**Exp: no: 6**

Performance analysis of TCP and UDP protocol.

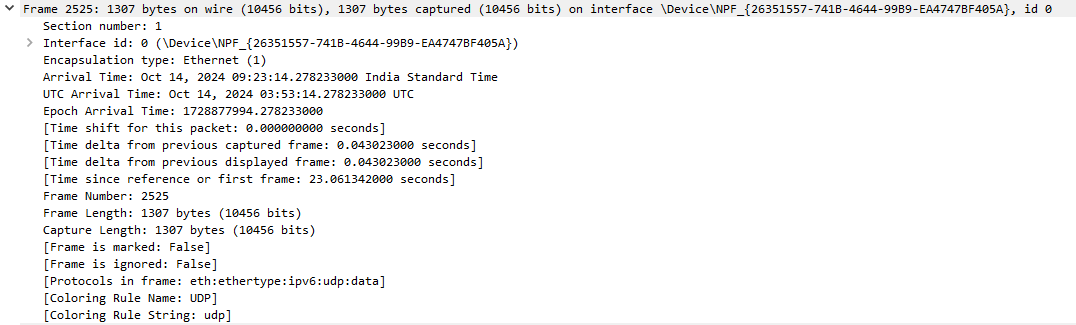
**Aim:**

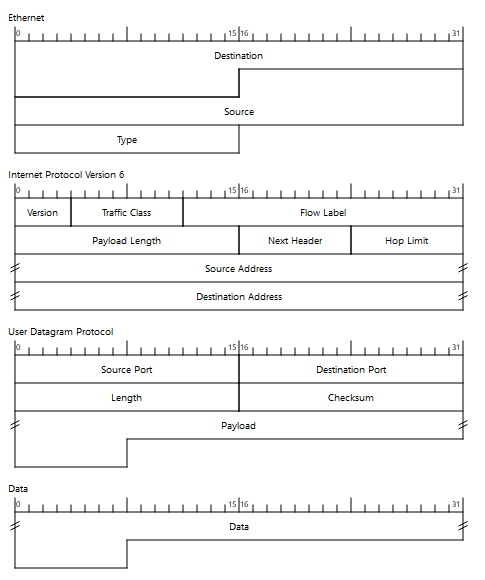
To analyse the performance of TCP and UDP protocols by comparing their data segments, frame formats, and key differences.

1. Draw the UDP frame format.

**Procedure:**

1. **Install and Open Wireshark**:
   1. Ensure Wireshark is installed on your system and launch it.
2. **Start a Capture**:
   1. Select the appropriate network interface (e.g., Wi-Fi or Ethernet) where you expect the UDP traffic to be.
   2. Click the **Start** button to begin capturing network traffic.
3. **Filter for UDP Packets**:
   1. In the **filter bar** at the top of the Wireshark window, enter the filter **udp** and press **Enter**. This filter will show only UDP packets.
4. **Initiate a UDP Transmission**:
   1. Open a network application that uses UDP (e.g., a video streaming service or VoIP application) to generate UDP traffic.
5. **Capture and Stop**:
   1. Once the UDP traffic is visible in Wireshark, click the **Stop** button to halt the capture.
6. **Select a UDP Packet**:
   1. From the captured packets list, select any UDP packet to analyze. Click on it to see its details.
7. **Examine the UDP Frame Format**:
   1. In the **Packet Details** pane, expand the **User Datagram Protocol (UDP)** section to view the fields of the UDP frame:
      1. **Source Port**: Port from which the packet was sent.
      2. **Destination Port**: Port to which the packet is being sent.
      3. **Length**: The length of the UDP header and the encapsulated data.
      4. **Checksum**: Used for error checking.
   2. You can also view the **Data Payload** carried by the UDP packet.
8. **Screenshot**: Take a screenshot of the UDP frame format for future reference or reports.

**Output:**



1. List out the differences between TCP and UDP.

|  |  |  |
| --- | --- | --- |
| **Feature** | **TCP (Transmission Control Protocol)** | **UDP (User Datagram Protocol)** |
| **Connection Type** | Connection-oriented (Requires handshake) | Connectionless (No handshake required) |
| **Reliability** | Provides reliability through error checking, acknowledgments, and retransmissions. | No reliability, no retransmissions. Once sent, data is not guaranteed to arrive. |
| **Flow Control** | Uses flow control to manage data transmission rate | No flow control mechanisms |
| **Congestion Control** | Employs congestion control algorithms like TCP Reno, TCP Tahoe | No congestion control |
| **Overhead** | Higher due to error-checking, sequencing, and acknowledgment mechanisms | Lower overhead due to minimal header fields |
| **Use Cases** | File transfer (e.g., FTP), web browsing (e.g., HTTP), email (e.g., SMTP) | Real-time services like video streaming, VoIP, DNS queries, and online gaming |
| **Speed** | Slower due to connection setup and error-checking | Faster as no connection setup is required |
| **Data Segmentation** | Segments data into smaller packets and reassembles them at the receiver's end | No data segmentation, sends data in the form of datagrams |
| **Error Detection** | Provides error detection and correction | Error detection is limited (only checksum) |
| **Header Size** | 20-60 bytes | 8 bytes |

1. Analyse the TCP and UDP data segments with the data parameters.

**Procedure for TCP Segment Analysis:**

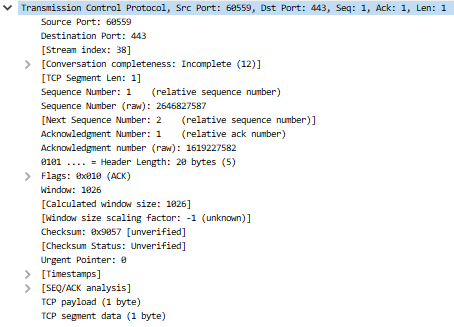
1. **Start Wireshark and Capture Network Traffic**:
   1. Select your network interface and click the **Start** button to begin capturing packets.
2. **Filter for TCP Traffic**:
   1. In the **filter bar**, type **tcp** and press **Enter** to display only TCP traffic.
3. **Generate TCP Traffic**:
   1. Open a TCP-based application (e.g., a web browser, download a file via HTTP, or initiate a file transfer using FTP) to generate TCP traffic.
4. **Capture and Stop**:
   1. Once you have some TCP packets captured, click **Stop**.
5. **Select a TCP Packet**:
   1. Select a TCP packet from the list of captured packets and click on it to see the detailed breakdown.
6. **Examine the TCP Data Segment**:
   1. In the **Packet Details** pane, expand the **Transmission Control Protocol (TCP)** section to view the following fields:
      1. **Source Port & Destination Port**: Identifies the sending and receiving applications.
      2. **Sequence Number**: Ensures data packets are received in order.
      3. **Acknowledgment Number**: Used to acknowledge the successful receipt of data.
      4. **Flags**: Shows TCP control flags (SYN, ACK, FIN, etc.).
      5. **Window Size**: Indicates how much data the receiver can accept.
      6. **Checksum**: Ensures data integrity.
   2. **Data Payload**: You can also analyze the data segment being transmitted by the TCP packet (i.e., the actual data from the application layer).
7. **Take Notes/Screenshot**: Take a screenshot of the segment analysis or record important data parameters.

**Procedure for UDP Segment Analysis:**

1. **Start Wireshark and Capture Network Traffic**:
   1. As before, select your network interface and click **Start**.
2. **Filter for UDP Traffic**:
   1. In the **filter bar**, type **udp** and press **Enter** to focus on UDP traffic.
3. **Generate UDP Traffic**:
   1. Open a UDP-based application (e.g., online gaming, VoIP service, or DNS queries).
4. **Capture and Stop**:
   1. Once UDP packets are visible, click **Stop** to halt the capture.
5. **Select a UDP Packet**:
   1. Choose any UDP packet from the capture list and click on it to view its details.
6. **Examine the UDP Data Segment**:
   1. In the **Packet Details** pane, expand the **User Datagram Protocol (UDP)** section to view:
      1. **Source Port & Destination Port**: The application ports involved in communication.
      2. **Length**: Total length of the UDP packet, including the header and payload.
      3. **Checksum**: Ensures data integrity for error detection.
   2. **Data Payload**: You can inspect the application data encapsulated within the UDP segment.
7. **Take Notes/Screenshot**: Like before, capture a screenshot or take notes for documentation.

**Output:**

TCP:

**

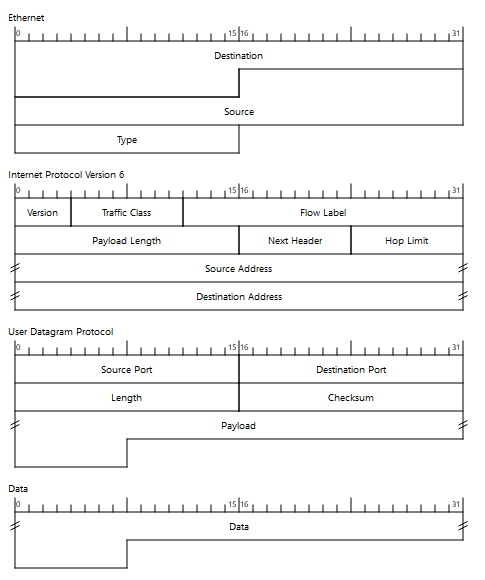
*A diagram of a number

Description automatically generated with medium confidence*

UDP:

*A screenshot of a computer

Description automatically generated*



**Result:**

The performance of TCP and UDP protocols was successfully analysed by examining their frame formats, data segments, and identifying the key differences.