SAGE-CAN: a slack-coupled elastic gateway for deterministic CAN-to-IoT convergence

An analytical and simulation study on ESP32-class edge hardware

[1]Pravat Satpathy, [2] Sonia Mohapatra
[1]Researcher, [2]Researcher,
[1,2] R & D Division
[1,2]Tekstein Solutions, India
[1]pravat@teksteinsolutions.com , sonia@teksteinsolutions.com

Abstract—Controller Area Network (CAN) segments and Internet-of-Things backhauls impose contradictory service disciplines: CAN is priority-arbitrated and analytically bounded, whereas Wi-Fi and MQTT transport is best-effort and subject to multi-second outages. Gateways bridging the two typically resolve this with a fixed or queue-driven aggregation window, both agnostic to the criticality of the frames they carry. This paper presents SAGE-CAN, in which the uplink aggregation window is coupled to the residual worst-case response-time slack of the CAN message set and hard-capped by a per-criticality escalation deadline, combined with tier-aware admission and delta-encoded batching. A slack estimator is derived from revised CAN response-time analysis and five algorithms are specified. Discrete-event simulation over a sixteen-message industrial rotating-machinery workload at 500 kbit/s under a Gilbert-Elliott backhaul shows tier-1 on-time delivery rising from 57.62 to 93.49 percent against the strongest baseline, mean tier-1 latency falling from 200.5 to 59.0 ms, worst-case latency bounded at 668.7 against 3299.1 ms, and uplink volume reduced 8.2-fold at 29 percent lower power. An ablation establishes that escalation capping and delta encoding dominate, while slack coupling and tiering remain dormant below approximately 85 percent bus utilisation, delimiting each mechanism's operating regime. All results are analytical or simulated; no hardware measurements are claimed and a validation protocol is specified.

Index Terms—Controller Area Network, CAN FD, edge computing, ESP32, industrial Internet of Things, real-time scheduling, response-time analysis, gateway architecture, condition monitoring, TWAI.

I. INTRODUCTION

Controller Area Network remains the dominant field bus for distributed embedded control, owing its longevity to non-destructive bitwise arbitration, a noise-tolerant differential physical layer, distributed error confinement and, critically, an analytical foundation permitting worst-case message latencies to be bounded before a system is built.

The pressure now acting on CAN installations comes from what is attached above them. Condition monitoring and predictive maintenance require field data to reach a cloud tier, and the gateway performing that translation occupies an awkward position: below it lies a network whose defining property is bounded latency, above it a transport for which no bound exists. A gateway forwarding each frame as an individual MQTT publication consumes uplink capacity and radio energy out of all proportion to the information carried and saturates long before the bus does. A gateway aggregating frames into periodic batches solves that and destroys the latency property that justified choosing CAN.

The prevailing response is a fixed window of one hundred to five hundred milliseconds, occasionally modulated by queue occupancy. Both variants share the defect this paper takes as its starting point: they are indifferent to what they carry. An emergency-stop annunciation and an ambient-temperature reading are held equally long and discarded with equal probability. The mismatch is structural rather than a matter of tuning, because the controlled variable, queue depth, carries no information about deadline proximity.

A. Research gap, problem and objectives

Three gaps emerge from Section II. CAN response-time analysis and IoT gateway design have developed as disjoint literatures, so the substantial body of work on worst-case response times is used to dimension buses rather than to parameterise the software draining them. Published gateway designs report aggregate throughput and mean latency but rarely disaggregate delivery by criticality, concealing the failure mode that matters. Evaluations are commonly conducted under a benign backhaul, whereas multi-second outages are the normal industrial condition.

The problem is therefore: given a schedulable CAN message set and an intermittent best-effort uplink of unknown instantaneous capacity, determine an aggregation and admission policy maximising the fraction of high-criticality frames delivered within a per-criticality end-to-end deadline, subject to gateway memory, radio duty and power constraints, without perturbing the timing of the CAN segment itself. The objectives are to derive an online slack estimator from revised response-time analysis; to define a window law bounded by a criticality escalation deadline; to specify a tiered admission discipline; to quantify the result against representative baselines; and to

identify by ablation which mechanisms are actually load-bearing.

B. Contributions

This paper contributes the coupling of CAN response-time slack to edge uplink scheduling, a link not previously exploited; an escalation-capped elastic window law giving an analytical bound on gateway-induced tier-1 latency (Eq. 10); criticality-preserving differential aggregation with a stated loss bound; and a component ablation identifying two of the four mechanisms as inactive below roughly 85 percent bus utilisation, retained rather than suppressed because it delimits where the framework should be deployed.

A note on evidentiary status: the following sections present expected or simulated performance pending hardware validation. No physical measurement is reported. Every result originates from closed-form analysis or from the discrete-event model of Section V, with parameters drawn from component datasheets and protocol specifications. Section VII states the protocol by which these predictions should be confirmed or refuted.

TABLE I
Comparative summary of surveyed literature

#	Author (Year)	Method	Limitation
1	Tindell & Burns (1994) [3]	First fixed-priority RTA for CAN	Later shown optimistic; no gateway model
2	Davis et al. (2007) [4]	Revised CAN RTA, arbitrary deadlines	Bus-local; ignores uplink egress
3	Davis et al. (2011) [5]	RTA with FIFO controller queues	No criticality differentiation
4	Xie et al. (2021) [6]	MILP/SA extensibility for CAN FD	Design-time only; no runtime adaptation
5	Ma et al. (2021) [8]	AUTOSAR CAN FD frame packing	Offline; no criticality-aware shedding
6	Song et al. (2020) [10]	Deep CNN IDS on CAN frames	Compute cost precludes MCU deployment
7	Seo et al. (2018) [11]	GAN-based IDS (GIDS)	Training cost; offline inference
8	Wu et al. (2020) [12]	Survey of in-vehicle IDS	Survey; no gateway scheduling
9	Olufowobi et al. (2020) [16]	Timing-specification IDS	Detection only; no transport policy
10	de Araujo-Filho et al. (2021) [14]	Low-cost intrusion prevention	Security scope; no QoS scheduling
11	Baccari et al. (2024) [13]	Survey of CAV anomaly detection	Survey; no edge resource model
12	Espressif (2024-25) [20][21]	TWAI / TWAI-FD specification	Reference documentation, not analysis

II. RELATED WORK

The relevant literature divides into three streams that have progressed largely independently, summarised in Table I.

CAN timing analysis begins with Tindell and Burns, who adapted fixed-priority scheduling theory to bus arbitration [3]. That formulation was commercially productised before Davis, Burns, Bril and Lukkien showed it optimistic, in that it could certify sets that would miss deadlines in the worst case, and supplied the corrected analysis adopted here [4]. Later refinements addressed non-ideal FIFO transmit queues [5]. The stream is mature but terminates at the bus boundary: it dimensions the network and says nothing about the software consuming it.

A second stream addresses CAN FD design-time optimisation. Frame packing has been treated as an integer program [9], as an AUTOSAR-compliant near-optimal problem [8], and jointly with security and extensibility objectives [6], [7]. These are static optimisations over a fixed signal set that assume the egress path is not a constraint, which is precisely the assumption failing when the consumer is reached over an intermittent radio link.

The largest stream concerns intrusion detection, spanning convolutional models over frame images [10], generative detection of unseen attacks [11], unsupervised high-dimensional models [15], timing-specification monitoring [16], one-class support vector data description [18], and edge-assisted recurrent models [17]; surveys map the field [12], [13]. Complementary work addresses authentication for CAN and CAN FD [19], [23]. Two observations apply. Most detection models sit at cloud or FPGA rather than microcontroller scale. More importantly, this stream decides whether traffic is legitimate, not which legitimate traffic to transmit when capacity is insufficient; the scheduling question is orthogonal and comparatively neglected.

Work on ESP32-class gateways targets smart-home telemetry over MQTT and REST [22], establishing platform practicality without confronting a deterministic field bus on ingress. The consolidated gap: no reviewed contribution uses residual CAN timing slack as a control input to the uplink policy, and none reports delivery disaggregated by criticality under a degraded backhaul.

III. ARCHITECTURE AND ANALYTICAL MODEL

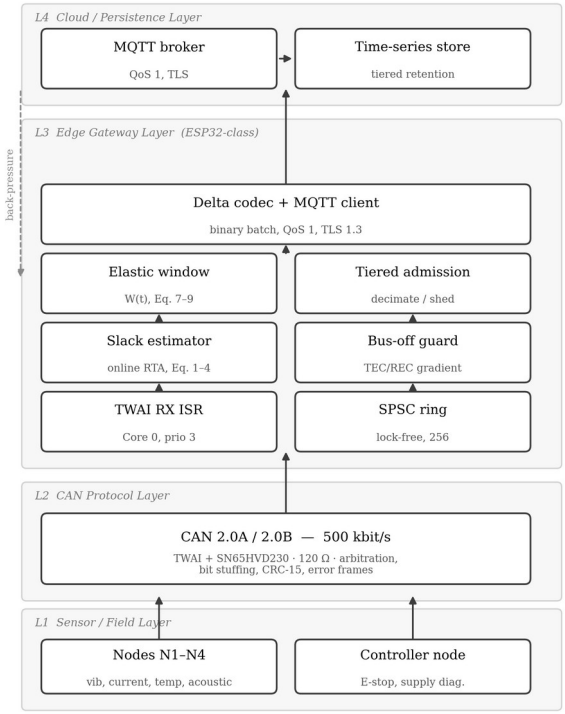

Fig. 1 SAGE-CAN layered architecture. Timing information propagates upward from the CAN protocol layer into the edge scheduler; back-pressure propagates downward. The CAN segment is unmodified.

A. Layered architecture

SAGE-CAN comprises four layers (Fig. 1). The reference node is an ESP32-class device. A point of frequent confusion warrants explicit statement: the TWAI controller in the original ESP32 and ESP32-S3 implements classical CAN only and interprets a flexible data-rate frame as an error [20]; CAN FD there requires an external controller such as the MCP2518FD over SPI, whereas later parts including the ESP32-C5 accept both formats natively [21]. Evaluation uses classical CAN at 500 kbit/s. In all cases the device integrates only the protocol controller; an SN65HVD230 transceiver converts logic signals to the differential pair, with 120 ohm termination at both extremities.

The dual-core organisation is exploited for temporal isolation, the principal implementation contribution. Wi-Fi and TCP/IP stacks generate bursty high-priority interrupt activity of unbounded duration, so a CAN receive path sharing a core acquires large and intractable jitter. Core 0 therefore hosts only the TWAI interrupt service routine, timestamping and the producer side of a lock-free 256-slot ring, performing no allocation or formatting; Core 1 hosts the window controller, admission filter, delta codec, MQTT client and Wi-Fi stack. The cores share only the ring and two atomic counters, so no priority-inversion path exists between radio and bus.

B. Frame timing and response time

With bit time $\tau bit = 1/Rb$, the worst-case transmission time of a classical frame carrying sm payload bytes, including maximum bit stuffing and the three-bit interframe space, is

$$Cm = (g + 10\,sm)\,\tau bit \qquad (1)$$

with g = 55 for standard and 80 for extended identifiers [4]; the factor of ten accounts for worst-case stuff bits. Bus utilisation is $U = \Sigma\, Cm/Tm$. Blocking arises because arbitration is non-preemptive once transmission begins:

$$Bm = \max_{k \in lp(m)} Ck \qquad (2)$$

and the queueing delay of instance q within the busy period satisfies

$$wm(q) = Bm + qCm + \Sigma_{k \in hp(m)} \lceil (wm(q) + Jk + \tau bit)/Tk \rceil\, Ck \qquad (3)$$

solved by iteration from wm(0) = Bm, giving Rm(q) = Jm + wm(q) − qTm + Cm and Rm = max Rm(q) over the busy period [4]. Deadlines are taken equal to periods.

C. The slack estimator

The quantity coupling bus to uplink is the minimum normalised residual slack,

$$Smin = \min_{m \in M} (Tm - Rm)/Tm \qquad (4)$$

which lies in (0,1] for any schedulable set and approaches zero at the feasibility boundary. It is computed once at configuration time and rescaled online by measured bus load, so runtime cost is a single multiplication. It expresses the fraction of each period unconsumed by worst-case contention, and therefore the share of the timing budget the gateway may legitimately spend aggregating.

D. Uplink efficiency, delay and energy

With Ω the invariant per-publication wire overhead (MQTT, TLS, TCP/IP and 802.11 headers), β the mean encoded record size and N the batch size, transport efficiency $\eta = N\beta/(\Omega + N\beta)$ is concave in N: with Ω = 139 bytes and β = 6.4 bytes it rises from 0.041 at N = 1 to 0.731 at N = 64. That concavity is why aggregation is attractive and equally why its marginal benefit collapses beyond a moderate batch, bounding the sensible window independently of latency. End-to-end delay decomposes as

$$Dete = Rm + Wwait + Qup + Tpub \qquad (5)$$

of which only Wwait is under gateway control and is precisely what fixed-window schemes hold constant. Modelling the uplink as a deterministic server, $E[Qup] = \rho up Tpub/(2(1-\rho up))$, which diverges as ρup approaches unity; per-frame forwarding drives ρup above unity for the workload of Table II. Mean power is modelled as

$$P = V(ITX(tair + \lambda tstack) + Iidle(1-\delta) + ICPU) \qquad (6)$$

with δ the radio duty cycle and λ = 0.55 scaling mean current during stack processing. Since tair scales with bytes while tstack scales with publication count, Eq. 6 predicts that energy is dominated by publication count at low batch sizes.

E. The elastic window law

Let ρq be staging-buffer occupancy, D1 the tier-1 deadline budget and a1 the age of the oldest resident tier-1 record. The proposed window is the minimum of a slack term and an escalation cap:

$$Wslack = Wmax \cdot Smin \cdot \sqrt{(1-\rho q)} \qquad (7)$$

$$Wesc = \kappa D1 - a1(t),\ \kappa = 0.45 \qquad (8)$$

$$W(t) = clamp(min(Wslack, Wesc), Wmin, Wmax) \qquad (9)$$

The square root contracts the window gradually under mild pressure and sharply near saturation. The coefficient κ

apportions the tier-1 budget between buffer residency and the remaining transport terms of Eq. 5, reserving the majority for publication and network delay, the dominant uncertainty. From Eq. 8, gateway-induced residency of any tier-1 record obeys

$$W_{wait}(\text{tier } 1) \le \kappa D1 = 67.5 \text{ ms} \qquad (10)$$

This bound is the analytical core of the contribution and is what the worst-case latencies of Section VI reflect.

F. Tiered admission, loss bound and bus-off prediction

Under scarcity, records of tier $\tau > 1$ are admitted with decimation factor $d(\tau, \rho q, c)$ where c is channel state; tier 1 is never decimated. For a signal decimated by d at period T, zero-order-hold reconstruction error is bounded by $\varepsilon \le dT \cdot \sup|ds/dt|$, so decimation is admissible when the decimated interval times the maximum slew rate stays within downstream tolerance. For the thermal and acoustic signals in tiers 2 and 3, whose slew rates are small relative to their periods, d = 12 remains within tolerance; this justifies the tier assignment rather than making it arbitrary. Finally, rather than react to bus-off, the framework predicts it from the transmit-error-counter gradient $\Gamma = (TEC(t) - TEC(t-\Theta))/\Theta$, projecting time to bus-off as $(256-TEC)/\Gamma$ and raising a tier-1 diagnostic when this falls below a configured horizon.

IV. ALGORITHMS

Algorithm 1: Sensor acquisition and frame construction

```
for each sensor s do
  x <- median_filter(ADC_read(s), w=5)
  y <- K[s].gain*x + K[s].offset      // 2-pt calib
  if |y-last[s]|>deadband[s] or age[s]>T[s] then
    twai_transmit(pack(id[s],scale(y),dlc[s]))
    last[s] <- y ; age[s] <- 0
end for
```

Algorithm 2: Slack estimation and elastic window

```
offline: R_m by Eq.3 ; S_min by Eq.4
each epoch:
  S <- S_min*(1-(U_hat-U_nom)/(1-U_nom))
  rho <- |Q|/Q_cap
  W_slack <- W_max*S*sqrt(max(0,1-rho))    // Eq.7
  if tier-1 record in Q then
    W <- min(W_slack, kappa*D_1 - age(oldest_t1))
  else W <- W_slack
  return clamp(W, W_min, W_max)             // Eq.9
```

Algorithm 3: Criticality-tiered priority allocation

```
on arrival of f with tier t:
  if t=1 then admit(f); return
  if channel=BAD or rho>0.55 then
    d <- (t=3)?4:2
    if channel=BAD and rho>0.75 then d<-(t=3)?12:4
    cnt[t]<-cnt[t]+1
    if cnt[t] mod d != 0 then shed(f); return
  if |Q|=Q_cap then
    v <- newest record of highest tier in Q
    if tier(v)>t then evict(v) else drop(f); return
  admit(f)
```

Algorithm 4: Cloud synchronisation with delta batching

```
wait until age(oldest)>=W or |Q|>=N_max
B <- first min(|Q|,N_max) records
hdr <- {node_id, epoch_ts, count, crc32}
for r in B: enc <- varint(id_index(r))
            || zigzag(ts(r)-epoch_ts)
            || zigzag(value(r)-baseline[id(r)])
            ; baseline[id(r)] <- value(r)
publish(topic, hdr||enc*, QoS=1)
if ack within T_ack then dequeue(B) else retry once
```

Algorithm 5: Fault detection and bus-off prediction

```
every Theta ms:
  Gamma <- (TEC-TEC_prev)/Theta
  if Gamma>0 and (256-TEC)/Gamma < H then
    raise DIAG_BUSOFF_IMMINENT; throttle by 2
  if REC>127 then raise DIAG_ERROR_PASSIVE
  if silent(id)>3*T[id] then raise DIAG_NODE_LOST
  TEC_prev <- TEC
```

The tier-1 escalation sequence is: the frame completes arbitration at t0; the Core-0 ISR timestamps and enqueues it; Algorithm 2 returns Wesc rather than Wslack on the next epoch; Algorithm 4 fires no later than t0 + κD1. The uplink task has four states, IDLE, ACCUMULATING, PUBLISHING and RETRY, with transitions on first admission, on window expiry or batch cap or escalation, on acknowledgement, on timeout, and on second failure with loss accounting.

V. SIMULATION METHODOLOGY

The following presents expected or simulated performance pending hardware validation; no physical measurement is reported.

The workload models a rotating-machinery condition-monitoring installation: four instrumentation nodes and one controller on a single 500 kbit/s segment carrying sixteen periodic messages under standard identifiers. Table II gives the specification with computed worst-case response times. Bus utilisation is 12.09 percent, aggregate frame rate 526 frames per second, minimum normalised slack 0.889. Three criticality tiers carry end-to-end budgets of 150, 1000 and 10 000 ms, assigned by the slew-rate argument of Section III.F.

The uplink is a two-state Gilbert-Elliott channel on a 100 ms tick. In the good state publication costs 6 ms fixed with 3500 kbit/s effective goodput after TLS; in the bad state 140 ms fixed, 220 kbit/s, and failure probability 0.22 triggering one QoS-1 retry. Transition probabilities were swept to give mean outage fractions from 0 to 27.71 percent.

Three baselines are compared. BASE-FWD publishes each frame individually as JSON. FIXED-AGG applies a 200 ms window with JSON batching and drop-tail queueing, representing common practice. QUEUE-ADAPT adapts the window to queue occupancy alone with binary batching, representing the strongest criticality-agnostic scheme. All share 512-record staging and a 128-record batch cap.

Each configuration ran over 24 independent seeds of 300 simulated seconds for the primary comparison and 8 to 12 for sweeps and ablations; intervals are 95 percent confidence intervals, and comparisons use Welch's t-test with Cohen's d. Two delivery metrics are distinguished and must not be conflated: packet delivery ratio is the fraction of frames reaching the broker at all, whereas on-time delivery ratio is the fraction arriving within their tier budget. The latter is the metric of interest, since a reading four seconds late has not been usefully delivered.

TABLE II
Evaluation message set and worst-case response times (500 kbit/s, U = 12.09 %)

ID	Message	T (ms)	DLC	C (µs)	R (µs)	Tier
0x001	EMERGENCY_STOP	10	2	150	520	1
0x002	BEARING_VIB_RMS_A	10	8	270	840	1
0x003	BEARING_VIB_RMS_B	10	8	270	1110	1
0x004	MOTOR_CURRENT_3PH	20	8	270	1430	1
0x005	SHAFT_SPEED	20	4	190	1620	1
0x006	TORQUE_EST	20	6	230	1850	2
0x007	TEMP_BEARING_A	100	4	190	2140	2
0x008	TEMP_BEARING_B	100	4	190	2330	2
0x009	TEMP_WINDING	100	6	230	2560	2

ID	Message	T (ms)	DLC	C (µs)	R (µs)	Tier
0x00A	LUBE_PRESSURE	100	4	190	2750	2
0x00B	VIB_SPECTRAL_BAND	50	8	270	3120	2
0x00C	ACOUSTIC_ENV	100	8	270	3390	3
0x00D	AMBIENT_TH	500	4	190	3680	3
0x00E	SUPPLY_RAIL_DIAG	500	6	230	3910	3
0x00F	NODE_HEARTBEAT	1000	2	150	4060	3
0x010	CONFIG_ECHO	1000	8	270	4160	3

TABLE III
Primary comparison at nominal load (U = 12.09 %, 24 seeds, 300 s each)

Metric	BASE-FWD	FIXED-AGG	QUEUE-ADAPT	SAGE-CAN
Publications s^{-1}	145.45	4.79	4.17	13.76
Uplink (kbit/s)	266.47	324.04	28.35	39.68
Mean power (mW)	565.56	245.76	147.41	173.97
Radio duty (%)	95.27	16.91	4.32	10.26
Mean window (ms)	—	200.0	1993.9	1777.7
Tier 1 PDR (%)	28.10	96.20	99.64	99.66
Tier 1 on-time (%)	0.02	57.62	54.79	**93.49**
Tier 1 mean lat. (ms)	3562.3	200.5	145.5	**59.0**
Tier 1 p99 (ms)	13921.5	2169.6	434.0	**384.0**
Tier 1 max (ms)	16922.9	3299.1	1620.7	**668.7**
Tier 2 on-time (%)	0.13	93.34	99.57	96.56
Tier 3 on-time (%)	25.90	96.15	99.64	95.00

VI. RESULTS AND DISCUSSION

A. Primary comparison

Table III reports the comparison at nominal load. BASE-FWD is infeasible: a 95.27 percent radio duty cycle indicates uplink saturation, the workload demanding roughly 3.2 times available publication capacity, so only 28.10 percent of tier-1 frames are delivered at all and effectively none within budget — consistent with the divergence predicted as ρup approaches unity.

Both aggregating baselines deliver almost all frames but only slightly over half of tier-1 frames on time: 57.62 percent for FIXED-AGG and 54.79 for QUEUE-ADAPT. The reasons differ. FIXED-AGG holds every record up to 200 ms, alone exceeding a substantial share of the 150 ms budget. QUEUE-ADAPT, its queue never approaching capacity at this load, holds its window near the 1994 ms maximum and is rescued only by the batch cap. Neither distinguishes a tier-1 record from a heartbeat.

SAGE-CAN achieves 93.49 percent, a gain of 35.87 points over the strongest baseline. Mean tier-1 latency falls to 59.0 ms and, more significantly, the worst case is bounded at 668.7 ms against 3299.1 ms. Uplink volume falls 8.2-fold and mean power sits 29.2 percent below FIXED-AGG though 18.0 percent above QUEUE-ADAPT: the framework trades a modest energy increment for a substantial determinism gain, the intended design point. Differences are significant with large effect sizes: $t = 30.81$, $d = 8.90$ against FIXED-AGG and $t = 32.98$, $d = 9.52$ against QUEUE-ADAPT across 24 seeds.

B. Backhaul degradation and utilisation sensitivity

Fig. 2 reports progressive backhaul degradation, the condition of practical interest. Under a perfect link SAGE-CAN attains 100 percent tier-1 on-time delivery with a 74.3 ms worst case, comfortably inside budget, while both baselines already fail about 40 percent of tier-1 deadlines. At 27.71 percent outage SAGE-CAN degrades to 72.36 percent against 45.47 and 42.22 percent, and its worst case rises only to 746.0 ms whereas FIXED-AGG reaches 4060.1 ms. Degradation is graceful and, more importantly, the tail stays bounded, which is exactly the property Eq. 8 was constructed to provide.

Fig. 3 sweeps bus utilisation by scaling all periods. The response-time analysis places the feasibility boundary between 80.57 percent utilisation, where minimum normalised slack is 0.153, and 92.97 percent, where slack turns negative and the set is unschedulable. Within the feasible region the mean window contracts monotonically from 1778 to 305 ms, tracking Eq. 7 as designed.

The sweep also qualifies the primary result, and this must be stated plainly. The large advantage at 12.1 percent utilisation narrows considerably above 24 percent, where all three schemes reach 92 to 98 percent. At higher frame rates the 128-record batch cap fires well before any window expires, so the baselines are involuntarily forced into short effective windows. The advantage of SAGE-CAN is therefore genuine but specific to the regime where arrivals are too sparse to trigger a capacity-driven flush. That is the common case in condition monitoring, which is characteristically light-loaded, but it is not universal.

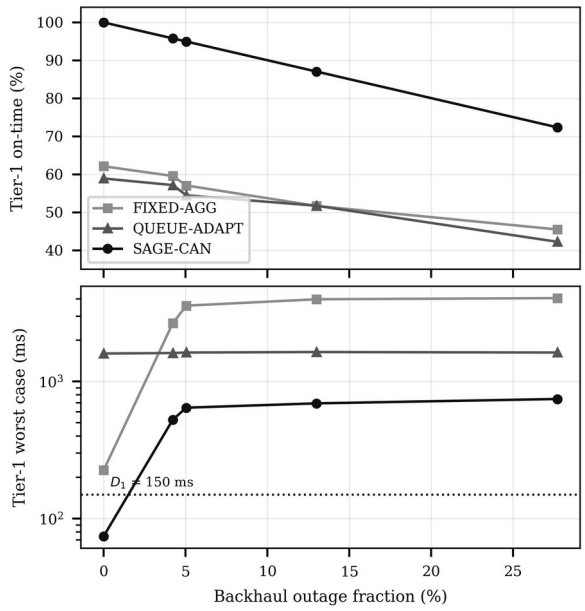

Fig. 2 Tier-1 on-time delivery (top) and worst-case latency (bottom) against backhaul outage fraction. SAGE-CAN degrades gracefully and retains a bounded tail; the baselines do neither.

C. Ablation

Table IV decomposes each mechanism, and the results are partly negative. Tier-1 escalation is decisive at nominal load: removing it collapses on-time delivery from 94.22 to 54.54 percent, reducing the framework to QUEUE-ADAPT. At high load its effect on the mean vanishes, though it still halves the worst case. Delta encoding is the second load-bearing mechanism, cutting uplink volume 8.3-fold and power by 38 percent; at high load its removal demands 1898 kbit/s, unrealisable on the modelled link.

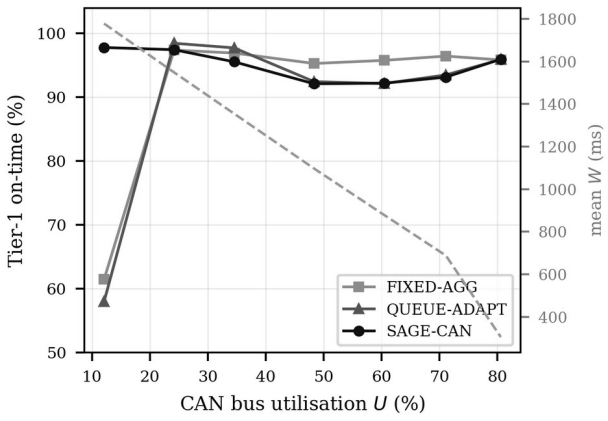

Fig. 3 Tier-1 on-time delivery against bus utilisation (left axis) and mean SAGE-CAN window (right axis, dashed). The advantage narrows above 24 %.

VII. LIMITATIONS, VALIDATION AND FUTURE WORK

TABLE IV
Component ablation: tier-1 on-time delivery (%) at two load points, with tier-1 mean and worst-case latency (ms) at nominal load

Configuration	U=12.1 %	U=71.1 %	Mean	Max
Full SAGE-CAN	94.22 ± 2.06	93.16 ± 9.83	57.1	658.8
w/o tier-1 escalation	54.54 ± 1.78	93.13 ± 9.83	145.8	1671.7
w/o slack coupling	94.22 ± 2.06	93.16 ± 9.83	57.1	658.8
w/o criticality tiering	93.53 ± 2.32	93.54 ± 9.14	60.4	669.6
w/o delta encoding	90.93 ± 2.92	96.49 ± 5.09	171.1	3527.8

Slack coupling produces no measurable effect at either load point, for a structural reason. Whenever a tier-1 record is resident, Eq. 8 evaluates to at most 67.5 ms while Eq. 7 evaluates to 1778 ms at nominal load and 688 ms at 71.1 percent utilisation, so Eq. 9 always selects the escalation term. Slack coupling can bind only where the slack term falls below κD1, requiring Smin below roughly 0.034 and hence utilisation above about 85 percent. Criticality tiering likewise contributes at most 0.7 points to tier-1 delivery while costing 4.1 points of tier-3 delivery, becoming worthwhile only when the staging buffer is genuinely contended.

The honest characterisation is therefore that SAGE-CAN is an escalation-capped, delta-encoded elastic gateway whose slack-coupling and tiering mechanisms are dormant reserves engaging near the schedulability boundary. Reporting four co-equal contributions would misrepresent it, and the practical implication is more useful than the positive findings: on a lightly loaded bus, implement escalation capping and delta encoding and omit the remainder, halving implementation effort at no measurable cost.

Two observations generalise. Queue depth is a poor proxy for urgency: a buffer holding one emergency-stop annunciation is more urgent than one holding four hundred heartbeats, and no function of occupancy alone distinguishes them. And BASE-FWD transmits fewer kilobits per second than FIXED-AGG, because saturation discards most of its input, yet consumes 2.3 times the power — per-publication fixed cost, not payload, is what the designer should minimise.

The limitations are substantial and stated without qualification. No hardware validation has been performed; every figure is analytical or simulated. The energy model rests on datasheet currents and one scaling coefficient, whereas real devices exhibit power behaviour sensitive to DTIM interval, association state and regulatory duty limits; predicted power should be treated as ordinal rather than absolute. Real industrial Wi-Fi degradation is neither two-state nor memoryless and correlates with machinery duty cycles a Gilbert-Elliott model cannot represent. The evaluation covers one message set on one segment at one bit rate, the advantage narrows above 24 percent utilisation, and the delta-encoding ratio of 0.42 is assumed rather than measured. Finally, no security analysis is offered: shedding relies on identifier fields an on-bus adversary can forge, and a spoofed tier-1 identifier would command uplink priority, so integration with the authentication schemes of [19], [23] is necessary before adversarial deployment.

The proposed validation protocol assembles a five-node testbed of ESP32-C5 modules with SN65HVD230 transceivers on a 40 m terminated segment. Bus timing ground truth is captured with a commercial analyser at 1 µs hardware-timestamp resolution and cross-referenced against gateway-

internal timestamps to characterise ISR latency and jitter. End-to-end latency is measured against a broker on the same subnet using PTP-disciplined clocks at both ends. Degradation is induced deterministically with a programmable attenuator and traffic shaper reproducing the outage fractions of Fig. 2, permitting point-by-point comparison against simulation. Energy is measured with a shunt analyser at 100 kHz over at least 10 000 publications, with λ in Eq. 6 fitted rather than assumed. Acceptance criteria are that measured tier-1 worst-case latency should not exceed the bound of Eq. 10 by more than the measured publication service time, and that measured and simulated on-time ratios agree within five percentage points. Beyond validation, four extensions are indicated: migration to CAN FD, which raises payloads to 64 bytes and alters Eq. 1 and hence the aggregation economics; multi-segment operation, introducing slack arbitration across buses of differing Smin; learned decimation conditioned on measured slew rate; and integration with lightweight message authentication.

CONCLUSION

This paper has argued that a CAN-to-IoT gateway should be scheduled on deadline proximity rather than buffer occupancy, and has given a framework doing so. An aggregation window derived from response-time slack and capped by a criticality escalation deadline yields an analytical bound on gateway-induced tier-1 residency, and simulation indicates the bound is reflected in practice: tier-1 on-time delivery rises from 57.62 to 93.49 percent, worst-case latency falls from 3299.1 to 668.7 ms, uplink volume falls 8.2-fold, and behaviour degrades gracefully to 72.36 percent under a 27.71 percent outage fraction. The ablation qualifies this in a manner the authors regard as the more useful contribution: two of four mechanisms carry the effect and the remainder are inactive below roughly 85 percent utilisation. The operating envelope has been stated rather than elided so practitioners can determine whether their deployment falls inside it. All results are analytical or simulated, and the protocol above is required before any quantitative claim is relied upon in a deployed system.

REFERENCES

[1] R. Bosch GmbH, "CAN Specification Version 2.0," Stuttgart, Germany, 1991.

[2] Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signalling, ISO 11898-1:2015, ISO, Geneva, 2015.

[3] K. W. Tindell and A. Burns, "Guaranteeing message latencies on Controller Area Network (CAN)," in Proc. 1st Int. CAN Conf., 1994, pp. 1–11.

[4] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised," Real-Time Systems, vol. 35, no. 3, pp. 239–272, Apr. 2007, doi: 10.1007/s11241-007-9012-7.

[5] R. I. Davis, S. Kollmann, V. Pollex, and F. Slomka, "Controller Area Network (CAN) schedulability analysis with FIFO queues," in Proc. 23rd Euromicro Conf. Real-Time Systems (ECRTS), 2011, pp. 45–56.

[6] Y. Xie, G. Zeng, R. Kurachi, F. Xiao, and H. Takada, "Optimizing extensibility of CAN FD for automotive cyber-physical systems," IEEE Trans. Intell. Transp. Syst., vol. 22, no. 12, pp. 7875–7886, Dec. 2021.

[7] Y. Xie, G. Zeng, R. Kurachi, H. Takada, and G. Xie, "Security/timing-aware design space exploration of CAN FD for automotive cyber-physical systems," IEEE Trans. Ind. Informat., vol. 15, no. 2, pp. 1094–1104, Feb. 2019.

[8] U. Bordoloi and S. Samii, "The frame packing problem for CAN-FD," in Proc. IEEE RTSS, 2014, pp. 284–293.

[9] W. Ma, G. Xie, R. Li, W. Liu, H. H. Li, and W. Chang, "Efficient AUTOSAR-compliant CAN-FD frame packing with observed optimality," in Proc. DATE, 2021, pp. 1899–1904, doi: 10.23919/DATE51398.2021.9473962.

[10] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," Vehicular Communications, vol. 21, p. 100198, 2020, doi: 10.1016/j.vehcom.2019.100198.

[11] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based intrusion detection system for in-vehicle network," in Proc. PST, 2018, pp. 1–6, doi: 10.1109/PST.2018.8514157.

[12] W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou, and K. Li, "A survey of intrusion detection for in-vehicle networks," IEEE Trans. Intell. Transp. Syst., vol. 21, no. 3, pp. 919–933, Mar. 2020.

[13] S. Baccari, M. Hadded, H. Ghazzai, H. Touati, and M. Elhadef, "Anomaly detection in connected and autonomous vehicles: A survey, analysis, and research challenges," IEEE Access, vol. 12, pp. 19250–19276, 2024, doi: 10.1109/ACCESS.2024.3361829.

[14] P. F. de Araujo-Filho, A. J. Pinheiro, G. Kaddoum, D. R. Campelo, and F. L. Soares, "An efficient intrusion prevention system for CAN," IEEE Access, vol. 9, pp. 166855–166869, 2021, doi: 10.1109/ACCESS.2021.3136147.

[15] M. Hanselmann, T. Strauss, K. Dormann, and H. Ulmer, "CANet: An unsupervised intrusion detection system for high dimensional CAN bus data," IEEE Access, vol. 8, pp. 58194–58205, 2020.

[16] H. Olufowobi, C. Young, J. Zambreno, and G. Bloom, "SAIDuCANT: Specification-based automotive intrusion detection using CAN timing," IEEE Trans. Veh. Technol., vol. 69, no. 2, pp. 1484–1494, Feb. 2020.

[17] K. Zhu, Z. Chen, Y. Peng, and L. Zhang, "Mobile edge assisted literal multi-dimensional anomaly detection of in-vehicle network using LSTM," IEEE Trans. Veh. Technol., vol. 68, no. 5, pp. 4275–4284, May 2019.

[18] X. Li, H. Zhang, Y. Miao, S. Ma, J. Ma, X. Liu, and K.-K. R. Choo, "CAN bus messages abnormal detection using improved SVDD in Internet of Vehicles," IEEE Internet Things J., vol. 9, no. 5, pp. 3359–3371, 2022.

[19] Y. Xiao, S. Shi, N. Zhang, W. Lou, and Y. T. Hou, "Session key distribution made practical for CAN and CAN-FD message authentication," in Proc. ACSAC, 2020, pp. 681–693.

[20] Espressif Systems, "Two-Wire Automotive Interface (TWAI) — ESP-IDF Programming Guide," 2024. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/peripherals/twai.html

[21] Espressif Systems, "TWAI — ESP32-C5 — ESP-IDF Programming Guide," 2025. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/stable/esp32c5/api-reference/peripherals/twai.html

[22] "Lightweight embedded IoT gateway for smart homes based on an ESP32 microcontroller," Computers, vol. 14, no. 9, p. 391, 2025, doi: 10.3390/computers14090391.

[23] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "CANAuth — A simple, backward compatible broadcast authentication protocol for CAN bus," in ECRYPT Workshop on Lightweight Cryptography, 2011.

[24] MQTT Version 5.0, OASIS Standard, Mar. 2019.