

**АНТОН АЛЕКСАНДРОВИЧ АНДРЕЕВ**

студент 4-го курса математического факультета, Петрозаводский государственный университет (Петрозаводск, Российская Федерация)  
*andreev@cs.karelia.ru*

**АЛЕКСАНДР СЕРГЕЕВИЧ КОЛОСОВ**

старший преподаватель кафедры информатики и математического обеспечения математического факультета, Петрозаводский государственный университет (Петрозаводск, Российская Федерация)  
*akolosov@cs.karelia.ru*

**ЮРИЙ АНАТОЛЬЕВИЧ БОГОЯВЛЕНСКИЙ**

кандидат технических наук, доцент, заведующий кафедрой информатики и математического обеспечения математического факультета, Петрозаводский государственный университет (Петрозаводск, Российская Федерация)  
*ybgv@cs.karelia.ru*

## **АВТОМАТИЗАЦИЯ ПОСТРОЕНИЯ ГРАФА КАНАЛЬНОГО УРОВНЯ ИКТ-ИНФРАСТРУКТУРЫ ЛОКАЛЬНОГО ПОСТАВЩИКА УСЛУГ ИНТЕРНЕТА\***

Рассматривается задача автоматизации построения графа канального уровня Сети, решение которой осложняется отсутствием поддержки стандартных средств обнаружения связей между устройствами на этом уровне. Таким образом, актуальной является задача разработки методов построения графа канального уровня Сети на основе данных, получаемых из различных неспециализированных источников. Приводится обзор существующих подходов к решению данной задачи, предлагается графовая модель для описания канального уровня Сети с учетом наличия нескольких широкоэмиттерных доменов, образованных сетями VLAN. На основе модели разработан новый комплексный алгоритм построения графа канального уровня, который использует данные, предоставляемые протоколами CDP, LLDP, STP, ARP, которые получаются из MIB сетевых устройств по протоколу SNMP. В силу разнообразия протоколов и технологий, применяемых на канальном уровне, существенно осложняется тестирование средств построения графа Сети. Для создания экспериментальных сетевых окружений авторами был использован инструмент имитационного моделирования Сетей GNS3. Приводятся результаты тестирования комплексного алгоритма как в различных экспериментальных окружениях, так и в реальной Сети Петрозаводского государственного университета.

Ключевые слова: Сеть, граф, канальный уровень, VLAN, экспериментальная среда

### **ВВЕДЕНИЕ**

Развитие ИКТ-инфраструктур (далее Сетей) локальных поставщиков сетевых услуг происходит стремительно, постоянно растет их масштаб и сложность. Существуют коммерческие компании и государственные организации, обладающие Сетями в несколько тысяч машин. Сопровождение таких Сетей требует не только более сложных методов логической структуризации (как, например, виртуальные локальные сети, далее VLAN), но и специализированных инструментов анализа.

Одним из основных инструментов сетевого управления является граф Сети – данные об аппаратных элементах и их связях. С помощью него можно решить множество задач: от поиска вариантов изменения топологии до моделирования потоков данных. Задачи планирования мощности Сети часто решаются с помощью имитационных моделей, также использующих граф [5].

Экспериментальная платформа для исследования моделей и методов управления Сетями Nest

[1], разрабатываемая на кафедре информатики и математического обеспечения Петрозаводского государственного университета, предоставляет средства для автоматизированного построения графа Сетей на сетевом уровне [2] и его визуализации.

Для обеспечения полноты инструментария по исследованию Сетей была поставлена задача разработать и реализовать метод построения графа канального уровня Сетей, построенных на базе стандартов IEEE 802 (который позволит получить представление о физическом устройстве Сети), включая сети VLAN.

### **ОБЗОР АЛГОРИТМОВ И ИСТОЧНИКОВ ДАННЫХ ДЛЯ ПОСТРОЕНИЯ ГРАФА КАНАЛЬНОГО УРОВНЯ СЕТИ**

В настоящее время отсутствует унифицированный способ представления данных об устройствах Сети и связях между ними. Существующие алгоритмы, как правило, извлекают необходимую для построения графа информа-

цию путем анализа различных неспециализированных источников данных.

Так, например, в [7] предлагается использовать данные об остовных деревьях Сети, которые строятся в каждом Ethernet-сегменте в соответствии со стандартом IEEE 802.1D (протокол STP) и представлены записями в BRIDGE-MIB коммутаторов. Преимуществом такого подхода является то, что указанный стандарт реализуется во всем современном оборудовании, однако не все устройства сопровождают BRIDGE-MIB, а при наличии в сегменте Сети нескольких VLAN доступ к данным о деревьях может быть затруднен или невозможен (о чем не сказано в [7]).

Другой способ основан на обработке данных из адресных таблиц пересылки (AFT), также сопровождаемых каждым сетевым коммутатором. Однако, во-первых, стандартами не гарантируется полнота таблиц AFT, во-вторых, не все устройства предоставляют содержимое этих таблиц посредством SNMP. Поэтому в [3] приводится алгоритм, позволяющий построить граф при отсутствии некоторой части информации, что приводит к росту вычислительной сложности и некоторому несоответствию получаемых графов реальной топологии Сети.

Наконец, в последнее время начали появляться протоколы, направленные непосредственно на поддержание целостного описания всех связей между устройствами Сети на канальном уровне. Одним из таких протоколов является Cisco Discovery Protocol (CDP), позволяющий каждому устройству сопровождать описание связанных с ним устройств и особенностей этих связей. Для успешного исследования Сети необходима поддержка данного протокола всеми сетевыми устройствами, которая, однако, слабо распространена на устройствах, производимых не корпорацией Cisco.

Открытым аналогом протокола CDP является Link Layer Discovery Protocol (LLDP, стандарт IEEE 802.1AB). Схема его работы и набор сопровождаемых данных схожи с CDP. LLDP был стандартизован сравнительно недавно и не поддерживается большинством устройств предыдущих поколений.

Отметим также, что в современных Сетях невозможно гарантировать предоставление всеми устройствами данных стандартной структуры или предоставление их вообще (например, устройство не поддерживает SNMP, доступ к информации закрыт).

## МОДЕЛЬ СЕТИ

Описание алгоритма построения графа канального уровня целесообразно приводить в терминах абстрактной модели Сети.

Введем множество сетевых устройств  $D$ . Для каждого  $d \in D$  обозначим множество его физи-

ческих интерфейсов как  $P_d$ . Множество всех интерфейсов всех устройств из  $D$  обозначим  $P = \bigcup_{d \in D} (P_d)$ . Для каждого интерфейса  $p \in P$  могут быть заданы MAC ( $p$ ) и IP ( $p$ ) – физический и сетевой адреса соответственно, а также ID ( $p$ ) и NM ( $p$ ) – целочисленный идентификатор интерфейса и его строковое имя. Введем симметричное отношение  $N \subseteq P \times P$  такое, что два интерфейса  $p, q \in P$  находятся в отношении  $N$  (обозначается  $p N q$ ), если они связаны физически друг с другом.

Сеть на канальном уровне можно описать как граф  $G = \langle V, E \rangle$ , где множество вершин  $V = D \cup P$ , а множество ребер  $E = C \cup L$ . Здесь множество ребер  $C = \{(d, p) \mid d \in D, p \in P_d\}$  содержит связи устройств с их интерфейсами, а множество ребер  $L = \{(p, q) \mid p, q \in P, p N q\}$  содержит связи между интерфейсами.

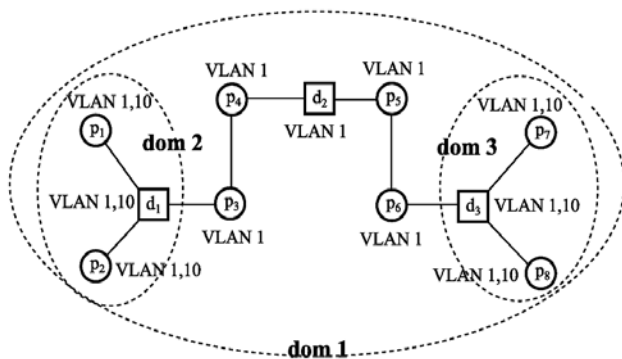
Пусть VID – множество идентификаторов VLAN, используемых в Сети. Припишем каждой вершине  $v \in V$  множество меток  $VLAN_v \subseteq VID$ , соответствующих идентификаторам виртуальных сетей, которым она принадлежит. При этом интерфейс считается принадлежащим некоторой виртуальной сети, если он сконфигурирован соответствующим образом на устройстве или, в противном случае, если он связан с другим интерфейсом, принадлежащим этой виртуальной сети. Устройства считаются принадлежащими некоторой VLAN, если хотя бы один из его интерфейсов принадлежит этой VLAN.

Подграф графа  $G$ , образованный вершинами с меткой  $i \in VID$ , будем обозначать  $G^i$ . Множество компонент связности этого подграфа  $CC(G^i)$  соответствует множеству широковещательных доменов, образованных VLAN с идентификатором  $i$ . Тогда множество  $BD = \bigcup_{i \in VID} (CC(G^i))$  соответствует всем широковещательным доменам Сети. Элементы множества  $BD$  будем обозначать  $dom$ , а идентификатор образовавшей его VLAN – ID ( $dom$ ).

На рисунке представлен пример модели канального уровня Сети, содержащей три коммутатора, для которых  $P_{d1} = \{p1, p2, p3\}$ ,  $P_{d2} = \{p4, p5\}$ ,  $P_{d3} = \{p6, p7, p8\}$ . Эти коммутаторы последовательно соединены:  $p3 N p4$  и  $p5 N p6$ . В изображенной сети имеется две VLAN:  $VID = \{1, 10\}$ . Для широковещательных доменов (выделены эллипсами), образованных этими VLAN:  $ID(dom1) = 1$ ,  $ID(dom2) = 10$ ,  $ID(dom3) = 10$ .

## РАЗРАБОТКА КОМПЛЕКСНОГО АЛГОРИТМА ПОСТРОЕНИЯ ГРАФА КАНАЛЬНОГО УРОВНЯ

Построение графа выполняется в ходе опроса и обработки данных, полученных по протоколу SNMP от устройств Сети. Опрос начинается с некоторого заданного устройства, как правило, корневого маршрутизатора Сети, затем обрабатываются все устройства, связанные с ним, и т. д.



Пример графа канального уровня

По мере опроса создаются вершины графа, представляющие устройства и их интерфейсы. При этом устройство добавляется в множество D, его интерфейсы – в P, а связи – в C и L.

Данные, которые используются в алгоритме, приведены в табл. 1.

Список устройств, которые стоят в очереди на обработку, будем обозначать DQ. Для обращения к какому-либо устройству с помощью SNMP требуется сетевой адрес интерфейсов устройства, который не всегда возможно определить по известному физическому адресу. Поэтому список отложенных связей  $PL \subseteq P$  будет содержать интерфейсы, сетевые адреса которых не получилось определить на момент первичной обработки, так как необходимая запись в таблице АТТ может появиться из MIB устройств, которые будут обработаны в дальнейшем. Список FL будет содержать устройства, сетевые адреса которых не удастся определить с помощью всех доступных данных из Сети.

Разработанный алгоритм разобьем на процедуры. Первые три процедуры принимают на вход вершину-интерфейс p устройства, от которого требуется обнаружить и построить связь с соседним устройством.

**Процедура 1.** Установление связей между устройствами в соответствии с текущим связующим деревом сегмента Сети в соответствии с IEEE 802.1D.

1. Определить DB (p) и произвести поиск IP (DB (p)) в АТТ.
2. Если адрес не был найден, то добавить p в PL и завершить процедуру.
3. Если вершина, представляющая DB (p), еще не создана, то создать ее и добавить в DQ.
4. Добавить ребро (p, DP (p)) в множество L.

**Процедура 2.** Установление связи между устройствами в соответствии с таблицами соседей CDP или LLDP.

1. Определить NA (p) и NP (p).
2. Выбрать такой  $b \in D$ , что  $NA (p) = IP (q)$ , для какого-либо  $q \in P_b$ .
3. Если такого элемента нет, то создать вершину b, представляющую устройство с адресом NA (p), добавить в DQ.
4. Выбрать такой интерфейс  $q \in P_b$ , что  $NM (q) = NP (p)$ .
5. Добавить ребро (p, q) в множество L.

**Процедура 3.** Установление связей между коммутаторами и оконечными устройствами по данным из таблиц коммутации. Если для одного интерфейса в таблице содержится несколько записей, то можно считать, что к такому интерфейсу подключен концентратор.

Таблица 1

Данные о канальном уровне Сети

Объекты	Описание данных	Обозначение
Источник: IF-MIB		
ifPhysAddress	Адрес интерфейса p	MAC (p)
ifIndex	Номер интерфейса p	ID (p)
Источник: BRIDGE-MIB		
dot1dBaseBridgeAddress	Адрес текущего устройства d	MAC (d)
dot1dStpPort	Номер интерфейса p текущего устройства d	ID (p)
dot1dStpPortDesignatedBridge	Адрес соседнего для интерфейса p устройства в основном дереве	DB (p)
dot1dStpPortDesignatedPort	Номер интерфейса соседнего для p устройства	DP (p)
dot1dTpFdbAddress	Таблица коммутации интерфейса p	AFT (p)
Источники: CISCO-CD-PMIB, LLDP-MIB		
cdpCacheAddress, lldpRemManAddrIfId	Адрес соседнего устройства для интерфейса p	NA (p)
cdpCacheDevicePort, lldpRemPortDescr	Имя интерфейса соседнего для p устройства	NP (p)
Источники: IP-MIB, RFC1213-MIB		
ipNetToMediaPhysAddress, atPhysAddress	Таблица соответствия MAC и IP адресов в пределах всей Сети	АТТ
Источники: CISCO-VTP-MIB, CISCO-VLAN-MEMBERSHIP-MIB, Q-BRIDGE-MIB		
vlanTrunkPortTable, vmMembershipTable, dot1qVlanCurrentTable	Информация о конфигурации VLAN на текущем устройстве d и всех его портах $p \in Pd$	$VID_d, VLAN_d, VLAN_p$

1. Если  $|AFT(p)| = 1$ , то создать вершину-устройство  $h$  с единственным интерфейсом с адресом из  $AFT(p)$ . Добавить ребро  $(p, q)$  в множество  $L$  (где  $q \in P_h$ ) и завершить процедуру.
2. Если  $|AFT(p)| > 1$ , то создать вершину-устройство  $s$  с количеством интерфейсов, равным  $|AFT(p)| + 1$ . Добавить ребро  $(p, q)$  в множество  $L$  (где  $q$  – один из элементов  $P_s$ ).
3. Для каждой записи из  $AFT(p)$  создать вершину-устройство аналогично шагу 2 и соединить с любым, не имеющим физической связи с другими интерфейсами,  $q \in P_s$ .

**Процедура 4.** Первый этап процесса построения множества ширококешательных доменов путем выделения компонент связности размеченных подграфов. На вход процедура получает данные вершины-устройства –  $d$ .

1. Выбрать из  $VLAN_d$  еще не обработанный элемент  $i$ .
2. Установить, есть ли такой  $p \in P_d$ , помеченный  $i$ , и такой  $q \in P$ , что  $q \cap p$  и  $i \in VLAN_q$ .
3. Если условие выполнено, то выбрать ширококешательный домен  $dom$  из  $BD$  такой, что  $q \in dom$  и  $ID(dom) = i$ . Иначе, создать новый домен  $dom$  с  $ID(dom) = i$ .
4. Добавить устройство  $d$  в  $dom$ . Также добавить в  $dom$  все  $p \in P_d$ , помеченные  $i$ .

**Алгоритм построения графа.** Комплексный алгоритм построения графа канального уровня Сети. При его запуске множество  $DQ$  состоит из единственной вершины, представляющей устройство, с которого начинается опрос Сети.

1. Выбрать  $d$  из  $DQ$ .
2. Наполнить  $ATT$  и  $AFT$  доступными из устройства  $d$  данными. Получаем данные от протоколов STP, CDP, LLDP. Наполнить множества  $VLAN_p$  для всех  $p \in P_d$  и  $VLAN_d$ .
3. Для всех  $p \in P_d$ , не имеющих физической связи с другими интерфейсами и  $DB(p)$  у которых отличен от  $d$ , произвести запуск Процедуры 1. Если  $p$  попадает в  $PL$  более трех раз – перенести  $p$  из  $PL$  в  $FL$ .
4. Для всех  $p \in P_d$  таких, что  $p \notin PL$ , произвести последовательный запуск процедур 2 (сначала для данных CDP, потом для LLDP) и 3, проверяя перед каждым вызовом, что  $p$  не имеет физической связи с другими интерфейсами (если имеет, то пропустить интерфейс).
5. Произвести запуск Процедуры 4 для  $d$ .
6. Удалить  $d$  из  $DQ$ . Если  $DQ \neq \emptyset$ , то вернуться к шагу 1.
7. Удалить из  $PL$  и  $FL$  все  $p$ , имеющие связь на канальном уровне.
8. Если  $PL \neq \emptyset$ , то для каждого  $p \in PL$  произвести запуск алгоритма начиная с шага 4.
9. Если  $FL \neq \emptyset$ , то для каждого  $p \in FL$  создать соседнюю вершину-устройство и ее интерфейсы со всей имеющейся по ним информацией. Установить с ней связь и добавить в  $L$ .

При реализации алгоритма необходимо учесть, что информация в BRIDGE-MIB может быть разделена по так называемым контекстам, соответствующим различным VLAN. Экземпляры BRIDGE-MIB для каждого контекста получаются отдельными SNMP-запросами.

Алгоритм строит граф Сети с учетом присутствия VLAN, обнаруживает связи, образующие циклы, и устройства, недоступные по SNMP. Использование данных из нескольких источников позволяет алгоритму работать с широким классом оборудования.

## РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ АЛГОРИТМА

При реализации вышеописанного алгоритма в ЭП Nest графовая модель была реализована с помощью модели SON [1] (которая используется в Nest для описания графа) следующим образом. Элементы множества  $D$  – это объекты класса Device. Для  $d \in D$  элементы  $P_d$  – это объекты класса LinkInterface. Класс модели VlanInterface описывает виртуальные интерфейсы с идентификаторами из  $VLAN_d$  для  $d \in D$ . Для описания ширококешательных доменов множества  $BD$  в модель SON был добавлен новый класс BroadcastDomain, представляющий VLAN. Связи из множества  $E$  описываются свойством link класса LinkInterface.

Алгоритм построения графа канального уровня Сети был реализован на языке Java в рамках подсистемы построения графа сетевого уровня ЭП Nest. Для выполнения SNMP-запросов использовалась библиотека SNMP4J [9]. В ходе проведенных работ в код ЭП было добавлено 12 новых классов и интерфейсов, 15 существовавших классов было переработано. Общее количество новых строк кода превысило 1500 (включая комментарии).

Тестирование разработанных алгоритмов проводилось на участке Сети ПетрГУ, включающем 9 устройств, предоставляющих доступ по SNMP, а также множество устройств и хостов без SNMP-доступа. Корректность построенных графов была проверена по записям сетевых администраторов.

В табл. 2 представлены результаты тестирования алгоритмов, использующих данные только CISCO-CDP-MIB, только BRIDGE-MIB, а также комплексного алгоритма. LLDP не был включен в тестирование, так как устройства Сети ПетрГУ не оборудованы поддержкой данного протокола. Время сбора данных включает время простоя при обращении к недоступным устройствам. В дополнение к устройствам, поддерживающим STP, комплексный алгоритм обнаружил 3 маршрутизатора, которые не фигурируют в основном дереве.

Тестовые эксперименты показали, что разработанный комплексный алгоритм позволяет обнаружить большее количество устройств при

Таблица 2

## Статистика тестирования алгоритма в Сети ПетрГУ

Конфигурация	Время сбора данных, сек.	Время построения графа, сек.	Обнаружено		
			Устройств	Интерфейсов	VLAN
Комплексный алгоритм	119,01	61,03	2749	5593	47
Только STP	106,26	51,33	2746	5581	46
Только CDP	118,03	24,65	24	158	36

схожих временных затратах по сравнению с алгоритмами, использующими источники информации по отдельности.

С целью проверки корректности работы алгоритма в конфигурациях, отличных от Сети ПетрГУ (построенной на оборудовании корпорации Cisco), а также с целью проведения полностью контролируемых тестов и получения воспроизводимых результатов проводились эксперименты в виртуальных сетевых окружениях. Для создания таких окружений используются программные средства моделирования Сетей, среди которых нами были рассмотрены GNS3 [6], Cisco Packet Tracer [4] и NetSim [8]. Последние две системы не позволяют использовать созданные окружения как полноценные сетевые сегменты и, следовательно, не могут быть использованы для отладки алгоритма, поэтому был выбран инструмент GNS3.

В рамках тестирования было создано три виртуальных лаборатории: 1) Сеть с поддержкой CDP; 2) Сеть с поддержкой LLDP; 3) Сеть с двумя

независимыми VLAN с одинаковым идентификатором.

Использование GNS3 на этапе тестирования позволило проверить работу алгоритма в различных конфигурациях VLAN, а также помогло реализовать и протестировать поддержку LLDP как источника данных.

### ЗАКЛЮЧЕНИЕ

Комплексный алгоритм построения графа канального уровня, разработанный в рамках данного исследования, показал свою эффективность в Сетях различного масштаба и состава. Используемый подход к построению – задействование нескольких источников данных – оказался удачным в условиях разнородности сетевого оборудования.

В будущем планируется провести расширенное тестирование в Сетях большего масштаба, а также в виртуальных лабораториях, эмулирующих сложные варианты топологий.

\* Разработка программного обеспечения и эксперименты выполнялись на компьютерном оборудовании, приобретенном по Программе стратегического развития ПетрГУ.

### СПИСОК ЛИТЕРАТУРЫ

1. Богдавленский Ю. А. Прототип экспериментальной платформы Nest для исследования моделей и методов управления ИКТ-инфраструктурами локальных поставщиков услуг Интернет // Программная инженерия. 2013. № 2. С. 11–20.
2. Колосов А. С., Богдавленский Ю. А. Параллельный алгоритм построения графа ИКТ-инфраструктуры интернет-провайдера // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2013. № 3. С. 105–110.
3. Breitbart Y. J., Gobjuka H. Ethernet Topology Discovery for Networks With Incomplete Information // IEEE/ACM Transactions on Networking. 2010. Vol. 18. № 4.
4. Cisco Packet Tracer // Cisco Packet Tracer – Networking Academy. 2013. Available at: <https://www.netacad.com/web/about-us/cisco-packet-tracer>
5. Claise B., Wolter R. Network Management: Accounting and Performance Strategies. Claise // Cisco Press. 2007. P. 631.
6. GNS3 // GNS3. 2014. Available at: <http://www.gns3.com>
7. Myung-Hee Son., Bheom-Soon Joo, Byung-Chul Kim, Jae-Yong Lee. Physical Topology Discovery for Metro Ethernet Networks // ETRI Journal. 2005. Vol. 27. № 4.
8. NetSim // Tetcos: Developers of NetSim, Network Simulator. 2014. Available at: <http://tetcos.com>
9. Snmp4j // SNMP4J – Free Open Source SNMP API for Java. 2014. Available at: <http://www.snmp4j.org>

Andreev A. A., Petrozavodsk State University (Petrozavodsk, Russian Federation)

Kolosov A. S., Petrozavodsk State University (Petrozavodsk, Russian Federation)

Bogoyavlenskiy Yu. A., Petrozavodsk State University (Petrozavodsk, Russian Federation)

### AUTOMATION OF ICT-INFRASTRUCTURE LINK LAYER GRAPH DISCOVERY FOR LOCAL INTERNET SERVICE PROVIDERS

The problem of automated Network link layer graph discovery is considered. A solution of the problem is complicated by the lack of standardized tools for device discovery and neighborhood notification. Therefore, the problem of new methods for the Network

link layer graph development with the use of different non-specialized data sources is important. The article provides an overview of existing methods for solving the noted problem and also suggests a graph model for the link layer description considering the presence of a number of broadcast domains produced by VLANs. A new complex algorithm for building the link layer graph was developed on the basis of the model. The algorithm uses data provided by CDP, LLDP, STP and ARP protocols, accumulated by network devices and retrieved from their MIBs with the use of SNMP. Testing of the Network graph building tools becomes very complicated because of the variety of link layer protocols and technologies. Experimental network environment built with the use of GNS3 network simulator was applied in the process of algorithm testing. Complex algorithm testing results obtained both in different experimental settings and in the real PetrSU network are presented.

Key words: Network, graph, link layer, VLAN, experimental environment

#### REFERENCES

1. Bogoyavlenskii Yu. A. Prototype of the Tested Nest for Research of Network Management Methods and Models at the Enterprise Network Level [Prototip eksperimental'noy platformy Nest dlya issledovaniya modeley i metodov upravleniya IKT-infrastrukturami lokal'nykh postavshchikov uslug Internet]. *Programmnyaya inzheneriya*. 2013. № 2. P. 11–20.
2. Kolosov A. S., Bogoyavlenskii Yu. A. A parallel algorithm for constructing a graph of a local Internet service provider's [Parallel'nyy algoritm postroeniya grafa IKT-infrastrukturnykh provaydera]. *Nauchno-tekhnicheskie vedomosti SpbGPU. Informatika. Telekommunikatsii. Upravlenie*. 2013. № 3. P. 105–110.
3. Breitbart Y. J., Gobjuka H. Ethernet Topology Discovery for Networks With Incomplete Information // *IEEE/ACM Transactions on Networking*. 2010. Vol. 18. № 4.
4. Cisco Packet Tracer // Cisco Packet Tracer – Networking Academy. 2013. Available at: <https://www.netacad.com/web/about-us/cisco-packet-tracer>
5. Claise B., Wolter R. Network Management: Accounting and Performance Strategies. Claise // Cisco Press. 2007. P. 631.
6. GNS3 // GNS3. 2014. Available at: <http://www.gns3.com>
7. Myung-Hee Son., Bheom-Soon Joo, Byung-Chul Kim, Jae-Yong Lee. Physical Topology Discovery for Metro Ethernet Networks // *ETRI Journal*. 2005. Vol. 27. № 4.
8. NetSim // Tetcos: Developers of NetSim, Network Simulator. 2014. Available at: <http://tetcos.com>
9. Snmp4j // SNMP4J – Free Open Source SNMP API for Java. 2014. Available at: <http://www.snmp4j.org>

Поступила в редакцию 05.11.2014