

# Computer Networks

ELEC2920

## Introduction to IP Routing

Lab # 2

Aubry Springuel

Oscar Medina Duarte

Abstract: The objective of this laboratory work was to experiment with the setup of some small sub-networks interconnected one each other by static routes exploiting Linux's utilities.

### Working Environment

For this laboratory work, Linux was used as a working environment because is very flexible and it allows somewhat low level configuration commands and procedures that would allow us to see how thing work. The most used commands during the lab were the ones related to network interface configuration and routing which are briefly explained here:

ifconfig – This command is used to configure network interfaces, which in our case were Ethernet network interfaces (ie. eth[0-9]+).

route – This command is used to set and display the static routing table of the a computer.

arp - This command if used to manipulate the ARP cache of the system, but in this lab we had only used it to display it.

netstat – This command is used to display networking statistics like network connections, open ports, routing tables and transmission statistics among some others.

ping – This command is a ICMP ECHO\_REQUEST sender implementation, which is used to know whether or not a computer reply to this messages in which case it can tell if a computer is connected to the network (But it doesn't necessarily diagnoses the opposite case).

iptables – This is a tool used to manage the IP packet filtering rules and NAT of the Linux kernel.

traceroute – This program is used to find out the routers required to traverse the network(hops) to reach a computer.

We were referred to the man pages every time there was some doubt about the usage or syntax of a command.

### Basic networks and static routing

A certain network structure (Appendix A - Figure1) was requested by assigning each computer on the lab (Machines A-F) a different network address and in the case of the machines B and E, two different network address one for each of its network interfaces (eth0, eth1). This was done using the ifconfig command like this:

```
$ ifconfig eth0 192.168.11.1 netmask 255.255.255.248 broadcast 192.168.11.7
```

\$

The result of this could be confirmed by displaying the configuration of the interface:

\$ ifconfig eth0

```
eth0  Link encap:Ethernet HWaddr 00:20:18:54:7E:2B
      inet addr: 192.168.11.1 Bcast:192.168.11.7 Mask:255.255.255.248
      inet6 addr: fe80::220:18ff:fe54:7e2b/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:15 errors:0 dropped:0 overruns:0 frame:0
      TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:1764 (1.7 KiB) TX bytes:2772 (2.7 KiB)
      Interrupt:18 Base address:0xa800
```

The meaning of the information displayed is :

eth0 .- Is the name of the Ethernet card in the system to be configured.

Link encap:Ethernet .- This is the lower encapsulation protocol (Ethernet).

HWaddr 00:20:18:54:7E:2B .- Hardware address of the Ethernet card.

inet addr: 192.168.11.1 .- Internet IP address assigned to the card.

Bcast:192.168.11.7 .- Broadcast address of the network card.

Mask:255.255.255.248 .- This is the network mask.

inet6 addr: fe80::220:18ff:fe54:7e2b/64 .-

MTU:1500 .- Maximum transfer unit.

RX packets:15 .- Number of received packets.

TX packets:12 .- Number of transmitted packets.

txqueuelen:1000 .- Length of the transmission queue.

RX bytes:1764 (1.7 KiB) .- Number bytes received.

TX bytes:2772 (2.7 KiB) .- Number bytes transmitted.

Interrupt:18 .- Hardware Interruption of the card.

Base address:0xa800 .- Base hardware address of the card.

After this, we were asked to check if all the network was interconnected using the ping command to try to reach the other computers in the network like this (If we are Machine C):

First to our neighbors :

Sending ping to machine B :

\$ ping 192.168.2.1

PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.

64 bytes from 192.168.2.1: icmp\_seq=1 ttl=64 time=0.766 ms

64 bytes from 192.168.2.1: icmp\_seq=2 ttl=64 time=0.420 ms

```
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=0.381 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=64 time=0.416 ms
64 bytes from 192.168.2.1: icmp_seq=5 ttl=64 time=0.422 ms
64 bytes from 192.168.2.1: icmp_seq=6 ttl=64 time=0.373 ms
```

--- 192.168.2.1 ping statistics ---

```
6 packets transmitted, 6 received, 0% packet loss, time 5000ms
rtt min/avg/max/mdev = 0.373/0.463/0.766/0.136 ms
```

Sending ping to machine A:

```
$ ping 192.168.1.1
```

```
PING 192.168.1.1 (192.168.2.1) 56(84) bytes of data.
```

```
From 192.168.1.1 icmp_seq=1 Destination Host Unreachable
```

```
From 192.168.1.1 icmp_seq=2 Destination Host Unreachable
```

```
From 192.168.1.1 icmp_seq=3 Destination Host Unreachable
```

--- 192.168.1.1 ping statistics ---

```
5 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3998ms
No response !!
```

No response was received and the same result was found when pinging computers D to F, that is because the routing was not yet configured, so to activate static routing at computers B and E we used the following route syntax:

```
$ route add -net 192.168.12.0/29 gw 192.168.11.2
```

This means that we will add the machine with IP 192.168.11.2 as gateway (router) to the routing table, which will give to the current machine the path to follow in order to reach computers in the network 192.168.12.0 with netmask 29 (255.255.255.248). With the specified netmask we are only giving a 3 bit space for addresses which would allow us only to have 5 normal IP addresses, from whom one of them would be the router IP, so we would have 4 free IP's. The other 2 addresses would be the first one (ie. 192.168.12.0) which in fact is the network address of the subnetwork and the last one (192.168.12.7) which would be the broadcast address.

Once we have done this we were asked to connect to a certain HTTP (port 80) server in the network using telnet.

```
$ telnet 80
asdasdas
```

### Usage of a network analyzer

We had used *ethereal* and *tcpdump* network analyzers to check out the behavior of the protocols during its usage, specifically the ping command while sending only one ICMP ECHO\_REQUEST packet with the following command : ping -c1 <IP address>, from which we could see the following in the tcpdump window:

```
$ tcpdump
```

```
ARP who has 123,123,123,1  
ICMP REQ  
ICMP Reply
```

From this output we can see that the first transmitted packet is an ARP "Who has" request from the originating computer to the rest of the network and following

### Usage of the netstat command

We have used the netstat command to display information and statistics of the network (see Appendix B – netstat command) from which the most interesting information is related to the current active tcp connections, but if we request more detailed information about the hardware networking interfaces by using the *-i* modifier, we would get:

```
$ netstat -i
```

Kernel Interface table

Iface	MTU	Met	RX-OK	RX-ERR	RX-DRP	RX-OVR	TX-OK	TX-ERR	TX-DRP	TX-OVR	Flg
eth0	1500	0	420	0	0	0	598	0	0	0	BMRU
lo	16436	0	104450	0	0	0	104450	0	0	0	LRU

Which is a listing of properties and statistics per interface, like Maximum Transfer Unit , transferred bytes, errors, packets etc...

We had also used the netstat command on the following way (see Appendix B - netstat command for complete listing) :

```
$ netstat -i s
```

Ip:

```
126284 total packets received
```

```
...
```

Icmp:

```
333 ICMP messages received
```

```
...
```

```
78 ICMP messages sent
```

```
..
```

```
destination unreachable: 8
```

```
echo replies: 70
```

Tcp:

```
2507 active connections openings
```

```
2469 passive connection openings
```

```
...
```

```
38 resets sent
```

Udp:

0 packets received

...

8 packets sent

TcpExt:

2445 TCP sockets finished time wait in fast timer

...

Which as we can see it displays more detailed statistics on the kind of packets and datagrams transmitted over the life of the link.

## Check of routing tables

To check the proper configuration of the routing tables we use first the ping program to send ICMP ECHO\_REQUEST packets to each of the connected computers as in Part1, and secondly, we use the traceroute program to find the routers on the way to reach a certain computer in the network, for instance, if we trace the route from machine C to machine F we would see something like this:

```
$ traceroute 192.168.12.2
```

```
traceroute to 192.168.12.1 (192.168.12.1), 64 hops max, 40 byte packets
```

```
1 192.168.2.3 (192.168.2.3) 49 ms 0 ms 0 ms
```

```
2 192.168.12.1 (192.168.12.1) 1 ms 0 ms 0 ms
```

Which means that we have to go through 2 routers with IP's 192.168.2.3 and 192.168.12.1 corresponding to computers G and E respectively, to reach machine with IP 192.168.12.2.

The traceroute program sends UDP segments to the destination with increasing time-to-live starting from one. It will receive one ICMP warning message of type 11 code 0 from each router that after decrementing the TTL by one obtains a TTL equal to 0. At the same time it will measure the round trip time starting from the moment it send the first datagram until it receives the corresponding ICMP warning message. Traceroute stops sending messages when it receives an ICMP message type 3 code 3 (port unreachable) from the destination host.

## Introduction to packet filtering

## Conclusion

Appendix A

Figure 1

## Appendix B – Listings

### netstat command

\$ netstat

Active Internet connections (w/o servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	telehost:ipp	telehost:32782	ESTABLISHED
tcp	0	0	telehost:ipp	telehost:32783	ESTABLISHED
tcp	0	0	telehost:34803	telehost:ipp	TIME_WAIT

...

tcp	0	0	telehost:34819	telehost:ipp	TIME_WAIT
tcp	0	0	telehost:34818	telehost:ipp	TIME_WAIT

Active UNIX domain sockets (w/o servers)

Proto	RefCnt	Flags	Type	State	I-Node Path
unix	2	[]	DGRAM		12645 @/var/run/hal/hotplug_socket
unix	2	[]	DGRAM		11827 @udev

...

unix	3	[]	STREAM	CONNECTED	12643
unix	3	[]	STREAM	CONNECTED	12597
unix	3	[]	STREAM	CONNECTED	12596
unix	2	[]	DGRAM		11824

\$ netstat -i s

Ip:

- 126284 total packets received
- 0 forwarded
- 0 incoming packets discarded
- 126259 incoming packets delivered
- 126415 requests sent out

Icmp:

- 333 ICMP messages received
- 0 input ICMP message failed.

ICMP input histogram:

- destination unreachable: 8
- redirects: 4
- echo requests: 70
- echo replies: 251

78 ICMP messages sent  
0 ICMP messages failed  
ICMP output histogram:  
    destination unreachable: 8  
    echo replies: 70

Tcp:

2507 active connections openings  
2469 passive connection openings  
0 failed connection attempts  
0 connection resets received  
4 connections established  
125784 segments received  
125784 segments send out  
0 segments retransmitted  
0 bad segments received.  
38 resets sent

Udp:

0 packets received  
8 packets to unknown port received.  
0 packet receive errors  
8 packets sent

TcpExt:

2445 TCP sockets finished time wait in fast timer  
6091 delayed acks sent  
18886 packets directly queued to recvmsg prequeue.  
1124761 of bytes directly received from backlog  
629791 of bytes directly received from prequeue  
16435 packet headers predicted  
19929 packets header predicted and directly queued to user  
6368 acknowledgments not containing data received  
50906 predicted acknowledgments  
0 TCP data loss events