



Теоретические вопросы информатики, прикладной математики, компьютерных наук и когнитивно-информационных технологий

Theoretical questions of computer science, computational mathematics, computer science and cognitive information technologies

УДК 004.942

DOI: 10.25559/SITITO.14.201801.013-026

МОДЕЛЬ ФУНКЦИОНИРОВАНИЯ ТЕЛЕКОММУНИКАЦИОННОГО ОБОРУДОВАНИЯ ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЕЙ

К.Е. Самуйлов, И.А. Шалимов, И.Г. Бужин, Ю.Б. Миронов

Российский университет дружбы народов, г. Москва, Россия

Аннотация

Строится математическая модель функционирования коммутатора программно-конфигурируемой сети в виде сети массового обслуживания, состоящей из двух систем массового обслуживания: первая моделирует входной буфер данных и устройство считывания информации из заголовка пакета, вторая – таблицу адресации коммутатора программно-конфигурируемой сети. Поступление данных в программно-конфигурируемых сетях имеет вероятностный характер при их детерминированной обработке в каналах связи и узлах коммутации. Поэтому данная математическая модель функционирования коммутатора программно-конфигурируемой сети была построена на основе систем и сетей массового обслуживания. Поступающий в сеть поток заявок был разделен на два пуассоновских потока разнотипных заявок, первый из которых соответствует пакетам,

Об авторах:

Самуйлов Константин Евгеньевич, доктор технических наук, профессор, заведующий кафедрой прикладной информатики и теории вероятностей, Российский университет дружбы народов (117198, Россия, г. Москва, ул. Миклухо-Маклая, д. 6); ORCID: <http://orcid.org/0000-0002-6368-9680>, ksam@sci.pfu.edu.ru

Шалимов Игорь Анатольевич, доктор технических наук, профессор, профессор кафедры прикладной информатики и теории вероятностей, Российский университет дружбы народов (117198, Россия, г. Москва, ул. Миклухо-Маклая, д. 6); ORCID: <http://orcid.org/0000-0003-0256-4291>, shalimov579@rambler.ru

Бужин Игорь Геннадьевич, аспирант кафедры прикладной информатики и теории вероятностей, Российский университет дружбы народов (117198, Россия, г. Москва, ул. Миклухо-Маклая, д. 6); ORCID: <http://orcid.org/0000-0001-8224-4065>, bigvrn93@mail.ru

Миронов Юрий Борисович, кандидат технических наук, старший преподаватель кафедры прикладной информатики и теории вероятностей, Российский университет дружбы народов (117198, Россия, г. Москва, ул. Миклухо-Маклая, д. 6); ORCID: <http://orcid.org/0000-0003-4790-9951>, mistiam@gmail.com

© Самуйлов К.Е., Шалимов И.А., Бужин И.Г., Миронов Ю.Б., 2018



приходящие на управляющий порт коммутатора (от контроллера), а второй поток - остальным прибывающим на коммутатор пакетам. Поток, соответствующий пакетам, приходящим на коммутатор от контроллера, имеет относительный приоритет над потоком из остальных прибывающих пакетов. В результате были получены формулы для расчета показателей производительности данного телекоммуникационного оборудования, такие как средние длины очередей ожидания приоритетных и неприоритетных заявок, среднее время ожидания заявок в очереди, вероятность потерь заявок на каждой фазе модели функционирования коммутатора. На основании полученных показателей качества функционирования данного телекоммуникационного оборудования можно дать оценку устойчивости телекоммуникационного оборудования программно-конфигурируемых сетей на различные информационные воздействия.

Ключевые слова

Телекоммуникационное оборудование; программно-конфигурируемые сети (ПКС); коммутатор ПКС; контроллер ПКС; таблицы адресации; конвейер адресации; протокол OpenFlow.

MODEL OF FUNCTIONING OF TELECOMMUNICATION EQUIPMENT FOR SOFTWARE-CONFIGURATED NETWORKS

Konstantin E. Samouylov, Igor A. Shalimov, Igor G. Buzhin, Yuriy B. Mironov

Peoples's Friendship University of Russia, Moscow, Russia

Abstract

A mathematical model of the functioning of the switch of a software defined networks is constructed in the form of a queuing network consisting of two queuing systems: the first simulates an input data buffer and a device for reading information from the header of the packet; the second is a table for addressing the switch of a software defined networks. The receipt of data in software defined networks has a probabilistic character in their deterministic processing in communication channels and switching nodes. Therefore, this mathematical model of the functioning of the switch of a software defined networks was built on the basis of queuing systems and networks. The stream of requests flowing into the network was divided into two Poisson streams of various types of applications, the first of which corresponded to the packets that came to the control port of the switch (from the controller), and the second flow to the remaining packets arriving on the switch. The flow corresponding to the packets arriving at the switch from the controller has a relative priority over the flow from the remaining arriving packets. As a result, formulas were obtained for calculating the performance indicators of this telecommunications equipment such as average waiting queues for priority and non-priority applications, the probability of loss of applications for each phase of the switch. Based on the received quality of service indicators for this telecommunications equipment, it is possible to assess the stability of switches in software defined networks for various information impacts.

Keywords

Telecommunication equipment; software defined networks (SDN); switch SDN; controller SDN; addressing tables; address pipeline; OpenFlow protocol.

Введение

Современное состояние и тенденции развития компьютерных сетей показали, что

потенциал роста производительности и пропускной способности сетей на основе



традиционных технологий практически исчерпан. Это связано с ростом затрат времени на маршрутизацию, с трудностями настройки сети и управления потоками в ней, особенно с ростом потребностей в виртуализации сетей, т.е. отображения нескольких логически изолированных сетей с независимыми политиками качества обслуживания на общий набор сетевых ресурсов. В настоящее время телекоммуникационные сети строятся на базе множества коммутационных устройств, каждое из которых может работать независимо от других устройств, самостоятельно определяя правила, по которым оно будет обрабатывать поступающие на него пакеты. Также современные коммутационные устройства вырабатывают свои правила коммутации пакетов по заложенным в них алгоритмам. Правила обмена служебной информацией между устройствами и способы её применения для выработки согласованных правил обработки пакетов определяются множеством разнообразных служебных протоколов, каждый из которых решает некоторое подмножество задач администрирования сети. Использование множества служебных протоколов отнюдь не делает сетевое администрирование простым. Решение данной проблемы значительно упростилось вместе со сравнительно недавним появлением концепции Программно-Конфигурируемых Сетей (ПКС). Такие сети позволят ускорить маршрутизацию, повысить удобство конфигурирования, виртуализации, настройки качества обслуживания сетей связи.

Цель работы

В современных условиях, когда на сети связи оказывается информационное воздействие, при проектировании программно-конфигурируемых сетей следует учитывать устойчивость функционирования телекоммуникационного оборудования на различные виды уязвимости к сетевым атакам. Таким образом, одной из проблем проектирования программно-конфигурируемых сетей является проблема оценки устойчивости функционирования телекоммуникационного оборудования данных сетей. Под оценкой устойчивости понимается оценка способности алгоритмов функционирования телекоммуникационного оборудования адекватно реагировать на информационные

воздействия. Таким образом, целью данной работы является разработка математической модели функционирования телекоммуникационного оборудования программно-конфигурируемой сети для дальнейшего анализа телекоммуникационного оборудования при информационном воздействии на него. Подобная математическая модель была построена в [2], но она оценивает только задержки, возникающие при различных методах управления коммутаторами, а не показатели качества функционирования коммутатора ПКС, необходимые для анализа телекоммуникационного оборудования при информационном воздействии.

Архитектура программно-конфигурируемых сетей

Главная идея ПКС заключается в отделении функций передачи трафика от функций управления (включая контроль как самого трафика, так и осуществляющих его передачу устройств). Это происходит за счет создания специального программного обеспечения, которое может работать на отдельном сервере (компьютере) и которое находится под контролем администратора сети. Все маршрутизаторы и коммутаторы объединяются под управлением контроллера ПКС или Сетевой Операционной Системы (СОС), которая обеспечивает приложениям доступ к управлению сетью и которая постоянно отслеживает конфигурацию средств сети.

Таким образом, функция коммутатора по управлению переносится на отдельное центральное устройство – контроллер ПКС. Такой подход позволяет управлять и контролировать за состоянием сети на логически централизованном контроллере. Кроме того, появляется возможность уровню управления отделяться от физической составляющей, используя логическое представление сети в целом. Взаимодействие между уровнем передачи данных осуществляется посредством единого унифицированного открытого интерфейса.

В архитектуре ПКС согласно [1] выделяют 3 уровня (рис. 1):

- уровень инфраструктуры сети, представляющий набор сетевых структур (коммутаторы и каналы связи);
- уровень управления, состоящий из



операционной системы, которая обеспечивает приложения сетевые сервисы и программный интерфейс для управления сетевыми устройствами и сетью;

- уровень приложений.

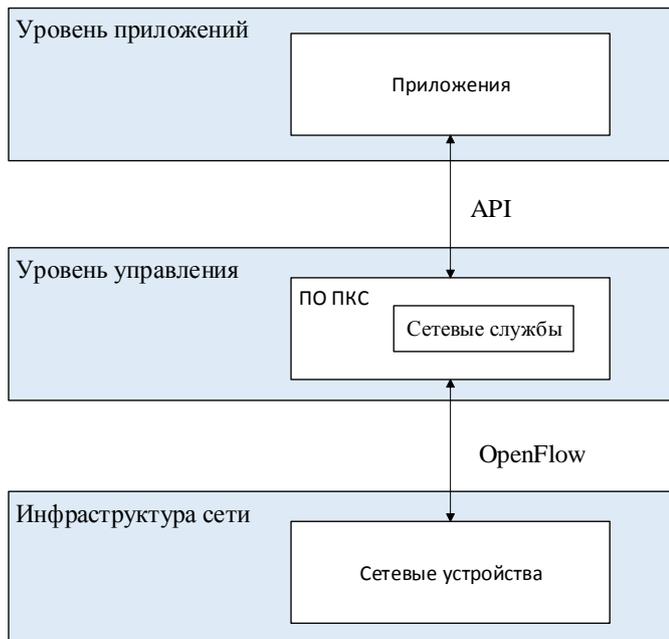


Рис. 1. Архитектура ПКС

Одно из центральных мест в архитектуре ПКС занимает сетевая операционная система, которая в общем смысле представляет собой операционную систему, обеспечивающую обработку, хранение и передачу данных в информационной сети. Сетевая операционная система или контроллер ПКС определяет взаимосвязанную группу протоколов верхних уровней, обеспечивающих основные функции сети: адресацию объектов, функционирование служб, обеспечение безопасности данных, управление сетью.

Контроллер ПКС формирует данные о состоянии всех ресурсов сети и обеспечивает доступ к ним для приложений управления сетью. Эти приложения управляют разными аспектами функционирования сети, включая построение топологии, принятие маршрутизирующих решений, балансировку нагрузки.

Для реализации этой идеи был разработан открытый протокол OpenFlow для управления сетевым оборудованием, не ориентированный на продукты определенного поставщика. С помощью этого протокола специалисты сами

могут определять и контролировать, какие узлы, при каких условиях и с каким качеством могут взаимодействовать в сети.

Принцип функционирования телекоммуникационного оборудования программно-конфигурируемой сети

Разделение уровня управления и передачи данных решается в рамках ПКС, работающих под управлением протокола OpenFlow. В коммутаторе ПКС реализован только уровень передачи данных. Каждый коммутатор ПКС имеет несколько входных и выходных портов. Порты коммутатора соединены с портами другого коммутатора физическими каналами связи. Порт управления соединен с узлом контроллера ПКС – по этому каналу происходит обмен OpenFlow сообщениями для управления коммутатором.

Коммутатор снабжен набором таблиц адресаций (flow tables), образующих конвейер адресации (pipeline), который состоит из одной или нескольких последовательно соединенных таблиц адресаций. Пакет, поступивший на один из входных портов коммутатора, сначала обрабатывается (считывается информация из заголовка пакета), затем он поступает на конвейер адресации. Начинается последовательная обработка пакета в таблицах адресаций. Под пакетом здесь понимается битовая строка, из которой можно выделить две части: заголовок и полезную нагрузку. Операции, выполняемые над пакетами в таблицах адресаций, не изменяют нагрузку пакета, но способны изменять его заголовок.

Заголовок пакета состоит из нескольких полей. Как правило, в этих полях указываются идентификаторы сетевых протоколов, которые должны обрабатывать пакет, и используемая ими служебная информация. В процессе прохождения пакета через конвейер таблиц адресаций к заголовку пакета могут быть добавлены служебные поля (metadata), которые предназначены для передачи служебной информации внутри конвейера, и сбрасываются при поступлении пакета в один из выходных портов коммутатора. Состав и размер служебных полей определяется техническими характеристиками конкретного коммутатора ПКС.

Таблица адресации содержит в себе записи о потоках и инструкции применения действий к пакету (см. рис. 2). Записи о потоках и нужная



информация из заголовков пакетов начинают находить соответствия. Если соответствие найдено, то над пакетом выполняется соответствующее действие. Его результатом может быть удаление пакета из коммутатора, переход на другую (большую по номеру) таблицу адресации, передача контроллеру через защищенный канал OpenFlow или отправка пакета на нужный выходной порт коммутатора

OpenFlow. Если коммутатор не может выполнить инструкцию применения действия к пакету, то информация о пакете отправляется на контроллер через защищенный канал OpenFlow. Если информация из заголовка пакета не совпала ни с одной записью о потоках, то пакет передается на следующую таблицу адресации.

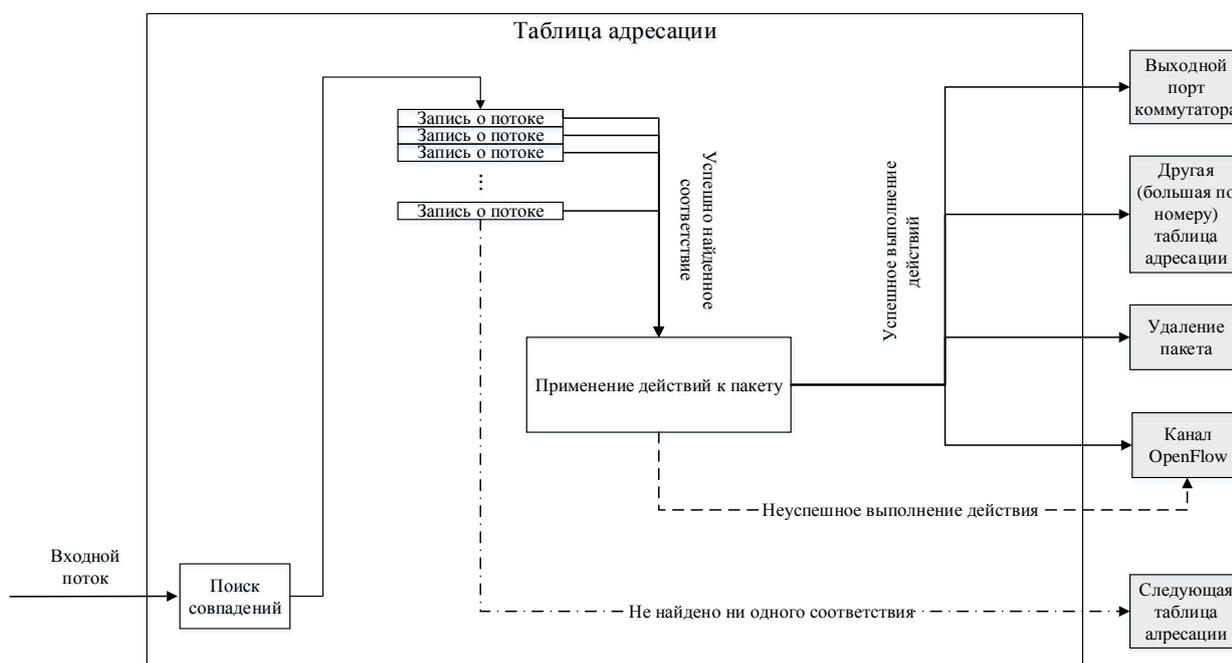


Рис. 2. Прохождение пакетов через таблицу адресации коммутатора ПКС

Если в последней таблице адресации между пакетом и записями о потоках не найдено соответствий, то пакет также удаляется из коммутатора.

Контроллер управляет содержимым таблиц адресаций подвластных ему коммутаторов. Протокол OpenFlow предполагает, что основной причиной того или иного изменения содержимого таблиц коммутаторов является

реакция контроллера на события из сети, и предполагает несколько типов уведомлений, чтобы коммутаторы могли оповестить контроллер о подобных событиях, а также несколько типов команд, с помощью которых контроллер мог бы вносить модификации в таблицы коммутаторов [1]. Контроллер ПКС управляет коммутатором ПКС при помощи протокола OpenFlow версии 1.5.1.

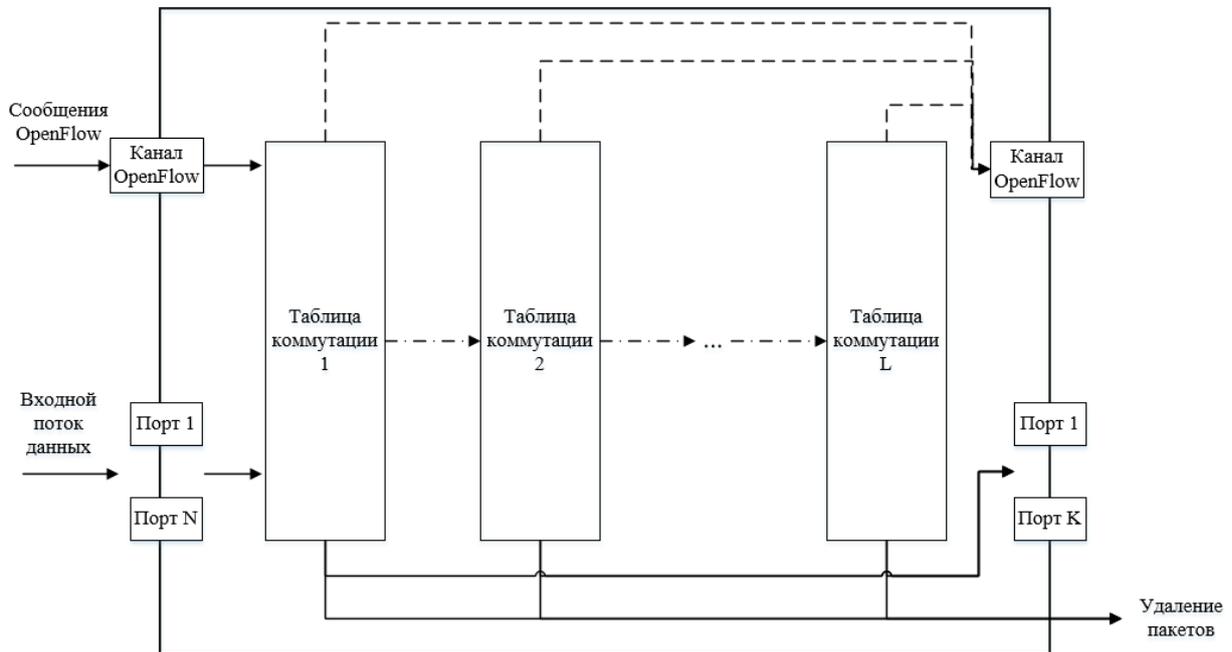


Рис. 3. Алгоритм функционирования коммутатора ПКС

Построение математической модели функционирования устройства считывания информации из заголовка пакета коммутатора программно-конфигурируемой сети

Алгоритм функционирования коммутатора ПКС может быть представлен следующим образом (см. рис. 3).

В коммутаторе может использоваться несколько таблиц адресаций $l, l = 1, \dots, L$, поддерживается защищенный канал для связи с удаленным контроллером ПКС. Каждая таблица коммутации потоков состоит из $i, i = 1, \dots, N$ записей о потоках. Коммутатор управляется контроллером посредством протокола OpenFlow версии 1.5.1. С помощью этого протокола контроллер может добавлять, обновлять и стирать записи в таблицах коммутаций, как в ответ на входящие пакеты, так и по своей инициативе.

Построим математическую модель функционирования коммутатора программно-конфигурируемой сети. Пусть в сеть поступают 2 пуассоновских потока разнотипных заявок с интенсивностями λ_1 и λ_2 и функциями распределения длительностей обслуживания B_1 и B_2 соответственно. Поступающие заявки с

интенсивностью λ_1 соответствуют пакетам, приходящим на управляющий порт коммутатора (от контроллера), заявки с интенсивностью λ_2 – остальным прибывающим на коммутатор пакетам. Затем заявки поступают в устройство считывания информации из заголовка пакета, которое представляет собой n идентичных обслуживающих приборов.

Далее, заявки поступают на первую таблицу адресации потоков. Каждая таблица адресации состоит из последовательного списка записей о потоках и устройства, реализующее действия над пакетом. Таким образом, в рамках ТМО модель функционирования коммутатора ПКС выглядит следующим образом (рис. 4).

Данная СеМО состоит из нескольких узлов (отдельных СМО): первый узел – очередь коммутатора и устройство считывания информации из заголовка пакета, второй и последующий за ним – таблицы адресации потоков.

Первый узел модели функционирования коммутатора ПКС представим в виде системы из n идентичных обслуживающих приборов и накопителя конечной емкости M ($2 \leq M < \infty$), в

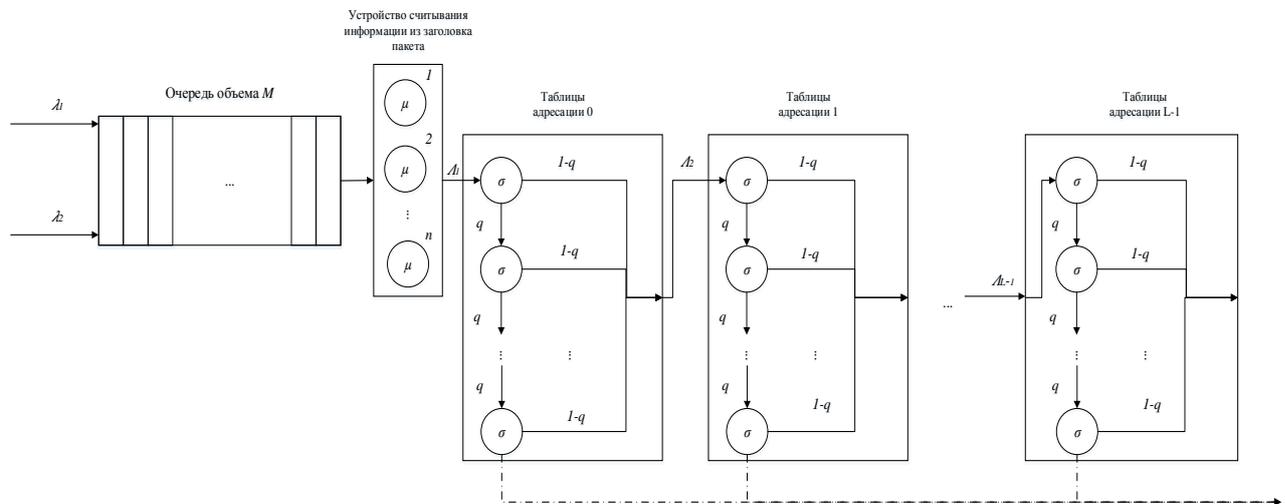


Рис. 4. Модель функционирования коммутатора ПКС

которую поступают два независимых пуассоновских потока заявок интенсивностей λ_1 и λ_2 с суммарной интенсивностью $\lambda = \lambda_1 + \lambda_2$. Данные потоки занумерованы в порядке убывания важности заявок. В момент окончания обслуживания на освободившийся прибор выбирается заявка из непустой очереди с минимальным номером.

Поступающие заявки с интенсивностью λ_1 соответствуют пакетам, приходящие на управляющий порт коммутатора (от контроллера), заявки с интенсивностью λ_2 – остальным прибывающим на коммутатор пакетам из внешних сетей.

Времена обслуживания заявок каждого из потоков независимы в совокупности и имеют общую функцию распределения $B(x) = 1 - e^{-\mu x}$, $x > 0$ одну и ту же для всех приборов. Если все приборы заняты, то поступившая заявка ожидает в накопителе, при этом заявки обоих потоков образуют общую очередь. К заявкам первого приоритета (1-заявки или приоритетные заявки) относятся пакеты, поступающие в коммутатор ПКС от контроллера. Они обладают относительным приоритетом по сравнению с заявками второго потока (2-заявки, или неприоритетные заявки). Это означает, что при наличии в очереди заявок обоих потоков на обслуживание выбирается 1-заявка, а 2-заявка может быть выбрана на обслуживание тогда, когда в очереди нет 1-заявок. Внутри каждого

приоритетного класса заявки обслуживаются в соответствии с дисциплиной обслуживания FIFO. При переполнении накопителя, т.е. когда в нем нет свободных мест, заявка любого типа теряется. Эти два типа заявок формируют две очереди, хотя и в общем накопителе, поэтому построим модель этих очередей.

Пусть $v_i(t), i = 1, 2$ – число i -заявок в накопителе и $v(t)$ – общее число заявок, обслуживаемых на приборах в момент t . Определим теперь процесс $\{\eta(t), t \geq 0\}$ следующим образом: если в момент t в системе меньше, чем n заявок, то положим $\eta(t) = v(t)$, в противном случае положим $\eta(t) = (v_1(t), v_2(t))$. Так как рассматриваемая СМО является экспоненциальной (входящие потоки пуассоновские, времена обслуживания экспоненциально распределены с параметром μ), то процесс $\{\eta(t), t \geq 0\}$ будет однородным марковским. Множество состояний X процесса $\{\eta(t), t \geq 0\}$ имеет вид:

$$X = \{(i), i = \overline{0, n-1}; (k, m), 0 \leq k + m \leq M\}.$$

Состояния из этого множества интерпретируются следующим образом: если $\eta(t) = (i)$ в некоторый момент времени t , то заняты обслуживанием i приборов, а если $\eta(t) = (k, m)$, то все n приборов заняты обслуживанием, а в накопителе ожидают k приоритетных заявок и m неприоритетных заявок.

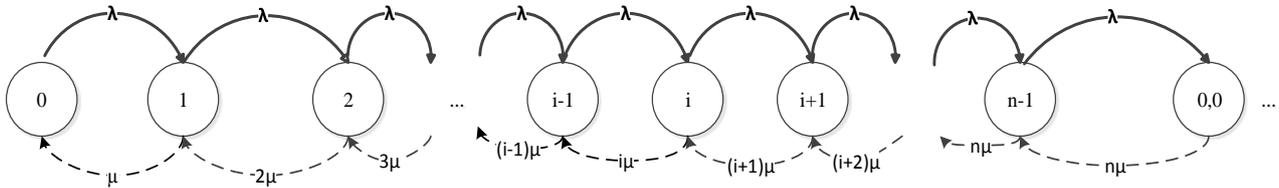


Рис. 5. Диаграмма переходов для $\eta(t) = v(t)$

Пусть $p_i = \lim_{t \rightarrow \infty} P\{\eta(t) = (i)\}$ и $p_{km} = \lim_{t \rightarrow \infty} P\{\eta(t) = (k, m)\}$. Так как все состояния процесса $\{\eta(t), t \geq 0\}$ сообщаются, а их число конечно ($n + \frac{(M+1)(M+2)}{2}$), то в силу эргодичности процесса $\{\eta(t), t \geq 0\}$ предельные вероятности p_i и p_{km} существуют, строго положительны, не зависят от начальных условий и удовлетворяют

системе уравнений равновесия (СУР). Получим СУР.

Найдем уравнения равновесия для процесса, когда в момент t в системе меньше, чем n заявок. Диаграмма переходов будет выглядеть следующим образом (см. рис. 5).

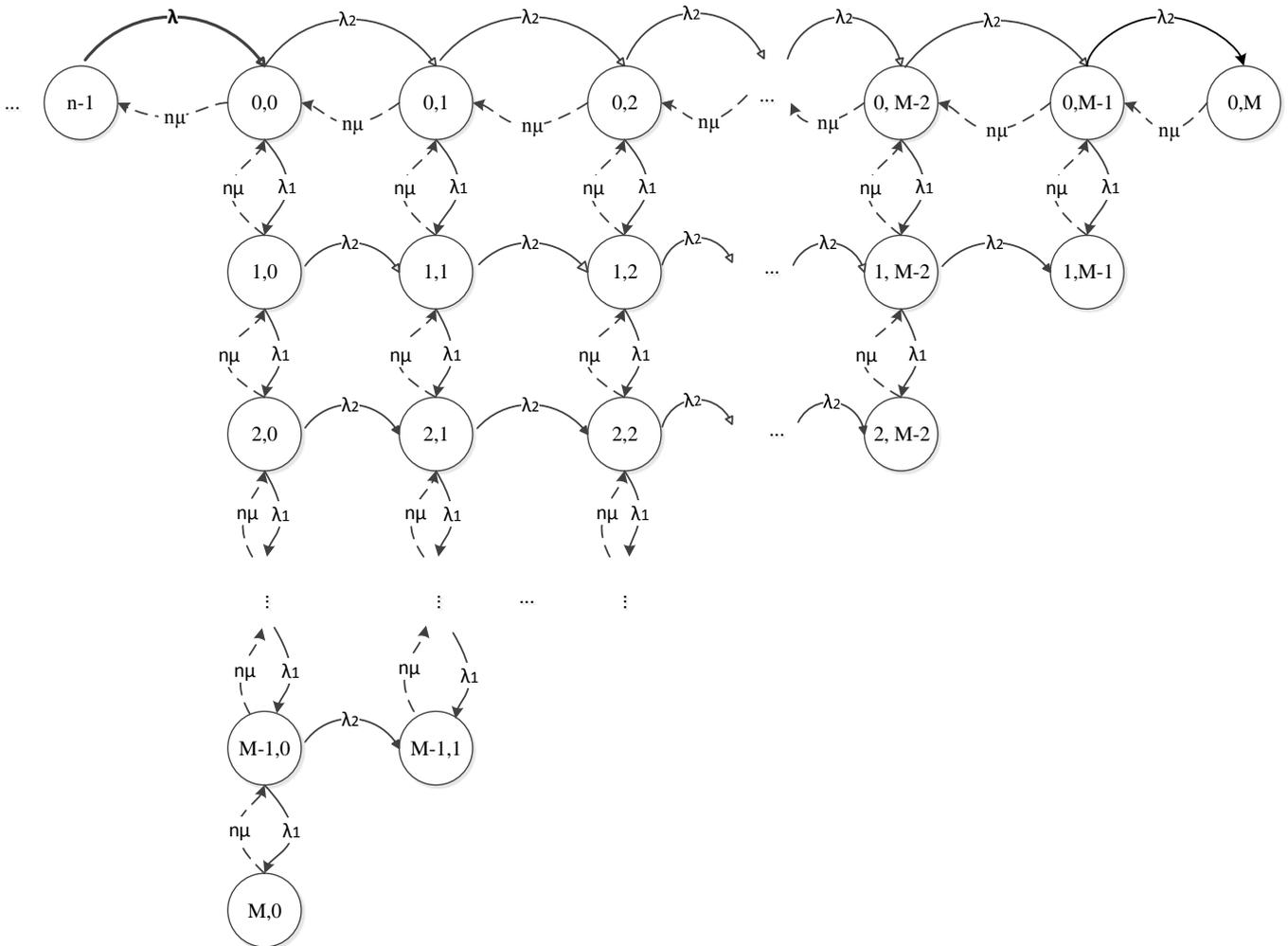


Рис. 6. Диаграмма переходов для $\eta(t) = (v_1(t), v_2(t))$

Тогда уравнения равновесия для данного процесса будут иметь вид:

$$\begin{aligned} 0 &= -\lambda p_0 + \mu p_1 \\ 0 &= -(\lambda + i\mu)p_i + \lambda p_{i-1} + (i+1)\mu p_{i+1}, \end{aligned} \quad (1)$$



$$i = \overline{1, n-1}. \quad (2)$$

Теперь найдем уравнения равновесия для процесса, когда все n приборов заняты обслуживанием, а в накопителе ожидают k приоритетных заявок и m неприоритетных заявок. Диаграмма переходов будет выглядеть следующим образом (см. рис. 6). Тогда получим оставшиеся уравнения для СУР:

$$0 = -(\lambda + n\mu)p_{0,0} + \lambda p_{n-1} + n\mu p_{1,0} + n\mu p_{0,1} \quad (3)$$

$$0 = -(\lambda + n\mu)p_{k,0} + \lambda_1 p_{k-1,0} + n\mu p_{k+1,0}, \quad k = \overline{1, M-1} \quad (4)$$

$$0 = -n\mu p_{M,0} + \lambda_1 p_{M-1,0} \quad (5)$$

$$0 = -(\lambda + n\mu)p_{0,m} + \lambda_2 p_{0,m-1} + n\mu(p_{1,m} + p_{0,m+1}), \quad m = \overline{1, M-1} \quad (6)$$

$$0 = -n\mu p_{0,M} + \lambda_2 p_{0,M-1} \quad (7)$$

$$0 = -(\lambda + n\mu)p_{k,m} + \lambda_1 p_{k-1,m} + \lambda_2 p_{k,m-1} + n\mu p_{k+1,m}, \quad 1 \leq k+m < M \quad (8)$$

$$0 = -n\mu p_{k,M-k} + \lambda_1 p_{k-1,M-k} + \lambda_2 p_{k,M-k-1}, \quad k = \overline{1, M-1}. \quad (9)$$

Решим получившуюся СУР (1) - (9). Введем стационарную вероятность p_{n+i} того, что в очереди имеется i заявок безотносительно к их виду. Тогда:

$$p_{n+i} = \sum_{k+m=i} p_{k,m}. \quad (10)$$

$$D_j = \begin{bmatrix} -1-\rho & 1 & 0 & \dots & 0 & 0 & 0 \\ \rho_1 & -1-\rho & 1 & \dots & 0 & 0 & 0 \\ 0 & \rho_1 & -1-\rho & \dots & 0 & 0 & 0 \\ & \vdots & & \ddots & & & \\ 0 & 0 & 0 & \dots & \rho_1 & -1-\rho & 1 \\ 0 & 0 & 0 & \dots & 0 & \rho_1 & -1 \end{bmatrix},$$

где j - количество строк в матрице, $D_0 = (1), D_1 = (-1)$, а также векторы $\vec{p}_m^T = (p_{1,m}, p_{2,m}, \dots, p_{M-m,m})$ и $\vec{e}_1^T = (1, 0, \dots, 0)$. Тогда уравнения (4), (5), (8) и (9) в матричной записи будут иметь следующий тип:

$$0 = D_M \cdot \vec{p}_0 + \rho_1 \cdot p_n \cdot \vec{e}_1, \quad (13)$$

$$0 = D_{M-m} \cdot \vec{p}_m + \rho_1 \cdot p_{0,m} \cdot \vec{e}_1 + \rho_2 \cdot \vec{p}_{m-1}, \quad m = \overline{1, M-1}. \quad (14)$$

Пусть каждая поступившая заявка может ожидать начала обслуживания не более случайного времени, распределенного по экспоненциальному закону с параметром γ . Тогда $p_n = p_0 \cdot \frac{\lambda^n}{\mu(\mu+\gamma)\dots(\mu+(i-1)\gamma)}$, где

$$p_0 = \left[1 + \sum_{n=1}^M \frac{\lambda^n}{\mu(\mu+\gamma)\dots(\mu+(i-1)\gamma)} \right]^{-1}.$$

Таким образом, уравнения (4), (5), (8) и (9) мы разбили на M подсистем относительно неизвестных векторов $\vec{p}_0, \vec{p}_1, \dots, \vec{p}_{M-1}$ меньшего

Тогда для фиксированного $i = \overline{0, M}$, объединяя уравнения (3), (6) и (7) таким образом, чтобы сумма индексов $k+m$ у вероятности $p_{k,m}$ в первых членах правых частей этих уравнений равнялась i , получим уравнения, которым удовлетворяют вероятности p_{n+i} , которые вместе с уравнениями (1) и (2) представляют СУР для СМО $\langle M|M|n|r = M \rangle$. Тогда из уравнений (1), (2), (3), (6) и (7) получаем:

$$p_l = \begin{cases} \frac{\rho^l}{l!} p_0, & \text{если } l = \overline{1, n-1} \\ \frac{\rho^l}{n! n^{l-n}} p_0, & \text{если } l = \overline{n, n+M}, \end{cases} \quad (11)$$

$\rho = \frac{\lambda}{\mu}$, а p_0 определяется из условия нормировки $\sum_{l=0}^{\infty} p_l = 1$, т.е.:

$$p_0 = \left[\sum_{l=0}^{n-1} \frac{\rho^l}{l!} + \frac{\rho^n}{n!} \cdot \frac{1 - (\frac{\rho}{n})^{n+1}}{1 - \frac{\rho}{n}} \right]^{-1}. \quad (12)$$

Рассмотрим теперь уравнения (4), (5), (8) и (9) и запишем их в матричном виде. Разделим все эти уравнения на $n\mu$ и пусть $\rho_1 = \frac{\lambda_1}{n\mu}, \rho_2 = \frac{\lambda_2}{n\mu}, \rho = \rho_1 + \rho_2$. Также введем матрицы вида:

порядка, которые могут быть последовательно решены.

В системе уравнений (13) вероятность p_n известна, необходимо найти \vec{p}_0 . Решим данную систему при помощи правила Крамера:

$$D_M \cdot \vec{p}_0 = -\rho_1 \cdot p_n \cdot \vec{e}_1. \quad (13')$$

Определитель матрицы D_j обозначим через d_j и разложим его по элементам первого столбца. Тогда получим следующие рекуррентные соотношения:

$$d_0 = 1, \quad d_1 = -1, \quad d_j = -(1 + \rho)d_{j-1} - \rho_1 d_{j-2}, \quad j \geq 2. \quad (15)$$

Пусть матрица $\vec{D}_j, j \geq 2$, получается из матрицы D_j заменой элемента в правом нижнем углу, равного -1 , на $-1 - \rho$, и $\vec{D}_0 = (1)$ и $\vec{D}_1 = (-1 - \rho)$. Тогда определитель \vec{d}_j этих матриц вычисляется по рекуррентной формуле (15), но с начальными условиями $\vec{d}_0 = 1, \vec{d}_1 =$



$-1 - \rho$.

$d_j = -(1 + \rho)d_{j-1} - \rho_1 d_{j-2}$, $j \geq 2$ - линейное рекуррентное соотношение с постоянными коэффициентами второго порядка и с начальными условиями $d_0 = 1$, $d_1 = -1$. Тогда его характеристическое уравнение имеет вид:

$$r^2 = -(1 + \rho)r - \rho_1. \quad (16)$$

Данное уравнение имеет 2 корня:

$$r_{1,2} = \frac{-(1 + \rho) \pm \sqrt{(1 + \rho)^2 - 4\rho_1}}{2}. \quad (17)$$

Тогда общее решение рекуррентного соотношения (15) имеет вид:

$$d_j = C_1 \cdot r_1^j + C_2 \cdot r_2^j, \quad j \geq 0. \quad (18)$$

Коэффициенты C_1 и C_2 определяются из начальных условий:

$$\begin{cases} d_0 = C_1 + C_2, \\ d_1 = C_1 r_1 + C_2 r_2. \end{cases}$$

Тогда $C_1 = \frac{r_2 - 1}{r_2 - r_1}$ и $C_2 = \frac{1 - r_1}{r_2 - r_1}$. Таким образом:

$$d_j = \frac{r_2 - 1}{r_2 - r_1} \cdot r_1^j + \frac{1 - r_1}{r_2 - r_1} \cdot r_2^j, \quad j \geq 0. \quad (18)$$

Применяя правило Крамера для решения системы уравнений (13'), получаем:

$$p_{k,0} = \frac{(-\rho_1)^k p_n d_{M-k}}{d_M}, \quad k = \overline{1, M}. \quad (19)$$

Далее, рассмотрим определитель $d_{M-m,k}$, который получается из d_{M-m} путем замены k -го столбца на столбец $\overrightarrow{p_{m-1}}$. Раскладывая определитель $d_{M-m,k}$ по элементам k -го столбца, получаем:

$$\begin{aligned} d_{M-m,k} &= d_{M-m-k} \sum_{s=1}^{k-1} (-1)^{k+s} p_{s,m-1} \rho_1^{k-s} \overline{d_{s-1}} \\ &\quad + \overline{d_{k-1}} \sum_{s=k}^{M-m} (-1)^{k+s} p_{s,m-1} d_{M-m-s}, \\ &\quad k = \overline{1, M-m}, \end{aligned} \quad (20)$$

где принято $\sum_{s=1}^0 = 0$.

Аналогичным образом с (18) находим общее решение рекуррентного соотношения:

$$\tilde{d}_j = \frac{r_2 + 1 + \rho}{r_2 - r_1} \cdot r_1^j + \frac{r_1 + 1 + \rho}{r_1 - r_2} \cdot r_2^j, \quad j \geq 0. \quad (21)$$

Применяя правило Крамера для решения системы уравнений (14), получаем:

$$p_{k,m} = \frac{-\rho_2 d_{M-m,k} + (-\rho_1)^k p_{0,m} d_{M-m-k}}{d_{M-m}}, \quad k = \overline{1, M-m}, \quad m = \overline{1, M-1}. \quad (22)$$

Показатели качества функционирования устройства считывания информации из заголовка пакета коммутатора программно-конфигурируемой сети

Поскольку заявки обоих типов скапливаются в общем накопителе, то при поступлении заявки любого типа в полностью заполненную систему она теряется с вероятностью:

$$\pi_1 = p_{n+M} = \sum_{k+m=M} p_{k,m}. \quad (23)$$

Пропускная способность (интенсивность выходного потока) системы Λ_1 определяется как среднее число заявок, обслуживаемой системой в единицу времени:

$$\Lambda_1 = (\lambda_1 + \lambda_2)(1 - \pi_1) = \lambda(1 - \pi_1). \quad (24)$$

Пусть $q_{1,j} = p_{j,*} = \sum_{m=0}^{M-j} p_{j,m}$, $j = \overline{0, M}$, - стационарная вероятность того, что заняты n приборов и в накопителе ожидают j приоритетных заявок, и $q_{2,j} = p_{*,j} = \sum_{m=0}^{M-j} p_{m,j}$, $j = \overline{0, M}$, - стационарная вероятность того, что заняты n приборов и в накопителе ожидают j неприоритетных заявок. Тогда стационарные средние длины Q_1 и Q_2 очередей приоритетных и неприоритетных заявок соответственно определяется формулами:

$$Q_1 = \sum_{j=1}^M j q_{1,j}, \quad (25)$$

$$Q_2 = \sum_{j=1}^M j q_{2,j}. \quad (26)$$

Стационарная средняя длина очереди Q безотносительно к типу заявок определяется как:

$$Q = Q_1 + Q_2. \quad (27)$$

Среднее время ожидания приоритетных и неприоритетных заявок, принятых в систему, функционирующую в стационарном режиме, можно найти при помощи формулы Литтла. Заметим, что заявки обслуживаются в порядке их поступления, т.е. согласно принципу FIFO. Пусть w_1 - среднее время ожидания приоритетной заявки, w_2 - среднее время ожидания неприоритетной заявки. Тогда по формуле Литтла:

$$\lambda_1(1 - \pi_1)w_1 = Q_1, \quad (28)$$

$$\lambda_2(1 - \pi_1)w_2 = Q_2. \quad (29)$$

Построение математической модели функционирования таблиц адресаций коммутатора программно-конфигурируемой сети

После обработки на устройстве считывания информации из заголовка пакеты поступают с интенсивностью Λ_1 , которая определяется по



формуле (24), в конвейер обработки пакетов - последовательность таблиц адресаций потоков. Таблицы адресации нумеруются с 0 в порядке прохождения пакета. Каждая таблица адресации коммутатора ПКС состоит из последовательного списка записей о потоках и устройства, которое реализует действия над пакетом. При обработке пакета таблицей адресации информация о пакете сопоставляется с записями о потоках в этих таблицах. Если запись в таблице совпадает с информацией о пакете, то над пакетом выполняется набор команд. Далее пакет может перейти в следующую таблицу адресации, поступить на выходной порт, быть удален либо поступить на порт OpenFlow для отправки на контроллер. Все эти действия определяет набор команд. Если запись в таблице не совпала с информацией о пакете, то пакет отправляется на следующую запись в таблице адресации потоков. Если не найдено ни одного соответствия в таблице, то пакет поступает на следующую таблицу адресации потоков этого же коммутатора ПКС. Если же в последней таблице адресации не найдено ни одного соответствия, то пакет отправляется на

контроллер ПКС по защищенному OpenFlow каналу связи.

Второй и последующие узлы модели коммутатора ПКС представимы в виде однолинейных СМО, в которые поступает пуассоновский поток с интенсивностью Λ_1 . Обработка пакета каждой записью $i, i = \overline{1, S}$, и выполнение дальнейших действий над пакетом будет представляется обслуживающим прибором со средним временем сопоставления пакета с записью о потоках $1/\sigma$. Также введем дополнительную вероятность q прохождения пакета к следующей записи и $1 - q$ - вероятность покидания системы на данной записи. Последней записью будет являться запись на позиции S или то, на котором произошло совпадение по его критерию. Пакет, достигнувший записи S , покидает систему обслуживания. Времена обслуживания заявок каждой записью функцию распределения $B(x) = 1 - e^{-\sigma x}, x > 0$ одну и ту же для всех приборов. Если все приборы заняты, то заявка покидает систему (блокируется).

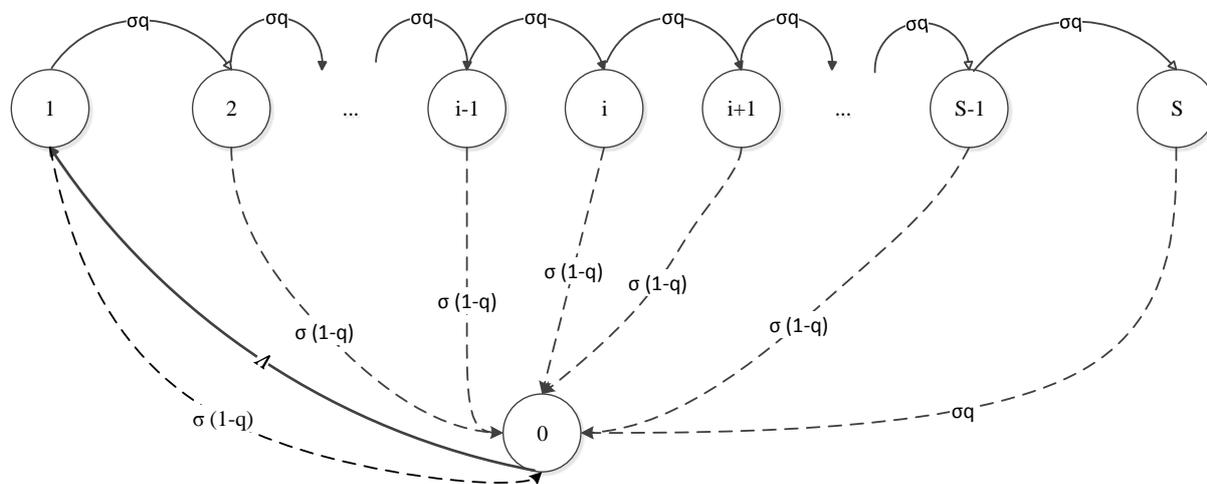


Рис. 7. Диаграмма переходов для $\gamma(t)$

Пусть $\gamma(t)$ - номер фазы обслуживания заявки на приборе в момент времени t , $0 \leq \gamma(t) \leq S$ (если $\gamma(t) = 0$, то в системе нет заявок). Процесс $\{\gamma(t), t \geq 0\}$, описывающий поведение СМО, является марковским случайным процессом с непрерывным временем и дискретным множеством состояний:

$$Y = \{(i), i = \overline{0, S}\}.$$

Обозначим через p_i - стационарную вероятность того, что система находится в состоянии (i) , т.е. на приборе обслуживается заявка на фазе под номером i (p_0 - стационарная вероятность того, что в системе нет заявок).

На основании описанной модели



разработана диаграмма состояний и переходов, представленная на рис.7.

Стационарное распределение удовлетворяет системе уравнений равновесия:

$$-A_1 p_0 + \sigma(1-q) \sum_{i=1}^{S-1} p_i - \sigma q p_S = 0, \quad (30)$$

$$A_1 p_0 - \sigma p_1 = 0, \quad (31)$$

$$\sigma q p_i - \sigma p_{i+1} = 0, i = \overline{2, S-1}, \quad (32)$$

$$\sigma q p_{S-1} - \sigma q p_S = 0. \quad (33)$$

Также стационарные вероятности удовлетворяют условию нормировки:

$$\sum_{i=0}^S p_i = 1. \quad (34)$$

Тогда:

$$p_1 = \frac{A_1}{\sigma} p_0, \quad (35)$$

$$p_i = q^{i-1} \frac{A_1}{\sigma} p_0, \quad i = \overline{2, S-1}, \quad (36)$$

$$p_S = p_{S-1} = q^{S-2} \frac{A_1}{\sigma} p_0, \quad (37)$$

где

$$p_0 = \left(1 + \frac{A_1}{\sigma} + \frac{A_1}{\sigma} \cdot \frac{q(q^{S-1}-1)}{q-1} + \frac{A_1}{\sigma} q^{S-1} \right)^{-1}. \quad (38)$$

Показатели качества функционирования таблиц адресаций коммутатора программно-конфигурируемой сети

Если на записи S не обнаружилось совпадения с информацией о пакете, то заявка блокируется (отправляется на контроллер). Тогда вероятность блокировки вычисляется по формуле:

$$\pi_2 = p_S = q^{S-2} \frac{A_1}{\sigma} \left(1 + \frac{A_1}{\sigma} + \frac{A_1}{\sigma} \cdot \frac{q(q^{S-1}-1)}{q-1} + \frac{A_1}{\sigma} q^{S-1} \right)^{-1}. \quad (39)$$

Пропускная способность (интенсивность выходного потока) второй фазы модели коммутатора ПКС A_2 определяется как среднее число заявок, обслуживаемой системой в единицу времени:

$$A_2 = A_1(1 - \pi_2). \quad (40)$$

Интенсивность заблокированных и отправляющихся на контроллер ПКС пакетов вычисляется по формуле:

$$A_{2loss} = A_1 \pi_2. \quad (41)$$

Выводы

В данной статье была построена модель функционирования коммутатора программно-конфигурируемой сети. В результате чего, получена система уравнений равновесия и успешно решена. На основании полученных результатов рассчитаны некоторые характеристики качества обслуживания коммутатора программно-конфигурируемой сети, а именно вероятность потери заявок, пропускную способность системы, среднюю длину очереди, среднее время ожидания заявок в буферах данных. На основании полученных характеристик можно провести анализ устойчивости коммутатора программно-конфигурируемой сети на различные виды информационных воздействий.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Open Networking Foundation. SDN ARCH 1.0 06062014 (Technical Reference), non-normative, type 2, 2014. 68 p. URL: https://www.opennetworking.org/wp-content/uploads/2013/02/TR_SDN_ARCH_1.0_06062014.pdf (дата обращения: 10.02.2018).
- [2] Малахов С.В., Тарасов В.Н., Карташевский И.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>
- [3] Смелянский Р.Л. Программно-конфигурируемые сети // Открытые системы. СУБД. 2012. № 9. С. 15-26. URL: <https://www.osp.ru/os/2012/09/13032491/> (дата обращения: 10.02.2018).
- [4] Вишневецкий В.М. Теоретические основы проектирования компьютерных сетей. Москва: Техносфера, 2003. 512 с.
- [5] Смелянский Р.Л. Компьютерные сети. Москва: Издательский центр «Академия», 2011. 240 с.
- [6] Бахарева Н.Ф., Тарасов В.Н. Аппроксимативные методы и модели массового обслуживания. Исследование компьютерных сетей. Самара: Изд-во СНЦ РАН, 2011. 327 с.
- [7] Tarasov V.N., Malakhov S.V. Statistical data handling program of Wireshark analyzer and incoming traffic research // Trudy ISP RAN [The Proceedings of ISP RAS]. 2015. Vol. 27, issue 3. Pp. 303-314. URL: <https://elibrary.ru/item.asp?id=23832950> (дата обращения: 10.02.2018).
- [8] SDN-Guard: Protecting SDN controllers against SDN rootkits / D. Tatang [et al.] // Proceedings of the 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). Berlin, Germany, 2017. Pp. 297 - 302. DOI: <https://doi.org/10.1109/NFV-SDN.2017.8169856>



- [9] Pujolle G. SDN (Software-Defined Networking). Vol. 1. Software Networks: Virtualization, SDN, 5G and Security. UK: ISTE Ltd; USA: John Wiley & Sons, Inc., 2015. 232 p. DOI: <https://doi.org/10.1002/9781119005100>
- [10] Shao Y.Zh., Scott-Hayward S., Jacquin L., Hill R. Guide to Security in SDN and NFV: Challenges, Opportunities, and Applications. Cham: Springer International Publishing, 2017. 331 p.
- [11] Nadeau T.D., Gray K. SDN: Software Defined Networks. O'Reilly Media, Inc. 2013. 384 p.
- [12] Azodolmolky S. Software Defined Networking with Openflow: Get Hands-on with the Platforms and Development Tools Used to Build Openflow Network Applications. Packt Publishing, Birmingham – Mumbai, 2013. 138 p.
- [13] Fei H. Network Innovation Through Openflow and Sdn: Principles and Design. CRC P Inc, Taylor & Francis Group, 2014. 520 p.
- [14] Alaaudhin Sh. Application Layer Firewall Using OpenFlow. Interdisciplinary Telecommunications Graduate Theses Dissertations, 2014. 69 p. URL: https://scholar.colorado.edu/cgi/viewcontent.cgi?article=1000&context=tlen_gradetds (дата обращения: 10.02.2018)
- [15] Coker O., Azodolmolky S. Software-defined Networking with Openflow – Second Edition. Packt Publishing, Limited, 2017. 246 p.
- [16] Gerrity A., Hu F. SDN/OpenFlow: Concepts and Applications / Ashley Gerrity, Fei Hu // Network Innovation through OpenFlow and SDN / Hu Fei. Taylor & Francis Group, LLC, 2014. Chapter 1. Pp. 3-16. DOI: <https://doi.org/10.1201/b16521-3>
- [17] Agouros K. Software Defined Networking. SDN-Praxis mit Controllern und OpenFlow. De Gruyter, 2016. Pp. 26–48. DOI: <https://doi.org/10.1515/9783110451870-003>
- [18] Agouros K. Software Defined Networking. OpenFlow-Implementierungen. De Gruyter, 2016. Pp. 49-70. DOI: <https://doi.org/10.1515/9783110451870-004>
- [19] Karyotis V., Khouzani M.H.R. Appendix B - Elements of queuing theory and queuing networks, In Malware Diffusion Models for Wireless Complex Networks, Morgan Kaufmann, Boston, 2016. Pp. 235-254. DOI: <https://doi.org/10.1016/B978-0-12-802714-1.00023-2>
- [20] MiniReal: A real SDN network testbed built over an SDN bare metal commodity switch / S. Wang, I. Lee // Proceedings of 2017 IEEE International Conference on Communications (ICC). Paris, France, 2017. Pp. 1-6. DOI: <https://doi.org/10.1109/ICC.2017.7997283>
- [21] Incremental SDN-Enabled Switch Deployment for Hybrid Software-Defined Networks / M. Huang, W. Liang // Proceedings of the 26th International Conference on Computer Communication and Networks (ICCCN). Vancouver, BC, Canada, 2017. Pp. 1-6. DOI: <https://doi.org/10.1109/ICCCN.2017.8038498>
- [22] Goransson P., Black C., Culver T. Software defined networks: A comprehensive approach. 2nd Edition. Elsevier, 2016. 436 p.
- [23] Load balancing – towards balanced delay guarantees in NFV/SDN / H. Wang, J. Schmitt // Proceedings of 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). Palo Alto, CA, USA, 2016. Pp. 240 – 245. DOI: <https://doi.org/10.1109/NFV-SDN.2016.7919504>
- [24] A low-delay SDN-based countermeasure to eavesdropping attacks in industrial control systems / G.K. Ndonga, R. Sadre // Proceedings of 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). Berlin, Germany, 2017. Pp. 1-7. DOI: <https://doi.org/10.1109/NFV-SDN.2017.8169840>

Поступила 26.01.2018; принята к публикации 20.03.2018; опубликована онлайн 30.03.2018.

REFERENCES

- [1] Open Networking Foundation. SDN ARCH 1.0 06062014 (Technical Reference), non-normative, type 2, 2014. 68 p. Available at: https://www.opennetworking.org/wp-content/uploads/2013/02/TR_SDN_ARCH_1.0_06062014.pdf (accessed 10.02.2018).
- [2] Malakhov S.V., Tarasov V.N., Kartashevsky I.V. Theoretical and experimental study of delay in software defined networks. *Infokommunikacionnye tehnologii*. 2015; 13(4):409-413. (In Russian) DOI: <https://doi.org/10.18469/ikt.2015.13.4.08>
- [3] Smelyanskiy R.L. Software defined networks. *Open Systems. DBMS*. 2012; 9:15-26. Available at: <https://www.osp.ru/os/2012/09/13032491/> (accessed 10.02.2018). (In Russian)
- [4] Vishnevskiy V. Theoretical foundations of computer network design. Moscow: The Technosphere, 2003. 512 p. (In Russian)
- [5] Smelyanskiy R. Computer networks. M.: Publishing Center «Academy», 2011. 240 p. (In Russian)
- [6] Bahareva N., Tarasov V. Approximation methods and queuing models. Research of computer networks. Samara: Publishing house of the SSC RAS, 2011. 327 p. (In Russian)
- [7] Tarasov V.N., Malakhov S.V. Statistical data handling program of Wireshark analyzer and incoming traffic research. *Trudy ISP RAN* [The Proceedings of ISP RAS]. 2015; 27(3):303-314. Available at: <https://elibrary.ru/item.asp?id=23832950> (accessed 10.02.2018).
- [8] Tatang D., Quinkert F., Frank J., Ropke C., Holz T. SDN-Guard: Protecting SDN controllers against SDN rootkits. *Proceedings of 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. Berlin, Germany, 2017. p. 297 – 302. DOI: <https://doi.org/10.1109/NFV-SDN.2017.8169856>
- [9] Pujolle G. SDN (Software-Defined Networking). Vol. 1. Software Networks: Virtualization, SDN, 5G and Security. UK: ISTE Ltd; USA: John Wiley & Sons, Inc., 2015. 232 p. DOI: <https://doi.org/10.1002/9781119005100>
- [10] Shao Y.Zh., Scott-Hayward S., Jacquin L., Hill R. Guide to Security in SDN and NFV: Challenges, Opportunities, and Applications. Cham: Springer International Publishing, 2017. 331 p.
- [11] Nadeau T.D., Gray K. SDN: Software Defined Networks. O'Reilly Media, Inc. 2013. 384 p.
- [12] Azodolmolky S. Software Defined Networking with Openflow: Get Hands-on with the Platforms and Development Tools Used to Build Openflow Network Applications. Packt Publishing, Birmingham – Mumbai, 2013. 138 p.



- [13] Fei H. Network Innovation Through Openflow and Sdn: Principles and Design. CRC P Inc, Taylor & Francis Group, 2014. 520 p.
- [14] Alaauddin Sh. Application Layer Firewall Using OpenFlow. Interdisciplinary Telecommunications Graduate Theses Dissertations, 2014. 69 p. Available at: https://scholar.colorado.edu/cgi/viewcontent.cgi?article=1000&context=tlen_gradetds (accessed 10.02.2018).
- [15] Coker O., Azodolmolky S. Software-defined Networking with Openflow - Second Edition. Packt Publishing, Limited, 2017. 246 p.
- [16] Gerrity A., Hu F. SDN/OpenFlow: Concepts and Applications // Network Innovation through OpenFlow and SDN / Hu Fei. Taylor & Francis Group, LLC, 2014. Chapter 1. p. 3-16. DOI: <https://doi.org/10.1201/b16521-3>
- [17] Agouros K. Software Defined Networking. SDN-Praxis mit Controllern und OpenFlow. De Gruyter, 2016. p. 26-48. DOI: <https://doi.org/10.1515/9783110451870-003>
- [18] Agouros K. Software Defined Networking. OpenFlow-Implementierungen. De Gruyter, 2016. p. 49-70. DOI: <https://doi.org/10.1515/9783110451870-004>
- [19] Karyotis V., Khouzani M.H.R. Appendix B – Elements of queuing theory and queuing networks, In Malware Diffusion Models for Wireless Complex Networks, Morgan Kaufmann, Boston, 2016. p. 235-254. DOI: <https://doi.org/10.1016/B978-0-12-802714-1.00023-2>
- [20] Wang S., Lee I. MiniReal: A real SDN network testbed built over an SDN bare metal commodity switch. *Proceedings of 2017 IEEE International Conference on Communications (ICC)*. Paris, France, 2017. p. 1-6. DOI: <https://doi.org/10.1109/ICC.2017.7997283>
- [21] Huang M., Liang W. Incremental SDN-Enabled Switch Deployment for Hybrid Software-Defined Networks. *Proceedings of the 26th International Conference on Computer Communication and Networks (ICCCN)*. Vancouver, BC, Canada, 2017. p. 1-6. DOI: <https://doi.org/10.1109/ICCCN.2017.8038498>
- [22] Goransson P., Black C., Culver T. Software defined networks: A comprehensive approach. 2nd Edition. Elsevier, 2016. 436 p.
- [23] Wang H., Schmitt J. Load balancing - towards balanced delay guarantees in NFV/SDN. *Proceedings of 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. Palo Alto, CA, USA, 2016. p. 240 – 245. DOI: <https://doi.org/10.1109/NFV-SDN.2016.7919504>
- [24] Ndonda G.K., Sadre R. A low-delay SDN-based countermeasure to eavesdropping attacks in industrial control systems. *Proceedings of 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. Berlin, Germany, 2017. p. 1-7. DOI: <https://doi.org/10.1109/NFV-SDN.2017.8169840>

Submitted 26.01.2018; Revised 20.03.2018; Published 30.03.2018.

About the authors:

Konstantin E. Samouylov, Doctor of Technical Sciences, Professor, Head of the Department of Applied Probability and Informatics, Peoples's Friendship University of Russia (6 Miklouho-Maclay St., Moscow 117198, Russia); ORCID: <http://orcid.org/0000-0002-6368-9680>, ksam@sci.pfu.edu.ru

Igor A. Shalimov, Doctor of Technical Sciences, Professor, professor of the Department of Applied Probability and Informatics, Peoples's Friendship University of Russia (6 Miklouho-Maclay St., Moscow 117198, Russia); ORCID: <http://orcid.org/0000-0003-0256-4291>, shalimov579@rambler.ru

Igor G. Buzhin, post-graduate student of the Department of Applied Probability and Informatics, Peoples's Friendship University of Russia (6 Miklouho-Maclay St., Moscow 117198, Russia); ORCID: <http://orcid.org/0000-0001-8224-4065>, bigvrn93@mail.ru

Yuriy B. Mironov, Candidate of Technical Sciences, senior lecturer of the Department of Applied Probability and Informatics, Peoples's Friendship University of Russia (6 Miklouho-Maclay St., Moscow 117198, Russia); ORCID: <http://orcid.org/0000-0003-4790-9951>, mistiam@gmail.com



This is an open access article distributed under the Creative Commons Attribution License which unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).