TCP Packet Encapsulation

ETHERNET	IP	TCP	HTTP	Website

Each data packet (header + encapsulated data) defined by a particular layer has a specific name.

Frame - Encapsulated data defined by the Network Access layer. A frame can have both header and trailer.

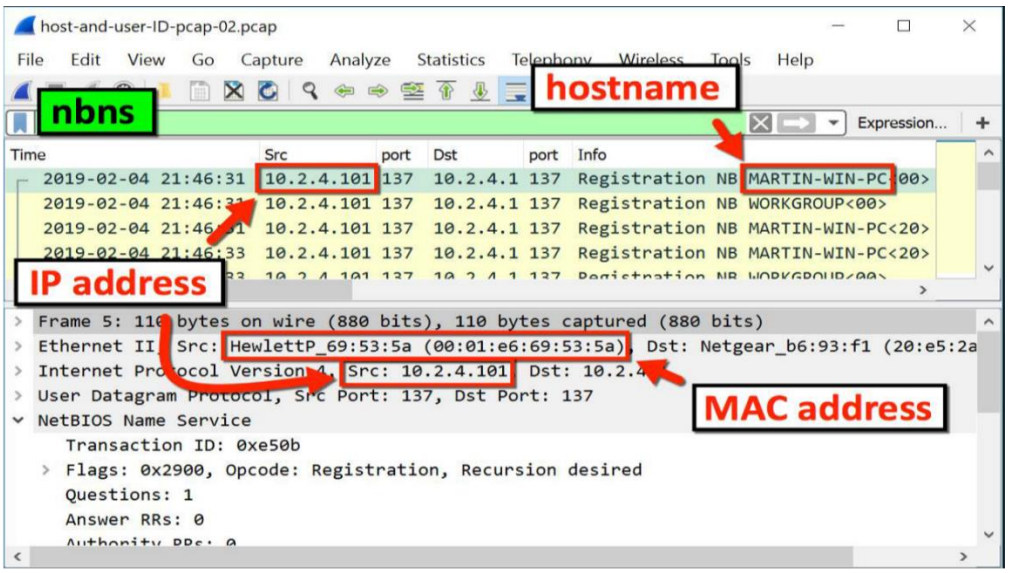
Packet - Encapsulated data defined by the Network Access layer. A header contains the source and destination IP addresses.

MAC On TCP/IP

TCP/IP networks use both IP and MAC addresses. A MAC address will remain fixed to a hardware device, but the IP address may alter dynamically in accordance with its TCP/IP network configuration.

In the OSI model, Internet Protocol operates at Layer 3, while the MAC protocol works at Layer 2.

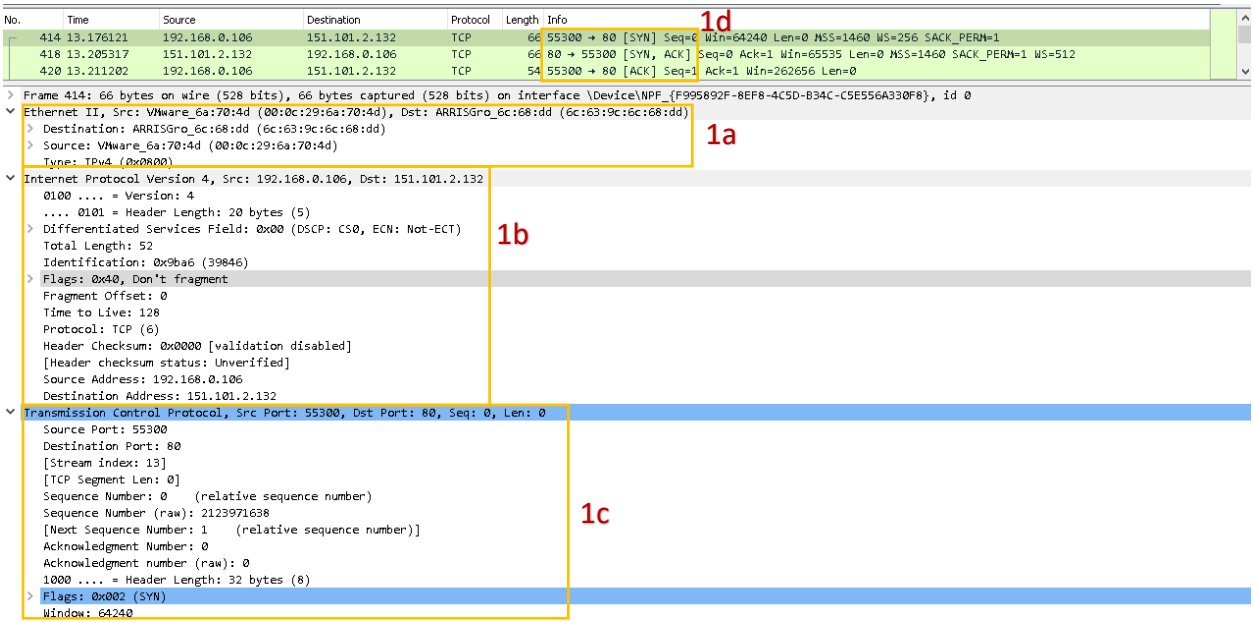
Media Access Control is able to support other networks besides TCP/IP, for this reason.



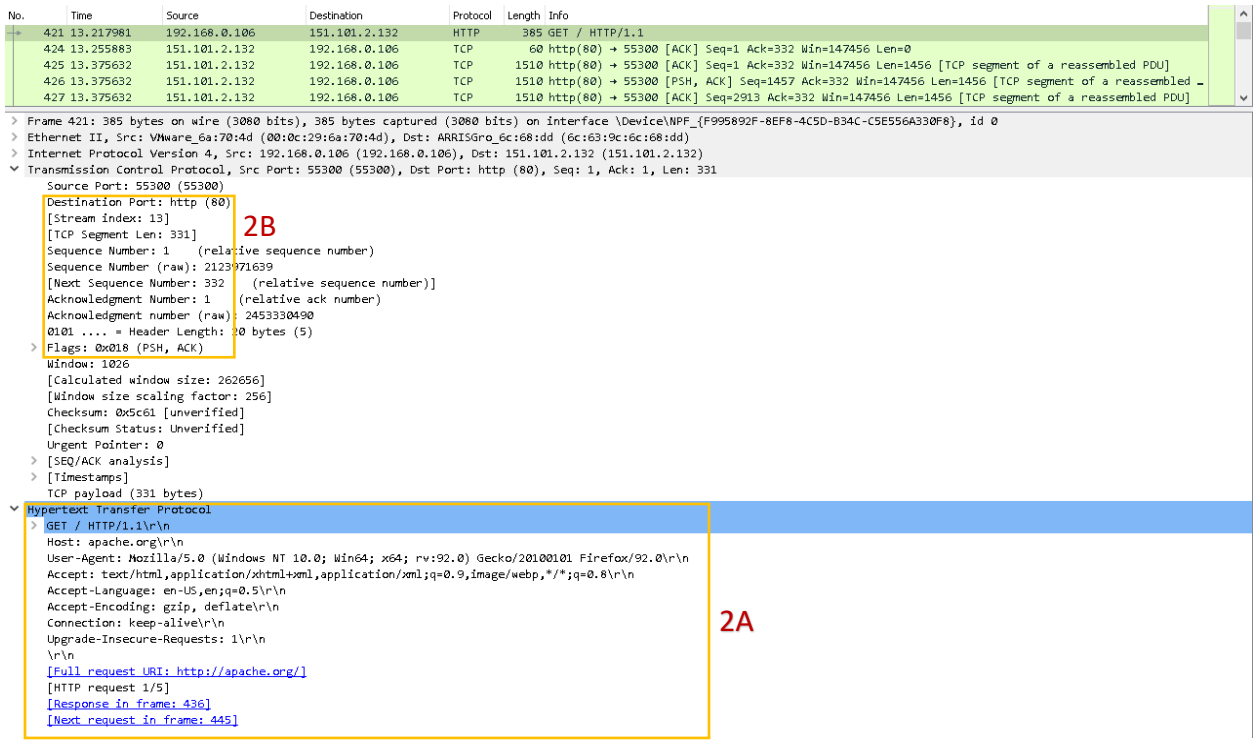
MAC Address - a unique identifier assigned to a network interface controller (NIC) for use as a network address in communications within a network segment.

OSI Model

	OSI	
01	Physical	
02	Data Link	**ETHERNET**
03	Network	**IP**
04	Transport	**TCP**
05	Session	
06	Presentation	
07	Application	**HTTP**

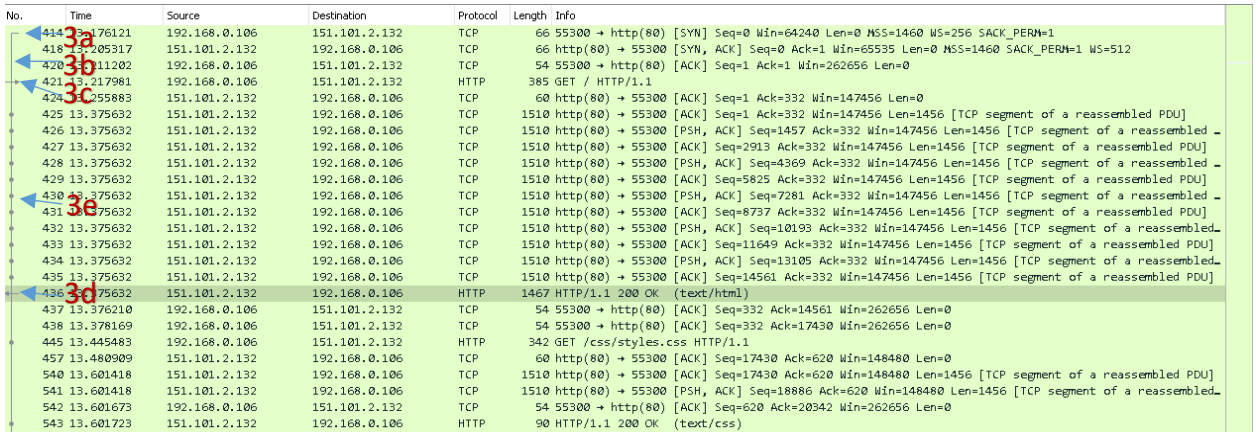


1. When you click on a packet in the capture window, it populates the detail window with information for that packet. I selected the first TCP packet and you can see the information for that packet. The Frame section tells how the packet is put together and info on how it was captured by Wireshark.
   a. The Ethernet II has the source device and destination device, but if you look closely at the destination you can see it is not the final destination. This is the router and its MAC address being used to pass the packet on. So we can use this information sometimes to determine an outbound path.
   b. The Internet Protocol Version 4 section contains the ip address of the sources of the packet and the destination. In this case, the private address of my machine is going to the ip addess of apache.org.
   c. The (TCP) Transmission Control Protocol section contains the source and destination port and it randomize the source and the destination in this case is 80, because we are going to make a request for the webpage content on http. We can also see the SYN flag since this is the initializing packet.
   d. In the packet list window we can see the three way hand shake to start the connection before the http request. Something to note in the detail screen, whenever you see [] around data it means it was added by Wireshark and was not part of the original packet info.



2. A. Now I have selected an http packet. You can see it has the same details as the TCP packet, except it has a Hypertext Transfer Protocol Section added. You can see the request is to Host: apache.org. It shows my browser was Mozilla Firefox, but sometimes it cannot detect the browser.

   B. Notice under the TCP section that the destination port is still 80, but the flag this time is a PSH, ACK. This is because it is making a request for the web content.



3. This is a request and response for a portion of the http page. Packet 421 is the request and packet 436, which is selected is the response. The packets in between are the pieces of the page

requested. By selecting the response I get some visual symbols to help me identify the conversation. Related packet symbols:

a. First packet in a conversation.
b. Part of the selected conversation.
c. Request.
d. Response.
e. The selected packet is related to this packet in some other way, e.g. as part of reassembly.
f. A few not pictured above. Not part of the selected conversation. Selected packet acknowledges this packet. Last packet in conversation.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.106	151.101.2.132	ICMP	74	Echo (ping) request id=0x0001, seq=140/35840, ttl=128 (reply in 2)
2	0.033837	151.101.2.132	192.168.0.106	ICMP	74	Echo (ping) reply id=0x0001, seq=140/35840, ttl=58 (request in 1)
3	1.016270	192.168.0.106	151.101.2.132	ICMP	74	Echo (ping) request id=0x0001, seq=141/36096, ttl=128 (reply in 4)
4	1.047961	151.101.2.132	192.168.0.106	ICMP	74	Echo (ping) reply id=0x0001, seq=141/36096, ttl=58 (request in 3)
5	2.047759	192.168.0.106	151.101.2.132	ICMP	74	Echo (ping) request id=0x0001, seq=142/36352, ttl=128 (reply in 6)
6	2.079530	151.101.2.132	192.168.0.106	ICMP	74	Echo (ping) reply id=0x0001, seq=142/36352, ttl=58 (request in 5)
7	3.062348	192.168.0.106	151.101.2.132	ICMP	74	Echo (ping) request id=0x0001, seq=143/36608, ttl=128 (reply in 8)
8	3.094182	151.101.2.132	192.168.0.106	ICMP	74	Echo (ping) reply id=0x0001, seq=143/36608, ttl=58 (request in 7)

4. This is standard ping request and reply.

No.	Time	Source	Destination	Protocol	Length	Info
7	1.760914	192.168.0.106	192.168.0.1	DNS	77	Standard query 0xae4e A b-ring.msedge.net
8	1.773954	192.168.0.1	192.168.0.106	DNS	212	Standard query response 0xae4e A b-ring.msedge.net CNAME b-ring.b-9999.b-msedge.net CNAME b-9999.b-ms…
33	1.981485	192.168.0.106	192.168.0.1	DNS	77	Standard query 0x4059 A a-ring.msedge.net
34	1.993611	192.168.0.1	192.168.0.106	DNS	212	Standard query response 0x4059 A a-ring.msedge.net CNAME a-ring.a-9999.a-msedge.net CNAME a-9999.a-ms…
66	2.177771	192.168.0.106	192.168.0.1	DNS	73	Standard query 0x47a5 A fp.msedge.net
69	2.190525	192.168.0.1	192.168.0.106	DNS	273	Standard query response 0x47a5 A fp.msedge.net CNAME 1.perf.msedge.net CNAME a-0019.a-msedge.net CNAM…
159	8.315266	192.168.0.106	192.168.0.1	DNS	81	Standard query 0xc00c A ow1.res.office365.com
160	8.326180	192.168.0.1	192.168.0.106	DNS	477	Standard query response 0xc00c A ow1.res.office365.com CNAME ow1.res.office365.com.edgekey.net CNAME …
264	8.705441	192.168.0.106	192.168.0.1	DNS	85	Standard query 0x69cd A bgpdefault-ata.msedge.net
273	8.765029	192.168.0.1	192.168.0.106	DNS	269	Standard query response 0x69cd A bgpdefault-ata.msedge.net CNAME m1-0155.m1-msedge.net A 13.107.218.1…

5. This is a machine making DNS request for several different sites and the response.

```
> Frame 7: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface \Device\NPF_{F995892F-8EF8-4C5D-B34C-C5E556A330F8}, id 0
> Ethernet II, Src: VMware_6a:70:4d (00:0c:29:6a:70:4d), Dst: ARRISGro_6c:68:dd (6c:63:9c:6c:68:dd)
v Internet Protocol Version 4, Src: 192.168.0.106 (192.168.0.106), Dst: 192.168.0.1 (192.168.0.1)
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 63
    Identification: 0xef8e (61326)
  > Flags: 0x00
    Fragment Offset: 0
    Time to Live: 128
    Protocol: UDP (17)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.0.106 (192.168.0.106)
```

```
0000  6c 63 9c 6c 68 dd 00 0c  29 6a 70 4d 08 00 45 00   lc·lh··· )jpM··E·
0010  00 3f ef 8e 00 00 80 11  00 00 c0 a8 00 6a c0 a8   ·?····· ·····j··
0020  00 01 fc e0 00 35 00 2b  81 f8 ae 4e 01 00 00 01   ·····5·+ ···N····
0030  00 00 00 00 00 00 06 62  2d 72 69 6e 67 06 6d 73   ·******b -ring·ms
0040  65 64 67 65 03 6e 65 74  00 00 01 00 01            edge·net ·····
```

6. The other portion of the packet window is the packet bytes pane. The packet bytes pane shows the data of the current packet (selected in the “Packet List” pane) in a hexdump style.