Lightweight Boolean Heartbreak Signaling with SDT and Adaptive Gain Factors for 5G AI-Driven NWDAF Layers

Enabling Scalable AI Monitoring via Minimal-Bandwidth LHB Health Signaling

Sreenivas Chandran
Department of Computer Science and Engineering
SRM Institute of Science and Technology
Email: sreenivaschandran@gmail.com
Alternate Email: sc3316@srmist.edu.in
ORCID: 0009-0002-4179-8090

Dr. Revathi Venkataraman
Department of Computer Science and Engineering
SRM Institute of Science and Technology
Email: revathin@srmist.edu.in

***Abstract*—Fifth-generation (5G) networks enable ultra-low latency, high bandwidth, and massive connectivity for mission-critical applications. Their microservice-based core requires continuous subsystem-level health monitoring to ensure reliability and performance. Emerging Beyond-5G and 6G architectures integrate AI-driven analytics through the Network Data Analytics Function (NWDAF), enabling predictive assurance and adaptive optimization. However, centralized AI pipelines often accumulate redundant telemetry, increasing inference latency, compute load, and cooling demands—thereby amplifying overall energy consumption. This paper presents LHB-X (Lightweight Heartbreak signaling), a novel engineered sustainable health-bit framework that reduces telemetry overhead while preserving detection fidelity. The Heartbreak Signal denotes a Boolean health-signal bit of subsystem well-being, generated at the source using surprisal-based scoring and refined through adaptive gain feedback from the NWDAF layer.**

Local edge-side computation ranges from $O(m)$ in structurally concrete subsystems to $O(m^3)$ in structurally complex correlation modes, while remaining lightweight for typical 5G deployments. Crucially, the exported LHB-X message is fixed-size, ensuring constant-time ($O(1)$) processing at the NWDAF layer regardless of subsystem structure or count. Comparative evaluation against baseline telemetry (raw, threshold-based, and sampled) demonstrates reduced bandwidth usage, lower CPU utilization, and stable true-positive rates under identical thresholds. By minimizing redundant data transfer, compute cycles, and cooling demands, LHB-X achieves measurable energy gains and supports dynamic, closed-loop optimization via NWDAF. The framework aligns with 6G sustainability goals and provides a deployable path toward efficient, AI-assisted monitoring across 5G/6G, IoT, and cyber-physical systems.

***Index Terms*—5G Core, 6G, NWDAF, Network Data Analytics Function, Boolean health signaling, Constant-time fault detection, Surprisal-based anomaly detection, Adaptive gain feedback, Network Digital Twin (NDT), Edge, Sustainable AI monitoring, Environmental efficiency, Heartbreak signal (HB), Closed-loop control, Cyber–physical systems, Industrial IoT, Operational Intelligence (OI)**

I. Introduction

Fifth-generation (5G) networks deliver ultra-low latency, high throughput, and massive connectivity for applications such as industrial automation, extended reality (XR), and real-time media [1]. Their microservice-based core comprises modules such as the Access and Mobility Function (AMF), Session Management Function (SMF), and User Plane Function (UPF), all of which require continuous subsystem-level health monitoring to ensure reliability. Failures in individual components can compromise end-to-end service continuity.

Beyond 5G, 6G research emphasizes integrating Artificial Intelligence (AI) through the Network Data Analytics Function (NWDAF) [2], [3] for predictive assurance and digital-twin optimization [5]. However, centralized AI monitoring streams large volumes of raw telemetry, inflating bandwidth usage, compute overhead, and cooling demand, thereby increasing overall energy consumption [6], [7]. The challenge intensifies in distributed deployments where long-distance transfers further add to latency [4].

To address these inefficiencies, we propose **LHB-X (Lightweight Heartbreak Signaling)**, a compact and adaptive framework for sustainable monitoring. The term **Heartbreak** refers to a Boolean **health-signal bit** representing subsystem well-being, enabling the source to offload redundant telemetry before it reaches AI pipelines. At the edge, LHB-X modules compute surprisal-based severity scores using Signal Detection Theory (SDT) and convert them into Boolean indicators [10], [11]. Within the core, NWDAF provides adaptive gain feedback that dynamically tunes sensitivity and stabilizes responses [2], [3]. Fig. 1 illustrates this distributed feedback loop.

LHB-X supports constant-time ($O(1)$) message interpretation at the NWDAF layer, while edge-side processing ranges from $O(m)$ (logical aggregation) to $O(m^3)$ (optional full-state correlation). In practical 5G Network Functions, subsystem groups remain small (typically $m \approx 5$–16), keeping edge computation lightweight while preserving overall constant-time signaling.

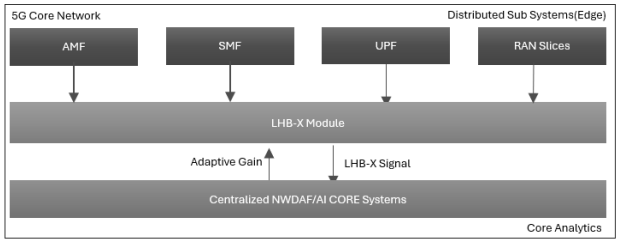

Fig. 1. LHB-X architecture overview

The main contributions of this paper are as follows:

- **Lightweight monitoring:** Integration of Boolean Heartbeak signaling, adaptive gain feedback, and surprisal-based risk scoring for AI-assisted reliability.
- **Compact schema:** Header–body message design with a 3-bit *Type* field enabling constant-time ($O(1)$) updates at the analytics layer.
- **Scalable complexity:** Edge-side processing supports $O(m)$–$O(m^3)$ modes while preserving constant-time downstream interpretation.
- **Cross-domain applicability:** Extendable to 6G, IoT, and cyber–physical systems requiring rapid, energy-aware fault signaling.

II. Challenges in 5G AI Monitoring

AI-driven 5G monitoring faces persistent bottlenecks that motivate the **LHB-X** design:

- **Telemetry overload:** Continuous KPI/log streaming inflates NF→NWDAF bandwidth and storage [2].
- **Compute cost:** Raw telemetry forces heavy AI/ML pipelines, raising CPU/GPU load and memory usage [3], [4].
- **Energy inefficiency:** Increased data flow expands power draw and cooling demand [6], [7].
- **Decision latency:** Inference queues and complex control loops delay slice or mobility decisions [3].
- **Scalability:** Distributed 5GC deployments struggle to maintain responsiveness under high-rate KPI reporting [4].
- **Lack of a lightweight standard:** KPI/log methods lack a unified Boolean health signal and feedback path [2], [9].

To address these issues, **LHB-X** fuses SDT with NWDAF-driven adaptive gain, introducing: (i) a **Boolean Heartbeak (HB)** bit, (ii) an **Instability Index** I for temporal decay, and (iii) a **Heartbreak Score** S_{HB} for gain-amplified bursts [10], [11]. Together, they yield a constant-time ($O(1)$), low-energy protocol for sustainable 5G/6G AI monitoring [8].

III. Background

This section summarizes the foundations of the LHB-X framework, covering: (i) surprisal-driven Signal Detection Theory (SDT), (ii) Boolean Heartbeak signaling with adaptive gain, (iii) constant-time message design, and (iv) its relation to existing telemetry-reduction and AIOps methods. These foundations construct a lightweight, feedback-compatible health-bit layer suitable for noisy and distributed 5G/6G environments.

A. Signal Detection Theory (SDT)

Signal Detection Theory provides a statistically grounded framework for distinguishing signal from noise under uncertainty [10]. Its surprisal-based formulation allows subsystem observations to be converted into compact one-bit decisions rather than continuous KPI/log streams. Recent work also applies SDT to human oversight of AI systems [11], reinforcing its relevance for safety-critical decision pipelines.

As illustrated in Fig. 2, the SDT threshold separates noise and signal-plus-noise distributions, enabling explicit control over Type I/II errors without retaining history buffers.

Comparison with Other Detectors: Several lightweight detectors partially address anomaly detection but lack the stateless, Booleanized, feedback-compatible design required for closed-loop NWDAF operation:

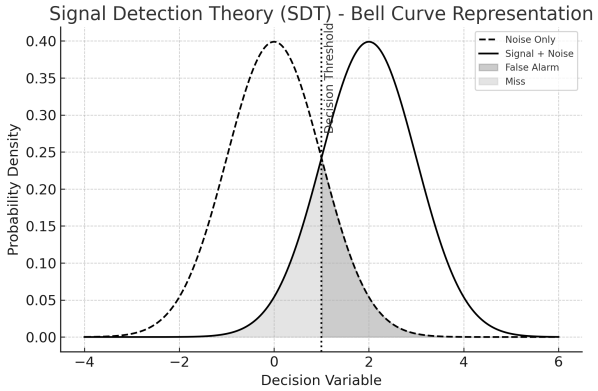

Fig. 2. Signal Detection Theory (SDT) showing noise and signal+noise distributions separated by a decision threshold.

- **Sketching Techniques** — CountMin, CountSketch, HLL, and Bloom filters [12], [13] compress frequency information but do not model anomaly severity, operate on raw counts, and typically require kilobytes of state.
- **Drift Detectors** — CUSUM, EWMA, Page–Hinkley, and ADWIN [14]–[16] detect gradual drift using sequential statistics but output continuous values requiring postprocessing.
- **Fixed-Threshold KPI Rules** [17] are simple but noise-sensitive and lack explicit Type I/II control adaptation.
- **ML-Based Detectors** — Autoencoders, LSTMs, GRUs, and GNNs [18]–[21] require raw telemetry, introduce inference latency, and increase compute/energy usage [6].
- **IoT/Edge Filters** [22], [23] identify local outliers but do not encode uncertainty or support feedback-aware adaptation.

SDT remains unique in providing stateless operation, explicit error control, a natural Boolean mapping, and consistent $O(1)$ per-event cost—making it an ideal foundation for LHB-X.

B. LHB-X Signaling and Constant-Time Design

LHB-X encodes subsystem well-being as a Boolean Heartbreak bit augmented with an adaptive gain factor $\kappa(n)$ returned by NWDAF along 3GPP-compliant feedback paths [2], [3]. A 3-bit Type field enables constant-time ($O(1)$) downstream interpretation, while deeper body fields are accessed only when HB = 1. This design minimizes CPU cycles, memory traffic, and telemetry volume, aligning with 6G sustainability goals emphasizing energy-aware observability [6], [8], [24].

C. Relation to Telemetry Reduction and AIOps

Conventional telemetry-reduction methods (sketches, sampling, learned summarizers) [25] and AIOps correlation pipelines [18] reduce analytic load but still rely on raw or semi-processed KPI streams. LHB-X differs by offering:

- **constant-time decoding** using a fixed 3-bit Type field,
- **statistically grounded Booleanization** via SDT with idle-aware normalization, and
- **adaptive gain feedback** replacing static threshold policies under NWDAF [2].

These properties position LHB-X as a minimal and sustainable telemetry backbone for 5G/6G NWDAF, AIOps, and emerging Network Digital Twin workflows [5].

TABLE I
FIELDS MAINTAINED IN THE SDT REGISTRY.

Field	Description
`sub_id`	Unique subsystem identifier.
`module`	Network Function (e.g., AMF, SMF, UPF).
`baseline`	Reference threshold for severity comparison.
`observed`	Current value reported by SDT.
`sensitivity`	Weight influencing health score.
`idle_count`	Consecutive inactive cycles.
`clock_interval`	Expected reporting cadence (e.g., 60 s).
`updated_at`	Timestamp of last update.

TABLE II
EXAMPLE SDT REGISTRY ENTRIES.

ID	Mod	Base	Obs	Sens	Idle	ClkInt	UpdAt
1	AMF	0.08	0.09	0.5	0	60 s	10:00
2	UPF	0.10	0.19	2.0	0	60 s	10:00

IV. PROPOSED WORK

Our objective is to detect service degradations within specific ms at FPR $\leq 10\%$, while minimizing telemetry bandwidth and energy per detected event under prototype-scale 5GC loads.

A. Step 1: SDT Registry Table (One-Time Setup)

A central *SDT registry* initializes subsystem identifiers, thresholds, and sensitivity weights at each monitoring window, ensuring deterministic Booleanization and lightweight LHB-X signaling. A `clock_interval` field defines expected reporting cadence for asynchronous ingestion and alignment across the network [2], [3].

The registry thus acts as the single source of truth for subsystem health, supporting cadence-aware, asynchronous SDT updates and consistent telemetry synchronization [3].

B. Step 2: Severity Ingestion via SDT

Each subsystem runs a **Signal Detection Theory (SDT)** module that computes a *suprisal-based severity score* at its configured `clock_interval` (e.g., AMF 60 s, SMF 120 s, UPF 60 s). Here, each observation is represented by its information content $I(x) = -\ln p(x)$, providing a compact measure of statistical deviation.

Assuming equal variance for tractability,

$$x \sim \mathcal{N}(\mu_0, \sigma^2) \text{ for } \mathcal{H}_0, \quad x \sim \mathcal{N}(\mu_1, \sigma^2) \text{ for } \mathcal{H}_1,$$

the decision threshold in the surprisal domain is

$$\tau_I = \ln\left(\frac{p(\mathcal{H}_0)C_{\text{FA}}}{p(\mathcal{H}_1)C_{\text{Miss}}}\right) + \frac{(\mu_1^2 - \mu_0^2)}{2\sigma^2}.$$

Severity is given by $s_i(n) = -\ln p(x_i(n))$ and Booleanization from (Eqn 5) flags statistically unlikely states. In practice, thresholds are estimated from empirical telemetry distributions, and the Gaussian model is used as a convenient approximation rather than a strict assumption on the data.

At each cycle, SDT: (i) computes $s_i(n)$ for subsystem i; (ii) updates `observed`/`updated_at`; (iii) finalizes a registry snapshot. The severity vector is

$$\mathbf{s}(n) = [s_1(n), s_2(n), \ldots, s_m(n)], \tag{1}$$

forwarded to Step 3 for Booleanization and LHB-X signaling.

C. Step 3: Booleanization via Normalized Risk

Given SDT-derived severity scores $\{s_i(n)\}_{i=1}^m$ and thresholds $\{\theta_i\}_{i=1}^m$ stored in the registry (Step IV-A), each subsystem's freshness is captured by the idle count $\text{IC}_i(n)$ (Eq. 9). Missing or delayed updates thus increase risk, and real-valued severities are converted into a binary **Boolean Error State** vector $\mathbf{e}(n)$ [10].

1) Per-subsystem normalized risk:

$$R_i(n) = \text{IC}_i(n)\frac{s_i(n)}{\theta_i} \tag{2}$$

where $s_i(n)$ is the observed severity, θ_i the reference threshold, and $\text{IC}_i(n)$ the missed-cycle count. The ratio $\frac{s_i}{\theta_i}$ normalizes deviation, while $\text{IC}_i(n)$ penalizes stale subsystems. Thus $R_i(n)$ grows larger both when a subsystem's condition worsens and when it fails to update on time.

2) Boolean Error State:

$$e_i(n) = \mathbb{1}\{R_i(n) > 1\} \tag{3}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function: $e_i(n)$=1 denotes a fault, and $e_i(n)$=0 denotes a healthy state. This binary transformation enables lightweight fault signaling.

3) Vectorized Form:

$$\mathbf{e}(n) = [\, e_1(n), e_2(n), \ldots, e_m(n) \,] \tag{4}$$

provides a compact binary snapshot of all subsystems at cadence n. This vector forms the basis for global indicators and matrix encoding in later steps.

4) Global Severity Score: The global severity score is derived from the following construction:

$$\delta_i(n) = \max\{\, R_i(n)\},$$

$$S(n) = \sum_{i=1}^{m} \delta_i(n), \tag{5}$$

$$S_{\text{flag}}(n) = \mathbb{1}\{\, S(n) > 0 \,\}.$$

In the prototype implementation, $S_{\text{flag}}(n)$ serves primarily as an aggregate trigger indicating that one or more subsystems have exceeded their normalized risk threshold.

D. Step 3b: Heartbreak Signal Generation

We propose different approaches for the Heartbreak signal $\mathbf{HB}(n)$ under different subsystem regimes.

1) Async Structurally Concrete Systems: For structurally concrete systems (fixed, weakly interacting subsystems), the Heartbreak signal is computed via Boolean OR logic:

$$HB(n) = \bigvee_{i=1}^{m} e_i(n), \tag{6}$$

where $\bigvee$ denotes logical OR across subsystems. Evaluation short-circuits once any $e_i(n)$=1, yielding best-case $O(1)$ and worst-case $O(m)$ complexity. A single subsystem breach triggers the global flag and generates a **Heartbreak signal, HB**(n) for downstream processing.

2) Async Structurally Complex Systems: In structurally complex settings with asynchronous interactions inside the 5G core layer, we optionally enrich the global view using a *complement–mirrored* state matrix $SM(n)$. Using the Boolean Error State vector $\mathbf{e}(n) = [e_1, e_2, e_3]$ and Global Health

Indicator $G(n)$ (Eq. 6), $SM(n)$ embeds subsystem faults in the upper triangle and their complements below the diagonal, with $G(n)$ on the main diagonal.

Complement–Mirrored Matrix:

For a 3-subsystem case:

$$a_{12} = e_1(n), \quad a_{13} = e_2(n), \quad a_{23} = e_3(n),$$

and the lower triangle (monitors) is set to the complements. The diagonal is $G(n)$. Thus,

$$SM(n) = \begin{bmatrix} G(n) & e_1(n) & e_2(n) \\ 1 - e_1(n) & G(n) & e_3(n) \\ 1 - e_2(n) & 1 - e_3(n) & G(n) \end{bmatrix}. \tag{7}$$

Heartbreak Signal (HB):

The operational *heartbreak signal* can be derived as a binary gate from the determinant of $SM(n)$:

$$HB(n) = \begin{cases} 1, & \det(SM(n)) > 0, \\ 0, & \det(SM(n)) \leq 0. \end{cases} \tag{8}$$

$G(n)$ captures binary fault presence, while $\det(SM)$ encodes an interaction-sensitive aggregate. The lower triangle stores complements $(1-e_i)$, so in the all-healthy pattern the constructed $SM(n)$ is singular ($\det=0$), whereas faulted patterns in our prototype exhibit non-zero determinants.

Example:: If e_i=0, G=0:

$$SM = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}, \ \det(SM)=0 \Rightarrow HB=0.$$

If $\mathbf{e}=[1,0,1]$, G=1:

$$SM = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}, \ \det(SM)=2 \Rightarrow HB=1.$$

This construction generalizes to square $SM(n)$ for $m > 3$ with an $O(m^3)$ cost per evaluation. In practice, the determinant-based indicator is treated as an optional heuristic for structurally complex subsystems; the OR-based form in Eq. (6) is sufficient for most deployments and is used as the primary HB construction in our experiments.

3) Sync Structurally Concrete Systems: In synchronous and structurally concrete systems, we can compute the **Heartbreak Signal** directly from the first error bit during the cadence window:

$$\text{GHI} = HB_s = \bigvee_{i=1}^{m} e_i(n),$$

which short-circuits on the first fault during a window. Complexity—Best: $O(1)$; Expected: $O(1/p)$; Worst: $O(m)$.

4) Unified Heartbreak Signal HB(n): The global health signal adapts $HB(n)$ across three operating regimes. **HB(n)=**

$$\begin{cases} 1, & \text{(synchronous)} \\ \bigvee_{i=1}^{m} e_i(n), & \text{(async., structurally concrete)} \\ 1\{\det(SM(n)) \neq 0\} & \text{(async., structurally complex)} \end{cases}$$

The first case represents synchronous OR-based short-circuit triggering, the second an asynchronous aggregation over independently reporting subsystems, and the third an optional structural coupling indicator derived from the complement–mirrored matrix $SM(n)$.

E. Step 4: Idle-Count Refresh

At each cadence $T_{\text{cad}}(n)$, missed reports are updated as

$$\text{IC}_i(n) = \max\left(0, \left\lfloor \frac{T_{\text{cad}}(n) - T_{\text{upd},i}(n)}{\Delta_i} \right\rfloor\right), \tag{9}$$

where Δ_i is the subsystem's reporting interval. For Δ_i=60 s: $T_{\text{upd},i}$=600 s $\Rightarrow$ IC_i=0 (fresh); $T_{\text{upd},i}$=420 s $\Rightarrow$ IC_i=3 (three missed cycles). The registry's `idle_count` column thus quantifies missed cadences and multiplies Eq. (2), linking freshness directly to fault sensitivity.

F. Step 5: Instability Index and HB Score with Adaptive Gain Feedback

This step models monitoring dynamics through two complementary functions that emulate **AI-like adaptive behavior** and incorporate feedback from NWDAF systems:

- **Instability Index** – The *Instability Index* $I(n)$ represents a bounded decay term that suppresses spurious bursts as idle time grows, acting as a regularization-like factor over delayed or noisy updates:

$$I(n) = \frac{HB(n)}{R(n)e^{IC_{\text{tot}}(n)} + e^0} \tag{10}$$

- **HB Score** – The *HB Score* $S_{HB}(n)$ is an exponential burst term amplified by the adaptive gain factor $\kappa(n)$ provided by NWDAF:

$$S_{HB}(n) = \left(R(n)e^{IC_{\text{tot}}(n)}\right)^{\kappa(n)} \tag{11}$$

Here, $IC_{\text{tot}}(n)$ is the total idle count, $R(n)$ denotes the normalized risk, and $\kappa(n)$ is the adaptive gain factor supplied by the NWDAF. For $HB(n) \in \{0,1\}$ and $R(n), IC_{\text{tot}}(n) \geq 0$, the Instability Index $I(n)$ remains in $[0,1]$, providing a bounded suppression term. Lower $R(n)$ values yield slower exponential growth in $S_{HB}(n)$, while higher $R(n)$ and $\kappa(n)$ accelerate the burst response, providing a tunable balance between suppression and excitation across subsystems. In this work, $S_{HB}(n)$ is used as a qualitative indicator of burst severity rather than an exact physical energy model.

1) Adaptive Gain Feedback Calculation: The gain factor $\kappa(n)$ evolves dynamically according to a simple error-driven update:

$$\kappa_{n+1} = \text{clip}\big(\kappa_n + \eta\big(E_{\text{tgt}} - E_n\big) + \beta\big(E_n - E_{n-1}\big),\ \kappa_{\min},\ \kappa_{\max}\big), \tag{12}$$

where E_n denotes a scalar error signal (e.g., deviation from a target false-positive rate or per-event energy budget), E_{tgt} is its reference value, η governs the proportional correction, and β provides a damping term based on recent error trends. The $\text{clip}(\cdot)$ operator constrains $\kappa(n)$ within $[\kappa_{\min}, \kappa_{\max}]$, ensuring that the gain remains bounded.

This update rule is intentionally lightweight and inspired by proportional-plus-damping control rather than designed as an optimal controller. Under a simple linearized approximation, one sufficient condition for a bounded response is to select $0 < \eta < 2/L$ and $0 \leq \beta < 1$, where $L = |\partial E/\partial \kappa|$ denotes a local sensitivity term. In our prototype experiments, $\eta \in [0.01, 0.1]$ and $\beta \in [0, 0.3]$ produced smooth, monotone adaptation across varying 5GC load conditions. A full control-theoretic analysis of convergence under richer fault dynamics is left as future work.

TABLE III
LHB-X HEADER FIELDS

Field	Var	Bits	Description
Type	t	3	Message type (HB(n), ACK, NACK, reserved)
Length	L	12	Total message length (bytes)
Flags	F	4	F1=HB, F2=burst, F3=decay, F4=test/downstream
Timestamp	$T(n)$	32	System cycle index or epoch
Checksum	C	16	CRC-16 integrity check
Reserved	(1–2)	–	Future extension fields
EOH	–	1	End-of-header delimiter

TABLE IV
LHB-X BODY FIELDS

Field	Var	Description
HB Score	$S_{HB}(n)$	Burst indicator under adaptive gain feedback
Instability Index	$I(n)$	Stability/decay metric (Step 5)
Active Severity Set	S_n^+	(i, s_i) pairs; reduced to $\widehat{S_n^+}$ for bandwidth
Adaptive Gain	$\kappa(n)$	Current gain; next value returned by NWDAF
Reserved	(1–4)	QoS, slice ID, metadata, or security extensions
EOM	–	End-of-message delimiter

G. Step 6: LHB-X Message Construction

After deriving subsystem severities, Boolean Heartbreak indicators, the Instability Index, and HB Score (burst factor), these components are assembled into a compact **LHB-X message**. This message supports: (i) constant-time ($O(1)$) downstream interpretation through a fixed-size header and 3-bit *Type* field, (ii) subsystem-aware signaling through the *active severity set* S_n^+, and (iii) extensibility for future control or diagnostic messages [3], [9]. Only the analytics layer benefits from $O(1)$ decoding; edge-side processing scales independently with subsystem structure.

*1) **Message Structure**:* The LHB-X message comprises a fixed-length *Header* and a variable-length *Body*. The header (Table III) enables immediate decoding and constant-time decisioning, while the body (Table IV) carries optional contextual fields for deeper analytics and ACK/NACK responses (Fig. 3).

Type codes (3-bit map):

- `000`: HB=0 (no heartbreak)
- `001`: HB=1 (heartbreak detected)
- `010–101`: Reserved
- `110`: NACK{id} (subsystem still faulty)
- `111`: ACK{id} (fault acknowledged)

A sample HB(1) message (t = 001) is encoded as:

```
header = {v:"01", t:1, L:128, T:1692782334, F
    :"1010",
       C:"0x3F2A", reserved:[null,null], EOH
           :1}
body = {S_HB:0.72, I:1.83, mask:26,
       S+:[{id:4,val:3.5},{id:2,val:2.1}],
       kappa:1.15, EOM:1}
```

This structure ensures **constant-time ($O(1)$) interpretation at the NWDAF layer**: a node inspects only the `Type` (3 bits) and $\kappa(n)$ field for fast decisions. Parsing of S_n^+ and additional fields is deferred to slower analytics paths [9].

*2) **Subsystem Identification via MAC Address**:* To maintain global uniqueness, subsystems may be identified through their *MAC address*, stored in the registry (Table I). The fault mask $M(n)$ encodes active bits, while S_n^+ carries (MAC, s_i) pairs. NWDAF resolves MAC → `nfInstanceId`/`nfType`/`URI` via NRF, returning ACK/NACK along the same mapping. This avoids ambiguity in multi-NF, multi-module deployments.

*3) **NWDAF Feedback Policy**:* NWDAF responses are subsystem-aware:

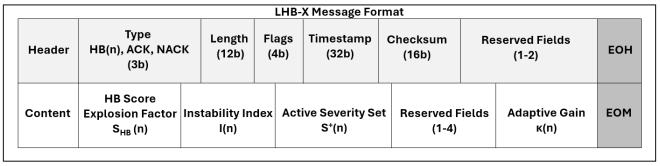

Fig. 3. LHB-X message format showing header, body, and field mapping

- **ACK** (`t=111`): returns updated adaptive gain $\kappa'(n)$ for the next monitoring window.
- **NACK** (`t=110`): instability persists; retain current $\kappa(n)$.

Thus, ACK/NACK operates not merely as acknowledgment but as a lightweight closed-loop feedback channel for continuous gain adaptation.

H. Step 7: LHB-X Encode–Route–Ack Algorithm

Each monitoring cadence constructs per-subsystem severities $s_i(n)$, thresholds θ_i, and contextual scalars $\{S_{HB}(n), I(n), \kappa(n)\}$. The 3-bit *Type* map encodes control signals as: `000`=HB= 0, `001`=HB= 1, `010–101`=reserved, `110`=NACK{id}, and `111`=ACK{id}. The active severity set is $S_n^+ = \{(i, s_i) \mid s_i > \theta_i\}$, with $\widehat{S_n^+}$ denoting the bandwidth-reduced subset [3].

The end-to-end encode–route–ack process ensures **constant-time ($O(1)$) decoding at the NWDAF layer** via fixed header slots for the Type and adaptive gain fields. Optional parsing of S_{HB}, $I(n)$, $M(n)$, and $\widehat{S_n^+}$ occurs only when HB(n) = 1 and is deferred to slower analytics layers [9].

Complexity. Encoding and routing of the fixed-length header execute in $O(1)$ time. Edge-side computation of Boolean vectors or correlation matrices scales with subsystem structure ($O(m)$–$O(m^3)$), but downstream NF/NWDAF decisioning remains constant-time since it inspects only the Type map and $\kappa(n)$.

Interpretation. This design allows control-plane nodes to react immediately based on the Type code and $\kappa(n)$ value, while analytics modules asynchronously extract HB Score, Instability Index, and active severity sets for trend learning and feedback estimation.

I. Step 8: Protocol and Message Transport

LHB-X employs a lightweight, event-driven message model aligned with 3GPP NWDAF interfaces. Each message captures an NF's instantaneous health using a Boolean flag and minimal metadata. Messages (< 120 B) may be serialized as JSON or compact binary formats and sent via REST/gRPC.

1) Transport: Agents communicate over **gRPC** through the Docker bridge. Aggregated data reach NWDAF via:

- **Nnwdaf_EventsSubscription:** NF-level KPI feeds;
- **Nnwdaf_AnalyticsInfo:** Burst metrics and adaptive gain feedback.

This separation keeps health signaling lightweight while enabling deeper, asynchronous analytics when required.

V. Reduced Complexity for Structurally Concrete Subsystems

When subsystem interactions are synchronous or weakly coupled, the LHB-X pipeline can be simplified for downstream AI telemetry without losing fidelity.

*1) **Header-only optimization**:* If $G(n) = 0$, the node emits a **header-only** LHB-X message (t=`000`, HB= 0), achieving pure $O(1)$ processing. If $G(n) = 1$, it emits t=`001` with a minimal body (mask $M(n)$, Top-k set $\widehat{S_n^+}$) and skips determinant-based checks. This reduces edge-side cost from $O(m^2)$ to $O(m)$ and, during healthy periods, collapses to constant-time signaling.

*2) **NWDAF feedback**:* This reduced mode remains fully compatible with closed-loop gain control:

- **ACK** (`t=111`): returns an updated gain $\kappa'(n)$.
- **NACK** (`t=110`): retains the current $\kappa(n)$.

Thus ACK/NACK continues to provide a lightweight feedback path for adaptive convergence even under reduced-complexity operation.

VI. Experimental Evaluation

A. Testbed Setup

Experiments were performed on a Docker–based 5G core (**Open5GS v2.7.6**) comprising NRF, AMF, SMF, UPF, and MongoDB containers on a shared virtual bridge. Python-based **LHB-X agents** and the **SDT supervisor** executed as sidecars to each NF, while an external **AI-driven NWDAF** (port 9400) provided adaptive gain feedback via gRPC.

Telemetry was streamed through gRPC/REST and collected using Prometheus, cAdvisor, and node-exporter. Containers ran on a laptop with:

- 8-core Intel i7 CPU (3.8 GHz), 16 GB RAM
- Windows 11 host OS
- Docker Desktop + WSL2 (Ubuntu 22.04) runtime

Energy traces were sampled using **Intel RAPL** exposed through WSL2's MSR interface. Container-level CPU residency from `docker stats` was used for cross-checking, showing $< 5\%$ variance across repeated measurements.

Faults (CPU overload, network delay, jitter, and loss) were injected using `stress-ng` and `tc netem`. LHB-X processed Boolean health bits ($HB(n) \in \{0, 1\}$) every 100 ms and updated $\kappa(n)$ in real time.

Note on two-node Docker deployment. A second Open5GS container was instantiated to emulate AMF/SMF and UPF split nodes. However, since all containers share the same host CPU, memory, and virtual switch under WSL2, cross-node effects (NIC queue separation, host-to-host RTT, or NUMA behavior) do not manifest. Thus, results reflect *method-level behavior* rather than infrastructure scaling. A true multi-host setup is under development to evaluate inter-node LHB-X propagation and delay variance.

B. Baselines and Metrics

Five monitoring pipelines were evaluated:

- **B1 – Raw Telemetry**
- **B2 – Fixed Threshold**
- **B3 – Sketch Sampling**
- **B4 – NWDAF KPI Polling**
- **P – LHB-X (SDT + Adaptive Gain)**

All baselines were tuned for TPR ≈ 0.9. Metrics included bandwidth (MB/s), latency (ms), CPU (%), TPR/FPR, F1-score, and energy (J/event). ROC and PR curves were computed using `scikit-learn`. Thresholds were derived empirically from telemetry distributions without assuming perfect Gaussianity.

TABLE V
COMPARATIVE PERFORMANCE OF MONITORING APPROACHES

Method	BW (MB/s)	Lat. (ms)	CPU (%)	TPR	FPR
B1 – Raw Telemetry	14.2	225	47	0.92	0.26
B2 – Fixed Threshold	9.8	182	41	0.88	0.21
B3 – Sketch Sampling	5.6	160	36	0.90	0.14
B4 – NWDAF KPI	9.1	175	39	0.84	0.19
P – LHB+SDT	**1.9**	**68**	**23**	**0.91**	**0.07**

C. Quantitative Results

LHB-X reduced telemetry bandwidth by **6.5×** and end-to-end latency by **2.6×** compared to NWDAF KPI polling. FPR dropped from 0.19 (B4) to 0.07, while maintaining TPR 0.9. Area-under-curve improved substantially (AUC: **0.93** vs. **0.78**).

D. Ablation and Complexity

Disabling adaptive gain increased FPR ($0.07 \rightarrow 0.15$) and latency ($68 \rightarrow 112$ ms), highlighting the stabilizing role of $\kappa(n)$. Runtime cost remained low (CPU $< 5\%$, memory < 25 MB). Message sizes stayed < 100 B with control-plane traffic $< 0.3\%$.

Downstream interpretation retained constant-time $O(1)$ since only the fixed header (Type and κ) is inspected. Edge-side Booleanization and optional structural correlation scale as $O(m)$–$O(m^3)$, depending on subsystem complexity.

E. Reliability and Integration

Sequence counters and gRPC ACKs ensured at-least-once delivery with $< 0.3\%$ overhead. In alignment with 3GPP TS 23.288, LHB-X treats agents as producers, SDT modules as local filters, and NWDAF as the feedback authority.

When multiple feedback records arrive simultaneously:

$$(\kappa^\star, p^\star, T^\star) = \arg\max_i (p_i, T_i), \qquad \kappa_{n+1} \leftarrow \kappa^\star.$$

LHB-X reports {HB, S_{HB}, I, S_n^+, mask} over `Nnwdaf_AnalyticsInfo`. Optional MQTT mirroring publishes JSON at `/lhb/nf_type/instance/metric`, enabling lightweight edge consumption.

F. Energy and Sustainability

Energy consumption was measured using **Intel RAPL** (CPU package + DRAM) inside WSL2. Container-level CPU residency from `docker stats` was used to verify measurement consistency ($< 5\%$ deviation).

Average power draw dropped from 7.2 W (B1) to 3.1 W (P), producing a **54% energy reduction**:

$$E_{\text{saved}} = k(B_{\text{raw}} - B_{\text{LHB}}).$$

Savings stem from reduced telemetry production and lower CPU cycles. LHB-X therefore achieves significant improvements in bandwidth, latency, and energy without compromising detection fidelity. Integration via `Nnwdaf_EventsSubscription/AnalyticsInfo` preserves full 3GPP compatibility. Future work includes multi-host evaluation and embedding LHB-X as a native Open5GS microservice.

VII. Future Work

Several open directions emerged during LHB-X development:

1) **Binary sensitivity:** A single-bit $HB(n)$ may miss mild or sub-threshold degradations. We plan to explore multi-level or probabilistic variants while preserving the lightweight design.
2) **Threshold tuning:** Subsystem thresholds θ_i influence the sparsity of S_n^+ and burst stability. Data-driven, non-Gaussian threshold adaptation remains a key area for refinement.
3) **Adaptive gain evolution:** The current gain update is a simple error-driven rule. Richer closed-loop policies and AI-assisted tuning (e.g., reinforcement learning) will be investigated.
4) **Evaluation scale:** A multi-host Open5GS setup with Ettus B200/B210 SDR-based RAN integration is under construction to measure LHB-X under realistic, distributed 5G/6G conditions.
5) **Complex subsystem growth:** The complement–mirrored matrix $SM(n)$ incurs $O(m^2)$ storage and $O(m^3)$ determinant cost. Sparse, hierarchical, or incremental encodings are being studied to improve scalability.
6) **Integration with Digital Twins:** LHB-X fits naturally as a minimal telemetry backbone for 5G/6G Network Digital Twins (NDT). Embedding LHB-X into NDT-based predictive assurance is ongoing work.

VIII. Other Application Domains

Beyond the 5G NWDAF layer, **LHB-X** generalizes to several domains requiring ultra-lightweight health signaling or rapid stability checks, including:

- **IoT and embedded systems** — $O(1)$ bit-level signaling for resource-constrained devices.
- **Cloud/edge environments** — federated fault isolation with minimal telemetry.
- **Industrial automation and robotics** — fast subsystem alerts under strict timing constraints.
- **UAV/robot coordination** — lightweight safety flags for multi-agent swarms.
- **Wearables and healthcare devices** — low-energy anomaly indicators for continuous monitoring.

Across these settings, LHB-X functions as a class of ultra-lightweight, constant-time fault-signaling protocols with adaptive sensitivity under feedback control.

IX. Conclusion

We introduced **LHB-X**, a lightweight Boolean heartbreak framework that combines Signal Detection Theory, surprisal-based scoring, and adaptive gain feedback for AI-driven NWDAF monitoring. Its compact header–body format and 3-bit Type field enable constant-time ($O(1)$) interpretation while sharply reducing telemetry, CPU load, and energy usage.

SDT-based Booleanization and gain-amplified metrics capture the required balance between stability and sensitivity, with experiments showing substantial improvements in bandwidth, latency, and energy efficiency without loss of diagnostic fidelity.

Although further refinement in threshold tuning, gain adaptation, and multi-host evaluation remains, LHB-X provides a principled basis for scalable, sustainable, and AI-integrated monitoring. Its lightweight design generalizes naturally to 6G, IoT, cloud, and cyber–physical systems, and serves as a practical building block for predictive, self-optimizing, and energy-conscious network observability.

References

[1] 3GPP, "System Architecture for the 5G System (5GS)," TS 23.501 V18.5.0, May 2024.
[2] 3GPP, "Architecture Enhancements for 5G System (5GS) to Support Network Data Analytics Services," TS 23.288 V18.5.0, May 2024.
[3] 3GPP, "5G System; Network Data Analytics Services; Stage 3," TS 29.518 V18.5.0, 2024.
[4] 3GPP, "5G System; Event Exposure Service for NWDAF and Other Analytics Functions," TS 29.552 V18.5.0, Jul. 2024.
[5] 3GPP, "Study on Management Aspect of Network Digital Twin," TR 28.915, Release 19, 2023–.
[6] International Energy Agency (IEA), *Energy and AI*. Paris, France: IEA, Apr. 2025.
[7] A. K. N. Lazarev *et al.*, "Towards Energy Efficient 5G vRAN Servers," Microsoft Research, Jun. 2024.
[8] 6G-IA, "Sustainability of 6G: Ways to Reduce Energy Consumption," Whitepaper v1.2.2, Jan. 2025.
[9] M. Yang *et al.*, "Towards NWDAF-enabled Analytics and Closed-Loop Automation in 5G Core," *arXiv:2505.06789*, May 2025.
[10] N. A. Macmillan and C. D. Creelman, *Detection Theory: A User's Guide*, 2nd ed. Mahwah, NJ, USA: Lawrence Erlbaum, 2005.
[11] M. Langer, S. van der Linden, S. Wachter, and B. Mittelstadt, "Human Oversight of AI: A Signal Detection Theory Approach," *Minds and Machines*, vol. 34, no. 4, pp. 615–644, 2024.
[12] G. Cormode and K. Yi, "Sketch techniques for approximate query processing," *Foundations and Trends in Databases*, vol. 4, no. 1–3, pp. 1–294, 2012.
[13] Y. Li and Y. Chen, "A survey on sketching techniques for network measurement and telemetry," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 157–195, 2022.
[14] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*. Prentice-Hall, 1993.
[15] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.
[16] A. Bifet, J. Montiel, J. Read *et al.*, "Machine learning for data streams with concept drift: A survey," *ACM Computing Surveys*, vol. 53, no. 4, pp. 1–36, 2021.
[17] A. Giaquinta, P. Bellavista, A. Corradi, and L. Foschini, "NWDAF in 5G: Architectures, use cases, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2751–2790, 2023.
[18] H. Zhang and M. Chen, "AIOps for networks: architectures, algorithms, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 2, pp. 899–945, 2024.
[19] J. Gan, Y. Liu, and X. Li, "Graph neural networks for network fault diagnosis: A survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 3, pp. 2104–2138, 2023.
[20] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short-term memory networks for anomaly detection in time series," in *Proc. ESANN*, 2015.
[21] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–36, 2019.
[22] J. Stankovic, "Research directions for the Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.
[23] A. Alrawais, F. Alharthi, and A. Alhothaily, "Lightweight security and anomaly detection in IoT systems: A survey," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3611–3632, 2023.
[24] M. Polese, A. Kosek-Szott, O. Simeone *et al.*, "OAM and observability in 6G networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1123–1157, 2023.
[25] C. Miao, J. Zhang, and X. Liu, "Telemetry reduction for cloud-native 5G: Sketches, sampling, and control-plane tradeoffs," *IEEE Transactions on Network and Service Management*, 2023.