FIT9137 Workshop Week 9

Topics

- Network Layer:
  - Routing
  - Static Routing
  - Dynamic Routing
- Transport Layer
  - Reliable Communication

Covered Learning Outcomes:

- Analyse and formulate the functions of communication architectures of local area networks, wide area networks and the Internet.
- Examine networks using the underlying fundamental theories, models, and protocols for data transmission.

Instructions

- One of the main purposes of an applied session is to build the learning community, create connections and include the learners. The other goal is to give and receive feedback from your peers and or your tutors.
- Form groups of 2 students (peers) to work through the exercises. If met a problem, try to solve it by asking direct questions to your peer. If the issue was not solved within peers, ask your tutor. If did not get a chance to solve the problem during your applied session with your peer or tutor, jump into one of many consultation hours and ask any of the tutors to help you. Please visit the "Teaching Team and Unit Resources" tile in the FIT9137 Moodle site.

ACTIVITY A: Network Routing

Activity A.1 Static Routing

Download the file `FIT9137_w9.imn` from Moodle and save it in the shared folder on the host machine (your laptop or PC). Open core and from the file menu open the downloaded core configuration from the shared folder in the VM (under `/media/sf_YOUR_SHARED_FOLDER_NAME`). Perform the following tasks.

1. Run the emulation and open a terminal on the node `clio` and ping the two interfaces of the node `zeus`. Do you receive replies? Explain why by inspecting the routing table of both nodes (`ip route` command, `man ip-route` for more information).

   Yes, we receive reply from `10.0.0.1` as it is in the same subnet as the node `clio`. The routing table of `clio` shows this fact:

```
default via 10.0.0.1 dev eth0
10.0.0.0/24 dev eth0  proto kernel  scope link  src 10.0.0.20
```

The second line tells the network layer that the destination host is in the same network. The MAC address of the node zeus is found by sending ARP request for the IP address 10.0.0.1.

For the eth1 interface of the node zeus we receive reply for the following reasons:

i) in routing table of clio:

```
default via 10.0.0.1 dev eth0
10.0.0.0/24 dev eth0  proto kernel  scope link  src 10.0.0.20
```

as the destination address 10.0.1.2 does not match the entry 10.0.0.0/24 the next entry checked is the default route which indicates the next hop as 10.0.0.1 hence the ping requests are sent to the default gateway.

ii) in the routing table of node zeus:

```
10.0.0.0/24 dev eth0  proto kernel  scope link  src 10.0.0.1
10.0.1.0/24 dev eth1  proto kernel  scope link  src 10.0.1.2
```

there is the entry 10.0.1.0/24 which will match with the destination IP address hence the ping request is routed. Once routed the router itself is the destination of the ping request and it sends the reply to destination 10.0.0.20.

iii) The routers response is checked against its routing table and the entry 10.0.0.0/24 is found hence the ping reply is routed toward the node clio

2. From the node clio ping the eth0 interface of the node hera.

a) Do you receive replies? Explain why or why not.

No, for the destination IP address 10.0.1.1 from clio the default route will match, from zeus the entry 10.0.1.0/24 will match however on here the routing table is as follows:

```
10.0.1.0/24 dev eth0  proto kernel  scope link  src 10.0.1.1
10.0.3.0/24 dev eth1  proto kernel  scope link  src 10.0.3.1
```

although the ping request matches the entry 10.0.1.0/24 and is delivered to the router. The router's reply does not find a match for the destination IP address 10.0.0.20 in its routing table and the response is dropped.

b) Open a terminal on the node hera and run the following command:

```
tcpdump -l -i eth0
```

The tcpdump is a command line packet capture tool (man tcpdump for more information). With the above options it starts capturing packets on eth0 interface and print a summary on the screen (standard output stdout)

Try the ping command from clio and observe whether hera receives the ping requests.

We see the ICMP echo requests in the terminal and no echo reply is sent.

c) If you did not receive replies in previous step then find out how to resolve the issue. Apply your fix and try the ping from clio again.

We need to add an entry to the routing table of hera for the subnet 10.0.0.0/24

```
ip route add 10.0.0.0/24 via 10.0.1.2
```

d) From `clio` ping the node `calliope`. Do you receive replies?

No, we receive `Destination Net Unreachable` from `10.0.0.1` as the destination IP address `10.0.3.20` does not match any entries in the `zeus` routing table (it matched default gateway of `clio`)

e) If you did not receive replies in previous step then find out how to resolve the issue. Explain why your fix resolves the issue.

To resolve the issue we need to add an entry to the routing table of `zeus` for the `10.0.3.0/24` network:

```
ip route add 10.0.3.0/24 via 10.0.1.1
```

Adding the routing entry will allow the ping requests to be routed to `hera` which finds the destination IP address can be routed through a directly connected network. The ping request arrives at `calliope` and the node sends the reply matching the default route entry. The replies reaches `hera` and it finds a match for the network `10.0.0.0/24` (the one we added in previous steps) and routes it to `zeus`. `zeus` finds a match for the destination IP address through one of its directly connected interfaces and routes it to that network.

d) Stop the emulation.

Note: Any changes you make while the emulation is running will be lost when you stop the emulation. To test this run the emulation again and try the steps and observe that none of the changes we made via command line on the nodes persisted. To make the changes persistent we need to use the core GUI and edit proper script/configuration files.

e) Place the necessary changes to the StaticRoute scripts of the identified nodes in previous steps so that `clio` can ping `calliope`. Save the configuration.

Note: If you make changes to a core configuration file and then close the window without saving the changes, you will not be warned, and the changes will be lost. If you have made changes that you want to keep, make sure to save the file before closing the core GUI window.

DYNAMIC ROUTING: Additional Tasks (Take home Exercise):

a) Now save the original file `FIT9137_w9.imn as` **`FIT9137_w9_Dynamic_routing_RIP.imn`** `and test the dynamic RIP routing.`

b) Now again save the original file `FIT9137_w9.imn as` **`FIT9137_w9_Dynamic_routing_OSPF.imn`** `and test the dynamic OSPF routing.`

Activity A.2 Routing Table Analysis

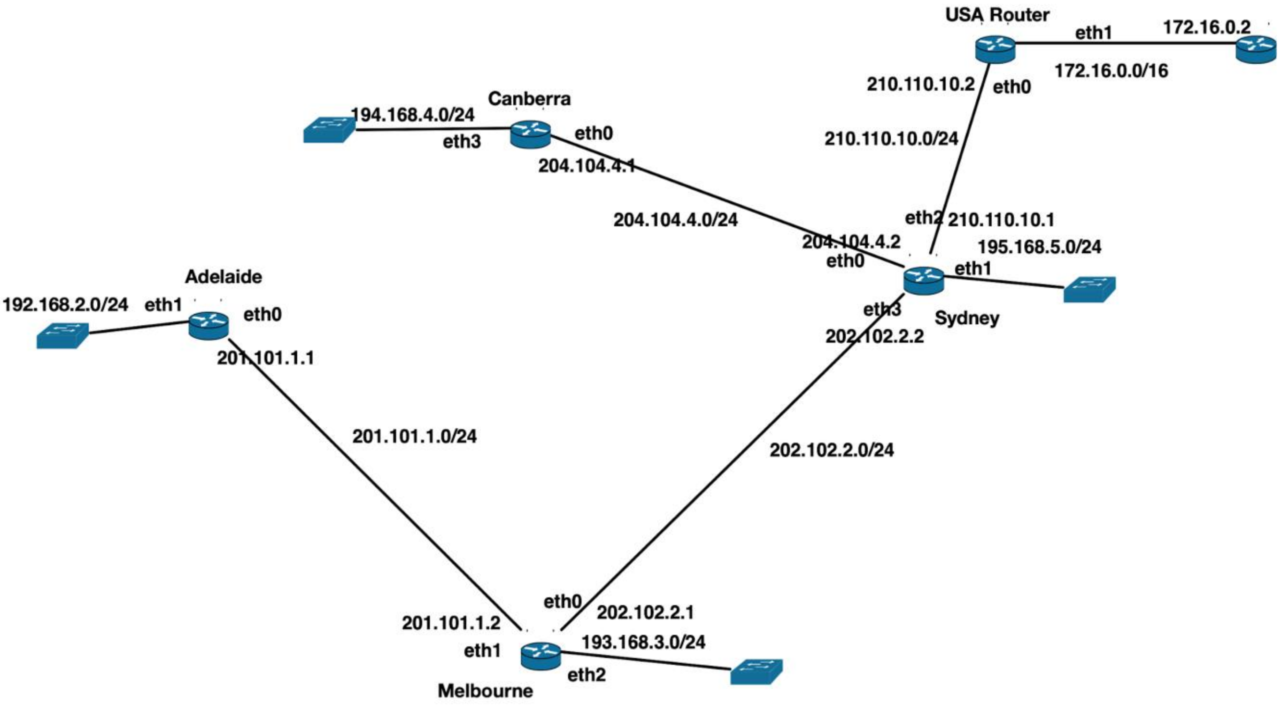
<u>From the captured routing tables (as shown in the Appendix I) of five routers</u>: Adelaide, Melbourne, Sydney, Canberra, and USA. We have constructed the network topology diagram showing the following details:

a) The linking of the routers

b) The interfaces on each router

c) all point-to-point links between routers and links to LANs and

d) all network addresses



Network topology diagram

Answer the following questions:

a) Which routers use RIP?

Routers with `R` entries, all except Adelaide

b) Which routers use static routes?

Routers with `S` entries, all except Canberra

c) Which routers use only static routes?

Routers with only `S` and `C`, Adelaide

d) Which router is the gateway to the Internet?

USA

e) Which routers learnt their gateway via RIP?

the entry `0.0.0.0/0` learned via RIP: Melbourne, Sydney, Canberra

f) Which routers use a static route to point to the gateway?

the entry `0.0.0.0/0` with code `S`: Adelaide and USA

g) Which networks are learnt via RIP on each Router?

entries with `R`

h) What do [120/1] and [1/0] mean? [Hint: https://en.wikipedia.org/wiki/Administrative_distance]

first number: Administrative distance check https://en.wikipedia.org/wiki/Administrative_distance for more information (lower is better).

second number: calculated cost/distance, in case of RIP this is hop count, the Link State or Hybrid protocols take into account other factors such as link speed, delay, congestion into account in calculating the weight/cost of a link (lower is better).

i) What do the times e.g., `00:01:09` mean in Melbourne Router? [hint: https://en.wikipedia.org/wiki/Routing_Information_Protocol#Timers]

RIP timeout indicating for how long the entry is valid. The entry will be removed if no update is received for that entry within the timeout period. https://en.wikipedia.org/wiki/Routing_Information_Protocol#Timers

j) Which routers have LANs?

all except USA (assuming `172.16.0.0/16` is the connection to the Internet)

Appendix I

Codes: `C` connected, `S` static, `R` RIP

The network address `0.0.0.0/0` stands for the default gateway.

Adelaide Router

```
S>* 0.0.0.0/0 [1/0] via 201.101.1.2, eth0
C>* 192.168.2.0/24 is directly connected, eth1
S>* 193.168.3.0/24 [1/0] via 201.101.1.2, eth0
S>* 194.168.4.0/24 [1/0] via 201.101.1.2, eth0
S>* 195.168.5.0/24 [1/0] via 201.101.1.2, eth0
C>* 201.101.1.0/24 is directly connected, eth0
```

Melbourne Router

```
R>* 0.0.0.0/0 [120/3] via 202.102.2.2, eth0, 00:01:09
R>* 172.16.0.0/16 [120/3] via 202.102.2.2, eth0, 00:01:09
S>* 192.168.2.0/24 [1/0] via 201.101.1.1, eth1
C>* 193.168.3.0/24 is directly connected, eth2
R>* 194.168.4.0/24 [120/3] via 202.102.2.2, eth0, 00:01:11
R>* 195.168.5.0/24 [120/2] via 202.102.2.2, eth0, 00:01:11
C>* 201.101.1.0/24 is directly connected, eth1
C>* 202.102.2.0/24 is directly connected, eth0
R>* 204.104.4.0/24 [120/2] via 202.102.2.2, eth0, 00:01:11
R>* 210.110.10.0/24 [120/2] via 202.102.2.2, eth0, 00:01:11
```

Sydney Router

```
R>* 0.0.0.0/0 [120/2] via 210.110.10.2, eth2, 00:02:02
R>* 172.16.0.0/16 [120/2] via 210.110.10.2, eth2, 00:02:02
S>* 192.168.2.0/24 [1/0] via 202.102.2.1, eth3
R>* 193.168.3.0/24 [120/2] via 202.102.2.1, eth3, 00:02:02
R>* 194.168.4.0/24 [120/2] via 204.104.4.1, eth0, 00:02:03
C>* 195.168.5.0/24 is directly connected, eth1
R>* 201.101.1.0/24 [120/2] via 202.102.2.1, eth3, 00:02:02
C>* 202.102.2.0/24 is directly connected, eth3
C>* 204.104.4.0/24 is directly connected, eth0
C>* 210.110.10.0/24 is directly connected, eth2
```

Canberra Router

```
R>* 0.0.0.0/0 [120/3] via 204.104.4.2, eth0, 00:01:38
R>* 172.16.0.0/16 [120/3] via 204.104.4.2, eth0, 00:01:38
R>* 192.168.2.0/24 [120/2] via 204.104.4.2, eth0, 00:01:39
R>* 193.168.3.0/24 [120/3] via 204.104.4.2, eth0, 00:01:39
C>* 194.168.4.0/24 is directly connected, eth3
R>* 195.168.5.0/24 [120/2] via 204.104.4.2, eth0, 00:01:39
R>* 201.101.1.0/24 [120/3] via 204.104.4.2, eth0, 00:01:39
R>* 202.102.2.0/24 [120/2] via 204.104.4.2, eth0, 00:01:39
C>* 204.104.4.0/24 is directly connected, eth0
R>* 210.110.10.0/24 [120/2] via 204.104.4.2, eth0, 00:01:39
```

USA Router

```
S>* 0.0.0.0/0 [1/0] via 172.16.0.2, eth1
C>* 172.16.0.0/16 is directly connected, eth1
R>* 192.168.2.0/24 [120/2] via 210.110.10.1, eth0, 00:00:15
R>* 193.168.3.0/24 [120/3] via 210.110.10.1, eth0, 00:00:14
R>* 194.168.4.0/24 [120/3] via 210.110.10.1, eth0, 00:00:15
R>* 195.168.5.0/24 [120/2] via 210.110.10.1, eth0, 00:00:15
R>* 201.101.1.0/24 [120/3] via 210.110.10.1, eth0, 00:00:14
R>* 202.102.2.0/24 [120/2] via 210.110.10.1, eth0, 00:00:15
R>* 204.104.4.0/24 [120/2] via 210.110.10.1, eth0, 00:00:15
C>* 210.110.10.0/24 is directly connected, eth0
```

ACTIVITY B: Transport Layer

Activity B.1: TCP Protocol: Reliable Communication

Open Wireshark in the VM and start capturing traffic on `enp0s3` interface. Open Firefox and clear the history and cache then visit the page
http://shell.cas.usf.edu/mccook/uwy/hyperlinks.html
Stop the capture and answer the following questions.

a) Find the `GET` request for the page `HTTP GET` message (`GET /mccook/uwy/hyperlinks.html HTTP/1.1`). and select the packet in the top pane. Is this the beginning of the communication between the client (VM) and the (Web server)? If no find out the first datagram sent from the client to the server to start the communication.

No! as TCP requires a connection to be established, we need to find the `TCP SYN` request from the client.

There are few ways to do this if there are too many captured packets.

One way is to right click on the selected packet and choose follow TCP stream. This will open a new window showing the communicated content between the client and server. You can close this window. You should now see a filter added in the **Display Filter** bar as `tcp.stream eq 3` (the last number may be different for you as it depends on the number of TCP connections at the time of capture).

Another way is to put a display filter using the `ip.addr eq 131.247.250.66` to limit the traffic between the client and Monash web server. This also allows looking at the transmitted picture from the server.

b) For each of the datagrams from the TCP `SYN` request to `HTTP/1.1 200 OK` observe the values of `Sequence number` and `Acknowledgement number`. What values are highlighted in the raw section (third window pane) when you select the mentioned fields in the middle section? Explain why Wireshark shows `(relative sequence number)` and `(relative ack number)`.

The client chooses a random sequence number in the `TCP SYN` request. The server also chooses a random number as its own sequence number for the `TCP SYN/ACK`. In the communication that follows the sequence number is incremented by the number of bytes in the message and the acknowledgement is the next expected byte added to the initial sequence number received from the other end. This, however, will be a bit complex to look at and to identify the relation between the two fields which is why Wireshark make it easy by showing the relative seq/ack values rather than the actual values.

The actual values are highlighted when we select the fields.

c) Identify the `GET` request for image002.gif the picture to HTTP/1.1 200 OK. How many TCP segments are used to transfer the gif file? How TCP is able to put the picture back together? What is the size of the picture?

This may be different for you, but in my case 2 segments. This is shown by Wireshark in the middle section under `[2 Reassembled TCP Segments]` when `HTTP/1.1 200 OK` segment for the picture is selected. This is Wireshark analysis that that shows the segment numbers (`in my case #53 & #56 and byte sizes for each segment as in my case it is 2920 bytes & 441 bytes`).

Each segment also between the request and the HTTP status shows the frame number of the reassembled PDU in the middle section.

The sequence numbers along with the segment length allows TCP to track the received segments and if there are any gaps it is going to request the missing segments by repeating the acknowledgement for next expected byte which would be the beginning of the gap. It will keep the segments received out of order and acknowledge them cumulatively once the missing segments are received. The other end when receives a repeated acknowledgement for a segment will retransmit the segment.

The size of the reassembled segments is 3361 and the size of the image002.gif is 3104 bytes. (`Select compuserve GIF File Interchange Format in middle pane under HTTP protocol and the status bar shows the file size`), alternatively the filed `Content-Length` in HTTP protocol header also shows the size of the image.

d) Explain how the TCP connection is closed down?

TCP performs a 4-way connection tear down. The server in this case sends a `FIN` request. The client sends back an `ACK`. The client sends a `FIN` request. The server sends an `ACK`. The two middle steps are sent in the same segment.