

Hands-on IV 1st - data link layer -

Information Network I
8 May. 2015

Let's use arp command!!

What's arp command

- arp command handles arp table on your PC.
- arp table means the list of MAC address, which is used to identify the PC in same L2 network.
- It is used to communicate with them on L2 network.

Let's start

- use terminal on Guest OS and Host OS.
- Firstly, please use arp command on both OS.
 - cygwin on Windows or terminal on linux or os x ...

```
$ arp -a (For mac, cygwin)
```

```
$ arp (For CentOS)
```
- This command shows arp table in your PC.

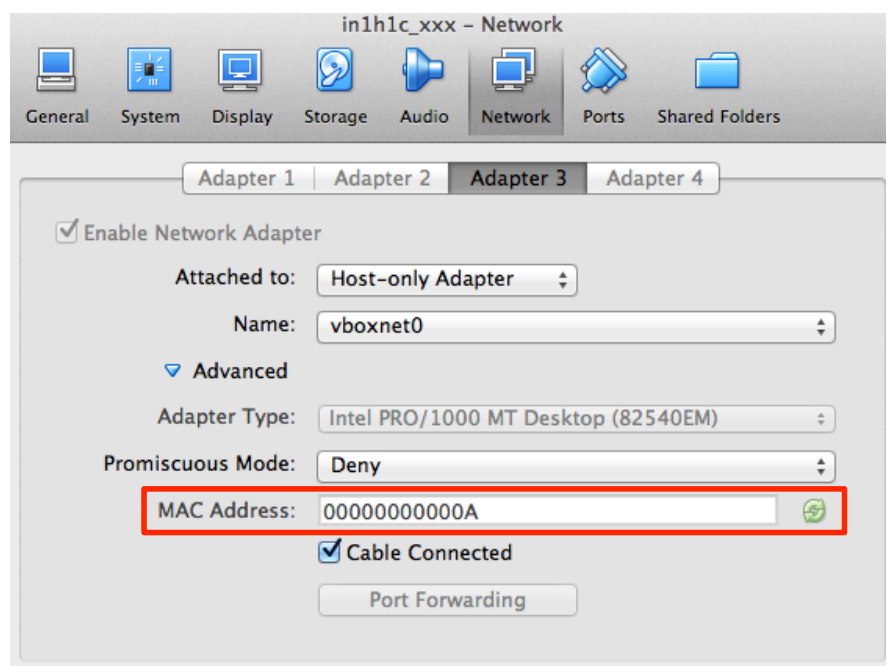
Result

- showing arp table in your PC
 - “arp -a” or “arp” shows all of your PC’s arp table
 - Please check it in detail.
- No output is OK.
 - It means there are no mac address list.

```
% arp -a
juniper-core2-v124.naist.jp (163.221.124.1) at 10:e:7e:c9:cb:f0 on en1 ifscope [ethernet]
aruba-bem.naist.jp (163.221.124.4) at 0:b:86:13:61:0 on en1 ifscope [ethernet]
naist-wavenet124-024.naist.jp (163.221.124.24) at 18:2a:7b:69:41:dd on en1 ifscope [ethernet]
naist-wavenet124-026.naist.jp (163.221.124.26) at ec:35:86:34:82:7e on en1 ifscope [ethernet]
naist-wavenet124-050.naist.jp (163.221.124.50) at b8:8d:12:d:7a:6a on en1 ifscope [ethernet]
naist-wavenet124-061.naist.jp (163.221.124.61) at 40:30:4:55:1d:71 on en1 ifscope [ethernet]
naist-wavenet124-070.naist.jp (163.221.124.70) at 84:38:35:63:d7:28 on en1 ifscope [ethernet]
naist-wavenet124-076.naist.jp (163.221.124.76) at 30:a8:db:70:59:35 on en1 ifscope [ethernet]
naist-wavenet124-091.naist.jp (163.221.124.91) at 4:e5:36:14:cc:15 on en1 ifscope [ethernet]
naist-wavenet124-092.naist.jp (163.221.124.92) at 70:11:24:3a:b3:7a on en1 ifscope [ethernet]
naist-wavenet124-095.naist.jp (163.221.124.95) at 18:2a:7b:69:41:dd on en1 ifscope [ethernet]
```

Change Guest OS's MAC address

- Choose your virtual os.
- From Setting, choose network.
- You can change MAC address from here.
- Please set it freely.
- This change can only be applied when that VM is shut down.



Let's Check it

- From your guest OS, please run :

\$ ifconfig <interface name>

“<interface name>” specifies the interface with the changed MAC address

Setting up a static entry in the arp table

- Arp table is made dynamically. If we want to set it statically, we run the following command:

\$ sudo arp -s <ip address> <mac address>

– This <ip address> is needed to belong to same network on your virtual box network.

- If you use 192.168.56.0/24 on Guest OS, please use 192.168.56.x
- 00:00:00:00:00:00 and FF:FF:FF:FF:FF:FF are reserved for special purposes.
- After setting, please check it.

Let's check it

```
sudo arp -s 192.168.56.2 00:00:00:00:00:0F
Password:

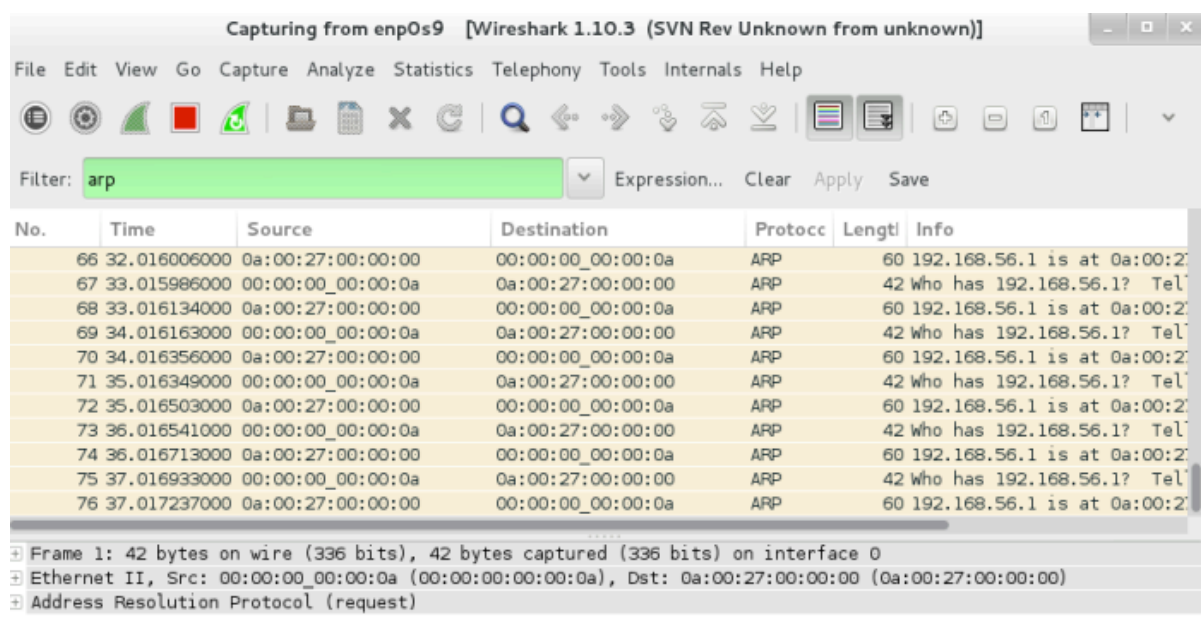
arp -i vboxnet0 -a
? (192.168.56.1) at a:0:27:0:0:0 on vboxnet0 ifscope permanent [ethernet]
? (192.168.56.2) at 0:0:0:0:0:f on vboxnet0 permanent [ethernet]
? (192.168.56.101) at 0:0:0:0:0:a on vboxnet0 ifscope [ethernet]
```

```
in1@localhost:~
File Edit View Search Terminal Help
[in1@localhost ~]$ arp
Address          Hwtype  Hwaddress      Flags Mask      Iface
192.168.56.1     ether   08:00:27:2f:03:0a  C              enp0s9
[in1@localhost ~]$ sudo arp -s 192.168.56.2 00:00:00:00:00:0F
[sudo] password for in1:
[in1@localhost ~]$ arp
Address          Hwtype  Hwaddress      Flags Mask      Iface
192.168.56.1     ether   08:00:27:2f:03:0a  C              enp0s9
192.168.56.2     ether   00:00:00:00:00:0f  CM             enp0s9
[in1@localhost ~]$
```



Capture an arp packet

- Please type this command on Guest OS:
`$ arping -I enp03s 192.168.56.1`
– It means asking who has 192.168.56.1.
- Please use wireshark to capture the arp packet on Guest OS.



Is ICMP(ping) using MAC address?

- On Host OS, please make new arp entry,
\$ arp -s 192.168.56.200 <MAC address using at Guest OS>
- After setting, on Host OS:
\$ ping 192.168.56.200
- On your Guest OS, please check the packet using wireshark/tcpdump for vboxnet adapter.

Guest OS

```
File Edit View Search Terminal Help
[ini@localhost ~]$ ifconfig
enp0s9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::200:ff:fe00:a prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:00:00:0a txqueuelen 1000 (Ethernet)
    RX packets 50 bytes 10462 (10.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 41 bytes 8069 (7.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Host OS

```
[ ]% sudo arp -s 192.168.56.200 00:00:00:00:00:0a
[ ]% arp -i vboxnet0 -a
? (192.168.56.1) at a:0:27:0:0:0 on vboxnet0 ifscope permanent [ethernet]
? (192.168.56.200) at 0:0:0:0:0:a on vboxnet0 permanent [ethernet]
```

ICMP to 192.168.56.200 can be seen on Guest OS

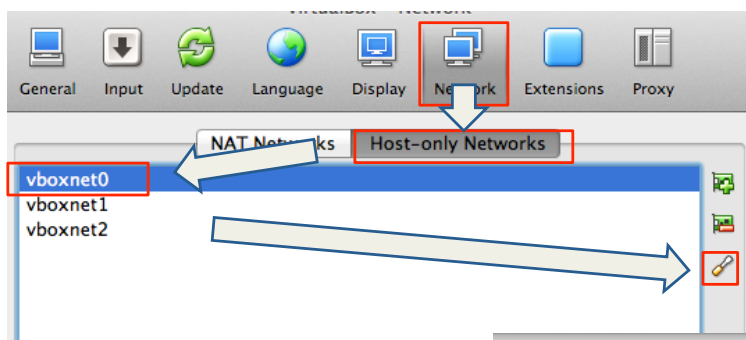
Filter:		Expression... Clear Apply Save			
No.	Time	Source	Destination	Protocol	Info
1	0.000000000	192.168.56.101	192.168.56.200	ICMP	Echo (ping) request id=0x168a, seq=
2	0.999305000	192.168.56.101	192.168.56.200	ICMP	Echo (ping) request id=0x168a, seq=
3	1.999231000	192.168.56.101	192.168.56.200	ICMP	Echo (ping) request id=0x168a, seq=
4	2.745324000	192.168.56.101	192.168.56.1	DHCP	DHCP Request - Transaction ID 0x25c
5	2.999381000	192.168.56.101	192.168.56.200	ICMP	Echo (ping) request id=0x168a, seq=
6	3.999632000	192.168.56.101	192.168.56.200	ICMP	Echo (ping) request id=0x168a, seq=
7	4.999727000	192.168.56.101	192.168.56.200	ICMP	Echo (ping) request id=0x168a, seq=
8	5.999575000	192.168.56.101	192.168.56.200	ICMP	Echo (ping) request id=0x168a, seq=
9	6.999252000	192.168.56.101	192.168.56.200	ICMP	Echo (ping) request id=0x168a, seq=
10	7.756969000	00:00:00_00:00:0a	0a:00:27:00:00:00	ARP	Who has 192.168.56.1? Tell 192.168.
11	7.999292000	192.168.56.101	192.168.56.200	ICMP	Echo (ping) request id=0x168a, seq=



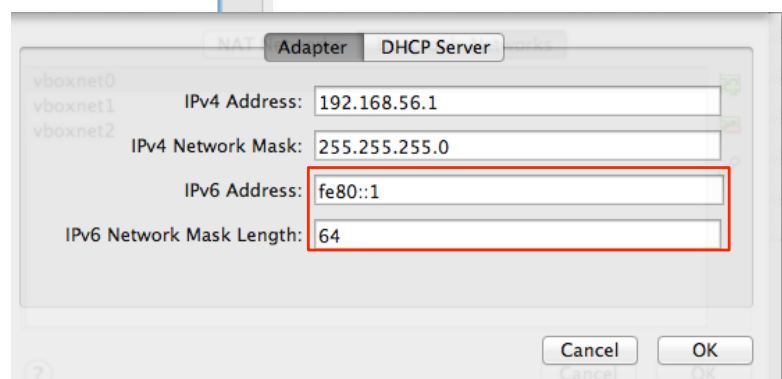
The next topic is Neighbor Solicitation in IPv6.
Let's configure IPv6.

how to set adapter ipv6 address

- From VirtualBox Preferences,



set IPv6
address like that



Neighbor Solicitation

- In IPv6, neighbor solicitation is used for getting MAC address from the other PCs, not arp.
- These data is collected to Neighbor Table.
 - how to check the neighbor table
 - \$ ip -6 neighbor show* (for Linux)
 - \$ ndp -an* (for OS X)
 - \$ netsh interface ipv6 show neighbor* (for cygwin)

Ping for IPv6

- Use it from Host OS and capture that packet.
 - \$ ping -6 <destination>%ifid* (for win)
 - \$ ping6 -I <interface_name> <destination>* (for others)
 - option is *-I*(capital i)
 - e.g. *\$ ping -6 fe80::1%26* (win)
 - \$ ping6 -I enp0s8 fe80::1* (linux)
- please prepare wireshark before using ipv6 ping.
- After ping, please check Neighbor Table.

```
[in1@localhost ~]$ ifconfig
enp0s9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::200:ff:fe00:a prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:00:00:0a txqueuelen 1000 (Ethernet)
    RX packets 73 bytes 13594 (13.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 51 bytes 9389 (9.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

In this case, Guest OS is using fe80::200:ff:fe00:a for IPv6 network.
So, ping6 command is below. Please confirm red line square.

```
[in1@localhost ~]$ ping6 -I enp0s9 fe80::200:ff:fe00:b
PING fe80::200:ff:fe00:b(fe80::200:ff:fe00:b) from fe80::200:ff:fe00:a enp0s9
: 56 data bytes
^C
--- fe80::200:ff:fe00:b ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

Neighbor Solicitation

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::200:ff:fe00:a	ff02::1:ff00:b	ICMPv6	86	Neighbor Solicitation for
2	1.001566000	fe80::200:ff:fe00:a	ff02::1:ff00:b	ICMPv6	86	Neighbor Solicitation for
3	2.003558000	fe80::200:ff:fe00:a	ff02::1:ff00:b	ICMPv6	86	Neighbor Solicitation for
4	3.011420000	fe80::200:ff:fe00:a	ff02::1:ff00:b	ICMPv6	86	Neighbor Solicitation for
5	4.011526000	fe80::200:ff:fe00:a	ff02::1:ff00:b	ICMPv6	86	Neighbor Solicitation for
6	5.013564000	fe80::200:ff:fe00:a	ff02::1:ff00:b	ICMPv6	86	Neighbor Solicitation for


```

+ Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
+ Ethernet II, Src: 00:00:00_00:00:0a (00:00:00:00:00:0a), Dst: IPv6mcast_ff:00:00:0b (33:33:ff:00:00:0b)
+ Internet Protocol Version 6, Src: fe80::200:ff:fe00:a (fe80::200:ff:fe00:a), Dst: ff02::1:ff00:b (ff02::1:ff00:b)
+ Internet Control Message Protocol v6

```

ICMP v6

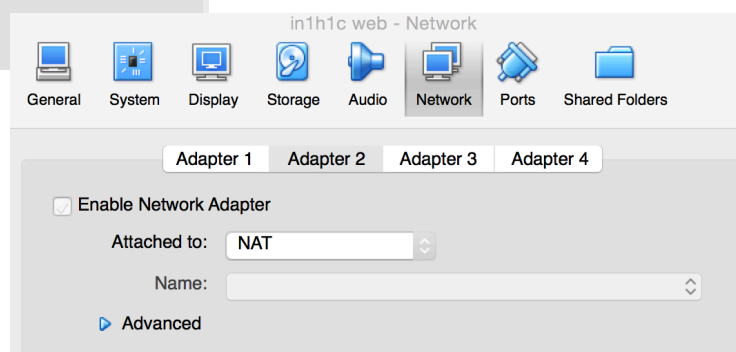
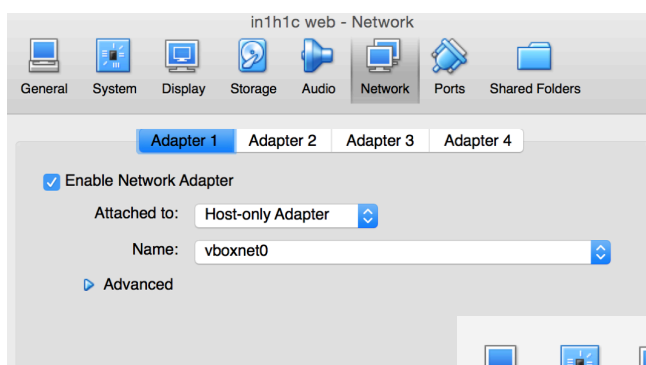
No.	Time	Source	Destination	Protocol	Length	Info
12	278.501291000	fe80::200:ff:fe00:c	fe80::200:ff:fe00:a	ICMPv6	118	Echo (ping) reply id=0x11
13	279.501702000	fe80::200:ff:fe00:a	fe80::200:ff:fe00:c	ICMPv6	118	Echo (ping) request id=0x
14	279.502070000	fe80::200:ff:fe00:c	fe80::200:ff:fe00:a	ICMPv6	118	Echo (ping) reply id=0x11
15	280.501627000	fe80::200:ff:fe00:a	fe80::200:ff:fe00:c	ICMPv6	118	Echo (ping) request id=0x
16	280.501887000	fe80::200:ff:fe00:c	fe80::200:ff:fe00:a	ICMPv6	118	Echo (ping) reply id=0x11
17	281.499379000	fe80::200:ff:fe00:c	fe80::200:ff:fe00:a	ICMPv6	86	Neighbor Solicitation for
18	281.499416000	fe80::200:ff:fe00:a	fe80::200:ff:fe00:c	ICMPv6	78	Neighbor Advertisement fe
19	281.501603000	fe80::200:ff:fe00:a	fe80::200:ff:fe00:c	ICMPv6	118	Echo (ping) request id=0x
20	281.501797000	fe80::200:ff:fe00:c	fe80::200:ff:fe00:a	ICMPv6	118	Echo (ping) reply id=0x11
21	282.505691000	fe80::200:ff:fe00:a	fe80::200:ff:fe00:c	ICMPv6	118	Echo (ping) request id=0x
22	282.506206000	fe80::200:ff:fe00:c	fe80::200:ff:fe00:a	ICMPv6	118	Echo (ping) reply id=0x11



Hands-on IV 2nd - network layer -

Information Network I
8 May. 2015

Changing the NIC settings



Ping

★ Trivia:

- probably the most famous network diagnostic tool
- developed by Mike Muuss in 1983
- believed to be an abbreviation for Packet Internet Groper, but the author has stated that the name comes from the sound that a sonar makes.

Let's ping

- Open Wireshark and start the packet capture on the **enp0s8** interface
- Run the following command from the VM
\$ ping 8.8.8.8

No.	Time	Source	Destination	Protocol	Length	Info
291	789.316551000	8.8.8.8	10.0.3.15	ICMP	98	Echo (ping) reply id=0x0d88, seq=65/1
292	790.253474000	10.0.3.15	8.8.8.8	ICMP	98	Echo (ping) request id=0x0d88, seq=66/1
293	790.318618000	8.8.8.8	10.0.3.15	ICMP	98	Echo (ping) reply id=0x0d88, seq=66/1
294	791.254564000	10.0.3.15	8.8.8.8	ICMP	98	Echo (ping) request id=0x0d88, seq=67/1
295	791.318928000	8.8.8.8	10.0.3.15	ICMP	98	Echo (ping) reply id=0x0d88, seq=67/1
296	792.257077000	10.0.3.15	8.8.8.8	ICMP	98	Echo (ping) request id=0x0d88, seq=68/1
297	792.320929000	8.8.8.8	10.0.3.15	ICMP	98	Echo (ping) reply id=0x0d88, seq=68/1
298	793.258854000	10.0.3.15	8.8.8.8	ICMP	98	Echo (ping) request id=0x0d88, seq=69/1
299	793.322932000	8.8.8.8	10.0.3.15	ICMP	98	Echo (ping) reply id=0x0d88, seq=69/1
300	794.259973000	10.0.3.15	8.8.8.8	ICMP	98	Echo (ping) request id=0x0d88, seq=70/1
301	794.322864000	8.8.8.8	10.0.3.15	ICMP	98	Echo (ping) reply id=0x0d88, seq=70/1

Frame 167: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0

Ethernet II, Src: CadmusCo_9a:70:3e (08:00:27:9a:70:3e), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)

Internet Protocol Version 4, Src: 10.0.3.15 (10.0.3.15), Dst: 8.8.8.8 (8.8.8.8)

Internet Control Message Protocol

Traceroute

★ Trivia:

- network diagnostic tool used to display the route to a certain address
- the original traceroute implementation was coded by Van Jacobson in 1987

Let's traceroute

- Open Wireshark and start the packet capture
- Run the following command from the VM
`$ traceroute 8.8.8.8`

No.	Time	Source	Destination	Protocol	Length	Info
16	0.000867000	10.0.3.15	8.8.8.8	UDP	74	Source port: 48559 Destination port: 33446
17	0.000928000	10.0.3.15	8.8.8.8	UDP	74	Source port: 44595 Destination port: 33446
18	0.001164000	10.0.3.15	8.8.8.8	UDP	74	Source port: 52269 Destination port: 33446
19	0.001251000	10.0.3.15	8.8.8.8	UDP	74	Source port: 52824 Destination port: 33446
20	0.002694000	163.221.124.1	10.0.3.15	ICMP	74	Time-to-live exceeded (Time to live exceeded)
21	0.003104000	163.221.124.1	10.0.3.15	ICMP	74	Time-to-live exceeded (Time to live exceeded)
22	0.003109000	163.221.124.1	10.0.3.15	ICMP	74	Time-to-live exceeded (Time to live exceeded)
23	0.003562000	10.0.3.15	163.221.8.11	DNS	81	Standard query 0x6f32 PTR 2.3.0.10.in-addr.arpa
24	0.003607000	163.221.6.97	10.0.3.15	ICMP	74	Time-to-live exceeded (Time to live exceeded)
25	0.003611000	163.221.6.97	10.0.3.15	ICMP	74	Time-to-live exceeded (Time to live exceeded)
26	0.003613000	163.221.6.97	10.0.3.15	ICMP	74	Time-to-live exceeded (Time to live exceeded)

Frame 16: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0

Ethernet II, Src: CadmusCo_9a:70:3e (08:00:27:9a:70:3e), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)

Internet Protocol Version 4, Src: 10.0.3.15 (10.0.3.15), Dst: 8.8.8.8 (8.8.8.8)

User Datagram Protocol, Src Port: 48559 (48559), Dst Port: 33446 (33446)

Data (32 bytes)

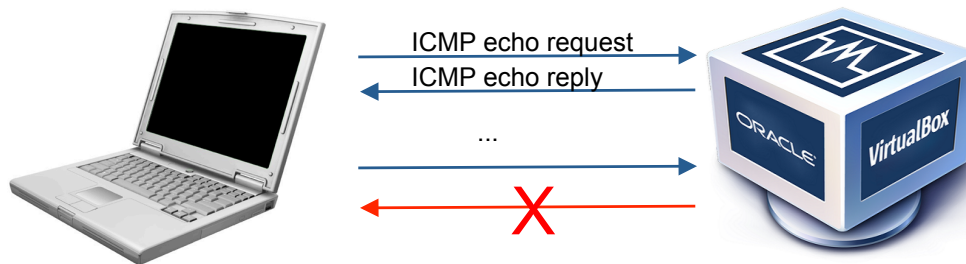


Traffic Control (TC) Loss rate

- Run the following command:
`$sudo tc qdisc add dev enp0s3 root netem loss 10`
* this command should emulate a 10% loss rate on the **enp0s3** interface
- Confirm the config by running:
`$sudo tc qdisc list`

Let's check the loss rate

- Run ping from the host machine to the VM:
`$ping 192.168.56.102`



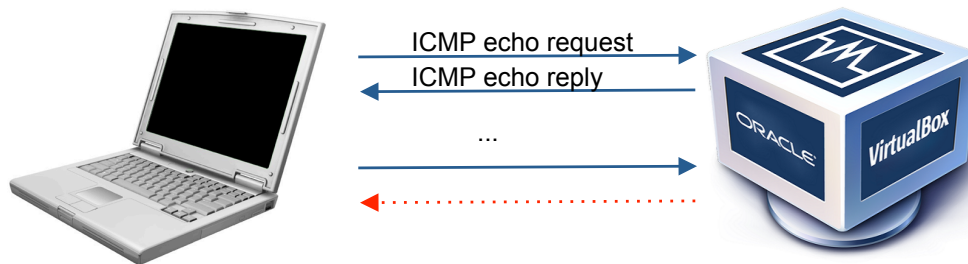
* for details use Wireshark to capture the ICMP traffic

Traffic Control (TC) corrupt

- Run the following command to remove the previous setting:
`$sudo tc qdisc del dev enp0s3 root netem loss 10`
- Run the following command:
`$sudo tc qdisc add dev enp0s3 root netem corrupt 15`
- * this command should emulate a 15% packet corruption rate on the **enp0s3** interface
- Confirm the config by running:
`$sudo tc qdisc list`

Let's check the corruption rate

- Run ping from the host machine to the VM:
`$ping 192.168.56.102`



* for details use Wireshark to capture the ICMP traffic

Traffic Control (TC) Delay

- Run the following command to remove the previous setting:

```
$sudo tc qdisc del dev enp0s3 root netem corrupt 10
```

- Run the following command:

```
$sudo tc qdisc add dev enp0s3 root netem delay 50ms
```

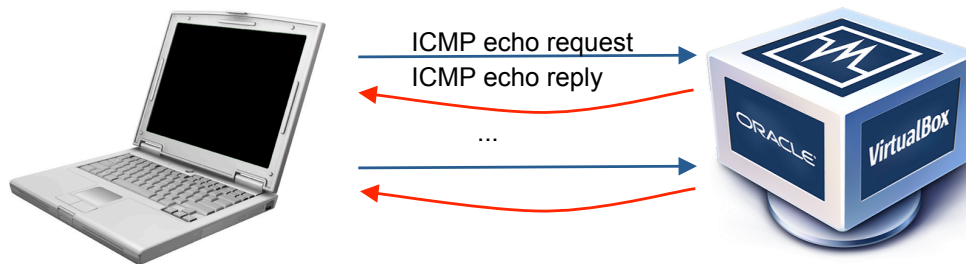
* this command should emulate a 50ms packet delay on the **enp0s3** interface

- Confirm the config by running:

```
$sudo tc qdisc list
```

Let's check the RT Delay

- Run ping from the host machine to the VM:
`$ping 192.168.56.102`



* for details use Wireshark to capture the ICMP traffic

