

УДК 621.372.061.3.001.63:681.3

Использование симулятора ns-3 для моделирования поведения сетевых протоколов

А. В. Борисов, А. В. Карпухин, Л. И. Маркова

Харьковский национальный университет имени В.Н. Каразина, Украина

Харьковский национальный университет радиоэлектроники, Украина

В статье рассмотрены вопросы моделирования перегрузок в компьютерных сетях с помощью сетевого симулятора ns-3. Определены основные параметры, которые определяют характеристики процессов в сетях с протоколом TCP. Предложена методика оценки влияния этих параметров на эффективность работы сети.

Ключевые слова: *сети передачи данных, компьютерный эксперимент, протокол TCP, управление перегрузками.*

У статі розглянуті питання моделювання перевантажень в комп'ютерних мережах за допомогою мережевого симулятора ns-3. Визначені основні параметри, які впливають на характеристики процесів в мережах з протоколом TCP. Запропонована методика оцінки впливу цих параметрів на ефективність роботи мережі.

Ключові слова: *мережі передачі даних, комп'ютерний експеримент, протокол TCP, управління перевантаженнями.*

In the paper simulations of overloads in computer networks by means of a network simulator ns-3 are considered. Principal parameters that characterize processes in the TCP networks were defined. Methodology of network overall performance estimation, which is based on this parameters, was proposed

Key words: *data networks, computer-based experiment, TCP protocol, congestion control.*

1. Суть проблемы и её актуальность

Изучение перегрузок, возникающих в компьютерных сетях, и методов их предотвращения является очень актуальным вопросом на современном этапе развития компьютерных сетей. Появляются новые приложения и технологии, такие как GRID и распределенные вычисления, которые передают достаточно большое количество данных по сети; видеоконференции и программы для передачи голоса предъявляют все более жесткие требования к качеству обслуживания (QoS – Quality of Service). Развивается и сама сеть Internet, растет количество пользователей и, как следствие, растет объем передаваемого трафика, который в конце 2009 года составил около 7500-12000 Пб/месяц. Все эти факторы являются предпосылкой для возникновения перегрузок в компьютерных сетях.

Многочисленные исследования процессов в сети Internet показали, что статистические характеристики трафика обладают свойством временной масштабной инвариантности (самоподобием). Большинство исследователей считают, что основной причиной самоподобия трафика является протокол TCP. [1]. Чтобы понять механизм воздействия протокола TCP на самоподобие сетевого трафика, необходимо описать в общих чертах алгоритм работы TCP.

Протокол TCP является основным протоколом работы в Internet, осуществляет доставку данных в виде байтовых потоков с установлением соединения и применяется в тех случаях, когда требуется гарантированная доставка сообщений. Он включает механизм контроля потока, который гарантирует, что отправитель не переполняет буфер приемника, и механизм контроля заторов, который пробует препятствовать попаданию слишком большого объема данных в сеть (что приводит к потерям пакетов).

Таким образом, хорошо изученные и отработанные методы теории вероятности не применимы для описания и предотвращения сетевых перегрузок. Также, определенные ограничения накладывает сама архитектура протоколов Internet. Так, например, конкурирующие TCP потоки, а также все множество транспортных протоколов, используемых в компьютерных сетях, «ничего не знают» друг о друге, но различные модели управления перегрузками в TCP и других протоколах должны мирно сосуществовать. Если же одна из реализаций TCP окажется агрессивнее других, то она будет использовать большую часть пропускной способности канала, что будет мешать передачам данных в «соседних» соединениях. Одновременно и излишняя консервативность алгоритма будет отрицательно сказываться на общей производительности протокола

2. Контроль перегрузки протоколом TCP

Регулирование трафика в протоколе TCP подразумевает существование двух независимых процессов: контроль доставки, управляемый получателем с помощью параметра rwnd (Receiver Advertised Window), и контроль перегрузки, управляемый отправителем с помощью окна перегрузки cwnd (Congestion Window) и процедуры медленного старта, контролируемого переменной ssthresh (Slow Start Threshold). Процесс доставки отслеживает заполнение входного буфера получателя, второй - регистрирует перегрузку канала, а также связанные с этим потери и понижает уровень трафика. Окно перегрузки cwnd и пороговое значение ssthresh позволяют согласовать полную загрузку виртуального соединения и текущие возможности канала, минимизируя потери пакетов при перегрузке. Меньшее из значений cwnd и rwnd в результате и определяет объем посылаемых данных. Управляя перегрузкой TCP, протокол проходит через четыре этапа: медленный старт, избежание перегрузки, быстрая передача и быстрое восстановление [2]. На каждом из этих этапов величина cwnd меняется согласно различным правилам, но, в общем, смысл алгоритма предотвращения перегрузки заключается в удержании значения cwnd в области максимально возможных значений. По существу эта оптимизация осуществляется с помощью потери пакетов. В различных вариантах протоколов TCP (Tahoe, Reno, Vegas) используются различные модификации алгоритма контроля перегрузок. Все эти тонкости в результате приводят к дополнительным осложнениям поведения протокола TCP в перегруженных сетях [3, 4].

3. Тестовый стенд

В данной работе для изучения поведения TCP потоков использовался дискретно-временной симулятор с открытым исходным кодом ns-3 (Network

Simulator 3). Он предоставляет исследователю набор классов, пользуясь которыми, наследуя и модифицируя их можно смоделировать широкий спектр протоколов и процессов, происходящих в компьютерных сетях. Также симулятор позволяет моделировать процессы в реальном времени и интегрировать его с испытательным стендом (testbed), делать испытательный стенд частью моделируемой сети и т. д. [5]. Симулятор ns-3 содержит множество тестов для всех компонент, что гарантирует достоверность получаемых результатов.

С помощью данного симулятора создавалась модель сети TCP/IP (см. рис. 1), где все хосты связаны с маршрутизатором соединением типа точка-точка. На хостах-отправителях моделировалась работа приложений, посылающих данные с постоянным битрейтом (bitrate) на хост-получатель, где работало приложение, принимающее данные от обоих хостов. Скорость генерирования данных отправителями (C_f), задержку (d_b) и пропускную способность (C_b) каналов в узком месте, а также задержку (d) и пропускную способность (C) каналов у хостов-отправителей можно было варьировать, задавая для каждого нового численного эксперимента свои параметры. Также изменялся еще один параметр – размер очереди типа Drop Tail Queue (Q_s) на сетевом интерфейсе маршрутизатора, соединенном с получателем. Окно принимающего хоста (rwnd) было преднамеренно сделано очень большим, чтобы лимитирующим фактором было только значение окна перегрузки (cwnd).

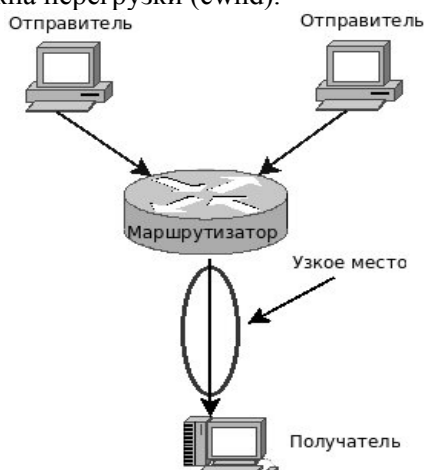


Рис. 1. Топология модельной сети

Очевидно, что состояние перегрузки в такой сети возникнет, когда суммарная скорость, с которой хосты-отправители посылают данные, будет превосходить пропускную способность канала получателя. Причем, ключевыми параметрами, влияющими на возникновение перегрузки, будут (при достаточно большом C и малом d): C_f , d_b , C_b , и Q_s . В дальнейшем, при описании численных экспериментов, будут приводиться значения только этих параметров.

4. Методика изучения поведения TCP протокола

Даже в такой простой системе с двумя TCP-соединениями количество переменных, характеризующих состояние данной системы, очень велико. Но

можно выбрать соответствующее сечение фазового пространства путем надлежащего выбора этих переменных. Было выбрано значение окна перегрузки (cwnd), так как оно непосредственно влияет на скорость пересылки данных.

Во время моделирования процесса работы двух ТСР-соединений отслеживалось изменение значений cwnd каждого ТСР потока от отправителей к получателю. При каждом изменении значения cwnd новое значение и момент времени, в который произошло это изменение, записывались в файл. В результате мы получаем два временных ряда (для двух хостов), которые задают ступенчатую функцию зависимости cwnd от времени (так как значение cwnd все время хранится в памяти машины, и изменяется в фиксированные моменты времени).

Но для того чтобы более подробно изучить данный процесс одной зависимости cwnd от времени мало. Проблема в том, что значения окна перегрузки – это всего лишь проекция полного набора переменных системы. К тому же функция cwnd(t) не является непрерывной. Другая проблема также состоит в том, что, начиная с определенного момента времени, значение окна перегрузки не позволяет делать выводы о состоянии системы во всех деталях.

В работе [6] было предложено использовать усредненные по N значения временного ряда: $[x_t, x_{t-\delta}, x_{t-2\delta}, \dots]$, как легко измеримую характеристику сложных систем и было показано, что можно использовать для восстановления скрытых многомерных траекторий. Данный метод, примененный к значениям окна перегрузки (cwnd), приводит к соотношениям [7]:

$$x[i] = \frac{1}{N} \sum_{j=1}^N cwnd_x[i-j]$$

$$y[i] = \frac{1}{N} \sum_{j=1}^N cwnd_y[i-j]$$

Здесь x и y обозначают два ТСР потока. Величина N отвечает за масштаб усреднения и чем больше N , тем больше скрытых размерностей системы может быть восстановлено.

В нашем случае функции cwnd(t) различны для каждого из хостов и зафиксированы только моменты изменения значения этой функции. Поэтому, для применения вышеописанного метода и построения фазового портрета необходимо взять значения окна перегрузки в одинаковые моменты времени.

5. Результаты

При определенных параметрах тестового стенда такая, казалось бы, довольно простая система проявляет довольно сложное поведение. В частности, если: $C_f=5\text{Mbps}$, $d_b=10\text{ms}$, $C_b=5\text{Mbps}$, $Q_s=20$ пакетов (1 пакет = 536 байт, во всех проведенных численных экспериментах), наблюдается регулярное биение (см. рис. 2); начальный интервал с переходным этапом установления данного режима на рисунке удален. Причем, каждый из ТСР-потоков попеременно получает преимущество друг над другом на определенный промежуток времени.

При проведении численных экспериментов с другими значениями параметров C_b , d_b , C_b , и Q_s система также могла проявлять подобное сложное

периодическое поведение, однако было замечено, что уменьшение значения размера очереди Q_s приводит к очень сильному ужесточению режима передачи данных. Например, можно уменьшить размер очереди до $Q_s=2$ пакета, а остальные параметры оставить неизменными ($C_f=5\text{Mbps}$, $d_b=10\text{ms}$, $C_b=5\text{Mbps}$, $Q_s=20$ пакетов). Результат можно видеть на рис. 3. Причем в области установившегося режима пики окна перегрузки одного хоста чередуются с пиками окна перегрузки другого.

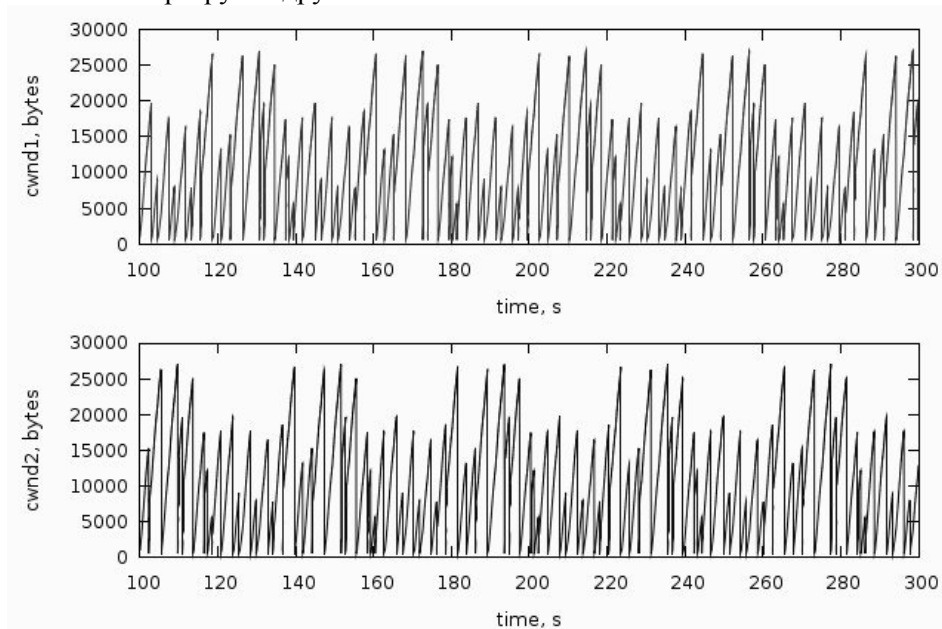


Рис. 2. Зависимость окна перегрузки от времени для двух конкурирующих TCP-потоков

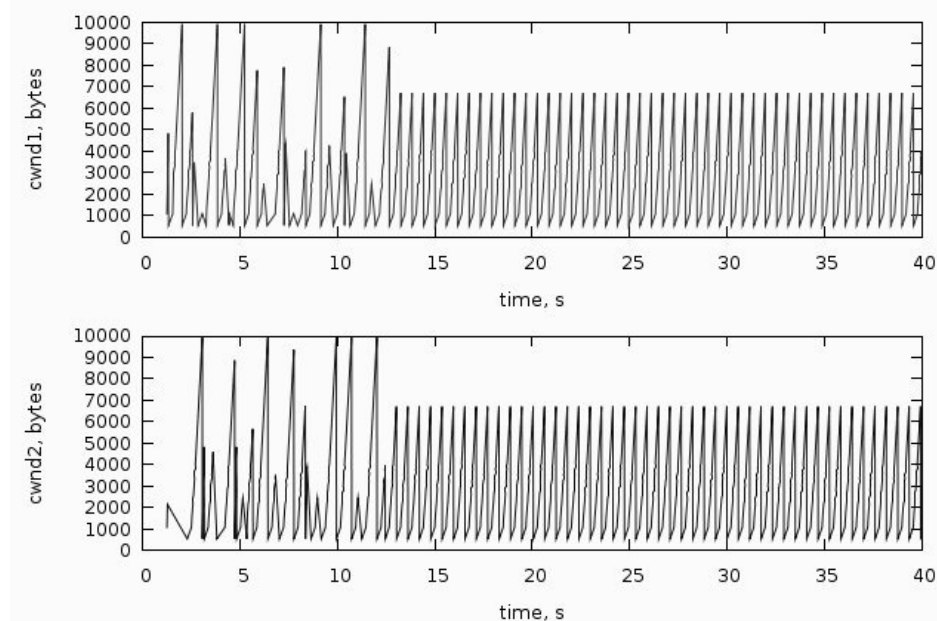


Рис. 3. Зависимость окна перегрузки от времени для двух конкурирующих TCP-потоков – более жесткий режим (переходной процесс показан)

Фазовый портрет, полученный после всех преобразований, описанных в п.4, для модели с $Q_s=2$ пакета, интервалом усреднения $N=2000$ и значениями $cwnd$, взятыми каждые 10ms, можно видеть на рис. 4.

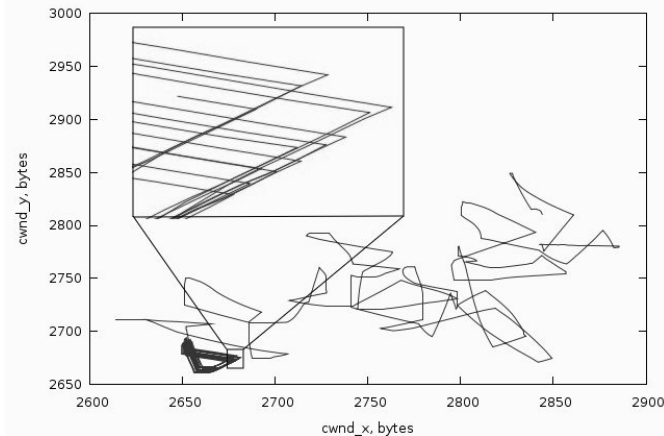


Рис. 4. Фазовый портрет ($Q_s=2$ пакета)

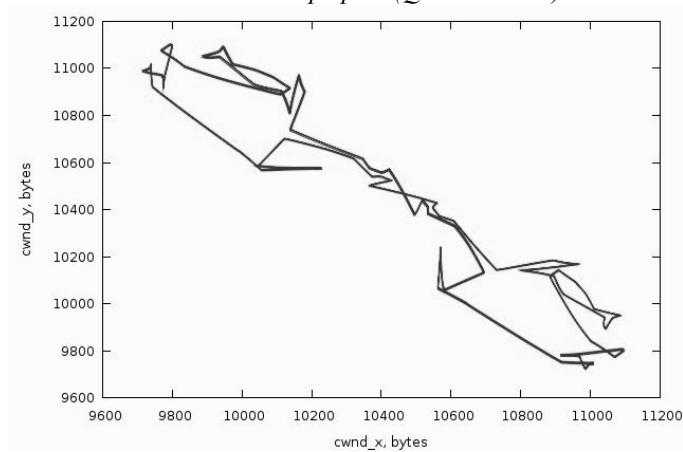


Рис. 5. Фазовый портрет – аттрактор более сложной формы ($Q_s=20$ пакетов)

Как можно видеть, фазовые траектории формируют петлю, которая имеет довольно тонкую структуру. Причем эта траектория довольно устойчива. При изменении времени старта ТСП потоков друг относительно друга (т.е. возмущая систему путем изменения положения точки старта) фазовая точка после небольшого «блуждания» начинает описывать ту же петлю. Такие траектории называются аттракторами. Более формальное определение аттрактора дается в [8].

Фазовый портрет для эксперимента с $Q_s=20$ пакетов можно видеть на рис. 5. В этом случае аттрактор имеет более сложную форму (тонкая структура не видна из-за выбранного масштаба изображения).

6. Выводы и направления дальнейших исследований

Таким образом, мы видим, что даже такая простая система, состоящая из двух конкурирующих ТСП потоков, в определенных условиях демонстрирует

очень сложное поведение. Проведенное исследование позволяет сделать вывод, что размер буфера маршрутизатора является одним из ключевых параметров, влияющих на работу протокола TCP.

Дальнейший путь, на наш взгляд, состоит в том, чтобы описывать сети связи информационных систем, использующие протокол TCP, как нелинейные динамические системы. В этом случае есть возможность применить весь арсенал классических методов анализа поведения таких систем, разработанный Пуанкаре, Ляпуновым, Биркгофом и т.д. Речь идет о качественном анализе поведения динамических систем в фазовом пространстве, что дает возможность определить все возможные режимы движения (работы) системы, а также определить значения параметров, при которых в системе наблюдаются нежелательные хаотические явления (потери пакетов, снижение производительности). Эти вопросы являются актуальными в первую очередь для ISP (провайдеров Internet), которые, к сожалению, до сих пор не обращают на них должного внимания.

В глобальном масштабе всей сети Internet решить проблему заторов и потерь пакетов, очевидно, не представляется возможным в связи с тем, что перестроить всю сеть нельзя в силу технических и экономических причин.

Однако, в ограниченных по размерам сетях (даже довольно больших) возможно дать рекомендации по проектированию (и дальнейшей эксплуатации) таких сетей, которые позволят свести к минимуму отрицательные явления хаотизации.

ЛИТЕРАТУРА

1. Leland W.E., Taqqu M.S., Willinger W., Wilson D.V. On the self-similar Nature of Ethernet Traffic, // IEEE/ACM Transactions of Networking, 1994, vol. 2(1), pp. 1-15, 1994.
2. Allman M, Paxson V. Request for Comments 5681. TCP Congestion Control [Электронный ресурс]. – Режим доступа: <http://www.ietf.org/rfc/rfc5681.txt>, 2009.
3. Карпухин А. В. Математическое моделирование хаотических явлений в высокоскоростных сетевых информационных системах с протоколом TCP. // Системи обробки інформації. – Х.: ХУПС, 2009. – Вип.4(78), – с.64-69.
4. Карпухин А. В. Особенности реализации протокола TCP в современных компьютерных сетях // Системи обробки інформації – Х.: ХУПС, 2009. – Вип.6(80), – с.49-53.
5. Симулятор ns-3 и сопутствующая документация [Электронный ресурс]. – Режим доступа: <http://nsnam.org>
6. Packard N. H., Crutchfield J. P., Farmer J. D. Shaw R. S., “Geometry from a Time Series”, //Phys. Rev. Lett., 45, 1980, pp. 712-716.
7. Veres A., Boda M.. The chaotic nature of TCP congestion control. // Computer and Communication Societies Proceedings. IEEE, 2000, vol3, pp. 1715-1723.
8. Peitgen H. O. Jurgен H. Saupe D., “Chaos and Fractals - New Frontiers of Science”, Springer-Verlag, 1992, NY.