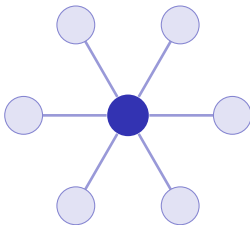


Communication

AS Computer Science ■ Topic 2

A-Level Computer Science



What we will cover

- 1 Networks: types & models
- 2 Topologies & cloud
- 3 Hardware & routing
- 4 Ethernet & streaming
- 5 Internet, IP & DNS
- 6 Recap





Why use a network?

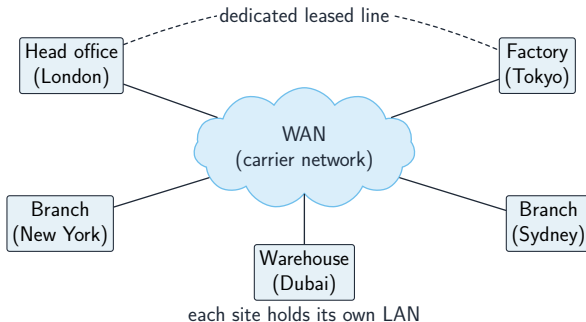
A **network** connects devices to communicate and share:

- share resources (printers, files, internet)
- central management & backup
- communication & remote access



Ethernet rj45

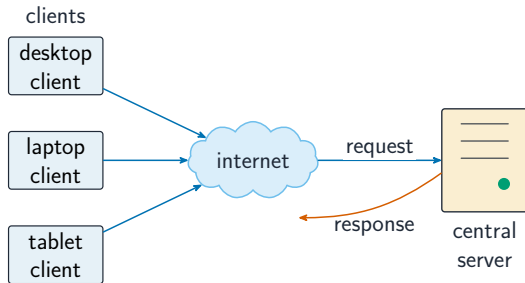
LAN vs WAN



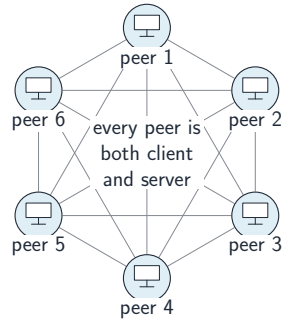
- **local area network (LAN)** – small area, one owner, fast, low latency
- **wide area network (WAN)** – large area, over **public switched telephone network** links, telecom links, slower; joins LANs

A LAN is usually owned by the organisation itself; a WAN leases telecom links. The internet is the largest WAN.

Client-server and peer-to-peer



client-server -: central



peer-to-peer : equal peers

Server: easy to manage but a single point of failure. P2P: robust but harder to secure.

Thin vs thick clients

thin client

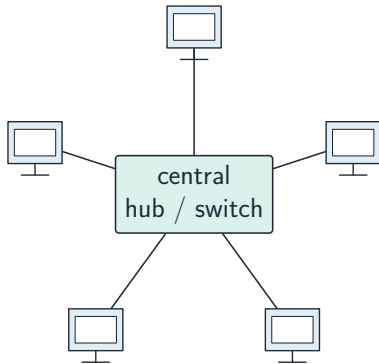
little local processing/storage
relies on a powerful server
(web terminal, remote desktop)

thick client

strong local processing/storage
runs full applications itself
(a normal desktop PC)

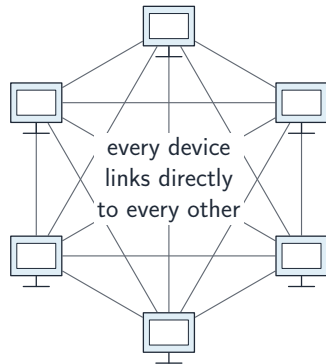


Network topologies (1)



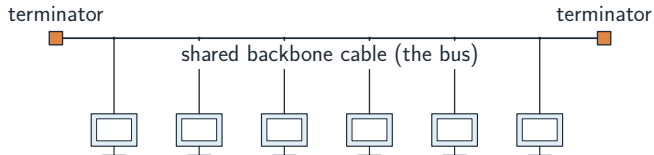
***star** : a central switch (most common)*

In a star one cable fault affects one device – but the switch is a single point of failure.

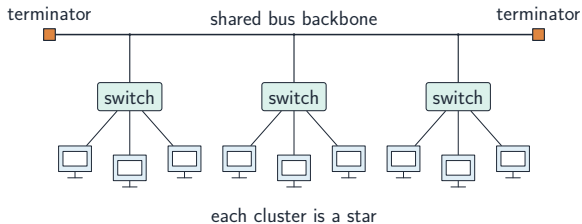


mesh** : many paths, **fault-tolerant

Network topologies (2)



bus : one shared cable – cheap, but one break stops everyone



Cloud computing and wired vs wireless

Cloud computing : services over the internet, hosted by a third party.

- + **scalability** , low cost, anywhere
- – needs internet; data held by others

Wired (**Ethernet** over **twisted-pair** or **fibre-optic**) : faster, lower latency, fewer errors, more secure. **Wireless** (Wi-Fi, Bluetooth, cellular): mobile and cable-free, but slower and open to interference and eavesdropping.

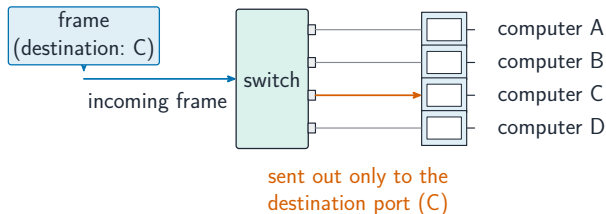
wired: faster,
safer, lower latency

wireless: mobile, no
cables, but slower

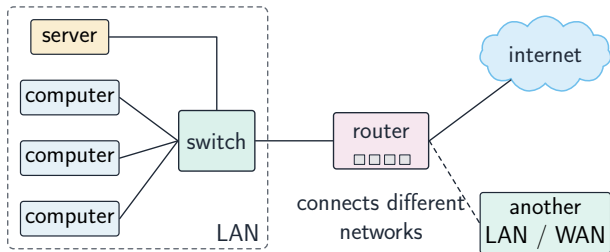


LAN hardware

- **network interface card (NIC)** – sends/receives; unique **MAC address** **MAC**
- **switch** – forwards a frame to the right port only
- **hub** – copies to all ports (obsolete)
- **wireless access point (WAP)** – joins **wireless network interface card** clients to a wired LAN
- cabling – twisted-pair for short runs, fibre for longer ones



Routers

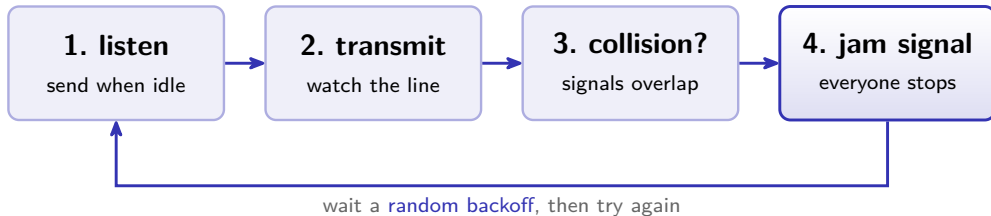


A **router** connects **different networks**:

- forwards **packets** by IP (routing table)
- **network address translation** (NAT), **DHCP**, and a **firewall** to block unwanted incoming traffic

Ethernet and CSMA/CD

On shared media a **collision** can happen – **CSMA/CD** :



Modern **full-duplex** switches avoid collisions altogether.

Wired against wireless, and how Ethernet shares a wire

for the same generation

wired wins on speed and reliability; **wireless** wins on convenience

1. carrier sense

listen first – if the medium is busy, wait

2. multiple access

any station may transmit when it hears silence

3. collision detection

keep listening while sending; a clash is a collision

4. back off

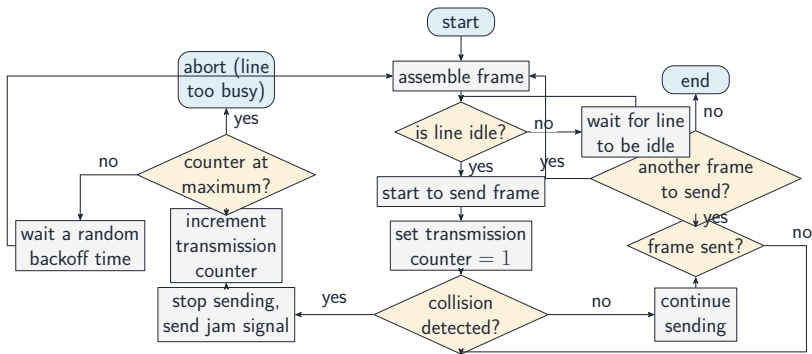
stop, wait a **random** time, and try again

why random

if both waited the same time they would collide again immediately

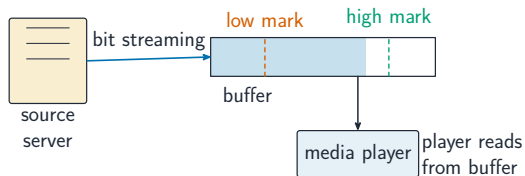
Wireless cannot detect a collision while transmitting, which is why Wi-Fi uses collision *avoidance* instead of detection.

The CSMA/CD process for handling collisions on



Listen first, and if two stations still transmit together, both **stop** and wait a **random** time before retrying – random, so they do not collide again.

Bit streaming



Bit streaming : play media as it arrives.

- **real-time** (live): low latency, no rewind
- **on-demand**: **buffer** ahead, can pause

The pipeline: capture and sample → encode with **lossy** compression → send → the receiver buffers briefly and plays, **dropping any late packet** – a live stream cannot wait – the encoded stream must fit the available **bandwidth** .

Bit streaming, and why it must be lossy

real-time streaming

live: the data is played as it arrives and not stored

on-demand streaming

the file exists on a server and is buffered as you watch

the buffer

a store filled ahead of the play point, so a brief slowdown does not stop playback

lossy compression

used here because moving pictures **hide small losses**, and the stream must be small enough to fit the **bandwidth**

the requirement

bit rate must be **no greater** than the available bandwidth, or the buffer empties

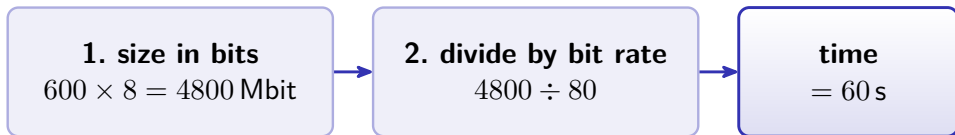
the trade

more compression means lower quality but fewer interruptions

Everything here is bandwidth against quality. Naming the buffer and what it protects against is what the question is asking for.

Worked example: how long to download?

A 600 MB video over an 80 Mbit/s connection – how many seconds?



File sizes are in **bytes** (MB); connection speeds are in **bits** per second (Mbit/s).
Multiply by 8 first – forgetting it is the classic exam slip.



Internet vs World Wide Web

Internet

the global network of
networks (TCP/IP)

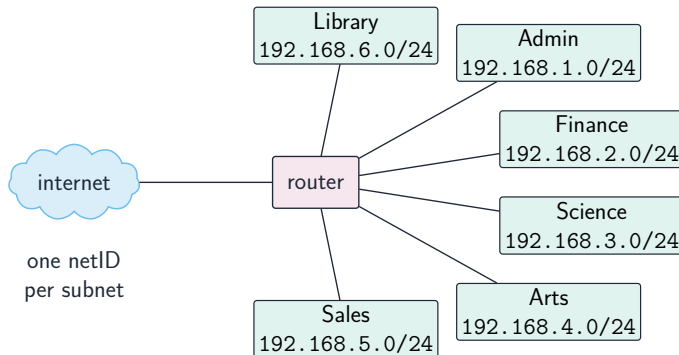
World Wide Web

one [service](#) on it:
linked pages via HTTP

The Internet is the [physical network](#) – cables, routers, addresses. The Web is [one service](#) running on top of it. Email and file transfer are other internet services, and are not part of the Web.

IP addresses and subnetting

- IPv4: 32-bit (192.168.1.10)
- IPv6: 128-bit, far more addresses
- **subnet** : split by a **subnet mask** (255.255.255.0)



Address kinds

private vs public

private: inside a LAN; public:
globally routable (from **ISP**)

static vs dynamic

static: fixed (servers); dynamic:
from **DHCP** , may change

NAT protection

devices behind NAT are not directly reachable from the internet

IP addresses, subnets and the URL

IPv4

32-bit, four denary numbers 0–255 – about 4.3×10^9 addresses, **now exhausted**

IPv6

128-bit, eight hexadecimal groups – effectively unlimited

a subnet

the mask splits the address into a **network** part and a **host** part

the mask

255.255.255.192 keeps 26 network bits, leaving 6 for hosts – 64 addresses per subnet

a URL

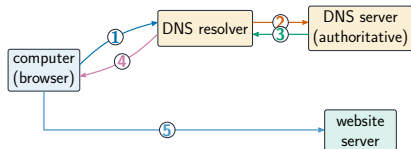
locates a resource on the WWW: **protocol**, **domain name**, then the **path** to the file

DNS

resolves the domain name to an IP address, asking successively higher name servers until one answers

AND the address with the mask to get the network. 192.168.10.130 and 192.168.10.200 under /26 are on **different** subnets – 128 and 192.

URL and DNS



- 1 computer asks the resolver for the IP
- 2 resolver queries a DNS server
- 3 server returns the IP address
- 4 resolver passes the IP back to the computer
- 5 browser connects to that IP and requests the page

A **URL** names one resource:

`protocol://domain/path`

- **https** – the **protocol**
- **example.com** – the **domain name**
- **/notes/1** – the path on that server

The **domain name system** (DNS) turns the domain name into an **IP address** before the browser connects.

Worked example – which subnet is this host on?

A host has IP 192.168.10.130, mask 255.255.255.192. Which network is it on, and is 192.168.10.200 on the same one?

- The mask's last octet 192 is 11000000, so the first **26** bits are the network part.
- Subnets step in blocks of $256 - 192 = 64$: .0, .64, .128, .192.
- 130 falls in the block from .128: the network is 192.168.10.128/26, hosts .129–.190, broadcast .191.
- 200 falls in the *next* block, so it is on a different subnet – and traffic between them must pass through a router.

Work in the last octet only. The block size is 256 minus the mask value, and every subnet starts at a multiple of it.

Exam tips

- Distinguish **LAN vs WAN** and **client-server vs peer-to-peer** by who stores and controls the resources.
- Match each **topology** (bus, star, mesh) to its advantages and drawbacks (cost, reliability, collisions).
- Know the job of each device: a **switch** directs within a LAN by MAC address, a **router** routes between networks by IP.
- Explain **bit streaming** and why buffering is needed (data arrives at a different rate from playback).
- Distinguish **IPv4 vs IPv6** and public vs private addresses; DNS turns a URL into an IP address.



Topic 2 – key takeaways

1

Types

LAN vs WAN; client-server vs peer-to-peer

2

Topologies

bus, star, mesh, hybrid

3

Hardware

switch forwards by MAC; router by IP

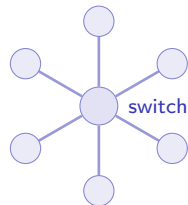
4

Internet

Web is a service on it; DNS maps name to IP

Check yourself

- 1 Which device forwards frames by MAC address?
- 2 In a star topology, what fails if the switch fails?
- 3 What does DNS convert?
- 4 Internet or Web: which is the service?



Answers 1. a switch 2. the whole network 3. domain name → IP address 4. the Web