Mic!

Lancaster University

Network Transport & UDP

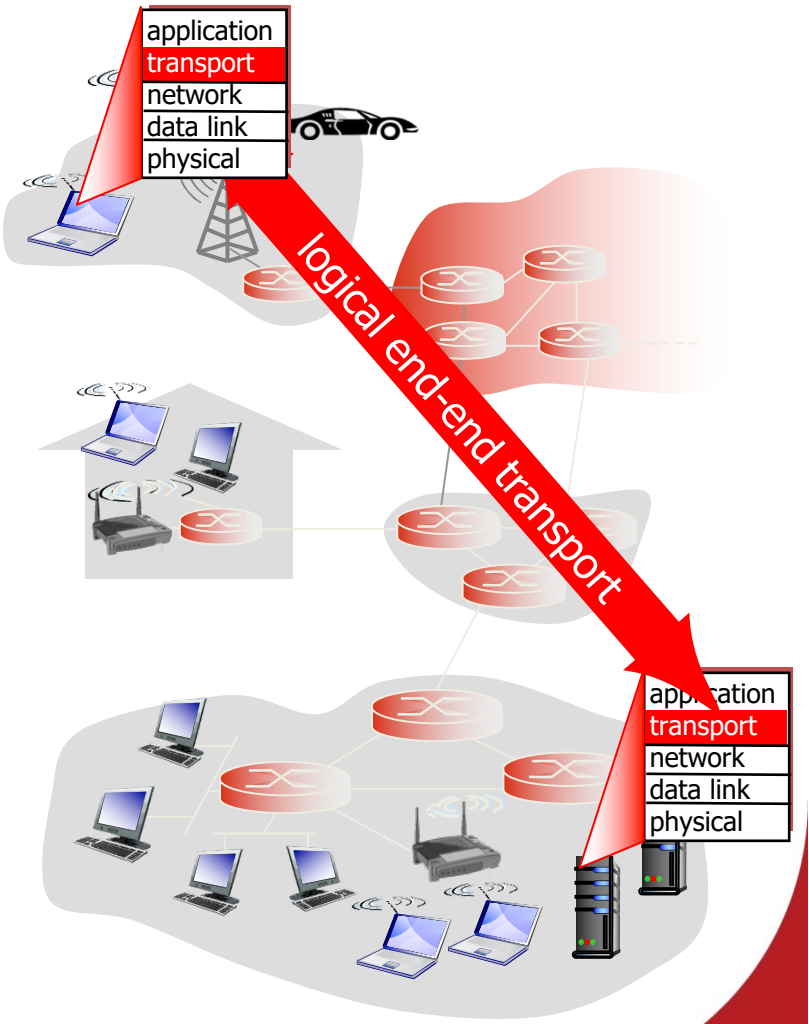
SCC. 203 – Computer Networks

Geoff Coulson

Week 15 Lecture 1

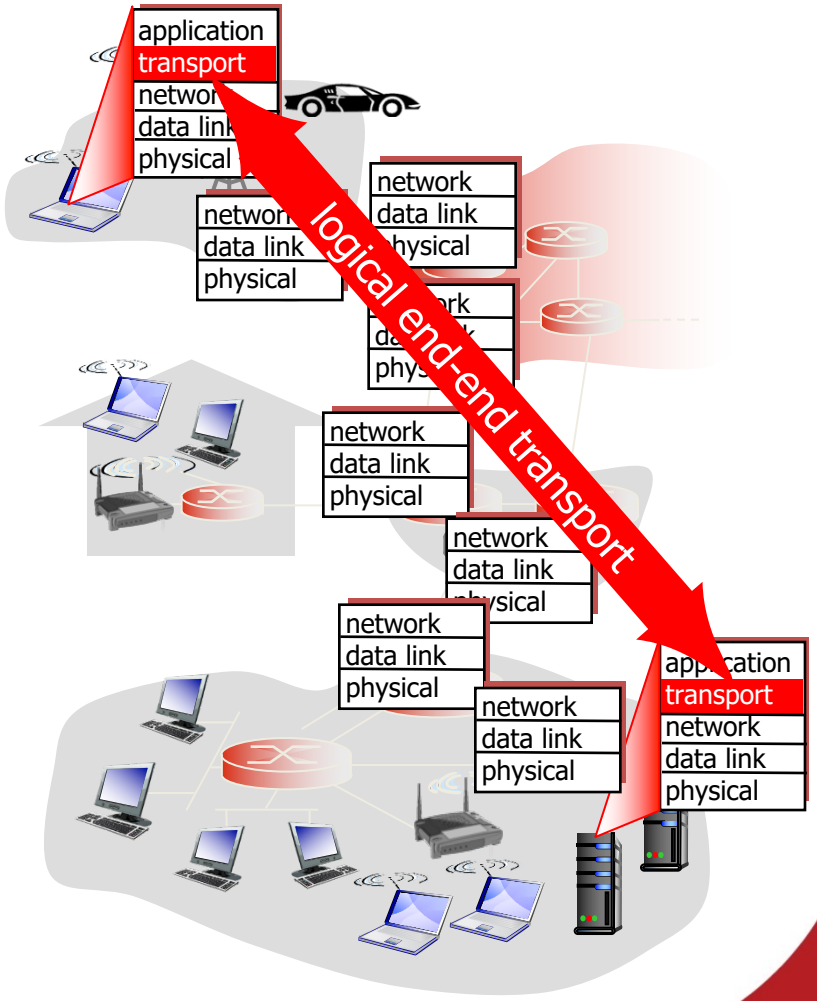
Reminder: transport services and protocols

- Provide logical, *end-to-end*, communication between application processes running on different hosts
- Transport protocols run in end systems only
  - Send side: breaks *application messages* into *segments*, and passes these to the network layer
  - Receiver side: reassembles segments into messages, and passes them up to the application layer
- More than one transport protocol available to apps
  - Internet: TCP and UDP



Internet transport-layer protocols

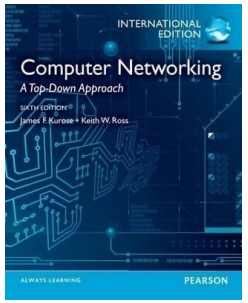
- Unreliable, unordered delivery: UDP
  - No-frills extension of "best-effort" IP
- Reliable, in-order delivery: TCP (and QUIC)
  - Connection setup
  - Control of flow and congestion
- Services *not* available from either:
  - Delay guarantees
  - Bandwidth guarantees



Lancaster University

Chapter 3.3

Connectionless Transport: UDP



UDP: no-frills, bare-bones transport protocol

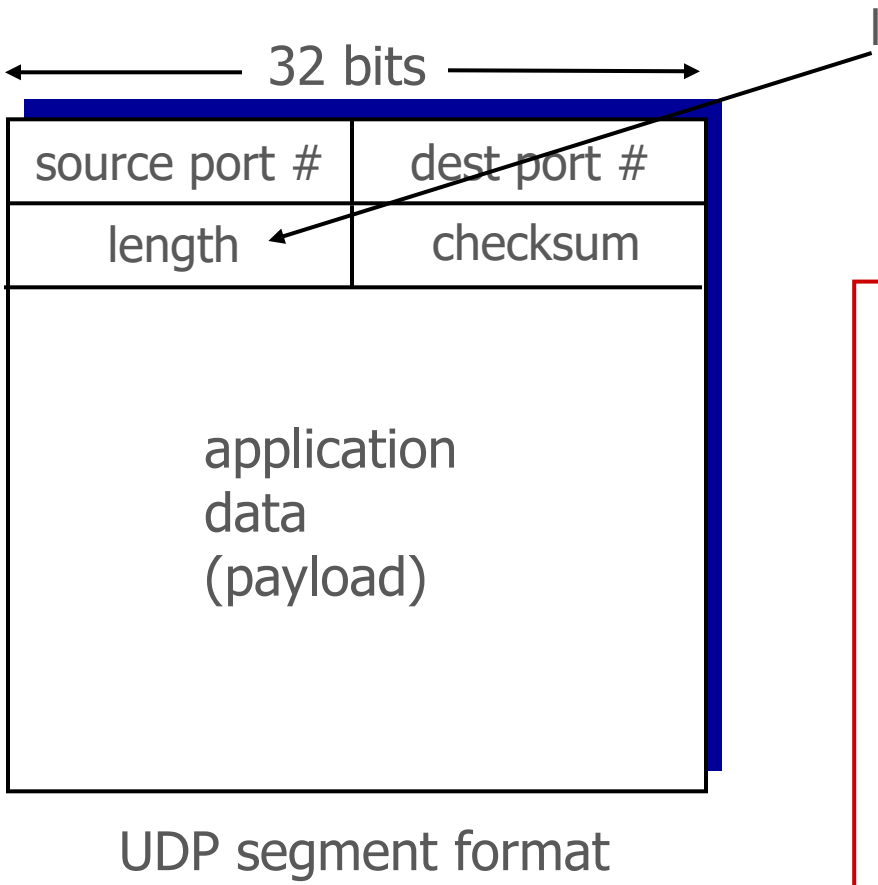
UDP: User Datagram Protocol [RFC 768]

Lancaster University

- "Best effort" service, UDP segments may be:
  - Lost
  - Delivered out-of-order to app
- *Connectionless:*
  - No handshaking between UDP sender, receiver
  - Each UDP segment handled independently of others
- UDP is "mostly IP with a short transport header"
  - Source and destination port
  - Ports enable dispatching of messages to receiver process

- UDP users include:
  - Streaming multimedia apps (**loss tolerant, rate sensitive**)
  - DNS
  - SNMP
- Want reliable transfer over UDP?
  - Add reliability etc., above, at application layer (cf. QUIC)
  - Application-specific error recovery!

UDP segment header



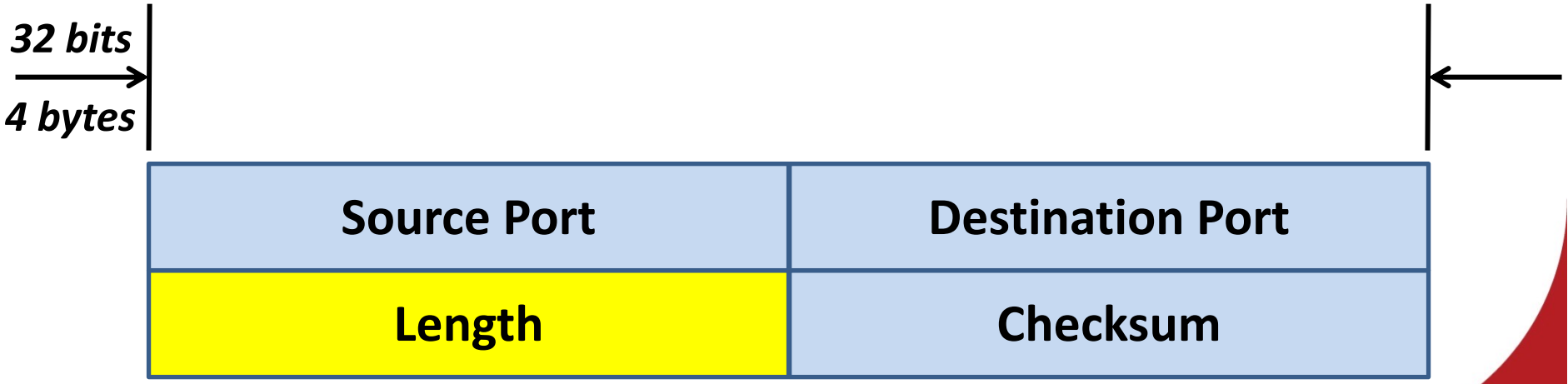
UDP segment format

why is there a UDP?

- No connection establishment (which can add delay)
- Simple: no connection state at sender, receiver
- Small header size (only 8 bytes)
- No congestion control: UDP can blast away as fast as desired

UDP length limits

- Length in bytes (UDP header and data)
  - Minimum is 8 bytes (length of UDP header)
  - Maximum is...
    - Length field is 16 bits, so $2^{16} - 1 = 65,535$ is the max. value
    - 65,535 – (UDP header length) 65,535 – 8 = **65,527 bytes**



UDP checksum

Lancaster University

Goal: detect "errors" (e.g., flipped bits) in transmitted segment

- Sender:
  - Treats segment contents, including header fields, as a sequence of 16-bit integers
  - Checksum := addition of these 'integers' (one's complement sum)
  - Puts checksum value into UDP checksum field
- Receiver:
  - Similarly, computes checksum of received segment
  - Checks if this equals given checksum field value:
    - NO - error detected
    - YES - no error detected (but there may be errors nonetheless!)

Lancaster University

Thanks for listening!
Any questions?