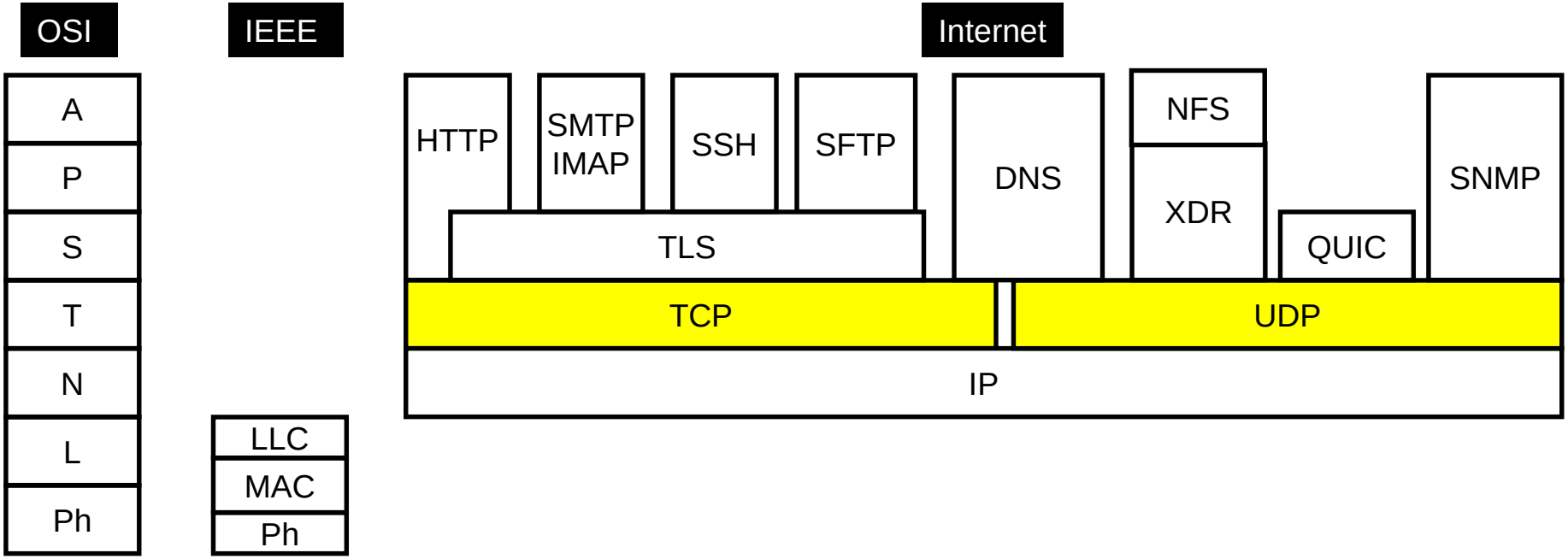
University of St Andrews

CS2003: Internet and the Web

TCP & UDP:
Service, protocol behaviour, and operation

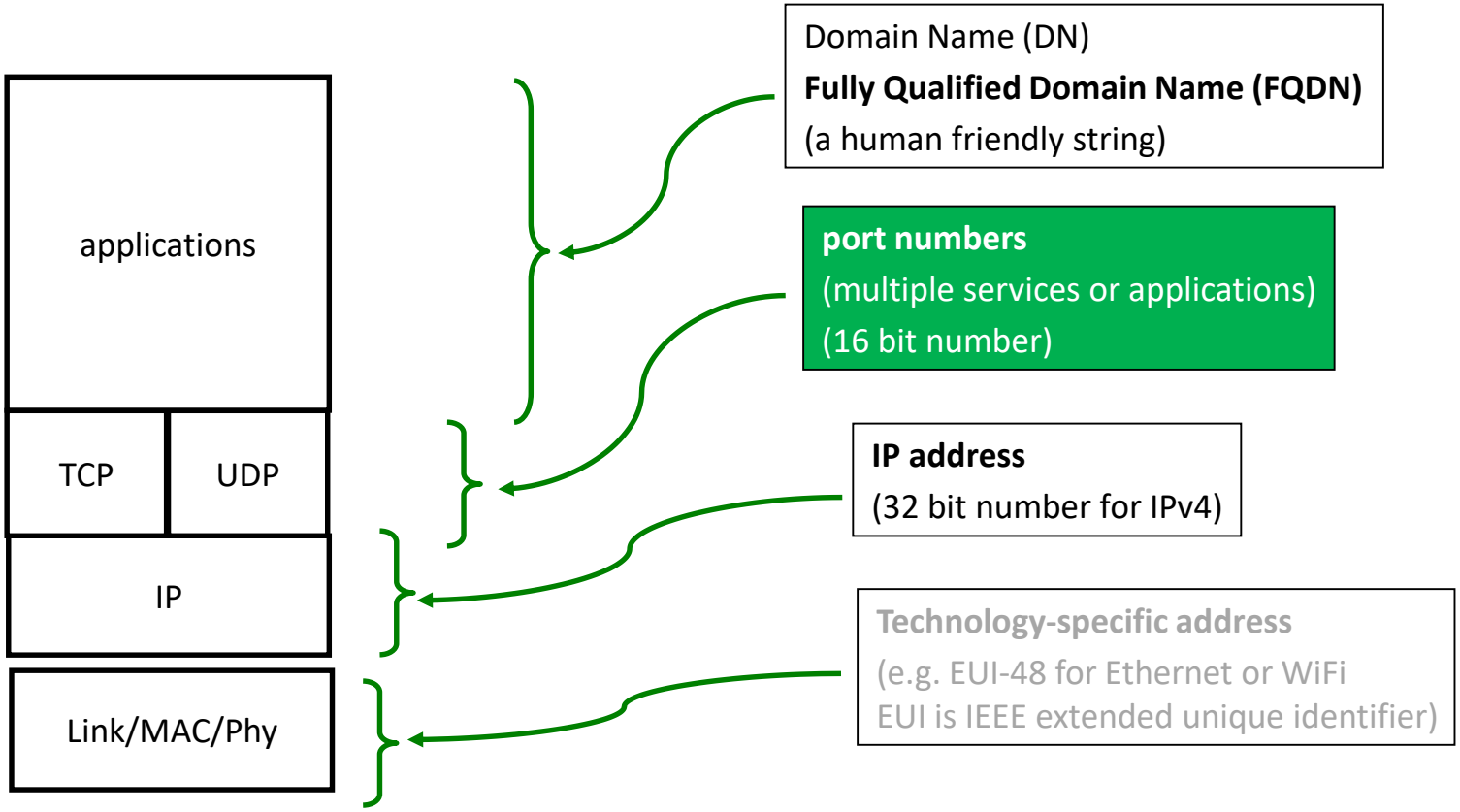
Transport layer protocols



DNS	Domain Name Service
HTTP	HyperText Transfer Protocol
SFTP	Secure File Transfer Protocol
IP	Internet Protocol
IEEE	Institute of Electrical and Electronic Engineers
LLC	Logical Link Control
MAC	Medium Access Control
NFS	Network File System
OSI	Open Systems Interconnection

POP	Post Office Protocol
QUIC	Quick UDP Internet Connections
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SSH	Secure Shell
TCP	Transmission Control Protocol
TLS	Transport Layer Security (aka SSL)
UDP	User Datagram Protocol
XDR	eXternal Data Representation

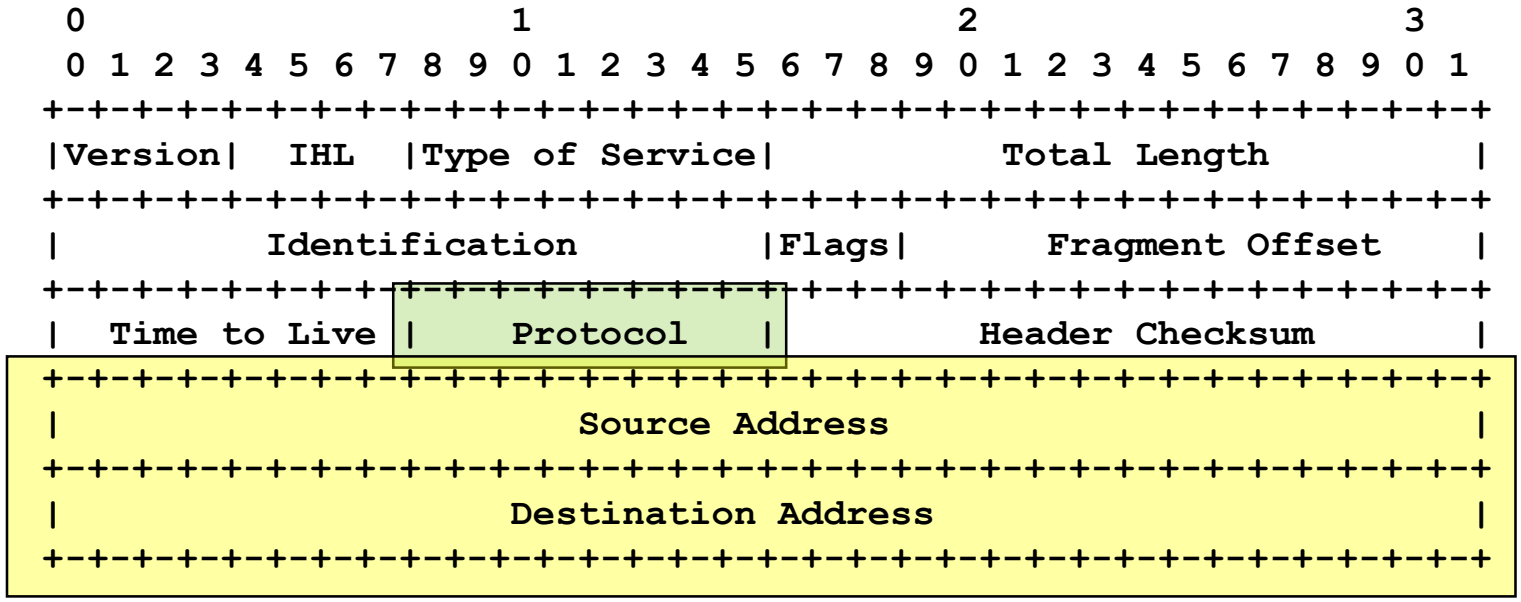
Naming and Address Bindings



A FQDN is looked up in a global directory service (the **Domain Name System**) and a corresponding IP address is found.

Strictly speaking, and IP address identifies an **interface on a host** (such as an Ethernet interface or wireless LAN interface) and **not** the host itself.

IPv4 header

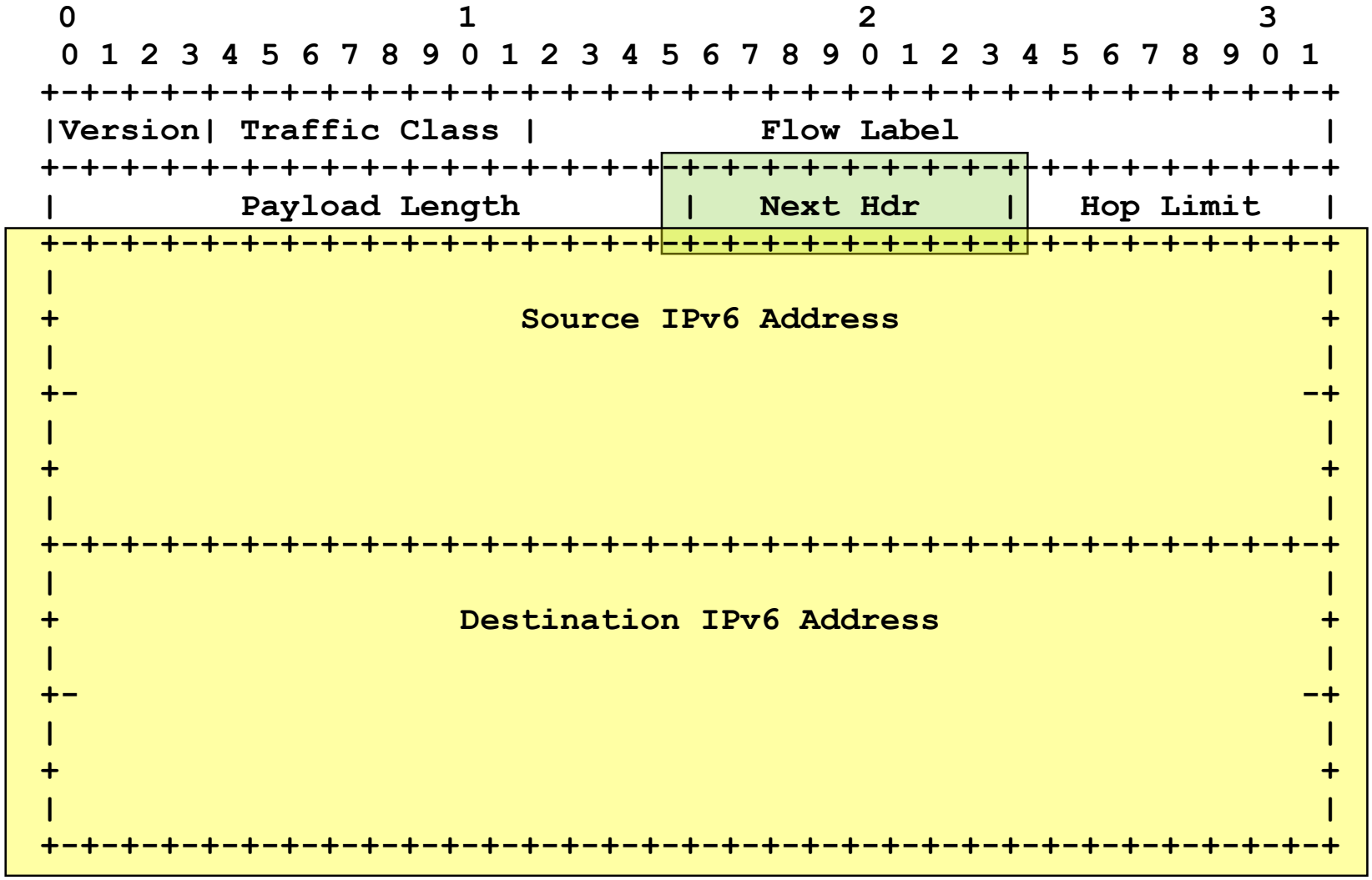


IP Header Format (from RFC791)

TCP protocol number 6 (0x06) (from RFC793)

UDP protocol number 17 (0x11) (from RFC768)

IPv6 header



(from RFC8200)

TCP and UDP

University of St Andrews

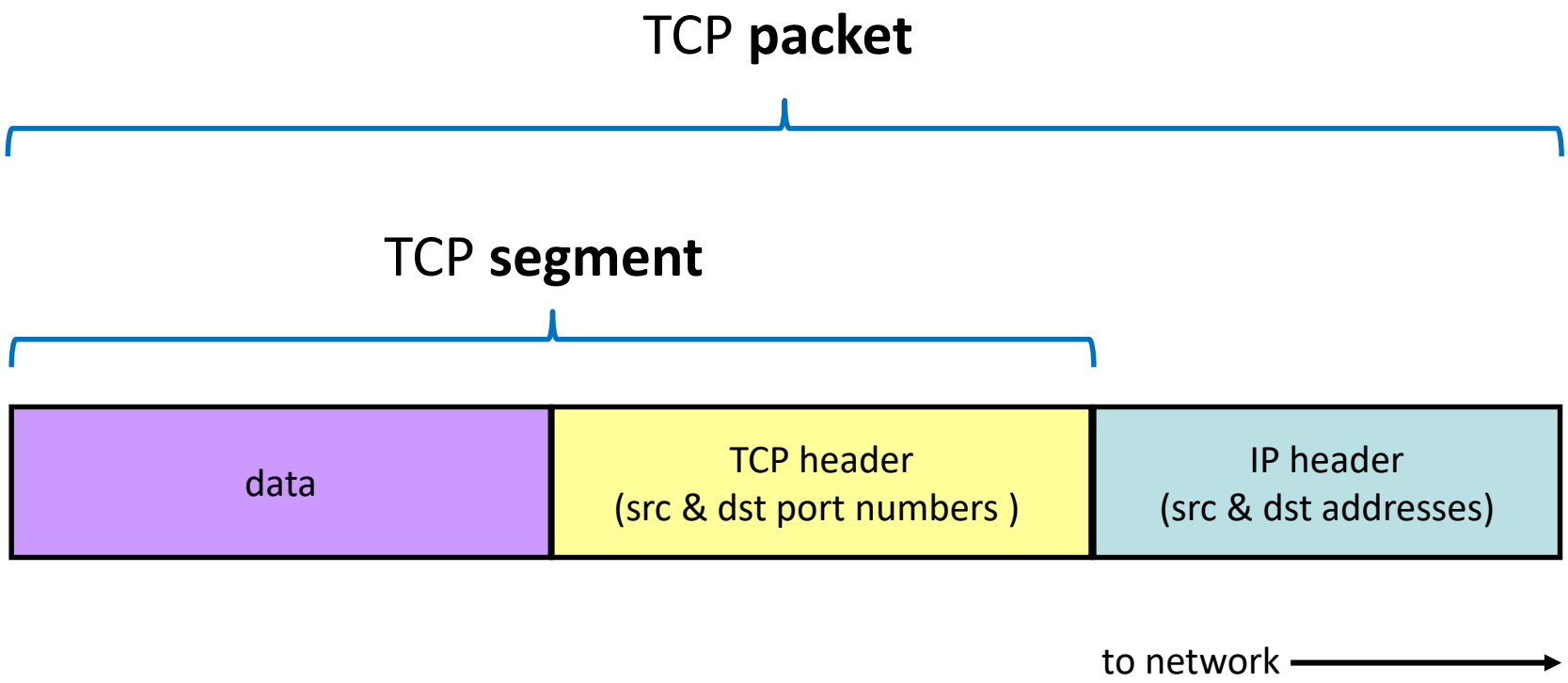
1. **TCP overview.**
2. UDP overview.

TCP protocol

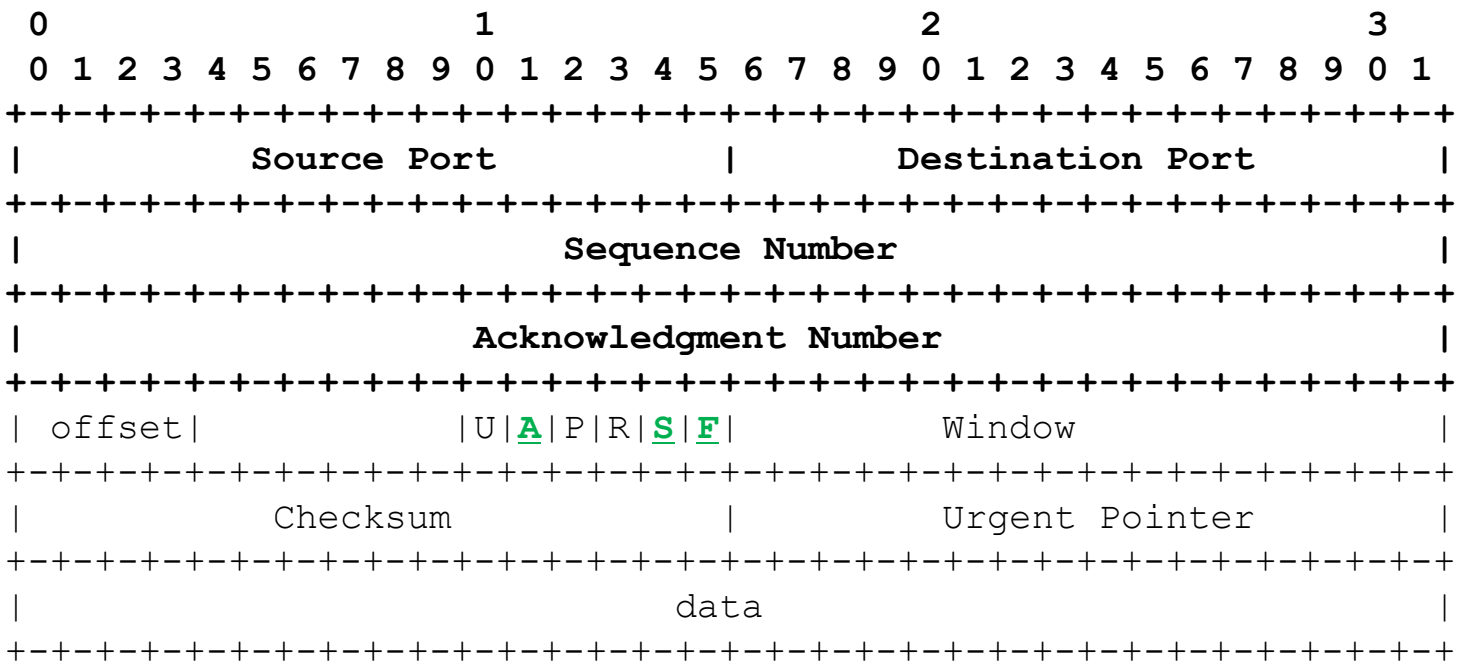
University of St Andrews

- Segment format:
  - Header: source and destination port numbers.
  - Payload: data from layer above.
- Service description:
  - connection-oriented (CO), ordered, confirmed, reliable: **byte-stream service.**
- Protocol:
  - **data structure** – segment header format.
  - **algorithm** – FSM for protocol state, plus various algorithms for reliability, flow control, congestion control (plus others …)

Simplified TCP packet



TCP header



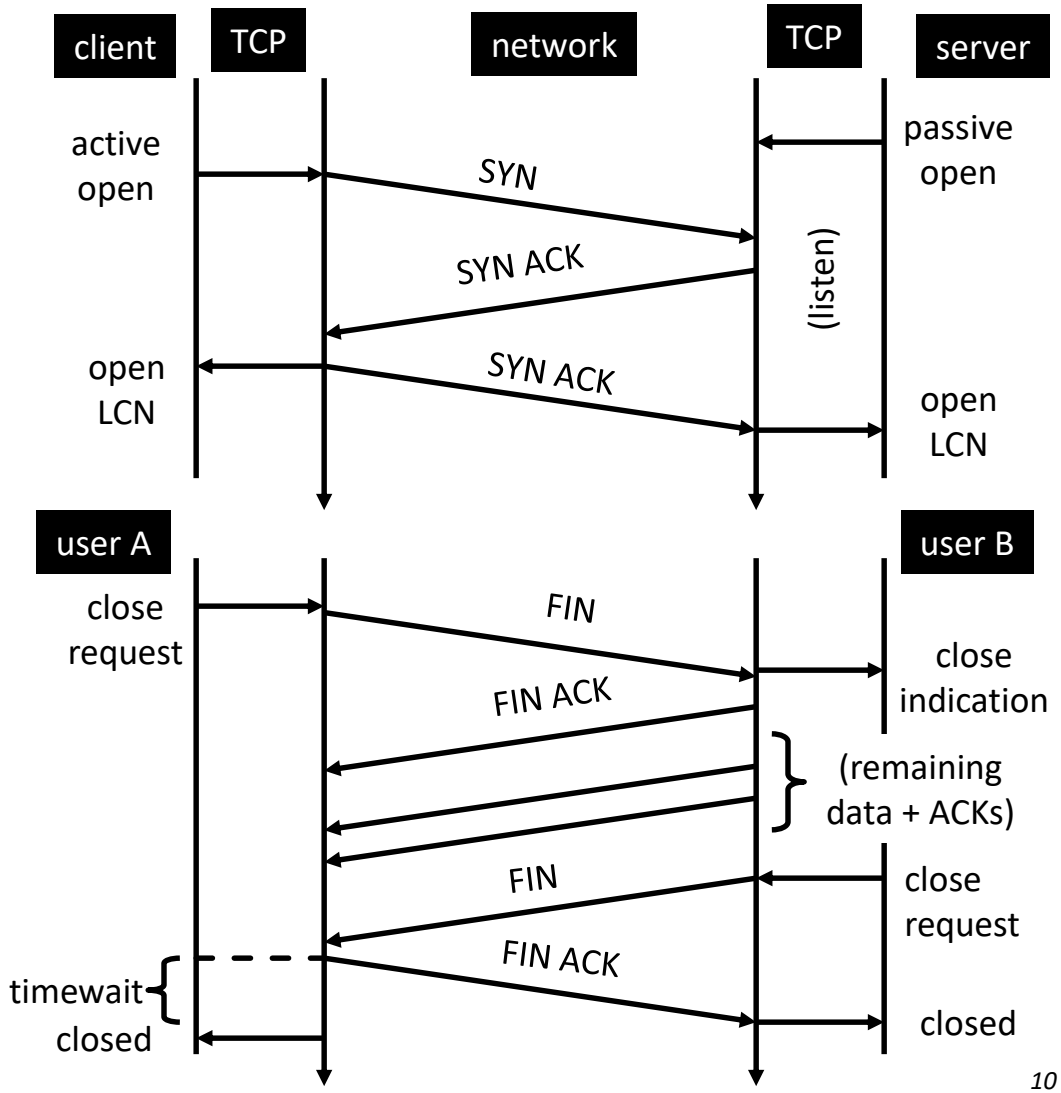
TCP Header Format (from RFC793, p14)

Flags

URG	urgent pointer significant	RST	reset connection (abortive)
ACK	**acknowledgement significant**	**SYN**	**synchronise sequence numbers**
PSH	push function	**FIN**	**no more data from sender**

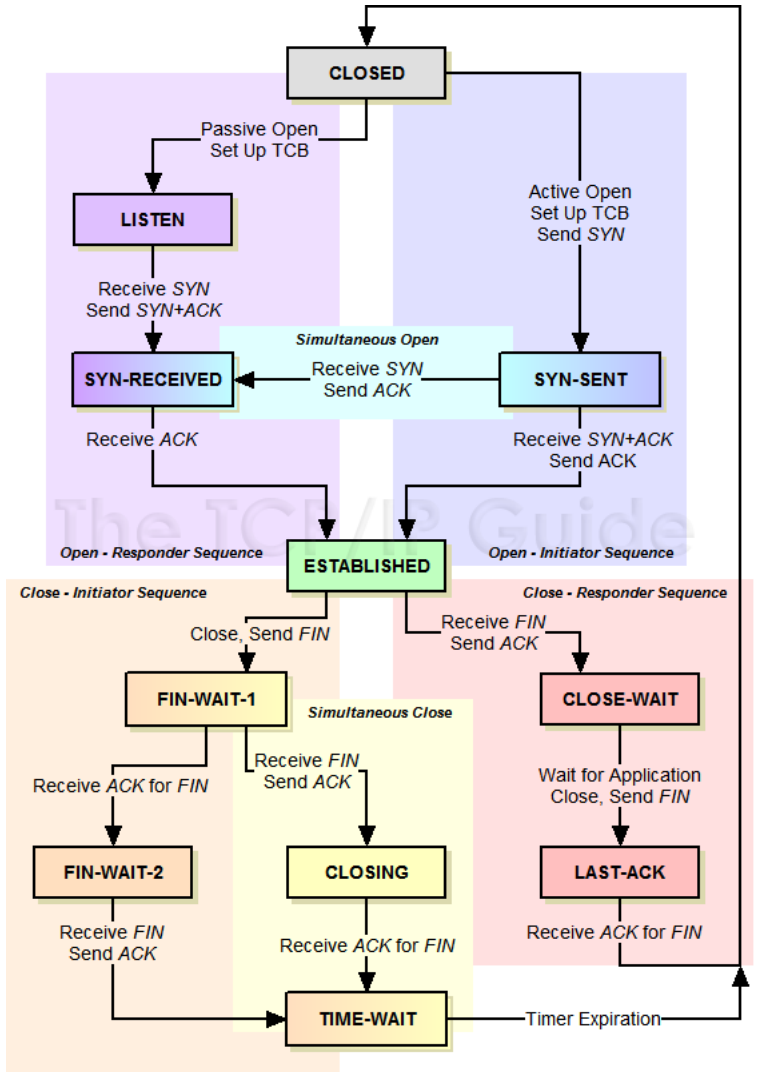
TCP connection management (1)

- Service:
  - reliable, ordered byte-stream delivery
  - connection-oriented
- Connection set-up:
  - active open: three-way handshake
  - passive open
  - local connection name
- Connection tear-down:
  - clean termination
  - abortive termination (option), uses RST (reset) flag.



TCP connection management (2)

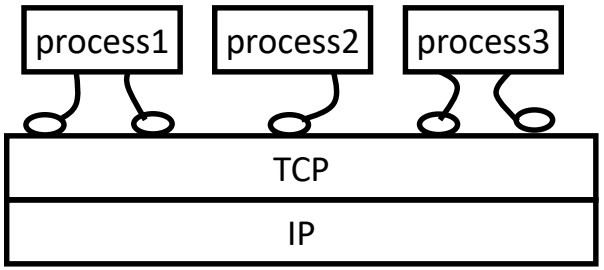
- TCP FSM
- Controls:
  - Connection setup (CS)
  - Connection release (CR)
  - Errors for CS and CR
  - Timeouts for CS and CR
- Once ESTABLISHED, then transmission follows **flow control** and **congestion control** algorithms.

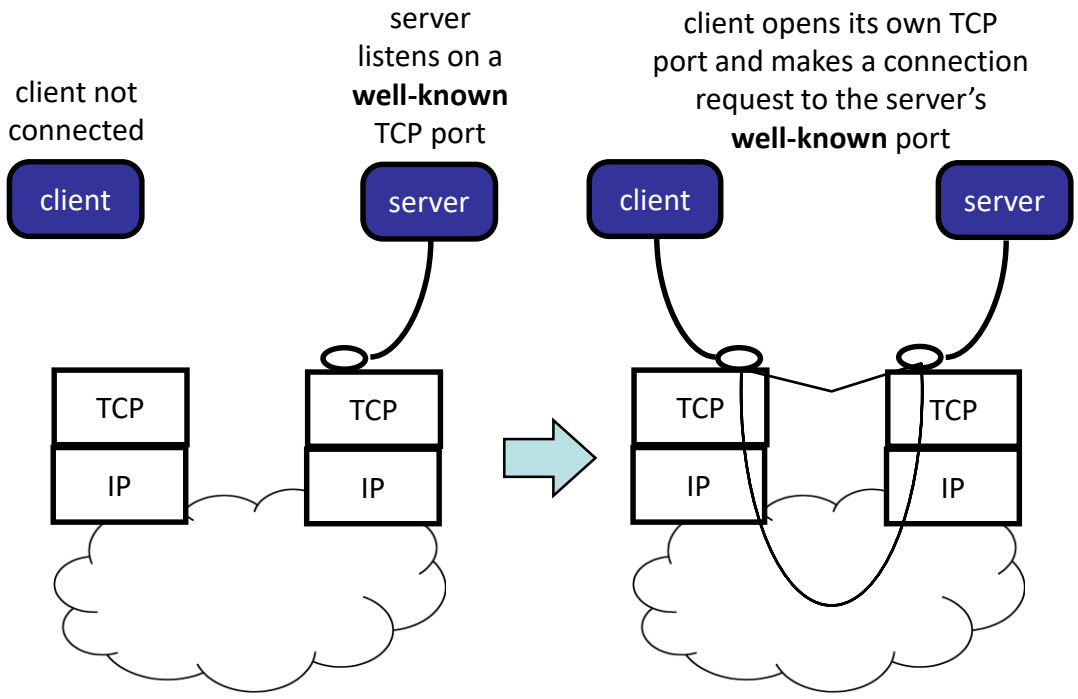


http://tcpipguide.com/

TCP ports

- Port numbers:
  - layer 4 de-multiplexers.
  - **assigned** – IANA.
  - **ephemeral** – dynamic.
- **Assigned numbers**:
  - "well-known" services.
  - /etc/services
- **Ephemeral**:
  - allocated by OS.
  - typically used by client.





TCP endpoints (1)

- Port numbers form part of transport identity.
- Transport layer tuple (5-tuple):
  - State maintained at hosts for each transport flow.
  - **Every unique 5-tuple is a unique flow.**

$$\langle TCP: A_{local}, P_{local}\ A_{remote}, P_{remote} \rangle$$

TCP	transport-layer protocol
A	IP address
P	Transport layer port number

TCP endpoints (2)

- Server port numbers, e.g.:
  - "well-known", e.g. from /etc/services
  - 22 for SSH
    80 for HTTP
    443 for HTTPS
- Client port numbers:
  - **Typically ephemeral.**
  - **Allocated by OS.**
  - **Connection to same server will have different client port numbers.**

<TCP: A_{local}, P_{local}, A_{remote}, P_{remote}>

For example:

connection 1: <TCP: A_C, **P_{C1}**, A_S, P_S>

connection 2: <TCP: A_C, **P_{C2}**, A_S, P_S>

C	client
S	Server

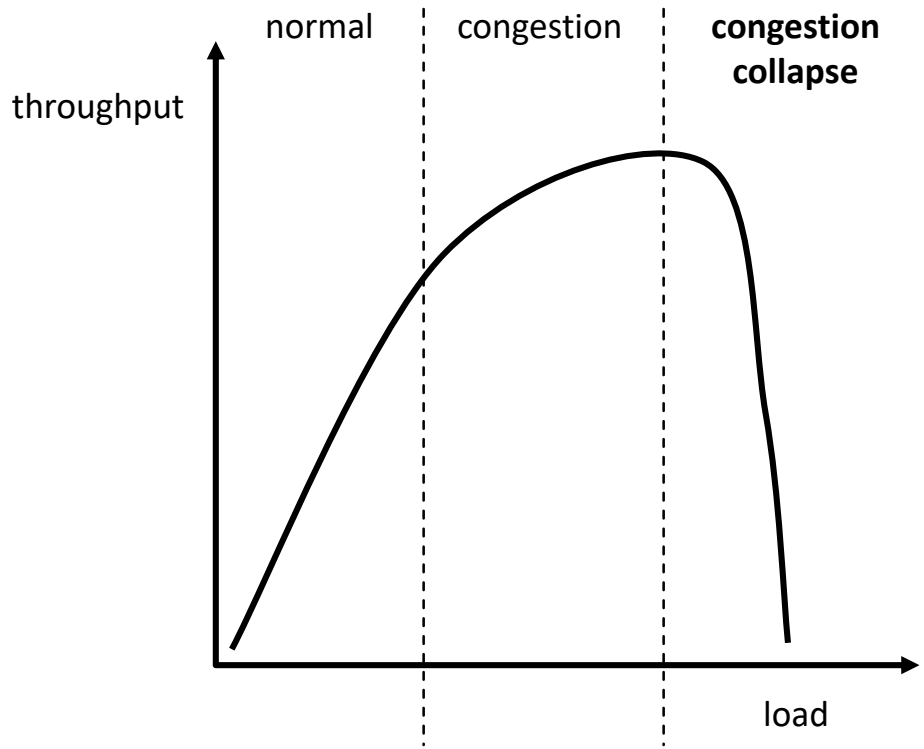
Remember that client and server are logical roles! Real applications can take one or both roles.

TCP – reliability

- Sequence numbers:
  - in TCP header.
  - implicit numbering of bytes in stream (index into stream).
- **Acknowledgement:**
  - in TCP header.
  - **next expected sequence number at receiver.**
- ACK "piggy-packed" on data:
  - header fields in each TCP segment.
  - might ACK single or multiple packets
- Sender has a **retransmission** strategy.
- Sequence numbers: ordering.
- Checksum:
  - uses pseudo header which includes, source and destination IP addresses.
- Checksum revaluated at receiver:
  - Bad checksum means packet should be discarded, no ACK should be sent.

Congestion effects

- Causes of congestion:
  - **too many packets**
  - buffer overflow in routers
  - unpredictable traffic patterns
  - route changes
  - time-of-day traffic
- Congestion effects:
  - higher end-to-end delay
  - lost packets
  - network instability
  - loss of service



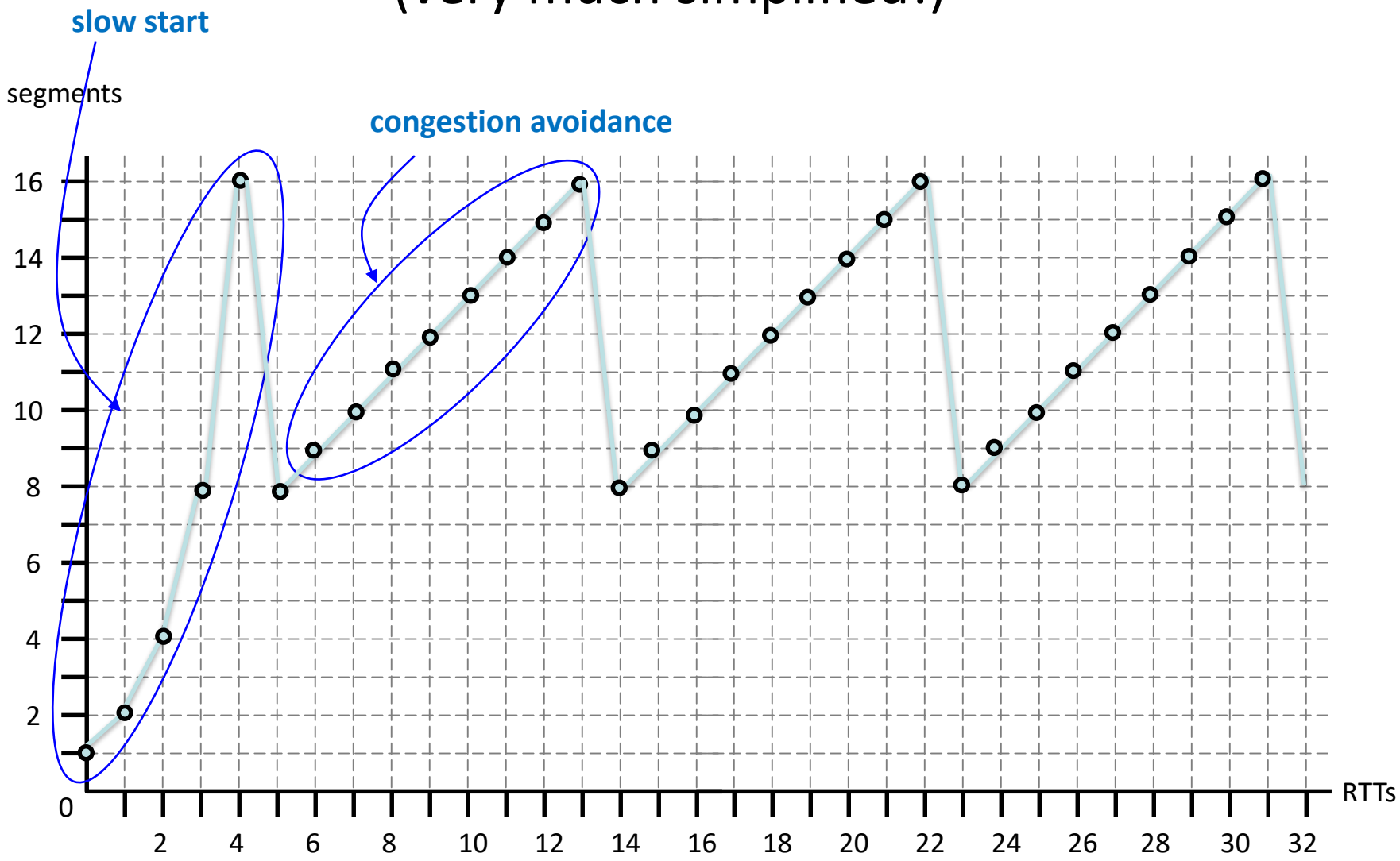
TCP – congestion control

University of St Andrews

- IP packets can enter the network unconstrained!
  - (line rate limit only)
- Imagine the aggregation of traffic from hundreds of workstations, each with a 1Gbps link!
- TCP-level transmission control required to prevent network congestion.

- **Slow start**:
  - Send 1 segment.
  - If successfully ACK'd, sends 2 segments, then 4, then 8 … (doubles on each ACK).
- **Congestion avoidance:**
  - Until a missing / delayed AC: then, drop to half, and increase linearly.
- **Overall result:**
  - **network does not get congested.**
  - **(missing / delayed ACK is an implicit signal from the network)**

TCP: sketch of throughput vs time
(very much simplified!)



TCP – flow control

- Window field:
  - 16 bits in TCP header.
  - **from receiver: number of bytes the receiver currently can accept.**
- Window size can be changed by receiver:
  - control transmission rate of the sender, e.g. busy server, resource constrained device.
- Window value:
  - can be changed **dynamically** by sender.
  - **not negotiated.**
  - (sender could ignore)
- **Flow control:**
  - **end-to-end signal from the receiver.**
  - **explicit control from the receiver.**

TCP is not secure

University of St Andrews

TCP transmissions are not protected from inspection, forgery, modification, or replay.

Reliability,
Flow Control, and
Congestion Control
are not security mechanisms.

TCP and UDP

University of St Andrews

1. TCP overview.
2. **UDP overview.**

Why UDP (User Datagram Protocol)?

University of St Andrews

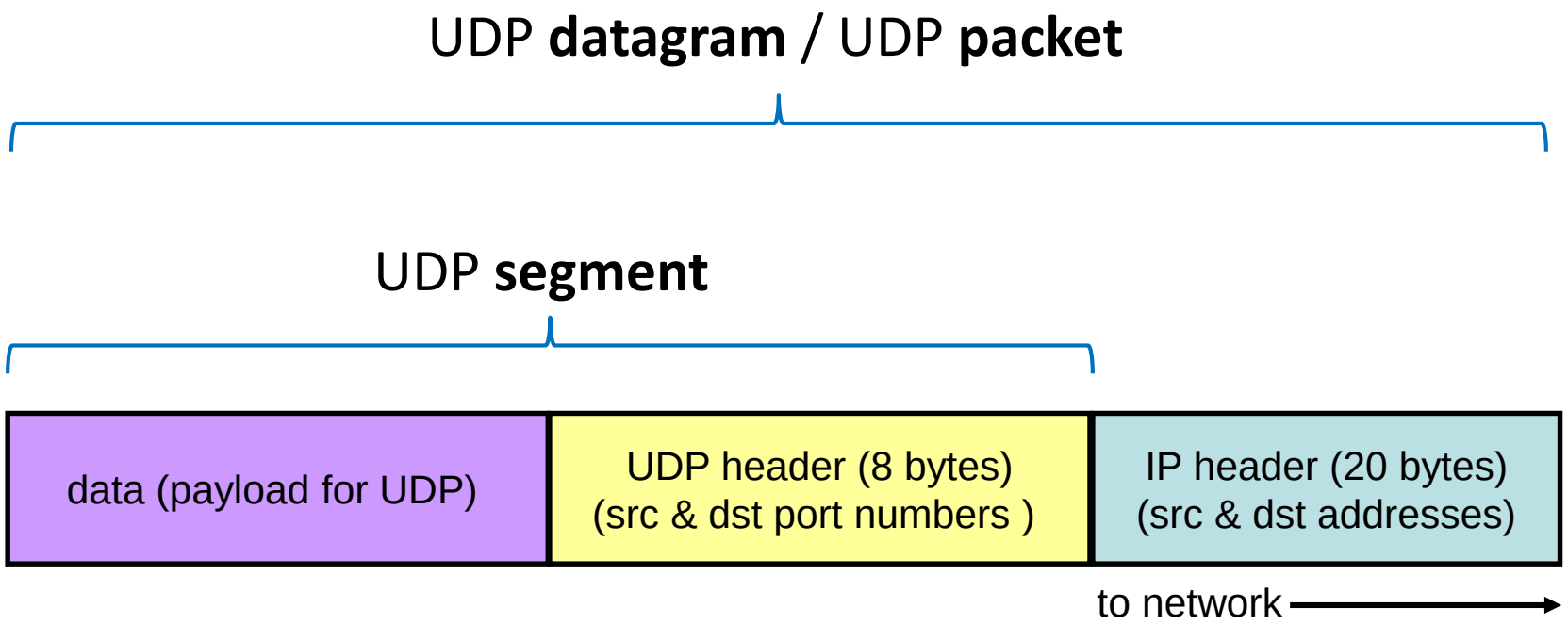
- End-to-end delay:
  - TCP reliability via retransmissions - adds **delay.**
  - TCP connection set-up - adds **delay.**
- Control of data transmission rate:
  - mechanisms of TCP not user controlled.
  - (congestion control, flow control).
- UDP used for various applications:
  - where reliability does not matter and/or delay does.
  - e.g. interactive voice and video (e.g. Skype).
  - e.g. (some?) online games.

UDP protocol

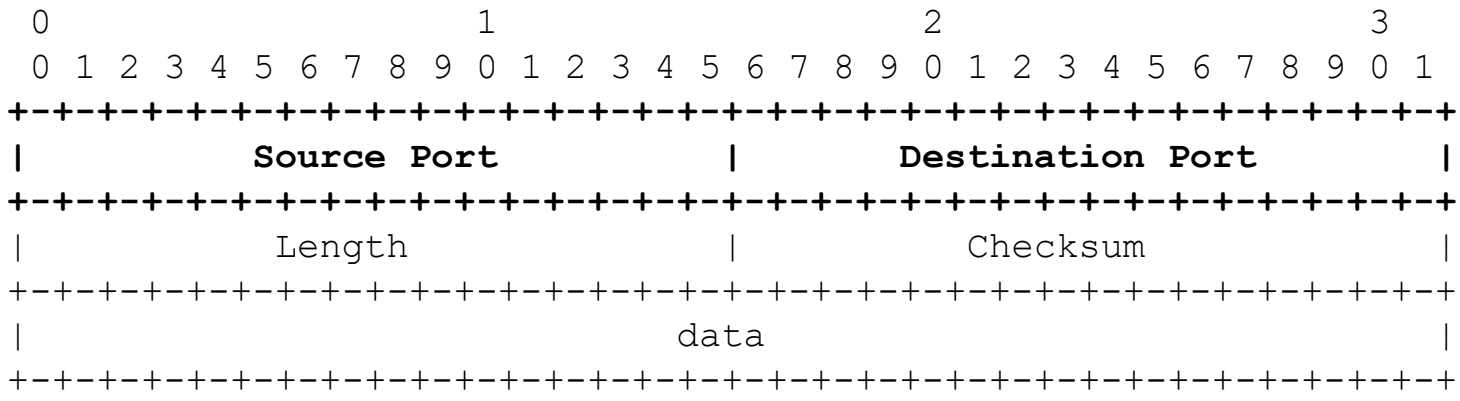
University of St Andrews

- Packet format:
  - header – source and destination port numbers.
  - payload – data from layer above.
- Service description:
  - connectionless (CL), unconfirmed, unreliable - **datagram service.**
  - **a thin layer on top of IP – best effort service, same as IP.**
- Protocol:
  - **data structure** – packet format.
  - **algorithm** – simple send/receive of independent packets.

A simplified UDP datagram



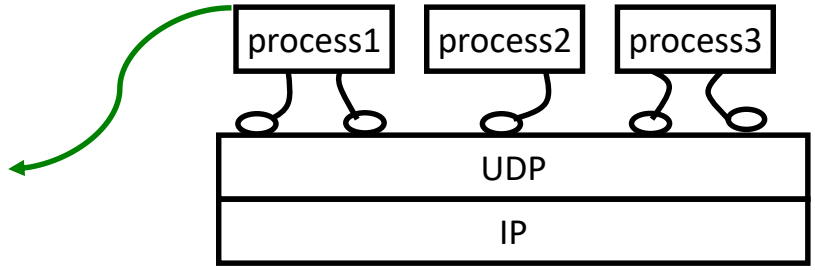
UDP header

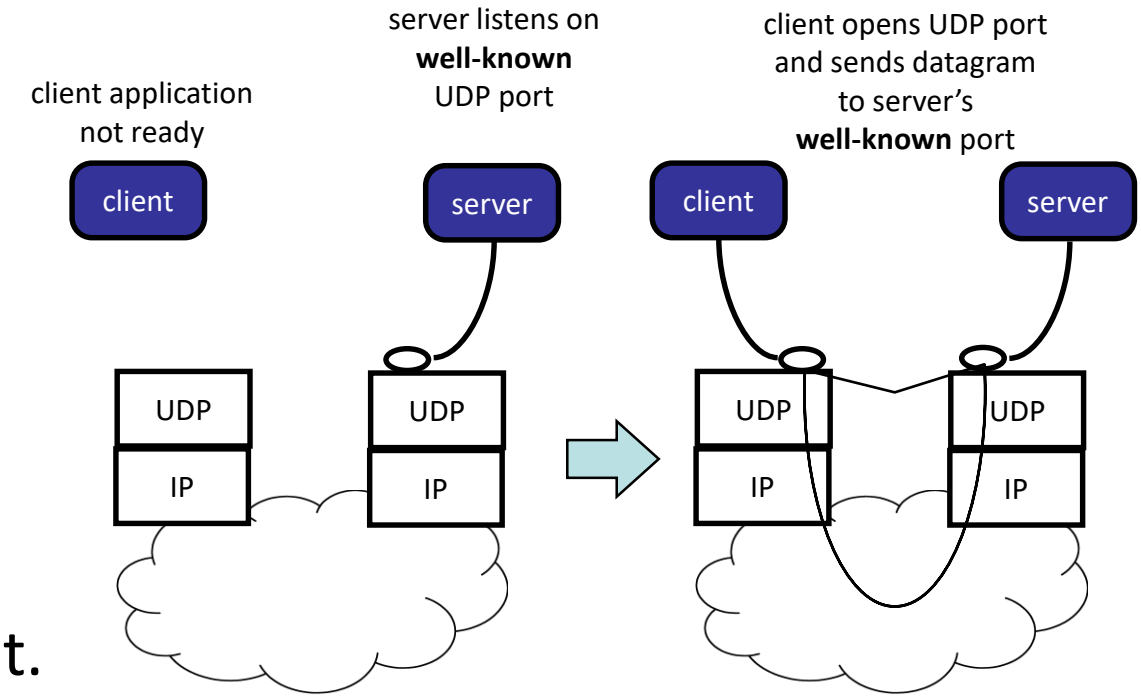


UDP Header Format (from RFC768, p1)

UDP ports

- Port numbers:
  - layer 4 de-multiplexers.
  - **assigned** – IANA.
  - **ephemeral** – dynamic.
- **Assigned numbers:**
  - well-known.
  - /etc/services
  - e.g. port 53 for DNS.
- **Ephemeral:**
  - allocated by OS.
  - typically used by client.





UDP endpoints

- Same principle as for TCP:
  - Local port numbers are different even if multiple UDP sessions to same remote host.
  - UDP 5-tuple.

<UDP: A_{local}, P_{local}, A_{remote}, P_{remote}>

Flow 1: <UDP: A_X, **P_{X1}**, A_Y, P_Y>
Flow 2: <UDP: A_X, **P_{X2}**, A_Y, P_Y>

X local host (could be a client)
Y remote host (could be a server)

UDP datagrams

- UDP datagrams are independent of each other
  - No flows
  - No flow control
  - No sequence numbers
  - No congestion control
- You can build flow control, congestion control etc on top of UDP
  - But then might you be better off using TCP?

TCP vs. UDP

University of St Andrews

TCP	UDP
• Connection oriented	• Connectionless
• Byte stream	• Datagram (packet)
• Reliable delivery	• Unreliable delivery
• Ordered delivery	• Unordered delivery
• Congestion control	• No congestion control
• Flow control	• No flow control
• **Send and wait:**	• **Send and hope(!):**
▪ **protocol will get your bytes to the receiver.**	▪ **delivery depends on network conditions.**

Summary

- Names, addresses and port numbers.
- TCP operation:
  - Connection management.
  - Reliability, flow control, congestion control.
- UDP operation:
  - Service and usage.
- Reading: Peterson & Davie Ch 5.1 and 5.2, Kurose & Ross Ch 3