

Литература.

1. Гмурман В.Е. Теория вероятностей и математическая статистика. — М.: Высшая школа, 1999. — 479 с.
2. Лапко В.А., Соколов М.И. Непараметрические методы обработки информации: Учеб. Пособие. — Красноярск: КГТУ, 2001. — 116 с.
3. Лапко А.В., Лапко В.А., Соколов М.И., Ченцов С.В. Непараметрические системы классификации. — Новосибирск: Наука, 2000. — 240 с.
4. Лапко А.В., Ченцов С.В. Непараметрические системы обработки информации. — М.: Наука, 2000 — 350 с.
5. Рубан А.И. Методы анализа данных: Учебное пособие, 2-е изд. — Красноярск: ИПЦ КГТУ, 2004 — 319 с.

Методика измерения пропускной способности в сетях TCP/IP

Султанов Т.Г., аспирант

Самарский Государственный Аэрокосмический Университет

В работе предлагается методика оценки пропускной способности канала. На сегодняшний день существуют системы, позволяющие измерять данную метрику, но они имеют ряд недостатков. Методика основана на тестировании сети пакетами разных размеров.

Ключевые слова: метрика, модель измерения, пакет, ширина канала, полоса пропускания, пропускная способность, avband, pipe test box.

Введение

В работе предлагается метод оценки доступной ширины полосы пропускания канала. Метод основан на измерении задержки в сети D_i для пакетов различного размера W_i .

Измерение таких метрик как доступная ширина полосы пропускания и загруженность канала позволяет предсказывать результирующую производительность сетевого приложения, производить динамический выбор маршрута, а также находить применение во многих других областях сетевых технологий. Метрика пропускной способности — одна из важнейших метрик для таких приложений как grid-вычисления, видео- и аудиопотоки, маршрутизация, пиринговые сети и многих других.

Различные интернет приложения реального времени, в первую очередь, передача голосовой и видео информации становятся все более и более популярными, однако для их качественной передачи требуются высокоскоростные сети. Основными факторами, определяющими качество изображения, являются: качество оборудования (кодека и видеосервера) и доступная полоса канала. Операторы

и их заказчики должны обеспечивать требуемую полосу пропускания для приложений голоса и видео, чтобы гарантировать присутствие требуемых ресурсов на сети.

Метрики, в наибольшей степени характеризующие канал это ширина полосы пропускания канала (C , *saracity*) и доступная ширина полосы пропускания B_{av} . Ширина полосы пропускания канала, C — максимальная пропускная способность IP-уровня, которую может обеспечить маршрут потоку, при условии не загруженности маршрута другими потоками. Доступная ширина полосы пропускания — это пропускная способность IP-уровня, которую может обеспечить маршрут потоку в загруженном другими потоками состоянии.

Хоть на сегодняшний день и существуют различные системы, позволяющие определять доступную полосу пропускания, но они имеют свои недостатки, поэтому поиск новых решений является востребованным. Среди них, такие как *iperf*, *netperf*, *pathrate* и *pathload*, *abget*, а также ряд малоизвестных программ *ncs*, *netest*, *pipechar* [1]. Покажем на рисунке ниже общий принцип работы с программными продуктами *iperf*, *netperf* и *pathrate*.



Рисунок 1. Принцип работы с системами измерений

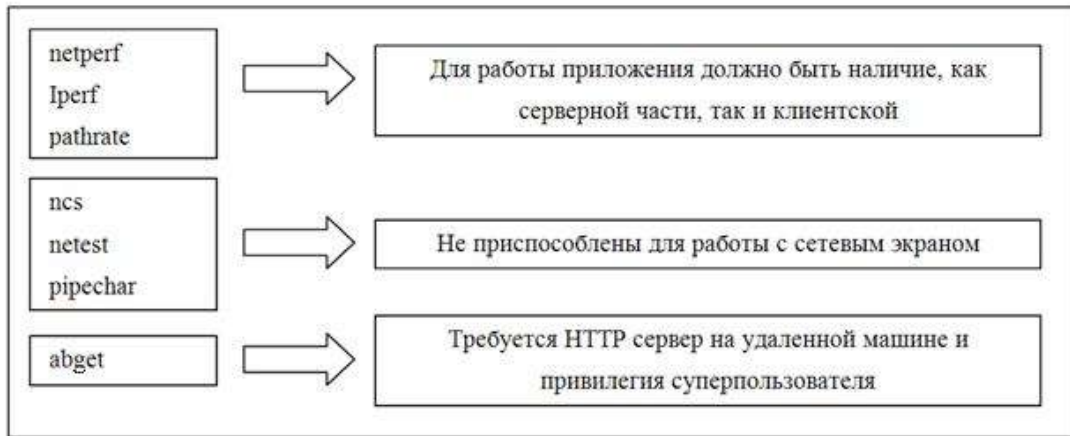


Рисунок 2. Недостатки существующих систем

Рассмотрим следующий программный продукт — утилиту *abget*. Утилита *abget* позволяет оценить доступную (свободную) пропускную способность сети в обоих направлениях, не имея доступа к удаленному серверу. Требуется HTTP сервер на удаленном сервере и привилегии суперпользователя. Для определения доступной пропускной способности сети от сервера к нам требуется указать URL достаточно большого файла на сервере. *abget* эмулирует ТСР протокол, чтобы спровоцировать сервер посылать данные в желаемом темпе.

У каждого из вышеперечисленных продуктов есть свои недостатки. Утилиты *iperf*, *netperf*, *pathrate* имеют одну особенность, которая является их существенным недостатком. Для того чтобы оценить пропускную способность сети требуется установить клиентскую и серверную части программы. Утилита *abget* требует HTTP сервер на удаленном сервере и привилегии суперпользователя, а что касается программ *ncs*, *netest*, *pipechar*, так как они не приспособлены для работы с сетевыми экранами, что в современных условиях делает их малоиспользуемыми. Проиллюстрируем эти положения на рисунке ниже.

В то же время вышеперечисленные программы используют алгоритмы оценки пропускной способности канала, основывающиеся на прогонке довольно большого количества пакетов по каналу связи, что снижает пропускную способность сети и требует значительного времени. Поэтому поиск нового метода, позволяющего оценить пропускную способность, и создание на базе него утилиты является актуальным.

Новая модель измерения пропускной способности

В работе [2] было показано, что преобразование Литтла с оценкой доступной полосы пропускания может быть записано как:

$$B_{av} = \frac{W}{D - a} \quad (1)$$

Уравнение (1) подсказывает альтернативный путь для нахождения доступной ширины B_{av} . Новый метод предполагает изменение размера тестового пакета для измерения доступной полосы пропускания. Если провести тестирование сети пакетами разных размеров W_1 и W_2 , тогда мы получим 2 разных значения времени задержки D_i .

Таким образом, система из двух полученных уравнений легко разрешима относительно доступной ширины B_{av} , а также величины a , имеющей смысл минимально возможной задержки:

$$B_{av} = \frac{W_1 - W_2}{D_1 - D_2} \quad (2)$$

$$a = \frac{W_1 D_2 - W_2 D_1}{W_1 - W_2} \quad (3)$$

Стоит отметить, что похожий результат был получен в ставшей классической работе Якобсона [3] «Предотвращение перегрузки и специализированный контроль».

Таким образом, проводя тестирование сети утилитой *ping* с помощью пакетов различного размера (например, 64 bytes и 1064 bytes) можно оценить доступную ширину исходящего канала между двумя точками (а точнее IP адресами) в глобальной сети. Этот метод прост и дает достаточно точные значения для исследуемых величин, а главное он не требует установки дополнительного, достаточно сложного программного обеспечения.

Была разработана утилита AvBand (Available Bandwidth), реализующая вышеописанный метод, используя в своей основе алгоритм *ping*. Этот алгоритм был разработан Майком Муусом в 1983 в США для операционной системы BSD [4]. Его преимущество в том, что можно работать с любым маршрутизатором или устройством, которое откликается на пакеты запросов ICMP Echo.

Данная утилита определяет пропускную способность исходящего канала связи между машиной, с которой проводится измерение и интересующим нас удаленным сервером. Для этого программа измеряет двустороннюю величину за-

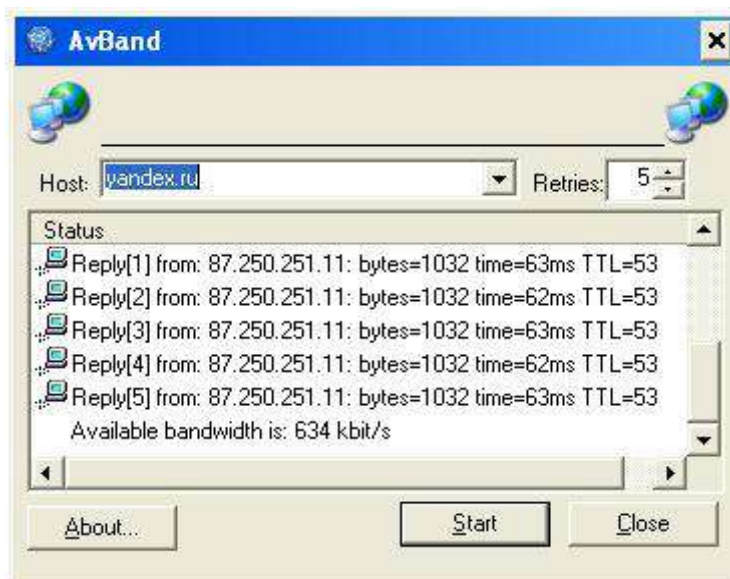


Рисунок 3. Пример работы утилиты AvBand

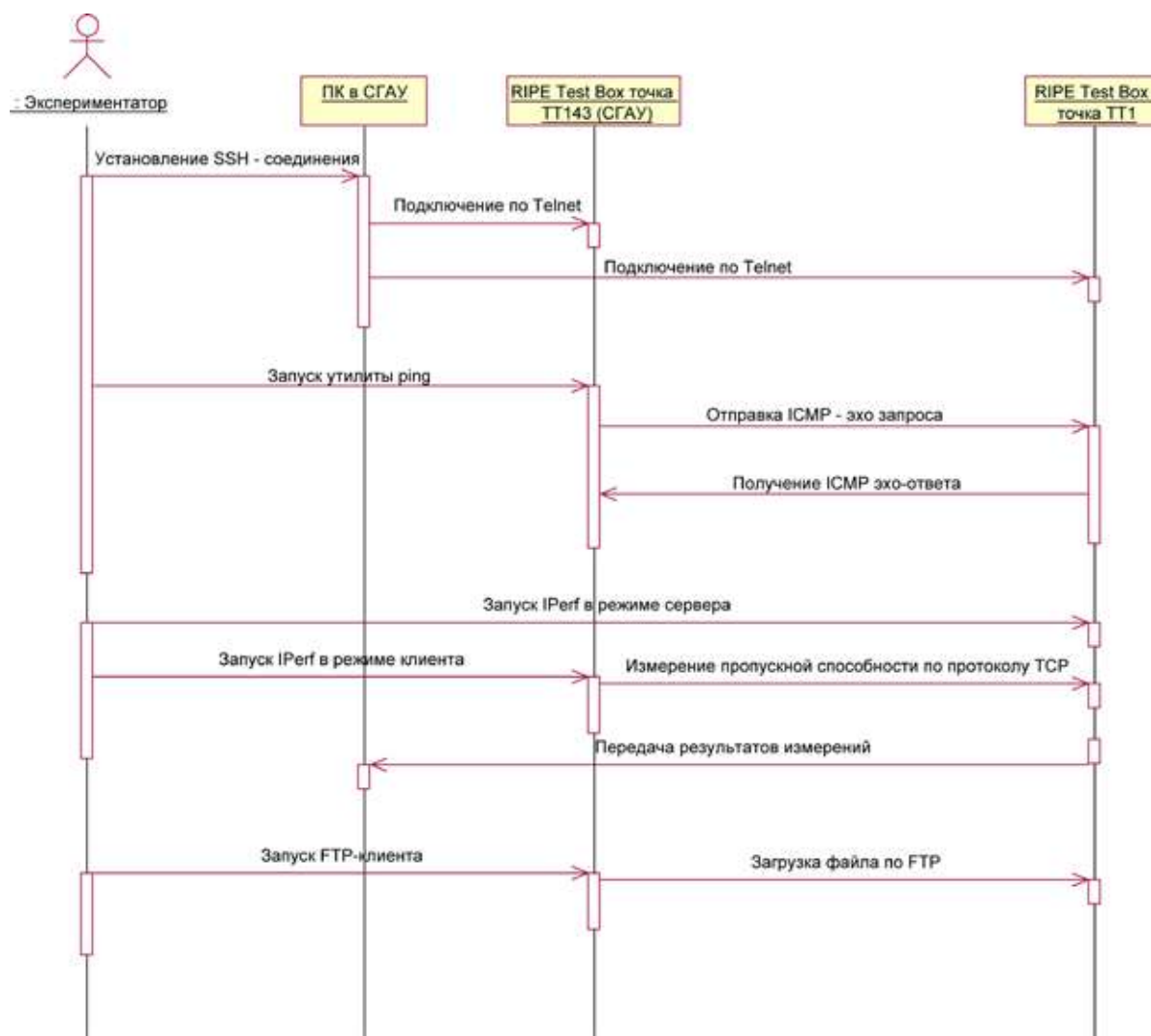


Рисунок 4. Схема проведения эксперимента

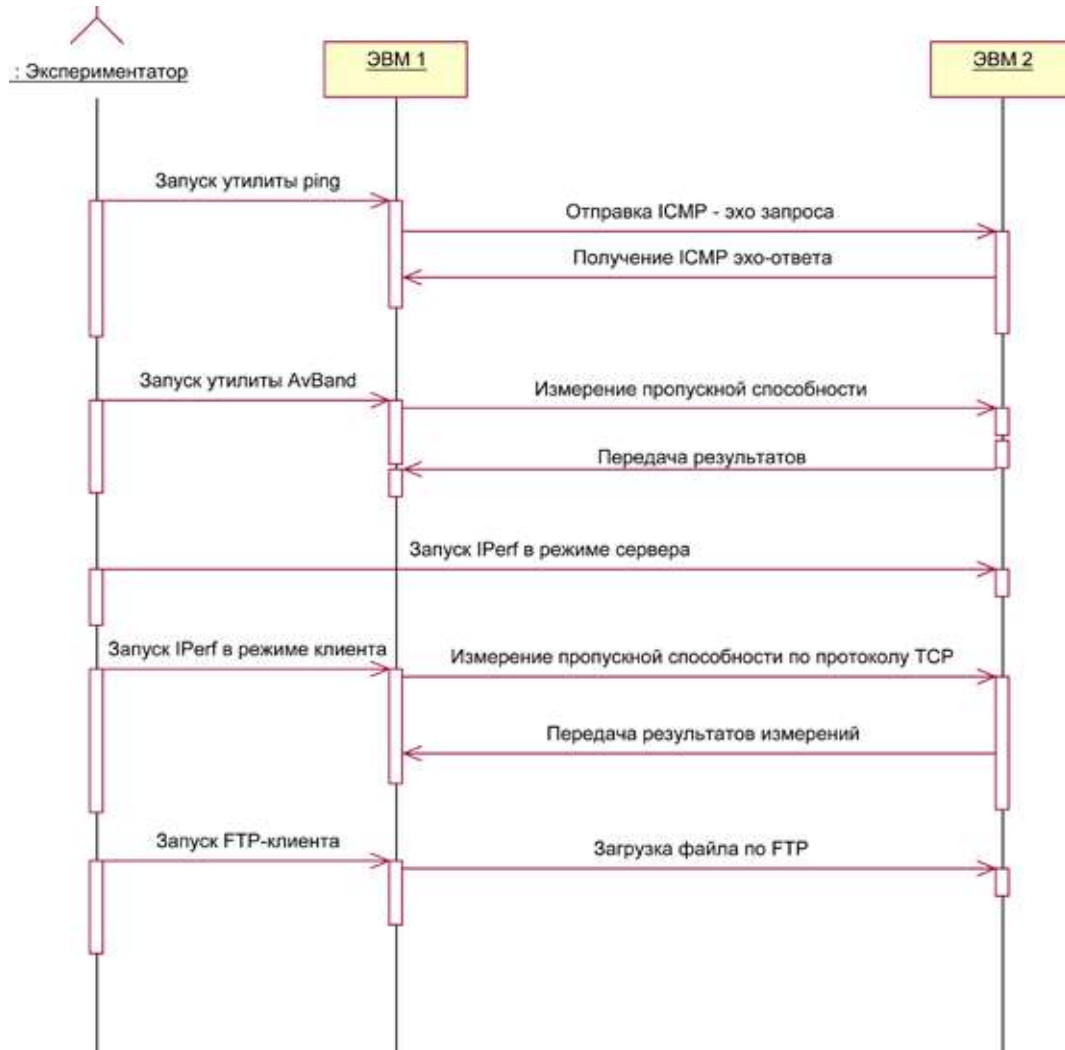


Рисунок 5. Схема проведения эксперимента

держки пакетов при прохождении по сети, то есть значение RTT (Round Trip Time) — время между отправкой запроса и получением ответа. При этом сначала генерируются пакеты стандартного Windows размера в 32 байта и определяется их задержка, а следующим шагом формируются пакеты большего размера в 1032 байта и замеряется их RTT. На рисунке 3 представлен скриншот работы программы.

В поле ввода “Host” задается имя сервера, ширину канала до которого мы собираемся измерять. В поле “Retries” указывается количество пингов, то есть эхо-запросов, которые будут отправлены на удаленный сервер. После этого достаточно нажать на кнопку “Start” и утилита отправит заданное количество пакетов размера 32 байта, далее такое же количество пакетов размером 1032 байта. Собранные значения полученных задержек по каждой из групп пакетов усредняются, а затем с помощью новой модели рассчитывается доступная ширина канала и выводится на экран (рисунок 2). Стоит отметить, что под шириной канала подразумевается пропускная способность от источника до сервера, с которым проводится изме-

рение. Другими словами это скорость передачи данных от источника к серверу.

Данная версия утилиты разработана для платформы Windows и использует библиотеку ICMP Windows API. В ближайшее время планируется разработка утилиты для Unix систем, прежде всего для семейства Linux.

Экспериментальная часть

В настоящее время проводится проверка модели несколькими способами, и первый из них — это прецизионное тестирование с помощью измерительной системы RIPE Test Box [5].

RIPE Test Box представляет собой персональный компьютер под управлением ОС Linux. В его состав входит также GPS приемник Acutime 2000 [http://www.ripe.net/test-traffic/] в качестве источника сигналов точного времени. Этот блок даст возможность измерять качество соединений с множеством других точек, так как он автоматически включается в глобальную измерительную

инфраструктуру. Измерительная точка RIPE Test Box развернута в СГАУ и используется в качестве NTP сервера уровня stratum 1, в работе [2] дано развернутое описание.

Характерные времена исследуемых процессов (задержки пакетов, джиттер) имеют порядок от 10 мкс до 1 сек, поэтому для их надежного измерения вполне достаточно достигнутой точности системных часов.

Прецизионный эксперимент проводится с точками RIPE (на первом этапе между ТТ143 и ТТ1). Он содержит 3 этапа:

- Прецизионное измерение задержки пакетов 64 и 1064 bytes с точностью 2 мкс;
- Измерение доступной полосы с помощью IPerf;
- Измерение полосы методом скачивания файла по FTP с точки ТТ143 на ТТ1.

На рисунке 4 приведена схема проведения эксперимента.

Таким образом, у нас сформируется три альтернативно измеренных величины полосы пропускания для последующего сравнительного анализа.

Следующей фазой эксперимента является проверка утилиты AvBand. Он будет проводиться с использованием следующих измерительных механизмов:

- Непосредственно сама утилита AvBand;
- Стандартный ping;
- Использование Iperf;
- FTP тестирование.

На рисунке 5 показана схема проведения второго эксперимента.

На настоящий момент были проведены измерения с университетом Огайо, а также ряд локальных экспериментов с использованием сетей различных операторов Самарской области. Это позволит собрать данные для последующего статистического анализа. Все данные по экспериментам приведены в таблице 1 ниже.

В настоящее время измерения еще не завершены, планируется набрать данные с не менее 20-ти точек, разбросанных по территории планеты. Таким образом, подводя итог проделанной работе, можно сделать основной вывод — теоретическая модель вычисления доступной полосы подтверждается.

Таблица 1. Результаты эксперимента

Место проведения эксперимента		Ping (64 b & 1064 b)	FTP (upload)	AvBand (>15 retries)	IPerf (TCP)
Сервер измерений	Удаленная точка				
СГАУ 89.186.245.217	Бременская школа университета прикладных наук 194.94.26.62	5517 Кбит/с	1840 Кбит/с	_____	_____
Бременская школа университета прикладных наук 194.94.26.62	СГАУ 89.186.245.217	3273 Кбит/с	935,11 Кбит/с	_____	_____
Университет Огайо 199.18.182.69	СГАУ 89.186.245.217	5744 Кбит/с	4044 Кбит/с	5744 Кбит/с	2453,44 Кбит/с
СГАУ 89.186.245.217	Университет Огайо 199.18.182.69	6359 Кбит/с	1253 Кбит/с	6359 Кбит/с	1203 Кбит/с
СГАУ 89.186.245.217	server2.hosting.reg.ru (Москва)	1150 Кбит/с	1040 Кбит/с	_____	_____
СГАУ 89.186.245.217	IOС RAS	20 Мбит/с	22,5 Мбит/с	20,6 Мбит/с	_____
Домашний компьютер (Оператор "Аист")	СГАУ 89.186.245.217	536 Кбит/с	600 Кбит/с	536 Кбит/с	659 Кбит/с
Домашний компьютер (Оператор "Волга-Телеком")	СГАУ 89.186.245.201	274 Кбит/с	_____	274 Кбит/с	283 Кбит/с
Домашний компьютер (Оператор "Инфолада", Жигулевск)	СГАУ stream.ip4tv.ru	346 Кбит/с	374 Кбит/с	_____	_____

Литература:

1. A Tigumala, J Ferguson, J Dugan, F Qin, K Gibbs, Available from: <http://dast.nlanr.net/Projects/Iperf>
2. А.П. Платонов, Д.И. Сидельников, М.В. Стрижов, А.М. Сухов Измерительная инфраструктура для изучения качества соединений в российском сегменте Интернет, Телекоммуникации, №1, 2009, с.11-16
3. Jacobson V. Congestion avoidance and control. In Proceedings of SIGCOMM 88 (Stanford, CA, Aug. 1988), ACM
4. Mike Muus, Ping documentation, <http://ftp.arl.mil/~mike/ping.html>
5. RIPE Test Box, <http://ripe.net/projects/ttm/>
6. Choi, B.-Y., Moon, S., Zhang, Z.-L., Papagiannaki, K. and Diot, C.: Analysis of Point-To-Point Packet Delay In an Operational Network. In: Infocom 2004, Hong Kong, pp. 1797-1807 (2004)
7. Padhye, J., Firoiu, V., Towsley, D., Kurose, J.: Modeling TCP Throughput: A Simple Model and its Empirical Validation. In: Proc. SIGCOMM Symp. Communications Architectures and Protocols, pp. 304-314 (1998).