Proceedings Paper:

Gray, Ian, Pinter, Andras and Soares Indrusiak, Leandro (2024) Evaluation of Early Packet Drop Scheduling Policies in Criticality-Aware Wireless Sensor Networks. In: IEEE 27th International Symposium on Real-Time Distributed Computing:Proceedings. IEEE 27th International Symposium on Real-Time Distributed Computing, 22-25 May 2024 IEEE, TUN.

Evaluation of Early Packet Drop Scheduling Policies in Criticality-Aware Wireless Sensor Networks

Andras Pinter
Department of Computer Science
University of York
York, United Kingdom
alp567@york.ac.uk

Leandro Soares Indrusiak
School of Computing
University of Leeds
Leeds, United Kingdom
L.SoaresIndrusiak@leeds.ac.uk

Ian Gray
Department of Computer Science
University of York
York, United Kingdom
ian.gray@york.ac.uk

Abstract—**This paper introduces three autonomous, criticality-aware packet scheduling policies that address the impact of high traffic loads and degraded conditions in wireless sensor networks. The proposed policies, collectively referred to as Early Packet Drop (EPD), leverage cross-layer information, including RPL Rank, link quality, and time-slotted Medium Access Control schedule, to mitigate Quality of Service degradation. Simulation results demonstrate that EPD consistently outperforms a Criticality-Monotonic Scheduling (CMS) baseline.**

Keywords*— *wireless sensor networks, criticality-aware packet scheduling, Early Packet Drop, service degradation management

I. INTRODUCTION

Large-scale machine-to-machine communication plays a key role in enabling the deployment of smart infrastructures in industrial or safety-critical settings. The aggregation and processing of high volumes of data in increasingly complex and dense wireless sensor networks (WSNs) introduces diverse challenges as network devices frequently contend with unpredictable environmental conditions, congestion, and internal interference. Real-time networks may experience surges in traffic during equipment failure, emergency situations or abnormal events prohibiting delivery of all data within stringent deadlines.

In this paper three autonomous, criticality-aware packet scheduling policies are proposed and evaluated, with emphasis on their ability to alleviate degradation of Quality of Service (QoS) during periods of high traffic. Autonomous scheduling policies are well-suited to enhance adaptivity, scalability and achieve improved responsiveness. By avoiding the dependency on coordinator nodes and centralised decision-making, and utilising up-to-date, locally available information, they effectively mitigate communication overhead and lead to reduced delays.

Simulation results indicate that employing Early Packet Drop (EPD) scheduling policies efficiently alleviate congestion, leading to significantly improved end-to-end packet delivery and collision reduction compared to a criticality-monotonic scheduling baseline. Notably, EPD based on cross-layer information, particularly Medium Access Control (MAC) schedule awareness, achieves more uniform degradation across the network, mitigating service starvation in peripheral nodes.

This paper is structured as follows. Section II provides an overview of dynamic and mixed-criticality packet scheduling strategies outlined in prior research. Section III presents the problem statement and objectives. The system model is defined in Section IV. Section V outlines the proposed approach to address the limitations of existing methods, introducing three EPD packet scheduling policies. The experimental setup and evaluation of simulation results are presented in Sections VI and VII. Section VIII concludes the discussion, identifying prospective areas for future research.

II. RELATED WORK

A. *Criticality aware and priority based packet scheduling*

Extensive research has been conducted on criticality aware MAC and packet scheduling for real-time WSNs. We adopt the definition of mixed criticality by Burns et al. [1], [2], referring to system components having different levels of operational importance and consequences of failure. The authors introduce AirTight, a hybrid scheduling protocol, which combines time-slotted medium access control with autonomous packet scheduling. Slot tables governing access to the wireless medium are constructed offline by a central coordinator node and are distributed during system initialisation. Decisions about the transmission of various criticality flows are made locally on each node at runtime based on the node's current operational criticality level. This is a dynamic property derived from fault conditions, not to be confused with the flows' criticality inherited from the associated application tasks. In AirTight, time-bounded latency guarantees facilitate deterministic communication by considering hop response times along a route. Individual responses are not required to meet a deadline, as long as the combined end-to-end latency is within the flow's deadline.

Zhang et al. [3] combine packet and transmission method selection to mitigate packet drops caused by missed deadlines. Packets are grouped by transmission method, and estimated transmission times are computed to determine the optimal choice under given channel conditions. The authors report simulation results indicating an overall reduction in the probability of packet drops. To further improve performance, packets that are expected to miss their deadlines within the current superframe are dropped. Early packet drop can enhance performance since the excluded packets do not contribute to network throughput or cause collisions. However, the proposed method solely focuses on the probability of deadline misses in the current superframe. Consequently, packets may occupy extended segments of multi-hop routes without eventually meeting deadlines, impeding the transmission of other packets in preceding superframes. Packet prioritisation and selection within groups is deadline-monotonic and does not take into account criticality.

Zhang et al. [4] consider the unreliability of wireless links, whilst aiming to meet timing and reliability goals. The proposed RD-PaS packet scheduling algorithm determines a minimum retransmission count based on each link's packet delivery ratio (PDR) to guarantee packet delivery with the specified level of confidence. Schedule synthesis in nominal mode uses static scheduling during network initialisation. Environmental degradation triggers mode shifts between nominal and rhythmic modes, prompting dynamic schedule updates with adjusted task parameters.

The approach represents a trade-off between deterministic support for real-time applications and flexibility for graceful degradation of non-critical services during network disturbances. However, it assumes prior knowledge of task specifications for static schedules and requires the distribution of updated parameters by a controller node in rhythmic mode during already compromised communication. Additionally, the precalculated number of required transmission slots during network initialisation remains static and does not adapt to actual link quality changes in nominal mode.

Nasser et al. [5] present Dynamic Multilevel Priority scheduling (DMP), a dynamic mixed-criticality scheme designed to minimise transmission delays and maintain data freshness. Local and forwarded packets are organised into a multi-level priority queue system based on criticality, hop distance between the source and receiving nodes, and data size. Packets generated by real-time services are placed in the highest priority queue and are forwarded on a First Come First Serve (FCFS) basis. Packets from remote nodes are placed in the second-highest priority queue, while locally generated packets are assigned the lowest priority. The policy allows for the pre-emption of higher priority tasks to prevent starvation. While this design decision improves fairness, the impact on the end-to-end delivery of real-time flows is not assessed. The authors assume infrequent execution of high-criticality emergency tasks causing the pre-emption of non-real-time tasks, which may not be the case in some practical scenarios. Additionally, the assumption that packets from remote nodes invariably have less laxity may not always hold. Packet deadlines may be considered instead for improved end-to-end PDR.

Chen et al. [6] suggest enhancements to DMP through dynamic priority adjustment (PAS) based on traffic awareness. When a predefined threshold is reached in a packet queue, it transitions to a "priority-promoting" state. During this state, packets are moved to higher priority queues until the backlog falls below the threshold or for a predetermined number of scheduling cycles. The authors observe improved average waiting times in queues and end-to-end delays. This cross-layer policy allocates the remaining time slots in the MAC schedule proportionally after transmitting all high-criticality packets. Similarly to DMP, PAS overlooks the impact of priority adjustments on critical tasks. While it reduces latency and average waiting time, it ignores packet deadlines when selecting packets for forwarding on a FCFS basis.

Striving for balance between performance optimisation and adaptability, current research predominantly focuses on packet scheduling policies assuming prior knowledge of network and service configurations, such as the presence of a coordinator node or preconfigured task parameters in different operational modes. Likewise, assumptions are commonly made about environmental conditions, such as static or worst-case link quality throughout the network's operation. Consideration of criticality is often overlooked. In contrast, heterogeneous WSNs consist of devices that may differ in terms of functionality, sensing capabilities and available resources where these assumptions may not be applicable. Therefore, there is a need to explore flexible, distributed packet scheduling policies that draw upon the merits of existing approaches whilst relaxing these assumptions to efficiently manage performance degradation.

B. Time-slotted Medium Access Control scheduling

To ensure determinism and facilitate schedulability analysis for the provision of reliability and timing guarantees, real-time scheduling protocols, such as AirTight, commonly employ time-slotted MAC on the Data Link Layer [1], [2]. This approach involves the division of time into discrete slots, which are allocated to individual nodes for packet transmission or reception. Network-wide time synchronisation and organised access to the wireless medium alleviate internal interference, collisions, and enhance energy efficiency by enabling nodes to reduce radio duty cycles through entering low-power states during inactive slots. Burns et al. extend AirTight's schedulability analysis to multichannel configurations based on affinity sets, partitioning flows to transmit on specific subsets of available frequency channels [7].

The combination of TDMA (Time Division Multiple Access) with FDMA (Frequency Division Multiple Access) is also adopted in Time-Slotted Channel Hopping (TSCH) [8]. In TSCH nodes transmit and receive on different frequencies in each time slot following a predetermined hopping sequence across superframes, which mitigates the effects of external interference and multi-path fading.

In our experimental setup we elected to adopt the Orchestra protocol for the synthesis of MAC schedules [9], [11]. In contrast with centralised and distributed protocols, Orchestra nodes do not need to negotiate with each other to agree on a communication schedule or transmit the network-wide schedule created by a single coordinator node. Instead, nodes compute their own schedules based exclusively on routing tables available locally. Orchestra integrates TSCH and, whilst in practical scenarios external interference may not be entirely eliminated by this method, this provides basis for the evaluation of packet scheduling policies without the need for incorporating environmental factors in the simulations. A detailed description of Orchestra's slot allocation mechanism and the protocol's configuration in the experimental setup is provided in Section VI.

III. PROBLEM STATEMENT AND OBJECTIVES

In large-scale WSN deployments, common challenges involve limited throughput and congestion, particularly in multi-hop configurations directing data towards aggregating gateway nodes. Surges in traffic may occur during disruptions caused by equipment failure or emergencies. Depending on spatial setup, environmental conditions, and medium access control, contention and resulting packet collisions can impact end-to-end PDR. Meeting Quality of Service (QoS) requirements across criticality levels requires effective resource allocation to ensure timely delivery of real-time data packets while minimising impact on lower criticality services. In distributed or autonomous scheduling, decision-making may be constrained by locally available information. To address these challenges, our objectives are to:

1. Propose and evaluate autonomous, criticality-aware packet scheduling policies that rely solely on locally available data, without the need for a central coordinator node.
2. Assess the policies' efficiency to manage QoS degradation during heightened traffic scenarios in heterogeneous WSNs, without prior knowledge of application task properties on other network nodes.
3. Evaluate best-effort compliance of reliability and timing requirements, assuming the deadline of each forwarded data packet is known.
4. Assess the potential advantages of integrating cross-layer information into packet scheduling decisions, particularly focusing on effectiveness in alleviating congestion and reducing packet collisions. Consideration is given to Network and Data Link layer metrics, including RPL Rank, link PDR and MAC schedule.

IV. SYSTEM MODEL

We consider a WSN configured in a multi-hop, tree topology with a single sink and multiple, distributed field devices (Fig. 1), denoted as $N = \{n_1, n_2, \ldots, n_k\}$. Nodes are resource constrained devices, equipped with a half-duplex radio transceiver with limited transmission range, which necessitates peer-to-peer packet switching and allows a node to either transmit to or receive from one neighbouring node at a time.

Multiple, non-interfering frequency channels are available for transmission. The IEEE802.15.4 protocol defines 16 channels in the 2.4GHz band although it is assumed that not all of these are accessible to the network due to factors such as external interference or regulatory constraints: C $(1 < C < 16)$.

Medium access control scheduling adopts TSCH. Communication is time-synchronised and divided into discrete time slots with a duration that is sufficient to accommodate the transmission of a single packet and its acknowledgement (L_{slot} = ~ 10ms). Nodes transmit and receive on a different frequency in each time slot following a predetermined hopping sequence. It is assumed that clock drift between nodes is negligible ($L_{drift} << L_{slot}$). This is a reasonable assumption as in Orchestra, the MAC scheduling protocol selected for our experimental setup, beacon frame transmissions are used for clock synchronisation between nodes [9], [10].

We assume the use of RPL (Routing Protocol for Low-Power and Lossy Networks) for network management and packet routing [11] . In RPL networks, the topologies take the form of Directed Acyclic Graphs (DAGs) that can be partitioned into one or more Destination Oriented DAGs (DODAGs). Each DODAG is associated with a sink node. Lower Rank values signify closer proximity to the root, effectively establishing a partial order of node positions within the RPL DODAG.

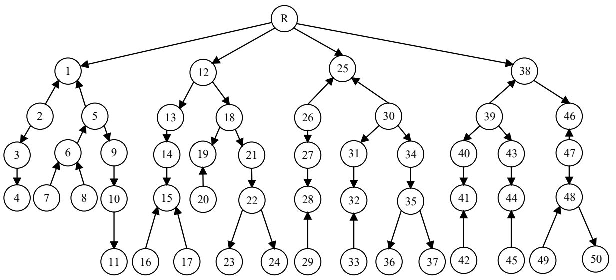


Fig. 1. Network routing topology in the experiment setup. The network consists of a single RPL DODAG (one root node) and 50 field devices.

The combination of autonomous, TSCH-based MAC scheduling and the use of RPL, results in a system model well-suited for our analysis. The objective is to evaluate package scheduling policy choices in isolation from other variables as comprehensively as possible. TSCH reduces the impact of external interference, minimises packet collisions and the resulting communication overhead involved in network repair. RPL also introduces minimal overhead of control messages for network management, provides mechanisms to optimise network topology for reduced latency, and the node ranking metric offers additional information for packet scheduling decisions.

Each node n_k executes a set of tasks, $T_k = \{\tau_{k,1}, \tau_{k,2}, \ldots, \tau_{k,l}\}$, defined by the following parameters: phase ($Ph_{k,l}$), period ($P_{k,l}$), relative deadline ($D_{k,l}$, where $D_{k,l} \leqslant P_{k,l}$), and criticality ($\Phi_{k,l}$). Tasks release jobs that generate application data, which is transmitted as packets through the wireless medium, traversing a multi-hop path to reach the network sink. The decision of which packets to transmit is made locally on each node based on a packet scheduling policy. Packets, generated on a node or received from descendant nodes for forwarding, are placed in a queue. In alignment with Chen et al. [6], we presume that each node possesses the capability to store a minimum of 50 packets. Packets selected for forwarding are dequeued before transmission. Therefore, in cases of collision, packets are lost and are not subject to re-transmission. Upward traffic flow is assumed to be dominant for application data.

V. PROPOSED APPROACH

A. Early Packet Drop

Focusing on mitigating network congestion, contention causing degradation of QoS across all criticality levels, we introduce three autonomous, deadline-aware Early Packet Drop (EPD) policies that expand on exclusively considering deadline misses within the current superframe, as presented in Zhang et al. [3], and instead address the broader impact of throughput bottlenecks across extended timeframes. EPD is augmented with the following locally available cross-layer information:

1. RPL Rank (Network Layer), indicating the node's routing distance from the root node.
2. PDR (Data Link Layer), indicating up-to-date information about link quality between the forwarding node and the next hop.
3. MAC schedule (Data Link Layer).

Incorporating up-to-date data on link quality is expected to yield more accurate estimates of required transmission slots per hop. Similarly, with the adoption of autonomous MAC scheduling policies, packet scheduling can be further enhanced as information about neighbours' schedules may be locally available.

B. Proposed packet scheduling policies

We evaluate the following package scheduling policies in mixed-criticality scenarios.

Criticality-Monotonic Scheduling (CMS): CMS is selected as our baseline. Priorities are assigned based solely on task criticality (Φ). Packet deadlines, and other network conditions, such as traffic load, are not considered. In mixed-criticality scenarios CMS may lead to starvation and deadline misses in lower criticality flows.

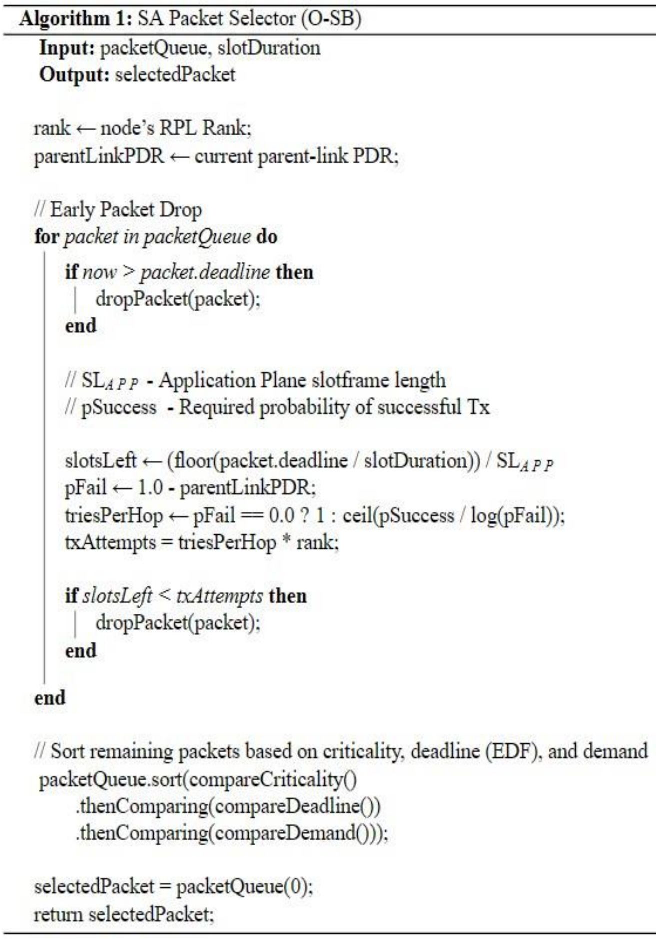
Criticality-Monotonic Scheduling with Early Packet Drop (CMS-EPD): CMS-EPD aims to improve performance by (1) assessing if delivery of each packet within its deadline is possible and (2) dropping packets that will definitely miss their deadlines. The policy assumes immediate transmission of the packet in the next transmission slot at each hop and considers the node's RPL Rank to determine the number of slots required to forward the packet to its destination.

CMS-EPD Earliest Deadline First (CMS-EPD-EDF): Earliest Deadline First (EDF) scheduling is commonly adopted in real-time communication systems and industrial automation use cases. In packet scheduling the approach prioritises packets for transmission based on their deadlines. Combined with criticality-monotonic scheduling, this policy may lead to the starvation of lower criticality tasks and tasks with later relative deadlines under heavy traffic.

In addition, a further improvement was implemented in our experimental setup. Nodes maintain information about traffic demand of descendant neighbours. In cases where criticality-monotonic and EDF policies identify multiple potential packets for transmission, packets received from less active nodes are prioritised to expedite freeing up bandwidth. This approach aims to address issues encountered in unbalanced networks.

Schedule-aware CMS-EPD-EDF (SA): SA represents a modification of CMS-EPD-EDF. Instead of assuming immediate packet transmission at each hop to assess whether the packet will meet its deadline, SA considers the current allocation of transmission slots in the node's MAC schedule. Although the precise time offset of Tx slots allocated to subsequent nodes may be unknown to the currently forwarding node, depending on the adopted MAC scheduling policy, informed predictions can be made. In autonomous scheduling, described in detail in Section VI, the transmission frequency remains constant for all nodes, with sender-based policies assigning one Tx slot per superframe [9], [10]**.** This approach provides a more accurate estimate of remaining transmissions before a packet exceeds its deadline.

The policy determines the number of packet forwarding attempts based on the packet delivery ratio of the link to the node's parent to ensure successful transmission with the required confidence. The same PDR is assumed for all subsequent links. If the available Tx slots before the deadline are insufficient compared to the estimated required slots, the packet is dropped. SA utilises traffic awareness to identify more active descendant nodes or subtrees in the same manner as CMS-EPD-EDF. The remaining packets are arranged in a criticality-monotonic order, followed by sorting based on earliest deadline first and, finally, least demand first (Fig. 2).



```
Algorithm 1: SA Packet Selector (O-SB)
  Input: packetQueue, slotDuration
  Output: selectedPacket

rank ← node's RPL Rank;
parentLinkPDR ← current parent-link PDR;

// Early Packet Drop
for packet in packetQueue do
    if now > packet.deadline then
        dropPacket(packet);
    end

    // $SL_{APP}$ - Application Plane slotframe length
    // pSuccess  - Required probability of successful Tx

    slotsLeft ← (floor(packet.deadline / slotDuration)) / $SL_{APP}$
    pFail ← 1.0 - parentLinkPDR;
    triesPerHop ← pFail == 0.0 ? 1 : ceil(pSuccess / log(pFail));
    txAttempts = triesPerHop * rank;

    if slotsLeft < txAttempts then
        dropPacket(packet);
    end
end

// Sort remaining packets based on criticality, deadline (EDF), and demand
 packetQueue.sort(compareCriticality()
      .thenComparing(compareDeadline())
      .thenComparing(compareDemand()));

selectedPacket = packetQueue(0);
return selectedPacket;
```

Fig. 2. SA algorithm pseudocode.

C. Network management and Medium Access Control

Whilst incorporating cross-layer data is expected to improve EPD, the proposed policies are intended to be evaluated independently from the adopted Data Link and Network Layer protocols without the need to consider communication overhead incurred in network management and the distribution and updates of MAC schedules. To this aim, we assume the adoption of RPL (Routing Protocol for Low-Power and Lossy Networks) for network management and packet routing, and the Orchestra protocol for the synthesis of medium access schedules [9], [11].

VI. EXPERIMENTAL SETUP

The evaluation is conducted under heavy and moderate traffic scenarios (Table 1), representing progressively less severe conditions where, nonetheless, the delivery of all packets is deemed infeasible. Performance is measured in terms of packet collision count and overall end-to-end Packet Delivery Ratio adjusted for packets that have met their deadlines. Furthermore, we examine degradation patterns across the network for two EPD scheduling policies, CMS-EPD, and

SA, by comparing overall end-to-end PDRs based on RPL Rank. The following sub-sections provide details of the experimental setup at the Data Link, Network and Application Layers of the protocol stack.

A. *Orchestra slotframe cell allocation*

Orchestra maintains a hierarchical organisation of 3 traffic planes (TSCH, RPL, and Application Planes in priority order) and corresponding schedule slotframes of different length. These planes are dedicated respectively to the transmission of Enhanced Beacon (EB) frames for time synchronisation and network association, RPL broadcast control messages to establish and maintain network routes and unicast packet switching for application data transfer.

By default, each plane is assigned a single channel to avoid collision of packets across the different types of traffic although the channel allocation is not static. Adhering to the TSCH protocol, the transmission frequency of a slot changes with each superframe cycle. Kim et al. [10] propose the enhancement of Orchestra through multi-channel utilisation. In our experimental setup we assign 4 channels to Application Plane traffic (C_{APP} (4)).

Furthermore, the slotframe lengths of each plane are coprime to avoid persistent collisions of overlapping slots [9]. When addressing overlaps, priority is given to traffic scheduled in the highest-priority plane. In our simulations the following slotframe lengths were used: $SL_{TSCH} = 397$, $SL_{RPL} = 31$, $SL_{APP} = 47$.

Slots in each slotframe are allocated based on each node's MAC ID and RPL neighbour (ancestor/descendant) relationship following receiver- or sender-based scheduling. In receiver-based scheduling (O-RB) a shared slot is allocated to node n_k at time offset

$$t_{Rx} = mod(hash(IDn_k), SL_{APP}) \tag{1}$$

for packet reception from any neighbour. Similarly, in sender-based scheduling (O-SB) a shared slot at time offset

$$t_{Tx} = mod(hash(IDn_k), SL_{APP}) \tag{2}$$

is allocated for packet transmission to any neighbour. Channel assignment for Application Plane data packets follows the same principle for O-RB (3) and O-SB respectively (4).

$$c_{Rx} = mod(hash(IDn_k), C_{APP}) \tag{3}$$

$$c_{Tx} = mod(hash(IDn_k), C_{APP}) \tag{4}$$

The scheduling policy, informed by IDn_k, enables neighbouring nodes to determine transmission and reception times and frequencies with node n_k. EB frames and RPL broadcast messages are transmitted on dedicated channels assigned to the TSCH and RPL Planes. The Tx and Rx time slots for EB frames are determined similarly based on IDn_k and SL_{TSCH}, while RPL messages are broadcast in a single predefined shared slot.

TABLE I. EXPERIMENTAL NETWORK TRAFFIC SCENARIOS

Network traffic scenarios	**Scenario ID**	**Task periods (sec)** [a]		
		P_{HI}	***P_{MED}***	***P_{LO}***
Heavy 1	H1	2.5	1.25	2.5
Heavy 2	H2	5	2.5	5
Moderate	M	15	7.5	15

[a] Packets' relative deadlines align with the period of the task that generated them ($D_i = P_i$)

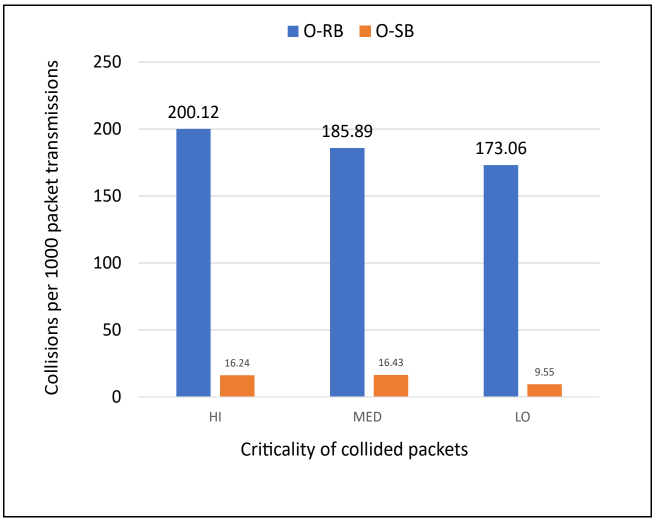


Fig. 3. Frequency of packet collisions in O-RB and O-SB scheduling (Scenario H1). Packet loss in O-RB predominantly occurs as a result of collisions rather than through competition for slots among packets of varying criticalities, which are orchestrated by the packet scheduling policy (CM).

B. *Contention in O-RB scheduling*

While scheduling overhead is minimal, the protocol's design presents a few intrinsic issues. In O-RB only one receiver-based slot per node is allocated to be used for communication. This may cause contention, especially during periods of heightened traffic, as several neighbours may try to communicate with a node in the same time slot. Comparison of the frequency of packet collisions using the two scheduling policies is shown in Fig. 3.

During our experiments this impact was consistently observed across various traffic scenarios and network topologies, obfuscating the distinct characteristics of the evaluated packet scheduling policies and the impact of local scheduling decisions. Therefore, we do not include O-RB in our simulation configuration and restrict the scenarios to sender-based scheduling.

O-SB presents another challenge. As Kim et al. [10] point out, although this technique alleviates contention, each node can only transmit to one neighbours in a slotframe, impacting latency. In addition, as both slot types are calculated using the modulo function, multiple nodes may try to transmit simultaneously, therefore packet collision and contention is not completely eliminated.

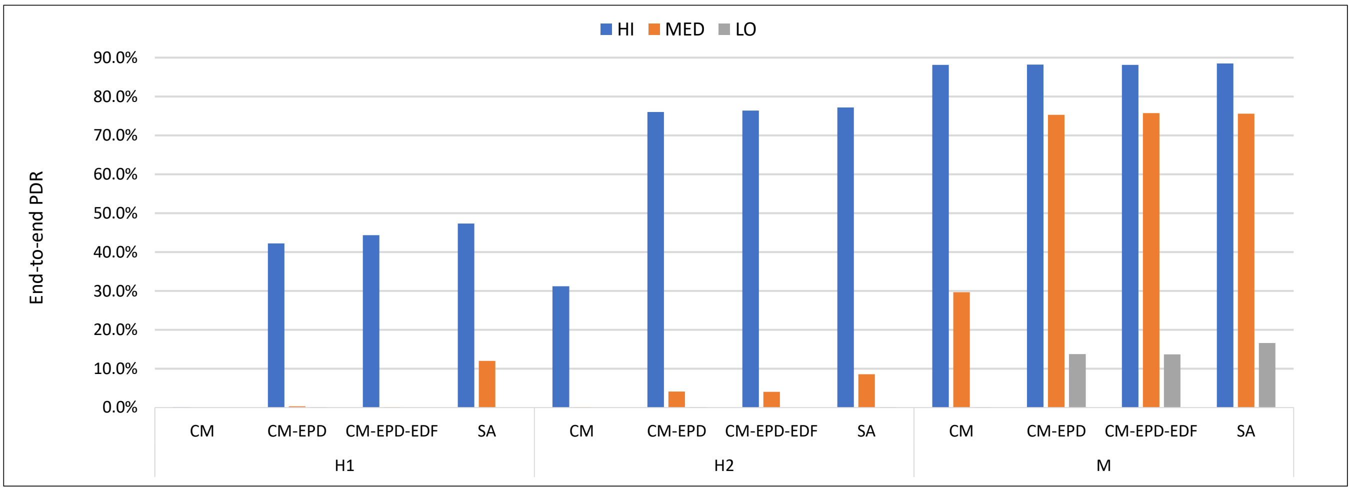


Fig. 4. End-to-end PDRs in all simulation scenarios. In scenario H1 almost all packets miss their deadlines under the CM policy. This is due to Tx slot allocation to packets that miss their deadlines, whilst blocking the transmission of packets that would meet theirs.

C. *Network properties*

The simulated sensor network comprises a single RPL DODAG instance with 50 field devices and one root node (Fig. 1). Our simulation scenarios replicate the adaptive functionality of RPL. Network nodes monitor link health through unacknowledged transmissions and in cases where nodes become unavailable due to internal interference and packet collisions, network repair is triggered. Attempts to join the network involve the selection of parent nodes with the lowest RPL Rank and the least number of descendant nodes.

D. *Application layer*

In our simulations, all nodes execute one high (HI), one medium (MED) and one low (LO) criticality task with a distribution of 25%, 50%, and 25% of packets generated by each task. The specific number of criticality levels is not relevant for our evaluation as the proposed policies may be applied in any multi-criticality scenarios.

Network performance is evaluated under three scenarios summarised in Table 1. Heavy traffic scenarios (H1, H2) represent severe conditions where MED and LO criticality flows are expected to be severely impacted as criticality-monotonic scheduling policies prioritise real-time (HI criticality) flows. In contrast, the moderate traffic scenario (M) is designed to assess system behaviour under a more manageable network load, where a smaller impact on HI criticality flows and a gradual improvement of QoS for lower criticality flows are anticipated.

We choose collision count and end-to-end Packet Delivery Ratios adjusted for packets delivered to the gateway (DODAG root) within their deadlines as performance metrics across all criticalities. In all cases, the relative deadline of packets aligns with the task's period that generated each packet ($D_i = P_i$). Packets are placed in a queue and dequeued when selected for transmission, therefore are not subject to retransmission in the case of collisions. These are noted by the sender node as unacknowledged frames. The evaluation environment was implemented using VisualSense, a collection packages for the modelling of WSNs [12], [13] based on the discrete event simulation engine of Ptolemy II [14], [15].

VII. PERFORMANCE EVALUATION

A. *End-to-end PDR*

All policies utilising early packet drop consistently outperform CM, criticality-monotonic scheduling which does not consider additional factors. Without early packet drop resources are allocated to transmit packets that miss their deadlines, which, in turn, impedes the transmission of packets that could otherwise meet their deadlines. This compound effect is particularly evident in scenario H1 (Fig. 4 and Fig. 5).

HI-criticality data flows are prioritised by all policies in all scenarios. With the reduction in traffic load a slowing trend is observable in the improvement of PDRs across all criticalities (e.g. improvement of HI criticality PDRs decelerates as lower criticality flows' PDRs increase).

Notably, the onset of the improvement in PDR for lower criticality flows occurs earlier in SA compared to other approaches. The advantages of schedule awareness and link quality based EDP are most pronounced in scenarios H1 and H2. These approaches, therefore, should be considered in efforts to mitigate the consequences of degradation in network conditions or during heavy traffic. SA achieves comparable PDRs for HI criticality flows in the moderate traffic scenario (M), with marginally higher PDRs for lower criticality flows (Fig. 4).

B. *Managing service degradation under heavy traffic*

Under severely degraded conditions or during periods of surge in traffic the transmitting nodes' ability to accurately assess the likelihood of a packet being successfully delivered to its destination becomes crucial as each packet that would miss its deadline needs to be discarded at the earliest possible time to free up bandwidth for viable transmissions. Under these circumstances criticality aware systems often have to compromise on the QoS of lower criticality services.

Simulation results for heavy traffic demonstrate that selective packet forwarding stands out in improving overall network performance. In scenario H1, CM exhibited poor performance as most packets failed to meet their deadlines (0.02% PDR for HI criticality flows, while no MED and LO criticality packets were delivered).

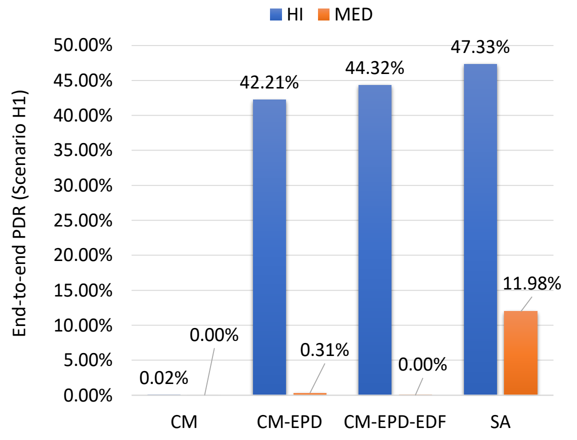


Fig. 5. Comparison of end-to-end PDRs under heavy traffic (H1). SA outperforms all other scheduling policies in the delivery of HI and MED criticality packets.

In contrast, employing early packet dropping CM-EPD, CM-EPD-EDF, and SA yielded markedly improved results: 42.21%, 44.32% and 47.33% PDR for HI criticality packets respectively. Furthermore, using locally available information about their medium access schedule, and applying this information to subsequent hops enables nodes to make more informed assumptions about conditions along the route. The schedule-aware SA policy achieves the highest end-to-end PDR of HI criticality packets within deadlines, while also succeeding in delivering non-real-time application data (11.98% of MED criticality packets). In comparison, the other evaluated EPD policies prioritise only HI criticality traffic and ignore MED and LO criticality flows (Fig. 5).

C. *Localised vs distributed service degradation*

Simulation findings revealed an additional distinguishing feature of schedule awareness. Nodes employing early packet dropping are expected to prioritise traffic closer to the root. Assuming homogeneous task configurations (e.g. identical tasks with similar periods across all nodes), packets traversing longer routes are more likely to miss their deadlines, leading to potential starvation of services in higher RPL Rank nodes during heavy traffic. Degradation is therefore expected to begin in the periphery of the network and progress toward the root node.

However, schedule awareness mitigates this characteristic. Our experiments demonstrate that SA more accurately estimates the number of upcoming transmission slots before specific deadlines, in contrast to CM-EPD and CM-EPD-EDF, which optimistically assume immediate forwarding at each hop. Orchestra allocates a single transmission slot in the superframe on the Application Plane, and in the event of unacknowledged packets, retransmission occurs no sooner than S_{LAPP} x L_{slot}. Using this information results in a more proactive packet dropping policy, enabling SA to efficiently clear bandwidth during heavy traffic for data originating farther from the root node (Fig. 6). However, this efficiency may lead to discarding more packets overall and the trade-off should be considered based on specific deployment requirements.

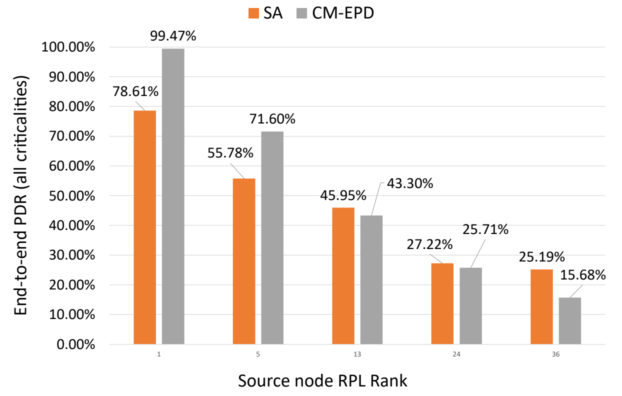


Fig. 6. Overall PDR by RPL Rank of packet source nodes (Scenario H1). During heavy traffic a higher percentage of packets originating in the periphery of the network (Rank 4 and 5 in our simulations) are delivered by SA than CM-EPD. In contrast, CM-EPD achieves significantly higher PDR for traffic originating closer to the network root.

D. *Collision avoidance*

Whilst O-SB scheduling mitigates the chance of collisions, the issue is not eliminated completely. Furthermore, contention between two nodes occur repeatedly until the slot table is updated as a result of network repair triggered by repeating unacknowledged transmissions following collisions or network topology changes as nodes join or leave the network.

In criticality-aware networks under heavy traffic the impact of collisions on HI criticality flows is most salient as lower criticality packets are de-prioritised. Our experiments show that, whilst all EPD policies succeed in reducing collisions compared with CM, SA achieves significantly better results. In addition to schedule awareness, SA considers the most up-to-date information about link quality (the current PDR of the link to the node's parent) for estimating the required number of transmissions per hop. The combination of schedule awareness and link quality-based estimation yields significantly better performance in scenarios H1 and H2 (Fig. 7).

VIII. CONCLUSION AND FUTURE WORK

In this study, we introduced and assessed three autonomous, criticality-aware packet scheduling policies, focusing on the effectiveness of early packet dropping in alleviating network congestion, reducing packet collisions, and mitigating service degradation during periods of heightened traffic. Simulation results indicate that integrating cross-layer information into decision-making, particularly incorporating schedule awareness and up-to-date data on link quality, enhances overall network performance. This improvement is reflected in a higher end-to-end packet delivery ratio, significantly fewer collisions, and the prevention of service starvation at peripheral network nodes.

The evaluation of the proposed policies has identified several potential areas for future research. As noted by Kim et al. [10], assigning a single transmission slot to each node in sender-based scheduling introduces latency, impacting early

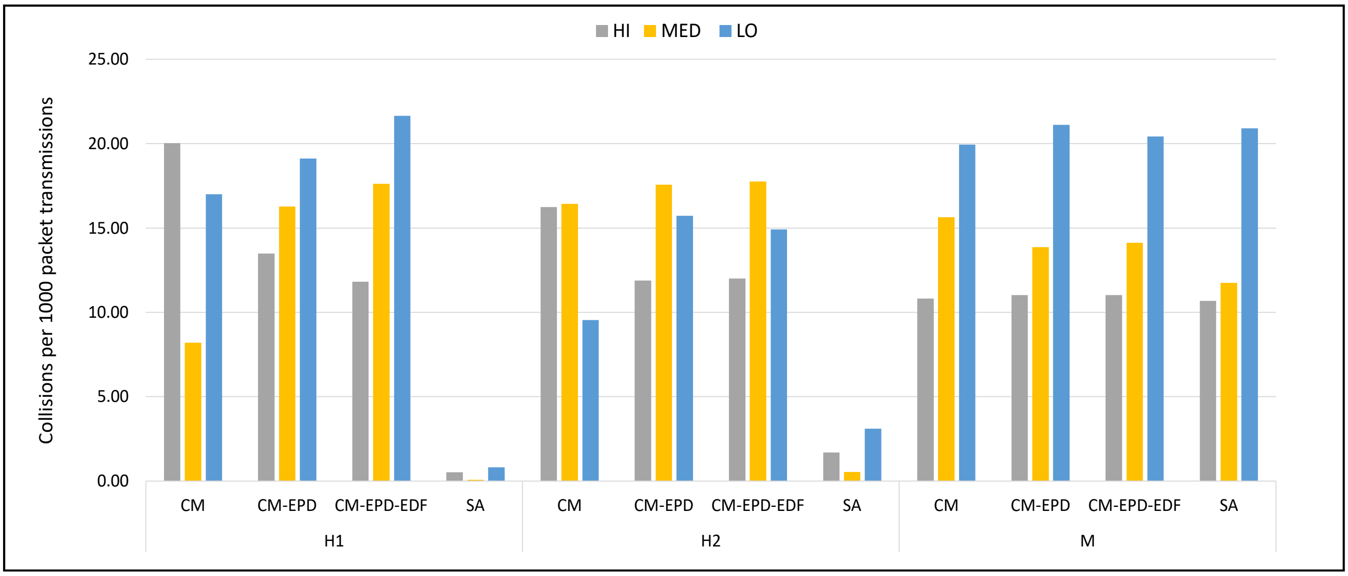


Fig. 7. Collision avoidance under heavy and moderate traffic scenarios. SA achieves significantly lower collision count during elevated traffic.

packet dropping decisions. Leveraging the characteristics of autonomous medium access control scheduling, the suggested schedule-aware policy could be refined by including information about neighbouring nodes' schedules. Schedules may be calculated for all nodes with known IDs, which may be distributed in sub-sections of the network with minimal communication overhead.

Furthermore, exploring the distinctions in how more or less optimistic EPD policies mitigate QoS degradation at close proximity to the root node compared with remote nodes could provide insights for selecting packet scheduling policies tailored to specific practical WSN deployments. Additionally, EPD policies may be augmented with efficient backoff strategies to reduce contention and prevent packet collisions.

REFERENCES

[1] A. Burns, J. Harbin, L. Indrusiak, I. Bate, R. Davis, and D. Griffin, “AirTight: A Resilient Wireless Communication Protocol for Mixed-Criticality Systems,” in *2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2018, pp. 65–75. doi: 10.1109/RTCSA.2018.00017.

[2] A. Burns, J. Harbin, R. Davis, L. Indrusiak, I. Bate, and D. Griffin, “The AirTight Protocol for Mixed Criticality Wireless CPS,” *ACM Transactions on Cyber-Physical Systems*, vol. 4, Sep. 2019, doi: 10.1145/3362987.

[3] L. Zhang, Y. Yu, F. Huang, Q. Song, L. Guo, and S. Wang, “Deadline-aware adaptive packet scheduling and transmission in cooperative wireless networks,” in *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, 2014, pp. 1980–1984. doi: 10.1109/PIMRC.2014.7136496.

[4] T. Zhang, T. Gong, M. Lyu, N. Guan, S. Han, and X. S. Hu, “Reliable Dynamic Packet Scheduling With Slot Sharing for Real-Time Wireless Networks,” *IEEE Transactions on Mobile Computing*, vol. 22, no. 11, pp. 6723–6741, 2023, doi: 10.1109/TMC.2022.3196922.

[5] N. Nasser, L. Karim, and T. Taleb, “Dynamic Multilevel Priority Packet Scheduling Scheme for Wireless Sensor Network,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1448–1459, 2013, doi: 10.1109/TWC.2013.021213.111410.

[6] X. Chen, X. Shi, and Y. Wang, “Packet Scheduling Algorithm Based on Priority Adjustment in Wireless Sensor Networks,” in *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2022, pp. 1632–1639. doi: 10.1109/ITAIC54216.2022.9836768.

[7] A. Gujarati, F. Cerqueira, and B. B. Brandenburg, “Multiprocessor real-time scheduling with arbitrary processor affinities: from practice to theory,” *Real-Time Systems*, vol. 51, pp. 440–483, 2014.

[8] T. Watteyne, M. R. Palattella, and L. A. Grieco, “Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement,” Internet Engineering Task Force, Request for Comments RFC 7554, May 2015. doi: 10.17487/RFC7554.

[9] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, “Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH,” Nov. 2015. doi: 10.1145/2809695.2809714.

[10] S. Kim, H. Kim, and C. Kim, “ALICE: autonomous link-based cell scheduling for TSCH,” Apr. 2019. doi: 10.1145/3302506.3310394.

[11] R. Alexander *et al.*, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” Internet Engineering Task Force, Request for Comments RFC 6550, Mar. 2012. doi: 10.17487/RFC6550.

[12] P. Baldwin, S. Kohli, E. A. Lee, and X. Liu, “VisualSense: Visual Modeling for Wireless and Sensor Network Systems,” University of California, Berkeley, UCB/ERL M04/08, Apr. 2004. [Online]. Available: http://chess.eecs.berkeley.edu/pubs/759.html

[13] P. Baldwin, S. Kohli, E. A. Lee, X. Liu, and Y. Zhao, “Modeling of Sensor Nets in Ptolemy II,” in

Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, in IPSN '04. New York, NY, USA: Association for Computing Machinery, 2004, pp. 359–368. doi: 10.1145/984622.984675.

[14] C. Ptolemaeus, Ed., *System Design, Modeling, and Simulation using Ptolemy II*. Ptolemy.org, 2014. [Online]. Available: http://ptolemy.org/books/Systems

[15] C. Brooks, E. A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, and H. Zheng, "Heterogeneous Concurrent Modeling and Design in Java (Volume 1: Introduction to Ptolemy II)," EECS Department, University of California, Berkeley, UCB/EECS-2008-28, Apr. 2008. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-28.html