Networks, Communications & Cyber Defence

Practical Session
IP version 4
Addressing

Peter Norris
University of Warwick

November 2021

Contents

Starting assumptions....2

IP addressing activities....2

Intended outcomes....2

Launching the Netkit lab....3

Look at the configuration....3

Change the configuration....4

Collisions and challenges....4

Convert to iproute2....5

IP4 subnet address ranges....5

Private addresses....5

Experiment....5

References....5

Starting assumptions

1: You have used Netkit **lstart** to launch a coordinated group of several virtual machines. Specifically, you are aware of how this group is coordinated via the **lab.conf** file.

2: You know how to shut down virtual machines using **lhalt** and **lcrash**.

3: You have used **ip addr** and **ping** within the virtual machines.

4: You have saved captured network traffic from a virtual machine using **tcpdump**.

5: You have viewed a packet capture file in the real Linux host using **wireshark**.

6: You are prepared to make accurate notes of what you do

7: You will complete unfinished activities before the next timetabled lab session.

8: You will resolve what you do not understand by conducting your own careful (and ethically sound) experimentation and / or further reading.

IP addressing activities

Intended outcomes

9: Can configure an individual Linux host via **ip addr**, **ifconfig**, **/etc/network/interfaces** and **ifup**.

10: Can configure several hosts to be on the same subnet.

11: Can explain how netmasks / VLSM and CIDR notation define the portion of the IP address that identifies the network (and by implication, how they also define the portion of the address that identifies unique hosts on a network).

Launching the Netkit lab

12: Open a Linux terminal, make the **~/nklabs/** directory your current directory:

```
13: cd nklabs/
14: pwd
```

15: As ever, use the **man** command to find out what a particular command does

16: Copy the zipped archive **lab05.tar.gz** and save it in the **~/nklabs/** directory. This contains the configuration information for around 14 virtual machines in the **lab05** directory.

17: This represents the configuration of nine hosts and five gateways / routers. Note that these are connected via hubs rather than switches. Connectivity is as the ascii art diagram in the file **lab.conf** (draw this in your own notes to annotate as the lab progresses):

18: Extract the the contents of **lab05.tar.gz** into the **~/nklabs** directory. Assuming your current directory is **~/nklabs** , this can be achieved by:

```
19: tar -xvzf lab05.tar.gz
```

20: Place yourself in the newly created **lab05** directory and list its contents:

```
21: cd lab05
22: ls
```

23: You will see several directories (some empty, some with content), one for each virtual machine.

24: With your current directory as the **~/nklabs/lab05** directory (which contains the **lab.conf** file), launch the virtual machines using the instruction:

```
25: lstart
```

26: You should see virtual machines start up in quick succession in groups of five. Arrange the windows on the screen so they are consistent with the network diagram you have created in your notes.

Look at the configuration

27: The subnets have been labelled W, X, Y, Z and D. For each subnet in turn, look at the network configuration of each host / network card connected. In particular note the IP address, the netmask and the broadcast address of each. Make sure you know how each machine got its IP address.

```
28:  ifconfig eth0               # old school net-tools
29:  ip addr show dev eth0       # modern iproute2
```

30: Start capturing the traffic from the gateway machine and saving it to your lab directory. For example, on subnet W, the gateway machine would be gwW so the instruction would be:

```
31:  tcpdump -s0 -i eth0 -w /hostlab/gwW-eth0-01.pcap
```

32: Ping between two machines in the same subnet. Stop capturing the the traffic (ctrl-c in the terminal where you started the tcpdump) and look at it using wireshark.

33: Look up the IP header format in RFC 791. Compare what you find in the standard with what you see in the bottom panel of the wireshark capture for one particular "ping" of your choice.

Change the configuration

34: Modify the running virtual machines on subnet W so that all the machines are on a different subnet. Specifically, place them all on the 146.227.150.64/26 subnet. Why can they not go on the 146.227.150.64/25 subnet?

35: Crash and restart the virtual machines on subnet W to restore them to the original configuration.

```
36:  lcrash m1 m2 gwW
37:  lstart m1 m2 gwW
```

38: Add an extra machine to the lab. Give it the name **Foo**, give it one Ethernet card and put it onto subnet X. Record precisely what changes you made so that the extra machine works with the other machines in the lab. You should find you need to edit **lab.conf**, create a directory and figure out how to ensure it starts up with the correct IP address.

Collisions and challenges

39: Give two machines on the same subnet the same IP address. What happens precisely when you try and interact with them?

40: Make one or two configuration changes so things are slightly broken. Challenge a colleague to identify and correct the problem.

Convert to iproute2

41: Convert all the configuration from old-school net-tools commands (ifconfig, route, etc) Replace the configuration using the **iproute2** command

```
42: ip addr ...
```

IP4 subnet address ranges

43: Explore IP4 address ranges on CyberChef. Ensure that you are clear precisely which range of addresses are on the subnet of any particular aaa.bbb.ccc.ddd/n CIDR address. Get to the stage where you can do this without needing to refere to CyberChef.

Private addresses

44: Look at rfc 1918 and the subsequent rfcs that make it obsolete. Record the number of each of these rfcs. Create a text file of the specific IPv4 addresses that are not routable on the public internet.

Experiment

45: Make DNS work with any changes that you make to the lab.

46: Experiment…

References

47: CyberChef; https://gchq.github.io/CyberChef/

48: RFC 791; Internet Protocol; https://datatracker.ietf.org/doc/html/rfc791

49: RFC 1918; Address Allocation for Private Internets; https://datatracker.ietf.org/doc/html/rfc1918 (see also 3330, 5735)