

Алгоритм распределения разнородного трафика в программно-конфигурируемой сети

Т.Н. Нишанбаев, С.О. Махмудов (ТУИТ)

В статье приводится алгоритм решения задачи распределения суммарного потока по каналам связи уровня инфраструктуры программно-конфигурируемой сети. Предлагается алгоритм поиска кратчайших путей на графе заданной структуры.

Мақолада дастурий конфигурацияланадиган тармоғи инфраструктура сатҳи алоқа каналларида ҳар хил турдаги трафик оқимларини тақсимлаш масаласи ечимининг алгоритми келтирилади. Берилган граф тугунлари орасида энг қисқа йўлни топиш алгоритми таклиф этилади.

The article provides an algorithm for solving the problem of distributing the total flow over communication channels of the infrastructure level of a software-defined network. An algorithm is proposed for finding the shortest paths on a graph of a given structure

В [1] транспортировку пакетов разнородного трафика в мульти-сервисной сети предлагается осуществлять на базе программно-конфигурируемой сети (ПКС). На базе потоковой модели формализуется оптимизационная задача распределения потоков разнородного трафика на уровне инфраструктуры ПКС. В качестве целевой функции обосновывается задачи оптимальной маршрутизации (распределения разнородного трафика) по критерию средней задержки.

В данной статье разрабатывается алгоритм решения задачи распределения суммарного потока по магистральным каналам связи уровня инфраструктуры программно-конфигурируемой сети.

Алгоритм основан на потоковой модели, описанной в [1] и реализуется в предположении стационарности системы, то есть при выполнении ограничений на реализуемости потока. Согласно модели, если нитка мультипотока любого z -класса $\Psi_{i,j}^{(z)}$ есть маршрут из узла i в узел j , на любом ребре которого (x, z) $\lambda_{xz}^{(z)} > 0$, то выпуклая функция $\bar{T}$, минимизируется мультипоток $\Lambda = \{\lambda_{ij}^{(z)}\}$ в том и только в том случае, когда любая нитка данного мультипотока есть кратчайший путь в метрике η_{ij} , то есть

$$\eta_{ij} = \frac{\partial \bar{T}_{ij}}{\partial \lambda_{ij}} = \mu_{ij} / \gamma(\mu_{ij} - \lambda_{ij})^2$$

Значение η_{ij} характеризует степень изменения целевой функции в результате приращения потока в (i, j) - ребре. В данном случае под приращением потока в (i, j) - ребре понимается переброска части трафика z - го класса из данной пути, содержащий (i, j) - ребров другой путь или изменение значения внешнего трафика z - го класса соответствующего узла, потоки которого протекают через данное ребро.

Задача, решаемая на основе потоковой модели сводится к следующему:

1) задаются исходные данные, которыми являются: параметры топологической структуры уровня инфраструктуры ПКС; пропускные способности каналов; интенсивности внешнего трафика пакетов всех Z - классовна входе коммутатора; значение матрицы тяготения распределения потоков от коммутаторов к другим коммутаторам;

2) $\varepsilon = 0$; формируется начальный вектор $\vec{\xi}^{(z, \varepsilon)}$ для трафиков всех классов: по заданным пропускным способностям каналов связи (весов ребер) (в случае экспоненциального распределения моментов обслуживания заявок за весь ребра может быть принята величина $\mu_i = 1/t_i$, где t_i - среднее время доставки), вычисляются K кратчайших путей между узлами и по этим кратчайшим путям согласно матрице тяготения направляются нитки потоков соответствующих внешних трафиков. В результате такого распределения в ребрах формируются многопродуктовые потоки:

$$\lambda_i = \sum_{i \in Z} \lambda_{z,i}, Z=1,2,3$$

Если во всех ребрах заданной топологии выполняются условие $\lambda_i \leq \mu_i$ (то есть выполняется условие реализуемости потока), то вектор потоков $\vec{\xi}^{(z, \varepsilon)} = \{\lambda_i\}$ принимается за начальный вектор;

3) для каждого ребра вычисляются новые веса:

$$\eta_{ij} = \frac{\partial \bar{T}}{\partial \lambda_i} = \mu_{ij} / \gamma(\mu_i - \lambda_i)^2$$

4) вычисляются значение коэффициента, по которому оценивается средняя задержка и соответствующая ей значение целевой функции $\bar{T}$:

$$\beta(\varepsilon) = \sum_{i \in M} \eta_i \lambda_i$$

5) определяются кратчайшие пути в метрике η_i (т.е. в метрике "маргинальных" задержек);

6) выполняются операции по перераспределению потоков соответствующих трафиков при учете новых вычисленных кратчайших путей. Рассмотрим этот этап несколько подробнее. Пусть:

L_{ij}^I - ранее определенный путь, по которому направлен поток всех трафиков между узлами (i, j) (на начальном этапе работы алгоритма данный путь определяется по весу ребра, равным μ_i);

L_{ij}^{II} - вычисленный кратчайший путь между этими узлами на основе метрики маргинальных задержек, к которой планируется осуществить переброску части потоков из ранее направленной пути.

L_{ij}^I, L_{ij}^{II} - соответственно характеризуют длины путей (под длиной в данном случае понимается число каналов в пути);

$\lambda_k^{L,1}$ - результирующий поток по K - каналу при направлении по нему потоков всех видов трафиков ранее определенным кратчайшим путем;

$\lambda_q^{L,2}$ - результирующий поток в q - м канале, находящийся в вычисленном кратчайшем пути между (i, j) узлами.

Естественно, возникает вопрос: поток какого трафика (речь, видео, данные) следует перебросить из старого пути в новый? Потоки речевого и видео трафика являются информацией реального времени и требуют максимального времени передачи, однако они не очень критичны к определенным потерям. Компьютерные данные являются информацией нереального времени и не критичны ко времени передачи (могут передаваться с определенными задержками), но не допускают потерь пакетов. Переброска потоков речевых или видео сообщений требуют максимального занятия каналов уже занятых в определенной степени каналов аналогичными трафиками от других источников и потребителей, что может вызвать перегрузку каналов. А переброска части трафика γ_2 не будет требовать максимального непрерывного времени передачи. Следующим фактором, поясняющим целесообразность переброски трафика γ_2 является то, что к коммуникационным узлам обычно соединяются локальные компьютерные сети учреждений, компаний, фирм и т. д., которые в своем составе имеют сервера (файл-сервер, сервер баз данных, Web-сервер), хранящие большой объем информации, и к которым часто обращаются многочисленные пользователи сети. Перекачка этих ресурсов ко каналам коммуникационной сети ("внутренний трафик") на другие точки сети может привести к росту непредусмотренных внешними трафиками потоков в каналах сети, которые уже перегружены другими видами информации. Переброска такого рода трафиков на другие недогруженные каналы улучшить значение целевой функции. Поэтому целесообразным является переброска на новые пути потоков трафика γ_2 . В случае отсутствия в данном канале потоков IP - трафика, целесообразно

отклонить части речевого потока, требующего меньше времени непрерывной передачи, чем поток трафика видео.

С учетом вышеизложенного процедура отклонения части общего потока сводится к следующему:

$$\varphi_k^{(L,1)} = \lambda_k^{(L,1)} - \Delta^2 \lambda_k^{(L,1)}$$

$$\varphi_q^{(L,2)} = \lambda_q^{(L,2)} + \Delta^2 \lambda_q^{(L,2)},$$

где Δ^2 - параметр, характеризующий долю отклоняемой части потока, из одной пути в другой. Индекс 2 означает, что отклоняемая часть потока соответствует трафику данные, если отклоняется потоки других трафиков, то индекс соответственно меняется (для речи - 1, видео - 3). Производя аналогичные отклонения потоков в других каналах $\varphi_k^{(L,1)}$, $\varphi_q^{(L,2)}$ в конечном итоге получим вектор потоков в сети на данной итерации:

$$\vec{\varphi}^{(\varepsilon)} = (\varphi_1^{(\varepsilon)}, \varphi_2^{(\varepsilon)}, \dots, \varphi_M^{(\varepsilon)})$$

7) вычисляются значение коэффициента δ , по которому оценивается средняя задержка в сети при данном варианте распределения потоков по кратчайшим путям и значение $\rightarrow$ целевой функции $\vec{T}(\varphi(\varepsilon))$:

$$\delta(\varepsilon) = \sum_{i \in M} \eta_i \varphi_i^{(\varepsilon)}$$

8) Если $\beta(\varepsilon) - \delta(\varepsilon) < \nu$ (где $\nu > 0$ - заданный допуск), то задача минимизации $\vec{T}(\varphi(\varepsilon))$ считается решенной. Этот оператор проверяет, обеспечивает ли распределение потоков на данном этапе поиска:

$$\min \vec{T}(\varphi(\varepsilon))$$

т. е. вычисляет степень изменения задержки в каналах сети за счет уменьшения части потоков по кратчайшим путям L_{ij}^I и увеличения части потоков в кратчайших путях L_{ij}^{II} , так как к ним были направлены (отклонены) части потоков. При невыполнении условия осуществляется переход к следующему шагу итерации, т. е.

$\varepsilon = \varepsilon + 1$ и осуществляется переход к шагу 3.

Полученные результаты на основе предложенного алгоритма сравнивались с известным алгоритмом Клейнрока о девиации потоков[2].

Учитывая более высокий приоритет речевого трафика в первую очередь решается задача распределения речевого трафика. Далее, выполняется алгоритм распределения потоков данных при различной загруженности каналов передачей речевой информации. Некоторые результаты вычислительного эксперимента приведены на рис. 1 (а,б) и 2. Параметр α характеризует степень загрузки сети речевым трафиком.

Как видно из рисунков, рост речевого трафика заметно увеличивает задержку пакетов данных (от **0,15** до **0,55** при изменении нагрузки речевого трафика от $\alpha = 0$ до $\alpha = 0,65$ (рис.2)) .

Также следует отметить, что во всех исследованных случаях алгоритм распределения потоков «парным» способом в области средней и сильной загруженности каналов обеспечивает меньшее значение времени задержки в сравнении с традиционным алгоритмом, использующий метод девиации потоков.

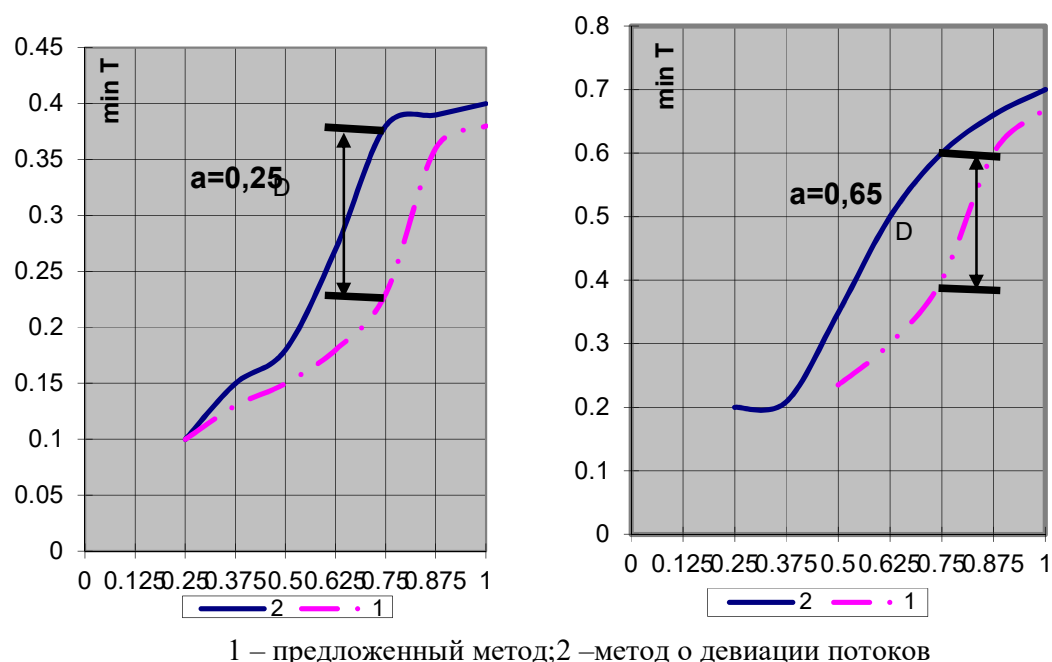


Рис. 1. Зависимость минимума суммарного среднего времени задержки данных от производительности каналов (ρ) при различной загруженности.

Данный алгоритм, рационально загружая каналы сети при загруженности их речевыми трафиками, не сильно увеличивает значение средней задержки при увеличении интенсивности потока данных. Например, $T_{cp} = 0.38$ при $\alpha = 0.25$, $T_{cp} = 0.6$ при $\alpha = 0.65$, $\Delta T_{cp} = 0.6 - 0.38 = 0.22$ при использовании метода о девиации потоков, аналогичный результат при предложенном алгоритме составляет $\Delta T = 0.4 - 0.23 = 0.17$.

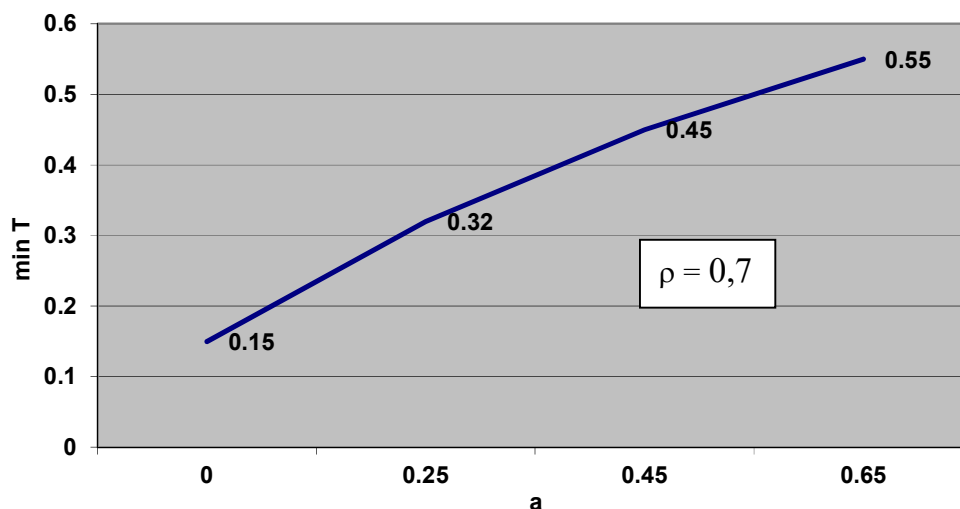


Рис .2. Характер изменения средней задержки данных при увеличении нагрузки речевого трафика.

Таким образом, предложенный алгоритм распределения потоков позволяет для каждой нитки потоков определить такие кратчайшие пути, при которых значение суммарной средней задержки информации в сети является меньшей в сравнении с алгоритмом девиации потоков.

Заключение

Всё выше изложенное позволяет сделать вывод о том, что, предложенная потоковая модель и алгоритм поиска кратчайших путей позволяют рационально управлять распределением разнородных потоков в условиях использования в транспортной части мультисервисной сети идеологию программно-конфигурируемой сети. В данном случае принятие решения по распределению потоков принимается на контроллере, а результаты передаются на соответствующие коммутаторы по специальным защищённым каналам. Выполнение этих операций в случае загрузки сети должны осуществляться за очень короткое время и использование разработанных алгоритмов позволят вычислить искомое решение за очень короткое время.

Литература

1. Нишанбаев Т.Н., Махмудов С. О. Формализация задачи распределения потоков разнородного трафика в программно-конфигурируемой сети. Информационные технологии, решения. 4(48)/2018. Ташкент 2018.
2. К. А. Литвинов, И. И. Пасечников Подходы к решению задачи маршрутизации в современных телекоммуникационных системах. Вестник ТГУ, т. 18, вып. 1, 2013.