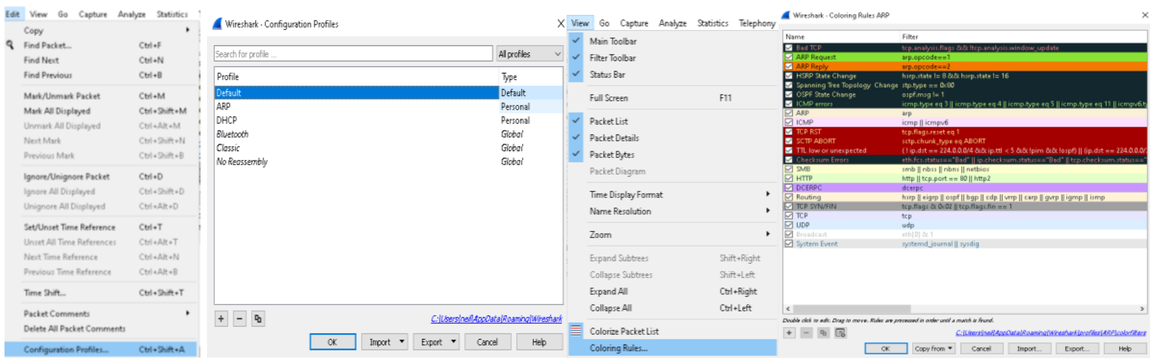
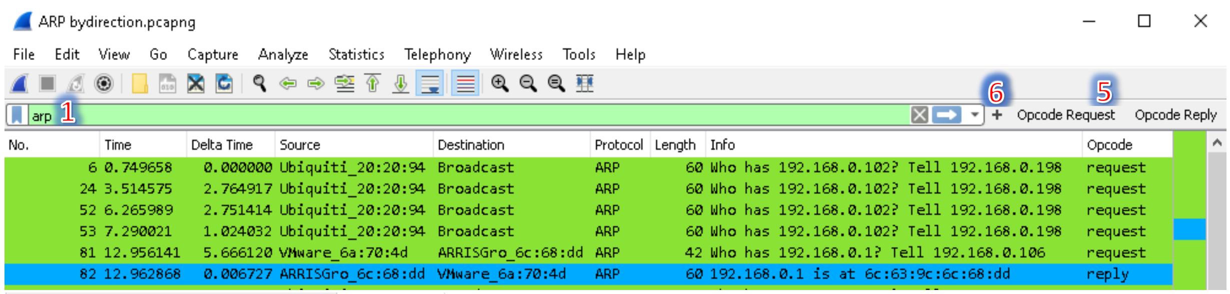
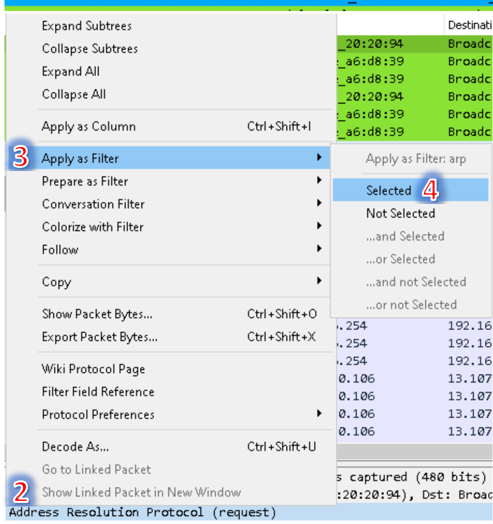


To get to profiles in Wireshark, click Edit and Configuration Profiles. Inside the Configuration Profiles window we can choose profile for current use. We can also add, import, or export profiles. On the bottom of the window is the path that personal profiles are stored. I would not suggest altering the Default Profile, always make a new one. You can find Coloring Rules under View menu. Coloring rules can help you identify specific traffic in a profile by site. Always leave the Bad TCP filter on top.





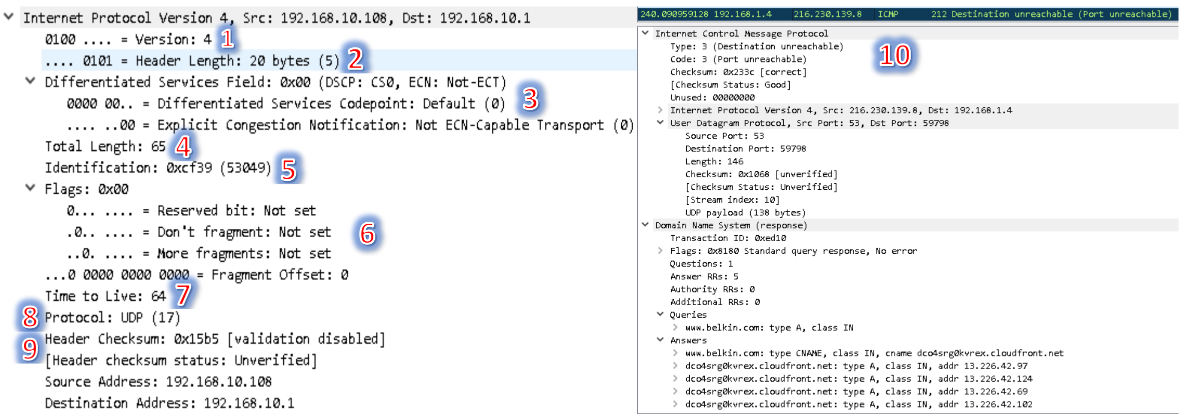
If you notice above in the Display Filter box (1), we have limited the traffic for this capture to only ARP traffic. It is easy to apply filter by selecting a packet of interest from traffic window above and right clicking a protocol like (2) ARP on the left and choosing (3) Apply as Filter and (4) Selected. You can easily add a display filter button (5) for commonly used filters, by clicking the + (6) and giving the button a Label Name and click Okay.

Core Network Protocols: UDP, TLS, IPv6, DNS, ARP, IP, ICMP. These protocols make up the majority of the traffic on your network. They are the most useful in finding issues on your network.

Address Resolution Protocol (ARP) bridges layer 2 and layer 3. ARP helps devices on the network keep track of which Media Access Control (MAC) addresses have which IP addresses, in order to complete their packets and send. ARP request are broadcast and that is why you usually see it anywhere you do a capture. ARP reply is unicast and usually only the requesting client sees those packets. You can identify scanning or unwanted traffic with ARP if one machine is sending ARP request for every IP address on the

network. Most clients don't do peer-to-peer traffic, so finding machines that are requesting IP addresses of other workstations is a good indication of infection propagating or malicious activity.
Devices keep an ARP table that it can reference, so that it doesn't have to do an ARP request every time it needs to communicate with another device. Devices will keep the most recent ARP reply that they receive. So if two devices reply to a request, the last reply that it gets will be the one it keeps.

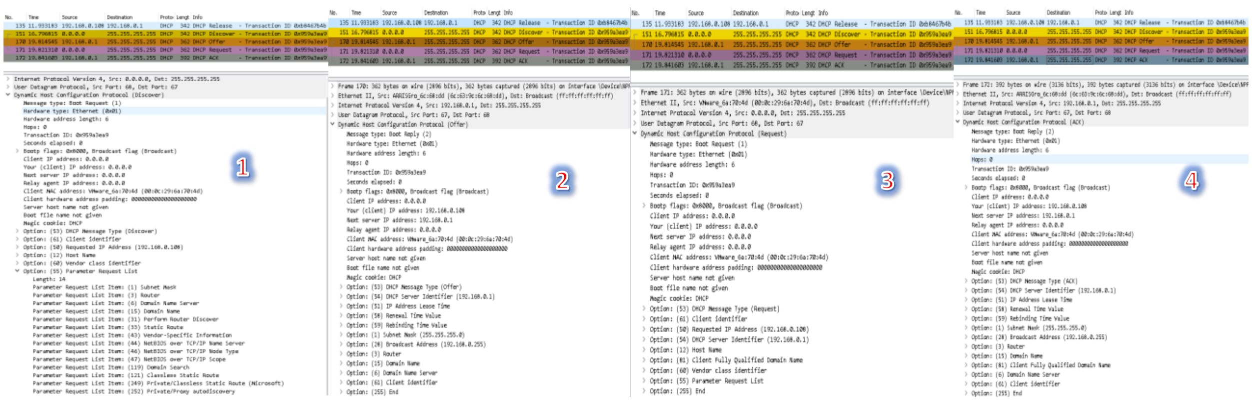
Internet Protocol (IP) is the network layer communication method. It allows end to end communication rather than point to point like Ethernet. IP Header Structure:



(1) Version- Usually 4
(2) Internet Header Length- Length of IP header in 4-byte (32-bit) units.
(3) Differentiated Services Codepoint (DSCP)- How a packet priority is marked. QoS uses this Byte to prioritize traffic. Explicit Congestion Notification (ECN) is an optional feature used by ECN-aware devices to notify devices of network congestion to avoid dropped packets. Standard machine traffic is Non ECN-Capable Transport (Non-ECT).
(4) Total Length- Length of the whole packet including the header in bytes.
(5) Identification- 16-bit number in each packet to help the destination host reassemble packet fragments.
(6) Flags- Only first 3 bits are defined. Bit 1 is reserved and always 0. Bit 2 tells if the packet can be fragmented. Bit 3 is set to 0 if it is the last fragment and 1 if more fragments follow. Fragment offset tells how many 8-byte blocks are in the packet fragment.
(7) Time to Live (TTL)- Number of seconds a packet can take to reach the destination. Routers decrease this value by a preconfigured amount, usually 1 and discard packets that arrive with field set to 0. Most networks interpret this field as a simple hop count between routers. TTL is usually set as 255, 128, or 64. So if a ping TTL return value is 60, we can be pretty sure that there are 4 hops to the destination.
(8) Protocol: Tells what protocol section is next and Wireshark uses this field to determine which dissector to load next. TCP, ICMP, UDP, etc.
(9) Header Checksum- Used to verify that the header has not been altered.

Internet Control Message Protocol (ICMP): An ICMP packet pictured above (10), we can see it is not used just for ping. It can be used to send information about network traffic and alerts about network problems. It can send returns to devices for routing issues, port problems, network congestion, or TTL issues. If you watch a capture for ICMP return packet on Network Unreachable, it can give you a place to start looking for potential problems. It will tell you if your traffic can't be routed because your packet needs to be fragmented.

User Datagram Protocol (UDP): Used for time-sensitive traffic like voice, video, and DNS lookups. By not establishing a connection like TCP, UDP is able to speed up data transfer. Because it doesn't establish a connection before sending data it is less reliable than TCP. If a datagram is lost in transit, there is no way for the sending device to know to resend it. So when you have a scratchy VOIP phone call you are experiencing datagram loss, but your conversation is not delayed like if it were being sent over TCP. This also makes UDP more susceptible to DDoS attacks. Attackers use the lack of TCP handshake to flood a target with UDP traffic and causes the denial-of-service for real traffic.

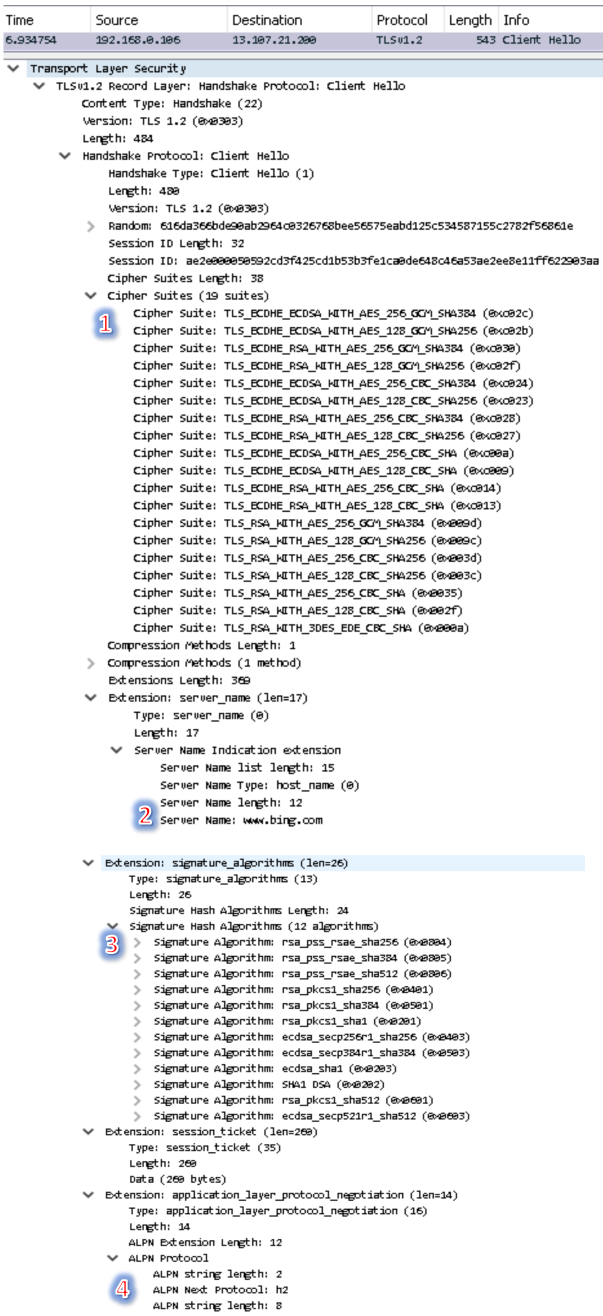


Dynamic Host Configuration Protocol (DHCP) is how we assign addresses dynamically to devices on the network from a single location. The DHCP procedure consist mainly of four packets. (1) The Discover packet is broadcast on the devices local segment to everyone. In the packet the device can ask for different options like DNS server, subnet mask, and default gateway. (2) The DHCP server responds with an offer from its ip address pool. (3) The device responds with a request that is broadcast again to everyone, so that if there were offers from other servers on the network, they know which offer was accepted. (4) The server responds with an acknowledgment and makes the IP official. The device then broadcast an ARP message with its MAC address and new IP address. This is how devices determine if they have a duplicate address. If it doesn't get a response that another device has that IP address, then it broadcast that it now has that IP address.

Domain Name System (DNS) is a naming system where names are resolved to the associated IP addresses and vice versa. It is similar to a phonebook, but for the internet or a local network. DNS is pretty simple it sends a request and the request is forwarded until a server has a cache of the requested name or it gets to an authoritative name server for that domain to be resolved and a response is sent back with the IP address for that domain name.

File Transfer Protocol (FTP) is a standard internet protocol used to transfer data from one device to another. The model it is built on, one devices acts as a server and the other devices as a client. It works really well with one big downfall, it is not a secure protocol and using Wireshark we can see login and password passed in plain text. Secure File Transfer Protocol (SFTP) addresses this issue by encrypting the login and transfer by a secure channel via Secure Shell (SSH). The other popular method of file transfer is Trivial File Transfer Protocol (TFTP) and is used mainly on a local network over UDP to push configuration information to devices like switches and VOIP phones.

Hypertext Transfer Protocol Secure (HTTPS) or the secure version of HTTP using Secure Socket Layer (SSL) / Transport Layer Security (TLS). TLS is an updated and more secure version of SSL. TLS 1.3 is the most common method currently for encryptions. Some older websites might still use TLS 1.2. TLS uses mathematically generated keys that the client and server agree on to encrypt and decrypt data to be exchanged. The client sends a public private key pair that initiates the handshake and tells the server which cipher it is capable of, as well as the TLS version. The server responds with a generated random key, the cipher that it chose, and data encrypted with the key. With one round trip communication both sides can now encrypt and decrypt data being sent. Below we can seem an example of a Client Hello and the response of Server Hello. Everything after this is scrambled, because it is encrypted.



The Client sends a Hello with (1) Ciphers that it can support. We can see the (2) server this Hello is going to and (3) Signature Hash that it can support. It declares the (4) http version that it can support.

The server responds with the chosen (5) Cipher, (6) http version, (7) Certificate, (8) Public Key, and (9) Hash.