Proactive Congestion Signaling Using Large Language Models in L4S Networks

Jonathan Kua Shiva Raj Pokhrel

IoT & Software Engineering Research Lab
Deakin University, Australia

Motivation

- Next-generation applications such as cloud gaming, AR/VR telepresence and real-time IoT require **sub-millisecond latency, minimal packet loss, and scalable throughput**.
- **The Low Latency, Low Loss, and Scalable Throughput (L4S) architecture (RFC 9330) [1]** combines Explicit Congestion Notification (ECN), DualPI2 Active Queue Management (AQM), and scalable congestion control (e.g., TCP Prague).
- L4S targets ultra-low latency and scalable throughput by coupling **Explicit Congestion Notification (ECN)** with **DualPI2 AQM** and scalable transports such as **TCP Prague**. Marking is preferred over loss to maintain sub-millisecond queuing delay.
- L4S is currently **reactive** - DualPI2 applies **Congestion Experienced (CE)** marks when queue thresholds are crossed. During short flows and bursty surges, feedback arrives late which increases delay variability and transient traffic overshoot, leading to poor performance.
- *Note:* ECN field distinguishes **Non-ECT, ECT(0)/ECT(1), and CE**. Receivers echo CE via ECE and senders reduce congestion window. Accurate ECN feedback is a prerequisite for L4S behavior.

Research Problem and Objective

- **Problem:** Threshold-based, queue-reactive ECN delays congestion feedback which limits responsiveness during transients and short flows.
- **Objective** Learn proactive CE marking from real-time telemetry. Train a compact LLM to infer CE decisions before queues exceed thresholds to stabilize rates and reduce queuing delay variance.
- **Question:** Can we integrate **Large Language Models (LLMs)** into the L4S pipeline [2] to perform predictive congestion signaling?

Our Proposed Approach: System Model and Formulation

- **NetLLM** [3] adaptation of Llama2 with three task-specific modules: a **Multimodal Encoder** for telemetry, a **Networking Head** that outputs ECN decisions, and a **Low-Rank Adaptation** pipeline for efficient fine-tuning.
- Experience tuples: Each sample is (s, a, r, done) where
  - $s = [\text{RTT}, \text{throughput}, \text{ECN}]$ is a time-aligned feature vector
  - $a \in \{0, 1, 3\}$ encodes ECN action: 0 Non-ECT, 1 ECT(1), 3 CE
  - $r = \max(0.1, \log_{10}(\text{throughput}/\text{RTT}))$ rewards high throughput at low latency
  - $\text{done} = \text{false}$ for supervised training.
- **Learning task:** Supervised ECN labeling from DualPI2 traces to approximate a proactive CE policy as a function of instantaneous latency and throughput. Outputs are constrained to network-valid actions by the networking head.

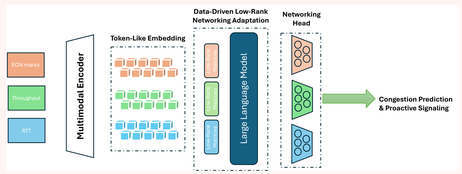


Figure 1. Our system design for proactive congestion signaling

LLM Integration into L4S

- **Telemetry path:** Router-side AQM exports instantaneous RTT estimates, per-flow throughput counters, and current ECN status as inputs to the encoder.
- **Decision path:** The LLM module returns a CE decision that can pre-mark packets prior to standard DualPI2 thresholds for L4S flows while preserving classic behavior for Non-ECT traffic.
- **Engineering goal:** Inference latency and footprint low enough for software routers, SmartNICs, and embedded platforms.

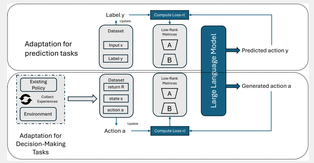


Figure 2. Key components of our proposed architecture

Results and Analysis

- Training convergence vs data scale:
  - Larger datasets improve stability and convergence speed.
  - 2k: Slow decrease from ~ 2.5 loss with broad dispersion indicating under-coverage of congestion regimes.
  - 4k: Faster initial learning but higher variance after $\sim$10 epochs.
  - 8k: Early, stable convergence with reduced variance.
  - 16k: Quickest convergence and narrow distribution; robust generalization.
- **Loss variability:** Train-loss standard deviation reflects diversity and stability. Joint mean–std analysis indicates 8k–16k is a balanced operating range.
- **Inference efficiency:** over a 2-hour test with 20 inference iterations, CPU $< 13\%$, RAM $< 15\%$, GPU peak $\sim 24\%$. Resource usage supports edge deployment.
- **Generalization:** Model distinguishes ECN states and predicts CE across flows with Prague, CUBIC, and NewReno which suggests transport-agnostic signal extraction.

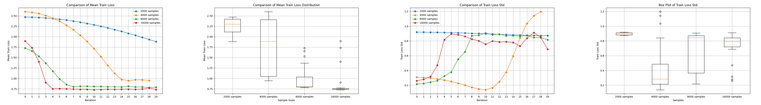


Figure 3. Mean training loss performance for each (2k, 4k, 8k, 16k) dataset.

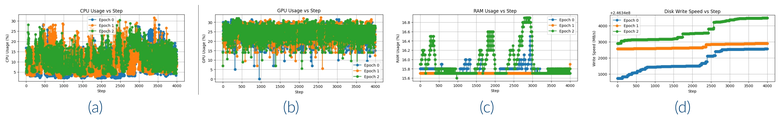


Figure 4. CPU, GPU, RAM and Disk performance for each (2k, 4k, 8k, 16k) dataset.

Experimental Testbed and Datasets

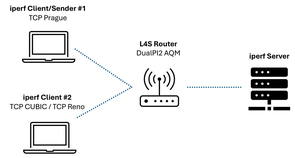


Figure 5. Network topology

- **Setup:** Ubuntu 22.04 router with DualPI2; clients generate flows with TCP Prague, CUBIC, and NewReno; traffic via iperf3; packet capture via tshark.
- **Data pipeline:** $\sim$ 220k packets captured; $\sim$ 82k usable samples with RTT and ECN after filtering. Training subsets: 2k, 4k, 8k, 16k.
- **Training:** Fine-tuned on cloud GPUs over 20 epochs; tracked mean loss and standard deviation; measured inference resource usage across repeated iterations.

Engineering and Deployment Considerations

- **Inline control:** Co-locate encoder and networking head with the queue discipline to avoid IPC overhead. Prefer CPU pinning with optional lightweight GPU acceleration when present.
- **Safety envelopes:** Gate proactive CE by minimum and maximum mark probabilities and rate-of-change constraints to preserve fairness and avoid oscillations.
- **Compression:** Quantization and low-rank adapters reduce latency and footprint while preserving CE decision fidelity.

Novelty and Contributions

- First demonstration of **LLM-driven congestion control within L4S networks**, and introduced a proactive AQM mechanism that marks congestion signals ahead of time.
- Designed a **supervised training pipeline** encoding RTT, throughput, and ECN telemetry into structured experience pools, and validated performance across datasets and hardware, showing scalability and low inference overhead.
- Released **open-source software and datasets** for reproducibility: `https://github.com/MPTCP-FreeBSD/L4S-LLM`

Conclusions and Future Work

- LLM-driven L4S anticipates congestion and issues earlier CE marks than reactive thresholding - improves responsiveness while remaining practical as a modular extension.
- Future work:
  1. **Accuracy target:** Validate CE decision accuracy on pre-labeled experience pools; proceed to predictive tests once accuracy exceeds $\sim 80\%$.
  2. **Temporal anticipation:** Measure whether proactive CE can pre-signal at $T - 0.2$ s relative to baseline CE at T under controlled bursts; quantify latency and throughput deltas.
  3. **In-router integration:** Prototype kernel-level integration with DualPI2 or SmartNIC offload to test fairness, stability, and coexistence with classic ECN.
  4. **Model lightening:** Explore smaller backbones, structured pruning, and mixed-precision inference tailored to embedded CPU or NIC instruction sets.

References

[1] Bob Briscoe, Koen De Schepper, Marcelo Bagnulo, and Greg White. Low Latency, Low Loss, and Scalable Throughput (L4S) Internet Service: Architecture. Internet Engineering Task Force (IETF) RFC 9330, January 2023.

[2] Shiva Raj Pokhrel, Deol Satish, Jonathan Kua, and Anwar Walid. Distilling Large Language Models for Network Active Queue Management. *arXiv preprint arXiv:2501.16734*, 2025.

[3] Duo Wu, Xianda Wang, Yaqi Qiao, Zhi Wang, Junchen Jiang, Shuguang Cui, and Fangxin Wang. NetLLM: Adapting Large Language Models for Networking. ACM SIGCOMM '24, 2024.