PROBLEM

Sender A wants to transmit 10 segments to Receiver B. The hosts are agreed to go with Go-Back-4. How many numbers of segments are transmitted by Sender A if every 6th segment that is transmitted by Sender A is either corrupted or lost. Also compare the number of transmissions of Go-Back-4 with Selective Repeat protocol using same window size.

0 1 2 3 4 5 6 7 8 5 6 7 8 9 7 8 9 = 17

0 1 2 3 4 5 6 7 8 5 9 = 11

Connectionless Transport: UDP

- UDP, defined in [RFC 768], aside from the multiplexing/demultiplexing function and some light error checking, it adds nothing to IP.
- UDP takes messages from the application process, attaches source and destination port number fields for the multiplexing/demultiplexing service, adds two other small fields, and passes the resulting segment to the network layer.
- The network layer encapsulates the transport-layer segment into an IP datagram and then makes a best-effort attempt to deliver the segment to the receiving host.
- If the segment arrives at the receiving host, UDP uses the destination port number to deliver the segment's data to the correct application process.
- With UDP there is no handshaking between sending and receiving transport-layer entities before sending a segment. For this reason, UDP is said to be connectionless.

Connectionless Transport: UDP

- DNS is an example of an application-layer protocol that typically uses UDP.
- When the DNS application in a host wants to make a query, it constructs a DNS query message and passes the message to UDP.
- Without performing any handshaking with the UDP entity running on the destination end system, the host-side UDP adds header fields to the message and passes the resulting segment to the network layer.
- The network layer encapsulates the UDP segment into a datagram and sends the datagram to a name server.

Connectionless Transport: UDP

Isn't TCP always preferable, since TCP provides a reliable data transfer service, while UDP does not?

1. Finer application-level control over what data is sent, and when.

- Under UDP, as soon as an application process passes data to UDP, UDP will package the data inside a UDP segment and immediately pass the segment to the network layer.
- TCP, on the other hand, has a congestion-control mechanism that throttles the transport-layer TCP sender when one or more links between the source and destination hosts become excessively congested.
- TCP will also continue to resend a segment until the receipt of the segment has been acknowledged by the destination.

Connectionless Transport: UDP

Isn't TCP always preferable, since TCP provides a reliable data transfer service, while UDP does not?

2. No connection establishment

- TCP uses a three-way handshake before it starts to transfer data.
- UDP just blasts away without any formal preliminaries. Thus UDP does not introduce any delay to establish a connection.
- This is probably the principal reason why DNS runs over UDP rather than TCP—DNS would be much slower if it ran over TCP.

Connectionless Transport: UDP

Isn't TCP always preferable, since TCP provides a reliable data transfer service, while UDP does not?

3. No connection state

- TCP maintains connection state in the end systems. This connection state includes receive and send buffers, congestion-control parameters, and sequence and acknowledgment number parameters.
- UDP does not maintain connection state and does not track any of these parameters.
- For this reason, a server devoted to a particular application can typically support many more active clients when the application runs over UDP rather than TCP.

Connectionless Transport: UDP

Isn't TCP always preferable, since TCP provides a reliable data transfer service, while UDP does not?

4. Small packet header overhead

- The TCP segment has 20 bytes of header overhead in every segment, whereas UDP has only 8 bytes of overhead.

Connectionless Transport: UDP

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP	TCP
Remote terminal access	Telnet	TCP
Web	HTTP	TCP
File transfer	FTP	TCP
Remote file server	NFS	Typically UDP
Streaming multimedia	typically proprietary	UDP or TCP
Internet telephony	typically proprietary	UDP or TCP
Network management	SNMP	Typically UDP
Routing protocol	RIP	Typically UDP
Name translation	DNS	Typically UDP

Figure 2.33 ◆ Popular Internet applications and their underlying transport protocols

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Connectionless Transport: UDP

UDP Segment Structure

- The UDP header has only four fields, each consisting of two bytes.
- The length field specifies the number of bytes in the UDP segment (header plus data).
- An explicit length value is needed since the size of the data field may differ from one UDP segment to the next.
- The checksum is used by the receiving host to check whether errors have been introduced into the segment.

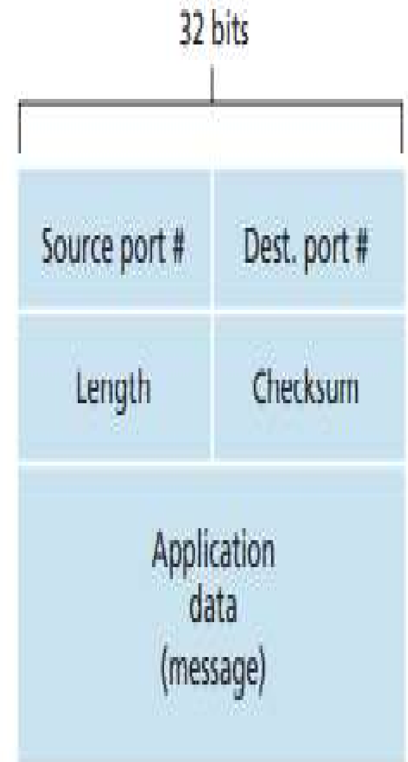

Figure 2.34 ♦ UDP segment structure

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Connectionless Transport: UDP

UDP Checksum

- The UDP checksum provides for error detection.
- Checksum is used to determine whether bits within the UDP segment have been altered as it moved from source to destination.
- UDP at the sender side performs the 1s complement of the sum of all the 16-bit words in the segment, with any overflow encountered during the sum being wrapped around.
- This result is put in the checksum field of the UDP segment.
- At the receiver, all 16-bit words are added, including the checksum. [Sum+checksum(1s complement of Sum)]
- If no errors are introduced into the packet, then clearly the sum at the receiver will be 1111111111111111.
- If one of the bits is a 0, then we know that errors have been introduced into the packet.

Connectionless Transport: UDP

UDP Checksum Example

Three 16-bit words
0110011001100000
0101010101010101
0000111100001100

The sum of first two of these 16-bit words is
0110011001100000 +
0101010101010101

1011101110110101

Adding the third word to the above sum gives
1011101110110101 +
0000111100001100

1100101011000010

1s complement of the sum 1100101011000010 is
0011010100111101 checksum

1100101011000010 + sum
0011010100111101 checksum

1111111111111111

Connectionless Transport: UDP

How 1's Complement Checksum Works

1. Divide the Data into 16-bit Chunks
   - If the data is not a multiple of 16 bits, pad it with zeros.
2. Sum All 16-bit Words Using 1's Complement Arithmetic
   - Add all 16-bit words.
   - If there's an overflow (carry beyond 16 bits), add the carry back to the sum.
3. Take the 1's Complement of the Sum
   - Invert all bits (flip 0s to 1s and 1s to 0s).
   - This becomes the checksum.
4. Verification at the Receiver
   - The receiver adds all 16-bit words (including the checksum).
   - A correct packet should result in 0xFFFF (all 1s).

     1111 1111 1111 1111
   - If the sum is not 0xFFFF, there is an error.

Connectionless Transport: UDP

Qn: A sender has two data items to send:

1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 and

1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

Compute checksum for the data.