Indian Institute of Technology, Kharagpur
Network Laboratory CS39006
Spring 2024-2025

Assignment 7: Design and Implement a CLDP using Raw Sockets

Name: Golla Meghanandh Manvith Prabhash
Roll no.: 22CS30027

Implementation Report

1. Introduction

The **Custom Lightweight Discovery Protocol (CLDP)** is a lightweight, custom network protocol built over raw IP sockets for intra-network host discovery and system information exchange. It allows hosts to announce their presence (via HELLO messages), respond to information requests (QUERY/RESPONSE), and dynamically interact over a local network without relying on standard higher-layer protocols.

This report documents the design, structure, and implementation details of CLDP, developed as part of a networking lab assignment.

2. Objectives

- To implement a discovery-based custom protocol over raw IP.
- To support custom packet formats and handling.
- To dynamically gather and return system information.
- To work entirely using raw sockets and protocol number 253 (user-defined).

3. Protocol Design

3.1 Message Types

CLDP defines three message types:

Type	Code	Purpose
HELLO	0x01	Broadcast announcement from server
QUERY	0x02	Request from client to server
RESPONSE	0x03	Response from server to client

3.2 CLDP Packet Format

Each packet consists of:

- A complete **IP Header** (manually constructed, with protocol field 253).
- A **CLDP Header**, followed by an optional payload.

CLDP Header (8 bytes total):

Field	Size	Description
msg_type	1 B	Type of message (HELLO, QUERY, etc.)
payload_len	1 B	Length of payload in bytes
transaction_id	2 B	Unique ID to map QUERY and RESPONSE
reserved	4 B	Reserved (set to 0)

Payload:

- **QUERY:** A string containing a combination of characters:
    - H = Hostname
    - T = Timestamp
    - S = System Info
- **RESPONSE:** Textual system information based on QUERY payload.

4. Implementation Details

4.1 Server (cldp_server.c)

- Sends **HELLO** messages every 10 seconds via broadcast (255.255.255.255).

- Listens for **QUERY** messages:
    - Parses the payload.
    - Retrieves system data accordingly using:
        - gethostname() – for Hostname.
        - gettimeofday() – for Timestamp.
        - uname() – for System Info.
- Constructs a RESPONSE and sends it back using sendto().

4.2 Client (cldp_client.c)

- Waits for HELLO messages for 10 seconds and builds a list of discovered servers (using a linked list).
- Sends **QUERY** messages to each discovered IP with a selected payload (based on user input index 0–6).
- Waits for **RESPONSE** messages and prints out the returned information.

4.3 QUERY Payload Options

Index	Payload	Information Requested
0	"H"	Hostname only
1	"T"	Timestamp only
2	"S"	System Information only
3	"HT"	Hostname + Timestamp
4	"HS"	Hostname + System Information
5	"TS"	Timestamp + System Information
6	"HTS"	Hostname + Timestamp + System Info

5. Key Features

- Uses raw sockets and full control over IP header creation (IP_HDRINCL).
- Implements a custom header and parsing logic.
- Sends broadcast HELLOs and receives unicast RESPONSEs.
- Dynamic query system with flexible payload support.
- Linked list to store and manage host IPs discovered via HELLO.

7. Assumptions and Limitations

- Designed and tested on **Linux** (Ubuntu) using **GCC**.
- Requires **root privileges** due to use of raw sockets.
- Only supports IPv4 local broadcast for HELLO discovery.
- No authentication, encryption, or retransmission mechanisms are implemented.
- Case-insensitive query parsing (e.g., "HtS" == "HTS").

8. Conclusion

The CLDP protocol provides a custom and educational approach to understanding raw socket communication, packet structure, and discovery mechanisms. It mimics basic aspects of network protocols and provides a strong foundation for extending functionality in network management, monitoring tools, or service discovery modules.

9. References

- RFC 791 – Internet Protocol
- RFC 1071 – Computing the Internet Checksum
- Linux Raw Sockets Documentation