OSI Reference Model Transport Layer

Transport Layer

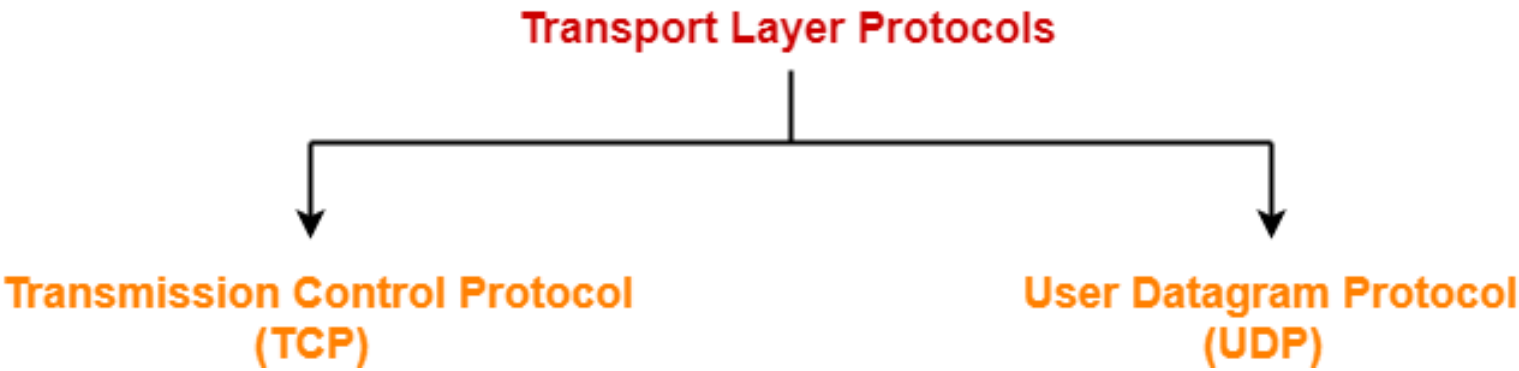
Munesh Singh

Indian Institute of Information Technology, Design and Manufacturing Kancheepuram, Chennai, Tamil Nadu 600127

November 15, 2020

Transport Layer Protocols

- There are mainly two transport layer protocols that are used on the Internet-



- **Transmission Control Protocol**
  - TCP is short for Transmission Control Protocol.
  - It is a transport layer protocol.
  - It has been designed to send data packets over the Internet.
  - It establishes a reliable end to end connection before sending any data.

Characteristics Of TCP

- **TCP is a reliable protocol.**
  - It guarantees the delivery of data packets to its correct destination.
  - After receiving the data packet, receiver sends an acknowledgement to the sender.
  - It tells the sender whether data packet has reached its destination safely or not.
  - TCP employs retransmission to compensate for packet loss.
- **TCP is a connection oriented protocol.**
  - TCP establishes an end to end connection between the source and destination.
  - The connection is established before exchanging the data.
  - The connection is maintained until the application programs at each end finishes exchanging the data.
- **TCP handles both congestion and flow control.**
  - TCP handles congestion and flow control by controlling the window size.
  - TCP reacts to congestion by reducing the sender window size.

- **TCP ensures in-order delivery.**
  - TCP ensures that the data packets get deliver to the destination in the same order they are sent by the sender.
  - Sequence Numbers are used to coordinate which data has been transmitted and received.
- **TCP connections are full duplex.**
  - TCP connection allows to send data in both the directions at the same time.
  - So, TCP connections are Full Duplex.
- **TCP works in collaboration with Internet Protocol.**
  - A TCP connection is uniquely identified by using-
    - Combination of port numbers and IP Addresses of sender and receiver.
  - IP Addresses indicate which systems are communicating.
  - Port numbers indicate which end to end sockets are communicating.
  - Port numbers are contained in the TCP header and IP Addresses are contained in the IP header.
  - TCP segments are encapsulated into an IP datagram.
  - So, TCP header immediately follows the IP header during transmission.

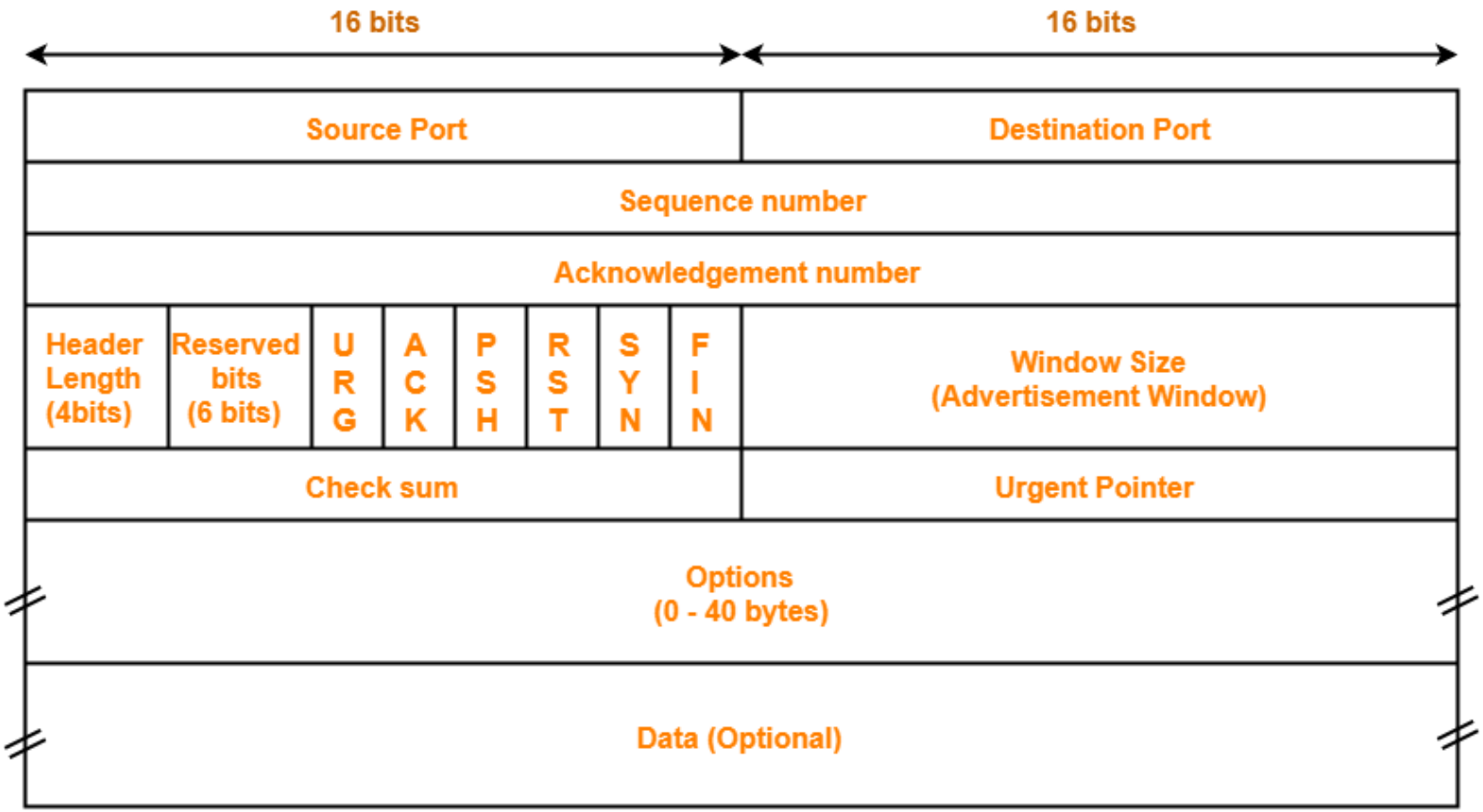
- **TCP can use both selective & cumulative acknowledgements.**
  - TCP uses a combination of Selective Repeat and Go back N protocols.
  - In TCP, sender window size = receiver window size.
  - In TCP, out of order packets are accepted by the receiver.
  - When receiver receives an out of order packet, it accepts that packet but sends an acknowledgement for the expected packet.
  - Receiver may choose to send independent acknowledgements or cumulative acknowledgement.
  - To sum up, TCP is a combination of 75% SR protocol and 25% Go back N protocol.
- **TCP is a byte stream protocol.**
  - Application layer sends data to the transport layer without any limitation.
  - TCP divides the data into chunks where each chunk is a collection of bytes.
  - Then, it creates a TCP segment by adding IP header to the data chunk.
  - TCP segment = TCP header + Data chunk.

- **TCP provides error checking & recovery mechanism.**
  - Checksum
  - Acknowledgement
  - Retransmission
- It continuously receives data from the application layer.
- It divides the data into chunks where each chunk is a collection of bytes.
- It then creates TCP segments by adding a TCP header to the data chunks.
- TCP segments are encapsulated in the IP datagram.

TCP segment = TCP header + Data chunk

TCP Header

- The following diagram represents the TCP header format-



TCP Header

- **Source Port**
  - Source Port is a 16 bit field.
  - It identifies the port of the sending application.
- **Destination Port**
  - Destination Port is a 16 bit field.
  - It identifies the port of the receiving application.
- **Note:**
  - A TCP connection is uniquely identified by using-
    - Combination of port numbers and IP Addresses of sender and receiver
  - IP Addresses indicate which systems are communicating.
  - Port numbers indicate which end to end sockets are communicating.
- **Sequence Number**
  - Sequence number is a 32 bit field.
  - TCP assigns a unique sequence number to each byte of data contained in the TCP segment.
  - This field contains the sequence number of the first data byte.

- **Acknowledgement Number**
  - Acknowledgment number is a 32 bit field.
  - It contains sequence number of the data byte that receiver expects to receive next from the sender.
  - It is always sequence number of the last received data byte incremented by 1.
- **Header Length**
  - Header length is a 4 bit field.
  - It contains the length of TCP header.
  - It helps in knowing from where the actual data begins.
- **Minimum and Maximum Header length**
  - The length of TCP header always lies in the range-
    [20 bytes , 60 bytes]
    - The initial 5 rows of the TCP header are always used.
    - So, minimum length of TCP header = 5 x 4 bytes = 20 bytes.
    - The size of the 6th row representing the Options field vary.
    - The size of Options field can go up to 40 bytes.
    - So, maximum length of TCP header = 20 bytes + 40 bytes = 60 bytes.

- **Concept of Scaling Factor**
  - Header length is a 4 bit field.
  - So, the range of decimal values that can be represented is [0, 15].
  - But the range of header length is [20, 60].
  - So, to represent the header length, we use a scaling factor of 4.
  - Header length = Header length field value x 4 bytes
- **Example**
  - If header length field contains decimal value 5 (represented as 0101), then-
    Header length = 5 x 4 = 20 bytes
  - If header length field contains decimal value 10 (represented as 1010), then-
    Header length = 10 x 4 = 40 bytes
  - If header length field contains decimal value 15 (represented as 1111), then-
    Header length = 15 x 4 = 60 bytes

- Header length and Header length field value are two different things.
- The range of header length field value is always [5, 15].
- The range of header length is always [20, 60].
- **Reserved Bits**
  - The 6 bits are reserved.
  - These bits are not used.
- **URG Bit: URG bit is used to treat certain data on an urgent basis.**
  - When URG bit is set to 1,
    - It indicates the receiver that certain amount of data within the current segment is urgent.
    - Urgent data is pointed out by evaluating the urgent pointer field.
    - The urgent data has be prioritized.
    - Receiver forwards urgent data to the receiving application on a separate channel.
  - **ACK Bit: ACK bit indicates whether acknowledgement number field is valid or not.**
    - When ACK bit is set to 1, it indicates that acknowledgement number contained in the TCP header is valid.
    - For all TCP segments except request segment, ACK bit is set to 1.
    - Request segment is sent for connection establishment during Three Way Handshake.

- **PSH Bit: PSH bit is used to push the entire buffer immediately to the receiving application.**
  - When PSH bit is set to 1,
    - All the segments in the buffer are immediately pushed to the receiving application.
    - No wait is done for filling the entire buffer.
    - This makes the entire buffer to free up immediately.
- **Note:**
  - Unlike URG bit, PSH bit does not prioritize the data.
  - It just causes all the segments in the buffer to be pushed immediately to the receiving application.
  - The same order is maintained in which the segments arrived.
  - It is not a good practice to set PSH bit $= 1$.
  - This is because it disrupts the working of receivers CPU and forces it to take an action immediately.

- **RST Bit-RST bit is used to reset the TCP connection.**
  - When RST bit is set to 1,
    - It indicates the receiver to terminate the connection immediately.
    - It causes both the sides to release the connection and all its resources abnormally.
    - The transfer of data ceases in both the directions.
    - It may result in the loss of data that is in transit.
  - This is used only when-
    - There are unrecoverable errors.
    - There is no chance of terminating the TCP connection normally.
  - **SYN Bit: SYN bit is used to synchronize the sequence numbers.**
    - When SYN bit is set to 1,
    - It indicates the receiver that the sequence number contained in the TCP header is the initial sequence number.
    - Request segment sent for connection establishment during Three way handshake contains SYN bit set to 1.

- **FIN Bit: FIN bit is used to terminate the TCP connection.**
  - When FIN bit is set to 1,
    - It indicates the receiver that the sender wants to terminate the connection.
    - FIN segment sent for TCP Connection Termination contains FIN bit set to 1.
- **Window Size**
  - Window size is a 16 bit field.
  - It contains the size of the receiving window of the sender.
  - It advertises how much data (in bytes) the sender can receive without acknowledgement.
  - Thus, window size is used for Flow Control.
- **Note**
  - The window size changes dynamically during data transmission.
  - It usually increases during TCP transmission up to a point where congestion is detected.
  - After congestion is detected, the window size is reduced to avoid having to drop packets.

- **Checksum**
  - Checksum is a 16 bit field used for error control.
  - It verifies the integrity of data in the TCP payload.
  - Sender adds CRC checksum to the checksum field before sending the data.
  - Receiver rejects the data that fails the CRC check.
- **Urgent Pointer**
  - Urgent pointer is a 16 bit field.
  - It indicates how much data in the current segment counting from the first data byte is urgent.
  - Urgent pointer added to the sequence number indicates the end of urgent data byte.
  - This field is considered valid and evaluated only if the URG bit is set to 1.
- **USEFUL FORMULAS**
  - **Number of urgent bytes = Urgent pointer + 1**
  - **End of urgent byte= Sequence number of the first byte in the segment + Urgent pointer**

- **Options**
  - Options field is used for several purposes.
  - The size of options field vary from 0 bytes to 40 bytes.
- **Options field is generally used for the following purposes-**
  - Time stamp
  - Window size extension
  - Parameter negotiation
  - Padding
- **Time Stamp**
  - Multiple segments having the same sequence number may appear at the receiver side.
  - This makes it difficult for the receiver to identify the correct segment.
  - If time stamp is used, it marks the age of TCP segments.
  - Based on the time stamp, receiver can identify the correct segment.

- **Window Size Extension**
  - Options field may be used to represent a window size greater than 16 bits.
  - Using window size field of TCP header, window size of only 16 bits can be represented.
  - If the receiver wants to receive more data, it can advertise its greater window size using this field.
  - The extra bits are then appended in Options field.
- **Parameter Negotiation**
  - Options field is used for parameters negotiation.
  - Example- During connection establishment,
    - Both sender and receiver have to specify their maximum segment size.
    - To specify maximum segment size, there is no special field. So, they specify their maximum segment size using this field and negotiates.

- **Padding**
  - Addition of dummy data to fill up unused space in the transmission unit and make it conform to the standard size is called as padding.
  - Options field is used for padding.
    - When header length is not a multiple of 4, extra zeroes are padded in the Options field.
    - By doing so, header length becomes a multiple of 4.
    - If header length = 30 bytes, 2 bytes of dummy data is added to the header.
    - This makes header length = 32 bytes.
    - Then, the value 32 / 4 = 8 is put in the header length field.
    - In worst case, 3 bytes of dummy data might have to be padded to make the header length a multiple of 4.

Thank You