EXTENSION · OPTIONAL · AS LEVEL

Topic 2: Communication

What really happens between pressing Enter and a web page appearing.

> **Optional stretch material.** This goes beyond the 9618 syllabus, so you will not be asked about it directly in the exam. Its purpose is to deepen ideas you already know. **Builds on:** 2.1 Networks (protocols, the Internet). Try the challenge questions on the last pages before you read the answers.

Layers and encapsulation

Network software is split into **layers**. Each layer does one job and wraps the data from the layer above in its own **header**, like putting a letter inside an envelope, then inside a parcel. The TCP/IP model has four layers:

Layer	Unit of data	Header adds	Example
Application	Message	Protocol commands	HTTP, SMTP, DNS
Transport	Segment	Port numbers, sequence numbers	TCP, UDP
Internet	Packet	Source and destination IP address	IP, routers
Link	Frame	MAC addresses, error check	Ethernet, Wi-Fi

Why bother? Each layer can be replaced without touching the others: HTTP works the same over Wi-Fi, Ethernet or mobile data. A router only needs to read as far up as the Internet layer. (The 7-layer OSI model splits the same job more finely.)

TCP: building reliability from an unreliable network

IP packets can be lost, duplicated or arrive out of order. **TCP** repairs this. Before any data moves it runs a **three-way handshake**:

```
Client --> Server   SYN        "I want to talk; my first sequence number is x"
Client <-- Server   SYN-ACK    "OK; mine is y, and I expect x+1 next"
Client --> Server   ACK        "Got it; I expect y+1 next"
```

Every segment carries a **sequence number**. The receiver acknowledges what has arrived; if the sender gets no acknowledgement before a timeout it **retransmits**. Sequence numbers also let the receiver put segments back in order.

UDP skips all of this: no handshake, no retransmission, much lower delay. It suits traffic where a late packet is worthless: live video calls, online games, DNS lookups.

NAT: why a billion devices share a few addresses

IPv4 has only $2^{32} \approx 4.3$ billion addresses, fewer than the number of devices. **Network Address Translation** lets every device in a home use a **private** address (such as 192.168.1.20) while the router shares one **public** address.

When your laptop contacts a web server, the router rewrites the packet's source to its own public address and a free port, and records the pairing in a **NAT table**. Replies are matched against the table and sent back to the right device.

A side effect: an *unsolicited* inbound packet matches no table entry and is dropped, a crude but useful firewall. Hosting a server at home therefore needs **port forwarding**, a permanent manual NAT entry.

Challenge questions

1. A 10 000 000-byte file is sent over TCP. Each segment carries at most 1460 bytes of data. How many segments are needed at minimum?
2. A laptop behind a home router can browse any website, yet a game server running on the same laptop cannot be reached from the Internet. Explain both observations.
3. A video-conferencing app uses UDP rather than TCP. Explain why, and say what the app can do when a packet is lost.

Worked answers

Only look once you have had a proper go at each question.

1. 10 000 000 ÷ 1460 = 6849.3, so **6850 segments** (6849 full segments carry 9 999 540 bytes; one final segment carries the remaining 460).

2. Browsing starts from inside: the router creates a NAT table entry when the request leaves, so replies match it and are forwarded. A stranger's inbound connection matches no entry, so the router drops it. Port forwarding adds a permanent entry that sends traffic arriving on the game's port to the laptop's private address.

3. A retransmitted frame arrives too late to be useful, and TCP would hold back later data until the missing segment is recovered (stalling the call). With UDP the app simply carries on: it can skip or conceal the missing frame, use forward error correction, or lower the quality.