Networks, Communications & Cyber Defence

Practical Session
IP version 4
Static Routing

Peter Norris
University of Warwick

September 2018

Contents

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 **ifconfig** 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.

IPv4 static routing activities

Intended outcomes

9: Can configure and view IPv4 static routes in Linux using **route** or **ip** instructions.

10: Can analyse a captured pcap file using wireshark and diagnose simple routing problems across small networks of networks.

11: Can adapt marginally incorrect instructions and / or Netkit configurations. (Note carefully - from now on, there will be some deliberate errors introduced so you are not always able to use instructions verbatim from this lab sheet; **there will be some deliberate errors**.)

Launching the Netkit lab

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

```
13:  cd nccd/
14:  pwd
```

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

16: Copy the zipped archive **lab06.tar.gz** and save it in the **~/nccd/** directory. This contains the configuration information for around 15 virtual machines split across two directories: **lab06a** directory and **lab06b** directory.

17: The overall arrangement of the virtual machines into subnets is almost identical to the lab05 arrangement. The lab **lab06a** contains the **W, X** and part of the **D** subnet. The lab **lab06b** contains the **Y** and **Z** subnets and the remainder of the **D** subnet.

18: Extract the the contents of **lab06.tar.gz** into the **~/nccd** directory. Assuming your current directory is **~/nccd** , this can be achieved by:

```
19:  tar -xvzf lab06.tar.gz
```

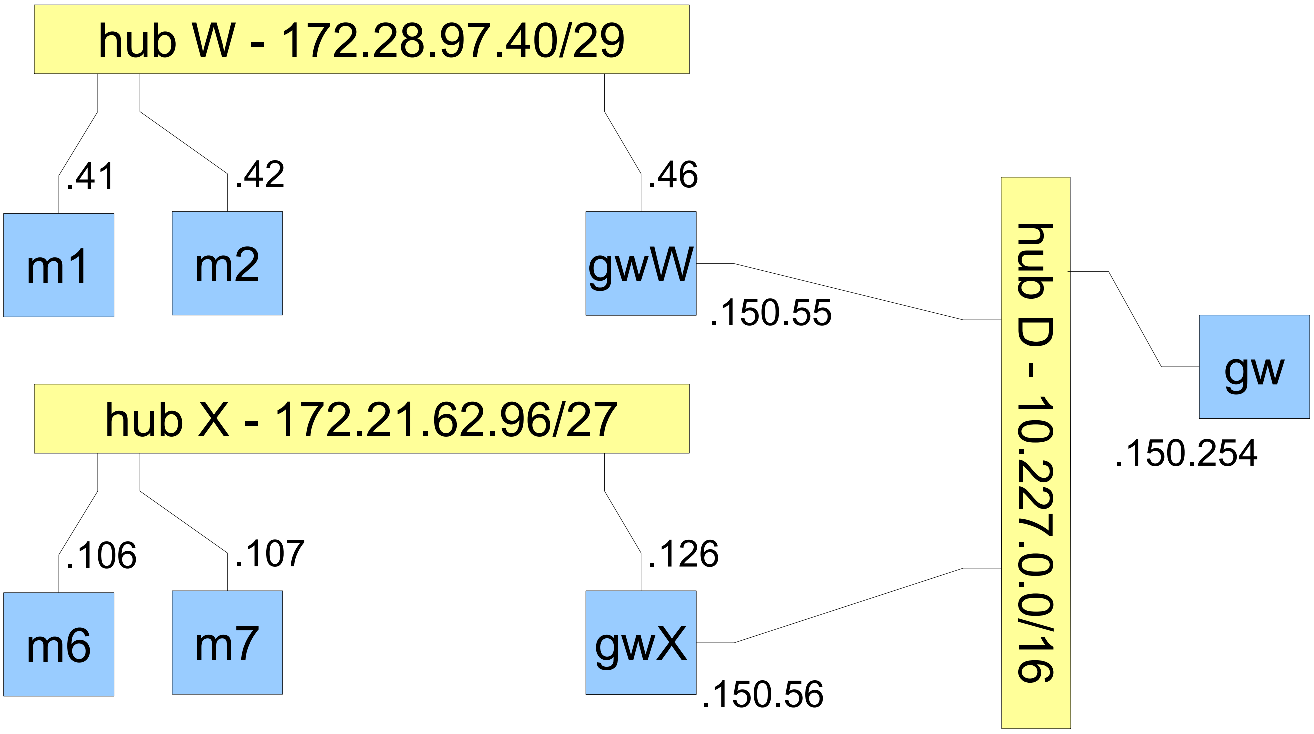
20: Place yourself in the newly created **lab06a** directory and list its contents:

```
21:  cd lab06a
22:  ls
```

23: You will see seven directories (some empty, some with content), one for each virtual machine. Identify which virtual machines have been given a **/etc/network/interfaces** file and confirm whether this is consistent with the information in the diagram in the lab.conf file.

```
24:  man interfaces
```

25: Look in the **lab.conf** file. Confirm that it corresponds to the diagram of the network you have made in your notes and the one below.



26: Look at the seven ***.startup** files, one for each machine. Identify which of these *directly* set the IP address via **ifconfig** and which use **ifup** to utilise the settings in **/etc/network/interfaces**.

27: With your current directory as the **~/nccd/lab06a** directory (which contains the **lab.conf** file), launch the seven virtual machines using the instruction:

```
28:  lstart
```

29: Look carefully at the virtual machines as they start. In particular record any errors you see.

Look at the configuration

30: Once all machines are up, use **tcpdump** on machines **m2, m7** and **gw** to store captured traffic in the files **m2-dump1.pcap**, **m7-dump1.pcap** and **gw-dump1.pcap** in the **lab06** directory you created in step 15 above. The instructions will be similar to the one below.

```
31:  tcpdump -s0 -i eht1 -w /hosthone/nccd/lab06/m2-dump1.pcap
```

32: Write down the exact instruction you used on each machine.

33: Now generate some traffic. From machine **m1** , ping each of the following IP addresses precisely twice:

```
34:  172.28.97.46
35:  10.227.150.55
```

```
36:  10.227.150.56
37:  172.21.62.126
38:  172.21.62.106
39:  172.21.62.99
```

40: For each of these IP addresses, write down:
a) which machine it is associated with,
b) the corresponding MAC address associated with each.

41: Stop the packet capture in **m2**, **m7** and **gw** (use CTRL-C in each virtual machine in turn.

42: Look at each of the three packet capture files in turn. Explain the MAC address and IP address of each packet. In particular, identify those packets where the destination MAC address is not consistent with the destination IP address that you noted above.

Tracing the route

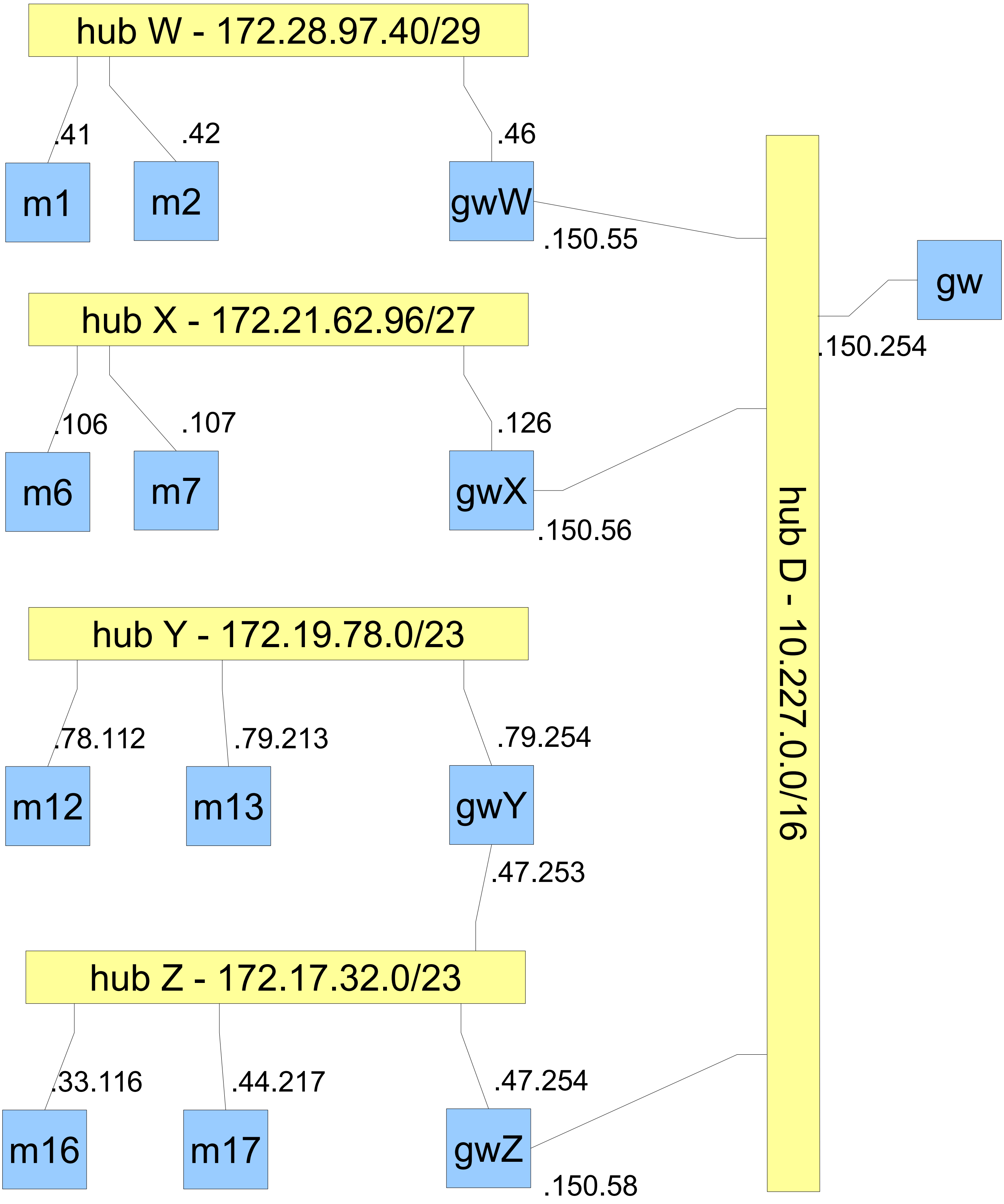
43: It is often useful to find the journey that a packet takes as it traverses networks of networks. Start packet capture again on **m2**, **m7** and **gw** but this time capture it into three different **...dump2.pcap** files.

44: From machine m1, execute the following:

```
45:  ping c1 172.21.62.106 -R
46:  traceroute -n 172.21.62.106
```

47: Stop the three packet capture files. Explain:
a) how **ping ... -R** works,
b) how **traceroute** works (and why use the -n option?).

48:


hub W - 172.28.97.40/29
.41
.42
.46
m1
m2
gwW
.150.55
gw
.150.254
hub X - 172.21.62.96/27
.106
.107
.126
m6
m7
gwX
.150.56
hub D - 10.227.0.0/16
hub Y - 172.19.78.0/23
.78.112
.79.213
.79.254
m12
m13
gwY
.47.253
hub Z - 172.17.32.0/23
.33.116
.44.217
.47.254
m16
m17
gwZ
.150.58

Fix the errors

49: Leave the first set of machines running which you started from the **lab06a** directory ie hubs **W** and **X** and some of hub D**.**

50: Debian provides multiple desktops aka workspaces. Select a new workspace (quickly switch between them using *ctrl alt right-arrow* or *ctrl alt left-arrow*) and in it open a terminal with the current directory set to **~/nccd/lab06b**.

51: Start the netkit lab in the **lab06b** directory using **lstart**. This lab will also use hub **D** to create the overall configuration shown below, plus two Domain Name Servers (**dns1** and **dns2**) on hub **D** which are not shown on the diagram on the next page. Note that in this lab, addressing and routing is achieved using the **ip** instruction rather than **ifconfig** and **route**.

52: Firstly find an fix all the IP address and routing problems in this lab so that each machine can ping each other machine. Construct a grid and tick off each machine pair that can ping. For each fault:
a) record the symptoms that you see,
b) explain what you did to locate what was wrong,
c) identify precisely what was wrong,
d) identify what you did to correct the fault,
e) record what you did to confirm the symptoms had gone.

53: Once the faults are fixed, on machines m2, dns1, m17 and m13 start a packet capture into the lab06 directory in your home directory.

54: On machine m1, execute:

```
55: traceroute 172.19.78.112
```

56: Stop the packet captures and explain what is happening.

RFC 1918 (and its friends)

57: Look at rfc1918. Write down the lowest and the highest addresses in each of the ranges. Consider what this means for "globally unique IPv4 addresses". What other RFCs now constrain special use IPv4 addresses. Learn these other ranges of special IPv4 addresses. How many look familiar?

Extension activities

58: Try creating an additional router between **W** and **Y** (eg give **m13** an extra network card and connect it to hub **W**). Now try to create a set of routes so that a packet to an unknown host will circulate until its time to live (TTL) drops to zero.

59: Make one or two changes to the lab configuration, then challenge a colleague to find and fix them.

60: Experiment with wireshark's visualisation of packets. In particular, try filtering so you see only the packets you want to (eg only with a source IP address of 172.28.97.46).

61: Remove the commenting out of the tap from gw in lab.conf. Restart gw `lcrash gw && lstart gw` and ping remote hosts on the real internet. Do this from various machines in the lab. Record traffic as you go. Useful ping targets are 8.8.8.8, 8.8.4.4, 9.9.9.9.

62: Manipulate the MTU from on specific network card. Note how you do this. Experiment with the size of the packets pass through that card. Interesting commands are `ping -c1 -s 4000` and `ifconfig eth0 172.28.97.43 mtu 500` (check routing table entries `route -n` before and after - fix these where needed).

63: Try creating a new host at 10.227.150.99 to be a squid proxy server. Make this part of lab6b. Then use policy based routing in gwZ to force traffic for tcp port 80 to be routed via the squid server. This is complex to think about and goes well beyond routing for this module. Interesting however.