
# COMPUTER COMMUNICATION NETWORKS

Department of Electronics and Communication Engineering

# COMPUTER COMMUNICATION NETWORKS

## CONNECTIONLESS TRANSPORT:
## UDP Segment format, checksum calculation

Dr. Arpita Thakre

Department of Electronics and Communication Engineering

## Connectionless Transport UDP

- UDP is defined in **RFC 768**
- "best effort" service, UDP

**Segments may be:**

- lost
- delivered out-of-order to client/server process.

- ***connectionless:***
- no handshaking between UDP sender, receiver
- UDP does not perform segmentation of upper layer messages.

- UDP use:
  - streaming multimedia apps (loss tolerant, rate sensitive)
  - DNS
  - SNMP
- no connection establishment (which can add delay)
- simple: no connection state at sender, receiver

## Connectionless Transport UDP

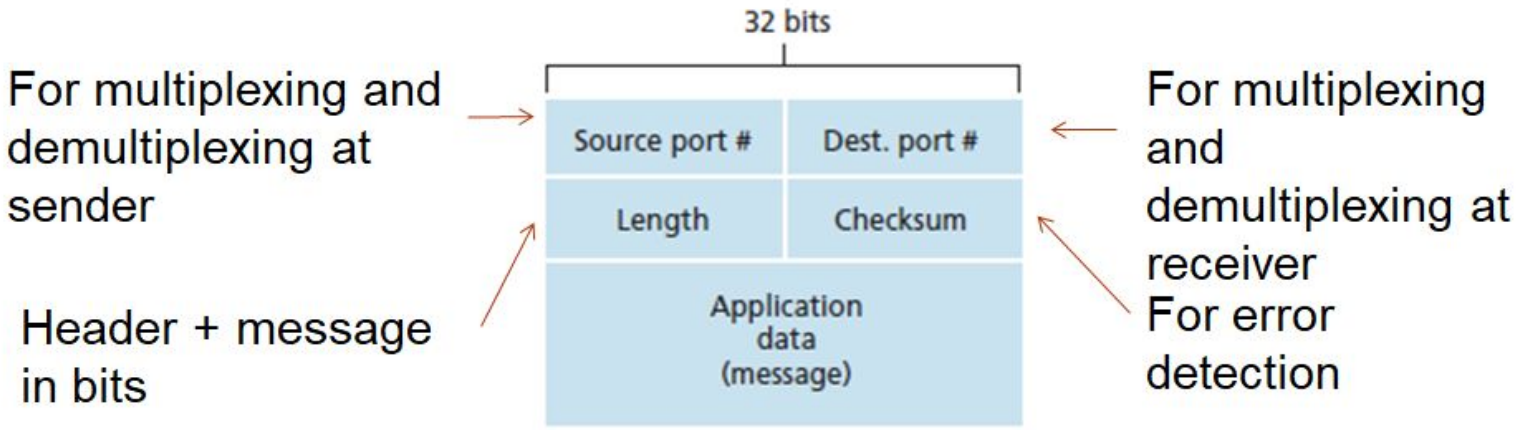
- Finer application-level control over what data is sent, and when.
- No connection establishment
- No connection state
- Small packet header overhead

## Connectionless Transport UDP

| 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 |
| Name translation | DNS | Typically UDP |

**Figure 3.6 Popular Internet applications and their underlying transport protocols**

## UDP Segment format



## *CHECKSUM CALCULATION*

**Checksum calculation at the sender side:**

a) The given data is split into 16-bit numbers (some zeros are padded if the given data is not an integral multiple of 16).

b) The 16-bit numbers are summed and the carry is wrapped around and added back to the resulting sum.

c) Take one complement of the sum and this result is called "checksum".

d) Append the data to the checksum.

## *CHECKSUM CALCULATION(Cont.)*

**Error detection at the receiver side:**

a) Separate the checksum from the data.

b) The data is split into 16-bit numbers (some zeros are padded if the given data is not an integral multiple of 16).

c) The 16-bit numbers are summed and the carry is wrapped around and added back to the resulting sum.

d) Add the checksum to the sum.

e) Data is treated as intact if the resulting sum contains only ones; Otherwise data was corrupted.

## *Checksum example:*

- Assume segment has 3 16-bit words

0110011001100000
0101010101010101
1000111100001100

The sum of first two of these 16-bit words is

0110011001100000
0101010101010101
1011101110110101

Adding the third word to the above sum gives

1011101110110101
1000111100001100
0100101011000010

1s complement of the sum 0100101011000010 is 1011010100111101

## *UDP checksum calculation:*

- The above sender and receiver operations are performed as before.
- However, the data in this context includes some fields from the network layer header and the UDP segment.
- On the sender side, prior to calculation of checksum assume that this field is all zeros.

Example: Suppose IPv4 datagram is carrying the UDP segment, the following IPv4 header fields are required for the calculation: source IP address, Dest. IP address, Upper layer protocol (binary value of 17), Datagram length

## *UDP checksum calculation:*

UDP must provide error detection at the transport layer, on an end-end basis, if the end-end data transfer service is to provide error detection. Example of end-to-end service.


# THANK YOU

Dr. Arpita Thakre

Department of Electronics and Communication Engineering