Computer Networks

Curated computer science study track (not a degree, no licensure claim). Original explanations and practice only, following the reviewed course handout.

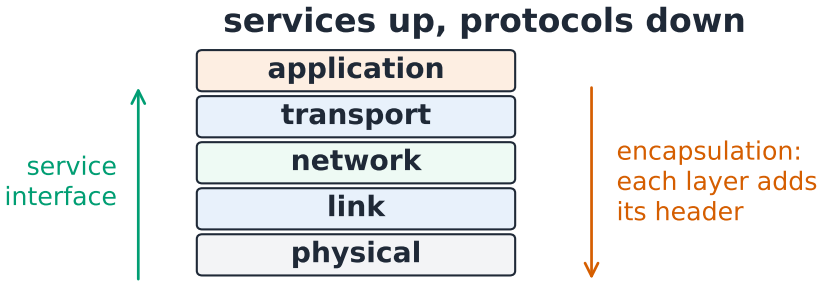
A message across a planet of strangers

- 9.1-9.3 move a frame (layers, **link** and IP, routing); 9.4-9.5 make it reliable and useful.
- 9.6-9.7 secure and scale it; every header here is decoded byte by byte in the suite.

Layering: services up, protocols down

1. **Stack Physical**, **link**, network, **transport**, application - each offers a service up and consumes one down.
2. **Bytes** A 100-byte payload leaves with 54 bytes of headers: 14 + 20 + 20, one per layer crossed.
3. **Shape** Narrow waist (IP), wide top and bottom: applications changed and links changed; IP did not.

Layer stack



a 100-byte payload gains 54 bytes of headers: 14 + 20 + 20

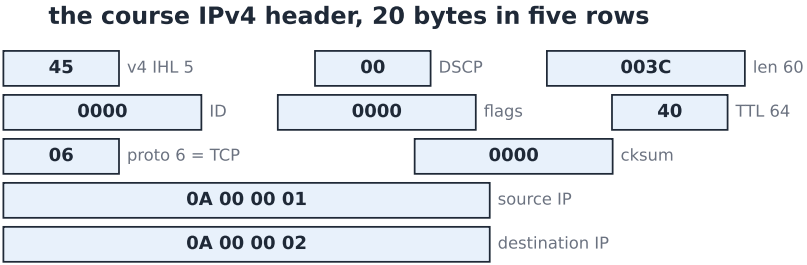
the hourglass: one narrow waist (IP), wide top, wide bottom

> Each layer adds a header going down and strips it coming up.

Link and IP: neighbour versus stranger

1. **Frame** Ethernet: 6-byte destination and source MACs, type, payload, **CRC** - the CRC guards the whole frame **link** by link.
2. **Header** 45 00 003C ... 40 06: IHL 5 words = 20 bytes, total length 60 (20 + 40 payload), **TTL** 64, protocol 6 = TCP.
3. **Scope** The IP checksum guards only the header: routers rewrite TTL but never touch the data.

Packet header layout



IHL 5 counts 32-bit words: 5 x 4 = 20 header bytes; total length 60 = 20 header + 40 payload

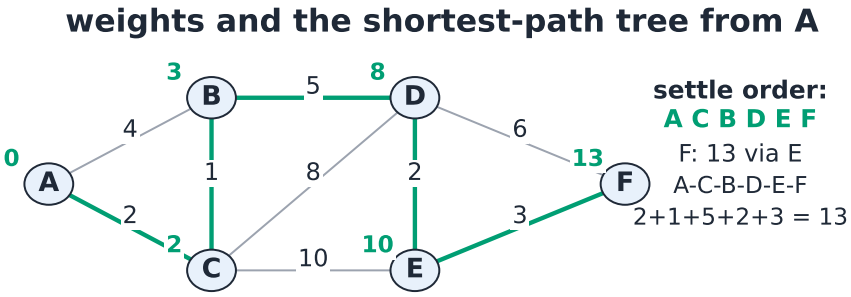
MAC names a neighbour; IP names the stranger

The IPv4 header decoded field by field: total length 60, **TTL** 64, protocol 6.

Routing: longest prefix, shortest path

1. **Match** 10.1.2.200 with /8, /16 and /24 entries present goes to the /24; 10.9.9.9 falls to the /8; the default catches the rest.
2. **Tree** Course weights from A settle C 2, B 3, D 8, E 10, F 13 - F via A C B D E F at $2+1+5+2+3 = 13$, beating D direct at 14.
3. **Aggregation** One /12 announcement covers 2^20 addresses: the table stays human-sized because host bits are thrown away.

Routing graph



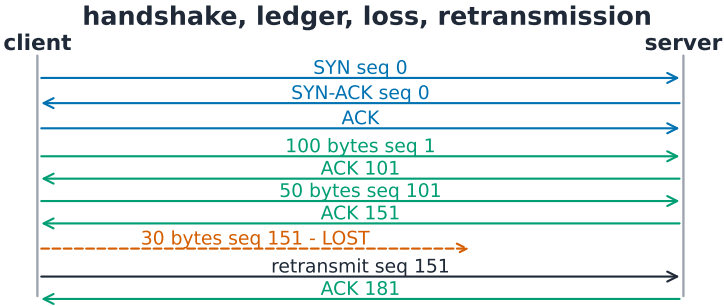
longest prefix match picks the /24 over the /16 over the /8; the default catches the rest

The course graph with weights; Dijkstra from A settles C, B, D, E, F.

Transport: the ledger and the postcard

1. **Ledger** SYN 0; 100 bytes: seq 1, ACK 101; 50 bytes: seq 101, ACK 151; a lost 30 at 151 holds the ACK until retransmission, then 181.
2. **UDP** 8 bytes - ports, length, checksum: **DNS** fits in one datagram and re-asks if lost; video tolerates loss, not delay.
3. **Windows** Two different windows: the receiver buffer and the network state - **TCP** guarantees order and delivery, never latency.

TCP sequence diagram



cumulative ACKs name the next byte expected; the gap holds the ACK number until the timeout retransmits

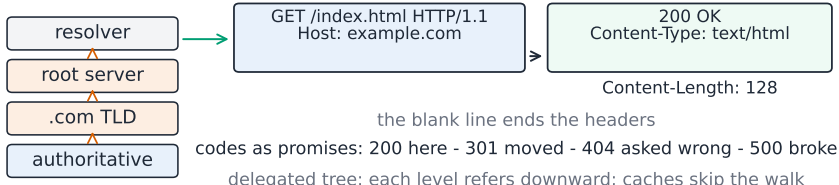
> Handshake, two segments with cumulative ACKs, a loss retransmitted after timeout.

Applications: names and pages

1. **DNS** Root refers to the TLD, the TLD to the authoritative server - and caching skips the walk for the answer **TTL**.
2. **HTTP** GET with a Host header; 200 OK with Content-Type and Content-Length 128, blank line, exactly 128 body bytes.
3. **Codes** 200 here it is; 301 moved (Location); 404 asked wrong; 500 broke - a crawler treats each differently.

Message flow

resolve the name, then speak HTTP



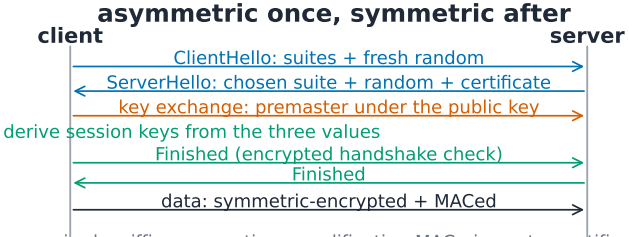
DNS: delegation + cache - HTTP: request, headers, blank line, body

A **DNS** delegation walk, then an **HTTP** exchange ending 200 OK.

Security: three properties, named attacks

1. **Handshake** ClientHello and ServerHello with fresh randoms and a certificate; the premaster travels under the public key; both derive session keys; Finished verifies.
2. **Pairing** Sniffing defeated by encryption; alteration by the **MAC**; impostors by certificates; replay by the fresh randoms - each attack names its defense.
3. **Honest** HTTPS makes the channel private and the server identified - the content can still lie.

TLS flow



attacks named, defenses paired: sniffing-encryption - modification-MAC - impostor-certificate - replay-fresh randoms

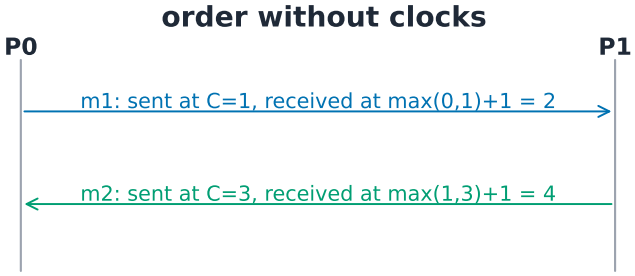
confidentiality + integrity + authentication, each defended by a named mechanism

> ClientHello, certificate, key exchange, both Finished - then **symmetric** data.

Distributed: order without clocks

1. **Clocks** P0 sends at C 1; P1 receives at max(0,1)+1 = 2; P1 sends at 3; P0 receives at max(1,3)+1 = 4 - causality preserved, time not used.
2. **One-way** If a **happens-before** b then C(a) < C(b); the converse is false - equal clocks imply nothing.
3. **Retries** deduct(user, 25, id=xyz) twice deducts once: the ledger records xyz as done - idempotent by key, not by hope.

Message ordering diagrams



causality only: if a happens-before b then C(a) < C(b) - the converse is false

idempotent operations plus a dedupe ledger make retries safe; a timeout means unknown, not failed

> Lamport counters across two processes: 1, 2, 3, 4 with causality intact.

Check yourself

- Headers around 100 payload bytes (Ethernet+IP+**TCP**)…? (54.)
- IHL 5 means a header of…? (20 bytes.)
- Course graph: F settles at…? (13.)
- P1 sends at C 3 to P0 at C 1: P0 becomes…? (4.)