

В.М. Григорьев

ВЫБОР ПЛАТФОРМЫ ДЛЯ РЕАЛИЗАЦИИ ЛАБОРАТОРНОГО ПРАКТИКУМА ПО КОМПЬЮТЕРНЫМ СЕТЯМ

Аннотация. Исследуется время загрузки большого числа виртуальных машин Qemu под управлением операционной системы RouterOS фирмы Mikrotik. в рамках хост-машины Ubuntu. Показано, что операционная система RouterOS является адекватным выбором для организации виртуальной лаборатории по компьютерным сетям.

Ключевые слова: время загрузки, виртуальная машина, RouterOS, Mikrotik, Qemu, GNS3, Ubuntu .

Актуальность темы. Большинство сетевых устройства (например, маршрутизаторы) работают под управлением операционных систем. В ряде случаев эти операционные системы можно запустить внутри виртуальных машин. Сетевые устройства в сети соединены каналами связи. Для организации виртуальной сетевой лаборатории надо соединить между собой виртуальные машины, в которых запущены операционные системы сетевых устройств.

Виртуальная лаборатория применима как испытательный стенд на начальной стадии проектирования реальных сетей, так и для обучения сетевым технологиям и подготовке к сертификации.

При комплектации реальной сетевой учебной лаборатории возникают трудности при определении производителя сетевого оборудования, подборе и обновлении сетевых устройств. Неприемлемы сетевые устройства, обладающие ограниченной функциональностью и Веб-интерфейсом для настройки. В виртуальной лаборатории подбор и смена «оборудования» не приводит к покупке новых сетевых приборов, а сводится к установке новых виртуальных машин.

Поместив лабораторию на общедоступном сетевом ресурсе, с помощью удалённого рабочего стола можно обеспечить совместную работу над сетевым проектом. Для коллективной работы не требуется

организация одновременного физического присутствия людей в одном месте. Появляется возможность давать студентам домашние задания по проектированию сетей, что в случае реального оборудования невозможно.

Виртуальная лаборатория обладает следующими преимуществами по сравнению с реальной:

- отсутствие затрат на оборудование,
- нет необходимости в помещении для развёртывания сетевого оборудования,
- высокая скорость сбора и коммутации сетевой топологии,
- каналы связи между сетевыми устройствами в виртуальной топологии имеют 100% надёжность и заявленную пропускную способность.

Анализ последних исследований Идея виртуальных сетевых лабораторий не нова, и для их реализации существуют специализированные решения. Для моделирования устройств фирмы Cisco, работающих под управлением операционной системы IOS, используются программы Cisco Packet Tracer и Boson NetSim, имеющие удобный графический интерфейс, позволяющий быстро создавать достаточно сложные сетевые топологии. Однако встроенная в них урезанная версия IOS позволяет изучать лишь сетевые технологии начального уровня

Для лабораторий Netsim фирмы Bosson и PacketTracer фирмы Cisco разработаны лабораторные практикумы для подготовки к сертификации CCNA и CCNP.

Большой интерес представляет использование операционных систем реальных сетевых устройств. Сетевая общественность для подготовки к сертификациям Cisco широко использует менеджер виртуальных машин Dynamips, поддерживающий широкий спектр операционных систем IOS. Виртуальная машина под управлением Dynamips запускается из командной строки и требует к себе бережного отношения. Для предотвращения 100% -й утилизации процессора необходимо подбирать параметр idlePC. Без должных настроек Dynamips использует все ресурсы компьютера уже для топологии из трёх маршрутизаторов.

Dynamips входит в состав контейнера виртуальных машин GNS3. GNS3 – графическая оболочка, позволяющая работать с визуальным представлением сетевой топологии в режиме Drag-and-Drop.

В связке Dynamips-GNS3 сетевая топология хранится в текстовом файле, а конфигурации сетевых устройств присутствуют в файлах в виде команд IOS. Настроенная в одном GNS3 топология легко переносима в другой GNS3.

Есть ряд платных курсов по сертификациям CCNP и CCIE, которые базируются на Dynamips и GNS3:

1. CCIE Service Provider Workbook фирмы ieMentor;
2. CCIE Lab Workbook Internetworkexpert (ine.com) .

Эти курсы снабжены файлами конфигураций и топологий для Dynamips и GNS3.

Есть огромные репозитории, хранилища топологий и конфигураций под Dynamips и GNS3.

<http://www.gns3-labs.com>

<http://gns3vault.com>

<http://7200emu.hacki.at>

gns3 workbench на <http://sf.net>

Для практически любой сетевой технологии можно найти реализацию в Dynamips и GNS3. Всё это положительно влияет на продвижение продуктов Cisco.

Казалось бы связка Dynamips и GNS3 – идеальный выбор для организации учебной лаборатории. Однако, опыт показал, что рядовой студент не относится к настройке параметра IdlePC в Dynamips с должной ответственностью и даже небольшая топология занимает всё процессорное время.

Нельзя не упомянуть менеджер виртуальных машин IOU для операционных систем Cisco IOS. В ней можно запустить пару операционных систем Cisco IOS с весьма мощной функциональностью, и она не требует такой настройки, как Dynamips. К сожалению, IOU не обладает графическим интерфейсом.

Для создания сетевых топологий в GNS3 используется технология Drag-and-Drop: зацепил устройство мышью и перетащил его на рабочее поле. GNS3 поддерживает три менеджера виртуальные машины: Dynamips, VirtualBox и Qemu. Выбор именно этих менеджеров машин для включения в GNS3 обусловлен наличием в их составе раз-

витых средств для соединения между собой виртуальных машин (в VirtualBox - с помощью API).

Под Qemu работает весьма широкий класс сетевых, встроенных и мобильных операционных систем: Juniper JunOS (FreeBSD), файрволы Cisco, множество встроенных Linux для сетевых и мобильных устройств (Vyatta, *WRT, Google Android, Mikrotik RouterOs и др).

Под Qemu можно запустить виртуальные машины не только для Intel-платформы.

VirtualBox также поддерживает множество ОС, но в составе GNS3 требует настройки отдельной виртуальной машины для каждого устройства сетевой топологии. Qemu для всех устройств с одинаковой ОС может использовать единые настройки.

Наш выбор при организации виртуальной лаборатории был сделан в пользу менеджера виртуальных машин Qemu в составе GNS3. Связка Qemu - GNS3 позволяет быстро создать сложную сетевую топологию из десятков устройств, соединив устройства топологии друг с другом.

Постановка задачи. Предполагается, что в виртуальной лаборатории по компьютерным сетям присутствует десяток пользователей, и каждый из них работает с сетевой топологией, состоящей из нескольких сетевых устройств. Одновременно работают десятки виртуальных машин. Возникает задача такого выбора операционных систем хост-машины и сетевых устройств, чтобы обеспечить пользователям комфортную одновременную работу при условии ограниченности ресурсов хост-компьютера. Удобство работы определяется временем запуска и скоростью работы операционной системы внутри виртуальной машины, что в свою очередь зависит от того, сколько ресурсов хост-машины потребляет виртуальная машина.

Следовало определиться, какую операционную систему использовать для хост-машины: Windows или Linux. GNS3 и Qemu задуманы, сделаны и развиваются в Linux. Qemu под Linux поддерживает аппаратную виртуализацию KVM (Kernel-based Virtual Machine) на базе процессоров Intel VT либо AMD SVM. Сам по себе KVM не выполняет эмуляции и используется совместно с виртуальными машинами Qemu. Мы будем использовать KVM без оптимизатора памяти ksmd.

Qemu под Windows не поддерживает KVM и при запуске нескольких экземпляров Qemu используется только одно ядро центрального процессора, что существенно замедляет работу с большими сетевыми топологиями. Разработчики Qemu винят в этом планировщик процессов Windows.

Возникает вопрос выбора дистрибутива Linux. GNS3 написан на Python и требует библиотеки Qt4. После ряда экспериментов с различными дистрибутивами Linux по установке GNS3 из исходных кодов выбор пал на настольную версию Ubuntu.

Обоснование полученных результатов. Определим операционную систему сетевого устройства для запуска под Qemu. Если потребовать, чтобы устройство поддерживало сетевую технологию MPLS, то выбор сразу сократится: это либо операционная система JunOS фирмы Juniper, либо RouterOS фирмы Mikrotik.

По объёму потребляемых ресурсов JunOS существенно превосходит RouterOS. Например, на компьютере с двухядерным процессором Intel Core2 6600 с частотой 2.4 ГГц время загрузки RouterOS версии 5.21 в Qemu под Ubuntu составляет несколько секунд (см. ниже), а JunOS версии Olive12.1R1.9 грузится 60-80 секунд. RouterOS требует минимум 64 Мб памяти, JunOS — 512 Мб. Образ диска RouterOS - 60 Мб, JunOS - 600 Мб.

Рассмотрим влияние KVM на время загрузки в Qemu (версия 13.0) в GNS3 нескольких экземпляров RouterOS версии 5.21 с памятью 64 Мб на компьютере с двухядерным процессором 2.4 ГГц и 4Гб памяти под управлением Ubuntu (табл. 1). Графики зависимости времени загрузки от числа экземпляров RouterOS приведены на рис.1. RouterOS считается загруженной, если она выдала в консоли слово «MikroTik». Это проверяется с помощью следующего скрипта на языке Python

```
tn = telnetlib.Telnet()
for p in range(3000,3079):
    while 1:
        try:
            tn.open("127.0.0.1",p)
            tn.write("\r")
            tn.read_until("MikroTik")
            break
        except socket.error:
```

```
print p
time.sleep(1)
continue
```

Из табл.1 видно, что KVM уменьшает время загрузки RouterOS более, чем в два с половиной раза и это время составляет около двух секунд. Для обычного бытового компьютера имеем приемлемое время загрузки 80-ти экземпляров RouterOS равное 3 минутам.

Загрузим одновременно 32 экземпляра RouterOS (62с). Не останавливая загруженные RouterOS, запустим в новом GNS3 одновременно ещё 32 экземпляра RouterOS. На это уйдёт 75с. Итого 137с., что сравнимо с временем одновременного старта 64-х RouterOS (140с., согласно, табл.1).

Таблица 1

Время загрузки нескольких экземпляров RouterOS в GNS3 в Ubuntu

Кол-во эк-земпляров RouterOS N	Qemu без KVM		Qemu с KVM	
	Общее время загрузки Tсек.	Время загрузки одного экземпляра N/T сек.	Общее время загрузки T сек.	Время загрузки одного экземпляра N/T сек.
16	82	5.125	30	1.875
32	166	5.1875	62	1.9375
48	260	5.416667	100	2.083333
64	470	7.34375	140	2.1875
80	Нет памяти		180	2.25

Число одновременно работающих экземпляров RouterOs под Qemu определяется свободной памятью хост-машины Ubuntu. Анализ показал, что каждый экземпляр RouterOS с памятью 64 Мб, запущенный в Qemu с KVM, требует у Ubuntu 32 Мб памяти, а каждый экземпляр RouterOS с памятью 128 Мб запущенный в Qemu с KVM, требует у Ubuntu 54 Мб памяти.

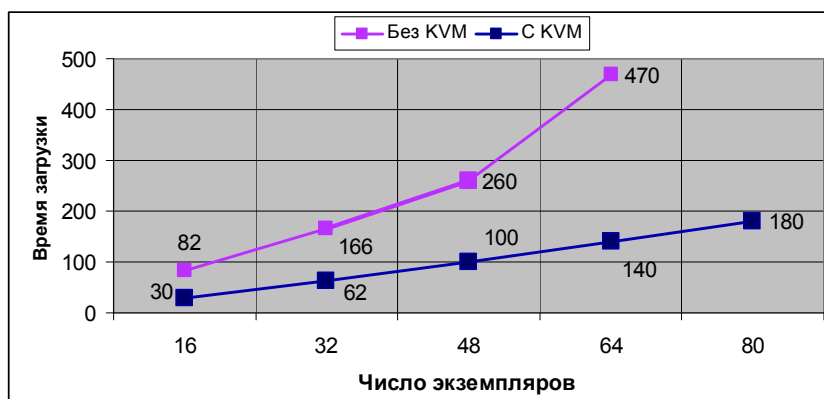


Рисунок 1 - Время загрузки нескольких экземпляров RouterOS в GNS3 в Ubuntu (2 ядра)

Операционную систему Ubuntu можно запускать также под управлением виртуальной машины, например HYPER-V фирмы Microsoft. Qemu в Ubuntu под управлением HYPER-V работает несколько медленнее, чем Qemu в Ubuntu на реальном компьютере. Это обусловлено, помимо прочего, и тем, что KVM не работает на виртуальной аппаратуре HYPER-V.

Хост-компьютер для HYPER-V имеет 4-ядерный процессор Intel Core i7 950 частотой 3066 МГц и 24Гб памяти. В HYPER-V запущена система Ubuntu с 24Гб памяти и 4-мя виртуальными ядрами. Рассмотрим время загрузки в Qemu в GNS3 нескольких экземпляров RouterOS версии 5.21 с памятью 64 Мб на виртуальной Ubuntu (табл.2). График зависимости времени загрузки от числа экземпляров RouterOS приведен на рис.2.

Таблица 2

Время загрузки нескольких экземпляров RouterOS в Ubuntu под HYPER-V

Кол-во экземпляров RouterOS N	Общее время загрузки T сек.	Время загрузки одного экземпляра N/T сек.
16	39	2.4375
32	75	2.34375
48	118	2.458333
64	155	2.421875
80	195	2.4375

Из табл.2 видим, что время загрузки RouterOS под Qemu в Ubuntu под HYPER-V составляет около двух с половиной секунд, что

сравнимо по времени с бытовым компьютером. Для виртуального компьютера имеем приемлемое время загрузки 80-ти экземпляров RouterOS равное 195 секундам.

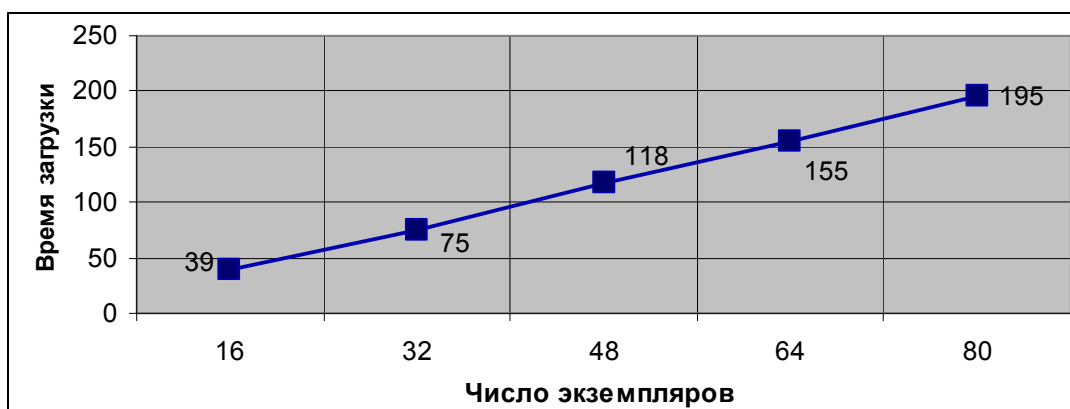


Рисунок 2- Время загрузки нескольких экземпляров RouterOS в GNS3 в Ubuntu под HYPER-V (4 ядра)

Анализ показал, что каждый экземпляр RouterOS с памятью 64 Мб, запущенный в Qemu, требует у Ubuntu под HYPER-V 57 Мб памяти, а каждый экземпляр RouterOS с памятью 128 Мб запущенный в Qemu, требует у Ubuntu HYPER-V 79 Мб памяти. То есть RouterOS в Qemu с KVM потребляет приблизительно вдвое меньше памяти, чем RouterOS в Qemu без KVM.

Время загрузки одного экземпляра RouterOS на бытовом 2-х ядерном компьютере без поддержки KVM в два раза больше, чем на виртуальном 4-х ядерном процессоре HYPER-V (см. табл.). Многоядерность процессоров распараллеливает загрузку нескольких экземпляров RouterOs. Поэтому реальное ядро бытового компьютера имеют приблизительно одинаковую мощность по сравнению с ядром виртуального процессора HYPER-V.

Рассмотрим влияние KVM на время загрузки в Qemu версии 13.0 в GNS3 нескольких экземпляров RouterOS версии 5.20 с памятью 64 Мб на более мощном компьютере с четырёхядерным процессором Intel i5-3550 под управлением Ubuntu (табл.3). Графики зависимости времени загрузки от числа экземпляров RouterOS приведены на рис.3.

Время загрузки нескольких экземпляров RouterOS в GNS3 в Ubuntu для компьютера с 4-х ядерным Intel i5-3550

Кол-во экземпляров RouterOS N	Qemu без KVM		Qemu с KVM	
	Общее время загрузки T сек.	Время загрузки одного экземпляра N/T сек.	Общее время загрузки T сек.	Время загрузки одного экземпляра N/T сек.
16	35	2.1875	23	1.4375
32	60	1.875	40	1.25
48	90	1.875	58	1.208333
64	120	1.875	73	1.140625
80	152	1.9	90	1.125

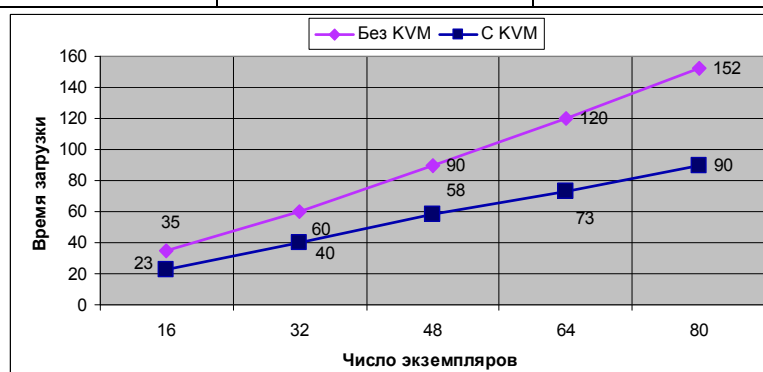


Рисунок 3 - Время загрузки нескольких экземпляров RouterOS в GNS3 в Ubuntu с 4-х ядерным Intel i5-3550

Из табл. 3 видим, что KVM уменьшает время загрузки RouterOS в полтора раза и это время составляет менее двух секунд. Время загрузки 80-ти экземпляров RouterOS с поддержкой KVM составляет полторы минуты.

Выводы. Операционная система RouterOs поддерживает практически все современные сетевые технологии: виртуальные частные сети 2-го и 3-го уровня, IPSec, MPLS, VPLS, VRF. С учётом функциональности, быстродействия и потребляемым ресурсам Mikrotik RouterOs под управлением виртуальной машины Qemu в составе GNS3 под управлением Ubuntu является лучшим выбором для организации виртуальной сетевой лаборатории.

ЛИТЕРАТУРА

1. Григорьев В.М. Виртуализация лабораторного практикума по компьютерным сетям // Системные технологии. Региональный межвузовский сборник научных трудов. - Выпуск 1 (78). - Днепропетровск, 2012. - С. 104–112.