

## EXP:05

## Experiments on Packet capture tool: Wireshark

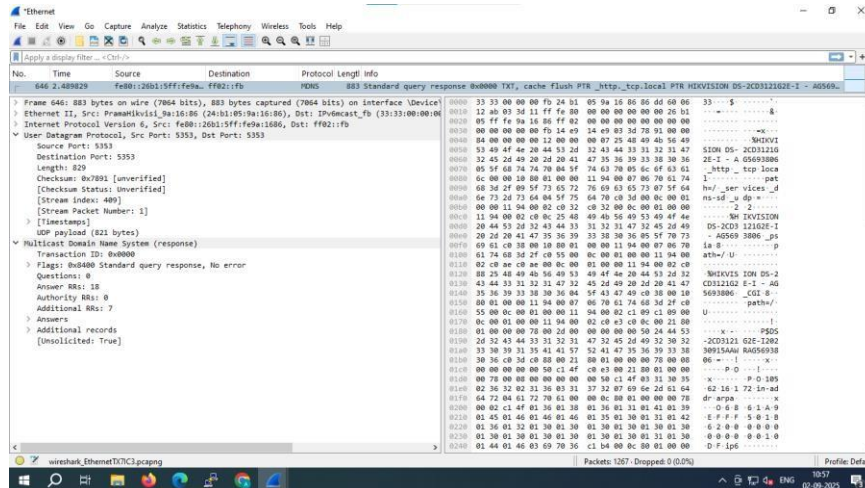
### Aim

To understand the features of Wireshark as a packet capture tool and analyze the encapsulation of information at various layers of the Protocol stack.

### Algorithm / Procedure

1. **Install** and launch the Wireshark packet capture tool.
2. **Select** the appropriate network interface for capturing traffic.
3. **Start** a packet capture session.
4. **Generate** network traffic (e.g., browse a website, ping a host).
5. **Stop** the capture and **analyze** the captured packets, focusing on the header information at the Data Link, Network, and Transport layers to understand encapsulation.
6. **Apply** filters (e.g., http, tcp, ip.addr == x.x.x.x) to isolate specific traffic.

### Output Images



## Result

Wireshark was successfully used to capture and analyze network traffic. The process of protocol encapsulation and the structure of packet headers at different layers were observed and understood.

EXP:06

## Error Correction at Data Link Layer

## Aim

To write a program to implement error detection and correction using the **Hamming Code** concept.

## Algorithm / Procedure

- Determine** the number of redundant (parity) bits required for the given data size.
- Calculate** the positions of the parity bits (powers of 2).
- Implement** the Hamming Code generation algorithm:
  - Place data bits and parity bits in their respective positions.
  - Calculate the value of each parity bit based on the data bits it covers.
- Implement** the error detection and correction algorithm:
  - Receive the transmitted codeword.
  - Recalculate the parity bits.
  - Calculate the syndrome (error position) by combining the recalculated parity bits.
  - If the syndrome is non-zero, flip the bit at the error position.