Ques 1: Difference b/w TCP header and UDP header

TCP (Transmission Control Protocal) and UDP (User datagram Protocal) are both transport-layer protocals.

1) TCP Header:

- TCP is ~~contio~~ connection-oriented.
- Provides reliable data transfer.
- Performs error recovery, flow control, and congestion control.
- Therefore, the TCP header is larger (20-60 bytes).

UDP header:

- UDP is connectionless.
- Provides best-effort, unreliable delivery.
- Does not provide flow control, sequencing, or retransmission.
- Therefore, the UDP header is very small (8 bytes).

Fields in TCP header that are Missing in UDP header:

1- Sequence Number.
2- Acknowledgment Number
3- Data Offset.
4- Reserved bits
5- Flags/Control bits (URG, ACK, PSH, RST, SYN, FIN)
6- Window Size
7- Cheak sum (UDP also has, so not missing)
8- Urgent Pointer.
9- Options and Padding.

UDP header only has 4 Fields:

- Source Port
- Destination Port.
- Length
- Cheak sum.

Reason for Absence:

→ UDP provides a simple, fast, connectionless service.

→ It doesnot estabish a connection, does not guarantee delivery, does not reorder, and does not control flow.

→ No options and padding field.

→ UDP doesnot use 3-way handshake.

Ques #2

$$\text{Efficiency} = \frac{\text{Useful Data Bytes}}{\text{Total Bytes Transmitted at UDP level.}}$$

- Data = 16 bytes
- UDP header = 8 bytes.
- Total Bytes transmitted at UDP level

$$\text{Total} = \text{Data} + \text{UDP header} = 16 + 8 = 24 \text{ bytes.}$$

$$\text{Efficiency} = \frac{16}{24} = 0.667$$

Efficiency = 66.7 %.

Ques 3 Interpreting TCP control field Values.

Total control bits field = 6 bits:
URG, ACK, PSH, RST, SYN, FIN

(a) 000000

- No control bit set.
- Indicates a normal data segment with no special function.

(b) 000001

- Bit pattern (from left: URG, ACK, PSH, RST, SYN, FIN):

→ last bit = FIN = 1

- This is a FIN segment, used to terminate a TCP connection gracefully.

(c) 01000

- Given a 5-bits, but assuming right-aligned in 6-bit field → 001000 ← Assume left zero padding.

PSH = 1

- This is an ACK segment, used to acknowledge received data or control segments.

Ques 4 Wrong destination port in TCP segment.

An IP datagram successfully reaches the correct host (same network 130.14.0.0/16), but the destination port inside the TCP header is corrupted.

What does receiving host do?

When IP data gram arrives:

1. IP hands the segment to TCP, based on protocol number (6 for TCP).
2. TCP cheaks the destination port number.

TCP behaviour in situation?

According to TCP portocol rule:

TCP sends a RESET (RST) segment back to sender.

- Receiving hosts sends a TCP RST packet to the source.
- The ~~Receiving~~ Receiving host checks its TCP port table.
- its find no application is listening on that corrupted port.

~~Why RST sent?~~

Why RST is sent:

Bcz:

- The segment is not part of existing connection.
- No valid application is bound to corrupted destination port.