

РАЗРАБОТКА ЭФФЕКТИВНОГО ГЕНЕТИЧЕСКОГО АЛГОРИТМА МАРШРУТИЗАЦИИ И ОБЕСПЕЧЕНИЯ КАЧЕСТВА ОБСЛУЖИВАНИЯ ДЛЯ ПРОГРАММНО-КОНФИГУРИРУЕМОЙ СЕТИ

В статье представлены результаты исследования, направленного на создание эффективных методов маршрутизации потоков данных в корпоративных программно-конфигурируемых сетях и облачных системах. Сформулирована оптимизационная задача проактивной маршрутизации и обеспечения качества обслуживания потоков данных с учетом гибких ограничений для потоков данных по минимальным гарантированным пропускным способностям и максимальным гарантированным задержкам. Предложен генетический алгоритм, решающий данную задачу и опирающийся на программно-конфигурируемые сети для получения информации о состоянии сети и установки правил в таблицы коммутаторов. Реализация предложенного алгоритма была предварительно протестирована с помощью разрабатываемого симулятора программно-конфигурируемой сети и показала свою эффективность. В будущем планируется детальное исследование предложенного алгоритма как с помощью симулятора, так и в условиях реальной программно-конфигурируемой сети.

Ключевые слова: корпоративные сети, облачные вычисления, программно-конфигурируемые сети, маршрутизация, обеспечение качества обслуживания.

В настоящее время большинство средних и крупных компаний имеют корпоративные сети или облачные системы, обеспечивающие функционирование их ИТ-инфраструктур. Как правило, они используются для развертывания различных корпоративных и бизнес-приложений, необходимых для эффективной работы компаний.

В корпоративных сетях и облачных системах используется технология Ethernet, для которой характерно применение классических протоколов маршрутизации, таких как OSPF, RIP и др. Основной их недостаток – отсутствие учета семантики передаваемых потоков данных, которая может быть использована для повышения эффективности маршрутизации данных.

Для исправления данного недостатка могут быть использованы программно-конфигурируемые сети (ПКС). В основе подхода ПКС лежит возможность динамически управлять пересылкой данных в сети с помощью открытого протокола OpenFlow. Все сетевые коммутаторы, поддерживающие OpenFlow, объединяются под управлением контроллера OpenFlow, который обеспечивает приложениям доступ к управлению сетью.

ПКС [1], [2] позволяют эффективно реализовывать собственные протоколы маршрутизации с помощью приложений для контроллера ПКС, это, в частности, дает возможность разработать алгоритмы маршрутизации потоков данных, учитывающих их семантику – комму-

никационные схемы и требования к качеству обслуживания (к максимальным задержкам и к минимальным пропускным способностям).

Созданный алгоритм маршрутизации использует проактивную схему (до появления потоков данных) установки правил коммутации в коммутаторы программно-конфигурируемой сети, что позволяет избежать дополнительных задержек.

В связи с ростом объемов мультимедийного, голосового (IP-телефония) и видео-трафика (видеоконференцсвязь) актуально обеспечение необходимых параметров качества обслуживания (максимальной гарантированной задержки, минимальной гарантированной пропускной способности, джиттера или процента потери пакетов) для соответствующих потоков данных. Созданный алгоритм маршрутизации и обеспечения качества обслуживания учитывает только первые два параметра качества обслуживания.

Проблема маршрутизации и обеспечения качества обслуживания

Задача маршрутизации сетевого трафика с помощью ПКС решается в работах [3], [4], [5], однако предложенные авторами алгоритмы не учитывают необходимость обеспечения параметров QoS для прокладываемого потока данных и ранее установленных потоков. Существующие алгоритмы обеспечения QoS [6], [7] для ПКС также недостаточно эффективны. В работе [6] описыва-

ется подход к динамической прокладке маршрутов передачи мультимедийных потоков, обеспечивающих гарантированную максимальную задержку с помощью алгоритма Lagrangian Relaxation Based Aggregated. Однако, авторы рассматривают только случаи единичных задержек по каждой сетевой связи и не учитывают минимальную гарантированную пропускную способность. Схожий подход описывается в [7], авторы ставят и решают оптимизационную задачу по передаче мультимедийного трафика без потерь по альтернативным путям, оставляя короткие пути для обычных данных. Однако, они осуществляют оптимизацию по задержкам и не учитывают необходимость обеспечения гарантированной пропускной способности.

В рамках настоящего исследования предложен подход, сочетающий маршрутизацию потоков данных и обеспечение для них двух параметров QoS – минимальной гарантированной пропускной способности и максимальной гарантированной задержки.

Пусть $G_T = (V, E)$ – ориентированный мультиграф, описывающий текущую топологию сети в некоторый момент времени t . Множество его вершин $V = Nodes \cup NetDevices$ является объединением множества узлов (серверов) и других сетевых устройств (коммутаторов, граничных шлюзов, СХД и т. п.).

Каждая дуга $e \in E$ соответствует некоторой сетевой связи между вершинами $beg(e) \in V$ и $end(e) \in V$. У нее также есть противоположная дуга, т. к. связь дуплексная. Между двумя вершинами может быть несколько параллельных дуг, например, параллельные соединения между маршрутизаторами. Это дает множество альтернативных маршрутов для передачи данных и обеспечения параметров QoS.

На множестве дуг E заданы две функции:

1. $b : E \rightarrow R^+ \cup \{0\}$ – отображение, характеризующее текущую пропускную способность каждой дуги в момент времени t .

2. $d : E \rightarrow R^+ \cup \{0\}$ – задержка на соответствующем выходном порту дуги в момент времени t .

Обозначим в качестве ориентированного графа $G_p = (V', C)$ коммуникационный паттерн некоторой распределенной программы, которая выполняется в корпоративной сети. Здесь $V' \subseteq Nodes$ – множество серверов, на которых работают компоненты распределенной про-

граммы, C – дуги, указывающие на потоки данных между ними.

Заметим, что при необходимости множество V' также может включать прочие сетевые устройства, например, граничные шлюзы, если большая часть трафика идет от удаленных пользователей к узлам.

На множестве дуг C заданы три функции:

1. $\underline{b} : C \rightarrow R^+ \cup \{0\}$ – минимальная гарантированная пропускная способность потоков данных, соответствующих данной дуге.

2. $\bar{d} : C \rightarrow R^+ \cup \{0\}$ – их максимальная гарантированная задержка.

3. $d : C \rightarrow R^+ \cup \{0\}$ – оценка средней задержки, которая возникнет при обработке пакетов потоков данных на портах сетевых устройств.

Алгоритм маршрутизации и обеспечения QoS должен построить такую функцию $\varphi : C \rightarrow P(G_T)$, которая каждому потоку данных $c \in C$ ставит в соответствие маршрут его передачи r , ведущий из вершины $beg(c)$ в $end(c)$. Здесь в качестве $P(G_T)$ обозначено множество маршрутов между любыми парами вершин в графе-топологии G_T .

Функция φ может быть описана в виде вектора $R = (r_1, \dots, r_{|C|})$, где $r_i = \varphi(c_i)$ – маршрут для потока данных $c_i \in C$.

Также введем дополнительное обозначение, пусть $\psi : E \rightarrow 2^C$ – функция, ставящая в соответствие каждой сетевой связи $e \in E$ множество потоков данных $c \in C$, для которых соответствующие маршруты проходят через e , т. е. $\psi(e) = \{c \in C \mid r = \varphi(c) \& e \in r\}$.

Очевидно, что вектор R должен содержать маршруты, которые удовлетворяют следующим ограничениям QoS:

1. Пропускная способность каждого маршрута r_i с учетом влияния других потоков данных не должна быть меньше гарантированной пропускной способности для потока c_i :

$$\forall r_i \min_{e \in r_i} \{b(e) - \sum_{c \in \psi(e) \setminus \{c_i\}} \underline{b}(c)\} \geq \underline{b}(c_i).$$

2. Суммарная задержка каждого маршрута r_i , с учетом влияния других потоков данных, не должна быть больше гарантированной задержки для потока c_i :

$$\forall r_i \sum_{e \in r_i} (d(e) + \sum_{c \in \psi(e) \setminus \{c_i\}} \bar{d}(c)) \leq \bar{d}(c_i).$$

Следует заметить, что данные ограничения являются гибкими и в некоторой степени могут

быть нарушены для части потоков данных (например, в случае, когда сеть перегружена), поэтому в качестве оптимизируемой функции будет принята следующая:

$R^A = (r_1^A, \dots, r_{k-1}^A, r_k^A, \dots, r_{|C|}^A)$ и $R^B = (r_1^B, \dots, r_{k-1}^B, r_k^B, \dots, r_{|C|}^B)$ формируются две дочерние хромосомы $R_1^{AB} = (r_1^A, \dots, r_{k-1}^A, r_k^B, \dots, r_{|C|}^B)$ и $R_2^{AB} = (r_1^B, \dots, r_{k-1}^B, r_k^A, \dots, r_{|C|}^A)$.
Операция мутации хромосомы $R = (r_1, \dots, r_{|C|})$

$$\begin{aligned}
 H(R) = & \sum_{\substack{c_i \in C: \\ \min_{e \in r_i} \{b(e) - \sum_{c \in \Psi(e) \setminus \{c_i\}} b(c)\} \geq \bar{b}(c_i) \& \\ \sum_{e \in r_i} d(e) + \sum_{c \in \Psi(e) \setminus \{c_i\}} d(c) \leq \bar{d}(c_i)}} \left[\alpha_b (\min_{e \in r_i} \{b(e) - \sum_{c \in \Psi(e) \setminus \{c_i\}} b(c)\} - \bar{b}(c_i)) + \alpha_d (\bar{d}(c_i) - \sum_{e \in r_i} d(e) - \sum_{c \in \Psi(e) \setminus \{c_i\}} d(c)) \right] + \\
 & + \beta_b \cdot \sum_{\substack{c_i \in C: \\ \min_{e \in r_i} \{b(e) - \sum_{c \in \Psi(e) \setminus \{c_i\}} b(c)\} < \bar{b}(c_i)}} \left[\min_{e \in r_i} \{b(e) - \sum_{c \in \Psi(e) \setminus \{c_i\}} b(c)\} - \bar{b}(c_i) \right] + \\
 & + \beta_d \cdot \sum_{\substack{c_i \in C: \\ \sum_{e \in r_i} d(e) + \sum_{c \in \Psi(e) \setminus \{c_i\}} d(c) > \bar{d}(c_i)}} \left| \bar{d}(c_i) - \sum_{e \in r_i} d(e) - \sum_{c \in \Psi(e) \setminus \{c_i\}} d(c) \right| \rightarrow \max.
 \end{aligned}$$

Здесь $\alpha_b > 0$ и $\alpha_d > 0$ – поощрения за соблюдение соответствующих ограничений по пропускной способности и задержкам, $\beta_b > 0$ и $\beta_d > 0$ – штрафы за их несоблюдение.

Также имеются жесткие ограничения на маршруты $r_i = (e_{i1}, \dots, e_{in_i})$:

1. r_i действительно является маршрутом:

$$\forall r_i \quad \forall j = \overline{1, n_i - 1} \quad \text{end}(e_{ij}) = \text{beg}(e_{i,j+1}).$$

2. r_i должен начинаться в вершине, соответствующей началу дуги c_i и заканчиваться в вершине, соответствующей концу c_i , т. е.:

$$\forall r_i \quad \text{beg}(e_{i1}) = \text{beg}(c_i) \& \text{end}(e_{in_i}) = \text{end}(c_i).$$

3. r_i не должен проходить несколько раз через одну и ту же вершину, т. е.:

$$\forall r_i \quad \forall j, k = \overline{1, n_i} \quad j \neq k \Rightarrow \text{beg}(e_{ij}) \neq \text{beg}(e_{ik}).$$

Разработка генетического алгоритма для маршрутизации и обеспечения качества обслуживания

Данная оптимизационная задача может быть решена с помощью генетического алгоритма.

Решение задачи кодируется хромосомой, представляющей собой вектор маршрутов $R = (r_1, \dots, r_{|C|})$. Популяция хромосом имеет фиксированный размер N .

Операция скрещивания двух хромосом представляет обычное одноточечное скрещивание. Выбирается случайное число $k \in \overline{2, |C| - 1}$, затем на основе родительских хромосом

представляет собой выбор случайного числа $k \in \overline{1, |C|}$, после чего маршрут r_k случайным образом меняется. Для этого случайно выбираются две его вершины $\text{beg}(e_{kj})$ и $\text{beg}(e_{ks})$ ($1 \leq j < s \leq n_k$) между которыми есть альтернативный маршрут (он не должен нарушать третье жесткое ограничение), и в r_k участок e_{kj}, \dots, e_{ks-1} заменяется альтернативным e'_{kj}, \dots, e'_{kp} .

Операция селекции сочетает в себе выбор P_{top} процентов лучших хромосом в каждом поколении, остальные хромосомы выбираются по принципу рулетки пропорционально значению оптимизируемой функции.

Условиями окончания работы генетического алгоритма будут: превышение максимального времени T_{max} и отсутствие существенных улучшений в среднем значении оптимизируемой функции на протяжении нескольких поколений. Последнее условие может быть записано с помощью следующего неравенства:

$$\max_{j=i-G+1, d} |\bar{H}_{Gi} - \bar{H}(R_j)| < \varepsilon,$$

где G – количество контролируемых поколений, \bar{H}_{Gi} – среднее значение оптимизируемой функции на протяжении G последних поколений, $\bar{H}(R_j)$ – среднее значение для j -го предыдущего поколения, ε – заданная степень толерантности.

Приведем по шагам разработанный алгоритм:

- Шаг 1. Замерить текущий момент времени T_{start} .

Шаг 2. Создать $Population_i$ начальную популяцию размера N . В качестве одной из хромосом R выбрать маршруты, проложенные с помощью алгоритма Дейкстры, запущенного из каждой вершины V' , руководствуясь минимизацией суммарных ограничений по задержкам. Остальные хромосомы сгенерировать случайным образом.

Шаг 3. Положить в качестве номера итерации значение $i := 1$.

Шаг 4. Пока $T_{current} - T_{start} < T_{max}$ и $i < G$ (или $\max_{j=i-G+1,i} |\bar{H}_{Gi} - \bar{H}(R_j)| \geq \epsilon$) необходимо выполнить следующие шаги:

Шаг 4.1. Над хромосомами $Population_i$ с вероятностью P выполнить операции скрещивания, объединив родительские хромосомы в случайные пары. Получаемые дочерние хромосомы сохранить в $Population'_i$.

Шаг 4.2. Для хромосом $Population'_i$ выполнить операцию мутации с вероятностью Q .

Шаг 4.3. Объединить родительские и дочерние популяции

$$Population''_i := Population_i \cup Population'_i$$

Шаг 4.4. Для $Population''_i$ выполнить операцию селекции, выбранные хромосомы сохранить в $Population_{i+1}$.

Шаг 4.5. Увеличить на единицу номер итерации $i := i + 1$ и перейти к шагу 4.

Шаг 5. Установить в таблицы коммутаторов OpenFlow все маршруты из $Population_i$.

В данном алгоритме параметр $T_{current}$ представляет собой текущее время, получаемое от системных часов.

Обсуждение полученных результатов

Созданный генетический алгоритм маршрутизации и обеспечения качества обслуживания

продемонстрировал свою эффективность в результате его экспериментального исследования на разрабатываемом симуляторе ПКС. Полученные результаты носят предварительный характер и будут опубликованы в будущем.

Разработанный алгоритм предназначен для проактивного вычисления маршрутов передачи данных распределенных программ в случае, если известны их коммуникационные паттерны. Прочий, не критичный к задержкам и ограничениям пропускной способности, трафик должен маршрутизироваться с помощью стандартных алгоритмов.

Следует отдельно заметить, что предложенный алгоритм работает в частном случае, когда G_p содержит две вершины, что соответствует вычислению одного маршрута в графе.

Выводы

В результате проведенного исследования сформулирована оптимизационная задача проактивной маршрутизации и обеспечения качества обслуживания потоков данных в ПКС с учетом гибких ограничений для потоков данных по минимальным гарантированным пропускным способностям и максимальным гарантированным задержкам. Предложен генетический алгоритм, решающий данную задачу и опирающийся на ПКС для получения информации о состоянии сети, установки правил в таблицы коммутаторов ПКС с целью реализации вычисленных маршрутов.

Реализация предложенного алгоритма была предварительно протестирована с помощью разрабатываемого симулятора ПКС и показала свою эффективность. В будущем планируется детальные исследования предложенного алгоритма как с помощью симулятора ПКС, так и в условиях реальной ПКС.

3.09.2014

Исследования выполнены при поддержке РФФИ (проекты №13-07-97046, №14-07-97034 и №13-01-97050), программы У.М.Н.И.К. Фонда содействия развитию малых форм предприятий в научно-технической сфере (договор №2628ГУ1/2014)

Список литературы:

1. Полежаев, П.Н. Математическая модель распределенного вычислительного центра обработки данных с программно-конфигурируемыми сетями его сегментов / П.Н. Полежаев // Вестник «Оренбургского государственного университета», 2013. – 5(154). – С. 198–204.

2. Полежаев, П.Н. Об эффективности алгоритмов планирования задач управления потоками данных облачных грид-систем / П.Н. Полежаев // Вестник «Оренбургского государственного университета», 2014. – 3(164). – С. 168–172.
3. Ibanez, G. A Small Data Center Network of ARP-Path Bridges made of Openflow Switches / G. Ibanez [и др.] // The 36th IEEE Conference on Local Computer Networks (LCN). – 2011.
4. Shimonishi, H. Building Hierarchical Switch Network Using OpenFlow / H. Shimonishi [и др.] // 2009 International Conference on Intelligent Networking and Collaborative Systems. – 2009. – P. 391–394.
5. Tavakoli, A. Applying NOX to the datacenter [Электронный ресурс] / A. Tavakoli [и др.] // Proceedings of the 8th ACM Workshop on Hot Topics in Networks (HotNets-VIII). – New York, 2009. – Режим доступа: <http://www.icsi.berkeley.edu/pubs/networking/applyingnox09.pdf>.
6. Egilmez, H.E. OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end quality of service over software-dened networks / H.E. Egilmez [и др.].
7. Kim, W. Automated and Scalable QoS Control for Network Convergence / W. Kim [и др.] // In Proc. INM/WREN, 2010.

Сведение об авторах:

Полежаев Петр Николаевич, преподаватель кафедры компьютерной безопасности и математического обеспечения информационных систем
Оренбургского государственного университета

460018, г. Оренбург, пр-т Победы, 13, ауд. 20520, тел.: (3532) 372534,
e-mail: peter.polezhaev@mail.ru

Бахарева Надежда Федоровна, профессор кафедры геометрии и компьютерных наук
Оренбургского государственного университета, доктор технических наук,
e-mail: nadin1956_04@inbox.ru

Шухман Александр Евгеньевич, заведующий кафедрой геометрии и компьютерных наук
Оренбургского государственного университета, кандидат педагогических наук, доцент

460018, г. Оренбург, пр-т Победы, 13, ауд. 1502, тел.: (3532) 372539,
e-mail: shukhman@gmail.com