

NS2 & Networking Using Desktop Virtualization: An Application of Virtual Box

Parth Shah^I, Vishwas Raval^{II}, Amit Nayak^{III}, Amit Ganatra^{IV}, Yogesh Kosta^{VI}

Abstract—Information Technology has dominated almost all the areas of human life now. Education is not exception in this. How to utilize the technology for imparting the education effectively and efficiently is more important rather than just using the technology. Often users must access applications in operating system (OS) environments other than the native OS on their personal laptop or desktop system. In this paper, we discuss how to share folder between host OS and guest OS, installation steps required to use ns2 simulator and setting networking properties for peer to peer communication and data transfer between host OS and guest OS.

Index Terms—Linux, Operating System, Virtual Box, IP Address, NS-2, Fedora, shared drives

I. INTRODUCTION TO VIRTUAL BOX

Utilizing the technology in effective and efficient way is more important than using the technology for the sake of being an end-user. Normally when one needs to use more than one operating systems all of those must be installed separately keeping the system multi boot. This consumes more resources of system. Especially for academicians, this problem is faced when operating systems concepts are to be taught either the any flavor of Linux must be installed on all PCs or via TelNet it can be used from remote terminal where the system has already been installed. Now Sun Microsystems has developed software called Virtual Box which can be used for the same purpose. We will see in details about Virtual Box and its utilization in coming texts.

The concept of virtualization is familiar to most businesses as a means of consolidating workloads and reducing costs. Server virtualization is recognized as a cost-effective way to condense enterprise application services on a single server, which can help to increase server utilization, improve manageability, reduce power use, and conserve datacenter footprint. Since virtualization technology has evolved in recent years to allow workloads to be securely partitioned on

a single server, multiple application tiers or many OS environments can now be easily consolidated on a single physical machine. Conceptually similar to server virtualization, desktop virtualization allows individual users to access strategic applications on multiple operating systems on a laptop or desktop, with similar benefits that extend from consolidating many virtual machines on a single virtualized platform. The native OS environment on the desktop is called the host environment, while operating system instances within virtual machines are called guest operating systems. As an example, Figure 1 shows a virtualized desktop with application windows that allow a user to run applications on Windows XP, Ubuntu Linux, and the OpenSolaris™ Operating System (OS) in addition to the native host Mac OS.

Usability Features [3] of virtual box.

- Near-native performance. The software ships with optimizations (called “guest additions”) for certain guest operating systems. Guest additions are installed inside a specific VM to accelerate performance or otherwise enhance the feature set of a particular guest environment.
- Rich host and guest integration. When a VM is configured in “seamless” mode, the software maps guest application windows directly on the host desktop, allowing guest and host windows to appear side-by-side. This gives the impression that a guest application window is running natively. Just as with native OS windows, the user can dynamically resize guest application windows as needed, or alternatively, run the guest VM in full-screen mode.
- Shared folders and shared clipboard. Applications running on host and guest operating systems can transparently share files and clipboard data. When a folder is configured as a shared folder, it becomes available to the guest environment as a network share. Optionally, the clipboard of a guest OS can be shared with the host OS.
- Time synchronization. Since the native OS manages the host system’s clock, some virtualized desktops do not adequately address time synchronization between host and guest OS environments. With VirtualBox, the guest operating environment is synchronized automatically with the native OS, allowing the user to see accurate time reporting.
- Powerful virtualized networking. Up to four virtual Gigabit-based network interfaces are supported in each VM and are then accessible from other machines on the physical network. When a VM is configured, the user individually selects what kind of hardware is presented

^I I Head, CE, Faculty of Engineering & Technology (parthshah.ce@ecchanga.ac.in)

^{II} Assistant Professor, CE-IT, Faculty of Engineering & Technology (vishwas.raval@gmail.com)

^{III} Assistant Professor, CE-IT, Faculty of Engineering & Technology (amitnayak.it@ecchanga.ac.in)

^{IV} Associate Professor, CE-IT, Faculty of Engineering & Technology (amitganu@yahoo.com)

^V Dean, Faculty of Engineering & Technology (ypkosta@yahoo.com) (IEEE Member) (SCPM, Stanford University)

Charotar University of Science Technology (CHARUSAT), Education Campus, Changa – 388421, Anand, Gujarat (INDIA)

to the virtual machine. Virtual network interfaces also support remote booting via PXE protocol.

- Robust virtualized hard disk controller support. VirtualBox implements virtualized instances of two of the most common hard disk controller types, IDE and SATA. Up to 32 SATA devices are supported in a guest OS, enabling access to a variety of device types, including devices with removable media.
- Support for Advanced Configuration and Power Interface (ACPI) reporting. This functionality allows an ACPI-aware guest OS to obtain host configuration specifics, which helps the guest configure virtual hardware capabilities. ACPI simplifies the task of cloning PC images from real machines or from third-party virtual machines into VirtualBox software. An ACPI-aware guest can also obtain host power status, such as a low battery condition on a laptop.
- Virtualized display support. VirtualBox supports all VESA standard resolutions and color depths. VMs can also be configured to support screen resolutions many times that of a typical physical display. This allows VMs to be spread over a large number of physical screens on a host system.
- Secure VM instances. Because VirtualBox prevents data leakage, VM instances are isolated and cannot impact applications running on the host OS or within other VMs. This allows companies to extend access to non-native applications while protecting core business functions.

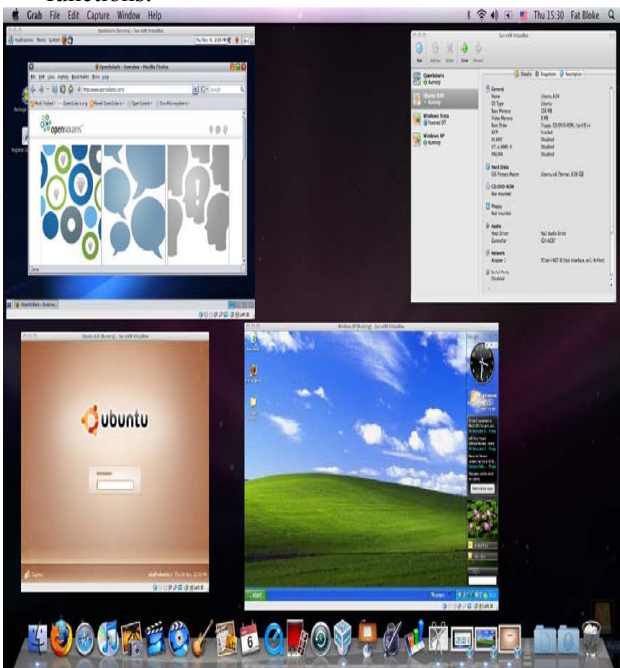


Figure 1. Sun Virtual Box running more than one Operating Systems

The installation of Virtual Box is as easy as installation of normal software. The following text explains how Linux can be used in Virtual Box environment.

II. THE VIRTUAL BOX AND FEDORA INSTALLATION

In order to install Fedor & Virtual Box, one need enough disk space to hold the virtual disk on hard drive. Installation of fedora11 KDE on virtual box is nicely describe in [1].

III. CONFIGURING DRIVES AND FOLDERS OF LINUX

One of the features that make desktop virtualization packages attractive is the ability to move files from the host to the guest virtual machine without the use of a network. Sun xVM VirtualBox has this functionality, so let's go through it for use on Windows systems. Following text explains Step by step instructions for getting shared folders to work in a Fedora 12 VM on VirtualBox after installing Fedora.

VirtualBox Shared Folders allow files residing on the host system to be accessed by guest operating systems running in virtual machines. In addition, assuming adequate permissions are granted on the shared folder, the guest is also able to transfer files onto the host by copying them into the shared folder.

VirtualBox folder sharing is currently only supported for Windows and Linux guests. Shared folders are not supported on virtual machines running Windows 3.1, Windows 95, Windows Me or Windows 98 guests [2].

Shared folders can be configured such that they are persistent (in other words the share is still configured even if the guest is restarted) or transient (such that they a lost after the virtual machine is shutdown or rebooted). Shared folders may also be declared as read-only to prevent the guest operating system from writing to the host folder.

The creation of a shared folder involves selecting the folder on the host system which is be to shared, assigning a share name and choosing whether the folder is to be read-only and transient or permanent.

A. Creating Shared Folders on a Powered Off Virtual Machine

VirtualBox Shared Folders on inactive virtual machines are configured through the Settings dialog. This dialog is accessed by selecting the desired virtual machine from the list and clicking the Settings button in the toolbar. Once the settings dialog is visible, click on the Shared Folders entry in the right hand pane to display the following screen:

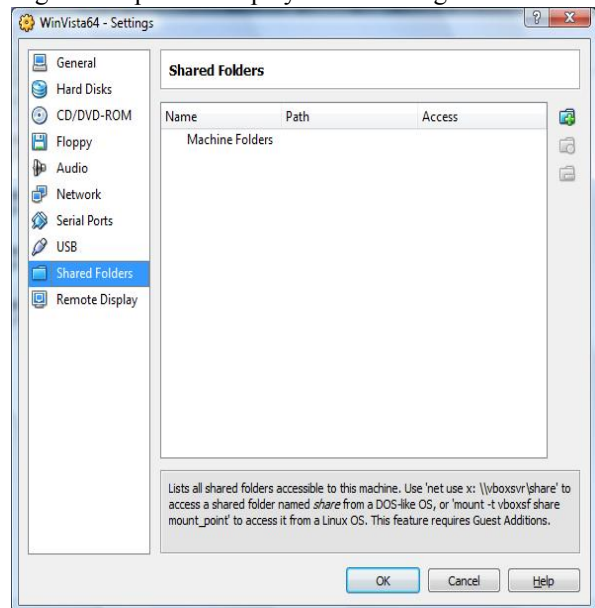


Figure 2. Setting Dialog Box

To add a new shared folder, click on the add folder button (the top button containing an icon of a folder with a green plus sign) and select a folder on the host system to be shared

with the selected guest. To browse for a specific folder, click the down arrow in the Folder Path text box and select other... from the drop down menu. Once a suitable folder has been selected, enter a name for the share in the Folder Name field. If the guest operating system is to be denied write access to the folder, ensure that the Read-only check box is selected before clicking the OK button to create the share:



Figure 3. Defining Shared Folder Name for Guest Operating System

Once the shared folder has been configured, start the virtual machine to access the folder. The mechanism for accessing the folder from within the guest will depend on the type of guest operating system running in the virtual machine, details of which are covered later in this chapter.

B. Creating Shared Folders for a Running Virtual Machine

Shared folders may be configured on a running virtual machine by selecting the Devices->Shared Folders. Menu of the virtual machine window. Once selected, this will display the Shared Folders dialog as illustrated below:

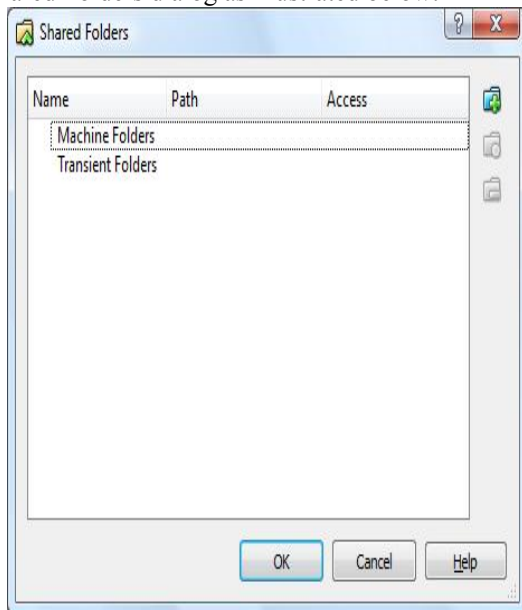


Figure 4. Types of Shared Folders

To add a new shared folder, click on the add folder button (the top button containing an icon of a folder with a green plus sign) and select a folder on the host system to be shared with the selected guest. To browse for a specific folder, click the down arrow in the Folder Path text box and select Other... from the drop down menu. Once a suitable folder has been selected, enter a name for the share in the Folder Name field. If the guest operating system is to be denied write access to the folder, ensure that the Read-only check box is selected.

If the share is to be transient (in other words it will not be restored when the virtual machine is restarted) leave the Permanent check box unselected before clicking the OK

button to create the share.

After the creation of the folder on host operating system next task is to configure guest operating system to access the content from host operating system. Following steps summarizes the configurations for accessing folder from host operating system.

- i) open a console window on fedora
- ii) change user to root by typing command su and specify password
- iii) type "yum install system-config-display"
- iv) type "sudo system-config-display"
- v) this will open a window in your GUI (KDE, Gnome, XFCE, ICE, whatever) choose the options for your display (if using Yakuake you need to hit F12 to make the console window go away so you can see this new window)
- vi) type "yum install gcc kernel kernel-devel kernel-headers"
- vii) restart the VM to boot with the newest kernel
- viii) goto the Devices menu and select install Guest additions
- ix) back to the console and type " mount /cdrom /media"
- x) type " /media/VBoxLinuxAdditions-x86.run"
- xi) (special note, if you have already run the VBox Additions from the cd, skip steps 7-10 and just type this "sudo /etc/init.d/vboxadd setup"
- xii) Now I do not remember if I had to create the share for My Documents in VirtualBox or if it was there by default, but I have one, called "documents" and that is the one I am using in this example. so, type the following "sudo mkdir /mnt/shared-docs"
- xiii) type "mount -t vboxsf documents /mnt/shared-docs"
- xiv) You can now access your Windows XP My Documents folder from your Fedora 12 install in the /mnt/shared-docs folder!!!

IV. NS-2 INSTALLATION

NS2 is basically a tool to simulate the network applications. However configuring NS2 on Linux in Virtual Box environment is not as easy as in normal Linux environment.

STEP 1

From terminal login as super user and do the following.(internet must also be connected in order to install the package dependencies.

```

yum install autoconf
yum install automake
yum install gcc-c++
yum install libX11-devel
yum install xorg-x11-proto-devel
yum install libXt-devel
And for nam installation (optional):
yum install libXmu-devel

```

STEP 2

From terminal go to the folder where the ns-allinone patch is placed and type the following to extract

```

$ tar -xzf ns-allinone-2.31.tar.gz
$ cd ns-allinone-2.29
$. /install

```

STEP 3

Now go to /etc folder and type

```
gedit ~/.bashrc
```

Add the following lines to the end of it. Remember replace "/your/path" by something like "/home/kazim". And accordingly also change the version numbers. This is for ns 2.31.

```
#LD_LIBRARY_PATHOTCL_LIB=/your/path/ns-allinone-2.31/otcl-1.13
NS2_LIB=/your/path/ns-allinone-2.31/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:$USR_LOCAL_LIB
#TCL_LIBRARYTCL_LIB=/your/path/ns-allinone-2.31/tcl8.4.14/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB
#PATHXGRAPH=/your/path/ns-allinone-2.31/bin:/your/path/ns-allinone-2.31/tcl8.4.14/unix:/your/path/ns-allinone-2.31/tk8.4.14/unix
NS=/your/path/ns-allinone-2.31/ns-2.31/
NAM=/your/path/ns-allinone-2.31/nam-1.13/
PATH=$PATH:$XGRAPH:$NS:$NAM
```

Let it take effect immediately

```
source ~/.bashrc2
```

Now, the installation has been completed. If you try:

```
$ ns
```

Then a "%" will appear on the screen.type "exit" to quit the mode and back to "\$"

V. NETWORK SETUP

VirtualBox has eight different networking modes. Each of the eight networking adapters can be separately configured to operate in one of the following five modes:

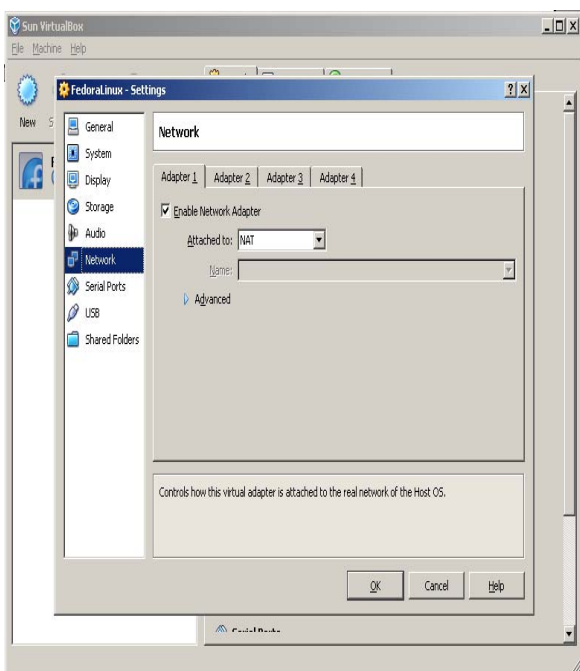


Figure 5. Setting Network Adapter for Guest Operating System

² Note: the step described above is important; otherwise, you cannot run ns successfully or you can restart your X windows i.e. logout and then login, or reboot your system, to make it work.

Not attached

In this mode, VirtualBox reports to the guest that a network card is present, but that there is no connection -- as if no Ethernet cable was plugged into the card. This way it is possible to "pull" the virtual Ethernet cable and disrupt the connection, which can be useful to inform a guest operating system that no network connection is available and enforce a reconfiguration.

Network Address Translation (NAT)

If all you want is to browse the Web, download files and view e-mail inside the guest, then this default mode should be sufficient for you, and you can safely skip the rest of this section. Please note that the ping utility does not work over NAT, and that there are certain limitations when using Windows file sharing.

Bridged networking

This is for more advanced networking needs such as network simulations and running servers in a guest. When enabled, VirtualBox sets up an additional, software-based network interface on the host to which the virtual machine is connected.

Internal networking

This can be used to create a different kind of software-based network which is visible to selected virtual machines, but not to applications running on the host or to the outside world.

Host-only networking

This can be used to create a network containing the host and a set of virtual machines, without the need for the host's physical network interface. Instead, a virtual network interface (similar to a loopback interface) is created on the host, providing connectivity among virtual machines and the host.

For the laboratory work (which requires peer to peer communication) in education, Bridged Networking mode can be used.

With bridged networking, VirtualBox uses a device driver on your host system that filters data from your physical network adapter. This driver is therefore called a "net filter" driver. This allows VirtualBox to intercept data from the physical network and inject data into it, effectively creating a new network interface in software. When a guest is using such a new software interface, it looks to the host system as though the guest were physically connected to the interface using a network cable: the host can send data to the guest through that interface and receive data from it. This means that you can set up routing or bridging between the guest and the rest of your network.

To enable peer to peer communication bridge networking requires on VirtualBox. Following are the steps for network configuration on Fedora guest operation.

- 1) Click on network configuration option (Figure 6) and popup menu will appear and select edit connection.

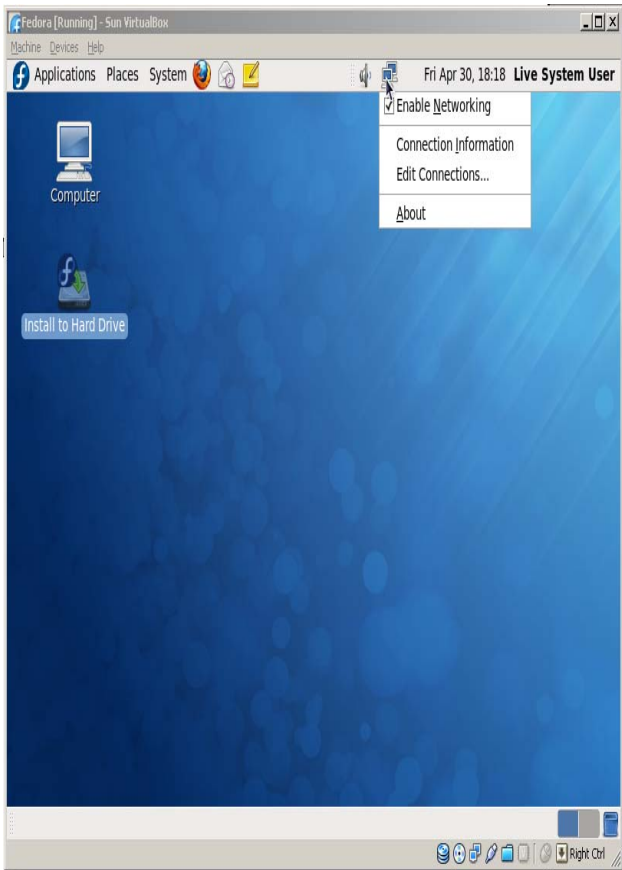


Figure 6. Opening Network Connection Wizard

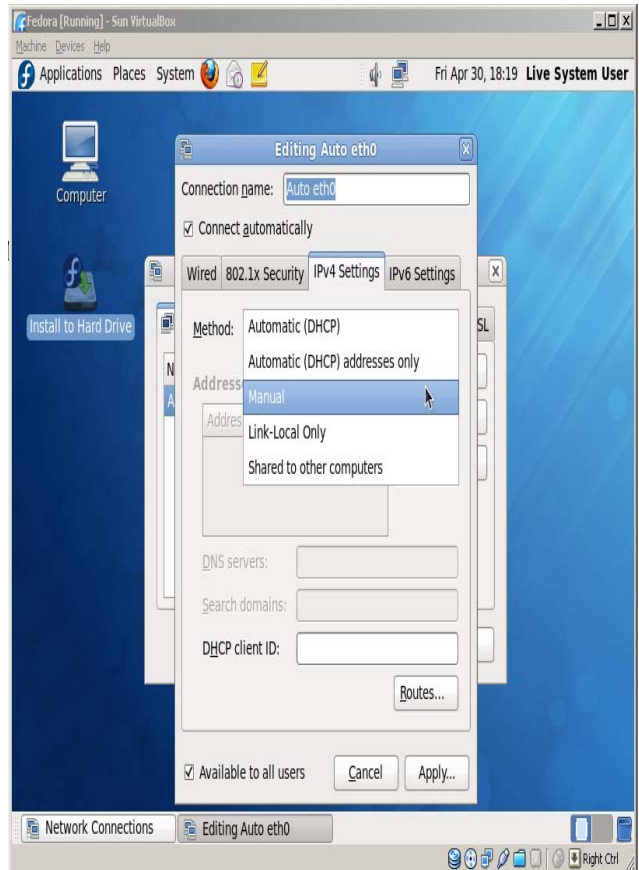


Figure 8. Editing Ethernet Connection as IPv4 (Manual)

2) The network connection wizard will be opened. Select auto eth0 and click on edit.

4) Open network adapter submenu from device main menu and click on attached to combo box. This will open list of available networking type in virtual box. By selecting bridge adapter user can now ping any machine from the current network.

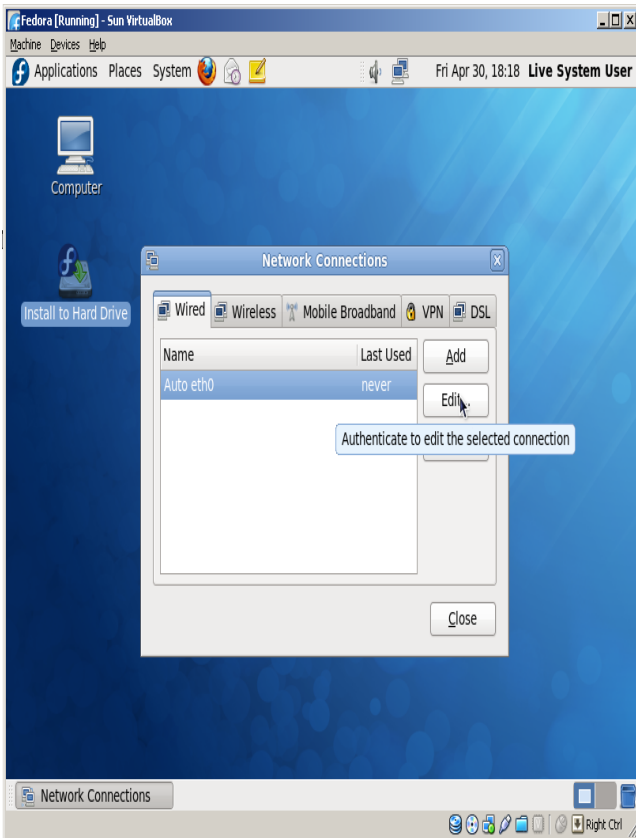


Figure 7. Opening Default Ethernet setting for Editing

3) From editing auto eth0 wizard select Ipv4 settings tab. From this tab select manual method.

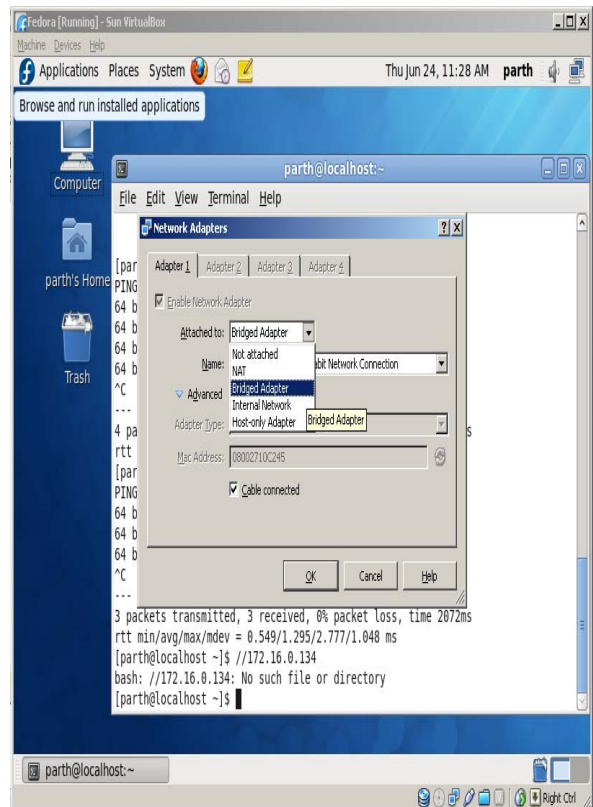


Figure 9. Changing Network Adapter Setting for Specific Ethernet

VI. CONCLUSION

Use of VirtualBox offers extensive flexibility to meet various desktop virtualization needs for a spectrum of users. It helps to create a powerful multiplatform desktop where any x86 operating system — including OpenSolaris, the Solaris OS, Linux, Windows, and Mac OS — can coexist without multi-boot system. Virtual Box occupies only a small portion of computers' resources whereas a dual boot setup allows each OS to take advantage of the computers' full resources. Virtual Box is easy to install and set up. It has very good documentation. No programming skills needed just a little systems administration knowledge.

ACKNOWLEDGMENT

We are thankful to The Omnipotent GOD for making us able to do something. We express our gratitude to the management of CHARUSAT; Shri Charotar MotiSattavis Kelavani Mandal, for providing us research opportunities and their wholehearted support for such activities. Finally, our acknowledgement can not end without thanking to the authors whose research papers helped us in making this research.

REFERENCES

- [1] <http://www.kombitz.com/2009/06/12/how-to-install-fedora-11-kde-on-virtualbox-2-2/>
- [2] http://www.virtuatopia.com/index.php/VirtualBox_Shared_Folders
- [3] OPTIMIZING THE DESKTOP USING SUN™ XVM VIRTUALBOX, Ulrich Möller, VirtualBox Software Team, Sun BluePrints™ Online
- [4] Sun xVM VirtualBox User Manual, Sun Microsystems, Inc., 2008.
- [5] Sun xVM VirtualBox Programming Guide and Reference, Sun Microsystems, Inc., 2008.
- [6] Sun xVM VirtualBox software: <http://www.sun.com/virtualbox>
- [7] xVM VirtualBox open source software and community: <http://virtualbox.org>