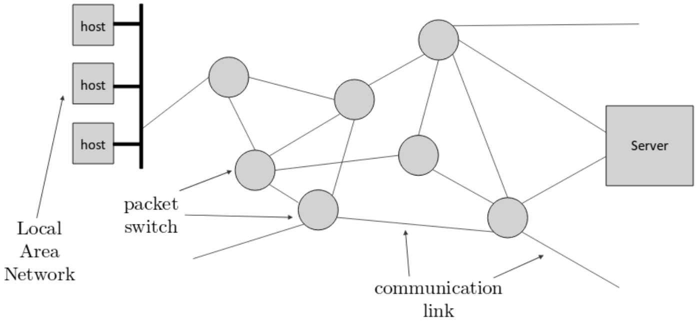
Reference Sheet for CO212 Networks and Communications

Autumn 2017

1 The Internet



- **Packet switch**: link-layer switch or router.
- **Communication link**: connection between packet switches and/or end systems.
  - Fibre optic cable, twisted pair copper wire, coaxial cable, wireless local area links, etc.
- **Route** (**path**): sequence of switches a packet goes through.
- **Protocol**: control the sending and receiving of information between end systems/packet switches

Packet Switching vs Circuit Switching Packet-switched networks (e.g. the Internet):

1. Information transmitted in **packets**: formatted unit of data.
2. Switches/routers operate on individual packets.
3. Switches/routers receive packets and **forward** them - forwarding decision taken on the basis of information within the packet.

Circuit-switched networks (e.g. the telephone network):

1. Has a setup phase: network reserves all resources for the connection (links, buffers, switches, etc.).
2. Links are dedicated for the entire duration of the connection.
3. Connection is destroyed and resources are freed.

Comparing packet and circuit switching:

- Packet switching is **connectionless**, circuit switching is **connection-oriented**.

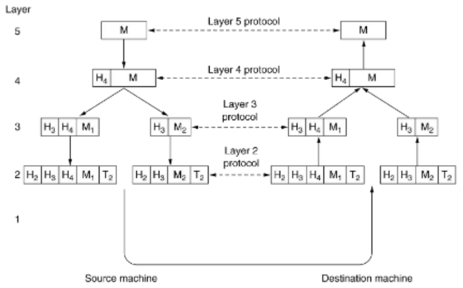
	Packet Switching	Circuit Switching
Setup cost	None	Expensive
Processing cost	For each forward	Little / none
Space overhead	For each packet	Little / none
Quality of Service	Difficult to guarantee	Easily guaranteed
Utilisation of links	Shares links - efficient	Limited sharing - inefficient

Communication Protocols An agreement on how communication is to proceed.

- Must be an **executable specification** which is **unambiguous** and **complete**.
- Needs to be able to solve **addressing**, **error control**, **flow control**, **multiplexing/demultiplexing** and **routing**.

1. **Handshake**: establish identities and/or contact.
2. **Conversation**: exchange information.
3. **Closing**: terminate conversation.

Protocol Layering:



- **Service**: set of primitives that layer provides to the layer above.
- **Protocol**: set of rules that prescribe layout and meaning of packets, and how the should be sent.

- Layer k puts its packet as data into a layer $k-1$ packet - which may add a header and/or trailer.
- **Fragmentation** may be required: layer k data may have to be split accross several layer $k-1$ packets.

Internet Protocol Stack:

1. **Application**: defines application functionality and message formats. E.g. Web (HTTP/HTTPS), E-mail (SMTP), BitTorrent, etc.
2. **Transport**: offers connection-oriented and connectionless services. Usually:
    (a) provides interface through **sockets**,
    (b) allows setting up connection, and delivering data reliably and in the order it was sent,
    (c) ensures fast senders don't overwhelm slow receivers (**flow control**),
    (d) supports **secure connections**.
3. **Network (Internet)**: describes how **routing** and **congestion** is solved.
4. **Data Link**: allows computers to share common channel and detects transmission errors (e.g. parity bit, checksum). E.g. Ethernet.
5. **Physical**: describes transmissions of raw bits.

Data Transfer

- **Bandwidth**: amount of information that can enter (or leave) a connection per time unit.
- **Throughput**: actual amount of information that enters (or leaves) connection per time unit.
- **Latency**: time it takes for one bit to go through the connection.
- **Transfer time**, $\Delta =$ propagation delay (latency), $d +$ transmission delay (packet size, L / throughput, R).

With two hops, we also include **router delay**, d_x, which is made up of:

1. **Processing delay**, d_{proc}: check bit errors, determine output link (usually negligible).
2. **Queuing delay**, d_q: time waiting at output link for transmission (depends on average packet arrival rate, a):
    (a) $\frac{La}{R} \approx 0$: small avg. queuing delay.
    (b) $\frac{La}{R} \to 1$: large avg. queuing delay.
    (c) $\frac{La}{R} > 1$: avg. queuing delay is infinite.

2 Application Layer

Hosts and Processes:

- **Host** (end system): May run multiple processes.
- **Process**: Addressed within its host by a port number.
- **Socket**: Network interface, managed by OS.

Clients and Servers:

- **Client**: process that initiates the communication (and establishes connection on a connection-oriented service).
    - Creates a socket C by connecting to a server application on host H and port P.
    - Uses C by reading and writing data into it.
    - Disconnects and destroys C.
- **Server**: process that waits to be contacted.
    - Creates a socket S by accepting a connection on port P.
    - Uses socket S by reading and writing data into it.
    - Disconnects and destroys S.
- **Peer-to-Peer**: Processes can act as both clients and servers.

2.1 Web (HTTP)

- Uses connection-oriented transport, TCP. (port 80).
- Consists of a sequence of requests issued by the client, and respoonses issued by the server.
- Stateless.

Requests

1. Request line: `METHOD URL Protocol/Version`.
2. Header lines: `Name:  Value`.
3. Empty line.
4. Object body.

Methods

- **GET**: retrieve object identified by URL.
- **POST**: submit data to the service.
- **OPTIONS**: request available communication options.
- **HEAD**: like GET, but without the body.
- **PUT**: store given object under given URL.
- **DELETE**: deletes given object.

Responses

1. Status line: `Protocol/Version Code Status`.
2. Header lines: `Name: Value`.
3. Empty line.
4. Object body.

Status Codes

- **1xx**: Informational.
- **2xx**: Successful operation.
- **3xx**: Redirection.
- **4xx**: Client error.
- **5xx**: Server error.

E.g. You can use `telnet` to make a request:

```
telnet www.doc.ic.ac.uk 80
GET /~js4416/index.html HTTP/1.1
Host: www.doc.ic.ac.uk
```

and get a response:

```
HTTP/1.1 200 OK
Date: ...
(Header lines) ...
Via: ...

<!DOCTYPE html>
(Page content) ...
```

TCP connections

- HTTP/1.0 used one TCP connection per object. Now possible with `Connection: close`.
- HTTP/1.1 introduced persistent connections - the same connection may be used to issues multiple requests and replies.

Web Caching Improves performance and security, but adds complexity and may reduce data 'freshness'.
A client request goes to a proxy (**cache**) server, which may:

1. Forward the request to the origin server.
2. Get the response from the origin server.
3. Store the object for some time.
4. Forward the response back to the client.

or

1. Respond immediately, using a cached object.

Implementation:

- Servers specify explicit expiration times using the `Expires` header or setting `max-age` and `must-revalidate` in the `Cache-Control` header.
- A client or proxy can also use a `Cache-Control` header to ensure they have a recent object. E.g. `Cache-Control: no-cache` or `Cache-Control: max-age=60`.
- A client or proxy can use a conditional `GET` with an `If-Modified-Since` header.

Sessions

- Server may use a `Set-Cookie` header, client may use `Cookie` header.
- Allows for sessions: users may be identified.

Dynamic Web Pages

1. **Common Gateway Interface**: identifies program and parameters in URL, which builds and returns a webpage.
2. **Scripting**: embedded scripts executed when page is processed - may be server-side (e.g. PHP) or client side (e.g. JS).

2.2 Domain Name System

- IPv4 (4 bytes) and IPv6 (16 bytes) addresses - easily processed by routers.
- Host names mapped to IP address by **Domain Name System** (DNS).
- Host names are within a hierarchical name space. There are 13 'root' DNS servers that know where TLD servers are.
- Each domain has an authoritative server, but caching used extensively.

DNS Query Types

- A: maps host name to address.
- NS: query for authoritative name server.
- CNAME: query for a canonical (primary) name.
- MX: query for mail exchange server.

DNS Protocol

- Uses connectionless transport protocol, UDP (port 53).
- Queries and replies have the same format.

Round Robin DNS Responds with a list of IP addresses , and order is changed to balance load.

Tools

- `host imperial.ac.uk` performs a simple DNS lookup.
- `dig` is similar to `host -v`.
- `nslookup imperial.ac.uk [ns0.ic.ac.uk]` queries nameservers, you can use `-type=` for a certain query type, and can also ask for an answer from a specific nameserver.

Content Distribution Networks Serve multiple copies at geographically distributed sites:

- **Enter deep**: CDN servers deep into many access networks.
- **Bring home**: smaller number of large clusters in points of presence near access networks.

2.3 Electronic Mail

- **User agent** allows user to read, compose, reply to, send and forward messages.
- **Mail servers**, accept messages for remote and local delivery (using transport protocol), and allow user agents to access local mailboxes (using access protocol).

SMTP Uses connection-oriented protocol, TCP.

1. Sets up TCP/IP connection between client and server (address can be found from DNS).
2. Client requests server to accept its messages.
3. Server responds, so that client can send.

Format:

```
HELO jordanspooner.com
MAIL FROM: jordan@jordanspooner.com
RCPT TO: js4416@ic.ac.uk
DATA
From: Jordan Spooner <jordan@jordanspooner.com>
(Header lines) ...
Subject: Test Email

(Content)...
.
QUIT
```

Only supports 7-bit content - but extended by the **Multipurpose Internet Mail Extensions** (MIME) specification.

POP3 Allows remote access, but assumes retrieved mail is deleted at the server. This is solved by IMAP.

3 Transport Layer

	TCP Transmission Control Protocol	UDP User Datagram Protocol
Type of service	Reliable connection-oriented	Unreliable connection-less
Data are called	Segments	Datagrams

Ports Cross-platform process identifiers.

- Socket connection identified by two pairs of IP address, port number, TCP/UDP.
- First 1024 ports are reserved. E,g, 80 for HTTP, 443 for HTTPS, 110 for POP3, 25 for SMTP, 21 for FTP, 22 for SSH, ...

Use `netcat` to read and write across network connections.

3.1 Transport Layer Interface

Berkeley socket interface:

1. Server and client each bind a transport-level address to a locally created socket.
   - `SOCKET(protocol)`: Create a new communication endpoint.
   - `BIND(socket, address)`: Attach a local address to a socket.
2. Server starts listening on this socket, waiting for connections from clients.
   - `LISTEN(socket, N)`: Announce willingness to accept N connections.
3. Server can accept or select connections from clients.
   - `ACCEPT(socket)`: Block until a remote client wants to establish connection.
4. A client connects to the socket, providing full transport-level address.
   - `CONNECT(socket, address, port, protocol)`: Attempt to establish a connection.
5. Client and sever communicate through send / receive operations on their respective sockets.
   - `SEND(socket, data)`: Send data over connection.
   - `RECEIVE(socket)`: Receive data over connection.
6. Communication ends when socket is closed.
   - `CLOSE(socket)`: Release the connection.

Without a connection (e.g. UDP): no need for `LISTEN`, `ACCEPT` and `CONNECT`.

3.2 TCP

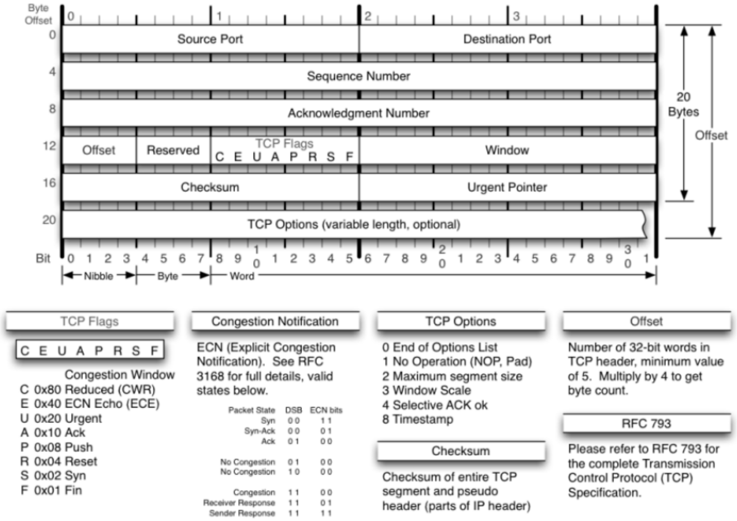
- **Reliable** data transfer.
- Uses **connections**.
- Uses **congestion control**.
- **Duplex**. Both endpoints can send and receive at the same time.

Segmentation TCP data are transmitted within TCP segments.

- **Maximum Segment Size** (MSS): maximum amount of application data transmitted in a single segment (excluding headers).
  - Usually related to MTU of connection, to avoid fragmentation.
- **Maximum Transmission Unit** (MTU): largest link-layer frame available to sender.
  - **Path MTU Discovery** (PMTUD): determine largest link-layer frame that can be sent on all links from sender host to receiver host.

Header Fields

- **Sequence number** indicates the place of the first byte carried by the segment.
  - When TCP connection is set up, a random Initial Sequence Number is decided.
- **Acknowledgement number** is the first sequence number **not yet** seen by receiver.



Congestion Window
C 0x80 Reduced (CWR)
E 0x40 ECN Echo (ECE)
U 0x20 Urgent
A 0x10 Ack
P 0x08 Push
R 0x04 Reset
S 0x02 Syn
F 0x01 Fin

ECN (Explicit Congestion Notification). See RFC 3168 for full details, valid states below.

Packet State	DSB	ECN bits
Syn	0 0	1 1
Syn-Ack	0 0	0 1
Ack	0 1	0 0
No Congestion	0 1	0 0
No Congestion	1 0	0 0
Congestion	1 1	0 0
Receiver Response	1 1	0 1
Sender Response	1 1	1 1

0 End of Options List
1 No Operation (NOP, Pad)
2 Maximum segment size
3 Window Scale
4 Selective ACK ok
8 Timestamp

Checksum of entire TCP segment and pseudo header (parts of IP header)

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

Please refer to RFC 793 for the complete Transmission Control Protocol (TCP) Specification.

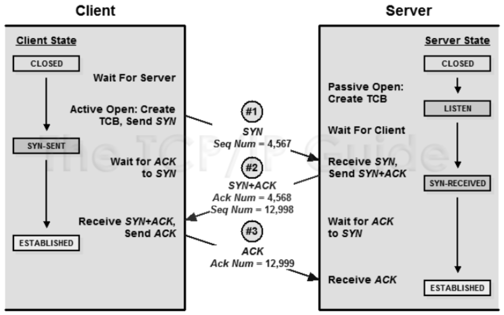
Three-way Handshake

1. Client sends `SYN` with its ISN.
2. Server responds with both:

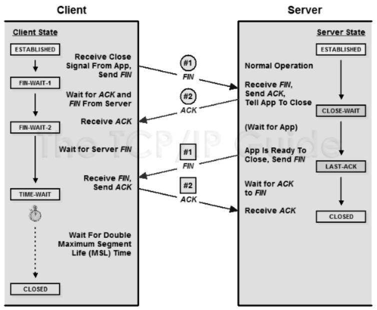
(a) SYN with its ISN.
(b) ACK with the first unseen client sequence number.

3. Client responds with ACK with the first unseen server sequence number, and its new sequence number.

Disconnection uses FIN instead of SYN, and usually two exchanges.



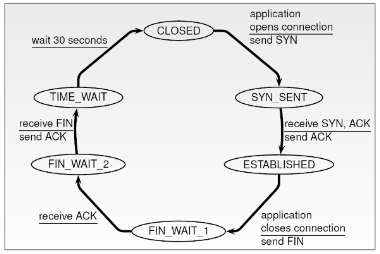
Connection Establishment



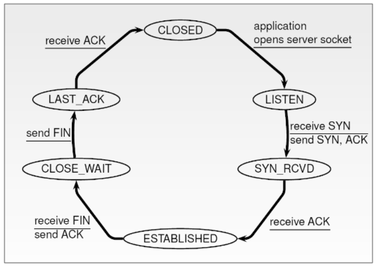
Connection Termination

States Use `netstat -at` to see current TCP connection states.

Client:


CLOSED
application opens connection send SYN
SYN_SENT
receive SYN, ACK send ACK
ESTABLISHED
application closes connection send FIN
FIN_WAIT_1
receive ACK
FIN_WAIT_2
receive FIN send ACK
TIME_WAIT
wait 30 seconds

Server:



3.3 Error Detection and Retransmission

- TCP adds reliability on top of unreliable data-link layer. Bit errors can occur, hence we use a checksum.
- **Error detection**: use a checksum.
- **Receiver feedback**: if packet is corrupted, wrong or timeout:
  - Could send a NACK, protected with an error-detection code.
    - Corrupted ACKs are interpreted as NACKs.
    - We may get **duplicate** segments, which we can ignore using SEQ numbers.
      - SEQ number may be a single bit (**alternating bit protocol**).
  - Could send an ACK for the next needed byte. Many methods:
    - **Delayed**: wait for the next segment, if it doesn't arrive, send ACK.
    - **Cumulative**: send ACK for multiple segments at once.
    - **Duplicate**: send another ACK if an out of order segment arrives.
    - **Immediate**: immediately send an ACK for the first gap.

Congestion Control Aims not to overflow the network.

- **Congestion window**: how many bytes can be pushed before waiting for an acknowledgement. I.e. last byte sent - last byte ACKed $\leq W$.
  - Where $W = \min(\text{congestion window}, \text{receiver window})$.
  - The resulting maximum output rate $\lambda = \frac{W}{RTT}$.
- **Slow Start**:
  - Initial W is MSS.
  - Double W every RTT, until:
    - $W >$ ssthresh, then use Congestion Avoidance.
    - $W =$ ssthresh, then use either Slow Start or Congestion Avoidance.
- **Congestion Avoidance**:
  - $W = W + MSS \times \frac{MSS}{W}$ (W increased by approximately 1 MSS every RTT).
- **AIMD Additive-Increase / Multiplicative-Decrease**:
  - W is increased with every good acknowledgement by CA.
  - W is halved at every packet loss event.
- **Reliability and Timeout**:
  - Timeout should be longer than RTT (to avoid unnecessary retransmissions) but not too long (to detect and retransmit lost segments quickly).
  - Use $T = \overline{RTT} + 4 \times \overline{DevRTT}$, based on estimated RTT (get by ping).
- **Fast Recovery**: three duplicate ACKs interpreted as a NACK.
  - Timeout:
    - Set ssthresh to half the current window size.
    - Go back to $W = MSS$ and run SS.
  - NACK: Cut W in half and run CA.

Flow Control Aims not to overflow the receiver.

- Receiver sends its window size (max number of bytes that can be sent) along with its acknowledgement.
- A window size of 0 is valid.

Wireless TCP TCP assumes IP is running over wires. When packets are lost, TCP assumes congestion and slows down. Possible solutions:

1. Split TCP to distinguish between wired and wireless IP.
2. Let the base station do some transmissions without informing the source.

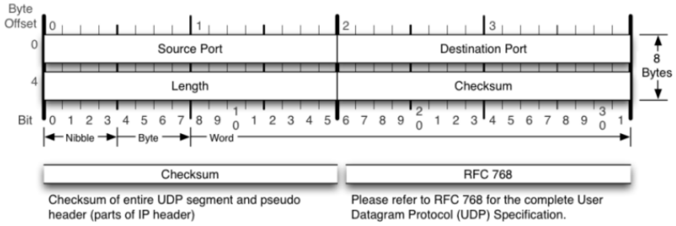
3.4 UDP

- No flow control, error control, retransmissions.
- Datagrams cannot be larger than 65K. (Usually use 500B or less).

Useful for:

- Finer Application Level control over what data are sent and when (e.g. real-time: Skype).
- No connection establishment (faster).
- Non connection state.
- Small packet header overhead.

Header Fields



3.5 Network Usage

$$\text{Utilisation Factor} = \frac{\text{how much we actually used the network}}{\text{how much we could have used it}}$$

4 Network Security

4.1 Terminology

- **Hacker**: once skilled, now cybercriminals. White/black/grey hats.
- **Virii**: e.g. ransomware/spyware/Trojans.
- **Anarchists**: physical.
- **Crackers**: use others' tools to infiltrate systems.
- **DDoSers**: participants in distributed denial of service attacks.
- **Spammers/botters**: mass senders of spam messages.
- **Pirates**: upload or download illegal content.
- **Cyberbullies**: harass, stalk, offend, threaten users.
- **Whistleblowers**: reveal inside secrets.
- **Social engineers/phishers/catfishes**: manipulators, pretend to be somebody else.

Black Hat Tools

- **Rootkits**: allow attackers to enter system.
- **Keyloggers**.
- **Trojans**: allow remote control of a system.
- **Evil twin**: free WiFi networks, sniff everything you send.

White Hat Tools

- **Tails**: OS that forgets everything.
- **Kali Linux**: provides security and pentesting tools.
- **Metasploit**: like `nmap`, identifies systems and vulns.

Network Security Issues

- **Access control**: only certain users are allowed to access a resource.
- **Authentication**: user knows that the resource really is what it says it is, and v.v.
- **Confidentiality**: user limits access to information / resources they own. (Encryption).
- **Integrity**: actions of a user should not affect integrity of resource.
- **Non-repudiation**: users cannot deny communication took place. (Monitoring, logging, auditing).

4.2 Firewalls

Access Control In a **secure channel**, the **guard** (e.g. firewall) controls:

- Which principals can access the resource.
- Where principals are allowed to be located.
- What requests principals are allowed to make.

A firewall:

- Controls access to the network (gateway between **internal** and **external** networks).
- Analyses inbound packets, and blocks / allows based on rules.

Examples:

- **Application-level gateway**: runs on the host.
  - Can block based on application-level content.
- **Proxy server**: runs on the entire network, can protect entire LAN.
  - Private network only accessible via proxy.
- **Circuit-level gateway**: act like a non-caching proxy.
- **Packet filtering**:
  - **Stateless**: checks source / destination IP address and ports.
  - **Stateful**: remembers connections and check contents of current and previous packets.

Bastion Hosts External hosts which expect to be attacked.

- Performs auditing / logging.
- Runs a trusted and secure OS.
- Administered via dedicated terminal.
- Only run necessary software, set file permissions, turn on quotas, process limits, remove regular user accounts, make filesystem read only, ...

Responsibilities:

- Relays connections and maintains state.
- Can authenticate users.
- Can drop connections if necessary.

Demilitarized Zone Area between you and the outside.

- External hosts can only speak directly to your internal hosts within the DMZ.
- All other non-DMZ hosts are protected by the gateway/router/firewall.
- Router uses NAT (network address translation) to get external messages to correct internal host.
- If you want to expose an internal host without putting it in the DMZ, you must **port forward** (from a given port of router's public IP to a given port of host's NAT-based LAN IP).

Getting Around Firewalls

- `ssh`.
- Spoof MAC/IP.
- Use a VPN (virtual private network) to tunnel around a firewall.
  - Firewall won't be able to know what you're doing.
  - As long as your tunnel uses SSL (secure sockets layer) / TLS (transport security layer).

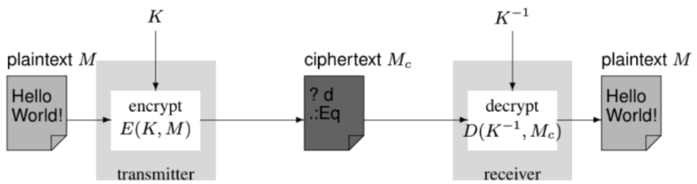
Access Control Lists Packet filtering rules.

- Rules are checked top to bottom. Allow what you need, block everything else or v.v.
- E.g. use `iptables` and `tcpd` to edit access control lists.

Keywords

- **IDS**: intrusion detection system.
- **IPS**: intrusion protection system.
- **NGFW**: next generation firewall: stateful firewall with IPS/IDS.
- **UMT**: unified thread management: NGFW but with extra capabilities (e.g. antispam/antivirus).

4.3 Cryptography



Symmetric (Secret Key) Encryption

- $K = K^{-1}$.
- K must be carefully distributed.

Asymmetric (Public Key) Encryption

- K is the **private-key**.
- K^{-1} is the **public-key**.
- Successfully decrypting the message using public key **authenticates** that the message came from the correct transmitter.
- Encrypting a message with the receiver's public key ensures **confidentiality** since only they can decrypt it.
- More secure but slower than symmetric encryption.

Secure Channel Establishment

1. Agree on a new key. **Diffie-Hellman** key exchange:
   (a) Choose a generator g and large prime p.
   (b) Bob chooses a secret b and Alice a secret a.
   (c) Bob calculates $x = g^b \mod p$ and Alice $y = g^a \mod p$, and share these.
   (d) They then calculate $y^b \mod p = (g^a \mod p)^b$ and $x^a \mod p = (g^b \mod p)^a$ respectively. The final key is $g^{ab} \mod p$.
2. Use trusted secure hosts. Obtain a key from the trusted host securely.

5 Network Layer

Provides facilities for getting data from a source to a destination.

- May make many hops (**routing**).
- Must know network **topology** to choose appropriate paths.
- Requires **load balancing** along routes.
- Has to deal with **network heterogeneity**.

Router

- Has **interfaces** (physical ports).

Forwarding

- Potentially multiple, asymmetric paths.
- Routers cooperate to find the best routes, by building a **sink tree** with **optimal routes**.
- Use `traceroute` command to see.

Routing Algorithms

- **Shortest path routing**: calculate using Dijkstra's algorithm.
- **Flood routing**: forward incoming packet across every outgoing link.
  - Always chooses **shortest path**.
  - Increases **overhead**.
  - Makes sense when **robustness** is required.

  1. After a packet was forwarded across N routers (**hop counter** $> N$), discard it.
  2. Avoid directed cycles by forwarding only once.
  3. Flood selectively, only in direction that makes sense.
- **Distance vector routing**: dynamic protocol that uses Bellman-Ford:
  - Good news propagates quickly.
  - Bad news propagates slowly (count-to-infinity problem). Use a longest acceptable path to avoid this.

  1. Consider costs that direct neighbours advertise to get packet to destination.
  2. Select neighbour whose advertised cost + cost to get to that neighbour is lowest.
  3. Advertise new cost to neighbours.
- **Link state routing**:
  1. Discover direct neighbours and get their addresses.
  2. Calculate cost for sending packet to each neighbour.
  3. Construct LSA (link state advertisement) packet with all information.
  4. Send / collect packet to / from **all** other routers.
  5. Run Dijkstra's algorithm locally to find shortest paths.

	DVR	**LSR**
Network knowledge	Local	Global
Computation	Global	Local
Synchronisation	Gradual	"Instant"

- **Hierarchical routing**: routing algorithm that can scale!
  - Go for suboptimal routes by introducing **regions**, and separate algorithms for intra-region and inter-region routing.
- **Broadcast routing**: attempt to send message to every host on network:
  - Use **flood routing**, if we can **limit** flood.
  - Multi-destination routing approach, requires list of all destinations.
  - **Multicast**. Build a sink tree and use for multicast route.
- **Multicast routing**:
  - **Solution 1**: Construct a spanning tree at each router using RPF (reverse-path forwarding).
    - Each router broadcasts packet to every adjacent router, except where it came from.
    - Accept packet only if it arrives on a direct path from origin.
  - Use a group ID to prune paths to nodes that do not contain members of the group.
  - **Solution 2**: Use **core-based trees**.
    - Single spanning tree per group with root near middle.
    - Hosts send multicasts to core.
    - Not optimal for all sources, but more scaleable.

Internetworking Construct gateways that interconnect different kinds of networks.

- **PAN**: personal area network, e.g. hotspot between phone and laptop.
- **LAN**: local area network, multiple users for the same purpose.
- **MAN**: metropolitan area network.
- **WAN**: wide area network, e.g. the Internet.

Devices

- **Repeaters** / **hubs**: physical layer devices that boost signals.
- **Switches**, **bridges**: data link layer devices that make **interconnections** based on MAC address.
- Multi-protocol **routers** / **gateways**: network layer devices that make decisions based on IP address.

The Internet Collection of **autonomous systems** connected by **backbones**.

- **Gateway routers** interconnect autonomous systems.
- Use **hierarchical routing**:
  - **Intra-AS routing protocols**: RIP (distance vector) and open shortest path first OSPF (link state).
  - **Inter-AS routing protocols**: BGP.

Border Gateway Protocol

- Neighbouring routers maintain connection to simplify reliability: provides reachability information from neighbour ASs.
- Determines routes to all outside subnets based on reachability information and policies.
- Path-vector protocol, based on distance-vector routing but paths instead of distances announced.
- Routers advertise routes to networks.
  - Destinations denoted by address prefixes.
  - May aggregate prefixes (**supernetting**).
- **Autonomous system number** (ASN): uniquely identifies each AS.
- BGP attributes: include AS-PATH and NEXT-HOP.
- BGP import policy: decides whether to accept or reject route advertisement.

5.1 Internet Protocol

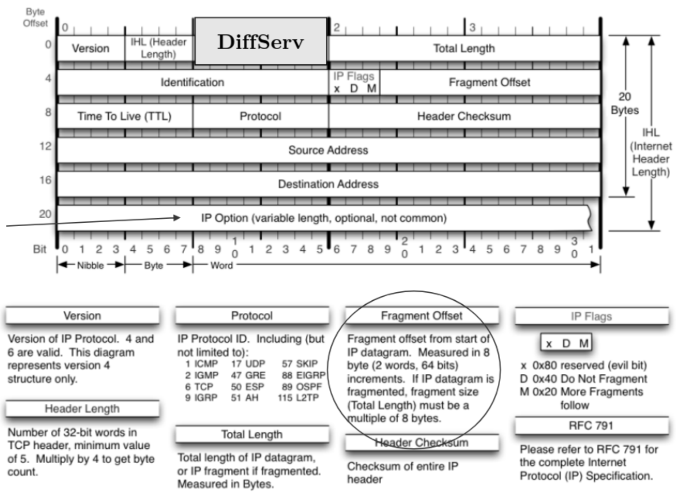
Best-effort: a packet-switched connectionless service.

- Datagram format.
- Fragmentation.
- Addressing.
- Packet handling.

Fragmentation

- If input datagram exceeds MTU of output link, IP datagram must be **fragmented**.
- Destination reassembles fragmented datagrams.
- **Fragment offset** is offset in units of 8 bytes.
- **Total length** is the number of bits in this fragment.

- **More fragments** flag will be set if more fragments are expected to follow.



Version of IP Protocol. 4 and 6 are valid. This diagram represents version 4 structure only.

IP Protocol ID. Including (but not limited to):

1 ICMP	17 UDP	57 SKIP
2 IGMP	47 GRE	88 EIGRP
6 TCP	50 ESP	89 OSPF
9 IGRP	51 AH	115 L2TP

Fragment offset from start of IP datagram. Measured in 8 byte (2 words, 64 bits) increments. If IP datagram is fragmented, fragment size (Total Length) must be a multiple of 8 bytes.

x 0x80 reserved (evil bit)
D 0x40 Do Not Fragment
M 0x20 More Fragments follow

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

Total length of IP datagram, or IP fragment if fragmented. Measured in Bytes.

Checksum of entire IP header

Please refer to RFC 791 for the complete Internet Protocol (IP) Specification.

Note:
Many of the available **IP Options** are not used because of security reasons

5.1.1 IPv4 Addressing

Each IP address is a associated with an interface.

Classful Addressing No longer used:

- Class A: 0NNNNNNN.HHHHHHHH.HHHHHHHH.HHHHHHHH.
- Class B: 10NNNNNN.NNNNNNNN.HHHHHHHH.HHHHHHHH.
- Class C: 110NNNNN.NNNNNNNN.NNNNNNNN.HHHHHHHH.
- Class D: 1110....: multicast address.
- Class E: 1111....: reserved for future use.

Subnet Masking Use a single network address for entire organisation, and divide internally into subnet addresses and host ids.

- **Classless Inter-Domain Routing** (CIDR) notation: address/prefix-length.
- Subnet mask for length p is p 1's followed by $32 - p$ 0's.
- Network name is given first IP.
- First IP of a host is the router.
- Last IP is the broadcast address.

Longest Prefix Matching Choose entry that matches destination address with longest prefix.

- 0.0.0.0/0 is the **default route** (otherwise broadcast).

Network Address Translation Assign each company / home a single IP, each device gets a unique local IP.

- Several private address ranges exist for local networks.
- You can find out your local IP, you can use `ifconfig`.
- When a connection is set up from a local IP on port X, the router uses its public address on port Y, and registers a mapping.
- Incoming connections require a static mapping from a port on the public address to a given port on a local address.

Criticism:

1. Violates architectural IP model of IP address uniquely identifying a machine.
2. Makes internet a connection-oriented network.
3. NAT makes assumptions about TCP, cannot easily support new transport protocols.
4. Uses behind NAT cannot be contact directly without port forwarding.

IPv6 could help deal with lack of IP addresses.

5.1.2 IPv6 Addressing

- Expanded addressing, supports anycast.
- Fragmentation done at end systems.
- Remove header checksum, since the transport protocols do this already.
- Remove options.

5.2 Internet Control Message Protocol

- Error reporting.
- Signalling.

E.g. used for `ping`.

5.3 Dynamic Host Configuration Protocol

Address configuration.

- Each new machine broadcasts DHCP DISCOVER packet when it connects.
- DHCP server replies with the assigned IP address.
- Mapping may be static, or assign different addresses each time a host connects.

6 Data Link Layer

6.1 Ethernet

Cables Goal is to protect against electromagnetic interference (EMI):

- **UTP** (unshielded twisted pair).
- STP (shielded/screened twisted pair).
- **FTP** (foiled twisted pair).
- **SFTP** (shielded and foiled twisted pair).

Pinouts

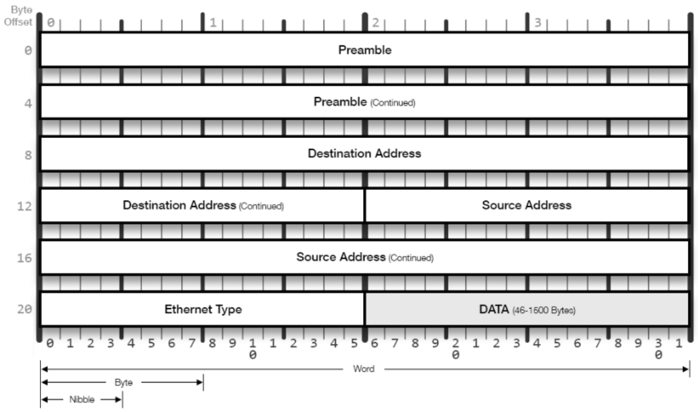
- **Straight-through**: communicate between **different** OSI layers.
- **Crossover** (media dependent interface MDI): communicate between devices of the **same** OSI layer.
- **Rollover** (media dependent interface with crossover MDIX): directly tap into a networking device.

Switches

- Forwards messages out of the right port.
- Uses a forwarding information base (FIB) MAC table.

Frames

- Has a cyclic redundancy check (CRC) checksum in the footer).



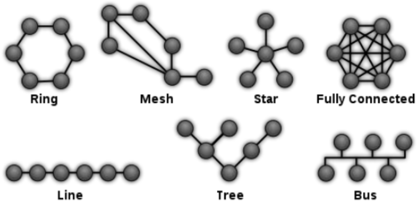
MAC Addressing

- Each network interface card (NIC) has a MAC address.
- 48 bits / 6 bytes.
- 8th (I/G) bit decides if individual or group address.
- 7th (U/L) bit decides if universally or locally administered.
- FF:FF:FF:FF:FF:FF is the **broadcast address**.
- IP is linked to MAC address, so routers can explain to switches which host a packet is for.

Address Resolution Protocol Use `arp` to see what your computer knows. Use `tcpdump` to see communication.

1. **Router**: asks each host on LAN if they have the request IP. (Broadcasts a frame).
   - `arp who has host_ip tell router_ip (router_mac)`
2. **Host**: checks it has the requested address, if so, send reply with its MAC.
   - `arp reply host_ip is_at host_mac (router_mac)`
3. **Router**: receive ARP message and caches IP and MAC.

6.2 Network Topologies



- **Bus**: used main coaxial cable to connect all hosts, use a BNC coaxial T connectors and terminators.
- **Ring**: requires 2 NICs per host. If a link gets cut, network dies. Also **dual-ring** which requires 4 NICs per host.
  - **Fibre Distributed Data Interface** (FDDI): supports long distance transmission, and can short circuit two rings together.
- **Token ring**: connect all hosts to a multi-station access unit (MSAU). Data flows by passing around a token.
  - **Frames**: **access control** bit at the start of all frames to identify presence of token, **frame status** bit to say whether address was found, **inter-frame gaps** to separate them.
  - **Active monitor**: takes responsibility for generating tokens, any host must be able to do this.
  - **Concentrators** avoid single failures breaking links by switching out faulty hosts.
- **Star**: no token. Central device is a single point of failure (SPoF).
- **Line**: ring that doesn't meet at the ends (terrible).
- **Tree**: a star bus.
- **Mesh**: some connected to some. More fault tolerant, but expensive.
- **Fully connected**: expensive and difficult to manage.
- **Hybrid**: mix of topologies.

6.3 Medium Access Control

Coordinates channel access.

- A separate layer within the Data Link layer (the other is Logical Link Control).

Strategies

1. **No control**. let stations retransmit after collision.
   - Fine if utilisation is low, otherwise very inefficient.
2. **Round-robin**. stations take turns using the channel.
   - Used by **token-based** MAC systems - only station with the token can transmit.
3. **Reservation**: stations obtain channel reservation before transmitting.
   - Used in **slotted** systems - need to manage reservations.

Static Channel Allocation

1. **Time division multiplexing** (TDM): station must wait its turn to transmit.
2. **Frequency division multiplexing** (FDM): station must use only limited frequency band.

Dynamic Channel Allocation

1. **ALOHA**: When a station has a frame to be transmitted, send it. When collisions occur, wait a **random** amount of time and retry.
2. **Slotted ALOHA**: Only specific time slots when stations may transmit.
   (a) **Carrier sensing**: transmission only allowed when channel is idle.
      i. **CSMA/CD**: Carrier sense multiple access / collision detection (Ethernet).
      ii. **CSMA/CA**: Carrier sense multiple access / collision avoidance (WiFi).

Collision Detection

- Station senses channel while transmitting to know whether its frame is OK.
- **CD**: transmission stops after collision, add **jamming signal**.
- Host must transmit for long enough to know frame is OK (min length is $2n$ where n is end-to-end transmission delay).

Back-Off

- **1-persistant CSMA**: Station keeps checking if the channel is free and then transmits immediately.
- **Non-persistent CSMA**: If channel is busy, wait for a random amount of time before checking again and then transmit immediately.

- p-**persistent CSMA**: Station keeps checking if channel is free and transmits with probability p.
- **Binary exponential back-off**: after c collisions, choose slot in range 0 to $2^c - 1$ for next attempt. Give up after 10 collisions (1023 upper limit).

Token Passing An alternative to CSMA/CD.

- Collisions are inevitable using CSMA/CD, stations may be delayed indefinitely.

Carrier Extension Pad any packet up to 512 bytes.

Frame Bursting Allow host to send multiple frames: many short frames with carrier in between.

Switched topology One host on each port to avoid collisions. Switches only transmit when channel available.

7 Physical Layer

Cables

	Repeater Spacing
Twisted Pair	2km
Coaxial Cable	1-9km
Fibre Optic	40km

- Attenuation, bandwidth.

Patch Panels Connects sockets to switches.

Wireless Transmission

- Convenient when communication required without wires.
- Bidirectional by default.

Information Representation

- **Baud rate**: symbol rate per second.
- **Bitrate**: how many times per unit **can** the symbol change?
- A digital channel can be implemented by an analogue channel using a **modem**.
- An analogue channel can be implemented by a digital channel using a **codec**.

Properties of Signals

$$c = f\lambda$$

If λ is in meters and f is in MHz, $\lambda f \approx 300$.

- **Waveform**: shape.
- **Amplitude**: range.
- **Wavelength**: distance signal travels before repetition.
- **Frequency**: number of repetitions per unit time.

Modulation Encode digital information signal to be transmitted effectively using bandwidth supported by a channel.

- **Baseband modulation**: unmodified.
- **Broadband modulation**: use physical carrier signal to encode information signal.
  - **Amplitude modulation** (ASK): carrier frequency is modulated in amplitude (prone to interference).
  - **Frequency modulation** (PSK).
  - **Phase modulation** (PSK): phase change represents a 1 bit.
  - More advanced modulation techniques transmit multiple bits per symbol - using combination of modulation schemes.

DSL

- Conventional phone lines can only reach 56Kbps download speed, limited to 3000Hz bandwidth.
- DSL (digital subscriber line) removes the bandwidth filter.
- ADSL (asymmetric). E.g. divides 1.1MHz bandwidth into 256 channels of 4000Hz each, mostly allocated to download.