Chapter 2 Networking in guest operating systems

In chapter 1, we have installed a guest operating system, Ubuntu 18.04, using four methods. Sometimes, you may wish to have multiple virtual machines at your disposal to repeat experiments for software development or learning purposes.

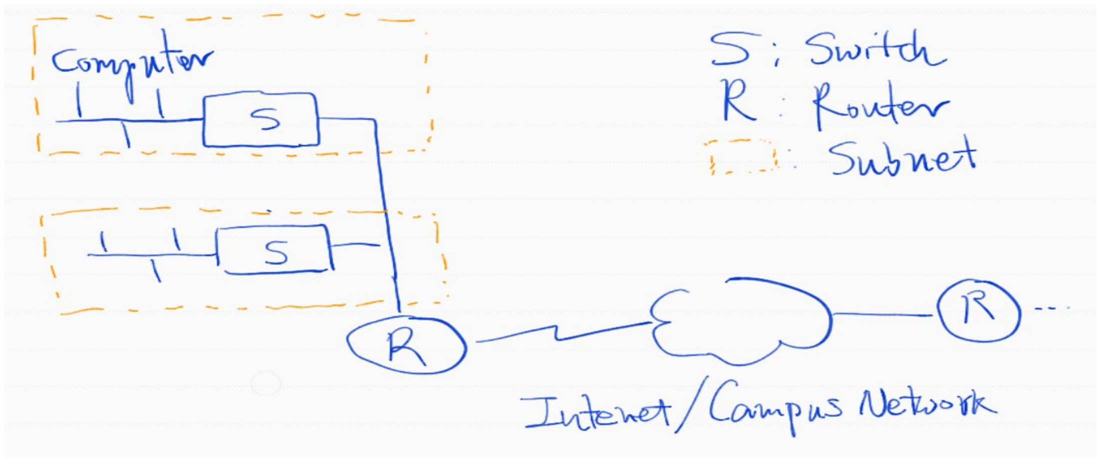
In this chapter, we assume you have created two virtual machines and they are able to connect to external networks (for example, Internet), but not talk to each other, yet. It is essential to understand network basics, because virtual machines operate in networks, physical and virtual.

This chapter covers:

- Networking basics
- Networking modes in VirtualBox
- Network adapters in a virtual machine
- Establishing a network between two virtual machines
    - Configuring VirtualBox
    - Configuring network adapters in virtual machines
- Installing Wireshark in a virtual machine
    - Seeing protocols between virtual machines

Networking basics

In essence, networks consist of switches and routers with computers and/or devices attached to them. Network switches form 'small' networks (typically, Local Area Networks) on their own, whereas routers interconnect them.



In TCP/IP (Transmission Control Protocol/Internet Protocol) networks, computers typically belong to a subnet (often, Local Area Network or WiFi Network) of a bigger network (for example, Enterprise or Campus Network). In order to communicate with another computer, a computer must be equipped with the following pieces of information:

- IP address - The unique (source) address of the computer on the subnet and/or on the Internet
- Subnet mask - The computer uses it to determine whether a destination IP address is on the same subnet or not
- Gateway IP address - The computer sends data to its Gateway, if the destination IP address is not on the same subnet
- IP address(es) of DNS server (Optional) - The computer obtains an IP address for its destination from a human friendly name (for example, cs.nott.ac.uk) via the Domain Name System (DNS)

On Ubuntu 18.04 LTS or CentOS 7, you can issue the command to inspect network interface information in a computer:

```
vagrant@ubuntu-bionic:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:82:7a:7b:51:94 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
    valid_lft 86173sec preferred_lft 86173sec
    inet6 fe80::82:7aff:fe7b:5194/64 scope link
    valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:67:1b:42 brd ff:ff:ff:ff:ff:ff
    inet 10.3.15.21/24 brd 10.3.15.255 scope global enp0s8 ❶
    valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe67:1b42/64 scope link
    valid_lft forever preferred_lft forever
```

❶ 10.3.15.21/24 represents an IP address and its network ID in CIDR (Classless Inter-Domain Routing) notation. /24 denotes the 24 bits out of 10.3.15.21 (32 bits) represent its network ID from the left.

On Windows 10, you can use the command:

```
D:\>ipconfig /all
```

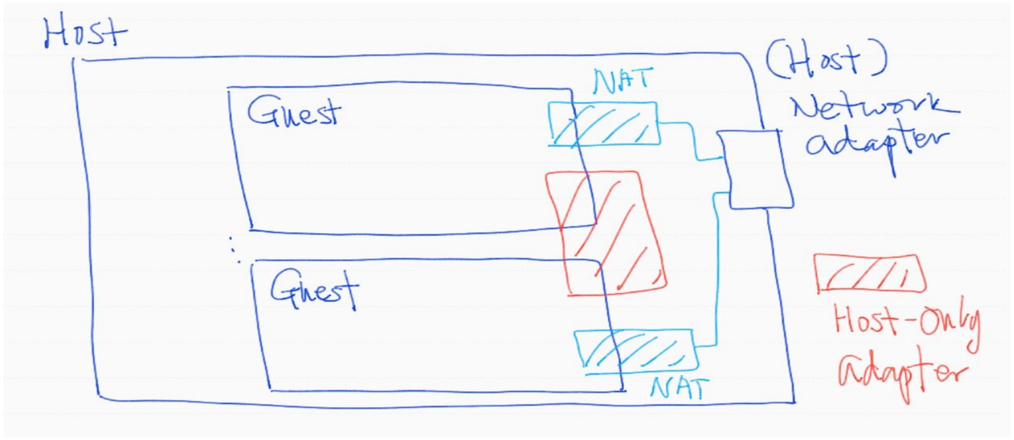
Networking modes in VirtualBox

According to the VirtualBox manual (https://www.virtualbox.org/manual/ch06.html), there are five network modes in VirtualBox, which makes it both flexible and complicated. In this book, the network requirements are:

- Virtual machines should be able to communicate with the external network outside the host. We need to download PostgreSQL packages from the Internet.
- One virtual machine should be able to communicate with another in PostgreSQL replication mode.

Networking modes in VirtualBox rely on virtual network switches and routers implemented in VirtualBox. Still, network basics mentioned earlier apply to virtual machines. A virtual machine must have an IP address(es) and belongs to a subnet, defined and managed by VirtualBox.

In a sense, Host-Only Adapter below acts like a virtual switch via which multiple virtual machines communicate with each other and the host. NAT (Network Address Translation) Adapter in VirtualBox plays a role of virtual router that allows one virtual machine to connect the host's physical network. Note that VirtualBox manages more than one Host-Only or NAT Adapter.



Specifically, VirtualBox supports the following networking modes:

Mode	VM→Host	VM←Host	VM1↔VM2	VM→Net/LAN	VM←Net/LAN
Host-only	+	+	+	–	–
Internal	–	–	+	–	–
Bridged	+	+	+	+	+
NAT	+	Port forward	–	+	Port forward
NATservice	+	Port forward	+	+	Port forward

<Source: https://www.virtualbox.org/manual/ch06.html>

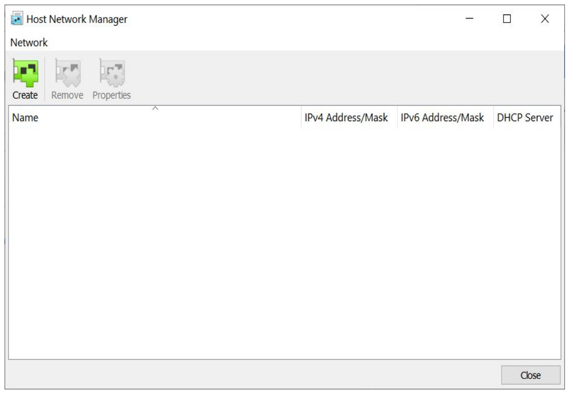
There are some points to consider which adapters to choose in VirtualBox for virtual machines in this book:

- NAT is the default mode in VirtualBox and it allows a virtual machine to access external networks. However, it does not allow one virtual machine to talk to another.
- The Bridge mode has an effect of having multiple host network adapters, but you can not use it when your host has only one static IP address, as is often the case with University computing labs.
- Host-only and Internal modes allow one virtual machine to talk to another. However, they do not have access to external networks outside the host.

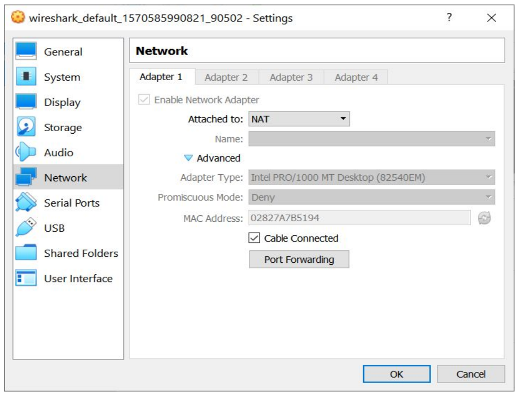
One way to satisfy the network requirements is to use NAT and Host-only modes together in virtual machines.

NAT adapter in virtual machines

Assuming that you have VirtualBox and Vagrant freshly installed on you host computer. Let's take a look at File -> Host Network Manager on the menu. You may wish to remove any network adapters to see what happens, when you create a virtual machine:



Name	IPv4 Address/Mask	IPv6 Address/Mask	DHCP Server



Enable Network Adapter: ✓
Attached to: NAT
Name:
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)
Promiscuous Mode: Deny
MAC Address: 02827A7B5194
Cable Connected: ✓

Let's create one virtual machine at the D:\wireshark directory, using the command:

```
D:\wireshark>vagrant init ubuntu/bionic64
```

Then, you can bring up the virtual machine by issuing the vagrant up command:

```
D:\wireshark>vagrant up
...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat ❶
```

❶ This line shows a Network adapter nat has been created.

At this stage, you will see no change to the Host Network in VirtualBox. If you look at the Settings of the virtual machine, you will notice that the network mode is set to NAT:

Let's login to the virtual machine to test network connectivity to the Internet.

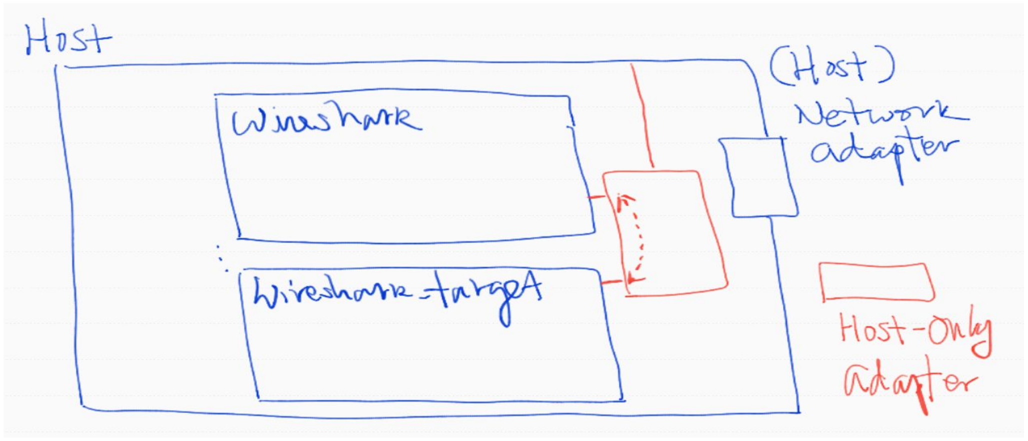
```
D:\wireshark>vagrant ssh
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-65-generic x86_64)
...
vagrant@ubuntu-bionic:~$ ping www.google.com
PING www.google.com (172.217.25.100) 56(84) bytes of data.
64 bytes from nrt13s51-in-f100.1e100.net (172.217.25.100): icmp_seq=1 ttl=50 time=105 ms
64 bytes from nrt13s51-in-f100.1e100.net (172.217.25.100): icmp_seq=2 ttl=50 time=67.8 ms
^C
--- www.google.com ping statistics ---
2 packets transmitted, 3 received, 0% packet loss, time 2042ms
rtt min/avg/max/mdev = 67.851/81.486/105.949/17.337 ms
```

Establishing a network between two virtual machines

As discussed in Networking modes in VirtualBox, NAT is the default mode in VirtualBox. Therefore, you could test network connectivity using the ping command.

If you have not cloned your virtual machine, it is time to do it. Assuming that you have two virtual machines (for example, wireshark and wireshark_target), you need to create a Host-only network interface to allow one virtual machine to communicate with the clone.

Note that before you create a new Host Network, it is good to halt the virtual machine and its clone.

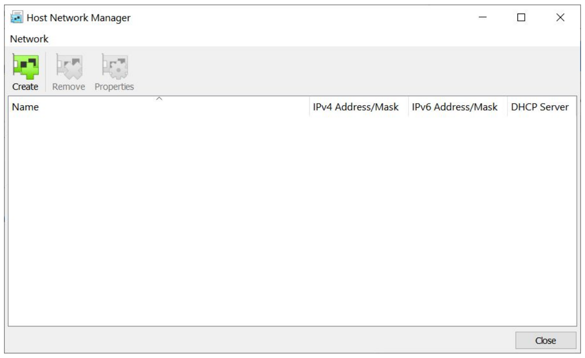


For this chapter, we can create and use any subnet in VirtualBox. Safely, we can go with a subnet, 10.1.15.1/24 and let's assign static 10.1.15.21 and 10.1.15.22 to the virtual machines, wireshark and wireshark_target.

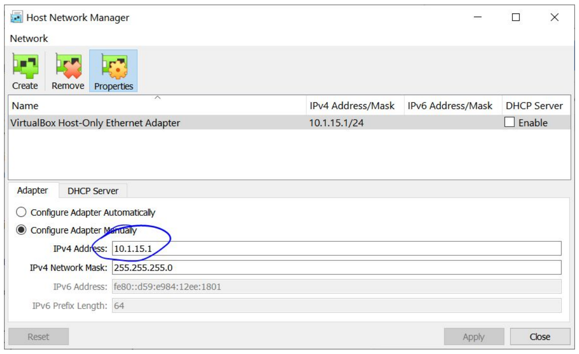
Note that a Host Network in VirtualBox supports DHCP (Dynamic Host Configuration Protocol). However, we do not use it in order to have a controlled environment with static IP addresses assigned to the virtual machines.

Step 1 Configuring VirtualBox

On the File menu of VirtualBox, click Host Network Manager and there is no Host Network Adapter yet.



Name	IPv4 Address/Mask	IPv6 Address/Mask	DHCP Server

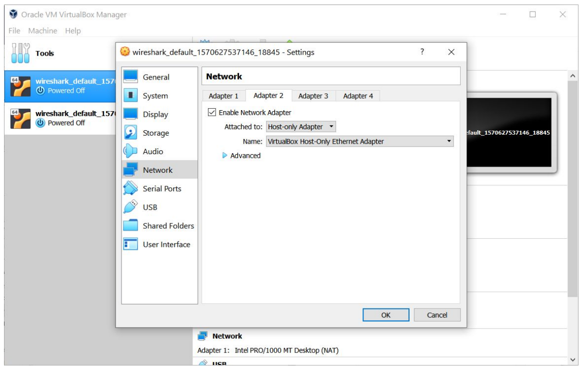


Name	IPv4 Address/Mask	IPv6 Address/Mask	DHCP Server
VirtualBox Host-Only Ethernet Adapter	10.1.15.1/24		☐ Enable

Configure Adapter Automatically
● Configure Adapter Manually
IPv4 Address: 10.1.15.1
IPv4 Network Mask: 255.255.255.0
IPv6 Address: fe80::d59:e984:12ee:1801
IPv6 Prefix Length: 64

Press Create above and you will have a new Host Network. Click Properties and modify IPv4 Address field, as shown above.

Now, you click on one virtual machine and you can go to Machine -> Settings -> Network and press Adapter 2 and pick Host-only Adapter in the Attached to: drop down list.



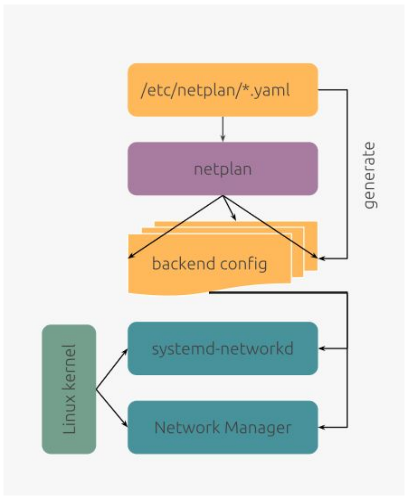
Network
☑ Enable Network Adapter
Attached to: Host-only Adapter
Name: VirtualBox Host-Only Ethernet Adapter

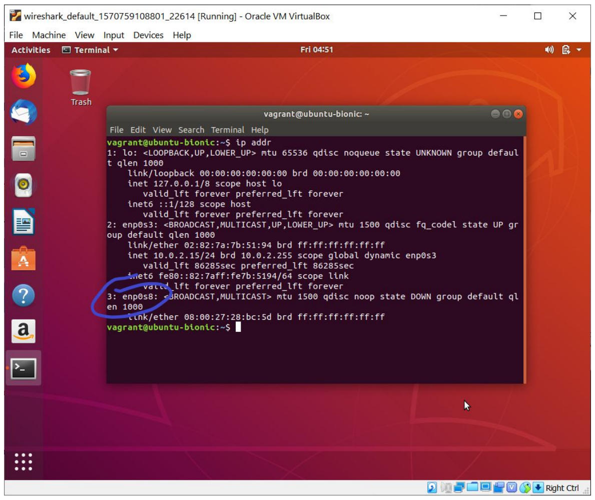
Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)

Similarly, you need to do this in the other virtual machine with the same IP address (in fact, subnet (10.1.15.1).

Step 2 Configuring network interfaces in virtual machines

You have created a Host Network (10.1.15.1/24) in VirtualBox that the two virtual machines can use to communicate with each other. The next step is to set up the IP address (10.1.15.21 or 10.1.15.22) on its Host-only adapter in each virtual machine.




wireshark_default_1570759108801_22614 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
Fri 04:51
Trash
vagrant@ubuntu-bionic: ~
File Edit View Search Terminal Help
Right Ctrl

```
vagrant@ubuntu-bionic:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 02:82:7a:7b:51:94 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 86285sec preferred_lft 86285sec
    inet6 fe80::82:7aff:fe7b:5194/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default ql
en 1000
    link/ether 08:00:27:28:bc:5d brd ff:ff:ff:ff:ff:ff
vagrant@ubuntu-bionic:~$
```

Source: https://netplan.io/

Inside the virtual machine, you can identify the Ethernet interfaces by issuing the command:

```
vagrant@ubuntu-bionic:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 1000
    link/ether 02:82:7a:7b:51:94 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
    valid_lft 85872sec preferred_lft 85872sec
    inet6 fe80::82:7aff:fe7b:5194/64 scope link
    valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:28:bc:5d brd ff:ff:ff:ff:ff:ff ❶
```

❶ This line shows that the network adapter enp0s8 is not configured (no IP address!) yet.

The virtual network adapter enp0s8 is the one you need to configure and you need to assign a static IP address (10.1.15.21) to it. On Ubuntu 18.04 LTS, a netplan utility is used to configure network interfaces.

You can find out the YAML definition for this virtual machine, wireshark by using the ls -l command:

```
vagrant@ubuntu-bionic:~$ ls -l /etc/netplan
```

```
-rw-r--r-- 1 root root 477 Oct 11 01:58 50-cloud-init.yaml
```

You need to add a new definition to /etc/netplan/50-cloud-init.yaml under ethernets:

```
network:
    ethernets:
      enp0s3:
            dhcp4: true
            match:
                    macaddress: 02:82:7a:7b:51:94
            set-name: enp0s3
      enp0s8:
            addresses:
            - 10.1.15.21/24
version: 2
```

Note the lines in **bold** are newly added. Or you may wish to create a new YAML file, /etc/netplan/enp0s8.yaml:

```
network:
    ethernets:
      enp0s8:
            addresses:
            - 10.0.15.21/24
version: 2
```

Then issue the netplan command:

```
vagrant@ubuntu-bionic:~$ sudo netplan generate
vagrant@ubuntu-bionic:~$ sudo netplan apply
```

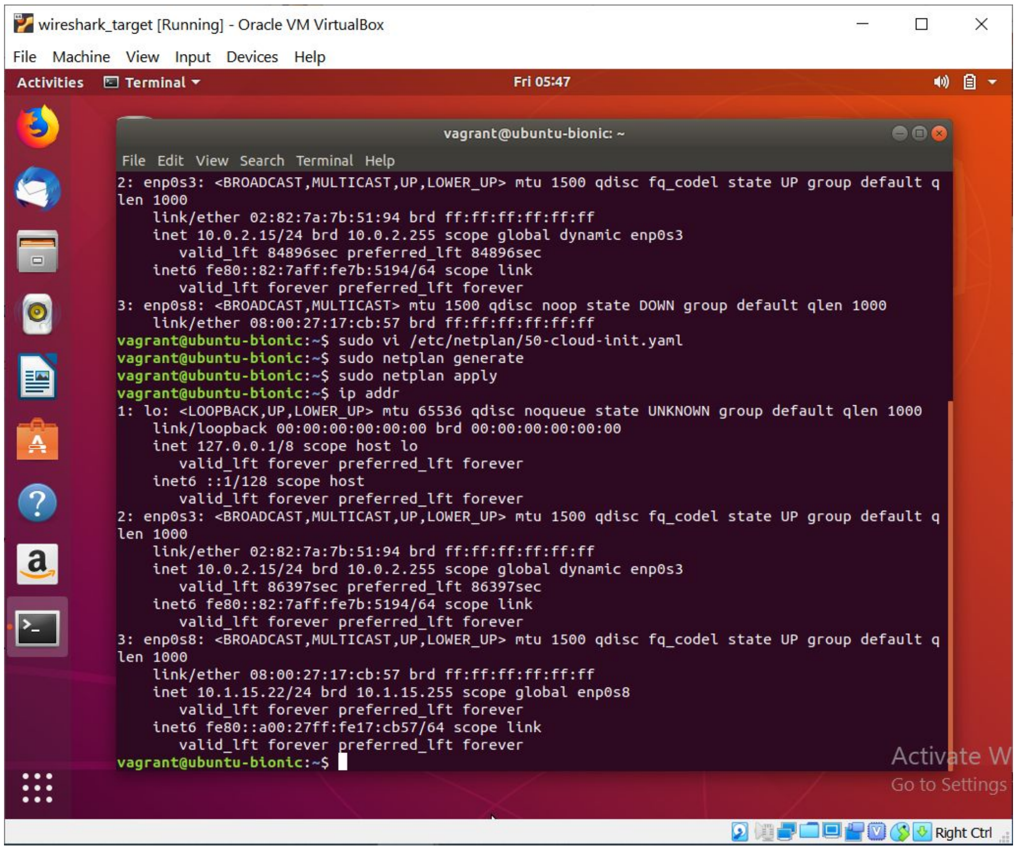
You can verify that the static IP address has been set:

```
vagrant@ubuntu-bionic:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
...
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 1000
    link/ether 02:82:7a:7b:51:94 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
    valid_lft 86398sec preferred_lft 86398sec
    inet6 fe80::82:7aff:fe7b:5194/64 scope link
    valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 1000
    link/ether 08:00:27:28:bc:5d brd ff:ff:ff:ff:ff:ff
    inet 10.1.15.21/24 brd 10.0.15.255 scope global enp0s8
```

```
        valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fe28:bc5d/64 scope link
        valid_lft forever preferred_lft forever
    vagrant@ubuntu-bionic:~$
```

You can do the same to wireshark_target with the IP address, 10.1.15.22:

- Modify the existing yaml file or create a new one (enp0s8.yaml).
- Issue the commands, netplan generate and netplan apply.


wireshark_target [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
Fri 05:47
vagrant@ubuntu-bionic: ~
File Edit View Search Terminal Help
Activate W
Go to Settings
Right Ctrl

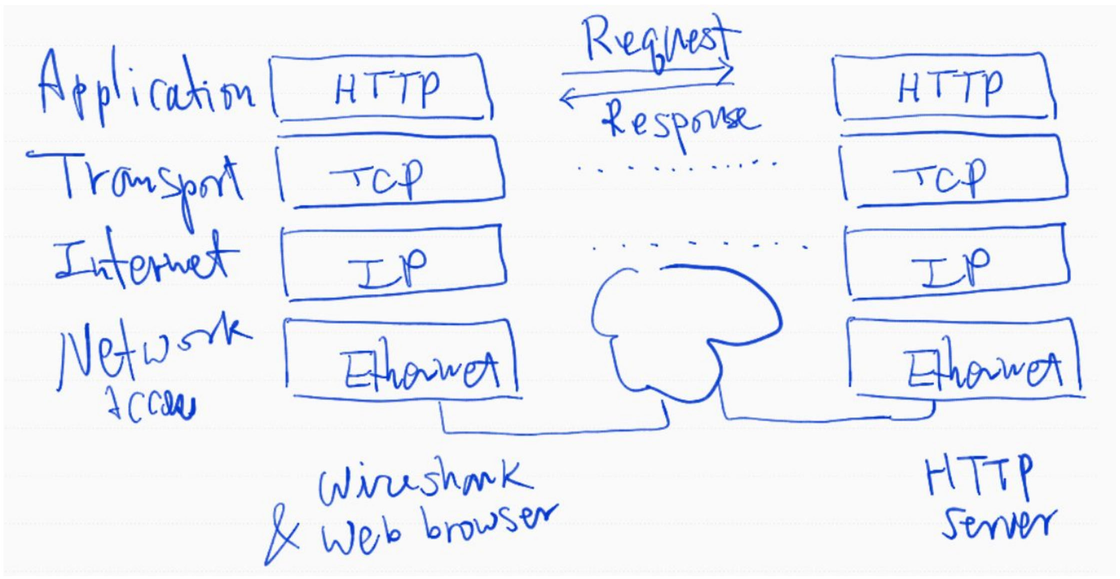
```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default q
len 1000
    link/ether 02:82:7a:7b:51:94 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 84896sec preferred_lft 84896sec
    inet6 fe80::82:7aff:fe7b:5194/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:17:cb:57 brd ff:ff:ff:ff:ff:ff
vagrant@ubuntu-bionic:~$ sudo vi /etc/netplan/50-cloud-init.yaml
vagrant@ubuntu-bionic:~$ sudo netplan generate
vagrant@ubuntu-bionic:~$ sudo netplan apply
vagrant@ubuntu-bionic:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default q
len 1000
    link/ether 02:82:7a:7b:51:94 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 86397sec preferred_lft 86397sec
    inet6 fe80::82:7aff:fe7b:5194/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default q
len 1000
    link/ether 08:00:27:17:cb:57 brd ff:ff:ff:ff:ff:ff
    inet 10.1.15.22/24 brd 10.1.15.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe17:cb57/64 scope link
       valid_lft forever preferred_lft forever
vagrant@ubuntu-bionic:~$
```

You can test network connectivity from 10.1.15.21 to 10.1.15 22 by using the ping command:

```
vagrant@ubuntu-bionic:~$ ping 10.1.15.22
PING 10.1.15.22 (10.1.15.22) 56(84) bytes of data.
64 bytes from 10.1.15.22: icmp_seq=1 ttl=64 time=1.68 ms
64 bytes from 10.1.15.22: icmp_seq=2 ttl=64 time=0.877 ms
64 bytes from 10.1.15.22: icmp_seq=3 ttl=64 time=1.01 ms
^C
--- 10.1.15.22 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2139ms
rtt min/avg/max/mdev = 0.877/1.192/1.689/0.357 ms
```

Wireshark on a virtual machine

It is common to use layered models in computer networks. However, it is often difficult to visualize and analyze what happens between two computers. Wireshark is a widely used protocol analysis tool in industry and academia. For example, Wireshark allows you to 'see' each layer (Ethernet, IP, TCP and Application Layer) associated with HTTP requests/responses, separately and collectively.



Now, we have two communicating Linux computers over a virtual network. On one virtual machine (wireshark_target), we run a simple web server. We install Wireshark on the other virtual machine and run a web browser to initiate some web traffic with Wireshark turned on.

You can install Wireshark using the command via SSH.

```
D:\wireshark>vagrant ssh
vagrant@ubuntu-bionic:~$ sudo apt update
vagrant@ubuntu-bionic:~$ sudo apt install wireshark
```

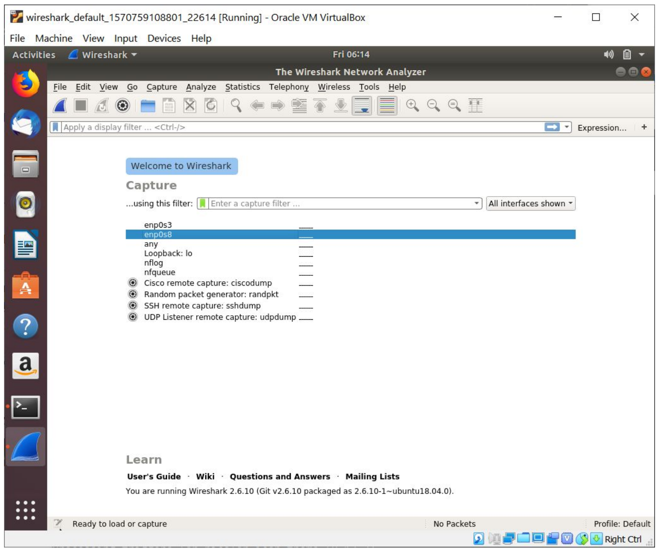
However, if you start Wireshark using the commands below outside virtual machine GUI, you will have an error, because it is built on the X Window System (X11):

```
vagrant@ubuntu-bionic:~$ sudo wireshark
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
qt.qpa.screen: QXcbConnection: Could not connect to display
Could not connect to any X display.
```

If you start Wireshark inside VirtualBox GUI using a terminal session, you will have an initial screen, as shown below:

```
$ sudo wireshark
```

Note that the interface name might be different, depending on the VirtualBox host. Also, depending on what you wish to monitor, you have to select (double click) a correct network interface. For external connectivity, you may wish to pick enp0s3. For traffic analysis between virtual machines, you need to pick enp0s8.


wireshark_default_1570759108801_22614 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Wireshark Fri 06:14
The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
Welcome to Wireshark
Capture
...using this filter: Enter a capture filter ...
All interfaces shown
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
Ready to load or capture
No Packets
Profile: Default
Right Ctrl

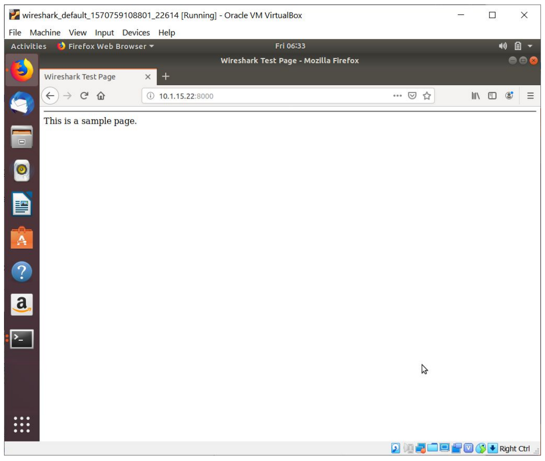
```
enp0s3
enp0s8
any
Loopback: lo
nflog
nfqueue
Cisco remote capture: ciscodump
Random packet generator: randpkt
SSH remote capture: sshdump
UDP Listener remote capture: udpdump
```

You are running Wireshark 2.6.10 (Git v2.6.10 packaged as 2.6.10-1~ubuntu18.04.0).

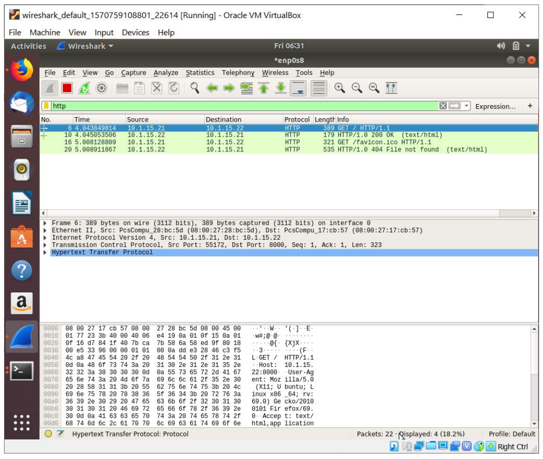
Seeing network protocols between two virtual machines

On the target virtual machine, you create a sample HTML (HyperText Markup Language) page, index.html at the vagrant home directory:

```
<HTML>
<HEAD><TITLE>Wireshark Test Page</TITLE></HEAD>
<P> This is a sample page.
</BODY>
</HTML>
```


wireshark_default_1570759108801_22614 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Firefox Web Browser Fri 06:33
Wireshark Test Page - Mozilla Firefox
Wireshark Test Page
10.1.15.22:8000
Right Ctrl

This is a sample page.


wireshark_default_1570759108801_22614 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Wireshark Fri 06:31
*enp0s8
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
Expression...
Hypertext Transfer Protocol: Protocol
Packets: 22 · Displayed: 4 (18.2%)
Profile: Default
Right Ctrl

```
No.   Time          Source       Destination  Protocol Length Info
    6 4.043849814   10.1.15.21   10.1.15.22   HTTP      389 GET / HTTP/1.1
   10 4.045053506   10.1.15.22   10.1.15.21   HTTP      179 HTTP/1.0 200 OK  (text/html)
   16 5.008128809   10.1.15.21   10.1.15.22   HTTP      321 GET /favicon.ico HTTP/1.1
   20 5.008911867   10.1.15.22   10.1.15.21   HTTP      535 HTTP/1.0 404 File not found  (text/html)

▸ Frame 6: 389 bytes on wire (3112 bits), 389 bytes captured (3112 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_28:bc:5d (08:00:27:28:bc:5d), Dst: PcsCompu_17:cb:57 (08:00:27:17:cb:57)
▸ Internet Protocol Version 4, Src: 10.1.15.21, Dst: 10.1.15.22
▸ Transmission Control Protocol, Src Port: 55172, Dst Port: 8000, Seq: 1, Ack: 1, Len: 323
▸ Hypertext Transfer Protocol

0000  08 00 27 17 cb 57 08 00  27 28 bc 5d 08 00 45 00   ··'··W·· '(·]··E·
0010  01 77 23 3b 40 00 40 06  e4 19 0a 01 0f 15 0a 01   ·w#;@·@· ········
0020  0f 16 d7 84 1f 40 7b ca  7b 58 6a 58 ed 9f 80 18   ·····@{· {XjX····
0030  00 e5 33 96 00 00 01 01  08 0a dd e3 28 46 c3 f5   ··3····· ····(F··
0040  4c a8 47 45 54 20 2f 20  48 54 54 50 2f 31 2e 31   L·GET /  HTTP/1.1
0050  0d 0a 48 6f 73 74 3a 20  31 30 2e 31 2e 31 35 2e   ··Host:  10.1.15.
0060  32 32 3a 38 30 30 30 0d  0a 55 73 65 72 2d 41 67   22:8000· ·User-Ag
0070  65 6e 74 3a 20 4d 6f 7a  69 6c 6c 61 2f 35 2e 30   ent: Moz illa/5.0
0080  20 28 58 31 31 3b 20 55  62 75 6e 74 75 3b 20 4c    (X11; U buntu; L
0090  69 6e 75 78 20 78 38 36  5f 36 34 3b 20 72 76 3a   inux x86 _64; rv:
00a0  36 39 2e 30 29 20 47 65  63 6b 6f 2f 32 30 31 30   69.0) Ge cko/2010
00b0  30 31 30 31 20 46 69 72  65 66 6f 78 2f 36 39 2e   0101 Fir efox/69.
00c0  30 0d 0a 41 63 63 65 70  74 3a 20 74 65 78 74 2f   0··Accep t: text/
00d0  68 74 6d 6c 2c 61 70 70  6c 69 63 61 74 69 6f 6e   html,app lication
```

And, you run a python HTTP server (a simple Web server) at the home directory:

```
$ python3 -m http.server
```

On the wireshark virtual machine, you should be running Wireshark and capturing the traffic already. Then, you bring up the browser (Firefox) and enter http://10.1.15.22:8000 at the address field.

You can 'see' live packets (HTTP, TCP, IP, Ethernet) with Wireshark in a virtual machine, as shown above right.

Exercises

1. Assuming you have just created a virtual machine with the command, vagrant init centos/7 and cloned the virtual machine twice. You want to assign two static IP addresses, 10.3.15.30 and 10.3.15.40 to the clones, using Host-only adapter in VirtualBox. Describe briefly the process.

2. Create two CentOS 7 virtual machines with static IP addresses (10.2.15.21 and 10.2.15.22) with Vagrant.