EdgeTrust-Offload: Quantifying Zero-Trust Overhead for Secure Task Offloading in Real-Time Edge Sensor Networks

Wanghley Soares Martins, Yifei Sun
ECE 654: Edge Computing

February 7, 2026

THE CORE IDEA

Continuous physiological monitoring (ICU telemetry to wearable temperature tracking) creates time-series streams that must be processed under tight latency budgets. Edge computing can meet these deadlines, but most offloading assumes *implicit trust within the local network*. For PHI under HIPAA [17] or sensor data under GDPR, that assumption fails.

This project builds an edge-trust offload strategy (Fig. 1) that wraps each IoT-to-edge transaction in a Zero-Trust Architecture (ZTA) envelope and *measures the cost*. An **ESP32-S3** runs two workloads: (1) a 1024-point FFT for ICU waveform analysis [14]; and (2) a 64-tap FIR filter for temperature smoothing [15]. Each task is executed locally or offloaded through a secure tunnel to an NVIDIA Jetson Nano worker.

The system comprises three tightly coupled components:

1. **Security Layer.** Tailscale/WireGuard [10] enforces mutual authentication (mTLS) and encrypted tunnels between node pairs.
2. **Continuous Sensing Engine.** ESP32-S3 pipelines: (i) 256 Hz ECG samples for FFT analysis and (ii) 1 Hz DS18B20 readings for FIR smoothing.
3. **Adaptive Scheduler.** A Python decision engine evaluates

$$C = \alpha \cdot T_{\text{exec}} + \beta \cdot T_{\text{net}} + \gamma \cdot E_{\text{crypto}} + \delta \cdot P_{\text{loss}}$$

where T_{exec} is compute time, T_{net} is round-trip latency, E_{crypto} is crypto overhead, and P_{loss} is packet-loss rate. The scheduler chooses local execution when $C_{\text{offload}} > C_{\text{local}}$.

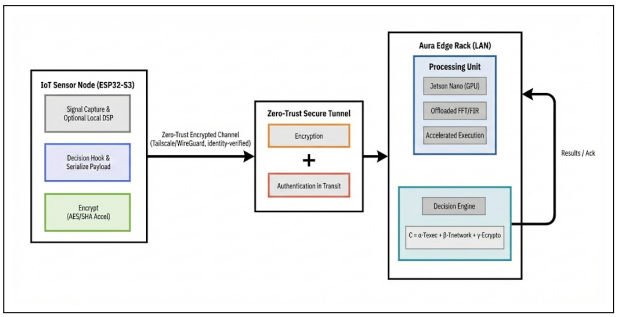

Figure 1: EdgeTrust-Offload System Architecture

The core goal is the **crossover point** (L^*): the maximum network latency where secure offloading remains faster than local ESP32 execution. We quantify crypto overhead and determine when offloading to the Jetson Nano still makes sense. No prior work measures this threshold for simple IoT devices processing continuous health sensor data.

THE FOUR WHYS

Why this? Offloading work optimizes latency and energy but ignores the "security tax" [2]. For HIPAA/FERPA deployments this is risky. We quantify when ZTA overhead exceeds gains and deliver rules: *offload iff* $L < L^*$ [1], [9].

Why now? NIST SP 800-207 formalized ZTA [1]; MCU crypto accelerators have matured [3]; and Duke Health/Smart Home are exploring privacy-sensitive edge sensing [4], [13]. The question is *when* ZTA is practical.

Why me? We have PCB/ESP32 experience, real-time sensing background, and the Aura rack testbed. Tailscale is already deployed and WireGuard throughput validated, reducing ramp-up risk, with a strong biomedical focus.

Why you? ECE 654 centers on low-latency, heterogeneous edge architectures—precisely the context where security overhead is most consequential. The crossover-point analysis directly answers the course's motivating question: *how do we architect edge systems that balance latency, security, and resource efficiency?* The findings and open-source code will be contributed back to Duke's ECE infrastructure for future student projects.

RELATED WORK

Zero-Trust Architecture. NIST SP 800-207 [1] defines ZTA, with Rose et al. [5] and Kindervag [6] extending the model. These frameworks focus on enterprise settings; none benchmark ZTA on microcontroller-class edge nodes.

Computation Offloading. Mao et al. [2] survey mobile edge offloading, Kumar et al. [7] formalize offload-vs-local, and Wang et al. [8] address DL inference. Lin et al. [9] study IoMT offloading. None include cryptographic overhead in the decision model.

Continuous Health Sensing at the Edge. Clifford et al. [14] show ICU alarms benefit from edge-local spectral analysis; Tamura et al. [15] highlight real-time temperature filtering; Dunn et al. [13] show wearable signals require low-latency, privacy-preserving pipelines.

Secure Edge Communication. Donenfeld [10] introduces WireGuard; Tailscale builds a user-space mesh with identity-based access; Cloudflare's Magic WAN [11] scales secure edge connectivity; and NVIDIA's TensorRT [12] accelerates inference.

Gap. Security and offloading are treated as orthogonal. We treat cryptographic cost as a *first-class scheduling constraint* alongside latency and compute time on representative sensor workloads.

METHODOLOGY & VALIDATION

Hardware Testbed: The Aura Rack

All experiments run on a physical 10-inch 9U rack ("Aura"):

- **Worker Nodes:** NVIDIA Jetson Nano (128 CUDA cores, 4 GB), Raspberry Pi 4 (4 GB), Orange Pi 4A
- **Sensor Node:** ESP32-S3 (T-Display S3) with AES-256/SHA-256 accelerators, DS18B20 probe, synthetic 256 Hz ECG via DAC
- **Network:** TP-Link TL-WR1502X Wi-Fi 6 router, TL-SG108 gigabit switch
- **Security:** Tailscale mesh (WireGuard tunnels, identity-based ACLs)

Workloads

1. **ICU Waveform Analysis (FFT).** A 1024-point single-precision FFT on a 256 Hz ECG buffer, representing spectral feature extraction for arrhythmia detection [14]. Executed via ESP-DSP locally or NumPy/CuPy on the Jetson.
2. **Temperature Smoothing (FIR).** A 64-tap low-pass FIR filter on a 1 Hz DS18B20 stream, modeling real-time thermal drift compensation for wearable health devices [15]. Lightweight enough to stress-test the overhead floor of ZTA.

Experimental Configurations

We isolate each variable with four scenarios per workload:

1. **Local Baseline.** Task executes on ESP32-S3. Measures T_{local}, E_{local}.
2. **Insecure Offload.** Raw TCP to Jetson (no encryption). Establishes $T_{\text{net}}^{\text{insecure}}$.
3. **Secure Offload (ZTA).** Offload through Tailscale/WireGuard. Measures $T_{\text{net}}^{\text{secure}}$ including encrypt/decrypt.
4. **Secure + Congested.** ZTA offload with Linux `tc`: 10–100 ms jitter, 1–10% packet loss.

Each configuration runs **500 trials**; we report median, 95th-percentile, and max latency.

Metrics

- **End-to-End Latency:** $T_{e2e} = T_{\text{ser}} + T_{\text{enc}} + T_{\text{tx}} + T_{\text{dec}} + T_{\text{comp}} + T_{\text{ret}}$
- **Security Overhead Ratio:** $R_{\text{sec}} = T_{e2e}^{\text{secure}} \,/\, T_{e2e}^{\text{insecure}}$
- **Crossover Point:** L^* where $T_{e2e}^{\text{secure}}(L^*) = T_{\text{local}}$
- **Throughput:** Tasks per second under sustained load
- **Energy per Task:** INA219 measurements on ESP32-S3 rail

PROJECT PLAN & TIMELINE

Week	Deliverable	Milestone
1–2 (Feb 7–21)	Literature review; Tailscale deployment; sensor wiring	Proposal
3–4 (Feb 22–Mar 7)	ESP32-S3 FFT & FIR implementations; local baselines	Development
5–6 (Mar 8–21)	Offload protocol (TCP & WireGuard); insecure vs. secure comparison	Progress Report
7–8 (Mar 22–Apr 4)	Scheduler decision engine; congestion injection experiments	Testing
9–10 (Apr 5–18)	Analysis; crossover-point plots; final paper & open-source release	Final Report

RISK MANAGEMENT

- **Risk:** ESP32-S3 SRAM (512 KB) limits encryption buffers for 1024-point FFT payloads.

  Mitigation: Segment into 256-sample chunks; leverage AES hardware; fall back to FIR-only if needed.

- **Risk:** Network jitter >50 ms causes TCP retransmissions inflating variance.

  Mitigation: Use UDP with sequence numbers and retransmit logic; report median over 500 trials; characterize jitter distribution.

- **Risk:** Tailscale coordination-server dependency introduces an external failure mode.

  Mitigation: Deploy Headscale on the Aura rack; keep direct WireGuard fallback.

- **Risk:** Jetson Nano thermal throttling under sustained crypto + compute load.

  Mitigation: Monitor `tegrastats`; add a 40 mm fan; report throttled vs. unthrottled throughput.

- **Risk:** Synthetic ECG lacks clinical fidelity, limiting generalizability.

  Mitigation: Use MIT-BIH Arrhythmia Database waveforms [16] via ESP32 DAC; validate spectral content.

DUKE COMMUNITY IMPACT

This work connects to two Duke initiatives. The **Duke Smart Home Program** deploys residential sensors; our crossover-point models guide when ZTA overhead is acceptable for FERPA-protected data. **Duke Health** is exploring edge processing of bedside telemetry; our empirical L^* thresholds provide device-class guidance [13].

All code (scheduler, firmware, analysis notebooks) will be released on the Duke GitHub organization. The Aura rack will remain available for future ECE 654 cohorts, and the latency dataset will serve as a benchmark for secure-offloading research.

References

[1] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," National Institute of Standards and Technology, Tech. Rep. NIST SP 800-207, 2020. DOI: 10.6028/NIST.SP.800-207.

[2] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017. DOI: 10.1109/COMST.2017.2745201.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," in *IEEE Internet of Things Journal*, vol. 3, 2016, pp. 637–646. DOI: 10.1109/JIOT.2016.2579198.

[4] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017. DOI: 10.1109/MC.2017.9.

[5] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," *NIST Special Publication*, vol. 800, p. 207, 2020.

[6] J. Kindervag, "No more chewy centers: Introducing the zero trust model of information security," *Forrester Research*, 2010.

[7] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," in *Mobile Networks and Applications*, vol. 18, 2013, pp. 129–140. DOI: 10.1007/s11036-012-0368-0.

[8] 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.

[9] K. Lin, Y. Li, Q. Sun, S. Zhou, and L. T. Yang, "Computation offloading toward edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1584–1607, 2019. DOI: 10.1109/JPROC.2019.2922285.

[10] J. A. Donenfeld, "Wireguard: Next generation kernel network tunnel," in *Network and Distributed System Security Symposium (NDSS)*, 2017. DOI: 10.14722/ndss.2017.23160.

[11] Cloudflare, Inc., *Magic wan: Secure, performant connectivity for branch offices and data centers*, https://www.cloudflare.com/magic-wan/, 2021.

[12] NVIDIA Corporation, *Tensorrt: Programmable inference accelerator*, https://developer.nvidia.com/tensorrt, 2019.

[13] J. Dunn, R. Runge, and M. Snyder, "Wearables and the medical revolution," *Personalized Medicine*, vol. 15, no. 5, pp. 429–448, 2018. DOI: 10.2217/pme-2018-0044.

[14] G. D. Clifford et al., "False alarm reduction in critical care," *Physiological Measurement*, vol. 37, no. 8, E5–E23, 2016. DOI: 10.1088/0967-3334/37/8/E5.

[15] T. Tamura, M. Huang, and T. Togawa, "Current developments in wearable thermometers," *Advanced Biomedical Engineering*, vol. 7, pp. 88–99, 2018. DOI: 10.14326/abe.7.88.

[16] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001. DOI: 10.1109/51.932724.

[17] U.S. Congress, *Health insurance portability and accountability act of 1996*, Public Law 104-191, 1996.