Computer Networks Major Task

Phase_1

Mini-RFC

Submitted by (Team N0.13):

Abdelrhman Mohammed Mahmoud	23P0370
Abdelrahman Mohamed Bahig	23P0390
Ahmed Mostafa Gomaa Atia	23p0375
Nagy Ahmed Nagy	23p0365
Abdallah Omar Moustafa Taman	22P0129
Kareem Younis Ahmed Foad Yehya	22P0136

1.Introduction:

LiteTelemetry Protocol (LTP) is a lightweight application-layer protocol designed for Internet-of-Things (IoT) devices that periodically transmit small telemetry readings (e.g., temperature, humidity, voltage) to a central collector or gateway.

LTP focuses on low latency, low overhead, and robustness in networks with moderate packet loss, making it ideal for constrained devices and embedded systems. It provides a simple yet efficient mechanism for data transfer, monitoring, and synchronization between IoT nodes and servers.

Key Design Principles:

- Compact binary messages (header ≈ 12 bytes)
- Operates over UDP to minimize connection and retransmission overhead
- Loss-tolerant by design (~5% packet loss acceptable)
- Easily parsed and logged by collectors
- Extensible structure to support future message types (e.g.,CONTROL,CONFIG)

2. Protocol Architecture

2.1 Entities

	ROLE	DESCRIPTION
SENSOR CLIENT	Sender	Periodically measures environmental data and sends packets (DATA, HEARTBEAT) to the collector.
COLLECTOR SERVER	Receiver	Receives, validates, and logs packets; detects duplicates, missing data, and reorders delayed packets.

2.2 Communication Flow

1. INIT Phase:
- When the sensor starts, it sends an INIT message containing its ID and configuration.
- The server acknowledges with an INIT-ACK, registering the client.

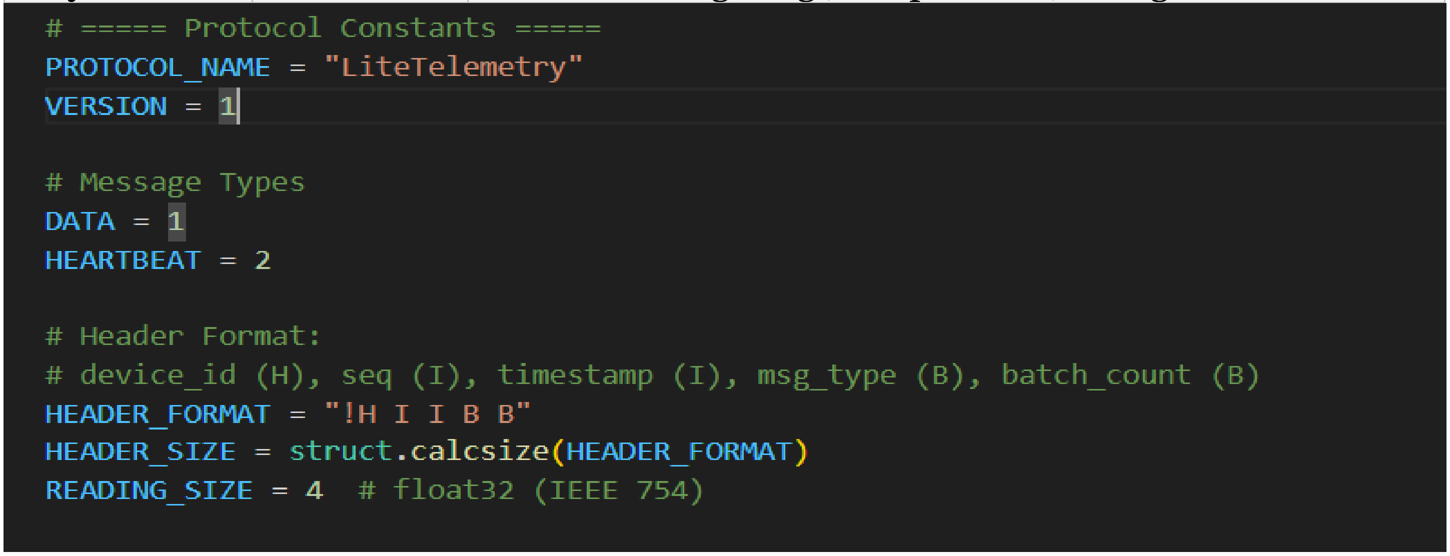
2. Operational Phase:
- Sensor sends DATA messages periodically with telemetry values.
- A HEARTBEAT message is sent when no new data is available.
- Server logs incoming packets and computes loss or delay statistics.

3. Optional Control Flow:
- The server can send a RESET or CONFIG message to update client behavior.

3. Message Formats

Each LTP message begins with a fixed 12-byte header followed by optional payload data (e.g., sensor readings).

Field Name	Size (bits)	Description
Version	4	Protocol version (e.g., 1)
MsgType	4	Message type: 0=INIT, 1=DATA, 2=HEARTBEAT, 3=CONFIG
DeviceID	16	Unique identifier for each IoT device
SeqNum	16	Incremental sequence number (detects loss/duplicates)
Timestamp	32	UNIX time when the reading was captured
Flags	8	Optional flags (e.g., checksum, batching, priority)
Payload	Variable	Sensor readings (e.g., temperature, voltage)



```
# ===== Protocol Constants =====
PROTOCOL_NAME = "LiteTelemetry"
VERSION = 1

# Message Types
DATA = 1
HEARTBEAT = 2

# Header Format:
# device_id (H), seq (I), timestamp (I), msg_type (B), batch_count (B)
HEADER_FORMAT = "!H I I B B"
HEADER_SIZE = struct.calcsize(HEADER_FORMAT)
READING_SIZE = 4  # float32 (IEEE 754)
```