

МЕТОДЫ УПРАВЛЕНИЯ ТРАФИКОМ ПО ВИРТУАЛЬНОМУ ПУТИ

Введение

Максимальное обеспечение производительности сети и является целью управления трафиком, в которой численное значение наиболее просто можно проанализировать когда завершены переходные процессы, связанные с изменением таблиц маршрутизации.

Этот анализ позволяет исследовать качественное влияние структурных особенностей используемых систем управления на установившихся режимах работы сетевых приложений. В этом случае с определенной степенью точности на интервалах наблюдения, соизмеримых с временем переходных процессов в протоколах маршрутизации, можно считать, что поток поступающих пакетов является некоррелированным, а процесс обслуживания описывается показательным законом распределения.

Качество функционирования системы передачи зависит от величины задержки при передаче пакетов и наличия канала подтверждения [1]. В последнем случае в модель системы необходимо добавить механизм управления в виде контура обратной связи. При подтверждении каждого отдельного пакета процесс формирования сигнала обратной связи можно связать с изменением состояния сетевого соединения, которое характеризуется номером ожидаемого к приему пакета. Пакеты могут передаваться по виртуальному каналу без получения подтверждения пока их число в канале меньше некоторого числа. Такую модель управления виртуальным соединением с помощью скользящего окна можно представить в форме замкнутой системы массового обслуживания.

Постановка задачи

Проанализировать возможные методы управления трафиком по виртуальному пути. Построить графики распределения вероятности дополнительной задержки для модифицированной схемы.

Решение

Схема, которая предусматривает управление потоком на каждом участке каждого виртуального соединения (рисунок 1) – кредитная схема, где применяются методы избыточного кодирования данных в смежных пакетах сериях в сочетании с динамическим резервированием кредитов для рассматриваемого виртуального соединения.

На принимающих портах коммутаторов АТМ (асинхронный режим передачи) под конкретное виртуальное соединение резервируются определенные объемы буферной памяти. Таким образом, источник трафика

как бы получает кредит на передачу определенного числа ячеек, которую он может осуществить не дожидаясь какой-либо управляющей информации. Это число ячеек определяется объемами выделенных буферов. При возникновении перегрузки и переполнении буферной памяти одного из коммутаторов он посылает управляющее сообщение на предыдущий коммутатор с требованием прекратить передачу данных по "перегруженному" соединению. Тот прекращает передачу и начинает накапливать поступающие данные в своем буфере. Это продолжается до тех пор, пока не исчезнет перегрузка на первом участке и коммутатор не придет разрешение на возобновление передачи. Важно отметить, что при такой схеме управления перегрузками для каждого виртуального соединения должен быть выделен индивидуальный буфер. Кредитная схема управления перегрузками может предотвратить потерю ячеек, что очень важно для работы некоторых приложений. Она позволяет максимально эффективно использовать полосу пропускания канала и, кроме того, дает возможность проходящим по одному физическому каналу виртуальным соединениям работать на разных скоростях.

Однако рассматриваемая схема не снискала популярности, так как ее применение требует серьезной доработки оборудования АТМ с целью поддержки индивидуальных буферов и сложных алгоритмов динамического расчета буферного пространства при установлении коммутируемых виртуальных соединений (SVC). Кредитная схема не была одобрена Форумом АТМ [2] и не вошла в стандарты по управлению трафиком.

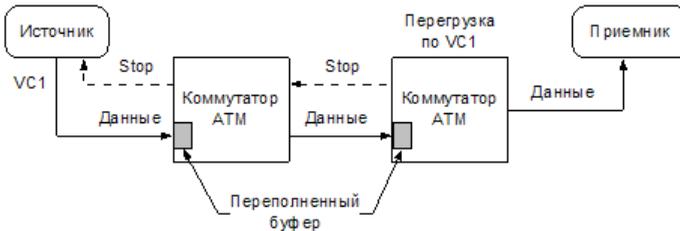


Рис. 1 – SVC схема управления потоками

Существует несколько скоростных схем управления перегрузками и все они предусматривают использование обратной связи для информирования источника о том, с какой скоростью в данный момент он может передавать ячейки по каждому виртуальному соединению [3].

FECN (прямое явное уведомление о перегрузке) – одна из скоростных схем управления перегрузками [1], использующая отрицательную обратную связь. Когда коммутатор АТМ испытывает перегрузку, он выставляет в проходящих через него ячейках бит EFCI (явное прямое уведомление о перегрузке) (находится в поле PFI заголовка ячейки), информируя тем самым приемник о перегрузке по конкретному виртуальному соединению. Получив сообщение о перегрузке, приемник направляет эту информацию источнику трафика.

В свою очередь источник принимает решение о снижении скорости передачи по “перегруженному” виртуальному соединению.

Принцип работы схемы FECN показан на рисунке 2. Испытывающий перегрузку коммутатор устанавливает во всех ячейках, проходящих по виртуальному соединению VC1, биты EFCI равными единице и тем самым информирует приемник о перегрузке по этому соединению. Получив данную информацию, приемник, чтобы проинформировать источник трафика о перегрузке, высылает ему специализированные ячейки управления ресурсами (RM). Получив ячейки RM, источник начинает снижать скорость передачи по соединению VC1.

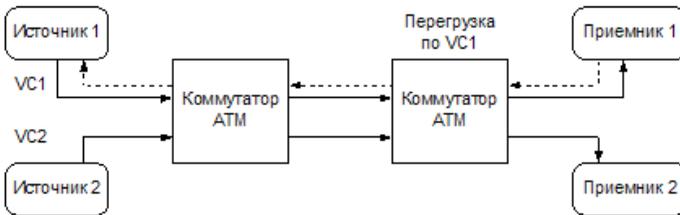


Рис. 2 – FECN схема управления потоками

По аналогичному принципу работает и схема BECN (обратное явное уведомление о перегрузке), но в ней предусмотрено, чтобы информацию о перегрузке направлял источнику трафика непосредственно сам испытывающий ее коммутатор рисунок 3. Очевидным преимуществом схемы BECN является более быстрая и, следовательно, более эффективная реакция на перегрузку. Но с другой стороны, при использовании этой схемы каждый транзитный коммутатор ATM должен уметь генерировать ячейки RM и вставлять их в проходящий поток данных.

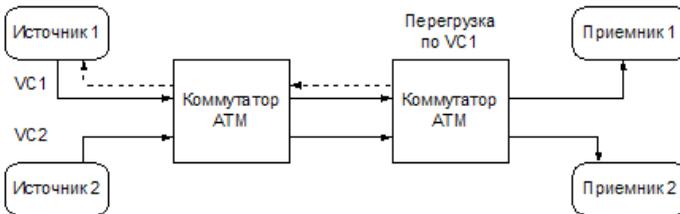


Рис. 3 – BECN схема управления перегрузками

В схемах FECN и BECN коммутатор ATM считается перегруженным, если очередь ячеек на обслуживание (коммутацию) превысит определенное пороговое значение. Получив информацию о перегрузке, источник должен начать снижение скорости передачи трафика, направляемого через перегруженный участок сети. Это снижение происходит до тех пор, пока продолжает поступать информация о перегрузке. Как только появление ячеек RM с информацией о перегрузке прекратится и они не будут

поступать в течение определенного интервала времени, источник может начать увеличение скорости передачи вплоть до ее максимального значения (PCR).

Если ячейки RM, направляющиеся к источнику трафика с информацией о перегрузке, сами попадают в перегруженный поток, то, вполне возможно, они никогда не смогут доставить эту информацию. Кроме того, сильно перегруженный коммутатор ATM может быть не в состоянии не только обслуживать транзитный трафик, но и генерировать ячейки RM. В обеих ситуациях сеть будет лишена информации о перегрузках. Это может усугубить положение дел в сети, ведь при отсутствии ячеек RM источник трафика начнет увеличивать скорость передачи, приближая ее к максимальной (PCR).

Таким образом, при использовании схем FECN и BECN появляется потенциальная возможность возникновения аварийной ситуации. Поэтому Форум ATM разработал более устойчивые схемы, например схему, основанную на алгоритме пропорционального управления скоростью (PRCA).

В отличие от предыдущих схем алгоритм PRCA основан на принципе положительной обратной связи, что позволяет избежать описанных выше проблем. При использовании PRCA источник трафика увеличивает скорость передачи только тогда, когда получает разрешение на это от приемника. В противном случае, т. е. при отсутствии такого разрешения, источник обязан последовательно снижать скорость пропорционально числу передаваемых ячеек.

Алгоритм PRCA работает следующим образом. В первой и в каждой N -й передаваемой источником ячейках бит *EFCT* устанавливается равным нулю, а в остальных ячейках - единице. Интервал N задается административно и определяет время реакции на перегрузку. Если приемник не перегружен, то в ответ на каждую ячейку с *EFCT*=0 он направляет источнику ячейку RM, разрешающую увеличить скорость передачи. Источник вправе увеличивать скорость только при получении такой ячейки. Испытывающий перегрузку коммутатор ATM может устанавливать бит *EFCT* равным единице (в ячейках, где он был равен нулю), запрещая тем самым приемнику генерировать ячейки RM, или же просто удалять ячейки RM, передаваемые по перегруженному каналу в направлении к источнику. В обоих случаях источник будет вынужден постепенно снижать скорость передачи до тех пор, пока не получит хотя бы одну ячейку RM.

С алгоритмом PRCA связана очевидная проблема. Она заключается в том, что при прохождении потока через несколько перегруженных участков число ячеек в нем с *EFCT*=1 будет значительно больше числа ячеек с *EFCT*=0 в других, не использующих этот алгоритм потоках. Следовательно, доступная данному потоку скорость (ACR) будет значительно ниже скоростей, доступных другим потокам. Даже если PRCA используют все потоки, то те из них, что прошли через разное число перегруженных участков, все равно будут иметь разную плотность битов *EFCT*=1, а, следовательно, и разную доступную скорость.

Подобных проблем не возникает при использовании усовершенствованного пропорционального алгоритма управления скоростью (EPRCA). В этом случае источник трафика посылает все информационные ячейки с битом *EFCI*, равным нулю. Через каждые *N* таких ячеек источник посылает ячейку *RM*, содержащую значения желаемой (обычно равна *PCR*) и текущей (*ACR*) скорости передачи показано на рисунке 4. Испытывающий перегрузку коммутатор *ATM* подсчитывает свое значение скорости *ACR*, которое зависит от того, какой объем трафика он может обработать в данный момент, и вставляет это значение в проходящую через него ячейку *RM*. Получив эту ячейку, приемник отправляет ее обратно источнику, а тот корректирует свою скорость в соответствии с новым значением скорости *ACR*. Таким образом, проходя по кругу, ячейка *RM* “собирает” информацию о наличии ресурсов со всех промежуточных коммутаторов и возвращается обратно к источнику со значением наименьшей доступной скорости для данного виртуального соединения.

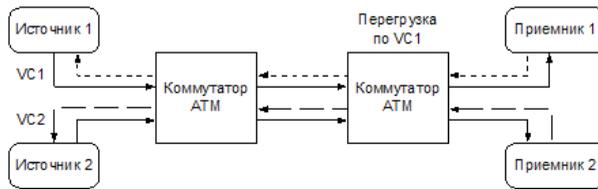


Рис. 4 – EPRCA схема управления перегрузками

Алгоритм EPRCA, как и алгоритм PRCA, обязывает источник трафика снижать скорость передачи при отсутствии ячеек *RM*, что избавляет сеть от возможных перегрузок, связанных с потерей этих ячеек. Он не имеет недостатков PRCA, так как скорость передачи трафика от источника не зависит от числа ячеек с *EFCI*=1, а задается в явном виде при передаче управляющей ячейки *RM*. Алгоритм EPRCA включен в принятый Форумом *ATM* стандарт *ATM* [2].

Дальше в статье рассмотрим подробно механизмы управления трафиком для модифицированной схемы и построим графики распределения вероятности дополнительной задержки.

Механизмы управления трафиком для модифицированной схемы должны включать в себя средства обратной связи для всех классов качества обслуживания (*QoS*), т.к. при необходимости нужно передавать источникам требования снизить скорость передачи, вплоть до *PCR*. Такая система обратной связи имеет и дополнительные преимущества. Одним из них является контроль параметров задержки со стороны источника, что позволяет предсказывать перегрузки в сети и предпринимать соответствующие меры, вплоть до заблаговременной ремаршрутизации и обхода перегруженного участка, что особенно актуально в случаях, когда перегрузки вызваны отказами промежуточных коммутаторов. Обход загруженного участка может также осуществляться для увеличения скорости передачи с целью минимизировать общую задержку передачи

трафика для данного соединения и более равномерного распределения загрузки по сети. Для реализации системы обратной связи целесообразно использовать стандартные – RM ячейки [3], а для передачи контрольных ячеек RM и управления цепью обратной связи целесообразно использовать стандартный алгоритм EPRCA.

CTD – задержка при передаче ячейки определяется как отрезок времени между двумя событиями – “входом ячейки” в точке приемнике и “выходом ячейки” из исходной точки. Параметр CTD характеризует максимальную задержку в сети при передаче ячеек от отправителя к получателю. Конкретная задержка складывается из задержек при передаче по линии связи между устройствами и задержек на каждом из промежуточных коммутаторов в АТМ:

$$CTD = T_0 + T_n, \quad (1)$$

где T_0 – время обслуживания в коммутаторах; T_n – суммарное время передачи.

С другой стороны,

$$CTD = CTD_{fixed} + CTD_{over}, \quad (2)$$

где CTD_{fixed} – фиксированное минимальное время доставки ячейки, которое определяется маршрутом и параметрами сети; CTD_{over} – дополнительная задержка, которая возникает из-за задержек в коммутаторах при загрузке или отказе последних (CDV).

Параметр CDV определяется разницей во времени между максимальной и минимальной задержкой при доставке ячеек от отправителя к получателю. Вариация задержки зависит от процедуры мультиплексирования в один физический канал связи потоков ячеек, принадлежащих множеству виртуальных соединений, и непостоянства задержки, вносимой очередями коммутаторов АТМ.

Для классической схемы параметр задержки (1) может быть представлен в следующем виде:

$$CTD = (T_{o..min} + T_n) + T_{o..don}. \quad (3)$$

При этом распределение вероятности возникновения дополнительной задержки выглядит следующим образом (согласно статистическим исследованиям форума в реальных сетях с помощью соответствующего оборудования) рисунок 5.

Для модифицированной схемы возможна минимизация задержек за счет минимизации времени передачи:

$$CTD = (T_{o..min} + \frac{1}{k} \cdot T_n) + T_{o..don}. \quad (4)$$

Тогда распределение вероятности возникновения дополнительной задержки будет выглядеть следующим образом рисунок 6.

Теоретическая минимально возможная задержка – это такое значение, которое принимала бы задержка при передаче данных с использованием максимально возможной полосы пропускания на данном маршруте.

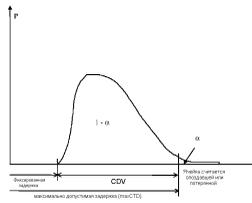


Рис. 5 – Распределение вероятности при возникновении дополнительной задержки согласно статистическим исследованиям форума

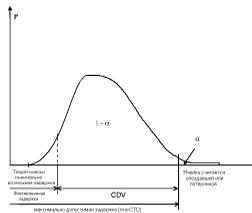


Рис. 6 – Распределение вероятности при возникновении дополнительной задержки

Кривизна кривой на участке возрастания и положение точки максимума по отношению к фиксированной задержке зависит от загрузки сети (т.к. она влияет на значения коэффициентов ускорения k_i), а также от составляющей доли трафика, поддерживающего режим ускорения. При этом возможен вариант, когда максимально вероятное время задержки будет меньше, чем теоретически вычисленная фиксированная задержка для классической схемы или равна ей. Подобный вариант, с точки зрения сети, означает минимальную загрузку коммутаторов, которая характеризуется отсутствием или минимальным значением отказов в обслуживании, а также минимальным уровнем потери ячеек. И, наоборот, при приближении времени задержки к области α , вероятность отказов в обслуживании и потери ячеек резко возрастают. Из вышесказанного вытекает, что в идеальном случае было бы не допустить превышения временем задержки некоторого порогового значения. Для этого предлагается использовать подход, описанный в следующем разделе.

Уменьшение времени задержки CTD осуществляется различным образом для различных классов обслуживания. Так, для классов гарантированного сервиса (CBR, VBR) такое уменьшение может осуществляться за счет выбора альтернативных маршрутов передачи, которые являются в данный текущий момент времени более выгодными, чем те, которые используются. Для сервиса ABR возможно уменьшение интенсивности трафика, вплоть до минимального значения MCR. Следует отметить, что уменьшение интенсивности трафика возможно также для классов обслуживания с гарантированным качеством сервиса в случаях, когда

их коэффициент ускорения больше единицы, а также для VBR классов с некоторым снижением качества передачи мультимедийной информации, телефонии и т.п.

Вышеперечисленные действия могут дать желаемый результат только при достаточно эффективной схеме управления обратной связью. Часть этой схемы, касающаяся установки соединения. Далее описаны дополнения этого алгоритма работы цепи обратной связи в ключе формирования трафика и управления сетевыми параметрами, необходимыми для модифицированной схемы: ввести два пороговых значения *CTD*, как показано на рисунке 7.

При $CTD \geq CTD_{w1}$, текущий коммутатор генерирует и отправляет предыдущему коммутатору на маршруте ячейку RM с установленным флагом *W1* и текущим значением *CTD*. Следует отметить, что здесь подразумевается общее время задержки доставки, однако пороговые значения *CTD* (или *CDV*) достигается на определенном коммутаторе при его чрезмерной загрузке или невозможности передать данные определенного соединения из-за отказов сети. Поэтому следует предпринять попытку “разгрузить” этот коммутатор или выбрать альтернативный маршрут для его обхода.

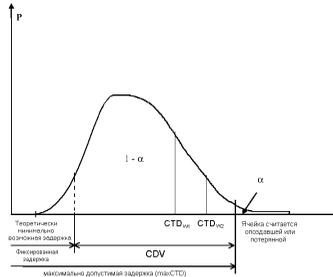


Рис. 7 – Распределение вероятности возникновению дополнительной задержки при введении двух пороговых значений *CTD*

При получении *W1*, коммутатор пытается определить альтернативный маршрут доставки, используя только свои связи. Альтернативный маршрут должен удовлетворять условию:

$$CTD_{new} \leq CTD + \Delta, \tag{5}$$

где Δ – параметр, учитывающий временные издержки на установку нового соединения, обновления содержимого маршрутных таблиц и фактор риска, необходимый для предотвращения новых перегрузок.

Если такой маршрут найден, то производится переустановка соединения и все сопутствующие ей действия, чтобы “перенести” перегруженное соединение на новый маршрут. Если маршрут не найден, то *W1* игнорируется.

При $CTD \geq CTD_{w2}$, текущий коммутатор генерирует и отправляет предыдущему коммутатору на маршруте ячейку RM с установленным флагом W2 и текущим значением CTD. W2 сигнализирует о перегрузке близкой к критической.

Если необходимый маршрут не может быть найден средствами данного коммутатора, то ячейка с W2 передается коммутатору предыдущему с точки зрения перегруженного маршрута, вплоть до источника. Источник при этом может реагировать на получение предупреждающих ячеек как изменением скорости передачи (ее понижением), так и выбором альтернативных маршрутов.

Вывод

Вышеперечисленные пункты алгоритма выполняются только для трафика с гарантированным качеством обслуживания. Для трафика “лучшей попытки” передается только ячейка W2, и передается она по цепочке коммутаторов к источнику с требованием понизить интенсивность передачи.

Источники трафика доступная битовая скорость (ABR) и неуказанная битовая скорость (UBR) могут также попытаться изменить маршрут, однако такое решение должно приниматься на уровне самих источников и поддерживаться сетью.

Для эффективной работы вышеописанного метода необходимо обеспечить “справедливую” дисциплину обслуживания, чтобы иметь некоторую гарантию того, что увеличение CTD для определенного направления связано с общей перегрузкой коммутатора, а не дискриминацией из-за присутствия более приоритетного трафика. Кроме того, она будет эффективно работать с передачей на повышенных скоростях и быстро реагировать на необходимость изменения скорости, чтобы предпринимать соответствующие меры.

Литература

1. Sfnj,K.; Ohta,S.; and Tokizawa,I. “Broad-band ATM Network Architecture Based on Virtual Paths”. IEEE Transactions on Communications, August -1990.
2. ATM Forum Technical Committee Traffic Management Specification Version 4.1 af-tm-0121.00.
3. Sato,K.; Ueda,H.; and Yoshikai, M. “The Role of Virtual Path Cross-connection”. IEEE LTS, August 1991.

Получено 13.11.2007