Transport Layer

Josh Wilcox (jw14g24@soton.ac.uk)

February 11, 2025

Contents

1 Transport Layer Addressing - Ports

- Both TCP and UDP use 16-bit port numbers
  - Numbers range from $0 \rightarrow 65535$
- Uniquely identifies a connection endpoint on the host
  - Two services can not listen on the same port at the same time
  - The operation will not allow this to happen
  - If multiple apps could use the same port, you wouldn't be able to till which application should get the data

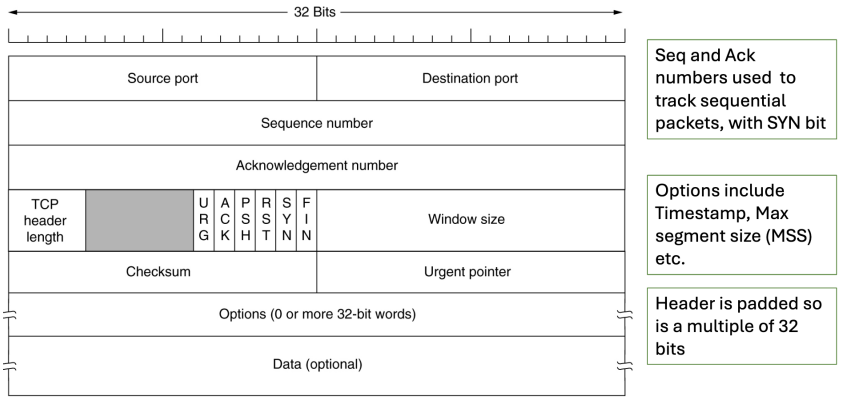
2 TCP

- TCP is **connection oriented**
- Includes *acknowledgments* and retransmissions
- Provide *flow control* and *congestion control* for segments it sends
- TCP adjusts sending rate over time

2.1 Properties

- Provides connection management
- Provides flow control
- Retransmission is done if you lose packets
- Receiver reassambles segments in the correct order

2.2 TCP Header



2.2.1 TCP Header Fields

- **Source Port** - Identifier for the sending port.
- **Destination Port** - Identifier for the receiving port.
- **Sequence Number** - Orders the packets in the stream.
- **Acknowledgment Number** - Confirms receipt of data.
- **Data Offset** - Indicates the header length.
- **Reserved** - Reserved for future use.
- **Flags** - Controls (e.g., SYN, ACK, FIN, etc.) the connection.

- **Window** - Manages flow and controls the data rate.
- **Checksum** - Ensures header and data integrity.
- **Urgent Pointer** - Indicates urgent data if set.
- **Options** (optional) - Provides additional functionalities.

2.3 TCP Connection Establishment

- Uses a **three way handshake**
  - SYN
  - SYN-ACK
  - ACK
- SYN - Opens connection with random sequence number

2.4 TCP Reliability

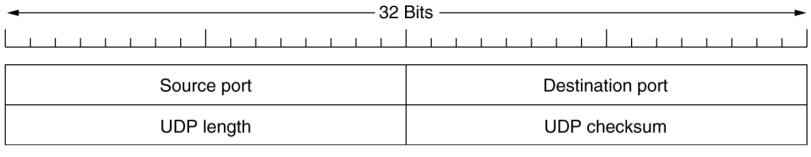
- ACKs are sent back by the receiver
- Sender must detect lost packets
  - This is done by **Retransmision Timeout**

3 UDP

- Completely connectionless
- Follows a "send and forget" mentality
- Always sends as fast as it can
  - No Acknowledgements
  - No sequance numbers
  - Very lightweight
  - Just send and hope
- Application has to handle any lost packets

3.1 Header

- Allows sending of datagrams without establishing a connection
- the UDP header is much simpler than TCP
- Checksum is optional
- Can multicast (one to many)



Then Payload (variable size)....

3.2 UDP Loss

- Lossy / congested network links can drop packets
- UDP does not handle these losses
- Applications need to detect these losses and deal with them themselves

4 TCP/UDP Service Model

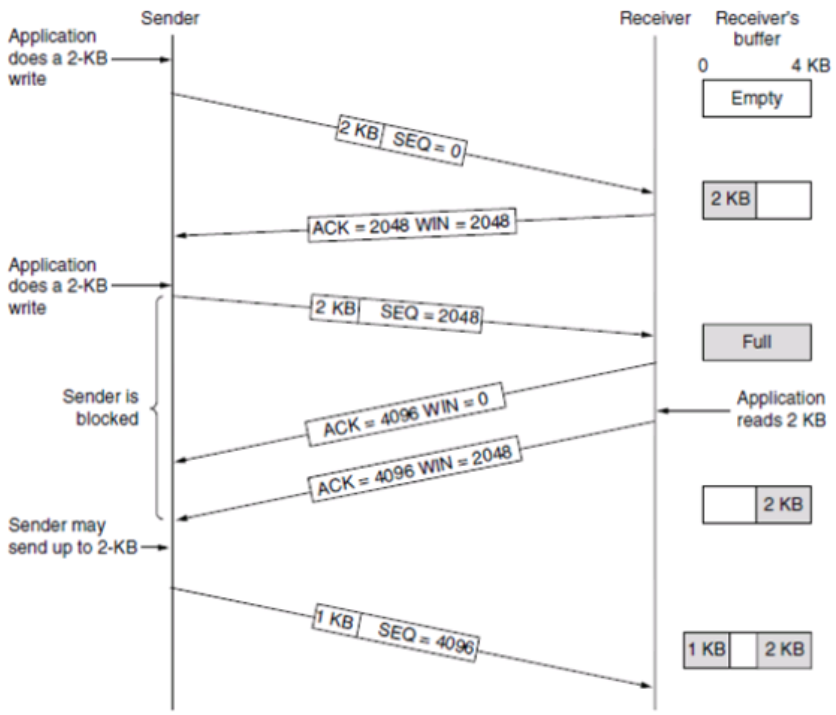
- A socket is an interface that facilitates network communication between devices.
- Each socket is bound to a specific IP address and port number, which uniquely identifies the communication endpoint.
- Sender and receiver each create a socket to act as their respective endpoints for data exchange.
- A complete communication session requires five pieces of information: source IP address, source port, destination IP address, destination port, and the protocol being used.

5 TCP Flow Control

- TCP uses a sliding window protocol to control how quickly data is sent
  - This is because the receiver has a limited incoming buffer size
- Sender should not send data unless the receiver indicates it has the buffer space to accept it
- The amount of buffer space will change over time - hence the "sliding window" term

5.1 Sliding Window Process

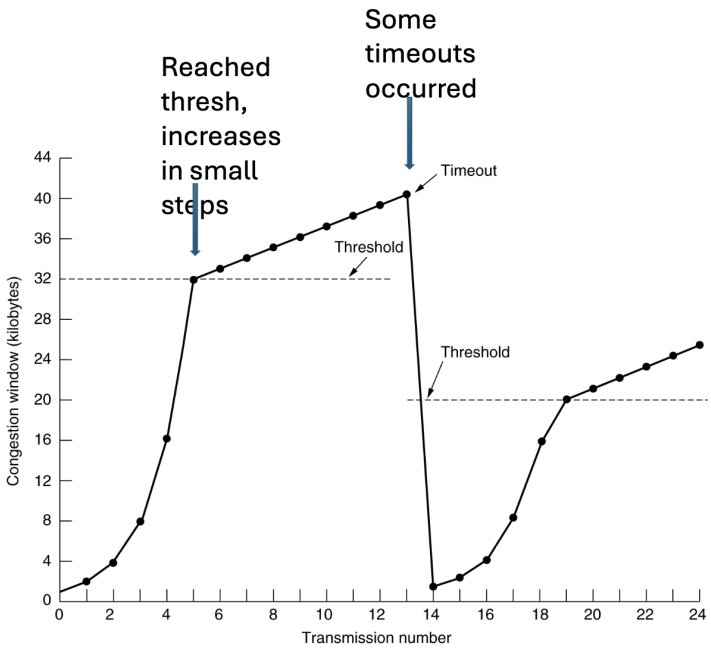
- Sender sends a segment with a sequence number and starts a timer
  - A segment in this context is a discrete unit of data that is transmitted over a network
- Reveiver replies with an ack number showing the next sequence number it expects **and available window size**
  - If the sender's timer goes off before this ack is received, the sender retransmits
- If the receiver's window size is 0, the sender may send a probe to get a new window advertisement



Window management in TCP

6 TCP Congestion Control

- Maintained by the sender
- The TCP congestion window indicates the number of bytes a sender may put into the network at any time
  - Packet loss is a signal of congestion
  - Runs alongside the receiver's sliding window
  - Uses the smaller of the two windows when sending
- The congestion window size starts small
  - Known as "slow start"
  - Size doubles every round trip until a threshold
  - After this threshold, the size increases linearly until some timeouts occur
  - Once a timeout occurs, the size goes back to the beginning and the threshold is lowered



7 Uses of UDP and TCP

7.1 Uses of TCP

- **Web Browsing**
  - Ensures reliable delivery of web pages and resources.
- **Email**
  - Guarantees that emails are delivered correctly.
- **File Transfers**
  - Protocols like FTP rely on TCP for reliable file transfer.
- **YouTube**
  - Uses TCP to ensure reliable streaming of video content.
  - Ensures that video data is received in the correct order and without errors.

7.2 Uses of UDP

- **Online Gaming**
  - Low latency is crucial; some packet loss is acceptable.
- **Streaming Media**
  - Live broadcasts and real-time video/audio streaming.
- **Voice over IP (VoIP)**
  - Real-time voice communication where speed is more important than reliability.
- **DNS Queries**
  - Quick resolution of domain names with minimal overhead.