

VE489 Homework 6

Wei, Yinyuo, 520370910103

July 2023

1. UDP features best effort transmission. The transmission is faster and a shorter header is needed, but the transmission is not guaranteed to be successful. It is connectionless: no handshaking no connection state. Besides, no flow control, no error control and no congestion control is performed. UDP datagrams can be lost or out-of-order.
As for TCP, different from UDP, it is transmitted byte by byte. It is connection-oriented. It features full-duplex unicast connection between client server processes. The transmission contains three major states: connection setup, monitor connection state and connection release. It has higher header overhead to guarantee successful transmission. It performs error control, flow control, and congestion control and has higher delay than UDP.
2. We need a stable connection to make sure the data is successfully transmitted and received. The IP addresses and ports of both source and destination nodes are needed. TCP sets up a bidirectional connection to ensure reliable and ordered data transmission. The three-way handshake establishes a stateful connection, enabling error detection, retransmission, and flow control. This approach allows TCP to guarantee data delivery and maintain full-duplex communication for efficient and secure data exchange between two endpoints.
3. TCP is designed as a byte-stream protocol because we need want to know how many bytes can be recieved from the receiver's side and transmit accordingly. It would be convenient if we transmit groups of bytes, which are flexible size segments. The receiver can reassemble the bytes. This mechanism facilitates flexible, reliable and ordered transmission.
4. The pseudo header is employed when calculating the checksum for the transport layer. It is not an actual component of the network packet but is utilized as input for the checksum algorithm. Its inclusion ensures a dependable and uniform method for confirming the integrity of the transport layer segment. It contains 32 bits for source and destination address, 8-bit 0, protocol and UDP length for UDP or TCP segment length for TCP.

5. An initial sequence number is selected during connection setup. ISN aims to avoid overlap with sequence numbers of prior connections. A local clock is used to select ISN. Time for clock to go through a full cycle is required to be greater than the maximum lifetime of a segment (MSL).
6. Because there are too many unpredictable behaviors from the users so that there are too many conditions to consider. We want to wait until the channel is stabilized to make sure the connection is released safely. We want to ensure that all data is received before terminating the connection and minimize the chances of data loss or inconsistencies during connection termination.
7. Flow control is controlled from the receiver's side. Receiver controls rate at which sender transmits to prevent receiver buffer overflow. TCP use 16 bits in its header to advertise its available window size to the transmitter. The sender will send bytes with SN from ACK to $ACK + window\ size$. Meanwhile, congestion control is implemented from the sender side. The sender maintains a congestion window $cwnd$ to control congestion at intermediate routers. Hence, the effective window size is the minimum of the congestion window and the advertised window. Besides, a dynamic BW control is implemented. Sources probe the network by increasing $cwnd$ and when congestion is detected, sources reduce rate.
8. (a) Slow start
Increase congestion window size by one segment upon receiving an ACK from receiver. The congestion window increases exponentially.
- (b) Congestion Avoidance
Algorithm progressively sets a congestion threshold. When $cwnd$ exceeds the threshold, slow down rate at which $cwnd$ is increased. Increase congestion window size by one segment per round-trip-time (RTT). Each time an ACK arrives, $cwnd$ is increased by $1/cwnd$. $cwnd$ grows linearly with time.
- (c) Fast Retransmit/Fast Recovery
During the congestion avoidance phase in TCP, if a packet loss is detected, TCP assumes network congestion. Instead of waiting for a timer to expire, TCP quickly retransmits the lost packet upon receiving three duplicate ACKs. These duplicate acknowledgments indicate that the receiver has received the same out-of-order packet multiple times and confirms the loss. To prevent further congestion, TCP reduces its congestion window size and performs congestion avoidance. Once all lost packets are successfully retransmitted, TCP resumes the congestion avoidance phase.
9. Slow start is necessary because the sender do not know the congestion status of the network at the beginning. And slow start can sense the available network bandwidth.
It is efficient because the window size explodes exponentially.

10. TCP can implement fast retransmission and recovery. That is to say, TCP quickly retransmits the lost packet upon receiving three duplicate ACKs and adjust its threshold and window size immediately instead of waiting for timeout. Besides, cwnd will oscillate around optimal value.