

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ
Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра дискретной
математики и информационных
технологий

**Программно-определяемые сети: использование протокола
Openflow в коммутируемых сетях**

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

Магистра 2 курса 271 группы
направления 09.04.01 «Информатика и вычислительная техника»
факультета компьютерных наук и информационных технологий
Юсра Абдулкарим Юсиф

Научный руководитель
к. ф.-м.н., доцент

подпись, дата

А.Д. Панферов

Заведующий кафедрой
к. ф.-м.н., доцент

подпись, дата

Л.Б. Тяпаев

Саратов 2017

ВВЕДЕНИЕ

Современные компьютерные сети обеспечивают очень высокие характеристики по пропускной способности, минимизации задержек при транспортировке данных и времени реакции на изменения конфигурации сети. Такие характеристики обеспечиваются максимальным раскрытием потенциала технологий канального уровня, заложенным еще в эпоху создания сетей относительно небольших масштабов – локальных вычислительных сетей (ЛВС). В современных решениях ограничения масштаба практически преодолены и управлением потоками данных на канальном уровне позволяет строить крупные легко масштабируемые сети.

Принципы управления коммутируемыми сетями строились на идее децентрализации и относительного равноправия формирующих их устройств. С ростом масштабов сетей это вело к ужесточению требований к производительности и вычислительным ресурсам, сложности настройки правил их взаимодействия друг с другом. Со временем сложилось понимание, что в рамках таких принципов потенциал роста производительности и пропускной способности практически исчерпан. В первую очередь это связано с ростом затрат времени на маршрутизацию, с трудностями настройки сети и управления потоками в ней. Особенно с учётом новых потребностей в качестве сервисов для высокоскоростных глобальных сетей и сетей центров обработки данных – наиболее масштабных и нагруженных сетевых решений.

Традиционный путь решения указанной проблемы - разработка еще более мощного, сложного оборудования и проприетарных программных решений для него. Однако в настоящее время все более популярен принципиально новый подход, называемый программно конфигурируемыми сетями (ПКС или в англоязычной терминологии SDN - Software Defined Networks). В рамках данного подхода задачи управления конфигурацией сети и правилами маршрутизации потоков данных решаются

централизованно а сетевые устройства упрощенной конфигурации только обеспечивают выполнение принятых решений. Следование такому подходу позволяет ускорить принятие управляющих решений, повысить удобство конфигурирования, виртуализации, настройки качества обслуживания, но требует дополнительных исследований и разработок.

Целью моей работы является исследование принципов функционирования ПКС и протокола Openflow, реализующего эту идеологию.

В соответствии с данной целью, задачами являются:

- изучение базовых принципов функционирования коммутируемых сетей;
- изучение протоколов децентрализованного управления коммутируемыми сетями;
- общая характеристика протокола OpenFlow, областей его применения;
- изучение практических аспектов управления программно-конфигурируемой сетью с использованием программного моделирования.

Информационной базой являются ресурсы глобальной сети Интернет, научные публикации по теме исследования, протоколы и спецификации, учебники и учебные пособия.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Коммутируемые сети. Технологической основой современных локальных сетей является Ethernet. Непрерывно развиваясь, эта технология стала наиболее масштабируемой и экономически эффективной технологией для построения сетей TCP/IP. Благодаря своей гибкости Ethernet сегодня преобладает в большинстве сетевых сред, а поддерживающее эту технологию оборудование производится в огромных объемах. Представленные сегодня решения обеспечивают скорость до 100 Гбит/с на один канал. Эти и другие

причины привели к тому, что Ethernet рассматривается основным кандидатом для конвергентной инфраструктуры ЦОД.

Основное требование, предъявляемое к компьютерной сети, заключается в выполнении тех функций и предоставлении тех услуг, для которых она предназначена. Например, предоставление доступа к ресурсам Интернет, ERP системе предприятия, обмен электронной почтой, обеспечение IP-телефонии и так далее. Таким образом, можно говорить об эффективности управления сетевой инфраструктурой КС и потоками данных в ней только в том случае, если она успешно выполняет задачи, для которых эта сеть создавалась.

Управляемость сети – это возможность управлять и контролировать централизованно за состоянием основных сетевых устройств, а также выявлять и решать проблемы (автоматически и за счет администратора), выполнять анализ производительности и планировать развитие сети. Производить накопления данных для дальнейшего планирования развития инфраструктуры и быть не зависимой от поставщиков оборудования.

Благодаря системам управления становятся возможными перераспределение нагрузки при сбоях оборудования, оптимизация трафика для повышения отдачи от существующих сетей и планомерное развитие сети за счет анализа трафика и моделирования потребностей пользователей и информационных систем в будущем. Средства управления и мониторинга состояния инфраструктуры позволяют избежать потерь связи или падения производительности сетевых сервисов за счет постоянного отслеживания и управления движением трафика.

Для управления работой сетей используется целый ряд технологий. Это виртуальные локальные сети или же просто VLAN – технология, позволяющая организовывать несколько независимых виртуальных сетей внутри одной физической сети. С помощью VLAN можно выполнять гибкое разнесение пользователей по различным сегментам сети с разной адресацией, даже если они подключены к единому устройству, а также дробить

широковещательные домены. Spanning Tree Protocol (STP) - протокол распределенного связующего дерева. Он применяется для формирования не содержащих циклов избыточных топологий коммутации в коммутируемых сетях. Необходимость в использовании протокола STP связана с тем, что наличие в сети мостов и повторителей неизбежно приводит к возникновению определенных проблем. Механизмы QoS представляют из себя средства, необходимые для обеспечения требуемого уровня сетевого обслуживания определенных пользователей, приложений или сетевых потоков.

Программно конфигурируемые сети (SDN) Под программно-конфигурируемой сетью (ПКС) или software-defined networking (SDN) в англоязычном варианте понимается сеть, логика работы которой формируется специальным программным приложением, функционирующем на специально выделенном для этого ресурсе [1-3]. Реализация этой логики осуществляется привычными сетевыми устройствами. Но на них теперь возлагается ответственность только за уровень передачи данных: прием пакета, его обработку (извлечение адреса) и передачу пакета коммутационной матрице. В том случае, если адреса нет в таблице коммутации, коммутатор обращается для определения нового правила действий к внешней управляющей программе и, получив ответ, осуществляет обработку всех последующих пакетов с этим адресом назначения по вновь созданному правилу.

В ПКС сетевые устройства освобождены от обязанностей по управлению и контролю за работой сети. В современных глубоко виртуализированных распределенных вычислительных средах такой подход оказывается очень удобным еще и потому, что для управляющей программы нет разницы является тот или иной коммутатор виртуальным или реальным физическим устройством. Фактически это обеспечивает легкую интеграцию реальных и виртуальных сетей и манипуляцию ими как единым целым.

Естественной средой для развертывания ПКС являются сети больших центров обработки данных. Они оказываются способны централизованно в

автоматическом режиме индивидуально реагировать на потребности фактически каждого приложения.

Программно-конфигурируемые сети SDN из объекта академических исследований уже перешли в сферу коммерческого применения, и сегодня находятся в центре внимания инженеров телекоммуникационной отрасли.

Симуляция сетей с использованием Mininet Mininet это программное обеспечение для эмуляции сети. Программа позволяет запустить виртуальную сеть, например, состоящую из конечных устройств, коммутаторов и контроллера SDN. В простейшем варианте «по умолчанию» в сети будет всего два хоста, один коммутатор и контроллер. Пользователю предоставляется доступ к этим виртуальным объектам для их настройки. Можно добавлять объекты, линки, менять настройки, расширять и совершенствовать сеть. Но при желании можно сгенерировать весьма не тривиальную сеть со сложной топологией всего одной командой. Если доступные топологии не устраивают пользователя в его распоряжение предоставляется специальный инструмент предварительного описания моделируемой сети с использованием синтаксиса Python. В этом случае его возможности по конструированию структуры сети практически безграничны в том числе при моделировании SDN сетей [4].

Разработка и моделирование сетей с гарантированной пропускной способностью Передо мной была поставлена задача провести моделирование работы сети с симметричной неблокируемой топологией. Неблокируемость обеспечивается большой избыточностью (с точки зрения классической древовидной архитектуры) соединений между коммутаторами. Фактически мы строим многопортовый составной коммутатор. Построение такой сети можно выполнить на массовых не дорогих коммутаторах с поддержкой OpenFlow . При наличии внешнего управляющего контроллера такая сеть может обеспечивать обслуживание массово параллельной вычислительной системы из нескольких сотен узлов. Так, при использовании коммутаторов с 48 портами можно обслуживать до 1152 вычислительных узлов. При этом

потребуется 72 таких коммутатора. В большинстве случаев маршруты передаваемых пакетов будут проходить через три коммутатора (это максимальная длина маршрута в данной топологии). При использовании современных аппаратных решений задержки на собственно перенаправление пакетов минимальны и мы получим чрезвычайно эффективную неблокируемую сеть для парных обменов потоками данных.

В качестве примера сети с топологией составного коммутатора мною была рассмотрена сеть для 18 вычислительных узлов, построенная с использованием 9 коммутаторов. Коммутаторы имеют по 6 интерфейсов, предназначенных для использования в целевой вычислительной сети. Коммутаторы первого слоя (аналог уровня доступа) используют три порта для подключения вычислительных узлов и три порта для подключения к коммутаторам центрального слоя (аналог совмещенной функциональности уровней ядра и распределения). Коммутаторы центрального слоя используют все шесть своих портов для соединения с каждым из коммутаторов первого слоя. Этим обеспечивается неблокируемость при любых сочетаниях парных обменов данными.

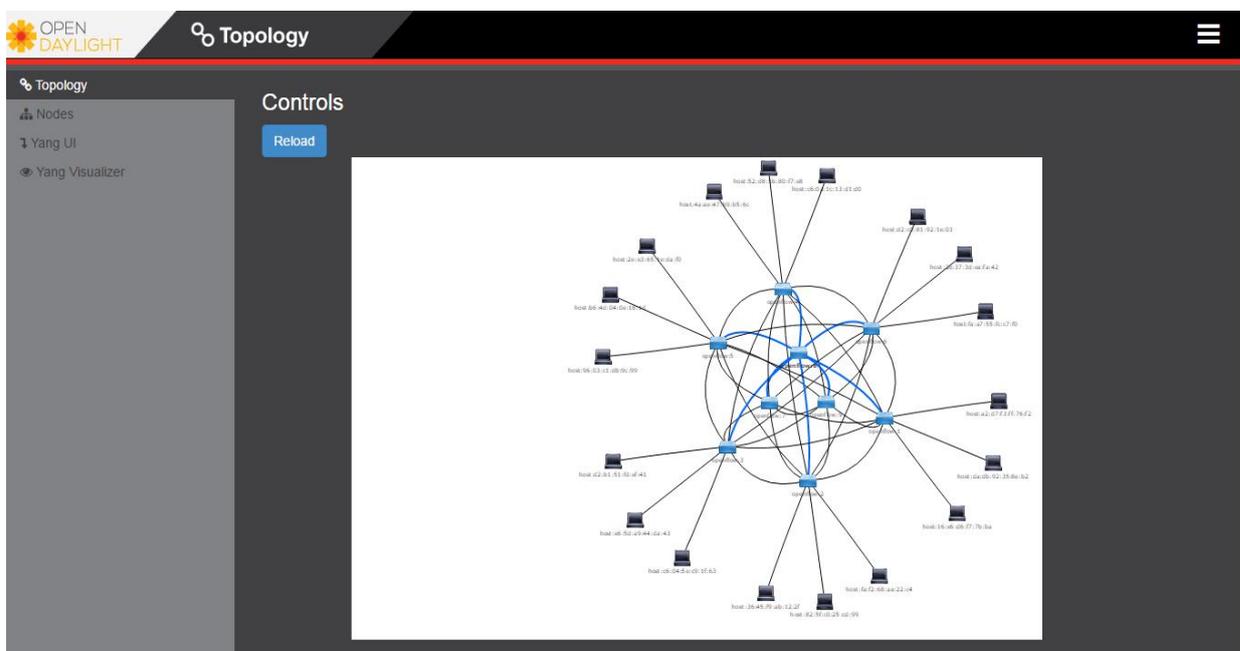
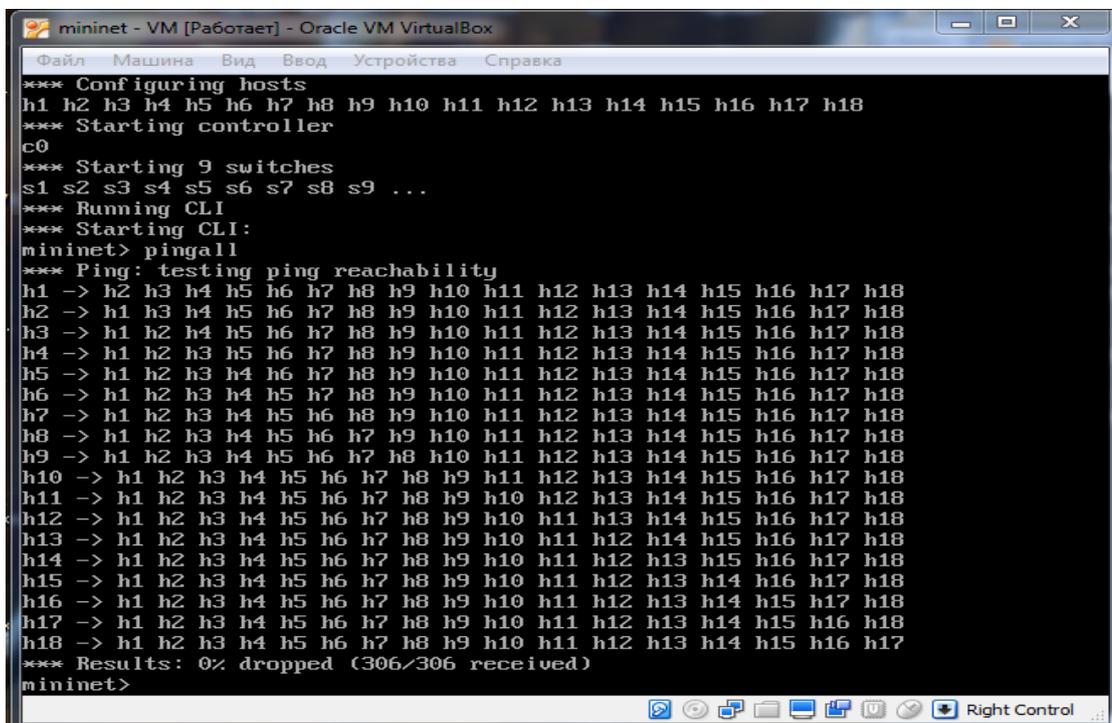


Рис. 1. Пример составного коммутатора на 18 вычислительных узлов

Наличие в представленной топологии множественных дублирующих соединений требует применения внешнего управляющего контроллера. Здесь и далее на схемах сети он не представлен. Не представлена и управляющая сеть, обеспечивающая обмен данными между коммутаторами и контроллером.

Для загрузки в Mininet этой модели был сформирован файл с её описанием с использованием синтаксиса Python. Полный текст описания модели представлен в приложении А. Управление сетью реализовано с использованием внешнего контроллера OpenDaylight, развернутого на отдельной виртуальной машине.



```
mininet - VM [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
*** Starting controller
c0
*** Starting 9 switches
s1 s2 s3 s4 s5 s6 s7 s8 s9 ...
*** Running CLI
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
h2 -> h1 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
h3 -> h1 h2 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
h4 -> h1 h2 h3 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
h5 -> h1 h2 h3 h4 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
h6 -> h1 h2 h3 h4 h5 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
h7 -> h1 h2 h3 h4 h5 h6 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
h8 -> h1 h2 h3 h4 h5 h6 h7 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
h9 -> h1 h2 h3 h4 h5 h6 h7 h8 h10 h11 h12 h13 h14 h15 h16 h17 h18
h10 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h11 h12 h13 h14 h15 h16 h17 h18
h11 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h12 h13 h14 h15 h16 h17 h18
h12 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h13 h14 h15 h16 h17 h18
h13 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h14 h15 h16 h17 h18
h14 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h15 h16 h17 h18
h15 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h16 h17 h18
h16 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h17 h18
h17 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h18
h18 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17
*** Results: 0% dropped (306/306 received)
mininet>
```

Рис. 2. Выполнение команды pingall для модельной сети

Моделирование было успешно реализовано. На Рис. 2 представлены результаты выполнения команды pingall для загруженной модели. Снимок экрана демонстрирует, что для любой пары хостов возможен обмен данными. Из выполненных 306 пингов все 100% успешны и потерянных пакетов нет.

Внешний контроллер OpenDaylight собирает и хранит подробную информацию о происходящих в сети событиях. Эти данные можно

просмотреть. На следующем рисунке представлено окно просмотра статистики передачи данных между узлами моделируемой сети.

Node Connector Id	Rx Pkts	Tx Pkts	Rx Bytes	Tx Bytes	Rx Drops	Tx Drops	Rx Errs	Tx Errs	Rx Frame Errs	Rx OverRun Errs	Rx CRC Errs	Collisions
openflow:8:LOCAL	0	0	0	0	0	0	0	0	0	0	0	0
openflow:8:8	117	117	9945	9945	0	0	0	0	0	0	0	0
openflow:8:7	117	117	9945	9945	0	0	0	0	0	0	0	0
openflow:8:6	117	117	10062	9945	0	0	0	0	0	0	0	0
openflow:8:5	309	1081	23502	77425	0	0	0	0	0	0	0	0
openflow:8:4	118	117	10148	9945	0	0	0	0	0	0	0	0
openflow:8:3	118	117	10148	9945	1	0	0	0	0	0	0	0
openflow:8:2	310	1081	23588	77425	0	0	0	0	0	0	0	0
openflow:8:1	890	501	64188	36825	1	0	0	0	0	0	0	0

Рис. 3. Просмотр статистики обмена данными в окне OpenDaylight

Следующим этапом работы было моделирование работы сети обмена вычислительными данными для 72 вычислительных узлов, построенной с использованием 18 двенадцати портовых коммутаторов. Как и в предыдущем примере дополнительные элементы управляющей сети на рисунке не представлены. Коммутаторы первого слоя используют шесть портов для подключения вычислительных узлов и шесть портов для подключения к коммутаторам центрального слоя. Коммутаторы центрального слоя используют все двенадцать портов для соединения с каждым из коммутаторов первого слоя. Такая сеть позволяет построить уже достаточно мощную массово параллельную систему. Используя в узлах сети двух процессорные серверы на базе процессоров Intel Xeon E5 четвертого поколения, даже без использования ускорителей Intel Xeon Phi или Nvidia Tesla, можно эффективно объединить до 3168 вычислительных ядер и обеспечить пиковую производительность 111.5 Tflops.

На Рис. 4 ниже представлена общая топология такой сети. Моделирование сети было выполнено.

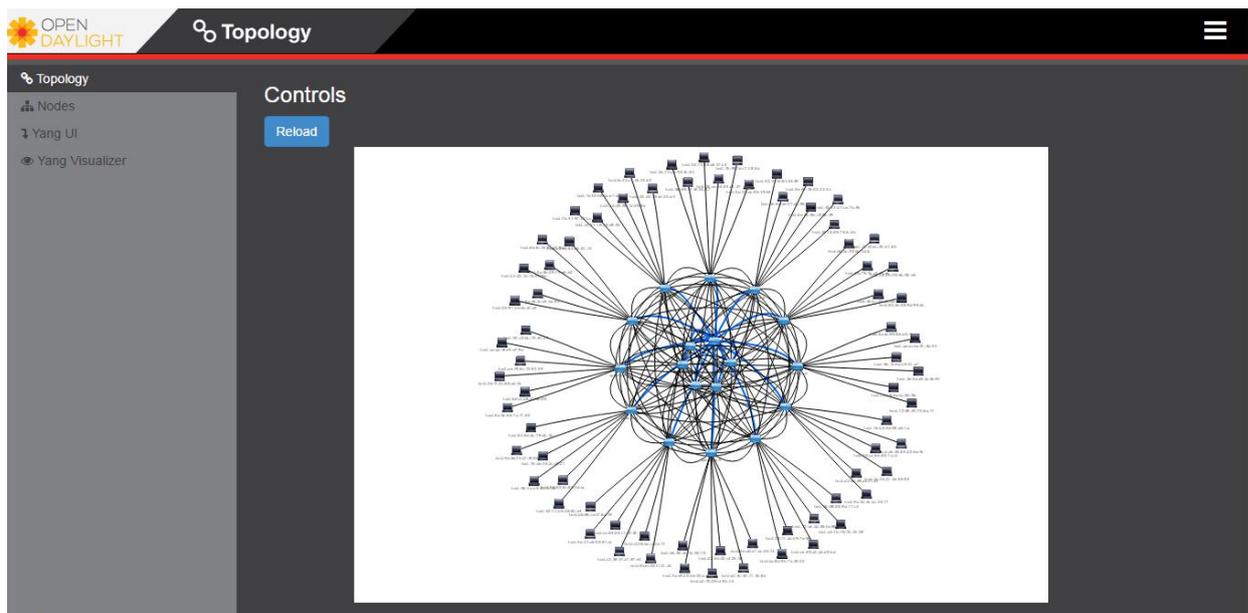


Рис. 4. Топология сети с 72 вычислительными узлами и составным двухуровневым коммутатором из 18 коммутаторов по 12 портов в каждом

В результате проделанной работы было убедительно продемонстрировано, что на базе относительно недорогих коммутаторов можно строить эффективные многопортовые составные коммутационные сети, обеспечивающие неблокируемое соединение в массово параллельных вычислительных системах. Такие решения могут быть полезны в любых конфигурациях с большим парком серверного оборудования и большими потоками данных.

ЗАКЛЮЧЕНИЕ

В представленной работе приведен подробный анализ технологий канального уровня, используемых в настоящее время для управления локальными сетями различного масштаба и назначения. Но заложенные в них идеи децентрализации и относительного равноправия устройств в настоящее время сталкиваются с проблемами при использовании в высоконагруженных сетевых средах с динамичными потоками данных. Принципы программного конфигурирования сети предложены для преодоления указанных трудностей.

Во второй главе изложены принципы построения таких сетей и работа протокола OpenFlow по управлению передачей потоков данных в них. Поскольку развертывание масштабных аппаратных стендов требует очень больших ресурсов, в настоящее время сложные сетевые решения удобно отрабатывать и апробировать в виртуальных сетевых средах. В качестве инструмента для построения таких виртуальных сетевых структур мною был выбран Mininet. Анализ его возможностей, преимуществ и процедур использования составляет содержание третьей главы.

Основной целью моей работы была разработка экономичного сетевого решения для массово параллельной вычислительной системы. К сети обмена данными таких систем предъявляются требования масштабируемости, отказоустойчивости и неблокируемости. Вне зависимости от текущего состояния сети и её загруженности сеть должна удовлетворять запросы на обмен данными между любыми вычислительными узлами. Для решения этой задачи мною была выбрана схема двухуровневого составного коммутатора. Такой подход (при условии использования внешнего управляющего контроллера) позволяет использовать массовые достаточно дешевые коммутаторы в качестве строительных элементов сети, удовлетворяющей высоким предъявляемым требованиям.

Мною были разработаны модели таких сетей и выполнены симуляции их работы в Mininet под управлением внешнего контроллера OpenDaylight. Моделирование было успешным и показало возможность построения сложных высокопроизводительных сетевых решений при использовании относительно простых массовых устройств, поддерживающих протокол внешнего управления потоками данных OpenFlow.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Технологии SDN и NFV: новые возможности для телекоммуникаций [Электронный ресурс] // Центр прикладных исследований компьютерных сетей. – 2015. –Режим доступа: <http://arccn.ru/media/1132>.

2 Барсков А. SDN: кому и зачем это надо?/ А. Барсков // Журнал сетевых решений/LAN. – 2012. -№12. – С. 11-16.

3 Brocade представила стратегию SDN [Электронный ресурс] // Открытые Системы. – 2012: <http://www.osp.ru/dcworld/2012/09/13017404.html>.

4 Stefano Salsano, Pier Luigi Ventre, Luca Prete, Giuseppe Siracusano, Matteo Gerola, Elio Salvadori OSHI - Open Source Hybrid IP/SDN networking (and its emulation on Mininet and on distributed SDN testbeds) // EWSDN 2014, 3rd European Workshop on Software Defined Networks, 1-3 September 2014, Budapest, Hungary.