Transport Layer Protocols

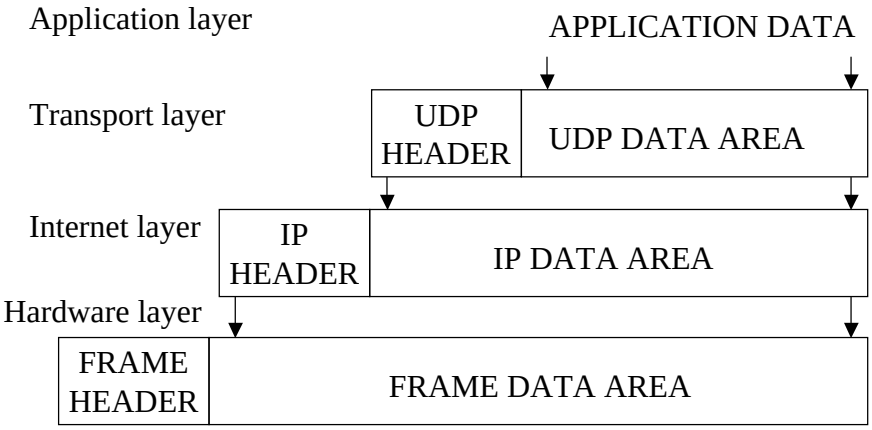
User Datagram Protocol (UDP)

Transmission Control Protocol (TCP)

Purpose of Transport Layer

- Transport Layer provides communication from an application to another application (= end-to-end communication)
- May be reliable or unreliable
- Must accept data from several applications

Encapsulation



User Datagram Protocol (UDP)

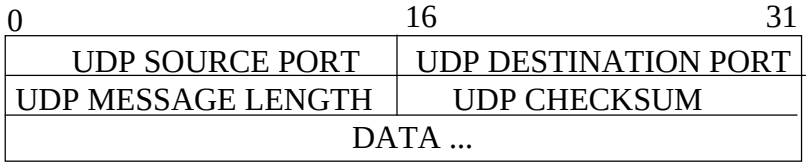
Comer's

chapter 12 (3rd and 4th ed.)

Repetition of UDP

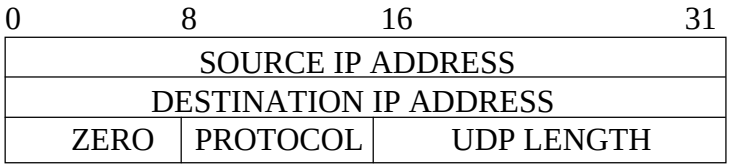
- User Datagram Protocol (UDP)
- Light protocol
- Several application programs may work in a host: UDP uses ports
- Unreliable connectionless delivery service
  - No acknowledgements, no retransmissions
  - Checksum (error detection) is optional

UDP Message Format



- Source port and checksum are optional
- Length: header + data length in octets
- Ports act as subaddresses in one host

UDP Pseudo-header



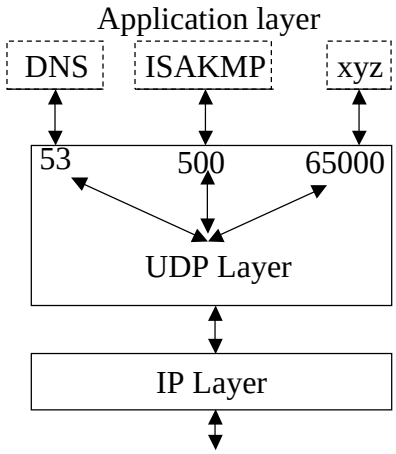
- Pseudo-header is used to verify that packet has reached the correct destination
  - IP addresses from IP header
  - Protocol number (17 = UDP) from IP header
  - Length of UDP datagram (UDP header + data)
- Pseudo-header is not sent over the network

Calculation of UDP Checksum

- Checksum is calculated from pseudo-header, UDP header and data.
  - Checksum is zero before calculation
  - Divide all data to 16 bits long pieces
  - Calculate one's complement sum from the pieces
  - Take one's complement from the result
- Checksum is similar in TCP

Ports and (De)Multiplexing

- Well-known ports
  - for standard protocols
  - www.iana.org
  - RFC 1700 (1994)
  - 0-1023
- Reserved ports
  - for services
  - 1024-49151
- Dynamic/private ports
  - 49152-65535



Summary

- User Datagram Protocol (UDP) offers connectionless unreliable delivery service
- UDP adds multiplexing in a host to service offered by IP
- Ports are divided to well-known, reserved and private ports

References

- Comer's chapter 12
- RFC 768 - User Datagram Protocol
- RFC 1122 - Requirements for Internet Hosts - Communication Layers
- M. Morris Mano: Computer Engineering - Hardware Design, 1988
- www.iana.org

Transmission Control Protocol

Comer's
chapter 13 (3rd and 4th ed.)

Repetition of TCP

- Transmission Control Protocol (TCP)
  - Connection-oriented
  - end-to-end (process-to-process)
  - Reliable
  - Two-way
- Applications usually need reliable data transmission over unreliable networks

Reliable Delivery Service

- Repetition from TLARK: Characteristics of reliable delivery service:
  - Stream orientation
  - Virtual circuit connection
  - Buffered transfer
  - Unstructured stream
  - Full duplex connection

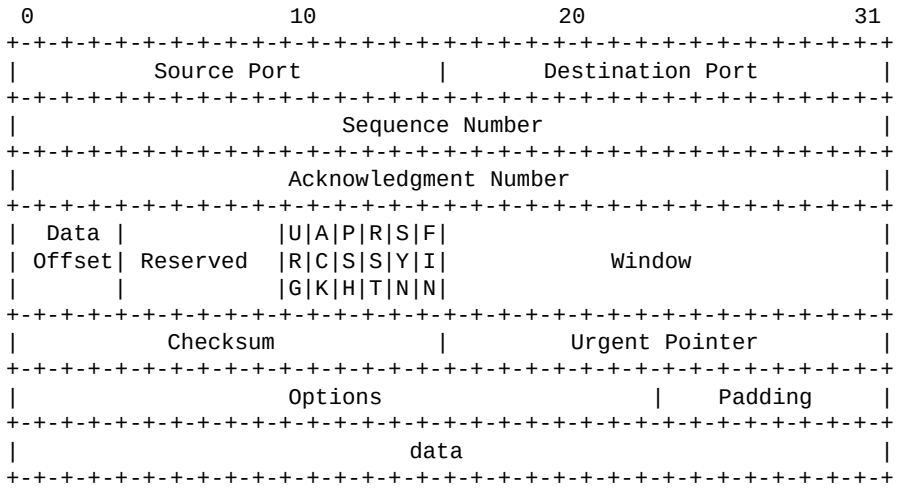
Required Functions to Implement Reliable Delivery Service

- Basic Data Transfer
- Reliability
- Flow control
- Multiplexing
- Connections
- Precedence and Security

Connection

- Identifier: pair of endpoints
  - Endpoint: (host, port) pair
  - For example: (130.233.45.31, 6000) and (130.233.220.31, 85) is an identifier
- A connection contains information:
  - Window size, sockets, sequence numbers, and other information needed for maintain the connection.

TCP Header Format (RFC 793)

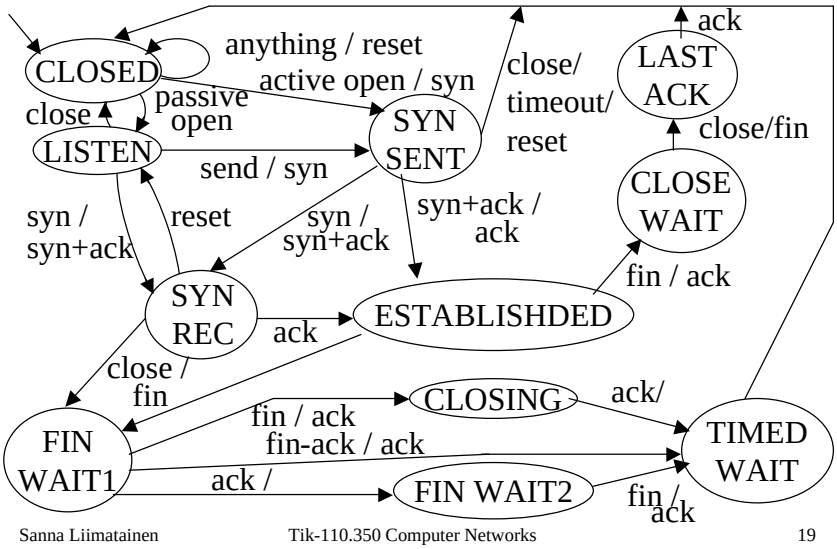


TCP Header - Options

- 0 = End of option list
  - After all options
- 1 = No-operation
  - Sometimes used between options
- 2 = Maximum Segment Size
  - In connection establishment, peers agreed on the maximum segment size

Fig. 13.15

State Machine of TCP

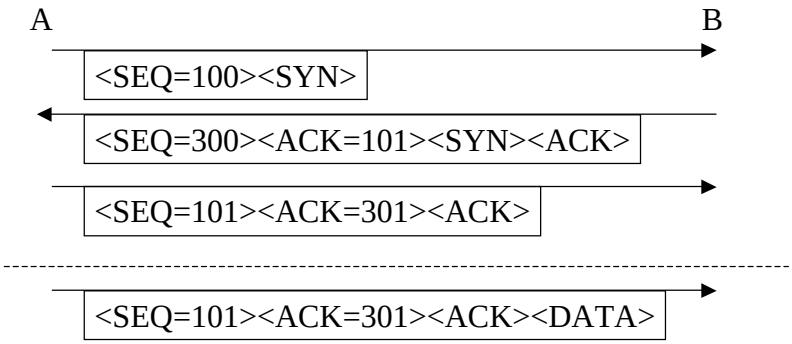


Sequence Numbers

- Begin from some value (not always 1)
- Calculate octets in data stream from the initial value
- Sequence numbering for both directions
- Used in acknowledgements

Connection Establishment

- Three-way handshake



Closing Connection

- Modified three-way handshake
  - Other end does not have data to send: closing the connection in one direction

→ The sender sends a segment with a FIN bit set

← The receiver acknowledges the FIN with ACK

  - No data to send also to other direction

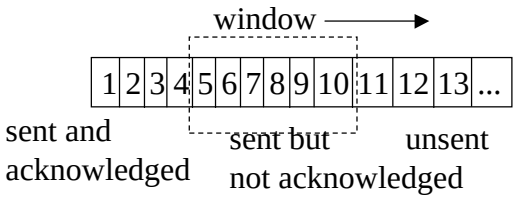
← Receiver sends FIN

→ Sender ACKs the FIN segment

Reliable Transmission

- Reliable delivery = data is delivered from one machine to other machine without duplicates or losses
  - Positive acknowledgements
  - Retransmissions
  - Error detection (timers, checksums)

Idea of Sliding Window



- More efficient transmission
  - A packet can be sent before previous packet has been acknowledged
  - Buffer for unacknowledged packets

TCP Window Mechanism

- For end-to-end flow control
  - Can also help avoiding congestion
- At each end, there are two windows
  - One for sent packets and one for received packets
- Window size can vary
  - How many octets the receiver can accept

Silly Window Syndrome (SWS)

- Applications with different speeds
  - Receivers buffer become full
  - It reads one octet and advertises window size 1
  - Sender sends one octet in a TCP packet
- Silly window may occur on both sender and receiver sides
- Small amout of data compared to header
- More calculation and processing

Avoiding SWS

- Receiver side
  - Advertise only ”significantly” large window
  - Delayed acknowledgements
- Sender side
  - Wait until enough data to send a maximum size segment
  - Wait until an acknowledgement arrives

Acknowledgements

- Cumulative acknowledgements
  - Easy to generate and unambiguous
  - Lost acknowledgements does not necessary generate retransmissions
  - Usually several sequences are acknowledged in one message
  - Sender either send all again or send only the first unacknowledged sequence

Retransmission

- If no acknowledgement arrives in a time interval, the packet is retransmitted
- Adaptive retransmission algorithm
  - TCP measures performance of connections
- Round trip time
  - RTT = (α*oldRTT)+((1-α)*newRTTsample)
- Timeout: β*RTT, β>1

Acknowledgement Ambiguity

- TCP uses cumulative acknowledments
- If a retransmission is sent, which sequence the acknowledgement belong to?
  - original datagram: longer timeouts
  - retransmitted datagram: too sort timeouts
  - discard these in the round trip calculation – not a good solution
- Timer backoff strategy

Variance in Delay

- Goal: retransmission timer is near actual roundtrip time
  - Also variance in round trip time is followed
  - More calcuation that effects to the timer

Congestion

- Congestion occurs at a switching point (e.g. router) somewhere between the endpoints
- Congestion means increased delay
- Transmission rate must be reduced to avoid congestion collapse
  - Congestion window

Global Synchronization

- Congestion has occured in the network
- All machines start sending at the same time
- Random Early Discard (RED) in routers
  - if queue is full → discard datagram
  - when queue is allmost full → discard datagram with some probablity
  - when queue is not full → deliver datagrams

Summary

- Transmission Control Protocol (TCP) offers connection-oriented, reliable delivery service
- Efficient use of network
- Flow control tells how much data can be handled in both endpoints

References

- Comer's chapter 13
- RFC 793 – Transmission Control Protocol
- RFC 1122 – Requirements for Internet Hosts – Communication Layers