Poster: Emergency-Aware TSCH Scheduling for High-Density Low-Power and Lossy Networks

Jiamei Lv, Hailang Zhang, Yeming Li, Wei Dong, Yi Gao,
Zhejiang University

***Abstract*—High-density Low-Power Lossy Networks (LLNs) for IoT applications (e.g., smart cities) require ultra-low latency during emergencies like fires. Current TSCH schedulers (Orchestra, OST) use static/hash-based allocation, causing high delays and collisions for urgent traffic due to lack of prioritization and conflict resolution. In this paper, we present ETS, an emergency-aware scheduler featuring: (1) Binary-tree resource partitioning with preemption, (2) Slot-segmented RTS/CTS conflict avoidance, and (3) Multi-sink load balancing. Evaluations show ETS reduces latency by 23–42%, maintains 95% PDR, and lowers duty cycle by 37–52% versus baselines.**

I. Introduction

Nowadays, low-Power and Lossy Networks (LLNs) have become fundamental in supporting numerous Internet of Things (IoT) applications, such as smart cities, smart homes, and agricultural automation. With stringent resource constraints, LLNs prioritize energy efficiency and reliability, often at the expense of transmission latency. Consequently, the Time Slotted Channel Hopping (TSCH) protocol, defined in IEEE 802.15.4e [1], has been adopted in time-sensitive applications, due to its strict time slot division and allocation. However, a key limitation of TSCH is its inability to effectively and timely handle emergent traffic, particularly in high-density emergency scenarios like fire alarms, due to its default static scheduling.

Existing works on TSCH scheduling [2]–[5] have made improvements by employing either hash-based autonomous slot allocation or dynamic slot allocation adapted to traffic load. However, these approaches lack the real-time responsiveness and conflict resolution capabilities essential for emergency scenarios. Specifically, Orchestra's hash-based allocation may assign multiple nodes to identical time slots, causing transmission conflicts in high-density networks. While OST dynamically adjusts slots, it fails to detect and prioritize urgent data packets, leading to critical delays. Moreover, neither scheme incorporates conflict avoidance mechanisms for emergency transmissions. To bridge this gap, this paper presents ETS: an emergency-aware centralized TSCH scheduling framework specifically designed for high-density LLNs.

II. ETS Design

The core design of ETS integrates three key mechanisms to achieve low-latency, reliable emergency communication:

First, ETS introduces a binary-tree to help scheduling resource. It segregates emergency and regular traffic by reserving the right subtree for urgent data while reserving the left subtree for regular data, enabling immediate allocation of idle time slots without prior negotiation. It is worth mentioning that if the left subtree resources allocated for regular traffic are exhausted, ETS allows regular traffic to hierarchically appropriate resources from the right subtree. However, emergency events retain the highest priority: when an emergency event occurs, it directly preempts resources from the right subtree, even if those resources are currently occupied by regular traffic. This mechanism ensures that emergency packets bypass the static scheduling queue, reducing wait times. This mechanism ensures high spectrum utilization efficiency while avoid static scheduling procedures for emergency data packets; thereby significantly reducing end-to-end latency."

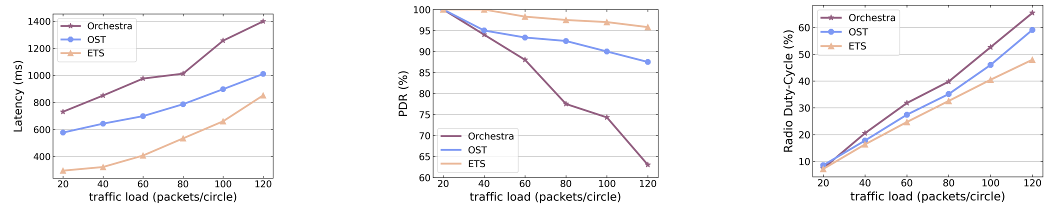
Second, ETS proposes a fine-grained conflict avoidance mechanism to mitigate collisions when multiple nodes attempt to use the same emergency slot simultaneously in dense networks. It employs the NHash function to compute slot offsets for emergency transmissions. Nodes calculate the offset $Offset_{SG}$ by: $Offset_{SG} = \mod(\text{NHash}(ID \ll 16 + NID), L_{SG})$, where ID represents the unique identifier of a node; NID denotes the network identifier, distinguishing different networks; L_SG is the number of segments into which a time slot is divided; and $\text{NHash}(x) = (x \oplus (x \ll 5) \oplus (x \gg 3) + c) \mod M$ (with $c = 0x9e3779b9$, $M = 256$). This splits each time slot into segments: three for RTS packets, one for CTS, and one for emergency data. Nodes send RTS packets at their calculated segment offset. The sink broadcasts CTS to authorize a node, ensuring exclusive channel use. This dynamic offset allocation and RTS/CTS handshake minimizes collisions by synchronizing transmission attempts .

Third, ETS introduces the multi-sink scheduler to tackle load imbalance in high-density networks, where uneven node distribution across sinks causes congestion in overloaded subnetworks and idle resources in others. The scheduler uses a "score" metric to quantify slot resource requirements. Mathematically, $\text{Score}(n) = M1 \times P_{emergency} + (1 - M1) \times P_{regular}$ (with $M1 = 0.2$). Score balances emergency and regular traffic priorities, giving higher weight to urgent transmissions. Subnetworks aggregate node scores to assess overall load, and new nodes join the subnetwork with the lowest total score, as determined by EB control packets synchronized via TSCH. This dynamic allocation ensures even distribution of traffic across sinks, preventing congestion in overloaded subnetworks and optimizing resource utilization .

III. Preliminary Evaluation

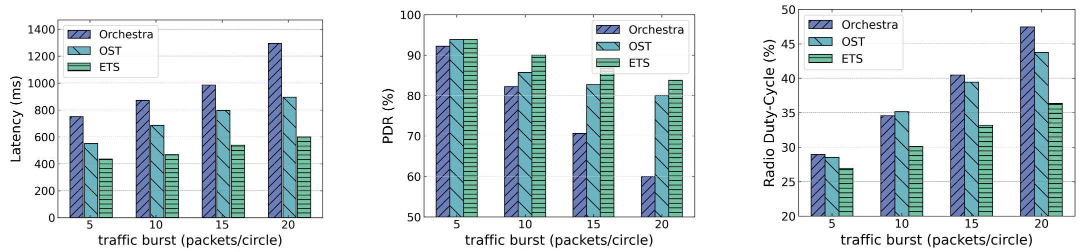
ETS is implemented on Contiki OS and evaluated using Cooja simulator, compared with state-of-the-art TSCH sched-

2325-1301/25/$31.00 ©2025 IEEE
DOI 10.1109/RTCSA66114.2025.00034



(a) End-to-end latency (b) End-to-end packet delivery ratio (c) Radio duty-cycle

Fig. 1: ETS performance with increasing traffic load



(a) End-to-end latency (b) End-to-end packet delivery ratio (c) Radio duty-cycle

Fig. 2: The performance of ETS with increasing traffic burst

ulers (Orchestra and OST). The setup includes 75 nodes and 3 sink nodes in a mesh network.

Impact of Traffic Load. Fig. 1 compares performance: *(a) Average end-to-end latency:* ETS reduces latency by 23% and 42% versus OST and Orchestra. Orchestra and OST incur queuing delays by deferring emergency packets. Latency increases with traffic load across all schemes due to contention. *(b-c) Packet Delivery Rate (PDR) and Radio Duty Cycle:* Rising traffic lowers PDR and raises duty cycle. OST maintains $>$ 80% PDR (down from 100%) due to delays and retransmissions. Orchestra performs worse, with collisions and drops caused by fixed slots and limited emergency capacity. ETS sustains $>$ 95% PDR via immediate emergency handling. Duty cycle (sink): All schemes degrade under load, especially Orchestra (contention from hash-based slots) and OST (negotiation overhead). ETS minimizes active time via direct scheduling within single slots.

Impact of Traffic Load Burst. We further evaluate ETS performance under bursty traffic conditions. As shown in Fig. 2 (a), rising burst frequency increases latency across all algorithms. Compared to uniform emergency data generation scenarios, ETS exhibits only marginal latency growth ($<$15%). Corresponding PDR degradation occurs with increased bursts (Fig. 2 (b)). This because resource contention causes queueing delays and subsequent packet loss. Crucially, ETS mitigates this by allowing non-contending nodes to immediately select alternative slots within the current cycle and significantly reducing wait-induced packet drops. For radio duty cycle (Fig. 2 (c)), ETS maintains superior efficiency despite inevitable increases under higher bursts. This stems from two competing factors: 1) Naturally elevated sink active time from increased data volume, and 2) ETS's streamlined scheduling that minimizes control overhead – reducing sink active time by 37-52% compared to Orchestra/OST in tested scenarios.

IV. Conclusion and Future work

ETS represents a significant advancement in TSCH-based LLNs by addressing the critical need for prompt, reliable communication in high-density emergency scenarios. Its integration of a binary resource tree for real-time slot allocation, a novel conflict avoidance mechanism, and a load-balanced multi-sink architecture ensures low latency, high reliability, and efficient resource use. Evaluations confirm ETS's superiority over state-of-the-art solutions in performance. For future work, we plan to extend ETS to handle more complex network topologies, e.g., those with mobile nodes or dynamic network sizes. Additionally, we aim to optimize the algorithm further by incorporating real-time network state information, enabling more adaptive scheduling decisions.

V. Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No. 62272407. the "Pioneer" and"Leading Goose" RD Program of Zhejiang under grant No. 2023C01033, the National Youth Talent Support Program and Yongiiang Talent Program of Ningbo.

References

[1] Watteyne T, Palattella M, Grieco L. Using IEEE 802.15.4e Time - Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement. 2015. RFC 7554.

[2] Simon Duquennoy, Beshr Al Nahas, Olaf Landsiedel, and Thomas Watteyne. "Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH," In Proc. of ACM SenSys, 2015.

[3] S. Jeong, H. -S. Kim, J. Paek and S. Bahk, "OST: On-Demand TSCH Scheduling with Traffic-Awareness," In Proc. of IEEE INFOCOM, 2020.

[4] Z. Yu et al.,"SmarTiSCH: An Interference-Aware Engine for IEEE 802.15.4e-based Networks," In Proc. of ACM/IEEE IPSN, 2022.

[5] Kim, Hongchan, et al. "Slot-size Adaptation and Utility-based Packet Aggregation for IEEE 802.15. 4e Time-Slotted Communication Networks." IEEE Internet of Things Journal 11.9 (2024): 16382-16397.