

Имитационное моделирование и оптимизация сетевого трафика

А. В. Горячев¹, Н. Е. Новакова²

Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)

¹avgoryachev@gmail.com, ²nenovakova@gmail.com

Аннотация. Рассмотрены проблемы управления трафиком с целью обеспечения качества сетевого обслуживания. Предложены динамические модели фильтра на основе алгоритма маркерной корзины и мультиплексора, поддерживающего контроль качества обслуживания. Рассмотрена задача выбора оптимальной стратегии управления параметрами фильтров трафика, работающих по алгоритму маркерной корзины.

Ключевые слова: сетевой трафик; сетевые ресурсы; пропускная способность сети; качество сетевого обслуживания; мультиплексор; моделирование; алгоритм маркерной корзины; оптимизация

I. ВВЕДЕНИЕ

Широко распространенные в последнее время сетевые приложения реального времени – такие, как программы для IP-телефонии и видеоконференций – предъявляют еще более жесткие требования к качеству сетевого обслуживания компьютерных сетей [1]. Особого контроля требуют следующие параметры сети: пропускная способность сети, потери пакетов, задержка доставки пакетов, варьирование времени задержки доставки («джиттер»), ошибки (неверное направление, повреждение пакетов). Таким образом, управлять сетевыми ресурсами необходимо так, чтобы гарантировать время реакции, пропускную способность и другие параметры сети. Совокупность технологий, призванных добиться этой цели, получила название Quality of Service (далее QoS) – «качество обслуживания». Используя механизмы управления трафиком, механизмы QoS назначают разный приоритет разным приложениям, пользователям или потокам данных так, чтобы добиться приемлемого уровня производительности для каждого потока данных.

В настоящее время QoS реализуется рядом протоколов [2]. Все эти технологии так или иначе задействуют механизм, называемый «Token Bucket», или маркерная корзина. Token Bucket – это, по сути, фильтр трафика. Концептуально он представляет собой «корзину» с «маркерами» («токенами»). Управление проходящим через маркерную корзину трафиком осуществляется на основании количества токенов в корзине.

II. МОДЕЛЬ QoS МУЛЬТИПЛЕКСОРА

Имитационная модель QoS мультиплексора на основе алгоритма маркерной корзины описана в [3]. Эта модель может использоваться для тестирования эффективности разрабатываемых алгоритмов путем проведения имитационных экспериментов.

Входящий трафик определяется вектором $\{T(t_k), k = 0, 1, 2, \dots\}$. По сути, это случайный процесс, наблюдаемый в моменты времени t_k . Результатом наблюдения является размер передаваемого пакета или фрейма $T(t_k)$. Интервалы времени $[t_{k-1}, t_k)$ можно принять равными и достаточно короткими для того, чтобы максимум один пакет или фрейм мог поступить на вход.

Введем обозначения для характеристик маркерной корзины: s – размер корзины; $r(t_k)$ – скорость пополнения токенами (число токенов за отрезок времени $[t_{k-1}, t_k)$); $T(t_k)$ – поступающий на вход трафик. В начале каждого очередного временного отрезка времени $[t_{k-1}, t_k)$ в корзину поступает число токенов $r(t_k)$. Результирующее число токенов в корзине ограничено ее размером s . Далее обрабатывается поступающий на вход трафик $T(t_{k-1})$. Таким образом, к моменту времени t_k весь входящий трафик обработан и фильтр переходит к следующему состоянию.

Состояние корзины в момент времени t определяется количеством токенов в ней. Обозначим его $a(t_k)$. В каждый промежуток времени $[t_{k-1}, t_k)$ этот параметр увеличивается на $g(t_k) = \min\{r(t_k), s - a(t_{k-1})\}$.

Корзина либо пропускает через себя весь поступающий на вход трафик (если содержит достаточное количество токенов), либо полностью его отбрасывает. Таким образом, объем трафика, проходящего через корзину к моменту t_k , будет равен:

$$p(t_k) = T(t_k)I(T \leq a(t_{k-1}) + g(t_k)),$$

где $I(x)$ – индикаторная функция:

$$I(x) = \begin{cases} 1, & \text{если } x \text{ истинно,} \\ 0, & \text{если } x \text{ ложно.} \end{cases}$$

Объем отброшенного на k -м интервале времени трафика равен, разности между поступившим на вход и пропущенным трафиками:

$$l(t_k) = T(t_k) - p(t_k).$$

Изменение числа токенов в маркерной корзине во времени определяется разностным уравнением

$$a(t_k) = a(t_{k-1}) + g(t_k) - p(t_k). \quad (1)$$

Первое слагаемое в правой части (1) определяет число токенов в предыдущий момент времени t_{k-1} , второе – число токенов, полученных в данный промежуток времени $[t_{k-1}, t_k)$, а последнее – число токенов, потраченных к моменту t_k .

III. ИМИТАЦИЯ РАБОТЫ УЧАСТКА СЕТИ

Для имитации работы участка сети построим модель с n источниками трафика, обслуживаемыми n соответствующими маркерными корзинами. Все они подключены к одному мультиплексору трафика, соединенному с внешней сетью.

Другая величина, определяющая эффективность работы сети, – это ее утилизация, т. е. степень использования ее ресурсов. Она определяется отношением реально пропускаемого сетью трафика в единицу времени и максимально возможного значения пропускной способности сети:

$$U = \frac{P}{R} = \frac{\sum_{i=1}^n \sum_{k=0}^K T_i(t_k) - \sum_{k=0}^K L(t_k) - \sum_{i=1}^n \sum_{k=0}^K l_i(t_k)}{R(t_K - t_0)}.$$

Все они подключены к одному мультиплексору трафика, соединенному с внешней сетью. Мультиплексор характеризуется, во-первых, максимальной длиной очереди (размером буфера) S , а во-вторых, скоростью R пересылки данных во внешнюю сеть. Обозначим длину очереди в момент t_k как $O(t_k)$. Значение длины очереди в начале очередного интервала времени $[t_{k-1}, t_k)$ равно разности длины очереди в момент t_{k-1} и части очереди, которая освобождается на этом интервале (т. е. скорости пересылки во внешнюю сеть):

$$O(t_k) = \max\{0, Q(t_{k-1}) - R\}.$$

Трафик, поступающий от всех маркерных корзин на вход мультиплексора на интервале времени $[t_{k-1}, t_k)$, равен суммарному трафику, пропущенному всеми маркерными корзинами, т. е.

$$I(t_k) = \sum_{n=1}^N p(t_k),$$

а объем трафика, который мультиплексор в состоянии обработать, ограничен размером буфера очереди:

$$A(t_k) = \min\{I(t_k), S - O(t_k)\}.$$

Размер потеранных на мультиплексоре пакетов можно найти, вычислив разность трафика, поступающего от всех

маркерных корзин, и трафика, добавленного мультиплексором в свою очередь:

$$L(t_k) = I(t_k) - A(t_k).$$

Состояние мультиплексора определяется длиной его очереди, изменение которой во времени можно выразить разностным уравнением (уравнение является разностным, так как $O(t_k)$ содержит $Q(t_{k-1})$):

$$Q(t_k) = O(t_k) + A(t_k).$$

Средняя пропускная способность сети равна объему трафика, пропускаемого сетью в единицу времени, т. е.

$$P = \frac{\sum_{i=1}^n \sum_{k=0}^K T_i(t_k) - \sum_{k=0}^K L(t_k) - \sum_{i=1}^n \sum_{k=0}^K l_i(t_k)}{t_K - t_0}.$$

Другая величина, определяющая эффективность работы сети, – это ее утилизация, т. е. степень использования ее ресурсов. Она определяется отношением реально пропускаемого сетью трафика в единицу времени и максимально возможного значения пропускной способности сети:

$$U = \frac{P}{R} = \frac{\sum_{i=1}^n \sum_{k=0}^K T_i(t_k) - \sum_{k=0}^K L(t_k) - \sum_{i=1}^n \sum_{k=0}^K l_i(t_k)}{R(t_K - t_0)}.$$

Однако главным показателем качества сетевого обслуживания являются потери пакетов. Как было показано в [2], потери пакетов могут возникать как на маркерных корзинах, так и на мультиплексоре. Кроме того, существуют задержки доставки, возникающие из-за ожидания пакетов передачи в очереди мультиплексора.

IV. ЦЕЛЕВАЯ ФУНКЦИЯ ДЛЯ ОПТИМИЗАЦИИ ТРАФИКА

Сформируем целевую функцию, учитывающую эти факторы:

$$Y(r) = \sum_{k=0}^K \left[\alpha(t_k) L(t_k) + \sum_{i=1}^n \beta_i(t_k) l_i(t_k) + \gamma(t_k) Q(t_k) \right]$$

Первое слагаемое в правой части выражения определяет взвешенное значение суммарных потерь трафика на мультиплексоре, второе – взвешенное значение суммарных потерь на всех маркерных корзинах, а последнее – взвешенное значение сумм длин очередей мультиплексора на всем отрезке времени $[t_0; t_K]$.

Параметры $\alpha(t_k)$, $\beta_i(t_k)$ ($i = 1, 2, \dots, n$) и $\gamma(t_k)$ – это веса соответствующих слагаемых.

Целевая функция зависит от управляющего вектора $\mathbf{r} = (r_1, r_2, \dots, r_n)$, определяющего значения скоростей поступления маркеров в маркерные корзины. Задача оптимизации заключается в определении такого значения вектора \mathbf{r} , при котором значение $Y(r)$ будет минимальным. Если в системе нет задержек, этот вектор определяется как

$$\mathbf{r}(t_k) = F(T(t_k), a(t_{k-1}), Q(t_{k-1})),$$

где первый параметр определяет входное воздействие на систему в момент t_k , а второй и третий – состояние системы в предыдущий момент времени. Вектор $\mathbf{a}(t_{k-1})$ определяет состояние всех маркерных корзин:

$$\mathbf{a}(t_{k-1}) = (a_1(t_{k-1}), a_2(t_{k-1}), \dots, a_n(t_{k-1})).$$

Если принять во внимание временную задержку передачи и обработки данных, то при том же законе управления F управляющий вектор примет вид

$$\mathbf{r}_d(t_k) = F(T(t_{k-d_1}), a(t_{k-1-d_2}), Q(t_{k-1-d_3})),$$

где d_1, d_2, d_3 – число интервалов времени, за которое информация доходит до объекта управления. Самый простой способ определения параметров маркерных корзин – это нахождение значения \mathbf{r} только по входному воздействию, т. е. по трафику, поступающему на маркерные корзины. Если известны статистические характеристики трафика – такие, как среднее, максимальное значения и дисперсия – значение \mathbf{r} можно определить, исходя из них.

V. ВЫБОР ПАРАМЕТРОВ МАРКЕРНЫХ КОРЗИН

Возможно использование двух подходов. Первый – использование пикового значения трафика: $r(t_k) = T_p$. При таком значении управляющего вектора число маркеров во всех маркерных корзинах всегда будет достаточным для трафика любого объема. Эта стратегия не слишком эффективна, так как максимальный битрейт потока данных с переменной скоростью часто бывает гораздо большим, чем средний, и, следовательно, большую часть времени потоку выделяется слишком много ресурсов. При этом маркерная корзина будет беспрепятственно пропускать трафик пиковых объемов, перегружая всю сеть.

Второй подход – использование среднего значения трафика: $r(t_k) = K \cdot T_m$, где K очевидно, близко к единице. По сравнению с предыдущей эта стратегия выбора параметров корзин гораздо эффективней. Экспериментально было установлено, что оптимальными являются значения коэффициента K на несколько десятых большие единицы.

Для описанных ранее методов регулировки параметров маркерных корзин управляющий вектор можно менять динамически, подстраиваясь под трафик по мере его поступления. Аналогично, для второй стратегии

$$\mathbf{r}(t_k) = \frac{\sum_{m=k-w}^k T_p(t_m)}{w}.$$

Для системы с задержкой:

$$\mathbf{r}_d(t_k) = \frac{\sum_{m=k-d-w}^{k-d} T_p(t_m)}{w}.$$

Потери на маркерных корзинах всегда предпочтительнее потерь на мультиплексоре, так как последние вызывают непредсказуемое поведение сети.

Если не вводить в алгоритм приоритеты потоков данных (считать приоритеты всех потоков равными), то очевидным решением будет распределение пропускной способности канала между источниками данных пропорционально объемам трафика, передаваемого каждым из источников. Другими словами, если доступная пропускная способность в момент t равна $\Theta(t_k)$, то пропускная способность канала каждого источника должна быть ограничена величиной

$$\theta_i(t_k) = \frac{T_i(t_k)}{\sum_{j=1}^n T_j(t_k)} \Theta(t_k), \quad (2)$$

где $i = 1, 2, \dots, n$.

Значение $\Theta(t_k)$ зависит от загруженности мультиплексора и определяется разностью размеров буфера очереди и самой очереди в начале соответствующего интервала времени:

$$\Theta(t_k) = S - O(t_k).$$

Выделять потоку данных пропускную способность выше требуемой для его беспрепятственной передачи нецелесообразно. Таким образом, действительное количество требуемых маркеров

$$\tau_i(t_k) = \max\{\theta_i(t_k), T_i(t_k)\}.$$

Результирующее значение скорости пополнения маркерных корзин маркерами определяется настоящими и требуемыми значениями числа маркеров:

$$r_i(t_k) = [\tau_i(t_k) - a_i(t_k)] \cdot I(\tau_i(t_k) > a_i(t_k)).$$

Таким образом, если маркерная корзина уже содержит необходимое число маркеров, новые маркеры в него не поступают. Если же существующих маркеров недостаточно, в корзину будут добавлены только недостающие.

Если разным пользователям требуется назначить разные приоритеты, то выражение для пропорционального распределения пропускной способности (2) нужно модифицировать следующим образом:

$$\theta_i(t_k) = \frac{\sigma_i(t_k) T_i(t_k)}{\sum_{j=1}^n \sigma_j(t_k) T_j(t_k)} \Theta(t_k),$$

где σ_i ($i = 1, 2, \dots, n$) – коэффициенты, определяющие приоритеты соответствующих маркерных корзин. Коэффициенты должны быть нормированы, т. е.

$$0 \leq \sigma_i(t_k) \leq 1, \quad \sum_{i=1}^n \sigma_i(t_k) = 1.$$

Если задержка передачи информации о поступающем трафике и/или о состояниях мультиплексора и маркерных корзин значительна, то приведенные ранее алгоритмы необходимо модифицировать для ее учета.

VI. ОПТИМИЗАЦИЯ СЕТЕВОГО ТРАФИКА

Для выбора параметров маркерных корзин также можно использовать алгоритмы оптимизации. Разработанная модель может использоваться для оптимизации только как «черный ящик»: связь между характеристиками возможных решений и результатами их использования достаточно сложна, а результат достигается с помощью имитационного моделирования. Единственным способом поиска оптимума является семплирование, т. е. получение большого числа реализаций. Выбор очередных действий в поиске решения возможен лишь на основе полученных ранее результатов семплирования (никакой иной информации о зависимости между решением и его оптимальностью не существует). Дискретность пространства возможных решений сводит задачу к комбинаторной. Кроме того, нахождение точного глобального оптимума не является обязательным условием, в особенности учитывая его вычислительную сложность. Таким образом, для данной задачи следует использовать метаэвристические алгоритмы Монте-Карло [4].

Очевидно, что вероятные решения задачи оптимизации будут представлять собой векторы $\mathbf{r} = \{r_k\} = \{(r_{1,k}, r_{2,k}, \dots, r_{n,k})\}$, где $k = 0, 1, \dots, K$ – индексы временных отрезков, на которых действует данный управляющий вектор; $r_{i,k} \in Z$.

Успешность решения будет определяться значением целевой функции, получаемым в результате проведения имитационного эксперимента с использованием разработанной модели. Для определения области поиска решения необходимо установить граничные значения для элементов решения. Очевидно, что скорость поступления токенов в маркерные корзины не может быть меньше нуля. Кроме того, бессмысленно устанавливать скорость больше, чем размер корзины. Таким образом, $0 \leq r_{i,k} \leq s_i \quad \forall k \in \{0, 1, \dots, K\}, i \in \{1, 2, \dots, n\}$, и задача сводится к комбинаторной оптимизации на конечном множестве потенциальных решений.

VII. РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

В экспериментах имитировалось поведение участка сети с двумя источниками трафика и соответствующими им фильтрами token bucket.

Использовались следующие параметры элементов системы:

- параметры целевой функции: $\alpha = 100$, $\beta = 1$, $\gamma = 0,1$;
- параметры фильтров: размер – 10000, частота поступления токенов – 5000;
- параметры мультимплексора: размер буфера – 15000, частота передачи – 10000;
- фильтры имеют равный приоритет.

Для динамического изменения параметров фильтров, основанных на определении характеристик трафика, было

установлено значение размера окна наблюдения, равное пяти пакетам.

В генетическом алгоритме применялись генетические операторы инверсии, мутации и одноточечного скрещивания. Размер популяции был выбран равным 25, а число итераций – 50.

Число итераций алгоритма подъема было установлено равным 100. Кроме того, операция мутации была реализована таким образом, чтобы иметь возможность весьма значительно изменить текущее решение. Для алгоритма гармонического поиска также было выбрано число итераций, равное 100.

Предустановленные параметры элементов сети и способов оптимизации были экспериментально подобраны так, чтобы обеспечить наилучший результат оптимизации. Для всех алгоритмов оптимизации критерием останова является достижение определенного числа итераций. Это позволяет быть уверенным в одинаковой продолжительности их работы при каждом запуске. Эта характеристика является важной, так как варьирование скорости обработки трафика приводит к непредсказуемому поведению сети.

В качестве исходных данных для проведения имитационных экспериментов были использованы готовые записи трафиков [5]. Наилучший результат оптимизации показал алгоритм подъема. Результаты первых четырех стратегий нестабильны и сильно зависят от характера трафика. Алгоритм пропорционального распределения доступной пропускной способности стабильно показывает хорошие результаты, имея при этом крайне небольшую вычислительную сложность.

VIII. ЗАКЛЮЧЕНИЕ

Рассмотрены проблемы управления и оптимизации сетевого трафика. В качестве логико-математической модели использован алгоритм маркерной корзины и такие методы оптимизации как генетический алгоритм, алгоритм подъема и алгоритм гармонии. Наилучший результат оптимизации показал алгоритм подъема в сочетании с генетическим алгоритмом.

СПИСОК ЛИТЕРАТУРЫ

- [1] Evans J., Filsfil C. Deploying IP and MPLS QoS for Multiservice Networks: Theory and Practice. Morgan Kaufmann, 2007.
- [2] Ahmed N.U., Qun. Wang, Orozco-Barbosa L. Systems Approach to Modeling the Token Bucket Algorithm in Computer Networks // Mathematical Problems in Engineering. 2002. № 8. pp. 265–279.
- [3] Ahmed N. U., Bo Li, Orozco-Barbosa L. Modeling and Optimization of Computer Network Traffic Controllers // Mathematical Problems in Engineering. 2005. № 6. pp. 617–640.
- [4] Stuart J Russell, Peter Norvig, Artificial Intelligence: A Modern Approach, 2-nd ed., Prentice Hall, 2003.
- [5] Bhattacharya A., Parlos A.G., Atiya A.F. Prediction of MPEG-coded video source traffic using recurrent neural networks // IEEE Transactions on Signal Processing. 2003. № 8. pp. 2177–2190.