The IP, TCP, UDP protocols

R.Venkatarathinam ,professor
Sri Balaji Chockalingam Engineering College , Arni

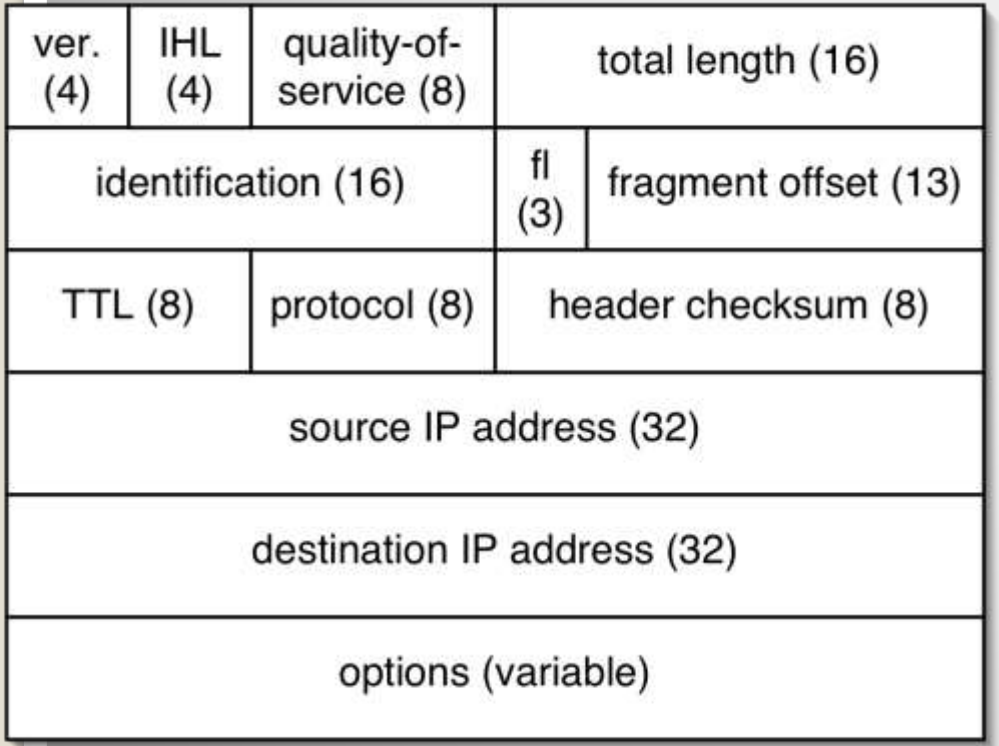
IP protocol

- Defines a uniform mechanism to access resources between internets
  - Enables networking across networks that are not connected at level 2 (data-link)., also written 128.220.23.0/24
  - Class B network: 192.168.x.x., also written 192.168.0.0/16
  - Class A network: 10.x.x.x, or 10.0.
  - Defines IP addresses and how to route network packets to a destination address.
- IP v.4, addresses: 4 octets, organized hierarchically
  - Single host: 128.220.23.4 or 192.168.33.1
  - Class C network: 128.220.23.x0.0/8

IP v.6

- 16 octet addresses, also hierarchical
- Represented by eight 4-digit hexadecimal values (one string of 0’s can be omitted)
  - Single host: 1080:0:0:0:0:800:0:417A *or* 1080::800:0:417A
- Internet routing is performed only on the 64 left bits (the remaining is for internal routing to hosts)
- Blocks of addresses denoted using the / notation.
  - 12AB::CD30:0:0:0:0/60 indicates a 68-bit wide space,
from 12AB:0:0:CD30:0:0:0:0
to 12AB:0:0:CD3F:FFFF:FFFF:FFFF:FFFF
  - (all addresses starting with prefix 12AB:0:0:CD3)

IP packet



ver. (4) | IHL (4) | quality-of-service (8) | total length (16)

identification (16) | fl (3) | fragment offset (13)

TTL (8) | protocol (8) | header checksum (8)

source IP address (32)

destination IP address (32)

options (variable)

IP v.4 header:
Only valid versions are 4 and 6.

IHL = internet header length, in 32-bit words

total length = length of header and data,
in 32-bit words

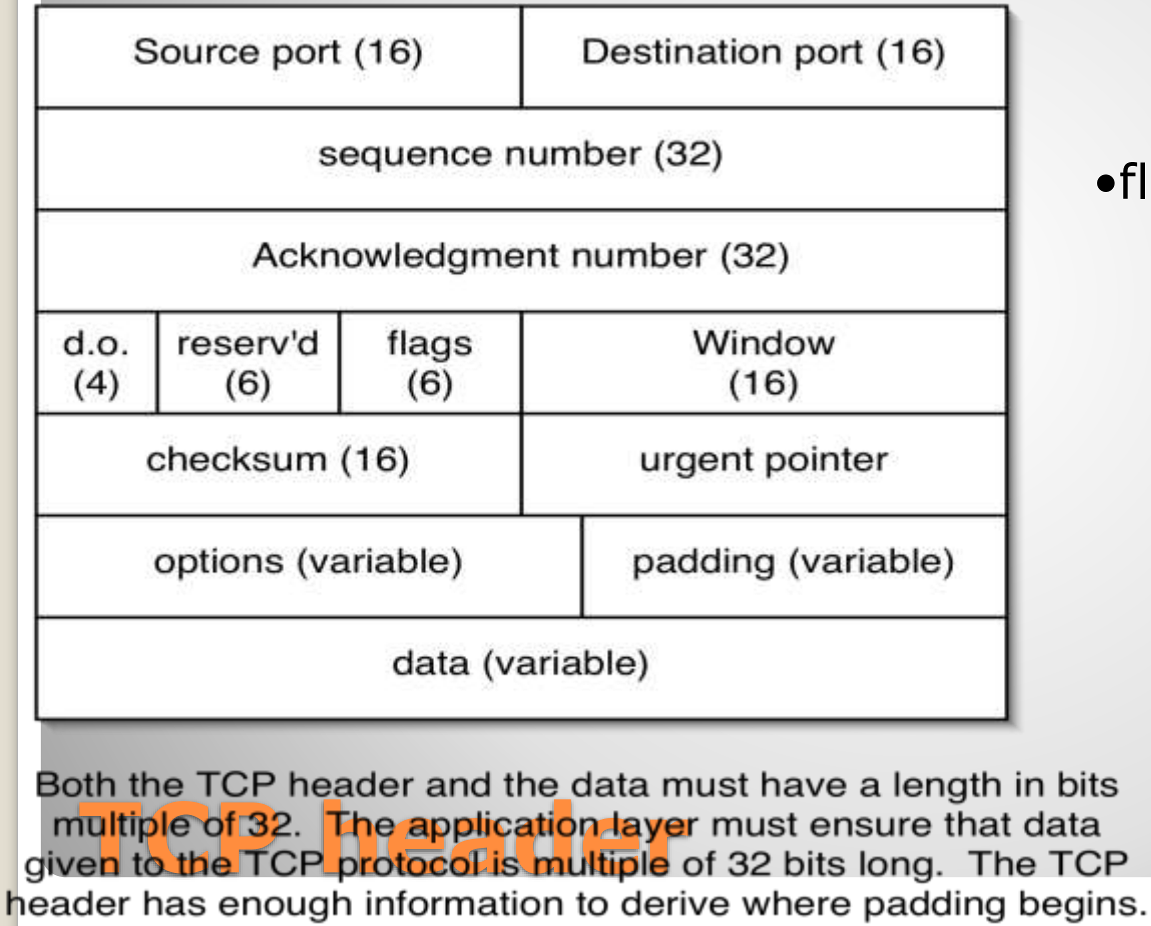
flags:
first bit reserved (0),
2nd bit (may fragment),
3rd bit (is last fragment)

TTL: must be decremented by at least 1
at each router, packet must be
discarded if TTL= 0

TCP/UDP

- Layer 4 (transport layer) protocols, run over IP
  - TCP and UDP packets are encapsulated into IP packets
- Use their own control information, stored in packet headers
  - Port numbers (indicate consuming program in the destination host)
- TCP is connection-oriented, and provides for reliable, order-preserving transmission of data
- UDP is not connection-oriented, does not guarantee data arrival, or proper ordering of arriving data

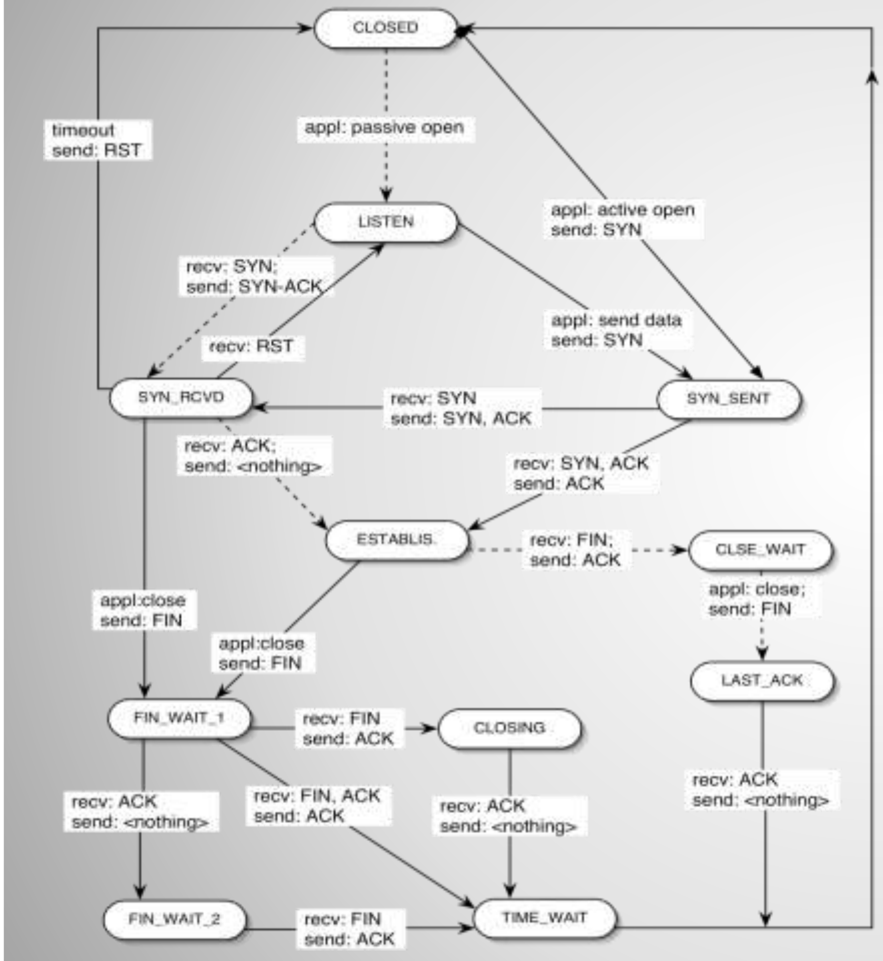
TCP header



<table>
<tr><td colspan="3">Source port (16)</td><td colspan="2">Destination port (16)</td></tr>
<tr><td colspan="5">sequence number (32)</td></tr>
<tr><td colspan="5">Acknowledgment number (32)</td></tr>
<tr><td>d.o.
(4)</td><td>reserv'd
(6)</td><td>flags
(6)</td><td colspan="2">Window
(16)</td></tr>
<tr><td colspan="3">checksum (16)</td><td colspan="2">urgent pointer</td></tr>
<tr><td colspan="4">options (variable)</td><td>padding (variable)</td></tr>
<tr><td colspan="5">data (variable)</td></tr>
</table>

Both the TCP header and the data must have a length in bits multiple of 32. The application layer must ensure that data given to the TCP protocol is multiple of 32 bits long. The TCP header has enough information to derive where padding begins.

- flags:
  - URG (urgent),
  - ACK (acknowledgment),
  - PSH (push function),
  - RST (close the connection),
  - SYN (synchronize sequence numbers),
  - FIN (end of data from sender)



TCP state diagram

TCP new connection:

- 3-way handshake, by TCP packets with SYN, SYN & ACK, and ACK flags set.

TCP close connection:

- Via RST, without ACK
- Via FIN, ACK, FIN, ACK (4 way teardown)

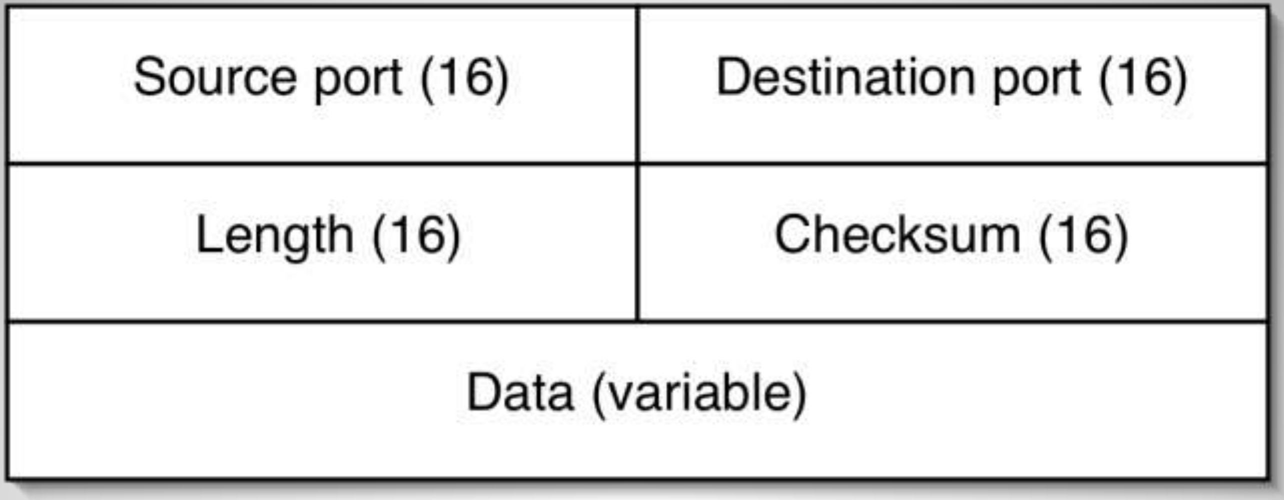
TCP SYN-flood attack

- An adversarial client can spoof many IP addresses and send large numbers of SYN-packets, requesting new TCP connections to the server.
- The server responds with SYN-ACK packets, and keeps record of the requested connections (in state SYN-received), waiting on each a period (timeout) for an ACK-packet that will never arrive.
- If the adversary's rate of requests is large enough, the server's resources (memory buffers) for handling pending connections will fill up quickly (well before timeout).
- The server will no longer be able to handle legitimate service requests. Moreover, it can result in OS instability and crashing

- Enable servers to continue to accept and serve legitimate connections even after its resources to handle pending connections (in SYN-received state) have been exhausted.
- Requires no modification of clients, can be unilaterally adopted by servers without modifying the underlying behavior of the IP protocol.
- Change the generation of server sequence numbers.
  - Of the 32-bit sequence counter, use the top 5 bits for a counter t than increases every 64 seconds.
- Use the bottom 24 bits to encode a secret function of the server and client IP addresses and port numbers, and of t.
- When out of resources, server just sends SYN-ACK as if ok. When the ACK is received, re-compute the secret function and verify if the ACK is a possible reply to an (unrecorded) SYN-ACK packet. Use the information in the ACK to re-create the connection table.

TCP SYN Cookies

UDP



Source port (16)	Destination port (16)
Length (16)	Checksum (16)
Data (variable)	

UDP packet: No state information for the communication session. UDP is a stateless protocol, without re-transmission of loss data or protection against data-reordering

Thank you