

УДК 621.391

Е.Б. ТКАЧЕВА, канд. техн. наук, ст. преп., ХНУРЭ, Харьков,
ИССАМ СААД, асп., ХНУРЭ, Харьков,
МОХАММЕД ДЖАМАЛ САЛИМ, асп., ОНАС им. А. С. Попова, Одесса

ПРИМЕНЕНИЕ АППАРАТА Е-СЕТЕЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ АНАЛИЗА И ВЕРИФИКАЦИИ ПРОГРАММНО- КОНФИГУРИРУЕМЫХ СЕТЕЙ

В статье предложены формализмы, позволяющие описывать набор основных свойств программно-конфигурируемых сетей. В рамках задачи анализа и верификации предложен метод, который базируется на модельном подходе и анализе последовательности смены состояний элементов модели сети. В качестве аппарата моделирования предложено использовать аппарат Е-сетей. Проводится анализ таких свойств модели, как достижимость, ограниченность, живость. Ил.: 1. Библиогр.: 11 назв.

Ключевые слова: задача анализа и верификации, программно-конфигурируемые сети, моделирование, Е-сеть.

Постановка проблемы и анализ литературы. Как правило, для успешного функционирования сетевой инфраструктуры требуется выполнение множества одновременных задач: от маршрутизации и мониторинга передаваемых данных, до контроля доступа и распределения нагрузки на сетевые элементы. Парадигма программно-конфигурируемых сетей (Software Defined Network, SDN) призвана упростить процесс передачи данных и управление сетевой инфраструктурой за счет логического разделения сети на подуровни: уровень управления и уровень передачи данных. SDN представляет собой распределенную систему, где один или несколько контроллеров управляют множеством коммутаторов, отвечающих за передачу пакетов между сетевым оборудованием.

Как и в традиционных сетях передачи данных, регулирование обмена сообщениями и согласованное взаимодействие между элементами сети, построенной на основе концепции SDN, обеспечивается сетевыми протоколами. Все функциональные возможности, гарантирующие выполнение требований, а также порядок взаимодействия контроллеров и коммутаторов SDN определяются протоколом OpenFlow [1].

Значения служебных полей протокола OpenFlow могут существенно отличаться в зависимости от требований сетевой инфраструктуры и сервисов, функционирующих в сети. На сегодняшний день протокол OpenFlow не имеет конечной реализации и основывается на ряде спецификаций, которые имеют значительные различия между собой.

Учитывая данные особенности, в процессе разработки и внедрения

© Е.Б. Ткачева, Иссам Саад, Мохаммед Джамал Салим, 2015

SDN, может возникнуть ряд несоответствий, приводящих к возникновению критических ошибок в процессе функционирования сети. В рамках устранения ошибок и проверки корректной работы протокола проводится его полный анализ и верификация на стадии проектирования.

Применение формальных методов в рамках задачи анализа и верификации протокола OpenFlow позволяет четко описать набор свойств, которыми должен обладать протокол, проверить их выполнимость и соответствие требованиям. Преимуществом формальных методов является то, что формализмы согласования и взаимодействия процессов, соответствующих корректной работе протокола OpenFlow, могут быть использованы в последующих разработках как прототипы, что позволяет существенно сократить временные и материальные затраты. Поэтому задача анализа и верификации сетей, построенных на основе концепции SDN, является актуальной.

На сегодняшний день существует небольшое количество научных трудов, посвященных формальным методам анализа и верификации SDN. В работе [2] приведена модель распределенной сети с описанием метода проверки свойств достижимости и принципов маршрутизации пакетов, который базируется на двоичных разрешающих диаграммах. В работах [3, 4] в рамках задачи анализа свойств достижимости применены методы булевой алгебры и преобразования ДНФ. В работах [5, 6] модель SDN представлена в виде системы переходов ВВГ, которая изменяет свое состояние в случае активности коммутатора.

Недостатком решений, предложенных в данных работах, является то, что модели SDN позволяют описывать поведение сети без учета характеристик контроллера, моделируя лишь сеть коммутаторов. Это позволяет проводить анализ и верификацию лишь моментальных сетевых конфигураций.

Целью статьи является определение основных свойств, которыми должна обладать сетевая инфраструктура SDN, установление критериев проверки их соблюдения в готовых реализациях SDN и разработка метода, позволяющего выполнить анализ и верификацию свойств на стадии разработки.

Постановка задачи. Рассмотреть ряд основных требований, предъявляемых к SDN, а также формально представить набор основных свойств протокола OpenFlow, которые гарантируют его корректную работу. Разработать метод анализа основных свойств протокола OpenFlow, который базируется на построении цепочек смены состояний протокола.

Построение сети на основе протокола OpenFlow: основные требования и их формализация. Протокол OpenFlow [1, 7] представляет собой новый стандарт обмена данными на уровне управления в SDN. Ключевыми элементами протокола является OpenFlow коммутатор, который выступает в роли пользовательской стороны и OpenFlow контроллер, который представляет собой централизованное устройство управления. OpenFlow коммутатор содержит таблицу переадресации и ряд правил, в соответствии с которыми передаются управляющие пакеты. SDN контроллеры содержат набор инструкций, которые устанавливают пути и методы доставки и обработки информации между конечными пользователями.

Сети, построенные на основе концепции SDN, представляют собой хорошо масштабируемую инфраструктуру, которая может быть эффективно адаптирована под любые сетевые задачи. На сегодняшний день концепция SDN применяется для построения распределенных сетей учебных заведений (сети "кемпус"), центров обработки данных, интеллектуальных распределенных сетей (сети с распределенными хранилищами данных Google).

Описание процессов и формализация обмена сообщениями. Структура сети SDN на уровне управления основывается на программном обеспечении, в частности, на программных реализациях SDN контроллера и OpenFlow коммутаторов. Такой подход позволяет значительно расширить функциональность процессов обработки и передачи информации, но, наряду с этим, повышает вероятность того, что логические ошибки в программной реализации могут привести к небезопасному или некорректному поведению сети в целом или отдельных ее элементов.

Независимо от структуры и масштабов сетевой инфраструктуры должны быть соблюдены все требования, соответствующие заданному уровню качества обслуживания. Для проверки и устранения подобных ошибок, необходимо проведение анализа, а в последующем и верификации программно-сетевой инфраструктуры на такие свойства:

– отсутствие потерь в процессе маршрутизации и переадресации пакета: любой пакет, который был отправлен, в конечном итоге должен быть доставлен получателю;

– отсутствие петель и зацикливаний: любой пакет должен быть доставлен непосредственно получателю, который идентифицирован в поле destination source таблицы переадресации;

– живость сети: все элементы сети, участвующие в передаче данных, должны быть доступны и активны;

- достижение заключительного состояния: все сетевые состояния, гарантирующие корректную работу протокола, должны быть достигнуты;
- эффективное формирование путей передачи данных: поиск оптимального пути передачи данных, учет изменения маршрутов передачи и мобильности конечных пользователей;
- безопасность: отсутствие несанкционированных процессов при функционировании протокола;
- гарантирование заданного уровня Quality of Services для каждого состояния.

Соблюдение первых трех свойств обеспечивает безотказную и корректную работу сети (протокола OpenFlow), гарантируя предоставление сервисов. Эффективное формирование путей передачи данных, безопасность передачи данных и обеспечение заданного качества обслуживания являются индивидуальными свойствами SDN: зависят от программной реализации контроллеров и OpenFlow коммутаторов, а также от версии самого протокола OpenFlow.

Обзор готовых SDN решений [7, 8] показал, что большинство ошибок, приводящих к некорректному функционированию сети в целом, возникает из-за различной интерпретации стандартов OpenFlow; логических ошибок проектирования; неполного соответствия реализации предъявляемым требованиям. В целях выявления и устранения подобного рода ошибок необходимо проведение анализа и верификации готовых решений.

Для проведения полноценного анализа, в первую очередь, необходимо выделить множество процессов P (включая терминальные P_t), которые определены в спецификации OpenFlow и соответствуют взаимодействию возможных элементов сети S_{SDN} (смене их состояний). Соответствие смены процессов изменению состояний можно представить в виде зависимости $f : P \rightarrow S_{SDN}$.

Так как сети, построенные на основе концепции SDN, имеют динамическую природу, то каждый набор элементов OpenFlow протокола S'_{SDN} из множества S ($S'_{SDN} \in \{S\}$), соответствует индивидуальному набору процессов P' , $P' \in \{P\}$. Логическая система, определяющая взаимодействие элементов OpenFlow и последовательности смены их состояний (f, S_{SDN}, P) , позволяет выполнить пошаговую проверку соответствия протокола или предоставляемого сервиса тем требованиям, которые к нему предъявляются. При этом соответствие требуемых свойств гарантируется тогда, когда, соблюдается требование

$$(f, S_{SDN}, P)_{\text{спец}} \equiv (f, S_{SDN}, P)_{\text{мод}}, \quad (1)$$

где $(f, S_{SDN}, P)_{\text{спец}}$ – множество, определяющее взаимосвязь состояний элементов OpenFlow протокола и последовательность их смены, соответствующее требованиям одной из версий спецификации; $(f, S_{SDN}, P)_{\text{мод}}$ – множество, определяющее взаимосвязь и последовательности смены состояний модели реализации OpenFlow протокола.

Также в процессе разработки и внедрения сетей, построенных на основе концепции SDN, необходима проверка соответствия и непротиворечивости требований, предъявляемых в различных версиях спецификаций OpenFlow [1, 7, 8]. Формально проверка спецификации на непротиворечивость команд и правил, заложенных в логику работы контроллера, может быть представлена следующим образом

$$(f, S_{SDN}, P)'_{\text{спец}} \equiv (f, S_{SDN}, P)''_{\text{спец}}, \quad (2)$$

где $(f, S_{SDN}, P)'_{\text{спец}}$ – множество, определяющее взаимосвязь состояний элементов OpenFlow протокола и последовательность их смены, которое соответствует требованиям спецификации версии 1.1.0; $(f, S_{SDN}, P)''_{\text{спец}}$ – множество, определяющее взаимосвязь состояний элементов OpenFlow протокола и последовательность их смены, которое соответствует требованиям спецификации версии 1.3.0.

SDN по своей природе относятся к сложным распределенным системам, количество элементов которых может значительно варьироваться. В рамках формального представления систем такого класса применяется модельный подход. Его преимуществом является четкая формализация элементов и их взаимодействия, а также учет структуры (топологии) и поведенческих свойств процессов.

Применение аппарата E-сетей для формального представления и анализа процессов протокола OpenFlow. В качестве моделей, отображающих смену состояний элементов SDN (функционирования протокола OpenFlow), предлагается использовать математический аппарат E-сетей [9], который обладает рядом преимуществ: позволяет проводить моделирование процессов синхронной и асинхронной природы. Наличие множества инвариантов меток позволяет учитывать особенности функционирования протокола OpenFlow,

функционирование которого определяется набором определенных атрибутов сети.

Аппарат E-сетей может быть применен в рамках решения задач разного уровня абстракции: как на уровне изменения таблиц маршрутизации на каждом отдельном OpenFlow коммутаторе (FlowTable) и его взаимодействии с SDN контроллером, так и в целях анализа маршрутов передачи данных, которые определяются правилами коммутации. При использовании аппарата E-сетей условия и выполнения команд (событий, имеющих место на уровне управления) отображаются (моделируются) переходами, изменение состояний коммутаторов и контроллера отображаются позициями.

Недостатком E-сетей является отсутствие возможности установления загруженности контроллера, а также некоторых характеристик, связанных с временем выполнения – момент обработки пакета, задержка, возникающая при обработке пакета. Однако, в рамках поставленных задач, данный фактор не учитывается. В качестве примера приведем модель E-сети, которая соответствует первичной стадии обмена управляющими сообщениями между контроллером и коммутатором (см. рис.). В связи с тем, что рисунок взят из пакета, на нем вместо индексных символов используются строчные.

Состояние *Switch learning (porti)* соответствует приходу нового сообщения на один из портов коммутатора (*porti*), об изменениях в топологии сети или добавлении нового пользователя. При этом перенаправление нового управляющего пакета контроллеру осуществляется в том случае, если в таблице переадресации коммутатора не найдено ни одного совпадения для поля *Match_j* (состояние *Don't find*). Если совпадения обнаружены (состояние *Find*), то коммутатор пересылает сообщение адресату, который указан в поле *destination source* таблицы переадресации (состояние *Forwarding data*) или осуществляет обработку сообщения (смена счетчиков, приоритета, адресата) – состояние *ChangeTable*. Состояние *ChangeCommand* соответствует передаче обработанного пакета коммутатору, которое может осуществляться напрямую или с помощью *FlowVisor* (состояние *FlowVisor*) посредством инкапсуляции различных пакетов управляющей информации в единый канал.

Получение пакета OpenFlow коммутатором осуществляется тем портом, который указан в заголовке *Match_j* (состояние *Receive porti*). Стоит отметить, что в соответствии со спецификацией OpenFlow пакеты могут быть доставлены как коммутатору, который инициировал установления соединения, так и всем коммутаторам сети, такую возможность в модели E-сети определяет состояние *Receive all ports*.

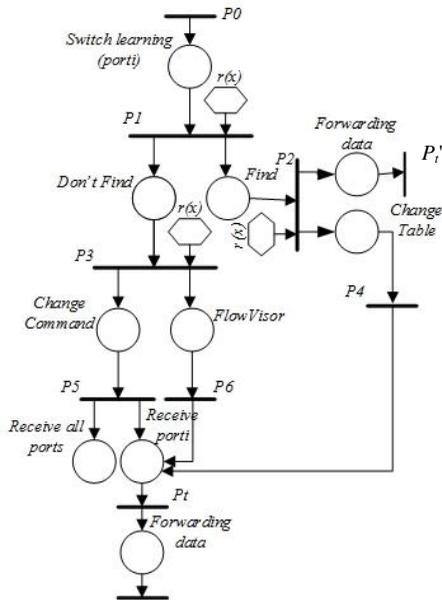


Рис. Модель процесса инициализации управляющего потока в сети по протоколу OpenFlow (на рис. вместо индексных символов используются строчные)

Переход P_0 модели E-сети соответствует процессу поступления нового сообщения (пакета) от устройств, находящихся на уровне передачи данных. Переход P_1 является управляемым, он соответствует процессу "обучения" коммутатора. Управляющий предикат $r(x)$ содержит информацию о данных, занесенных в таблицу переадресации. Здесь работает следующее требование: пакет пересылается контроллеру только в том случае, если в таблице переадресации коммутатора не найдено ни одного совпадения в уже имеющемся поле $Match_j$: $P_1 \rightarrow Tf(Match_j \notin Match_{Tf})$. Процесс P_2 также содержит предусловия выполнения, в соответствии с которыми либо осуществляется его передача адресату, либо последующая обработка полей и передача адресату – этому процессу соответствует переход P_4 . Переход P_3 соответствует процессу обработки пакета контроллером с дальнейшей его передачей по тому же маршруту (переход P_5) либо инкапсулируя его

посредством Flowvisor (переход P_6). Переход P_i и P_i' соответствует процессу обработки полученного от контроллера пакета с набором команд, которые должны быть к нему применены: занесение в таблицу переадресации, модификация $Tf(f_{i_{din}}(Modif, t))$ или отбрасывание пакета $Tf(Drop)$.

Таким образом, аппарат E-сетей позволяет представить каждый процесс, соответствующий функционированию протокола OpenFlow с необходимым уровнем детализации. Свойства E-сетей широко применяются для анализа таких свойств, как живость, безопасность, сохраняемость, которые соответствуют необходимым свойствам сетей, построенных на основе концепции SDN.

Проведем анализ свойств модели E-сети, приведенной на рис. Сначала формально представим множество всех возможных процессов вида $f : P \rightarrow S_{SDN}$. Данное множество соответствует возможной последовательности срабатывания переходов:

$f_1 : P_0 P_1 P_3 P_3 P_t$ – последовательность соответствует достижению состояния *Forwarding data*;

$f_2 : P_0 P_1 P_2 P_t'$ – последовательность соответствует достижению состояния *Forwarding data*;

$f_3 : P_0 P_1 P_2 P_4 P_t$ – последовательность соответствует достижению состояния *Forwarding data*;

$f_4 : P_0 P_1 P_3 P_6 P_t$ – последовательность соответствует достижению состояния *Forwarding data*;

$f_5 : P_0 P_1 P_3 P_5$ – последовательность соответствует достижению состояния *Receive all ports*.

Последовательности переходов $f_1 - f_4$ соответствуют достижению заключительного состояния процесса инициализации. Модель сети удовлетворяет свойствам живости (каждый переход может быть запущен), все состояния сети достижимы, однако последовательность f_5 приводит в возникновению дублирования меток, что соответствует небезопасности сети, а, следовательно, некорректной работе.

Возникновение подобной ситуации обусловлено тем, что спецификация протокола OpenFlow не имеет на сегодняшний день четких инструкций, которые описывают тип доставки управляющих сообщений (наличие multicast рассылки контроллером, правила приема, отсутствие поддержки протокола IPv6). В данном примере анализ фрагмента сети

показал, что при multicast рассылке, процесс инициализации заканчивается некорректно.

Принципы работы FlowVisor. Чаще всего концепция SDN применяется для сетей средних и крупных масштабов. Примерами таких сетей служат Европейская сеть Ofelia, сеть США Geni, сети корпораций Google и Amazon. В целях эффективного использования ресурсов крупной распределенной сети под разные сервисы и сетевые задачи применяется технология виртуализации, которая позволяет группировать несколько OpenFlow в одну логическую сеть [8].

Виртуализация позволяет повысить эффективность распределения сетевых ресурсов и сбалансировать нагрузку на них; изолировать потоки разных пользователей и приложений в рамках одной физической сети; администраторам разных срезов использовать свои политики маршрутизации и правила управления потоками данных; проводить эксперименты в сети, используя реальную физическую сетевую инфраструктуру; использовать в каждом срезе только те сервисы, которые необходимы конкретным приложениям. В соответствии со спецификацией протокола OpenFlow каждая подобная логическая сеть (network slices), может быть выделена по ряду признаков; топологии, пропускной способности, производительности оборудования и др [10].

Одним из примеров виртуализации ресурсов SDN, разделения сети на срезы и управления ими является FlowVisor – программа-посредник (проxy) [11], действующая на уровне между OpenFlow-коммутаторами и различными контроллерами SDN. Посредством FlowVisor можно создавать логические сегменты сети, использующие разные алгоритмы управления потоками данных, обеспечивая изоляцию данных сетей друг от друга.

Каждый контроллер SDN управляет только своей логической сетью и не может оказывать влияния на функционирование других. Разделение сети на логические подсети является достаточно трудоемкой задачей. Процесс усложнен отсутствием единой логики построения и формирования каналов, методов изоляции потоков передаваемых данных, определение свойств Flowvisor (формирование дополнительных соединений, используемые протоколы шифрования).

FlowVisor, как и любое программное сетевое решение, имеет свои особенности. Модификация и конфигурация срезов зависят от логики программного кода, сетевых характеристик и версии спецификации. На сегодняшний день работа FlowVisor полноценно поддерживается только версией протокола OpenFlow 1.1, реализация остальных версий зачастую содержит ошибки из-за неточностей спецификации.

Рассмотрим пример формирования срезов. Формальное представление процессов и их последующий анализ.

Конфигурация срезов происходит в соответствии с определенными правилами [8, 10], которые зависят от требований к виртуализации, а также спецификации протокола. Базовыми правилами FlowVisor для каждого среза являются [11]:

- данные (управляющая информация) доступна только для чтения и пересылки;
- разрешена модификация данных;
- данные необходимо отбросить.

В дальнейшем срезы формируются в соответствии с правилами и политиками, заложенными при работе SDN. Однако, некоторые правила в зависимости от спецификации могут перекрывать друг друга, что противоречит правилам изоляции трафика.

Некоторые последовательности могут описывать процессы образования петель, как следствие некорректной работы протокола. Причиной возникновения петель и отбрасывания пакетов при недостаточной пропускной способности канала логического среза служит отсутствие механизма своевременного предупреждения SDN контроллера о переполнении и отсутствие управляющих команд. Данная проблема связана с отсутствием четких требований к работе FlowVisor в спецификациях протокола OpenFlow.

Выводы. В соответствии с поставленной целью проведен обзор свойств, которыми должны обладать сети, построенные на основе концепции SDN. Установлено, что основными свойствами, которые соответствуют корректному функционированию сетевой инфраструктуры, являются живость, отсутствие тупиков и зацикливаний. В результате проведенной работы предложены формализмы, которые позволяют выполнить пошаговую проверку соответствия реализации SDN или ее составляющих предъявляемым требованиям и установить возможность возникновения противоречий.

В качестве метода анализа свойств SDN и их основных компонентов предложено применение аппарата E-сети и поведенческих цепочек. Данный подход позволяет провести анализ всех заданных и возможных граничных состояний элементов сети на стадии разработки.

Анализ модели E-сети процесса инициализации управляющего потока показал, что ряд действий приводит к возникновению тупиков: $f_5 : P_0 P_1 P_3 P_5$. Было выявлено, что причиной из возникновения является отсутствие точных требований спецификации о доставке управляющих команд от контроллера к коммутатору OpenFlow.

Таким образом, формализация свойств и анализ протокола OpenFlow с помощью аппарат Е-сетей позволяет выявить ряд ошибок, которые приводят к некорректной работе протокола.

Список литературы: 1. *Egawa T.* SDN standardization Landscape from ITU-T Study Group / *T. Egawa* // ITU Workshop on SDN, 4 June 2013. – Geneva: GTU, 2013. – P. 41-46. 2. OpenFlow Switch Specification (Series) [Electronic resource] // Open Networking Foundation. – 2014. – 120 p. – Режим доступа: <https://www.opennetworking.org/sdn-resources/onf-specifications/openflow>. 3. *Kim H.* Improving network management with software-defined networking / *H. Kim, N. Feamster* // Communications Magazine. IEEE. – 2013 – № 6 – P. 114-119. 4. *Al-Shaer E.* Network Configuration in a Box: Toward End-to-End Verification of Network Reachability and Security / *E. Al-Shaer, W. Marrero, A. El-Atawy* // 17th IEEE International Conference on Network Protocols (ICNP'09) – New Jersey: BTH, 2009. – P. 123-132. 5. *Mai H.* Debugging of the Data Plane with Ant eater / *H. Mai, A. Khurshid, R. Agarwal* // Proceedings of the ACM SIGCOMM conference. – NY: CU, 2011. – P. 290-301. 6. *Мухайлов Н.К.* Концепции системного анализа и проектирования базовых телекоммуникационных средств единого информационного пространства Украины / *Н.К. Мухайлов* // Праці УНДІРТ. – 2004. – № 2 (38). – С. 3-10. 7. Отчёт о НИР по теме: Создание прототипа отечественной ПКС платформы управления сетевыми ресурсами и потоками с помощью сетевой операционной системы (СОС) на основе анализа и оценки существующих сетевых операционных систем для ПКС сетей и выбора одной из них для последующего. – М.: МГУ им. М.В. Ломоносова, 2014. 8. FlowVisor: A Network Virtualization Layer [Electronic resource] // Open Networking Foundation – 2009. – 56 p – Режим доступа: <http://archive.openflow.org/downloads/technicalreports/openflow-tr-2009-1-flowvisor.pdf>. 9. *Iversen V.B.* Teletraffic engineering and network planning / *V.B. Iversen*. – DTU Course 34340, 2010. – 623 p. 10. *Лосев Ю.И.* Применение методов анализа Е-сетей к моделям СОД / *Ю.И. Лосев, С.И. Шматков, Е.В. Дуравкин* // Радиотехника: Всеукр. межвед. науч.-техн. сб. – 2002. – Вып.132. – С. 149-151. 11. *Oppenheimer P.* Top-Down Network Design. Third Edition / *P. Oppenheimer*. – Cisco Systems, Inc., 2011. – 447 p.

Bibliography (transliterated): 1. *Egawa T.* SDN standardization Landscape from ITU-T Study Group / *T. Egawa* // ITU Workshop on SDN, 4 June 2013. – Geneva: GTU, 2013. – P. 41-46. 2. OpenFlow Switch Specification (Series) [Electronic resource] // Open Networking Foundation. – 2014. – 120 p. – Режим доступа: <https://www.opennetworking.org/sdn-resources/onf-specifications/openflow>. 3. *Kim H.* Improving network management with software-defined networking / *H. Kim, N. Feamster* // Communications Magazine. IEEE. – 2013 – № 6 – P. 114-119. 4. *Al-Shaer E.* Network Configuration in a Box: Toward End-to-End Verification of Network Reachability and Security / *E. Al-Shaer, W. Marrero, A. El-Atawy* // 17th IEEE International Conference on Network Protocols (ICNP'09) – New Jersey: BTH, 2009. – P. 123-132. 5. *Mai H.* Debugging of the Data Plane with Ant eater / *H. Mai, A. Khurshid, R. Agarwal* // Proceedings of the ACM SIGCOMM conference. – NY: CU, 2011. – P. 290-301. 6. *Mihajlov N.K.* Koncepcii sistemnogo analiza i proektirovanija bazovih telekommunikacionnih sredstv edinogo informacionnogo prostranstva Ukrainy / *N.K. Mihajlov* // Praci UNDIRT. – 2004. – № 2 (38). – S. 3-10. 7. Otchjot o NIR po teme: Sozdanie prototipa otechestvennoj PKS platformy upravlenija setevymi resursami i potokami s pomoshh'ju setevoj operacionnoj sistemy (SOS) na osnove analiza i ocenki sushhestvujushhijh setevyh operacionnih sistem dlja PKS setej i vybora odnoj iz nih dlja posledujushhego. – М.: МГУ им. М.В. Ломоносова, 2014. 8. FlowVisor: A Network Virtualization Layer [Electronic resource] // Open Networking Foundation – 2009. – 56 p – Режим доступа: <http://archive.openflow.org/downloads/technicalreports/openflow-tr-2009-1-flowvisor.pdf>. 9. *Iversen V.B.* Teletraffic engineering and network planning / *V.B. Iversen*. – DTU

Course 34340, 2010. – 623 p. **10.** *Losev Ju.I.* Primenenie metodov analiza E-setej k modeljam SOD / *Ju.I. Losev, S.I. Shmatkov, E.V. Duravkin* // Radiotekhnika: Vseukr. mezhdved. nauch.-tehn. sb. – 2002. – Vyp.132. – S. 149-151. **11.** *Oppenheimer P.* Top-Down Network Design. Third Edition / *P. Oppenheimer.* – Cisco Systems, Inc., 2011. – 447 p.

Поступила (received) 10.03.2015

Статью представил д-р техн. наук, проф. Харьковского национального университета радиоэлектроники Дуравкин Е.В.

Tkachova Olena, PhD, Ass. professor
Kharkiv National University of Radio Electronics
Str. Lenina, 16, Kharkov, Ukraine, 61166
Tel.: (050) 0567283, e-mail: korov4enko@mail.ru
ORCID ID: 0000-0003-1890-5388

Isaam Saad, postgraduate student
Kharkiv National University of Radio Electronics
Str. Lenina, 16, Kharkov, Ukraine, 61166
Tel.: (057) 7076841, e-mail: asSaad@mail.ru
ORCID ID: 0000-0003-5367-5349

Mohammed Jamal Salim, postgraduate student
Odessa National Academy of Telecommunications named after A.S. Popov
Str. Kuznechnaya, 1, Odessa, Ukraine, 65029
Tel.: (048) 723-23-44, e-mail: 71dkh@ukr.net
ORCID ID: 0000-0002-5367-3319