УДК 004.72(045)

Кудзиновская И. П.

АНАЛИЗ МЕТОДОВ ОБЕСПЕЧЕНИЯ КАЧЕСТВА ОБСЛУЖИВАНИЯ В ВЫСОКОСКОРОСТНЫХ КОМПЬЮТЕРНЫХ СЕТЯХ

Институт компьютерных технологий
Национальный авиационный университет

Проанализированы методы обеспечения качества обслуживания в сетях. Рассмотрено влияние свойства самоподобия трафика на показатели QoS. Обоснована необходимость разработки алгоритмов формирования трафика и предотвращения перегрузки с учетом самоподобия трафика

Введение

Современная тенденция конвергенции сетей различных типов, а также увеличение объема трафика и появление приложений, работающих в режиме реального времени, мультимедийных приложений привели к необходимости переноса сетью различных видов трафика, в том числе, чувствительного к задержкам [1]. Поэтому традиционные *TCP/IP* сети не гарантируют необходимое приложениям качество обслуживания и возникает необходимость в разработке дополнительных средств предоставления приложениям требуемого уровня сервиса.

Качество обслуживания

Под качеством обслуживания *(Quality of Services, QoS)* понимают [2] интегральный полезный эффект от обслуживания, который определяется степенью удовлетворения пользователя как от полученной услуги, так и от самой системы обслуживания. Критерий качества обслуживания представляют в виде интегрального показателя совершенства обслуживания, учитывающего не только качество услуги, но и способность сети обрабатывать нагрузку.

Услуга может предоставляться с различными уровнями качества. Приемлемый уровень качества услуги согласовывается между сервис-провайдером и его клиентом и включается в текст соответствующего сервисного соглашения *(Service Level Agreement - SLA).* Уровень качества услуги задается конкретным набором значений определяющих параметров качества услуги. Выделяют следующие уровни: желаемый, гарантированный, измеренный. Трафареты сервисного соглашения относительно уровня оказываемых услуг, темплеты *SLA (Service Level Agreement Templates)* представляют собой определения стандартных уровней услуги, которые могут быть предложены покупателям услуги в рамках *SLA*. Например, могут быть предложены трафареты, которые определяют характеристики так называемой «золотой услуги» или «серебряной услуги» и т.п.

Требования к качеству обслуживания приложений различных типов

Внедрение механизмов *QoS* предполагает обеспечение со стороны сети соединения с определенными ограничениями по производительности, основными характеристиками которой являются [3] полоса пропускания, задержка, джиттер и уровень потери пакетов. Характеристики *QoS* особенно важны в том случае, когда сеть передает одновременно трафик разного типа, например, трафик вебприложений и голосовой, так как различные типы трафика предъявляют разные требования к характеристикам *QoS*. Учесть одновременно все характеристики *QoS* для всех видов трафика сложно, поэтому виды трафика, существующие в сети, классифицируют, относя каждый к одному из распространенных типов, а затем пытаются достичь одновременного выполнения определенного подмножества из набора требований для этих типов трафика.

В качестве основных критериев классификации приняты три характеристики трафика [1]: относительная пред-

сказуемость скорости передачи данных, чувствительность трафика к задержкам пакетов и чувствительность трафика к потерям и искажениям пакетов (рис. 1).

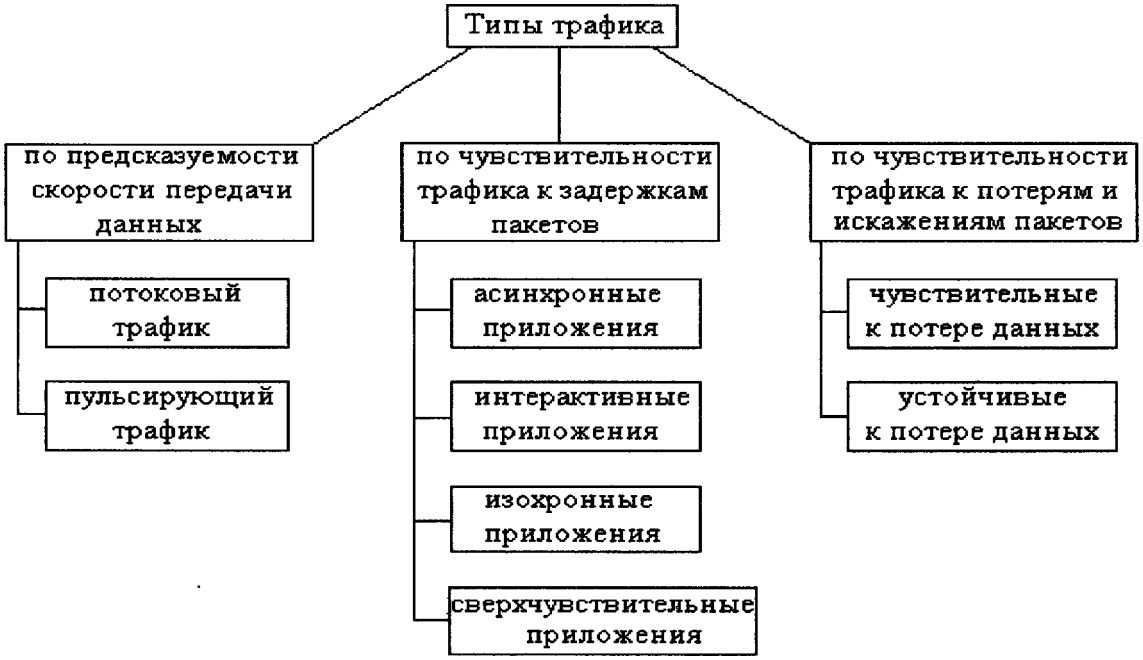

Рис. 1. Классификация типов сетевого трафика

Приложения с потоковым трафиком порождают равномерный поток данных, который поступает в сеть с постоянной битовой скоростью *(CBR)*. При использовании метода коммутации пакетов трафик таких приложений представляет собой последовательность пакетов одинакового размера (равного *B* бит), следующих друг за другом через один и тот же интервал времени *T*. *CBR* может быть вычислена путем усреднения на одном периоде: $CBR = B/T$ бит/с.

Приложения с пульсирующим трафиком отличаются высокой степенью непредсказуемости, когда периоды молчания сменяются пульсацией, в течение которой пакеты «плотно» следуют друг за другом. В результате трафик имеет переменную битовую скорость *(VBR)*. Практически любой трафик, даже трафик потоковых приложений, имеет ненулевой коэффициент пульсации (для пульсирующего трафика – от 2:1 до 100:1, для потокового – приблизительно 1:1).

К асинхронным приложениям относятся приложения, которые практически не имеют ограничений на время задержки (эластичный трафик), пример – электронная почта.

Интерактивные приложения – это приложения, на функциональности которых задержки не сказываются негативно, например – текстовый редактор, работающий с удаленным файлом.

Изохронные приложения имеют порог чувствительности к вариациям задержек, превышение которого резко снижает функциональность приложения, пример – передача голоса.

Функциональность сверхчувствительных к задержкам приложений задержка сводит к нулю, пример – приложения, управляющие техническим объектом в реальном времени.

Приложения, чувствительные к потере данных, – приложения, передающие алфавитно-цифровые данные (текстовые документы, коды программ, числовые массивы и т. п.). Все традиционные сетевые приложения (файловый сервис, сервис баз данных, электронная почта и т. д.) относятся к этому типу приложений.

Приложения, устойчивые к потере данных, – приложения, передающие трафик с информацией об инерционных физических процессах. Их устойчивость к потерям объясняется тем, что небольшое количество отсутствующих данных мож-

но определить на основе принятых. К этому типу относится большая часть приложений, работающих с мультимедийным трафиком (аудио- и видеоприложения). Однако процент потерянных пакетов не может быть большим (например, не более 1 %).

Механизмы обеспечения качества обслуживания

С точки зрения экономической целесообразности необходимо [2] стремиться к наиболее полной загрузке сетевых ресурсов, чтобы передавать в обусловленные промежутки времени как можно большие объемы данных. Но пульсации трафика, существующие в пакетных сетях, не позволяют добиться качественного обслуживания при нагрузках, приближенных к максимальным для данной сети. Сеть работает эффективно, когда каждый её ресурс существенно загружен, но не перегружен. Следовательно, с одной стороны, необходимо стремиться к улучшению качества обслуживания трафика, т.е. стараться снизить задержки в продвижении пакетов, уменьшить потери пакетов и увеличить интенсивности потоков трафика, с другой стороны, необходимо стараться максимально увеличить загрузку всех ресурсов сети с целью повышения экономических показателей. Компромисс в достижении вышеупомянутых целей необходимо искать [2] на пути использования средств и механизмов борьбы с перегрузками в сети, а именно:

– осуществлять рациональную настройку параметров сетевого оборудования с целью недопущения бесконтрольного увеличения интенсивности входных потоков;

– реализовывать алгоритмы управления очередями, оптимизированные к условиям работы сетевого оборудования и к условиям *SLA*;

– оптимизировать пути прохождения трафика через сеть, пытаясь максимизировать загрузку дорогостоящих элементов сети при соблюдении заданных требований к качеству обслуживания потоков данных.

На данный момент существует несколько вариантов реализации *QoS* в сетях, но каждый из них не оптимален.

Для обеспечения качества обслуживания в рамках сетевых элементов используются следующие средства *QoS* [4]:

– классификация, идентификация и маркирование потоков;

– управление перегрузкой, организация очередей, дифференцированное обслуживание потоков;

– избежание перегрузок, предотвращение заполнения очередей, а также принятие мер для общего снижения вероятности перегрузок;

– повышение эффективности канала, методы уменьшения задержек на низкоскоростных каналах;

– управление сетевым трафиком, сетевое планирование и оптимизация.

Для осуществления функций контроля и управления интенсивностью трафика, а также обеспечения качества обслуживания существуют специальные алгоритмы [4], которые основываются на принципе "корзины маркеров" *(Token Bucket)* или его модификациях. Этот алгоритм имеет два режима функционирования – полисинг *(traffic-policing)*, при котором происходит сбрасывание неконформной нагрузки, а также шейпинг *(traffic-shaping)*, буферизующий неконформные пакеты. Алгоритм полисинга используется для измерения и управления интенсивностью трафика. Профиль трафика задается согласованным размером всплеска трафика B_c за определенный интервал времени T_c. При этом интенсивность генерирования маркеров (*CIR*) определяется как $CIR = B_c / T_c$.

Алгоритм шейпинга, в отличие от полисинга, пакеты, не соответствующие заданному профилю, не отбрасывает, а буферизует и обрабатывает при первой возможности. Это позволяет уменьшить потери при дальнейшей обработке трафика, но задержки, которые вносятся, ограничивают применение алгоритма для систем обработки информации реального времени.

Таким образом, шейпинг и полисинг можно классифицировать как методы статического задания пропускной способности.

Алгоритмы управления очередями – это механизмы борьбы с перегрузками в сетях. Наиболее распространенным механизмом обслуживания очередей является алгоритм *FIFO*. Он достаточно эффективен, но не предусматривает [3] приоритетной обработки чувствительного к задержкам трафика путем его перемещения во главу очереди, проведения действий по предотвращению перегрузки или уменьшению размера очереди для снижения времени задержки.

Алгоритм произвольного раннего обнаружения *(Random Early Detection, RED)* [1, 3] позволяет предотвратить перегрузку сети путем превентивного отбрасывания пакетов для уведомления о возможной перегрузке источников *TCP*-соединения с помощью механизма сквозного адаптивного управления с обратной связью. Этот метод позволяет смягчить эффект от потери пакетов при больших нагрузках. Данный алгоритм, изначально разработанный для протокола *TCP*, может быть применим к трафику любого протокола, когда сеть не гарантирует доставки. Модификация этого алгоритма – взвешенный алгоритм произвольного раннего обнаружения *(Weighted Random Early Detection – WRED)*, позволяющий настраивать различные *RED*-параметры в зависимости от значения поля *IP*-приоритета или класса трафика. Алгоритм *WRED* на основе потока *(flow WRED)* представляет собой расширение алгоритма *WRED*, предусматривающее возможность назначения штрафа с ненулевой вероятностью тем потокам, которые пытаются завладеть слишком большой долей доступных ресурсов. Алгоритм явного уведомления о перегрузке *(Explicit Congestion Notification, ECN)* позволяет предупредить *TCP*-источник о начинающейся перегрузке сети путем маркировки (а не отбрасывания) пакетов.

Метод приоритетных очередей [1, 5] используется для обслуживания трафика, чувствительного к задержкам и имеющего небольшую интенсивность, например, голосового. При обслуживании трафика, чувствительного к задержкам, но имеющего большую интенсивность, например, видеотрафика, качество обслуживания других типов трафика будет очень низким. В отличие от приоритетного, заказное обслуживание очередей [3] обеспечивает минимальную полосу пропускания для каждого типа трафика.

Очереди на основе классов *(Class Based Queuing, CBQ)* – это алгоритм, при котором трафик делится на несколько классов. Каждый класс имеет собственную очередь и ему выделяется некоторая часть пропускной способности канала.

Взвешенная справедливая очередь *(Weighted Fair Queuing, WFQ)* [3] – частный случай *CBQ*, когда классам соответствуют независимые потоки. Выделение дополнительной пропускной способности для больших потоков позволяет уменьшить задержку при их обработке.

Интегрированная служба *IntServ* и дифференцированная служба *DifServ* были разработаны [1, 3, 5] для предоставления качества обслуживания в сетях *Internet*. Эти службы не определяют специальные протоколы маршрутизации или их выполнение, они представляют собой методологии или архитектуры, добавляющие маршрутизаторам новую функциональность, позволяющую запрашивать уровень *QoS* непосредственно из сети.

Архитектура *IntServ* предлагает два вида услуг: гарантированный сервис и сервис с максимальными усилиями. Каждый пакет связывается с потоком данных и механизм *IntServ* позволяет пользователю запросить необходимое качество обслуживания для всего потока, при этом обеспечивается предварительное планирование и резервирование ресурсов. В качестве сигнального протокола предлагается протокол резервирования ресурсов *(Resource Reservation Protocol - RSVP)*, который позволяет конечным приложениям, требующим определенные гарантированные услуги, проводить сквозную сигнализацию своих *QoS*-требований.

К недостаткам архитектуры *IntServ* относятся проблемы масштабирования, не позволяющие эффективно использовать ее в крупных сетях и, в особенности, в *Internet*, характеризующейся наличием десятков тысяч потоков трафика.

Другим способом обеспечения *QoS* в сетях является архитектура *DifServ*, которая была разработана с целью обеспечения поддержки легкомасштабируемых дифференцированных услуг в пределах *Internet*. Архитектура *DifServ* – модель, обеспечивающая параметры *QoS* не на базе потоков, а на основании требований различных групп пользователей, дифференцируя трафик по установленному номеру класса. Такой механизм снижает объем служебной информации по сравнению с архитектурой *IntServ*. Модель *DifServ* поддерживает три вида обслуживания: гарантированное обслуживание, обслуживание с предпочтением и сервис с максимальными усилиями. *DifServ* не требует сложного и дорогого оборудования в сети - в этом ее преимущество перед *IntServ*. Недостаток данной модели заключается в том, что, несмотря на высокий приоритет, данные все равно могут быть подвержены непредсказуемым задержкам при перегрузках в сети.

Дополняющим компонентом к *DifServ* является технология многопротокольной коммутации по меткам *MPLS* [1, 3], позволяющая оптимизировать распределение трафика с различными требованиями к качеству обслуживания и поддерживающая механизмы маркировки пакетов и управления очередями.

Влияние самоподобия трафика на качество обслуживания

Анализ многочисленных измерений информационных потоков на пакетном уровне [5, 6] указывает на специфическую природу процессов в компьютерных сетях, не укладывающуюся в рамки известных случайных моделей. Характерным для пакетного трафика являются обнаруженные на практике свойства самоподобия или масштабной инвариантности статистических характеристик.

В работе [6] проведено имитационное моделирование по оценке параметров *QoS* и показано, что с увеличением параметра Херста мультиплексированного потока процент потерянных пакетов *(drop)* и средняя задержка на *IP*-пакет (*delay*) возрастают (рис. 2). Также увеличиваются среднее значение джиттера на *IP*-пакет и коэффициент использования сети.

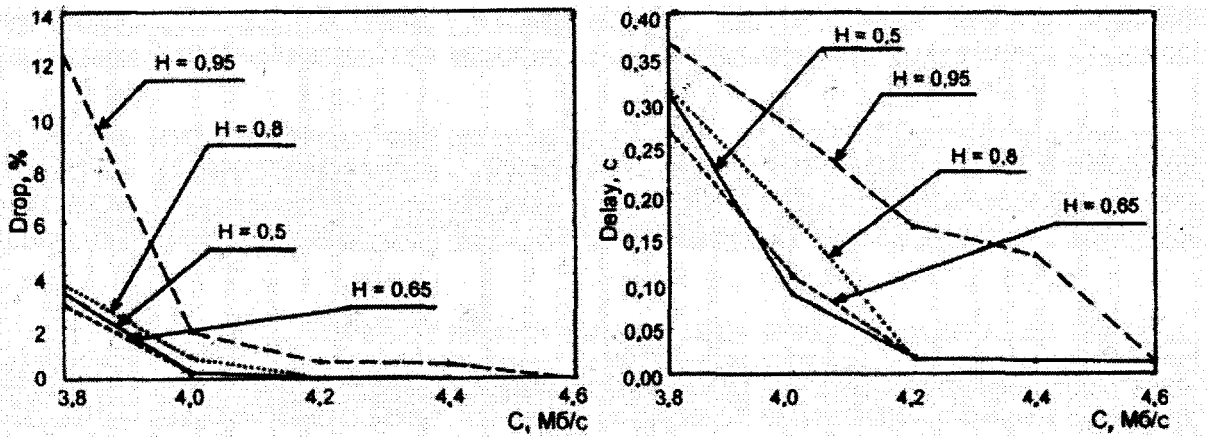

Рис. 2. Оценка влияния показателя Херста *(H)* мультиплексированного потока на показатели качества обслуживания

Таким образом, самоподобие речевого трафика ухудшает показатели качества обслуживания. Тем не менее, наличие свойства самоподобия позволяет разработать алгоритмы прогнозирования, которые смогут посредством анализа трафика на относительно небольшом отрезке времени предсказать его поведение на более длительных интервалах.

Алгоритмы "корзины маркеров" могут быть эффективными для трафика с относительно невысоким уровнем пачечности. В случае берстного самоподобного трафика профиль трафика будет выравниваться за счет значительных вносимых задержек, которые будут неприемлемы для большинства систем обработки информации реального времени. Поэтому необходимо модифицировать алгоритмы "корзины маркеров" с учетом влияния

эффекта самоподобия.

С точки зрения управления виртуальными соединениями необходимо определить моменты времени, в которые уровень сетевого процесса начинает превышать некоторое пороговое значение, определяемое эмпирически и зависящее от конфигурации конкретной сети и работы алгоритмов управления очередями, а также степени заполнения очередей в конкретных маршрутизаторах. В простейшем случае это максимально допустимый уровень трафика, выше которого произойдет переполнение очереди и, соответственно, начнутся потери пакетов. Поэтому необходим алгоритм для анализа роста очереди в буфере сетевого устройства с учетом свойства масштабной инвариантности статистических характеристик трафика, который позволит определить момент времени, когда необходимо запустить механизм подавления сверхактивных источников.

Выводы

На данный момент существует несколько вариантов реализации *QoS* в сетях, но каждый из них имеет недостатки, поэтому необходима разработка дополнительных средств предоставления приложениям требуемого уровня сервиса.

Свойства масштабной инвариантности статистических характеристик трафика отрицательно влияют на показатели качества обслуживания. Поэтому в дальнейшем необходимо разработать алгоритмы формирования трафика и предотвращения перегрузки с учетом самоподобия трафика, что позволит улучшить показатели *QoS*.

Список литературы

1. *Олифер В. Г., Олифер Н. А.* Компьютерные сети. – 3-е изд., – С.Пб.: Питер, 2006. – 957 с.

2. *Конахович Г. Ф., Чуприн В. М.* Сети передачи пакетных данных. – К.: МК-Пресс, 2006. – 272 с.

3. *Шринивас Вегешна* Качество обслуживания в сетях IP. – М.: Вильямс, 2003. – 368 с.

4. *Кучерявый Е. А.* Управление трафиком и качество обслуживания в сети Интернет. – С.Пб.: Наука и техника, 2004. – 336 с.

5. *Столлингс В.* Современные компьютерные сети. – СПб.: Питер, 2003. – 783 с.

6. *Осин А. В.* Имитационное моделирование процесса мультиплексирования цифровых потоков на выходе гибридных кодеков речи // Радиоэлектроника, электротехника и энергетика: Труды / Десятая международная научно-техническая конференция студентов и аспирантов. – М.: МЭИ, 2-3 марта 2004. – Т.1. – С. 131 – 132.