CSRP: an Enhanced Protocol for Consistent Reservation of Resources in AVB/TSN

Daniel Bujosa, *Student Member, IEEE*, Inés Álvarez, *Student Member, IEEE*
and Julián Proenza, *Senior Member, IEEE*

***Abstract*—The IEEE Audio Video Bridging (AVB) Task Group (TG) was created to provide Ethernet with soft real-time guarantees. Later on, the TG was renamed to Time-Sensitive Networking (TSN) and its scope broadened to support hard real-time and critical applications. The Stream Reservation Protocol (SRP) is a key work of the TGs as it allows reserving resources in the network, guaranteeing the required quality of service (QoS). AVB's SRP is based on a distributed architecture, while TSN's is based on centralized ones. The distributed version of SRP is supported and used in TSN. Nevertheless, it was not designed to provide properties that are important for critical applications. In this work we model SRP using UPPAAL and we study the termination and consistency. We verify that SRP does not provide such properties. Furthermore, we propose an improved protocol called Consistent Stream Reservation Protocol (CSRP) and we formally verify its correctness using UPPAAL.**

***Index Terms*—AVB, CSRP, SRP, TSN, UPPAAL.**

I. Introduction

THE IEEE Audio Video Bridging (AVB) Task Group (TG) [1] was created in 2005. Its purpose was creating a set of standards to provide Ethernet with soft real-time capabilities oriented to applications related to audio/video streaming. Specifically, the AVB TG started three projects, namely the IEEE Std 802.1AS [2], dedicated to clock synchronization; the IEEE Std 802.1Qav, which standardized the Credit-Based Shaper [3]; and, finally, the IEEE Std 802.1Qat, which standardized the Stream Reservation Protocol (SRP) [4]. In addition, the TG created a profile that sets a series of rules to ensure a minimum Quality of Service (QoS) when using the aforementioned standards. The profile is the IEEE Std 802.1BA-2011: Audio Video Bridging Systems [5]. This set of standards is commonly referred to as AVB standards.

Over time, the interest in the work done by the TG grew, also in areas of application beyond audio/video streaming, such as automotive [6], automation [7] and energy distribution [8]. For this reason, in 2012 the group was renamed to Time-Sensitive Networking (TSN) TG and its target broadened to meet the needs of these new applications, which are usually based on Critical Distributed Embedded Systems (CDES). Specifically, the TSN TG aims at providing Ethernet with proper support for mixed hard and soft real-time communications, flexibility of the traffic requirements and fault tolerance mechanisms. The set of standards developed by the TG is usually referred to as TSN standards.

Although the number of standardization projects carried out by the TSN TG grows at high speed, there are some projects that can be considered the core of the TG activity. One of the key projects is SRP, which was originally standardized by the AVB TG in [4] and subsequently reviewed by the TSN TG in [9]. SRP allows Ethernet to reserve resources along the path that connects a transmitter to one or more receivers. More concretely, SRP only authorizes the transmission of messages after verifying that the network can convey such messages with the required QoS. This prevents frame delays over the predefined limits during transmission and frame losses due to overflows of buffers in bridges. Moreover, SRP allows modifying the traffic requirements at run-time, providing a certain degree of flexibility to the network.

Currently, there are three different SRP architectures defined by the TSN TG in [9]. The first one is the fully distributed architecture, created in the context of AVB; whereas the other two architectures are centralized and they were defined in the context of TSN. TSN relies on YANG [10] to configure the network when using centralized architectures. YANG is a data modeling language which allows to define which data must be used to configure a network device and which is the format of said data. This allows to standardize and simplify the integration of different processes and applications in distributed systems.

We must note that, at the moment of writing this paper, there is no available YANG model to support the online configuration of the reservations of event-triggered traffic using the centralized architectures proposed in TSN [11]. Therefore, the distributed version of SRP is still used in TSN to manage the network resources of event-triggered real-time traffic. For this reason, and for the interest that certain areas such as automotive have shown on the distributed SRP [12], [13], [14], [15], [16], [17], [18], we believe that this version of SRP is going to continue to be used in the upcoming years.

As we have said, TSN targets critical applications, which means that all protocols, including every version of SRP, must exhibit certain properties which are common in CDES to ensure the proper behavior of the overall system. In this work we focus on termination and consistency. On the one hand, it is common for applications executed by CDES to carry certain actions within a bounded time and, thus, termination must be guaranteed. On the other hand, nodes in CDES usually need to have a consistent view of the network or share data consistently to interact with each other correctly.

This work is supported in part by the Spanish Agencia Estatal de Investigación (AEI) and in part by FEDER funding through grant TEC2015-70313-R (AEI/FEDER, UE). Corresponding author: D. Bujosa.

D. Bujosa is with the Mälardalen University, Västerås, Sweden (e-mail: daniel.bujosa.mateu@mdh.se).

I. Álvarez and J. Proenza are with the University of the Balearic Islands, Palma, Spain (e-mail: ines.alvarez@uib.es and julian.proenza@uib.es).

Nevertheless, the distributed version of SRP was not developed to be used in CDESs. Even though we can see that the distributed SRP does not guarantee termination nor consistency with a simple analysis, it is not effective to thoroughly identify the potential scenarios without using any formal tool. For this reason, in this work we use a formal model checker to verify in an exhaustive manner whether the distributed version of SRP provides these properties and in which cases it does not. From now on, whenever we say SRP we refer to the distributed version of the protocol.

Specifically, we use the UPPAAL model checker [19] to build a model of SRP. Modeling any system or protocol requires to abstract certain details, as analyzing all possible scenarios in an exhaustive manner requires a great amount of memory and time. For this reason, our SRP model abstracts implementation details of the protocol. Nevertheless, the level of abstraction used in this work is the typical when modeling communication networks. Moreover, we validate our model, understanding the term validate as the evaluation done to ensure that the model is properly implemented and that it is a faithful representation of the behavior of SRP.

We then use our UPPAAL model to verify that SRP does not provide termination nor consistency and to detect in which scenarios this happens. We use the term verify to refer to the evaluation done to ensure that a system presents a certain property, i.e., our system is the right one for our needs. Moreover, we discuss the consequences derived from the absence of termination and consistency. We propose different ways to modify SRP in order to provide it with the aforementioned properties and we select what we consider to be the best one. Finally, we create a UPPAAL model of the modified protocol, which we call Consistent Stream Reservation Protocol (CSRP). Again, we validate our model and we verify the correctness of our design.

The remainder of the document is structured as follows. Section II summarizes the related work, while Section III explains the parts of SRP that are most relevant for this work. Section IV provides an overview of the SRP model we implemented in UPPAAL while Section V and Section VI show the termination and consistency issues detected and their consequences. Section VII describes the solution proposed. Section VIII describes the changes applied to the SRP model to implement the proposed CSRP while Section IX and Section X show the formal verification of the correctness of CSRP. Finally, Section XI summarizes the work done.

II. RELATED WORK

Due to the great relevance of the AVB and TSN standards, the community has carried out a significant amount of work related to their study, application and improvement. For example, in [20] authors describe the requirements for an AVB network, summarize the methods described in the standards and describe how they can be used by several higher layer protocols; while in [21] authors provide an up-to-date comprehensive survey of the TSN standards and the related research studies.

Furthermore, there are many works related to the study of AVB's efficiency, such as [14], [16], [22], [23]. On the other hand, in the work presented in [24] the authors detect a drawback in the resource reservation de-registration specification, which leads to the waste of the network resources, and proposed some solutions. Moreover, some works present solutions to provide fault tolerance against permanent faults using SRP [25].

Nevertheless, to the best of the authors' knowledge, there are no works related to the study of the termination and consistency of the distributed version of SRP, apart from the preliminary work presented in [26]. We next list the contributions of this paper compared to the work presented in [26]. In this work we carry out an improvement of the UPPAAL model of SRP resource reservation mechanism which is explained in more detail in Section IV. We study the termination and consistency of the reservations in different scenarios and components with this new UPPAAL model. We design an enhanced version of SRP that does exhibit the termination and consistency properties, thereby eliminating all the issues we have identified for SRP and that provides the network devices with enough information to make rather complex decisions about the reservation of resources within a bounded time. Finally, we implement the proposed solution, i.e., CSRP, in the UPPAAL model and we verify that it is correct.

III. SRP OVERVIEW

As we anticipate in Section I, SRP is a key piece for many of the projects developed by the AVB and TSN TGs. Specifically, SRP is key to provide real-time guarantees to Ethernet-based communications. More concretely, SRP allows verifying that there are enough resources in the network to convey the traffic and reserving said resources. This allows to bound the end-to-end delay of the frames and to avoid the loss of packets due to the buffer overflow. Moreover, SRP can be used to modify the traffic requirements at run-time, giving the network a certain degree of flexibility.

SRP follows the publisher-subscriber paradigm, where the publisher is called talker and the subscribers, listeners. The real-time data communications are made through streams. A stream is a logical communication channel that carries traffic defined by a set of parameters, such as the period or frame size. For example, if one temperature sensor, the talker, wants to transmit its measurements with a period of 10 ms and a payload of 1 byte to other nodes, the listeners, the network has to check that there are enough resources and, if there are, it must create a stream with the specified period and payload.

As we have already said, there are three different SRP architectures. Nevertheless, as this work focuses on the distributed version of SRP, next we only explain the resource reservation mechanism of this version. Further details on the other architectures can be found in [9].

It is important to note that all the decisions regarding the reservation of resources are taken using local information only. Nevertheless, there is important information related to the reservations that must be propagated throughout the network; e.g. the amount of resources needed for a stream, whether a certain bridge has resources available or not, etc. This

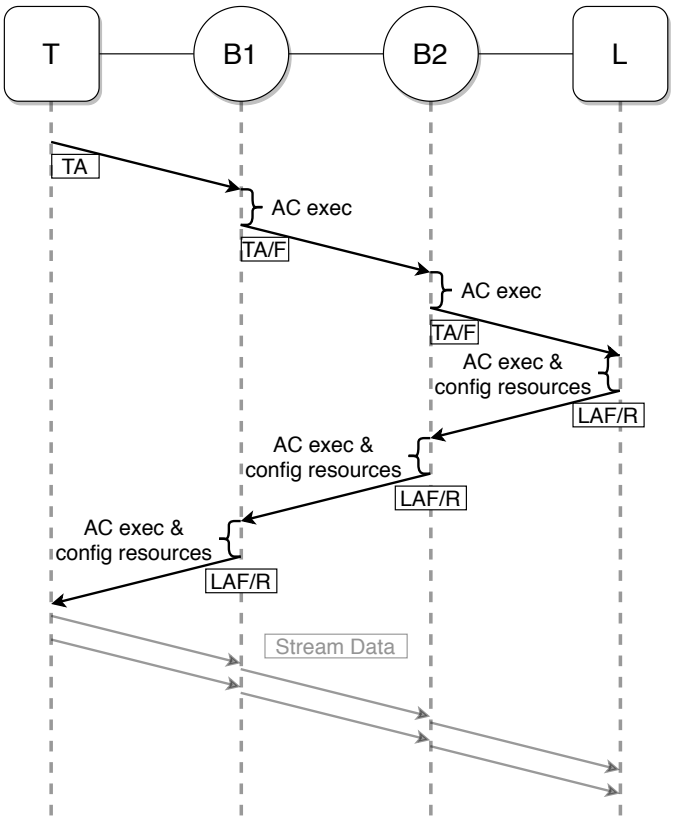


Fig. 1: Time diagram of the resource reservation mechanism in a network with a line topology.

information is conveyed within special messages called talker and listener attributes. The arrows in Fig.2 represent the direction in which the attributes are propagated throughout the network. Next we describe the process in detail.

Fig.1 shows the time diagram of the resource reservation mechanism in a network with a line topology. This network consists of one talker (T), one listener (L) and two bridges (B1 and B2). As we can see in the figure, when a talker wants to transmit a set of frames with certain parameters, it must first create the stream to convey such frames. To create a stream the talker has to declare its intention to communicate by transmitting in broadcast mode a special message called Talker Advertise (TA) message. This message conveys stream identification information, as well as the resources needed to convey the traffic. This information is then used in the rest of devices of the network to check whether there are enough resources for the stream so that it can be created. This evaluation of available resources is called Admission Control (AC). Note that SRP relies on other mechanisms that eliminate the loops in the network to prevent the TA message from circulating the network indefinitely.

The TA message transmitted by the talker is received by the bridge connected to it. When a bridge receives a TA message, each forwarding port checks if it has enough resources for the stream or not by executing the AC. In this protocol, a forwarding port is any port through which the TA message was not received, e.g., the port that connects B1 to B2 in Fig.1. If the port has enough resources, the TA message is forwarded to the next device, i.e., the next bridge or node. On the other hand, if the port does not have enough resources, it sends a so called Talker Failed (TF) message instead. A TF message conveys the same information as the TA message plus the reason for the failure in the reservation. Bridges that receive a TA message transmitted by another bridge through one of their ports behave as we have just described. In contrast, if the message received is a TF message, bridges transmit a TF message through all their forwarding ports, without carrying the AC.

Regarding nodes, we have to note that not all nodes are listeners for all streams. Therefore, if a node that does not want to become a listener of the stream receives a TA or TF message, it does not carry any further actions. In fact, it does not even inform the talker about its lack of interest in the stream. On the other hand, if a node receives a TA or TF message and is willing to listen to the stream there are three possible scenarios to consider: (i) the listener receives a TF message and cannot therefore become a receiver of the stream that is being created, so it sends a message called Listener Asking Failed (LAF) to the bridge; (ii) the listener receives a TA message but, while checking its resources it realizes that it does not have enough resources to receive the stream, so it sends an LAF message to the bridge; and, (iii) the listener receives a TA message and, while checking its resources it realizes that it has enough resources to receive the stream, so it sends a message called Listener Ready (LR) message to the bridge.

Each port of the bridges connected to a listener can receive an LR or LAF message. If a port receives an LAF message it does nothing else. If a port receives an LR message the port checks its resources again. If it does not have enough resources the port changes the LR received to an LAF; otherwise, if it has enough resources, the port reserves the resources (*config resources* in the figure). Whenever a bridge is connected to several bridges or nodes, it may have several listener responses to forward. In this case the bridge combines the responses into a single one and transmits it towards the talker. The result of combining the responses is the following: (i) if the bridge receives an LR in all the ports, it transmits to the talker an LR message; if the bridge receives an LAF in all the ports, it transmits to the talker another LAF message; and, if the bridge receives LR messages in some ports and LAF messages in other ports, it will transmits to the talker a new message called Listener Ready Failed (LRF) message. Whenever a bridge receives an LRF message it forwards an LRF message to the talker, regardless of the other listener attributes it receives. Note that in Fig.1 listener attributes cannot be LRF because in a linear topology there is only one port receiving responses, therefore, bridges cannot receive LR messages through some ports and LAF messages through other ports.

Finally, the talker waits until it receives an LR or LRF message to start the data transmission. Once the stream has been created, the talker can delete it at any time by means of the unadvertise stream mechanism. The talker transmits a message to eliminate the stream from all devices. This message is also transmitted in broadcast mode to ensure that all bridges and listeners receive the indication to eliminate the stream.

We call a reservation distribution to each possible combination of paths that reserve resources. We need to note that we consider to be good and bad reservation distributions. For instance, let us assume that we have a network with a star topology with one talker, two listeners and one bridge in the middle. One example of good reservation distribution is one where all the ports have reserved the required resources, whereas one example of bad reservation distribution is one where the listener ports have reserved the required resources but the talker port has not. In this last case there is a waste of resources in the links to the slaves because, as the port of the talker is not reserved, the slaves are not going to receive anything from that talker, regardless of their reservations.

IV. SRP UPPAAL MODEL

This section introduces the model developed in this work. The model is implemented using the UPPAAL model checker which is a tool for modeling real-time systems and formally verifying their properties [19]. In UPPAAL the systems are modeled by means of interconnected timed automata (finite-state machines extended with clocks that progress at the same pace). Each automaton is specified by a template that can be instantiated several times. At the same time, templates are constructed using locations, edges, local variables and local clocks, and can synchronize through different types of channels.

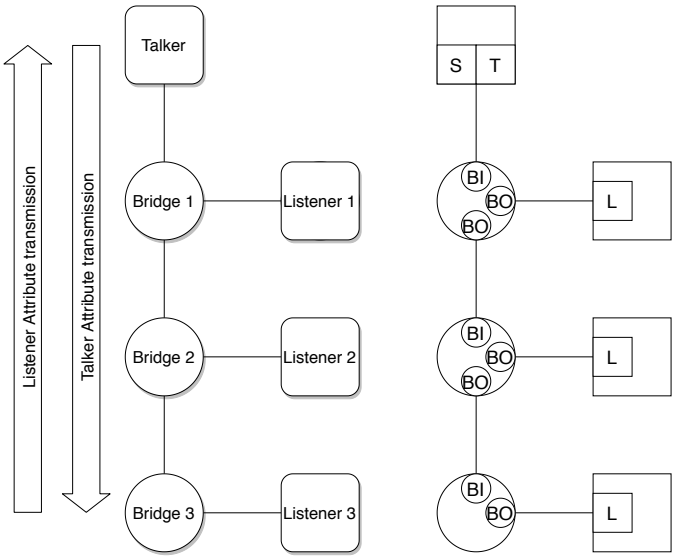
In addition, UPPAAL provides a formal query language that allows defining properties that the system should exhibit.

Using the model and the queries as inputs, the tool performs an exhaustive check of the properties, i.e., it explores all the possible execution paths of the model to verify whether the properties hold. After this, UPPAAL informs the user about the result and, if a property does not hold, it shows an execution path in which the property is violated.

In this paper we abstract the description of the model. Section III of [27] describes our model in an exhaustive way. The complete model files and the files used for its verification can also be found in [27].

Fig.2 (a) represents the network we modeled with UPPAAL while Fig.2 (b) represents the resulting UPPAAL model. As can be seen, our SRP model is made of 5 different templates: Talker template, Stream template, Listener template, BridgeInput template and BridgeOutput template (represented as T, S, L, BI, BO respectively in Fig.2(b)). These templates model the different relevant actions of the protocol carried out by the talkers, bridges and listeners. Specifically, as we can see in Fig.2, our model has one instantiation of the Talker and Stream templates to model the actions carried out by one talker. It also has three instantiations of the Listener template to model the actions carried out by three listeners. And, finally, it has 3 instantiations of the BridgeInput template and five instantiations of the BridgeOutput template to model the actions carried out by three AVB bridges. Other network elements, such as links, are represented in the model by variables, clocks and channels.

As we can see in Fig.2(a), and as we have already said, our model is made up of one talker, three bridges and three



(a) Modeled network consisting of one talker, three listeners and three bridges.

(b) Abstraction of the network model made with UPPAAL.

Fig. 2: Representation of the modeled network and its model by means of templates where T represents the Talker template, S the Stream template, BI the BridgeInput template, BO the BridgeOutput template and L the Listener template.

listeners, each connected to one bridge. We decided to use three listeners for many reasons. The first reason is that, in many critical systems is usual to use active replication, using three replicas which perform majority vote on each result, in order to tolerate the failure of nodes. Moreover, three listeners are enough to have all relevant combinations of responses of the listeners. This is because, as listeners can only be or not interested in the stream or interested with or without resources, having more listeners would increase the times one of this options appears but would not produce a different scenario. On the other hand, we connected one listener to each bridge to have paths with different lengths and end-to-end delays, factors that increase the likelihood of encountering consistency issues. Finally, we used a line topology because SRP relies on other protocols that eliminate the loops of the network, such as the Rapid Spanning Tree Protocol [28] or the Shortest Path Bridging Protocol [29].

Like any model of a system, our SRP model has a series of abstractions. First, we only model the transmission of one stream because allowing the model to transmit several streams would lead to the explosion of the state space without providing any benefit, on the contrary, it would make the model more difficult to analyze. We neither model the transmission of data frames because it is not part of SRP and it would increase the complexity of the model unnecessarily, as it would distort the model without giving greater precision to the analysis of the protocol. Finally, we did not take into account the presence of errors for several reasons. First, the property issues we detected

appear in the absence of faults in the network. Secondly, there are some works like the one presented in [30] that allow tolerating faults in the channel by using proactive replication of frames. It is important to note that this level of abstraction is typical for network models and that we have validated our model to ensure that it is a faithful representation of SRP.

The model used in [26] abstracted the number of ports of the devices. Specifically, all devices had a single reception port and a single transmission port, even though AVB bridges can have an undetermined number of ports. This abstraction reduced the state space and the complexity of the Bridge template, and, in addition, avoided specifying a specific number of ports on the bridges which would decrease the generality of the model. However, this abstraction moved slightly away from reality, reduced the information present in the bridges and transferred certain decisions from the bridges' output ports to the input ports of the devices connected to them.

In this work we present a more detailed, yet analyzable and general model. Specifically, our model divides the Bridge template into two, one for the reception port of the talker attributes and transmission of the listener attributes and another for the reception of the listener attributes and transmission of the talker attributes. These templates can be instantiated as many times as necessary for each bridge, so the generality of the model is maintained. In addition, the model conforms more to reality and allows to keep decisions and information on the bridge. Nonetheless, this new model has an increased state space compared to the one in [26], which increases the time required for the analysis.

V. Evaluation of the Termination of SRP

As we said in Section I, termination is a basic property of the CDESs. Thus, all TSN protocols should provide it in order to support this kind of systems, even more if we talk about SRP which is responsible to accept the streams and reserve the resources, a key piece to ensure a good behavior of the system.

This analysis does not aim at demonstrating that the distributed version of SRP does not present termination, since this is relatively obvious once we know the protocol. Instead, we use it to analyze in depth the cases of non-termination and their consequences. In addition, this analysis is helpful to find a solution to the problem and to have a reference to evaluate the proposed solution.

In this work we differentiate two levels of termination: termination for the application and for the infrastructure. The first one affects the nodes and, therefore, the application. The lack of termination at the application level can cause malfunction of some critical applications. This is due to the fact that many of those applications require to know the result of the reservation to make important decisions.

The infrastructure level refers to the bridges of the network. Even if in an ideal system these devices do not require termination, it is important to provide it to prevent unforeseen and undesirable effects in future reservations. For example, if a bridge receives many requests without resolution, it would be possible to cause an overflow of the buffer that could prevent the bridge from accepting new reservation requests or force it to eliminate some already accepted ones.

We next present the problems detected but it is important to note that the issues are mainly due to the fact that in SRP listeners do not inform the bridges nor the talkers when they are not interested in binding to a stream. Section IV of [27] shows and explains in more detail the queries used.

A. Termination at the Application Level

Using the UPPAAL model, we find a series of scenarios where the talker does not receive any response from the listeners and, thus, it waits indefinitely. This can happen even in the absence of faults, when there are no listeners interested in the stream. As we have said before, many critical applications require to know the result of the reservations to make important decisions. Thus, the lack of termination can cause a malfunction of those applications, such as blocking the decision process or leading to incorrect decisions due to the lack of knowledge.

To check the termination for the application level and determine the causes of the issues detected we used the query (Q1), which corresponds to query 1 in [27]. This query checks if there is a path of states in the system in which the talker never receives any listener response. The query is satisfied which means that there are scenarios in which a talker does not receive any listener response leading to a termination issue. Then we checked if it is possible that this happens if at least one listener is interested in the stream. To do that we used another query (Q2), which corresponds to query 2 in [27]. This query checks if, at the end of the listeners actions, the response of at least one listener, i.e., there is at least one listener interested in the stream, implies the reception of responses by the talker. This query is satisfied, which means that if at least one listener is interested in the stream, the talker receives at least one response. Finally, we checked that the non-reception of responses by the talker was due to the non-transmission of responses by the listeners. This was checked with another query (Q3), which corresponds to query 3 in [27]. This query checks if, at the end of the listeners actions, if no listener has responded, the talker receives no response.

The use of the previous queries allowed us to determine that the only case where termination is not achieved in the application level is when no listener is interested in the stream that the talker is announcing.

B. Termination at the Infrastructure Level

A bridge that forwards the request of a talker waits for the responses of the listeners indefinitely. Also, bridges register talkers' attributes in all their ports, and they do so for all the talkers willing to transmit. Similar to what happens for termination at the application level, we find some scenarios where some bridges do not receive any response from the listeners, even in the absence of faults and even if the first level of termination is actually achieved by the protocol. Thus, bridges can wait indefinitely, e.g., if there are no listeners interested in the stream connected directly or indirectly to the bridge. This can cause an unnecessary use of memory

in bridges and can later prevent the creation of streams with listeners willing to bind due to a lack of memory.

To check the termination at the infrastructure level, and determine the causes of the issues detected, we used three different queries for each port. These queries are similar to the ones used in the verification of the termination at the application level, but have the particularity that must be checked for each port of the bridge. The first of these three queries (Q4), which corresponds to queries 4, 7, 10, 13 and 16 in [27], checks if there is any path of states in the system in which the port of the bridge does not receive any listener response. As the query is satisfied, it is possible that the port of a bridge does not receive any listener response, leading to a termination issue in the bridges.

Then, we used another query (Q5), which corresponds to queries 5, 8, 11, 14 and 17 in [27], to verify if this is possible even if at least one listener connected directly or indirectly to the bridge is interested in the stream. This query checks if the response of at least one listener, i.e., if at least one listener is interested in the stream, leads to the reception of responses by the bridge port. As this query is satisfied, if at least one listener is interested in the stream, the port of the bridge receives at least one response.

Finally, we checked that the non-reception of responses by the port of a bridge was due to the non-transmission of responses by the listeners connected directly or indirectly to the port of the bridge. This was checked with the third query (Q6), which corresponds to queries 6, 9, 12, 15 and 18 in [27]. This query checks if when no listeners have responded at the end of the listeners actions, so there are no listeners interested in the stream, the port of the bridge receives no response.

The use of these queries allowed us to conclude that the only case where there is no termination at the infrastructure level is the one where no listeners connected directly or indirectly to a bridge are interested in the stream.

VI. Evaluation of the Consistency of SRP

As it was introduced in Section I, consistency is an important property in CDESs. For instance, SRP should do the reservation of resources in a consistent manner if we intend to have several replicated nodes. The nodes should have the same inputs and outputs, which cannot be guaranteed without a consistent reservation of resources. This is just a simple example that illustrates the importance of consistency, even though consistency can be required in any distributed system.

As in Section V, this analysis was not performed to demonstrate that the distributed version of SRP does not present consistency, which is relatively obvious once you know the protocol, but to analyze in depth the scenarios that can cause inconsistencies and their consequences. In addition, again, this analysis is helpful to find a solution to the problem and to have a reference to evaluate the proposed solution.

As in the previous section, we differentiate two levels of consistency: consistency for the application level and for the infrastructure level. Again, the first one affects the nodes and, therefore, the application. The lack of consistency at the application level can cause malfunction of some critical applications. Some of the applications targeted by TSN require the different nodes to carry out coordinated actions because, e.g., they may rely on active replication of the nodes. In these applications, consistency in the communications is key to guarantee the correct operation of the overall system. The first step towards achieving consistent communications is to reserve the network resources consistently. Thus, at this level, SRP should guarantee that enough listeners have resources reserved for the communication before starting to transmit.

As before, the infrastructure level refers to the bridges of the network. As we will see later, inconsistencies when reserving resources in bridges can cause the waste of resources. This, in the long term, causes that streams, for which there would be sufficient resources, cannot be declared due to the resources reserved and wasted in some bridges.

As in the evaluation of the termination, despite the importance of consistency, we found some issues in both levels even in the absence of faults. We next present the problems detected but it is important to note that the issues are mainly due to the fact that information related to the reservations is propagated in a single direction. That is, the talker attribute transmitted by a talker is forwarded always towards the listeners; while, when listeners and bridges reply to a stream declaration, the information is only forwarded towards the talker. Thus, not all the devices involved in the reservation of a stream receive the same information. We next describe the consistency issues detected and their effects. Section V of [27] shows and explains in more detail the queries used.

A. Consistency at the Application Level

In SRP, resources can be reserved for a subset of listeners, even when there are listeners willing to communicate that do not have resources to do it. In this case, the talker only communicates to a subset of listeners, generating an unnoticed inconsistency in the exchange of data. This means that actually starting a stream (with some listeners) has priority over doing it consistently (with either all or none of them). In addition, talkers cannot know which listener has enough resources and which one does not. A talker only knows if all interested listeners have enough resources when it receives LR messages; if all interested listeners have not enough resources when it receives LAF messages; if no listener is interested when it does not receive any answer; or, if at least one interested listener has enough resources when it receives LRF messages. This limited information does not allow the talker to take intelligent decisions. Furthermore, we have to take into account that this information can change during the execution of the SRP mechanism e.g. it is possible for a talker to receive an LR message and then receive an LRF message. Something similar can happen in listeners. They may be interested in the stream and have sufficient resources, but they do not receive anything because during the transmission of the response, the route to the talker did not have enough resources.

Furthermore, even when all listeners willing to bind have enough resources to do so, there are scenarios where consistency for the application is not guaranteed all the time. This can happen for two reasons, first the paths between a

talker and different listeners may differ in length and end-to-end delay and, second, the talker starts transmitting as soon as it receives the response of one listener ready to receive. Therefore, some listeners willing to bind to the stream, with enough resources throughout the whole path towards the talker, may miss the first frames transmitted by the talker. This can cause, for example, two replicas to be in two different states so that, although from that moment they receive the same data, they will not provide the same result.

To check the consistency for the application level, and to determine the causes of the issues detected, we used query (Q7), which corresponds to queries 19, 20 and 21 in [27], to check if there is at least one state in which the talker is already transmitting, a listener is interested in the stream and, from his point of view, has sufficient resources but the route from the talker to the listener has not reserved the necessary resources for that stream. This test shows that the talker can start transmitting even when there are interested listeners that will not be able to receive the stream. Moreover, it also shows that there are listeners that believe they are going to receive the stream but never will.

Another query (Q8), which corresponds to query 22 in [27], is used to verify that, even when all interested listeners can bind to the stream, some of them may miss the first messages because the talker starts transmitting before finishing the resource reservation. Specifically, it checks if a talker transmitting, a listener waiting for the stream and the route not yet reserved may lead the system to a state in which the route will never be reserved. As the query is not satisfied we proved the inconsistency in the data received at the beginning of the stream.

B. Consistency at the Infrastructure Level

In this work we also find out that bridges can make inconsistent decisions regarding the reservation of resources of a stream. Specifically, in SRP it is possible that some bridges reserve resources for a stream but other bridges in the same route to the listener do not. This implies a waste of resources in the bridges that reserved the resources because the listener for which they reserved the resources is not going to receive the stream because of the bridges in the same route that did not reserve the resources. This may not be problematic at first, but, with an utilization close to 100%, this may cause streams, for which there would be sufficient resources, cannot be declared due to the resources wasted in these bridges.

To check the consistency for the infrastructure level, and to determine the causes of the issues detected, we used query (Q9), which corresponds to queries 23 and 24 in [27], to check if there is at least one state, after the mechanism has been executed, in which the stream is being transmitted while the link that supplies one or more bridges is not reserved but the links of the bridges are. This reservation distribution implies a waste of resources in all the links reserved by the bridges affected because, as the link that supplies them is not reserved, they are not going to receive data messages from this stream. Finally, another query (Q10), which corresponds to query 25 in [27], checks if the transmission of the stream always leads to one of all correct distributions of resource reservations. As it is not satisfied, we can determine that incorrect distributions of resource reservations (with waste of resources) may happen using SRP.

VII. CSRP Description

We designed a series of solutions to deal with the drawbacks described in Sections V and VI. The main objective is to provide network devices with a consistent view of the reservation of resources so that they can make rather complex decisions within a bounded time.

A trivial solution could consist in modifying the current SRP in the following way:

1) The talker multicasts the talker attribute instead of broadcasting it. With this change, the network avoids sending the talker attribute to non-interested listeners, eliminating the termination issue of the application and infrastructure level. The problem is that it makes the network less open and adaptive, understanding open as a network where nodes can join or leave a stream dynamically and adaptive as a network where stream requirements change during run-time.
2) A bridge that has decided that it has enough resources when retransmitting the talker attribute cannot change this decision when it receives the listeners' responses. This would allow bidirectional propagation of decisions without adding another round of transmissions solving the consistency issue at the infrastructure level. The problem is that it can hinder the creation of streams that attempt to be declared simultaneously and does not prevent the waste of resources in unnecessary reservations.
3) The listeners multicast the listener responses, instead of unicasting them, conveying their ID in the response. In this way the listeners can inform the talker, the bridges and other listeners of whether they can receive or not, solving the consistency issue at the application level.The problem is that listeners can flood the network with control messages because each of them would transmit its status in multicast mode.
4) The talker, listener and bridges must make decisions deterministically based on the information received which, in the absence of faults, will be consistent.

Despite solving the problems detected, this solution greatly modifies the mechanism and has several limitations. For these reasons we decided to develop the following solution which is a compendium of the different solutions proposed in [26].

The most relevant differences between this new protocol and previous proposals are:

1) The listener responses convey the ID of the listeners that sent them. This gives the talkers and bridges the necessary information to know which listeners can receive and which not.
2) Bridges' and listeners' decisions can change depending on the decision taken by the talker.
3) The talker, after a bounded time limited by a timer in it, will decide which listener can receive the stream and which cannot based on the information received. After

this, the talker will send the result of his decision to all the devices so that they have a consistent view of the status of the reservations and carry out the necessary actions.

Even if this solution requires some changes in SRP, it can be implemented in a way that non-modified devices can still work normally. For example, in the implementation we are proposing next, listeners do not need to implement the solution although this would imply that these listeners would not have a consistent view of the network. Furthermore, it provides the talker the necessary information to make a centralized, rather complex and bounded in time decision which, in addition, will be sent to the rest of the network to maintain the consistent view of the reservation of resources. Moreover, all this would be achieved by adding only one additional broadcast transmission, the one by the talker with the final decision, avoiding the multiple multicast transmissions of the listeners added in the previously proposed trivial solution.

The name of this new protocol is CSRP and its resource reservation mechanism is as follows:

The transmission of talker and listener attributes proceeds as in SRP, as it is described in Section III. The first modification of the protocol is found in the transmission of listeners attributes by the bridges. Bridges receive the listener attributes and combine them to send them to the talker. In order to accomplish this, bridges analyze the responses received by each port and then generate the new response that they transmit towards the talker. Whenever a bridge receives an LR message through a port, it checks whether the port has enough resources. If there are enough resources, the LR remains unchanged and the port reserves the necessary resources provisionally, instead of definitely like in SRP; otherwise, the LR becomes an LAF message. On the other hand, if the bridge receives an LAF message the value is left unchanged and the port does not reserve the resources. In case of concurrent requests, and this is another change with respect to SRP, the provisional reservation is made for the first LR or LRF message received, while the rest are transferred to a first-in, first-out (FIFO) list. The items in this list are only deleted when their reservation processes are completed or when the reservation of resources is confirmed.

After processing the listener attributes, each bridge must join them to forward an updated one to the talker. This process is the same as in SRP and is described in Section III. It is important to note that bridges do not wait for the reception of all the listener attributes, but they are continuously joining and retransmitting them as they receive new answers. In this way a bridge can transmit an LR or LAF message and then transmit an LRF message, just like in SRP. Nevertheless, in CSRP bridges must specify in the listener attribute which listeners can receive and which listeners cannot. To do so, CSRP relies on two lists, one for successful reservations and one for unsuccessful ones. Specifically, edge bridges introduce the identifier of the node that sends the LR or LAF message in the corresponding list and sends them embedded in the response to the talker. Whenever a bridge receives a response from another bridge, it checks the lists and updates them accordingly when joining the responses.

The talker waits for the answers for a bounded period of time. This time will depend on the application, topology and size of the network. Basically, it must ensure that the response from the farthest slave has enough time to reach the talker. This is implemented by means of a local timer in the talker which is activated at the beginning of the transmission of the TA and expires after the predefined time. After that time, the talker uses the lists with the nodes identifiers to know which listeners can receive and which listeners cannot and it decides whether to transmit the stream to all the listeners that can receive, to a subgroup or to none of them. This decision is communicated by transmitting in broadcast mode a message called Final Decision (FD), which contains a list of listeners that will receive the stream and listeners that will not receive the stream.

When a bridge receives the FD message it knows which listeners must receive the stream and which must not. In this way, bridges can lock the resources or eliminate unnecessary reservations. Listeners, on the other hand, can know whether they are subscribed to the stream or not and do not wait indefinitely for the data transmission.

Once the FD message has been transmitted and the resource reservation mechanism has finished, the talker starts transmitting the data stream. Finally, as in standard SRP, once the stream has been created, the talker can delete it at any time by means of the unadvertised stream mechanism.

VIII. CSRP UPPAAL MODEL

The UPPAAL model of CSRP has the same topology, same templates, same instantiations of the templates and same abstractions as the model of the standardized SRP explained in Section IV. To formally verify the correction of the improved mechanism (CSRP's resource reservation mechanism), we modified as little as possible the model shown above to include the changes proposed in our solution. Again, the templates and the modifications applied to them can be found in Section VII of [27]. The complete model files and the files used for its verification can also be found in [27]. We next explain in an abstract way the main changes done to the SRP model.

In the Talker template we basically eliminated the instantaneous transmission of data that occurred as soon as the speaker received an LR or LRF message. On the other hand, we added a timer to define the waiting time for listener responses and implement the transmission of the FD message.

In the bridge templates we implemented the reception and forwarding of the FD message and the mechanisms to change the resource reservations based on it.

Finally, in the Listener template we implemented the reception of the FD message and the mechanism so that listeners know if they can receive or not. It is important to remember that these modifications in listeners are not essential. However, not implementing them would imply that listeners remain unsure of whether they will receive or not until they receive any data message of the stream.

IX. EVALUATION OF THE TERMINATION OF CSRP

We next describe the verification of CSRP from the termination point of view. Again, we address the issues at the

application and infrastructure level. To do that, we used the same queries that proved the non-termination in SRP plus some additional queries. These are explained in more detail in Section VIII of [27].

Just like in SRP, if in CSRP no nodes want to bind to a stream, the talker and bridges do not receive any listener responses. Thus, we provide termination with the timer in the talker and the FD message in the bridges, as now both, talkers and bridges, know when to stop waiting for listener responses.

A. Termination at the Application Level

In CSRP it is still possible that the talker does not receive any response from the listeners (see Sub-section V-A). However, using the timer, the talker always stops waiting for an answer, makes a decision based on the information it has received and informs about it by means of the FD message to the rest of the network. We used the UPPAAL model of CSRP to verify the behavior of the protocol. Specifically, we check that all the nodes finish the resource reservation process within a bounded time determined by the timer in the talker and the distance between the talker and the listeners.

To check the termination at the application level of CSRP we used the same three queries that we used to evaluate SRP (Q1, Q2 and Q3), which correspond to queries 26, 27 and 28 in [27]. The results of evaluating these queries allow us to determine that in CSRP it is still possible that the talker never receives any response from the listeners. However, we used another query (Q11), which corresponds to query 29 in [27], to evaluate whether the talker stops waiting for a response after the timer has expired, makes a decision based on the received information and informs about it by means of the FD message to the other devices of the network. This query proved that CSRP provides termination at the application level.

B. Termination at the Infrastructure Level

With this evaluation we see how bridges' ports may not receive any listener response. However, thanks to the FD message sent by the talker, bridges always stop waiting for an answer and change their reserved resources based on the talker decision. We used the UPPAAL model of SRP to verify that all bridges finish the resource reservation process within a bounded time.

Using the same queries as the ones used to evaluate SRP (Q4, Q5 and Q6), which correspond now to queries 30 to 36, 38 to 40, 42 to 44 and 46 to 48 in [27], we found that bridges' ports may not receive any listener response. However, another query (Q12), which corresponds to queries 33, 37, 41, 45 and 49 in [27], demonstrated that, thanks to the FD message, the bridges always stop waiting for an answer and change their reserved resources based on the talker decision. Thus, CSRP provides termination at the infrastructure level.

X. EVALUATION OF THE CONSISTENCY OF CSRP

In this section we describe the verification of CSRP from the consistency point of view, at the application and infrastructure level. To do so, we use the same queries that proved the inconsistency in SRP plus some additional queries. These are explained in detail in Section IX of [27]. Nevertheless, this solution solves all the detected consistency issues. We achieved this by centralizing the decisions in the talker and ensuring the homogeneous propagation of information related to the reservation of resources.

A. Consistency at the Application Level

First, note that this solution does not aim at providing resources for all the listeners that want to bind. Instead, it aims at ensuring that all listeners know what is the status of the reservation regardless of whether they can receive or not. This was not guaranteed in the standard SRP but it is achieved in CSRP thanks to the FD message. We verify the consistent view of the network. Specifically we prove that when CSRP finishes the reservation process, all devices know which nodes are subscribed to the stream and which are not, including the nodes.

By using query Q7, which corresponds now to queries 50, 53 and 56 in [27], also used to evaluate SRP, we found that in CSRP there are states where a listener wants to bind to the stream and it thinks it can but the resources have not been reserved. However, another query (Q13), which corresponds to queries 51, 52, 54, 55, 57 and 58 in [27], demonstrated that, thanks to the FD message, now listeners know when they can and when they cannot receive the stream.

Finally, we used two queries (Q14 and Q15), which correspond to queries 59 and 60 in [27], to verify the consistent view of the network. These queries verify that always at the end of the reservation process all the lists with the successful and unsuccessful reservations explained in Section VII are consistent. Therefore, we can conclude that CSRP provides consistency at the application level.

B. Consistency at the Infrastructure Level

Finally, at the infrastructure level, we verify that CSRP avoids wasting resources with unnecessary reservations thanks to the FD message that informs the bridges about which listeners have been able to bind to the stream and which listeners have not, so that the bridges can free the resources they reserved for the listeners that cannot receive. We carry out this verification using the CSRP UPPAAL model.

To check the consistency at the infrastructure level of CSRP we used queries Q9 and Q10, which correspond now to queries 61, 62 and 63 in [27], also used in the evaluation of SRP. These queries prove that, at the infrastructure level, not only we avoid wasting resources with unnecessary reservations but also only the appropriate reservation distributions are generated. Moreover, we can conclude that CSRP provides consistency at the infrastructure level.

XI. CONCLUSIONS

The IEEE AVB TG defined a series of standards to provide Ethernet with soft-real time capabilities. Later on, the TG broadened its scope and was renamed to TSN. The new TG aims at providing hard real-time guarantees, flexibility of

the network configuration and fault tolerance mechanisms to Ethernet. Therefore, the work carried out by the TSN TG is intended to enable the use of standard Ethernet as the network technology for critical distributed embedded systems. One of the most important projects of the TGs is SRP, as it allows to reserve resources to provide timing guarantees and prevent frame losses.

In this work we present a model of the AVB's distributed version of SRP in UPPAAL. Thanks to it, we identify multiple scenarios in which SRP does not exhibit termination nor consistency and we determine the possible adverse effects that their lack can cause. Moreover, we have proved that these properties can be violated even in the absence of faults. Some of the problems detected and their effects are: (i) talkers and bridges in most of the cases do not know whether the resource reservation process is over, (ii) talkers never know which listeners have bound to the stream and which ones have not, (iii) subscribed listeners can miss the first data messages of the stream and (iv) some bridges can waste resources because they don't know if the bridges they depend on (bridges located on the route between the talker and them) have been able to reserve resources for them.

Thus, in this work we propose an enhanced version of SRP that enforces termination and consistency, thereby eliminating all the issues we have identified for SRP. This solution is called CSRP and not only it solves the aforementioned problems, but it also provides the nodes and bridges with enough information to make rather complex decisions about the reservation of resources within a bounded time. For instance, now a talker can advertise a stream and know which listeners can receive and which ones cannot, after a certain time determined by an internal clock. Based on this information, the talker can decide whether all, a subset or none of the nodes that manifest their interest in the stream can subscribe.

Finally, we have developed a UPPAAL model of CSRP and we have used it to verify that CSRP provides termination and consistency to the reservation of resources in the absence of faults. Further work will be done to extend the study of the behavior of CSRP in other scenarios and even using an experimental implementation to evaluate new aspects, such as its performance in real networks.

REFERENCES

[1] (2019, April) IEEE Audio and Video Bridging Task Group. https://1.ieee802.org/tsn/.

[2] "IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks," *IEEE Std 802.1AS-2011*, pp. 1–292, March 2011.

[3] "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, Jan 2009.

[4] "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)*, Sept 2010.

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

[6] 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, JUNE 2018.

[7] 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, March 2017.

[8] 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., Sep 2018.

[9] "IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks – Amendment 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements," *IEEE Std 802.1Qcc-2018 (Amendment to IEEE Std 802.1Q-2018 as amended by IEEE Std 802.1Qcp-2018)*, pp. 1–208, Oct 2018.

[10] M. Björklund, J. Schönwälder, P. Shafer, K. Watsen, and R. Wilton, "NETCONF Extensions to Support the Network Management Datastore Architecture," RFC 8526, ISSN: 2070-1721, https://tools.ietf.org/html/rfc8526, Tech. Rep., 2019.

[11] (2018) YANG Modules. https://1.ieee802.org/yang-modules/.

[12] S. Thangamuthu, N. Concer, P. J. L. Cuijpers, and J. J. Lukkien, "Analysis of Ethernet-switch Traffic Shapers for In-Vehicle Networking Applications," in *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2015, pp. 55–60.

[13] L. Lo Bello, "Novel Trends in Automotive Networks: A Perspective on Ethernet and the IEEE Audio Video Bridging," in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, Sep. 2014, pp. 1–8.

[14] P. Meyer, T. Steinbach, F. Korf, and T. C. Schmidt, "Extending IEEE 802.1 AVB with Time-Triggered Scheduling: A Simulation Study of the Coexistence of Synchronous and Asynchronous Traffic," in *2013 IEEE Vehicular Networking Conference*, Dec 2013, pp. 47–54.

[15] H. Lim, D. Herrscher, and F. Chaari, "Performance Comparison of IEEE 802.1Q and IEEE 802.1 AVB in an Ethernet-Based In-Vehicle Network," in *2012 8th International Conference on Computing Technology and Information Management (NCM and ICNIT)*, vol. 1, April 2012.

[16] R. Queck, "Analysis of Ethernet AVB for Automotive Networks using Network Calculus," in *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, July 2012, pp. 61–67.

[17] T. Steinbach, F. Korf, and T. C. Schmidt, "Real-time Ethernet for Automotive Applications: A Solution for Future In-Car Networks," in *2011 IEEE International Conference on Consumer Electronics -Berlin (ICCE-Berlin)*, Sep. 2011, pp. 216–220.

[18] A. Gothard, R. Kreifeldt, and C. Turner, "AVB for Automotive Use White Paper," AVnu Alliance, Tech. Rep., Oct 2014.

[19] G. Behrmann, A. David, and K. G. Larsen, *A Tutorial on Uppaal*, M. Bernardo and F. Corradini, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[20] M. D. Johas Teener, A. N. Fredette, C. Boiger, P. Klein, C. Gunther, D. Olsen, and K. Stanton, "Heterogeneous Networks for Audio and Video: Using IEEE 802.1 Audio Video Bridging," *Proceedings of the IEEE*, vol. 101, no. 11, pp. 2339–2354, Nov 2013.

[21] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, "Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards and Related 5G ULL Research," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 88–145, Firstquarter 2019.

[22] H. Lim, L. Völker, and D. Herrscher, "Challenges in a Future IP/Ethernet-based in-car Network for Real-Time Applications," in *2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2011, pp. 7–12.

[23] J. Migge, J. Villanueva, N. Navet, and M. Boyer, "Insights on the Performance and Configuration of AVB and TSN in Automotive Ethernet Networks," in *9th European Congress on Embedded Real Time Software and Systems (ERTS 2018)*, Toulouse, France, Jan. 2018.

[24] D. Park, J. Lee, C. Park, and S. Park, "New Automatic De-Registration Method Utilizing a Timer in the IEEE802.1 TSN," in *2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)*, Oct 2016, pp. 47–51.

[25] O. Kleineberg, P. Fröhlich, and D. Heffernan, "Fault-Tolerant Ethernet Networks with Audio and Video Bridging," in *ETFA2011*, Sept 2011, pp. 1–8.

[26] D. Bujosa, I. Alvarez, D. Čavka, and J. Proenza, "Analysing Termination and Consistency in the AVB's Stream Reservation Protocol," in *Proceedings of the IEEE 24th International Conference on Emerging Technologies and Factory Automation (ETFA 2019)*, October 2019.

[27] D. Bujosa, I. Álvarez, and J. Proenza, "Description of the UPPAAL Models for SRP and CSRP and Verification of their Termination and Consistency Properties," arXiv:2007.15712 [cs.NI], https://arxiv.org/abs/2007.15712, Tech. Rep., 2020.

[28] "IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges," *IEEE Std 802.1D-2004 (Revision of IEEE Std 802.1D-1998)*, pp. 1–281, June 2004.

[29] "IEEE Standard for Local and Metropolitan Area Networks–Media Access Control (MAC) Bridges and Virtual Bridges," *IEEE Std 802.1Q, 2012 Edition, (Incorporating IEEE Std 802.1Q-2011, IEEE Std 802.1Qbe-2011, IEEE Std 802.1Qbc-2011,IEEE Std 802.1Qbb-2011, IEEE Std 802.1Qaz-2011, IEEE Std 802.1Qbf-2011,IEEE Std 802.1Qbg-2012, IEEE Std 802.1aq-2012, IEEE Std 802.1Q-2012*, pp. 1–1782, Dec 2012.

[30] I. Alvarez, J. Proenza, and M. Barranco, "Towards a Time Redundancy Mechanism for Critical Frames in Time-Sensitive Networking," in *Proceedings of the IEEE 22nd International Conference on Emerging Technologies and Factory Automation (ETFA 2017)*, January 2018.

Daniel Bujosa received the degree in automation and industrial electronic engineering from the University of the Balearic Islands (UIB), Palma de Mallorca, Spain, in 2017 and a master in industrial engineering from the same University in 2019.

He is currently working towards the Ph.D. degree at the School of Innovation, Design and Engineering at Mälardalen University. His research interests include dependable and real-time systems, fault-tolerant distributed systems and dependable communication topologies.

Mr. Bujosa is a Student Member of the IEEE and IEEE Industrial Electronics Society (IES) since 2020.

Inés Álvarez received the degree in computer science with a specialization in computer engineering from the University of the Balearic Islands (UIB), Palma de Mallorca, Spain, in 2014 and a master in computer engineering from the same University in 2016.

She is currently working towards the Ph.D. degree in information and communications technologies at UIB and she is a Lecturer with the Department of Mathematics and Computer Science, UIB. Her research interests include dependable and real-time systems, fault-tolerant distributed systems and dependable communication topologies.

Ms. Alvarez is a Student Member of the IEEE since 2016.

Julián Proenza received the first degree in physics (Licenciatura en Ciencias Físicas) and the doctorate in informatics from the University of the Balearic Islands (UIB), Palma de Mallorca, Spain, in 1989 and 2007, respectively.

He is currently a Lecturer with the Department of Mathematics and Computer Science, UIB. His research interests include dependable and real-time systems, fault-tolerant distributed systems, adaptive systems, clock synchronization, dependable communication topologies, and field-bus and industrial networks.

Dr. Proenza is a Member of the IEEE Industrial Electronics Society (IES) since 2009 and Senior Member since 2012. He is also a member of the IES Technical Committee on Factory Automation.