Networks, Communications & Cyber Defence

Practical Session
LANs, Layer 2, Switches, MAC Addresses

Peter Norris
University of Warwick
September 2021

Contents

Acknowledgements

1: Various students have contributed to the evolution of these labs over the years. Particular thanks goes to Josh Hawking and his team for migrating Netkit to a modern kernel and Luke Spademan for migrating from net-tools to iproute2 commands.

Starting assumptions

2: 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** and **lab.dep** files.

3: You know how to shut down virtual machines using **lhalt** or **lcrash**.

4: You have used **ip, ifconfig** and **ping** within the virtual machines.

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

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

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

8: You will complete unfinished activities.

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

Switch activities

Intended outcomes

10: Can configure a switch in Linux via **ip**.

11: Can explain how switches use MAC addresses.

12: Can interrogate a switch in Linux via **bridge**.

13: Can analyse LAN traffic in a pcap file using wireshark.

Launching the Netkit lab

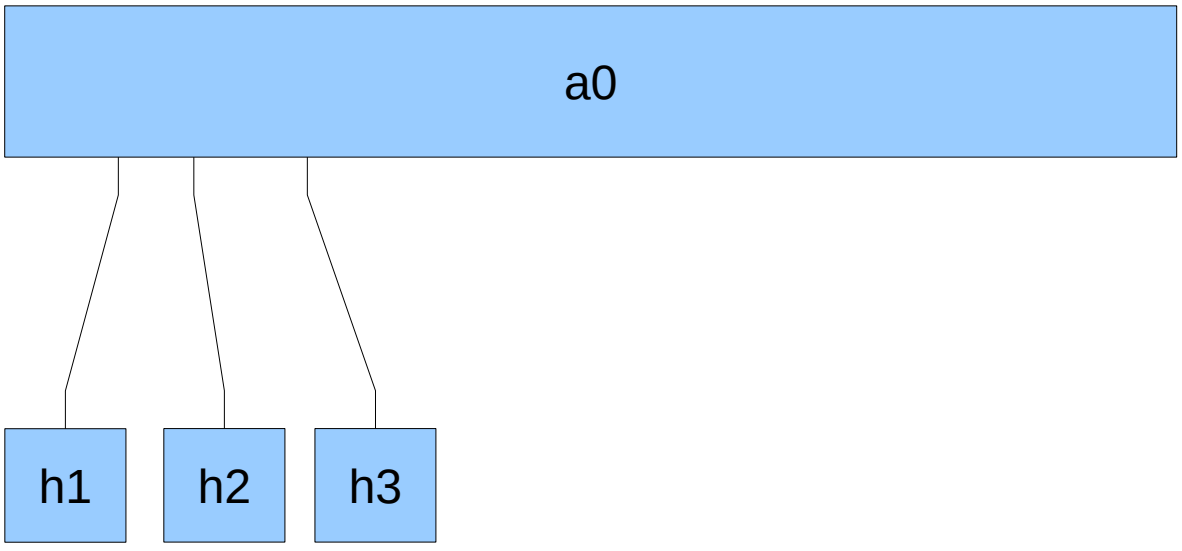
14: Open a Linux terminal, make the ~**/nklabs/** directory your current directory:

```
15: cd ~/nklabs/
16: pwd
```

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

18: Copy the zipped archives **lab3a.tar.gz** and **lab3b.tar.gz** and save them in the **~/nklabs/** directory. These are a pair of labs - **lab3a** can function on its own - **lab3b** expects to have **lab3a** already running.

19: This represents the configuration of a switch (**a0**), three hosts (**h1**, **h2**, **h3**). The switch (**a0**) will **not** have a network (IP) address. Each of the three hosts will get a network (IP) address. We have given the switch (**a0**) eight ports, connected three of them to **h1**, **h2**,and **h3,** left three unused and set up two two interact with **lab3b**. Connectivity is as the diagram below (draw this in your own notes to annotate as the lab progresses):



20: Extract the the contents of **lab3a.tar.gz** into the **~/nklabs** directory. Assuming your current directory is **~/nklabs** , this can be achieved by:

```
21: tar -xvzf lab3a.tar.gz
```

22: Place yourself in the newly created **lab3a** directory and list its contents:

```
23: cd lab3a
24: ls
```

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

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

27: Look at the three ***.startup** files, for the three hosts. Note the MAC address and IP address of each network interface card (NIC) on your diagram.

28: The file **a0.startup** defines the configuration of the bridge (aka switch).

29: With your current directory as the **~/nklabs/lab3a** directory (which contains the **lab.conf** file), launch the four virtual machines using the instruction:

```
30: lstart
```

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

32: Once all machines are up, dump all traffic passing through the switch (**a0**) for later analysis with wireshark using:

```
33: tcpdump -s0 -i sw0 -w /hostlab/a0-sw0-dump01.pcap
```

34: This file will persist because we have used **/hostlab** .

Look at the configuration

35: On all machines except the switch (**a0**), note the MAC address and IP address that each machine has for its eth0 NIC.

```
36: ip addr show dev eth0
```

37: Confirm your understanding of precisely where **h1** and **h3** got their MAC / IP addresses from. Make changes on the host file system using **gedit** so each has a different MAC and gets a different IP address.

```
38: lcrash h1 h3
```

```
39: lstart h1 h3
```

40: In the switch (a0), use ctrl-C to stop the packet capture and look carefully at the traffic. Associate what you find with the various activities so far. In particular, note the frame numbers in the capture file that associate with your activities in the various hosts.

41: Determine the default time for which the switch will cache entries in its table of MAC address entries. Then manipulate the ageing time of entries in the switch so they are cached for only 20 secs.

```
42: brctl ???
```

ARP activities

Intended outcomes

43: Can explain how ARP enables a host to associate a MAC address with an IP address.

44: Can analyse ARP traffic in a pcap file using wireshark.

45: Can configure ARP parameters on a Linux host.

Generating ARP traffic

46: Using the machines from the DHCP activity, dump all traffic passing through the switch (**a0**) for later analysis with wireshark using:

```
47: tcpdump -s0 -i sw0 -w /hostlab/a0-sw0-dump02.pcap
```

48: As with the previous dump, this file will persist after you log out because we have used **/hostlab** (which maps to the directory containing **lab.conf**)

49: Look in the running **h3** virtual machine's ARP cache to see what it already knows. We will do this two ways: the old fashioned way using **arp** and the more modern way using **ip neigh**. We use the **-n** option to prevent IP addresses being resolved to host names.

```
50: arp -n
51: ip neigh show
```

52: On **h3**, quickly ping the IP address of each machine in turn, together with one non-existent address. Since we are pinging an IP address, ARP is going to have to discover the corresponding MAC address so it can send an ethernet frame to the correct NIC.

```
53: ping -c1 192.168.97.41
54: ping -c1 192.168.97.42
55: ping -c1 192.168.97.201
56: ping -c1 192.168.1.1
```

57: Look again at what **h3** knows about MAC address / IP address associations. Write these down.

```
58: arp -n
59: ip neigh show
```

60: Repeat this for **h3** Record what you find before and after pinging the other machines (remember the IP addresses will not all be exactly the same). Identify any discrepancies between what you found on the three machines. Look in their associated *.startup files and ensure you can explain why they differ.

61: Stop the traffic dump in a0 and open the file in wireshark. Ensure you can associate the frames that you see in wireshark with your ping activities.

References

[RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, November 1982. http://www.ietf.org/rfc/rfc0826.txt

Hawking, J et al, Netkit-jh, https://github.com/netkit-jh/netkit-jh-build/releases [accessed 25th Oct 2021]