

References

1. Sheluhin O.I., Tenjakshev A.M., Osin A.V. *Fractal Processes in Telecommunications*. Moscow: Radiotekhnika, 2003, 479 p. (In Russian.)
2. Linets G.I. et al. Simulation model of asynchronous transformation of self-similar traffic in switching nodes using a queue. *Infokommunikacionnye tehnologii*, 2019, vol. 17, no. 3, pp. 293–303. (In Russian.)
3. Linets G.I., Govorova S.V., Voronkin R.A. A program for generating a dataset for studying the statistical characteristics of a self-similar traffic transformation model. *Svidetel'stvo o gos. registratsii programmy dlja EVM № 2019619275*, 2019. (In Russian.)
4. Bronshtein I.N. et al. *Handbook of Mathematics*. 6th ed. Berlin: Springer, 2015, 1151 p. DOI: <https://doi.org/10.1007/978-3-662-46221-8>.
5. *Basic principles of machine learning using linear regression as an example*. URL: <https://habr.com/ru/company/ods/blog/322076> (accessed 01.04.2020).
6. Koel'о L.P., Richard V. *Building Machine Learning Systems in Python*. Trans. from Eng. by A.A. Slinkin. Moscow: DMK Press, 2016, 302 p. (In Russian.)
7. *Isotonic Regression*. URL: <https://scikit-learn.org/stable/modules/isotonic.html> (accessed 01.04.2020).
8. Sharden B., Massaron L., Bosketti A. *Large-Scale Machine Learning With Python*. Trans. from Eng. by A.V. Logunov. Moscow: DMK Press, 2018, 358 p. (In Russian.)
9. Rashka S. *Python and Machine Learning*. Trans. from Eng. by A.V. Logunov. Moscow: DMK Press, 2017, 418 p. (In Russian.)
10. *Support Vector Regression (SVR) using linear and non-linear kernels*. URL: https://scikit-learn.org/stable/auto_examples/svm/plot_svm_regression.html?highlight=svr (accessed 01.04.2020).
11. Westling T., Gilbert P., Carone M. Causal isotonic regression. *arXiv:1810.03269*, 2019. URL: <http://arxiv.org/abs/1810.03269> (accessed 23.05.2020).

Received 25.06.2020

УДК 004.942

АНАЛИТИЧЕСКАЯ МОДЕЛЬ КОММУТАТОРА КОРПОРАТИВНОЙ ПРОГРАММНО-УПРАВЛЯЕМОЙ СЕТИ SDN

Мочалов В.П., Линец Г.И., Братченко Н.Ю.
Северо-Кавказский федеральный университет, Ставрополь, РФ
E-mail: n.b.20062@yandex.ru

Реализация практически неограниченных возможностей программно-управляемой сети вызывает необходимость дополнительного исследования ее инфраструктурного уровня, оценки телекоммуникационной составляющей. Поэтому целью исследования является разработка аналитической модели для анализа основных показателей качества сетевых коммутаторов и каналов связи. Для формирования математической модели коммутатора программно-управляемой сети, функционирующей в стационарном режиме и обслуживающей запросы случайного объема, предлагается использовать теорию системы массового обслуживания, а также аппарат полумарковских процессов. Модель коммутатора формально представлена в виде открытой сети массового обслуживания с пуассоновским потоком поступающих пакетов, отказами и блокировками. Первый этап обслуживания запросов включает в себя процессы записи сетевых пакетов в многолинейную буферную память коммутатора. В следующем центре осуществляется проверка соответствия адреса поступившего пакета записям в таблицах потоков. Третий однолинейный центр, реализующий процесс передачи выходящих из коммутатора пакетов, может быть описан системой с независимыми однолинейными системами массового обслуживания с n -местными буферами. Получены зависимости времени ожидания обслуживания и времени обслуживания поступающих сетевых пакетов от нагрузки, а также выражения для определения объема буферной памяти коммутатора при допустимой вероятности потери сообщений, показатели качества его передающей части.

Ключевые слова: таблицы потоков, OpenFlow коммутатор, модели массового обслуживания, контроллер, вероятностно-временные характеристики

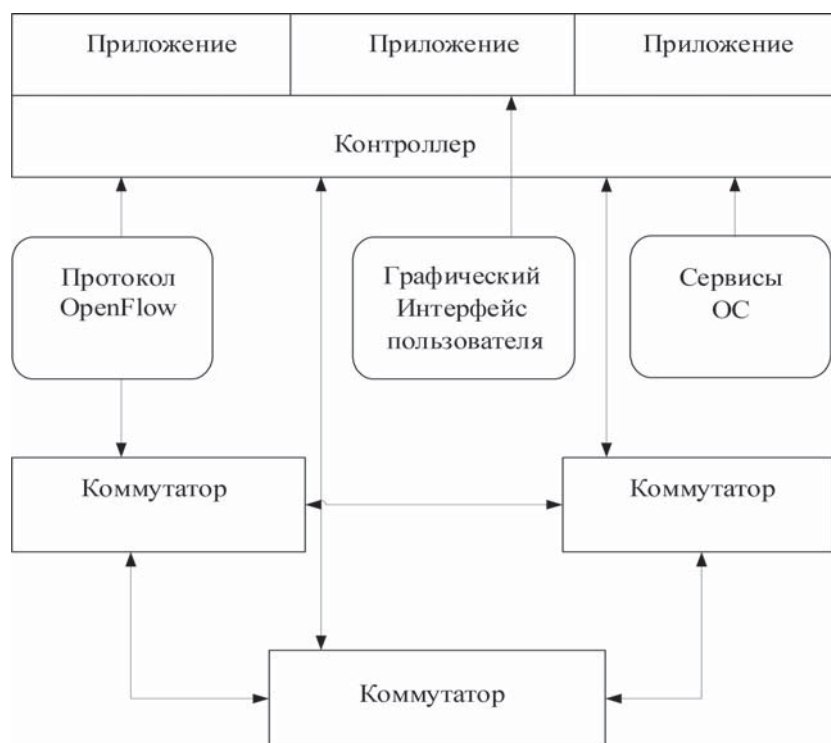


Рисунок 1. Типовой вариант архитектуры SDN-сети

Введение

Проблемы и ограничения современных компьютерных сетей [1–4] привели к разработке и построению программно-конфигурируемых сетей (Software Defined Networks, SDN). Основные подходы концепции SDN представлены в рекомендациях Международного союза электросвязи – МСЭ-T серии Y.3000, предполагающие разделение процессов передачи и управления данными, логически централизованный уровень управления, применение унифицированного интерфейса OpenFlow, а также виртуализацию физических ресурсов. В отличие от традиционных методов коммутации и маршрутизации, базирующихся на IP- и MAC-адресах, протокол OpenFlow способен реализовать более сорока критериев выбора маршрутов передачи сетевых пакетов. Типовой вариант архитектуры SDN-сети [5–6] приведен на рисунке 1.

Контроллер SDN, сетевая операционная система периодически обновляют свои внутренние данные о состоянии элементов сети, топологии, маршрутах передачи данных, потоках, ресурсах. При поступлении запроса на обслуживание контроллер производит обработку первого сетевого пакета соответствующего потока и устанавливает правила управления и пересылки для всех последующих, т. е. управление данными происходит на уровне потоков. Первый пакет каждого нового потока пересылается контроллеру, который

создает в коммутаторе соответствующую запись в таблице передачи. Каждый коммутатор заполняет свои таблицы адресации только по данным контроллера. Если считать, что очередь обслуживаемых сетевых пакетов на входе контроллера отсутствует, можно описать его экспоненциальной системой массового обслуживания (СМО) с бесконечным числом обслуживаемых приборов, а модель коммутатора SDN, включающая фазы приема и обработки заголовков пакетов, управления потоками, внесения изменений в таблицы адресации, может быть представлена многофазной системой массового обслуживания (СМО) с неограниченной памятью. Функционирование сети в стационарном режиме, а также ее переходы в различные состояния могут быть описаны цепью Маркова [7–9].

Порт управления коммутатора связан с процессором контроллера защищенным каналом обмена OpenFlow сообщениями. При этом может применяться как специальная сеть управления, так и существующая транспортная сеть. Каждый коммутатор включает в себя цепочку последовательно соединенных таблиц адресации потоков пакетов (см. рисунок 2), которые содержат алгоритмы и инструкции перераспределения пакетов: пересылки на следующую по номеру таблицу, на один из выходных портов или на управляющий вход контроллера.

При поступлении входных пакетов проверяется соответствие адреса поступившего пакета

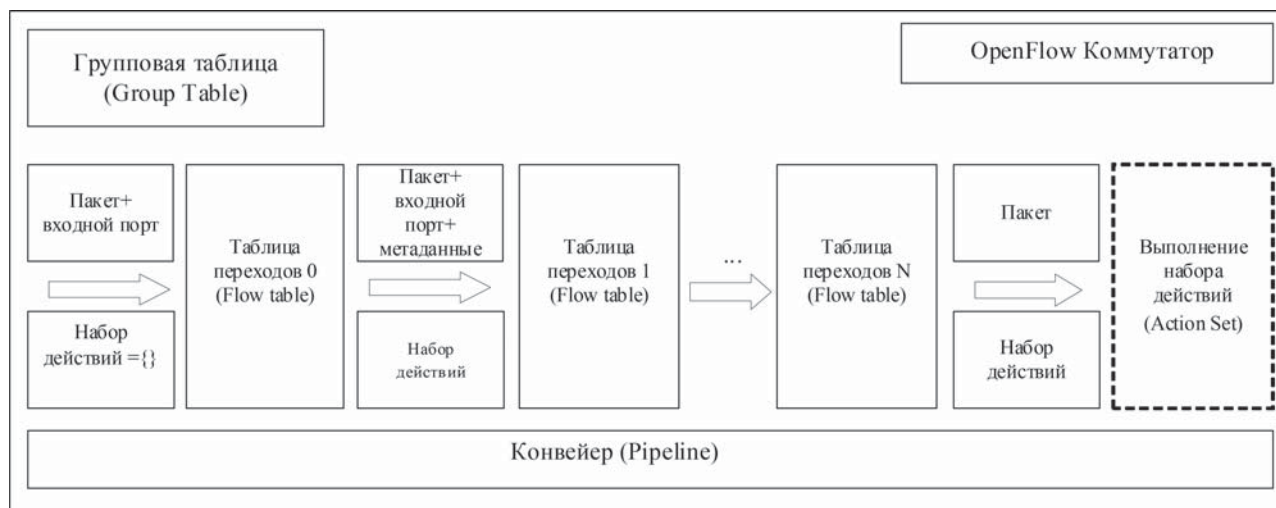


Рисунок 2. Формальная модель распределенной системы управления телекоммуникациями



Рисунок 3. Структура набора записей коммутатора

записям в таблицах потоков. Если соответствия адресов не установлено, то пакет отправляется контроллеру, который определяет правила его обработки и устанавливает их в таблицах адресации коммутатора. При этом контроллер, в зависимости от состояния сети (топологии, нагрузки, тупиков и блокировок потоков пакетов), может изменять содержимое таблиц адресации, а также сопоставляет информацию о состоянии элементов.

Важнейшими задачами разработки и исследования таких сетей являются определение архитектуры, выбора параметров сетевых элементов, совершенствование протоколов управления и обмена данными, обеспечивающих требуемое качество обслуживания. Решение данных задач возможно путем разработки моделей функционирования соответствующего оборудования,

определения и исследования его основных вероятностно-временных характеристик.

Методология исследования

Каждый коммутатор содержит набор записей (flow entries) о потоках, которые включают в себя поля проверки (match fields), счетчики (counters) и указатели определяемых действий (инструкции), связанных с предварительной стандартной обработкой и пересылкой пакетов. Структура набора записей коммутатора приведена на рисунке 3.

Операции над потоками пакетов можно разделить на этапы: прием коммутатором пакетов от абонентов, проверка наличия записей о поступивших потоках пакетов в таблицах адресации, пересылка запросов контроллеру, принимающему решение по обработке пакетов. Вариант архитектуры коммутатора представлен на рисунке 4.

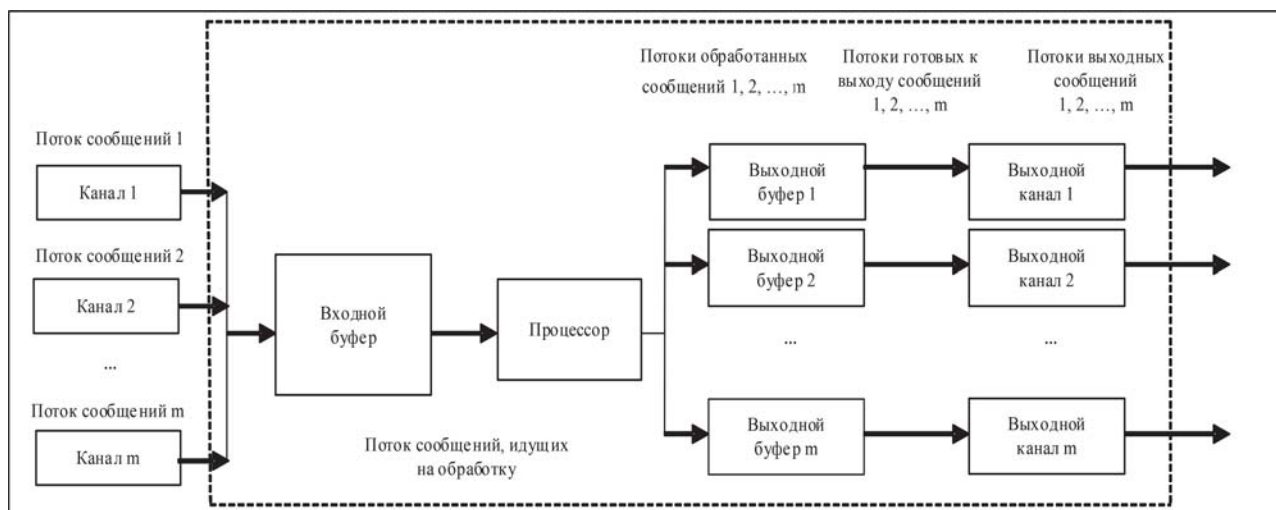


Рисунок 4. Вариант архитектуры коммутатора

Обработка пакетов включает в себя следующие действия [7; 10–11].

1. Пакет вновь формируемого потока со скоростью канала связи поступает на входной порт коммутатора и размещается в его буферной памяти.

2. Формирование потоков пакетов. Проверка соответствия адреса поступившего пакета записям в таблицах потоков. Если соответствие выявлено, то выполняется шаг 5.

3. Если соответствие не выявлено, пакет отправляется контроллеру SDN.

4. Согласно алгоритму маршрутизации контроллер SDN добавляет соответствующую запись в коммутатор и остальные коммутаторы по тракту передачи данного потока.

5. Ожидание освобождения канала в направлении исходящего порта. Инкапсуляция пакета и передача по заданному адресу.

Каждый фрагмент SDN-сети содержит несколько коммутаторов, связанных между собой высокоскоростными дуплексными каналами связи. Считаем, что используется асинхронное уплотнение каналов, связывающих абонентов с коммутатором и не оказывающее значительного влияния на общую оценку производительности сети. При этом, если пакет сразу после поступления отправляется на выходной порт высокоскоростного канала коммутатора, считаем, что время его обслуживания равно нулю; если пакет остается в памяти коммутатора, а на контроллер отправляется его заголовок, считаем, что время обслуживания описано произвольной функцией распределения.

В реальных сетевых элементах объем памяти всегда ограничен, поэтому при превышении установленного размера буфера велика вероятность потери пакетов. Буферная память является

общим разделяемым ресурсом для всех каналов связи. Оценить требуемый объем памяти можно, исходя из заданной вероятности отказов входящим в узел пакетам. Для приближенных оценок объема памяти коммутатора будем использовать зависимости, связывающие объем требований, время обслуживания сообщения и вероятность потерь. При этом число обслуживающих приборов равно числу мест накопителя m , и очередь отсутствует. Память может быть представлена совокупностью m независимых однолинейных СМО с отказами, блокировкой и дисциплиной обслуживания, распределенной по экспоненциальному закону.

Число обслуживающих элементов такого многолинейного центра равно количеству поступивших на вход коммутатора потоков пакетов. Применяемый в SDN протокол OF-CONFIG допускает проведение распределения ресурсов коммутатора, формируя несколько виртуальных коммутаторов из одного физического, распределяя при этом физическую память между потоками. Объемы сообщений распределены по экспоненциальному закону $L(x) = 1 - e^{-fx}$. Сообщения поступают на прием по m каналам с интенсивностями a_i ($i = 1, m$). Поток принятых сообщений простейший с интенсивностью $\lambda = \sum_{i=1}^m a_i$.

В [12] получены явные выражения для определения объема памяти при допустимой вероятности отказа в приеме сообщения. Для СМО $M|G|1|\infty$ найдено преобразование Лапласа – Стильгеса (ПЛС) функции распределения (ФР) объема обслуживаемого сообщения $R(x)$:

$$l_i(s) = 1 - \frac{a_i}{g_i} \left[\frac{1}{f} + \frac{f}{(s+f)^2} \right], \quad (1)$$

где i – номер приемного канала; a_i – интенсивность входного потока ($i = \overline{1, m}$); $g_i > 0$ – время приема сообщения по i -му каналу; f – параметр ФР объема сообщения.

Отсюда первые два момента объема требования будут равны:

$$l_{1i} = \frac{2a_i}{g_i f^2} = \frac{2\rho_i}{f}, \quad l_{2i} = \frac{2a_i}{g_i f^3} = \frac{6\rho_i}{f^2}, \quad (2)$$

где ρ_i – нагрузка i -го канала;

$$(l_{2i} - l_{1i}^2) = \frac{2\rho_i}{f^2} (3 - 2\rho_i).$$

Тогда ПЛС стационарного суммарного объема сообщений будет равно

$$\delta(s) = \prod_{i=1}^m \left\{ 1 - \frac{a_i}{g_i} \left[\frac{1}{f} + \frac{f}{(s+f)^2} \right] \right\}. \quad (3)$$

Отсюда следует, что моменты первого и второго порядка суммарного объема сообщений будут равны:

$$\delta_1 = \frac{2}{f} \sum_{i=1}^m \rho_i, \quad (\delta_1 - \delta_1^2) = \frac{2}{f^2} \sum_{i=1}^m \rho_i (3 - 2\rho_i). \quad (4)$$

Расчет объема памяти V проводится с учетом вероятности потерь сообщения:

$$p_{\pi} = 1 - R(V), \quad (5)$$

где

$$R(V) = \int_0^V D(V-x) dL(x), \quad L(x) = 1 - e^{-fx}$$

– ФР объема сообщения; $D(x) = p(\delta < x)$ – ФР стационарного суммарного объема сообщений δ . Первые два момента ФР $R(x)$ равны:

$$r_1 = \delta_1 + \varphi_1, \quad r_2 = \varphi_2 + 2\varphi_1\delta_1 + \delta_2, \quad (6)$$

где $\varphi_1 = 1/f$ – средний объем сообщения; δ_1, δ_2 – моменты суммарного объема сообщений. Для решения данного уравнения можно воспользоваться выражением

$$D(V) = p_0 + (1 - p_0) \frac{\gamma(p, gx)}{\Gamma(p)}, \quad (7)$$

где p_0 – вероятность непоступления требований;

$$p = \frac{r_1^2}{r_2 - r_1^2}, \quad g = \frac{r_1}{r_2 - r_1^2}. \quad (8)$$

Используя расчетные таблицы математической статистики, можно получить требуемый объем памяти V коммутатора при заданной вероятности потери сообщения. Результаты моделирования представлены в таблице.

Следующий этап обработки запросов – анализ служебной части пакетов и формирование потоков сетевых пакетов. С каждым пакетом осуществляются следующие действия [13; 14].

Таблица. Результаты моделирования

V	P_{π}		
	$\rho = 0.4$	$\rho = 0.5$	$\rho = 0.7$
20	0.093	0.171	0.279
30	0.017	0.083	0.073
40	0.0063	0.0038	0.017

1. Идентификация пакета и реализация процедуры определения его принадлежности к потоку.

2. Осуществление поиска по всем типам потоков.

3. Если поток не будет найден, то соответствующий пакет передается контроллеру и формируется новый поток с информацией о текущем пакете.

Исходя из этого, считаем, что требование обслуживается мгновенно при идентификации пакета, в противном случае, то есть при отсутствии идентификации, описывается экспоненциальным распределением с параметром p . При этом ФР времени обслуживания входного потока пакетов будет иметь вид $B(t) = p + (1-p)(1 - e^{-pt})$, а ее ПЛС есть

$$\beta(q) = p + \frac{(1-p)p}{p+q} = \frac{p(1+q)}{p+q}, \quad (9)$$

откуда получим среднее значение времени обслуживания

$$\beta_1 = -\beta'(0) = \frac{1-p}{p}. \quad (10)$$

Если нагрузка системы определяется как $\rho = a\beta_1 = a(1-p)/p$, то ПЛС времени ожидания обслуживания определится как

$$W(q) = \frac{(1-p)(p+q)}{p+q-a(1-p)} = \frac{(1-p)(p+q)}{q+p(1-\rho)}. \quad (11)$$

Среднее значение времени ожидания равно $W_1 = -W'(0) = \frac{p}{p(1-\rho)}$. Если изображение имеет

вид рациональной дроби $\frac{A_n(p)}{B_n(p)}$, где P_1, P_2, \dots, P_n – корни кратности r_1, r_2, \dots, r_n так, что

$$r_1 + r_2 + \dots + r_n = m, \quad (12)$$

$$B_m(p) = \beta_0 (p - P_1)^{r_1} (p - P_2)^{r_2} \dots (p - P_n)^{r_n}, \quad (13)$$

то оригинал может быть найден по формуле

$$f(t) = \sum \operatorname{Re} s \left[\frac{A_n(p) e^{pt}}{B_m(p)} \right]. \quad (14)$$

Если корни знаменателя P_1, P_2, \dots, P_m являются простыми, то

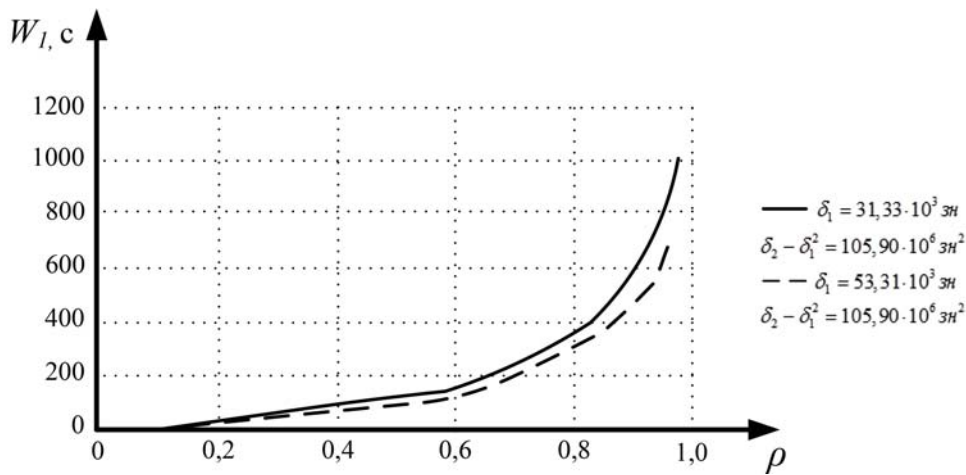


Рисунок 5. Зависимость среднего значения времени ожидания от нагрузки

$$f(t) = \sum \frac{A_n(P_k)}{B_m(P_k)} e^{pk^t}. \quad (15)$$

Тогда обращение ПЛС функции $W(q)$ будет определяться соотношением

$$W(t) = \sum \text{Re} s \left[\frac{(1-\rho)(\rho+q)}{q(q+p(1-\rho))} e^{qt} \right]. \quad (16)$$

Функция распределения случайной величины (СВ) U имеет вид

$$U(y) = p \{U < t\} = \int_0^t W(t-u) dB(u) = \int_0^t e^{-\mu u} W(t-u) dU, \quad (17)$$

где $B(t) = 1 - e^{-\mu t}$. Для случая

$$p \{W > 0\} = 1 - W(0) = \frac{(np)^n p_0}{n!(1-\rho)}$$

среднее значение стационарного времени ожидания представляет собой

$$W_1 = EW = \int_0^\infty dW(t) = \frac{n^{n-2} \rho^n p_0}{\mu(1-\rho)^2 (n-1)!}. \quad (18)$$

Среднее значение стационарного времени обслуживания будет равно

$$U_1 = EU = \int_0^\infty t dU(t) = \beta_1 + T_1 = \frac{1}{\mu} + \frac{n^{n-2} \rho^n p_0}{\mu(1-\rho)^2 (n-1)!}, \quad (19)$$

где EW, EU – математические ожидания СВ W и U .

Для упрощения расчетов полученные случайные величины были аппроксимированы ФР:

$$Z(x) = p_0 + (1-p_0) \frac{\gamma(p, gx)}{\Gamma(p)},$$

где

$$\gamma(p, gx) = \int_0^{gx} t^{p-1} e^{-t} dt, \quad \Gamma(p) = \gamma(p, \infty),$$

$$p = \frac{\delta_1^2}{(1-p_0)\delta_2 - \delta_1^2}, \quad g = \frac{(1-p_0)\delta_1}{(1-p_0)\delta_2 - \delta_1^2}, \quad (20)$$

где p_0 – стационарная вероятность отсутствия требований.

Представленные зависимости позволяют определить зависимости динамических характеристик коммутатора от нагрузки для двух объемов сообщений: $\delta_1 = 31,33 \cdot 10^3$ зн, $\delta_1 = 53,31 \cdot 10^3$ зн и дисперсии $\delta_2 - \delta_1^2 = 105,90 \cdot 10^6$ зн². На этапе передачи данных коммутатор исполняет инструкции, связанные с данным пакетом, добавляет поступившую от контроллера SDN информацию о маршрутизации и передает сформированный поток пакетов на исходящий порт для отправки в канал связи [10; 15]. Методы передачи сетевых пакетов, определяющие как скорость передачи данных по каналам связи, так и время хранения копий сообщений в буферной памяти, существенно зависят от применяемых линейных протоколов, особенностей этапов квитирования и time-out [17]. Будем считать, что копии передаваемых пакетов после окончания их передачи хранятся в буферной памяти в течение некоторого времени time-out до получения подтверждения о доставке пакетов. Отсутствие подтверждения вызывает необходимость повторной передачи пакета. Это устраняет возможные ошибки и потери пакетов из-за нехватки буферной памяти в принимающем коммутаторе. Очевидно, что фаза передачи выходящих из коммутатора сетевых пакетов может быть описана системой $M | M | m | n$, т. е. m независимыми однолинейными СМО с n -местными буферами, простейшим входящим потоком и

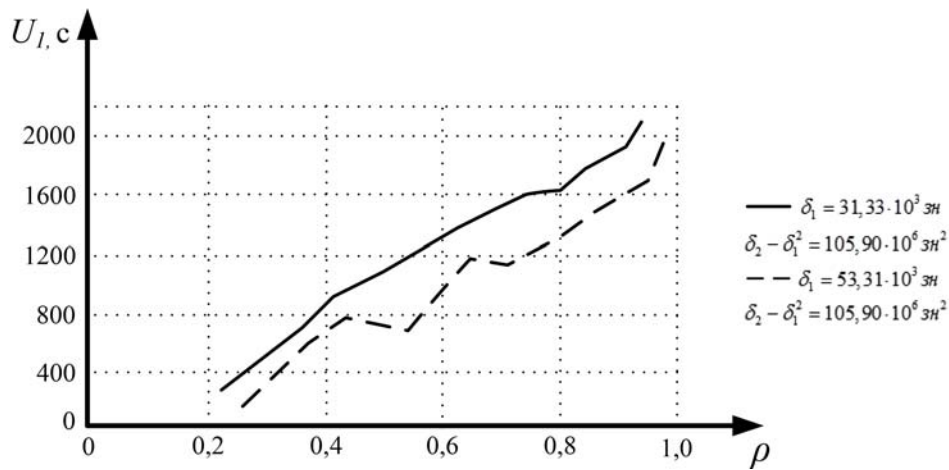


Рисунок 6. Зависимость среднего значения времени обслуживания от нагрузки

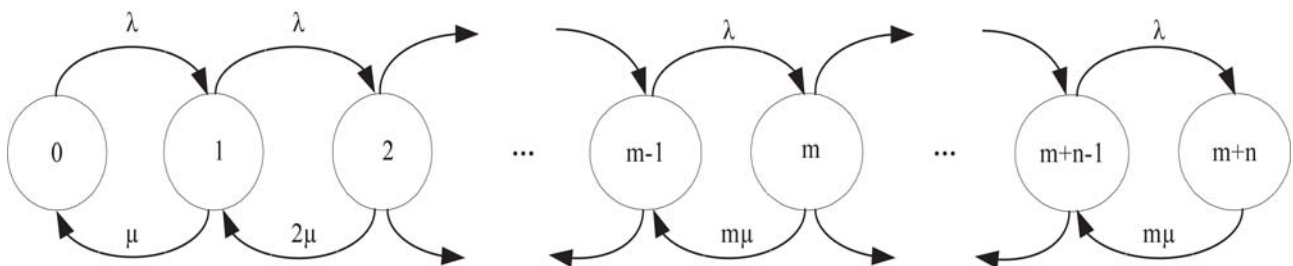


Рисунок 7. Граф состояний и переходов СМО типа $M | M | m | n$

показательным распределением времени обслуживания. Диаграмма интенсивностей переходов СМО типа $M | M | m | n$ приведена на рисунке 7.

Известно [11; 16], что вероятность нахождения системы $M | M | m | n$ в состоянии p_k равна

$$p_k = \frac{\frac{\rho^k}{k!}}{\sum_{k=0}^m \frac{\rho^k}{k!} + \frac{\rho^{m+1}}{m \cdot m!} - \frac{1 - (\frac{\rho}{m})^n}{1 - \frac{\rho}{m}}}, \quad (21)$$

где $0 \leq k \leq m$, $\rho = \frac{\lambda}{\mu}$ – нагрузка.

Соответственно, вероятность состояния p_{m+s} равна

$$p_{m+s} = \frac{\frac{\rho^m}{m!} (\frac{\rho}{m})^s}{\sum_{k=0}^m \frac{\rho^k}{k!} + \frac{\rho^{m+1}}{m \cdot m!} - \frac{1 - (\frac{\rho}{m})^n}{1 - \frac{\rho}{m}}}, \quad (22)$$

где $1 \leq s \leq n$.

Считаем, что из-за отсутствия свободного буфера в принимающем коммутаторе переданный пакет с вероятностью p может быть потерян. Тогда интенсивность выходного потока определится

как $y = \lambda - \lambda p_{m+n} = \lambda \sum_{i=0}^{m+n-1} p_i$. Соответствующие вероятности состояний рассматриваемой системы есть

$$p_0 = \frac{\rho}{\sum_{k=0}^m \frac{\rho^k}{k!} + \frac{\rho^{m+1}}{m \cdot m!} - \frac{1 - (\frac{\rho}{m})^n}{1 - \frac{\rho}{m}}}, \quad (23)$$

$$p_{m+n} = \frac{\frac{\rho^m}{m!} (\frac{\rho}{m})^n}{\sum_{k=0}^m \frac{\rho^k}{k!} + \frac{\rho^{m+1}}{m \cdot m!} - \frac{1 - (\frac{\rho}{m})^n}{1 - \frac{\rho}{m}}} = \quad (24)$$

$$= \frac{\rho^m}{m!} \left(\frac{\rho}{m}\right)^n p_0.$$

С учетом потерянных пакетов интенсивность выходного потока будет равна

$$y = \lambda(1 - p_{m+n}) = \lambda \left(1 - \frac{\rho^m}{m!} \left(\frac{\rho}{m}\right)^n p_0 \right) = \quad (25)$$

$$= \lambda \frac{\rho - \frac{\rho^m}{m!} \left(\frac{\rho}{m}\right)^n}{\rho} p_0.$$

Кроме данных характеристик оценить показатели качества передающей части коммутатора можно такими параметрами, как:

– время пребывания пакетов в системе

$$V = \frac{N}{m\mu(1-p_0)} = \frac{\sum_{k=0}^{m+n} kp_k}{m\mu(1-p_0)}; \quad (26)$$

– время ожидания пакетов в очереди

$$W = \frac{N_0}{m\mu(1-p_0)} = \frac{\sum_{k=m+1}^{m+n} (k-m)p_k}{m\mu(1-p_0)}; \quad (27)$$

– время обслуживания

$$\begin{aligned} T_{об} &= V - W = \\ &= \frac{\sum_{k=0}^{m+n} kp_k}{m\mu(1-p_0)} - \frac{\sum_{k=m+1}^{m+n} (k-m)p_k}{m\mu(1-p_0)} = \\ &= \frac{1}{m\mu} + \frac{(m-1)\sum_{k=m}^{m+n} p_k + \sum_{k=2}^{m-1} (k-1)p_k}{m\mu(1-p_0)}. \end{aligned} \quad (28)$$

Представленные оценки процесса функционирования передающей части коммутатора зависят от времени. Известно, что при рассмотрении случайных стационарных процессов, когда характеристики системы не изменяются с течением времени, систему можно рассматривать в установленном режиме как стационарную с параметрами, не зависящими от времени.

Результаты и дискуссия

В работе установлена связь между задержкой сообщений в сетевом коммутаторе и объемом его буферной памяти. Недостатками данной модели является предположение о дискретном характере обработки сообщений, а также возможность проведения исследований только на уровне частных распределений СВ, что усложняет ее применение в реальных информационных системах.

В статье представлена аналитическая модель для исследования показателей качества коммутатора программно-управляемой сети SDN, формально представленная в виде сети массового обслуживания (СеМО) с пуассоновским потоком поступающих пакетов, экспоненциальным обслуживанием, отказами и блокировками. Известно, что при проведении исследования данной СеМО можно независимо исследовать ее составляющие узлы, представляющие СМО типа.

Поэтому операции, выполняемые коммутатором, условно разделены на три группы: приема и размещения сетевых пакетов в буферной памяти, обработки и формирования потоков пакетов, передачи сетевых пакетов в каналы связи. При этом используется аппарат производящих функций и преобразований Лапласа – Стилтеса. Первый этап обслуживания запросов включает в себя процессы записи пакетов в многолинейную буферную память коммутатора. Число однолинейных элементов памяти здесь соответствует числу мест буферной памяти, очередь отсутствует. В следующем центре осуществляется проверка соответствия адресов поступивших пакетов записям в таблицах потоков.

Если соответствие выявлено, то происходит формирование потока пакетов, если соответствие не выявлено, пакет отправляется контроллеру SDN. Третий однолинейный центр, реализующий процесс передачи выходящих из коммутатора пакетов, описан системой, то есть m независимыми однолинейными СМО с n -местными буферами. Учитывая, что в данном случае зависимость между этапами обработки не является значимой, рассматриваемые этапы обработки пакетов можно считать независимыми, поэтому показатели качества данной СеМО получены с учетом параметров однофазных СМО.

Результаты исследования могут быть использованы при проектировании и эксплуатации сетей связи, реализующих концепцию SDN [18; 19].

Заключение

Рассмотрены основные подходы к построению и анализу процессно-ориентированных распределенных систем управления телекоммуникациями на основе открытой цифровой архитектуры (Open Digital Architecture) концепции TM Forum Framework, при которых функции системы управления реализуются совокупностью связанных между собой процессорных модулей с установленными на них программными компонентами, каждая из которых выполняет одну или несколько бизнес-задач, являющихся составными элементами бизнес-процесса, и могут служить основой для разработки более гибких систем управления телекоммуникациями.

Разработана аналитическая модель для исследования показателей качества коммутатора сети SDN, представленная в виде СеМО с пуассоновским потоком поступающих пакетов, экспоненциальным обслуживанием, отказами и блокировками. Путем использования формального аппарата полумарковских процессов, преобразо-

ваний Лапласа и производящих функций получены оценки показателей качества обработки запросов случайного объема.

Финансирование

Данные исследования выполнены при поддержке Российского фонда фундаментальных исследований (РФФИ), грант № 19-07-00856/20.

Литература

1. Bholebawa I.Z., Dalal U.D. Performance analysis of SDN / OpenFlow controllers: POX versus floodlight // *Wireless Personal Communications*. 2008. Vol. 98. № 2. P. 1679–1699. DOI: <https://doi.org/10.1007/s11277-017-4939-z>.
2. Queueing analysis of software defined network with realistic OpenFlow-based switch model / Y. Goto [et al.] // *Computer Networks*. 2019. Vol. 164. P. 106892. DOI: <https://doi.org/10.1016/j.comnet.2019.106892>.
3. Fundamentals of queueing theory. 1st ed. / D. Gross [et al.]. Hoboken: John Wiley & Sons, 2008. 509 p. DOI: <https://doi.org/10.1002/9781118625651>.
4. OSS/BSS Framework based on NGOSS / L. Hanhua [et al.] // 2009 International Forum on Computer Science-Technology and Applications. 2009. P. 466–471. DOI: <https://doi.org/10.1109/IFCSTA.2009.120>.
5. Software-defined networking: a comprehensive survey / D. Kreutz [et al.] // *Proceedings of the IEEE*. 2015. Vol. 103. № 1. P. 14–76. DOI: <https://doi.org/10.1109/JPROC.2014.2371999>.
6. Lechler T., Taylor B.J., Klingenberg B. The telecommunications carriers' dilemma: Innovation vs. Network Operation // *PICMET '07 – 2007 Portland International Conference on Management of Engineering & Technology*. 2007. P. 2940–2947. DOI: <https://doi.org/10.1109/PICMET.2007.4349638>.
7. Lee E.A. The problem with threads // *Computer*. 2006. Vol. 39. № 5. P. 33–42. DOI: <https://doi.org/10.1109/MC.2006.180>.
8. Li T., Chen J., Fu H. Application scenarios based on SDN: an overview // *Journal of Physics: Conference Series*. 2019. Vol. 1187. № 5. P. 052067. DOI: <https://doi.org/10.1088/1742-6596/1187/5/052067>.
9. Тарасов В.Н., Каргашевский И.В., Малахов С.В. Теоретическое и экспериментальное исследование задержки в программно-конфигурируемых сетях // *Инфокоммуникационные технологии*. 2015. Т. 13. № 4. С. 409–413. DOI: <https://doi.org/10.18469/ikt.2015.13.4.08>.
10. Mochalov V.P., Bratchenko N.Yu., Yakovlev S.V. Analytical model of integration system for program components of distributed object applications // 2018 International Russian Automation Conference (RusAutoCon). 2018. P. 1–4. DOI: <https://doi.org/10.1109/RUSAUTOCON.2018.8501806>.
11. A survey of software-defined networking: past, present, and future of programmable networks / B.A.A. Nunes [et al.] // *IEEE Communications Surveys & Tutorials*. 2014. Vol. 16. № 3. P. 1617–1634. DOI: <https://doi.org/10.1109/SURV.2014.012214.00180>.
12. Olszewsk, M., Ansel J., Amarasinghe S. Kendo: Efficient deterministic multithreading in software // *ACM SIGPLAN Notices*. 2009. Vol. 44. № 3. P. 97. DOI: <https://doi.org/10.1145/1508284.1508256>.
13. Rao N.S. Performance comparison of SDN solutions for switching dedicated long-haul connections // *The International Symposium on Advances in Software Defined Networking and Network Functions Virtualization*. 2016. URL: <https://www.osti.gov/biblio/1267045-performance-comparison-sdn-solutions-switching-dedicated-long-haul-connections>.
14. Модель функционирования телекоммуникационного оборудования программно-конфигурируемых сетей / К.Е. Самуйлов [и др.] // *Современные информационные технологии и ИТ-образование*. 2018. Т. 14. № 1. С. 13–26. DOI: <https://doi.org/10.25559/SITITO.14.201801.013-026>.
15. Simoes J., Wahle S. The future of services in next generation networks // *IEEE Potentials*. 2011. Vol. 30. № 1. P. 24–29. DOI: <https://doi.org/10.1109/MPOT.2010.939761>.
16. Analytical modelling of software and hardware switches with internal buffer in software-defined networks / D. Singh [et al.] // *Journal of Network and Computer Applications*. 2019. Vol. 136. P. 22–37. DOI: <https://doi.org/10.1016/j.jnca.2019.03.006>.
17. Sutter H., Larus J. Software and the concurrency revolution // *Queue*. 2005. Vol. 3. № 7. P. 54. DOI: <https://doi.org/10.1145/1095408.1095421>.
18. Priya A.V., Radhika N. Performance comparison of SDN OpenFlow controllers // *International Journal of Computer Aided Engineering and Technology*. 2019. Vol. 11. № 4/5. P. 467. DOI: <https://doi.org/10.1504/IJCAET.2019.10020284>.
19. Y.3300: Framework of software-defined networking. URL: <https://www.itu.int/rec/T-REC-Y.3300-201406-I/en> (дата обращения: 01.06.2020).

Получено 01.07.2020

Мочалов Валерий Петрович, д.т.н., профессор кафедры инфокоммуникаций (ИК) Северо-Кавказского федерального университета (СКФУ). 355028, Российская Федерация, Ставропольский край, г. Ставрополь, пр. Кулакова, 2 (корп. 9). Тел. +7 865 295-69-97. E-mail: mochalov.valery2015@yandex.ru

Линец Геннадий Иванович, д.т.н., заведующий кафедрой ИК СКФУ. 355028, Российская Федерация, Ставропольский край, г. Ставрополь, пр. Кулакова, 2 (корп. 9). Тел. +7 865 295-69-97. E-mail: kbytw@mail.ru

Братченко Наталья Юрьевна, к.ф.-м.н., доцент кафедры ИК СКФУ. 355028, Российская Федерация, Ставропольский край, г. Ставрополь, пр. Кулакова, 2 (корп. 9). Тел. +7 865 295-69-97. E-mail: n.b.20062@yandex.ru

ANALYTICAL MODEL OF CORPORATE SOFTWARE-CONTROLLED SDN NETWORK SWITCH

Mochalov V.P., Linets G.I., Bratchenko N.Yu.

North-Caucasus Federal University, Stavropol, Russian Federation

E-mail: n.b.20062@yandex.com

Implementing the almost limitless possibilities of a software-controlled network requires additional study of its infrastructure level and assessment of the telecommunications aspect. Therefore the aim of this study is to develop an analytical model for analyzing the main quality indicators of network switches and communication channels. To form a mathematical model of the switch for software-controlled network operating in a stationary mode and serving the requests of variable volume, it is proposed to use the queuing system theory, as well as the apparatus of semi-Markov processes. The switch model is formally presented as an open queuing network with Poisson flow of incoming packets, refusals and blockings. The first stage of servicing requests includes the processes of writing network packets to the switch multiline buffer memory. In the next center the correspondence of incoming packet address to the records in the flow tables is being checked. The third single-line center implementing the transmission process of the packets leaving the switch can be described by a system with independent single-line queuing systems with n-place buffers. Dependencies of queuing time and service time of incoming network packets on load are obtained, as well as equations for finding the volume of a switch buffer memory with acceptable probability for message loss, the quality indicators of its transmitting part.

Keywords: *flow tables, OpenFlow switch, queuing models, controller, probabilistic and temporal characteristics*

DOI: 10.18469/ikt.2020.18.3.05

Mochalov Valeriy Petrovich, North-Caucasus Federal University, 2 (building 9), Kulakov Avenue, Stavropol, Stavropol Territory, 355028, Russian Federation; Professor of Infocommunication Department, Doctor of Technical Science. Tel. +7 865 295-69-97. E-mail: mochalov.valery2015@yandex.ru

Linets Gennadiy Ivanovich, North-Caucasus Federal University, 2 (building 9), Kulakov Avenue, Stavropol, Stavropol Territory, 355028, Russian Federation; Head of Infocommunication Department, Doctor of Technical Science. Tel. +7 865 295-69-97. E-mail: kbytw@mail.ru

Bratchenko Natalya Yurevna, North-Caucasus Federal University, 2 (building 9), Kulakov Avenue, Stavropol, Stavropol Territory, 355028, Russian Federation; Associate Professor of Infocommunication Department, PhD in Physics and Mathematics. Tel. +7 865 295-69-97. E-mail: n.b.20062@yandex.ru

References

1. Bholebawa I.Z., Dalal U.D. Performance analysis of SDN / OpenFlow controllers: POX versus floodlight. *Wireless Personal Communications*, 2008, vol. 98, no. 2, pp. 1679–1699. DOI: <https://doi.org/10.1007/s11277-017-4939-z>.

2. Goto Y. et al. Queueing analysis of software defined network with realistic OpenFlow-based switch model. *Computer Networks*, 2019, vol. 164, pp. 106892. DOI: <https://doi.org/10.1016/j.comnet.2019.106892>.
3. D. Gross et al. *Fundamentals of Queueing Theory*. 1st ed. Hoboken: John Wiley & Sons, 2008, 509 p. DOI: <https://doi.org/10.1002/9781118625651>.
4. Hanhua L. et al. OSS/BSS Framework based on NGOSS. *2009 International Forum on Computer Science-Technology and Applications*, 2009, pp. 466–471. DOI: <https://doi.org/10.1109/IFCSTA.2009.120>.
5. Kreutz D. et al. Software-defined networking: a comprehensive survey. *Proceedings of the IEEE*, 2015, vol. 103, no. 1, pp. 14–76. DOI: <https://doi.org/10.1109/JPROC.2014.2371999>.
6. Lechler T., Taylor B.J., Klingenberg B., Innovation vs. Network Operation. *PICMET '07 – 2007 Portland International Conference on Management of Engineering & Technology*, 2007, pp. 2940–2947. DOI: <https://doi.org/10.1109/PICMET.2007.4349638>.
7. Lee E.A. The problem with threads. *Computer*, 2006, vol. 39, no. 5, pp. 33–42. DOI: <https://doi.org/10.1109/MC.2006.180>.
8. Li T., Chen J., Fu H. Application scenarios based on SDN: an overview. *Journal of Physics: Conference Series*, 2019, vol. 1187, no. 5, pp. 052067. DOI: <https://doi.org/10.1088/1742-6596/1187/5/052067>.
9. Tarasov V.N., Kartashevskij I.V., Malahov S.V. Theoretical and experimental study of latency in software-configurable networks. *Infokommunikacionnye tehnologii*, 2015, vol. 13, no. 4, pp. 409–413. DOI: <https://doi.org/10.18469/ikt.2015.13.4.08>. (In Russian.)
10. Mochalov V.P., Bratchenko N.Yu., Yakovlev S.V. Analytical model of integration system for program components of distributed object applications. *2018 International Russian Automation Conference (RusAutoCon)*, 2018, pp. 1–4. DOI: <https://doi.org/10.1109/RUSAUTOCON.2018.8501806>.
11. Nunes B.A.A. et al. A survey of software-defined networking: past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, 2014, vol. 16, no. 3, pp. 1617–1634. DOI: <https://doi.org/10.1109/SURV.2014.012214.00180>.
12. Ansel J., Amarasinghe S. Kendo: Efficient deterministic multithreading in software. *ACM SIGPLAN Notices*, 2009, vol. 44, no. 3, p. 97. DOI: <https://doi.org/10.1145/1508284.1508256>.
13. Rao N.S. Performance comparison of SDN solutions for switching dedicated long-haul connections. *The International Symposium on Advances in Software Defined Networking and Network Functions Virtualization*, 2016. URL: <https://www.osti.gov/biblio/1267045-performance-comparison-sdn-solutions-switching-dedicated-long-haul-connections>.
14. Samujlov K.E. et al. Model of functioning of telecommunication equipment of software-defined networks. *Sovremennye informatsionnye tehnologii i IT-obrazovanie*, 2018, vol. 14, no. 1, pp. 13–26. DOI: <https://doi.org/10.25559/SITITO.14.201801.013-026>. (In Russian.)
15. Simoes J., Wahle S. The future of services in next generation networks. *IEEE Potentials*, 2011, vol. 30, no. 1, pp. 24–29. DOI: <https://doi.org/10.1109/MPOT.2010.939761>.
16. Singh D. et al. Analytical modelling of software and hardware switches with internal buffer in software-defined networks. *Journal of Network and Computer Applications*, 2019, vol. 136, pp. 22–37. DOI: <https://doi.org/10.1016/j.jnca.2019.03.006>.
17. Sutter H., Larus J. Software and the concurrency revolution. *Queue*, 2005, vol. 3, no. 7, p. 54. DOI: <https://doi.org/10.1145/1095408.1095421>.
18. Priya A.V., Radhika N. Performance comparison of SDN OpenFlow controllers. *International Journal of Computer Aided Engineering and Technology*, 2019, vol. 11, no. 4, p. 467. DOI: <https://doi.org/10.1504/IJCAET.2019.10020284>.
19. Y.3300: *Framework of software-defined networking*. URL: <https://www.itu.int/rec/T-REC-Y.3300-201406-I/en> (accessed 01.06.2020).

Received 01.07.2020