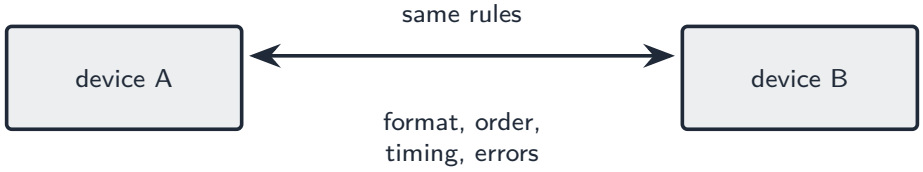
Communication and internet technologies

A-Level Computer Science

Why protocols are needed

A **protocol** 协议 is a **set of rules** for how devices **communicate**. Both ends must follow the same rules, or one side's signals are meaningless to the other. Protocols define the **format** of the data (where addresses and payload sit), the **order** of messages (who speaks first, when to acknowledge), the **meaning** of each message, the **timing** (timeouts, retransmits), and **what to do on error**. Without an agreed protocol, communication fails — like two people speaking different languages with no translator.



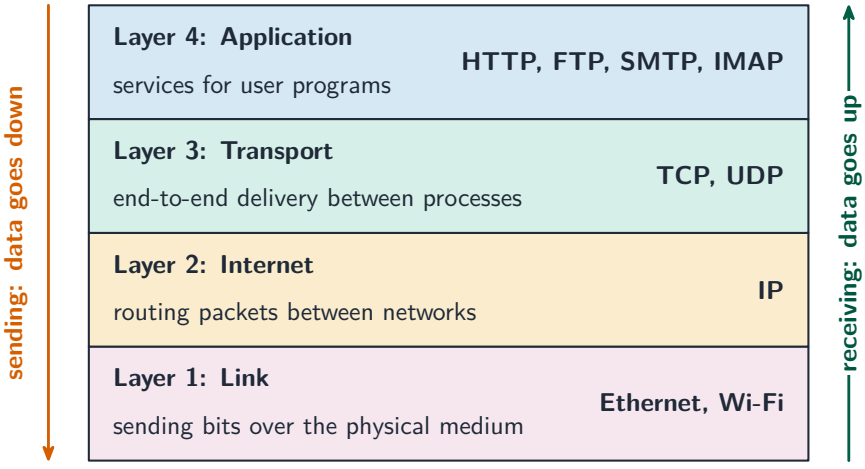
A protocol is the shared rules: format, order, timing and errors

Layered protocols

Networking is complex, so it is split into **layers** 层, each with one focused job, talking only to the layer above and below. Benefits: **modularity** 模块化 (replace one layer — say Ethernet with Wi-Fi — without touching the others), standardisation (vendors interoperate), and **abstraction** 抽象 (you ignore details handled elsewhere). The internet uses the **TCP/IP protocol suite** 协议栈 (4 layers).

TCP/IP protocol suite

Layer	Purpose	Examples
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	sending bits over the physical medium	Ethernet, Wi-Fi



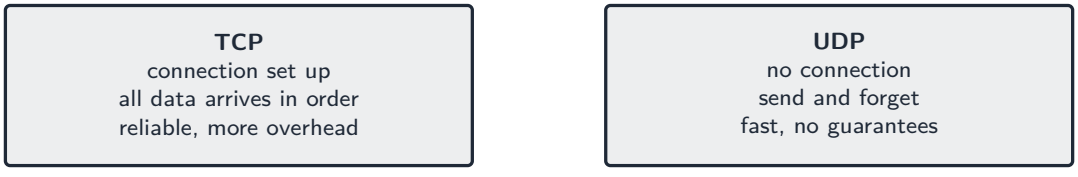
The four layers of the TCP/IP protocol suite

Application layer

The **application layer** 应用层 gives services to user programs and defines the protocols they speak (HTTP for web, SMTP for email). This is where a programmer most often works.

Transport layer

The **transport layer** 传输层 delivers data **end-to-end** between processes, identified by **port numbers** 端口号. Two protocols:



TCP connects and delivers in order; UDP sends and forgets

- **TCP** 传输控制协议 — **connection-oriented** 面向连接: sets up a connection, ensures **all data arrives in order**, retransmits lost **packets** 数据包, controls flow. Reliable but with overhead. Used by HTTP, HTTPS, SMTP, FTP.
- **UDP** 用户数据报协议 — **connectionless** 无连接: sends and forgets, with no acknowledgements or ordering. Low overhead, no guarantees. Used for streaming, DNS and gaming, where speed beats reliability.

Internet layer

The **internet layer** 网络层 carries packets between hosts using **IP**. Each packet has a source and destination **IP address** IP 地址, and **routers** 路由器 forward it onward. It does not guarantee delivery — that is TCP's job.

A home router does this job for your house: it reads each packet's destination address and sends it on towards the internet, and back to the right device.

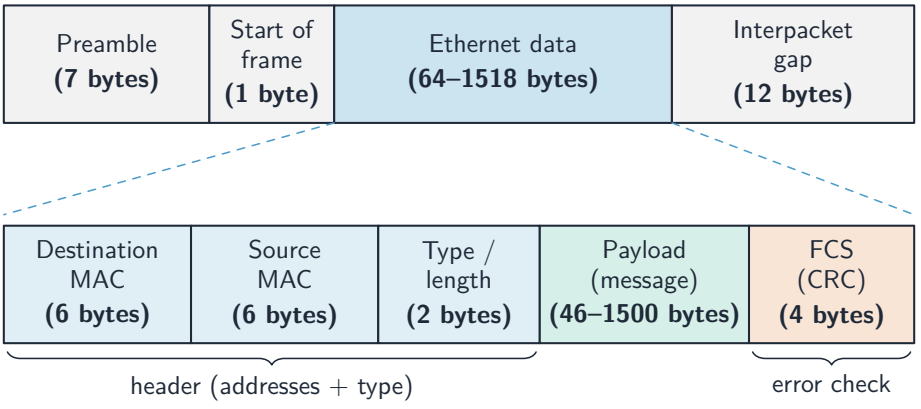
A home Wi-Fi router: it forwards packets between your devices and the internet

Before the router reaches the wider internet, a **modem** 调制解调器 connects the home to the internet provider over the provider's cable or phone line. Its lights show the link is up and online.

A cable modem connects a home network to the internet provider

Link layer

The **link layer** 链路层 sends bits over one physical link (Ethernet, Wi-Fi). It adds a frame header with **MAC addresses** MAC 地址 and handles medium access (e.g. **CSMA/CD** 载波侦听多路访问/冲突检测 on Ethernet).

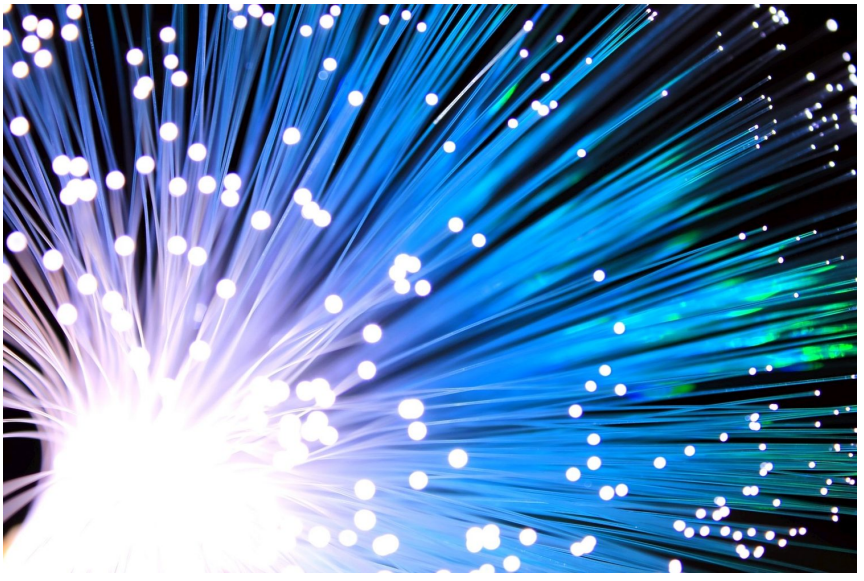


The parts of a typical Ethernet frame

On a wired local network, a **switch** 交换机 joins many devices together. Each device plugs into a port with an Ethernet cable (an RJ45 plug), and the switch uses the MAC addresses in each frame to send it only to the correct port.

A network switch connects many wired devices on a local network

The physical link can be a copper wire, a radio signal (Wi-Fi), or a **fibre-optic cable** 光纤. In a fibre-optic cable, the bits travel as flashes of light through very thin strands of glass, which is fast and carries data a long way.



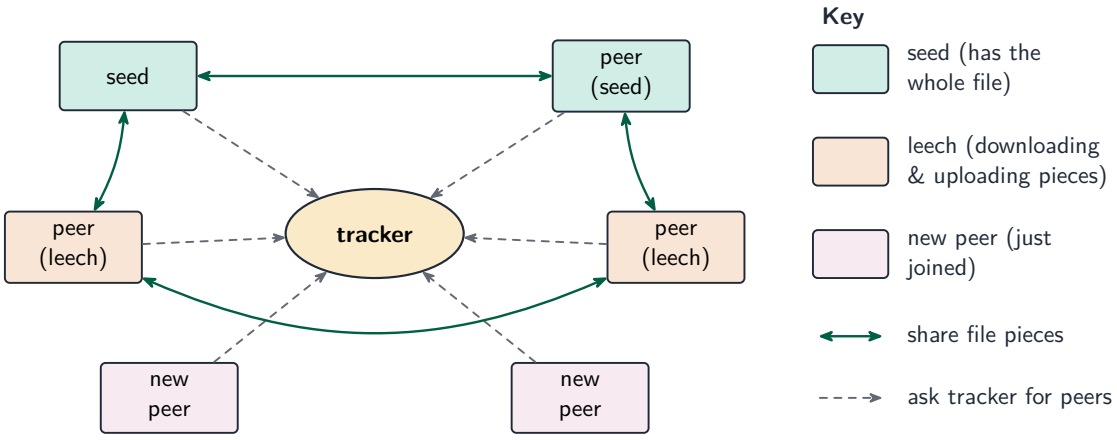
A fibre-optic cable: data travels as light through thin glass strands

A radio link can reach much further. A **satellite dish** 卫星天线 sends and receives radio signals to and from a satellite, carrying data to places that wired links cannot easily reach.

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

Common application-layer protocols

- **HTTP** 超文本传输协议 — browsers fetch web pages from servers (over TCP, port 80). **HTTPS** is HTTP over TLS — encrypted, port 443.
- **FTP** 文件传输协议 — transfer files between client and server.
- **SMTP** 简单邮件传输协议 — **send** email between client and server, and between servers. Receiving uses POP3 or IMAP.
- **POP3** — downloads email and usually deletes it from the server. **IMAP** — leaves email on the server and syncs across devices, so the same inbox appears everywhere.
- **BitTorrent** — a **peer-to-peer** 对等网络 protocol; a file is split into pieces downloaded from many peers in parallel, so no single server carries all the load.

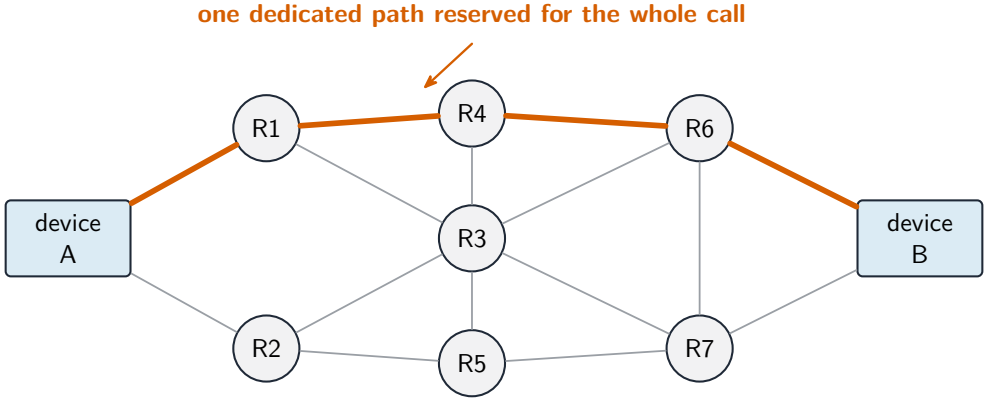


BitTorrent: a tracker helps peers find each other, then they share file pieces directly

Circuit switching vs packet switching

Circuit switching

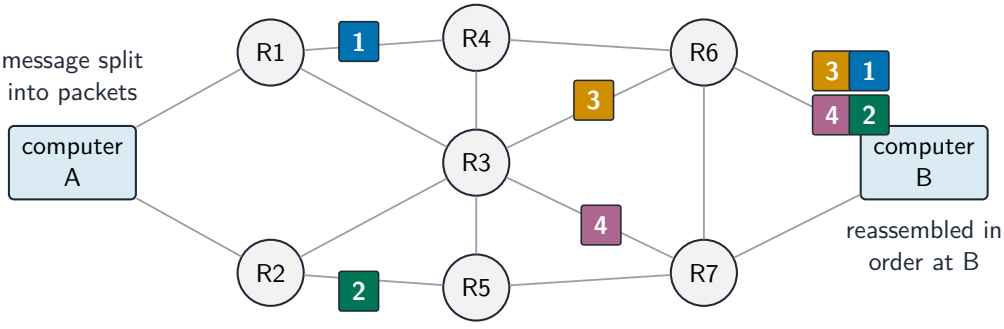
A **dedicated path** is set up between the two ends before any data is sent (**circuit switching** 电路交换), reserved for the whole conversation, then released. It gives **reserved bandwidth** 带宽 and in-order delivery, but is **inefficient** during silences and slow to set up. Classic example: the traditional telephone network.



Circuit switching: one dedicated path is reserved end to end

Packet switching

The data is split into **packets**, each sent **independently** (**packet switching** 分组交换). Each packet carries the destination address; routers make per-packet decisions, so packets may take different routes and arrive out of order, and the destination reassembles them. It is **efficient** (one link is **multiplexed** 多路复用 across many conversations), **robust** (reroute around a failure), but has **variable latency** 延迟 and possible loss (TCP handles reliability). Used by the internet.



Packet switching: packets travel independently and may take different routes

Aspect	Circuit switching	Packet switching
Path	dedicated, reserved	shared, per-packet
Setup time	slow	none
Bandwidth use	inefficient	efficient
Order	in order	may be out of order
Robustness	one failure cuts the circuit	reroute around failures
Suits	constant-rate flows (voice)	bursty flows (web, email)

Modern networks use packet switching for its efficiency and resilience.

Describing packet switching in a few sentences

A good exam answer: "The message is broken into small packets. Each packet carries the destination and source addresses and a sequence number. Each packet travels through the network independently, with routers choosing the next hop per packet. Packets may take different paths and arrive out of order. The destination uses the sequence numbers to reassemble the message, and missing packets can be requested again."

Worked example. 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, and it would suffer badly if pieces arrived late or out of order - so **circuit switching** suits it: a dedicated path is set up for the whole call and its capacity

is reserved for the duration. A **file download** does not care about timing or arrival order, because the receiver reassembles it, and it benefits from using whatever capacity happens to be spare - so **packet switching** suits it: the file is split into packets that travel **independently**, each carrying source and destination addresses and a **sequence number**, with routers choosing a next hop per packet. Name the **property of the traffic** that decides it: reserved capacity and low delay for the call, efficiency and resilience for the download.

Exam tips

- Explain **why protocols and layers** are used: each layer has one job and can change independently.
- Place common protocols in the **TCP/IP** stack (HTTP/FTP/SMTP application; TCP/UDP transport; IP internet).
- Compare **circuit switching vs packet switching** (a dedicated path vs independent packets) with a use for each.