

ВЫБОР И УСОВЕРШЕНСТВОВАНИЕ ГЕНЕРАТОРА WEB ТРАФИКА ДЛЯ ТЕСТИРОВАНИЯ ПОВЕДЕНИЯ МАРШРУТИЗАТОРА В СОСТОЯНИИ ПЕРЕГРУЗКИ

Ю. К. Гликман

Санкт-Петербургский институт информатики и автоматизации РАН
199178, Санкт-Петербург, 14-я линия В.О., д.39
glickman@fokus.gmd.de

УДК 681.326(075)

Ю. К. Гликман. **Выбор и усовершенствование генератора WEB трафика для тестирования поведения маршрутизатора в состоянии перегрузки** // Труды СПИИРАН. Вып. 1, т. 3. — СПб: СПИИРАН, 2003.

Аннотация. Рассмотрены наиболее популярные генераторы трафика. Произведён выбор генератора трафика для проведения эксперимента с маршрутизатором. Показаны пути усовершенствования программы. — Библ. 12 назв.

UDC 681.326(075)

Y. K. Glickman. **The selection and improvement of WEB traffic generator for testing of routers at overload condition** // SPIIRAS Proceedings. Issue 1, v.3. — SPb: SPIIRAS, 2003.

Abstract. A briefly review of the most popular traffic generators. The traffic generator was chosen for the test of router. The improvement ways were showed. — Bibl. 12 items.

При разработке методов борьбы с атаками на WEB узлы «отказ в обслуживании» (distributed denial of service) для проверки гипотез требуется проведение экспериментов с большим числом задействованных пользователей. Для реалистичности эксперимента число этих пользователей должно исчисляться десятками тысяч человек и более. Конечно же, обеспечить подобное условие для проведения эксперимента практически невозможно. Остаётся единственный выход: симулировать работу участка Internet.

Проблема генерации трафика является одним из ключевых аспектов в симуляции работы сети Internet. Для проведения небольших экспериментов очень часто достаточно использовать один или несколько источников трафика. Иногда для проведения эксперимента бывает достаточно использование только программы ping. Но очень часто исследователи нуждаются в более реалистичной смеси трафика. Таким образом, основной проблемой является вопрос: как представить различные источники трафика в эксперименте.

Основной задачей эксперимента является исследование поведения маршрутизатора в состоянии перегрузки. Экспериментальная установка, как показано на рис. 1, будет состоять из трёх основных частей:

- 1) серверная часть;
- 2) маршрутизатор;
- 3) клиентская часть.

Клиентская часть посылает запрос серверу. Сервер, обработав запрос, посылает ответ. Поскольку мы собираемся исследовать поведение маршрутизатора в условиях, как можно более приближенных к реальным, то необходимо, по меньшей мере, чтобы клиентов было много, то есть, чтобы запросы шли с разных IP адресов. При этом характеристики генерируемого трафика должны быть как можно сильнее приближены к реальным. Кроме того, трафик должен быть весьма высокой интенсивности, так как мы хотим исследовать поведение маршрутизатора в состоянии перегрузки. Следовательно, для проведения подобно-

го эксперимента очень важно правильно подобрать генератор трафика. Рассмотрим же наиболее известные генераторы трафика.

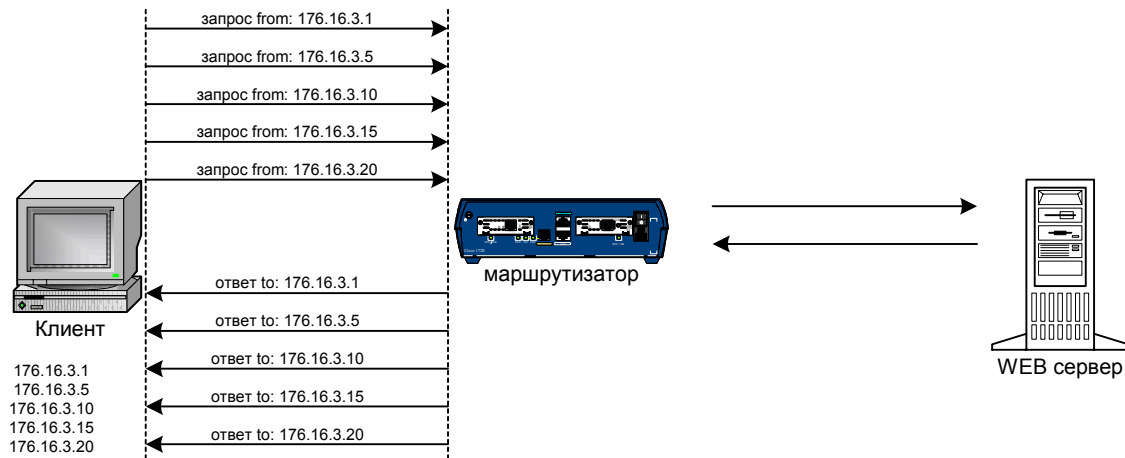


Рис. 1. Схема проведения эксперимента

Пользователи операционной системы Unix часто применяют систему для тестирования пропускной способности сети ТТСП. Выбирая источник и место назначения, можно анализировать различные участки сети, в частности, данная программа позволяет пользователю менять параметры передачи, а именно: число посылаемых пакетов, размер пакетов, номер порта, с которого посылаются пакеты, номер порта, на котором принимаются пакеты и ряд других параметров. Изменяя эти параметры, вы можете тестировать различные механизмы буферизации в сетевом оборудовании между источником и получателем пакетов [9]. Данный пакет не предназначен для имитации реального WEB трафика. Он предназначен для тестирования пропускной способности сети. Как генератор трафика всё, что он умеет, это посылать TCP пакеты различного размера с разными временными интервалами. Что делает его непригодным для использования в нашем эксперименте.

Пакет Netperf был разработан компанией Hewlett-Packard ещё в 1993 году, и с тех пор претерпел ряд изменений. Он представляет из себя набор программ для измерения различных аспектов сетевой производительности. В основе данного пакета лежит измерение производительности при посылке массива данных, запросе данных, получении ответа с использованием TCP или UDP и интерфейса сокетов Berkeley.

Netperf разработан в соответствии с моделью клиент-сервер. В состав пакета входят две исполняемых программы: netperf и netserver. При запуске программы устанавливается служебное соединение с удалённой системой. Это соединение будет использоваться для обмена тестовой информацией и результатами с удалённой системой. После того как служебное соединение было создано, будет открыто специальное отдельное соединение для выполнения теста как такового с использованием API и протокола, предварительно выбранного для теста. Затем будет выполнен тест и показаны результаты [8]. Этому пакету присущи те же недостатки, что и предыдущему. Отличительной его особенностью является возможность посылки не только TCP пакетов, но и UDP, что расширяет возможности его использования.

NetPIPE — независимое от протокола средство для измерения производительности сети. Данная программа содержит в себе всё лучшее, что было в программах ТТСП и Netperf. Кроме этого, в программе присутствует возмож-

ность получения данных, которые могут быть легко использованы для графического отображения производительности.

Рассматривая сеть с точки зрения end-to-end, NetPIPE ясно показывает состояние сети, ассоциированное с различными протокольными уровнями. NetPIPE отвечает на такие вопросы как: насколько быстро будет получен пакет данных размера k в месте назначения? Какая сеть и протокол передадут пакет данных размера k быстрее? Что обеспечивает максимум пропускной способности и насыщения? Имеется ли блок данных размера k , для которого пропускная способность максимальна? Как быстро небольшое по размеру управляющее сообщение ($< 1\text{kbyte}$) будет доставлено, и какая сеть и протокол лучше всего подходят для этой цели?

NetPIPE состоит из двух частей: независимый от протокола драйвер и специальная коммуникационная составляющая. Коммуникационная составляющая содержит необходимые функции для установления соединения, отправки и получения данных и закрытия соединения. Эта часть имеет различную реализацию для каждого протокола.

Реализация драйвера базируется на принципах, изложенных в HINT computer performance metric. Безусловно, производительность сети не может быть точно описана с использованием только одного размера передаваемого блока данных. NetPIPE увеличивает размер передаваемого пакета данных, начиная с одного байта и заканчивая размером, когда время передачи превысит одну секунду. Следовательно, NetPIPE — это программа с изменяемой продолжительностью измерения производительности, и она может масштабироваться на все скорости сети. В отличие от программ измерения производительности с фиксированным размером пакета, NetPIPE не устареет и не станет менее точной по мере технологического развития. Для увеличения универсальности NetPIPE информация в ней измеряется в битах, а не в байтах [10].

NetPIPE использует пинг-понг передачу. Это заставляет сеть передавать только блок данных без потока других блоков данных, которые находятся в том же сообщении. Результатом является время передачи одиночного блока, зная которое, можно сказать, какой размер блока наилучший или какая пропускная способность обеспечивается блоком размера k .

NetPIPE создаёт файл, содержащий время передачи, пропускную способность, размер блока и колебание времени передачи для каждого варианта размера блока данных. Используя эти данные, легко построить набор графиков, воспользовавшись любым графическим пакетом.

Данный пакет, безусловно, является более удобным, чем те, что были описаны выше. Наболее значимым его преимуществом является полная независимость от протокола. Это прекрасный инструмент, но он предназначен несколько для других целей, чем те, которые мы преследуем в нашем эксперименте. К сожалению, эта программа не способна имитировать реальный WEB трафик.

NetSpec — это программа, предназначенная для определения производительности сети. Она была разработана в университете штата Канзас, США. Программа обеспечивает тестирование сетей с большими потоками данных и позволяет менять типы источника трафика и его режимы. Для задания параметров эксперимента и поддержки управления экспериментами по измерению производительности с произвольным числом соединений посредством LAN или WAN программа NetSpec позволяет использовать простой блочноструктурированный язык программирования [5].

NetSpec представляет много возможностей, которые не поддерживаются аналогичными средствами, описанными выше. Среди этих возможностей можно особо отметить:

- наличие параллельных и последовательных множественных соединений;
- выбор типа трафика (FTP, HTTP, MPEG и т.д.);
- наличие наиболее широко используемых транспортных протоколов (TCP, UDP);
- три различных режима трафика;
- масштабируемость;
- возможность собирать информацию системного уровня от коммуникационных систем так же хорошо, как от промежуточных сетевых узлов.

На рис. 2 показана основная архитектура NetSpec 3.0 [6]. Контроллер (controller) — это процесс, поддерживающий пользовательский интерфейс, который в настоящий момент представляет из себя файл с описанием эксперимента на встроенном языке.

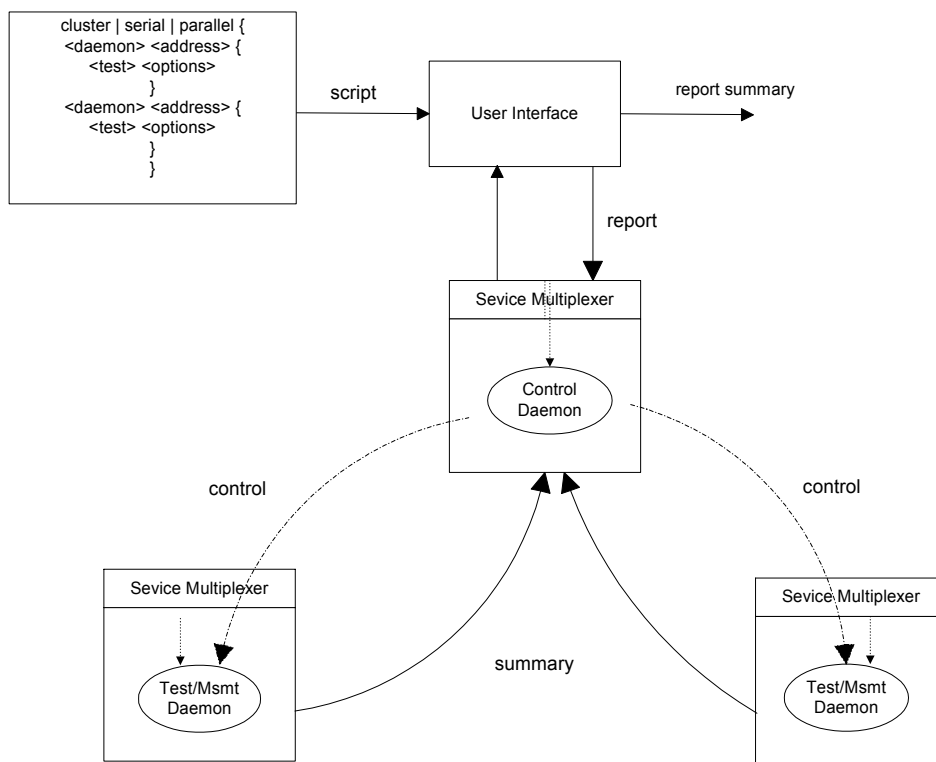


Рис. 2. Архитектура NetSpec 3.0

В данном языке соединение (connection) — это основной элемент эксперимента, который через управляющий демон (control daemon) управляет демонами, реализующими непосредственно тестирование. Для каждого соединения в эксперименте создаются надлежащие тест-демоны (test daemons). Каждый демон ответственен за генерацию собственного отчёта после выполнения эксперимента, и демоны измерения (measurement daemons) концентрируют своё внимание на как можно более тщательном собирании данных, не беспокоясь о выполнении транспортных функций. Исходящий отчёт доставляется контроллеру через управляющий демон для просмотра пользователем. Связь между контроллером и демонами реализуется с помощью использования базирующегося на ASCII языка, который увеличивает переносимость и расширяемость.

NetSpec поддерживает три основных режима трафика:

- Полный потоковый режим (Full Stream Mode)
При этом режиме данные передаются так быстро, как только возможно.
- Пакетный режим (Burst Mode)
При этом режиме данные передаются пакетами по прерыванию таймера. Пользователь может задавать временной интервал и размер пакета. Недостаток данного режима состоит в том, что если процедура записи пакета не успеет завершиться прежде, чем произойдет следующее прерывание, то возникнет ошибка записи для следующего пакета.
- Пакетный режим с очередью (Queued Burst Mode)
Является разновидностью основного пакетного режима. Для этого режима характерны те же параметры, что и для предыдущего. Преимуществом этого режима является то, что изменение частоты прерываний не вызовет ошибку записи пакета, поскольку последовательность посылки пакетов регламентируется специальной очередью. Недостатком этого алгоритма является зависимость характеристик трафика от времени задержки в очереди.

NetSpec может потенциально имитировать трафик FTP, TELNET, VBR Video Traffic (MPEG, Video-Teleconferencing), CBR Voice Traffic и HTTP (World Wide Web traffic) на уровне приложения. Эти возможности делают NetSpec прекрасным инструментом для сетевого тестирования. Однако, следует отметить, что при разработке этой программы не учитывались принципы поведения реального пользователя. Поэтому WWW трафик, создаваемый этой программой не отличается высокой реалистичностью.

Программа TG была разработана совместно фирмами SRI International и 3Com Corporation в 1998 году. Программа предназначена для оценивания производительности сети, работающей с протоколами, основанными на пакетной передаче данных. Она посылает и принимает однонаправленные потоки пакетных данных, передаваемые от клиентского процесса операционной системы Unix между узлом источника трафика и узлом стока в сети. С помощью этой программы можно тестировать для сравнения производительности различные протоколы и их версии [7].

Программа TG управляется простым, но гибким конфигурационным языком, который позволяет оперировать с различными рабочими режимами, протоколами, адресными функциями и экспериментировать с параметрами трафика. Использование конфигурационного языка позволяет создавать трафик, состоящий из пакетов нескольких различных размеров, и задавать временное распределение. В текущей реализации поддерживаются транспортные протоколы TCP и UDP, а также поддерживается IP Multicasting. К сожалению, в текущей реализации TG не способна поддерживать более одного активного соединения одновременно, что является существенным недостатком с точки зрения нашего эксперимента. Кроме того, в ней отсутствует возможность имитации реального трафика.

Эта программа была разработана в Бостонском университете в 1997 году. Ее появлению предшествовала большая теоретическая работа, проделанная разными учёными, по определению аналитических моделей транзакций WWW. Данная программа состоит из комбинации этих моделей. Целью SURGE явля-

ется обеспечение масштабируемой основы, которая с точки зрения сервера создаёт запросы, имитирующие поведение реальных пользователей.

Если смотреть сверху вниз, то SURGE состоит из набора клиентов, обращающихся к серверу, как показано на рис. 3 [1]. Каждый из клиентов выполняет набор нитей, каждая из которых может находиться либо в активном, либо в пассивном состоянии. Каждая нить запрашивает набор документов, которые затем передаются сервером. Этот временной интервал называется активным временем (ON time). После получения набора документов нить засыпает на некоторое время. Это время называется пассивным временем (OFF time). Наличие разделения времени на активное и пассивное является очень важным преимуществом пакета SURGE по сравнению с другими аналогичными программами.

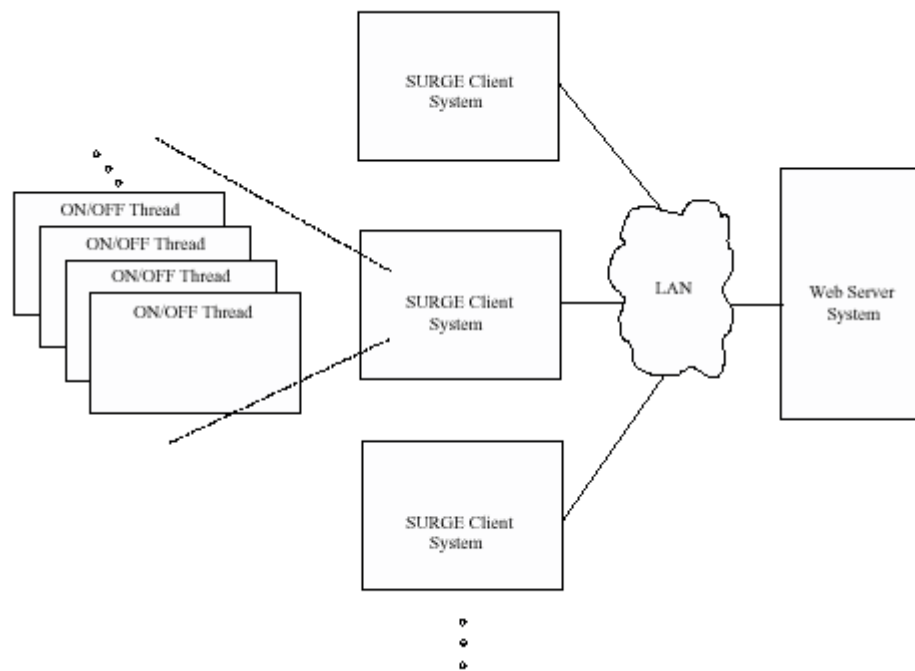


Рис. 3. Архитектура SURGE

Когда SURGE запускается на клиентской системе, он начинает создавать массивы данных в соответствии с аналитическими моделями поведения WWW клиента. SURGE использует следующие модели [2]:

- модель распределения уникальных размеров файлов;
- модель распределения размеров всех файлов, передаваемых с сервера;
- модель популярности всех запрашиваемых с сервера файлов;
- модель последовательности имён запрашиваемых с сервера файлов;
- модель пассивного времени (OFF time);
- модель активного времени (ON time);
- модель числа документов, передаваемых в течение активного времени.

После создания необходимых массивов данных запрашиваемые документы копируются на сервер. Затем происходит непосредственный запуск клиентской программы, которая начинает обращаться к серверу с запросами по описанной модели схеме. Особенностью этой программы является то, что в отличие от

всех, описанных выше, она основана на целом ряде полноценных статистических исследований. Практическое использование которой показало, что трафик, создаваемый ею, практически невозможно отличить от настоящего. В качестве же недостатка, можно отметить отсутствие возможности имитации трафика FTP, TELNET, VBR Video Traffic (MPEG, Video-Teleconferencing), CBR Voice Traffic. Данная программа, к сожалению, умеет имитировать только HTTP. Но так как имеются исходные коды программы, то она может быть модернизирована.

Проведённый обзор генераторов трафика показал, что наиболее сильно поставленным требованиям отвечают программы NetSpec и SURGE. Однако обе эти программы нуждаются в модернизации, чтобы соответствовать требованиям эксперимента. В них отсутствует возможность генерации трафика от имени различных IP адресов. Предпочтение было отдано программе SURGE, так как она имеет более простую и расширяемую организацию. Кроме этого, в отличие от NetSpec, в ней намного лучше проработаны модели поведения клиента WWW [3,4].

Проведение эксперимента предполагает разработку системы измерения и анализа результатов, которая должна включать: параметры системного уровня (кумулятивный показатель использования процессора на каждого пользователя, использование реальной и виртуальной памяти, объем доступной и загруженной памяти, объем свободной памяти, использование дискового пространства и каналов ввода/вывода); параметры уровня мониторинга процессов (количество процессов и их типы, взаимосвязь между процессами, время жизни процесса, текущее состояние, соотношение времен различных процессов); параметры сетевой информации (количество подключений и их статусы, усредненное количество посланных и полученных пакетов, продолжительность соединения, тип соединения, используемый порт и протокол). Данные измерений и их анализ позволят выявить причины и слабые места тестируемой системы и прогнозировать поведение за пределами тестируемых параметров.

Таким образом, за основу генератора трафика для проведения экспериментов с маршрутизатором будет взята программа SURGE, куда будет добавлена возможность посылать запросы от имени разных IP адресов с одной клиентской машины. Одним из путей реализации этого нововведения может послужить использование виртуальных IP адресов [11, 12]. Дополненная этой полезной возможностью программа SURGE станет особенно удобным инструментом в руках исследователя сетевых технологий.

Литература

- [1] *Barford P., Crovella M.E.* An Architecture for a WWW Workload Generator — Wide Web Consortium Workshop on Workload Characterization, 1997.
- [2] *Barford P., Crovella M.E.* Generating representative web workloads for network and server performance evaluation — ACM Sigmetrics '98, 1998.
- [3] *Feldman A., Gilbert A., Huang P., Willinger W.* Dynamics of IP traffic: A study of the role of variability and the impact of control — SIGCOMM '99, 1999.
- [4] *Floyd S., Paxson V.* Difficulties in Simulating the Internet — IEEE/ACM Transactions on Networking, 2001.
- [5] *Jonkman R.* NetSpec: Philosophy, Design and Implementation. Master's Thesis — University of Kansas, Informations and Telecommunications Sciences Laboratory, 1996.
- [6] *Lee B. O., Frost V. S., Jonkman R.* NetSpec 3.0 Source Models for telnet, ftp, voice, video and WWW traffic — ITTC-TR, 1997

- [7] *McKenney P. E., Lee D. Y., Denny B.A.* Traffic Generator Software (Release Notes) — SRI International and 3Com Corporation, 1998.
- [8] Netperf: A Network Performance Benchmark — Hewlett-Packard Company, 1995.
- [9] Network performance testing with TTCP — http://www.ccci.com/product/network_mon/tnm31/ttcp.htm
- [10] *Snell Q. O., Mikler A. R., Gustafson J. L.* NetPIPE: A Network Protocol Independent Performance Evaluator — <ftp://ftp.scl.ameslab.gov/pub/netpipe>.
- [11] *Stevens W. R.* TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols — Addison-Wesley, 1996.
- [12] *Stevens W. R.* UNIX network programming. Networking APIs: Sockets and XTI —Prentice Hall PTR, 1998.