First Analysis of the AVB's Stream Reservation Protocol in the Context of TSN

Daniel Bujosa† Drago Čavka* Inés Álvarez† Julian Proenza†
†Department of Mathematics and Informatics, University of the Balearic Islands, Spain
†{daniel.bujosa, ines.alvarez, julian.proenza}@uib.es
*Faculty of Electrical Engineering, University of Banja Luka, Bosnia and Herzegovina
*drago.cavka@etf.unibl.org

1. INTRODUCTION

In 2005 the IEEE Audio Video Bridging Task Group started working in a new set of projects to provide Ethernet with soft real-time capabilities. Nevertheless, the interest in the work of the group grew. In 2012 the group was renamed to Time-Sensitive Networking Task Group (TSN TG) and its scope broadened, to provide Ethernet with hard and soft real-time communications, flexibility of the traffic and fault tolerance mechanisms for critical applications.

One of the projects developed by the groups is the Stream Reservation Protocol (SRP), which was originally standardized by the AVB TG in [1] and later extended by the TSN TG in [2]. SRP provides Ethernet with support for resource reservation between a transmitter and its receivers. SRP is a key piece to support many of the projects developed by the TSN TG. This is so because resource reservation prevents frame delays beyond predefined limits and losses due to buffer overflow. Furthermore, SRP allows modifying the traffic in run-time, providing a certain degree of flexibility to the network.

Some of the applications targeted by the TSN TG are critical. The mechanisms developed for these applications must exhibit a series of properties to guarantee their correct operation, including those that operate at the network level. Nonetheless, AVB's distributed version of SRP was not designed considering these applications and, thus, it was not designed to fulfil these properties. Moreover, even though the revision of SRP included two new architectures, it still supports the distributed one with no modifications.

In this work we study whether the distributed version of SRP can be deployed in critical systems as it is included in TSN's new standard. To do so, we modelled the protocol (see [4] for a description of this model), using the Uppaal model checker. We carry out the study in the absence of faults, as transient faults can be tolerated using mechanisms such as those shown in [3]; while permanent faults have been left for future work. In this context, we evaluate whether SRP provides termination and consistency, properties that are typical in distributed systems for critical applications. Next we identify scenarios in which termination and consistency for the reservations are not achieved, we discuss the consequences derived from the absence of these properties and we provide a first overview of a series of mechanisms to enforce them.

2. EVALUATION OF THE TERMINATION

We evaluate whether the protocol provides mechanisms to ensure that the reservation process is concluded using the Uppaal model of SRP that we developed. In SRP when listeners receive a message from the talker announcing a new stream, they send a message announcing their intention to bind to that stream in case they are interested in doing so regardless of the availability of resources. In contrast, listeners do not inform the bridges nor the talker when they are not interested in the stream.

2.1 Termination in the Talker

We checked if SRP provides termination to the talkers. We saw that there are scenarios where the talker does not receive any response from the listeners, even in the absence of faults, and the talker waits indefinitely. 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. Not only it can block the decision process, but it can also lead to incorrect decisions due to the lack of knowledge.

A possible solution is to introduce a timeout in the talker. If the talker does not receive any listener response before the timeout expires, it should tear the stream down using the mechanisms available in SRP. The value assigned to this timeout should be adjusted depending on the topology, to ensure that there is enough time for the furthest listeners to communicate their intention to bind.

2.2 Termination in Bridges

A bridge that forwards the request of a talker waits for the responses of the listeners indefinitely. Also, bridges register talker's attributes in all their ports, and they do so for all the talkers willing to transmit. Using the model we saw that there are scenarios where the bridge sends a talker advertise and does not receive any response, because none of the listeners connected to the bridge (directly or indirectly) want to bind to the stream. This situation causes an unnecessary use of memory in bridges and can later prevent the creation of streams with listeners willing to bind.

A possible solution is to introduce a timeout in each bridge and listener. This mechanism would delete the stream registration from the memory of the bridge when it does not receive a response and it would delete the stream registration from the memory of the listener when it doesn't want to bind to the stream. The value assigned to the timeout should be high enough to guarantee that all listeners can announce their will to bind to the stream before the registration is torn down. Otherwise, a listener could try to bind to the stream when the stream has already been deleted in the bridges.

Another option could consist in introducing a timeout only in bridges, but not in listeners. Instead, the declarations in listeners would be removed when receiving a frame. Specifically, the bridge closer to the talker with no reply from the listeners would delete the stream registration from its memory and then transmit the frame to trigger the elimination of the registration in the rest of the network. As in the previous solution the timeout should be high enough to guarantee that all listeners can announce their will to bind to the stream before the registration is torn down. Furthermore, the time should be high enough to guarantee that the first timer that expires if the one in the bridge closer to the talker. Otherwise, more than one bridge might start the transmission of the frame to tear the declaration down. The advantage of this solution is that nodes do not have to have a timer for each stream they receive. Nevertheless, bridges must be able to create a frame to tear the declaration in other devices down.

3. CONSISTENCY

Some of the applications targeted by TSN require the different nodes to carry out coordinated actions. In these applications, consistency is key to guarantee the correct operation of the overall system. For instance, in a system that uses hardware redundancy in the nodes to provide fault tolerance, it is key that nodes exchange their information in a consistent manner. The first step towards achieving consistent communications is to reserve the network resources consistently. However, we found some consistency issues even in the absence of faults.

The problems detected are mainly due to the fact that the propagation of the information related to the reservations is unidirectional. That is, the declaration of a stream transmitted by a talker is forwarded always towards the listeners. Furthermore, when listeners and bridges reply to a stream declaration, the information is only forwarded towards the talker. Thus, not all bridges receive the same information. Moreover, the talker receives the result of the reservation, but listeners do not.

3.1 Consistent reservations in nodes

In SRP talkers are responsible for triggering the creation of streams; whereas listeners must announce their intention to bind. Nevertheless, the information exchanged by the talker and the listeners is limited, *e.g.*, the talker is informed if a listener that wants to bind cannot do it, but cannot know which listener it is. Moreover, as mentioned, this information is not shared with all the nodes. This asymmetry can lead to an inconsistent view of the network.

This problem could be addressed exchanging additional information among the nodes, e.g., it would be possible to exchange information to identify which listeners can and cannot bind. Moreover, it would be important to also exchange information about the reason that prevented a listener from binding; e.g., there is not enough bandwidth. Finally, it would be necessary to guarantee that this information is exchanged in a consistent manner.

3.2 Consistent reservations in bridges

In SRP bridges make reservations using local information mostly and provide limited information about their results to other bridges. Furthermore, the information that bridges do send does not reach all participants, as we said at the beginning of this section. This can lead to inconsistencies in the reservations also in bridges, as the information two bridges receive from another bridge changes depending on whether they are on the path to the talker or to the listener.

Let us use an example to illustrate the type of problem that this can cause in bridges. Let us assume we have a talker attached to a listener through two bridges in a line topology (T-B1-B2-L). When the listener replies to a stream declaration and bridge B2 has enough resources, B2 reserves them. However, if bridge B1 does not have resources it cannot reserve them, but it does not inform bridge B2. Thus, B2 is wasting resources because it reserved them but the stream is not created.

This problem could be addressed exchanging information that is already available in the network. However, just like in the previous solution, it is important to guarantee that this exchange of information is carried out in a consistent manner.

3.3 Consistent communication start

In SRP talkers start transmitting as soon as they receive information of at least one listener ready to receive. Therefore, a talker can start transmitting frames through a stream before it receives the response from all listeners. This can happen, as the paths between a talker and different listeners may differ in length and end-to-end delay. Therefore, a listener 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 would lead to a lack of consistency in the input of the listeners, which could be a problem if they have to carry out coordinated actions.

Similarly to the solution proposed to guarantee termination, the talker could rely on a timeout to wait for all listeners to transmit their will to bind to the stream before starting the communication.

Acknowledgements

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). Drago Čavka was supported by a scholarship of the EUROWEB+ Project, which is funded by the Erasmus Mundus Action II programme of the European Commission.

4. REFERENCES

[1] 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.

[2] 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)*, pages 1–208, Oct 2018.

[3] I. Álvarez, J. Proenza, and M. Barranco. Towards a Time Redundancy Mechanism for Critical Frames in Time-Sensitive Networking. In *ETFA2017*, pages 1–4, Sept 2017.

[4] D. Bujosa, D. Cavka, I. Álvarez, and J. Proenza. First Analysis of the AVB's Stream Reservation Protocol in the Context of TSN. Technical Report A-02-2019, University of the Balearic Islands (UIB), May 2019.