The Transfer Layer

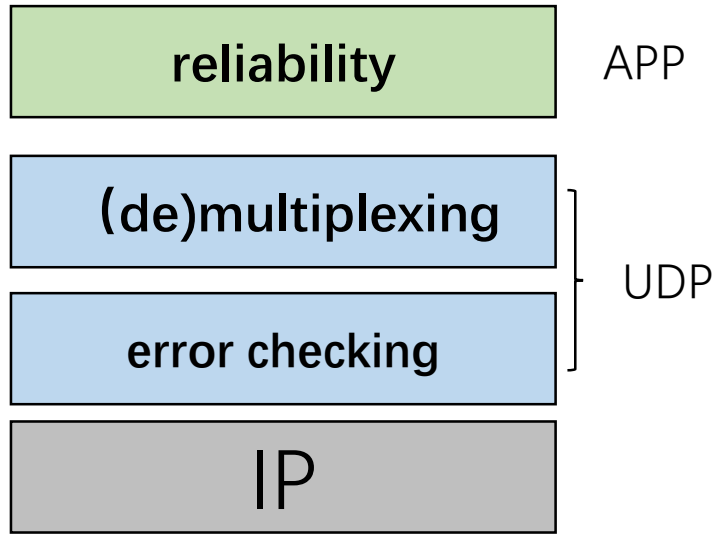
UDP Service

School of Software Engineering
South China University of Technology
Dr. Chunhua Chen
chunhuachen@scut.edu.cn
2019 Spring

Connectionless Transport: UDP

a no-frills, bare-bones transport protocol,with just a small enhence of IP service

- Connectionless
  - no handshaking between sending and receiving transport-layer entities before sending a segment.
- No reliability?
  - Can be built into the application itself.



Why UDP?

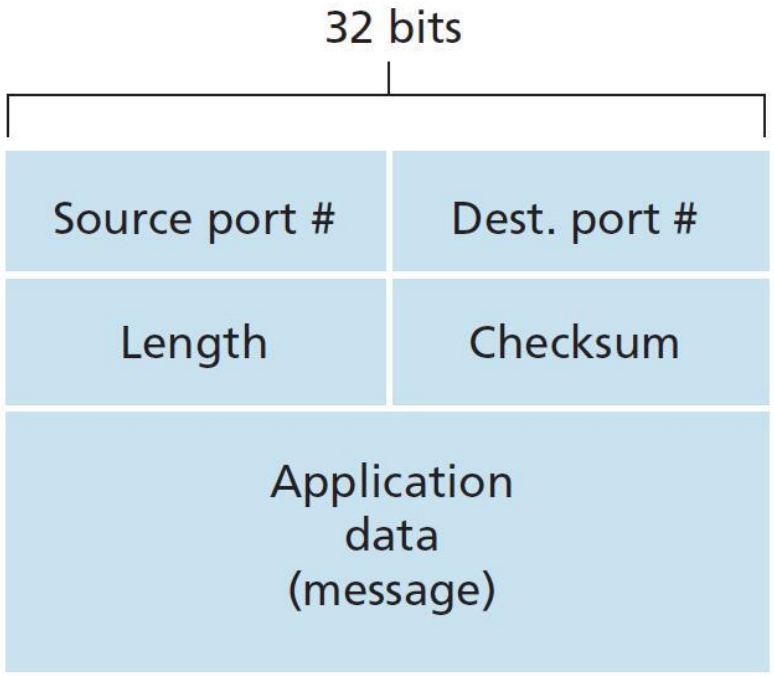
- Finer application-level control over what data is sent, and when.
  - Consider real-time applications requiring a minimum sending rate.
- No connection establishment
  - Comparing DNS and Web
- No connection state
  - support many more active clients (than TCP)
- Small packet header overhead
  - UDP: 8 bytes, TCP: 20 bytes

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP	TCP
Remote terminal access	Telnet	TCP
Web	HTTP	TCP
File transfer	FTP	TCP
Remote file server	NFS	Typically UDP
Streaming multimedia	typically proprietary	UDP or TCP
Internet telephony	typically proprietary	UDP or TCP
Network management	SNMP	Typically UDP
Routing protocol	RIP	Typically UDP
Name translation	DNS	Typically UDP

Figure 3.6 ♦ Popular Internet applications and their underlying transport protocols

UDP Segment Structure

- **Port** number fields for (de)multiplexing
- **Length** field specifies the number of bytes in the UDP segment (header plus data).
- **Checksum** is used by the receiving host to check whether errors have been introduced into the segment.
  - i.e. error detection.
- Only error detection, no error correction



UDP Checksum

- UDP at the sender side performs the 1s complement of the sum of all the 16-bit words in the segment, with any overflow encountered during the sum being **wrapped around**.
  - RFC 1071

Sender

```
0110011001100000
0101010101010101
1000111100001100
```

```
0110011001100000
0101010101010101
1011101110110101
1011101110110101
1000111100001100
0100101011000010
```

```
0100101011000010
        ↓ 1s complement
1011010100111101
```

Receiver

```
0100101011000010
1011010100111101
1111111111111111
```

?

Why error detection at UDP? The end-end principle

- Bit errors
  - During transmitting
  - During buffering.
- Many link layer protocols (including the popular Ethernet protocol) also provide error checking.
- But **No Guarantee** that all the links between source and destination provide error checking;
- Bit errors could be introduced when a segment is stored in a router's memory