УДК 621.394.6;654.1

А.С. Крутолапов[1], А.К. Абулев[2], С.Ю. Ошкин[3]
([1]Санкт-Петербургский университет Государственной противопожарной службы МЧС России, [2]ГУ МЧС России по г. Москве, [3]Академия ГПС МЧС России;
e-mail: krut75@mail.ru)

ОБ УПРАВЛЕНИИ ПОТОКАМИ В СЕТЯХ ПЕРЕДАЧИ ДАННЫХ ГПС МЧС РОССИИ

Проведён анализ проблем обслуживания пользователей сети передачи данных ГПС МЧС России, предложены архитектура распределённой информационной среды и структура её программного обеспечения.

Ключевые слова: сеть, передача данных, архитектура, программное обеспечение.

A.S. Krutolapov, A.K. Abulev, S.Y. Oshkin

ABOUT MANAGEMENT OF FLOWS IN THE NETWORKS OF DATA TRANSFER OF THE STATE FIRE SERVICE OF EMERCOM OF RUSSIA

The analysis of problems of client service from the data transfer network of the State Fire Service of EMERCOM of Russia, proposed architecture of distributed information environment and structure of its software.

Key words: network, data transfer, architecture, software.

Иерархическую структуру классов качества обслуживания пользователей в сети передачи данных можно представить в виде дерева. В листовых узлах дерева содержатся реальные очереди конечных классов. Внутренние узлы дерева обеспечивают реализацию правил обслуживания на группах листовых и внутренних классов.

Для простоты описания примем, что листовые узлы дерева располагаются на одном уровне h (высота дерева). Каждому классу обслуживания K может быть сопоставлена абстрактная очередь Q (для листа она является реальной) и весовой коэффициент φ, определяющий неизменную составляющую приоритета класса.

Очередь листового класса можно представить как множество элементов, на котором задано отношение полного порядка. Тогда под абстрактной очередью внутреннего класса понимается множество элементов, на котором задано отношение полного порядка, являющееся объединением попарно непересекающихся подмножеств дочерних классов, которые также являются абстрактными очередями. Рассмотрим фрагмент структуры дерева классов, представленный на рис. 1.

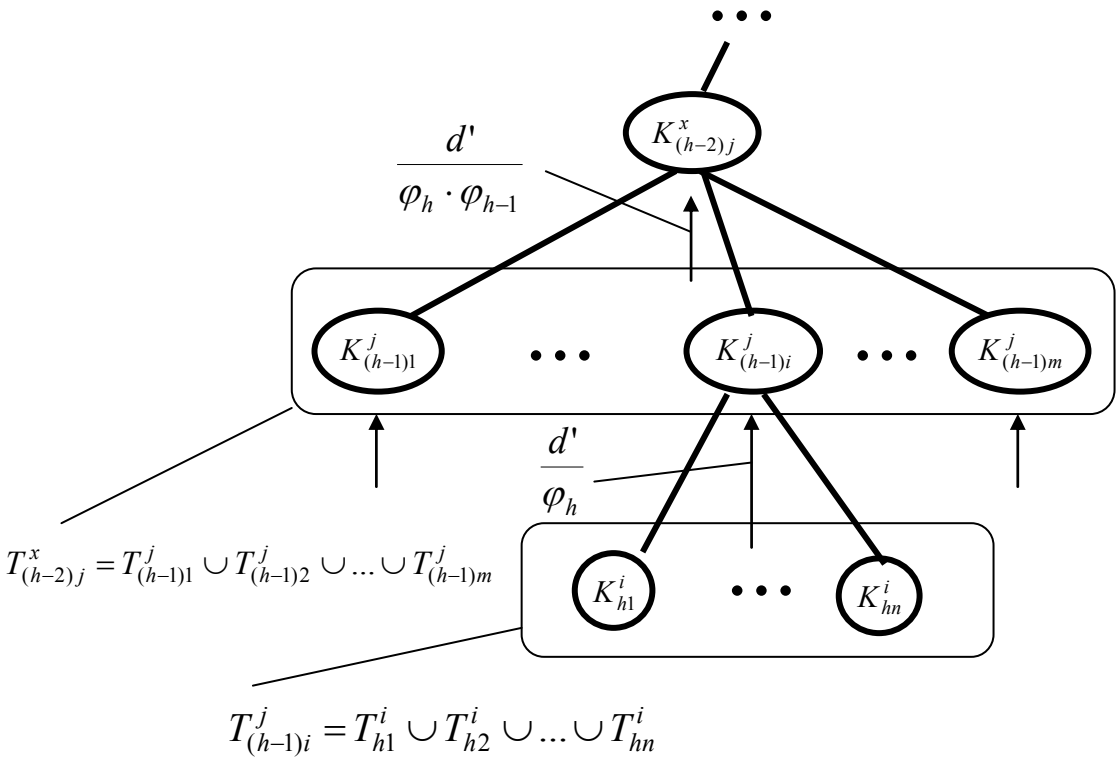

Рис. 1. Фрагмент иерархической структуры классов обслуживания

На уровне листовых узлов h порядок следования элементов в множестве T^i_{hl} класса K^i_{hl}, являющегося l-ым потомком класса с индексом i, определяется их текущими задержками $d_h = d'$. На предпоследнем уровне множество элементов абстрактной очереди класса $K^j_{(h-1)i}$ является объединением подмножеств своих потомков $T^j_{(h-1)i} = T^i_{h1} \cup T^i_{h2} \cup \ldots \cup T^i_{hn}$. Причем порядок следования элементов в соответствии со сформулированной одноуровневой моделью определяется взвешенными значениями текущих задержек $d_{h-1} = \frac{d'}{\varphi_h} = \frac{d_h}{\varphi_h}$, где φ_h – весовой коэффициент класса на уровне h. На уровне $h - 2$ имеем множество $T^x_{(h-2)j} = T^j_{(h-1)1} \cup T^j_{(h-1)2} \cup \ldots \cup T^j_{(h-1)m}$ с порядком по величине $d_{h-2} = \frac{d_{h-1}}{\varphi_{h-1}}$.

Имеет место рекуррентная зависимость, формально описывающая величину, определяющую порядок выбора заявки на обслуживание на произвольном уровне k дерева классов:

$$\begin{cases} d_k = \dfrac{d_{k+1}}{\varphi_{k+1}}, \ k = 1 \ldots h-1; \\ d_k = d', \quad k = h. \end{cases} \qquad (1)$$

Реальный выбор заявки на обслуживание происходит из листового класса. На формальном уровне на основе рекуррентной зависимости можно представить выбор на уровне 1. Приведем рекуррентную зависимость (1) к замкнутому виду:

$$d_1 = \frac{d'}{\prod_{k=2}^{h} \varphi_k}. \qquad (2)$$

Примем, что в любой момент времени отношение текущих задержек двух последовательно выбранных заявок классов i и j стремится к отношению их весовых коэффициентов:

$$\frac{d'_i}{d'_j} \to \frac{\varphi_i}{\varphi_j}, \qquad (3)$$

С учётом полных весовых коэффициентов это условие запишется в виде:

$$\frac{d'_i}{d'_j} \to \frac{\prod_{k=2}^{h_i} \varphi_{ki}}{\prod_{k=2}^{h_j} \varphi_{kj}} \qquad (4)$$

и сформулируем окончательные условие справедливости и правила выбора заявок в иерархической модели.

Определение 1. Распределение ресурса обслуживающего устройства является справедливым, если:

1. В любой момент времени отношение текущих задержек двух последовательно выбранных заявок листовых классов i и j стремится к отношению их полных весовых коэффициентов:

$$\frac{d'_i}{d'_j} \to \frac{\prod_{k=2}^{h_i} \varphi_{ki}}{\prod_{k=2}^{h_j} \varphi_{kj}}.$$

2. Класс обслуживания не "штрафуется" за использование ресурса обслуживающего устройства в предшествующие периоды.

Определение 2. Выбор следующей заявки на обслуживание осуществляется в соответствии со следующими правилами:

1. Право занять обслуживающее устройство предоставляется заявке из очереди i-го листового класса обслуживания, где

$$i = \max_{d'_j / \prod_{k=2}^{h_j} \varphi_{kj}} \{ j \mid j \in A \}.$$

2. Если таких классов окажется более одного, выбор определяется случайным образом.

3. Порядок следования заявок внутри каждой очереди определяется алгоритмом частично кругового циклического обслуживания с выбором позиции.

Методика управления потоком заявок

Любая система обеспечения качества обслуживания управляет потоком с целью предупреждения перегрузок. Под перегрузкой в данном случае понимается перегрузка системы, то есть переполнение её очередей.

Основными целями управления потоком являются:

- управление размерами очередей диспетчера с целью строгого ограничения максимального среднего размера;

- обеспечение справедливого управления потоками, то есть выделение более и менее приоритетных классов с точки зрения выделения им места в буфере диспетчера.

Для регулирования длины очередей применяются различные алгоритмы, реализующие политику отбрасывания пакетов (packet drop policy). Традиционный алгоритм обслуживания очередей FCFS использует достаточно простую политику "отбрасывания хвоста" (tail drop), в соответствии с которой любая попытка постановки пакета в полную очередь заканчивается его отбрасыванием [1].

С точки зрения работы транспортного протокола TCP, политика отбрасывания хвоста имеет ряд недостатков. С целью поддержки механизма предотвращения заторов в сети источники TCP-соединения используют так называемое окно перегрузки (congestion window) и алгоритм медленного старта [2, 3]. Сочетание механизмов "отбрасывания хвоста" и медленного старта приводит к одновременным периодическим резким снижениям интенсивности трафика и перегрузкам во многих TCP-соединениях. Такие подобные волне изменения размера очереди получили название эффекта глобальной синхронизации (global synchronization) [4].

В нашем случае механизм "отбрасывания хвоста" также не является решением задачи, так как его использование не позволит достичь последней цели реализации механизма управления потоком.

Появление эффекта глобальной синхронизации привело к необходимости создания механизма, позволяющего превентивно управлять очередью с целью сигнализации о перегрузке сети до фактического переполнения очереди. ***Методика произвольного раннего обнаружения (Random Early Detection – RED)*** предложена Салли Флойдом (Sally Floyd) и Ваном Якобсоном (Van Jacobson) [5]. На данный момент разработано несколько модификаций этой методики [6, 7].

В реализации, предложенной в работе [5], методика раннего произвольного обнаружения имеет основной целью обеспечение предупреждения перегрузок. Дополнительные цели включают:

- минимизацию дрожания задержки пакетов путем контроля за средним размером очереди;

- предотвращение эффекта глобальной синхронизации TCP-трафика;

- обеспечение непредвзятого обслуживания трафика, характеризующегося кратковременными всплесками;

- строгое ограничение максимального среднего размера очереди.

Цели работы алгоритма RED отчасти совпадают со сформулированными нами целями алгоритма управления потоком в разрабатываемой системе. Одна из модификаций – взвешенный алгоритм произвольного раннего обнаружения (Weighted Random Early Detection – WRED), предоставляет различные уровни обслуживания пакетов в зависимости от вероятности их отбрасывания, то есть поддерживает возможность настройки параметров с целью справедливого обслуживания на основании приоритетов.

Оптимальным можно считать подход к построению требуемого алгоритма путем модификации алгоритма произвольного раннего обнаружения RED.

Методика RED использует превентивный подход к предотвращению перегрузки сети. Вместо ожидания фактического переполнения очереди, RED начинает отбрасывать пакеты с ненулевой вероятностью, когда средний размер очереди превысит определенное минимальное пороговое значение (отбрасывание пакета служит сигналом TCP-источнику о необходимости уменьшить интенсивность передаваемого трафика путем перезапуска алгоритма медленного старта). Вероятностный подход к отбрасыванию пакетов позволит быть уверенными в том, что будут отброшены пакеты лишь нескольких произвольно выбранных потоков, тем самым помогая избежать эффекта глобальной синхронизации.

Если средний размер очереди будет продолжать увеличиваться, это приведет к линейному росту вероятности отбрасывания. Вероятность отбрасывания пакетов растет прямо пропорционально увеличению среднего размера очереди от минимального до максимального порогового значения. При достижении максимального порогового значения вероятность отбрасывания пакетов достигает 100 %.

В нашем случае цели и решаемые задачи отличаются от управления потоком на 3-4 уровнях эталонной модели МОС [8, 9].

Во-первых, нет априорной информации о распределении заявок по классам и, следовательно, невозможно обоснованно распределить емкость буфера диспетчера очередей, а распределение поровну не отвечает принципам справедливого обслуживания. Отсюда вытекает необходимость использования общего буфера. Очереди большего размера будут образовываться у классов высокой интенсивности. Однако такая "несправедливость" по отношению к классам, проявляющим меньшую активность, в данном случае является главным признаком справедливого обслуживания, так как отвечает потребностям пользователей.

Во-вторых, теряет свой смысл раннее произвольное обнаружение, в старом смысле, так как источник запроса (в нашем случае пользователь) не может повлиять на общую интенсивность поступления запросов в данном классе. Однако вероятностный подход может быть использован с целью реализации концепции справедливого обслуживания на основе дифференциации качества предоставляемых услуг.

В-третьих, основной целью использования в алгоритмах RED вместо реального размера усредненного размера очереди является непредвзятое отношение к трафику, характеризующемуся кратковременными всплесками. Фактически используется фильтр низких частот, чтобы исключить влияние кратковременных всплесков на результирующую величину. В нашем случае поток заявок формируется большим количеством независимо работающих пользователей. Такие потоки заявок имеют свойство компенсации активности одних пользователей за счет пассивности других. Следовательно, в нашем случае можно использовать реальное значение размера очереди.

Предлагаемая методика управления потоком

При поступлении очередной заявки на обслуживание текущий суммарный размер очередей *sum_n* сравнивается с двумя пороговыми значениями (thresholds): минимальным *min_th* и максимальным *max_th*. Если суммарный размер очередей не превышает минимального порогового значения, заявка передается для постановки в очередь соответствующего класса обслуживания. Если суммарный размер очередей достиг максимального порогового значения, то заявка отбрасывается с передачей соответствующего сообщения источнику.

Если суммарный размер очередей находится между минимальным и максимальным пороговыми значениями, то поступившая заявка отбрасывается с ненулевой вероятностью *p_a*. Вероятность является функцией суммарного размера очередей, а также зависит от суммарного весового коэффициента листового класса, к которому принадлежит поступившая заявка.

Литература

1. ***Braden B., Clark D., Crowcroft J.*** Recommendations on Queue Management and Congestion Avoidance in the Internet: Tech. Rep. IEEE RFC 2309, 1998. 16 p. http://xml.resource.org/public/rfc/html/rfc2309.html.

2. ***Wang Z., Crowcroft J.*** A New Congestion Control Scheme: Slow Start and Search (Tri-S) // Computer Communication Review, 1991. № 1. V 21. P. 32-43.

3. ***Stevens W.*** TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms: Tech. Rep. IETF RFC, 1997. http://rfc2001.x42.com.

4. ***Floyd S., Jacobson V.*** The Synchronization of Periodic Routing Messages // In Proceedings of ACM SIGCOMM'93, 1993. P. 33-44.

5. ***Floyd S., Jacobson V.*** Random Early Detection gateways for Congestion Avoidance // IEEE/ACM Transactions on Networking, 1993. № 4. V 1. P. 397-413.

6. ***Floyd S., Gummadi R., Shenker S.*** Adaptive RED: An Algorithm for Increasing the Robustness of RED's active queue management: Tech. Rep. ICSI, 2001. http://www.icir.org/floyd/papers/adaptiveRed.pdf.

7. ***Bodin U., Schelen O., Pink S.*** Load-tolerant Differentiation with Active Queue Management // SIGCOMM Computer Communication, 2000. № 3. V 30. P. 4-16.

8. ***Бертсекас Д., Галлагер Р.*** Сети передачи данных. М.: Мир, 1989. 554 с.

9. ***Блэк Ю.*** Сети ЭВМ: Протоколы, стандарты, интерфейсы. М.: Мир, 1990. 506 с.

Статья опубликована 27 февраля 2013 г.