2ND WORKSHOP
“MACHINE LEARNING & NETWORKING” (MaLeNe)
PROCEEDINGS

SEPTEMBER 4, 2023

CO-LOCATED WITH
THE 5TH INTERNATIONAL CONFERENCE ON NETWORKED SYSTEMS (NETSYS 2023)
POTSDAM, GERMANY

Adapting to the Flow: Reinforcement Learning for Dynamic Priority Assignment in TSN

Nurefşan Sertbaş Bülbül and Mathias Fischer
University of Hamburg, Germany
Email:{sertbas, mfischer}@informatik.uni-hamburg.de

***Abstract*—Real-time systems employ prioritization schemes to accommodate different traffic classes with specific quality of service (QoS) requirements. However, in some scenarios where numerous high-priority packages are transmitted, lower-priority packages may fail to meet their deadlines, leading to a significant decline in scheduling performance. Sending high-priority flows excessively early does not provide any additional benefits beyond meeting the deadline. Instead, it is more effective to utilize this buffer time for lower-priority traffic and ensure on-time transmission of high-priority traffic. We propose an adaptive dynamic priority assignment scheme that utilizes reinforcement learning (RL) to address this issue. This enables adaptation to changing network conditions and continual improvement in performance over time. Additionally, we present and discuss two potential configuration candidates that can be utilized within the proposed scheme.**

***Index Terms*—dynamic priority, reinforcement learning, time-sensitive networks, deadline**

I. Introduction

The Internet of Things (IoT) and time-sensitive networking (TSN), i.e., real-time communication protocols, are enablers for future mission-critical systems and respective applications that require bounded latency as well as seamless and fault-tolerant communication. In this context, the delivery of traffic within a specific timeframe, referred to as the deadline, is significant, with potentially severe consequences when packets are late. In certain critical applications like robotics, any failure to meet these deadlines can have catastrophic consequences. The respective traffic has a hard deadline. However, numerous applications operate with soft deadlines, e.g., best-effort traffic, meaning they can tolerate occasional misses without experiencing a significant decline in performance or output quality. For instance, in video streaming, a few late packets do not have a significant impact on the overall video quality, e.g., the codec contains enough redundancy to recover. In such cases, the system can be more flexible in meeting the deadlines, focusing on providing the best possible output while still striving to adhere to the specified deadline.

The IEEE task group has proposed the TSN standards for providing such a QoS for each traffic class within the same network by describing different handling mechanisms for varying traffic requirements. It uses eight priorities to distinguish traffic classes, representing which traffic is more important with hard real-time constraints. The TSN mechanisms handle traffic on switches based on these priority values.

Although the prioritization mechanism enables the coexistence of different traffic classes, when a large number of high-priority packages are sent, it can lead to lower-priority packages failing to meet their deadlines. Despite having a lower priority, these packages still have their deadlines. This becomes particularly unfortunate when high-priority packages reach their destination much earlier than they are supposed to arrive. From the application perspective, it does not make a huge difference to receive a packet shortly before its deadline or well in advance of it. Thus, instead of sending high-priority traffic early on, it might be better for all flows in the network to send high and low-priority traffic on time. To address that, we propose dynamic priority handling that reassigns packet priorities so that lower-priority packets can make up time and do not arrive late. This can increase the utilization of network resources and decrease the number of late packets.

The authors of [1] have proposed an elastic queuing structure to avoid frame drops due to queue size limitations and with the help of the underlying hardware. However, this is an additional dependency and also increases the hardware requirements. Meng et al. [2] have proposed to use the Fuzzy Analytic Hierarchy Process (FAHP) to compute the priority of packets considering energy consumption, running time, and deadline. Then, a heapsort-based dynamic sorting algorithm selects the optimal scheduling subset from the task set using the new priorities. Even though results promise to reduce the deadline miss rate, it is not directly applicable to TSN with design criteria like energy consumption.

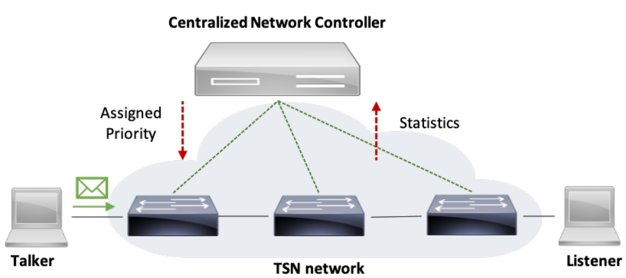
Unlike traditional approaches, RL offers several advantages for dynamic real-time networks as it can adapt and learn from experience in changing environments. In dynamic real-time networks, RL can continuously update its policies and make near-optimal decisions based on the current network state. This autonomous optimization capability allows RL to adapt to changing network conditions and improves network performance over time without human intervention. Moreover, the capabilities of RL to learn from interactions and effectively represent complex relationships position it as a highly suitable approach for addressing challenges encountered in dynamic network environments. Thus, RL has also been used in TSN with different goals, such as finding a routing path [3] or configuring per-hop latency guarantees [4]. Several studies leverage RL for dynamic priority assignment problems in the context of real-time environments [5]. However, the main objective of these studies is to determine a schedulable priority assignment that can accommodate a greater number of flows in the network. Instead, we aim to use resources more efficiently while accommodating, at least, the same number of flows but providing better QoS satisfaction regarding deadlines. This problem becomes more obvious in the case of imbalanced traffic classes, where certain classes may become overloaded while there is an available capacity for other traffic classes.

Accordingly, this paper outlines our roadmap for utilizing reinforcement learning for dynamic priority assignments in time-sensitive networks. Our main goal is to reduce the number of missed flow deadlines resulting from inefficient resource utilization. To achieve this objective, we propose two configuration schemes based on TSN standards: centralized and distributed. We examine these schemes and discuss their limitations in meeting near real-time requirements and providing strict QoS guarantees, considering the constraints imposed by the time-sensitive environment.

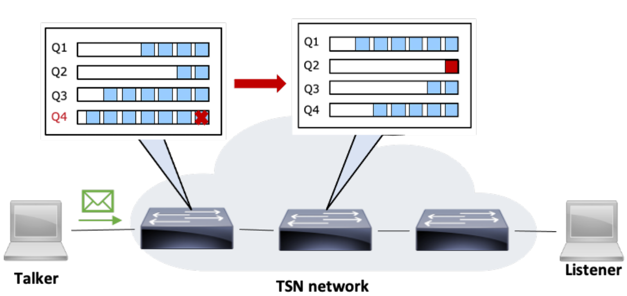
II. Dynamic Priority Assignment with Reinforcement Learning

To leverage the advantages of reinforcement learning, we present two potential configuration schemes in Figure 1 that can be applied to time-sensitive networks:

*a) **Centralized Scheme**:* In a centralized scheme, as illustrated in Figure 1a, it is assumed that the centralized network controller (CNC) has a global network view and collects statistics such as queue waiting time and queue utilization. The RL agent is deployed on top of the CNC so that it receives real-time network data and can utilize RL algorithms to learn and determine global network policies. Since it perceives the network as a whole, this scheme enables coordinated decision-making and optimization across multiple network elements. Thus, CNC can generate *best* policies for the dynamic priority assignment based on a centrally deployed RL agent.



(a) Centralized scheme



(b) Distibuted scheme

Fig. 1: RL-based dynamic priority assignment scheme.

Since there is no policy or apriori information about the network initially, CNC can monitor the current assignments and train the RL agent. In other words, CNC can get hop-by-hop statistics and merge them to compute a final reward value for the action, e.g., the current priority of the packet. Here, considering the QoS requirements of the packets, the reward can get negative values as well, e.,g., a penalty value. During this training (exploration) time, RL can assign random priorities to packets or leave them as they are. After the pre-training time, CNC can benefit from the developed RL policy to reassign a new priority to the packets dynamically on the runtime.

*b) **Distributed Scheme**:* In a distributed scheme, as illustrated in Figure 1b, RL agents are directly deployed at network nodes, e.g., at TSN switches. These agents locally monitor the network, collect real-time data, and learn optimal decision-making policies based on the observed conditions. Each switch aims to compensate for a potential latency in the previous hop by dynamically re-assigning packet priorities determined by RL. Thus, each switch has its own RL agent to develop a policy. It may also be possible to benefit from the *transfer reinforcement learning* concept. Switches may collaboratively help each other to develop the best assignment policy.

However, without a centralized controller, the problem gets harder. Now, the switch has to assess, based on limited knowledge, whether the packet will miss its deadline and needs to get reprioritized. For that, it may need to know the topology or path the packet will be routed. Also, the packet must be marked as a late packet to be handled differently at the next hop switch. Thus, there is a need for a communication protocol and a local or distributed algorithm between switches to address these points.

III. Conclusion

In conclusion, the proposed adaptive dynamic priority assignment scheme leveraging reinforcement learning presents a promising solution for future networks. It can dynamically reconfigure the priorities of existing packets and offers a practical approach to decreasing missed flow deadlines. By adaptively adjusting priorities based on real-time conditions, it can effectively manage resource allocation and meet stringent QoS requirements.

References

[1] A. G. Mariño, F. Fons, A. Gharba, L. Ming, and J. M. M. Arostegui, “Elastic queueing engine for time sensitive networking,” in *IEEE 93rd Vehicular Technology Conference*, 2021, pp. 1–7.

[2] S. Meng, Q. Zhu, and F. Xia, “Improvement of the dynamic priority scheduling algorithm based on a heapsort,” *IEEE Access*, vol. 7, pp. 68 503–68 510, 2019.

[3] N. S. Bülbül and M. Fischer, “Reinforcement Learning assisted Routing for Time Sensitive Networks,” in *IEEE GLOBECOM Global Communications Conference*, 2022, pp. 3863–3868.

[4] A. Grigorjew, M. Seufert, N. Wehner, J. Hofmann, and T. Hoßfeld, “ML-Assisted Latency Assignments in Time-Sensitive Networking,” in *IFIP/IEEE International Symposium on Integrated Network Management*, 2021, pp. 116–124.

[5] H. Lee, J. Lee, I. Yeom, and H. Woo, “Panda: Reinforcement learning-based priority assignment for multi-processor real-time scheduling,” *IEEE Access*, pp. 185 570–185 583, 2020.