

CS144: TCP

Datagrams

- IP's service is **Internet datagrams**
 - unreliable, **best-effort** delivery between two hosts
- UDP's service is **user datagrams**
 - unreliable, **best-effort** delivery between two user-space (unprivileged) programs
- Both of these are **unreliable!** Datagram can be:
 - Lost
 - Delivered more than once
 - Delivered out of order
- But don't have to worry about ~~truncation~~ and ~~corruption~~, because of tools:
 - Header checksum (IP)
 - Data checksum (UDP)

What if we want a “reliable” exactly-once datagram?

- Basic approach: send the datagram over and over again, until acknowledged
 - Tools:
 - Acknowledgment (from receiver to sender)
 - Timer and timeout (at sender)
 - Retransmission (by sender)

What if we want a “reliable” large message?

- Basic approach: break message into numbered **segments**
- Send each segment as a reliable datagram
- Tools:
 - Acknowledgment (from receiver to sender)
 - Timer and timeout (at sender)
 - Retransmission (by sender)
 - Sequence number

Web transaction
(reliable GET/POST of a URL)

HTTP

Internet stream socket (TCPSocket)
(reliable byte-stream between two programs anywhere in the world)

TCP

Internet datagrams
(unreliable packets between two computers anywhere in the world)

IP

Frames (unreliable packets within one network)

Ethernet

Reality
(electrons, photons, voltages)

Web transaction
(reliable GET/POST of a URL)

HTTP3

Reliable byte stream

QUIC

User datagrams

(unreliable packets between two programs anywhere in the world)

UDP

Internet datagrams

(unreliable packets between two computers anywhere in the world)

IP

Frames (unreliable packets within one network)

Ethernet

Reality

(electrons, photons, voltages)

TCP: reliable bytestreams over datagrams

Basic rules of TCP:

- ByteStream in each direction
- **Every** byte tagged with its place in sequence. Also:
 - Beginning of stream counts as one place in sequence
 - End of stream also counts as one
- Receiver tells sender:
 - next byte it needs to reassemble
 - how many more bytes it's willing to accept

Format of each TCP segment

- Sender-to-receiver information:
 - Index of the first byte in the segment (sequence no.)
 - **SYN bit:** is this the beginning of the stream?
 - Contents of segment
 - **FIN bit:** is this the end of the stream?
- Receiver-to-sender information:
 - Index of the next byte needed to assemble (ack no.)
 - # of bytes beyond this willing to accept (window size)