TCP/IP Protocols: ARP and IP

ADDRESS RESOLUTION PROTOCOL(ARP)

Local IP Address

When two computers try to communicate, an ARP request is initiated. If the IP address is on the local network, the source host checks its ARP cache to see if it already has the hardware address (MAC address) of the receiving host. If not, a broadcast is sent to all local hosts. If the receiving host finds that the IP address of the source host matches its own then it sends a reply to the source host with its hardware address. When received by the source host, its ARP cache is updated to include this info. If no hosts respond to the broadcast then the request is discarded.

Remote IP Address

This is a little different. When the destination address is found to be a remote host, the source host checks the local routing table for a path to the receiving host. If one is not found then a broadcast is sent to the router (gateway). The router replies with its hardware address and then the packet is sent to the router. Essentially the router follows the same pattern. It checks its cache for a path to the receiving host. If one is found then it forwards the packet. If not, it sends a broadcast and waits for a reply from the host. It may again determine that it is a remote host and then the process repeats with a broadcast to the next router and so on
and so forth. Once the receiving host gets the request, it sends an ICMP echo request.

Would You Like To Know More?

ARP entries can be static or dynamic. If a dynamic entry is not used within 2 minutes then it is deleted. If it is used then it will remain for 10 minutes. A static entry will hang around until the computer is rebooted, it is deleted with arp -d, or a new hardware address is received via broadcast in which case the entry becomes dynamic. A tornado or earthquake could also remove your static entries.

ARP COMMANDS

1) `arp -a`
   `arp -g`

- Both of these commands do the same thing. They display the contents of the current arp cache.

2) `arp -s <ip_address> <hardware_address>`

- This commands a static entry to the arp cache.

3) `arp -d ip_address`

- Removes an entry from the arp cache.

```
C:\>arp -a

Interface: 172.0.0.127 on Interface 0x1000003
  Internet Address      Physical Address      Type
  172.0.0.2             00-00-00-00-00-00     invalid
  172.0.0.3             00-20-c5-00-25-ee     dynamic

C:\>
```

INTERNET PROTOCOL (IP)

Background

IP is a connectionless protocol, which means that a session is not created before sending data. IP is responsible for addressing and routing of packets between computers. It does not guarantee delivery and does not give acknowledgement of packets that are lost or sent out of order as this is the responsibility of higher layer protocols such as TCP.

IP HEADER STRUCTURE

***VERSION:
This field uses 4 bits to denote the version of IP.

***HEADER LENGTH:
4 bits denote the number of 32-bit words in the header. The minimum length is 20 bytes.

***TYPE OF SERVICE:
8 bits that indicate the quality of service that the packet should receive. Includes precedence, delay, throughput and reliability.

***TOTAL LENGTH:
16 bits denote the total length of the packet.

***IDENTIFICATION:
16 bits are used as a unique identifier so the packet can be reassembled in the event that it is fragmented.

***FRAGMENTATION FLAGS:
3 bits used in the fragmentation process.

***FRAGMENT OFFSET:
13 bits used to determine the location of the fragment in regards to the original IP packet.

***TIME TO LIVE (ttl):
8 bits that indicate the maximum number of hops that a packet can travel before being thrown away. Asleep yet?

***PROTOCOL:
8 bits are used to identify the original upper-layer protocol used.

***HEADER CHECKSUM:
16 bits used to check for errors in the header only.

***SOURCE ADDRESS:
32 bits that indicate the IP address of the sending host.

***DESTINATION ADDRESS:
32 bits that indicate the IP address of the receiving host.

***OPTIONS AND PADDING:
Stores IP options.

TRANSMISSION CONTROL PROTOCOL (TCP)

BACKGROUND

As opposed to IP, TCP is connection oriented and assures reliable delivery of packets. When the destination host receives a segment it sends back an acknowledgement (ACK). If an ACK is not received by the source host within a certain period of time then the data is retransmitted. TCP uses sockets and ports to exchange data between applications. Ports provide a specific and universal location for message delivery, while sockets use the host IP address, port number and the type of service (TCP or UDP) to create a reliable connection. TCP uses sliding windows to buffer data between hosts. A buffer that is too large or small can cause poor network performance. For example, if you are shotgunning a beer your throat is like the buffer. If your throat isn't opened wide enough then the beer travels to your stomach very slowly. If your throat is open too wide, then some of the beer packets get lost in your lungs or you just throw up. You then have to retransmit the beer back to your stomach. (Sick!)

THREE-WAY HANDSHAKE

A TCP session begins with a three-way handshake. This process synchronizes the sending and receiving of data.

1) The source host sends a segment with the SYN flag set "on."

2) The destination host sends a reply with SYN flag "on", a sequence number and an ACK that relays the next packet that the destination host is expecting.

3) The source host sends and ACK with received sequence number and an acknowledgement number. The session is ended with a similar process.

TCP HEADER STRUCTURE

***SOURCE PORT:

This is the TCP port of the source host.

***DESTINATION PORT:

TCP port of receiving host

***ACKNOWLEDGEMENT NUMBER:

The sequence number of the packet that the receiving host is expecting next.

***DATA LENGTH:

Length of segment

***RESERVED:

Crappy explanations everywhere so it must not be important.

***FLAGS:

Denotes the content of the segment.

***WINDOW:

Specifies how much space is left in the TCP window.

***CHECKSUM:

Makes sure that the header is not corrupted.

***URGENT POINTER:

If there is a flag in the flags section that indicates that there is urgent data included, this field shows where the end of this urgent data is.

USER DATAGRAM PROTOCOL (UDP)

BACKGROUND

UDP is a connectionless service that sends small amounts of data at one time and does not guarantee delivery. It is commonly used with applications such as NETSTAT, TFTP, SNMP, NETBIOS name service and NETBIOS datagram service. Like TCP, UDP uses ports to provide the location to send packets.

HEADER STRUCTURE

***SOURCE PORT:

UDP port of the source host

***DESTINATION PORT:

UDP port of receiving host

***MESSAGE LENGTH:

The total size of the UDP packet.

***CHECKSUM:

Verifies that the header is intact.