ОБЕСПЕЧЕНИЕ КАЧЕСТВА ОБСЛУЖИВАНИЯ В СЕТИ ПЕРЕДАЧИ ДАННЫХ

Батыр С.С., Хорхордин А.В.
Донецкий национальный технический университет, г. Донецк,
кафедра автоматики и телекоммуникаций
e-mail: sbatyr@fcita.dn.ua

Abstract

Batyr S.S., Horhordin A.V. Guaranteeing quality of service in computer networks. *Analysis of QOS was made in this article. Also SNMP was analyzed. Special SNMP parameters for routers were chosen. Linear approximation of exponential RED was presented.*

Общая постановка проблемы. Одной из ключевых задач при обеспечении заданного качества обслуживания в сети передачи данных является управление трафиком.

Под управлением трафиком понимается совокупность алгоритмических средств, реализованных как аппаратно, так и программно, направленных на обеспечение функционирования рассматриваемой сети с требуемым качеством обслуживания и эффективным использованием ресурсов. С точки зрения топологии сети, управление трафиком включает в себя сетевое планирование и оптимизацию. Сетевое планирование является процессом, в результате которого определяется топология сети и пропускная способность линий (каналов), при этом должны приниматься во внимание объемы предполагаемого трафика, возможные темпы его роста и т.п. Под оптимизацией подразумевается управление распределением трафика в существующей сети.

Постановка задач исследования. Как и в любой пакетной сети, обеспечивающей качество обслуживания, для реализации последнего необходимо использование определенной функции «управления допустимостью соединения». Данная функция является набором действий, осуществляемых сетью на фазе установления нового соединения (или восстановлении соединения) с целью определения, возможна ли ею поддержка с требуемыми параметрами и качеством обслуживания или нет. Важнейшей частью процесса обработки пакетов с целью обеспечения качества обслуживания в сети Интернет, является алгоритм управления очередью в буферах сетевого оборудования. Алгоритм управления очередью – это набор методов, управляющих поступлением, хранением и передачей на обслуживание поступающих в систему пакетов. Выбор алгоритма управления очередью является очень сложным, т.к. каждый тип этих алгоритмов обладает как преимуществами, так и недостатками. В связи с тем, что трафик и, как следствие, качество его обслуживания является разнородным, очевидно наличие приоритетов при его обслуживании. Пакеты с различными приоритетами в маршрутизаторах должны помещаться в различные очереди, в связи с чем задача распределения времени центрального процессора маршрутизатора становится актуальной.

Опыт создания и эксплуатации сетей передачи данных показал, что система динамического управления потоками данных является обязательным элементом сетевого оборудования. Под «управлением потоками» будем понимать совокупность механизмов, управляющих доступом к сетевым ресурсам и обеспечивающих согласование параметров сети и пользовательского трафика. Задачей системы управления потоками является предотвращение и устранение перегрузок и разрешение тупиковых ситуаций. При этом система управления потоками должна функционировать так, чтобы был обеспечен принцип «справедливого распределения ресурсов» и гарантированы требуемые показатели качества обслуживания для пользователей. Под перегрузкой понимается такое состояние сети, при котором основные показатели качества обслуживания существенно ухудшаются. В

зависимости от типа сети эти ухудшения могут выражаться в увеличении числа потерянных пакетов и в увеличении средних значений и джиттера задержки. Перегрузки могут возникать как на отдельных участках сети (локальные перегрузки), так и распространяться на всю сеть (глобальные перегрузки).

В сети Интернет состояние перегрузки характеризуется резким ростом размеров очередей в отдельных узлах. В случае, если отсутствует механизм контроля/управления перегрузкой, или он не справляется с возложенными на него задачами, это может привести к невозможности доступа пользователей к ресурсам отдельных участков, а в отдельных случаях — к ресурсам всей сети в целом. Такое состояние определяется как блокировка сети. Производительность сети (число обслуженных пакетов) стремится к нулю, задержки — к бесконечности.

Также перегрузку для сети Интернет можно определить как такое состояние сетевых узлов, когда сеть не может гарантировать требуемое качество обслуживания для уже установленных соединений, а также и для устанавливаемых соединений. Как правило, перегрузка может быть вызвана флуктуациями потоков трафика или выходом из строя какого-либо сетевого элемента, что может привести как к несоблюдению обязательств сети по обеспечению качества обслуживания существующих соединений, так и невозможностью установления нового соединения с запрошенным качеством обслуживания.

Необходимо отметить фактор влияния профиля нагрузки на качество функционирования механизмов. Внедрение таких новых приложений в сети Интернет как передача компрессированной речи, видео, и, в целом, мультимедийный трафик, определяет новые требования к методике контроля и управления трафиком. Практически каждое новое приложение предъявляет свои собственные требования по показателям качества обслуживания. В первую очередь, именно эти факторы и определяют необходимость разработки новых методов и механизмов управления потоками и борьбы с перегрузками.

Управление ресурсами в сети Интернет затруднено, в первую очередь, из-за того, что по своей природе эта сеть является децентрализованной и состоит из множества автономных систем с индивидуальным управлением. Поэтому для такой гетерогенной сети задачу управления ресурсами разумно решать совместно «из-конца-в-конец» и локально (в рамках каждого сетевого узла). В действительности определено два базовых класса механизмов управления нагрузкой в сети Интернет: базирующихся в хостах (host-based) и базирующихся в маршрутизаторах (router-based).

Как известно, в сети Интернет протоколы передачи данных реализованы «из-конца-в-конец» на транспортном уровне, т.е. реализуются только в хостах. В связи с этим у маршрутизаторов отсутствует возможность каким-либо образом влиять на создаваемую хостами нагрузку (в маршрутизаторе реализуются только три нижних уровня в соответствии с моделью osi или TCP/IP). С другой стороны, источник должен иметь возможность получать каким-либо образом информацию о наличии перегрузки в транзитном сетевом узле. Т.к. сетевой узел не имеет реализации транспортного уровня, т.е. не может модифицировать поток, то единственным возможным решением в данной ситуации остается посылка источнику служебных пакетов с информацией о наступившей (пли грозящей) перегрузке. Причем источником этой информации является сеть, а приемником - источник нагрузки. Такой механизм называется «обратной связью» (feedback) и относится к классу router-based. Совместная реализация обоих классом механизмов управления перегрузкой позволяет добиться достаточно хороших результатов. Для совместной реализации необходимо, чтобы механизм управления перегрузкой в хосте имел поддержку семантики обратной связи, т.е. должен быть реализован дополнительный односторонний протокол маршрутизатор — хост. При получении от маршрутизатора информации о перегрузке на сети механизм управления нагрузкой, реализованный в хосте, должен понизить скорость передачи пакетов в сеть. Без реализации механизма «обратной связи» источник не способен принимать решение об из-

менении скорости передачи. Источник нагрузки может получать информацию при помощи «обратной связи» как прямо от сети, так и через хост-приемник нагрузки.

Неявная «обратная связь» подразумевает, что источник нагрузки получает информацию через хост-приемник и процесс мониторинга параметров передачи нагрузки (задержки, потери) осуществляется обоими хостами и они же ответственны за определение состояния сети. и. соответственно, за определение скорости передачи. Достаточно сложной задачей в данном случае является точность принятия решения о состоянии сети, и, следовательно, точность изменения скорости передачи. Самой простой и широко используемой в реальной сети Интернет причиной функционирования «обратной связи» является потеря пакета: сеть через приемник неявным образом информирует источник нагрузки о потере пакета. Однако нельзя утверждать, что потеря пакета является следствием перегрузки, например. в беспроводных сетях потеря может быть вызвана ошибкой в радиоканале. Существует ряд других методов для реализации «обратной связи», построенных, например, на базе измерения скорости получения пакетов приемником или на базе измерения задержки «из-конца-в-конец».

Явным преимуществом этого типа «обратной связи» является простота реализации в маршрутизаторах, т.к. нет необходимости вносить какие-либо дополнительные функциональные возможности.

Механизм явной «обратной связи» должен явно информировать источник нагрузки о состоянии сети. Существует два типа явной «обратной связи»: «уведомление о перегрузке» и «индикация скорости». Возможности механизма достаточно сильно ограничены тем, что его информация переносится, как правило, в заголовках пакетов, а их размер и количество неиспользуемых бит ограничено. Явная «обратная связь» может быть как бинарной — «уведомление о перегрузке», так и многозначной — обычно количество значений ограничено несколькими, т.е. существует возможность сообщить источнику нагрузки об определенном уровне перегрузки. Реализацией этого механизма для TCP/IP сетей являются Explicit Congestion Notification и ICMP Source Quench.

Не принимая во внимание тип сети, в рамках которой мы хотим реализовать поддержку качества обслуживания (корпоративная сеть, сеть доступа или магистральная), передачу данных «из-конца-в-конец» можно представить просто как прохождение пакетов через множество каналов и их обработку множеством маршрутизаторов. Таким образом, в предельном случае, задачу обеспечения качества обслуживания можно свести к задаче обработки (перенаправления) пакетов маршрутизаторами.

Рассмотрим детально параметры и события, которые влияют на прохождение пакетов по сети передачи данных.

Одним из основных параметров является задержка пакета при передаче «из-конца-в-конец». Она может быть определена как сумма задержек в линиях маршрута и времени обработки пакета на каждом узле. Задержки, вносимые линиями передачи, достаточно легко прогнозируются, т.к. они построены на таких известных технологиях как SDH, SONET, ATM и пр., в то время как время обработки пакета в узле можно лишь оценить снизу и сверху (определить минимальное и максимальное значения). Эти значения зависят от множества факторов как в отдельности, так и вместе, например, программного обеспечения узла, уровня нагрузки в нем, информации о перегрузке на направлении и т.д.

Теоретическая задержка пакета «из-конца-в-конец» для некоторой гипотетической сети может быть оценена как сумма средних задержек на каждом этапе передачи между транзитными узлами и среднего времени обработки пакета в каждом узле. Кроме того, время передачи по каналам примерно фиксировано, поэтому если для пакетов некоторого потока обеспечить верхний предел значения задержки в каждом узле маршрута «из-конца-в-конец», то можно гарантировать, что время доставки пакета «из-конца-в-конец» не будет превышать некоторого значения. Джиггер задержки также является важным параметром при

обеспечении качества обслуживания, особенно для мультимедийных приложений реального времени. Значение джиттера определяет дисперсию задержки, или, другими словами, стабильность параметров маршрута «из-конца-в-конец».

Основным фактором ухудшения качества обслуживания является неуправляемая потеря пакетов. В современных телекоммуникационных фиксированных сетях, построенных на базе протокола IP, подавляющее количество потерь пакетов происходит в буферах маршрутизаторов.

Потери вызываются, в первую очередь, перегрузками. Буферы маршрутизаторов имеют ограниченный размер, поэтому при высокой нагрузке вероятность того, что буфер будет полностью занят пакетами, стремительно растет. Если в момент времени, когда в маршрутизатор поступает новый пакет, все пространство буфера занято, то новый пакет удаляется, т.е. происходит потеря.

Состояние маршрутизатора можно определить с помощью средств сетевой диагностики. Первичная диагностика возложена на протокол ICMP. Также возможно управление параметрами маршрутизатора в ходе его работы. Учитывая важность функции управления, для этих целей было создано два протокола SNMP (Simple Network Management Protocol, RFC-1157, -1215, -1187, -1089 разработан в 1988 году) и CMOT (Common Management Information services and protocol over TCP/IP, RFC-1095, в последнее время применение этого протокола ограничено). Обычно управляющая прикладная программа воздействует на сеть по цепочке SNMP-UDP-IP-Ethernet.

Протокол SNMP работает на базе транспортных возможностей UDP (возможны реализации и на основе TCP) и предназначен для использования сетевыми управляющими станциями. Он позволяет управляющим станциям собирать информацию о положении в сети Интернет. Протокол определяет формат данных, а их обработка и интерпретация остаются на усмотрение управляющих станций или менеджера сети. SNMP-сообщения не имеют фиксированного формата и фиксированных полей. При своей работе SNMP использует управляющую базу данных (MIB - management information base, RFC-1213, -1212).

Алгоритмы управления в Интернет обычно описывают в нотации ASN.1 (Abstract Syntax Notation). Все объекты в Интернет разделены на 10 групп и описаны в MIB: система, интерфейсы, обмены, трансляция адресов, IP, ICMP, TCP, UDP, EGP, SNMP. В группу "система" входит название и версия оборудования, операционной системы, сетевого программного обеспечения и пр.. В группу "интерфейсы" входит число поддерживаемых интерфейсов, тип интерфейса, работающего под IP (Ethernet, LAPB etc.), размер дейтограмм, скорость обмена, адрес интерфейса. IP-группа включает в себя время жизни дейтограмм, информация о фрагментации, маски субсетей и т.д. В TCP-группу входит алгоритм повторной пересылки, максимальное число повторных пересылок и пр..

Вся управляющая информация для контроля ЭВМ и маршрутизаторами Интернет концентрируется в базе данных MIB (Management Information Base, RFC-1213 или STD0017). Именно эти данные используются протоколом SNMP. Система SNMP состоит из трех частей: менеджера SNMP, агента SNMP и базы данных MIB. Агент SNMP должен находиться резидентно в памяти объекта управления. SNMP-менеджер может быть частью системы управления сетью NMS (Network Management System), что реализуется, например, в маршрутизаторах компании CISCO (CiscoWorks).

MIB определяет, например, что IP программное обеспечение должно хранить число всех октетов, которые приняты любым из сетевых интерфейсов, управляющие программы могут только читать эту информацию.

Согласно нормативам MIB управляющая информация делится на восемь категорий:

1. system Операционная система ЭВМ или маршрутизатора.
2. Interfaces Сетевой интерфейс.
3. addr.trans Преобразование адреса (напр., с помощью ARP).

4. IP Программная поддержка протоколов Интернет.
5. ICMP Программное обеспечение ICMP-протокола.
6. TCP Программное обеспечение TCP-протокола.
7. UDP Программное обеспечение UDP-протокола.
8. EGP Программное обеспечение EGP-протокола.
9. SNMP Программное обеспечение SNMP-протокола.

Описание сетевых интерфейсов маршрутизатора согласно MIB имеет следующий вид:

Таблица 1. Переменные **IFtable** (интерфейсы)

Переменная описания интерфейсов (iftable)	Тип данных	Описание	ifEntry
IFindex	integer	Список интерфейсов от 1 до ifnumber.	1
IfDescr	displaystring	Текстовое описание интерфейса.	2
IfType	integer	Тип интерфейса, например, 6 - ethernet; 9 – 802.5 маркерное кольцо; 23 - PPP; 28 - SLIP.	3
IfNumber	integer	Число сетевых интерфейсов.	
IfMTU	integer	mtu для конкретного интерфейса;	4
IfSpeed	gauge	Скорость в бит/с.	5
IfPhysaddress	physaddress	Физический адрес или строка нулевой длины для интерфейсов без физического адреса (напр. последовательный).	6
IfAdminStatus	[1...3]	Требуемое состояние интерфейса: 1 - включен; 2 - выключен; 3 - тестируется.	7
IfOperStatus	[1...3]	Текущее состояние интерфейса: 1 - включен; 2 - выключен; 3 - тестируется.	8
IfLastchange	timeticks	Sysuptime, когда интерфейс оказался в данном состоянии.	9
IfInOctets	counter	Полное число полученных байтов.	10
IfInUcastpkts	counter	Число пакетов, доставленных на верхний системный уровень (unicast).	11
IfInNUcastpkts	counter	Число пакетов, доставленных на верхний системный уровень (not-unicast).	12
IfInDiscads	counter	Число полученных но отвергнутых пакетов.	13
IfInErrors	counter	Число пакетов, полученных с ошибкой;	14
IfInUnknown Protos	counter	Число пакетов, полученных с ошибочным кодом протокола;	15

IfOutOctets	counter	Число отправленных байтов;	16
IfOutUcastPkts	counter	Число unicast- пакетов, полученных с верхнего системного уровня;	17
IfOutNucastPkts	counter	Число мультикастинг- и широковещательных пакетов, полученных с верхнего системного уровня;	18
IfOutDiscads	counter	Количество отвергнутых пакетов из числа отправленных;	19
IfOutErrors	counter	Число отправленных пакетов, содержащих ошибки;	20
IfOutQlen	gauge	Число пакетов в очереди на отправку;	21

В результате проведенного анализа баз данных MIB были выделены наиболее важные поля, которые позволяют задавать параметры процесса управления очередью. Из [3] были выделены параметры, позволяющие задать тип используемого алгоритма для управления очередью входного буфера маршрутизатора, а также задать индивидуальные параметры алгоритма wRED, который может использоваться для предотвращения перегрузки.

```
caqQueueThreshDropAlgorithm OBJECT-TYPE
   SYNTAX INTEGER {
      tailDrop(1),
      wred(2)
      }
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
      "Indicates the drop algorithm used at this queue and threshold.
      tailDrop(1) indicates that tailDrop is used.
      wred(2) indicates that WRED is used."
   ::= { caqQueueThresholdEntry 4 }

caqQueueThreshDropThreshold OBJECT-TYPE
   SYNTAX Gauge(1..100)
-- Units
--   percent
   ACCESS read-write
   STATUS mandatory
   DESCRIPTION
      "This object specifies the drop threshold parameter for a
      pair of queue and threshold of an interface queue type when the
      drop algorithm is tail drop. Once the packets in the buffer is
      more than the value of this object, the incoming packets of the
      buffer are dropped. The value is a percentage of the full
      buffer.

      This object is instantiated only if the value of
```

```
        caqQueueThreshDropAlgorithm is tailDrop(1)."
      ::= { caqQueueThresholdEntry 5 }

caqQueueThreshMinWredThreshold OBJECT-TYPE
    SYNTAX Percent
--   Rsyntax INTEGER(0..100)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This object specifies the min WRED threshold parameter of a
        threshold number for the specific port type when WRED drop
        algorithm is used.
        WRED (Weighted Random Early Detect) is a mechanism which drops
        packets fairly during congestion so that adaptive applications
        can react to congestion. This object specifies a percentage of
        the buffer size.
        This object is instantiated only if the value of
        caqQueueThreshDropAlgorithm is wred(2)."
      ::= { caqQueueThresholdEntry 6 }

caqQueueThreshMaxWredThreshold OBJECT-TYPE
    SYNTAX Gauge(1..100)
--  Units
--    percent
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This object specifies the max WRED threshold parameter of a
        threshold number for the specific port type when WRED drop
        algorithm is used.
        This object is instantiated only if the value of
        caqQueueThreshDropAlgorithm is wred(2)."
      ::= { caqQueueThresholdEntry 7 }
```

Данные параметры доступны для чтения и записи по сети в режиме mandatory, то есть возможно динамическое управление и изменение параметров алгоритма управления с соблюдение требований безопасности, предъявляемых к сетевому оборудованию.

В работе [4] были разработаны математическая и имитационная модели участка сети передачи данных с маршрутизатором, которая учитывала механизм AIMD (Additive Increase – Multiply Decrease) для TCP-соединений передачи данных, а также предложено усовершенствование алгоритма RED – замена кусочно-линейного полинома вычисления вероятности потери пакета экспоненциальной кривой.

Результаты работы [4] показали, что теоретически предложенное усовершенствование должно уменьшить количество потерянных пакетов и обеспечить до 10% прироста скорости передачи данных. Но в силу лицензионных ограничений, непосредственное изменение алгоритмов управления очередями, заложенное в аппаратное обеспечение маршрутизаторов, невозможно.

Поэтому для проверки алгоритма предложено аппроксимировать экспоненциальную кривую набором прямых, являющихся касательными к кривой. И в зависимости от уровня

загрузки входного буфера изменять параметры стандартного алгоритма RED для получения аппроксимации экспоненциальной кривой.

Для стандартной реализации алгоритма RED необходимо задать уровни pmax, Tmin и Tmax (из [4]). Выражение для определения вероятности отбрасывания пакета при использовании экспоненциальной кривой из [4]:

$$p(q) = e^{-K_{AQM}(1-q/Q)^2},$$

тогда параметры уравнения касательной вида $y=k*(q/Q) + b$ в произвольной точке q при условии $q \geq 0$ и $q \leq Q$ будут:

$$k = p'(q) = -K \cdot 2(1-q/Q) \cdot (-1) \cdot e^{-K_{AQM}(1-q/Q)^2}$$

$$b = p(q) - k * \left(q/Q \right)$$

Соответственно параметры pmax, Tmin и Tmax с учетом накладываемых на них ограничений примут значения:

$$p\max = \begin{cases} (k+b) \ если \ (k+b) < 1 \\ 1 \end{cases},$$

$$T\min = \begin{cases} -b/k \ если \ -b/k > 0 \\ 0 \ если \ -b/k \leq 0 \end{cases},$$

$$T\max = \begin{cases} 1 \ если \ \dfrac{1-b}{k} > 1 \\ \dfrac{1-b}{k} \ если \ \dfrac{1-b}{k} \leq 1 \end{cases}.$$

Динамическое изменение параметров алгоритма RED позволит обеспечить более качественную передачу данных по сравнению с фиксированными параметрами.

Выводы. Таким образом, в статье:

1. Произведен анализ сети передачи данных для обеспечения качества обслуживания. Выделены основные элементы сети и показатели качества.

2. Проанализированы способы динамического управления сетью и произведен отбор необходимых параметров.

3. Предложен алгоритм динамического управления маршрутизатором с целью улучшения качества обслуживания.

Литература

1. Кучерявый Е. А. Управление трафиком и качество обслуживания в сети Интернет. — СПб. Наука и Техника, 2004. — 336 с: ил
2. http://book.itep.ru
3. CISCO-QOS-PIB-MIB.txt, http://www.cisco.com
4. Батыр С.С., «Анализ и моделирование процессов в сетях передачи данных», квалификационная работа магистра, ДонНТУ, Донецк, 2005.