

User Datagram Protocol

Advantages:

- Small packet size (UDP header 8 bytes, TCP header 16 bytes)
- No connection first before sending out data
- More control over when the data is sent

Disadvantages:

- UDP doesn't try to recover after corruption. The corrupted segment is discarded. Sometimes it can return only warning flag to the application - *no compensation for lost packets*
- *Packets can arrive out of order*
- *No congestion control* (перегрузка). UDP will try to cram those packets. Its a bad strategy because packets get dropped more often.

Transmission Control protocol

Advantages:

- Negotiate the connection first before doing anything - *three way handshake*.
 - Initiator asks the acceptor if it wants to set up a connection. Acceptor sends a reply, initiator receives it, then acceptor sends packets. The similar thing when connection is closed.
- *Delivery acknowledgement* - receiver acknowledge that it has got the data. TCP segments carry a number for this.
- TCP provides *retransmission* - if sender doesn't get a delivery acknowledgement within a certain amount of time, it will assume that the packet got lost, so it will send it again.
- *In-order delivery* - packets can arrive out of order, but TCP rearranges them.
- *Congestion control* - delay the transmission of data when the network is congested.

Disadvantages:

- *Bigger header*
- *Data doesn't always get sent out immediately* - side effect of congestion control.
- *Bigger overhead* - retransmission of packets, delivery acknowledgements.

Examples

Text communication - **TCP** (in-order, retransmission)

File Transfer - **TCP**

Remote protocols such as SSH - **TCP**

Multimedia streaming - **UDP** (no delay, data loss can be masked, less overhead)

Small transactions, DNS lookups - **UDP** (no need to create connection first)

Bandwidth-intensive apps that tolerate packet loss - **UDP**