School of
Science

CSG1105 Workshop Two

1 INTRODUCTION

This week will be a short lab to build some basic skills and understanding regarding IP addresses and Protocol layering.

2 IP ADDRESSES

An **IP Address** is required for any two devices on an IP-based network to communicate.

2.1 Finding your IP address

Most computer users are unaware of the underlying mechanisms used for establishing an Internet client/server connection. Most modern devices have an operating system that hides the complexity via additional software that attempts to automatically link the device to a Wi-Fi router/modem connected to the Internet Service Provider (ISP) via copper, fibre optic or cellular connections.

2.1.1 Windows

1. Open a command prompt (WinKey+R, "cmd")
2. Type `ipconfig` and Enter

```
 Wireless LAN adapter WiFi:

Connection-specific DNS Suffix .  :  modem
IPv6 Address. . . . . . . . . . . :  2001:8004:1df1:9238:ec12:cd1e:54e:b01e
Temporary IPv6 Address. . . . . . :  2001:8004:1df1:9238:7806:8bac:5606:add4
Link-local IPv6 Address . . . . . :  fe80::ec12:cd1e:54e:b01e%12
IPv4 Address. . . . . . . . . . . :  192.168.0.100
Subnet Mask . . . . . . . . . . . :  255.255.255.0
Default Gateway . . . . . . . . . :  fe80::a691:b1ff:fe71:659f%12
192.168.0.1
```

2.1.2 OS X

1. Use Finder to launch Applications->Utilities->Terminal
2. Type `ifconfig` and Enter

```
en1:  flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
options=400<CHANNEL_IO>
ether 1c:36:bb:ef:cf:3a
inet6 fe80::495:223c:a5eb:6f39%en1 prefixlen 64 secured scopeid 0x5
inet 10.1.1.8 netmask 0xffffff00 broadcast 10.1.1.255
nd6 options=201<PERFORMNUD,DAD>
media:  autoselect
status:  active.
```

2.1.3 Private and Public IP

For both Windows and OS X, the interface name will be dependant upon your system configuration and also if you're using a cabled or wired connection. If working from home, the IP may begin with either 192.168 or 10.0. These both belong to **private** IP ranges. They will be mapped to a public IP using a protocol called **NAT** (More on this later).
To find your public IP, use a browser to open **google.com** and search for "what is my IP". My public IP is 58.169.xxx.yyy **(don't reveal your public IP!)**. Whatever is returned will have been allocated to you by your ISP automatically and may change from time to time. Later we will be looking at the mechanism for how IPs are allocated, both private and public.

2.1.4 Network Masks

The Windows example shows a dotted-decimal subnet mask of `255.255.255.0`.
In binary, this is `11111111.11111111.11111111.00000000`.
Applying the subnet mask to the IP will reveal a network of **192.168.0.0**.
The OS X example shows a subnet mask (or netmask) as a hexadecimal number `0xffffff00`.
In binary, this is also `11111111.11111111.11111111.00000000` (check this for yourself).
Applying the subnet mask to the IP will reveal a network of **10.1.1.0**.

In other words, both these networks have 24 bits of network address and 8 bits of host address.

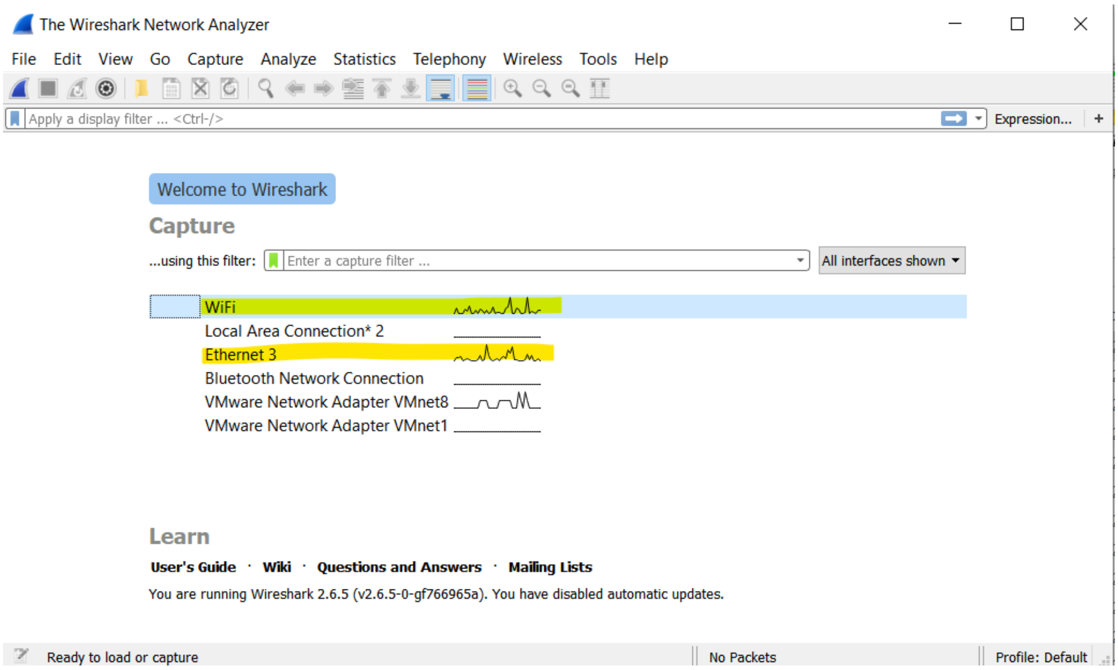
3 Using Wireshark

3.1 What is Wireshark?

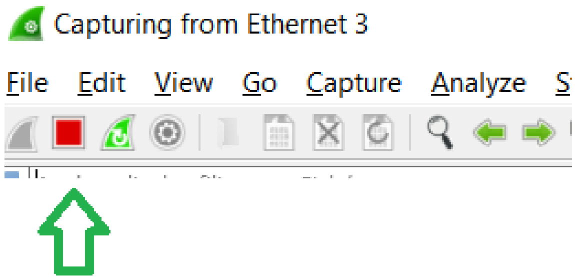
Wireshark is a network packet analyser. It allows us to capture traffic on a network interface and save it to a file for latter analysis. Wirehark is available at no cost for Windows, OS X and Linux. It may be downloaded from https://www.wireshark.org/. (If you are running this on your own machine, please download and install Wireshark at this point. Follow the instructions for your OS/platform.) In today's exercise, we are going to use it to examine protocol stacking.

3.1.1 Run Wireshark

Use the appropriate OS menu to locate and launch Wireshark

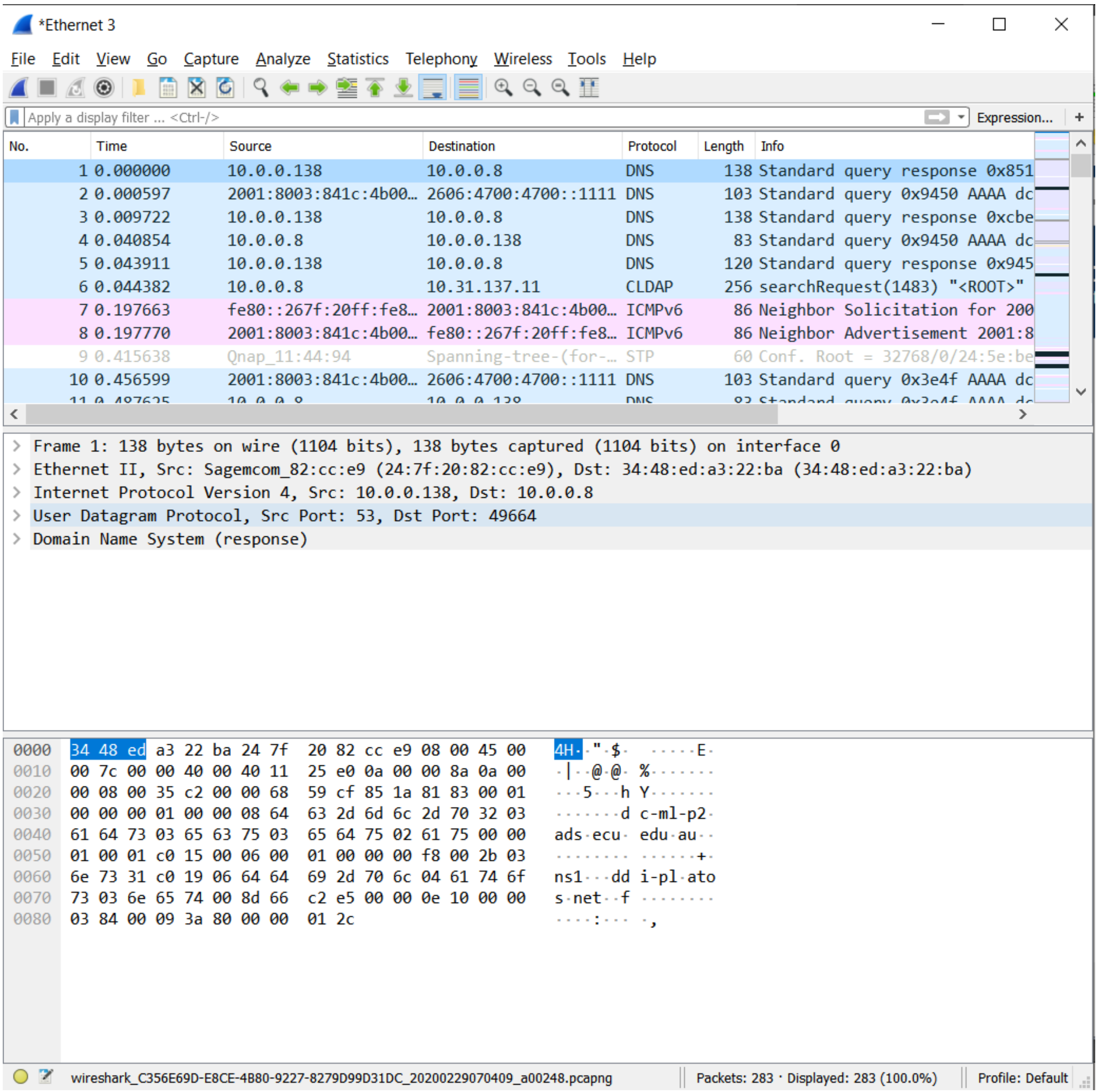


1. In the above screen capture, I have highlighted the two available network interfaces, WiFi and Wired (Ethernet 3). Again, this will vary, depending upon the PC you are working on. You can see the activity graph, indicating traffic on the specific interface. Double-clicking on an interface begins capturing traffic from that interface.
2. Double click on the interface that connects your PC to the Internet.
3. Quickly open a browser to http://www.bom.gov.au
4. Wait for a minute, then stop the capture by clicking the stop icon.



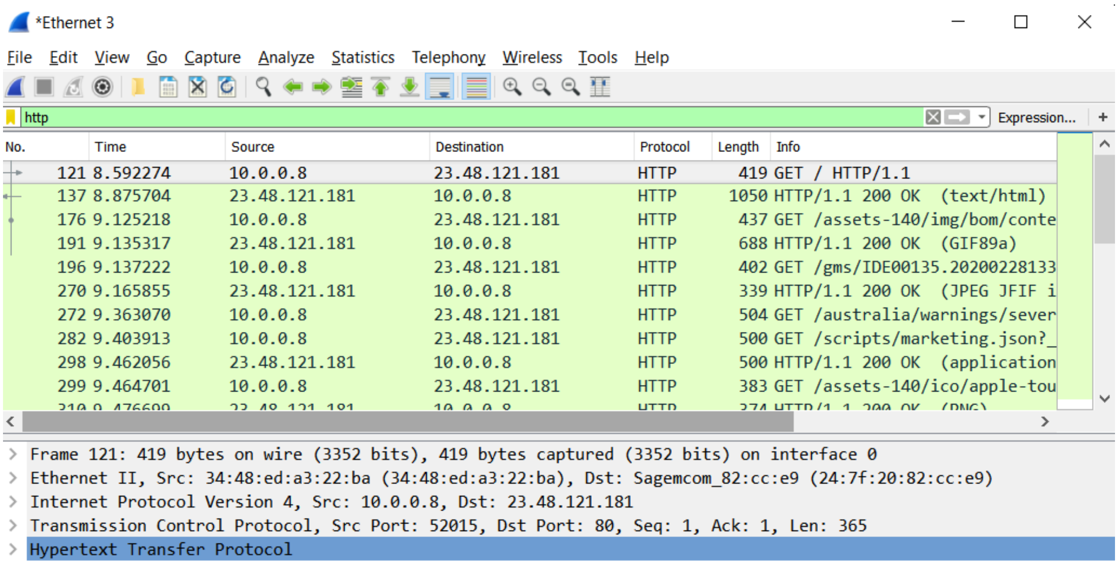
5. . You'll notice there is three main sections to the captured data.

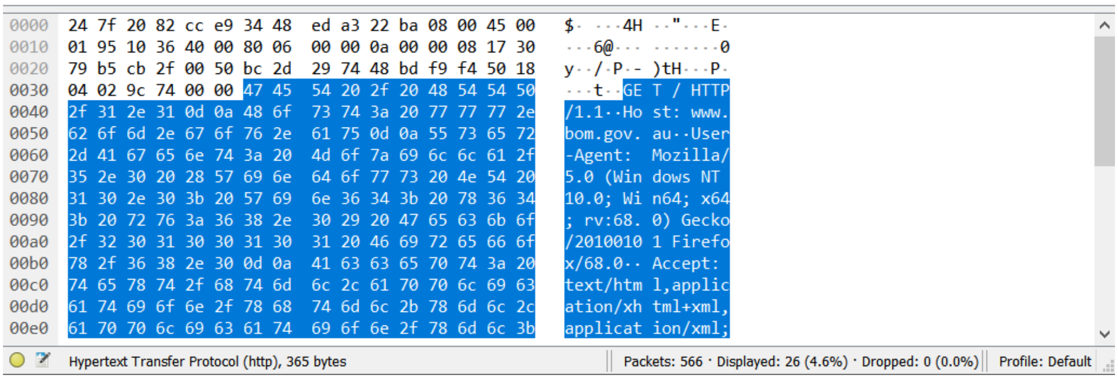
   a) The list of all captured packets; this contains information such as source and destination address, protocol, length and info.
   b) A layered view of the selected packet we are viewing, with more information provided to us
   c) The view of the frames in hexadecimal and ASCII



We can order our list of captured packets by any of the columns, so if we wanted to view the largest traffic, we can sort by Length or how many ARP or TCP requests are being made we could sort by Protocol, for now though, leave it sorted by the packet number.

6. Above our list of captured packets, you'll notice a **Apply a display filter:** field. In this field we want to track the information we just generated by our telnet request. So, let's filter only HTTP protocol packets by typing in HTTP and then clicking on **apply** (the white arrow on a grey background) to the right. Below is an example of our new view:





7. If we start selecting these packets we can see the information in our layer viewer has the same structure for every HTTP packet, but the information contained within changes. Without clicking on the '>' next to each one we can select the layer and find the part of the frame in the frame viewer below that it responds to. If we select the first one starting with 'Frame' you'll see the entire frame below is highlighted, moving down, when we select 'Ethernet II' we can see what part of the frame the Ethernet information is (MAC addresses etc.). Selecting 'Internet Protocol Version 4' shows the source and destination addresses part of the frame; selecting 'Transmission Control Protocol' shows the where in the frame the ports and sequence information is stored. Finally, selecting 'HTTP' shows us the section of the frame specifying that it is HTTP and the data being transferred. Clicking on the '>' next to each layer allows us to see the individual information provided in each layer, clicking on that newly revealed information shows where in the frame it is stored.

8. In the following steps, refer back to the lecture notes regarding the TCP/IP network model layers and PDU names for each layer.

9. Clicking on the '>' next to **Transmission Control Protocol** displays the contents of the TCP header. In this header, information on the TCP **segment** may be seen. In the screen capture, the source port (52015) and destination port (80, standard for a HTTP server) are followed by additional fields including the segment length and sequence numbers.

```
> Frame 121: 419 bytes on wire (3352 bits), 419 bytes captured (3352 bits) on interface 0
> Ethernet II, Src: 34:48:ed:a3:22:ba (34:48:ed:a3:22:ba), Dst: Sagemcom_82:cc:e9 (24:7f:20:82:cc:e9)
> Internet Protocol Version 4, Src: 10.0.0.8, Dst: 23.48.121.181
v Transmission Control Protocol, Src Port: 52015, Dst Port: 80, Seq: 1, Ack: 1, Len: 365
    Source Port: 52015
    Destination Port: 80
    [Stream index: 3]
    [TCP Segment Len: 365]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 366    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)

0030  04 02 9c 74 00 00 47 45  54 20 2f 20 48 54 54 50   ···t··GE T / HTTP
0040  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 77 77 77 2e   /1.1··Ho st: www.
0050  62 6f 6d 2e 67 6f 76 2e  61 75 0d 0a 55 73 65 72   bom.gov. au··User
0060  2d 41 67 65 6e 74 3a 20  4d 6f 7a 69 6c 6c 61 2f   -Agent:  Mozilla/
0070  35 2e 30 20 28 57 69 6e  64 6f 77 73 20 4e 54 20   5.0 (Win dows NT
```

10. Clicking on the '>' next to **Internet Protocol Version 4** displays the contents of the IP header. In this header, information on the IP **packet** may be seen. In this capture, a series of field values followed by the source address (10.0.0.8) and destination address (23.48.121.181) may be seen.

```
> Frame 121: 419 bytes on wire (3352 bits), 419 bytes captured (3352 bits) on interface 0
> Ethernet II, Src: 34:48:ed:a3:22:ba (34:48:ed:a3:22:ba), Dst: Sagemcom_82:cc:e9 (24:7f:20:82:cc:e9)
v Internet Protocol Version 4, Src: 10.0.0.8, Dst: 23.48.121.181
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 405
    Identification: 0x1036 (4150)
  > Flags: 0x4000, Don't fragment
    Time to live: 128
    Protocol: TCP (6)
    Header checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]

0000  24 7f 20 82 cc e9 34 48  ed a3 22 ba 08 00 45 00   $· ···4H ··"···E·
0010  01 95 10 36 40 00 80 06  00 00 0a 00 00 08 17 30   ···6@··· ·······0
0020  79 b5 cb 2f 00 50 bc 2d  29 74 48 bd f9 f4 50 18   y··/·P·- )tH···P·
0030  04 02 9c 74 00 00 47 45  54 20 2f 20 48 54 54 50   ···t··GE T / HTTP
0040  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 77 77 77 2e   /1.1··Ho st: www.
```

11. Clicking on the '>' next to **Ethernet II** displays the contents of the Ethernet header. In this header, information on the Ethernet **frame** may be seen. In this capture, the Ethernet source address (23:48:ed:a3:22:ba) and destination address (24:7f:20:80:cc:e9) may be seen (more on these next week). The header also has a Type field which shows IPV4.

```
> Frame 121: 419 bytes on wire (3352 bits), 419 bytes captured (3352 bits) on interface 0
v Ethernet II, Src: 34:48:ed:a3:22:ba (34:48:ed:a3:22:ba), Dst: Sagemcom_82:cc:e9 (24:7f:20:82:cc:e9)
  > Destination: Sagemcom_82:cc:e9 (24:7f:20:82:cc:e9)
  > Source: 34:48:ed:a3:22:ba (34:48:ed:a3:22:ba)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 10.0.0.8, Dst: 23.48.121.181
> Transmission Control Protocol, Src Port: 52015, Dst Port: 80, Seq: 1, Ack: 1, Len: 365
> Hypertext Transfer Protocol

0000  24 7f 20 82 cc e9 34 48  ed a3 22 ba 08 00 45 00   $· ···4H ··"···E·
0010  01 95 10 36 40 00 80 06  00 00 0a 00 00 08 17 30   ···6@··· ·······0
0020  79 b5 cb 2f 00 50 bc 2d  29 74 48 bd f9 f4 50 18   y··/·P·- )tH···P·
0030  04 02 9c 74 00 00 47 45  54 20 2f 20 48 54 54 50   ···t··GE T / HTTP
0040  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 77 77 77 2e   /1.1··Ho st: www.
```

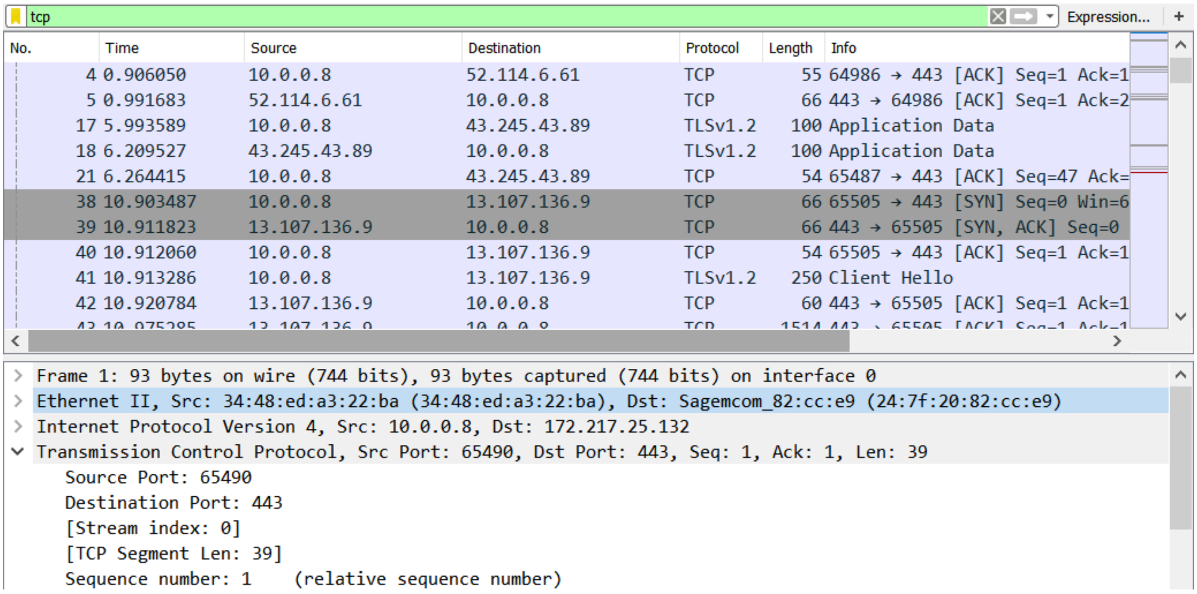
12. Got to the menu at the top of the Wireshark screen and select the **Analyze** option. Scroll down to the **Follow** option and select **HTTP Stream**. This will open a new window displaying the HTTP requests in Red and the HTTP Server responses in Blue. Scroll through the window and try to relate some of the content to what was displayed in the browser.

Wireshark · Follow HTTP Stream (tcp.stream eq 13) · Ethernet 3

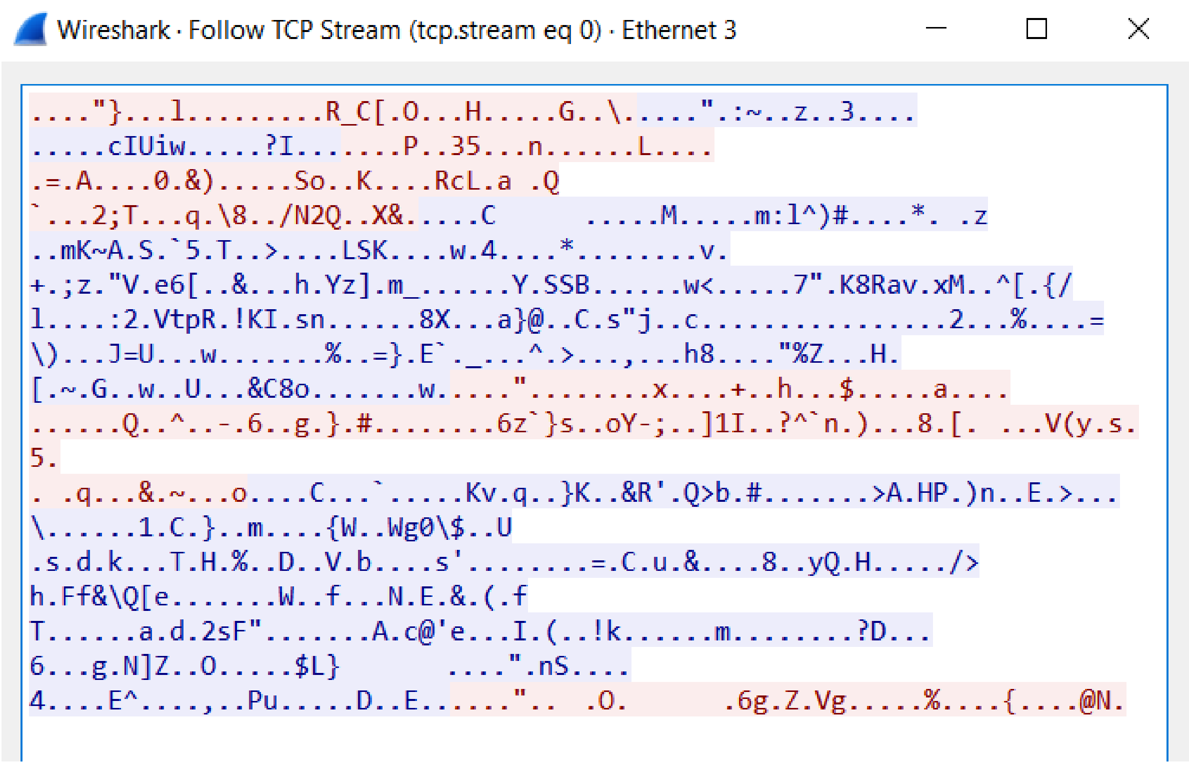
```
GET / HTTP/1.1
Host: www.bom.gov.au
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/
20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/
*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Cookie: _easyab_seed=215.20803983200375
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Content-Encoding: gzip
Content-Type: text/html; charset=UTF-8
Server: Apache
Vary: Accept-Encoding
X-Akamai-Transformed: 9 - 0 pmb=mRUM,2
Date: Sat, 29 Feb 2020 00:34:31 GMT
Content-Length: 9460
Connection: keep-alive
Server-Timing: cdn-cache; desc=HIT
Server-Timing: edge; dur=1

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">
<meta name="robots" content="noodp">
<meta name="viewport" id="viewport" content="width=device-width">
```

13. The server at `http://www.bom.gov.au` is not using encryption, so the contents of the frames may be viewed as we have seen. Start a new Wireshark capture (by clicking on the shark fin icon in the menu) and in your browser, open `https://www.ecu.edu.au`, then stop the capture.

14. In the capture, apply a filter for TCP and find a frame that contains your IP as the origin. You will notice that the destination port is now **443**, indicating that we are communicating with an encrypted HTTP server.



15. Use **Analyse->Follow->TCP Stream**. The resulting capture will be encrypted and you should not be able to decipher anything.



16. Start a new Wireshark capture and let it run for a few minutes as you browse the web. Stop the capture and explore the content captured, noting the types of traffic in the **protocol** column. There are many to be seen, some of which we have discussed (DNS, TCP, HTTP) some of which will be covered in future weeks (ARP, DHCP, ICMP, TLS) and some which are out of scope for this unit.