

УДК 004.738.057.4

В.Т. ЕРЕМЕНКО, П.А. СЫСОЕВ, А.С.ЗАСИМОВ

ИССЛЕДОВАНИЕ ПОВЕДЕНИЯ ТРАНСПОРТНЫХ ПРОТОКОЛОВ В КОРПОРАТИВНЫХ СЕТЯХ В УСЛОВИЯХ ИНТЕНСИВНОГО ТРАФИКА

В работе рассматриваются различные версии протокола TCP (Transmission Control Protocol). Основная задача состоит в определении оптимальной версии TCP на основе изучения поведения этих протоколов в условиях интенсивного трафика и измерения их производительности. Для этого предлагается использовать моделирование транспортных сообщений с помощью раскрашенных сетей Петри, а также симулятора сетей (ns).

In work various versions of protocol TCP (Transmission Control Protocol) in the loaded network are considered. The basic purpose consists in definition of optimum version TCP on the basis of studying behaviour of these protocols in conditions of the intensive network traffic and measurement of their productivity. For this purpose it is offered to simulate models of transport reports with the help of painted networks Petri, and also a package network simulator (ns).

На сегодняшний день наблюдается тенденция к всеобщему переходу на стек протоколов TCP/IP, и большинство корпоративных сетей поддерживает именно его. Следует учитывать, что в Интернете, который может использоваться корпоративными сетями в качестве WAN (Wide Area Networks), могут встречаться участки с разными протоколами (Ethernet, ATM, SDH, Frame Relay, PPP и т.д.). Эти технологии имеют разные алгоритмы обработки ситуаций перегрузки (или не имеют их вовсе), а отправитель и получатель, как правило, не имеют данных о том, какие протоколы реализуют виртуальное соединение.

Для достижения высокой производительности конечных систем и сети в целом очень важно правильно выбрать транспортный протокол. В стеке TCP/IP основным транспортным протоколом является TCP (Transmission Control Protocol, RFC 793). Он обеспечивает надежную передачу данных через ненадежную среду передачи. В частности, на протокол TCP возложена задача управления потоками и перегрузками, что подразумевает согласование скорости передачи данных с техническими возможностями узлов сети.

С точки зрения архитектуры TCP/IP все узлы сети можно разделить на два класса. Первый класс составляют конечные системы, которые генерируют информационный трафик либо являются его потребителями. Например, такими системами могут быть web- или ftp-серверы, а также компьютеры пользователей, которые запрашивают с этих серверов информацию. Второй класс составляют системы, которые являются посредниками в передаче информационного трафика. В роли таких систем чаще всего выступает различное сетевое оборудование, например, маршрутизаторы. Транспортные протоколы, такие как TCP, работают только на конечных системах.

При передаче пакеты информации могут быть потеряны. Потеря пакета может быть следствием либо перегрузки сети, либо ненадежности канала связи. Надежность доставки обеспечивается с помощью стратегии подтверждений – получатель должен подтвердить прием данных путем передачи подтверждения отправителю. Если же сегмент был потерян, то он передается заново с помощью механизма повторной передачи. Протокол TCP обеспечивает управление потоком между конечными системами. С момента своей начальной публикации (RFC 793) протокол претерпел ряд изменений, целью большинства из которых являлось улучшение механизма управления потоком. Появилось несколько версий протокола, например, TCP-Reno (RFC 2581), TCP-NewReno (RFC 2582, RFC 3782), TCP-Vegas [1], TCP-Westwood [2].

В настоящее время более популярной является версия TCP-NewReno, (RFC 3782), использующая алгоритм быстрой повторной передачи (Fast Retransmit) и быстрого восстановления (Fast Recovery). Алгоритм, описанный в RFC 3782, соответствует варианту *Careful* алгоритма TCP NewReno из RFC 2582, и исключает проблему множественных повторных пересылок. Этот алгоритм использует переменную *recover*, значение которой соответствует исходному порядковому номеру посланного пакета. После каждого таймаута

повторной передачи наибольший порядковый номер переданного пакета записывается в переменную *recover*.

Существует несколько эвристических подходов, основанных на временных метках или на преимуществах поля кумулятивных (групповых) подтверждений, которые отправитель в некоторых случаях использует, чтобы различить случаи с тремя задублированными АСК, следующие за повторно посланным пакетом, который был потерян, и тремя задублированными подтверждениями, сопряженными с откликами на повторно посланные по ошибке сегменты. Отправитель может использовать эвристику, чтобы решить, следует ли запускать режим быстрой повторной передачи, даже если задублированные АСК не перекрывают диапазон номеров, определенный переменной *recover*.

TCP-Vegas контролирует размер окна путем мониторинга отправителем RTT для пакетов, посланных ранее. Если обнаруживается увеличение RTT, система узнает, что сеть приближается к перегрузке и сокращает ширину окна. Если RTT уменьшается, отправитель определит, что сеть преодолела перегрузку, и увеличит размер окна. Следовательно, размер окна в идеальной ситуации будет стремиться к требуемому значению.

Версия TCP-Westwood позволяет достичь большей эффективности использования канала. В этой модификации протокола используется новый алгоритм управления окном перегрузки, основанный на оценке потока данных (RE – Rate Estimation) и текущего значения полосы пропускания (BE – bandwidth estimation). На основе этих оценок производится вычисление *cwin* и *ssthresh*. Для больших произведений полосы на RTT этот алгоритм может дать лучший результат, чем NewReno. Помимо управления перегрузкой со стороны отправителя в TCP предусмотрен механизм управления со стороны получателя. Получатель в отклике АСК посылает значение параметра *rwnd*, определяющее число сегментов, которое готов принять получатель. Такой механизм управления скользящим окном особенно важен при работе в сетях с большой величиной RTT.

Сравнительно недавно консорциум IETF одобрил версию TCP SACK (Selective Acknowledgement, RFC 2018), которая позволяет подтверждать прием данных не в порядке их поступления, как это было раньше, а выборочно. Это позволяет получить два преимущества. Во-первых, повышается эффективность повторной передачи сегментов TCP благодаря сокращению времени выполнения этой процедуры. Обычно протокол TCP использует алгоритм повторной передачи, опираясь на информацию, которую он получает на основе упорядоченных подтверждений. Такой вариант вполне приемлем, однако в случае его применения для восстановления каждого потерянного сегмента требуется примерно один цикл обращения. SACK же позволяет осуществлять в одном цикле повторную передачу сразу нескольких потерянных сегментов.

Во-вторых, благодаря выборочным подтверждениям, протокол TCP точнее оценивает доступную ширину полосы пропускания в условиях нескольких последовательных потерь сегментов и способен обойтись без алгоритма «Медленный старт». TCP SACK играет важную роль в соединениях, в которых используется окно большого размера. Во время работы данной версии протокола, при потере сегмента или серии сегментов, получатель имеет возможность точно проинформировать отправителя о том, какие данные были приняты успешно, и указать на потерянные данные в потоке сегментов. Отправитель может повторно передавать только потерянные данные, а не весь их блок.

Исследование поведения протокола, оценка его производительности являются актуальной задачей. Так, известно, что стандартный алгоритм управления потоком TCP имеет следующие недостатки:

- Потеря сегмента расценивается протоколом как наступление перегрузки сети, что в общем случае неверно, т.к. пакет с данными может быть потерян вследствие помех в канале. Это снижает эффективность работы протокола, например, на беспроводных каналах связи.
- Для оценки свободных ресурсов сети стандартный алгоритм управления потоком искусственно создает состояние перегрузки сети.

Остается актуальной и задача построения более эффективного алгоритма управления потоком. Для этого оценка свободной пропускной способности сети может быть принята как эвристическое правило. Доказано [3], что такая оценка является довольно точной. Это дает основания для использования такой оценки в будущих версиях транспортных протоколов.

Моделирование сетевого трафика. Ряд работ по исследованию сетевого трафика показывает, что этот трафик более качественно моделируется с помощью статистически самоподобных процессов, чем с помощью традиционной Пуассоновской модели.

В работе [4] приводится доказательство фундаментального свойства самоподобия сетевого трафика. Трафик рассматривается как комбинация источников, которые его генерируют. Каждый источник имеет структуру ON/OFF. Некоторый период времени источники могут генерировать пакеты информации (периоды ON), при этом внутри одного периода пакеты приходят с одинаковыми интервалами. Затем следует период OFF, когда источник не генерирует пакеты. Размер периодов ON и OFF является случайной величиной, которая, как показано в [5], должна иметь конечное математическое ожидание и бесконечную дисперсию. Предлагается длительность периодов ON и OFF рассматривать как случайную величину, распределенную по Парето по формуле $F(x) = K^\alpha x^{-\alpha}$, $1 < \alpha < 2$.

Элементами трасс самоподобного сетевого трафика являются время прихода пакета и его размер. Для генерации трасс самоподобного трафика может быть выбран параметр $\alpha = 1,4$, что соответствует реальному трафику в сетях Ethernet, согласно [5].

Сценарии модельных экспериментов. Работа заключается в наиболее полном рассмотрении различных аспектов работы транспортных протоколов, из которых выделяются два: корректность работы и производительность протоколов.

Один из используемых подходов состоит в моделировании и анализе транспортных протоколов с помощью раскрашенных сетей Петри, поскольку этот формализм является универсальным средством для исследования свойств различных систем [6].

Предлагаемый подход состоит в моделировании с помощью этого формализма протоколов TCP, а также элементов сетевой структуры с целью выполнения (симуляции) и проведения количественного анализа эффективности работы протоколов. Например, построенная в терминах раскрашенных сетей Петри модель протоколов TCP может быть использована для анализа корректности работы протоколов с помощью метода Model Checking. Вообще анализ корректности работы транспортных протоколов может быть сведен к проверке следующего правила: протокол должен передать от отправителя к получателю все требуемые для передачи данные за конечный период времени.

Анализ различных характеристик производительности транспортных протоколов в разных режимах работы является актуальной задачей [7, 8]. Основной подход состоит в создании некой тестовой сетевой инфраструктуры, нахождении в этой инфраструктуре «узких» мест и использовании самоподобных трасс для моделирования передачи трафика по этим каналам. Кроме самоподобного трафика, в моделируемой системе могут существовать и потоки, генерируемые с помощью алгоритмов TCP. Целью работы является сравнительный анализ различных параметров производительности этих протоколов (например, размер средних длин очередей маршрутизаторов, справедливость и эффективность распределения сетевых ресурсов).

С модели в терминах раскрашенных сетей Петри можно снимать различные показатели производительности. Однако, на наш взгляд, специализированные прикладные пакеты для моделирования работы протоколов подходят гораздо лучше [9]. Наиболее перспективным специализированным пакетом для исследования различных характеристик производительности протоколов является пакет Network Simulator [10] (далее – ns).

Ns позволяет строить модели коммуникационных сетей практически любой сложности. Одной из главных сильных сторон пакета является то, что модели протоколов, реализованные в этом пакете, были представлены либо непосредственными разработчиками протоколов, либо построены под их руководством. Представленные модели протоколов прошли квалифицированную экспертную оценку на соответствие спецификациям и реальным реализациям в операционных системах. Таким образом, эти модели с большой долей вероятности можно считать «эталонными».

В исследованиях работы транспортных протоколов важную роль должны играть современные версии протокола TCP, а также его экспериментальные модели. Внимание, в первую очередь, уделяется работе этих протоколов для непрерывной передачи больших массивов данных за один сеанс. В реальных сетях, например, в Интернете, такому режиму соответствует, в частности, передача файлов с помощью протокола FTP. Однако для исследования важны и другие режимы, которые моделируют, например, интерактивный трафик. Такая ситуация возникает, когда с помощью транспортного протокола передается html-страница от web-сервера клиенту, который обычно представлен браузером. Такая страница может содержать изображения, которые передаются с помощью отдельных транспортных соединений. В совокупности при передаче такой страницы образуется

несколько транспортных потоков, по каждому из которых передается относительно небольшое количество информации. Так, в [7] было показано, что передача такого вида данных создает значительно большую нагрузку на коммуникационную сеть (например, на буферы маршрутизаторов), чем передача небольшого количества объемных файлов того же размера. Таким образом, при анализе производительности транспортных протоколов необходимо обязательно учитывать поведение приложения, использующего сервис транспортного уровня.

Другим полезным средством является использование трасс трафика, причем как моделируемых, так и реальных. Моделируемые трассы могут быть получены, например, с помощью средств самого ns. Несмотря на то, что ns позволяет моделировать сети, состоящие из нескольких тысяч узлов, использование трасс помогает избежать моделирования одновременной работы большого количества информационных потоков в коммуникационных сетях. Кроме того, система ns позволяет использовать при анализе моделей трассы, которые были получены с помощью измерения реального трафика.

Современные коммуникационные сети являются в большей части гетерогенными – в них работают различные протоколы, в том числе и транспортные. Более того, практически невозможно обеспечить переход сразу всей сетевой инфраструктуры для использования более новой версии какого-либо транспортного протокола. Поиски решения оптимизации протокола TCP можно вести по двум направлениям. Модифицировать сам протокол, адаптируя его для новых условий и требований, или изменять сетевую среду, делая ее более дружелюбной по отношению к TCP. Любое изменение протокола должно обеспечить обратную совместимость, чтобы миллионы «старых» программ могли по-прежнему работать в этой среде. Поэтому большой интерес представляет исследование производительности транспортных протоколов в средах, где работают различные версии протоколов. Это необходимо для того, чтобы быть уверенным, что внедрение нового транспортного протокола или его версии не создаст проблем в сети.

ЛИТЕРАТУРА

1. http://www.isoc.org/inet2000/cdproceedings/2d/2d_2.htm
2. http://www.cs.ucla.edu/NRL/hpi/tcpw/tcpw_papers.html
3. Jain M., Dovrolis K. End-to-end available bandwidth: measurement, dynamics, and relation with TCP throughput // ACM SIGCOMM2002. - August, 2002. - P. 295-309.
4. Taqqu M., Willinger W., Sherman R. Proof of a fundamental result in self-similar traffic modeling // Computer Communication Review. - 1997. - № 4.
5. Leland W., Taqqu M., Willinger W., Wilson D. On the self-similar nature of Ethernet traffic // IEEE/ACM. Transactions on Networking. - 1994, February. - Vol. 2(1). -P. 1-15.
6. Крылов, В.В. Теория телеграфика и ее приложения [Текст]/ В.В. Крылов, С.С. Самохвалова. – СПб.: БХВ-Петербург, 2005. – 288 с.
7. Youngmi Joo, Ribeiro V., Feldmann A., Gilbert A., Willinger W. TCP/IP traffic dynamics and network performance: A lesson in workload modeling, flow control, and trace-driven simulations // ACM Computer Communication Review. - April, 2001.
8. Mathis M., Semke J., Mahdavi J., Ott T. The macroscopic behaviour of the TCP congestion control algorithm // ACM Computer Communication Review. - July, 1997.
9. Сысоев, П. А. Моделирование процессов оценки скорости передачи данных в стеке протоколов TCP/IP [Текст]/ П. А. Сысоев, В. Т. Еременко // Изв. ОрелГТУ. Информационные системы и технологии, 2006. – № 2. – С. 174-182.
10. Ns-2 network simulator. - <http://www.isi.edu/nsnam/ns/>.

Еременко Владимир Тарасович

Орловский государственный технический университет, г. Орел
Доктор технических наук, профессор, зав. кафедрой «Проектирование
и технология электронных и вычислительных систем»
Тел.: +7(4862) 41-57-65
E-mail: vladimir@orel.ru

Сысоев Павел Анатольевич

Орловский государственный технический университет, г. Орел
Аспирант кафедры «Информационные системы»
Тел.: +7(4862) 36-53-55
E-mail: spam@orel.ru

Засимов Алексей Сергеевич

Орловский государственный технический университет, г. Орел
Аспирант кафедры «Информационные системы»
Тел.: +7(4862) 76-75-06
E-mail: axcelhost@mail.ru