Lab 08 - Networking

1. *Although in this virtual environment we only have on single machine "under the same network", what could be the reason to allocate a random MAC address in a virtual environment of more than one virtual machine? Consider what would happen if the same value was in use by more than one system.*

More than one MAC address would wreak havoc at layer two: neither of the virtual machines would be able to network effectively as a result of collision, packet loss. Duplicate MAC addresses are fine as long as they're separated by a router.

2. *What is the current IP-address given to the system and the MAC address allocated to the Virtual Box?*

```
[root@UWS ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.5.246.135  netmask 255.255.0.0  broadcast 10.5.255.255
        ether 02:0a:6d:c0:9c:47  txqueuelen 1000  (Ethernet)
        RX packets 3097  bytes 241199 (235.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14  bytes 2295 (2.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 2  bytes 140 (140.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2  bytes 140 (140.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

The IPV4 address is `10.5.246.135` and the MAC address is `02:0a:6d:c0:9c:47`.

3. *Summarize in your own words what the command `ifconfig` does.*

`ifconfig` is a tool for configuring network devices. Interfaces can be enabled and disabled by passing the `up` or `down` arguments respectively, and IP addresses can be assigned. With no arguments it displays the settings for each available ethernet adapter, with the `-a` argument it displays available and disabled adapters, and individual NICs can be accessed by passing their names as arguments, e.g. `ifconfig eth0`.

4. *How many individual systems can a subnet with netmask `255.255.240.0` accommodate?*

14

5. *What is the meaning of the double exclamation mark `!!` in Unix-like operating systems?*

`!!` repeats the last command.

6. *Explain the `ping` command in your own words. What do the summary entries min/avg/max/mdev represent?*

`ping` is a network diagnostic tool which allows the user to communicate with another computer by passing in the destination IP address. By default the programme sends an ICMP echo request message each second until the user interrupts operation. The min, avg, max, and mdev values represent the minimum, average, maximum, round trip time, mdev represents the standard deviation.

7. *Explain why in both cases, the summary output is very similar, for each of the min/avg/max and mdev values.*

The question is ambiguous. There exists no such environment variable as `IP_loc_machine`. Running `ping` against my own IP address both internal external yields no results.

```
[root@UWS ~]# IP_loc_machine
-sh: IP_loc_machine: command not found
[root@UWS ~]# ping 80.229.232.182
PING 80.229.232.182 (80.229.232.182) 56(84) bytes of data.
^C
--- 80.229.232.182 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2071ms
[student@UWS ~]$ ping 192.168.1.138
PING 192.168.1.138 (192.168.1.138) 56(84) bytes of data.
^C
--- 192.168.1.138 ping statistics ---
14 packets transmitted, 0 received, 100% packet loss, time 13530ms
```

Further, there no such environment variable exists and there are no instructions to set an environment variable:

```
[root@UWS ~]# IP_loc_machine
-sh: IP_loc_machine: command not found
```

Below, the result when running `ping` against the local loopback and IPV4 address:

```
[root@UWS ~]# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.000 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.000 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.000 ms
^C
--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2062ms
rtt min/avg/max/mdev = 0.000/0.000/0.000/0.000 ms
[root@UWS ~]# ping 10.5.67.79
PING 10.5.67.79 (10.5.67.79) 56(84) bytes of data.
64 bytes from 10.5.67.79: icmp_seq=1 ttl=64 time=0.000 ms
64 bytes from 10.5.67.79: icmp_seq=2 ttl=64 time=0.000 ms
64 bytes from 10.5.67.79: icmp_seq=3 ttl=64 time=0.000 ms
^V^C
--- 10.5.67.79 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.000/0.000/0.000/0.000 ms
```

`127.0.0.1` represents a loopback address. Running `ping` against it is simply testing the NIC network stack, which is functionally identical to running `ping` against the local IP address of the device (in this case, `10.5.67.79`).

8. *Given that the loopback `127.0.0.1` is your `IP_loc_machine` kernel interface, why are both summary outputs not identical? Give a short explanation.*

The question makes no sense given the answer to question 7.

9. *Do you see an output? if not, review the network settings or restart the virtual machine.*

```
[root@UWS ~]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=168 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=175 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=168 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2010ms
rtt min/avg/max/mdev = 168.000/170.333/175.000/3.316 ms
```

10. *How much longer are the times now for min/avg/max and mdev?*

The RTT when running `ping` against the local machine are zero. When running `ping` against the external IP `8.8.8.8` the times are `rtt min/avg/max/mdev = 168.000/170.333/175.000/3.316 ms`.

This question makes little sense in light of the answer to question 7. Please see Q7!

11. *Why are the times longer for the remote machine?*

Physics!

12. *Summarise the `route` command in your own words.*

The `route` command alows users to view and interact with the routing table.

13. *What are the Destinations, Gateways and Genmasks in the kernel?*

Destination represents the destination host or network, a gateway the address to access that destination, and genmask is the subnet mask of the destination network.

14. *How many packets have been received OK and transmitted error free by your loopback interface?*

```
[root@UWS ~]# netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0      1500     8022      0      0 0            66      0      0      0 BMRU
lo       65536       22      0      0 0            22      0      0      0 LRU
```

22

15. *Is your ethernet a point-to-point connection?*

```
[root@UWS ~]# netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0      1500     8022      0      0 0            66      0      0      0 BMRU
lo       65536       22      0      0 0            22      0      0      0 LRU
```

No.

16. *Are loopback and ethernet broadcasting?*

```
[root@UWS ~]# netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0      1500     8022      0      0 0           66      0      0      0 BMRU
lo       65536       22      0      0 0           22      0      0      0 LRU
```

A broadcast address has been set for `eth0`.

17. *How many packages have been dropped on your system by the ethernet interface?*

```
[root@UWS ~]# netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0      1500     8022      0      0 0           66      0      0      0 BMRU
lo       65536       22      0      0 0           22      0      0      0 LRU
```

Zero.

18. *What are the local and Foreign addresses for the `ESTABLISHED` net-connection?*

```
[root@UWS ~]# netstat -na | head -2; netstat -na | grep 'ESTABLISHED'
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:53710         127.0.0.1:22            ESTABLISHED
tcp        0      0 127.0.0.1:22            127.0.0.1:53710         ESTABLISHED
```

19. *How many TCP (protocol based) connections are open at the moment?*

```
[root@UWS ~]# netstat --statistics
-- snip --
Tcp:
    1 active connection openings
    1 passive connection openings
    0 failed connection attempts
    0 connection resets received
    2 connections established
    747 segments received
    747 segments sent out
    0 segments retransmitted
    0 bad segments received
    0 resets sent
-- snip --
```

Two. One passive, one active.

20. *How many segments have been received and send via TCP/IP?*

```
[root@UWS ~]# netstat --statistics
-- snip --
Tcp:
    1 active connection openings
    1 passive connection openings
    0 failed connection attempts
    0 connection resets received
    2 connections established
```

```
    747 segments received
    747 segments sent out
    0 segments retransmitted
    0 bad segments received
    0 resets sent
-- snip --
```

747 received, 747 sent.

21. *Use your established knowledge to explain the command that you just issued on the remote system. What was the purpose of the semicolon?*

```
[root@UWS /]# cd / ; find /
-- snip --
[root@UWS ~]# netstat --statistics
-- snip --
Tcp:
    1 active connection openings
    1 passive connection openings
    0 failed connection attempts
    0 connection resets received
    2 connections established
    4548 segments received
    4548 segments sent out
    0 segments retransmitted
    0 bad segments received
    0 resets sent
-- snip --
```

Taking the question as wrote, the command `cd / ; find /` first performs `cd` with the root directory as the target, the semicolon separates the first command from the next command (which *will* be executed as `;` is not a conditional statement), and `find /` finds and displays the path of every file on the system.

22. *How many segments have been received and send via TCP/IP now? (I hope you see the difference.)*

```
[root@UWS /]# cd / ; find /
-- snip --
[root@UWS ~]# netstat --statistics
-- snip --
Tcp:
    1 active connection openings
    1 passive connection openings
    0 failed connection attempts
    0 connection resets received
    2 connections established
    4548 segments received
    4548 segments sent out
    0 segments retransmitted
    0 bad segments received
    0 resets sent
-- snip --
```

I do! 4,548 segments both received and sent out.