# IC322 – Computer Networking

## Fall AY2013

## Student Directions – Milestone I (v1.0)

**Glossary**

CLASS: refers to all students in a single section.
TEAM: refers to a team of 4-5 students.
STUDENT: refers to an individual within a team.
HOST OS: the operating system of the physical PC that each student is using
GUEST OS: the operating system that is running within a Virtual Machine
PC: the physical workstation for each student

For Milestone I, each student must complete the following six tasks and complete the checklist at the end of this document. Hand in the completed checklist to your instructor. Be prepared to show your instructor your running network the day that the Milestone is due.

**Goals**

By the end of this Milestone, each student should:

1. be able to create new Virtual Machines,
2. be able to perform basic administrative tasks on both Windows and Linux,
3. be familiar with the NAT and Bridged networking modes in VMware,
4. be able to set a static IP address on both Windows and Linux to get any machine onto a network,
5. be able to perform basic Windows and Linux command-line functions.

> Important – Task 6 includes the steps to return your Host PC to a network-ready condition. Make sure you run Task 6 every time you finish working on your VMs in the lab.

**Background**

We will be building a network of both real and Virtual Machines for this project.

"Virtual Machines" allow us to simulate multiple computers on a single piece of hardware. We will be using VMware Player to create our virtual environments. Think of VMware as a program that simulates computer hardware, including the CPU, memory, disk drive, network card, etc. We can load an operating system into that simulated computer, and it will act just

like any physical computer. Much like in the Matrix – most guest OSs are not even aware that they are running in a simulated world.

VMs are great for learning about networking, since they let us easily built and connect multiple machines. We can have multiple VMs running on a single PC and create a virtual network between them. This is much faster and easier than setting up multiple real computers.

> **VMware tip:**
> You can learn more about VMs here:
> http://en.wikipedia.org/wiki/Virtual_machine

Your virtual machines will be stored in a directory on your classroom PC's hard drive. We will need to access these VMs while we are disconnected from the Academy network, so we cannot store them on a shared drive. Make sure that you always sit at the same PC while working on your project, or else you will not be able to access your VM files.

**<u>Task 1:</u>**

This task will walk you through downloading and installing your first VM. We will use Ubuntu Desktop - a popular version of Linux that is considered easy to install and maintain.

Create a directory on your Desktop named "IC322Project". The Desktop maps to a local directory: `C:\Users\<your-login>\Desktop\`. After creating the directory, browse to this location with the Windows Explorer and verify that you can see the directory on your C: drive. You will need to store all of your VMs in this directory.

> **VMware tip:**
> When you see a file named *.iso, that file was created by taking a snapshot of a complete CD or DVD. We usually use ISO files rather than actual CDs to install new OSs, since they are easier to obtain. When we load an OS from an ISO, we are simulating putting an install disk in the drive and rebooting the machine.

1. **Create a Linux VM**
    a. Login to your classroom PC
    b. Download an ISO file. Go to the following URL in a browser on your host machine: http://www.ubuntu.com/download/desktop/ Under "Choose your flavour", select "32-bit". Your computers can handle the 64-bit version, but the 32-bit version of many applications is more stable. Click "Get Ubuntu". You may be taken to a screen that asks for a donation. Linux is free to download, so this is not required. Click "Not now, take me to the download" to continue. Save the ISO file in the default location: C:\Users\<loginname>\Downloads\.
    c. Start the VMware player (Start→All Programs→VMware→VMware Player)
    d. Click "Create a New Virtual Machine"
    e. Click on "Installer disc image file (iso)"
    f. Click "Browse" and select your ISO file from the Downloads directory. Click "Next".

g. "Personalize Linux page": By default, Ubuntu does not use a "root" account - the first user-account created is automatically a regular user who gain root privileges by running "`sudo`".
    i. Enter your actual full name
    ii. For username, use your last name
    iii. Pick a unique password - do not recycle one you have used before
    iv. Click "Next" when done
h. "Name the virtual machine":
    i. Pick a unique name for each machine in your Team
    ii. Under location, Browse to your private Desktop directory. Add "Ubuntu_desktop " at the end to create a subdirectory where your files will be installed.
    iii. Click "Next"
i. "Specify Disk Capacity" page: This indicates how much space will be taken up by the virtual hard drive. Normally you want to take the default value, but since we are not going to install much new software or load files, we can use a lot less. Reduce it to 5GB and click "Next".
j. "Ready to Create Virtual Machine" page: This shows the settings for virtual memory, CPUs, peripherals, etc. Click "Finish" to accept the defaults. Your computer is going to spend several minutes installing. Work through the next step while you are waiting.

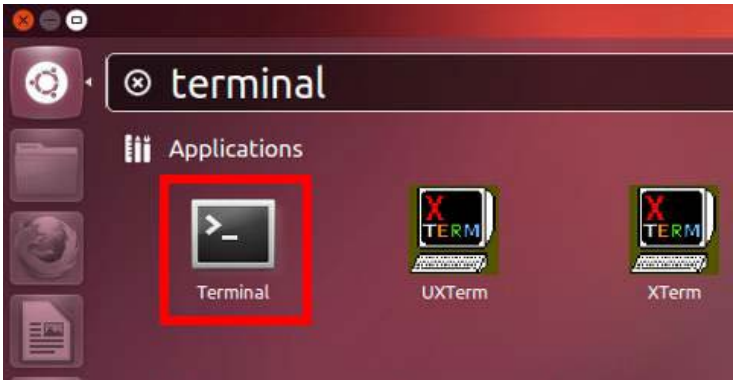
2. **Examine virtual files:**
    a. Open `C:\Users\<loginname>\Desktop\IC322\Ubuntu_desktop\` in Windows Explorer. These are the files that define your VM. Read through this page to understand what each one does: http://www.vmware.com/support/ws5/doc/ws_learning_files_in_a_vm.html
    b. Fill out the item in the Checklist that asks about the purpose of these files.
3. **Linux installation process.** Your computer is going to take several minutes to run the virtual install disk. This is the same process that would happen if you put an actual install disk in your computer and rebooted, except it creates a VM in a subdirectory instead of overwriting your entire hard drive. The installation process for Ubuntu is almost entirely automated. When complete, it will prompt you for your password. Enter it to login.

> **VMware tip:**
> VMware gives you lots of popups asking you to upgrade or install more software. Disregard all of them.

4. **Open a Terminal window.** We traditionally control Linux with command-line functions rather than GUIs. Ubuntu is trying to move its users towards GUIs by making it difficult to create a terminal window. Here is how to create your first one:

a. Click the "Dash" icon highlighted here:

b. Type "terminal" in the black box to find matching applications. Click the "Terminal" icon that pops up:




c. A new icon will show up in the Launcher bar to the left of the screen. This icon indicates that you currently have a terminal window running. RMB on the icon and click "Lock to Launcher". That will keep that icon permanently attached to the Launcher:

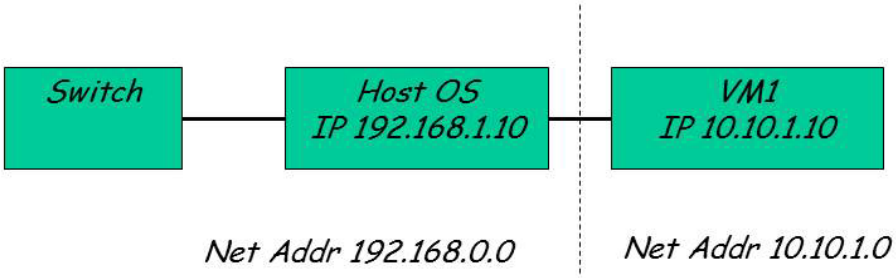
**VMware tip:**
Depending on whether VMware Tools is installed, the guest OS may not want to let go of your mouse. If your mouse gets trapped in the virtual machine, press “Ctrl-Alt” to release it.

d. Verify that the new launcher icons works - close your terminal window and open a new one by clicking on your terminal icon.

e. To open a second terminal window, RMB on the icon and select "New Terminal". It is common to have multiple terminals open at a time.

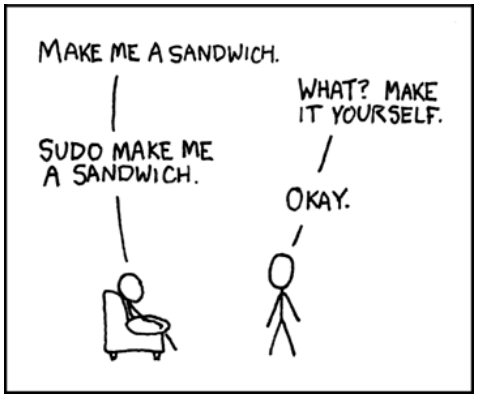
5. **Connect to the Internet with NAT:** VMware creates a virtual network card for your VM.

There are three ways that it can connect your VM to a network" NAT, Bridged, and Host-only. We will investigate the first two of these. NAT (Network Address Translation) allows your VM to share the IP address of the Host OS for its communications with the outside world. The diagram below shows a network diagram for a NAT'ed VM. When the VM wants to talk to a machine on the 192.168.0.0 network, it sends a message to the Host OS, who them repackages the message with its own IP address. So the recipient thinks that the message originated from 192.168.1.10. When the host gets a reply, it recognizes the packet as belonging to the VM, changes the IP address to match the VM's, and passes it along. The same procedure is more commonly used in routers. See pages 349-352 of *Computer Networking, 6th Edition* for more details.




a. The toolbar along the lower edge of the VMware window contains icons for accessing virtual hardware items: optical drives, memory, hard drives, network cards, etc. Mouse-over each of them and find the one that controls the network card. Click on it to open up the network control panel.
b. Select "NAT" from the list of options for connecting to the network.
c. Ensure that "Connected" and "Connect at startup" are both checked on. These determine whether the virtual network card is activated.
d. Click "Okay" to apply these changes.

6. **Install Software Updates.** Malicious hackers are constantly finding new security holes in installed software. They can send exploits against those holes to gain access to your computer and its network. Operating System developers patch these holes as soon as they are found. The single best defense against hackers is to keep your software patches up to date. You should always patch new machines as soon as they are built, and have your OS automatically check for updates.




a. Open a terminal window
b. Type `apt-get update`. The OS will complain that it could not open the lock file. This is because you ran the command as a regular user without root privileges. To get the root privileges, you could either login as root (which is disabled by default in Ubuntu) or you could use the "`sudo`" command to use your account's root access. This is the equivalent of using "Run as Administrator" in the Windows GUI.

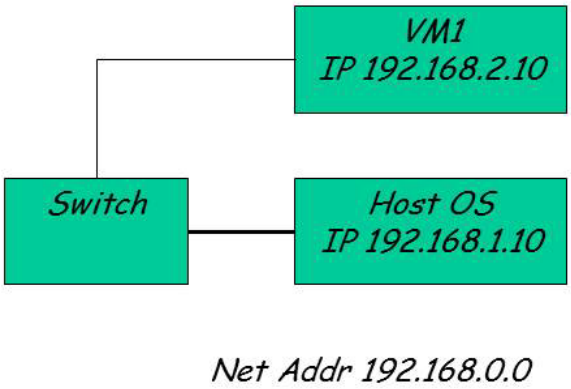
c. Type `sudo apt-get update`. "apt-get update" tells your OS to see whether there are any new updates to your installed packages. The `sudo` command tells the OS that you have root privileges. It will prompt you for your regular password. This prevents somebody from running commands as root if you walk away from your keyboard.
d. Type `sudo apt-get upgrade`. This will run the installer to update any outdated packages. Note that sudo will not prompt you for a password since you just typed it a few moments ago.
e. There is a GUI method for doing this, but that takes all the fun out of learning Linux. You should on learning the command-line methods because they are the same in almost all versions of the OS. The GUI is different for every flavor of Linux and may not always be available.
f. Minimize (do not close) the terminal that is running the updates. We can let it run in the background. You will be able to tell when it is finished because your disk will suddenly get quieter.

7. **Create a new user**
   a. Open a new terminal window
   b. Run "`cat /etc/passwd`". This displays a colon-delimited file that lists all of the users for this machine, plus information about their userID, default shell, home directory, etc. Most of these are "user accounts" that exist only to run specific servers - you cannot log in as them, but processes can run as them. You should see "root" as the first user and your own login name as the last.
   c. run "`sudo useradd -m anyuser`". Enter your own password when prompted by sudo. The -m creates the user's home directory automatically. You can view its contents with "`ls -al /home/anyuser`"
   d. Run "`cat /etc/passwd`". Your new user is now appended to the end of the passwd file. You must create a password before you can log in as that name.
   e. Run "`sudo passwd anyuser`". You will be prompted to create a password. Select one that you are willing to give away to the other members of your team.
8. **Log in as the new user**
   a. Run "`su - anyuser`". This logs you into the new user's account in the terminal window. (You might have to run "`sudo su - anyuser`". The sudo reminds the OS that you have root access.) su allows you to log in as any user.
   b. Run "`whoami`". The OS should reply with "anyuser".
   c. Open a second terminal window and run "`whoami`". Note that su only affected the one window - everywhere else you are logged in with your regular account. It is possible to have multiple windows open, all logged in as different people.
   d. In your "anyuser" terminal, run "`exit`", then "`whoami`". This should have logged you out from the "anyuser" account and returned you to your own. If you type "exit" again, it will close your terminal window.

9. **Suspend the VM.** Close the VMware Player by clicking on the "X" in the upper-right hand corner of the window. The app will ask if you want to Power Off or Suspend. Powering-off is the same as shutting down a computer - you will have to boot up and log in next time you want to use it. Suspending turns off the player, but next time you start it up, your VM will be running exactly as you left it. Click Suspend this time.
10. **Restart the Ubuntu_desktop VM:** Now that you have created your VM, there are two ways we can restart it:
    a. Start VMware. Ubuntu_desktop will be in the list of recent files. Click it to run. If your VM does not start automatically, click “Play Virtual Machine”.
    b. Browse to the Desktop directory with your virtual files. Double-click the one with the *.vmx extension.

**<u>Task 2:</u>**

This Task demonstrates connecting a VM with Bridged networking. The Bridged mode assigns a network IP address to each VM, the same as your Host OS. The figure below is a network diagram showing a Host OS and a VM operating in Bridged mode. Both machines are on the same network. Either one can be seen by a remote host elsewhere on the network. Although the VM must physically use the Host PC’s network card to communicate, the VM’s IP address appears on the network like any other machine.




With NAT, the VM can initiate communications to remote machines, but remote machines cannot initiate comms with your VM. i.e. – you cannot ping a NAT’ed VM from a remote PC. With Bridged networking, machines can communicate in either direction. Many of the servers you use daily are actually Virtual Machines running in Bridged mode. It is almost impossible to tell the difference between a VM and a real machine.

You will start by creating a network on your local machine and then connecting it to your teammates with a hub or switch. Use the table to determine your network and IP addresses.

| | |
|---|---|
| Assign a unique number (1..254) to each PC in the Team: | X == _____ |
| Network address: | 192.168.255.255 |
| Netmask: | 255.255.0.0 |
| Static IP of Host OS: | 192.168.1.X |
| Static IP of Ubuntu_desktop VM: | 192.168.2.X |
| Static IP of Ubuntu_server VM: | 192.168.3.X |

1. **Unplug your Host PC from the Academy network.** We are going to be setting new IP addresses on it, and leaving it on the network could cause problems for the rest of USNA.
2. **Set the static IP for your Host OS**
    a. In a Host OS terminal, run `ncpa.cpl`
    b. Local Area Connection→Properties→IPv4→Properties
    c. Select: Use the following IP address
        i. IP address: <from table above>
        ii. Subnet mask: <from table above>
        iii. Default gateway: <leave blank>
    d. Select: Use the following DNS server addresses.
    e. Leave the DNS servers blank
    f. Click "OK"
    g. Click "Close" ← (the IP change does not take effect until here)
    h. Click "Close"
    i. Run `ipconfig` to verify your changes were applied
3. **Set the static IP for your Ubuntu VM**
    a. Boot up your Ubuntu VM and log in.
    b. Open a VM terminal
    c. `cd /etc/network/`
    d. `cat interfaces` ← this file is the IP configuration in Linux
    e. `sudo gedit interfaces &`
    The file should look something like this:

```
auto lo
iface lo inet loopback

auto eth1
iface eth1 inet dhcp
```

Change it to the following, save, and exit:

```
auto lo
iface lo inet loopback

auto eth1
iface eth1 inet static
address 192.168.2.<number from table above>
netmask 255.255.0.0
```

f. `sudo /etc/init.d/networking restart` (← apply changes)

g. Run "`ifconfig`" to verify your changes

4. **Verify that you can ping both ways between each of your machines:**
   a. Host OS ↔ Ubuntu VM
5. **Plug each of your teams' Host PCs into a hub or switch.**
   a. Verify that your VM can ping at least one VM on another host.
   b. Verify that at least one VM on another host can ping your VM.

**Task 3:**

Draw a network diagram on the bottom of your Checklist. The diagram should include the IP address and OS of each Host and its Ubuntu_desktop VM in your team's network (8-10 total) when configured in Bridged mode. Neatness counts – the use of a ruler or drafting template is recommended.

**Linux tip:**
Most Linux flavors have both a *Desktop* and a *Server* version. The Desktop has a GUI and usually comes loaded with word processors, spreadsheets, and games. They are designed to load on a desktop computer and be used by a human. Server versions have no GUI and much less HCI software. Their owners need to use ssh and command-line to administer them.

**Task 4:**

Now that you have figured out how to build a VM image and connect it to a network, you are going to do the next one on your own. We are going to create a new VM that uses the same operating system, but does not have the user-friendly components like GUIs.

1. **Download the 32-bit Ubuntu Server ISO from here:** http://www.ubuntu.com/download/server
2. Install it to C:\Users\<loginname>\Desktop\IC322\Ubuntu_server\
3. Try shrinking the disk size to 2 GB. Removing the GUI requires a lot less space.
4. Use your last name as the username, but do not recycle a previous password
5. Update your software packages
6. Assign the appropriate IP address from the table above.
7. Connect to the network and ping your other VM

**Ubuntu Server tip:**
gedit is not available since the GUI is not installed. Try nano, emacs, or vi.

**Task 5:**

It is important to learn command-line functions in both Windows and Linux. Read through the following tutorials to understand some basic commands on each OS. Run at least 20 different commands on each OS to get a feel for how some of them work.

1. Linux: http://linuxcommand.org/lc3_learning_the_shell.php (pages 1-10)
2. Windows: http://www.c3scripts.com/tutorials/msdos/commands.html
3. Both OSs have built-in help functions to explain commands. Run these two:
    a. Windows: "`help dir`"
    b. Linux: "`man ls`"

**Task 6:**

Every time you finish working in the lab, do the following to return your Host PC to an appropriate network-ready state:

1. **Suspend your Linux VMs.** You may stay logged in for convenience, since these machines are password-protected by your personal Academy account.
2. **Reconfigure the Host OS for DHCP.**

    Cheat sheet: ncpa.cpl
    Local Area Connection→Properties→IPv4→Properties
    Select: Obtain an IP address automatically
    Select: Obtain DNS server address automatically
    Click "OK"
    Click "Close" ← (the change does not take effect until here)
    Click "Close"
3. **Plug the Host PC back into the Academy network.**
4. **Ping a site on the Internet to verify that you have reconnected properly.**

Name: _____ Section: _____ Team: _____

# IC322 – Computer Networking

## Checklist – Milestone I

- Briefly explain the purpose of each of the following VMware files:

  `*.log` _____

  `*.nvram` _____

  `*.vmdk` _____

  `*.vmsd` _____

  `*.vmx` _____

  `*.vmxf` _____

- Do for each Linux VM (desktop and server)
  - Install and configure Ubuntu image on your local PC
  - Connect to the Internet with "NAT" networking
  - Update patches on VM
  - Create a user
- Connect your two VMs and your Host OS using "Bridged" networking
- Connect to other teammates' VMs using a hub/switch and "Bridged" networking.
- Shut down the VMs and return the host PC to the Academy network.
- Draw a network diagram below showing your team's Host OS and Ubuntu_desktop VMs (8-10 total machines) and its hubs or switches. Leave out the Ubuntu_servers.