

# МОДЕЛИ И УПРАВЛЕНИЕ КАЧЕСТВОМ ПРОГРАММНО-ОПРЕДЕЛЯЕМЫХ СЕТЕЙ

В.А. ВИШНЯКОВ, Б.А. МОНИЧ

*Учреждение образования «Белорусская государственная академия связи»,  
ул. Ф. Скорины, 8/2, Минск, 220114, Беларусь*

*Поступила в редакцию 13 февраля 2019*

Приведено понятие качества (QoS) программно-определяемых сетей (ПОС) в облачных технологиях. Рассмотрен подход к описанию модели качества ПОС. Даны два вида услуг QoS: ограничение скорости на входе и гарантирование минимальной пропускной способности на выходе. Определена спецификация OpenFlow для поддержки функционала в области QoS, показаны примеры настройки качества программно-определяемых сетей. Обсуждено взаимодействие модуля прототипа обеспечения QoS с сетевой ОС. Рассмотрена схема обеспечения QoS в OpenFlow-коммутаторе.

*Ключевые слова:* программно-определяемая сеть, качество обслуживания – QoS, модели QoS, поддержка QoS, управление QoS.

## Введение

У организаций возникает необходимость конфигурировать крупномасштабные сети с облачными вычислениями. Упростить эту задачу могут технологии программно-конфигурируемых сетей (Software-Defined Networking – SDN), которые позволяют перевести сетевые элементы под контроль настраиваемого ПО, сделать их более интеллектуальными. Они выполняют согласованное быстрое управление сетями, разрешая управлять всей сетью с одной консоли управления [1]. При этом необходимо решать вопросы обеспечения качества сервиса (QoS). Среди параметров QoS самые важные: **Bandwidth (BW)** – полоса пропускания, которая описывает номинальную пропускную способность среды передачи информации, определяет ширину канала; **Delay** – задержка при передаче пакета; **Jitter** – вариация задержки при передаче пакетов; **Packet Loss** – потери пакетов [2].

Основа реализации качества обслуживания (QoS) базируется на контроле входа и выхода пакета из оконечных устройств. Какая единица данных при этом используется (поток или пакет) – не имеет значения. Реализация QoS сводится к определению приоритетов конкретных пакетов. Контроль над прохождением пакетов через сеть доступен только в пределах центра обработки данных (ЦОД), за пределами ЦОД вся ответственность ложится на провайдеров телекоммуникационных услуг. Протокол OpenFlow позволяет программному обеспечению SDN взаимодействовать с элементами сети – маршрутизаторами и коммутаторами через открытые интерфейсы API. QoS предназначен для решения вопросов: перегрузка каналов и потеря пакетов, которые приводят к снижению производительности приложений. Перегрузка каналов и потери пакетов могут произойти по причинам: превышение максимальной пропускной способности, высокий коэффициент загрузки, а также неправильная конфигурация сетевых устройств [2].

Управление полосой пропускания может уменьшить проблемы, связанные с ограниченной пропускной способностью каналов связи, но только каналов между ЦОД и Интернет. Если пропускная способность ограничена на другом конце канала («последняя миля» провайдера), этот метод не улучшит производительность, так как контроллер OpenFlow не имеет сведений об этих ограничениях. Вне контекста приложений QoS редко дает заметное улучшение в производительности с точки зрения конечного пользователя. Контекст является необходимым для того, чтобы применять правильные методы и политику в нужное время, чтобы обеспечить оптимальную производительность для пользователей конкретного приложения. Согласно концепции SDN вся логика управления выносится в так называемые контроллеры, которые способны отслеживать работу всей сети [3].

## Подход к модели качества ПОС

Для обеспечения функции глобального управления производительностью и качеством обслуживания сети необходима единая модель, в которой будет учитываться топология, полоса пропускания и настройки QoS для каждого потока. Предлагаемая модель применения политик QoS должна основываться на базовых возможностях коммутаторов в области качества обслуживания: ограничителях скорости и статических очередях с приоритетами.

Введем следующие обозначения:  $f_0$  – поток данных,  $rf_0$  – требуемая пропускная способность,  $df_0$  – допустимая задержка. Для обеспечения потоку  $f_0$  требуемой скорости  $rf_0$  необходимо вычислить остаточную пропускную способность  $arf$  как разность максимально доступной пропускной способности  $C$  и суммы текущих (пиковых) скоростей остальных потоков. Затем проводится сравнение: не превышает ли  $rf_0$  остаточной полосы пропускания  $arf$ .

Данное ограничение учитывается на коммутаторе, куда впервые попадает искомый поток данных. Делается это с той целью, чтобы ограничить влияние потока на сеть. Однако при глобальном управлении сетью минимизация задержек на протяжении всей сети представляет собой проблему. В большинстве OpenFlow-коммутаторов на каждый физический порт приходится восемь очередей. Очередь с номером восемь – самая низкоприоритетная.

Предположим, что на одну из очередей ( $q_8$ ) порта восемь ассоциированы шесть потоков  $f_1$ – $f_6$  с произвольными приоритетами (у  $f_1$ – $f_3$  – приоритет  $q = 6$ , у  $f_4$ – $f_6$  –  $q = 4$ ). Добавляется еще один поток  $f_0$ , который ассоциируется с очередью  $q_5$  с приоритетом пять. Задержка потока  $f_0$  –  $df_0$  – зависит от того, насколько быстро прибывают пакеты в очередях с такими же или более высокими приоритетами и насколько быстро канал сможет отправить пакеты. Затребованная пропускная способность порта  $R_{q:8}$  определяется как сумма всех пропускных способностей  $rf_i$ ,  $i = q \dots 8$ . Задача вычисления  $df_0$  определяется как зависимость  $f(q_5, R_{5:8}, C_8)$ , где  $q$  – приоритет очереди ассоциированной с потоком  $f_0$  ( $q_5 = 5$ ),  $R_{q:8}$  – сумма пиковых пропускных способностей  $rf_5 \dots rf_8$  для всех очередей с приоритетом пять и выше.

Каждый коммутатор на пути следования пакета добавляет свой вклад в задержку. Более приоритетные потоки прибавляют задержки менее приоритетным потокам ( $f_4$ – $f_6$  прибавляют задержку потокам  $f_1$ – $f_3$ ). При добавлении  $f_0$  потокам  $f_1$ – $f_3$  задержка еще больше увеличивается. На следующем по пути коммутаторе контроллер может не найти место в очереди для потока  $f_0$ , если другой похожий поток с тем же или более высоким приоритетом на том же физическом порту уже подходит к верхнему пределу максимальной задержки. Контроллер должен учитывать взаимодействие потоков, проходящих по одному пути, и будущее взаимодействие потоков на следующих по пути коммутаторах для решения задачи ассоциации потока с очередью.

Оптимальное распределение потоков по очередям и назначение им приоритетов являются сложными вычислительными задачами. Расчет этих параметров в соответствии с требованиями по задержкам потока  $f_0$  не должен нарушать требования к остальным потокам. Кроме того, необходимо уменьшать время установки новых потоков в таблицу коммутатора. По всему пути следования потока  $f_0$  на каждом исходящем порту ассоциируется очередь. Путь следования и очереди выбираются по минимальной вероятности потери потока. Очередь назначается в соответствии с выбранным приоритетом на конкретном порту. Если приоритет будет установлен в максимум, поток будет иметь наименьшую задержку, но наибольшее влияние на остальные потоки, и наоборот.

На каждом порту необходимо вычислить наивысший возможный приоритет для потока  $f_0$ , при котором еще возможно минимальное влияние на остальные потоки, и минимально возможный приоритет, при котором в текущей ситуации возможно выполнить требования по производительности потока  $f_0$ . Диапазон приоритетов от меньшего к большему назовем  $low_{f_0} \dots hi_{f_0}$ . Для каждого значения приоритета можно вычислить значение задержки в данных условиях для текущих уровней загрузок портов и потоков. Затем проводится серия расчетов для решения задачи минимизации  $\sum(q, R_{q:8}, C) \rightarrow \min$  при  $q \rightarrow \min$  для каждого порта следования потока каждого коммутатора. Особенностью является обратно пропорциональная нелинейная зависимость задержки  $Df$  от приоритета  $q$ .

## Поддержка услуг качества ПОС

Основным элементом концепции SDN является протокол OpenFlow, который обеспечивает взаимодействие контроллера с сетевыми устройствами. Протокол OpenFlow при идентификации

трафика оперирует понятием «поток». Ключевым элементом коммутатора, поддерживающего этот протокол, является таблица потоков (Flow Table). Группа столбцов в левой части таблицы формирует поля соответствия, где указаны характеристики потоков: это могут быть различные параметры, включая MAC- и IP-адреса отправителя и получателя, идентификатор VLAN, номера протокольных портов TCP и UDP, а также другая информация. Эти данные с помощью протокола OpenFlow записывает в таблицу коммутатора контроллер, он же определяет приоритет разных потоков: чем выше приоритет, тем выше соответствующая запись в таблице потоков [3].

Большинство коммутаторов и маршрутизаторов поставляются с некоторой встроенной поддержкой QoS. Некоторые продукты поддерживают только тип DiffServ (RFC-2474 и RFC-2475, небольшое количество очередей и политик), другие поддерживают тысячи очередей и большое количество политик. На практике при реализации QoS в сетях с коммутацией потоков поток отображается на класс трафика QoS, то есть поток получает идентификатор уже имеющегося класса [3].

В настоящее время существует два основных типа QoS услуг: ограничение скорости на входе и гарантирование минимальной пропускной способности на выходе. Первая обычно делается на основе измерений скорости входного порта или потока, после превышения определенной скорости пакеты отбрасываются в соответствии с некоторым алгоритмом. Ограничение минимальной гарантированной полосы пропускания означает, что каждому потоку будет доступна определенная часть от имеющейся пропускной способности. По умолчанию восемь очередей на физическом порту имеют следующие минимальные гарантированные полосы пропускания от целого канала:

- а) первая очередь (low priority): 2 %;
- б) вторая очередь (low priority): 3 %;
- в) третья очередь (normal priority): 30 %;
- г) четвертая очередь (normal priority): 10 %;
- д) пятая очередь (medium priority): 10 %;
- е) шестая очередь (medium priority): 10 %;
- ж) седьмая очередь (high priority): 15 %;
- з) восьмая очередь (high priority): 20 %.

Спецификация OpenFlow 1.0.0 предусматривает поддержку ограниченного функционала в области QoS. На каждый порт можно подключить одну или несколько очередей. Каждый поток ассоциируется с конкретной очередью. Обработка в очереди происходит в соответствии с настройкой конкретной очереди. По умолчанию все очереди имеют настройку на предоставление минимальной гарантированной пропускной способности, другие типы очередей не поддерживаются текущей версией OpenFlow. Для обеспечения QoS в свойствах конкретного потока существует поле IP ToS (Type of Service), в нем содержится шести битный идентификатор DSCP (Differentiated Services Code Point), который используется для обозначения специального байта данных стандартного заголовка IP-пакета и несет информацию о приоритете трафика. Input VLAN Priority служит для приоритезации одной виртуальной сетью (VLAN) над другой. Для обработки потока существуют следующие действия:

- а) `OFPAT_SET_VLAN_PCP` устанавливает приоритет VLAN 802.1q;
- б) `OFPAT_SET_NW_TOS` устанавливает новое значение DSCP потоку;
- в) `OFPAT_SET_QUEUE` устанавливает новый идентификатор очереди для данного потока, т. е. ассоциирует конкретный поток с очередью вне зависимости от DSCP и VLAN Priority.

Для настройки самих очередей QoS, необходимы особые команды, не относящиеся к OpenFlow. Очереди настраиваются средствами операционной системы коммутатора (через специальные команды или посредством внешних утилит). OpenFlow-контроллер может только запросить информацию об очередях на конкретном порту: количество очередей, их тип, количество пакетов в очереди. Также можно запросить статистику по всем очередям коммутатора: переданные пакеты, переданные байты, количество переполнений очереди. Каждая очередь имеет определенный размер в байтах и тип: простая FIFO очередь или с ограничением по гарантированной полосе пропускания. Для второго типа очереди настраивается гарантированная полоса пропускания в процентах (одна единица измерения равна 0.1 процента) от максимально доступной полосы пропускания порта коммутатора. Таким образом, существует два шага для работы с QoS в OpenFlow: настройка очередей на коммутаторах с привязкой к портам или без нее и связывание конкретных потоков.

Существует альтернативный метод обеспечения QoS перенаправить обработку пакетов конкретного потока на операционную систему коммутатора, и тот обработает их классическими методами. Это возможно только на коммутаторах с одновременной поддержкой OpenFlow обработки пакетов и обычной обработки пакетов.

Однако базовая концепция QoS предполагает, что все пакеты всегда идут по одному направлению в течение довольно длительного времени, если топология сети статична, а путь между двумя точками детерминирован. Сети OpenFlow не позволяют применять сложные механизмы QoS именно из-за недетерминированности топологии и путей следования пакетов. К тому же очень часто при глубокой интеграции виртуализированных ресурсов в ЦОД требования к обеспечению необходимых показателей работы сети кардинально меняются в процессе работы для разных виртуальных сетей [5].

### Примеры настройки качества программно-определяемых сетей

В связи с описанными ограничениями реализации QoS предлагается использовать первый подход к реализации – настройка очередей на коммутаторах и сопоставление битов QoS и конкретных потоков на контроллере. Затем всю обработку QoS наиболее рационально перенести на операционную систему коммутатора в соответствии с битами QoS, что позволит максимально использовать все преимущества дополнительных возможностей очередей коммутаторов.

Для корректной настройки параметров QoS необходима точная информация о трафике в сети. В протокол OpenFlow встроена возможность анализа проходящего трафика средствами измерений. Все измерения записываются в таблицу, которая состоит из записей для каждого потока. Измерения для каждого потока позволяют OpenFlow реализовывать различные простые операции QoS, такие как ограничение скорости.

Контроллер может уведомить коммутатор о различных типах обработки очередей:

а) удаление (отбрасывание) пакета. Этот тип обработки может быть использован для определения диапазона ограничения скорости.

б) изменение DSCP. Этот тип позволяет уменьшать приоритет DSCP в IP-заголовке пакета. Используется для определения простого ограничителя DiffServ.

Для управления произвольным коммутатором OpenFlow любого производителя существует linux-утилита `dpctl`, позволяющая добавлять, удалять, просматривать данные о потоках, очередях, счетчиках. Для реализации QoS на уровне OpenFlow (программном уровне) необходимо создать очередь и настроить ее параметры. Так как очередь в OpenFlow может быть только с политикой WFQ [4] (Weighted Fair Queue, взвешенная справедливая очередь) с нижней границей скорости потока, команда для настройки очереди для третьего приоритета на втором порту коммутатора IP = 10.1.1.1 для минимальной гарантированной скорости 1000 Кбит/с будет выглядеть следующим образом:

```
#dpctlmod-queue tcp:10.1.1.1:6633 3 2 1000.
```

Для классификации конкретного потока можно использовать как параметры ранее присвоенного ToS/DSCP, так и присвоить эти параметры непосредственно OpenFlow препроцессором или перенаправить в нужную очередь (локальная классификация). Правило для приоритезации (ассоциации с третьей очередью) пакета от 10.0.0.2 к 10.0.0.3 выглядит следующим образом:

```
#dpctl add-flow tcp:10.1.1.1:6633 "ip, nw_dst=10.0.0.3, nw_src=10.0.0.2, action=mod_vlan_pcp:3, output:23",
```

где `mod_vlan_pcp` задает приоритет пакета (номер очереди) при обработке.

При необходимости изменить ToS можно использовать параметр

```
mod_nw_tos:<tos>;
```

где `output` задает номер исходящего порта (коммутация).

Минимальная гарантированная скорость на порту задается при дополнительной настройке физической очереди. Перенаправить пакет в физическую очередь №3 порта 23 коммутатора можно командой:

```
#dpctl add-flow tcp:10.1.1.1:6633 "ip, nw_dst=10.0.0.3,nw_src=10.0.0.2, action=enqueue:23:3, output:23"
```

где `enqueue` – номер очереди конкретного порта, `enqueue` поддерживается не всеми коммутаторами.

Для создания ограничения сверху конкретного потока используется следующая запись другой утилиты, входящей в состав продукта OpenvSwitch:

```
#ovs-ofctl add-limiter tcp:10.1.1.1:975 123 drop 512 kbps,
```

где 123 – номер ограничителя; drop – реакция на превышение лимита; 512 kbps – размер виртуального канала.

После создания ограничителя необходимо связать конкретный трафик с ним:

```
#ovs-ofctl ladd-flow tcp:10.1.1.1:975 "priority=32769,idle_timeout=0,ip, nw_dst=10.0.0.3,  
nw_src=10.0.0.2, action=rate_limit:123,output:23,
```

где rate\_limit – номер ограничителя.

Как видно из приведенных примеров, настройка QoS для большой сети с множеством потоков и динамической топологией является нетривиальной задачей, к тому же ограниченной ручным созданием и настройкой очередей.

В больших сетях контроллеру нет возможности управлять выделением ресурсов каждому потоку трафика и применять QoS на конкретный поток данных. Коммутаторы имеют ограничение на количество записей в таблице OpenFlow, поэтому целесообразно использовать группировку (агрегацию) нескольких потоков в единую логическую группу и управление выделением ресурсов на эту группу. Большинство QoS политик классификации работают именно по этому принципу. В OpenFlow существуют понятия aggregative VLAN и aggregative flow, которые позволяют применять правила агрегации на уровне коммутатора и контроллера, не теряя при этом данные отдельного потока.

Контроллер должен постоянно проводить мониторинг всей сети с целью обеспечения глобального управления ресурсами. Например, все коммутаторы могут пересылать LLDP, DHCP, ARP пакеты на контроллер, который всегда будет иметь у себя полную актуальную топологию сети со всеми связями и конечными устройствами. При необходимости контроллер может рассчитать места внедрения или удаления политик QoS, оптимальные маршруты с учетом применения QoS и требований приложений.

Кроме того, предлагается регулярно делать копию таблиц OpenFlow с коммутаторов и проводить их анализ для выявления несоответствий. Дополнительно во многих коммутаторах предоставляется возможность запроса QoS конфигурации.

### **Структура управления качеством ПОС**

Схема взаимодействия модуля прототипа обеспечения QoS с СОС и обязательными компонентами системы работает следующим образом. Управление модулем обеспечения QoS осуществляется через веб-интерфейс, предоставляемый стандартными средствами веб-приложений платформы Django. Через веб-интерфейс возможно добавление, изменение и просмотр сведений о потоках, для которых будет гарантированы параметры QoS. Модуль обеспечения QoS регулярно выполняет проверку изменений в БД и вносит изменения в таблицу потоков OpenFlow для заданных передач данных. Контроллер NOX с подключенными модулями snmp, topology выполняет функции управления, модуль pyswitch имеет дополнение в виде вызова модуля обеспечения QoS при добавлении нового потока в коммутатор. Система сбора статистики MRTG по протоколу SNMP собирает статистику с коммутаторов (загрузку портов, размеры очередей), через интерфейс модуля NOXsnmp эти сведения заносятся в систему и используются модулем обеспечения QoS для расчета параметров потоков.

Используются два действия для обеспечения QoS в OpenFlow: настройка очередей на коммутаторах и связывание конкретных потоков. Существует альтернативный метод обеспечения QoS – передача обязанностей по обработке пакетов конкретного потока операционной системе коммутатора. Это возможно только на коммутаторах, поддерживающих как традиционную обработку пакетов, так и обработку с помощью OpenFlow в соответствии с рис. 1.

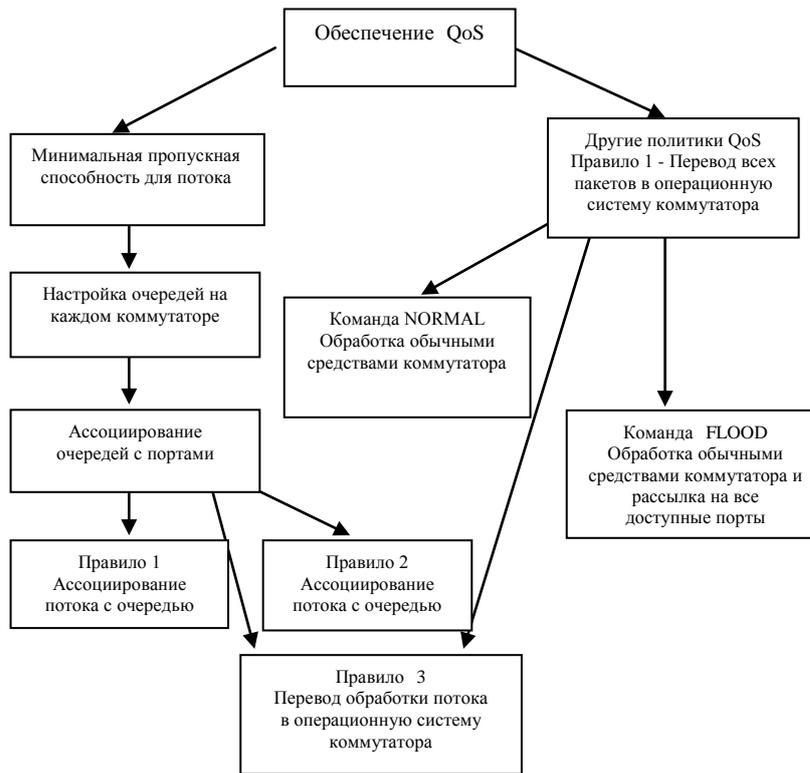


Рис. 1. Структура обеспечения QoS в OpenFlow коммутаторе

Однако базовая концепция QoS предполагает, что все пакеты всегда идут по одному направлению в течение длительного времени, топология сети статична, путь между двумя точками детерминирован. Сети OpenFlow не позволяют применять сложные механизмы QoS именно из-за недетерминированности топологии и путей следования пакетов. К тому же, как правило, при глубокой интеграции виртуализированных ресурсов в ЦОД требования к обеспечению необходимых показателей работы сети меняются в процессе работы кардинально для разных виртуальных сетей.

В связи с описанными ограничениями реализации QoS предлагается использовать первый подход к реализации – настройка очередей на коммутаторах и сопоставление битов QoS и конкретных потоков на контроллере. Затем всю обработку QoS наиболее рационально перенести на операционную систему коммутатора в соответствии с битами QoS, что позволит максимально использовать все преимущества дополнительных возможностей очередей коммутаторов.

### Заключение

Основа реализации качества обслуживания (QoS) базируется на контроле входа и выхода пакетов из устройства. Реализации QoS сводится к определению приоритетов конкретных пакетов. Контроль над прохождением пакетов через сеть доступен только в пределах центра обработки данных (ЦОД), за пределами ЦОД вся ответственность ложится на провайдеров телекоммуникационных услуг. Протокол OpenFlow позволяет программному обеспечению SDN взаимодействовать с элементами сети – маршрутизаторами и коммутаторами – через открытые интерфейсы API.

Модель применения политик QoS основывается на базовых возможностях коммутаторов в области качества обслуживания: ограничителях скорости и статических очередях с приоритетами. Ограничитель скорости в OpenFlow коммутаторах реализован с некоторыми отличиями от RSCP в обычных коммутаторах. На каждом порту необходимо вычислить наивысший возможный приоритет для потока  $f_0$ , при котором еще возможно минимальное влияние на остальные потоки, и минимально возможный приоритет, при котором в текущей ситуации возможно выполнить требования по производительности потока  $f_0$ .

Рассмотрена структура управления качеством ПОС. Управление модулем обеспечения QoS осуществляется через веб-интерфейс, возможно добавление, изменение и просмотр сведений о потоках, для которых будет гарантированы параметры QoS. Модуль обеспечения QoS регулярно

выполняет проверку изменений в БД и вносит изменения в таблицу потоков OpenFlow для заданных потоков данных. Контроллер NOX с подключенными модулями snmp, topology выполняет свои функции управления, модуль rpswitch имеет дополнение в виде вызова модуля обеспечения QoS при добавлении нового потока в коммутатор. Система сбора статистики MRTG по протоколу SNMP собирает статистику с коммутаторов (загрузку портов, размеры очередей), через интерфейс модуля NOXsnmp эти сведения заносятся в систему и используются модулем обеспечения QoS для расчета параметров потоков.

## MODELS AND QUALITY MANAGEMENT OF SOFTWARE-DEFINED NETWORKING

U.A. VISHNIAKOU, B.A. MONICH

### Abstract

The concept of quality (QoS) of software-defined networks (SDN) in cloud technologies is carried out. Considered the approach to quality model of SDN. Two QoS services are discussed: the speed limit on the entrance and guaranteeing minimum bandwidth on the output. The specification of OpenFlow for support of functionality in the field of QoS is defined. The quality setting of software-defined networks is shown. The description of the interaction module of the prototype providing QoS with network OS is discussed. The scheme for providing QoS in the OpenFlow switch is done.

### Список литературы

1. Программно определяемые сети (Software Defined Networks): настоящее и будущее [Электронный ресурс]. – Режим доступа : <https://habr.com/en/company/hpe/blog/160531>. – Дата доступа : 15.01.2019.
2. Дубинин, В. Программно-определяемые сети: от концепции к технической реализации / В. Дубинин // ITWeek. – 2016. – № 5. – С. 15–20.
3. SDN: кому и зачем это надо [Электронный ресурс]. – Режим доступа : <https://www.osp.ru/lan/2012/12/13033012>. – Дата доступа : 15.01.2019.
4. Логинов, С. С. Об уровнях управления в программно-конфигурируемой сети (SDN) [Электронный ресурс]. – Режим доступа : <https://cyberleninka.ru/article/n/ob-urovnyah-upravleniya-v-programmno-konfiguriruemoj-seti-sdn>. – Дата доступа : 15.01.2019.
5. Вишняков, В. А. Технология программно-определяемых сетей в облачных вычислениях / В. А. Вишняков, Б. А. Монич // Проблемы инфокоммуникаций. – 2018. – № 2. – С. 70–75.