УДК 004.057.4, 004.042

О.В. РЫЖКОВА

Таврический национальный университет им. В.И. Вернадского, Украина

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ ИСПОЛЬЗОВАНИЯ АЛГОРИТМОВ ИЗМЕНЕНИЯ РАЗМЕРА ОКНА ПЕРЕГРУЗОК В СЕТЯХ CLOUD COMPUTING

Анализируется проблема сетевой перегрузки в сетях Cloud Computing и методы ее избегания путем изменения размера окна перегрузок CWND в TCP протоколе. Алгоритмы реализации этих методов рассматриваются как процедуры параметрической адаптации системы управления объектом CWND. Предлагаются варианты использования оптимальных с точки зрения функционирования сети алгоритмов в соответствии со спецификой технологии Cloud. Выбор алгоритма контроля перегрузок следует делать в соответствии такими особенностями сетей Cloud Computing, как мультисервисность, многопоточность, избыточность, с учетом больших объемов и типов передаваемого трафика.

Ключевые слова: *сетевая перегрузка, окно перегрузок, cloud computing.*

Введение

Проблема сетевой перегрузки [1] в высоконадежных, высокопроизводительных сетях Cloud Computing, предоставляющих услуги как сервис, на сегодняшний день особенно актуальна [2]. Предпосылок и причин сетевой перегрузки достаточно как на аппаратном уровне, так и на программном. На аппаратном уровне сетевая перегрузка напрямую связана с переполнением очередей буферов на интерфейсах устройств, ненадежностью среды передачи данных, нехваткой пропускной способности каналов передачи данных, несовершенством коммутационной матрицы сетевых устройств. На программном уровне эта проблема связана с мультиплексированием виртуальных каналов, потоков с гарантированной передачей и без, выделением пропускной способности для каждого соединения в зависимости от типа трафика, приложения, протокола, и т.д. Кроме того, мировая тенденция перехода с IPv4 на IPv6 также вносит определенный вклад в объемы передаваемой информации по IP-сетям, так как объем служебной информации в заголовках IP-пакетов существенно увеличивается.

Для борьбы с сетевой перегрузкой на аппаратном уровне используются известные методы [1]: увеличение объема буферной памяти на интерфейсах сетевых устройств, расширение пропускной способности каналов передачи данных, использование высоконадежных коммутаторов и маршрутизаторов с наилучшими показателями наработки на отказ, резервирование каналов связи и сетевых устройств. На программном уровне необходимо эффективно распределить полосу пропускания между потоками данных (это включает в себя служба контроля качества QoS), а также эффективно управлять объемом трафика, который передается от отправителя к получателю.

В наиболее популярных на сегодняшний день сетях TCP/IP для подавления перегрузки используется механизм динамического изменения размера окон перегрузки CWND (количество сегментов данных, который может быть передан от отправителя получателю без подтверждения), который управляется потерей пакетов. Если на пересылаемый пакет не пришло подтверждение о доставке *ACK* (квитанция), то пакет считается утерянным и требует повторной передачи. При этом размер окна перегрузки уменьшается. Чем меньше размер окна перегрузки (или так называемого «скользящего» окна), тем меньшими порциями будет передаваться информация по сети от источника получателю. В идеале размер «скользящего» окна должен стремиться к такому значению, при котором пакеты доставляются без потерь.

Существует несколько алгоритмов управления размерами окон перегрузки как в реализациях самого протокола TCP, так и отдельными механизмами. Цель данной работы – анализ существующих и выбор рационального алгоритма управления размером окна перегрузки в сетях Cloud Computing, который обеспечит надежное функционирование сети и минимальное количество потерь пакетов.

1. Учет особенностей Cloud Computing

Концепция облачных вычислений и расположения ресурсов в «облаке» предполагает надежную передачу данных из сети клиента в «облако» и в обратном направлении.

При выборе алгоритма контроля перегрузками сети следует опираться критерии, определяемые особенностями Cloud сетей [2].

1. Мультисервисность. Облачные технологии ориентированы на несколько основных типов сервисов: программное обеспечение как услуга (SaaS), платформа как услуга (PaaS), инфраструктура как услуга (IaaS), хранение данных как услуга (dSaaS).

2. Многопоточность. Так как «облако» предоставляет различные виды сервисов и услуг (хранение данных, потоковое видео, IP-телефония), по каналам передачи будут передаваться различные типы и классы трафика, с гарантированной доставкой (TCP), без гарантии (UDP), с различными приоритетами.

3. Избыточность. Резервирование каналов связи (резервные маршруты) а также самих сетевых устройств.

4. Большие объемы передаваемых данных и открытых сессий (соединений) между получателями и отправителями.

Эти и другие особенности следует учесть при выборе алгоритма управления размером окна перегрузок в качестве критериев, на основании которых делается выбор. Проанализируем варианты реализаций алгоритмов управления контролем перегрузок в сетях.

2. Адаптация в процессе управления объектом окна перегрузок

Различные реализации алгоритмов управления размером окна можно рассматривать с точки зрения адаптации как способы управления объектом CWND в обстановке неопределенности среды – среды передачи данных. Неопределенность среды связана не только с влиянием помех на состояние среды, но и со всплесками объемов трафика, передающегося по каналам передачи данных, которые появляются случайным образом. Так как речь идет об изменении параметров объекта управления – размера окна перегрузок, будем говорить о параметрической адаптации.

Согласно [3], пусть X – состояние взаимодействующей с объектом среды, в которой находится объект CWND, а Y – состояние объекта. Тогда объект можно представить как преобразователь F^0 состояния среды в состояние объекта: $Y = F^0(X)$, где F^0 – оператор входа X и выхода Y объекта.

С учетом фактора управления U, объект Y можно представить как $Y = F^0(X, U)$, где U – канал управления состоянием объекта Y.

Блок-схема системы управления объектом CWND в общем случае будет выглядеть таким образом [3], как показано на рис. 1

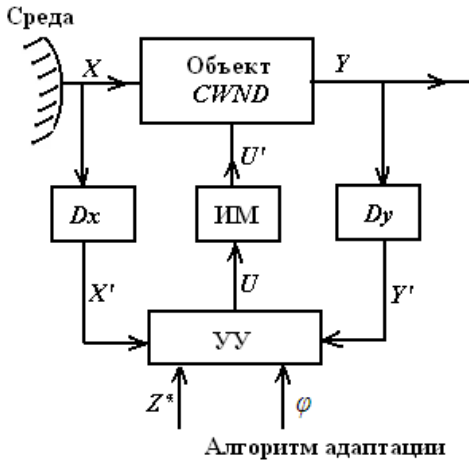

Рис. 1. Блок-схема системы управления объектом окна перегрузок CWND

На рис. 1 Dx, Dy – датчики, измеряющие состояние среды и объекта соответственно (таймеры высокого разрешения на отправителе, механизмы оценки пропускной способности канала, потока данных, загруженности буферов интерфейсов устройств). Результаты измерений X' = DX (X), Y' = DY (Y) поступают на управляющее устройство (УУ), которое вырабатывает команды управления U. Эти команды обрабатываются исполнительными механизмами (ИМ) с целью изменения состояния управляемого входа U' объекта. Для функционирования управляющего устройства ему нужно сообщить цель управления Z* и алгоритм φ– алгоритм управления размером окна перегрузки: U = φ (X', Y', Z*), следовательно $Y = F^0(X, \varphi (X', Y', Z^*))$.

Для того, чтобы явно задать цели Z*, их необходимо унифицировать. Они могут выражаться в виде требований к определенным критериям, заданным на состоянии Y объекта. Эти требования необходимо свести к одной из следующих форм:

- критерии-неравенства $H(U) \geq h$,
- критерии-равенства $G(U) = 0$,
- критерии минимальности $Q(U) \rightarrow \min$,

где H (U), G (U), Q (U) – набор критериев определения цели Z* объекта Y, изменяющегося с помощью адаптируемых параметров U; h – некоторое пороговое значение.

Цель адаптации заключается в решении задачи $Q(U) \rightarrow \min \rightarrow U^*$, при $H(U) \geq h$ и $G(U) = 0$.

В зависимости от реализации каждого конкретного алгоритма контроля перегрузок будет использован свой набор таких критериев. В общем случае критериями-неравенствами могут служить значения некоторых пороговых переменных, сигнализирующих о приближающейся перегрузке. Критерии-равенства можно представить в виде количества потерь пакетов равного нулю. Минимизируемыми критериями могут выступать время RTT (round trip time) – сумма времен

транспортировки сегмента от отправителя к получателю и времени движения отклика от получателя к отправителю, длина очереди (степень загрузки буфера) на интерфейсах устройств.

Восстановление работоспособности сети после перегрузки требует определенного времени, которое может оказаться критическим для функционирования ряда систем Cloud Computing, таких, как системы реального времени с высоким коэффициентом готовности. Поэтому, учитывая специфику Cloud, следует отметить то, что задачей адаптации также будет являться предотвращение перегрузки и переполнения очередей буферов устройств, а не только оперативное ее подавление. Также может возникнуть проблема синхронной реакции на перегрузку. Когда в результате потери пакетов все сетевые устройства идентифицируют перегрузку, они синхронно уменьшают размер окна перегрузки до минимума, обеспечивая тем самым недогруженность сети и неэффективное использования полосы пропускания, а затем так же синхронно его увеличивают вплоть до появления новой перегрузки.

3. Алгоритмы управления размером окна перегрузки CWND

В настоящее время предложено и опробовано несколько разновидностей алгоритмов управления окном перегрузки [4]. В рамках данной статьи анализируются наиболее известные из них.

TCP-Tahoe.

Данный алгоритм является одним из самых первых механизмов управления размером «скользящего» окна CWND в протоколе TCP. Размер «скользящего» окна перегрузки меняется циклически и проходит несколько этапов (фаз): фаза медленного старта, фаза исключения перегрузки, фаза повторной передачи и восстановления.

В фазе медленного старта размер CWND растет экспоненциально. Переход из фазы медленного старта в фазу исключения перегрузки обусловлен превышением текущего размера окна CWND величины порога ssthresh, который является своеобразным индикатором приближающейся перегрузки. В этой фазе рост CWND становится линейным. Рост CWND будет продолжаться до тех пор, пока не произойдет сетевая перегрузка, т.е. истечение времени RTO (таймаут) либо отсутствие подтверждения ACK для переданного получателю сегмента данных. Потерянный пакет и все, посланные после него, пакеты (вне зависимости оттого, подтверждено их получение или нет) пересылаются повторно. При большой вероятности потери это существенно понижает пропускную способность и увеличивает и без того высокую загрузку канала. Алгоритм адаптации представлен графически на рис. 2 [1].

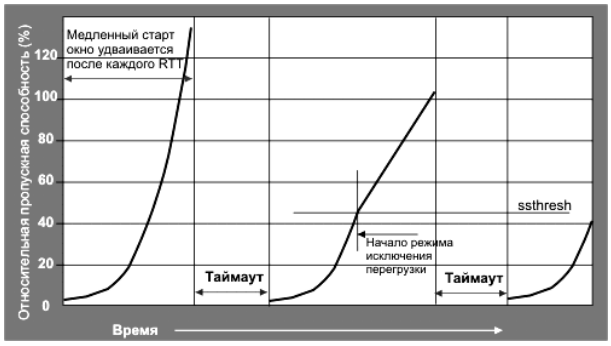

Рис. 2. Алгоритм TCP-Tahoe адаптации CWND при потере пакетов

В условиях реализации этого алгоритма критериями-неравенствами являются CWND <(≥) ssthresh и время $t \le RTO$ (таймаут). Цель алгоритма заключается в удержании значения CWND в области максимально возможных значений. По существу эта оптимизация осуществляется с помощью потери пакетов. Если потерь пакетов не происходит, значение CWND достигает искомого значения. Следовательно, критерий минимизации потерь пакетов является целью адаптации.

TCP-Reno.

Алгоритм TCP-Reno работает по аналогичному принципу, что и TCP-Tahoe, разница лишь в линейном возрастании CWND в фазе медленного старта и других некоторых модификациях.

Алгоритм адаптации размера окна в момент времени $t + t_A$:

$$CWND(t + t_A) =$$

$$= \begin{cases} CWND(t) + 1, \text{если } CWND(t) < ssth; \\ CWND(t) + \dfrac{1}{CWND(t)}, \text{если } CWND(t) \ge ssth. \end{cases}$$

Критерии остаются аналогичными. Недостаток этого алгоритма, согласно результатам практических исследований в [1], показывают, что происходит потеря как минимум двух пакетов в каждом эпизоде перегрузки. Некоторое улучшение такой ситуации дает модель алгоритма NewReno. Преимущество заключается в использовании механизмов Fast Retransmit & Fast Recovery (быстрая повторная пересылка и быстрое восстановление). Алгоритм NewReno использует дополнительную переменную recover (восстановление), которая служит еще одним порогом критерия-неравенства. В случае, когда доступна опция выборочного подтверждения (SACK), отправитель знает, какие пакеты следует переслать повторно на фазе быстрого восстановления (Fast Recovery).

TCP Vegas.

TCP-Vegas контролирует размер окна путем мониторинга отправителем времени 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. Поэтому в данном алгоритме параметр RTT можно рассматривать как критерий минимизации.

TCP-Westwood.

Алгоритм TCP-Westwood позволяет достичь большей эффективности использования пропускной способности канала. Он основан на оценке потока данных (RE - Rate Estimation) и текущего значения полосы пропускания (BE - Bandwidth Estimation). вычисление CWND и порога ssthresh производится на основе этих оценок.

Все рассмотренные на этом этапе алгоритмы являются модификациями самого протокола TCP. Отправитель настраивает соответствующую модификацию у себя на устройстве перед тем, как начать передачу данных. К сожалению, ни одна из них практически не учитывает особенности, присущие сетям Cloud, о которых говорилось в начале статьи: мультисервисность, многопоточность, избыточность, большие объемы передаваемого трафика. Алгоритмы модификаций TCP контролируют ширину «скользящего» окна на основании не менее важных критериев, но без учета типа трафика, протокола передачи данных, типа приложения, наличия альтернативных маршрутов, т.е. функций QoS. Поэтому эти алгоритмы будут эффективны в сочетании со службами балансировки нагрузки, механизмами управления очередями на интерфейсах устройств, QoS.

Более того, у всех рассмотренных алгоритмов есть существенный недостаток: сеть узнает о перегрузке уже после того, как она произошла, но не предотвращает ее. С этой точки зрения очень эффективен алгоритм случайного раннего обнаружения RED и его модификация WRED на интерфейсах устройств, учитывающий тип передаваемого трафика в поле ToS (Type of Service) заголовка IP-пакета. Алгоритм предполагает стохастическую выборку пакетов из очереди на интерфейсах сетевых устройств и их отбрасывание еще до того, как заполнится очередь. Это заставляет TCP плавно уменьшать размер «скользящего» окна перегрузки и предотвращает волны синхронизаций. Установка пакета в очередь или его отбрасывание осуществляется на уровне IP-протокола. Алгоритм RED состоит из 2 основных частей: определения среднего значения длины очереди и принятии решения о том, будет ли отброшен пакет или нет. Решение принимается на основании следующих критериев: минимальный порог сбрасывания пакетов (MIN TH) – среднее значение длины очереди пакетов, выше которого пакеты выборочно сбрасываются; максимальный порог сбрасывания пакетов (MAX TH); превышение – максимальное количество пакетов, которые могут быть приняты в очередь сверх максимального порога.

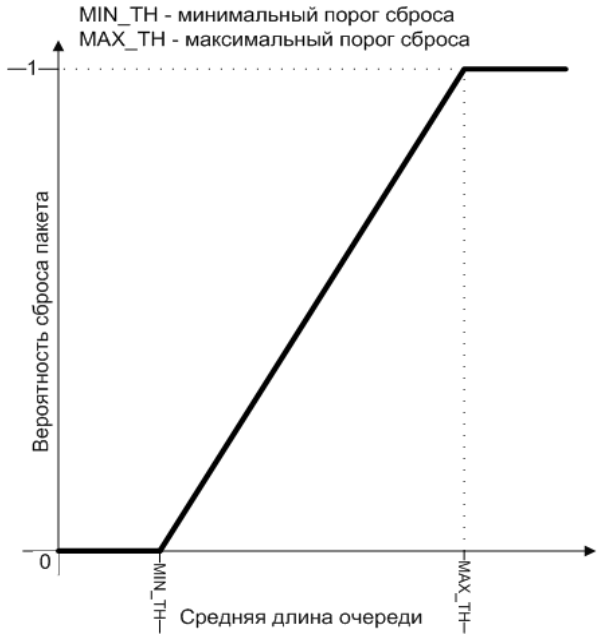

Рис. 3. Адаптация алгоритма RED
в зависимости от средней длины очереди

В случае использования этого алгоритма постановка задачи адаптации заключается в минимизации критерия средней длины очереди пакетов на интерфейсе.

Разновидность алгоритма WRED (Weighted RED) в качестве еще одного критерия учитывает тип трафика и его приоритет. Поэтому при использовании этого алгоритма особенности многопоточного, многосервисного трафика в сетях Cloud Computing будут учтены. Алгоритм RED/WRED должен поддерживаться сетевыми устройствами. Поддержка RED/WRED имеется на маршрутизаторах Cisco серии 7000, 12000 и коммутаторах 3 уровня, таких как Catalyst 3550.

Более эффективная реализация этого механизма в сочетании с алгоритмом ECN (Explicit Congestion Notification). ECN также использует поле ToS в заголовке IP-пакета, устанавливая в нем битфлаг CE (Congestion Experienced) в том случае, если ожидается перегрузка. Отправитель должен среагировать соответствующим образом на эту ситуацию, уменшив размер окна перегрузки. Система может адекватно реагировать на состояние среды передачи данных не отбрасывая пакеты, с необходимостью их дальнейшей повторной передачи, а сохраняя их, повышая надежность и готовность сети, что является актуальным для Cloud Computing. При этом необходима настройка ECN на обеих сторонах (отправителя и получателя) и поддержка на транспортном уровне. Маршрутизаторы Cisco поддерживают ме-

ханизм ECN, если настроен WRED, начиная с версии IOS 12.2 и выше.

Еще одним решением эффективного мультиплексирования виртуальных каналов, смеси потоков данных с гарантией и без может служить менеджер перегрузок (Congestion Manager). Это отдельный модуль, взаимодействующий с сетевыми приложениями. Перед тем, как послать данные, модуль на программном уровне анализирует всю доступную информацию о характеристиках каналов связи, количестве переданных и принятых байт информации, результатах измерения RTT, после чего подбирает рекомендуемое значение размера «скользящего» окна для надежной передачи. На рис.4 показано взаимодействие компонентов менеджера перегрузки (МП) между собой и потоками данных протоколов различных уровней. Контроллер перегрузки Congestion Controller регулирует скорость передачи и объем потоков данных на основании оценки загруженности сети во время предыдущих передач, которую получает через интерфейс API самих приложений. Планировщик Sheduler распределяет доступную полосу пропускания между различными потоками и уведомляет приложение, когда разрешена передача.

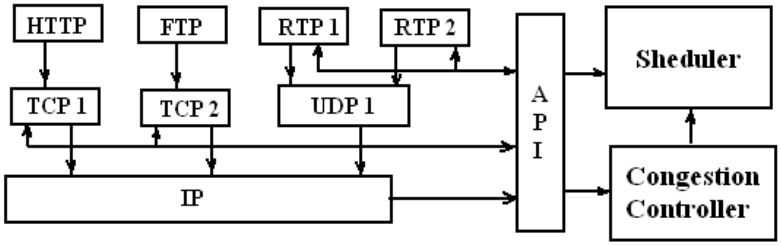

Рис. 4. Взаимодействие компонентов МП

Несмотря на все рассмотренные преимущества, которые выгодно использовать в сетях Cloud, опираясь на [5], можно выделить существенный недостаток: МП наиболее эффективен при использовании в сетях с отсутствием резервирования, исключающих наличие резервных маршрутов с различными характеристиками для потоков данных между одной и той же парой устройств, что является неприемлемым с точки зрения критерия избыточности в сетях облачных технологий.

Выводы

С учетом особенностей построения и функций сетей Cloud Computing, можно дать следующие рекомендации при выборе алгоритмов контроля окон перегрузки:

1. Высокую эффективность работы сети Cloud можно получить при использовании алгоритмов контроля перегрузок, цель которых – не только своевременно адекватно среагировать на перегрузку, уменьшив соответствующим образом ширину окна, но и максимально предотвратить ее появление. Вариантом такого алгоритма является механизм случайного раннего обнаружения RED/WRED, который предполагает стохастическую выборку пакетов из очереди на интерфейсах сетевых устройств и их отбрасывание еще до того, как очередь заполнится. Этот метод удобен при использовании в сочетании с одной из разновидностей протокола TCP, реализующей соответствующий метод изменения ширины «скользящего окна», например с TCP-Westwood.

2. Повышение показателей надежности сети Cloud подразумевает минимизацию потерь пакетов. Это позволяет сделать алгоритм ECN, настроенный совместно с механизмами RED/WRED на интерфейсах маршрутизаторов.

3. Выбор алгоритма контроля перегрузок следует делать в соответствии такими особенностями сетей Cloud Computing, как мультисервисность, многопоточность, избыточность, с учетом больших объемов и типов передаваемого трафика. С этой точки зрения эффективным будет использование механизма WRED совместно с ECN, который учитывает тип и приоритет различных видов трафика, а также менеджера перегрузки (Congestion Manager) на участках, где не предусмотрено резервирование каналов связи с различными характеристиками.

Литература

1. Семенов, Ю.А. Модели реализации протокола TCP и его перспективы [Електронний ресурс] / Ю.А. Семенов. – Режим доступа: http://book.itep.ru/4/44/tcp.htm. – 18.02.2012.

2. Jones, M. Tim. Cloud computing with Linux [Text] / M. Tim Jones. – February, 2009. – 322 p.

3. Растригин, Л.А. Адаптация сложных систем [Текст] / Л.А. Растригин. – Рига.: Зинатне, 1981. – 375 с.

4. Allman, M. TCP Congestion Control [RFC 2581] [Text] / M. Allman, V. Paxson. – April, 1999.

5. Balakrishnan, H. The Congestion Manager [RFC 3124] [Text] / H. Balakrishnan, S. Seshan. – June, 2001. – 214 p.

Поступила в редакцию 18.02.2012

Рецензент: д-р техн. наук, проф., заведующий кафедрой компьютерных систем и сетей В.С. Харченко**,** Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина.

ПОРІВНЯЛЬНИЙ АНАЛІЗ ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ АЛГОРИТМІВ ЗМІНИ РОЗМІРУ ВІКНА ПЕРЕВАНТАЖЕНЬ У МЕРЕЖАХ CLOUD COMPUTING

О.В. Рижкова

Аналізується проблема мережевого перевантаження в мережах Cloud Computing і методи її уникнення шляхом зміни розміру вікна перевантажень CWND у TCP протоколі. Алгоритми реалізації цих методів розглядаються як процедури параметричної адаптації системи керування об'єктом CWND. Пропонуються варіанти використання оптимальних з точки зору функціонування мережі алгоритмів відповідно до специфіки технології Cloud. Вибір алгоритму контролю перевантажень слід робити відповідно такими особливостями мереж Cloud Computing, як мультисервісність, багатопотоковість, надмірність, з урахуванням великих обсягів і типів переданого трафіку.

Ключові слова: мережеве перевантаження, вікно перевантажень, cloud computing.

COMPARATIVE EFFECTIVENESS OF ALGORITHMS RESIZING CONGESTION WINDOW IN CLOUD COMPUTING NETWORKS

O.V. Ryzhkova

The problem of network congestion in Cloud Computing networks and the methods of its avoidance by changing the size of congestion window CWND in TCP protocol are analyzed. Algorithms for the implementation of these methods are described as procedures of parametric adaptation of the system managing object CWND. Different solutions are offered for using the optimal algorithms from the point of view of the network operation, in accordance with the specifics of Cloud technology. The choice of congestion control algorithm should be done according to such characteristics of networks Cloud Computing, as a multiservice, multi-threading, redundancy, considering the large volumes and types of the traffic.

Key words: network congestion, congestion window, cloud computing.

Рыжкова Ольга Владимировна – ассистент кафедры компьютерной инженерии и моделирования Таврического национального университета им. В.И. Вернадского «ТНУ», Симферополь, Украина.