LETRA: Mapping Legacy Ethernet-Based Traffic into TSN Traffic Classes

Daniel Bujosa Mateu*, Mohammad Ashjaei*, Alessandro V. Papadopoulos*, Julian Proenza‡, Thomas Nolte*
*Mälardalen University, Västerås, Sweden
‡ University of the Balearic Islands, Palma, Spain

***Abstract*—This paper proposes a method to efficiently map the legacy Ethernet-based traffic into Time Sensitive Networking (TSN) traffic classes considering different traffic characteristics. Traffic mapping is one of the essential steps for industries to gradually move towards TSN, which in turn significantly mitigates the management complexity of industrial communication systems. In this paper, we first identify the legacy Ethernet traffic characteristics and properties. Based on the legacy traffic characteristics we presented a mapping methodology to map them into different TSN traffic classes. We implemented the mapping method as a tool, named Legacy Ethernet-based Traffic Mapping Tool or LETRA, together with a TSN traffic scheduling and performed a set of evaluations on different synthetic networks. The results show that the proposed mapping method obtains up to 90% improvement in the schedulability ratio of the traffic compared to an intuitive mapping method on a multi-switch network architecture.**

***Index Terms*—TSN, traffic mapping, Ethernet message characteristics, legacy Ethernet networks**

I. Introduction

New technologies often offer new solutions or improvements for companies that can lead to an advantage over the competitors; reducing costs or offering a better product, or lead to an environmental improvement; improving performance, optimization of resources, and/or emission reduction. However, new opportunities imply new challenges. These challenges are often related to their integration with existing legacy solutions and their implementation. For many industrial applications, it may not be cost-effective to adopt the new technologies, as it may require redeveloping all previous solutions and systems.

One of the new technologies, which can change the current paradigm of industrial communications and seems to be a key to the transition to Industry 4.0, is Time-Sensitive Networking (TSN). Everything started when, in 2005, the IEEE Audio-Video Bridging (AVB) Task Group (TG) was created. The purpose was to provide Ethernet with soft real-time capabilities oriented to audio/video streaming. The AVB TG developed three projects: (i) the IEEE Std 802.1AS [1], dedicated to clock synchronization, (ii) the IEEE Std 802.1Qav, which standardized the Credit-Based Shaper (CBS) [2]; and, finally, the IEEE Std 802.1Qat, which standardized the Stream Reservation Protocol (SRP) [3]. Additionally, another profile with series of rules, called IEEE Std 802.1BA-2011: Audio Video Bridging Systems [4], was created to ensure a minimum QoS when using the aforementioned standards. These standards together are commonly referred to as AVB standards. Over time, areas of applications, such as automotive [5], automation [6] and energy distribution [7], were interested in the work done by the TG so in 2012 the group was renamed to TSN TG and its objective was expanded to meet the needs of these new applications. The set of standards developed by the TG is usually referred to as TSN standards and it presents several interesting features. Specifically, TSN seems to provide Ethernet with proper support for mixed hard and soft real-time communications, flexibility of the traffic requirements and fault tolerance mechanisms. For these characteristics, TSN seems promising to enable new solutions within the context of modern industrial systems and solutions enabling, among other things, the integration of multiple legacy networks onto one TSN network. However, current TSN networks do not support all Ethernet-based legacy system message implementation characteristics, such as jitter in some legacy network devices, while currently used legacy technologies do not meet all TSN requirements. Moreover, it is cost-effective and beneficial for companies if they gradually move towards new technologies instead of completely replacing existing ones. Therefore, solutions to integrate a legacy system into a TSN network are essential so that services are not disturbed. An example of that is when a network consists of 4 nodes connected via different protocols, would be able to replace the end-to-end links with a TSN switch connected to all 4 nodes. This would allow the communications between networks while letting previous communications benefit from TSN features hence improving their real-time behavior, synchronization, and fault tolerance, to name a few.

Contributions: To allow industry to adopt TSN solutions, and the desire integration, a proper migration methodology of the legacy Ethernet-based traffic to TSN traffic classes should be designed. In this work we propose three steps to achieve this goal, as follows:

1) We develop a Legacy Ethernet-based Traffic model that can describe messages of any Ethernet-based communication protocols. Moreover, apart from the model, we require a methodology to identify the parameters of the messages inherited from the legacy system. The identification methodology is out of the scope of this paper. However, some works like [8] have already addressed this issue.
2) We develop a mapping methodology, and its corresponding implementation as a mapping tool, named Legacy

Ethernet-based Traffic Mapping Tool or LETRA, that can map the legacy Ethernet-based messages characterized by the proposed model into different TSN traffic classes. To the best of our knowledge, this is the first attempt to map the Ethernet-based legacy traffic into TSN traffic classes considering a full spectrum of message characteristics. To evaluate the mapping methodology we implemented it as a mapping tool and compared its performance with an intuitive mapping methodology on different networks.

3) We integrated a pre-existing TSN scheduling method and a TSN schedulability analysis method into LETRA, which can map the messages, schedule the TSN traffic and evaluate their real-time behavior.

Paper outline: The paper is organized as follows. Section II presents the related work. Section III presents the legacy Ethernet-based traffic model. Section IV describes the background of TSN and TSN traffic classes. Then, Section V proposes the traffic mapping methodology and development of LETRA, while Section VI presents the experiments and evaluations. Finally, Section VII concludes the paper and gives future directions.

II. Related work

Due to the great relevance of the work done by the TSN TG since 2012, the community has carried out a significant amount of work related to their study, application, and improvement. For example, the work in [9] studied the effects of the time-aware shaper, the work in [10] analyzed the fault tolerance issues, while the work in [11] proposed time redundancy to tolerate temporary faults, the work in [12] studied the scheduling policies and the load balancing was studied in [13]. Moreover, the work in [14] provided an up-to-date comprehensive survey of the TSN-related research.

Within the context of TSN traffic mapping, the work in [8] presented a network monitoring method to obtain the traffic properties based on measurements. A meta-heuristic method is proposed in [15] that maps mixed-criticality applications into the TSN traffic classes. Although the aim is similar to this paper, the proposed method does not cover all cases that are studied and exist in industrial applications. The method, thus, becomes suitable for cases where only very few mixed-criticality levels are assumed in the legacy system with no extensive timing information, whereas in this paper we consider different traffic characteristics including many timing characteristics and constraints when we map them into the TSN traffic classes.

There are also few works on integrating legacy networks into TSN networks. For example, the work in [16] integrated a few of the TSN standards into Sercos III, which is a closed system that allows standard Ethernet devices to be plugged, to improve its performance. Moreover, an integration methodology of wireless TSN (802.11) was proposed in [17]. Finally, an integration that focuses on the clock synchronization for EtherCAT and TSN was proposed in [18].

Nevertheless, to be the best of our knowledge, the proposed mapping methodology and the tool LETRA in this paper is the first attempt to map messages from any Ethernet-based legacy network into TSN traffic classes considering a full spectrum of message characteristics.

III. Legacy Ethernet-based traffic model

This section introduces a message model to describe the legacy Ethernet-based messages. To handle legacy messages from different Ethernet-based protocols, we identified a set of characteristics with which we can model the Ethernet-based messages. The model is used by the mapping tool to map the messages into different TSN traffic classes. Note that the extraction of values in the model is out of the scope of this paper, which can be done by measurements in the legacy networks as described in [8].

The parameters used to characterize the Ethernet messages are divided into three categories, including: *common parameters*, *periodic message parameters* and *non-periodic message parameters*. Note that not all messages need to have all parameters to map them. Following is the description of the parameters.

1) Common parameters: The common parameters are those that are independent of the behavior of the messages that are presented by $\{S, D, ML, DL, LRT, FLR, Prec\}$. In the above set, S and D represent the source and destination(s) of the message. The length of the message in bytes is denoted by ML which can be a range of sizes to consider variable message sizes. The message deadline is shown by DL. Moreover, LRT shows if the message is soft or hard real-time, i.e., missing deadlines lead to a degradation of functionality or a complete system failure, respectively. FLR denotes the message lost percentage, and $Prec$ denotes the precedence constraints between two or more messages which identifies if some messages should be transmitted or received in a specific sequential order.

2) Periodic message parameters: The periodic message parameters are those dependent on the periodicity of the messages, that is presented by $\{P, O, JI, JO\}$. P are the message period, while O is the offset, i.e., time shift of the message release time. JI represents the maximum jitter on the message release, while JO is the maximum jitter in the message reception. Note that jitter is the variation of delays that can be on transmission and/or reception of the message.

3) Non-periodic message parameters: The main parameter for non-periodic messages is the minimum inter-arrival time, identified by MIT, which is the minimum time between two consecutive releases of a message in non-periodic messages.

Figure 1 illustrates a graphical representation of some of the presented parameters in the model.

IV. TSN traffic characteristics

This section gives a brief background about the TSN standards and TSN traffic classes. Communication in a TSN network is done among end-stations through routes of links and switches through Ethernet messages. A port in a TSN

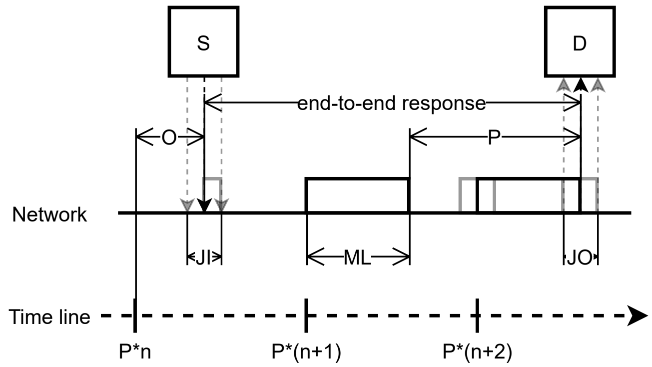


Fig. 1: Graphical representation of the presented model.

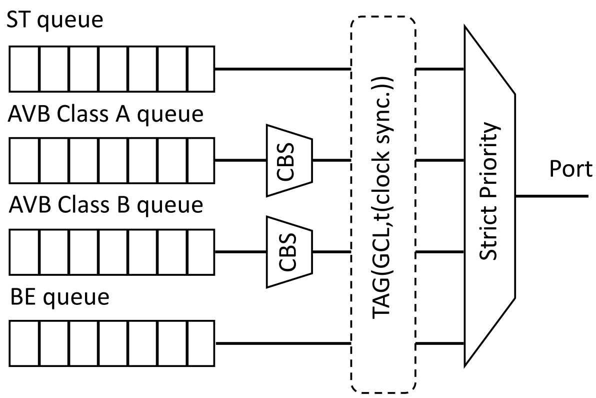


Fig. 2: A TSN egress port.

switch supports eight FIFO queues. A typical port with four TSN traffic classes (queues) is shown in Fig. 2. The TSN standards define three traffic classes including Scheduled Traffic (ST), Audio-Video Bridging (AVB) traffic, and Best Effort (BE) traffic. The AVB traffic is named by classes A and B, where class A has higher priority than class B. Following, we describe the details of TSN traffic classes.

A. ST traffic class

The ST traffic is scheduled offline, which makes them fully deterministic with zero jitter in the message delivery. The TSN standard [19] defines a Time-Aware Gates (TAG) that is controlled by a Gate Control List (GCL). The GCL specifies at which specific time of the network-wide reference time gates are open, and thus, the link is available for a queue to send messages. Note that the reference clock is achieved thanks to the synchronization protocol defined in [1] which allows clock synchronization between end-stations and switches.

B. AVB traffic class

The AVB TG [2] introduced the CBS that defines credits to AVB queues. The credit is consumed when a message in that queue is sent, otherwise, it is replenished when there is a pending message in the queue (or the credit is still negative). The AVB queues can only transmit when their credit is positive or zero and their gate is open according to the GCL. CBS defines priority classes (classes A and B) but allowing transmission of low-priority traffic even if high-priority traffic is waiting according to their credits. This reduces buffering and improves lower-priority traffic QoS. Even though the activation time of AVB traffic is unknown due to possible blocking from other AVB classes or ST queues, there are analysis methods to calculate their worst-case response time. For example, the analysis used in the experiments of this paper is the one presented in [20].

C. BE traffic class

BE has no real-time guarantees and is the lowest priority. This queue is not shaped by CBS and can only be sent if its gate is open and all other AVB queues have negative credit or there is no AVB traffic ready for transmission.

V. Proposed traffic mapping methodology

To map the legacy Ethernet-based traffic characterized by the parameters modeled in Section III into the TSN traffic classes, we developed three logic-based equations explained in the following sub-sections. As the equations are logic-based, in this section all parameters are treated as logical Boolean variables. In this sense, a parameter is True if the frame is affected by the parameter, otherwise False. For example, if one frame is non-periodic and has deadline of $100ms$, as $P = NULL$, $P = 0$ in the equation. On the other hand, as $DL = 100ms$, $DL = 1$ in the equation.

A. Mapping to the ST traffic class

The mapping first checks whether the legacy Ethernet traffic should be mapped into the ST traffic, according to the following expression:

$$ST = P \ \& \ (JO \parallel (!JI \ \& \ DL)) \tag{1}$$

where $\&$ is the logical "and" operator, $\parallel$ is the logical "or" operator, and ! is the logical "not" operator. First of all, Eq. (1) checks whether the message is periodic (P). This is mandatory as, otherwise, it would be impossible to implement a proper GCL. That is because, as described in Section IV, GCL is a list of open/close instructions executed repeatedly at certain times. Therefore, even if it is possible to schedule the message instances offline it would imply generating a long GCL table that makes the schedule impractical. Secondly, Eq. (1) checks if the message has JO constraints, and it is periodic, it must be transmitted as ST traffic as it is the only way to ensure meeting the JO requirement. That is because, as mentioned in Section IV, ST traffic is the only fully deterministic TSN traffic with zero JO. However, if the message has no JO requirements but it has DL and no JI then it can be also sent as an ST message. The reason for having D is to benefit from the ST characteristics while the JI conditions are to prevent waste of resources. JI implies considering larger open gates for the ST traffic to ensure the message instance to be transmitted within that time. This intuitively means allocating more bandwidth, which can be a waste of network bandwidth resources. Note that this happens only with JI and not with JO as once the message reaches the first TSN switch the messages will be sent just when the window starts.

B. Mapping to the AVB traffic class

Eq. (2) indicates whether the message can be sent as an AVB message. First, the message should have DL constraints to benefit from being sent as an AVB message. Moreover, the message must not have JO constraints unless it is not a hard real-time message. If the message has JO constraints but it is not hard real-time, another type of analysis, such as utilization-based analysis [15] may be needed. Note that Eq. (2) only specifies if the message can be transmitted as AVB traffic but it says nothing about the AVB possible priority classes. In this work, we consider only one AVB class queue for mapping as currently LETRA only identifies the messages as suitable or not for each traffic class. Later, it will be integrated or improved through constraint programming and/or metaheuristics to reach the desired specification level. In Eq. (2), HRT indicates the hard real-time requirement for the message.

$$AVB = DL \ \& \ (!JO \parallel !HRT) = DL \ \& \ !(JO \ \& \ HRT) \quad (2)$$

C. Mapping to the BE traffic class

Eq. (3) indicates whether the message can be sent as a BE class message. It checks whether the message has real-time requirements or not, which is the only requirement to be in the BE class.

$$BE = !JO \ \& \ !DL = !(JO \parallel DL) \quad (3)$$

D. Resulting truth table

As a result of the presented mapping methodology, we summarized the equations with a truth table shown in Table I. As it can be seen, we just use P, JI, JO, DL, and LRT to map legacy Ethernet messages. The other parameters in the model do not affect the mapping but on the scheduling and analysis of the traffic after the mapping is performed to verify the timing properties.

To show the performance of the proposed mapping methodology we also define an intuitive mapping methodology. The intuitive mapping methodology classifies all periodic messages as ST traffic class and all non-periodic messages as AVB traffic class to still have a level of timing guarantee for them.

Note that the presented mapping methodology cannot be compared with the mapping presented in [15], which is only based on the criticality level of messages. The reason is that we consider more specific variables to map the messages, which means that [15] cannot map 90% of the messages considered in this work. However, [15] maps messages between ST class and AVB class which, according to our tool, are suitable for both classes. In this sense, the combination of both tools would expand the number of mappable messages, while resolving some ambiguities in LETRA.

P	JI	JO	DL	HRT	ST	AVB	BE
0	X	X	0	0	0	0	1
0	X	X	0	1	0	0	1
0	X	X	1	0	0	1	0
0	X	X	1	1	0	1	0
1	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1
1	0	0	1	0	1	1	0
1	0	0	1	1	1	1	0
1	0	1	0	0	1	0	0
1	0	1	0	1	1	0	0
1	0	1	1	0	1	1	0
1	0	1	1	1	1	0	0
1	1	0	0	0	0	0	1
1	1	0	0	1	0	0	1
1	1	0	1	0	0	1	0
1	1	0	1	1	0	1	0
1	1	1	0	0	1	0	0
1	1	1	0	1	1	0	0
1	1	1	1	0	1	1	0
1	1	1	1	1	1	0	0

TABLE I: Truth table of the mapping methodology.

E. Evaluation tool

To have a complete evaluation tool, we implemented LETRA, a TSN traffic scheduling, and a schedulability analysis to be able to evaluate the solution. Moreover, we developed a network generator and an intuitive mapping tool which implements the intuitive mapping methodology described in Section V. The tools that are used for this evaluation from previous works include the ST scheduling tool in [21] and the AVB traffic schedulability analysis in [20].

The integration of the mentioned tools is shown in Fig. 3. First, the Network Generator generates the network messages according to the network configuration specified in the presented model. The generated messages are used as inputs for LETRA and the intuitive mapping tool, thus, obtaining two different classifications for the messages. The ST messages of both mapping tools are scheduled through the ST traffic scheduler [21] and, finally, the AVB traffic is checked through the AVB analyzer [20], which checks whether the AVB traffic are schedulable considering the transmission algorithms based on a response time analysis. The results of the scheduler and the AVB analysis are used to compare the mapping performance.

The Network Generator tool uses the parameters in the model explained in Section III as inputs to generate the messages randomly. Besides the parameters in the model, the network topology is an input.

VI. Experiments and results

This section presents the experiments that we conducted to evaluate the proposed mapping methodology using the developed integrated tools (Fig. 3). We first present the experimental setup and then we illustrate and discuss the results.

A. Experimental setup

For the evaluation in this paper, we considered two network architectures, including a single-switch and a three-switch

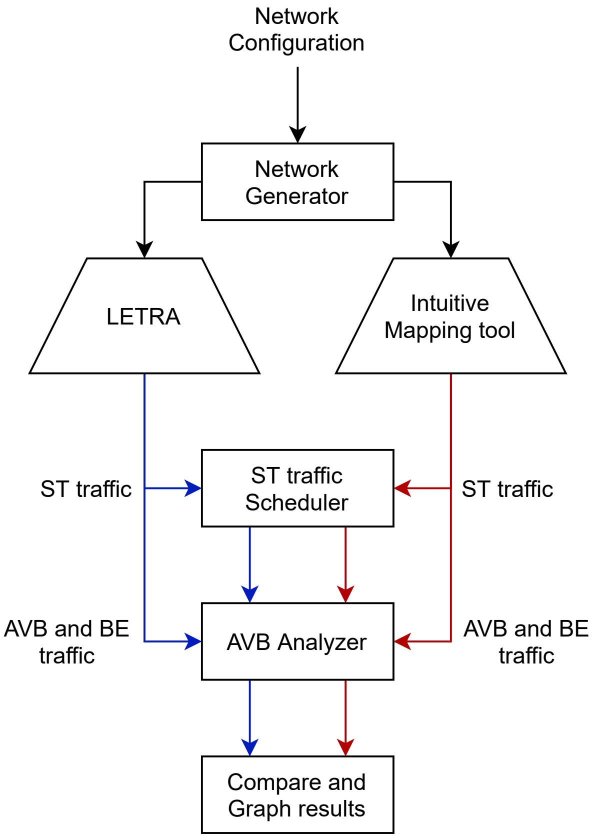
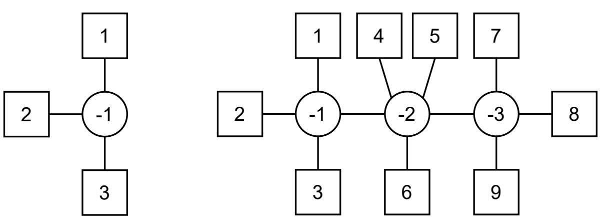


Fig. 3: Integration of the tools for evaluation.



(a) A single-switch network. (b) A three-switches network.

Fig. 4: Experimental network architectures.

network. The topology is a line-star topology with the switches connected in a line and nodes connected to the switches in the form of stars, as shown in Fig. 4. The line topology, apart from being widely used in the industry in many layers of the automation pyramid, is simpler and share many similarities with a tree topology. This allows us to extend the results of these experiments to a greater number of communication networks that could be developed in the future.

The network generator is designed such that we can select the input probabilities to uniformly distribute the probability among all possible messages, which is listed in Table I. To achieve that we set the probability of all parameters to 50% except for periodicity. This means, as an example, that the messages have a 50% chance to have deadline constraints. With this setup, we ensure that all possible combinations with the same probability will be generated.

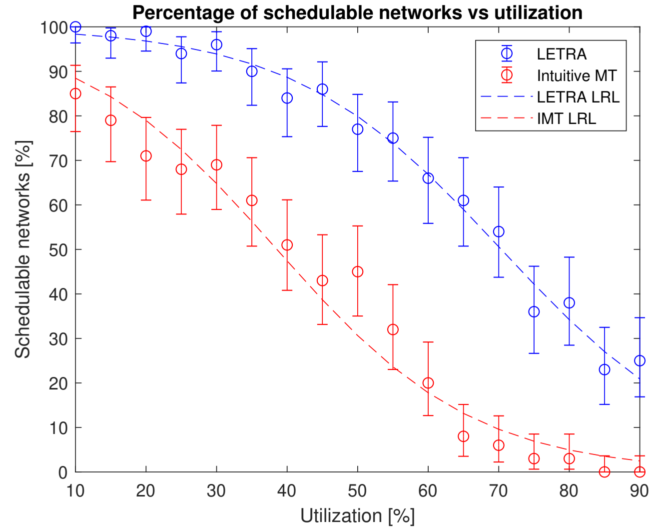


Fig. 5: Percentage of schedulable networks with respect to the bandwidth utilization for the single-switch network where the schedulability percentage is represented by a circle with error bars and the dashed curve corresponds to its logistical regression tendency line (LRL).

The parameters to generate the messages are set as follows. The network bandwidth is set to 10Mbps to prevent generating too many messages in case of generating a high load. The maximum link utilization is varied within the range [10%,90%] with the interval of 5%. Note that the messages will be generated such that the utilization in all links will be the one selected as an input, i.e., when we select 10% utilization the message generator selects the message sizes and routes to obtain 10% on all links. The maximum number of generated messages is set to 100, however, depending on the selected link utilization the message number can be different. The message length is selected within the range [64,1530] Bytes. The maximum allowed period and minimum inter-arrival time for messages are set to $1000\mu s$. We also allowed arbitrary deadlines, which can be selected within the range $[500,1000]\mu s$. The input and output jitter values are also selected within the range $[1,100]\mu s$. We generated 100 networks for each link utilization, e.g., 100 networks with the load of 10% on links, hence 1700 networks are generated for each network architecture shown in Fig. 4.

The next sub-sections present the results of the experiments. We compare the results of mapping the generated messages with LETRA and an intuitive mapping tool based on three different variables, including the link utilization, the number of messages, and the time it takes to schedule the ST messages.

B. Results of the single-switch network

Fig. 5 shows the percentage of schedulable networks with respect to the utilization. The horizontal axis shows the utilization of the generated networks, while the vertical axis shows the percentage of networks that are schedulable with

Mean IMP [%]	Min IMP [%]	Max IMP [%]
86.65	37.52	149.47

TABLE II: Performance improvements with respect to the percentage of bandwidth utilization for the single-switch network.

Mean IMP [%]	Min IMP [%]	Max IMP [%]
77.57	35.93	120.85

TABLE III: Performance improvements with respect to number of messages for the single-switch network.

two different traffic mapping tools. The circles in the figure are the schedulable percentage of generated networks with specific network utilization and the error bars are calculated through the binomial analysis with 95% certainty. In addition, the dashed lines show the trend of the data as logistic regression. As it can be seen, LETRA results in more schedulable networks in all generated utilization compared to the intuitive mapping tool. For instance, when we generated traffic with 90% utilization on all links, which is a very high network utilization, LETRA gives just below 30% of the networks schedulable, whereas the intuitive mapping results in very few schedulable networks. Table II shows the mean improvement by using LETRA compared to the intuitive mapping tool while Min IMP and Max IMP are the maximum and minimum possible improvements due to the result errors. LETRA results in 86.65% more schedulable networks compared to the intuitive mapping on average for the single-switch network architecture.

Fig. 6 shows the percentage of schedulable networks by varying the number of messages. Similar to the previous results, the vertical axis is the percentage of schedulable networks, and the horizontal axis is the number of generated messages. The figure also shows the error bars calculated through the binomial analysis with 95 % certainty and the tendency line calculated through logistic regression. Again, LETRA shows a significant improvement in the schedulability of networks compared to the intuitive mapping tool. For example, the intuitive mapping tool cannot schedule the networks when the number of messages is more than 35, however, LETRA can schedule a few of the generated networks up to 45 messages in the single-switch network. Table III shows the mean improvement between the two mapping tools, where it shows that on average 77.75% improvement by using LETRA.

Fig. 7 illustrates the time that it takes to schedule the ST messages in a generated message after mapping. In the figure, times are shown in ms and the error bars are calculated through the gamma distribution analysis with 95% certainty. We also used a linear trend line to show the overall trend of data. In this figure, the values present high variability and they do not follow any basic tendency curve due to the number of parameters that can affect the scheduling time, such as memory or CPU utilization, which could not be monitored during the execution of the experiments due to hardware limitations. However, it can show, in general, apart from having better performance, LETRA is also delivering the ST schedules faster compared to the intuitive mapping tool.

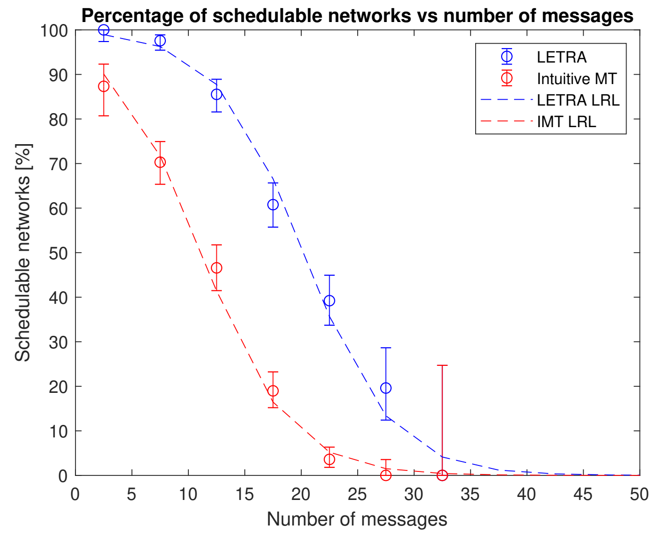


Fig. 6: Percentage of schedulable networks with respect to the number of messages for the single-switch network where the schedulability percentage is represented by a circle with error bars and the dashed curve corresponds to its logistical regression tendency line (LRL).

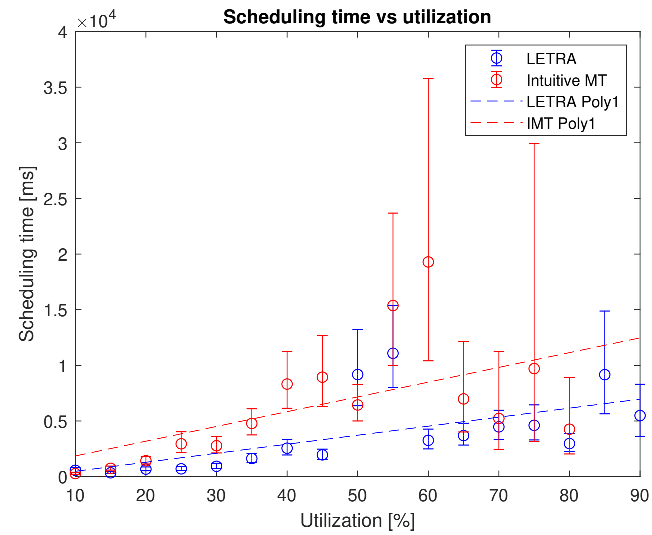


Fig. 7: Scheduling time with respect to bandwidth utilization for the single-switch network where the scheduling time is represented by a circle with error bars and the dashed line corresponds to a linear regression (Poly1).

The reason is that the intuitive mapping tool selects more messages to be ST messages, while LETRA decides based on many timing requirements which in general leads to less number of ST messages. This means that the legacy messages are not unnecessarily mapped into the ST class.

C. Results of the three-switch network

Fig. 8 shows the percentage of schedulable networks with respect to the bandwidth utilization for the three-switch net-

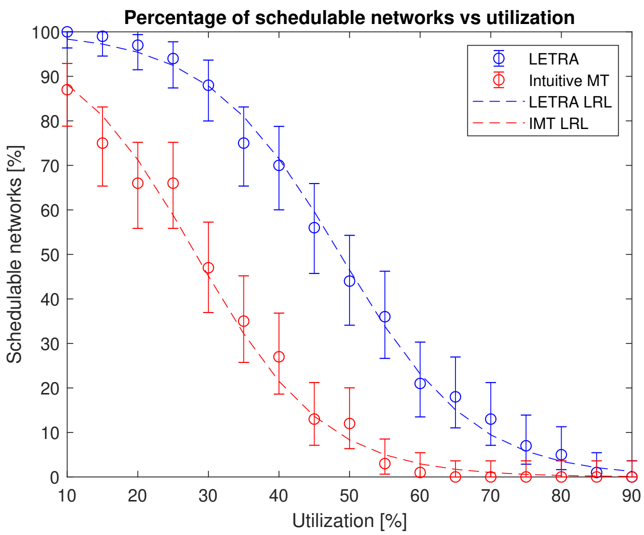


Fig. 8: Percentage of schedulable networks with respect to the bandwidth utilization for the three-switch network where the schedulability percentage is represented by a circle with error bars and the dashed curve corresponds to its logistical regression tendency line (LRL).

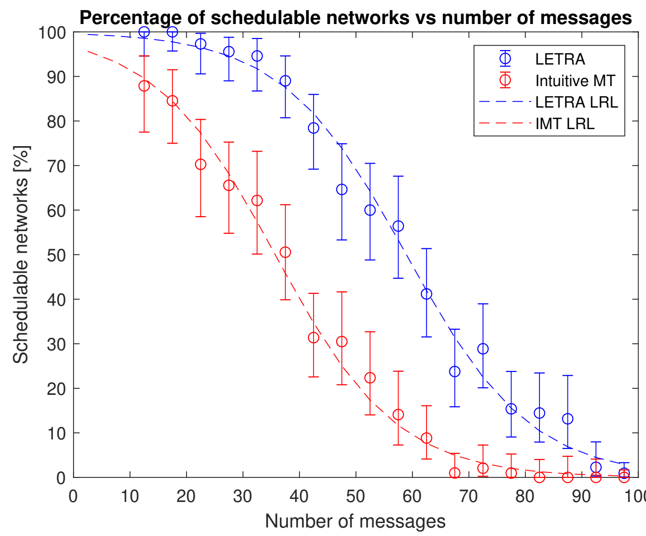


Fig. 9: Percentage of schedulable networks with respect to the number of messages for the three-switch network where the schedulability percentage is represented by a circle with error bars and the dashed curve corresponds to its logistical regression tendency line (LRL).

Mean IMP [%]	Min IMP [%]	Max IMP [%]
90.74	32.30	165.28

TABLE IV: Performance improvements with respect to the percentage of bandwidth utilization for the three-switch network.

work. Again, LETRA exhibits better performance compared to the intuitive mapping tool for all ranges of network utilization in larger networks. Table IV shows mean improvement, which is 90.74% better performance with LETRA on average compared to the intuitive mapping tool. Although the performance of both mapping tools decreases with the size of the network, according to the results, we can conclude with 95% certainty that the percentage of improvement remains constant with the size of the network.

Fig. 9 shows the percentage of schedulable networks with respect to the number of messages. LETRA exhibits better performance compared to the intuitive mapping tool for the entire range of the number of messages analyzed. Table V shows that the amount of improvement in the larger network is 83.41% on average when using LETRA to map the traffic.

Finally, Fig. 10 presents the time that it takes to schedule the ST messages after the mapping. Again, the results show that, besides LETRA having better performance, it is also faster in scheduling the ST messages compared to the intuitive mapping tool. Another interesting observation is that in the bigger network with high utilization, the ST messages can be scheduled within a reasonable time, while it is not the case if we map the traffic with the intuitive mapping tool where after 50% network utilization the ST messages cannot be scheduled. We believe that the reason is mainly due to the high amount of messages and utilization of the network.

Mean IMP [%]	Min IMP [%]	Max IMP [%]
83.41	27.19	157.84

TABLE V: Performance improvements with respect to the number of messages for the three-switch network.

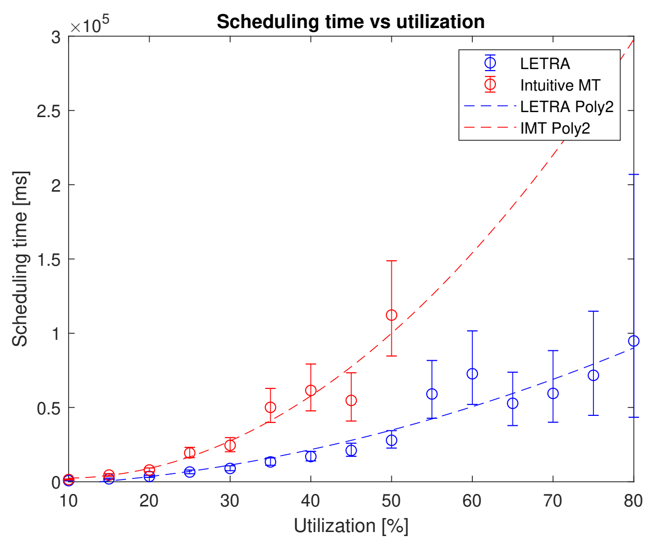


Fig. 10: Scheduling time with respect to the bandwidth utilization for the three-switch network where the scheduling time is represented by a circle with error bars and the dashed curve corresponds to a quadratic regression (Poly2).

VII. Conclusions and Future Work

We argued that one of the essential steps towards migrating from legacy Ethernet-based networks to TSN-based networks in industries is efficiently mapping the traffic into TSN traffic classes. Therefore, in this paper, we took a three-step strategy to achieve such a missing step. The steps include: (i) identifying the properties of legacy Ethernet-based messages by modeling them, (ii) map the Ethernet messages into different TSN classes, including ST, AVB, and BE classes, according to several timing properties, and (iii) develop a set of tools to evaluate the proposed mapping methodology, including the mapping tool, called LETRA, an ST scheduling tool and a schedulability analysis for the AVB messages. We also developed an intuitive mapping tool to show the performance of LETRA compared with that. We performed a set of experiments using two network architectures, being single-switch and three-switch architectures. We generated a set of messages randomly with a specific network utilization to show which mapping tool can result in more schedulable networks. The results show that in both network sizes LETRA performs much better, in concrete, 86.65% better performance in the smaller network and 90.74% in the larger network in average.

All these results were obtained for a specific network topology and traffic configuration. In future work, we plan to run similar evaluations for other kinds of networks and schedulers to better define the mapping criteria and their performance. Moreover, we want to integrate it with the scheduler so the messages that can be placed in different traffic classes according to the mapping criteria can be specified while running the scheduler. On the other hand, we also plan to continue working on the other steps of the legacy integration. In this sense, we plan to develop a formal standard legacy Ethernet-based traffic model and adapt the schedulers to those message characteristics.

Acknowledgment

This research is supported by the Swedish Knowledge Foundation (KKS) under the FIESTA project, the Swedish Governmental Agency for Innovation Systems (VINNOVA) under the DESTINE and PROVIDENT projects, the Spanish Agencia Estatal de Investigación (AEI) and the FEDER funding through grant TEC2015-70313-R (AEI/FEDER, UE).

In addition the authors would like to acknowledge Daniel Hallmans at Hitachi-ABB Power Grids, Ludvika, Sweden, for useful insights, discussions, and feedback concerning large industrial legacy Ethernet-based communication networks.

References

[1] "IEEE Standard for Local and Metropolitan Area Networks–Timing and Synchronization for Time-Sensitive Applications," *IEEE Std 802.1AS-2020 (Revision of IEEE Std 802.1AS-2011)*, pp. 1–421, 2020.

[2] "IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams," *IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005)*, pp. C1–72, 2010.

[3] "IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks Amendment 14: Stream Reservation Protocol (SRP)," *IEEE Std 802.1Qat-2010 (Revision of IEEE Std 802.1Q-2005)*, 2010.

[4] "IEEE Standard for Local and Metropolitan Area Networks–Audio Video Bridging (AVB) Systems," *IEEE Std 802.1BA-2011*, pp. 1–45, 2011.

[5] S. Samii and H. Zinner, "Level 5 by Layer 2: Time-Sensitive Networking for Autonomous Vehicles," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 62–68, 2018.

[6] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[7] R. Salazar, T. Godfrey, L. Winkel, N. Finn, C. Powell, B. Rolfe, and M. Seewald, "Utility Applications of Time Sensitive Networking White Paper (D3)," IEEE, Tech. Rep., 2018. [Online]. Available: https://mentor.ieee.org/802.24/dcn/18/24-18-0022-03-sgtg-utility-applications-of-time-sensitive-networking-white-paper-d3.docx

[8] M. Gutiérrez, W. Steiner, R. Dobrin, and S. Punnekkat, "Learning the parameters of periodic traffic based on network measurements," in *2015 IEEE International Workshop on Measurements & Networking (M&N)*. IEEE, 2015, pp. 1–6.

[9] G. Alderisi, G. Patti, and L. Lo Bello, "Introducing support for scheduled traffic over IEEE audio video bridging networks," in *Conf. Emerging Technologies Factory Automation*, 2013.

[10] S. Kehrer, O. Kleineberg, and D. Heffernan, "A comparison of fault-tolerance concepts for IEEE 802.1 Time Sensitive Networks (TSN)," in *IEEE Emerging Technology and Factory Automation*, 2014.

[11] I. Álvarez, I. Furió, J. Proenza, and M. Barranco, "Design and experimental evaluation of the proactive transmission of replicated frames mechanism over time-sensitive networking," *Sensors*, vol. 21, no. 3, p. 756, 2021.

[12] K. S. Umadevi and R. K. Sridharan, "Multilevel ingress scheduling policy for time sensitive networks," in *Int. Conf. Microelectronic Devices, Circuits and Systems*, 2017.

[13] F. A. R. Arif and T. S. Atia, "Load balancing routing in time-sensitive networks," in *Int. Scientific-Practical Conference Problems of Infocommunications Science and Technology*, 2016.

[14] M. Ashjaei, L. Lo Bello, M. Daneshtalab, G. Patti, S. Saponara, and S. Mubeen, "Time-sensitive networking in automotive embedded systems: State of the art and research opportunities," *Journal of Systems Architecture*, vol. 110, pp. 1–47, September 2021.

[15] V. Gavriluţ and P. Pop, "Traffic-type assignment for tsn-based mixed-criticality cyber-physical systems," *ACM Trans. Cyber-Phys. Syst.*, vol. 4, no. 2, 2020.

[16] S. Nsaibi, L. Leurs, and H. D. Schotten, "Formal and simulation-based timing analysis of industrial-ethernet sercos iii over tsn," in *2017 IEEE/ACM 21st International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, 2017, pp. 1–8.

[17] O. Seijo, Z. Fernández, I. Val, and J. A. López-Fernández, "SHARP: towards the integration of time-sensitive communications in legacy LAN/WLAN," in *2018 IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–7.

[18] D. B. Mateu, D. Hallmans, M. Ashjaei, A. V. Papadopoulos, J. Proenza, and T. Nolte, "Clock synchronization in integrated tsn-ethercat networks," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 214–221.

[19] "IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic," *IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-2014/Cor 1-2015)*, pp. 1–57, 2016.

[20] M. Ashjaei, G. Patti, M. Behnam, T. Nolte, G. Alderisi, and L. L. Bello, "Schedulability analysis of ethernet audio video bridging networks with scheduled traffic support," *Real-Time Systems*, vol. 53, no. 4, pp. 526–577, 2017.

[21] N. Reusch, L. Zhao, S. S. Craciunas, and P. Pop, "Window-based schedule synthesis for industrial ieee 802.1 qbv tsn networks," in *2020 16th IEEE International Conference on Factory Communication Systems (WFCS)*. IEEE, 2020, pp. 1–4.