Definitions

- datagram - the name given to packets of data sent across the internet layer.
- transport protocol - a protocol providing end-to-end delivery between end points of a connection.
- Internet Protocol (IP) - a protocol in the internet layer that provides “unreliable” datagram service between hosts. Both TCP and UDP use IP as their delivery service.

Definitions

- Transmission Control Protocol (TCP) - a protocol in the transport layer providing reliable data delivery.
- User Datagram Protocol (UDP) - a protocol in the transport layer providing unreliable datagram service.

UDP

- UDP delivers datagrams between applications or processes on host computers.
  - “Best effort” or unreliable delivery.
  - Checksum (optional) guarantees integrity of data.
- protocol port or port - an end point of UDP.
- Each UDP transmission identifies the internet address and port number of the destination and the source.
- The destination port and the source port may be different.

UDP Headers

- UDP datagrams have a header that comes after the hardware and IP headers.
- The UDP header is very simple. It contains:

  - port numbers;

  - message length;

  - checksum.
- UDP ports are similar in concept to TCP ports but are otherwise unrelated.

TCP

- TCP is the most widely used transport protocol.
- TCP is a high-overhead protocol that provides reliable data delivery.
- This is accomplished using IP’s unreliable datagram delivery service!
- TCP compensates for packet loss, delay, and duplication.

Features of TCP

- Connection Orientation.
- Point-to-Point Communication.
- Reliability.
- Full-Duplex Communication.
- Stream Interface.
- Reliable Connection Startup.
- Graceful Connection Shutdown.

A TCP Connection

- TCP is an end-to-end protocol.
- A TCP connection is a virtual connection achieved by software, not by the underlying network system.
- The end points are identified by ports.
- TCP uses IP as its data delivery service. TCP data travels in IP datagrams.
- IP treats TCP like data and does not interpret the contents of a TCP message.
- Internet routers look only at the IP header to forward datagrams.

TCP & Reliable Delivery

TCP recovers from:

- lost packets;
- duplicate packets;
- delayed packets;
- corrupted data;
- transmission speed mismatches;
- congestion;
- system reboots.

TCP Segments & Sequence Numbers

- An application sends arbitrarily large chunks of data to TCP as a stream.
- TCP breaks the data into segments, each of which fits into an IP datagram.
- Each segment contains a sequence number for the data being carried by that segment.
- The receiver uses the sequence numbers to reorder segments that arrive out of order.

Lost Packets

- TCP uses positive acknowledgment with retransmission to achieve reliable data delivery.
- The receiver sends an acknowledgment (ACK) control message back to the sender to verify successful receipt of data.
- The ACK appears in a segment as an acknowledgment number specifying the sequence number of the received data.
- The sender sets a timer when data is transmitted. If the timer expires before an ACK is received, the sender retransmits.

Setting the Timeout

- Inappropriate timeout can cause poor performance.
  - Too long: The sender waits longer than necessary before retransmitting.
  - Too short: The sender generates unnecessary traffic.
- Timeout values must be different for each connection and set dynamically.
  - Hosts on a LAN should have shorter timeouts than hosts 20 hops away.
  - Delivery times on an internet can change.

Setting the Timeout

- TCP uses adaptive retransmission.
  - Timeout should be based on round-trip time (RTT).
  - However, a sender doesn’t know the actual RTT of a packet before it’s transmitted.
  - So a sender picks a retransmission timeout (RTO) based on previous RTTs.
  - Various algorithms are used to estimate the current RTT and set a current RTO.

TCP Sliding Window

- TCP uses a sliding window for flow control.
- Along with an ACK, a receiver specifies its current window size in bytes with a window acknowledgment.
- The sender can transmit any amount of data in any size segment up to the window acknowledgment.

TCP Headers

- A TCP segment has a more complicated header than a UDP datagram.
- The same header format is used in both the send and receive directions.
- Thus, a segment can carry both outgoing data and an acknowledgment and window advertisement for incoming data.

Three-Way Handshake

- TCP uses three-way handshake for reliable connection establishment and termination.

Congestion Control

- Packet loss is more likely caused by congestion than hardware failure.
- Transport protocols that retransmit can exacerbate congestion and lead to congestion collapse.
- To avoid congestion collapse, TCP always interprets packet loss as congestion.
- When a packet is lost, TCP begins congestion control and slows the transmission of packets.

Congestion Control

- When entering congestion control mode, TCP sends a single packet.
- If acknowledgment is received, TCP sends 2 more packets.
- If acknowledgment is received, TCP sends 4 more packets.
- This exponential increase continues until half the advertised window size is reached, at which time TCP slows the rate of increase.

General Comments

- TCP is designed for reliable transmission of data.
  - If data is lost or damaged, TCP ensures that data is resent.
  - If packets arrive out of order, TCP puts them back in the correct order.
  - If data is coming too fast, TCP throttles back the speed.
  - A program never needs to worry about receiving incorrect or out-of-order data.
  - TCP’s reliability comes at a price: speed.

General Comments

- UDP is an alternate but unreliable protocol.
  - There's no way off knowing when or even if UDP data arrives.
  - There's no way of knowing if the data arrives in order.
  - Pieces that do arrive generally arrive more quickly.

Why Would You Ever Want to Use an Unreliable Protocol?

- If data is worth sending, shouldn't you care about whether it arrives correctly?
- UDP is not a good choice for applications like file transfer (FTP) that require reliable transmission.
- UDP is a good choice for applications requiring raw speed.
  - In real-time audio or video, lost or swapped packets appear simply as static.
  - Static is tolerable, but long pauses while TCP corrects bad data are intolerable.

An Analogy

- TCP is like the telephone system.
  - When you dial a number, the phone is answered and a connection established.
  - As you talk, the other party hears your words in the order you say them.
  - If the phone is busy or no one answers, you find out right away.
- UDP is like the postal system.
  - You send packets of mail to an address.
  - Most of the letters arrive, but some may be lost.

An Analogy

- The letters probably arrive in the order you mailed them, but that’s not guaranteed.
- The further away the recipient, the more likely letters will be lost or arrive out or order.
- If this is a problem, you can write numbers on the envelopes and ask the recipient to arrange them in the proper order and send you mail telling you which ones arrived so that you can resend those that didn’t. However, you and the recipient must do this; the post office won’t.