International Journal of Research in Computing

ORIGINAL RESEARCH
Volume 5, Issue 2 (2026)

ShieldLink: Retry-Aware Authenticated Encryption for Secure and Reliable Chiplet Interconnects

Michél Nguyen[1]

[1]University of the People, 595 E. Colorado Blvd., Suite 623, Pasadena, CA, 91101, USA

[1] michel_ng@icloud.com | https://orcid.org/0000-0001-6834-4422

Abstract

Chiplet-based systems-in-package increasingly rely on high-speed die-to-die links such as UCIe and CXL, where link-layer retry mechanisms and authenticated encryption are often implemented as separate reliability and security functions. This separation can create a time-of-check/time-of-use risk when acknowledgments advance before authentication completes. This study introduces ShieldLink, a retry-aware authenticated-encryption protocol that integrates link-layer delivery, buffer retirement, and cryptographic verification to enable secure and reliable chiplet interconnects. ShieldLink formalizes a deliverability invariant requiring CRC, sequence, and AEAD verification before receiver advancement. It defines per-frame authentication (Mode A) and epoch authentication (Mode B), evaluates them with a discrete-event Gilbert–Elliott burst-error model, and includes bounded safety exploration plus RTL control-plane resource sizing. Mode A removes the validity-before-verification race while improving goodput by approximately 2.4–2.5 percentage points over the secure naive baseline at representative burst probabilities. Mode B improves wire efficiency, reaching about 0.926 at $M = 32$, but its epoch-retransmission cost causes a crossover around $\pi_B \approx 0.04$ under the default $\beta = 0.2$ stress regime. The results show that authenticated delivery must be treated as part of retry semantics rather than an afterthought. ShieldLink provides an auditable design invariant and practical guidance on mode selection for future secure chiplet interconnect adapters, with relevance to SDG 9: Industry, Innovation and Infrastructure.

KEYWORDS
Chiplet Interconnect, Authenticated Encryption, ARQ, Gilbert–Elliott model, UCIe.

ARTICLE HISTORY
Published: 16 July 2026

DATA/CODE AVAILABILITY
Data and code are publicly available at: `https://github.com/pinkysworld/ShieldLink`; Code, Results, RTL and Figures.

SDG ALIGNMENT
SDG 4
SDG 9
SDG 16

1 Introduction

Chiplet integration is rapidly becoming the default path to scale compute density and yield, pushing more system functionality onto high-speed die-to-die (D2D) links. Standards such as Universal Chiplet Interconnect Express (UCIe) and Compute Express Link (CXL) [1], [2] define interoperable PHY and transaction semantics. Still, they implicitly assume that link reliability mechanisms and cryptographic integrity checks are mutually safe. In practice, designers often deploy

(i) CRC + automatic repeat request (ARQ) to tolerate transient bit errors, and

(ii) authenticated encryption with associated data (AEAD) to protect confidentiality and integrity against an interposer adversary.

This paper focuses on a subtle but high-impact composition hazard: a receiver may acknowledge (ACK) a frame after it passes a fast CRC check and before the slower AEAD verification completes. That early ACK can free buffers and advance sequence windows, creating a validity-before-verification race (a link-layer TOCTOU [8]). An active attacker who can induce or inject corrupted ciphertext can exploit this window to desynchronize the reliability state from the cryptographic state, thereby causing silent drops, use-after-free of payload buffers, or head-of-line stalls that are invisible to the ARQ layer.

ShieldLink addresses this problem by enforcing a deliverability invariant (D-Inv): the receiver must not emit an ACK that advances the sender's base sequence beyond any frame whose payload might later be rejected by AEAD. Intuitively, ACK implies deliverable. Figure 1 contrasts naive stacking with ShieldLink's co-designed gate. Figure 2 sketches the race exploited by early ACK, and Figure 3 summarizes the receiver control states that separate reliability NAKs (recoverable) from security drops (non-recoverable without rekey/reset).

Contributions:

1. We formalize the deliverability invariant (D-Inv) for D2D links with retransmission and authenticated delivery.
2. We propose ShieldLink Mode A (per-frame authentication with ACK gating) and Mode B (epoch authentication) as two operating points.
3. We evaluate wire overhead, goodput, and tail commit latency under a Gilbert–Elliott burst-error model, and report confidence intervals across random seeds.
4. We provide supplementary scripts and RTL stubs to support reproducibility and independent validation.

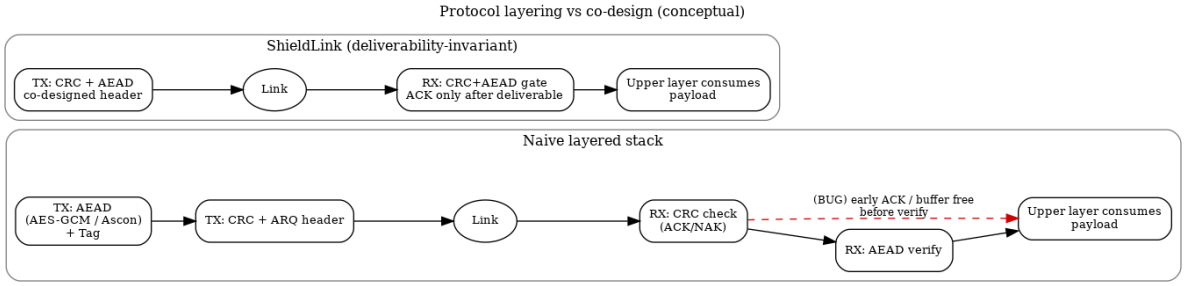

Figure 1: Naive layering can ACK on CRC before AEAD finishes, violating deliverability. ShieldLink couples verification to ACK.

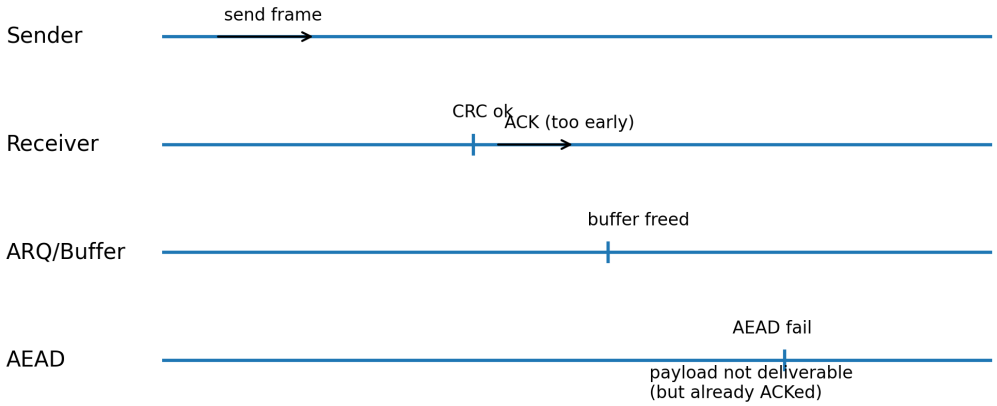

Figure 2: Validity-before-verification race: early ACK/buffer free precedes the AEAD verdict. The gap Δt between CRC_OK and AEAD_FAIL is the vulnerability window in naive stacking.

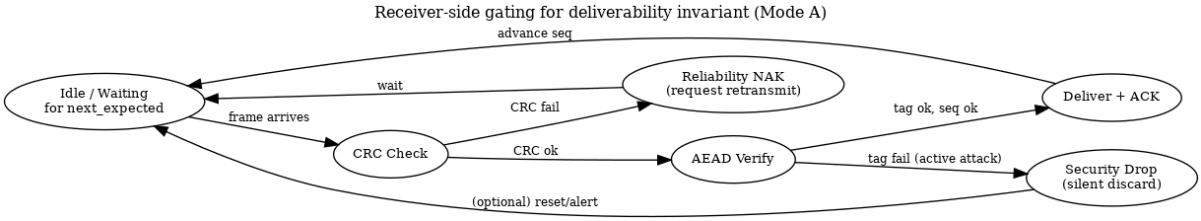

Figure 3: Receiver control separates reliability NAKs (CRC failures) from security drops (AEAD failures without an ACK), thereby avoiding cryptographic oracles and enabling attack detection via telemetry.

2 Literature Review / Background

Chiplet links commonly incorporate CRC, ARQ, and increasingly link security. PCIe and CXL specify Integrity and Data Encryption (IDE)—typically AES-GCM-based AEAD—to provide confidentiality, integrity, and replay protection. However, IDE engines still rely on proper interaction with link-layer sequencing and retransmission: the ShieldLink problem is not a cryptographic primitive but the composition of retransmission state machines with delayed verification.

Prior work motivates these assumptions. Contactless probing demonstrations show that chiplet interposer wiring and I/O drivers can be physically observed with lab equipment, making on-package link confidentiality and integrity relevant even when board-level links are protected [3]. Complementary defenses aim to prevent IP theft and reverse engineering (e.g., dynamic interposer obfuscation) but, by themselves, do not prevent on-path data manipulation [4]. At the system level, root-of-trust proposals for 2.5D integration address supply-chain trust and key provisioning across dies, and can be used to bootstrap ShieldLink security associations [5].

Our threat model overlaps with two well-studied classes: (i) active interposer adversaries that can observe and modify link traffic, and (ii) fault injection attacks (e.g., EMFI) that can transiently corrupt signals. In both cases, a layered CRC/ARQ + AEAD design can mistakenly treat "correct CRC" as "safe to consume". The general TOCTOU hazard is widely documented in software systems; ShieldLink shows an analogous pitfall at the D2D link layer.

2.1 Standards context: where reliability and IDE sit

UCIe defines a die-to-die adapter abstraction that can host reliability mechanisms (CRC, retry buffers, and ARQ) and optional security services. In contrast, CXL/PCIe defines Integrity and Data Encryption (IDE) for board-level links. In practice, these features are often implemented as distinct blocks: a link/transport controller that performs replay buffering and retries, and an IDE engine that performs AEAD-style protection (commonly AES-GCM) with monotonically increasing IV/counter state. The semantic mismatch addressed in this work arises precisely at this boundary: reliability seeks to retire and reuse buffers as early as possible, whereas IDE requires that retirement occur only after authentication completes. ShieldLink should therefore be understood as an adapter-internal contract that ensures CRC/ARQ and IDE are mutually consistent, rather than as a replacement for IDE itself. Our design is compatible with standardized management planes in UCIe (e.g., error counters and link reset hooks) and with IDE threat assumptions in CXL/PCIe. [1], [2], [6]

2.2 Physical and fault-injection threat reality

While cryptographic integrity is often motivated by a "man-in-the-middle" adversary, chiplet packages also admit adversaries that induce structured faults. Recent demonstrations show contactless probing and manipulation of chiplet interposer wiring and I/O die interfaces [3], and broader work emphasizes that 2.5D/3D integration expands the attack surface beyond on-die threats. Even without direct probing, electromagnetic fault injection (EMFI) and related perturbations can create correlated, burst-like errors rather than independent bit flips—an observation that motivates our use of the Gilbert–Elliott (GE) channel model [9], [10], and our discussion of active injection feasibility [12]. ChipletQuake further illustrates that attackers can exploit package-level properties (e.g., interposer impedance) to mount stealthy attacks and to complicate detection, reinforcing the need for link-visible telemetry and reset policies. [11]

2.3 TOCTOU lessons for chiplet links

The core failure mode—acting on a preliminary validity check before a stronger verification completes—is a time-of-check/time-of-use (TOCTOU) pattern. TOCTOU attacks are widely documented in software and firmware systems (e.g., CWE-367) [8] and have also been observed in hardware trust chains (e.g., TOCTOU attacks on device attestation flows) [13]. ShieldLink's "validity-before-verification" vulnerability is a link-layer instance of the same class: a CRC check is necessary for robustness but insufficient for security, and any state transition that depends solely on CRC can be exploitable if the AEAD result is still pending. Positioning the problem this way helps clarify why the proposed invariant is the appropriate abstraction: it elevates "ACK and buffer release" to a use that must be guarded by the strongest check.

2.4 Zero-trust chiplet integration and MPC-style authentication

Recent chiplet security work also explores boot-time and integration-time defenses, including roots of trust for 2.5D systems [5] and mechanisms that treat chiplets as mutually distrustful parties requiring explicit authentication and policy enforcement. Such approaches (including those based on multi-party computation or distributed authentication) primarily address who is allowed to participate and how keys are provisioned, whereas ShieldLink focuses on how the runtime data plane avoids semantic races once keys are in place. These themes are complementary: a zero-trust provisioning path can establish per-link keys and epochs, while ShieldLink ensures that retries and acknowledgments cannot subvert the resulting security guarantees.

2.5 Positioning against IDE/ARQ baselines

Table 1 summarizes the novelty boundary. Existing CRC/ARQ mechanisms provide robustness against noise, and IDE-class AEAD engines provide cryptographic protection, but neither alone specifies the receiver-side contract needed when retransmission state and authentication verdicts are separated by an implementation boundary. ShieldLink contributes to the contract through D-Inv and shows two ways to enforce it.

Table 1: Comparison matrix positioning ShieldLink against reliability-only, IDE-style, and layered baselines.

Approach	Reliability / retry behavior	Authentication behavior	ACK and buffer-retirement semantics
CRC+ARQ only	Retries frames that fail CRC or sequence checks.	No confidentiality or cryptographic integrity.	ACK can retire a CRC-valid frame, but the frame is not security-verified.
IDE / AEAD engine used as a separate block	Depends on surrounding link controller and replay-buffer policy.	Provides confidentiality, integrity, and replay protection after verification completes.	Public standards rarely expose whether ARQ retirement is gated on AEAD; safety depends on integration.
Naive CRC+ARQ + AEAD stacking	Reliability controller may accept and ACK before delayed AEAD verdict.	AEAD can reject after the reliability layer has already advanced.	Illustrates the TOCTOU class: ACK/buffer release is decoupled from authenticated deliverability.
ShieldLink Mode A	Per-frame retry with immediate NAK on CRC failure.	Per-frame AEAD verification.	ACK and NEXT_EXPECTED advance only when CRC_ok, AEAD_ok, and Seq_ok all hold.
ShieldLink Mode B	Epoch retry; all-or-nothing baseline flush on epoch failure.	One tag authenticates M-frame epoch.	ACK is delayed until epoch verification, trading lower overhead for higher buffering and tail latency.

3 Methodology

3.1 Study design and goals

We follow a co-design methodology: define an invariant that captures safe interaction between reliability and security, design protocol mechanisms that enforce it, and evaluate performance and tail behavior under burst errors. The primary correctness objective is D-Inv: ACKs must never outpace authentication.

Deliverability Invariant (D-Inv): For every ShieldLink frame (SLF) s,

$$\text{Deliverable}(s) \equiv \text{CRC\_ok}(s) \wedge \text{AEAD\_ok}(s) \wedge \text{Seq\_ok}(s).$$

Only deliverable SLFs may advance NEXT_EXPECTED and trigger an ACK (buffer retirement); CRC failures trigger a reliability NAK, while AEAD failures trigger a security drop and may contribute to a link-reset threshold.

3.2 Protocol overview

ShieldLink frames (SLFs) carry a fixed 256 B payload, a 16 B link header, and a 4 B CRC; authentication material depends on the operating mode. For comparison, the CRC+ARQ-only baseline in Table 2 assumes a minimum 8-B header (sequence/control) plus a CRC. Mode A appends a per-frame AEAD tag, verifies AEAD before emitting the ACK that advances the sender base, and uses NAKs only for CRC failures. If AEAD fails, the receiver treats it as a security drop (non-recoverable without rekeying or resetting) and requests a link reset/rekey via sideband. Mode B amortizes tags by authenticating M frames with a single tag; the receiver buffers the epoch and appends the tag on the final frame. A CRC failure anywhere in the epoch triggers go-back retransmission of that epoch.

3.3 Simulation model

We implement a discrete-event simulator that models (i) serialization of frames over a link of width $W_{link} = 8$ B/cycle, (ii) CRC delay (1 cycle), (iii) AEAD/epoch verification delay (8 cycles), (iv) propagation delay (2 cycles), and (v) go-back-N ARQ with window $N_{win} = 64$ frames. Errors follow a two-state Gilbert–Elliott channel [9], [10] with good-state corruption probability $p_G = 10^{-6}$, bad-state corruption probability $p_B = 0.1$, and bad-state exit probability $\beta = 0.2$ (mean bad-state duration of five frames). For each bad-state probability π_B, we derive α such that $\pi_B = \alpha/(\alpha+\beta)$. Each point represents the average of five random seeds and reports a 95% confidence interval.

Parameter rationale: p_B is treated as an effective frame-corruption probability in the Bad state rather than a raw bit-error rate. The value $p_B = 0.1$ intentionally stresses the retry/security interaction so that epoch retransmission effects are visible within a bounded simulation. $\beta = 0.2$ corresponds to a mean bad-state duration of $1/\beta = 5$ frames, matching the intended burst-error stress regime; in deployment, these parameters should be calibrated from link telemetry rather than treated as universal constants.

3.4 Metrics

We report normalized goodput (the delivered payload rate divided by the raw link capacity) and p99 latency to assess deliverability. For per-frame protocols (Mode A and Naive), latency is measured from the start of the first transmission to the time the frame becomes deliverable. For Mode B, the deliverable unit is an epoch; we report the p99 epoch-head latency (from the start of the first-frame transmission to epoch commit).

3.5 Bounded safety exploration

To avoid overclaiming formal verification, we treat our protocol proof effort as a bounded safety sanity check, not as a complete formal proof. In addition to the TLA+-style sketch summarized in Appendix D, we implemented a small, explicit-state explorer in Python that models a single stream with go-back-N retransmission, CRC/AEAD outcomes, and duplicate-delivery handling. For a bounded configuration (sequence numbers modulo 8, ARQ window $N_{win} = 4$, exploration depth 14 steps), the included `formal_sanity_check.py` explorer enumerates 845,625 reachable states and finds no counterexamples to the core safety properties: (P1) the receiver never emits ACK unless $\mathrm{CRC\_ok} \wedge \mathrm{AEAD\_ok} \wedge \mathrm{Seq\_ok}$ holds, and (P2) receiver state advances only on deliverable frames. These checks do not prove liveness under all adversarial schedules, but they increase confidence that the FSM and its ACK-gating logic are internally consistent. [14]

3.6 Preliminary RTL structural cost analysis (FPGA/synthesis-friendly control plane)

To address concerns about implementation realism without requiring a full PHY or end-to-end RTL prototype, we developed a small, synthesizable RTL skeleton of the ShieldLink control plane (Modes A and B). The goal is to make the hardware impact of the Deliverability Invariant explicit in terms of (i) state bits, (ii) buffer bits, and (iii) incremental combinational logic depth. We do not claim measured PPA in this section; instead, we report exact memory sizing and a conservative gate-equivalent estimate for the incremental D-Inv gating.

The RTL follows the paper's microarchitecture: CRC and AEAD blocks produce registered one-bit verdicts (crc_ok, aead_ok). The ACK/NAK FSM consumes these registered verdicts, so D-Inv adds only a constant-depth gate on the control path (ack_en = crc_ok ∧ aead_ok ∧ seq_eq). In this organization, the datapath critical path is dominated by the AEAD pipeline and the PHY boundary rather than by the ACK gate itself.

The synthesizable SystemVerilog RTL skeleton, resource-sizing script, and updated simulation generator are provided as supplementary material (Appendix C). They enable independent synthesis experiments, but any LUT/FF/BRAM, f_{max}, area, or power numbers for a specific FPGA or ASIC target should be interpreted as future implementation results rather than claims made by this paper.

4 Results

4.1 Wire efficiency and buffering

Table 2: Overhead and buffering by scheme (payload = 256 B, W_{link} = 8 B/cycle).

Scheme	Bytes/frame	Cycles/frame	Wire efficiency	Verification	RX buffering
Reliability-only (CRC+ARQ)	268	34	0.955	per-frame CRC	≈1 frame
Naive Stacking (secure)	296	37	0.865	per-frame AEAD	≈1 frame
ShieldLink Mode A	288	36	0.889	per-frame AEAD	≈1 frame
ShieldLink Mode B ($M = 32$)	276+tag/32	≈35.0	0.926	per-epoch AEAD	≈16 KiB (2 epochs)

Mode B reduces per-frame overhead by amortizing tags, improving wire efficiency. Figure 4 moves the epoch-size sensitivity plot into the main results to make this amortization explicit. However, Mode B intentionally delays delivery until an epoch is verified, thereby increasing tail latency and making the design more sensitive to bursts.

4.2 Tail latency and goodput under burst errors

In this configuration, Mode B outperforms Mode A at low burst probability due to lower per-frame overhead. Still, once π_B reaches roughly 0.04 in the default $\beta = 0.2$ stress regime, epoch flushes dominate and goodput falls below Mode A. Mode A preserves low tail latency by allowing the ARQ machine to retry immediately on CRC failure while never advancing the ACK beyond verified data.

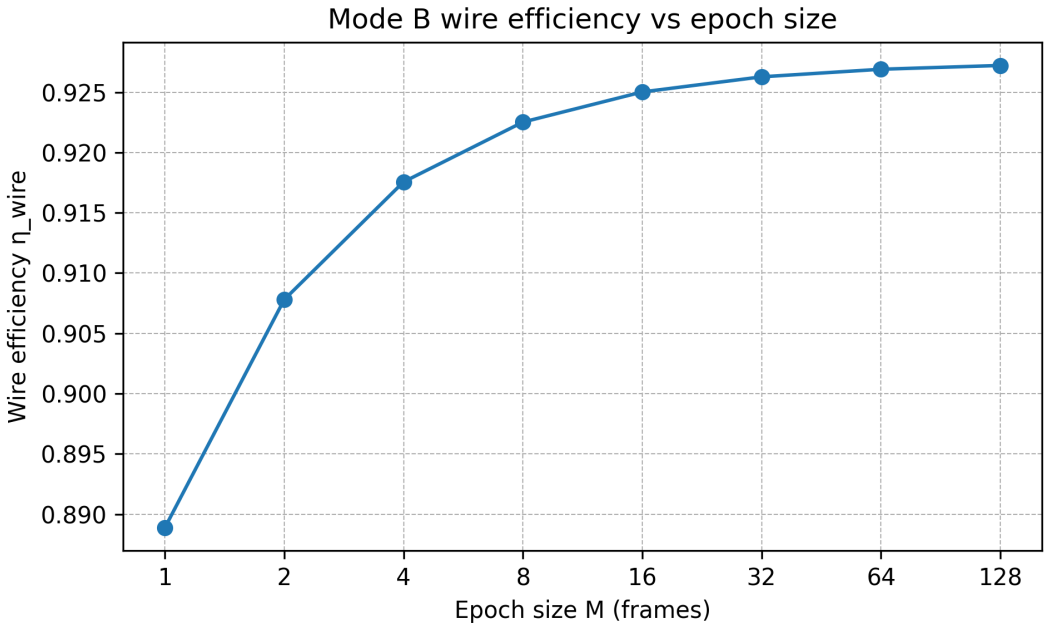

Figure 4: Mode B wire efficiency vs epoch size M (analytic). Larger epochs amortize the tag more effectively but increase buffering and commit granularity.

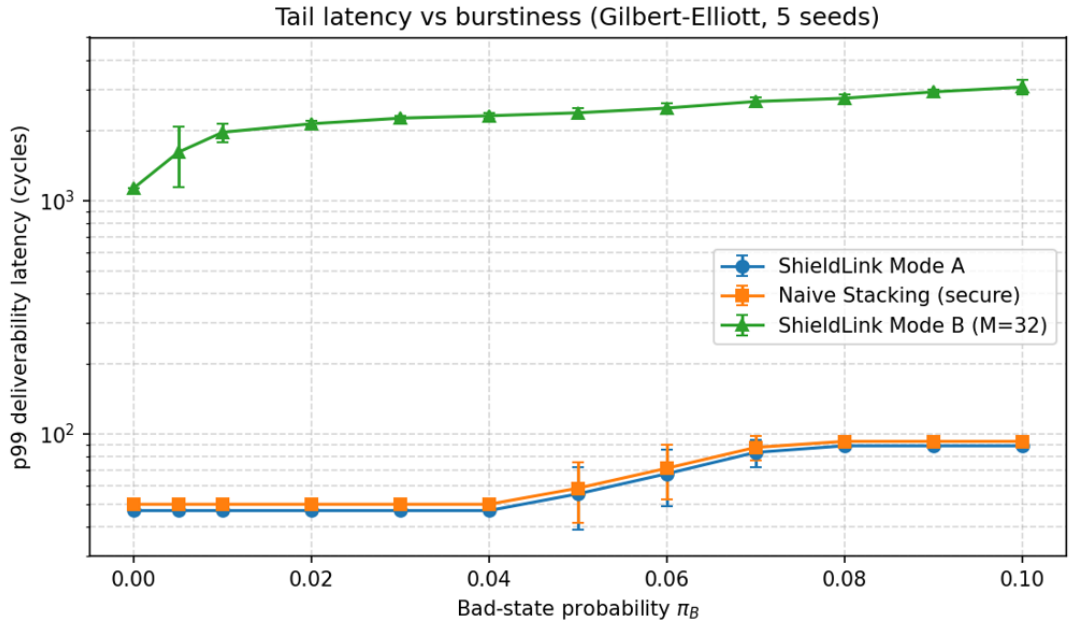

Figure 5: p99 deliverability latency vs bad-state probability π_B, updated with five random seeds. Mode B shows an abrupt tail increase due to epoch retransmissions.

4.3 Sensitivity and limitations

Appendix B reports a β sweep showing that the Mode B crossover depends on burst dynamics. Under a fixed, stationary bad-state probability π_B, larger β (shorter but more frequent bursts) tends to shift the crossover to smaller π_B because more bursts touch distinct epochs. Our simulator is intentionally minimal; it does not model higher-layer traffic patterns, adaptive PHY equalization, or selective repeat ARQ. Future work should include trace-driven evaluation.

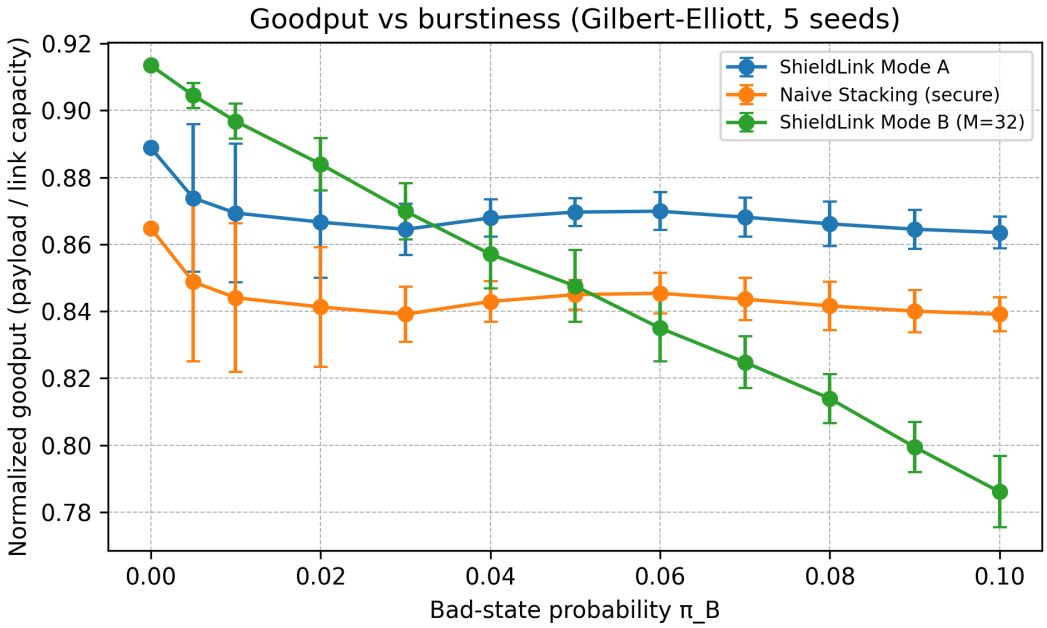

Figure 6: Normalized goodput vs bad-state probability π_B, updated with five random seeds. Mode B crosses below Mode A near $\pi_B \approx 0.04$ for this stress regime.

Table 3: Representative discrete-event simulation points ($\beta = 0.2$, $p_B = 0.1$), reported as mean ±95% CI across five random seeds.

π_B	Scheme	Goodput (mean ±95% CI)	p99 commit latency (cycles; mean ±95% CI)
0.01	Mode A	0.869 ± 0.021	47 ± 0
0.01	Naive stacking (secure)	0.844 ± 0.022	50 ± 0
0.01	Mode B ($M = 32$)	0.897 ± 0.005	1962 ± 182
0.05	Mode A	0.870 ± 0.004	55 ± 16
0.05	Naive stacking (secure)	0.845 ± 0.004	59 ± 17
0.05	Mode B ($M = 32$)	0.848 ± 0.011	2376 ± 112

4.4 Sender buffer pressure from ACK gating

A reviewer might ask whether delaying ACK until after AEAD verification increases sender replay-buffer pressure or requires a larger ARQ window to keep the link saturated. This effect is real but can be quantified with a bandwidth-delay product (BDP) argument: for a link that transmits one SLF in t_{tx} cycles and returns an ACK after RTT cycles, the minimum window to avoid transmitter idle time is $N_{win} \geq \lceil RTT/t_{tx} \rceil$. In Mode A, RTT includes serialization, two-way propagation, CRC check, AEAD verification, and a slight control delay. With our nominal parameters (256-byte payload, 8 B/cycle serialization, $d_{prop} = 2$ cycles, $d_{crc} = 1$ cycle, $d_{AEAD} = 8$ cycles), the required window remains small (Table 4).

This simple calculation explains why Mode A's ACK gating does not automatically imply large replay buffers: the AEAD verification delay is slight compared to serialization time at typical flit sizes. Mode B differs in construction—buffering is the price of epoch amortization—so it is best suited to traffic classes that can tolerate bulk commit latency.

Table 4: ACK-gating and minimum ARQ window size needed to sustain continuous transmission (nominal parameters).

Scheme	t_{tx} (cycles/SLF)	RTT to free first buffer (cycles)	$N_{win,min} \approx \lceil RTT/t_{tx} \rceil$	Notes
Mode A	36	50	2	ACK after per-SLF AEAD; window ≈2 frames is sufficient.
Naive stacking (secure)	37	52	2	ACK after per-SLF AEAD through an IP boundary; slightly larger wire time due to redundant header/metadata.
Mode B ($M = 32$)	36	1166	33	ACK amortized over an epoch; requires buffering $\gtrsim M$ frames to avoid mid-epoch stalls.

4.5 RTL/FPGA resource sizing (preliminary)

Table 5 reports the exact buffer sizes implied by the reference parameters used throughout this manuscript (payload $P = 256$ B, header=16 B, CRC=4 B, tag=12 B, $N_{win} = 64$, Mode B epoch $M = 32$). For FPGA intuition, we also map these memory bits to common block RAM granularities (18 KB and 36 KB). Table 6 complements this by summarizing control-plane FF/LUT structural estimates for the prototype RTL. These are resource-sizing estimates only, not vendor synthesis, post-place-and-route timing, power, or f_{max} results.

Table 5: RTL/FPGA buffer sizing and BRAM mapping (reference parameters).

Buffer	Total (KiB)	Total (bits)	18Kb BRAM blocks	36Kb BRAM blocks
Mode A TX retry buffer (N_{win})	18.000	147456	8	4
Mode B RX epoch buffer (M)	8.625	70656	4	2
Incremental storage for tags (Mode A vs. no-tag)	0.750	6144	1	1

Table 6: RTL control-plane resource estimate (logic only; excluding cipher core and buffers).

Block (prototype RTL)	FF (bits)	LUT-eq (rough)	Notes
Baseline CRC+ARQ control (unsafe early-ACK if paired with later AEAD)	195	≈200	ACK/advance on CRC+Seq only
ShieldLink Mode A control (D-Inv gated)	196	≈200	Δ ≈+1 FF, +1 AND vs baseline
ShieldLink Mode B control (epoch bookkeeping)	299	≈250–300	Adder+compare+bitmap reduction; flush-all baseline

LUT-eq counts are conservative RTL-structure estimates, not vendor-synthesis or post-place-and-route measurements. They are dominated by 64-bit add/compare logic that already exists in reliability-only ARQ; the incremental D-Inv gating is constant-depth and should not materially affect the control-plane critical path. Full timing closure and power validation remain future work for an integrated adapter/PHY implementation.

Figure 7 depicts the buffering cost for Modes A and B under the same reference parameters.

As a conservative logic-depth estimate, a 64-bit equality comparator (Seq==NextExpected)

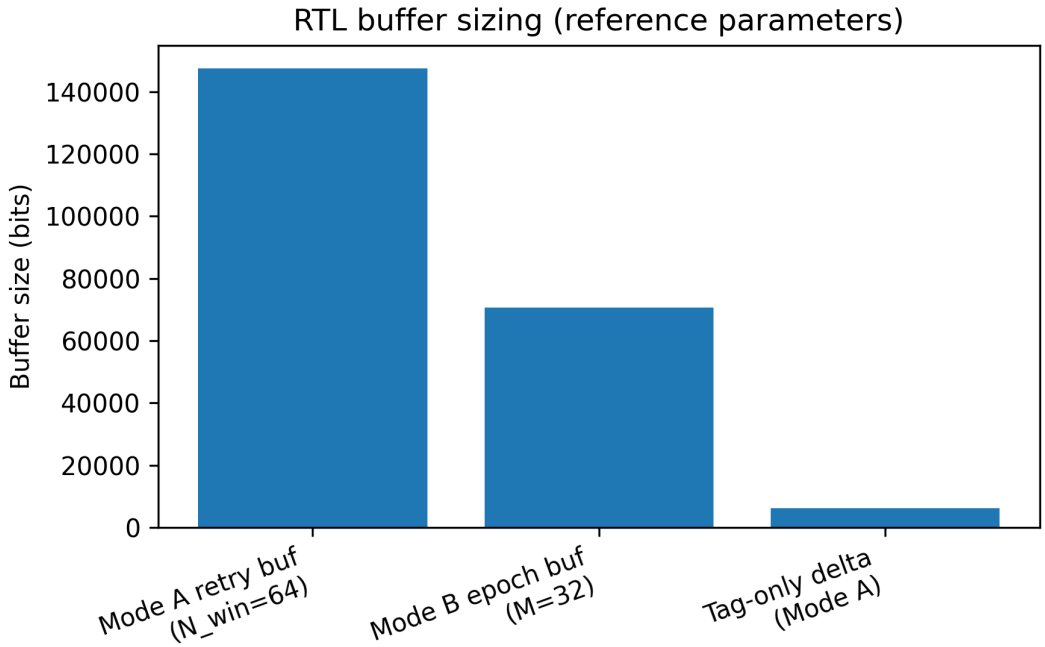

Figure 7: RTL buffer sizing for Mode A TX retry buffering and Mode B RX epoch buffering (reference parameters).

can be decomposed into 64 XNORs plus an AND-reduction tree (63 ANDs), i.e., ≈ 319 two-input gate-equivalents (assuming XNOR$\approx$ 4 2GE). The incremental D-Inv gating adds ≈ 2 2GE (gating ACK/state advance on aead_ok), which is $< 1\%$ of the comparator's cost. This supports the claim that D-Inv logic is negligible relative to the cipher core and buffering.

5 Discussion

ShieldLink's central design choice is to treat authentication as part of reliability: ACK is a promise that the corresponding payload is safe to deliver. This shifts security failures into an explicit state (a security drop) rather than allowing them to masquerade as progress in reliability. In deployment, a security drop would typically trigger a link reset and a key refresh, as many link encryption engines respond to integrity failures.

Mode selection can be adaptive: Mode B is appropriate for bulk transfers over benign channels, whereas Mode A is preferable when bursts are common or when latency is critical. Future work includes selective retries within epochs (bitmap NAK), dynamic epoch sizing, and combining ShieldLink with lightweight FEC to reduce CRC failures before ARQ is engaged.

5.1 Mitigation costs: AEAD delay, buffering, and timing closure

A recurring concern is whether enforcing the Deliverability Invariant (D-Inv) meaningfully complicates timing closure. ShieldLink's datapath does not place CRC and AEAD on the same critical combinational arc; CRC checking and AEAD verification are naturally pipelined and are already treated as multi-cycle operations in many link designs. The primary architectural cost of D-Inv is not additional cryptographic logic but the requirement to retain replay-buffer state until the AEAD result is known. For per-SLF authentication (Mode A and the secure naive baseline), Table 4 shows that the additional holding time yields only a slight increase in the bandwidth-delay product under our nominal parameters; consequently, a modest ARQ window

continues to saturate the link. The Mode B case is qualitatively different because it intentionally defers commitment until epoch authentication completes; the required buffering is therefore proportional to the epoch size M and should be considered an explicit design knob rather than a hidden overhead.

In addition to motivating the issue with a concrete race, the work offers a link-layer invariant that is easy to audit: an implementation is correct if and only if the receiver emits an ACK exactly when an SLF is deliverable (CRC_ok $\wedge$ AEAD_ok $\wedge$ sequence_valid). The two operating modes indicate that this invariant does not constrain a single point in the latency–efficiency design space: Mode A prioritizes tail latency for coherence-like traffic, whereas Mode B amortizes per-frame overhead for bulk transfers but exhibits a crossover region under burst errors. Future work should (i) implement selective retry within Mode B epochs, (ii) prototype the integrated adapter on FPGA/RTL to quantify f_{max}, area, and power under realistic PHY timing, and (iii) extend the evaluation to trace-driven CXL/UCIe traffic mixes and multi-hop topologies. These steps would turn the present design into a deployable template for secure, reliable chiplet interconnects.

Implementation note: Table 5 and Figure 7 show that buffering dominates the incremental cost, not the D-Inv gate. For the reference parameters, Mode B's additional epoch buffer is ≈ 8.6 KiB per protected stream, while Mode A's tag storage adds only 0.75 KiB over a no-tag retry buffer at $N_{win} = 64$. Because D-Inv consumes registered CRC/AEAD verdicts, it can be placed off the datapath critical path; this is an architectural expectation, not a measured f_{max} claim.

5.2 Is "naive stacking" a strawman?

It is fair to ask whether "naive stacking" actually occurs in practice. We do not claim that any particular commercial CXL/PCIe controller releases replay buffers on a CRC pass before IDE verification; public documentation rarely specifies such microarchitectural details. Instead, the baseline represents a class of unsafe integration contracts: reliability and IDE are implemented as separable blocks with an interface boundary that decouples "frame accepted by the link" from "frame authenticated." In this sense, ShieldLink is both a concrete design and an audit checklist. If an implementation cannot demonstrate that ACK implies CRC_ok $\wedge$ AEAD_ok $\wedge$ sequence_valid, then it is vulnerable to the semantic mismatch regardless of the underlying cipher. [8], [13]

5.3 Security drops, link reset, and higher-layer recovery

Silent drops on AEAD failure are controversial because they trade integrity-oracle resistance for availability: an active attacker can force timeouts and retries. ShieldLink adopts a conservative stance: an explicit "authentication failed" NAK would create a decryption oracle, whereas a silent drop preserves D-Inv. To make the resulting availability behavior operationally acceptable, the receiver must treat repeated authentication failures as a detectable anomaly and escalate. One practical policy is a link-reset threshold: after N_{auth} consecutive AEAD failures (or after an AEAD_fail-to-CRC_fail ratio exceeds a threshold), the receiver tears down the link, rekeys, and reports an error via sideband telemetry. At the system level, higher layers can then apply standard recovery actions—reroute to a redundant path, fail over to a spare chiplet, or mark the traffic class as degraded—analogous in spirit to the "poison/viral" reliability signaling used in CXL ecosystems. [2]

5.4 Mode B rigidity and a selective-retry path

The evaluation intentionally models Mode B with an all-or-nothing epoch-containment policy: if any SLF within an epoch fails CRC or AEAD, the receiver discards the buffered epoch, and the sender retransmits it. This flush policy is not presented as the only engineering choice;

rather, it is a clean baseline that exposes the head-of-line (HoL) mechanism behind Mode B's "goodput cliff." A practical optimization is selective retry within an epoch, for example, by attaching a compact bitmap to a NAK that identifies which SLFs in the epoch failed CRC checks, while maintaining a single epoch-level authentication tag. Such a scheme would preserve most of Mode B's header amortization while reducing retransmission volume during bursts, at the cost of additional receiver bookkeeping and sender-scheduling complexity. We treat this as low-hanging-fruit work because it can be layered onto the same D-Inv foundation: ACK remains gated on deliverability, but the retransmission granularity becomes finer.

5.5 Beyond single links: calibration and multi-hop extensions

Our simulation models a single D2D link; multi-link fabrics and multi-hop chiplet topologies introduce additional failure modes. First, burst parameters (α, β, and p_B) should be calibrated to the physical environment. Rather than guessing, a deployed system can estimate these parameters from CRC/NAK counters and recovery events observed during runtime, and then select Mode A or Mode B (or adjust M) to remain on the favorable side of the crossover region. Second, active fault-injection techniques, such as EMFI, can produce spatially and temporally correlated errors across adjacent links, thereby increasing the value of link-local telemetry when multiple links share an interposer or I/O die. Finally, extending D-Inv across multi-hop paths suggests a compositional view: each hop enforces its own deliverability invariant, while higher layers maintain end-to-end ordering and key epochs. Trace-driven CXL/UCIe traffic evaluation is an important next step for generalizability. [9]–[12]

6 Conclusion

Naively composing link-layer ARQ with delayed AEAD verification can create a validity-before-verification race: the reliability layer can commit data before cryptographic integrity is established. ShieldLink enforces D-Inv by gating acknowledgments and buffer retirement on AEAD verification (Mode A) or epoch verification (Mode B), cleanly separating recoverable link noise from non-recoverable security violations.

More broadly, ShieldLink makes an implicit contract explicit: on retrying links, delivery semantics and security semantics must align. Our analytical and simulation results quantify the wire overhead/latency trade-off and guide decisions about whether per-frame authentication (Mode A) or amortized authentication (Mode B) is preferable. Future work includes end-to-end FPGA validation for timing closure and power, and the exploration of tighter integration with standard IDE-class crypto backends.

References

[1] UCIe Consortium, "UCIe Consortium Introduces 3.0 Specification With 64 GT/s Performance and Enhanced Manageability," *Business Wire*, Aug. 5, 2025. [Online]. Available: `https://www.businesswire.com/news/home/20250805909613/en/UCIe-Consortium-Introduces-3.0-Specification-With-64-GTs-Performance-and-Enhanced-Manageability`. Accessed: Dec. 14, 2025.

[2] CXL Consortium, "CXL Consortium Releases the Compute Express Link 4.0 Specification Increasing Speed and Bandwidth," Nov. 18, 2025. [Online]. Available: `https://computeexpresslink.org/wp-content/uploads/2025/11/CXL_4.0-Specification-Release_FINAL_Website-Copy.pdf`. Accessed: Dec. 14, 2025.

[3] A. Deric, K. Mitard, S. Tajik, and D. Holcomb, "Evaluating Vulnerability of Chiplet-Based Systems to Contactless Probing Techniques," arXiv:2405.14821, 2024.

doi:10.48550/arXiv.2405.14821.

[4] J. Talukdar, A. Chaudhuri, J. Kim, S. K. Lim, and K. Chakrabarty, "Securing Heterogeneous 2.5D ICs Against IP Theft Through Dynamic Interposer Obfuscation," in Proc. DATE, 2023. doi:10.23919/DATE56975.2023.10137145.

[5] M. Nabeel, M. Ashraf, S. Patnaik, V. Soteriou, O. Sinanoglu, and J. Knechtel, "2.5D Root of Trust: Secure System-Level Integration of Untrusted Chiplets," *IEEE Trans. Comput.*, vol. 69, no. 11, pp. 1611–1625, Nov. 2020, doi: 10.1109/TC.2020.3020777.

[6] M. Dworkin, "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC," NIST SP 800-38D, Nov. 2007. [Online]. Available: `https://csrc.nist.gov/pubs/sp/800/38/d/final`. Accessed: Dec. 14, 2025.

[7] M. S. Turan et al., "Ascon-Based Lightweight Cryptography Standards for Constrained Devices: Authenticated Encryption, Hash, and Extendable Output Functions," NIST SP 800-232, Aug. 2025. doi:10.6028/NIST.SP.800-232. [Online]. Available: `https://csrc.nist.gov/pubs/sp/800/232/final`. Accessed: Dec. 14, 2025.

[8] MITRE, "CWE-367: Time-of-check Time-of-use (TOCTOU) Race Condition," 2024. [Online]. Available: `https://cwe.mitre.org/data/definitions/367.html`. Accessed: Dec. 14, 2025.

[9] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *Bell Syst. Tech. J.*, vol. 39, no. 5, pp. 1253–1265, 1960.

[10] E. O. Elliott, "Estimates of Error Rates for Codes on Burst-Noise Channels," *Bell Syst. Tech. J.*, vol. 42, no. 5, pp. 1977–1997, 1963.

[11] S. Khalaj Monfared, M. Saadat Safa, and S. Tajik, "ChipletQuake: On-Die Digital Impedance Sensing for Chiplet and Interposer Verification," *Sensors*, vol. 25, no. 15, Art. no. 4861, Aug. 2025, doi: 10.3390/s25154861.

[12] N. Mishra, A. Chakraborty, and D. Mukhopadhyay, "Faults in Our Bus: Novel Bus Fault Attack to Break ARM TrustZone," in Proc. Network and Distributed System Security Symposium (NDSS), 2024, doi: 10.14722/ndss.2024.24499.

[13] S. Hristozov, M. Wettermann, and M. Huber, "A TOCTOU Attack on DICE Attestation," in Proc. ACM Conference on Data and Application Security and Privacy (CODASPY), 2022, doi: 10.1145/3508398.3511507.

[14] L. Lamport, *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*. Addison-Wesley, 2002.

Appendix

A Frame format and nonce construction

Mode A SLF = [Header (16 B) | Payload (256 B) | CRC (4 B) | Tag (12 B)]. We use a 96-bit truncated tag to reduce bandwidth overhead; the same framing also supports a full 128-bit tag. Mode B SLF = [Header | Payload | CRC] for frames 0 to $M-2$, with a 12 B epoch tag appended to frame $M-1$. Nonces are derived from (link_id, direction, sequence_number). Retransmissions send identical encrypted bits without re-encryption, so the nonce remains unchanged across retries.

B Additional plots and sensitivity

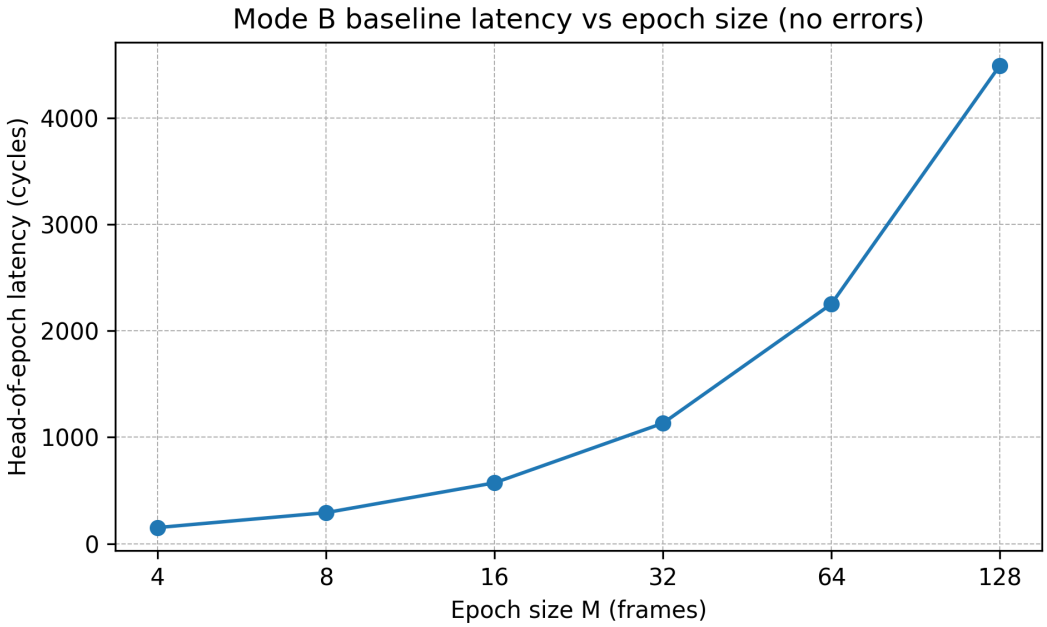

Figure 8: Mode B baseline head-of-epoch latency grows roughly linearly with M (no errors).

Table 7: Goodput crossover points where Mode B becomes worse than Mode A (targeted five-seed coarse sweep).

β	Mean bad burst length (frames)	Crossover π_B
0.05	20.0	0.055
0.10	10.0	0.050
0.20	5.0	0.040
0.50	2.0	0.025

C Reproducibility

Supplementary materials accompanying this revised submission include (i) `generate_assets.py`, which regenerates the Gilbert–Elliott simulation sweeps, Figures 4–6, Appendix B plots, and the coarse Mode B crossover table; (ii) the revised five-seed simulation CSV files; (iii) `resource_estimator.py` and RTL sizing data; (iv) `formal_sanity_check.py` for bounded safety exploration; and (v) the prototype SystemVerilog RTL skeleton of the ShieldLink control plane. The scripts are deterministic given their random seeds.

Supplementary RTL: The accompanying RTL package provides a synthesizable SystemVerilog skeleton for the sender/receiver controllers, the retry buffer, and baseline CRC+ARQ logic, along with the Mode A ACK-gating modifications that enforce D-Inv. The package is intended for independent reproduction and extension, not as a final PPA/timing sign-off artifact.

D Bounded protocol sanity check

This appendix summarizes the bounded explicit-state exploration described in Section 3.5. The goal is to sanity-check safety (no ACK without deliverability), not to prove liveness, timing

closure, or resistance to denial-of-service under all schedules.

D.1 Safety properties checked

We check two core properties over all explored interleavings:

- P1 (ACK gating): The receiver emits ACK only if CRC_ok ∧ AEAD_ok ∧ Seq_ok.
- P2 (monotonic delivery): The receiver advances next_expected only when it delivers a frame.

Here, Seq_ok means either (i) the frame's sequence equals next_expected (new in-order delivery) or (ii) the frame is a duplicate of an already-delivered sequence number (duplicate ACK without re-delivery).

D.2 Explorer model (high-level)

The explorer models a single stream with a go-back-N sender, a FIFO channel, and a receiver that implements ShieldLink's CRC/AEAD decision structure. At each step, it nondeterministically chooses among the following: the sender transmits, the receiver consumes a data frame, and the sender consumes an ACK/NAK. CRC_ok and AEAD_ok are treated as nondeterministic outcomes per transmission, capturing both noise and active manipulation. The included script reports 845,625 reachable states for the stated bounded configuration and no safety counterexample. The reference implementation is provided as `code/formal_sanity_check.py` in the supplementary bundle.

Funding Statement

No funding received.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Ethical Considerations

The article does not contain any studies with human participants or animals performed by the author.

AI Usage Statement

AI tools were used in the following way(s): Grammarly for language editing; Figures: Anthropic Claude Opus 4.7; OpenAI Codex 5.5 Medium for GitHub README files and push & commit.