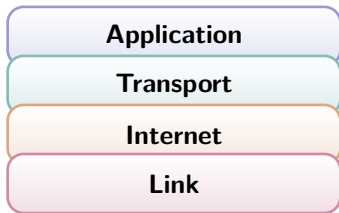


Communication & Internet Technologies

A-Level Computer Science ■ Topic 14

A-Level Computer Science



What we will cover

- 1 Why protocols?
- 2 Layered protocols & TCP/IP
- 3 Application-layer protocols
- 4 Circuit vs packet switching
- 5 Recap



A protocol is a shared language

A **protocol** is a *set of rules* for how devices communicate. Both ends must follow the same rules, or the signals are meaningless.

format

where addresses
and payload sit

order

who speaks
first; when to
acknowledge

timing

timeouts and
retransmits

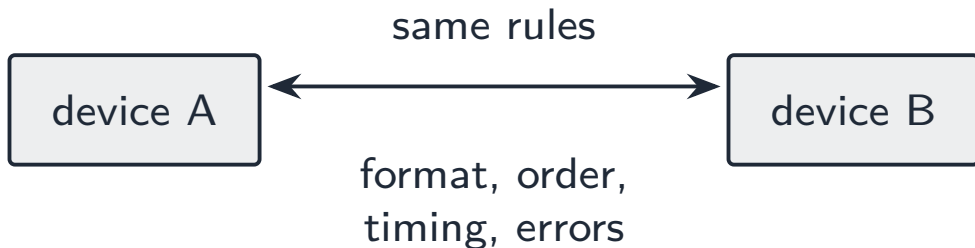
errors

what to do when
something fails

Without an agreed protocol two devices cannot understand each other – even on the same wire.

What a protocol fixes

Format, order and timing – agreed in advance, so two machines that have never met can talk.





Why split a network into layers?

Networking is complex, so it is split into **layers** . Each layer has one focused job and talks only to the layer above and below.

- **modularity** – replace Ethernet with Wi-Fi without touching the rest
- **standardisation** – vendors interoperate
- **abstraction** – you ignore details handled elsewhere

Application

Transport

Internet

Link

The internet uses the four-layer **TCP/IP suite** .

The four TCP/IP layers

Application

what the user program does

HTTP, FTP, SMTP, IMAP

Transport

end-to-end delivery between processes

TCP, UDP

Internet

routing packets between networks

IP

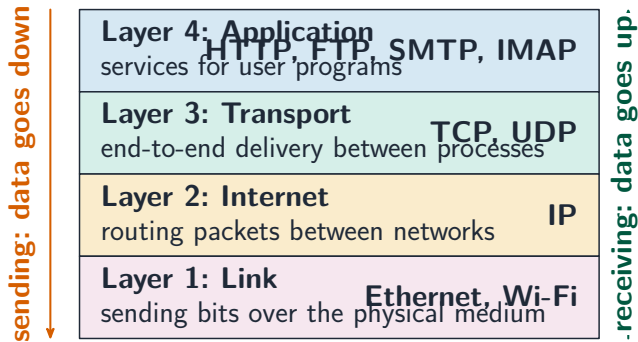
Link

bits over the physical medium

Ethernet, Wi-Fi

The four layers

Each layer talks to its opposite number on the other machine and uses only the layer below.



Transport – TCP vs UDP

The **transport layer** delivers data end-to-end between processes, identified by **port numbers**.

TCP

connection-oriented

sets up a connection; all data arrives *in order*; lost packets are retransmitted; flow control.

HTTP, HTTPS, SMTP, FTP

UDP

connectionless

sends and forgets – no acknowledgements, no ordering. Low overhead, no guarantees.

streaming, DNS, gaming

TCP against UDP

TCP acknowledges and retransmits; UDP just sends. Streaming prefers late-free to loss-free.

TCP

connection set up
all data arrives in order
reliable, more overhead

UDP

no connection
send and forget
fast, no guarantees

Internet layer and link layer

Internet layer

- **IP** carries packets between hosts
- each packet has a source and destination **IP address** IP
- **routers** forward it onward
- IP does **not** guarantee delivery – that is TCP's job

Link layer

- sends bits over one physical link
- frame header with **MAC addresses** MAC
- Ethernet uses **CSMA/CD** /
- a **switch** forwards a frame only to the correct port

Copper, Wi-Fi or **fibre-optic** (bits as flashes of light) are all link-layer media.

Internet layer



*A home Wi-Fi router: it forwards packets between
your devices and the internet*

A cable modem connects a home network to the internet provider

Link layer

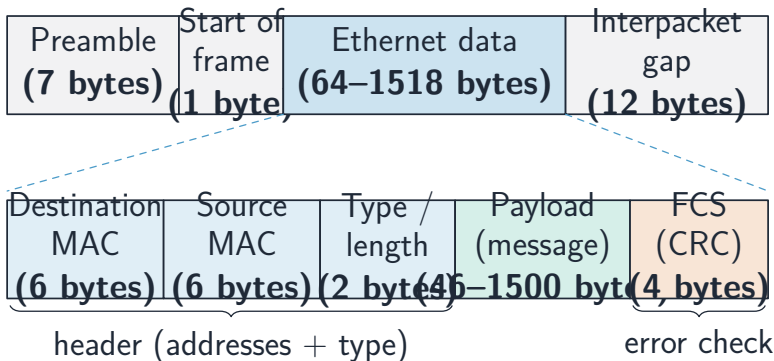


A satellite dish sends and receives data by radio over a long distance

A network switch connects many wired devices on a local network

Inside an Ethernet frame

Addresses, payload and a check sequence – the link layer's envelope.





Common application-layer protocols

HTTP HTTPS	80 443	browsers fetch pages; HTTPS is HTTP over TLS
FTP	21	transfer files between client and server
SMTP	25	send email; receiving uses POP3 or IMAP
POP3 IMAP	110 143	POP3 downloads and usually deletes; IMAP syncs
DNS	53	names to IP addresses

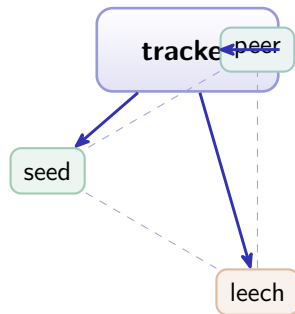
A fibre-optic cable carries all of this as light through thin glass strands

BitTorrent – sharing without a single server

BitTorrent is a **peer-to-peer** protocol.

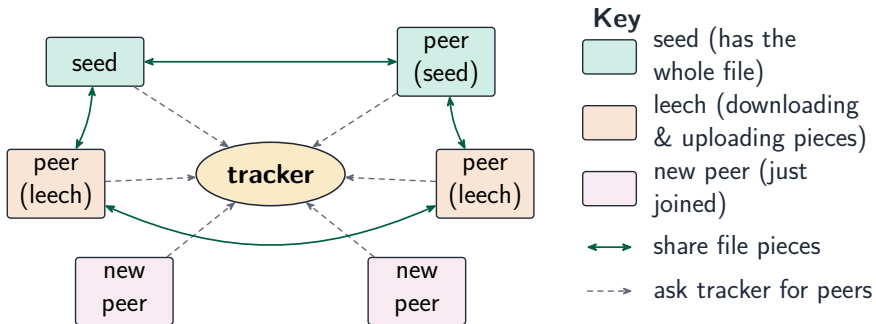
- a file is split into pieces
- a **tracker** helps peers find each other
- peers then share pieces directly
- no single server carries all the load

Seeds have the whole file; leeches are still downloading.



A BitTorrent swarm

Every peer that has a piece can serve it, so popular files get **faster** as demand grows.

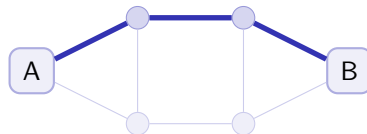




Circuit switching – one reserved path

Circuit switching sets up a **dedicated path** before any data is sent, reserved for the whole conversation, then released.

- reserved bandwidth, in-order delivery
- inefficient during silences
- slow to set up
- classic example: the traditional telephone network

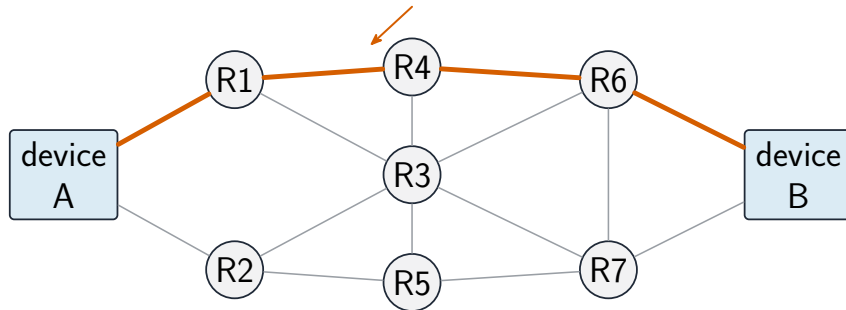


one path reserved

Circuit switching

A path is reserved for the whole call: guaranteed capacity, wasted when idle.

one dedicated path reserved for the whole call



Packet switching – independent packets

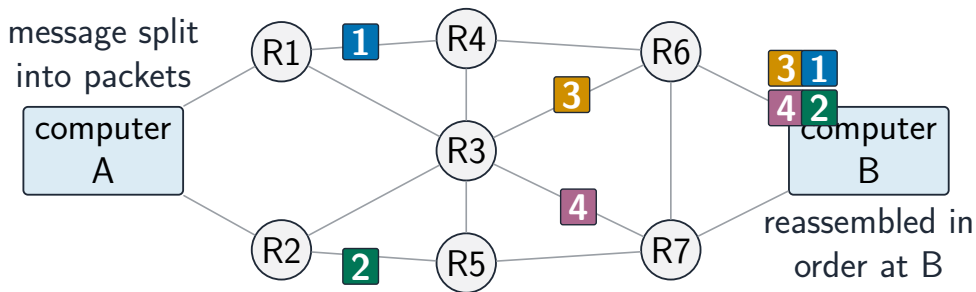
In **packet switching** the data is split into **packets**, each sent **independently**.

- each packet carries source, destination and a sequence number
- routers choose the next hop *per packet*
- packets may take different routes and arrive out of order
- the destination reassembles them

Efficient (links are **multi-plexed**) and robust (reroute around a failure). Variable **latency**; TCP handles loss.

Packet switching

Packets are routed independently and reassembled – resilient, shared, but variable in delay.



Circuit vs packet – the exam comparison

Aspect	Circuit	Packet
Path	dedicated, reserved	shared, per-packet
Setup	slow	none
Bandwidth	inefficient in silence	efficient
Order	in order	may be out of order
Failure	one cut kills the circuit	reroute around it
Suits	constant-rate voice	bursty web / email

Worked example – a call and a download

A phone call and a large file download share a network. Which switching method suits each, and why?

A **phone call** needs a steady stream with low delay – **circuit switching**: a dedicated path is reserved for the whole call.

A **file download** does not care about timing; the receiver reassembles it – **packet switching**: packets travel independently, each with addresses and a sequence number.

Name the *property of the traffic*: reserved capacity and low delay vs efficiency and resilience.



Topic 14 – key takeaways

1 Protocols

format, order, timing, errors – a shared language

2 TCP/IP

App / Transport / Internet / Link; TCP vs UDP

3 Application

HTTP(S), FTP, SMTP, POP3/IMAP, DNS

4 Switching

dedicated path vs independent packets

Check yourself

- 1 Which layer does IP sit in?
- 2 TCP or UDP for guaranteed, in-order delivery?
- 3 SMTP sends email – which protocols receive it?
- 4 One advantage of packet switching over circuit switching?

Answers 1. Internet
bandwidth efficiently 2. TCP 3. POP3 or IMAP 4. reroutes around failures / uses