КОМБИНИРОВАННЫЕ ПОДХОДЫ К УПРАВЛЕНИЮ ПЕРЕГРУЗКАМИ В СЕТЯХ TCP/IP

Аноприенко А. Я., Рычка С. В.
Кафедра ЭВМ ДонНТУ
anoprien@cs.dgtu.donetsk.ua, rsv@donntu.edu.ua

Abstract

Anoprienko A., Richka S. The combined approaches to congestions control in TCP/IP networks. *The differences between methods of congestion control in TCP/IP networks, based on network and transport layers, are considered. Researches show, that the best are the combined methods of congestion control.*

Введение

Проблема перегрузки в сетях TCP/IP возникает в случае, когда количество передаваемых данных начинает приближаться к значению допустимой пропускной способности сети. При этом ухудшаются основные показатели качества обслуживания. Эти ухудшения могут выражаться в увеличении числа потерянных пакетов и времени задержек. Управление перегрузками является актуальной задачей, так как количество конечных пользователей глобальной сети, а, следовательно, объемы передаваемых данных, постоянно увеличиваются. Растет также доля мультимедийного трафика реального времени, влияние перегрузок на который особенно критично. Задача механизмов управления перегрузками заключается в том, чтобы поддерживать количество данных, передаваемых по сети, ниже уровня, при котором пропускная способность сети начинает резко падать, ограничивая потоки входящего и исходящего трафика.

В случае перегрузки производительность сети, как правило, меньше, чем производительность сети, в которой действуют методы управления перегрузками. Это было наглядно показано в [1]. На рисунке 1 представлена зависимость интенсивности исходящего трафика и времени задержки от интенсивности входящего трафика, где Pmax – максимальная пропускная способность канала, Рвх – интенсивность входящего трафика, Рвых – интенсивность исходящего трафика, Тз – время задержки.

Данные графики показывают качественный характер реакции производительности и задержки на изменение входной нагрузки. График под номером 1 характеризует идеальное поведение сети при увеличении

нагрузки, когда на выходе обеспечивается максимальная пропускная способность.

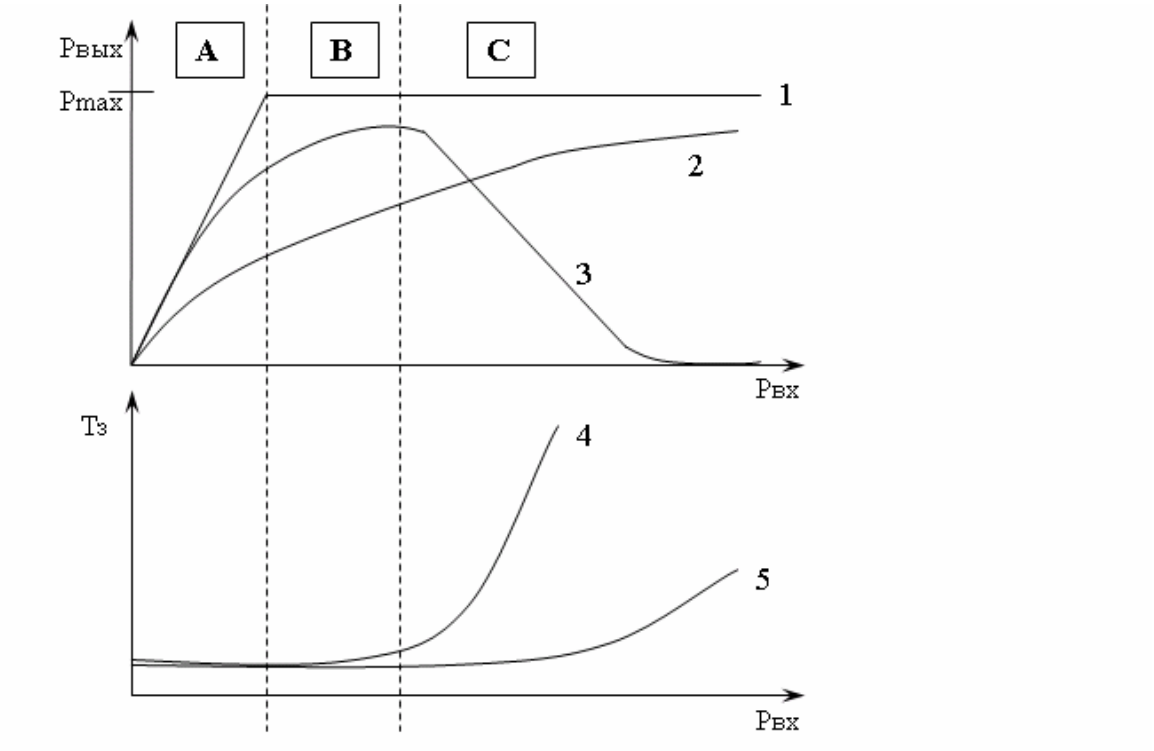

Рис. 1. Зависимость интенсивности исходящего трафика и времени задержки от интенсивности входящего трафика

Графики 2 и 4 соответствуют отсутствию механизмов управления перегрузками, а 3 и 5 – соответствуют режиму с наличием механизмов управления перегрузками и управления потоком. Отрезок А соответствует низкой нагрузке на сеть, режим В называется управляемой перегрузкой, когда сеть еще справляется с нагрузкой, хотя и с увеличенной задержкой. При отсутствии механизмов управления перегрузками на отрезке С происходит переполнение приемных буферов приемников, результатом чего является отбрасывание пакетов и попытки их повторной передачи. В результате задержки начинают резко увеличиваться, а производительность начинает стремиться к нулю.

В управлении перегрузками в сетях TCP/IP участвуют как промежуточные узлы сети, через которые проходит трафик, так и конечные системы, обменивающиеся данными. Существующие на данный момент механизмы управления перегрузками разделяются на механизмы, базирующиеся в хостах (реализация в протоколе TCP) и базирующиеся в маршрутизаторах (реализация в протоколе IP). Наиболее удачным решением проблемы управления перегрузками являются комбинированные методы, базирующиеся на взаимодействии механизмов сетевого и транспортного уровней.

Механизмы управления перегрузками в протоколе TCP

На данный момент существует целый ряд реализаций методов управления перегрузками в протоколе TCP, различающиеся в применимости к решениям различных задач. Фактически первой реализацией алгоритмов управления перегрузками в TCP была реализация TCP Tahoe, предложенная Якобсоном в 1988 году. В 1989 году в [2] были определены требования к реализации механизмов работы канального, сетевого и транспортного уровней, в частности, алгоритм управления перегрузками в TCP. Сейчас большинством операционных систем используется реализация TCP Reno, предложенная Якобсоном в 1990 году, отличающаяся от TCP Tahoe усовершенствованным алгоритмом “быстрая повторная передача” и добавлением алгоритма “быстрое восстановление”.

Существуют два параметра, используемые в алгоритмах управления перегрузками в TCP/IP, реализованных в протоколе TCP [3]:

– rwnd (receiver window) – окно приемника – количество данных, которое приемник может принять подряд;

– cwnd (congestion window) – окно перегрузки – ограничивает количество данных, которое источник может послать приемнику без подтверждения доставки.

Параметр rwnd находится в поле заголовка сегмента TCP и сигнализирует о том, сколько места свободно в приемном буфере приемника. С его помощью приемник может управлять скоростью передачи данных источником. На базе этого параметра в TCP реализована функция управления потоком.

Посредством изменения окна перегрузки количество переданных данных источником может варьироваться. Именно параметр cwnd реализует функцию управления перегрузкой в протоколе TCP. Среднюю пропускную способность соединения TCP можно приблизительно оценить по формуле:

$$BW = cwnd * MSS / RTT,$$

где BW – пропускная способность канала связи, MSS – максимальный размер сегмента, RTT – время возврата сегмента. Таким образом, с помощью управления окном перегрузки можно влиять на пропускную способность соединения.

С помощью cwnd приемник также может управлять скоростью передачи источником, например, отбросив один из принятых сегментов TCP. Это вызовет уменьшение окна перегрузки, и источник будет посылать меньше данных без подтверждения. Так поступают некоторые алгоритмы управления трафиком.

Механизм управления перегрузкой в протоколе TCP состоит из следующих алгоритмов: аддитивное увеличение – мультипликативное снижение, медленный старт, предотвращение перегрузки, быстрая повторная передача, быстрое восстановление. Основные принципы реализации этих алгоритмов и методы управления трафиком, реализованные в протоколе TCP, более подробно рассмотрены в [4].

Необходимо отметить, что существующие на данный момент реализации протокола TCP/IP ориентированы на гарантированную доставку данных без ограничений на параметры задержки и вероятности потери пакета. Протокол TCP/IP не имеет средств непосредственного определения состояния сети. Таким образом, механизм управления перегрузками в протоколе TCP не может эффективно управлять перегрузкой и не обеспечивает необходимое качество обслуживания при высокой интенсивности входящего трафика.

Преимущества использования комбинированных механизмов управления перегрузкой

Под комбинированными механизмами управления перегрузкой будем понимать совместное использование алгоритмов на сетевом и транспортном уровне. В работе [5] в 1984 году было предложено использовать комбинированные механизмы управления перегрузками, базирующиеся на сетевом и транспортном уровнях, в частности, использование отбрасывания пакетов приемником без посылки ICMP-сообщения источнику (ICMP-подавление), использующееся сейчас во многих комбинированных механизмах управления перегрузкой. Протокол передачи данных TCP реализуется на транспортном уровне только на конечных системах. Промежуточные маршрутизаторы используют только 3 нижних уровня модели OSI (самый верхний – сетевой уровень IP), так что они не могут непосредственно управлять нагрузкой, создаваемой конечными системами в сети. Источник должен каким-то образом получить информацию от промежуточного маршрутизатора о потенциальной или возникшей перегрузке и уменьшить поток передаваемых данных. Так как маршрутизатор неспособен изменить данные TCP на транспортном уровне, единственным способом извещения хоста-источника остается использование сигнальных битов в протоколе IP на сетевом уровне. Этот механизм управления перегрузками называется базирующимся в маршрутизаторах и работает по принципу обратной связи. Совместное использование механизма, базирующегося в маршрутизаторах, с механизмом, реализованном в протоколе TCP, позволяет добиться хороших результатов при решении проблемы управления перегрузками в сетях TCP/IP. Такое совмещение используется в одном из новых механизмов TCP ENC.

Комбинированный механизм управления перегрузками ECN

В 1994 году было предложено использовать новый механизм ECN (Explicit Congestion Notification) – явное уведомление о перегрузке, в первую очередь призванный управлять нагрузкой соединений TCP, чувствительных к задержкам и потерям пакетов. При этом не используется сброс пакета для подавления источника передачи данных в качестве индикации перегрузки. В случае использования ECN передатчик должен быть проинформирован о перегрузке заранее, то есть до ее наступления, а не по факту самой перегрузки. Зачастую информацией о перегрузке в сети обладают маршрутизаторы, работающие на 3-м уровне модели OSI, через которые проходит поток данных соединения TCP. В связи с этим для реализации ECN было необходимо использовать некоторые биты в IP-заголовке. В 1999 году [6] было предложено добавить возможность реализации ECN в IP-заголовок, а в 2001 году [7] реализация ECN была дополнена и стандартизирована. На данный момент реализация ECN имеет экспериментальный статус. ECN использует два бита (6 и 7) поля ToS (Type of Service) – тип услуги:

– бит ECT (ECN-capable transport) – бит 6: источник устанавливает значение бита в 1 для информирования приемника о том, что он использует протокол TCP с поддержкой ENC;

– бит CE (congestion experienced) – бит 7: маршрутизатор устанавливает значение бита в 1, информируя приемник о возникновении перегрузки.

Источник и приемник должны договориться о совместном использовании механизма ECN при установке соединения, в результате ECT-биты во всех последующих пакетах будут установлены в 1. В результате каждый из маршрутизаторов на пути соединения будет учитывать, что источник и приемник используют механизм ECN. При возникновении перегрузки на любом из маршрутизаторов, он путем установки значения CE-бита в 1 информирует приемник о перегрузке. Приемник в свою очередь, получив от маршутизатора единичный CE-бит, информирует источник о перегрузке путем посылки сегмента подтверждения с установленным флагом ENC-Echo. Этот флаг располагается в заголовке TCP и использует 9-й бит в поле reserved. После получения сегмента с установленным флагом ENC-Echo источник уменьшает значение размера окна перегрузки cwnd. Получение

информации о перегрузке источник индицирует путем посылки приемнику сегмента с установленным флагом CWR (congestion window reduced) – уменьшение размера окна перегрузки, для которого используется 8-й бит поля reserved заголовка TCP. Таким образом, в механизме ECN наблюдается совместное использование сетевого и транспортного уровней модели OSI для управления перегрузкой – на сетевом уровне сигнализируется перегрузка от маршрутизатора, а конечные устройства используют сигнализацию с помощью обратной связи на транспортном уровне.

Управление трафиком как часть механизма управления перегрузками

Одной из основных задач при управлении перегрузками и обеспечении необходимого качества обслуживания является управление трафиком. В общем виде, под управлением трафиком понимается совокупность алгоритмов, реализованных как аппаратно, так и программно, призванных обеспечить требуемое качество обслуживания. Управление трафиком включает в себя сетевое планирование, в результате которого определяется топология сети и пропускная способность линий, и оптимизацию, подразумевающую непосредственно эффективное распределение трафика. С другой стороны, борьба с перегрузкой подразумевает эффективное использование сети при высоком уровне нагрузки. Это условие обеспечивается при использовании механизмов управления трафиком, которые используют более сложные алгоритмы, по сравнению с базовыми методами управления перегрузкой, реализованных в TCP. Рассмотрим основные направления реализации данных алгоритмов.

В последнее время были разработаны наиболее чувствительные и эффективные методы управления трафиком и борьбы с перегрузками, представляющие собой две взаимодействующие структуры, работающие на уровне IP: интегрированные службы (Integrated Service, IS) и дифференцированные службы (Differentiated Services). Впервые архитектура IS была предложена в 1994 году [8], а архитектура DS – в 1998 году [9]. Назначение полей DS в заголовках IPV4 и IPV6-пакетов было описано в [10]. Интегрированные службы, реализованные в сетевых узлах, сначала изучают суммарные требования трафика. Далее, они ограничивают поддерживаемый трафик объемами, соответствующими текущим возможностям сети и резервируют ресурсы для предоставления определенного уровня качества обслуживания в соответствии с заданными требованиями. Структура дифференцированных служб, наоборот, не пытается заранее резервировать сетевые ресурсы. Вместо этого трафик классифицируется по группам. Каждая группа соответствующим образом помечается, а услуга, предоставляемая сетевыми элементами, зависит от

членства в той или иной группе. При этом пакеты, которые относятся к разным группам, обслуживаются по-разному. На данный момент службы IS и DS реализуются в коммутаторах и маршрутизаторах. Принципы реализации и преимущества использования данных служб описаны в [11].

Комбинированные механизмы борьбы с перегрузками и управления трафиком в TCP/IP, реализуемые в маршрутизаторах на программном уровне

Зачастую “узким местом” в процессе обмена данными с глобальной сетью является канал связи между организацией-потребителем и поставщиком услуг связи Internet (ISP – Internet Service Provider). Это объясняется тем, что ISP предоставляют конечному пользователю определенную полосу канала передачи данных и очередь конечного размера на своем сетевом узле. Этот механизм гарантирует только то, что пользователь будет получать данные в необходимом объеме и при превышении лимита скорости часть данных может быть помещена в очередь. В случае перегрузки канала связи и, следственно, переполнении очереди в сетевом узле ISP, пакеты начинают теряться, а задержки начинают расти. Это приводит к резкому падению производительности сети на стороне клиента.

Потери пакетов существенно влияют на скорость передачи данных, вне зависимости от величины пропускной способности канала связи. Потери пакетов можно свести к минимуму за счет увеличения размера очередей. Следовательно, возрастает скорость передачи данных. Поэтому ISP зачастую используют очереди большого объема. С другой стороны, при увеличении размеров очередей возникают задержки в трафике реального времени. Пакеты, подготовленные к отправке, помещаются в очередь для исходящего трафика, и обычно проходит до секунды, пока они будут получены удаленным узлом. Помимо задержек, накладываемых задержкой в исходящей очереди и канале передачи данных, накладывается задержка, вызванная временем пребывания пакета в большой входящей очереди на стороне ISP. Так как очереди поставщика услуг не подвластны клиентской стороне, они должны быть организованы на маршрутизаторе со стороны клиента.

Одним из предлагаемых решений проблемы является использование на стороне клиента маршрутизатора, имеющего программное обеспечение, позволяющее регулировать потоки входящего и исходящего трафика относительно клиентской стороны не только на уровне IP, как это делают

практически все аппаратные коммутаторы и маршрутизаторы, но и на уровне TCP. Такой подход значительно расширяет возможности управления трафиком и борьбы с перегрузками, по сравнению с методами, построенными на взаимодействии механизмов IP, реализованных в сетевых узлах, и TCP, реализованных в конечных станциях. Одним из главных преимуществ программных решений управления трафиком, по сравнению с IP-ориентированными алгоритмами, базирующимися в маршрутизаторах, является возможность осуществлять более эффективное управление на уровне TCP. Такие программные решения, в основном, используются в операционных системах с открытым программным кодом. Авторами, для предотвращения перегрузок и обеспечения необходимого качества обслуживания для сети ДонНТУ, используются механизмы управления трафиком, реализованные в ядре ОС Linux. Использование программных методов управления трафиком позволяет на уровне TCP/IP:

– управлять полосой пропускания канала для разных компьютеров;

– разделять полосу пропускания канала между различными типами трафика;

– изменять приоритет трафика в полосе пропускания канала, согласно нуждам;

– ограничивать полосу пропускания, как входящего, так и исходящего трафика.

Основные методы управления трафиком на уровне TCP/IP в программных маршрутизаторах подразделяются на следующие [12]:

– ограничение исходящего трафика (shaping);

– планирование (scheduling);

– классификация (classifying);

– ограничение входящего трафика (policing);

– уничтожение (dropping);

– маркирование (marking).

Процесс планирования определяет, из какой очереди следующий пакет будет обслуживаться маршрутизатором. По-сути, именно планировщик решает задачу разделения полосы пропускания исходящего канала связи в соответствии с заданными условиями. В общем случае, руководствуясь принципом “справедливого распределения ресурсов”, задача планировщиков может быть выражена в организации распределения пропускной способности r для каждого потока i по следующему принципу:

$$r[i] = BW / n,$$

где BW – пропускная способность рассматриваемого канала связи, n – количество активных потоков, разделяющих канал связи, причем $i \in [1,n]$.

Организация очереди позволяет управлять скоростью передачи данных. Можно непосредственно управлять лишь скоростью передачи

отправляемых данных. На сегодняшний день специфика глобальной сети Интернет такова, что очень сложно контролировать объем входящего трафика. Если маршрутизатор связывает локальную сеть с внешним миром и необходимо ограничивать скорость загрузки данных во внутреннюю сеть, то нужно это делать на внутреннем интерфейсе маршрутизатора, с которого данные передаются компьютерам в локальной сети, то есть относительно маршрутизатора это будут исходящие данные. Если ввести ограничение для скорости исходящего трафика на уровне, который немного меньше, чем пропускная способность канала, то можно ликвидировать исходящую очередь в сетевом узле ISP. Можно сказать, что исходящая очередь переместиться в маршрутизатор. Управление очередью исходящего трафика на стороне клиента имеет еще один очень важный момент. В этой очереди должна быть полоса для TCP-ACK-пакетов, которая должна иметь максимальный приоритет. Фактически это делается для того, чтобы входящий трафик не блокировался исходящим. Так как в случае высокой загрузки исходящего канала TCP-ACK-пакеты могут задерживаться или теряться, это может нарушить нормальную работу механизма обратной связи, управляющего перегрузками на уровне TCP за счет изменения окна перегрузки. Одной из самых распространенных на данный момент дисциплин обработки очереди исходящего трафика в ядре Linux является HTB (Hierarchical Token Bucket), предложенная Мартином Девера [13].

Концепция построения очередей строится на приложении нотации Кендалла, построенной на ряде допущений, к производным от формулы Литла. Для обозначения основных допущений нотация Кендалла выглядит в виде:

$$X/Y/N,$$

где X – распределение интервалов времени между поступлением запросов, Y – распределение времени обслуживания, N – количество серверов. Наиболее важное допущение касается скорости поступления запросов. Предполагается, что интервалы времени между поступлениями запросов распределены экспоненциально, а значит, могут быть подчинены распределению Пуассона. То есть запросы поступают случайным образом и не зависят друг от друга. В данной нотации предполагается, что время обслуживания также подчиняется экспоненциальному распределению или постоянно. Наиболее распространенной моделью, согласно нотации Кенделла, является M/M/1 – модель с одним сервером, в которой количество поступающих запросов распределено по Пуассону и с экспоненциальным распределением интервалов времени обслуживания [14]. Формула Литла для нескольких серверов выглядит так:

$$r = \lambda * T_r,$$
$$p = \lambda * T_s / N,$$

где r – среднее количество запросов в системе, находящихся в очереди и обслуживающихся, p – коэффициент использования сервера, λ – среднее число поступающих в секунду запросов, T_r – среднее время, которое запрос проводит в очереди и на стадии обслуживания, T_s – среднее время запроса на стадии обслуживания, N – количество серверов.

Ограничение скорости входящего трафика, по сути, является прямым механизмом борьбы с перегрузками и управления трафиком. С сетевыми устройствами в ОС Linux взаимодействуют дисциплины обработки очередей и все, что помещается в очередь к сетевому устройству, предварительно попадает в очередь дисциплины обработки очереди. Такая организация накладывает несколько ограничений:

– непосредственно ограничить скорость передачи данных можно только для исходящего трафика;

– нельзя определить общие ограничения, так как дисциплина обработки очереди классифицирует и обрабатывает поступающие данные для каждого сетевого устройства отдельно.

Управлять скоростью входящего трафика сложнее, чем исходящего, так как отсутствует возможность непосредственно влиять на скорость поступления данных. Относительно каждого сетевого интерфейса поток данных делится на входящий и исходящий, причем, трафик, который является входящим для одного сетевого устройства-приемника, для другого сетевого устройства-источника будет исходящим. А как было указано выше, исходящим трафиком можно управлять. В стадии экспериментальной поддержки на уровне ядра Linux сейчас находится дисциплина обработки очереди входящего трафика IMQ (Intermediate queueing device) [15], работающая по вышеуказанному принципу. Дисциплина IMQ пытается решить вышеуказанные проблемы и представляет собой механизм, работающий через псевдоинтерфейс. То есть на стороне сетевого устройства-приемника добавляется псевдосетевое устройство-источник. С помощью пакетных фильтров, встроенных в ядро ОС Linux или установленных в виде дополнительных модулей, можно классифицировать пакеты и часть из них направлять через псевдоинтерфейс, к которому могут подключаться различные дисциплины обработки очередей. Таким образом, можно управлять скоростью входящего трафика и осуществлять управление перегрузками.

Необходимо отметить, что задача регулирования потоков входящего и исходящего трафика не только на уровне IP, но и на уровне TCP, станет актуальной в обозримом будущем и для аппаратных маршрутизаторов, используемых в глобальной сети и на ее границах. Это обусловлено необходимостью модификации протокола TCP по разным причинам.

Одной из них можно назвать возможность возникновения ситуации повторения номеров сегментов в рамках одного и того же соединения, для соединений, проходящих по высокоскоростным каналам связи скоростью ≥ 1Гбит/с. В работе [16] рассматривается вышеуказанный вариант и предлагается модификация протокола TCP, при которой аппаратные маршрутизаторы должны поддерживать фильтрацию пакетов на 4-м уровне модели OSI, что еще раз подтверждает актуальность данной темы.

В качестве подтверждения изложенных выше данных был проведен эксперимент, суть которого заключалась в сравнении производительности сети в зависимости от входной нагрузки в реальных условиях при использовании комбинированных механизмов управления перегрузками и без них. В ходе эксперимента с удаленного сетевого узла генерировался трафик в виде пакетов размером 700 байт. Число передаваемых пакетов увеличивалось от 1 до 100. Контрольными точками измерения производительности были значения полученных пакетов в зависимости от генерации 25, 50, 75 и 100 пакетов соответственно. При генерации 100 пакетов, теоретически требуемая пропускная способность канала равнялась 560 Кбит/с, что составило 20% от суммарной загруженности канала, загруженного на 80%. Пусть Pr1 – количество полученных пакетов без использования механизмов управления перегрузками, Pr2 – количество полученных пакетов при использовании комбинированных механизмов управления перегрузками, Ps – количество сгенерированных и отправленных пакетов. В таблице 1 приведена зависимость количества принятых от количества переданных пакетов для обоих случаев.

Таблица 1
Зависимость количества принятых пакетов
от количества переданных пакетов

Ps	Pr1	Потери, %	Pr2	Потери, %
25	19	24	24	2
50	35	30	47	5
75	47	37	69	8
100	59	41	90	10

На рисунке 2 представлен график зависимости количества принятых пакетов от количества переданных пакетов, построенный по результатам эксперимента.

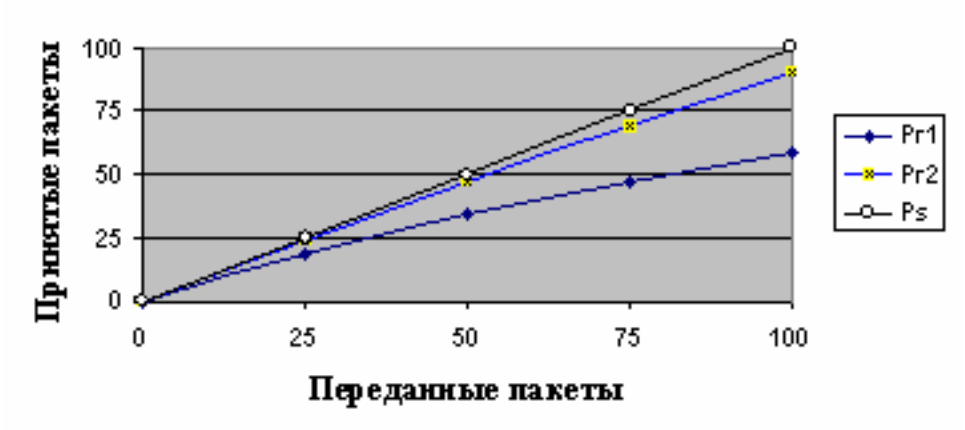

Рис. 2. График зависимости количества принятых пакетов от количества переданных пакетов

Полученные в результате эксперимента данные свидетельствуют о том, что при загрузке канала связи на 70-90% использование комбинированных механизмов управления перегрузками позволило исключить переполнения очередей маршрутизатора и потери пакетов, а также добиться увеличения производительности примерно на 30-40%.

Выводы

Управление перегрузками является одной из основных задач при обеспечении высокой производительности сети и требуемого качества обслуживания. Постоянное увеличение доли мультимедийного трафика, передающегося в реальном времени, ужесточает требования к механизмам управления трафиком и борьбы с перегрузками. На данный момент одной из главных целей алгоритмов управления перегрузками является поддержание высокой производительности сети при ее максимальной загрузке. В теории очередей есть грубое правило, гласящее о том, что 80-процентное заполнение очереди является предвестником перегрузки. Судя по рисунку 1, качественные зависимости производительности сети и задержки от входной нагрузки имеют схожие характеристики. Таким образом, еще одной задачей управления перегрузками в сетях TCP/IP можно определить регулирование входной нагрузки на сеть на уровне, не превышающем заданный порог, при котором велика вероятность перегрузки. Актуальность данной темы обуславливается увеличением роста запросов на доступ к сетевым ресурсам по сравнению с увеличением пропускной способности каналов связи и постоянным расширением глобальной сети. Реализация механизмов управления перегрузками особенно критична на стыках крупномасштабных, разветвленных локальных сетей, какой является сеть ДонНТУ, с глобальной сетью Internet, так как пропускная способность внутри локальной сети, зачастую,

как минимум на порядок выше, чем скорость канала связи с глобальной сетью.

На данный момент базовые алгоритмы рассмотренных механизмов управления перегрузками и управления трафиком используются авторами для повышения продуктивности использования канала связи, объединяющего сеть ДонНТУ с глобальной сетью. При этом предлагается использование комбинированных механизмов борьбы с перегрузками и управления трафиком, реализованных в программных маршрутизаторах для повышения эффективности работы канала связи между конечными системами и глобальной сетью, который зачастую является “узким местом”. Проведенные на базе Internet-узла ДонНТУ эксперименты показали, что ограничение интенсивности входящего трафика из сети Internet на уровне 70-90 % и использование комбинированных механизмов управления перегрузками позволяет значительно уменьшить переполнение очередей маршрутизатора и количество потерянных пакетов, а также добиться увеличения производительности примерно на 30-40 % за счет отсутствия повторных передач потерянных пакетов.

Литература

1. Gerla M., Kleinrok L. “Flow Control: A Comparative Survey”. IEEE Transactions on Communications, vol. 28 (4), 1980, pp. 553-574.
2. Braden R. Requirements for Internet Hosts - Communication Layers. RFC1122, Oct. 1989, http://www.ietf.org/rfc/rfc1122.txt?number=1122.
3. Allman M., Paxson V., Stevens W. TCP Congestion Control. RFC2581, April 1999, http://www.ietf.org/rfc/rfc2581.txt?number=2581.
4. Стивенс Р. У. TCP/IP крупным планом. Главы 17-24, http://www.soslan.ru/tcp/.
5. Nagle J. Congestion control in IP/TCP internetworks. RFC896, January 1984, http://www.ietf.org/rfc/rfc0896.txt?number=896.
6. Ramakrishnan S., Floyd S. A Proposal to add Explicit Congestion Notification (ECN) to IP. RFC2481, January 1999, http://www.ietf.org/rfc/rfc2481.txt?number=2481.
7. Ramakrishnan K., Floyd S., Black D. The Addition of Explicit Congestion Notification (ECN) to IP. RFC3168, September 2001, http://www.ietf.org/rfc/rfc3168.txt?number=3168.
8. Braden R., Clark D., Shenker S. Integrated Services in the Internet Architecture: an Overview. RFC1633, June 1994, http://www.ietf.org/rfc/rfc1633.txt?number=1633.
9. Blake S., Black D., Carlson M., Davies E., Wang Z., Weiss W. An Architecture for Differentiated Service. RFC2475, December 1998, http://www.ietf.org/rfc/rfc2475.txt?number=2475.

10. Nichols K., Blake S., Baker F., Black D. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC2474, December 1998, http://www.ietf.org/rfc/rfc2474.txt?number=2474.
11. Столлингс В. Современные компьютерные сети. 2-е изд. – СПб.: Питер, 2003. – с. 527-570.
12. Traffic control HOWTO. Traditional elements of traffic control. http://www.tldp.org/HOWTO/Traffic-Control-HOWTO/elements.html.
13. HTB - Hierarchical Token Bucket, http://luxik.cdi.cz/~devik/qos/htb/.
14. Столлингс В. Современные компьютерные сети. 2-е изд. – СПб.: Питер, 2003. – с. 226-243.
15. IMQ - Intermediate Queueing Device, http://www.linuximq.net/.
16. Семенов Ю. А. Новый протокол TCP. Электронный научный журнал “Исследовано в России”, 2006, http://zhurnal.ape.relarn.ru/articles/2006/025.pdf.

Дата надходження до редакції 12.10.2006 р.