSUT, St. Petersburg, 193232, Russian Federation, aenye.aaa@yandex.ru

Voronov Alexey – Student, SPbSUT, St. Petersburg, 193232, Russian Federation, romankurazhev@gmail.com