Of packets and their journeys.

Hardik Rajpal

November 28, 2023

Contents

Chapter 1

Misc

1.1 Big Fat Protocol Table

- Model Layer refers to the layer that the protocol is modelled to be a part of.

Name	Operates over	Model Layer	Function	Remarks
ARP	N/A	Link	Returns the MAC address corresponding to an IP address	Further discussion.
DHCP	UDP	Application	Used to obtain new IP address assignments for hosts.	Further discussion.
ICMP	IP	Network	Used by hosts and routers to communicate network level information	Futher discussion. Why is it on the network layer?
UDP	IP	Transport	Used for (process-) to-process delivery.	Read more. Source process is unnecessary or deduced by application layer (ex. DHCP $\implies$ delivery to source is based on link-layer broadcast and transaction id Xid)

1.2 Addresses

- Ethernet addresses are 48bits=6Bytes long.
- IPv4 addresses are 32bits=4Bytes long.

1.3 Data Units

Reference: baeldung. and Gateoverflow, I'm sorry, no offence.

1. Packet (aka IP Datagram): data unit used at the network layer.
    - Comprises of the network layer header (src/dst IP addresses) and the payload (transport layer or otherwise).
    - Basic unit of communication between a source and a destination in a network.
2. Fragment: Network layer unit. A packet can be split into multiple fragments, so as to ensure that each fragment is under the **Maximum Transmitted Unit** limit of the network.

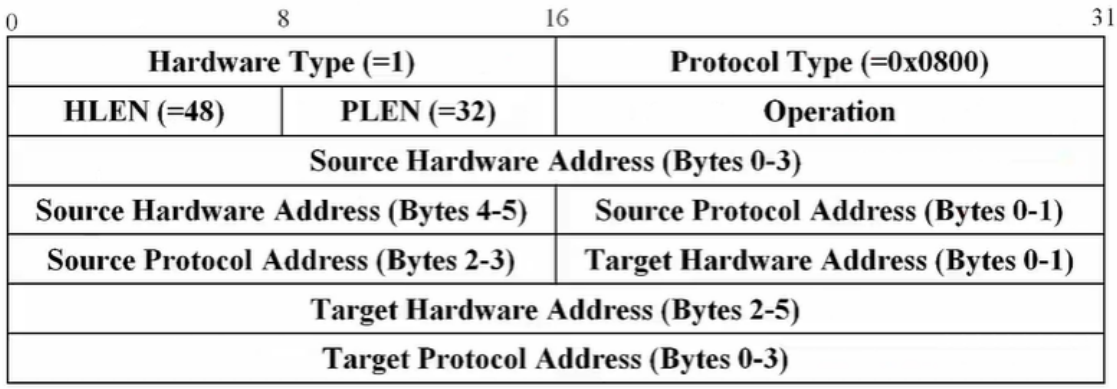
3. Frame: Link layer unit, refers to one network layer fragment (or entire packet), encapsulated with link layer headers and trailers.
    - There exist two types of framing: fixed-length and variable-length.
    - Fixed-length: the constant size eliminates the need for a delimiter.
    - Variable-length: it's essential to define the start and end of the frame.
4. (UDP) Datagram: Unit used at the transport layer by UDP only. It refers to the application layer payload and the UDP header encapsulating it.
5. Segment: A segment is a transport layer unit. It refers to a piece of a packet and the TCP header encapsulating it.
    - Segments increase the efficiency of network performance and improve security. (TODO: How?)
6. Message: Unit of data at application layer.

Chapter 2

Link Layer

2.1 Address Resolution Protocol

- Operates at link layer.
- Used to find the MAC address m corresponding to an IP address a.
- Broadcasts "who is a? tell srcIPAddress." message. Host with IP address a replies.
  - Ex: `who is 10.11.63.71?  tell 10.09.63.43.`
- Each intermediate host maintains a cache of IP to MAC translations and updates its cache on parsing ARP replies and requests.
- The requesting host saves the reply MAC in its cache.
- Entries in said cache timeout periodically.
- The packet format is:



  - Hardware type specifies what link level technology we're using. For ex, it's set to 1 for ethernet.
  - Protocol Type refers to higher level protocol. It's 0x0800 for IP.
  - HLEN specifies length of the MAC address in bits.
  - PLEN specifies length of the protocol address in bits. It's 32 for IP address in bits.
  - Operation can be: request or reply.
- The terms involved are:

  - Originator: Host that generates ARP request.
  - Target: Host replying to the ARP request. It updates its cache with srcIP, srcMAC.

- When a host has to forward a datagram that specifies a destination IP address (that is within the LAN),
  1. It first checks its ARP cache for a map from dstIP to MAC.
  2. If no entry is found, it broadcasts an ARP request.
  3. While the request and reply move through the LAN, intermediate hosts refresh their caches.
- Note: Intermediate hosts NEVER reply to ARP requests.

2.1.1 Gratuitous ARP

- Generated by a host to inform others of its IP and MAC address.
- According to this, gratuitous ARPs are request packets and not reply packets.
- Both IPdst and IPsrc are set to IP host, and src MAC is set to host MAC.
- dst MAC is the broadcast address: `ff:ff:ff:ff:ff:ff`
- No reply is expected.
- Gratuitous ARP is used to:
  1. Inform hosts of changes to my IP or MAC address.
  2. Inform hosts that a host is now available.
  3. Help rectify ARP entries.
  4. Report IP address conflicts (duplicate IP addresses).
  5. Inform learning bridges of the new location of the host, or the location of a new host.
- Note that since ARP is a stateless protocol, even replies that were never requested are parsed and processed and thus, can function as gratuitous ARPs.

Chapter 3

Network Layer

3.1 Obtaining IP Addresses

- Organizations get the address blocks from ISPs, from whom they purchase internet routers. Ex: Reliance, Tata, Sprint, AT&T.
- ISPs in turn get address blocks from Regional Internet Registries (RIR) which are controlled by Internet Corporation for Assigned Names and Numbers (ICANN).
- There are five RIRs:

- ISPs follow CIDR (covered before maybe) to assign blocks to organizations based on their size, and splits up blocks between organizations using different bits after the block's identifying bits.
- ISP routers advertise the block address that they've been given by an RIR to other routers for receiving all those packets.
- They don't advertise individual organize prefixes, but aggregate them all into the largest possible block.
- Given that the organization has an IP prefix, each host needs a unique IP address (hence a suffix).
- The address needs to be unique and location (subnet) dependent.
- Additionally, it needs to be reconfigurable, unlike ethernet addresses which are fixed by the manufacturer. (Though even those can be modified in some OSes, like Linux).
- Before any communication, the host needs:
  1. an IP address
  2. a mask that specifies what the network portion corresponds to

3. the default router's IP address.
4. the DNS server's IP address.

- Manual configuration is one option.
- Remote configuration is difficult and error prone.
- Enter DHCP:

3.2 DHCP

- Operates at application layer, above UDP.
- Used to obtain a new IP address assignment on joining a network or booting.
- DHCP server maintains a pool of available IP address which it leases out on requests.
- Leases expire periodically unless the hosts renew them.
- It's advantages include:
  - Easy of configuration, as it's automated.
  - Reuse of IP addresses, in the case where the total number of hosts is large but the number of hosts live at any time is very small.
  - Support for portability of hosts across the network, across different subnets.
- The packet format is:

<table>
<tr><td>Operation (1)</td><td>Htype (1)</td><td>Hlen (1)</td><td>Hops (1)</td></tr>
<tr><td colspan="4">Xid (4)</td></tr>
<tr><td colspan="2">Secs (2)</td><td colspan="2">Flags (2)</td></tr>
<tr><td colspan="4">Ciaadr (4)</td></tr>
<tr><td colspan="4">Yiaddr (4)</td></tr>
<tr><td colspan="4">Siaddr (4)</td></tr>
<tr><td colspan="4">Giaddr (4)</td></tr>
<tr><td colspan="4">Chaddr (16)</td></tr>
<tr><td colspan="4">Sname (64)</td></tr>
<tr><td colspan="4">File (128)</td></tr>
<tr><td colspan="4">Options (312)</td></tr>
</table>

  - Operation specifies whether it's a request or reply.
  - Htype: hardware type: ethernet etc.
  - Hlen: specifies length of hardware addresses. It's 6 (Bytes) (48 bits) for ethernet.
  - Hops: indicates number of relays traversed. It's set to zero by the host.
  - Xid: transaction ID to match requests and replies. All messages corresponding to one DHCP transaction have the same Xid.

- Secs: time elapsed since the client started the communication. A larger value means the server gives it more priority in the queue.
- Flags: only 1 of 16 bits is used, corresponding to the broadcast flag: whether or not the server broadcasts its replies.
  * It's set to 1 when the host broadcasts the discover message, so the server broadcasts its replies as the client doesn't have an IP address yet.
  * It's set to 0 when the host is renewing the lease, as it has an IP address that the server can send its replies on.
- Ciaddr: specifies the current IP address (if a previous assignment exists).
- Yiaddr: (Your IP addr) the IP address that the server offers the client.
- Siaddr: (Server IP addr) specified only if the client has to contact some other server as a part of the operation.
  * Additionally, the client uses this in the request message to inform all DHCP servers which may have sent it a request of the server whose offer it has chosen to proceed with.
  * As the messages are broadcast, those rejected DHCP servers note this request message and terminate the transaction.
- Giaddr: IP address of Relay agent if any.
- Chaddr: Holds the hardware address corresponding to the client.
- Sname: hostname of the server if it has one.
- File: boot file if the server wants to specify one, it's typically not used.
- Options: The client mentions the offered IP address here in the DHCP request packet. The client can also request additional information here, and the server can respond with additional information.

- Note that the same packet format is used by the client and the server.
- DHCP has its origins in BOOTP which was used earlier in booting of machines where configuration files were sent over the network.

3.2.1 Operation

1. Host broadcasts a DHCP discover message, with the broadcast (dstIP=`255.255.255.255`) restricted to the physical network.
   - Note that all IP-level broadcasts are link-level broadcasts. Routers don't allow letting such messages out into the internet.
   - All messages in this exchange have their dst IP address set to broadcast.
   - This helps in the case of multiple DHCP servers being around; so that all servers receive the request message meant for a chosen offer, to know they have to terminate the connection.
   - This allows the rejected servers to recycle the available IP addresses faster.
   - The client sets its src IP address to `0.0.0.0` if it's requesting an IP address for the first time.
   - In some cases of field values in the discover packets, unicast mayb be used. See this **if necessary.**
   - Otherwise, it uses its assigned IP address.
   - The server as expected puts its IP address in the src IP address field.
2. DHCP server responds with DHCP offer message, while other hosts ignore the discover message.
   - The offered IP address is derived from the subnet IP address, and possibly the hosts' MAC address.
3. Host receives the offer and requests the offered IP address: DHCP request message.

4. DHCP server confirms the assignment: DHCP ack. The server also informs the client of the expiration time.

- If the host requests for it, the DHCP server also passes:
    1. the subnet mask
    2. default router's IP address
    3. domain name
    4. DNS server info
- One DHCP server can be used over multiple subnets, if DHCP Relays are installed in each subnet that forward messages to (and from) DHCP server via a unicast link.
- Routers can act as DHCP relays.

3.2.2 Router Configuration

- Sysads manually configure the interface addresses on a router using network management tools.

3.3 ICMP

- Stands for Internet Control Message Protocol.
- Used by hosts and routers to communicate network-level information.
- Operates on top of the Internet Protocol. (IP).

Chapter 4

Transport Layer

4.1 Introduction

- Provides logical communication between two processes.
  - The devices running the processes are identified by their IP addresses.
  - On each device, the process (application) involved in a connection is identified by a port.
- Helps in multiplex, demultiplexing of packets for delivery to the right processes.
- Aka end-to-end protocols.
- Unit of data at transport layer is a segment.

4.1.1 Sockets

- Sockets function as the interface between processes and the transport layer.
- On a device (fixed IP address), a socket is identified by a port number.
- There are two types of sockets that provide two corresponding types of (de)multiplexing.
  1. Connectionless:
     - Used with UDP sockets.
     - A socket on a machine (fixed IP address) is identified by a port number.
     - Thus, once on the network, the socket is identified by a 2-tuple: (dst IP address, dst Port Number).
     - Packets with the tuple are directed to the same socket (thus, process), regardless of the source IP address and port.
     - It is the process' job to segregate the data from different clients.
  2. Connection-oriented:
     - Used with TCP sockets.
     - A socket on a machine (fixed dst IP address) is identified by src IP, src Port and dst Port.
     - Thus, once on the network, the socket is identified by a 4-tuple: (dst IP, dst Port, src IP, src Port).
     - The server creates a new socket for each client that connects to it.
       - In the program, the socket is identified by a file descriptor or a reference to some object wrapping around it.
       - In the kernel, the fd is associated with the 4-tuple mentioned earlier.

    * When a packet arrives with the four tuple, the kernel looks up the tuple's corresponding socket in a table, and sends the payload to that socket.
    * The program access this by the read functions in the object wrapping around the socket's fd.
    * Note: I am not sure how the lookup is done as this is a transport layer functionality which requires access to the network layer header (src IP).

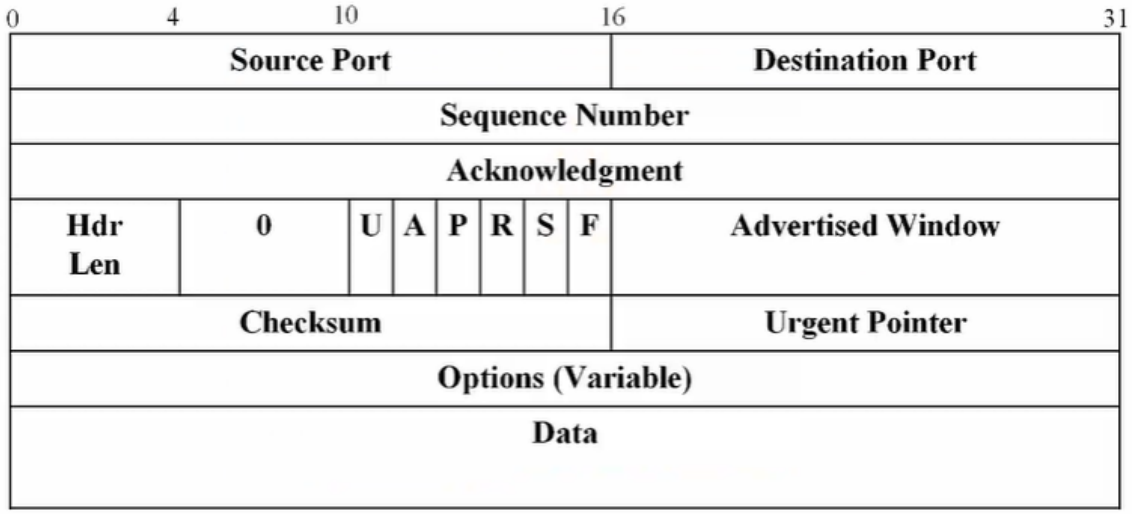
4.1.2 Port Numbers

- Ports are a logical construct, an abstraction, a figment of the programmers' inter-subjective imagination.
- Physical ports refer to ports on a switch, not these ports.
- 16bit numbers (0 to 2^16 - 1) to identify processes using the network on a machine.
- One process can listen to multiple ports, but one port can only be used by one process.
- The point of the port abstraction is to allow for multiple processes to share the network without conflicts.
- Some OS provide extensions to the above, allowing for port reuse. See this.
- Note that one computer may have two (active) IP addresses, by having two network cards, and two processes may use the ports with the same number on each IP address.

4.2 Transmission Control Protocol

- Connection-oriented byte stream protocol.
- Provides:
  - (De)Multiplexing.
  - Reliable process to process data transfer.
  - Full duplex connection.
  - Flow control: receiver has control over sending rate.
  - Congestion control: stops sender from overwhelming routers and network while using these resources sufficiently well.
- Key idea: sliding window protocol.
- Key challenges that TCP addresses:
  1. Connection management: explicit connection establishment between two processes and tear-down.
  2. Round Trip Time: Adaptive timeout mechanism to correspond to varying congestion rates and route availability.
  3. Reordering: Arbitrary delays for each packet implies they need to be reordered at the destination.
  4. Flow Control: Mechanism to ensure destination host isn't overwhelmed with the data transfer rate.
  5. Congestion Control: Mechanism to ensure network (possibly changing) isn't overwhelmed with the data transfer rate.

4.2.1 Segment Format



Sequence Number and Ack

- Each byte is assigned a sequence number.
- The Sequence number field in the header refers to that of the first byte.
- The ack field acknowledges the data flow from the other direction. It carries the sequence number of the next byte the host is expecting.
- Unless specified otherwise, the ack is cumulative.
- ACK=85 $\implies$ all bytes with sequence number ≤ 84 have been received, and 85 onwards are expected.

Flags

1. U flag
   - Urgent flag to indicate that the segment contains urgent data.
   - Used along with urgent pointer, which indicates the length of the urgent segment, thus indicating where the non-urgent data begins.
   - Assumes that urgent data (if any) precedes non-urgent data in the application payload.
2. A flag:
   - The Ack flag is set if the acknowledgement field is valid.
   - It's used to separate single direction data flow connection packets (A=0) from two-way flows (A=1).
3. P flag:
   - The push flag indicates whether the receiver should pass data to higher layer immediately.
4. R flag: Reset flag, used to abort connection.
5. S/F flags: Syn and fin flags are used during connection establishment and termination.

Advertised Window

Used by each host for flow control, to control the sending rate at the other.

Checksum

- Similar to UDP checksem.
- Compulsory in both IPv4 and IPv6.
- Calculated over TCP header, data and pseudoheader: src, dst IP addresses, IP protocol number and total length of TCP segment.

Options

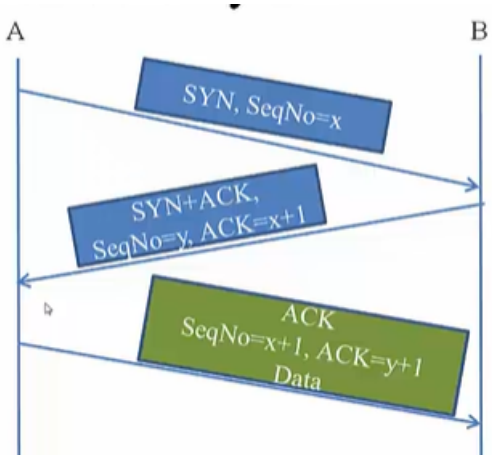
- Used at connection setup to negotiate certain parameters.
- Also used later to carry certain info for implementing specific functionality.
- Ex:
  - Can negotiate maximum segment size (based on MTU supported by network between two hosts).
  - Can perform window scaling.
  - Permits use of selective acks: both to indicate device support for selacks and to carry actual selack info.
  - Permits use of alternative checksums.

4.2.2 Connection Management

- Being a connection oriented protocol, all TCP connections start with a connection establishment phase. This phase helps in:
  - Exchanging and initiating state variables: MaxSegSize, ISNs, ACK types.
  - Allocating resources for connection (send and receive buffer spaces).
  - The buffer space allocation depends on the OS, ranges from 4KB to MB.

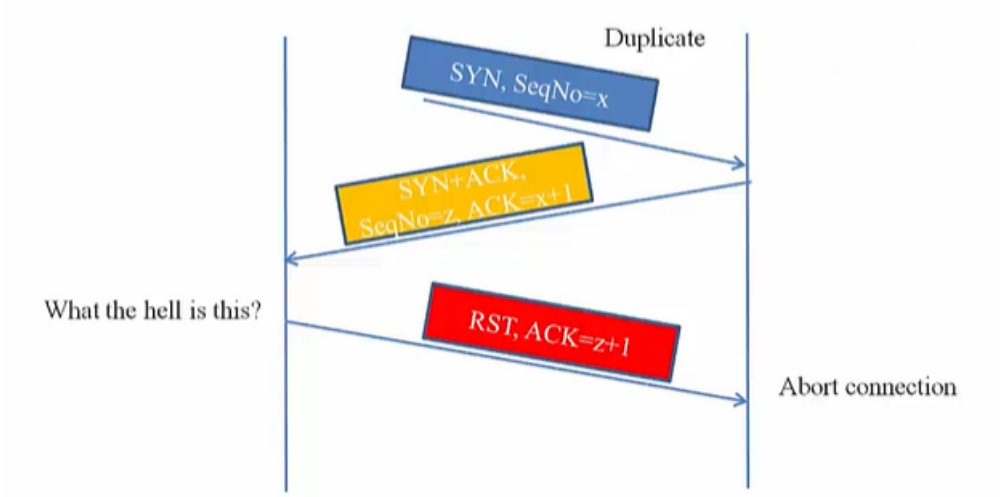
Connection Setup

- A simple request-grant-data model is susceptible to errors from packet duplication and timeout-induced replays.
- The three-way handshake helps counter this.

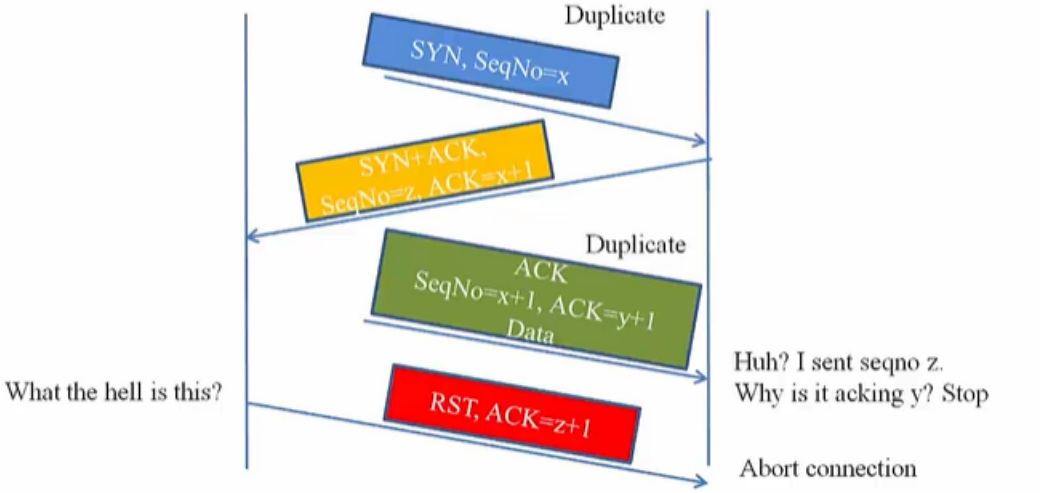


- SYN $\implies$ synchronization.

- Case 1: Only SYN is duplicated.
  - On receiving the response to the duplicated SYN (SN=z, ACK=x+1) (which was in the network and just got delivered), the server A sends a RST packet as it has not initiated any connection with B.



- Case 2: Both SYN and SYN+ACK are duplicated.
  - RST packets are essential to close such erroneous transfers.

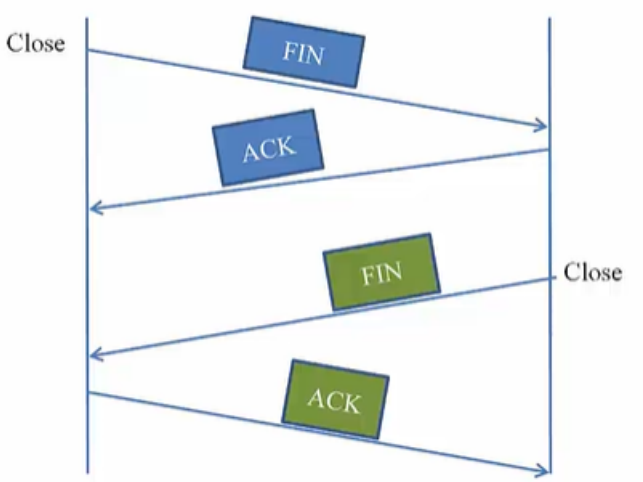


Initial Sequence Number

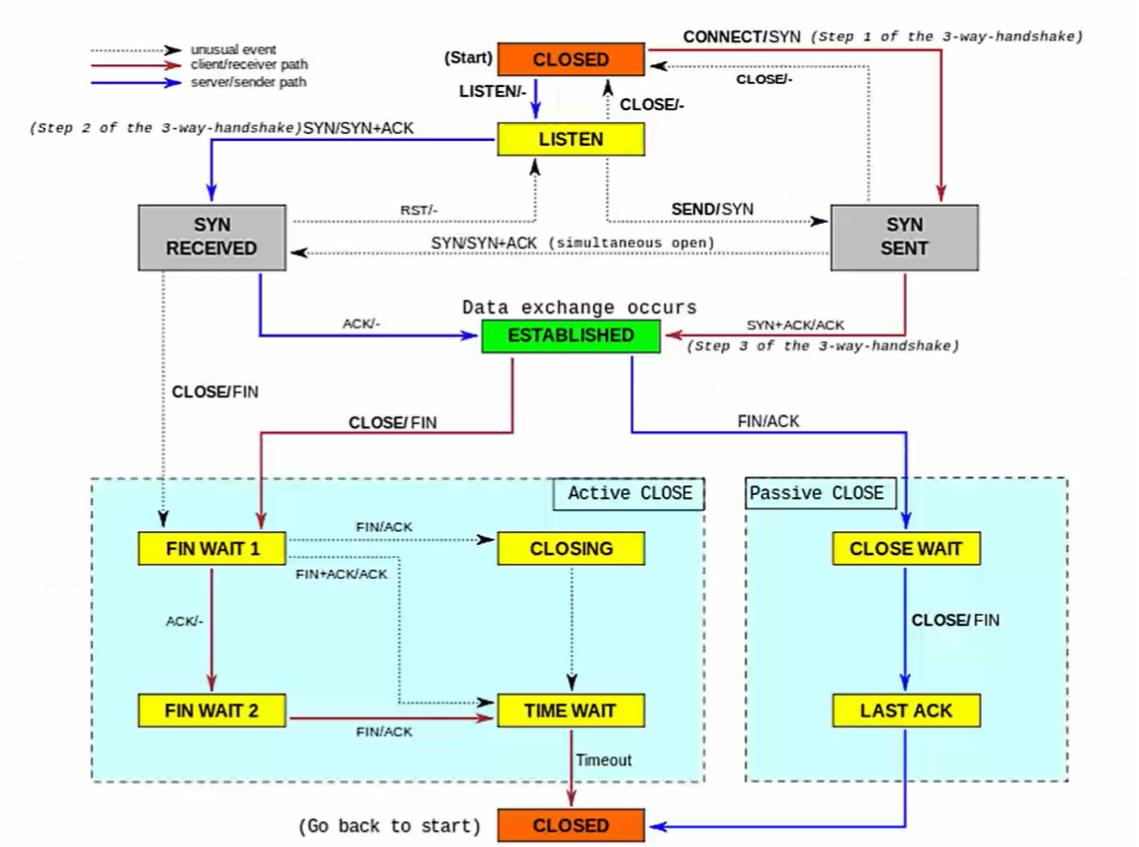
- We don't start with zero so segments from different connections don't get mixed up.
- Additionally, there's a security risk with predictable ISNs.
- Current implementation uses random ISN.
- Older implementations considered using the system clock to choose ISNs, for connection segregation, but this still suffered from having predictable ISNs.

Connection termination

- Asymmetric release (just hang-up) leads to data loss.
- Symmetric release:
  - Treat connection as two separate unidirectional connections.
  - Each side should be released separately.



  - After sending a FIN, the host doesn't send anymore data. It continues to ACK any data it receives.



- Note that the client is in the time-wait state before it closes by timeout.
  - It waits here for 2*(Max Seg Lifetime).
  - This mechanism is expected to clear out older packets in the network and prevent them from interfering with the new connection.
  - In particular, if a connection with a 4-tuple is closed and a new connection with the same 4-tuple is started shortly after (as can happen with HTTP requests), packets from the previous connection (which are forwarded to the socket identified by the 4-tuple being used by the new connection) can result is a RST packet and the consequent connection termination.
  - With timewait, said packets are cleared out.
  - In the time-wait state, attempts to open a connection with the same 4-tuple results in a "bind failed" error.

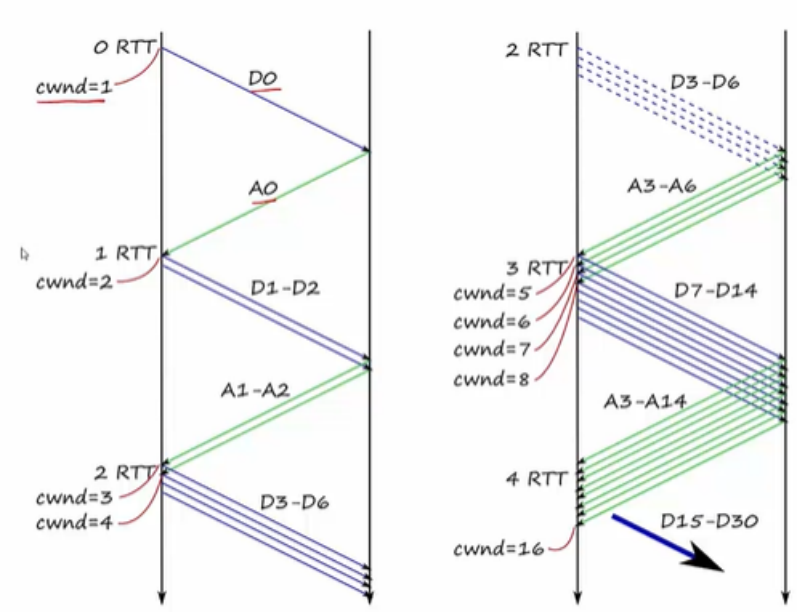
4.2.3 Congestion Control

- $\text{Throughput}|_{SWP} = \frac{W \times MSS}{RTT}$,where
  - W is the window size in number of segments.
  - MSS is the maximum segment size
  - RTT is the round trip time.
  - Here the transmission time is ignored, consider a large W.
  - Considering the transmission time Tx, RTT is replaced by RTT + (W-1)×Tx. This accounts for the NIC only being able to send one packet over a period of Tx.
- The need for congestion control:
  - Many flows (TCP or UDP) pass through a router $\implies$ number varies with time.
  - Link capacities also vary across routers.
  - Throughput achieved by a given flow depends on many parameters.
  - We need to estimate W such that each flow gets a fair share of the bandwidth.
  - A fixed low W results in underutilization while a large W causes congestion $\implies$ W varies over time.
  - Congestion control refers to preventing sources from sending too much data too fast and thereby congesting the network.
- The capacity of the network in SWP is defined as the bandwidth-delay product: BW × RTT.
- Removal of data from the pipe (the sliding window) is done on receiving an ACK for a given packet, and subsequently, a new packet is sent in the window. This feature of TCP is said to make it self-clocking.
- We view the network as a pipe of dynamic capacity given by the BW-delay product. It is used to derive a congestion window variable CW.
- TCP uses self-clocking to pump packets into the network.

Getting to Equilibrium

- A large initial value of CW (by every client in the network for each TCP connection) will push the network into congestion.
- A value of CW that is "just right" means that bursty transmissions can lead to losses. This is due to buffer overflow at bottleneck routers (high input BW, low output BW) in the presence of other users.
- Hence, we start with a conservative, low, value of CW and slowly fill up the pipe: "Slow start."

- Slow start:
  - Use a variable called CW to capture the maximum allowed number of outstanding data in the network. (sent, unacked at sender).
  - Set CW to 1 initially.
  - On each ack for new data, increase CW by 1 on every ack. (Linear increment with ACKs, but exponential in RTT.)
  - CW = $2^{\frac{t}{RTT}}$



  - Consecutive ACKs are separated by the Tx time at the bottleneck link in the network, and for each ACK, we send two packets. $\implies$ At most, two packets every Tx time of the bottleneck bandwidth.
  - Hence, we send data at at most twice the bottleneck link on the path $\implies$ bound to overestimate the capacity at some time in the future.

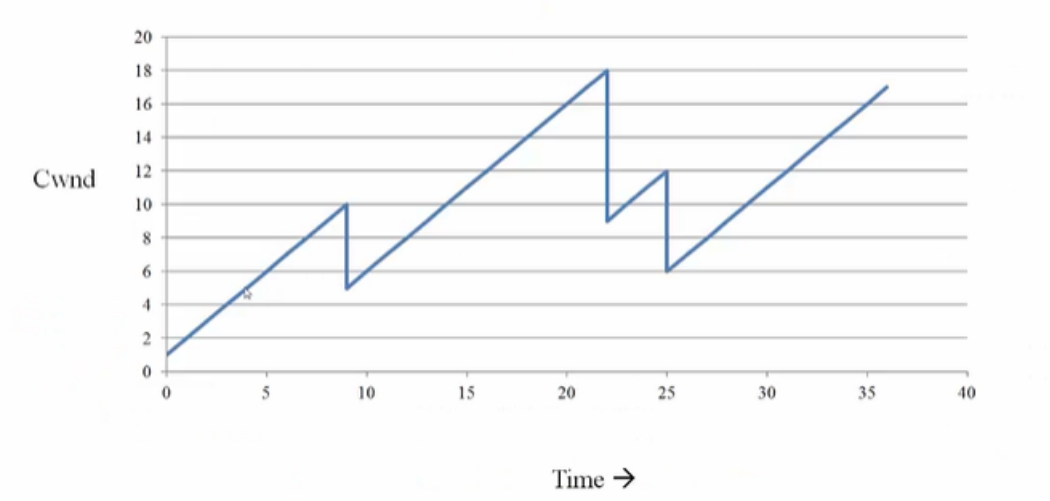
Conservation at Equilibrium

- Conservation: Don't put a packet unless a packet is removed (ACKed at sender).
- This is particularly important if the network is in a congested state.
- At slow start, two packets are added for each removed, and thus this is violated before attaining equilibrium.
- It is also violated on timeouts (perceived losses). Thus,it is crucial to estimate RTT properly, so that delayed packets are not interpreted as lost.
- RTT increases as the network gets more congested.
- Original Algorithm for RTT estimation:
  - Sample RTT = RTT for the SN whose ACK has just been received.
  - Est. RTT = a $\times$ Est. RTT + (1-a)$\times$ Sample RTT.
  - Lower value of a means Est. RTT is heavily influenced by temporary fluctuations.
  - Higher value of a means Est. RTT is not quick enough to adapt to real changes.

  - Timeout = 2 × Est. RTT.
  - a ∈ (0.8,0.9) $\implies$ works fine with low variance of RTT, but fails with a congested network, where the variance of RTT is large.
  - However, this algorithm led to a network collapse.
- Jacobson/Karels Algorithm:
  - It takes into account variance of RTTs.
  - Trust Est. RTT if variance is low, if it is large, the timeout shouldn't depend heavily on Est. RTT.
  - Difference = Sample RTT. - Est. RTT
  - Est. RTT = Est. RTT + (d × Difference)
  - Deviation = Deviation + (d×(—Difference—-Deviatoin)), where d $\tilde{0}$.125
  - Timeout = u×Est.RTT + q×Deviation, u=1, q=4.
  - Exponential timeout backoff to control spacing between retransmits (in the absence of new Sample RTTs).
  - Timeout doubles for every timeout.
  - With stable RTTs, we react to losses quicker, but in cases of high deviation, the timeout is larger and allows for larger delays.

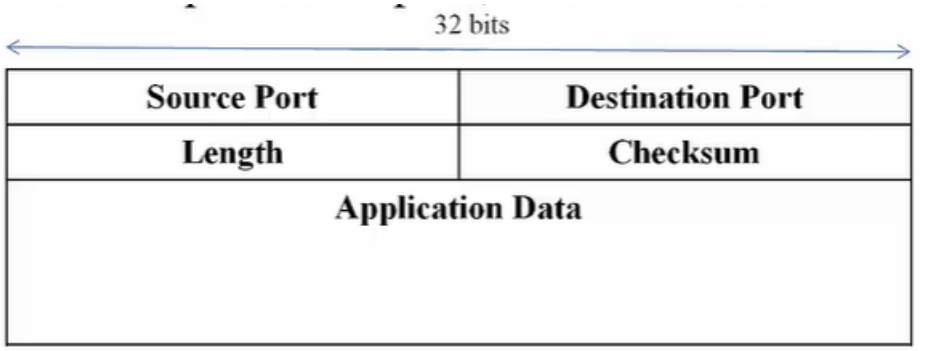
Congestion Avoidance: AIMD

- Control theory says that an unstabel system can be stabilized by adding exponential damping.
- A network system subject to random load shocks and prone to congestive collapse can be stabilized by adding exponential damping to its primary excitation (traffic sources).
- Essentially, we want to give the system enough time to recover before we send more packets.
- Adapting to the path:
  - Estimation method for W may overestimate or underestimate W; we need to correct this.
  - The available BW also changes over time; need to adapt to this.
  - We need a feedback mechanism from the network to realize that the estimate is wrong.
- On overestimation,
  - Overestimation leads to congestion.
  - Note that routers don't send any "source quench" messages to inform the host of congestion.
  - If losses are due to congestion and timers are working correctly, timeouts indicate congestion. (At least on wired systems, bit error rate is very low.)
  - Multiplicative decrease yields better stability: $W_i = dW_{i-1}$, d¡1, typically 0.5.
- On underestimation,
  - Underestimation leads to lower utilization.
  - Exponential (multiplicative) increase leads to instability and overestimation is inevitable.
  - We thus prefer additive increase: $W_i = W_{i-1} + u$, u ¡¡ W_{max}, typically 1.
  - $\implies$ we increase W by 1 segment every RTT, NOT on every ACK (which is what slow start does).
  - This is equivalent to incrementing by 1/W on every ACK.
- This algorithm results in a saw-tooth graph:



4.3 User Datagram Protocol

- Provides (de)multiplexing over best effort network layer service.
- UDP segments can be lost, duplicated, delivered out of order.
- Connectionless $\implies$ no handshaking phase between client and server.
- Packet delivery to socket identified by (dst IP, dst Port)
- Each DHCP segment is handled independently of the other.
- Pros:
  - Not having a handshake phase, it has zero connection establishment overhead.
  - Small segment header: less data overhead per packet.
  - Server can support multiple clients due to the simplicity of the protocol.
  - ? No congestion control: rate is independent of congestion in the network.
  - No retransmission delay: useful for realtime applications like VoIP, games.
- Cons:
  - No delivery guarrantees $\implies$ applications have to implement these capabilities on their own.
- Protocols using UDP:
  - DHCP
  - RIP
  - DNS
  - SNMP
- UDP datagram format:



- Length refers to the total length of the segment in bytes = 8B header + app data.
- Checksum is calculated over UDP header, app data and pseudo header from IP header.
  - Pseudo header includes the protocol version number, source and destination IP addresses.
  - Checksum on pseudo header verifies correct delivery.
  - Checksum is optional in IPv4 but compulsory in IPv6 as it doesn't have a checksum in its header.