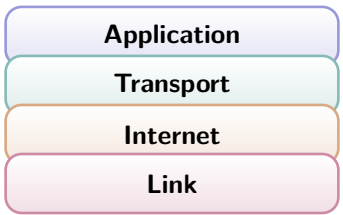
Communication & Internet Technologies

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

1 Why protocols

A protocol is a shared language

A **protocol** 协议 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

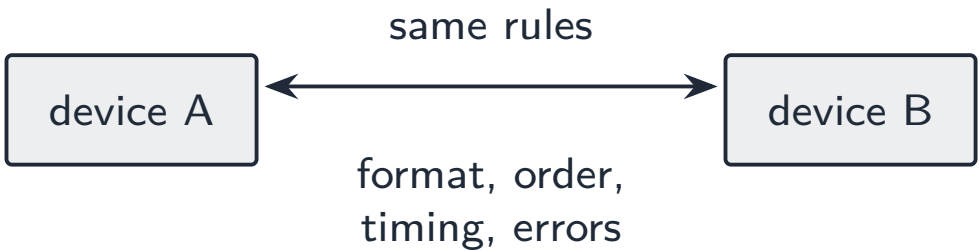
errors

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.

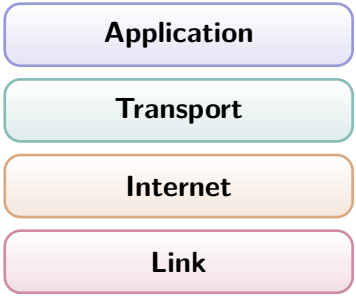


2 Layers

Why split a network into layers?

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

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



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

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

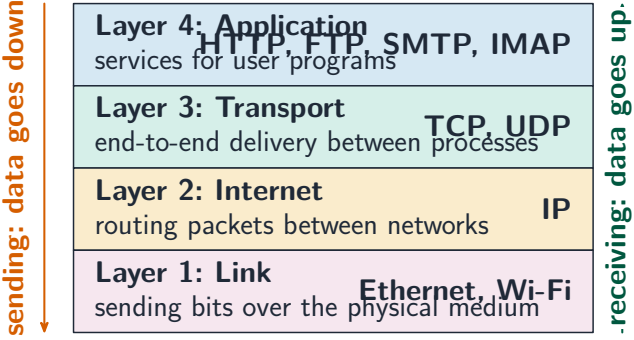
Link

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** 传输层 delivers data end-to-end between processes, identified by **port numbers** 端口号.

TCP

传输控制协议

connection-oriented

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

HTTP, HTTPS, SMTP, FTP

UDP

用户数据报协议

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** 网际协议 carries packets between hosts
- each packet has a source and destination **IP address** IP 地址
- **routers** 路由器 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 地址
- Ethernet uses **CSMA/CD** 载波侦听多路访问/冲突检测
- a **switch** 交换机 forwards a frame only to the correct port

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

Internet layer

A cable modem connects a home network to the internet provider

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

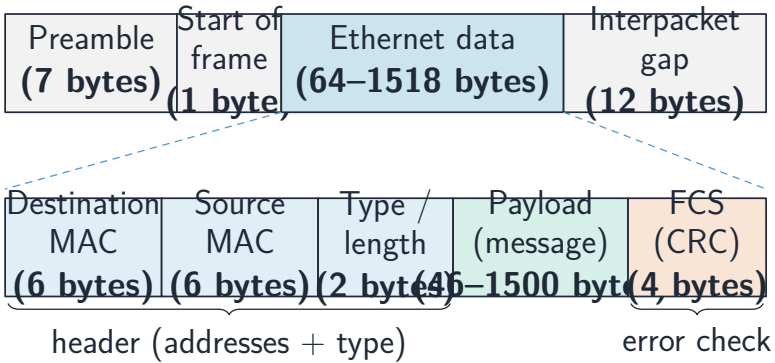
A network switch connects many wired devices on a local network

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

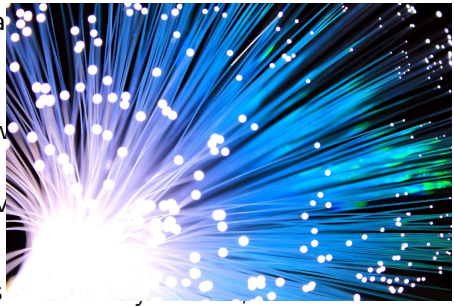
Addresses, payload and a check sequence – the link layer's envelope.



3 Application protocols

Common application-layer protocols

HTTP HTTPS	80 443	browsers fetch pa over TLS
FTP	21	transfer files betw
SMTP	25	send email; receiv IMAP
POP3 IMAP	110 143	POP3 downloads IMAP syncs
DNS	53	names to IP addr



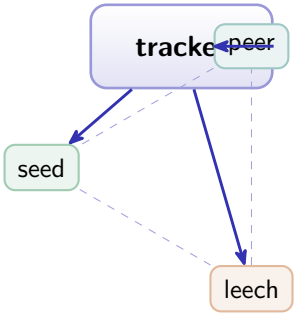
A fibre-optic cable: data travels as light through thin glass strands

BitTorrent – sharing without a single server

BitTorrent 比特洪流 is a **peer-to-peer** 对等网络 protocol.

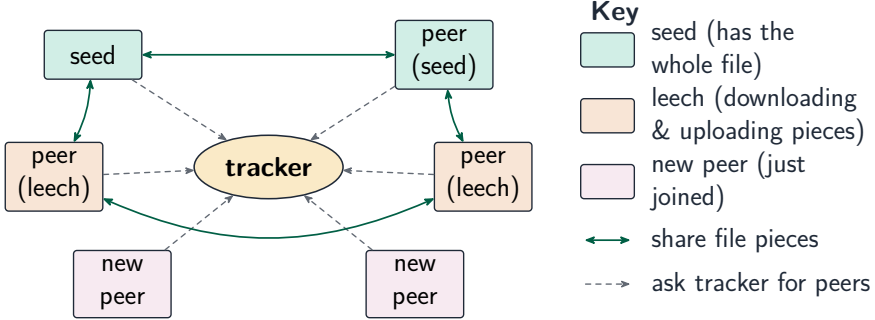
- a file is split into pieces
- a tracker helps peers find each other
- peers then share pieces directly
- no single server carries all the load

> 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.

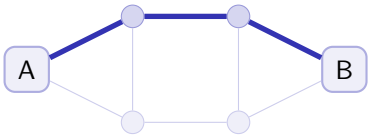


4 Switching

Circuit switching – one reserved path

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

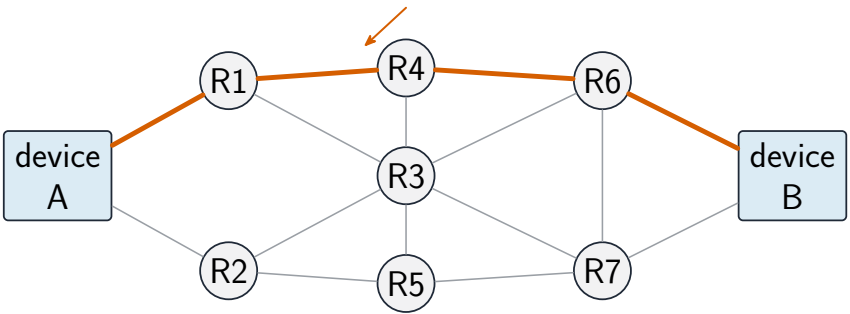
- reserved bandwidth, in-order delivery
- inefficient during silences
- slow to set up
- classic example: the traditional telephone network



one path reserved

Circuit switching

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



Packet switching – independent packets

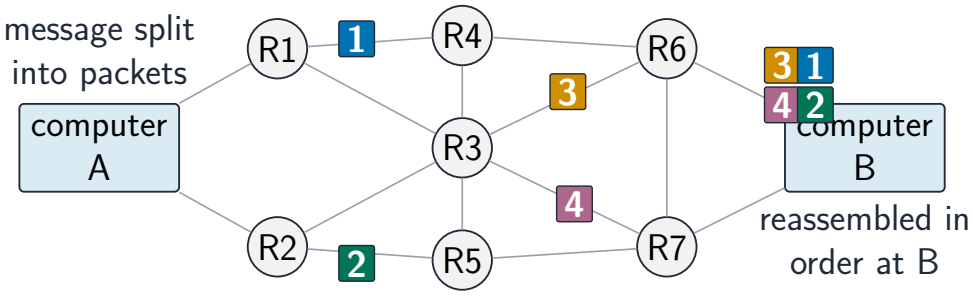
In **packet switching** 分组交换 the data is split into **packets** 数据包, each sent independently.

- 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 **multiplexed** 多路复用) and robust (reroute around a failure). Variable **latency** 延迟; 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.

5 Recap

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 **2.** TCP **3.** POP3 or IMAP **4.** reroutes around failures / uses bandwidth efficiently