Introduction to TCP/IP networking

Source: Ganesh Sittampalam

TCP/IP protocol family

- IP : Internet Protocol
  - UDP : User Datagram Protocol
    - RTP, traceroute
  - TCP : Transmission Control Protocol
    - HTTP, FTP, ssh

What is an internet?

- A set of *inter*connected *net*works
- The **I**nternet is the most famous example

- Networks can be completely different
  - Ethernet, ATM, modem, …
  - (TCP/)IP is what links them

What is an internet? (cont)

- *Routers* (nodes) are devices on multiple networks that pass traffic between them
- Individual networks pass traffic from one router or endpoint to another
- TCP/IP hides the details as much as possible

ISO/OSI Network Model (Don't need to know this)

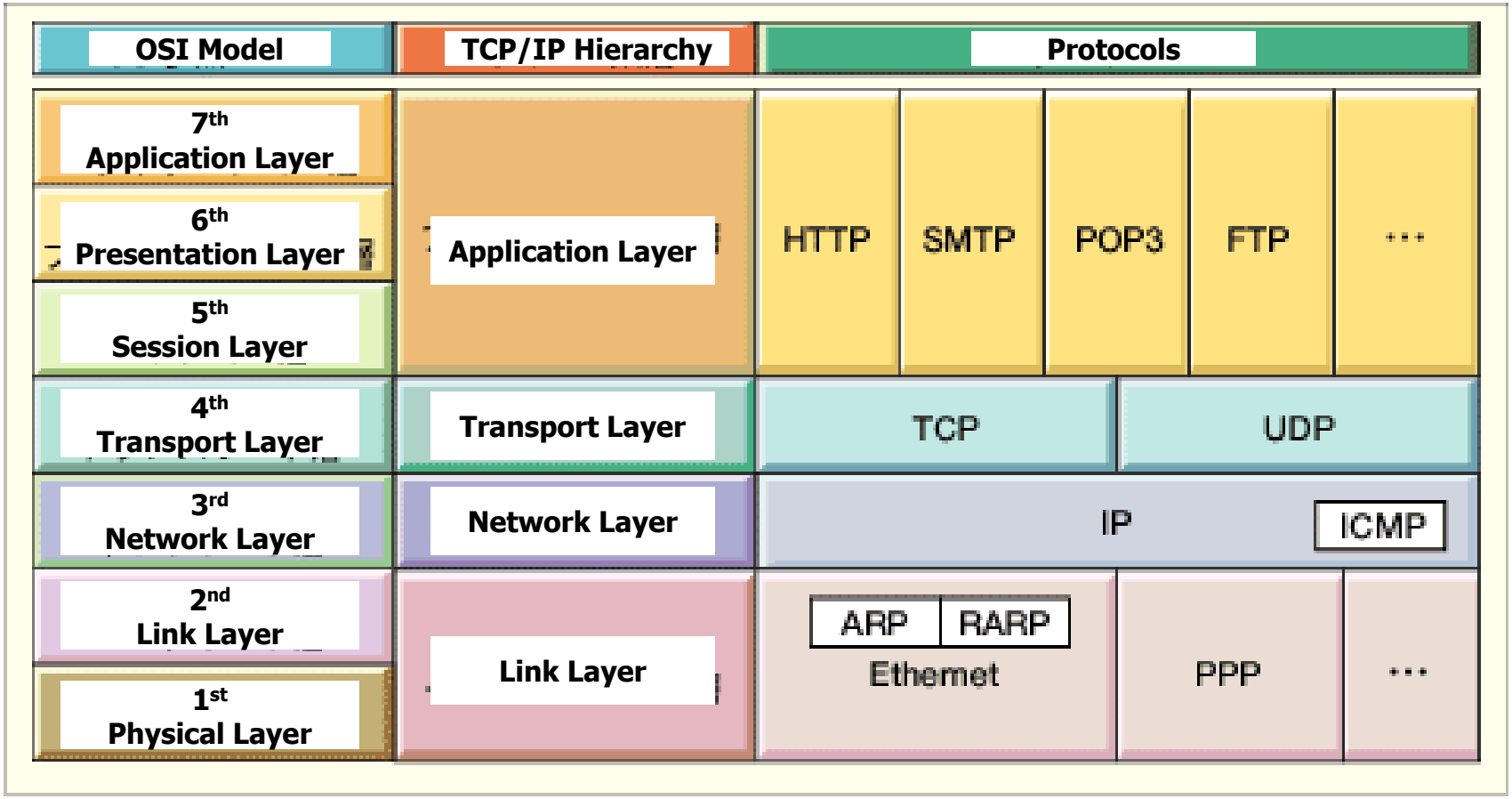
- Seven network "layers"
  - Layer 1 : Physical – cables
  - Layer 2 : Data Link – ethernet
  - Layer 3 : Network – IP
  - Layer 4 : Transport – TCP/UDP
  - Layer 5 : Session
  - Layer 6 : Presentation
  - Layer 7 : Application

You don't need to know the layers just the idea that it is layered

TCP/IP Network Model

- Different view – 4 layers
  - Layer 1 : Link (we did not look at details)
  - Layer 2 : Network
  - Layer 3 : Transport
  - Layer 4 : Application

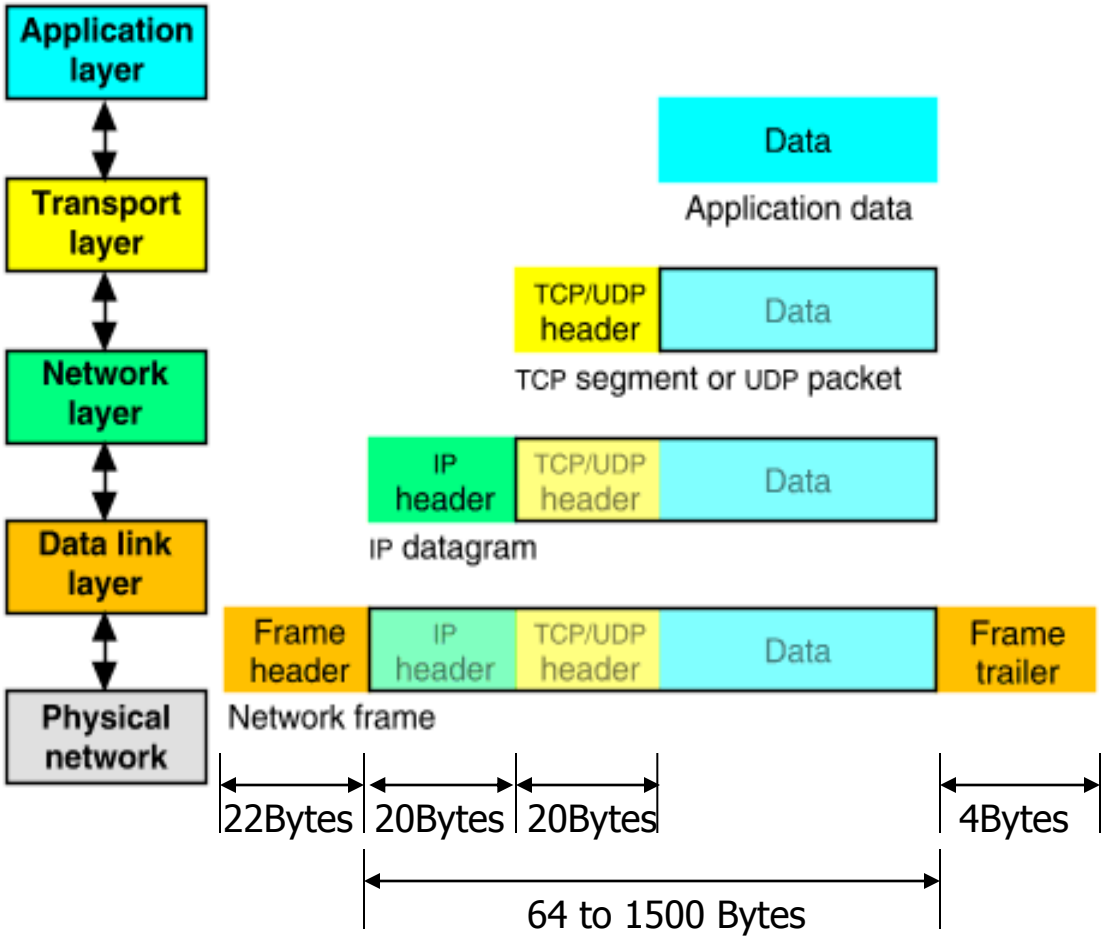
OSI: Open Systems Interconnect



Link Layer : includes device driver and network interface card
Network Layer : handles the movement of packets, i.e. Routing
Transport Layer : provides a reliable flow of data between two hosts
Application Layer : handles the details of the particular application

Packet Encapsulation

- The data is sent down the protocol stack
- Each layer adds to the data by prepending headers



IP

- Responsible for end to end transmission
- Sends data in individual packets
- Maximum size of packet is determined by the networks
  - Fragmented if too large
- Unreliable
  - Packets might be lost, corrupted, duplicated, delivered out of order

IP addresses

- 4 bytes
  - e.g. 163.1.125.98
  - Each device normally gets one (or more)
  - In theory there are about 4 billion available
- But…

Routing

- How does a device know where to send a packet?
  - All devices need to know what IP addresses are on directly attached networks
  - If the destination is on a local network, send it directly there

Routing (cont)

- If the destination address isn't local
  - Most non-router devices just send everything to a single local router
  - Routers need to know which network corresponds to each possible IP address

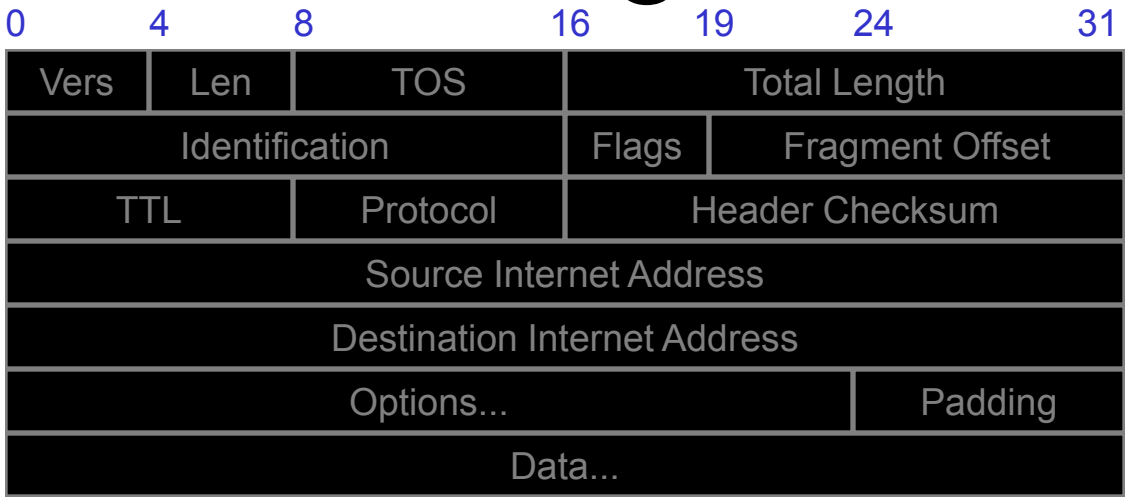
Allocation of addresses

- Controlled centrally by ICANN
  - Fairly strict rules on further delegation to avoid wastage
    - Have to demonstrate actual need for them
- Organizations that got in early have bigger allocations than they really need

IP packets

- Source and destination addresses
- Protocol number
  - 1 = ICMP, 6 = TCP, 17 = UDP
- Various options
  - e.g. to control fragmentation
- Time to live (TTL)
  - Prevent routing loops

IP Datagram

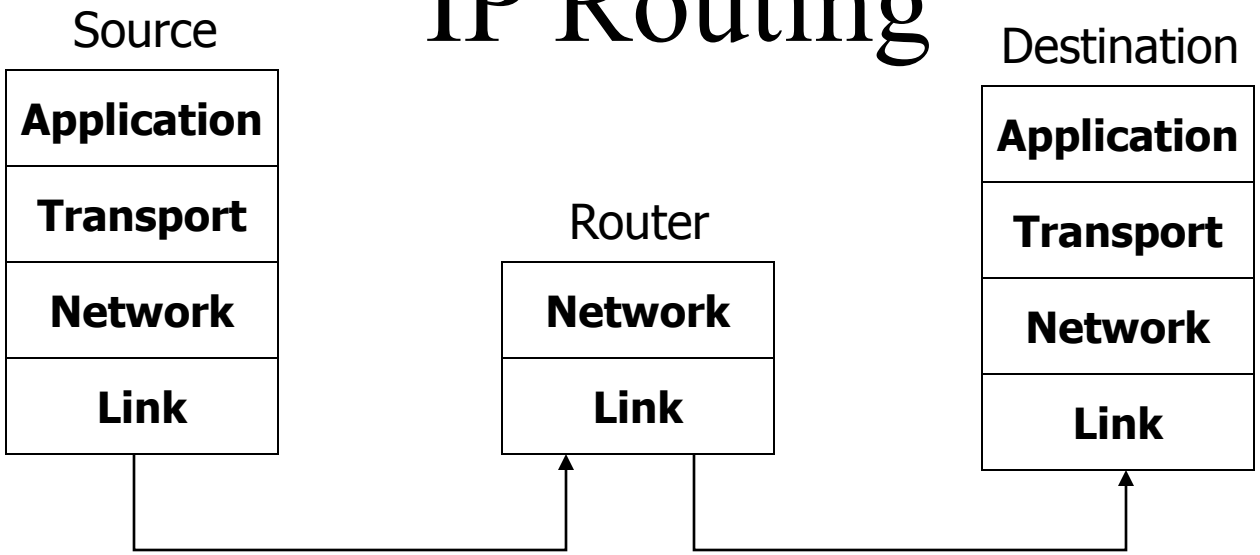


Field	Purpose
Vers	IP version number
Len	Length of IP header (4 octet units)
TOS	Type of Service
T. Length	Length of entire datagram (octets)
Ident.	IP datagram ID (for frag/reassembly)
Flags	Don't/More fragments
Frag Off	Fragment Offset

Field	Purpose
TTL	Time To Live - Max # of hops
Protocol	Higher level protocol (1=ICMP, 6=TCP, 17=UDP)
Checksum	Checksum for the IP header
Source IA	Originator's Internet Address
Dest. IA	Final Destination Internet Address
Options	Source route, time stamp, etc.
Data...	Higher level protocol data

We only looked at the IP addresses, TTL and protocol #

IP Routing

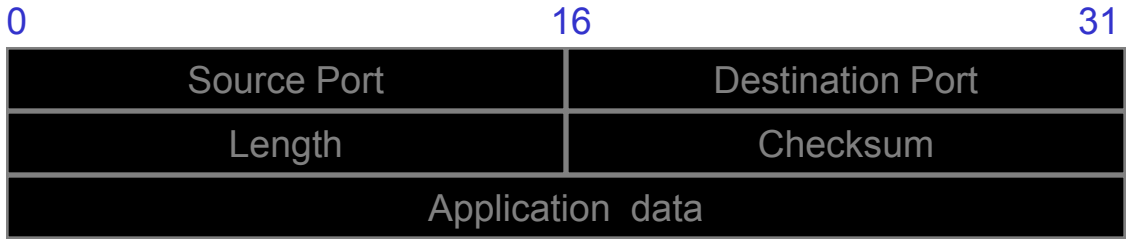


- Routing Table
  - Destination IP address
  - IP address of a next-hop router
  - Flags
  - Network interface specification

UDP

- Thin layer on top of IP
- Adds packet length + checksum
  - Guard against corrupted packets
- Also source and destination *ports*
  - Ports are used to associate a packet with a specific application at each end
- Still unreliable:
  - Duplication, loss, out-of-orderness possible

UDP datagram



Field	Purpose
Source Port	16-bit port number identifying originating application
Destination Port	16-bit port number identifying destination application
Length	Length of UDP datagram (UDP header + data)
Checksum	Checksum of IP pseudo header, UDP header, and data

Typical applications of UDP

- Where packet loss etc is better handled by the application than the network stack
- Where the overhead of setting up a connection isn't wanted

- VOIP
- NFS – Network File System
- Most games

TCP

- Reliable, *full-duplex*, *connection-oriented*, *stream* delivery
  - Interface presented to the application doesn't require data in individual packets
  - Data is guaranteed to arrive, and in the correct order without duplications
    - Or the connection will be dropped
  - Imposes significant overheads

Applications of TCP

- Most things!
  - HTTP, FTP, …
- Saves the application a lot of work, so used unless there's a good reason not to

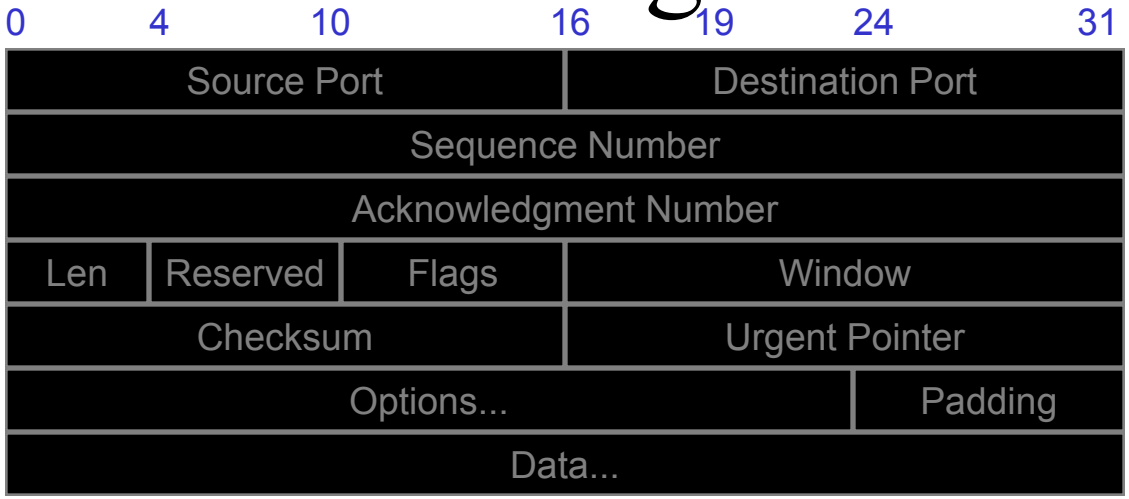
TCP implementation

- Connections are established using a *three-way handshake*
- Data is divided up into packets by the operating system
- Packets are numbered, and received packets are acknowledged
- Connections are explicitly closed
  - (or may abnormally terminate)

TCP Packets

- Source + destination ports
- Sequence number (used to order packets)
- Acknowledgement number (used to verify packets are received)

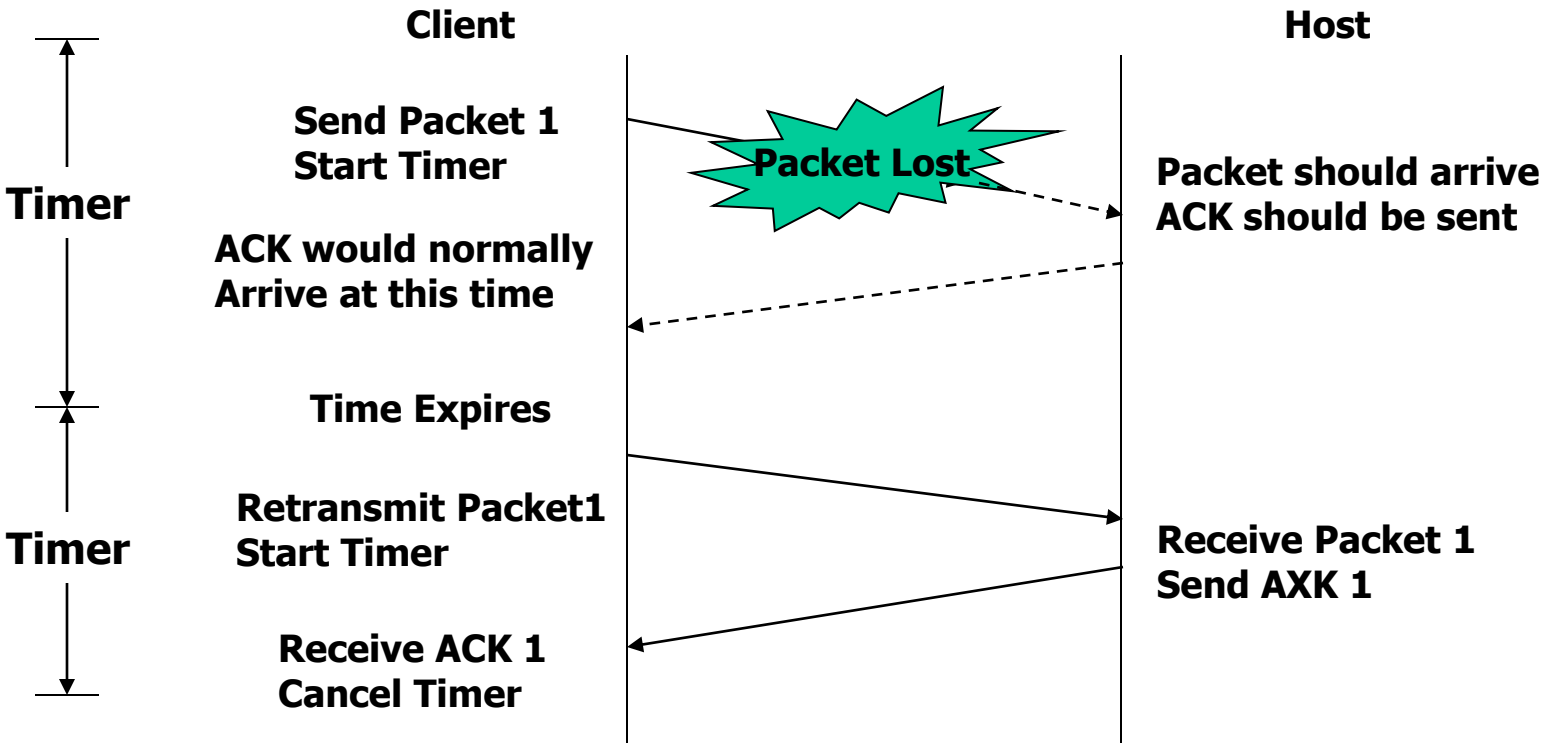
TCP Segment



Field	Purpose
Source Port	Identifies originating application
Destination Port	Identifies destination application
Sequence Number	Sequence number of first octet in the segment
Acknowledgment #	Sequence number of the next expected octet (if ACK flag set)
Len	Length of TCP header in 4 octet units
Flags	TCP flags: SYN, FIN, RST, PSH, ACK, URG
Window	Number of octets from ACK that sender will accept
Checksum	Checksum of IP pseudo-header + TCP header + data
Urgent Pointer	Pointer to end of “urgent data”
Options	Special TCP options such as MSS and Window Scale

You just need to know port numbers, seq and ack are added

TCP : Data transfer



IPv6

- 128 bit addresses
  - Make it feasible to be very wasteful with address allocations
- Lots of other new features
  - Built-in autoconfiguration, security options, …
- Not really in production use yet