

Lab 8: Network virtualization with Virtualbox

Alexander Hoffmann

March 15, 2020

1 Configuration

For this lab, we will configure our virtual machines in a host-only network. Host-only networking can be thought of as a hybrid between the bridged and internal networking modes. As with bridged networking, the virtual machines can talk to each other and the host as if they were connected through a physical Ethernet switch. As with internal networking, a physical networking interface need not be present, and the virtual machines cannot talk to the world outside the host since they are not connected to a physical networking interface.

When host-only networking is used, Oracle VM VirtualBox creates a new software interface on the host which then appears next to your existing network interfaces. In other words, whereas with bridged networking an existing physical interface is used to attach virtual machines to, with host-only networking a new loopback interface is created on the host. And whereas with internal networking, the traffic between the virtual machines cannot be seen, the traffic on the loopback interface on the host can be intercepted.

1. First, we need to configure the virtual interfaces in the VirtualBox GUI. Here is the current configuration:

Name	IPv4 Address/Mask	IPv6 Address/Mask	DHCP Server
VirtualBox Host-Only Ethernet Adapter	192.168.11.1/24		<input type="checkbox"/> Enable
VirtualBox Host-Only Ethernet Adapter #2	192.168.22.1/24		<input type="checkbox"/> Enable
VirtualBox Host-Only Ethernet Adapter #3	192.168.33.1/24		<input type="checkbox"/> Enable

These are our three networks. Next, we'll manually configure the IP addresses inside the virtual machines. The command is as follows:

`ifconfig <interface> <ip-address> netmask <netmask> up`

Which yields the following results after configuration:

```
alex@wenger:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.11.1 netmask 255.255.255.0 broadcast 192.168.11.255
    inet6 fe80::a00:27ff:fea5:9659 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:a5:96:59 txqueuelen 1000 (Ethernet)
    RX packets 64 bytes 19664 (19.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 37 bytes 5722 (5.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

PC Router

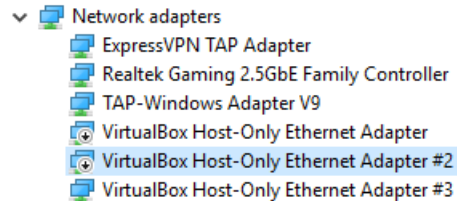
```
alex@wenger:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.11.10 netmask 255.255.255.0 broadcast 192.168.11.255
    inet6 fe80::a00:27ff:fe9b:4ed0 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:9b:4e:d0 txqueuelen 1000 (Ethernet)
    RX packets 67 bytes 19274 (19.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 38 bytes 6894 (6.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

PC1

```
alex@wenger:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.11.2 netmask 255.255.255.0 broadcast 192.168.11.255
    inet6 fe80::a00:27ff:fede:a3bc prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:de:a3:bc txqueuelen 1000 (Ethernet)
    RX packets 84 bytes 24208 (24.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 45 bytes 7636 (7.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Server

The next step is to disconnect the host machine from host-only network 1 and 2.



2. Now we do the same for the second interface.

```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.22.1 netmask 255.255.255.0 broadcast 192.168.22.255
    inet6 fe80::a00:27ff:fec7:7e7c prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:c7:7e:7c txqueuelen 1000 (Ethernet)
    RX packets 22 bytes 1992 (1.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 9 bytes 726 (726.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

PC Router

```
alex@wenger:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.22.10 netmask 255.255.255.0 broadcast 192.168.22.255
    inet6 fe80::a00:27ff:fefa:b3f1 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:fa:b3:f1 txqueuelen 1000 (Ethernet)
    RX packets 59 bytes 11412 (11.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 34 bytes 5008 (5.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

PC2

3. Finally, configure the third interface.

```
enp0s9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.33.2 netmask 255.255.255.0 broadcast 192.168.33.255
    inet6 fe80::a00:27ff:fe09:3dd1 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:09:3d:d1 txqueuelen 1000 (Ethernet)
    RX packets 117 bytes 10764 (10.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 13 bytes 1006 (1.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

PC Router

```
Ethernet adapter VirtualBox Host-Only Network #3:

Connection-specific DNS Suffix . : 
Link-local IPv6 Address . . . . . : fe80::1c07:7da8:b183:99c2%44
IPv4 Address. . . . . : 192.168.33.1
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
```

PC Host

2 Testing

1. Testing pings from various devices.

```
alex@wenger:~$ ping 192.168.11.1
PING 192.168.11.1 (192.168.11.1) 56(84) bytes of data.
64 bytes from 192.168.11.1: icmp_seq=1 ttl=64 time=0.435 ms
64 bytes from 192.168.11.1: icmp_seq=2 ttl=64 time=0.246 ms
64 bytes from 192.168.11.1: icmp_seq=3 ttl=64 time=0.220 ms
^C
--- 192.168.11.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2031ms
rtt min/avg/max/mdev = 0.220/0.300/0.435/0.096 ms
```

Ping PC Router from PC1

```
alex@wenger:~$ ping 192.168.22.1
PING 192.168.22.1 (192.168.22.1) 56(84) bytes of data.
64 bytes from 192.168.22.1: icmp_seq=1 ttl=64 time=0.409 ms
64 bytes from 192.168.22.1: icmp_seq=2 ttl=64 time=0.178 ms
64 bytes from 192.168.22.1: icmp_seq=3 ttl=64 time=0.244 ms
^C
--- 192.168.22.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2026ms
rtt min/avg/max/mdev = 0.178/0.277/0.409/0.097 ms
```

Ping PC Router from PC2

```
C:\Users\Pc>ping 192.168.33.1

Pinging 192.168.33.1 with 32 bytes of data:
Reply from 192.168.33.1: bytes=32 time<1ms TTL=128
Reply from 192.168.33.1: bytes=32 time<1ms TTL=128
Reply from 192.168.33.1: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.33.1:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Ping PC Router from PC Host

```
alex@wenger:~$ ping 192.168.22.10
connect: Network is unreachable
```

Ping PC2 from Server

```
C:\Users\Pc>ping 192.168.11.1

Pinging 192.168.11.1 with 32 bytes of data:
Request timed out.
Request timed out.

Ping statistics for 192.168.11.1:
    Packets: Sent = 2, Received = 0, Lost = 2 (100% loss),
```

Ping PC1 from PC Host

2. There are two pings that have failed:

- ping from Server to PC2
- ping from PC Host to PC1

Both these pings failed because we tried to access devices on different networks without advertising them in the first place. We need to add gateways in order to allow the devices on different networks to communicate with each other through the router.

3. We have to add default gateways on each device so that it forwards unknown traffic. Use this command:

```
route add default gw <ip-address>
```

Also, it is necessary to enable routing mode on PC Router using this command:

```
sysctl -w net.ipv4.ip_forward=1
```

On PC Host, which is a Windows computer, use this command:

```
route add <ip-address> mask <netmask> <gateway-ip-address>
```

4. Now let's test our pings.

```
alex@wenger:~$ ping 192.168.22.10
PING 192.168.22.10 (192.168.22.10) 56(84) bytes of data:
64 bytes from 192.168.22.10: icmp_seq=1 ttl=63 time=0.424 ms
64 bytes from 192.168.22.10: icmp_seq=2 ttl=63 time=0.409 ms
64 bytes from 192.168.22.10: icmp_seq=3 ttl=63 time=0.285 ms
64 bytes from 192.168.22.10: icmp_seq=4 ttl=63 time=0.356 ms
^C
--- 192.168.22.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3074ms
rtt min/avg/max/mdev = 0.285/0.368/0.424/0.057 ms
```

Ping PC2 Router from PC1

```
C:\Windows\system32>ping 192.168.22.10

Pinging 192.168.22.10 with 32 bytes of data:
Reply from 192.168.33.1: Destination host unreachable.
Reply from 192.168.22.10: bytes=32 time<1ms TTL=64
Reply from 192.168.22.10: bytes=32 time<1ms TTL=64
Reply from 192.168.22.10: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.22.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Ping PC2 from PC Host

```
alex@wenger:~$ ping 192.168.22.10
PING 192.168.22.10 (192.168.22.10) 56(84) bytes of data.
64 bytes from 192.168.22.10: icmp_seq=6 ttl=63 time=0.418 ms
64 bytes from 192.168.22.10: icmp_seq=7 ttl=63 time=0.435 ms
64 bytes from 192.168.22.10: icmp_seq=8 ttl=63 time=0.455 ms
^C
--- 192.168.22.10 ping statistics ---
8 packets transmitted, 3 received, 62% packet loss, time 7168ms
rtt min/avg/max/mdev = 0.418/0.436/0.455/0.015 ms
```

Ping PC2 from Server

3 DHCP Server

This first screenshot shows that all DHCP servers have been disabled in the VB interface manager.

5. First, edit the file `/etc/default/isc-dhcp-server` with root permissions:

```
INTERFACES="enp0s8 enp0s9"
```

The following entry defines the LAN and the router of the LAN. The IP-addresses 192.168.1.1 to 192.168.1.255 are typical for an intranet. Here only the range 192.168.1.1 to 192.168.1.254 are permitted.

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.1 192.168.1.254;
    option routers 192.168.1.1;
}
```

Attribute manually the first available address on each subnet to the gateway.

```
allow-hotplug enp0s8
iface enp0s8 inet static
    address 192.168.1.1
    netmask 255.255.255.0

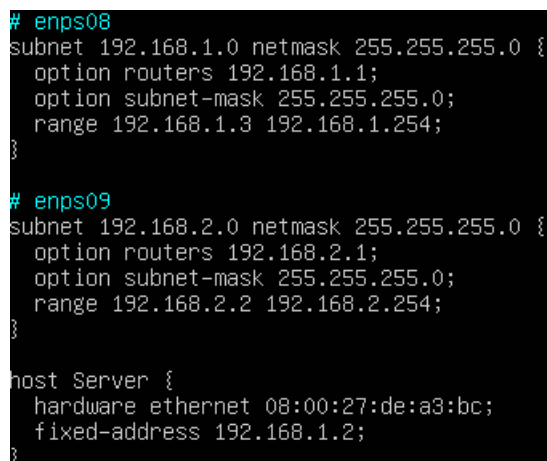
allow-hotplug enp0s9
iface enp0s9 inet static
    address 192.168.2.2
    netmask 255.255.255.0

allow-hotplug enp0s10
iface enp0s10 inet dhcp
```

Contents of `"/etc/network/interfaces"`

To assign a fixed address to a particular machine add a statement like the following to the configuration file. The cryptic number 00:0D:87:B3:AE:A6 is the hardware address of the interface of Server.

```
host Server {  
    hardware ethernet 00:0D:87:B3:AE:A6;  
    fixed-address 192.168.1.2;  
}
```

A screenshot of a terminal window showing the contents of the file /etc/dhcp/dhcpd.conf. The text is as follows:

```
# enps08  
subnet 192.168.1.0 netmask 255.255.255.0 {  
    option routers 192.168.1.1;  
    option subnet-mask 255.255.255.0;  
    range 192.168.1.3 192.168.1.254;  
}  
  
# enps09  
subnet 192.168.2.0 netmask 255.255.255.0 {  
    option routers 192.168.2.1;  
    option subnet-mask 255.255.255.0;  
    range 192.168.2.2 192.168.2.254;  
}  
  
host Server {  
    hardware ethernet 08:00:27:de:a3:bc;  
    fixed-address 192.168.1.2;  
}
```

Contents of "/etc/dhcp/dhcpd.conf"

To make all the changes effective, restart the DHCP daemon and reset the interfaces.

```
service isc-dhcp-server restart  
sudo ifdown enp0s8  
sudo ifup enp0s8  
sudo ifdown enp0s9  
sudo ifup enp0s9
```

Here is the current IP configuration:

```

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::a00:27ff:fec7:7e7c prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:c7:7e:7c txqueuelen 1000 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 55 bytes 3496 (3.4 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.2.2 netmask 255.255.255.0 broadcast 192.168.2.255
        inet6 fe80::a00:27ff:fe09:3dd1 prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:09:3d:d1 txqueuelen 1000 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 9 bytes 726 (726.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s10: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.33.2 netmask 255.255.255.0 broadcast 192.168.33.255
        inet6 fe80::a00:27ff:fe31:eaf8 prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:31:ea:f8 txqueuelen 1000 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 17 bytes 3462 (3.4 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

PC Router

```

alex@wenger:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.2 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::a00:27ff:fede:a3bc prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:de:a3:bc txqueuelen 1000 (Ethernet)
        RX packets 1071 bytes 124918 (124.9 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 233 bytes 31560 (31.5 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Server

Now let's test pinging on PC1 from PC2.

```

alex@wenger:~$ ping 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data:
64 bytes from 192.168.2.2: icmp_seq=1 ttl=63 time=0.418 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=63 time=0.477 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=63 time=0.423 ms
^C
--- 192.168.2.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2052ms
rtt min/avg/max/mdev = 0.418/0.439/0.477/0.031 ms

```

PC1 to PC2


```
alex@wenger:~$ ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.
64 bytes from 192.168.1.3: icmp_seq=1 ttl=63 time=0.432 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=63 time=0.432 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=63 time=0.482 ms
^C
--- 192.168.1.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2033ms
rtt min/avg/max/mdev = 0.432/0.448/0.482/0.033 ms
```

PC2 to PC1

4 HTTP Server

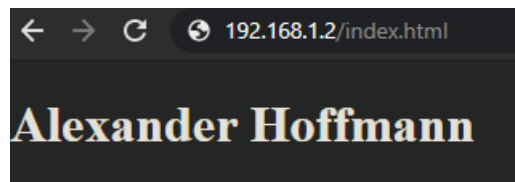
8. Instead of using `ssh`, we will be using `scp` which is a means of securely transferring computer files between a local host and a remote host or between two remote hosts. It is based on the Secure Shell (SSH) protocol.

```
scp alex@192.168.33.1:index.html /var/www/html
```

Now we need to reset the `apache2` server.

```
service apache2 reload
```

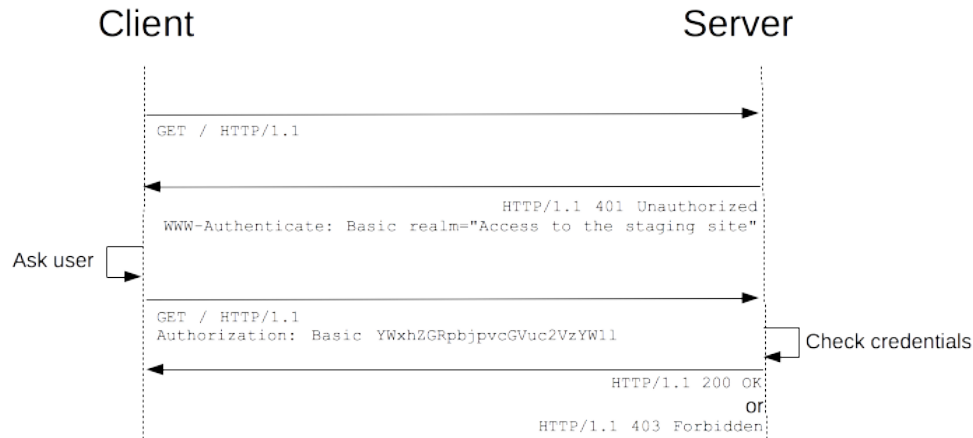
Next, open a web browser and type `192.168.1.2/index.html`. We get the following web page.



9. Capture with Wireshark an HTTP traffic on PC-Host.

http						
No.	Time	Source	Destination	Protocol	Length	Info
29	16.678111	192.168.33.1	192.168.1.2	HTTP	517	GET /tpvm2.html HTTP/1.1
31	16.680789	192.168.1.2	192.168.33.1	HTTP	503	HTTP/1.1 200 OK (text/html)

HTTP provides a general framework for access control and authentication. The most common HTTP authentication is based on the "Basic" schema. The screenshot below shows an introduction to the HTTP framework for request and reply as well as authentication.



Basic authentication HTTP protocol

5 FTP Server

5.1 TCP Understanding

In this section, we have to change the IP configuration of the lab. At ECE School, the private network is of the form 10.X.X.X whereas my home network has 192.168.1.X. Therefore, we'll simplify the network.

We want to filter out the file transfer traffic, therefore we show only the TCP traffic in Wireshark. To connect to the machine, use:

`ftp <machine-ip-address>`

At times we wish to copy files from a remote machine using anonymous FTP. When the remote machine asks for a login, type in the word **anonymous**.

```

C:\Users\Pc>ftp 192.168.244.2
Connected to 192.168.244.2.
220 (vsFTPd 3.0.3)
200 Always in UTF8 mode.
User (192.168.244.2:(none)): anonymous
230 Login successful.
ftp>
  
```

Anonymous login on the FTP Server

On the following screenshot, we can clearly see the connection process. On line 306, we sent a request for the user **anonymous** which was accepted on line 307.

298	1272.857650	192.168.244.1	192.168.244.2	TCP	66 57722 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1440
299	1272.857851	192.168.244.2	192.168.244.1	TCP	66 21 → 57722 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
300	1272.857894	192.168.244.1	192.168.244.2	TCP	54 57722 → 21 [ACK] Seq=1 Ack=1 Win=8192 Len=0
301	1272.858862	192.168.244.2	192.168.244.1	FTP	74 Response: 220 (vsFTPd 3.0.3)
302	1272.861054	192.168.244.1	192.168.244.2	FTP	68 Request: OPTS UTF8 ON
303	1272.861123	192.168.244.2	192.168.244.1	TCP	60 21 → 57722 [ACK] Seq=21 Ack=15 Win=64256 Len=0
304	1272.861202	192.168.244.2	192.168.244.1	FTP	80 Response: 200 Always in UTF8 mode.
305	1272.901301	192.168.244.1	192.168.244.2	TCP	54 57722 → 21 [ACK] Seq=15 Ack=47 Win=8146 Len=0
306	1277.059598	192.168.244.1	192.168.244.2	FTP	70 Request: USER anonymous
307	1277.091537	192.168.244.2	192.168.244.1	FTP	77 Response: 230 Login successful.
308	1277.132042	192.168.244.1	192.168.244.2	TCP	54 57722 → 21 [ACK] Seq=31 Ack=70 Win=8123 Len=0
320	1459.386554	192.168.244.1	192.168.244.2	FTP	82 Request: PORT 192,168,244,1,225,159
321	1459.386905	192.168.244.2	192.168.244.1	FTP	105 Response: 200 PORT command successful. Consider using PASV.
322	1459.390822	192.168.244.1	192.168.244.2	FTP	60 Request: NLST
323	1459.391169	192.168.244.2	192.168.244.1	TCP	74 20 → 57759 [SYN] Seq=0 Win=64240 Len=0 MSS=1440
324	1459.391231	192.168.244.1	192.168.244.2	TCP	66 57759 → 20 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
325	1459.391288	192.168.244.2	192.168.244.1	TCP	60 20 → 57759 [ACK] Seq=1 Ack=1 Win=64256 Len=0
326	1459.391403	192.168.244.2	192.168.244.1	FTP	93 Response: 150 Here comes the directory listing.
327	1459.391430	192.168.244.2	192.168.244.1	TCP	60 20 → 57759 [FIN, ACK] Seq=1 Ack=1 Win=64256 Len=0
328	1459.391441	192.168.244.1	192.168.244.2	TCP	54 57759 → 20 [ACK] Seq=1 Ack=2 Win=2107648 Len=0
329	1459.391536	192.168.244.2	192.168.244.1	FTP	78 Response: 226 Directory send OK.
330	1459.391552	192.168.244.1	192.168.244.2	TCP	54 57722 → 21 [ACK] Seq=65 Ack=184 Win=8009 Len=0

Now let's test the `ls -l` command and capture the traffic.

```
ftp> ls -l
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
```

Line 351 shows the requested command. After that, we receive the directory listing as displayed on line 355.

349	1648.965731	192.168.244.1	192.168.244.2	FTP	82 Request: PORT 192,168,244,1,225,175
350	1648.966083	192.168.244.2	192.168.244.1	FTP	105 Response: 200 PORT command successful. Consider using PASV.
351	1648.969981	192.168.244.1	192.168.244.2	FTP	63 Request: NLST -l
352	1648.970375	192.168.244.2	192.168.244.1	TCP	74 20 → 57775 [SYN] Seq=0 Win=64240 Len=0 MSS=1440
353	1648.970431	192.168.244.1	192.168.244.2	TCP	66 57775 → 20 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
354	1648.970545	192.168.244.2	192.168.244.1	TCP	60 20 → 57775 [ACK] Seq=1 Ack=1 Win=64256 Len=0
355	1648.970641	192.168.244.2	192.168.244.1	FTP	93 Response: 150 Here comes the directory listing.
356	1648.970657	192.168.244.2	192.168.244.1	TCP	60 20 → 57775 [FIN, ACK] Seq=1 Ack=1 Win=64256 Len=0
357	1648.970667	192.168.244.1	192.168.244.2	TCP	54 57775 → 20 [ACK] Seq=1 Ack=2 Win=2107648 Len=0
358	1648.970798	192.168.244.2	192.168.244.1	FTP	78 Response: 226 Directory send OK.
359	1648.970814	192.168.244.1	192.168.244.2	TCP	54 57722 → 21 [ACK] Seq=102 Ack=298 Win=7895 Len=0
360	1648.973100	192.168.244.1	192.168.244.2	TCP	54 57775 → 20 [FIN, ACK] Seq=1 Ack=2 Win=2107648 Len=0
361	1648.973164	192.168.244.2	192.168.244.1	TCP	60 20 → 57775 [ACK] Seq=2 Ack=2 Win=64256 Len=0

Next, we send the file to the server.

```
put <file>
```

```
ftp> put hello.txt
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
ftp: 1642200 bytes sent in 0.00Seconds 410550.00Kbytes/sec.
```

Let's take a look at the corresponding traffic.

No.	Time	Source	Destination	Protocol	Length	Info
2	4.091798	192.168.244.1	192.168.244.2	FTP	82	Request: PORT 192,168,244,1,226,185
3	4.092138	192.168.244.2	192.168.244.1	FTP	105	Response: 200 PORT command successful. Consider u
4	4.096334	192.168.244.1	192.168.244.2	FTP	70	Request: STOR hello.txt
5	4.096893	192.168.244.2	192.168.244.1	TCP	74	20 → 58041 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 S
6	4.096949	192.168.244.1	192.168.244.2	TCP	66	58041 → 20 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
7	4.097034	192.168.244.2	192.168.244.1	TCP	60	20 → 58041 [ACK] Seq=1 Ack=1 Win=64256 Len=0
8	4.097112	192.168.244.2	192.168.244.1	FTP	76	Response: 150 Ok to send data.
9	4.098603	192.168.244.1	192.168.244.2	FTP-DA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
10	4.098604	192.168.244.1	192.168.244.2	FTP-DA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
11	4.098604	192.168.244.1	192.168.244.2	FTP-DA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
12	4.098604	192.168.244.1	192.168.244.2	FTP-DA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
13	4.098605	192.168.244.1	192.168.244.2	FTP-DA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
14	4.098605	192.168.244.1	192.168.244.2	FTP-DA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
15	4.098605	192.168.244.1	192.168.244.2	FTP-DA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
16	4.098605	192.168.244.1	192.168.244.2	FTP-DA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
17	4.098606	192.168.244.1	192.168.244.2	FTP-DA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
18	4.098606	192.168.244.1	192.168.244.2	FTP-DA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
19	4.098716	192.168.244.2	192.168.244.1	TCP	60	20 → 58041 [ACK] Seq=1 Ack=13941 Win=55808 Len=0

The figure above corresponds to the first 19 frames of the exchange between the host and the remote FTP server.

10. Identify the connection establishment and connection release of this file transfer.

```

> Frame 5: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interf
> Ethernet II, Src: PcsCompu_b3:7d:98 (08:00:27:b3:7d:98), Dst: 0a:00:27:00:00
> Internet Protocol Version 4, Src: 192.168.244.2, Dst: 192.168.244.1
✦ Transmission Control Protocol, Src Port: 20, Dst Port: 58041, Seq: 0, Len: 0
  Source Port: 20
  Destination Port: 58041
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  Sequence number (raw): 1172669120
  [Next sequence number: 1 (relative sequence number)]
  Acknowledgment number: 0
  Acknowledgment number (raw): 0
  1010 .... = Header Length: 40 bytes (10)
> Flags: 0x002 (SYN)
  Window size value: 64240
  [Calculated window size: 64240]
  Checksum: 0x196d [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0

```

We analyze the packet establishing the connection down after. Let's explore the flags.

Flags: 0x002 (SYN)

```

000. .... = Reserved: Not set
...0 .... = Nonce: Not set

```

```

.... 0... .... = Congestion Window Reduced (CWR): Not set
.... .0... .... = ECN-Echo: Not set
.... ..0. .... = Urgent: Not set
.... ...0 .... = Acknowledgment: Not set
.... .... 0... = Push: Not set
.... .... .0.. = Reset: Not set
.... .... ..1. = Syn: Set
.... .... ...0 = Fin: Not set

```

The SYN flag synchronizes sequence numbers to initiate a TCP connection.

```

Ethernet II, Src: PcsCompu_b3:7d:98 (08:00:27:b3:7d:98),
        Dst: 0a:00:27:00:00:0c (0a:00:27:00:00:0c)
Destination: 0a:00:27:00:00:0c (0a:00:27:00:00:0c)
Source: PcsCompu_b3:7d:98 (08:00:27:b3:7d:98)
Type: IPv4 (0x0800)

```

The source MAC address corresponds to the MAC address of the FTP server as can be seen on the screenshot below.

```

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.244.2 netmask 255.255.255.0 broadcast 192.168.244.255
    inet6 fe80::a00:27ff:feb3:7d98 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:b3:7d:98 txqueuelen 1000 (Ethernet)
    RX packets 1466 bytes 1724947 (1.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 386 bytes 44830 (44.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

The destination MAC address corresponds to the virtual interface of the VirtualBox Host-Only Adapter.

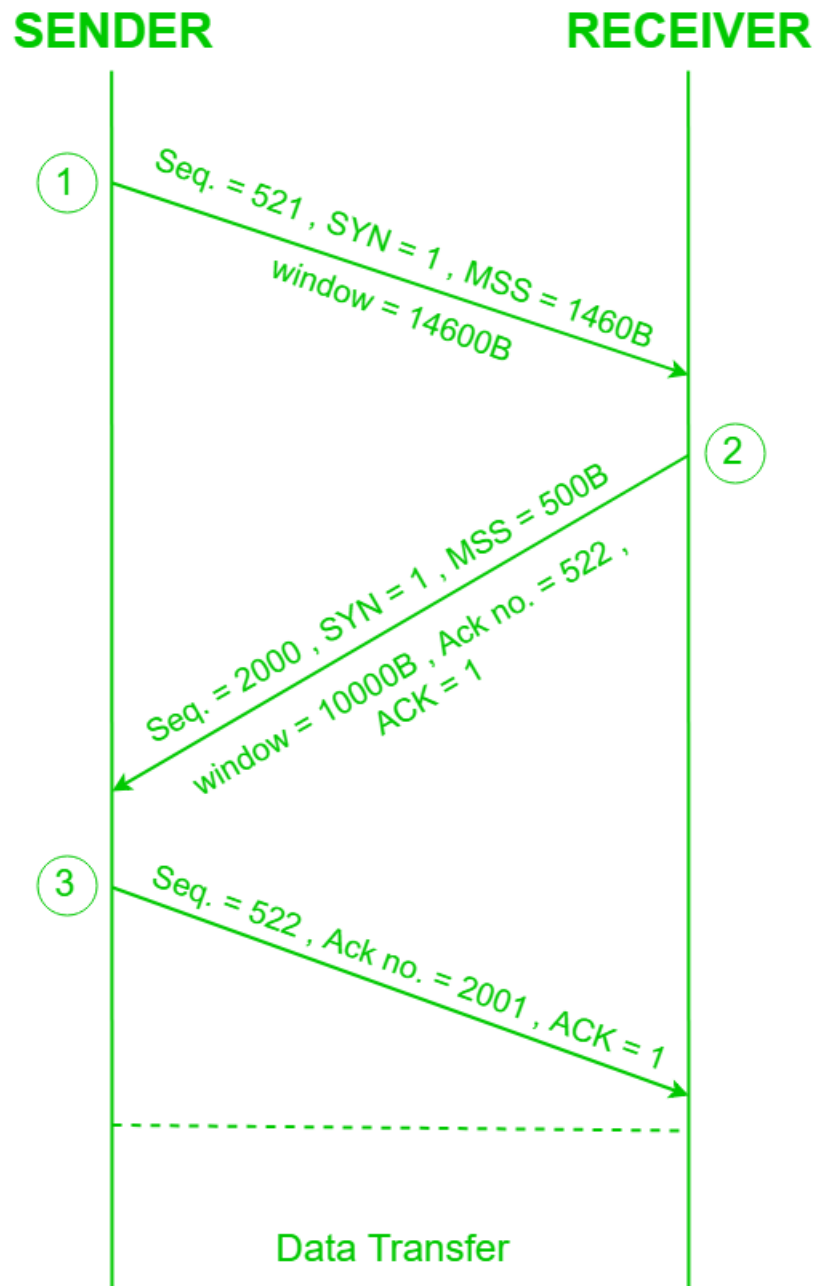
```

Ethernet adapter VirtualBox Host-Only Network:

Connection-specific DNS Suffix  . : 
Description . . . . . : VirtualBox Host-Only Ethernet Adapter
Physical Address. . . . . : 0A-00-27-00-00-0C
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::f584:63b5:7465:190%12(Preferred)
IPv4 Address. . . . . : 192.168.244.1(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 

```

The figure below describes the TCP connection establishment.



Now let's take a look at the closing packet.

```

  ▾ Ethernet II, Src: 0a:00:27:00:00:0c (0a:00:27:00:00:0c), Dst: PcsCompu_b3:7d:98 (08:00:27:b3:7d:98)
    > Destination: PcsCompu_b3:7d:98 (08:00:27:b3:7d:98)
    > Source: 0a:00:27:00:00:0c (0a:00:27:00:00:0c)
    Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: 192.168.244.1, Dst: 192.168.244.2
  ▾ Transmission Control Protocol, Src Port: 58041, Dst Port: 20, Seq: 1642201, Ack: 1, Len: 0
    Source Port: 58041
    Destination Port: 20
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence number: 1642201 (relative sequence number)
    Sequence number (raw): 1374930941
    [Next sequence number: 1642202 (relative sequence number)]
    Acknowledgment number: 1 (relative ack number)
    Acknowledgment number (raw): 1172669121
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x011 (FIN, ACK)
    Window size value: 8233
    [Calculated window size: 2107648]
    [Window size scaling factor: 256]
    Checksum: 0x60ef [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0

```

Flags: 0x011 (FIN, ACK)

```

000. .... = Reserved: Not set
...0 .... = Nonce: Not set
.... 0... = Congestion Window Reduced (CWR): Not set
.... .0.. = ECN-Echo: Not set
.... ..0. = Urgent: Not set
.... ...1 = Acknowledgment: Set
.... .... 0... = Push: Not set
.... .... .0.. = Reset: Not set
.... .... ..0. = Syn: Not set
.... .... ...1 = Fin: Set

```

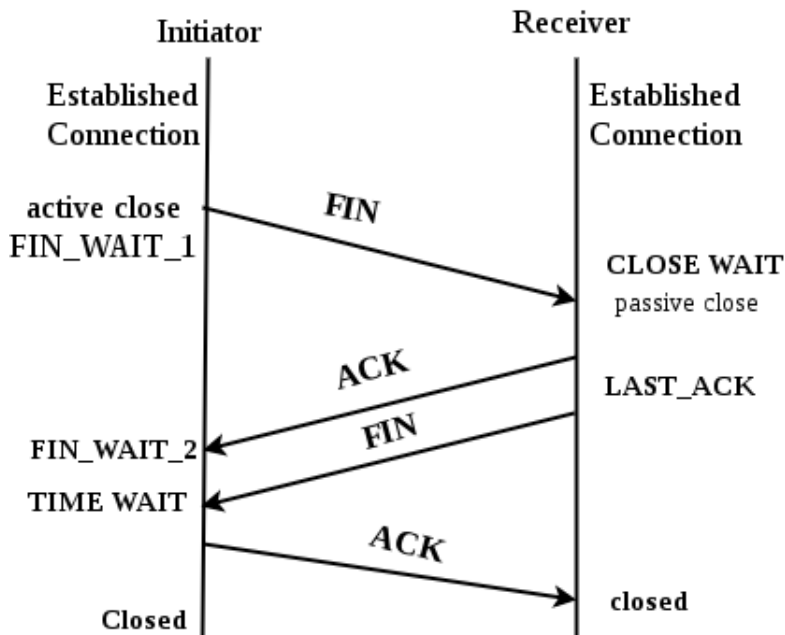
The FIN flag indicates the end of data transmission to finish a TCP connection. This frame is also an acknowledgment.

```

Ethernet II, Src: 0a:00:27:00:00:0c (0a:00:27:00:00:0c),
      Dst: PcsCompu_b3:7d:98 (08:00:27:b3:7d:98)
Destination: PcsCompu_b3:7d:98 (08:00:27:b3:7d:98)
Source: 0a:00:27:00:00:0c (0a:00:27:00:00:0c)
Type: IPv4 (0x0800)

```

The figure below describes the TCP connection termination.



11. Here are the 10 first data segments.

9	4.098603	192.168.244.1	192.168.244.2	FTP-DATA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
10	4.098604	192.168.244.1	192.168.244.2	FTP-DATA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
11	4.098604	192.168.244.1	192.168.244.2	FTP-DATA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
12	4.098604	192.168.244.1	192.168.244.2	FTP-DATA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
13	4.098605	192.168.244.1	192.168.244.2	FTP-DATA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
14	4.098605	192.168.244.1	192.168.244.2	FTP-DATA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
15	4.098605	192.168.244.1	192.168.244.2	FTP-DATA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
16	4.098605	192.168.244.1	192.168.244.2	FTP-DATA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
17	4.098606	192.168.244.1	192.168.244.2	FTP-DATA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)
18	4.098606	192.168.244.1	192.168.244.2	FTP-DATA	1448	FTP Data: 1394 bytes (PORT) (STOR hello.txt)

Here are the first two packets.

Transmission Control Protocol, Src Port: 58041, Dst Port: 20,
Seq: 1, Ack: 1, Len: 1394

```

Source Port: 58041
Destination Port: 20
[Stream index: 1]
[TCP Segment Len: 1394]
Sequence number: 1      (relative sequence number)
Sequence number (raw): 1373288741
[Next sequence number: 1395      (relative sequence number)]
Acknowledgment number: 1      (relative ack number)
Acknowledgment number (raw): 1172669121
  
```



```

0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
  000. .... = Reserved: Not set
  ....0 .... = Nonce: Not set
  .... 0... = Congestion Window Reduced (CWR): Not set
  .... .0.. = ECN-Echo: Not set
  .... ..0. = Urgent: Not set
  .... ...1 = Acknowledgment: Set
  .... .... 0... = Push: Not set
  .... .... .0.. = Reset: Not set
  .... .... ..0. = Syn: Not set
  .... .... ...0 = Fin: Not set
  [TCP Flags: .....A....]
Window size value: 8233
[Calculated window size: 2107648]
[Window size scaling factor: 256]
Checksum: 0x5e93 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
[SEQ/ACK analysis]
  [iRTT: 0.000141000 seconds]
  [Bytes in flight: 1394]
  [Bytes sent since last PSH flag: 1394]
[Timestamps]
TCP payload (1394 bytes)

```

```

Transmission Control Protocol, Src Port: 58041, Dst Port: 20,
                               Seq: 1395, Ack: 1, Len: 1394

```

```

Source Port: 58041
Destination Port: 20
[Stream index: 1]
[TCP Segment Len: 1394]
Sequence number: 1395      (relative sequence number)
Sequence number (raw): 1373290135
[Next sequence number: 2789      (relative sequence number)]
Acknowledgment number: 1    (relative ack number)
Acknowledgment number (raw): 1172669121
0101 .... = Header Length: 20 bytes (5)

```

```

Flags: 0x010 (ACK)
  000. .... = Reserved: Not set
  ...0 .... = Nonce: Not set
  .... 0... = Congestion Window Reduced (CWR): Not set
  .... .0.. = ECN-Echo: Not set
  .... ..0. = Urgent: Not set
  .... ...1 = Acknowledgment: Set
  .... .... 0... = Push: Not set
  .... .... .0.. = Reset: Not set
  .... .... ..0. = Syn: Not set
  .... .... ...0 = Fin: Not set
[TCP Flags: .....A....]
Window size value: 8233
[Calculated window size: 2107648]
[Window size scaling factor: 256]
Checksum: 0xeb23 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
[SEQ/ACK analysis]
  [iRTT: 0.000141000 seconds]
  [Bytes in flight: 2788]
  [Bytes sent since last PSH flag: 2788]
[Timestamps]
TCP payload (1394 bytes)

```

The size of the payload is 1394 bytes. The sequence number starts with 1 and increases by 1394 bytes at every packet sent. That's why we have 1, 1395, 2789, etc. as sequence number.

12. Identify the segments that acknowledge the reception of these segments.

```

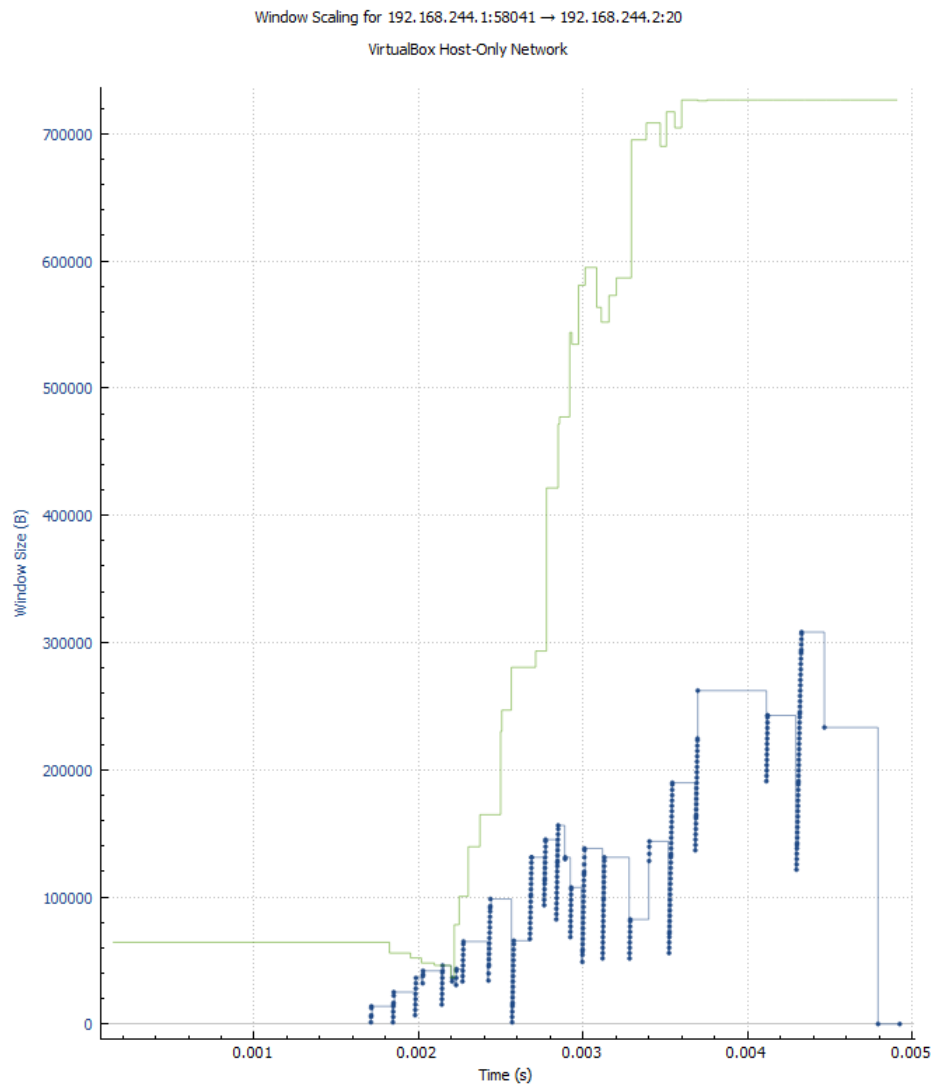
> Frame 19: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\
> Ethernet II, Src: PcsCompu_b3:7d:98 (08:00:27:b3:7d:98), Dst: 0a:00:27:00:00:0c (0a:00:27
> Internet Protocol Version 4, Src: 192.168.244.2, Dst: 192.168.244.1
▼ Transmission Control Protocol, Src Port: 20, Dst Port: 58041, Seq: 1, Ack: 13941, Len: 0
  Source Port: 20
  Destination Port: 58041
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 1 (relative sequence number)
  Sequence number (raw): 1172669121
  [Next sequence number: 1 (relative sequence number)]
  Acknowledgment number: 13941 (relative ack number)
  Acknowledgment number (raw): 1373302681
  0101 .... = Header Length: 20 bytes (5)
> Flags: 0x010 (ACK)
  Window size value: 436
  [Calculated window size: 55808]
  [Window size scaling factor: 128]
  Checksum: 0x57e2 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
> [SEQ/ACK analysis]
> [Timestamps]

```

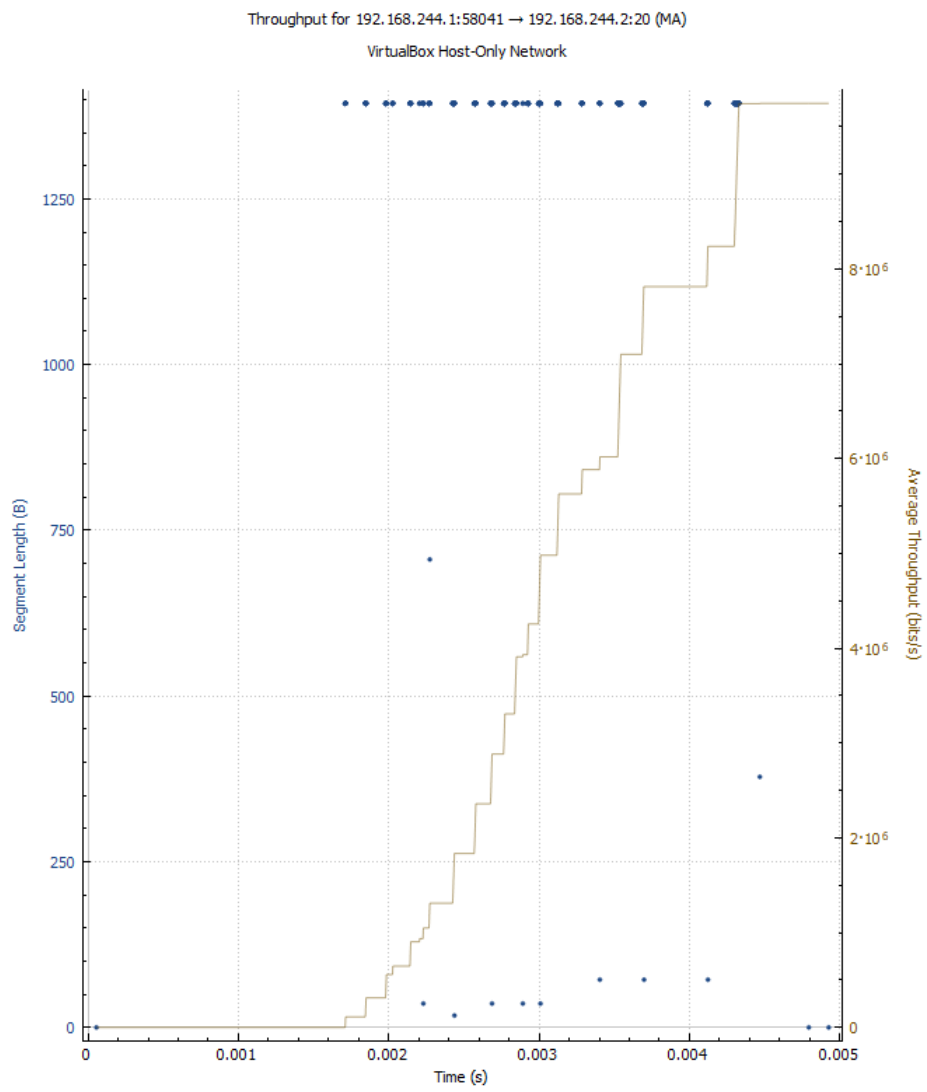
This frame is an acknowledgment for the 10 first packets.

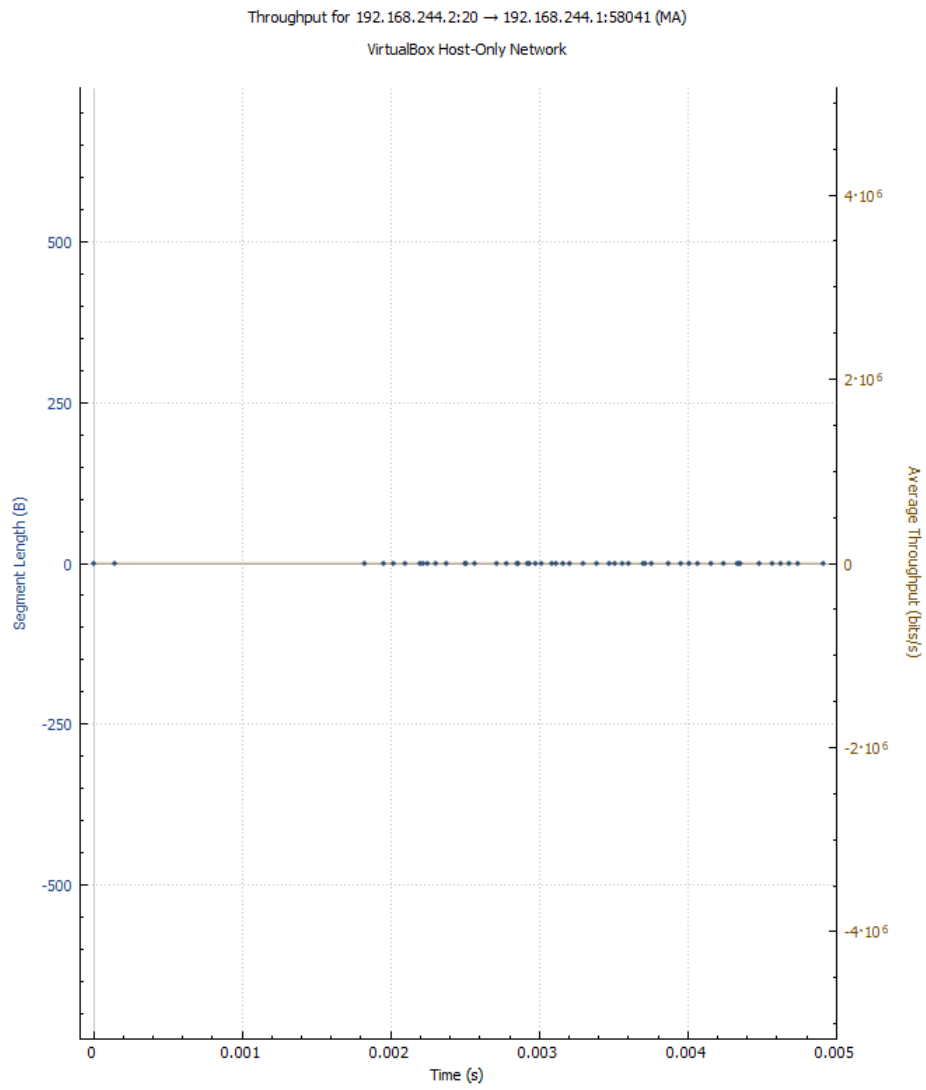
13. There is no re-transmission of any frame. This would be seen if we had a frame with the same sequence number.

14. Study and analyze the impact of the receiver's buffer space on the sender (based on window size advertisement). Display the window scaling graph.



15. Display throughput graphs in both connection directions.





5.2 FTP Understanding

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.244.1	192.168.244.2	FTP	81	Request: PORT 192,168,244,1,243,31
2	0.000263	192.168.244.2	192.168.244.1	FTP	105	Response: 200 PORT command successful. Consider usin
3	0.005919	192.168.244.1	192.168.244.2	FTP	70	Request: RETR file1.txt
4	0.006159	192.168.244.2	192.168.244.1	TCP	74	20 → 62239 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK
5	0.006216	192.168.244.1	192.168.244.2	TCP	66	62239 → 20 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MS
6	0.006303	192.168.244.2	192.168.244.1	TCP	60	20 → 62239 [ACK] Seq=1 Ack=1 Win=64256 Len=0
7	0.006391	192.168.244.2	192.168.244.1	FTP	120	Response: 150 Opening BINARY mode data connection fo
8	0.006441	192.168.244.2	192.168.244.1	FTP-DATA	60	FTP Data: 6 bytes (PORT) (RETR file1.txt)
9	0.006473	192.168.244.2	192.168.244.1	TCP	60	20 → 62239 [FIN, ACK] Seq=7 Ack=1 Win=64256 Len=0
10	0.006483	192.168.244.1	192.168.244.2	TCP	54	62239 → 20 [ACK] Seq=1 Ack=8 Win=2107648 Len=0
11	0.006594	192.168.244.2	192.168.244.1	FTP	78	Response: 226 Transfer complete.
12	0.006604	192.168.244.1	192.168.244.2	TCP	54	62089 → 21 [ACK] Seq=44 Ack=142 Win=7452 Len=0
13	0.011642	192.168.244.1	192.168.244.2	TCP	54	62239 → 20 [FIN, ACK] Seq=1 Ack=8 Win=2107648 Len=0
14	0.011749	192.168.244.2	192.168.244.1	TCP	60	20 → 62239 [ACK] Seq=8 Ack=2 Win=64256 Len=0
62	300.008946	192.168.244.2	192.168.244.1	FTP	68	Response: 421 Timeout.
63	300.008989	192.168.244.2	192.168.244.1	TCP	60	21 → 62089 [FIN, ACK] Seq=156 Ack=44 Win=502 Len=0
64	300.009008	192.168.244.1	192.168.244.2	TCP	54	62089 → 21 [ACK] Seq=44 Ack=157 Win=7438 Len=0