NanoTelemetry Protocol v1.0 (NTP-v1)

Draft Mini-RFC (Sections 1–3)
Status: Prototype / Phase 1
Authors: Team 22 (Computer Networks Project)
Date: November 2025

---

1. Introduction

NanoTelemetry Protocol v1.0 (NTP-v1) is a lightweight, connectionless telemetry protocol for constrained IoT sensors that periodically send small numeric readings to a central collector.

The design goals are:
• Minimal packet overhead (≤ 12-byte header).
• Operation over UDP only (no retransmission).
• Detection of duplicates and sequence gaps.
• Simple extensibility via a flags field.

Assumptions and Constraints:
• Transport = UDP.
• Maximum application payload ≤ 200 bytes.
• Typical reporting intervals = 1 s, 5 s, 30 s.
• Expected loss ≈ ≤ 5%; no retransmission required.
• Each device has a unique Device ID (1 byte in prototype).

2. Protocol Architecture

Entities	
Sensor (Client):	Sends periodic DATA packets containing sensor readings and occasional HEARTBEAT messages.
Collector (Server):	Listens on UDP port 12000, records packets to CSV, tracks sequence numbers, and flags duplicates or gaps.

3. Message Formats

3.1 Header Layout

Binary Header Structure (12 bytes total)

Field	Size(bits)	Offset (bytes)	Description
Version	8	0	Protocol version (1 = v1)
MsgType	8	1	1 = DATA, 2 = HEARTBEAT
Flags	8	2	Reserved for future use (batch mode, etc.)
DeviceID	8	3	Unique sensor identifier
SeqNum	32	4	Monotonic sequence number per device
Timestamp	32	8	Unix epoch seconds (at send time)

3.2 Payload Formats

Message Type	Payload Format	Description
DATA	1 float (32 bits, IEEE-754)	Sensor reading (e.g., temperature)
HEARTBEAT	None	Indicates device liveness

Encoding Example (Python struct format):

HEADER_FMT	'!BBBBII' (12 bytes total)
DATA_FMT	'!f' (float payload)

Sample Packet

Version	1
MsgType	1(DATA)
Flags	0
DeviceID	1
SeqNum	42
Timestamp	1730913600
Payload	25.42

3.3 Example Exchange

1. **Sensor → Server**: DATA (seq = 1, value = 25.3)
2. **Sensor → Server**: DATA (seq = 2, value = 25.1)
3. **Sensor → Server**: HEARTBEAT (seq = 3)
4. Server logs each packet in CSV and updates sequence tracker.