A First Qualitative Comparison of the Admission Control in FTT-SE, HaRTES and AVB

Inés Álvarez, Luis Almeida
Instituto de Telecomunicações
Universidade do Porto,
Porto, Portugal
ines.alvarez.91@gmail.com, lda@fe.up.pt

Julián Proenza
Departament de Matemàtiques i Informàtica
Universitat de les Illes Balears
Illes Balears, España
julian.proenza@uib.es

***Abstract*—Ethernet is gaining importance in fields such as automation, avionics and automotive. In these fields novel multimedia-based applications must coexist with traditional control systems, which leads to high diversity in size, intensity and timing requirements of the traffic traversing the channel. Multimedia traffic is characterised by having large size, low intensity and soft real-time requirements, while control traffic usually conveys small amounts of information with a high intensity and hard real-time requirements. Moreover, many modern applications must support on-line connection and disconnection of participants. Since Ethernet was designed as a general purpose data network protocol it lacks appropriate support for real-time communications and dynamic quality of service management. Several protocols were proposed to cope with these drawbacks, including Flexible Time-Triggered Switched Ethernet and, more recently, Audio Video Bridging. In this paper we discuss the importance of the admission control and make a comparison of the implementations carried out in the aforementioned protocols.**

I. Introduction

In the last years Ethernet has become a leading technology for the development of applications in the fields of automation, avionics and automotive, due to its high bandwidth, low cost, easy scalability, high flexibility and IP-based network compatibility. Novel in-car applications, such as Advanced Driver Assistance Systems (ADAS) or infotainment applications, must coexist with critical control systems. Traffic conveyed by the aforementioned applications is diverse in size, intensity (average occupancy of the channel) and timing requirements. On the one hand ADASs and infotainment applications convey multimedia traffic, with large size, low intensity and soft real-time requirements [1]. On the other hand, traffic conveyed by control applications is characterised by having a small size, high intensity and hard real-time requirements. Moreover, since ADASs and infotainment applications can be launched at arbitrary moments of the system operation, the communication subsystem must support the on-line connection and disconnection of network participants while guaranteeing the adequate Quality of Service (QoS) level to each application.

Nevertheless, Ethernet was designed as a general purpose data network protocol and thus does not provide the required services to support real-time traffic. For instance, Ethernet does not prevent traffic bursts from happening, which can lead to unbounded delays in the transmission of packets and packet loss. In this context, several protocols were proposed to provide Ethernet with real-time capabilities, such as Time-Triggered Ethernet [2] and Dynamic-TDMA Ethernet [3], beyond several real-time industrial Ethernet protocols. Nevertheless, full support for reconfigurability and adaptivity has generally lacked. This led to the proposal of several implementations of the *Flexible Time-Triggered* (FTT) paradigm [4] based on switched Ethernet networks, supporting hard, soft real-time traffic and non-real-time traffic, as well as dynamic management of the QoS, with admission control. Currently two different approaches for FTT over switched Ethernet exist: FTT Switched Ethernet (FTT-SE) and Hard Real-Time Ethernet Switching (HaRTES).

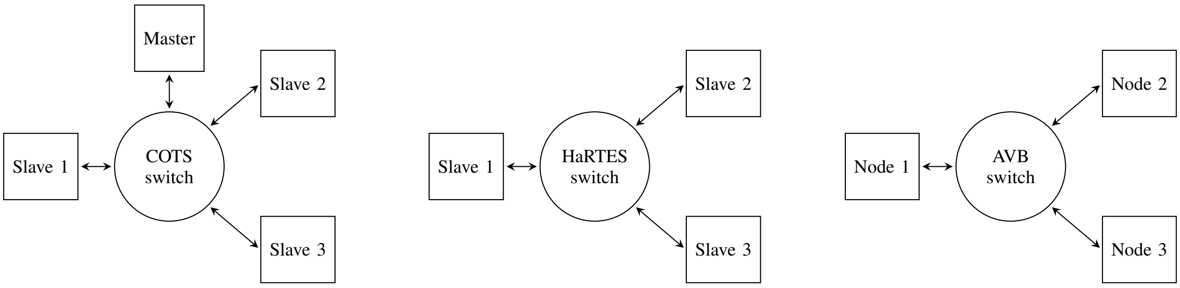
Recently the IEEE published a series of technical standards based on Ethernet that are known as *Audio Video Bridging* (AVB) [5] [6] [7]. Similarly to FTT, AVB aims at providing Ethernet with soft real-time capacities and with flexibility by conveying several types of traffic in the same network infrastructure and supporting on-line changes of the QoS. AVB implements synchronisation services IEEE Std 802.1AS-2011, frame forwarding for AVB traffic IEEE Std 802.1Qav-2010 and admission control of streams in the bridges IEEE Std 802.1Qat-2010 to guarantee bounded maximum latency in the transport of audio and video frames.

When working in dynamic environments that may change in unpredictable manners admission control is of great importance to guarantee timing constraints. In fact it ensures that enough resources are available throughout the network to transmit the traffic accepted, preventing packet delays beyond predefined limits and packet loss due to buffer overflow. In this paper we compare the admission control process in each of the aforementioned protocols from three different points of view: 1) reliability, 2) flexibility and 3) performance. This comparison will be based on a mono-hop architecture. To the authors best knowledge it does not exist in the literature any complete comparison of the aforementioned protocols or the admission control they carry out.

II. Overview of FTT over Ethernet

FTT is a communication paradigm that supports event and time-triggered traffic and provides mechanisms for dynamically changing the communication requirements. FTT follows a master/multi-slave scheme, that is, the communication among the application nodes (slaves) is managed and coordinated by the master node. The master organises the communication in slots of fixed duration called *Elementary Cycles* (EC). In each

©2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.



(a) FTT-SE network architecture (b) HaRTES network architecture (c) AVB network architecture

Fig. 1: Network architectures for FTT-SE, HaRTES and AVB.

EC the master synchronises and triggers the communication in several slaves by means of a special message called *Trigger Message* (TM). The transmission of the TM is carried out in the guard window, isolated from the rest of the traffic. The rest of the EC is divided into two different phases, the synchronous window for time-triggered data messages and the asynchronous window for event-triggered messages.

The communication is carried out through virtual communication channels called message streams. All streams in the network have an application identifier that allows slaves to subscribe to them to receive the desired information. All information related to the streams is stored in the *System Requirements Database* (SRDB) in the master and in the *Node Requirements Database* in the slaves (NRDB). Slaves that want to start communicating with others must first request to the master the creation of a stream. This is done by sending an *add_full_message* to the master specifying the stream identifier, the type of traffic that the stream will convey, the maximum load of the messages transmitted through the stream and the period or minimum time separation of transmissions. The slave that will trigger the creation of the stream is set at the application level. Then, an *add_one_end_connection* message must be sent for each slave willing to attach to a stream, specifying the stream identifier, the node identifier and the role it will play in the communication, i.e., whether it will be a publisher or a subscriber. When the master receives the *add_one_end_connection* message, it checks for the availability of resources in the link connected to the requesting node and in case there are resources available the master updates the SRDB and notifies the slaves to update their NRDBs. The admission control process is repeated for each node that requests the attachment to a message stream.

FTT was implemented on Ethernet in two flavours, FTT-SE that uses Commercial Off-The-Shelf (COTS) switches and HaRTES that uses custom switches (Fig. 1). By using COTS switches FTT-SE (Fig. 1a) preserves some of Ethernet desirable features, such as low cost, high availability of components and general purpose LANs compatibility. Nevertheless, it requires all nodes to be FTT-aware, since non-FTT-compliant nodes could send their traffic at any moment, interfering with the timeless of the protocol messages.

The HaRTES architecture (Fig. 1b) is based on a custom switch that contains the master node. Thus, the master has complete vision of the communication, which allows implementing traffic shaping, i.e., the master confines the traffic sent by the nodes to the appropriate windows. Thus, non-FTT-enabled nodes can be attached to the network, since their traffic is only forwarded by the master during the asynchronous widow when enough bandwidth is available.

III. Overview of Audio Video Bridging

AVB is a set of standards that provides synchronisation services, frame forwarding policies for AVB traffic and admission control in the scope of Ethernet. The admission control is performed by the *Stream Reservation Protocol* (SRP), that allows for the reservation of resources along the path between two or more nodes willing to communicate. SRP considers two classes of real-time traffic, A and B, with different QoS, the former providing lower latencies and higher priority.

The communication is done through virtual channels called streams. A node can be attached to a stream whether as a *talker*, the node that generates the traffic, or as a *listener*, nodes that consume the traffic. Streams are defined by an identifier, the bandwidth required for the communication and the class they belong to. All this information is stored in the switches to manage the admission control and the communication. The creation of streams is triggered by the talker by sending a *talker advertise* message. A bridge that receives a talker advertise checks for the availability of resources in all the output ports that support the stream class, except for the port the talker advertise arrived to. If the required resources are available in the port, the switch forwards the talker advertise and registers its transmission in the *talker advertise table* of the port. Otherwise the switch forwards a *talker failed* message, that contains the reason for the failure. Thus, a node only receives a talker advertise if there are resources available along the path that connects it to the talker. When a listener that wants to join to the stream receives the talker advertise it sends a *listener ready* message through the same port the talker advertise arrived. Otherwise, if the listener receives a talker failed it sends a *listener asking failed* message.

When a switch receives a listener ready message it checks again for the availability of resources in the arriving port. If resources are still available the switch locks them, otherwise it changes the listener ready message to a listener asking failed. Every switch only forwards one listener attribute through the path back to the talker. Nevertheless, since a switch may forward the talker advertise through several ports, it may receive several possibly different responses. All these listener responses must be merged into a single one, as follows:

- If all the listener responses are listener ready, the switch forwards a listener ready, indicating that the reservation has been successful for all the downstream listeners.
- If the switch receives at least one listener ready and one or more *listener ready failed* or listener asking failed, the switch forwards a listener ready failed, meaning that there is at least one path with sufficient resources and at least another one with insufficient resources.
- If the switch receives listener asking failed messages through all the ports, it forwards a listener asking failed.

When a switch transmits the listener attribute through a port, it registers its transmission in the *listener ready table* of said port. When a talker receives a listener ready or a listener ready failed knows that the stream was successfully created for at least one listener and can start the transmission.

IV. Comparing reliability

Networked systems are specially vulnerable to temporary faults in the communication channel due to electromagnetic interference. Nevertheless, none of the presented protocols implement specific mechanisms to face the loss of messages caused by said faults. The consequences of message loss during the admission control are different depending on the approach and the specific message affected by the fault.

In FTT-SE and in HaRTES if a slave request message is lost the admission control will not take place and the slave will not be able to create or attach to the stream and, therefore, to communicate. On the other hand, when the admission control is finished the master sends the result to all slaves by means of a broadcast message. Nevertheless, since transient faults may only affect a subset of links in the network it is possible that some slaves receive the message from the master while others do not receive it. Only slaves receiving the message update their NRDBs, causing an inconsistent view of the network among slaves. Moreover, in FTT-SE asynchronous messages are scheduled using the signalling mechanism. Thus, if the signalling message of a given slave is lost, the master will not schedule the transmission of any asynchronous message for said slave in that EC and the admission control will be delayed until a signalling message reaches the master.

Regarding AVB, both talkers and listeners must periodically send the talker advertise and listener ready messages to announce their intention to maintain the communication. Thus, the loss of one of these messages would delay the admission control until one of the following transmissions gets to the destination. Nevertheless, no retransmissions are considered for messages forwarded by the switch. Therefore, if a talker advertise message is lost when forwarded by the switch only nodes receiving it will be able to attach as listeners. On the other hand, if the message lost is the listener attribute the talker will not be able to initiate the communication even if the resources are reserved along the path between the talker and the listeners. Moreover, since the messages are lost after being registered in the output ports of the switch, further talker advertise and listener ready messages with the same stream identifier are not forwarded through those ports, which makes it impossible for nodes to latter attach to the communication.

Temporary faults in nodes are not considered in any of the presented protocols, and their analysis is left as future work. With respect to permanent faults we looked in particular to the possible presence of *single points of failure* (SPoF). In what concerns FTT-SE, the communication channel, the master node and the switch are SPoF and no mechanisms are considered to deal with the failure of said components. Therefore, a link affected by a permanent fault would lead to the isolation of the node attached to it, preventing it from communicating and, in case the link connects to the master node, preventing the transmission of the TM and therefore the slaves would not be able to communicate. In case of a permanent fault in the master, again, the communication among slaves would not take place due to the lack of the TM. Finally, in case of a permanent fault in the switch the communication would be seriously jeopardised, since the switch is responsible for forwarding all messages. Regarding the HaRTES architecture, both, the communication channel and the HaRTES switch represent SPoF. In the project *Fault Tolerance for Flexible Time-Triggered* (FT4FTT) several mechanisms were proposed to eliminate the existing SPoF of the HaRTES architecture [8], by means of spatial, time and information redundancy. Finally in AVB we can also find two SPoF, the communication channel and the switch. No mechanisms to eliminate those SPoF are considered in AVB, nevertheless the IEEE is currently working in the definition of the second generation of AVB standards, a.k.a *Time Sensitive Networking* (TSN) which will provide support for spatial redundancy of the communication channel.

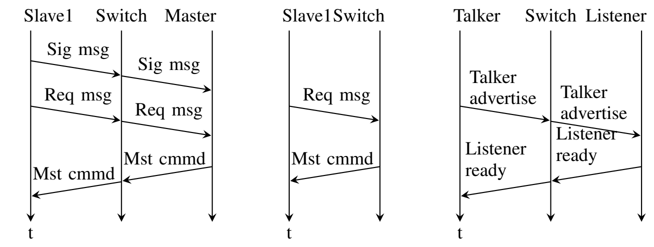
V. Comparing flexibility

The admission control is said to provide protocols with flexibility, since it allows changing the QoS of the communication on-line. Nevertheless, different levels of flexibility can be achieved depending on the specific implementations and the services considered by the protocols.

Both FTT-SE and HaRTES allow slaves to dynamically change the QoS requirements of an exiting stream by sending a message to the master. When receiving the request the master evaluates whether the resources are sufficient to carry out said change in all links and, if so, it broadcasts the new QoS parameters to all slaves to upload their NRDBs. Conversely to FTT, nodes that want to change the QoS provided by a given stream in AVB must first tear the existing stream down. Moreover, after tearing down the stream nodes must wait an amount of time predefined by the protocol before requesting the creation again. During this time other nodes can perform the reservation of resources for different streams and exhaust the available resources, thus preventing the creation of the updated stream.

On the other hand, in both FTT and AVB streams belong to a class. Specifically, a stream in FTT-SE and HaRTES can be synchronous or asynchronous, while an AVB stream can belong to class A or B. All streams belonging to the same class share resources and follow the same policies. Neither FTT-SE, HaRTES or AVB provide with support for dynamically changing the stream group, instead a new stream must be created and all nodes involved in the communication must then attach to the new stream.

Finally, FTT-SE and HaRTES allow for a slave participant in the network to declare the creation of a stream and the attachment of other slaves to that stream. This way legacy



(a) FTT-SE messages (b) HaRTES messages (c) AVB messages

Fig. 2: Messages exchanged during the admission control.

applications can be deployed on top of FTT if there is a node that performs said requests. In AVB this is not possible since the node sending the talker advertise must be the same one that latter on carries out the transmission through that stream.

VI. Comparing performance

To compare the performance obtained by each approach we decided to measure the overhead caused by the process in terms of number of messages exchanged. We will consider a simple scenario where only two nodes communicate.

During the FTT admission control process, the slaves send at least three requests, one to request stream creation, one for the publisher and one for each subscriber to attach to the stream. Slaves can send several requests in the same message. Therefore, a slave that wants to attach to a stream can also request its creation in the same message. Afterwards, the master sends at least one message announcing the result of the admission control. That is, as depicted in Fig. 2b, in HaRTES two messages are exchanged during the admission control per each participant, that is, four messages are exchanged in our simple scenario. On the other hand, in FTT-SE slaves that want to attach to a stream must first inform the master of pending request messages using a *signalling message*. The master schedules the transmission of the asynchronous messages carrying the slave requests and inform the slaves by means of the TM. Therefore, as seen in Fig 2a in FTT-SE the admission control requires the transmission of six messages per participant, that is, twelve messages for our simple scenario.

With respect to AVB, as depicted inf Fig. 2c at least four messages are exchanged during the admission control. First, a talker that wants to communicate sends a talker advertise message to the switch, that will in turn check for the availability of resources in each output port and then forward the corresponding talker attribute (talker advertise or talker failed) through them. Then, nodes receiving a talker attribute transmit a listener ready or a listener asking failed through the incoming port to the switch, that checks again for the availability of resources and sends the corresponding listener attribute. Therefore, we can conclude that the overhead in the communication introduced by FTT-SE during the admission control is significant when compared to HaRTES or AVB.

VII. Conclusions and future work

This paper presents a qualitative comparison of three admission control protocols that provide switched Ethernet with timing and flexibility services: FTT-SE, HaRTES and AVB; considering reliability, flexibility and performance. Table I summarizes the comparisons carried out. Concerning reliability, we observed that none of the presented protocols provide adequate services to face transient faults, while HaRTES is currently the only one handling permanent faults; nevertheless the IEEE is developing a new set of standards to overcome some of these problems. In terms of flexibility, we saw that FTT allows to dynamically change the QoS of the streams, providing a higher level of flexibility than AVB. Finally, concerning the performance obtained by each protocol we concluded that FTT-SE introduces a significant overhead in the communication channel when compared to HaRTES and AVB. We are currently working on extending this comparison and carrying out a quantitative one, through simulation.

TABLE I: Summary of the comparison.

	Reliability		Flexibility			Perf.
	Trans. Faults	Perm. Faults	QoS change	Class change	Legacy nodes	Overh.[a]
FTT-SE	✗	✗	✓	✗	✓	12
HaRTES	✗	✓	✓	✗	✓	4
AVB	✗	✗	✗	✗	✗	4

[a] Minimum number of massages exchanged during the admission control.

Acknowledgements

This work was supported by project TEC2015-70313-R (Spanish Ministerio de Economa y Competividad), FEDER funding and by a grant of the Instituto de Telecomunicações, Portugal.

References

[1] L. Lo Bello, "Novel trends in Automotive Networks: A perspective on Ethernet and the IEEE Audio Video Bridging," in *Proc. 19th IEEE Conf. on Emerging Technologies & Factory Automation (ETFA)*, Sep. 2014.

[2] H. Kopetz, A. Ademaj, P. Grillinger, and K. Steinhammer, "The Time-Triggered Ethernet (TTE) Design," in *8th IEEE International Symposium on Object-oriented Real-time distributed Computing (Seattle, Washington: TU Wien)*, May 2005, p. 2233.

[3] G. Carvajal, L. Araneda, A. Wolf, M. Figueroa, and S. Fischmeister, "Integrating dynamic-tdma communication channels into cots ethernet networks," *IEEE Transactions on Industrial Informatics*, vol. PP, 2016.

[4] P. Pedreiras and L. Almeida, "The Flexible Time-Triggered (FTT) paradigm: an approach to QoS management in distributed real-time systems," in *Proc. Int. Parallel and Distributed Processing Symposium*. IEEE Computer Society, 2001.

[5] *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.

[6] *IEEE Standard for Local and metropolitan area networks, Virtual Bridged Local Area Networks, Amendment 14: Stream Reservation Protocol*, IEEE Std. 802.1Qat, 2010.

[7] *IEEE Standard for Local and metropolitan area networks, Virtual Bridged Local AreaNetworks, Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams*, IEEE Std. 802.1Qav, 2009.

[8] D. Gessner, J. Proenza, and M. Barranco, "A Proposal for Managing the Redundancy Provided by the Flexible Time-Triggered Replicated Star for Ethernet," in *Proc. 10th IEEE Int. Workshop on Factory Communication Systems (WFCS)*, Toulouse, France, May 2014.