International Journal of Wireless and Ad Hoc Communication

♦ **ISSN:** 2692-4056 **Vol:** 10, No. 01, 2026 ♦ PP. 35–42
Article DOI: https://doi.org/10.54216/IJWAC.100106

FULL LENGTH ARTICLE

Communication-Aware Digital-Twin Reliability Budgeting for Fog-Assisted Wireless Sensor Ad Hoc Networks

Safina Shokeen[1,*] **Vishal Srivastava**[2]
[1] School of Engineering and Technology, VIPS-TC, India
[2] Department of Industrial Internet of Things, School of Engineering and Technology, Vivekananda Institute of Professional Studies, Technical Campus, Delhi 110034, India
Emails: safina.shokeen@vips.edu · vishal.srivastava@vips.edu

Received: February 09, 2026 Revised: March 19, 2026 Accepted: April 18, 2026 ⋆ Corresponding author

ABSTRACT

Wireless sensor IoT systems are increasingly deployed as infrastructure-light communication fabrics in which battery-powered devices exchange event streams through local gateways, fog nodes, and sometimes multi-hop ad hoc routes. In such settings, reliability cannot be judged only by how fast a packet reaches a server. A reading may be fresh but untrusted, energy-efficient but delayed, or successfully delivered through a route that overloads the next fog node. This article revises the problem as a communication-aware reliability budgeting task for fog-assisted wireless sensor ad hoc networks. It reviews core studies on wireless sensor networking, fog and edge computing, digital twins, edge intelligence, federated learning, and IoT security, then introduces an extended Digital-Twin Reliability Budgeting model. The model maintains compact fog-side twin states and uses them to govern route choice, event compression, fog offloading, replication, and cloud escalation. Three mathematical algorithms are presented for twin synchronization, route-and-action selection, and adaptive budget learning. The analysis develops delay, energy, freshness, loss, trust, and occupancy terms and shows how they interact across multi-hop communication paths. The resulting framework supports a more disciplined design philosophy: fog nodes should not only process sensor data near the edge; they should regulate the reliability budget of each communication decision before network resources are consumed.

Keywords: Wireless sensor IoT ▪ Ad hoc communication ▪ Fog computing ▪ Digital twin ▪ Reliability budgeting

1. INTRODUCTION

Wireless sensor IoT networks have moved from simple environmental monitoring toward applications that observe infrastructure, healthcare spaces, energy systems, vehicles, and industrial assets. The original promise of wireless sensor networks was low-cost distributed observation, but contemporary deployments are asked to deliver a much richer service: timely information, stable operation, secure forwarding, and continuous adaptation under battery, bandwidth, and link-quality constraints. Earlier surveys of wireless sensor networks identified energy and connectivity as central limitations, while IoT architectures added cloud connectivity and service composition to that problem space [1, 2, 3].

Fog computing emerged to narrow the distance between sensors and cloud services. Rather than forwarding every raw packet to distant data centers, fog nodes can preprocess data, maintain local context, and support delay-sensitive decision making. Foundational work on edge and fog computing describes this continuum as a practical response to latency, mobility, and locality requirements [4, 5]. Broader fog-IoT surveys further show that fog nodes can provide computation, storage, and networking support for services that cannot wait for remote cloud processing [6, 7].

However, fog computing alone does not solve the reliability problem of wireless sensor IoT. A sensor reading may arrive quickly but consume excessive energy; a packet may be compressed to save bandwidth but lose useful detail; a task may be placed at a fog node but become stale while waiting in a queue. This creates a multi-dimensional design problem. Recent edge intelligence and federated learning studies demonstrate that models can be trained or executed closer to the data, but they also show that heterogeneity, communication cost, privacy, and system drift remain unresolved challenges [13, 14, 15].

Digital twins provide a useful but still underused route for addressing this gap. In IoT, a digital twin can maintain a logical representation of a physical device or process and use that representation to support monitoring, prediction, and control [17, 18]. Network digital-twin studies extend the idea to communication infrastructure, where virtual replicas assist planning, service adaptation, and policy testing [19]. Yet fog-assisted wireless sensor IoT still lacks a compact model that combines twin state, reliability budgets, and local control decisions.

This article offers a full-length review and model-oriented analysis of digital-twin reliability budgeting for fog-assisted wireless sensor ad hoc networks. It differs from a conventional offloading review by treating communication itself as a governed process: a route, a fog placement, and a data-reduction action are selected only after delay, loss, freshness, residual energy, trust, and fog occupancy have been considered together. This framing is consistent with the resource-management literature in fog computing, but it adds a communication-aware control layer for wireless sensor paths in which every hop can change the cost of the event [10, 11].

The paper makes four contributions. First, it reorganizes prior work around communication reliability budgets rather than around individual technologies. Second, it introduces an extended Digital-Twin Reliability Budgeting (DTRB) model for deciding among local filtering, route-preserving fog offloading, event compression, replication, and cloud escalation. Third, it rewrites the DTRB process into three professional algorithms that can be implemented at a fog gateway or cluster head. Fourth, it adds mathematical analysis for multi-hop reliability propagation, cost normalization, budget feasibility, and adaptive weight learning.

The remainder of the paper is organized as follows. Section 2 reviews fog, edge intelligence, digital twins, and security-oriented sensing. Section 3 presents the review synthesis and a comparative mapping of prior studies. Section 4 introduces the proposed communication-aware DTRB model. Section 5 gives the mathematical analysis and extended scenario interpretation. Section 6 discusses design implications and open challenges. Section 7 concludes the paper.

2. BACKGROUND AND RELATED WORK

2.1 Fog and edge support for sensing systems

Fog computing is often positioned as an intermediate layer between wireless sensors and cloud platforms. Its core value is not merely faster processing, but the ability to bring decision logic close to the data source. Surveys by Yousefpour et al. [5], Puliafito et al. [6], and Bellavista et al. [7] established the taxonomy of fog services, emphasizing locality, mobility, and distributed resource management. Porambage et al. [8] connected this discussion to multi-access edge computing and IoT realization, while Abbas et al. [9] surveyed mobile edge computing as a closely related paradigm.

The resource-management problem has received detailed attention. Deng et al. [10] studied workload allocation between fog and cloud, showing that computation placement can be expressed as a trade-off between latency and energy. Ni et al. [11] reviewed resource allocation strategies and highlighted that fog nodes operate under limited capacity and variable demand. These findings are important for wireless sensor IoT because sensor traffic is not uniform: alarms, bursts, and environmental changes create temporary demand spikes.

2.2 Intelligence near the wireless edge

Edge intelligence studies shift the discussion from where computation is placed to how learning and decision making are supported. Li et al. [12] argued that edge computing can enable deep learning for IoT by reducing dependence on remote cloud servers. Zhou et al. [13] described edge intelligence as a pathway for bringing artificial intelligence to the network edge, while Wang et al. [14] reviewed the convergence between edge computing and deep learning. Iftikhar et al. [16] later systematized AI-based resource management in fog and edge environments, emphasizing the need for adaptive methods under dynamic workloads.

Federated learning is relevant because it allows devices or edge nodes to collaborate without sending raw data. Lim et al. [15] showed that federated learning at mobile edge networks must address communication efficiency, privacy, and heterogeneous participation. Mothukuri et al. [21] further examined privacy and security limitations of federated learning. In wireless sensor IoT, such findings imply that local knowledge and distributed model updates must be governed carefully rather than treated as cost-free intelligence.

2.3 Digital twins and security-aware sensor governance

Digital twins extend edge intelligence by keeping a live software representation of physical or network entities. Minerva et al. [17] placed digital twins within the IoT context, and Fuller et al. [18] summarized enabling technologies and open research challenges. Wu et al. [19] surveyed digital twin networks and emphasized their potential for network management and service adaptation. In industrial IoT, Xu et al. [20] showed that digital twins can support high-fidelity cyber-physical representations.

Security also shapes fog-assisted sensing. Ferrag et al. [22] introduced Edge-IIoTset as a realistic cybersecurity dataset for IoT and IIoT,. Healthcare-oriented fog systems such as HealthFog demonstrate that application-level decisions can be moved near the edge, but they also show that the resulting architecture must account for latency, reliability, and application criticality [24, 23]. Together, these studies justify a budget-based model that treats reliability as a joint property of sensing, communication, computation, and security.

Table 1. Representative literature mapped to reliability-budget dimensions

Study	Fog	AI	Twin	Energy	Fresh.	Security	Main relevance
Yousefpour et al. [5]	Yes	No	No	Yes	No	Yes	Fog taxonomy and continuum view
Puliafito et al. [6]	Yes	No	No	Yes	No	Yes	Fog support for IoT services
Bellavista et al. [7]	Yes	No	No	Yes	No	Yes	Integration requirements for fog-IoT
Deng et al. [10]	Yes	No	No	Yes	No	No	Latency-energy workload trade-off
Zhou et al. [13]	Yes	Yes	No	Yes	No	No	Edge intelligence architecture
Wang et al. [14]	Yes	Yes	No	Yes	No	No	Deep learning at edge
Lim et al. [15]	Yes	Yes	No	Yes	No	Yes	Federated learning at edge
Minerva et al. [17]	No	No	Yes	No	Yes	No	Digital twin architecture in IoT
Fuller et al. [18]	No	Yes	Yes	No	Yes	Yes	Enabling technologies for twins
Wu et al. [19]	Yes	Yes	Yes	No	Yes	Yes	Digital twin networks
Ferrag et al. [22]	Yes	Yes	No	No	No	Yes	IoT/IIoT security benchmark

Table 1 shows that the literature is rich but fragmented. Fog studies discuss placement and service proximity; edge-intelligence papers emphasize learning; digital-twin papers formalize virtual representations; and security datasets focus on attack visibility. Few works combine all of these dimensions into a single operational model for wireless sensor IoT.

3. REVIEW SYNTHESIS

3.1 From resource placement to reliability budgets

A recurring limitation across the reviewed literature is that the optimization target is often too narrow. Placement studies optimize delay or energy; security studies optimize detection quality; and intelligence studies optimize learning or inference. Real sensor deployments, however, require a budget view. A fog decision is acceptable only when it preserves sufficient delay slack, energy reserve, information freshness, and trust at the same time. This is the motivation for the DTRB model introduced in Section 4.

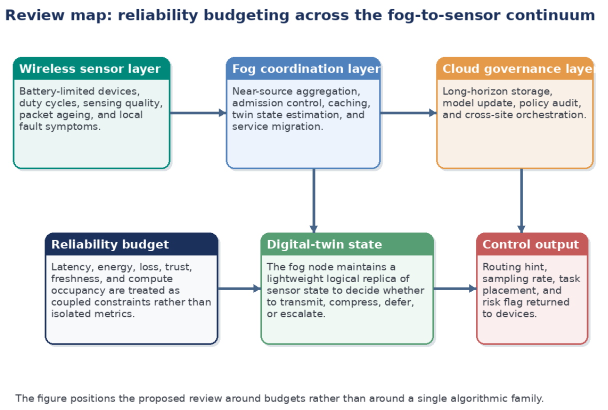


Figure 1. Review map linking fog support, digital-twin state, and reliability budgets.

Figure 1 summarizes the shift proposed in this paper. The wireless sensor layer supplies imperfect and energy-constrained observations. The fog layer provides local coordination and state estimation. The cloud layer handles longer-horizon governance. Reliability budgeting connects these layers by translating raw sensor status into actionable constraints.

3.2 Critical reading of prior contributions

The reviewed studies can be criticized along three axes. First, many assume that the network state is observable enough for accurate placement decisions, even though low-power sensors often report sparse and delayed status. Second, some intelligence methods require model updates or inference workloads that may exceed the capacity of fog nodes. Third, digital-twin studies rarely specify how the twin should influence individual sensor actions such as sampling, compression, or replication.

Table 2. Critical comparison of reviewed directions

Direction	Real-time	Battery-aware	Twin-based	Cross-layer	Main gap
Fog-IoT surveys [6, 7]	Yes	Partial	No	Partial	Broad taxonomies but limited operational decision rules
Workload allocation [10, 11]	Yes	Yes	No	Partial	Energy-delay trade-off without live logical state
Edge intelligence [12, 13]	Yes	Partial	No	Partial	Inference focus without sensor reliability budget
Federated edge learning [15, 21]	Partial	Partial	No	No	Privacy-aware learning but high communication sensitivity
Digital twin networks [17, 19]	Yes	No	Yes	Partial	Strong representation but weak sensor actuation policy
IoT security datasets [22]	Partial	No	No	No	Useful attack data but limited fog control logic
Application fog systems [23, 24]	Yes	Partial	No	Partial	Domain value but limited reusable governance model

Table 2 explains why a new model is needed. The issue is not absence of fog computing, artificial intelligence, or digital twins. The issue is the lack of a small operational bridge that turns sensor state into fog-level reliability decisions.

3.3 Review maturity across reliability attributes

The literature can also be read by asking how mature each reliability attribute is in practice. Delay and offloading have the strongest coverage because they can be modeled with measurable service time and network time. Energy is also widely discussed, but it is often handled as a device-side cost rather than as a shared constraint across sensor, fog, and cloud layers. Freshness and trust receive less unified treatment. This imbalance explains why a digital twin can be useful: it can keep a compact representation of the missing state and expose it to the placement logic.

Table 3. Maturity of reliability attributes in the reviewed literature

Attribute	Coverage	Common method	Remaining limitation
Delay	High	Placement and queue models	Limited treatment of bursty sensor events
Energy	High	Duty cycle and offloading cost	Rarely coupled with freshness and trust
Freshness	Medium	Age-of-information reasoning	Often separated from fog placement
Trust	Medium	Intrusion detection or reputation	Usually detached from resource decisions
Fog occupancy	Medium	Load balancing	Limited link to sensor-level risk
Explainability	Low	Post-hoc reporting	Weak support for operator audit

Table 3 indicates that future progress depends on joining attributes rather than adding another isolated optimization target. For example, an energy-aware design can still fail if it permits stale readings in an industrial safety loop. A security-aware design can also fail if it treats all suspicious events as high-priority traffic and overloads the fog node. The proposed DTRB model responds to this gap by using one compact budget vector.

4. COMMUNICATION-AWARE DIGITAL-TWIN RELIABILITY BUDGETING MODEL

The proposed communication-aware DTRB model treats every sensor-to-fog interaction as a budgeted communication decision. A sensor does not simply offload data when a fog node exists. Instead, the fog node keeps a lightweight digital twin of the sensor state and checks whether the next action satisfies reliability budgets. The model is intentionally compact because wireless sensor IoT devices cannot support heavyweight control loops.

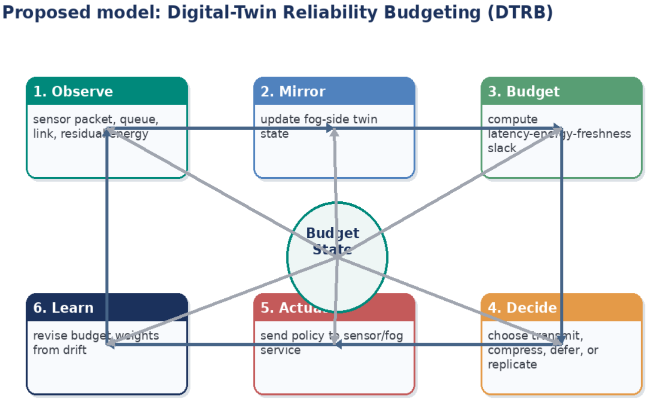


Figure 2. Operational cycle of the proposed communication-aware Digital-Twin Reliability Budgeting model.

As shown in Figure 2, DTRB contains six steps: observe, mirror, budget, decide, actuate, and learn. The observe step collects sensor packet status, queue length, link indicators, and residual energy. The mirror step updates the fog-side twin. The budget step calculates whether delay, energy, freshness, and trust constraints remain feasible. The decide step chooses the action. The actuate step sends the selected policy to the device or fog service. The learn step updates budget weights when traffic or link behavior drifts.

4.1 State variables and decision space

Let i denote a wireless sensor and t a decision epoch. The fog-maintained twin state is represented as

$$\mathbf{s}_{i,t} = \{e_{i,t}, q_{i,t}, \ell_{i,t}, a_{i,t}, r_{i,t}, u_t\}, \tag{1}$$

where e is residual energy, q is device queue level, ℓ is link loss estimate, a is age of information, r is trust or risk score, and u is fog occupancy. The action set is

$$\mathscr{A} = \{\text{local}, \text{fog}, \text{compress}, \text{replicate}, \text{cloud}\}. \tag{2}$$

The budget vector for sensor i is

$$\mathbf{b}_{i,t} = \{B_D - D_{i,t}, B_E - E_{i,t}, B_A - A_{i,t}, B_R - R_{i,t}\}, \tag{3}$$

where B_D, B_E, B_A, and B_R are maximum allowable delay cost, energy cost, information age, and risk respectively. A feasible action must satisfy every non-negative component of $\mathbf{b}_{i,t}$.

4.2 Communication-aware algorithmic design

The DTRB controller is implemented as three linked procedures. The first procedure keeps the fog-side digital twin consistent with the latest sensor and link evidence. The second procedure chooses the communication action and route. The third procedure updates the importance of each reliability budget after observing whether the selected action improved or weakened the system state.

Algorithm 1. Fog-Side Twin Synchronization and Budget Construction

Input: event descriptor $\mathbf{x}_{i,t}$, previous twin state $\mathbf{s}_{i,t-1}$, budget limits $\mathbf{B}_i$.
Output: updated twin state $\mathbf{s}_{i,t}$ and residual budget vector $\rho_{i,t}$.

```
1. Receive $\mathbf{x}_{i,t} = \{q_{i,t}, e_{i,t}, \ell_{i,t}, \gamma_{i,t}, \eta_{i,t}\}$ from sensor $i$, where $q$ is queue state, $e$ is residual energy, $\ell$ is loss estimate, $\gamma$ is link indicator, and $\eta$ is event priority.
2. Update freshness using $A_{i,t} \leftarrow 0$ if a valid update is delivered; otherwise set $A_{i,t} \leftarrow A_{i,t-1} + \Delta t$.
3. Update trust as $R_{i,t} \leftarrow \lambda R_{i,t-1} + (1-\lambda)\widehat{r}(\mathbf{x}_{i,t})$.
4. Form the compact twin state $\mathbf{s}_{i,t} = \{e_{i,t}, q_{i,t}, \ell_{i,t}, A_{i,t}, R_{i,t}, u_t\}$.
5. For each budget component $k \in \{D, E, A, R, U\}$, compute the residual margin $\rho_{k,t} = B_k - C_k(\mathbf{s}_{i,t})$.
6. Return $\rho_{i,t} = \{\rho_{D,t}, \rho_{E,t}, \rho_{A,t}, \rho_{R,t}, \rho_{U,t}\}$.
```

Algorithm 1 gives the fog node a compact picture of the sensor and its current communication environment. The objective is not to duplicate the entire physical process, but to maintain enough state to avoid blind offloading and blind forwarding. Queue pressure, link condition, event priority, trust, and freshness are converted into the same budget language before any decision is made.

Mathematically, the procedure transforms raw event observations into a bounded vector in which each component has the same sign convention: positive values indicate remaining reliability margin, while negative values indicate a violation. The update is linear in the number of budget components and therefore has complexity $O(K)$ per event. Because K is small and fixed in the proposed model, the synchronization step can operate at the fog gateway without measurable scheduling overhead.

Algorithm 2. Route-and-Action Selection under Ad Hoc Communication Budgets

Input: residual budget $\rho_{i,t}$, candidate actions $\mathscr{A}$, candidate routes $\mathscr{P}_{i,t}$.
Output: selected pair (a^*, p^*).

```
1. For every route $p \in \mathscr{P}_{i,t}$, estimate route reliability $\Gamma(p) = \prod_{h \in p}(1-\ell_h)$ and route delay $D_p = \sum_{h \in p}(d_h + w_h)$.
2. For every action $a \in \mathscr{A}$ and route $p \in \mathscr{P}_{i,t}$, estimate normalized cost vector $\widehat{\mathbf{C}}(a,p)$.
3. Reject $(a,p)$ if any hard constraint $C_k(a,p) > B_k$ is violated.
```

```
4. Compute the admissible utility
   $\Phi(a,p) = \eta_{i,t}\Gamma(p) - \sum_k \omega_k \widehat{C}_k(a,p).$
5. Select $(a^*,p^*) = \arg\max_{(a,p)} \Phi(a,p)$ among all admissible pairs.
6. If no pair is admissible, select a controlled degradation action using the priority order: compress, defer, replicate, cloud escalate.
7. Return $(a^*,p^*)$ to the local scheduler or routing controller.
```

Algorithm 2 is the communication-aware core of the model. Unlike a placement-only rule, it evaluates the route and the action jointly. A fog node may have adequate compute capacity, but the path to that node may suffer high loss or multiple unstable hops. Conversely, a stable path may still be rejected if it drains the sensor battery or generates stale information.

The route term $\Gamma(p)$ makes the model compatible with ad hoc networks because reliability decreases multiplicatively as additional lossy hops are added. The action term penalizes delay, energy, information age, risk, and occupancy in a normalized cost space. For M candidate actions and P candidate routes, the complexity is $O(MPK)$. In practical fog-assisted sensor networks, both M and P are bounded by gateway policy, making the search feasible for online operation.

Algorithm 3. Adaptive Reliability-Weight Learning

```
Input: previous weights $\omega_{t-1}$, observed outcome $\mathbf{y}_t$, desired budget target $\mathbf{B}$.
Output: updated weights $\omega_t$.
1. Observe the post-decision outcome vector $\mathbf{y}_t = \{D_t, E_t, A_t, R_t, U_t\}$.
2. Compute normalized budget error $\varepsilon_{k,t} = \max(0, y_{k,t} - B_k)/B_k$ for each dimension $k$.
3. Update each weight by
   $\omega_{k,t} = \frac{\omega_{k,t-1} + \mu\varepsilon_{k,t}}{\sum_j(\omega_{j,t-1} + \mu\varepsilon_{j,t})}.$
4. If repeated violations occur in the same dimension, increase the corresponding hard-priority flag.
5. Return the normalized vector $\omega_t$ for the next decision epoch.
```

Algorithm 3 prevents the controller from using fixed design weights when the deployment changes. A sensor network may begin as energy-limited, then become freshness-limited during a burst or trust-limited during an attack. The algorithm adapts budget weights only when observed outcomes violate target limits, making the learning step conservative rather than continuously unstable.

The mathematical update is a projected additive correction on the probability simplex. The numerator increases the importance of violated dimensions, and the denominator preserves $\sum_k \omega_k = 1$. The learning rate μ controls how quickly the fog node reacts to repeated violations. A small μ is appropriate for stable environmental monitoring, while a larger value may be justified for industrial or transportation sensing where late reactions are costly.

5. MATHEMATICAL ANALYSIS

5.1 Reliability cost

For an action a, define the normalized reliability cost as

$$\mathscr{C}(a) = \omega_D\hat{D}(a) + \omega_E\hat{E}(a) + \omega_A\hat{A}(a) + \omega_R\hat{R}(a) + \omega_U\hat{U}(a). \tag{4}$$

Here, $\hat{D}$, $\hat{E}$, $\hat{A}$, $\hat{R}$, and $\hat{U}$ are normalized delay, energy, freshness, risk, and fog-occupancy terms. The selected action is

$$a^* = \arg\min_{a\in\mathscr{A}} \mathscr{C}(a), \quad \text{s.t. } C_k(a) \le B_k. \tag{5}$$

The key feature of this formulation is that a single low-cost dimension cannot dominate the decision. A low-delay action with unacceptable energy cost is rejected; a low-energy action with stale information is also rejected. This makes the model better aligned with wireless sensor IoT than single-objective placement approaches.

5.2 Freshness and energy coupling

Information freshness can be approximated using age of information:

$$A_{i,t+1} = \begin{cases} \Delta_t, & \text{if an update is delivered,} \\ A_{i,t} + \Delta_t, & \text{otherwise.} \end{cases} \tag{6}$$

Energy evolves according to

$$e_{i,t+1} = e_{i,t} - E_{\text{tx}}(a) - E_{\text{cmp}}(a) - E_{\text{idle}}. \tag{7}$$

The two equations show why reliability budgeting is needed: reducing age normally requires transmission or computation, which consumes energy. DTRB exposes this coupling explicitly to the fog controller.

5.3 Ad hoc route reliability and budget propagation

In an ad hoc wireless sensor path $p = \{1, 2, \ldots, H\}$, a delivery decision is affected by every hop. If hop h has packet-loss estimate ℓ_h, the path reliability is

$$\Gamma(p) = \prod_{h=1}^{H}(1 - \ell_h). \tag{8}$$

For small ℓ_h, the logarithmic approximation

$$\log\Gamma(p) \approx -\sum_{h=1}^{H} \ell_h \tag{9}$$

shows that a route with many small losses can be as harmful as a short route with one weak hop. This is why the proposed controller evaluates route reliability together with fog load and event value.

The residual communication budget after a route is selected is

$$\rho_{i,t}(p) = \mathbf{B}_i - \left[D_p, E_p, A_{i,t+1}, R_{i,t+1}, U_{f,t+1}\right]. \tag{10}$$

A route-action pair is admissible only when every component of $\rho_{i,t}(p)$ is non-negative. This condition gives a simple operational rule: a route is not acceptable merely because it is connected; it must preserve the reliability budget.

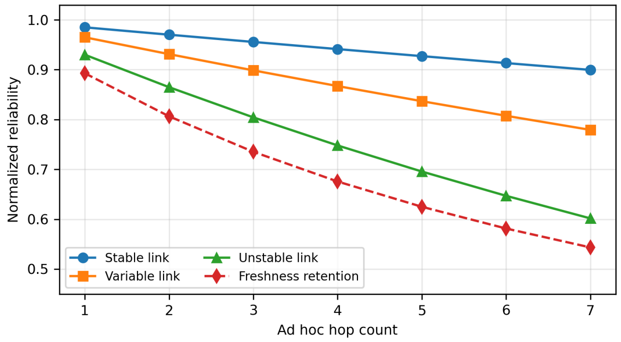


Figure 3. Multi-hop reliability decay under different ad hoc link conditions.

Figure 3 illustrates how route length affects reliability. The curve for unstable links declines more rapidly, which means that multi-hop communication should be used carefully when link loss is high. The freshness-retention curve also decreases with hop count, reinforcing the need to make route selection part of the same budget as sensing and fog processing.

5.4 Feasibility and stability of the decision rule

Let $\mathscr{F}_{i,t}$ be the feasible set of all action-route pairs satisfying the budget constraints. The proposed decision rule is stable in the following sense: if all costs are bounded and the weight vector remains in the simplex, then the selected utility $\Phi(a^*, p^*)$ is bounded above and below. This follows from

$$0 \le \sum_k \omega_k \widehat{C}_k(a,p) \le 1, \quad 0 \le \Gamma(p) \le 1, \tag{11}$$

when normalized costs and route reliability are kept in $[0,1]$. Hence the utility score cannot diverge even when sensor events arrive in bursts.

The practical implication is that DTRB can degrade gracefully. If $\mathscr{F}_{i,t}$ is nonempty, the controller selects the best admissible pair. If $\mathscr{F}_{i,t}$ is empty, it selects a controlled degradation action, such as compression or deferral, rather than forwarding all events blindly.

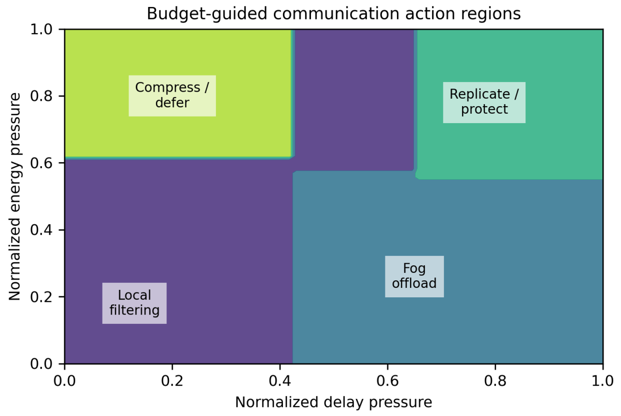
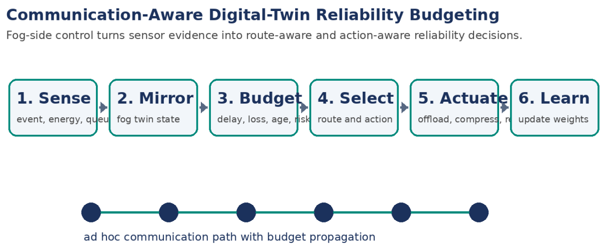

Budget-guided communication action regions

Figure 4. Illustrative action regions produced by delay and energy pressure under reliability budgeting.

Figure 4 demonstrates the behavior of the budget rule in a simplified two-dimensional view. Low delay and low energy pressure favor local filtering. High delay pressure with tolerable energy cost favors fog offloading. When both pressures become high, replication or protective escalation becomes reasonable only for valuable events. Routine events should instead be compressed or deferred.

5.5 End-to-end communication control flow

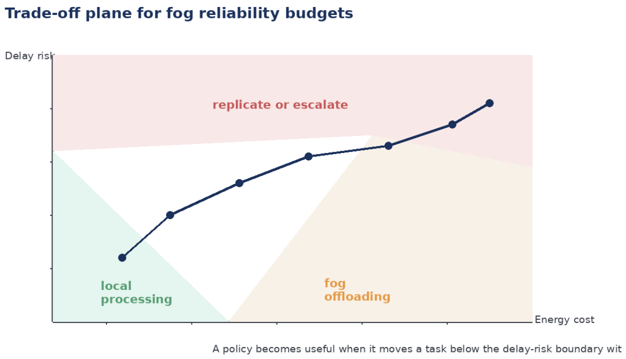
The algorithms can be interpreted as a closed communication loop: sensing evidence is mirrored by the twin, converted into budgets, and then used to select a route-aware action. Figure 5 summarizes this process and highlights the role of ad hoc path propagation.


Communication-Aware Digital-Twin Reliability Budgeting

Fog-side control turns sensor evidence into route-aware and action-aware reliability decisions.

Figure 5. Communication control flow for DTRB in fog-assisted wireless sensor ad hoc networks.

The flow confirms that the proposed model is not a cloud-centric analytics pipeline. It is a fog-side communication governor. The most important decision is not where to execute a model, but whether the next sensed event should be filtered, forwarded, replicated, compressed, or escalated under the present communication state.


Trade-off plane for fog reliability budgets

A policy becomes useful when it moves a task below the delay-risk boundary wit

Figure 6. Trade-off regions between delay risk and energy cost under fog reliability budgeting.

Figure 6 illustrates the decision boundary. Local processing is preferred when delay risk and energy cost are both low. Fog offloading becomes useful when local computation is expensive but fog delay remains acceptable. Replication or cloud escalation is reserved for high-risk cases where local and fog actions cannot satisfy the reliability budget.

6. SCENARIO-BASED ANALYSIS

The proposed model can be interpreted through recurring wireless sensor IoT and ad hoc communication scenarios. The purpose of this section is not to report a new benchmark, but to analyze how the reviewed design principles behave when the fog node must make a practical decision under constraints.

Table 4. Scenario analysis of the proposed reliability-budgeting logic

Scenario	Dominant pressure	DTRB decision tendency	Risk if budget is ignored
Industrial vibration	Freshness and trust	Replicate only abnormal event windows	False stability under delayed alerts
Smart building	Energy and occupancy	Compress routine readings, offload exceptions	Premature battery depletion
Healthcare sensing	Delay and auditability	Fog processing with cloud summary	Unexplained late decision
Agriculture field	Energy and loss	Defer low-value updates during poor links	Empty batteries with limited information gain
Transportation sensor	Delay and risk	Local filtering plus fog escalation	Congestion during short bursts

In industrial vibration monitoring, a small change can be more important than a large volume of normal samples. DTRB therefore favors replication or escalation only when the twin detects a state change that threatens the reliability budget. This differs from ordinary periodic offloading, where all updates may be treated as equal even when their operational value differs substantially.

In smart-building and agricultural sensing, the reliability problem is usually energy dominated. The fog node should avoid frequent offloading when a sensor is stable, but it should also avoid keeping silent long enough to make the twin stale. DTRB resolves this by treating freshness and residual energy as competing terms in the same budget. Healthcare and transportation cases place more emphasis on delay and auditability, because a late or unexplained control action can be operationally unacceptable even when the raw packet is eventually delivered.

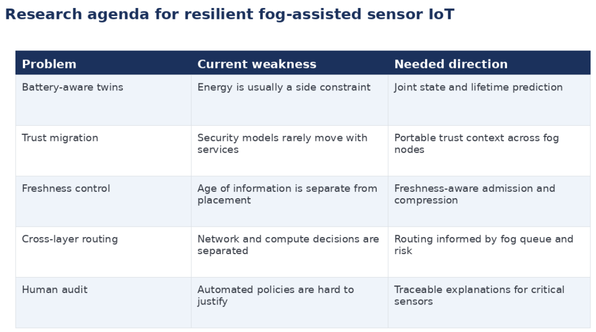
6.1 Design boundary conditions

DTRB is most appropriate when three conditions hold. First, sensors generate repeated events whose value depends on state changes. Second, a fog node can maintain a low-dimensional twin state without storing complete raw streams. Third, the application can express acceptable boundaries for delay, energy, freshness, and risk. When these conditions are absent, a simpler rule-based offloading scheme may be sufficient.

The model also has limitations. Budget weights may be difficult to tune during early deployment. Trust and risk values may be noisy if the underlying intrusion detection layer is immature. In addition, multi-hop wireless sensor networks can introduce route changes that are not fully visible to a single fog node. These limitations do not invalidate the model, but they show where future empirical validation is needed.

7. DISCUSSION AND RESEARCH DIRECTIONS

The proposed reading of the field leads to three observations. First, fog-assisted sensing should be evaluated through joint budgets, not isolated metrics. A system that improves delay while exhausting sensor batteries is not reliable. Second, digital twins should be operational, not merely descriptive. A twin is valuable when it influences sampling, placement, compression, or routing. Third, security must be embedded in the same budget as latency and energy, because attacks and faults change the reliability of sensed information.



Research agenda for resilient fog-assisted sensor IoT

Problem	Current weakness	Needed direction
Battery-aware twins	Energy is usually a side constraint	Joint state and lifetime prediction
Trust migration	Security models rarely move with services	Portable trust context across fog nodes
Freshness control	Age of information is separate from placement	Freshness-aware admission and compression
Cross-layer routing	Network and compute decisions are separated	Routing informed by fog queue and risk
Human audit	Automated policies are hard to justify	Traceable explanations for critical sensors

Figure 7. Research agenda derived from the review and proposed model.

Figure 7 converts the review into a research agenda. Battery-aware twins, trust migration, freshness-aware admission control, cross-layer routing, and human-auditable fog policy are presented as key directions. These directions are compatible with recent work on AI-based fog resource management [16], network digital twins [19], and IoT security datasets [22].

Table 5. Design implications for fog-assisted wireless sensor IoT

Design issue	Conventional response	DTRB-oriented response
Delay bursts	Increase fog capacity	Admit only events with feasible budget
Battery depletion	Reduce sampling rate	Couple sampling with twin-state urgency
Stale information	Forward more updates	Balance freshness against energy and loss
Suspicious traffic	Run separate detector	Treat risk as a reliability budget term
Fog overload	Migrate services	Use twin state before migration decisions
Cloud dependence	Forward summaries	Escalate only when local/fog action is infeasible

Table 5 emphasizes that DTRB does not replace existing fog, edge intelligence, or digital twin methods. Instead, it gives them a governing layer. This is particularly useful for sparse or battery-constrained sensor deployments where repeated wrong decisions can be more damaging than a single delayed packet.

7.1 Practical adoption path

A realistic deployment can adopt DTRB gradually. The first stage is passive monitoring, where the fog node estimates budgets without changing sensor behavior. The second stage is advisory control, where the fog node recommends compression or offloading but does not enforce it. The third stage is closed-loop control, where budget violations trigger direct actuation. This staged adoption is important because many sensor systems are safety-critical and cannot accept abrupt policy changes.

The model is also compatible with federated edge learning. A fog node may learn local budget weights from its own sensor population, while sharing only compact model updates with neighboring fog nodes. This design follows the communication-aware reasoning of federated mobile edge networks [15] and the privacy concerns surveyed by Mothukuri et al. [21]. The twin state remains local unless escalation is justified by a budget violation.

8. CONCLUSION

This review examined fog-assisted wireless sensor IoT through the lens of reliability budgeting. The literature shows strong progress in fog computing, edge intelligence, federated learning, digital twins, and IoT security, but these directions are often developed separately. The proposed DTRB model offers a compact bridge by maintaining a fog-side digital twin and making each sensor action subject to delay, energy, freshness, risk, and occupancy budgets.

The main conclusion is that future fog-assisted sensing systems should not be designed as simple offloading pipelines. They should operate as reliability-governed continua, where local devices, fog nodes, and cloud services coordinate around the current state of the sensor and the value of the sensed event. Further research should validate DTRB in multi-hop deployments, integrate it with routing protocols, and study how budget weights can be learned without reducing explainability.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002, doi: 10.1016/S1389-1286(01)00302-4.

[2] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008, doi: 10.1016/j.comnet.2008.04.002.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013, doi: 10.1016/j.future.2013.01.010.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016, doi: 10.1109/JIOT.2016.2579198.

[5] A. Yousefpour et al., “All one needs to know about fog computing and related edge computing paradigms: A complete survey,” *Journal of Systems Architecture*, vol. 98, pp. 289–330, 2019, doi: 10.1016/j.sysarc.2019.02.009.

[6] C. Puliafito, E. Mingozzi, C. Vallati, F. Longo, and G. Merlino, “Fog computing for the Internet of Things: A survey,” *ACM Transactions on Internet Technology*, vol. 19, no. 2, pp. 1–41, 2019, doi: 10.1145/3301443.

[7] P. Bellavista, J. Berrocal, A. Corradi, and S. K. Das, “A survey on fog computing for the Internet of Things,” *Pervasive and Mobile Computing*, vol. 52, pp. 71–99, 2019, doi: 10.1016/j.pmcj.2018.12.007.

[8] P. Porambage et al., “Survey on multi-access edge computing for Internet of Things realization,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2961–2991, 2018, doi: 10.1109/COMST.2018.2849509.

[9] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, “Mobile edge computing: A survey,” IEEE Internet of Things Journal, vol. 5, no. 1, pp. 450–465, 2018, doi: 10.1109/JIOT.2017.2750180.

[10] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, “Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption,” *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1171–1181, 2016, doi: 10.1109/JIOT.2016.2565516.

[11] J. Ni, K. Zhang, X. Lin, and X. S. Shen, “Resource allocation in fog computing: A survey,” *IEEE Access*, vol. 6, pp. 41667–41686, 2018, doi: 10.1109/ACCESS.2018.2859159.

[12] H. Li, K. Ota, and M. Dong, “Learning IoT in edge: Deep learning for the Internet of Things with edge computing,” *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018, doi: 10.1109/MNET.2018.1700202.

[13] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, “Edge intelligence: Paving the last mile of artificial intelligence with edge computing,” *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019, doi: 10.1109/JPROC.2019.2918951.

[14] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, “Convergence of edge computing and deep learning: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 869–904, 2020, doi: 10.1109/COMST.2020.2970550.

[15] W. Y. B. Lim et al., “Federated learning in mobile edge networks: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020, doi: 10.1109/COMST.2020.2986024.

[16] S. Iftikhar et al., “AI-based fog and edge computing: A systematic review, taxonomy and future directions,” *Internet of Things*, vol. 21, 100674, 2023, doi: 10.1016/j.iot.2022.100674.

[17] R. Minerva, G. M. Lee, and N. Crespi, “Digital twin in the IoT context: A survey on technical features, scenarios, and architectural models,” *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1785–1824, 2020, doi: 10.1109/JPROC.2020.2998530.

[18] A. Fuller, Z. Fan, C. Day, and C. Barlow, “Digital twin: Enabling technologies, challenges and open research,” *IEEE Access*, vol. 8, pp. 108952–108971, 2020, doi: 10.1109/ACCESS.2020.2998358.

[19] Y. Wu, K. Zhang, and Y. Zhang, “Digital twin networks: A survey,” *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13789–13804, 2021, doi: 10.1109/JIOT.2021.3079510.

[20] H. Xu, J. Wu, Q. Pan, X. Guan, and M. Guizani, “A survey on digital twin for industrial Internet of Things: Applications, technologies and tools,” *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2569–2598, 2023, doi: 10.1109/COMST.2023.3297395.

[21] V. Mothukuri et al., “A survey on security and privacy of federated learning,” *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021, doi: 10.1016/j.future.2020.10.007.

[22] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, “Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning,” *IEEE Access*, vol. 10, pp. 40281–40306, 2022, doi: 10.1109/ACCESS.2022.3165809.

[23] A. M. Rahmani et al., “Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach,” *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018, doi: 10.1016/j.future.2017.02.014.

[24] S. Tuli et al., “HealthFog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments,” *Future Generation Computer Systems*, vol. 104, pp. 187–200, 2020, doi: 10.1016/j.future.2019.10.043.