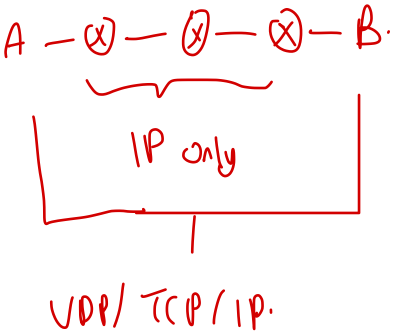
Transport supports Application layer.

eg. | TCP | HTTP |

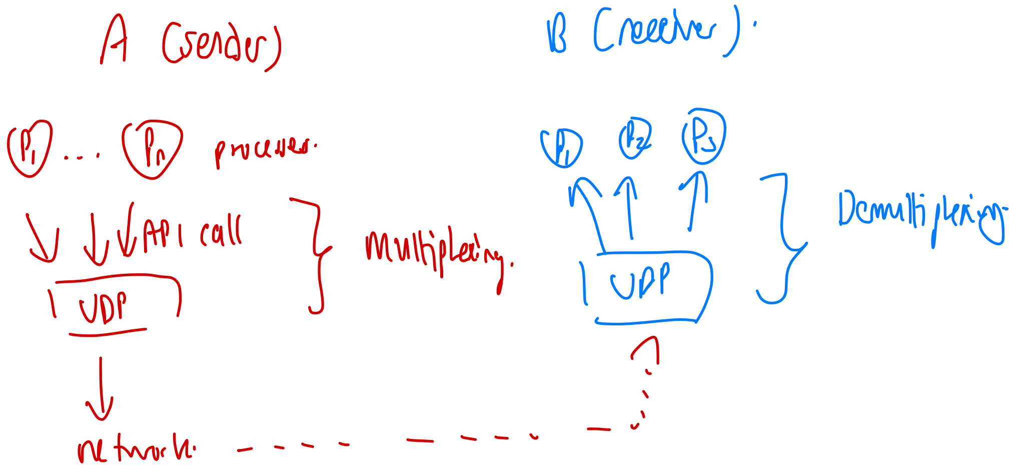
2 protocol @ Transport. <
- UDP
- TCP.

Only in hosts, not routers.



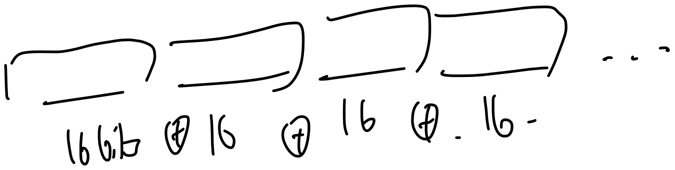
unreliable.
- unable to recover if packet lost.

UDP (simple) — Checksum — bit error detection.
- multiplexing



UDP packet — Header <
- Source, dest port
- length, checksum.

Checksum.



→ Wrap around carry.

→ Flip around.

Reliable Data Transfer. (rdt).

→ Corrupt, dropped, ~~re-ordered~~, long delays in packets.
(lost)

deals w/ packet corruption.

→ 1.0, 2.0, 2.1, 2.2, 3.0...

1.0 ↓ no error ↓ nothing

2.0 ↓ data bit error ↓ Checksum/ ACK/NAK.

2.1, 2.2 ↓ + ACK/NAK bit error. ↓ + sequence number ↓ + NAK free | Simplifies design.

3.0 ↓ + packet loss. ↓ + timeout/re-transmission.

Bit Error.

→ Receiver detect → Checksum.

→ Receiver feedback → Recover

ACK ↓ OK.

NAK. ↓ corrupted.

Corrupted? ⇓ duplicated packet. ‖

Rdt 2.0.

⇓

every packet a sequence no.

⇓

rdt 2.1

rdt 3.0

↳ packet may be lost.

↓

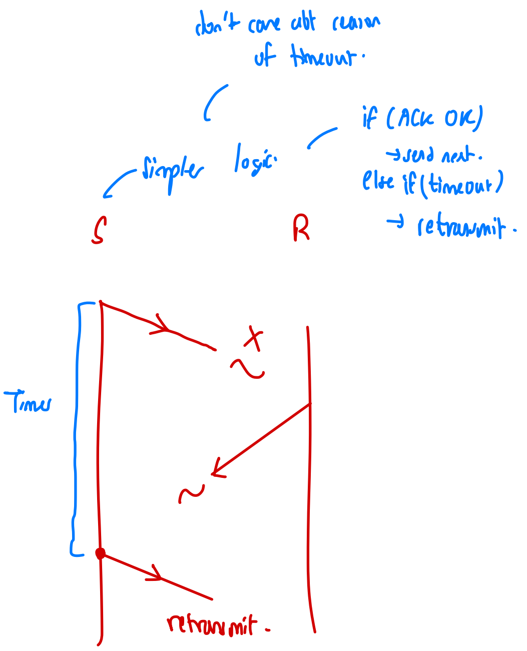
set a timer for every packet. very powerful.

↓

if no ACK, assume lost.

↓

retransmit upon timeout.


don't care abt reason of timeout.
simpler logic.
if (ACK OK) → send next.
else if (timeout) → retransmit.
S
R
Timer
retransmit.

All rdt protocals are home-made (not used in practice) even for GBA / SR.

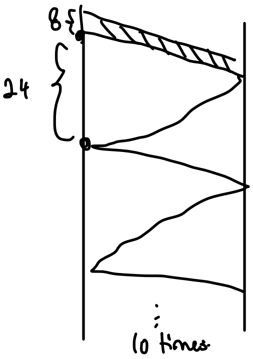
⇓

Foundation for TCP

DNS query is directly encapsulated in UDP segment

... | IP | UDP | DNS | ...

Stop & wait



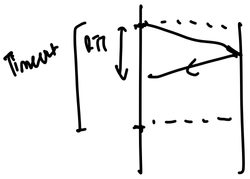
$d_t = \frac{8000 \text{ bits}}{10^6} = 8 \times 10^{-3}$

$10 \times (8 + 24) = 320$ ms.

Sequence no. → feedback corrupted → duplicate data packet.

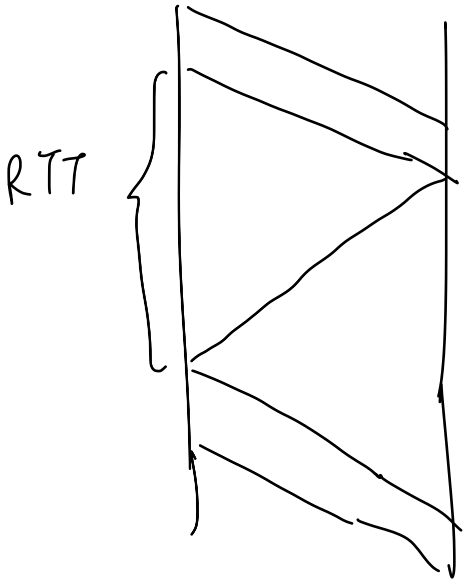
RTT known → timer set properly → no long delay / pre-mature timeout.

→ no duplicate packet.



rdt 3.0 :

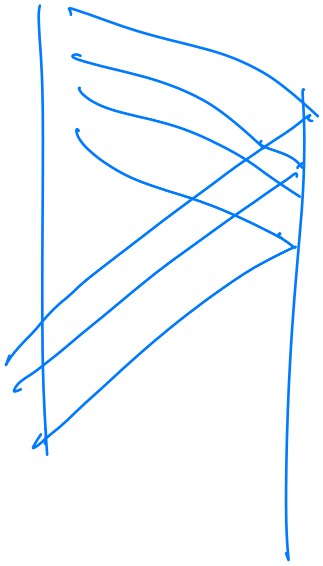
Stop & wait Operation



⇒ very slow.

∴ pipeline for multiple packets.

Pipelining



Go-Back-N - GBN.

Selective Repeat - SR.

Send up to N packet before ACK returned.

GBN.

- One timer for oldest unACK packet.
- Retrans everything once run out.
- Culmulative ACK — Once receive ACK 4, all 1, 2, 3 are received.
- Slightly simpler.
- Discards out-of-order packets.

SR

- N timers → a timer for each packet.
- Retrans that packet.
- Individual ACK.
- More efficient. — buffers out-of-order packets

TCP — built on top of previous protocols.

- point-to-point → One socket for one client
  → one-to-one transmission
  → label every packet w/ seq. no.
- Connection oriented → 3 way hand-shake
- Fully duplex — send & receive simultaneously. (bi-directional flow)

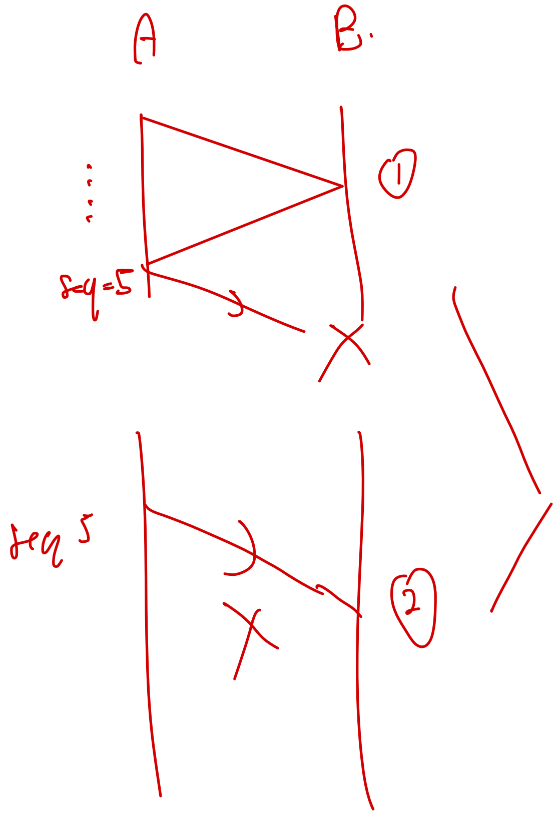
· in-order byte stream

sequence no. to label bytes.

diff. from GBN or SR.

first byte of data in a segment.

⇒ initial seq. no. is not always 0.



second seq 5.
retransmission or not?

⇓

∴ not always start from 0.

∴ In practice ⟹ random seq no.

⟹ both session unlikely to use same seq. no.

· Cumulative ACKs.

· Shake hand. → agree on connection & exchange connection parameters.

→ no data on first 2 handshake.

· TCP packet can be reorder.