УДК 004.7

О ЗАДАЧЕ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ ЗАГРУЗКОЙ ОЧЕРЕДЕЙ МАРШРУТИЗАТОРА

И.С. Гарькавый
Московский государственный университет имени М.В.Ломоносова
Россия, 119991, Москва, Ленинские горы, д. 1
E-mail: garx@asvk.cs.msu.ru

Е.П. Степанов
Московский государственный университет имени М.В.Ломоносова
Россия, 119991, Москва, Ленинские горы, д. 1
E-mail: estepanov@lvk.cs.msu.ru

Ключевые слова: задачи управления, компьютерные сети, балансировка нагрузки, Network Powered by Computing.

Аннотация: В сетях нового поколения, таких как Network Powered by Computing, проблема минимизации задержки передачи данных является одной из ключевых из-за повышений требований приложений. Одним из подходов к минимизации задержки является обеспечение сбалансированной загрузки очередей на портах сетевых маршрутизаторов. Этого можно достичь при помощи балансировки пакетов одного и того же потока по разным выходным портам маршрутизатора. Задача такой балансировки пакетов по очередям также востребована в магистральных сетях, сетях SD-WAN. В докладе представлена математическая модель функционирования маршрутизатора. В рамках представленной модели поставлена задача оптимального управления загрузкой очередей, минимизирующего отклонения загрузки очередей разных портов маршрутизатора от среднего по портам.

1. Введение

В сетях нового поколения, таких как Network Powered by Computing, проблема минимизации задержки передачи данных является одной из ключевых из-за повышений требований приложений [1].

Работ, посвященных проблеме оптимального управления очередями в терминах математической теории оптимального управления, мало. Одна из основных трудностей при таком подходе заключается в том, что целевая функция требует моделирования наполнения очередей в каждый момент времени. В работах [2-4] построена модель наполнения очередей при помощи разностных уравнений и ставится схожая задача оптимального управления загрузкой очередей на выходных портах маршрутизатора. При этом также рассматривается сбалансированность загрузки очередей маршрутизаторов как критерий оптимальности управления. Однако, в цитируемых работах рассматривается случай направления пакетов одного потока на один и тот же выходной порт, а не на несколько.

Одним из подходов к минимизации задержки является обеспечение сбалансированной загрузки очередей на портах сетевых маршрутизаторов. Этого можно достичь при помощи балансировки пакетов одного и того же потока по разным выходным портам маршрутизатора. Задача такой балансировки пакетов по очередям также востребована в магистральных сетях, сетях SD-WAN. В работе задача попакетной балансировки нагрузки по выходным портам маршрутизатора рассматривается в рамках теории оптимального управления. Чтобы предотвратить частое переупорядочивание пакетов на стороне получателя, в задаче рассматривается балансировка флоулетов [5-10].

В докладе представлена математическая модель функционирования маршрутизатора. Для каждого поступающего в маршрутизатор пакета задано множество допустимых выходных портов для его передачи. Функция управления отображает количество поступивших пакетов на матрицу распределения пакетов по выходным портам. Длина очереди в каждый момент времени определяется при помощи нелинейных разностных уравнений.

В рамках представленной модели поставлена задача оптимального управления загрузкой очередей. Требуется найти функцию управления, минимизирующую вариацию длин очередей на разных выходных портах на рассматриваемом временном интервале. То есть, поступающие в маршрутизатор пакеты должны распределяться по выходным портам маршрутизатора, равномерно заполняя очереди выходных портов.

Таким образом, в терминах построенной модели функционирования маршрутизатора поставлена задача оптимального управления загрузкой очередей, и показана актуальность поставленной задачи. В дальнейшем планируется исследовать методы решения этой задачи, такие как обучение с подкреплением, динамическое программирование, синтез управления с помощью методов природной оптимизации.

2. Модели рассматриваемых объектов и постановка задачи

2.1. Модели входных потоков и работы маршрутизатора

Дан маршрутизатор с n входными и n выходными портами.

Маршрутизатор проверяет новые поступившие пакеты на входные порты с периодом t_p секунд. Будем измерять время в модели в тактах длины t_p.

$\{t\} = \{0, \dots, T\}$ (единица времени – один такт) – интервал дискретного времени, на котором рассматривается работа маршрутизатора. В каждый из этих моментов t (включая $t = 0$) маршрутизатор извлекает все пакеты, поступившие в маршрутизатор в течение интервала $(t-1, t]$, и распределяет их по выходным портам. Временем передачи пакета с входного на выходной порт пренебрегается.

На каждом выходном порту есть единственная FIFO-очередь пакетов на выдачу в канал. Размер очереди (максимальная длина) не ограничен. В начальный момент времени очереди пусты.

Каждый поступающий в маршрутизатор пакет принадлежит некоторому потоку.

$\{c\} = \{1, \dots, m\}$ – множество потоков.

В данной модели работы маршрутизатора будем использовать частный случай попакетной балансировки нагрузки по выходным портам – балансировка флоулетов. Последовательность пакетов каждого потока разбивается на *флоулеты* (flowlet) таким образом, чтобы длина интервала между поступлением флоулетов на маршрутизатор была больше порога флоулетов (flowlet gap). Существуют разные способы определения порога флоулетов [5-10]. Балансировка флоулетов решает проблему частого переупорядочивания пакетов на стороне получателя, которая возникает при попакетной балансировке.

Введем вероятностное пространство $\{\Omega, \mathfrak{A}, P\}$: Ω – множество элементарных исходов, $\mathfrak{A}$ – сигма-алгебра подмножеств Ω, P – мера вероятности.

Элементарный исход $\omega \in \Omega$ – расписание пакетов, поступающих в маршрутизатор на интервале $\{t\}$.

$\boldsymbol{a}(t) = \boldsymbol{a}(t)(\omega) \in \mathbb{R}_{\geq 0}^{m \times 1}, \omega \in \Omega$ – векторный случайный процесс, обозначающий суммарный размер пакетов, поступивших на все входные порты в интервале $(t-1, t]$, для каждого потока ($c = \overline{1,m}$), $t \in \{t\}$. Таким образом, элемент $a_c(t)$ вектора $\boldsymbol{a}(t)$ – скалярный случайный процесс, равный суммарному размеру пакетов потока c, поступивших в интервале $(t-1, t]$.

Распределение $\boldsymbol{a}(t)$ дано (с учетом характеристик потоков в исследуемой сети).

Также дан следующий векторный случайный процесс:

$\boldsymbol{h}(t) = \boldsymbol{h}(t)(\omega) \in \mathbb{R}_{\geq 0}^{m \times 1}$ – вектор порогов флоулетов потоков в момент t.

Распределение $\boldsymbol{h}(t)$ дано. $\boldsymbol{h}(t)$ может, например, рассчитываться по определенным правилам на основе таких характеристик потоков, как задержки передачи их пакетов от рассматриваемого маршрутизатора до получателя.

Для каждого потока дано множество допустимых выходных портов для передачи любых пакетов потока. Зададим все эти множества в виде одной матрицы:

$W \in \{0,1\}^{m \times n}$ – матрица допустимых выходных портов ($\overline{1,n}$) для потоков ($\overline{1,m}$).

$W_{cj} = 1$ тогда и только тогда, когда на выходной порт j можно передавать пакеты потока c, $j = \overline{1,n}$, $c = \overline{1,m}$.

$S^t = \langle \{\boldsymbol{a}(\tau)\}_{\tau=0}^{t}, W, \{\boldsymbol{h}(\tau)\}_{\tau=0}^{t} \rangle$ – история входных данных на интервале $[0, t]$.

Определим вспомогательные обозначения:

$\boldsymbol{q}(t) \in \overline{\mathbb{R}}^{m \times 1}$ – вектор последних моментов поступления пакетов потоков на момент t. Определяется через $\boldsymbol{a}(0), \dots, \boldsymbol{a}(t)$ следующим образом:

$$q_c(t) = \begin{cases} \max\limits_{\substack{\tau \leq t, \\ a_c(\tau) > 0}} \tau, \exists \tau \leq t: a_c(\tau) > 0 \\ -\infty, \text{иначе} \end{cases}.$$

Заметим, что $\boldsymbol{q}(t)$ можно также выразить через $\boldsymbol{a}(t)$ и $\boldsymbol{q}(t-1)$, $t \geq 1$.

$\boldsymbol{\nu}(t) \in \{0,1\}^{m \times 1}$ – вектор, обозначающий, прошло ли к моменту t время, равное текущему flowlet gap, с момента поступления последнего пакета соответствующего потока (0, если да, и 1, если нет). Определяется следующим образом:

$$\boldsymbol{\nu}(t) = \{\nu_1(t), \dots, \nu_m(t)\}, \nu_c(t) = \begin{cases} 0, t - q_c(t-1) \geq h_c(t) \\ 1, t - q_c(t-1) < h_c(t) \end{cases}, c = \overline{1,m}.$$

Если в момент t поступил пакет потока c, то при $\nu_c(t) = 0$ он будет принадлежать новому флоулету потока, а при $\nu_c(t) = 1$ – последнему из флоулетов потока.

За $O_{m,n}$ ($J_{m,n}$) обозначим матрицу $m \times n$ строку из нулей (соответственно единиц).

Введем следующую функцию:

$U(t, \boldsymbol{a}(t), S^t) \in \{0,1\}^{m \times n}$ – функция управления – отображение количества $\boldsymbol{a}(t)$ поступивших пакетов в момент t на матрицу распределения потоков по выходным портам, с учетом входных данных S^t от начального момента 0 до текущего момента t. Если $a_c(t) > 0$, то элемент $U_{cj}(t, \boldsymbol{a}(t), S^t)$ обозначает, передавать ли пакеты потока c на выходной порт j (1, если передавать, и 0, если нет).

Ограничения на функцию управления:

В рамках одного такта, для каждого потока должен быть определен единственный выходной порт для передачи:

(1) $$\boldsymbol{g}_1(U, t, S^t) = U(t, \boldsymbol{a}(t), S^t) \cdot J_{n,1} - J_{m,1} = O_{m,1}.$$

Пакеты должны быть распределены по допустимым выходным портам:

(2) $$G_2(U, t, S^t) = U(t, \boldsymbol{a}(t), S^t) \odot \left(J_{m,n} - W\right) = O_{m,n},$$

где $\odot$ – операция поэлементного умножения матриц.

Все пакеты потока в пределах текущего флоулета передаются на один и тот же выходной порт:

(3) $$G_3(U,t,S^t) = \left(U(t,\boldsymbol{a}(t),S^t) - U(t-1,\boldsymbol{a}(t-1),S^{t-1})\right) \odot \left(\boldsymbol{\nu}(t)\cdot J_{1,n}\right) = \\ = O_{m,n}, t \geq 1.$$

Определим $\{U\}$ – множество функций управления, удовлетворяющих ограничениям (1)-(3) при любых входных данных:

$$\{U\} = \left\{ U(t,\boldsymbol{a},S) \;\middle|\; \begin{matrix} \forall t \in \{t\}, \forall \widetilde{\boldsymbol{a}}^0,\dots,\widetilde{\boldsymbol{a}}^t, \widetilde{\boldsymbol{h}}^0,\dots,\widetilde{\boldsymbol{h}}^t \in \mathbb{R}_+^{m\times 1}, \\ \forall W \in \{0,1\}^{m\times n}, \tilde{S}^t = \langle \{\widetilde{\boldsymbol{a}}^\tau\}_{\tau=0}^t, W, \{\widetilde{\boldsymbol{h}}^\tau\}_{\tau=0}^t \rangle \\ \boldsymbol{g}_1(U,t,\tilde{S}^t) = O_{m,1} \\ G_2(U,t,\tilde{S}^t) = O_{m,n} \\ G_3(U,t,\tilde{S}^t) = O_{m,n}, t \geq 1 \end{matrix} \right\}.$$

Также дано: $\boldsymbol{r} \in \mathbb{R}_{\geq 0}^{n\times 1}$ – вектор пропускных способностей (размер данных, передаваемый за один такт) выходных каналов, смежных с выходными портами $\overline{1,n}$.

$\boldsymbol{b}(t) \in \mathbb{R}_{\geq 0}^{n\times 1}$ – случайный векторный процесс, обозначающий длины очередей на выходных портах $\overline{1,n}$ к моменту t. Определяется соотношениями:

$$\boldsymbol{b}(t+1) = [\boldsymbol{b}(t) + U^T(t,\boldsymbol{a}(t),S^t)\cdot \boldsymbol{a}(t) - \boldsymbol{r}]^+, t \in \{0,\dots,T-1\};$$
$$\boldsymbol{b}(0) = O_{n,1},$$

где $[x]^+ = \max(0,x)$.

Введем вспомогательные обозначения:

$\mu(t) = \frac{1}{n}\sum_{i=1}^n b_i(t)$ – средняя по выходным портам длина очереди.

$\delta(t) = \frac{1}{n}\sum_{i=1}^n \left(b_i(t) - \mu(t)\right)^2$ – средний (по портам) квадрат отклонения длины очереди от средней длины очереди (по портам).

2.2. Задача оптимального управления загрузкой очередей маршрутизатора

Дано:

- Модель входных потоков (случайный процесс $\boldsymbol{a}(t)$, матрица допустимых выходных портов W, случайный процесс $\boldsymbol{h}(t)$);
- Модель работы маршрутизатора.

Найти: функцию управления $U(t,\boldsymbol{a}(t),S^t)$:

- удовлетворяющую ограничениям (1)—(3);
- минимизирующую средние отклонения длин очередей:

$$U^* = argmin_{U\in\{U\}} F(U), F(U) = \mathbb{E}\left(\frac{1}{T+1}\sum_{t=0}^T \delta(t)\right).$$

3. Возможные методы решения поставленной задачи управления загрузкой очередей маршрутизатора

Опишем простейший алгоритм, получающий субоптимальное решение поставленной задачи. Он заключается в решении задачи оптимизации $\delta(t) \to \min$ в каждый момент времени t.

Пусть фиксирован момент $t \in \{t\}$ и элементарный исход ω, известна реализация $\widetilde{S^t}$ набора случайных величин S^t, а также значения $U(\tau,\widetilde{\boldsymbol{a}}^\tau,\tilde{S}^\tau)$, $0 \leq \tau < t$. Тогда, значение $U(t,\widetilde{\boldsymbol{a}}^t,\tilde{S}^t)$ определяется следующим образом:

$$U\left(t, \widetilde{\boldsymbol{a}}^t, \tilde{S}^t\right) = \underset{\substack{\boldsymbol{g}_1(U,t,\tilde{S}^t)=0 \\ G_2(U,t,\tilde{S}^t)=0 \\ t\geq 1 \to G_3(U,t,\tilde{S}^t)=0}}{argmin} \delta(t).$$

Эту задачу оптимизации можно решать, например, полным перебором значений матрицы U, коих не более mn в силу ограничения (1).

Данный алгоритм не учитывает ни полную историю потока, ни модель потоков, поэтому не достигает желаемого эффекта балансировки. Учет полной истории потока позволил бы с некоторой точностью предсказывать всплески потоков в будущем. Можно показать, что имея предсказание всплеска потока в будущий момент времени, можно улучшить значение $F(U)$ по сравнению с приведенным выше алгоритмом.

Перспективными методами решения поставленной задачи могут быть обучение с подкреплением, динамическое программирование, синтез управления с помощью методов природной оптимизации.

4. Заключение

В терминах построенной модели функционирования маршрутизатора поставлена задача оптимального управления загрузкой очередей с использованием балансировки флоулетов по выходным портам. В дальнейшем планируется исследовать методы решения этой задачи, такие как обучение с подкреплением, динамическое программирование, синтез управления с помощью методов природной оптимизации.

Список литературы

1. Smeliansky R. L. Network powered by computing // 2022 International Scientific and Technical Conference on Modern Computer Network Technologies (MoNeTeC). Moscow: IEEE, 2022. P. 1-5.
2. Лемешко А. В., Симоненко Д. В. Динамическая модель балансировки буферных и канальных ресурсов транспортной сети телекоммуникационной системы // 2010 Проблемы телекоммуникаций. Харьков, Украина. 2010. № 2 (2), С. 42-49.
3. Ali A. S. Dynamic model of multipath routing supporting traffic engineering queues with packets priorities // 2015 Second International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T). Kharkiv, Ukraine: IEEE, 2015. P. 79-81.
4. Lemeshko O., Lebedenko T., Al-Dulaimi A. Improvement of Method of Balanced Queue Management on Routers Interfaces of Telecommunication Networks // 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT). Lviv, Ukraine: IEEE, 2019. P. 170-175.
5. Benet C. H., Kassler A. J. FlowDyn: Towards a Dynamic Flowlet Gap Detection using Programmable Data Planes // Proceedings of the 8th International Conference on Cloud Networking (CloudNet). Coimbra, Portugal: IEEE, 2019. P. 1-7.
6. Alizadeh M., Edsall T., Dharmapurikar S., Vaidyanathan R., Chu K., Fingerhut A., Lam V. T., Matus F., Pan R., Yadav N., Varghese G. CONGA: distributed congestion-aware load balancing for datacenters // Proceedings of the 2014 ACM conference on SIGCOMM. Chicago, Illinois, USA: ACM, 2014. P. 503-514.
7. Katta N., Ghag A., Hira M., Keslassy I., Bergman A., Kim C., Rexford J. Clove: Congestion-Aware Load Balancing at the Virtual Edge // Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies. Incheon, Republic of Korea: ACM, 2017. P. 323-335.
8. Katta N., Hira M., Kim C., Sivaraman A., Rexford J. HULA: Scalable Load Balancing Using Programmable Data Planes // Proceedings of the Symposium on SDN Research. Santa Clara, CA, USA: ACM, 2016. P. 1-12.
9. Sinha S., Kandula S., Katabi D. Harnessing TCP's Burstiness with Flowlet Switching // Proceedings of the 3rd ACM Workshop on Hot Topics in Networks (Hotnets-III). Portland, Oregon, USA: Citeseer, 2004.
10. Vanini E., Pan R., Alizadeh M., Taheri P., Edsall T. Let it Flow: Resilient Asymmetric Load Balancing with Flowlet Switching // Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation. Boston, MA, USA: NSDI, 2017. P. 407-420.