## Overview

In this assignment, you will set up the virtual machine containing the network emulator for our miniature data center. You will then run the emulator and observe the impact of routing policies on application performance. **Note that you will not need to submit any code for this assignment.**

Be sure to watch the video explanation for Programming Assignment 1 before getting started.

## Instructions

### Software Requirements

1. Download and install the following programs:

- VirtualBox
- 7-Zip

2. Download the VM disk (455 MB).

3. Extract moocvm.7z to obtain the file *MoocVm-disk1.vmdk*. (Note: the vmdk is approximately 2.4 GB.)

### VM Setup

**The details may vary minutely on different operating systems and corresponding versions of VirtualBox. If necessary, please request help in the Week 5 Discussion Forum by posting details of your setup.**

1. Open VirtualBox Manager
2. From the menu bar, select **Machine --> New**
3. Create a VM with the name *cloudnetmooc*, type *Linux* and version *Ubuntu (64bit),* and click **Next**

4. Set the memory size to at least 2048MB and click **Next**
5. Select the bullet **Use an existing virtual hard drive file**
6. Click the **folder icon** next to the drop down menu and navigate to the folder containing *MoocVm-disk1.vmdk*
7. Select **MoocVm-disk1.vmdk** and press **Open**
8. Click **Next**

**Recommended: Configure the VM with additional processors and video memory.**

1. In the VirtualBox Manager, select the newly created VM **cloudnetmooc**
2. Click the **Settings** icon below the menu bar
3. In the Settings menu, select **System** from the left side menu
4. Click on the **Processor** tab and increase the count to at least 2
5. Select **Display** from the left side menu
6. Increase the Video Memory to at least 32MB

**Start the VM.**

1. In the VirtualBox Manager, select the newly created VM **cloudnetmooc**
2. Click the **Start** icon below the menu bar
3. When the VM starts, log in with the username mooc and password mooc (Note: the sudo password is also mooc)
4. To start the desktop environment, type startx and press **Enter**

**To change the VM guest display size, you can configure the display to auto-adjust to the VM window size. From the VM window, select View --> *Auto-resize Guest Display*. Alternatively, you can set the resolution manually inside the VM by performing the follow steps.**

1. Click **Applications Menu** --> **Settings** --> **Display**
2. Select the desired resolution from the **Resolution** drop down menu
3. A confirmation window may appear, click **Keep this Configuration**
4. Close the Display window

Note: To keep the distributed image small, we have included only the lightweight graphical editor mousepad. You can install your preferred editor using the command sudo apt-get install [emacs, vim, nano]. Remember, the sudo password is also mooc.

## Common Problems

When creating a new VM, there is no option for 64-bit Ubuntu in VirtualBox.

- This may be caused by several issues: virtualization extensions are not enabled or Hyper-V may conflict with VirtualBox.
- **Enable Virtualization Extensions:** Ensure you have enabled virtualization extensions (VT-x/AMD-V) in your system's BIOS. The exact instructions will depend on your motherboard, but generally the instructions follow:

1. Reboot your machine and open the **BIOS** menu. Depending on your system, this typically is done by pressing the *delete* key, the *F1* key, the *F4* key, or the *Alt* key
2. Open the **Processor** menu
3. Enable Intel Virtualization Technology (Intel VT-X). AMD-V extensions will already be enabled. The extensions may be also labeled "Virtualization Extensions" or "Vanderpool" or various other names
4. Select **Save & Exit**
5. Reboot the machine

- **Disable Hyper-V:** On Windows machines, Hyper-V can conflict with VirtualBox. Disable Hyper-V under **Settings** --> **Control Panel** --> **Turn Windows Features On or Off**.

## I've enabled virtualization extensions but the VM does not start

- Ensure virtualization extensions are enabled in VirtualBox for your new VM. In VirtualBox, select the VM and click **Settings**. On the left side menu, click **System**, then select the **Acceleration** tab. Make sure the box labeled **Enable VT-x/AMD-V** is checked.

## My hardware doesn't support virtualization extensions

- You can download a 32-bit version of the VM image here.

## There is no 7-zip client for my OS

- OS X users can download one of several 7-zip clients:

-Rar extractor

-Keka

-7zX

- Ubuntu users can install 7zip using sudo apt-get install p7zip-full and extract the image using 7za x moocvm.7z.

### After downloading VM disk, how I can I be sure it was not corrupted during transfer?

- You can verify the integrity using md5sum. The checksum for the disks are:

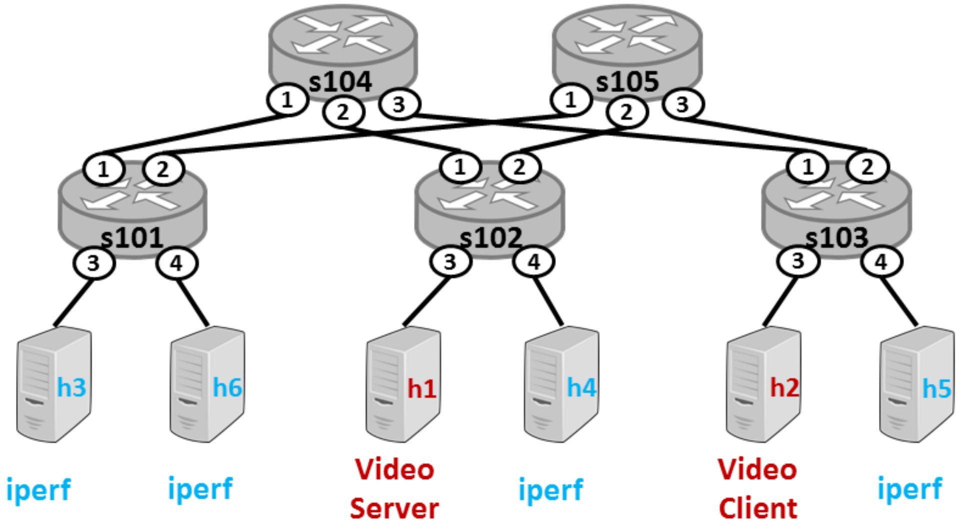
- moocvm.7z: a4da4d88339cd0e462540c0d7c3abf7f
- moocvm32.7z: 6ddf313d51f212d598f14a2a37c918cd

## Video Streaming

Now that the VM is set up, you will learn how to use the environment that emulates the miniature datacenter. Our system uses the network emulator mininet to run a software-defined network (SDN). To control the SDN, we'll use the controller Ryu. This software is already installed and configured in the VM.

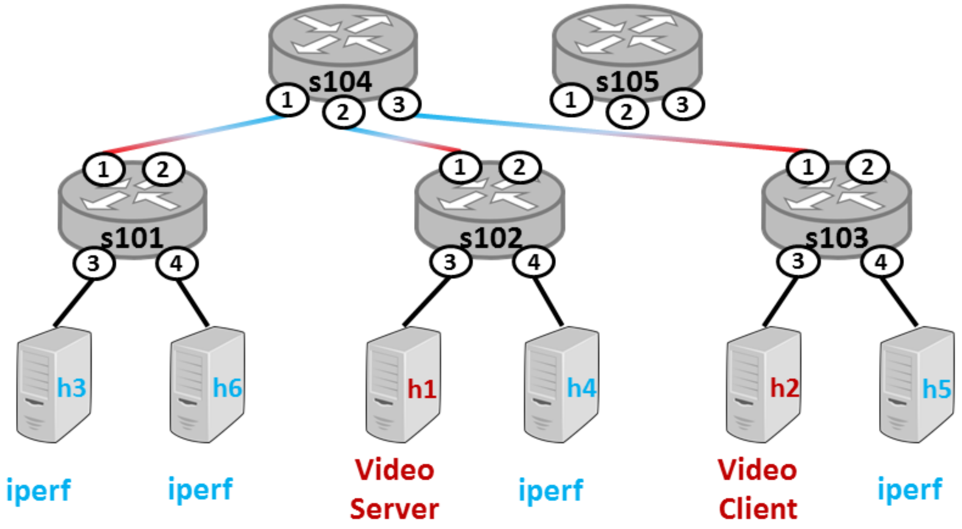
The datacenter will use a topology consisting of core switches, edge switches, and hosts, as shown below. Switches s104 and s105 are **core switches**, which connect to **edge switches** s101, s102, and s103. Each edge switch connects to all core switches, as well as two **hosts** (e.g., two of h1–h6). The datacenter will have two tenants – one running iperf to simulate bulk transfers and another streaming video. The video server will adapt the video quality to the available bandwidth between the video server and client.

Each switch connects to another switch or host through a physical or virtual port. The physical ports are shown in the topology diagram as circled numbers. Each switch also has virtual ports to send packets to the SDN controller or flood traffic (i.e., send a packet out all physical ports except the one on which it was received.)



Next, you will start the emulated environment and see the impact of a poor routing policy on application performance across different tenants. Specifically, you will see how traffic from the iperf tenants interferes with the available bandwidth to the video streaming tenant. This lack of performance isolation between tenants in a datacenter is not ideal. In later assignments, we will write policies that achieve better isolation.

The naive policy you will see in action will route all traffic from the edge switches to a single core switch, as show by the mixed red and blue lines. Notice that s104 can become a bottleneck in the network, and s105 is completely unused. As a result, the traffic of one tenant can easily impact another tenant's perceived network performance.



To start the emulator, open two terminals in the VM. There is a shortcut provided on the desktop.

1. In terminal 1, cd to *~/cloudnetmooc* and run *sudo ./mdc --vid*

2. In terminal 2, cd to *~/cloudnetmooc/minidc/controller*

- Start Ryu: *ryu-manager controller.py*
- The default (naive) routing policy will be loaded automatically.

3. Now that Ryu has started, press <enter> in **terminal 1**.

- A Chrome window will pop up. Press play to start the video.
- Observe the video quality is poor and plays only in low quality. The video may even pause to buffer.

4. End the experiment:

- In terminal 1, type *exit*<enter> (do not Ctrl-c to exit; this will interrupt the teardown process). If you accidentally Ctrl-c and interrupt the process, run *sudo mn -c*.
- In terminal 2, Ctrl-c to stop Ryu.

## What to Submit

Nothing in Coursera. You do not need to submit any code for this assignment. When you have finished Programming Assignment 1A, please consider completing the **optional Programming Assignment 1A Survey.**

Mark as completed