UNIVERSITÀ DEGLI STUDI DI CATANIA
Dipartimento di Ingegneria Elettrica, Elettronica ed Informatica

Dottorato di Ricerca in Ingegneria Informatica e delle Telecomunicazioni
XXVIII Ciclo

ETHERNET IN AUTOMOTIVE: THE CASE OF IEEE AUDIO VIDEO BRIDGING

Ing. Giuliana Alderisi

Coordinatore
Chiar.ma Prof.ssa V. Carchiolo

Tutor
Chiar.ma Prof.ssa L. Lo Bello

to my grandmother, Nonna Paola

SOMMARIO

I veicoli moderni sono equipaggiati con più di 100 sistemi embedded, detti Electronic Control Unit (ECU), che gestiscono il funzionamento del veicolo stesso. Sono ad esempio ECU i seguenti dispositivi: Engine Control Module (ECM), Powertrain Control Module (PCM), Brake Control Module (BCM), General Electric Module (GEM), e altri ancora. Con l'incremento del numero di ECU, crescono e si evolvono le esigenze delle reti in automotive. Sempre più spesso il traffico generato in un dominio funzionale è richiesto, per essere elaborato o più semplicemente mostrato all'utente, da un altro dominio. Si pensi ad esempio alle ECU dedite a monitorare la temperatura del motore per avviare procedure di raffreddamento in caso di necessità. Le informazioni rilevate da questi sistemi sono però allo stesso tempo richieste anche per informare, in tempo reale, il guidatore sullo stato del motore stesso. Questo è un classico esempio di condivisione da parte di due domini funzionali differenti (powertrain e infotainment) di un flusso di traffico. Le connessioni punto-punto non sono più valide (per complessità di manutenzione, peso dei cavi, conseguente incremento nel consumo di carburante ecc..). Per questo le soluzioni switched, ed

in particolare Ethernet switched, sono considerate promettenti. Tra tutti i protocolli Ethernet-based, in particolare, lo standard IEEE Audio Video Bridging (AVB) ha suscitato un notevole interesse sia presso la comunità scientifica che presso l'industria automobilistica.

Tuttavia, sebbene il supporto a traffico di tipo multimediale sia ormai largamente disponibile con molti protocolli di comunicazione, ivi compreso AVB, le reti Ethernet-based non forniscono supporto per la trasmissione di traffico di controllo o deterministico, detto traffico schedulato.

Il contributo di questo lavoro di tesi è duplice: viene presentata un'indagine sul protocollo AVB, in realistici scenari automotive, confrontando AVB anche con altri protocolli di comunicazione attualmente impiegati in automotive, e vengono presentate due soluzioni innovative, chiamate AVB_ST e AVB_P, che migliorano AVB rendendolo capace di fornire supporto al traffico schedulato.

ABSTRACT

Modern vehicles are equipped with more than 100 embedded systems, called Electronic Control Unit (ECU), which handle the functioning of the vehicle itself. Engine Control Module (ECM), Powertrain Control Module (PCM), Brake Control Module (BCM), General Electric Module (GEM) are some examples of ECU. With the growing number of ECU, the needs of the automotive networks change. More often the traffic generated in a functional domain is required by another functional domain to be processed, or even just shown to the driver. For instance, the ECU used to monitoring the engine temperature send traffic information to those units in charge of starting the cooling procedure if needed. At the same time, that traffic is required to be shown directly to the driver to inform him/her on the state of the engine, in real-time. This is a classic example of a traffic flow shared among two different functional domain (powertrain e infotainment). Point-to-point connections are not effectual (for maintenance complexity, cable weight, consequent increasing of fuel consumption etc..). For this reason, switched solutions, like switched-Ethernet solutions, are considered promising. Among all the Ethernet-based protocols, both

the scientific community and the automotive industry think that the IEEE Audio Video Bridging (AVB) standard is a promising candidate. However, although the support to multimedia is largely provide by a number of communication protocols, including AVB, Ethernet-based networks do not provide support to deterministic control traffic, here called schedule traffic.

The contribution of this dissertation is twofold: first an investigation on AVB, in realistic automotive scenarios, even comparing AVB to others communication protocol currently used in automotive, is presented. Second, two novel solutions, called AVB_ST and AVB_P, that extend AVB providing support to schedule traffic are proposed.

CONTENTS

CHAPTER

ONE

INTRODUCTION AND MOTIVATIONS

Ethernet is nowadays considered a promising candidate for in-car communications, thanks to the high bandwidth provided (100 Mbps onwards) that paves the way for applications, like Advanced Driver Assistance Systems (ADASs), which make the volume of exchanged data in automotive communication continuously grow. Other strengths of Ethernet are the well-experienced technology, that allows for better testing, maintenance and development, and the wide use and open standardization, that entail a large availability of high-quality chips on the market and therefore low production costs. In addition Ethernet technology is scalable, thus meeting the requirements imposed by today's automotive systems, where the number of nodes to interconnect steadily increases. As reported in [1], the Ethernet Unshielded Twisted Single Pair successfully passes the electromagnetic compatibility immunity test, thus showing that Ethernet can correctly operate as automotive network. Furthermore the Reduced Twisted Pair Gigabit Ethernet (RTPGE) PHY is also an appealing solution currently under consideration for automotive communications at 1Gbps within the IEEE 802.3 bp Task Force [2][3]. Another strong point in favour of Ethernet is the support for the Internet Protocol (IP) stack that opens the way to enhanced navigation functionalities, remote diagnostics and location-based services. Investigations into the usage of Ethernet in automotive is in progress in academia, the car industry and companies producing automotive electronic devices. Attention is paid to the IEEE AVB standard family [4],[5] and [6] for multimedia, infotainment and driver assistance. AVB is then considered a promising candidate for the enhanced QoS provided, the IEEE standardization, no need for license fees and its cost and quality, comparable to those of standard Ethernet. Although AVB is, for all the reasons above, considered appealing, it is a new technology and the car industry is not inclined to the change, if it not strongly motivated by economic profit. So assessments of AVB and between AVB and other technologies currently applied in automotive networks, are needed.

This need motivates the work presented in the first part of this dissertation, which is a deep study of the IEEE Audio Video Bridging (AVB) protocol. In fact chapters 4-5 show performance assessments of AVB in a realistic automotive scenarios, thus:

- Proving the AVB capabilities to deal with automotive functional domains, such as ADAS, Infotainment and Multimedia.

- Investigating multiple topologies, in order to identify which is the more convenient to use for AVB networks.

- Comparing AVB and other technologies currently applied in automotive networks in order to prove if AVB is able to outperform those technologies, thus motivating the changing.

Furthermore, a recent joint study by Broadcom and Bosch estimated that using "unshielded twisted pair (UTP) cable to deliver data at a rate of 100Mbps, along with smaller and more compact connectors can reduce connectivity cost up to 80 percent and cabling weight up to 30 percent" [7]. For this reason the target is to use an Ethernet network as a backbone between all the functional domains. This entails that the Ethernet-based protocol should provide support to control, hard real-time, traffic as well as to multimedia/Infotainment traffic. The first generation of AVB standard, as described in section 2.5, is not able to do that. For this reason a committee (the Time Sensitive Networking (TSN) group) is entitled to create a second generation of AVB standard, which will deal with control traffic and will present also other features (see section 2.5).

This expression of interest towards the so called *Deterministic Ethernet*, in addition to the results of the performance assessments of AVB, motivate the second part of this dissertation, which presents two novel novel solutions that extend AVB in order to make it able to provide support to the scheduled traffic.

1.1 Structure of this Dissertation

This dissertation is organised in nine chapters (including this introduction) as follows:

Chapter 2 provides basic definitions related to the most relevant automotive network protocols. It also presents a concise description of the AVB features.

Chapter 3 summarizes all the assessments of the AVB protocol, for ADAS and infotainment traffic, that are presented in this dissertation.

Chapter 4 explores the AVB capabilities in a realistic automotive scenario. The study is made through a simulative assessment and the results confirm AVB as a promising candidate for in-car communication.

Two comparative assessments, i.e., between AVB and Time-Triggered Ethernet and between AVB and Media Oriented System Transport, are presented in Chapter 5 and Chapter 6 respectively. The results, in terms of latency and jitter, are discussed highlighting the strengths and the weaknesses of AVB, for different application domains.

Chapter 7 presents AVB_ST, an extension of AVB able to provide support to hard real-time control traffic on AVB networks. This solution is evaluated through a comparative assessment between AVB_ST, AVB and TTE.

In Chapter 8 another novel solution for providing support to scheduled traffic on AVB networks, for both automotive and industrial applications, is presented: AVB_P. AVB_P exploit a phase-based schedule thus simplifying the procedure for allowing temporal isolation for the scheduled traffic.

Chapter 9 summarizes the conclusions and proposes further works related to the presented subject.

CHAPTER

TWO

TECHNICAL BACKGROUND

2.1 Automotive Networks

In the past, the primary function of the car was to move people efficiently. Today cars have instead become sophisticated networked embedded systems in which electronic components and control systems, interconnected by communication networks, implement functions to improve performance and safety. Several functional domains are found in a car, that correspond to different applications and feature diverse constraints [1]. Traditional domains include:

- Powertrain, which is relevant to the control of engine and transmission.

- Chassis, which deals with the control of the vehicle's stability, dynamics, agility, according to steering/braking solicitations and driving conditions (e.g., ground surface, wind, etc.).

- Body, which implements comfort functions, such as, the control of doors, windows, seats and air conditioning.
- Multimedia/Infotainment, which is relevant to Audio and Video players, TV, Rear Seat Entertainment, navigation information services, and also includes Human Machine Interface, such as, advanced display technologies, haptic devices giving feedback to the driver, speech input to lower the driver's distraction when operating on navigation devices, radio or mobile phones.
- Camera-based Advanced Driver Assistance Systems (ADAS), which implement functions that assist the driver to improve driving safety, such as Lane Departure Warning, Traffic Sign Recognition, Night vision and Bird's-eye view.

2.2 Local Interconnect Network (LIN) and Controller Area Network (CAN)

Local Interconnect Network (LIN) LIN [8] was born as a projects started in 1998 by a consortium of car-makers (such as Audi,BMW, Daimler-Chrysler,Volvo and Volkswagen) together with Motorola. Since 2001 it is introduced in car series production but the first version of LIN became a standard (open standard) in 2000 and in 2003 a second version, i.e., LIN-2.0 was standardized. Today, it still is being used in the automotive application domain, as well as CAN. LIN provides speeds of up to 20 KBps and for this reason is typically used in body and comfort systems to control devices like seat control, light sensors and climate control [9].

Controller Area Network (CAN) The Controller Area Network (CAN) [10],was developed in the fist half of the '80s by Bosch. Today CAN is the widely used vehicular network in the automotive industry. CAN provides a datarate of 1 Mbps and is used in several application domains such as chassis, body/comfort, diagnostic and, above all, powertrain [9]. Over the years several different CAN standards have been developed and used in different applications and, more recently, an flexible data-rate version of CAN, called CAN FD was presented by Bosh. CAN FD uses a different frame format, thus allowing for a different data length as well as optionally switching to a faster bit rate after the arbitration.

2.3 Time Triggered Ethernet (TTE)

Time-Triggered Ethernet (TTE) [11], SAE standard (AS6802), supports three different traffic types, i.e., time-triggered (TT), rate-constrained (RC) and best-effort (BE). Time-triggered (TT) messages are transmitted at predefined times and have precedence over the other kinds of traffic. They are suitable for brake-by-wire, steer-by-wire systems (avionics). Rate-constrained (RC) messages are sent at a bounded transmission rate that is enforced in the network switches, so that for each application a max predefined bandwidth, together with delays and temporal deviations within given limits, are guaranteed. RC messages do not follow a sync time base, so multiple transmissions may occur at the same time and messages may queue up in the switches, leading to increased transmission jitter. Rate-constrained messages are suitable for multimedia or ADAS automotive applications, like the ones addressed in this paper. Best-effort (BE) messages use the spare bandwidth left from the higher priority classes and so have no guarantee on the delay and on the delivery at the destination. BE traffic is suitable for legacy Ethernet traffic (e.g. Internet protocols) without any QoS requirement. TTE provide support multiple topologies:

- Star topology: a single TTE switch is connected to multiple end systems. Each path between two end systems, i.e., through the switch, is called *channel*. In a star topology a channel, by definition, consists of a switch and a set of links connecting that switch directly to end-nodes.

- Tree topology: several switches connected to both end systems and other switches. In this case, a channel may consists of more than a switch and all the links connecting the switches among them and to the end systems.

The Time-Triggered Ethernet message format is based on the format of the standardized Ethernet message according to the IEEE 802.3 standard [12]. The contents of the Type Field, for uniquely identifying a Time-Triggered Ethernet message and the associated message protocol, is 0x88d7 [13].

2.4 Media Oriented System Transport (MOST)

The Media Oriented Systems Transport (MOST) [14] is a master-slave, function-oriented, high-speed multimedia technology able to network

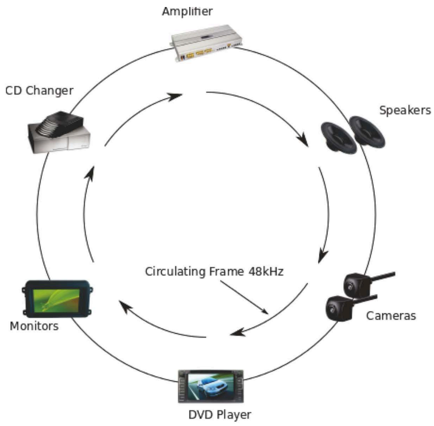

Figure 2.1: *MOST ring topology*

up to 64 devices, called MOST nodes. The MOST specifications allow for two kinds of topology, star and ring.

In the star topology, all the slaves are connected to the master (which handles transmissions according to a logical ring between nodes), while in the ring each node connects to two other nodes forming a physical ring, as shown in Figure 2.1

MOST is a synchronous network in which the master, called Timing Master, provides the system clock by circulating continuous frames, called MOST frames, with a fixed frequency. All the other devices, i.e., the Timing Slaves, synchronize their operation to these frames. Depending on the bandwidth offered, there are three different MOST versions, i.e., MOST25, MOST50 and MOST150, which provide 25, 50 and 150 Mbps, respectively. In addition, MOST 150 provides support for transmitting isochronous data traffic, i.e., a kind of traffic characterized by a transmission rate not necessarily equal to the sample frequency of the MOST frame. A MOST network provides the following types of data transport mechanisms:

- The *Control Channel*, used to transfer packets for commands, status and diagnostics messages to specific addresses and to initiate the streaming data connection between sender and receiver.

- The *Streaming Data Channel*, used to transfer continuous data streams that demand high bandwidth and require time-synchronized transmission. The connections are dynamically managed through appropriate control messages sent over the Control Channel. A connection label and the required bandwidth characterize a streaming connection. Among the real-time streams, that here are called streaming data, a distinction can be made between synchronous and isochronous data, i.e., between the data with the same sample rate as the MOST network and data that are received or sent at a rate that is unrelated to the MOST system clock. The bandwidth allocated for streaming data connections is always available and is reserved for the dedicated streams, so there are no interruptions, collisions, or delays in the transport of the data streams.

- The *Packet Data Channel*, used for transmissions that require high bandwidth in a burst-like manner, e.g., for transmitting large data blocks.

The MOST150 frame, consists of 3072 bits (384 bytes). Four of the first 12 bytes are used for the Control Channel, i.e., for sending control data, and the next 372 bytes are used for the Packet Data

Channel and the Streaming Data Channel [15]. The MOST150 frame is therefore subdivided into three regions:

- The header region, used for the transmission of management, preamble and control data (e.g., to manage connections);
- The stream data region, used for the transmission of real-time streams (e.g., audio and video) related to the Streaming Data Channel;
- The packet data region, used for the transmission of non-real time data belonging to the Packet Data Channel (e.g., TCP/IP traffic).

When a slave is traversed by the MOST frame it reads and/or writes data in the frame. In the Control Data channel, arbitration is carried out in a CSMA way (note that only four bytes of control data can be transmitted in each frame, so a complete transmission takes from 6 to 18 frames [14]). In the Stream Data channel, for each established connection, space is allocated within the MOST frame for streaming data transmission. So, when a slave is traversed by the MOST frame, it knows exactly where to read and/or write data. Finally, in the Packet data channel access is ruled by a token passing policy.

The Media Oriented Systems Transport (MOST) protocol [14] has been introduced in the infotainment domain. MOST 150 provides a very high payload efficiency (defined as the ratio between the payload and the effective sent data [16]), however, there are some limitations. First, spreading of the MOST technology outside the automotive domain is still limited, and the smaller market penetration entails higher production costs. Second, since MOST is a bus system, its total network bandwidth is shared among all connected devices, while switched networks utilize bandwidth more efficiently.

2.5 IEEE Audio Video Bridging (AVB)

The IEEE 802.1 Audio/Video Bridging (AVB) standard provides the specifications for time-synchronized low latency streaming services through IEEE 802 networks and includes three specifications, i.e.

- The IEEE 802.1Qat [4] Stream Reservation, which allows for the resource reservation within switches (buffers, queues) along the path between sender and receiver.

- The IEEE 802.1Qav [5] Queuing and Forwarding for AV Bridges, which splits time-critical and non- time-critical traffic into different traffic classes extending methods described in the IEEE 802.1Q standard and applies traffic shaping at the output ports of switches and end nodes to avoid traffic bursts.

- The IEEE 802.1AS Time Synchronization [6], which provides precise time synchronization of distributed local clocks with a reference that has an accuracy of better than 1 μs.

In [4], the term *talker* is used for a traffic flow source while *listener* is used for a traffic flow destination. The traffic classes for which is possible to reserve resources (bandwidth, buffer and queues) are called Stream Reservation Classes (SR-Class). Although the AVB standard foresees up to seven SR-Classes, the configuration parameter, i.e., the

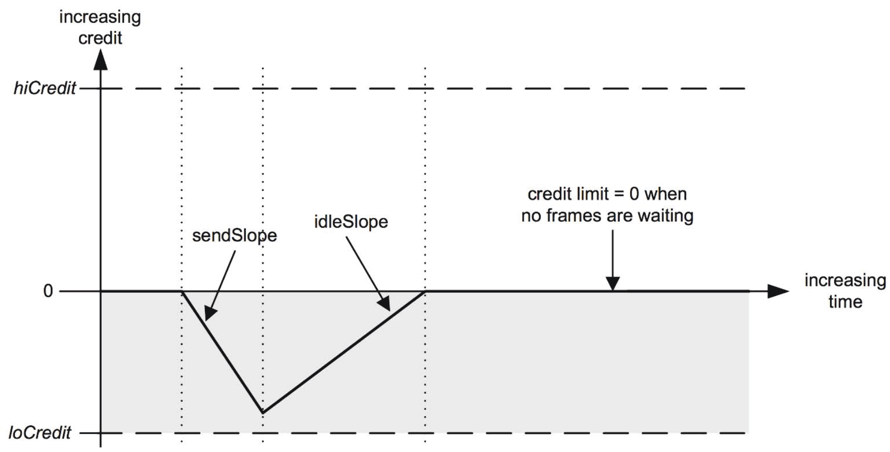

Figure 2.2: *Credit-based shaper operation*

class measurement interval, is specified just for two traffic classes, namely Class A and Class B. For those two traffic classes, a fixed upper bound for latency, for seven hops within the network, is guaranteed:

- Class A, that provides a maximum latency of 2ms;

- Class B, that provides a maximum latency of 50ms.

In an AVB network, both the talker and the listener are in charge of guaranteeing that the path is available and of reserving the resources.

In [5] time critical and non-time critical traffic are split into different traffic classes extending the methods described in the IEEE 802.1Q standard and credit-based shaping (CBS) is applied at the output ports of the switches and at the end nodes to avoid traffic bursts. An example of how CBS works is shown in Figure 2.2 [5].

Each SR-Class traffic class has an associate *credit* parameter, and two limits, *loCredit* and *hiCredit*. *Credit* decreases at the *sendSlope* rate defined for the class during the transmission of frames belonging to that class. *Credit* increases at the *idleSlope* rate defined for the class whether the frames of that class are waiting for transmission or no more frames of the class are waiting, but credit is negative. *hiCredit* and *loCredit* are computed as in Equations 2.1 and refeq:locredit, respectively.

$$hiCredit = maxInterferenceSize \times \frac{idleSlope}{portTransmitRate} \tag{2.1}$$

$$loCredit = maxFrameSize \times \frac{sendslope}{portTransmitRate} \tag{2.2}$$

where *portTransmitRate* is the transmission rate, in bits/s, that the underlying MAC service provides and *maxInterferenceSize* is the maximum size, in bits, of any burst of traffic that can delay the transmission of a frame that is available for transmission for this traffic class. *Credit* is immediately reset to zero when credit is greater than zero and no more frames of that class are waiting.

With a careful planning of periodic execution and mapping to the high priority queues within switches, AVB is able to guarantee low jitter.

The standardization process of AVB is still in progress. The Time-Sensitive Networking Task Group of IEEE 802.1 (TSN) is working on AVB extensions with the aim of making AVB capable to handle time-sensitive traffic. Several amendments are in progress, which deal with enhanced time synchronization [17], robustness [18], redundancy [19], stream reservations [20], and support for scheduled traffic [21], i.e., a traffic class that requires to schedule frame transmission based on a timing reference (derived from the IEEE 802.1AS standard).

Another delay factor under study by the TSN group is the latency introduced by store-and-forward bridges. In fact, while in the presence of interfering frames there is little advantage in using cut-through bridges, if interference is removed, the target port of the bridge is idle when scheduled traffic frames arrive and the benefit of using cut-through bridges becomes significant. In [22], assuming a 64-byte internal buffering (i.e., a 64-byte cut-through point), it is shown that the latency achieved by cut-through bridges is almost half of that obtained using store-and-forward bridges (i.e., 2.074 μs vs. 4.122 μs). Moreover, such latency can be guaranteed and is also independent of the size of the time-sensitive frame.

Stream Reservation Protocol (SRP) A Stream Reservation (SR) Class is defined as a traffic class that is forwarded on the network according to the procedures described in the Multiple Stream Reservation Protocol (MSRP). Currently, only two SR traffic classes, namely, Class A and Class B, are supported, as the specifications in [4] define the class measurement interval parameter for these two traffic classes only. The class measurement interval is a period of time during which a station can place up to *MaxIntervalFrames* data frames, of a size no longer than *MaxFrameSize* each, into the queue associated to the stream. The *MaxIntervalFrames* and *MaxFrameSize* parameters are used by the MSRP. The first is defined as a frame rate, in a frames-per-class measurement interval, for a given stream, while the second is the maximum number of bits per every frame of the same stream. The class measurement interval value is 125 μs for Class A and 250 μs for Class B. The SR classes are handled by the CBFQ. The SRP according to [6], foresees that each traffic flow of SR Class A and SR Class B, before being sent on the network, must be registered on the bridges that will forward it on the network to the listener. The registration is possible if, and only if, every node that participates in the forwarding process of the stream, from the talker to the listener, has sufficient bandwidth and resources. If the registration is successfully completed, the amount of bandwidth required by the new flow is subtracted, at each port that is traversed by this flow, from the total amount of bandwidth available. Once the percentage of bandwidth to be reserved to the Classes A and B is determined, the remaining one is left over for best effort traffic. Three of the parameters that are involved in these procedures are:

- portTransmitRate – PTR: defined in the previous paragraph as the transmission rate, in bits/s, that the underlying MAC service provides.

- adminIdleSlope(N): The bandwidth, in bit/s, that has been requested to be reserved for use by the queue associated with traffic class N.

- deltaBandwidth(N): A percentage of the PTR; this is the bandwidth that can be reserved for use by the queue associated with traffic class N.

IEEE 802.1Qav [4] defines eight traffic classes and foresees that at least one of them must be used as SR class. Traffic classes that are not used as SR Classes are left to best effort traffic, without any bandwidth reservation or guarantee. Each traffic class has a priority level (from 0 to 7, where 7 is the highest priority).

Synchronization in IEEE Audio Video Bridging Clock synchronization is performed as described in the IEEE 802.1AS standard [6], a variant of the IEEE 1588 [23] standard that was updated to the latest version in 2011. To synchronize the network according to IEEE 802.1AS, all the nodes, both bridges and end-stations, are required to be time-aware stations, i.e., systems that make explicit reference to time. Among the nodes, one is selected as a reference node for synchronization, according to the Best Master Clock Algorithm (BMCA) and is called the grandmaster node. The BMCA determines the grandmaster and constructs a time-synchronization spanning tree rooted at the grandmaster. The synchronized time is transported from the grandmaster to other time-aware systems via the time-synchronization spanning tree. In the IEEE 802.1AS standard procedures for synchronization and syntonization are described (here we recall that two stations are synchronized if their clocks do not differ in time, while they are syntonized if they use, for all the time interval measurements, the same time base). The time synchronization capability of the IEEE 802.1AS for industrial applications derives from the IEEE 1588 protocol [23]. The performance of the IEEE 802.1AS protocol was investigated for industrial automation [24] and for automotive scenarios [25][26] and the results showed that typical implementations have an accuracy better than ± 300 ns for 7 hops.

CHAPTER

THREE

ON THE ASSESSMENTS OF IEEE AUDIO VIDEO BRIDGING FOR ADAS AND INFOTAINMENT

Multiple and heterogeneous networks support the different automotive functional domains [27][9]. As stated [28], although recently automotive industry mainly uses MOST, communication for camera-based Advanced Driver Assistance Systems is mainly supported by point-to-point Low-Voltage Differential Signalling (LVDS) wires or by analogue Colour Video Blanking Signal (CVBS) cables can still be found. However, the usage of point-to-point dedicated connections for audio and video content has to be discontinued due to wiring complexity, that affects maintenance, reliability and weight, and costs, in terms of wires, connectors and fuel consumption.

Moreover, a growing interest towards Ethernet as an in-vehicle network for today's cars has been largely shown by both industry and academia. Among all the Ethernet-based communication protocol,the IEEE Audio Video Bridging (AVB) standard is a promising candidate for automotive communication.

The interest towards AVB comes from the academia [3, 29, 30], and also from the manufacturers of automotive electronic devices (e.g., Freescale [31, 32], Broadcom [33], Bosch [34], Continental [35]). The motivations for this interest are summarized in [36], i.e., the enhanced QoS support provided, together with the IEEE standardization, no need for license fees and its cost and quality, that are comparable to those of standard Ethernet.

For this reason, in chapters 4,5 and 6, we present three assessments on the suitability of the IEEE AVB standards [4, 5, 6], in the ADAS and infotainment domain. The papers goal is not to define a winner, but to investigate the behaviour of these technologies when supporting ADAS and multimedia traffic under a high and varying workload.

The novelty of the contribution in [37], discussed in chapter 4, is a quantitative performance evaluation of AVB for ADAS, multimedia and infotainment, obtained through simulation in scenarios that use realistic traffic patterns. The purpose of this work is to demonstrate that, even in conditions of high workload, AVB performance are adequate and promising.

The work in [38], described in chapter 5, extends the work [37] in several respects. First, different topologies, i.e. both a single and a double star, are addressed. Second, the simulation scenarios are very different, as in [38] there are cross-domain flows, that are not present in the other work, and a highly varying workload with significant traffic bursts, while in [37] the traffic distribution is constant. Last, but not least, the work [37] deals with AVB only, while this paper deals with both AVB and TTE and provides comparative assessments.

Among the automotive functional domains in which AVB, TTE and MOST might step in, the works described in chapter 5 and chapter 6 focus on ADAS, multimedia and infotainment systems.

The results of the studies in [37] and [38] motivate a performance comparisons between AVB and MOST in typical automotive scenarios. In fact, although currently MOST is very popular for in-car multimedia/infotainment, it is very likely that in the future it will be replaced by AVB. The main reason is that AVB is an Ethernet-based standard, and this paves the way for both a broad spreading of the technology and the availability of multiple technology providers, while MOST is only used in the automotive domain and its small market penetration entails high production costs. This is a non-trivial issue, as in the automotive domain costs should always be as low as possible. The comparison between AVB and MOST is presented in chapter 6 and especially focuses on the camera-based ADAS and multimedia domains, as such domains are the expected battlefield for the two competing protocols. In chapter 6, only MOST150 is taken into account, as it is the MOST version that provides a bandwidth comparable with the AVB one.

3.1 Related Work

A number of recent studies addressed the performance of Ethernet as in-car network. In [16], MOST 150 and Ethernet AVB for ADAS and infotainment support are addressed and a discussion about the payload efficiency and network utilization of the two networks is provided. The outcome of the comparison is that MOST 150 outperforms AVB as far as payload efficiency is concerned, while AVB is the preferred solution in terms of network utilization, thanks to the multiplied bandwidth, that allows it to support ADAS (for instance, multiple live video streams for cameras), while MOST, that shares the available bandwidth, cannot support the same amount of traffic on the same network. The comparison provided in [16] is based on calculations, not on simulations. Moreover, it only refers to payload efficiency and does not address the AVB and MOST performance in terms of latency and jitter.

These considerations suggest the need for a MOST/Ethernet gateway, that is addressed in [39], with a concept for a migration of MOST to AVB. In [40] the performance of an in-car network with a double star topology under intensive streaming flows is assessed and the outcome of the study is the need for QoS mechanisms. The findings in [40] further motivate the use of AVB for in-vehicle network as it provides advanced QoS management. The same authors, in [41], addressed the QoS offered by three different network topologies, i.e. , a star, a daisy-chain and a tree-based one, in a mixed traffic scenario. The outcome of the work is that the star topology outperforms all the others in terms of end-to-end delay.

The work in [42] focuses on audio and video communications and compares several network topologies in terms of Quality of Service (QoS) and cost. In [43] a star-based topology for an in-car network using Gigabit Ethernet is evaluated under a workload consisting of control and video traffic. In [44] the performance of AVB and TTE are proven to be comparable in a network with a tree-based structure and a mix of control traffic and streaming traffic. Such a scenario significantly differs from the one in [38], both for the topologies investigated and for the type and amount of traffic exchanged on the network. In [38] star topologies instead of neither the tree-based structures are addressed and the control traffic is not considered. Moreover, the main focus is on ADAS and, for this reason, in the considered scenario the network load for ADAS traffic is significantly higher than in [44]. In [45] Ethernet is investigated as a common networking technology to be used not only in a single functional domain, but also as in-car backbone for inter-domain communications. The double star is, again, the topology that offers the best performance in terms of end-to-end delay and packet loss. The work [36] addresses the suitability of Ethernet for automotive communications and indicates AVB and TTE as possible candidates. In [25] encouraging simulation results obtained with the IEEE 802.1AS standard for in-car networking are provided. The work [46] highlights open issues in AVB worst-case latency analysis, pointing out some limitations of current theoretical formulations used for AVB latency estimation in [47]. Basically, starting from the findings in [48], the authors explain two effects that affect the latency estimation of AVB and that are not encompassed in the formulas provided by the AVB standard [47]. The first of these effects is the so-called "own-priority and higher-priority blocking", that occurs when several streams of same or different priority share the same port. In this case, bursts can accumulate over multiple hops, thus eventually interfering with other streams and increasing their latency. The second effect is called "shaper blocking" and refers to the large blocking times that a flow may experience in certain scenarios (i.e., in daisy-chains) due to traffic shaping. Such a blocking may get worse when combined with priority inversion and the priority blocking described above. The simulation results shown in [38] are compliant with these findings.

In the study in [49] Ethernet is proven to consume less power than MOST.

The work in [50] discusses the role of Ethernet in ADAS and describes a first prototypal implementation of a camera-based ADAS for automotive applications, with special focus on electric mobility. The integration of video-camera systems in ADAS with MOST technology is addressed in [28]. A prioritization method for bandwidth allocation in MOST network to support ADAS and multimedia traffic on the same network is proposed in [51].

CHAPTER

FOUR

ASSESSING IEEE AUDIO VIDEO BRIDGING FOR AUTOMOTIVE NETWORKS

In this chapter a preliminary assessment of IEEE Audio video Bridging in a realistic automotive scenario is described [37].

4.1 Simulation Scenario

In the scenario under study, shown in Figure 4.1, different traffic types are present:

- Advanced Driver Assistance System (ADAS);
- High-quality Multimedia Audio;
- HD-Video entertainment.

The ADASs here considered are based on a system composed by 6 IP-cameras, i.e. Front, Night Vision, Left, Right, Rear, and one

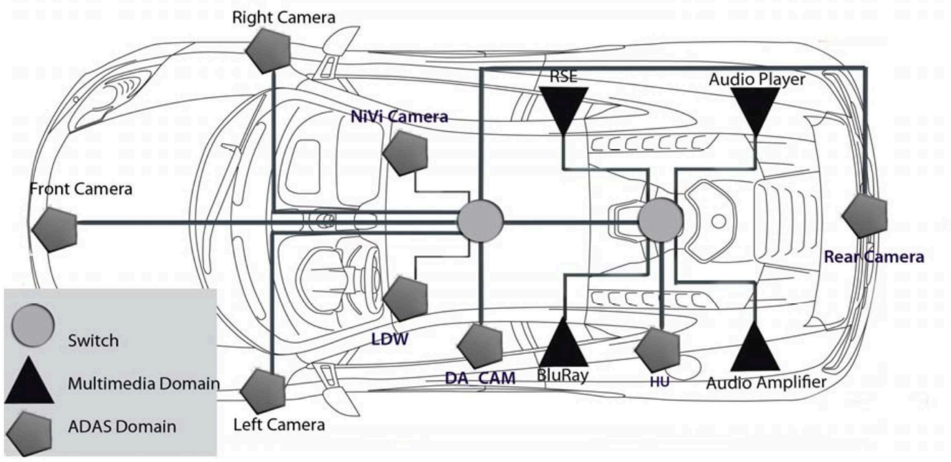

Figure 4.1: *The network topology*

Lane Departure Warning/Traffic Sign Recognition (LDW/TSR) camera. All cameras generate video streams (one for each camera) and send them to a specialized Driver assistance Electronic Control Unit (ECU), named DA-Cam. The DA-Cam processes the streams and produces both "views" (e.g. Top view, Side view) and navigation warnings. Both the views and navigation warnings are sent to a Head Unit (HU in the picture) that is equipped with a monitor, installed on the car's dashboard, on which the received views and warnings are displayed. In our scenario, the video frame rate generated by an IP-camera is 30 frames per second (fps), while the video resolution selected for displaying the stream on the Head Unit monitor is 640 x 480 pixel. We supposed a maximum video frame size of 27.3 KB, as in [52] and we adopted the small modification in the encoding algorithm described in [53] to reduce the variability in the generated

Type	Bandwidth [Mbps]	AVB SR-Class
Cameras	32.75	SR-Class A
LDW/TSR camera	13.10	SR-Class A
DA-Cam Video traffic		
Single flow (Case A)	32.75	SR-Class A
Aggregated flow (Case B)	32.75	SR-Class A
DA-Cam Warning traffic	0.016	SR-Class A
Bluray	40	SR-Class B
Audio	8	SR-Class B

Table 4.1: *Traffic Model and configured traffic/priority classes*

traffic and minimize the delay due to the acquisition and coding of the video frame. The car is also equipped with a multimedia in- car audio system allowing to listen to music at a very high quality and a HD-Video entertainment (e.g., a BluRay) system [41]. The BluRay video stream (encoded with MPEG4 High Definition standard) is directly sent to the rear seats monitor (Rear Seat Entertainment, RSE), while the audio stream is coded with AAC (Advanced Audio Coding) and sent to the multimedia in-car audio system. Alternatively, the audio stream produced by a multimedia audio player can be sent to the in-car audio system instead of the BluRay audio stream. Table 4.1 shows the requirements for the traffic flows in our scenario.

Our performance metrics are:

- **Latency**, defined as the one-way frame end-to-end delay, i.e., the time from the source sending a packet to the destination receiving it.

- **Jitter**, defined as the absolute value of the difference between two consecutive inter-arrival times. The *inter-arrival time* is the difference between the arrival times of two consecutive frames of the same stream. This jitter is calculated at the destination as in Eq. Equation 4.1

$$J_n = |(a_n - a_{n-1}) - (a_{n-1} - a_{n-2})| \tag{4.1}$$

where $n > 2$, The arrival time an of the n-th Ethernet frame, and so the latency and the jitter, are measured at the application level.

The considered topology is a double-star where ADAS and entertainment traffic are transmitted on the same physical infrastructure. The network is shown in Figure 4.1 and consists of two directly connected switches. The first switch connects all the six cameras with the DA-Cam, the second one connects all the entertainment units and the Head Unit. The flows that traverse both the switches originate from the DA-Cam and go into the Head Unit. These flows consist of either a single view, i.e. a single flow (corresponding to case A in section 4.3), or multiple views aggregated in a single flow (corresponding to case B in section 4.3), in addition to navigation warnings present in both cases. The entertainment traffic traversing the second switch is the multimedia video traffic sent by the BlueRay player to be displayed on the RSE and the multimedia audio flow that the relevant audio player streams to the Digital Audio Amplifier. Table 4.1 summarizes the traffic flows we simulated in our scenario. It can be seen that the network operates under a high workload. As shown in Table 4.1,in the considered scenario all the ADAS-related traffic flows are mapped onto the AVB SR-Class A, while the entertainment traffic, i.e., the BluRay stream and multimedia audio, are mapped onto AVB

SR-Class B. This is because we assumed that all the traffic related to ADAS is more important than the entertainment traffic. According to the IEEE 802.1Qat standard [5] only 75% of the total bandwidth can be reserved to Class A and Class B, to leave room for best-effort traffic. As the standard [5] specifies that the percentage of bandwidth assigned can be sized according to the traffic needs, and given that there is no best-effort traffic in our scenario, here we assigned 90% of the Switch1 bandwidth to Class A and 5% to Class B, and 50% of the Switch2 bandwidth to Class A and the other 50% to Class B.

4.2 The simulation model

The network performance was evaluated using the OMNeT++ simulation tool and the INET-framework. The simulaton time was 600s. OMNeT++ [54] is a modular, component-based C++ simulation tool, primarily used for building network simulators. Components (modules) of the simulation are programmed in C++, then assembled into larger components and models using a high-level language called NEtwork Description (NED) language. AVB simulator consists of two kind of modules:

- End-node (either talker or listener), each consisting of *application*, *queue* and *mac* submodule;
- Bridges, each consisting of *relay unit*, *queue* and *mac* submodule

The end-node submodules were designed from stretch, while the bridge modules were obtained modifying the mac layer in the 802.1 bridge modules, included in the INET-framework [55], adding the SRP

Type	**Case A**		**Case B**	
	Avg	Max	Avg	Max
Cameras	0.321	0.487	0.333	0.487
LDW/TSR camera	0.365	0.569	0.385	0.569
DA-Cam Video traffic	0.238	0.238	0.242	0.242
DA-Cam Warning traffic	0.098	0.170	0.105	0.180
Bluray	0.239	0.239	0.239	0.239
Audio	0.239	0.239	0.239	0.239

Table 4.2: *Latency results, expressed in ms, for Ethernet frames in case A and case B*

and the CBS functionalities. For each end-node is possible to define multiple flows to send and/or receive as well as the association between each queue in the outgoing port and a SR-Class. Configuration files are edited to change the traffic characteristic of each flow, including SR-Class, payload of each packet, period of the flow.

4.3 Results

In this section, latency and jitter results, measured for the ADAS video frames (Table 4.4 and Table 4.5) and for Ethernet data frames of all the traffic flows (Table 4.2 and Table 4.3), are discussed. As each video frame is split into 30 Ethernet frames for transmission, to measure the latency and jitter of the video frames we are interested in determining the latency of sequences of 30 Ethernet frames. We calculate the time between the instant when the first Ethernet frame of a sequence is sent and the instant when the last Ethernet frame of the sequence arrives at

Type	**Case A**		**Case B**	
	Avg	Max	Avg	Max
Cameras	1.45	54	6.02	81
LDW/TSR camera	555	881	555	960
DA-Cam Video traffic	0	0	0	0
DA-Cam Warning traffic	0.52	6.2	0.62	7.8
Bluray	0	0	0	0
Audio	0	0	0	0

Table 4.3: *Jitter results, expressed in μs, for Ethernet frames in case A and case B*

Type	**Case A**		**Case B**	
	Avg	Max	Avg	Max
Cameras	32.68	32.68	32.43	32.43
LDW/TSR camera	16.50	16.50	16.51	16.51
DA-Cam Video traffic	32.41	32.41	32.43	32.43

Table 4.4: *Latency results, expressed in ms, for ADAS video frames in case A and case B*

Type	**Case A**		**Case B**	
	Avg	Max	Avg	Max
Cameras	2.80	81	2.86	135
LDW/TSR camera	6.24	78	7.61	118
DA-Cam Video traffic	0	0	0	0

Table 4.5: *Jitter results, expressed in ms, for ADAS video frames in case A and case B*

Type	**Workload (Mbps)**		**Throughput (Mbps)**	
	Case A	Case B	Case A	Case B
Switch 1	57.64	87.78	57.64	87.78
Switch 2	58.27	87.49	58.27	87.49

Table 4.6: *Switch workload and throughput*

destination. As far as latency measure is concerned, we determine the time elapsed between the arrival time of the first Ethernet frame of a video frame and the arrival time of the first Ethernet frame of the next video frame and obtain an interarrival time. The difference between two interarrival times is the jitter. Here we preliminary assessed that we do not have packet loss. This behaviour was obtained by a suitable configuration of the switch buffers size and thanks to the support provided by the AVB protocol. Moreover, the sum of all incoming traffic on different ports that is directed to the same outgoing port does not exceed the available bandwidth.

Table 4.6, that compares workload and throughput obtained by simulation, confirms our findings. Table 4.4 and Table 4.5 show the results for video frames. The maximum transmission time for a full video frame is 32.68 ms, the maximum experienced jitter is $135\mu s$. The DA-Cam video frames have no jitter even though there is another parallel flow on the same link (i.e., navigation warnings) as both flows originate from the same unit and the navigation flow has a definitely lower bandwidth and service rate.

The null jitter is due to approximations in the simulator, that does not model the jitter of the hardware implementation. This is not a big issue, as here we want to show the effects of the protocol and not of the hardware. And, such a jitter is well below $1\mu s$. For the Ethernet frames of the cameras (see Table 4.2 and 4.3) in Case A (i.e. a single flow from the DA-Cam to the HU) the jitter is due to the fact that several video streams are sent from the cameras to the same receiver (DA-Cam), and so the Ethernet frames are delayed in the relevant Switch1 outgoing queue. In Case B (aggregated flow) we see an increase of jitter due to the higher workload, but such an increase is not significant. The simulation results show that the AVB protocol performs well under the scenarios provided.

In particular, the latency experienced by the navigation warnings messages is way below the reaction time of a human driver (usually not lower than 0.5s [56]) and therefore fully compliant with the ADAS application constraints. The latency of BluRay traffic meets the requirements stated in [40]. The traffic from the cameras used for direct services experiences a latency which is always less than 33ms, while for this class of traffic the limit can be set to 45ms [40]. Since the LDW/TSR camera has a frame rate twice as the previous ones, its maximum allowed latency must be set to $45/2 = 22.5$ms, but its latency is always less than 16.55ms (Table 4.4), so such a limit is met. The traffic originated by the DA-Cam ECU and addressed to the HU meets the same constraints as the Cameras traffic. Throughput and workload are almost the same in both the topologies and cases, thus confirming that there is no packet loss (see Table 4.6).

4.4 Conclusions

This preliminary performance assessment represented a first step of the investigation about IEEE AVB standard in automotive scenarios. According to the results here obtained, IEEE AVB proved to be a promising standard for being exploited in in-car networks. The work in [37] inspired comparative assessment between IEEE AVB and TTethernet that was realized in [38].

CHAPTER

FIVE

ASSESSMENTS OF IEEE AUDIO VIDEO BRIDGING AND TIME-TRIGGERED ETHERNET FOR IN-CAR COMMUNICATION

In this chapter a comparative assessment between IEEE Audio video Bridging and Time-Triggered Ethernet for in-car communication is described [38].

5.1 The considered scenario

In the scenario under study, shown in Table 5.1, different traffic types are considered:

- Advanced Driver Assistance Systems (ADASs).
- CD Audio.
- DVD Video entertainment.

- Cross-domain traffic.

The ADASs here considered are based on a system composed by 6 cameras that generate video streams (one for each camera) and send them to a specialized Driver Assistance Electronic Control Unit, named DA-Cam, that processes them. According to the services they provide, in the considered scenario the cameras are split in two service groups [52]:

- Indirect services, aimed to improve road safety, that support the driver with navigation warnings derived from the processed video streams. The camera used for indirect services, i.e. Lane Departure Warning/Traffic Sign Recognition (LDW/TSR) is positioned on the windshield close to the rear mirror. LDW is a mechanism devised to warn the driver when the car crosses a road lane marking or the edge of the road. TSR is a technology that enables a vehicle to recognize the traffic signs on the road.

- Direct services, that support the driver with visual information in the form of views. In our scenario, there are five direct service cameras: Front, NightVision, Left, Right, Rear (see Figure 5.1).

The flows processed by the DA-Cam are sent to the a Head Unit that is equipped with a monitor, installed on the cars dashboard, on which the received streams are displayed. The output flows produced by the DA-Cam are new flows, either augmented with additional graphics to assist the driver or resulting from processing multiple camera flows to produce single views (e.g., Top view, Side view, etc.). A flow streamed to the Head Unit therefore consists of either a single view (as multiple views cannot be displayed at the same time on the

Head Unit monitor because this is usually too small), or multiple views aggregated in a single flow in such a way that the Head Unit can select and extract the needed view and display it on the monitor or, alternatively, send it to the RSE. As said before, the DA-Cam also produces navigation warnings that are displayed at the Head Unit monitor. In order to make the simulation scenario realistic, had to be decided how to model both the traffic generated by cameras and the streaming of the flows displayed on the Head Unit monitor, taking into account both current practices and state of the art. Some previous works [42] considered MPEG-2 Transport Stream compression (MPEG-2/TS) for both IP cameras and multimedia video. This can be considered as a conservative approach, as MPEG-2 traffic has been thoroughly analysed and modelled and hardware compression is available at reasonable costs. On the other hand, MPEG-2 has been recently replaced by more effective codecs based on the MPEG-4 standard for a number of reasons. In most cases recent IP cameras for video surveillance adopt MPEG-4 SP (Simple Profile) for video coding as a trade-off between encoding/decoding complexity and bitrate/quality combination. Compared to other standards for video compression, MPEG-4 SP requires a smaller bandwidth (a few Mbps in the scenario considered in [38]) while offering higher quality. Therefore, in [38] the MPEG-4 SP is the final choice. In the considered simulation scenario, the video frame rate generated by a camera is 30 frames per second (fps), while the video resolution selected for displaying the stream on the Head Unit monitor is 640 x 480 pixel. To model an increasing video traffic workload, two DA-Cam Aggregated flows is used. The first one, indicated in Table 5.1 as DA-Cam Aggregated (4-flows), starts at t=0s and stops at t=400s. The second one, named DA-Cam Aggregated 5-

flows, models the workload increase corresponding to turning on in our scenario the fifth direct camera (i.e. the Night Vision one). It starts at t=400s and continues until the end of the simulation at t=600s. This way the behaviour of the network when the video traffic workload increases over time could be assessed. The car is also equipped with a CD Audio and a DVD Video entertainment system. The DVD video stream, encoded with the MPEG-2 Program Stream standard (MPEG-2/PS), is directly sent to the rear seats monitor, while the audio stream is encoded with AC3 (Dolby Digital) and sent to the in-car digital audio amplifier. Alternatively, the audio stream produced by the audio-CD player can be sent to the in-car audio digital amplifier instead of the DVD audio stream. The features of the CD Audio and DVD video flows are shown in Table 5.1. The considered scenario also envisages cross-domain traffic, consisting of periodic data coming from the car control network which is gathered from a gateway device and injected in our network, directed to a Cross-Domain Processing Unit which extracts relevant data useful to the navigation and driver warning functions. When a given condition is detected, which requires a driver warning, the Cross-Domain Processing Unit turns on the corresponding dashboard indicator. Here an increasing cross domain traffic is modelled using two flows. The first one, indicated as Cross-domain 1st flow in Table 5.1, starts at t=0s and stops at t=300s. The second one, named Cross-domain 2nd flow, is twice the first one. It starts at t=300s and continues until the end of the simulation at t=600s. This way the network behaviour could be assessed when the cross domain workload doubles over time. Table 5.1, in addition to the characteristics of our traffic flows, also shows their mapping on the traffic classes provided by AVB and TTE, respectively. The first column gives, for

Type	**Bandwidth [Mbps]**	**Activation interval [s]**	**AVB SR-Class**	**TTE Priority**
Cameras	[5,15]	4x[0,600] + 1x[400,600]	A	RC (Prio 3)
LDW/TSR camera	[2,6]	[200,600]	A	RC (Prio 3)
DA-Cam single flow	[1,3]	[0,600]	A	RC (Prio 3)
DA-Cam				
Aggregated (4-flows)	[4,12]	[0,400]	A	RC (Prio 3)
Aggregated (5-flows)	[5,15]	[400,600]	A	RC (Prio 3)
DA-Cam warning	0.016	[200,600]	A	RC (Prio 2)
DVD player	10.08	[500,600]	B	RC (Prio 4)
Cd Audio Player	1.41	[0,600]	B	RC (Prio 4)
Cross Domain (1st flow)	0.0736	[0,300]	A	RC (Prio 1)
Cross Domain (2nd flow)	0.147	[300,600]	A	RC (Prio 1)

Table 5.1: *Characteristic of traffic model and configured traffic/priority classes*

each traffic type, the corresponding workload. For instance, [5, 15] for cameras means that the overall amount of video traffic generated by all the cameras varies between 5 and 15 Mbps. For camera flows, it is known that the workload is highly dependent on the specific video, i.e. on the scene being captured. As in [38] the main focus on evaluating the performance of the two addressed networks under high and very varying workload, the workload ranges typical of the adopted encoding is took as reference . Hence, the parameters of statistical distributions able to generate such workloads are derived. Table 5.1 also shows the activation interval of each traffic type. Every device in the network has in fact an activation interval that indicates when the device is switched-on and so the relevant traffic starts.

5.2 Simulation setup

The network performance was evaluated using the OMNeT++ simulation tool and the INET-Framework as described in 4.2. A protocol stack where the Application level is directly on top of Ethernet is simulated, for both AVB and TTE. The protocol overhead is therefore 18 bytes for both TTE and AVB. For all the simulations, the duration is set to 600s. For each topology, two cases are investigated.

- Case A: the DA-CAM sends to the Head Unit a single flow (with a workload varying in the range [1,3] Mbps);
- Case B: DA-CAM sends to the Head Unit a flow resulting from the aggregation of flows from the Cameras that provide direct services (with a workload varying in the range [4,12] Mbps for the first 400s, and in the range [5,15] Mbps for the last 200 s of simulation).

Given the high variability of the video payload, traffic bursts are present in both scenarios. All the other traffic flows are the same in both cases and the relevant values are shown in Table 5.1.

AVB Setup In AVB simulation, all the traffic is mapped on the Stream Reservation (SR-AVB) classes. A priority value is associated with each SR-AVB class. For SR-AVB Class A the priority value is “3”, for SR-AVB Class B the priority value is “2”. As shown in Table 5.1, all the video flows from the cameras to the DA-Cam, the video flows and the navigation warnings from the DA-Cam to the Head Unit and cross-domain traffic are mapped onto Class A, while the entertainment traffic (i.e., the DVD stream and audio CD) are mapped onto SR- AVB Class B. The rationale behind this mapping is that AVB supports only two classes of time-sensitive traffic (i.e., Class A and Class B), and cameras video, DA-Cam video, navigation warnings and cross domain traffic are relevant to the driver safety. Those flows could have been split onto Class A and B but, doing so, entertainment traffic would be mapped either onto Class B, to the detriment of the safety-related flows in that class, that would have experienced interference from very bandwidth-greedy entertainment flows at the same priority level, or on best-effort traffic, without any guarantee on quality of service. With the chosen mapping the mix between safety-critical and non-safety critical flows is avoided, while some QoS guarantees also to the entertainment flows are provided.

This choice therefore reflects the different criticality of the flows from the user perspective (being safety-related traffic more critical than entertainment one). Bandwidth is reserved using the Stream Reservation Protocol (SRP) and the related signalling protocol, the Multiple Stream Registration Protocol (MSRP) for stream reservations across the network. This protocol provides end stations with the ability to reserve network resources that guarantee the transmission and reception of data streams across a network with the requested QoS. According to the IEEE 802.1Qat Standard [5] only 75% of the total bandwidth can be reserved to Class A and Class B, to leave room for best-effort traffic. Although in the considered scenario there is no best-effort traffic, the standard specifications are followed. The percentage for the reservable bandwidth is 40% for AVB SR Class A and 35% for AVB SR Class B. Each stream is identified from unique *StreamID*. The forwarding process provides one or more queues for a given switch port, each corresponding to a distinct traffic class. Each traffic class is assigned a distinct priority level, so there is one queue for each priority level.

TTE Setup TTE supports three traffic classes. In the considered scenario the time-triggered traffic class is used for the Cross-domain traffic flows and the rate-constrained traffic class is used for the others. Four priority levels, with different queues, are used. Cross-domain traffic has the highest priority, i.e. prio 1, as it has a small payload and introduces a light workload, so we preferred to provide it with a preferential handling. For the same reason, the second priority level, i.e., level 2, is given to navigation warnings, while cameras and DA-Cam video have the third priority level, i.e. level 3. Finally, CD audio and DVD video streams have the fourth priority level, i.e. level 4. Strict Priority scheduling is used between the switch queues, FIFO within the same queue. The TT traffic is sent at regular time intervals. To support this traffic it is necessary to configure a complete off-line schedule. In the cycle time, i.e., the period after which the schedule is repeated, we define the slots in which every network device can send TT messages. In the offline planned slots no other traffic can be sent. TT traffic is actually generated at a well-defined time at the sending device and it must be received at a known time by the receiving device. A different receiving time of the frame could cause incorrect scheduling. For simulation purposes, an exact receiving time is not defined, but a very small temporal interval, called a receiving window, within which the reception of TT traffic is expected. This window is defined for every cycle time (which has a value of 5 ms) and is expressed in ticks (1 tick = 80 ns). The RC traffic is based on the concept of Virtual Link (VL), i.e., a logical unidirectional connection from one source end system to one or more destination end systems. The use of VL allows full-duplex communication channels on which multiple streams, each one identified by a CT-ID (Critical Traffic-Identifier), can be sent. Each VL is associated with a Bandwidth Allocation Gap (BAG), that is the minimum delay between two consecutive frames on the same VL. The application sending the message has to respect the constraint of the relevant configured BAG, otherwise the Ethernet frame will be considered not valid and will be dropped by the switch [3]. The model used for the TTE simulation was evaluated e validated using analytical methods and real-world measurements using TTE hardware in [57]. The switches are characterized by a switch processing time of 8μs. The switch model used allows us to consider the propagation delay, that in our case has been fixed equal to 5 ns per meter and varies according to the length of the connections between the devices in the interval [0.3, 3.2] m.

5.3 Topologies

In [38] two different topologies are investigated. In the first one, called a single-star topology (Figure5.1), two separate switches are used, while in the second one, called a double- star topology (Figure5.2), there are two interconnected switches.

Single-star Topology In the single-star topology (Figure5.1), the first switch connects the ADAS cameras to the Driver Assistance Cam ECU (DA-Cam) unit. This choice, that differs from the approach in [41], where point-to-point connections are used between cameras and ADAS, is motivated by the need to reduce wiring complexity,

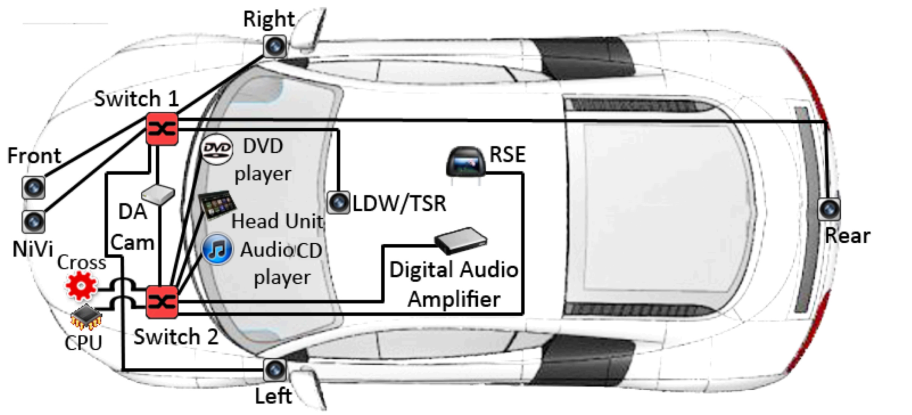

Figure 5.1: *The single-star network topology*

weight and costs (as explained in Chapter 1) replacing the current point-to-point Low-Voltage Differential Signalling (LVDS) wires with a switched network. The second switch connects the Head Unit with the DA-Cam, the Cross domain device with the Cross-domain Processing Unit (CPU), the DVD player with the Rear Seats Entertainment (RSE) system, and an audio player (CD-Audio) with the relevant digital audio amplifier. The reason for two switches is to separate the flows originating from the cameras and directed to the DA-Cam from the rest of traffic, to avoid that entertainment and cross-domain traffic on the same switch could affect the performance of ADAS traffic. In the single-star topology, the DA-Cam is therefore a specialized ECU, equipped with two ports to be connected to two different switches, respectively. It produces new traffic flows (resulting from processing the ones received from the cameras and traversing the first switch), that are sent through the second switch to be displayed on the Head-Unit

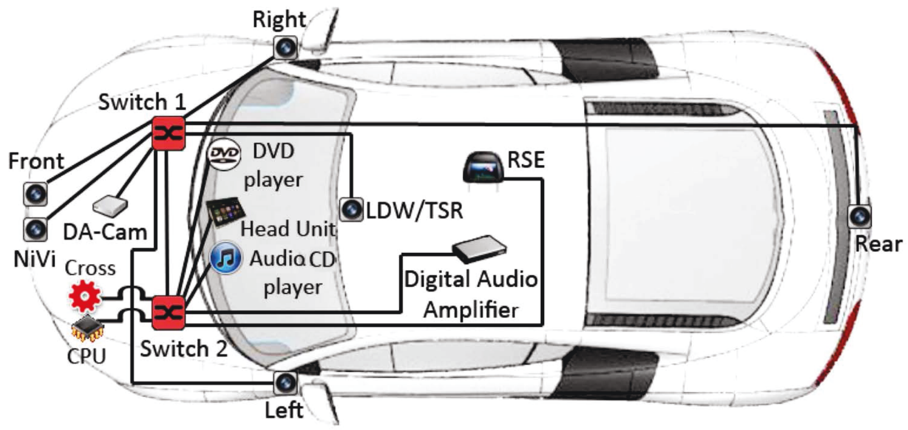

Figure 5.2: *The double-star network topology*

monitor. The second switch, in addition to the flows sent by the DA-Cam to the Head Unit, also handles the multimedia video traffic sent by the DVD player to be displayed at the Rear Seat Entertainment (RSE) system, the multimedia audio flow that the Audio-CD player streams to the Digital Audio amplifier and the cross-domain traffic that is processed by the Cross- domain Processing Unit (CPU).

Double-star Topology The second topology under study in [38], called a double-star topology, is shown in Figure5.2. Differently from the star topology, here the two switches are directly connected, so the DA-Cam is a one-port device. The flows that traverse both the switches originate from the DA- Cam and go to the Head Unit. Like in the single-star topology, these flows consist of either single views, or multiple views aggregated in a single flow, in addition to navigation warnings present in both cases. The cross-domain and entertainment flows traversing the second switch in the double star topology are the same that traverse the second switch in the single star topology. The double star topology is investigated to assess the network performance when the traffic of the three domains, i.e. cross-domain, ADAS and entertainment, is transmitted on the same physical infrastructure, and the network operates under a high workload. In this case, in fact, the first switch is also traversed by the ADAS traffic generated by DA-Cam, so its workload is higher than in the single-star topology.

5.4 Results

The performance metrics considered for the comparative evaluations in [38] are the same described in section 4.1.

Latency Assessment - Single Star topology - Case A Table 5.2 shows the results for Ethernet AVB and TTE for Ethernet frames. Mean latency values for both protocols are in the same order of magnitude, although AVB has slightly higher values than TTE, with the exception of cameras and DVD flows. As far as maximum latency is concerned, TTE has slightly higher values for cameras, LDW/TSR and navigation flows, but lower values for the other traffic flows. For both protocols, a latency increase is observed, during simulation, whenever devices are dynamically activated. For example, when the LDW/TSR and NiVi cameras are added (at t=200s and t=400s, respectively), the latency increases about 80 μs for every device. LRD is high in cameras and LDW/TSR flows for both protocols, as several video streams at the same priority level are sent from the cameras to the same receiver (DA-Cam), and so the Ethernet frames are delayed in the relevant switch outgoing queue. The slight different LRD between the two protocols is due to the different frame scheduling in TTE (FIFO) and AVB (Credit-Based Shaping). The LRD values equal to 0 for Audio CD and DVD in TTE are because both these flows do not suffer from interference from other flows in the relevant queues. This behaviour is observed with TTE in all topologies and in all cases here addressed. For navigation flow in AVB, the LRD depends on the presence in the same switch output queue of large frames of other Class A traffic, i.e. DA-Cam flows. The maximum LRD for the DA-Cam flow shows that this flow is sometimes affected by the navigation warning traffic, that determines rare peaks starting from 200 s onwards. Average LRD values are instead very close to TTE results. The results prove that TTE benefits from using different priorities for navigation warnings and DA-Cam, while in AVB these traffic flows have the same priority level, as both flows are mapped onto Class A, so they mutually interfere. As explained in the AVB setup section, this is because AVB supports only two classes of time-sensitive traffic, Class A and Class B. Cross-domain traffic also obtains the best results in terms of latency and maximum latency deviation with TTE, thanks to the mapping onto the highest priority class (i.e., the time-triggered one).

Latency Assessment - Single Star topology - Case B In this scenario, instead of a single flow, DA-Cam streams to the Head Unit an aggregated video stream. The simulation results are shown in Table 5.3, where *DA-Cam Aggregated (4-flows)* and *DA-Cam Aggregated (4-flows)* are used to indicate the aggregated flow before and after the activation time of the night vision camera at time 400s (as indicated in Table 5.1), respectively. With TTE, there is almost no difference

Type	Latency [μs]				LDR [μs]			
	AVB		TTE		AVB		TTE	
	Avg	Max	Avg	Max	Avg	Max	Avg	Max
Cameras	109	441	169	469	411.2	852	148.31	688.66
LDW/TSR camera	209	443	121	459	414.1	852	86.18	733.610.569
DA-Cam	122	245	111	166	67	481	56.52	191.52
Navigation Warnings	62	113	33	163	32	189	5.92	173.12
DVD	399	1400	794	1350	526	2100	0	0
Cd-Audio	412	489	238	238	637	971	0	0
Cross-domain	160	170	60	61	0.2	1.5	0.22	1.12
Cross-domain	160	175	60	61	0.2	1.5	0.22	1.04

Table 5.2: *Latency and LDR results, in Single Star - Case A*

in the latency results for DA-Cam Aggregated (4-flows) as compared to the single DA-Cam flow in case A (see Table 5.2). For DA-Cam Aggregated (5-flows), in TTE we see a slight latency increase, while the LRD is still very close to the one obtained for the single flow in Case A in Table 5.2). This means that TTE offers a more stable behaviour than AVB for DA-Cam flows. AVB results, in fact, show an increase in the mean latency values and a significant increase in the LRD for both DA-Cam Aggregated (4-flows) and DA-Cam Aggregated (5-flows). Similar results for the two protocols are obtained for navigation warnings latency and LRD between Case A and Case B, with AVB providing slightly higher values. Even in this Case, the cross- domain flow obtains the best results with TTE, because it is managed as time-triggered traffic. LRD values are generally better in TTE, as mentioned before. In AVB, the maximum value of LRD for the traffic sent by DA-Cam to Head Unit in Table 5.3) is less than in the previous case, and this occurs at the expense of the maximum LRD of navigation warnings, that is larger than the one in Case A. These results confirm the mutual interference between these two traf-

Type	Latency [μs]				LDR [μs]			
	AVB		TTE		AVB		TTE	
	Avg	Max	Avg	Max	Avg	Max	Avg	Max
Cameras	100	441	169	469	411.2	852	148.31	688.66
LDW/TSR camera	209	443	121	459	414.1	852	86.18	733.61
DA-Cam Aggregated (4-flows)	203	225	111	161	308	340	56.64	190.88
DA-Cam Aggregated (5-flows)	204	228	171	389	330	372	56.67	192.64
Navigation Warnings	89	154	42	168	67	200	20.20	178.16
DVD	399	1400	794	1350	526	2100	0	0
Cd-Audio	412	489	238	238	644	971	0	0
Cross-domain 1st flow	160	170	60	61	0.2	1.5	0.22	1.04
Cross-domain 2nd flow	160	175	60	61	0.2	1.5	0.22	0.96

Table 5.3: *Latency and LDR results, in Single Star - Case B*

fic flows in our AVB settings, and show that a larger amount of traffic with big-size frames, such as DA-Cam one, affects the performance of competing traffic in the same class with smaller-size frames, such as navigation warnings.

Latency Assessment - Double Star topology - Case A For the double star topology in case A the latency increases, as compared to the single star, for both the DA-Cam and navigation warnings flows with both AVB and TTE, albeit more significantly with AVB (see Table 5.4). This is due to the increased amount of traffic that traverses switch 1 in this topology. As cross domain, CD-Audio and DVD video flows are not affected by the double star topology, their results remain the same as in the single star, Case A. Results show that the frames that are forwarded through two switches now experience a higher latency. Also in this scenario the activation of navigation warning traffic affects the AVB latency of DA-Cam flow, from t=200 s onwards, by introducing rare peaks up to t=912 μs. This is because these two flows have the same priority and content for transmission in the same

Type	Latency [μs]				LDR [μs]			
	AVB		TTE		AVB		TTE	
	Avg	Max	Avg	Max	Avg	Max	Avg	Max
Cameras	131	453	169	471	431.7	852	148.34	717.83
LDW/TSR camera	209	457	121	471	431.7	852	86.48	757.27
DA-Cam	189	912	170	250	102.3	1400	84.87	286.56
Navigation Warnings	77	118	57	250	47	119	11.72	241.04
DVD	399	1400	794	1350	526.1	2100	0	0
Cd-Audio	412	489	238	238	637	971	0	0
Cross-domain 1st flow	160	170	60	61	0.2	1.5	0.22	1.04
Cross-domain 2nd flow	160	175	60	61	0.2	1.5	0.22	1.04

Table 5.4: *Latency and LDR results, in Double Star - Case A*

queue, so, under certain circumstances, own- priority blocking and traffic shaping may determine very high latency for some frame. The LRD values increase in TTE for cameras, LDW/TSR, DA-Cam, navigation warnings as this traffic directed to Head Unit must cross both the switches, and Switch1 is traversed by more traffic compared to the single star case.

Latency Assessment - Double Star topology - Case B In the double star topology, between Case A and Case B, with TTE there is no significant latency increase for DA-Cam Aggregated (4-flows) flow as compared to DA-Cam single flow, while for Cam Aggregated (5-flows) there is a latency increase, although the LRD remains stable. Conversely, AVB latency is larger for both Cam Aggregated (4-flows) and Cam Aggregated (4-flows) (see Table 5.5). No latency peaks as in Case A are found with AVB for DA-Cam aggregated flows in this configuration, thus showing that in this case the own priority effect affected more navigation traffic than DA-Cam one. In fact, here with AVB navigation traffic obtained higher latency values than in Case A, due to the increased amount of interfering DA-Cam traffic at the same priority. As far as DA-Cam LRD is concerned, AVB suffers from significant increase for the mean value, but not for the maximum. Conversely, TTE for DA-Cam obtained results quite close to the ones found in Case A, thus showing a more stable behaviour.

Type	Latency [μs]				LDR [μs]			
	AVB		TTE		AVB		TTE	
	Avg	Max	Avg	Max	Avg	Max	Avg	Max
Cameras	130	442	169	469	411.2	852	148.47	709.60
LDW/TSR camera	203	457	121	459	417.1	852	86.23	742.88
DA-Cam Aggregated (4-flows)	223	265	170	246	401	434	84.89	287.76
DA-Cam Aggregated (5-flows)	237	279	231	537	410	482	84.97	288.96
Navigation Warnings	129	339	70	252	77	224	33.60	242.96
DVD	399	1400	794	1350	526.1	2100	0	0
Cd-Audio	412	489	238	238	637	971	0	0
Cross-domain 1st flow	160	170	60	61	0.2	1.5	0.22	1.04
Cross-domain 2nd flow	160	175	60	61	0.2	1.5	0.22	0.96

Table 5.5: *Latency and LDR results, in Double Star - Case B*

5.5 Discussion and conclusions

The outcome of the comparative assessment between AVB and TTE presented in [38] is that the best results for both protocols are obtained by the single star topology, but both topologies are suitable to fulfil the requirements of the traffic here considered, that is, 33 ms for all the video flows from cameras and DA-Cam and 100 ms for audio and multimedia video flows [42][40]. The results for cross-domain traffic are also adequate.

CHAPTER

SIX

COMPARING MEDIA ORIENTED SYSTEMS TRANSPORT AND THE IEEE AUDIO VIDEO BRIDGING FOR AUTOMOTIVE COMMUNICATIONS

In this chapter a comparative assessment between IEEE Audio Video Bridging and Media Oriented Systems Transport, for automotive communications, is described.

6.1 Application in the assessed scenario

- Camera-based Advanced Driver Assistance Systems (ADAS), i.e., a system composed of 5 IP-cameras that generate video streams (one for each camera) and a central Electronic Control Unit (ECU), named DA-Cam, which analyses and processes the streams and send them to a Head Unit.

- High-quality multimedia audio, i.e. a multimedia in-car audio system allowing to listen to music at a very high quality. This system is composed of a CD audio player and an audio amplifier.

- HD-Video entertainment, i.e., a DVD system composed of a DVD player and two monitors placed on the back of the front seats, called Rear Seats Entertainment monitors.

- Control system, based on information collected in other domains and reported to the driver, i.e., the car engine temperature, the fuel consumption level, etc. All the services are defined below.

The ADAS services are provided through real-time video processing and analysis system. In our scenario, which is shown in Figures 6.1-6.2, there are five cameras, i.e., Front/ Night Vision, Left, Right, Rear, and Traffic Sign Recognition/Lane Departure Warning (TSR/LDW) continuously acquiring the scenery surrounding the vehicle. The first four cameras are used for direct services, i.e., the services that support the driver through views displayed on the monitor of a so-called Head Unit installed on the car dashboard. To this end, the images collected by the cameras are sent to the DA-Cam which is in charge of producing new output video flows that are streamed to the Head Unit. These flows are either augmented with additional graphics to assist the driver, or resulting from processing multiple camera flows to produce single views (e.g., Top view, Side view, etc.). The Front Camera is placed close to the rear-view mirror that is inside the car and is used to produce the so-called bird's eye view. The Left and the Right cameras are placed near the external rear-view mirrors and the

Rear camera is placed on the back of the car, and is especially useful for the parking assistant system. The NiVi Camera is an infra-red camera used to capture video in low light conditions. The LDW/TSR camera is placed near the front camera and is used for Lane Departure Warning/Traffic Sign Recognition services. The Lane Departure Warning (LDW) is a driver assistance system that is based on a video sensor. A camera integrated near the rear-view mirror captures the 50 meters of road ahead of the car in daylight or within the headlight beam at night. The Traffic Sign Recognition (TSR) is a camera-based driver assistance system that helps the driver to maintain a legal speed and obey local traffic instructions, or urban restrictions, by recognizing various traffic signs. Both these services, LDW and TSR, are called indirect services, and support the driver with navigation warnings derived from video streams processed and displayed on the Head Unit monitor to assist the driver to improve road safety. The DA-Cam processes the streams and produces both "views" and navigation warnings that are sent to a Head Unit. The Head Unit is equipped with a monitor, installed on the car's dashboard, which displays the received streams (views) and warnings. In the adopted simulation scenario, each IP-camera generates a video stream with a frame rate of 30fps and frame size 640 x 480 pixel with a 24 bit colour depth. This results in a raw data flow of more than 221Mbit/s for each camera, which is unpractical without compression. While in the past the most widely adopted video compression standard for ADAS applications was MJPEG, due to its inherent low latency and decoding simplicity, recently most IP-cameras use the H.264/MPEG-4 Part 10 AVC (Advanced Video Coding) standard. AVC requires a considerably smaller bandwidth than MJPEG, while maintaining a comparable (or even higher) video quality, however, its generated packets may have largely variable size and a mechanism to reduce such variability is needed when the predictability of the traffic is important. The approach proposed in [27], and adopted here, significantly reduces the variability of the generated traffic and minimizes the acquisition and decoding delay, at the price of a small change in the encoding algorithm. In accordance with [52], we assume that for the video stream, the frame size is equal to 27.3KB. The high-quality multimedia audio consists of a player that provides an audio stream that is coded with AAC (Advanced Audio Coding) and sent to the multimedia in-car audio amplifier system. The car is also equipped with and a HD-Video DVD entertainment system [52]. The DVD video stream (encoded with MPEG4 High Definition standard) is directly sent to the rear seats entertainment monitors (RSE). Our scenario also includes a service based on the transmission of cross-domain traffic. The cross-domain traffic consists of periodic data gathered from a gateway device called a Cross-domain Unit (CU) and addressed to a Cross-domain Processing Unit (CPU). Such a unit processes data, extracting information useful to the navigation and driver warning functions. The dashboard is provided of indicators that are turned on when the CPU detects a special condition that requires a driver warning.

In this work, two topologies are considered:

- The switched topology for AVB
- The ring topology for MOST150

In the AVB switched scenarios, shown in Figure 6.1, the two switches, namely SwitchA and SwitchB, are directly connected. The

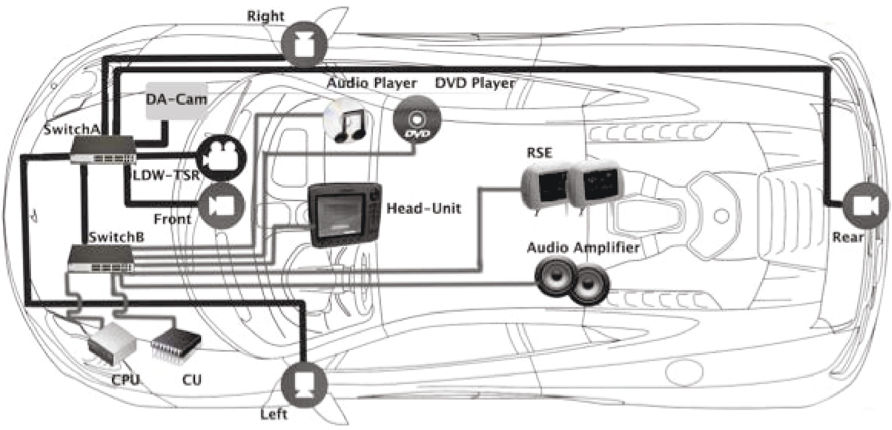

Figure 6.1: *The AVB switched network topology*

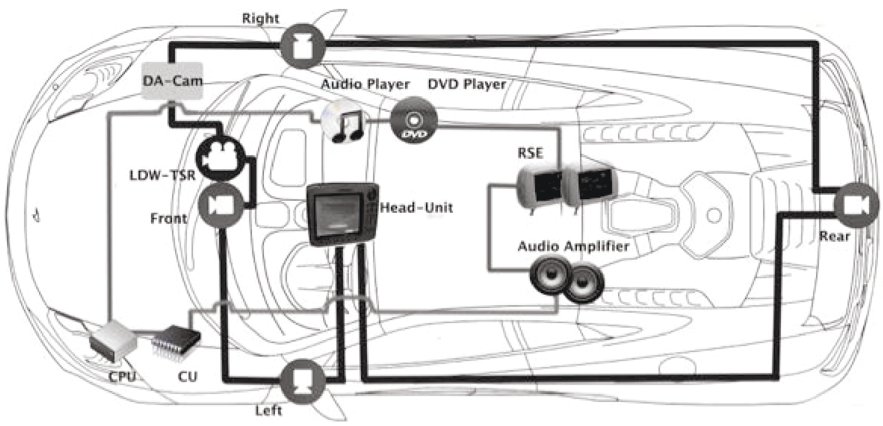

Figure 6.2: *The MOST ring network topology*

flow that originates in DA-Cam and goes to the Head Unit consists of multiple views aggregated in a single flow or in navigation warnings. The cross-domain and entertainment traffic only traverses the SwitchB. In Figure 6.1, the connections related to the ADAS switch are shown as a black line, while the multimedia ones are represented by a grey line. Unlike the approach in [41] where point-to-point connections are used between cameras and DA-Cam, the adoption of a switched or ring topology here is motivated by the need to reduce wiring complexity, weight and costs (as explained in chapter 3) replacing the current point-to-point Low-Voltage Differential Signalling (LVDS) wires with a switched network.

Here we recall that the MOST protocol specifies two topology options, i.e. ring and star. In the ring topology messages must traverse all the nodes between the sender and the receiver to reach the destination and this may entail increased latencies. In the star topology, instead, a logical ring is foreseen. The master sends the MOST frame to the first slave of the ring which, once received the frame, writes in the frame the byte relevant to its own traffic flow and sends it back to the master, which sends the frame to the second node, and so on. In this work, in order to reduce the cable length and latency, the ring topology was preferred to the star one.

As shown in Figure 6.2, there are two separate rings. One ring is dedicated to ADAS communications and it is shown as a black thick line, while the other one is dedicated to Multimedia and cross-domain traffic and is represented as a grey line. The choice to use two separate networks was inspired by the work [16] and is motivated by the need of reducing latency for ADAS traffic which, otherwise, could experience larger delay due to the high number of network nodes to cross.

Type	Bandwidth [Mbps]	App. Payload [Byte]	Service rate [ms]	Activat. Time [s]	AVB SR-Class
Cameras (3+1)	[4,12]	uniform(206, 620)	1.65	3 at 0 + 1 at 400	A
LDW/TSR camera	[2,6]	uniform(206, 620)	0.825	200	A
DA-Cam single flow	[1,3]	uniform(206, 620)	1.65	0	A
DA-Cam Aggregated (3+1)	[4,12]	uniform(155, 620)	0.413	3 at 0 + 1 at 400	A
DA-Cam warning	0.037	46	10	200	A
DVD player	11	1050	0.76	500	B
Cd Audio Player	1.41	1050	5.96	0	B
Cross Domain (1st flow)	0.037	46	10	0	A
Cross Domain (2nd flow)	0.074	46	5	300	A

Table 6.1: *Traffic flow characterization and traffic classes mapping for the AVB simulation*

6.2 Simulation Setup

Table 6.1 summarizes the traffic flows for our AVB simulations and shows how the flows were mapped onto the traffic classes provided by the protocol. Every device in the network has an activation time that indicates when the relevant traffic starts to be generated. Such a choice has been made to evaluate the protocol with a varying and increasing workload.

Since MOST has no traffic class distinctions, all the flows in the MOST scenario are mapped on to the same traffic class, as shown in Table 6.2, which describes the traffic flows in the MOST scenario. By comparing Table 6.1 and Table 6.2, is evident that the bandwidth demand is the same in both tables for the same flow, while the application payload and the service rate values are different. This is because, according to the MOST protocol, each node can fill up a part of each MOST frame and, for this reason, the service rate for each flow can be the same as the MOST frame (20.8 μs for synchronous traffic) or

Type	Bandwidth [Mbps]	App. Payload [Byte]	Service rate [ms]	Activat. Time [s]
Cameras (3+1)	[4,12]	uniform(5, 15)	41.6	3 at 0 + 1 at 400
LDW/TSR camera	[2,6]	uniform(5, 15)	20.8	200
DA-Cam single flow	[1,3]	uniform(1, 3)	41.6	0
DA-Cam Aggregated (3+1)	[4,12]	uniform(20, 60))	41.6	3 at 0 + 1 at 400
DA-Cam warning	0.037	2	457.6	200
DVD player	11	28	20.8	500
Cd Audio Player	1.41	4	20.8	0
Cross-domain (1st flow)	0.037	2	457.6	0
Cross-domain (2nd flow)	0.074	2	228.8	300

Table 6.2: *Traffic flow characterization and traffic classes mapping for the MOST simulation*

a multiple of this service rate (for isochronous traffic).

We assessed the performance of AVB in the switched topology and MOST adopting the ring topology.

For each topology, two cases were investigated as far as ADAS traffic is concerned.

- Case A: the DA-Cam sends to the Head Unit a single flow;
- Case B: the DA-Cam sends to the Head Unit a flow resulting from the aggregation of flows from the cameras that provide direct services.

In both cases the multimedia and cross-domain traffic flows are the same. The network performance was evaluated using the OMNeT++ simulation tool and the INET-framework. The AVB simulator used here is the same described in section 4.2, while the MOST was specifically designed and developed for this work. Both the simulators are validated as shown in section 6.4 For all the simulations, the duration was set to 600s and six simulations for the same scenario were realized to evaluate the results varying by the seed that determines the uniform distribution used to obtain the frame length. For the simulation scenario here proposed, as our purpose is to assess the network behaviour when the cross-domain workload doubles over time, we modelled an increasing cross-domain traffic using two flows. The first one, indicated as *Cross-domain (1st flow)* in Table 6.1 and Table 6.2 , starts at t=0s and stops at t=300s. The second one, named *Cross-domain (2nd flow)*, is twice the *Cross-domain (1st flow)*. It starts at t=300s and continues until the end of the simulation, i.e., until t=600s.

The considered performance metrics are:

- Latency: the one-way frame end-to-end delay, defined as the time interval between the instant at which the packet is sent and the instant at which it is fully received at the destination. Latency is measured at the MAC level.

- Jitter, defined as in equation Equation 4.1

where n¿2. The arrival time an of the Ethernet frames, as well as the latency and the jitter, are measured at the MAC level.

AVB and MOST setup Considering the heterogeneity of the traffic flows, in the AVB simulation we chose to exploit the two traffic classes provided by the standard i.e., the SR-AVB Class A (highest priority) and the SR-AVB Class B (the second highest priority). We assumed that the ADAS traffic is more critical than the entertainment traffic and, for this reason, as shown in Table 1, all the ADAS-related traffic are mapped onto the SR-AVB Class A. Conversely, the entertainment traffic, i.e., the DVD stream and multimedia audio are instead mapped onto SR-AVB Class B. According to [4] only 75% of the total bandwidth can be reserved to Class A and Class B, in order to leave room for best-effort traffic. For Class A and Class B traffic, the Credit Based Shaper (CBS), proposed in [4], is used to fairly share the link capacity and avoiding bursts of data. The switching processing time is supposed equal to 8 μs. Table 6.2 shows the traffic flows for the MOST scenario. According to [14], there is no traffic class distinction, so in Table 6.2 there is no column dedicated to the traffic class mapping. All the flows are considered Source Streaming Data, as they are all periodic data flows. The LDW/TSR camera, the DVD Player and the CD Audio player flows are Synchronous Data, as their service rate is the same of the MOST frame, while all the other flows are Isochronous Data. The messages used to set the connections are transmitted as Control Data, as required by the protocol. The service rate, shown in the fourth column, is expressed in μs and all the values are multiple of the service rate of the MOST frame (that is 20.8μss), which means that the sampling rate is equal to 48kHz. The activation time value, shown in the fifth column, is the same as in Table 1 and represents the time at which the connection manager receives the request to establish the connection between the source and the sink of a specific flow. As shown in Figure 6.2, for the MOST notwrok two separate rings are considered. One is dedicated to ADAS flows and the other one is dedicated to multimedia and cross-domain flows. Such a choice is made to avoid that ADASs flows experience long delays due to the large number of devices that to traverse on the same ring.

Type	Latency [μs]				LDR [μs]			
	AVB		MOST		AVB		MOST	
	Avg	Max	Avg	Max	Avg	Max	Avg	Max
Cameras	217	431	229	4002	396	456	5	8
LDW/TSR camera	265	423	229	4002	396	456	5	8
DA-Cam single flow	92	108	229	658	86	128	5	8
DA-Cam warnings	68	99	473	473	23	45	0	0
DVD Player	167	178	594	594	0	0	0	0
CD Audio Player	169	175	62.7	62.7	0	0	0	0
Cross-domain (1st flow)	16.8	19	473	473	0	0	0	0
Cross-domain (2nd flow)	16.8	19	473	473	0	0	0	0

Table 6.3: *Case A - Mean and maximum Latency and Jitter results*

Type	Latency [μs]				LDR [μs]			
	AVB		MOST		AVB		MOST	
	Avg	Max	Avg	Max	Avg	Max	Avg	Max
Cameras	217	431	233	6340	396	446	6	9
LDW/TSR camera	265	423	235	6345	396	446	6	9
DA-Cam								
Aggregated (3+1)	87	118	330	1654	92	147	6	7
DA-Cam warnings	86	143	472	472	0	0	0	0
DVD Player	165	178	594	594	0	0	0	0
CD Audio Player	170	174	62.7	62.7	0	0	0	0
Cross-domain (1st flow)	16.8	19	473	473	0	0	0	0
Cross-domain (2nd flow)	16.8	19	473	473	0	0	0	0

Table 6.4: *Case B - Mean and maximum Latency and Jitter results*

6.3 Results

Table 6.3 and Table 6.4 show the latency and jitter values for AVB and for MOST frames obtained in the Case A and Case B simulation scenario, respectively.

As the assessment proposed in this paper is made between heterogeneous communication protocols, in order to present a meaningful comparative evaluation, the following considerations are needed. As the frame length for ADAS traffic flows is obtained as a pseudo–random number (generated from a uniform distribution), the same sequence of pseudo-random numbers for packet lengths was utilized in both the AVB and MOST simulations for the sake of comparison. For the ADAS, DVD and cross-domain traffic flows, the MOST latency is calculated as the time required to receive all the fragments needed to re-construct a frame as large as the relevant AVB one. In the MOST protocol the latency value for the CD Audio flow that we would obtain considering the time required to receive all the fragments needed to re-construct a frame as large as the relevant AVB one is equal to 11.98ms, which is a high value. However, comparing this value to that obtained using AVB would be unfair, because in the MOST network the audio samples are delivered at the same rate as they are generated (i.e., 4 bytes at 48kHz, thus providing a good quality of service). For this reason, in order to correctly assess the MOST performance for the CD Audio flow, the latency results shown in Table 4 and 5 are obtained for a single packet, 4-byte long (i.e., for a single audio sample).

Case A assessment As shown in Table 6.3, the latency results for AVB are acceptable as the requirements of the traffic here considered that is, 33 ms for all the video flows from cameras and DA-Cam and 100 ms for audio and multimedia video flows [16][34], are fulfilled. The maximum experienced jitter is 456 μs. The jitter is due to the fact that several video streams are sent from the cameras to the same receiver (the DA-Cam), so the Ethernet frames are delayed in the relevant switch-outgoing queue. The latency values obtained for MOST are higher than those obtained for AVB or all flows. This is due to the ring topology, as the MOST frame containing, for instance, data for the DA-Cam from the right camera, has to cross all the nodes to reach the destination. Although the MOST network has been split in two separate networks, one for ADASs and the other one for multimedia and cross-domain, the negative effect on the latency cannot be avoided and worsens with the growing number of devices in the network.

Case B Assessment As shown in Table 6.4, in the second case, no significant increase of latency and jitter was found, except for the DA-Cam video traffic flow that, in this case, is an aggregated flow. For all the simulations both the latency and jitter related to the multimedia traffic flows are the same as those obtained in Case A. This is because the traffic increase in the ADAS network does not affect the multimedia and cross-domain links. Furthermore, in Case B, there is no difference as far as the latency and jitter values obtained for cameras flows are concerned. As the DA-Cam warning traffic flow shares the link with the DA-Cam video aggregated flow, i.e., both flows originate from the same node and are addressed to the Head Unit, the packet size increase for the DA-Cam flow determines an increase also for the mean latency of the navigation warnings flow. This is due to the delay experienced in the switch queue by the navigation warnings packets that have to wait that one or multiple larger video frames are forwarded to the Head Unit. Among the two networks, AVB offers higher jitter values than MOST, as in MOST follow a fixed order, so the jitter only depends on the variable length of the video frames, while in AVB all the frames originating from the cameras and directed to the DA-Cam are enqueued within switch S1 in a variable way. The jitter values for multimedia and cross-domain traffic, i.e. for constant payload traffic flows, are very low for both AVB and MOST. This is due to the packet size of the stream, which is fixed. Actually, the low jitter obtained in both the simulations is here set to zero, due to approximations in our simulator that does not model the jitter of the hardware implementation, which is much lower than 1 μs. As the evaluation of the hardware implementation is not the purpose of this work, it can be considered negligible. The highest latency value in the MOST network is obtained for the Audio CD flow and is equal to 11.9 ms. However, such a high value is not an issue, as the sampling rate of the MOST frames is equal to the standard audio sampling rate used by professional digital video equipment. This means that, even if 11.9 ms are needed to collect, at the destination, 1050B of data, the streaming of the audio flow is not interrupted.

6.4 Assessment of the simulators

In this work, three simulators are used, i.e., the AVB/AVB_ST simulator, the TTE simulator and the MOST simulator. All of them are realized using OMNeT++ simulation tool and the INET-framework. The validation for the TTE simulator is available in [58], while, the AVB/AVB_ST and MOST simulators were validated through a worst-case latency analysis, which is presented in the following.

Table 6.6 summarizes the notations. For AVB we adopted the compositional analysis proposed in [59], which allows to obtain the worst-case latency for the flows in an AVB network.

According to the model presented in [59], $w_i(q)$ can be bound by maximizing and adding all the delay contributions, as in Equation 6.1:

$$w_i(q) \leq t_{transfer}(q) + I_{LPB} + I_{SPB}(w_i(q)) + I_{HPB}(w_i(q)) \tag{6.1}$$

As explained in [59], under the assumption of respecting the minimum Ethernet frame size (i.e., 46 bytes) and the frame overhead plus the inter-frame gap (i.e., 38 bytes), C_i^+ can be obtained as in Equation 6.2:

$$C_i^+ = \frac{(38 + max(46, dataLength))}{pTxRate} \tag{6.2}$$

In the worst-case analysis, I_{LPB} is assumed to be calculated as in Equation 6.3

$$I_{LPB} = max_{j \in ip(i)}(C_j^+) \tag{6.3}$$

I_{SPB} depends on the arrival time $a_i(q)$ of the q-th frame of stream τ_i due to the FIFO scheduling within the same priority. It can be bounded a in Equation 6.4:

$$I_{SPB} = \sum_{j \in sp(i)} \{\eta_j^+(a_i(q)) \cdot C_j^+\} \tag{6.4}$$

I_{TPB} is the delay due to the traffic shaper blocking and depends on the allowed rate (idleSlope) of the corresponding class. The amount of credits consumed by a transmission lasting C_trans time units is $K_consumed$. If we consider the traffic shaper blocking observed by stream τ_i, C_trans can be bounded by the time required to transmit own and same-priority frames, i.e., $C_trans \leq t_{transfer}(q) + I_{SPB}(a_i(q))$. Hence, to replenish the credits consumed by previous frame transmissions and assuming that the first frame does not experience any traffic shaper blocking, hence the (q-1) in Equation 6.5, the following time is required:

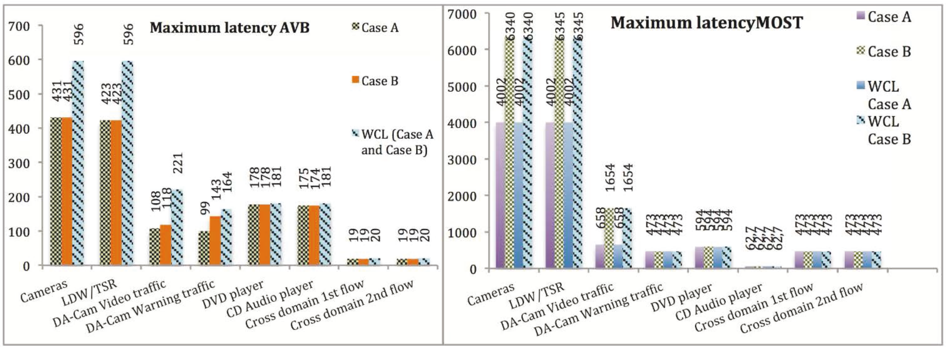

Figure 6.3: *Maximum latency obtained, through simulation and worst-case analysis, for both AVB and MOST*

$$
\begin{aligned}
I_{TBS}(a_i(q)) &= \frac{K_{consumed}}{idleSlope} = \\
= \frac{-sendSlope_i}{idleslope_i} \cdot C_{trans} &\leq [(q-1) \cdot C_i^{+} + I_{SPB}(a_i(q))] \cdot \frac{(-sendSlope_i)}{idleSlope}
\end{aligned}
\tag{6.5}
$$

Using Equation 6.1, the worst-case latency can be calculated for each flow taking into account the number of hops that each flow must go through to reach the destination and adding, for each hop, the switching processing time. In Figure 6.3 the maximum latency values are compared to the worst-case latency calculated using Equation 6.1 and considering a switching processing time equal to 8 μs.

For the MOST protocol the latency can be easily calculated and corresponds to the time needed for an entire MOST frame to traverse the ring, as in Equation 6.6

$$WCL = InterN \cdot \frac{(l_{MOSTFRAME} \times 8)}{dataRate_{MOST}} \cdot frameNumber \qquad (6.6)$$

The simulation model was assessed by comparing the Ethernet AVB and MOST timing parameters calculated using Equations 6.1 and 6.6 with those obtained in the simulation. Figure 6.3 shows the maximum latency results obtained through simulation and the worst-case latency calculated using Equation 6.1, for AVB and Equation 6.6, for MOST. We notice that, for both AVB and MOST, the maximum values obtained through simulations were never higher than the worst-case latency values obtained through the worst-case timing analysis. For AVB the worst-case latency is the same for Case A and Case B, this is due to the maximum frame payload values, which are the same in both cases. On the contrary, two different worst-case latency values are obtained, respectively, for Case A and Case B, this is because the maximum frame payload values used for MOST in the two cases are different and, consequently, the amount of byte allocated in the MOST frame for each flow changes between the two cases.

Since we used the same AVB simulator for all the works described in this dissertation, all the simulation models considered in this dissertation are validated.

6.5 Conclusions

The paper proposed comparative assessments of the IEEE AVB and the MOST protocols in automotive scenarios, providing simulation results obtained with realistic traffic patterns. The outcome of this study is that AVB performs better than MOST for camera-based ADAS traffic, as the maximum latency values experienced by the video frames sent by the cameras are significantly lower in AVB than in MOST. Moreover, AVB is far more scalable than MOST. In fact, when the number of nodes in a MOST network increases, so does the latency. In addition, MOST does not scale well, as the MOST frame has a fixed size, so the number of flows that can be accommodated for transmission in one MOST frame is limited. Another weak point of MOST is that the delay for a flow depends on the position of the node on the ring, i.e., on the distance between the flow source and destination node. To alleviate this problem, multiple MOST rings could be deployed, but this choice would entail more wiring, weight and network complexity, features that are not convenient for the automotive context. AVB does not outperform MOST in the support provided to CD Audio flows because of its higher latency. Nevertheless, the latency values for both protocols are suitable for multimedia applications, thanks to the presence of end-to-end buffers (which are commonly used in multimedia streaming, but would not be advisable for time-constrained applications such as camera-based ADAS). As a result, AVB proved to be suitable for both camera-based ADAS and multimedia/infotainment applications, thus paving the way for its adoption in these functional domains.

Symbol	Definition
τ_i	The i-th stream.
$a_i(q)$	The arrival time of the q-th frame of stream τ_i
q	The number of frames of a stream.
$sp(i)$	The set of same-priority streams mapped onto the same resource of stream τ_i.
$pTxRate$	The network data rate.
$\eta_j^+(a_i(q))$	The maximum number of frames expected before the arrival of the q-th frame of stream τ_i.
$dataLenght$	The frame length.
$idleSlope$	The rate of credit replenishment.
C_x^i	The worst-case execution time, i.e., the maximum time a frame takes to pass through a port without experimenting interference, i.e., without waiting in the queue for the end of other transmissions.
I_{TSB}	The delay due to traffic shaper blocking, that depends on the allowed rate (idleSlope) of the corresponding class.
$t_{transfer}(q)$	The maximum transfer time for q packets of a stream τ_i(i.e., $t_{transfer} = q \cdot C_i^+$).
$sendSlope$	The rate of credit consumption.
$lp(i)$	The set of lower priority frames in the same queue.
C_{trans}	The time needed for accomplishing a transmission.
I_{LPB}	The set of lower priority frames in the same queue.
$K_{consumed}$	The amount of credits consumed by a transmission lasting C_{trans} time units (i.e., $K_{consumed} = -sendSlope_i \cdot C_{trans}$)
I_{SPB}	The maximum delay due to the on-going transmission of a lower priority frame.
$w_i(q)$	The longest time required to forward q frames of a stream τ_i.
I_{HPB}	The number of nodes in the ring between the sender and the receiver.
WCL	The worst-case latency.
$InterN$	The number of nodes in the ring between the sender and the receiver.
$l_{MOSTFRAME}$	The length of the MOST frame, i.e., 384 byte.
$dataRate_{MOST}$	The transmission rate of MOST 148 Mbps.
$frameNumber$	The number of MOST frames required to deliver to the receiver all the bytes needed to reconstruct a frame of the same length as the relevant AVB one.

Table 6.6: *Summary of notations*

CHAPTER

SEVEN

PROVIDING SUPPORT TO SCHEDULED TRAFFIC ON IEEE AUDIO VIDEO BRIDGING NETWORKS

As mentioned in section 2.5, the IEEE Time Sensitive Networking (TSN) task group of the IEEE 802.1 working group is preparing the next generation of IEEE 802.1 standards, called AVB gen2. Among all the standards that the TSN group is working on, i.e., [17, 18, 19, 20, 21], significant attention is given to the IEEE 802.1Qbv standard. Inspired by the ongoing activity within the IEEE Time Sensitive Networking task group, the work described in chapters 8 and 9 introduce two novel strategies for proving support to scheduled traffic in IEEE AVB networks. Scheduled traffic is characterized by small-payload, high priority and off-line scheduled. Differently from the current trend in [60], that investigates the inclusion of scheduled traffic in one of the traffic classes already provided by the IEEE AVB standard (i.e., the Stream Reservation Class A), in chapter 8 we propose a separate class for scheduled traffic, called the Scheduled Traffic (ST) class, that should be added on top of the already existing traffic classes provided by the IEEE AVB standard.

Using the approach described in chapter 8, a sequence of ST frames arriving one after the other with short interleaving may prevent the transmission of a non-ST frame for a considerable amount of time, as the TABS is activated for each ST frame. To overcome the above mentioned limitation while at the same time simplifying the planning of the ST transmissions, in chapter 9 a phase-based mechanism is proposed. The phase-based mechanism has two main advantages. First, it does not require TABSs, but a simple counter. Second, it makes easier the management of the ST transmissions merging all the ST_Windows in a single large interval, called the red phase, during which multiple ST frames are transmitted. After the red phase, other phases follow (green, orange) during which the other traffic classes can be transmitted without suffering from the interference of the ST flows. Phases follows each other in a cycle. Isolation between ST transmissions and those of other traffic classes is enforced through a yellow phase, during which no new transmissions are allowed, which is only used to complete any ongoing transmission before the start of the red phase of the next cycle.

7.1 Related Work

A number of works focused on the Ethernet AVB ability to cope with the requirements of the real-time traffic typically found in industrial automation [61]. In [62] a performance comparison between AVB and standard Ethernet is presented. The outcome of the study is that, although AVB allows for determining the worst case latency for all real-time message classes, further improvements are still needed for use in industrial automation. In fact, as the CBFQ used in AVB adopts non-preemptive scheduling, in the worst case a real-time frame might be delayed in every bridge by the ongoing transmission of a maximum sized frame not belonging to the real-time class. Approaches to mitigate this interference, such as packet preemption, fragmentation, and synchronous scheduling [63], are discussed in [64]. The work in [65] addressed the performance of AVB in aeronautic networks and showed that, when using time synchronization, the timing requirements are met.

CHAPTER

EIGHT

SCHEDULED TRAFFIC OVER IEEE AUDIO VIDEO BRIDGING: THE AVB_ST APPROACH

In this chapter the AVB_ST approach [66] is described

8.1 Scheduled Traffic Class (ST-Class)

In [66] a type of time-sensitive control traffic, called Scheduled Traffic (ST) is introduced. By definition, ST traffic flows are periodic and the characteristics of this traffic (period, frame size) are fixed and a priori known. The transmission sequence for ST messages is off-line planned by a scheduler and offset scheduling techniques [67] are adopted in the designing phase to make sure, by design, that conflicts between transmissions of ST messages will not occur in the entire network, i.e. at end stations as bridges. The expected receive times of the ST messages are computed off-line, as will be explained in Sect.

8.2. All the nodes that are either traversed by ST flows or are the intended recipient of ST flows, i.e. either bridges or listeners, are fed with the schedule of the relevant flows. As mentioned before, the TABS mechanism here adopted, that will be detailed in the following subsections, avoids any interference on ST messages from other traffic classes. Therefore also the expected receiving time of any ST message can be calculated on the receiving side, albeit with some uncertainty depending on the clock synchronization protocol. For this reason, suitable time intervals, here called ST_Windows, are defined and Sect. 8.2 discusses the ST_Windows sizing. The support of the ST Class according to our approach requires a reliable synchronization of the network nodes, such that every node knows when it is the right time to transmit its ST messages. As mentioned in Sect. 2, in AVB clock synchronization is performed by the IEEE 802.1AS protocol, whose performance for a switched Ethernet network were evaluated in [25]. In [66] some assumptions, taken from [25], are made:

- Fast Ethernet 100 Mbit/s full-duplex links.
- Switch processing time: 10 μs.
- Clock speed: 40 ns tick interval at least, or 25 MHz clock frequency.

Under these assumptions, it was shown in [25] that the synchronization process based on the IEEE 802.1AS is not affected by high network workload and that the synchronization error and its accuracy remain below the specified value of 1 μ s over seven or fewer hops. The commonly suggested value for the synchronization interval is 125 ms. However, for the purposes of the approach described in [66], is safer to chose a shorter sync interval of 62.5 ms that, as suggested in [25], reduces the maximum synchronization error to approximately 50% and only requires 0.027 Mbit/s per link to exchange synchronization messages. The protocol described in [66], called AVB_ST, relies on the support provided by the IEEE 802.1AS as far as clock synchronization and the notion of global time are concerned. Thanks to the syntonization, the time base used in the entire network is the one of the grandmaster and is the same for every node. ST messages are tagged with the highest priority TAG according to the IEEE 802.1Q standard. SR Classes A and B take the second and the third highest priority, respectively. The TAG, which is part of the IEEE 802.1Q header, has values ranging from 0 to 7. In the AVB_ST approach, the mapping of the priority TAG on each traffic class (i.e., ST, SR Class A, SR Class B or best effort) is known to all the devices that belong to the network (both end-station and bridges). ST traffic is handled in a separate queue and does not undergo traffic shaping. On the contrary, SR Classes are handled by the credit based shaper (CBS).

8.2 Sizing the windows

The communication mechanism foreseen in the AVB_ST approach avoids any interference on ST messages from other messages belonging either to the same class or to any other traffic class. As a result, the receiving time of any ST message can be calculated. However, the calculation of the reception instant for any ST message has to take into account the synchronization error between the nodes and the drift of each station. This drift is expressed in parts per million (ppm) and

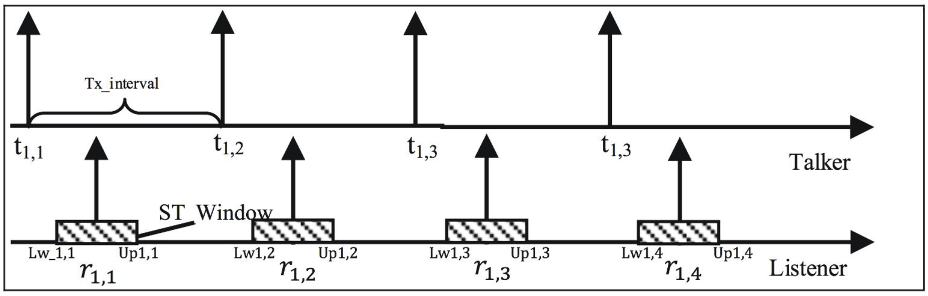

Figure 8.1: *The ST_Windows*

represents the clock speed. The difference between the drift of a node and the drift of the reference node can be different from node to node. According to the drift, during a synchronization interval, i.e., between a synchronization instant and the next one, the local clock of each node deviates from the reference clock. The difference between two clock values measured at the same instant of time is called skew.

On each receiving side (the bridge or the listener) for each ST message, a time window within which the message has to be received, called an ST_Window, is defined. If the ST message arrives outside this window, it is considered lost. The ST_Window is centred on the expected receiving time, called *expectedReceiveTime*, and is equal to twice the worst deviation (here indicated with δ) between the local clock of the receiver and the local clock of the sender, as in equation 8.1:

$$ST\_Windows = 2 \times \Delta \qquad (8.1)$$

This is because two time-aware systems cannot differ in time by more than $2 \times \Delta$, as this is the maximum time difference between two local clocks of two different nodes. Δ depends on the synchronization protocol used and on the accuracy of the clock, it is computed as in equation 8.2:

$$\Delta = max\_sync\_err + max\_skew \tag{8.2}$$

where max_sync_err is the synchronization error (1μs for IEEE 802.1AS) and max_skew is the maximum possible skew between two consecutive nodes, that depends on the difference between their drifts and is computed after an entire synchronization interval. Figure8.1 shows the message transmission sequence for an ST traffic stream. In the figure, $t_{i,j}$ represents the send time of the j-th message of the i-th stream from the talker. Transmission occurs every *TxInterval* and $r_{i,j}$ is the

expected receive time of the j-th message of the i-th stream (sent at $t_{i,j}$), calculated as:

$$expectedReceiveTime_{i,j} = r_{i,j} = TxTime + t_{i,j} \tag{8.3}$$

where TxTime is the time needed to transmit the message from the sender to the receiver. Eq. 8.3 does not include cable delay, as it is negligible over distances shorter than 100 meters. TxTime depends on the number of hops for the stream and is calculated similary to [60], as eq. 8.4:

$$TxTime = TxTime_{Talker} + \sum_{n=0}^{BridgeHops} TxTime_{Bridge_n} \tag{8.4}$$

Where $TxTime_Talker$ is the time needed to transmit one frame out of the Talker, given by:

$$TxTime_Talker = (Payload_i + Overhead + IFG) \times 8 \times 10 \tag{8.5}$$

where:

- Payload is the payload of each frame of the i-th stream expressed in bytes;

- Overhead represents the MAC layer overhead added to each IEEE 802.1Q frame (22 bytes);

- IFG is the inter-frame gap (according to the Ethernet standard recommendation, 12 bytes);

- 10 is the transmission time for one bit on Fast Ethernet, expressed in nanoseconds.

In Eq.8.4, BridgeHops is the number of bridges that a frame has to cross to reach the Listener. $TxTimeBridge_n$ is the time a frame takes to traverse the n-th bridge, as shown in eq. 8.6

$$TxTimeBridge_n = t_{BRIDGE} + t_{TX} \tag{8.6}$$

where:

- t_{BRIDGE} is the time needed to process the frame within the bridge. By design, ST traffic does not wait in the bridge queue. This is because ST traffic is properly staggered through off-line scheduling, so as to not pile up in the queues. Moreover, ST traffic does not wait for other traffic classes thanks to TABS. As a result, t_{BRIDGE} is a constant that depends on the bridge specifications. Here we assume Store and Forward bridges and a t_{BRIDGE} value of 10 μs.

- $t_T X$ is the time needed to transmit the frame of j-th stream out of the bridge, computed as in eq: 8.7

$$t_{TX} = (Payload + Overhead + IFG) \times 8 \times 10 \tag{8.7}$$

To assess whether the j-th message of the i-th stream has been entirely received within its ST_Window, the following values are observed for each j-th message of the i-th stream.

- $UpperSTWindows_{i,j}$ ($Up_{i,j}$ in Figure8.1), i.e., the upper value of the ST_Window;
- $lowerSTWindows_{i,j}$ ($Lw_{i,j}$ in Figure8.1), i.e., the lower value of the ST_Window;

These two values are calculated as:

$$Up_{i,j} = expectedReceivedTime + ST\_Windows/2 \tag{8.8}$$

$$Lw_{i,j} = expectedReceivedTime - ST\_Windows/2 \tag{8.9}$$

If the message is delivered out of its ST_Window, it is considered invalid and is discarded.

8.3 AVB with scheduled traffic support: Outline

To enforce temporal isolation for the ST Class, thus preventing any non-ST message transmission that could delay the transmission of the next ST message, all the non-ST traffic goes through a TABS both in the bridges and in the talkers. The idea of TABS was proposed in [60], to preserve the QoS of SR Class A traffic in AVB networks. In [60] the TABS blocks any lower priority transmission (i.e., from SR Class B or non-AVB traffic) that would interfere with the upcoming transmission of SR Class A traffic. For instance, if a given non-SR Class A queue has a frame ready, but the transmission of such a frame, if allowed, would delay the start time of the next transmission of SR Class A traffic, such a transmission is not allowed. In [66], is proposed to adopt TABSs to provide temporal isolation between the ST traffic and all the other traffic types. Thanks to the TABS mechanism, any non-ST message that is enqueued, ready for transmission, has to wait not only for the duration of an ST message transmission, but also for an additional time if the time needed for its transmission is longer than the difference between the send time of the next ST message and the current time. This time difference, called a minimal distance, is enforced to avoid the transmission of non-ST traffic that would delay the next ST message. This way, an ST message ready for transmission will never experience blocking due to the ongoing transmission of a non-ST message, as the transmission of a non-ST message that could delay the ST message will not be allowed. In the AVB_ST approach, therefore, there is no need for preempting the ongoing transmission of a non-ST message, as if such a message is being transmitted, this means that it will not interfere with any ST message, otherwise it would have been blocked in advance. To avoid bandwidth waste, in [60] it is suggested to make the TABS inspect the next priority queue and see whether such a queue has a smaller frame whose transmission would finish before the start time of the next Class A transmission. If this is the case, this frame is allowed to start. In our AVB_ST approach, therefore, in the time interval between two consecutive ST message send times, the queues of non-ST traffic, starting from the non-empty highest priority one, are looked for a ready message whose transmission would finish before the send time of the next ST message. If the ready message in the highest non-empty priority queue (e.g., SR Class A) requires a transmission time greater than the time interval up to the next send time for ST traffic, the message is not selected for transmission, and the next highest priority queue is inspected (e.g. SR Class B). In the end, either a suitable message is found or no transmission is allowed. The maximum time taken by the search is the time needed to perform a comparison between two different time values multiplied by the queues number. Such a number is negligible, as the time difference computation is made "on the fly", and concerns the propagation time in logic gates. The TABS mechanism represents a trade-off between the need to temporally isolate ST traffic, avoid any blocking due to lower priority traffic, and bandwidth utilization. In the AVB_ST approach, therefore, the messages in the queues associated to the SR Classes undergo both the TABS and credit shaping, while best effort messages go through the TABS only, as no shaping is foreseen for best effort traffic in the standard [6]. In our design, shaping for SR Classes is performed after the TABS, so that credits are consumed only when the frame transmission is allowed by the TABS. Summarizing, in our design offset based scheduling prevents ST traffic from experiencing queuing delays due to other ST traffic, TABSs avoid interference from the other traffic classes and strict priority handling for ST traffic avoids the additional delays due to CBS. This design makes the ST traffic latency both low and predictable.

8.4 SRP with Scheduled Traffic

In AVB_ST the bandwidth reservation has to be accomplished taking into account the bandwidth reserved to ST streams. In addition, the computation of the correct *adminIdleSlope* for each traffic class (described in section 2.5) has also to consider the guard band, i.e., the bandwidth not utilized when the enforcement of the minimal distance (as seen in section 2.5) does not allow any transmission between two consecutive ST send times. For this reason, a new parameter for the SRP, here called *realPortTransmitRate* (RPTR), has to be defined, as follows:

$$RPTR = PTR\text{–}adminIdleSlope(ST) \tag{8.10}$$

where PTR is defined, as in section 2.5, and adminIdleSlope(ST) is the bandwidth reserved to ST traffic, which is calculated off-line. The RPTR is therefore the bandwidth that is left to SR Classes and best effort traffic. The bandwidth reserved to SR Classes is defined by the adminIdleSlope(N) parameter. The adminIdleSlope for a specific SR stream is obtained as in the standard [6], with the difference being that the new RPTR parameter is used instead of the RPTR. However, this new definition of the transmission rate does not take into account the guard band mechanism.

The bandwidth not utilized due to the guard band depends on the number of ST transmissions and on the size of the maximum non-ST frame that may interfere with these ST transmissions. A simple way to calculate the guard band is to make a worst case assumption, i.e., considering that the interfering frame has always the maximum Ethernet frame length, but this assumption is overly pessimistic. For this reason, here a more realistic method for dynamically computing the guard band is proposed, that uses the largest frame between all the SR Class flows that are registered at the node. The mechanism works as follows. At the time of registration attempt of an SR Class A or SR Class B stream at a node, the bandwidth availability for the specific traffic class is checked. If there is bandwidth available then the frame size for the specific flow is examined. If this size exceeds the maximum frame size of all the other SR flows already registered at the relevant node, then the ratios between such a size and the period of each ST stream are computed, as for each ST message a guard band to prevent the largest SR Class frame from interfering is needed. The sum of the values thus obtained represents the maximum guard band needed to isolate the ST traffic of that specific node. This estimate is still pessimistic, but safe. If the guard band amount exceeds the given threshold, then the SR stream cannot be registered. We heuristically set a threshold equal to 95% of the bandwidth left over for best effort traffic to leave room for network traffic such as, synchronization messages, that, as calculated in [25], require 3% of the network bandwidth to realize a synchronization period of 62.5 The same admission control mechanism is repeated every time a new AVB SR flow enters the registration procedure. Summarizing, in AVB_ST, for each node, the PTR of each port is initially reduced by the bandwidth reserved to ST traffic (that is known a priori, thanks to the off-line scheduling). The remaining bandwidth is subsequently distributed between the SR Classes, according to the deltaBandwidth(N) (seen in section 2.5) parameter. What remains is left for both best effort traffic and the guard band. If the guard band reaches 95% of the remaining bandwidth, then it will not be possible to register any SR traffic flow with a frame size larger than the one used to compute the current guard band.

8.5 Performance evaluation of AVB_ST

This section presents a proof-of-concept assessment of the AVB version proposed inm[66], also in comparison with standard AVB and TTEthernet (TTE). The network performance was evaluated using the OMNeT++ simulation tool and the INET-framework. The AVB simulator used here is the same described in section 4.2, and the AVB_ST simulator was obtained extending the AVB one. The simulated protocol stacks feature the Application level on top of Ethernet, therefore the overhead is 22 bytes for both the AVB versions and 18 bytes for TTE, respectively. Three simulations, called AVB_ST, AVB and TTE were performed. Each simulation was 600 s long so as to collect a large number of samples, and all the flows started at the same time to assess a critical scenario from the interference point of view. In the following the definitions of the considered performance metrics are given:

- Latency: the one-way end-to-end frame delay, i.e., the time from the source sending a packet to the destination receiving it.
- Absolute Jitter: the difference between the maximum and the minimum latency value for a given flow.

The simulation scenario considered in [66] consists of 6 nodes and 4 bridges B_j, with j=1..4. The network topology is shown in Figure8.2. There are 5 talkers, called $node_k$ with k=1..5,and one listener, i.e., $node_6$. Every talker sends frames to $node_6$, this because the purpose of this simulations is to load as much as possible the queues of the B_1

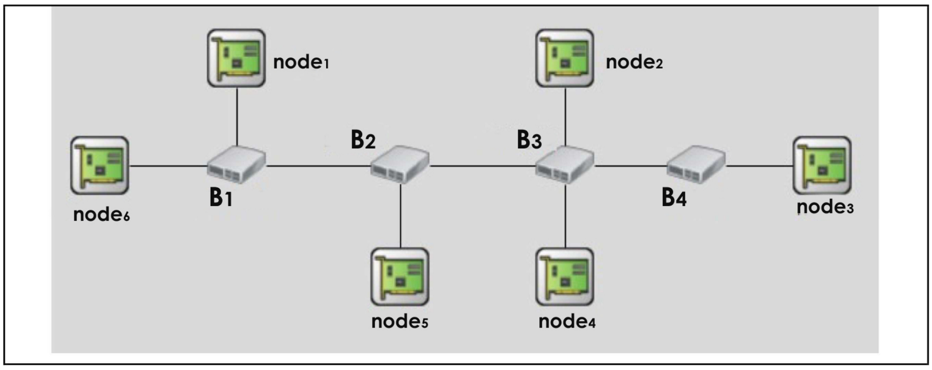

Figure 8.2: *The network topology*

and generate a sort of bottleneck to evaluate the interfering problems. Table 8.1 shows the traffic model and the mapping of the traffic flows on the available traffic classes for each simulation. All the flows are time-sensitive, but with some differences, as the ones generated by $node_k$, with k=1..4, are assumed to be more time-critical than the flow generated by $node_5$. In particular, the flows generated by $node_3$ and $node_4$ and labelled as ST are scheduled traffic, while the non-ST flows are indicated as NST in Table 8.1. The second column of Table 8.1 gives the Application level payload. All the flows in Table 8.1 are periodic, with period equal to the service rate (shown in the third column), and have constant payload. The switch processing time is equal to 10 μs.

The AVB_ST simulation In this simulation, which refers to the AVB version with scheduled traffic, the priority value assigned to the ST class is 7, as this flow is given the highest priority, while the values for the SR Class A and B are 6 and 5, respectively. Following the

Talker type	Payload (Byte)	Service rate (μs)	Traffic Class AVB_ST	Traffic Class AVB	Traffic Class TTE
$node_1$	256	250	SR A (prio 6)	SR SR A (prio 7)	RC(prio 2)
$node_2$	256	250	SR A (prio 6)	SR SR A (prio 7)	RC(prio 2)
$node_3$ - NST	256	300	SR A (prio 6)	SR SR A (prio 7)	RC(prio 2)
$node_3$ - ST	128	500	ST (prio 7)	SR SR A (prio 7)	RC(prio 1)
$node_4$ - NST	256	500	SR A (prio 6)	SR SR A (prio 7)	RC(prio 2)
$node_4$ - ST	128	500	ST (prio 7)	SR SR A (prio 7)	RC(prio 1)
$node_5$	256	500	SR B (prio 5)	SR SR B (prio 6)	RC(prio 3)

Table 8.1: *Traffic model and configured traffic classes*

$node_1$	$node_2$	$node_3$	$node_4$	$node_5$	$node_6$	B_1	B_2	B_3	B_4
-30	20	10	-15	-50	0	30	-15	40	50

Table 8.2: *Drift difference between time-aware systems and the reference node [ppm]*

approach proposed in this work, the bandwidth that is left to the SR Classes and best effort traffic is the difference between the total available bandwidth and the sum of the bandwidth needs of the ST traffic, i.e., 4.096 Mbit/s in our case, plus the bandwidth required to support the 62.5 ms synchronization interval, i.e., about 3 Mbit/s according to [25]. The 75% of the remaining bandwidth, similar to what is foreseen in the IEEE 802.1Qat standard [4], is reserved to SR Class A and B, while the other 25% is left for best effort traffic and the guard band. As a result, the overall bandwidth that SR Classes can be reserved is 69.7 Mbit/s, while 23 Mbit/s are left for best effort traffic and guard band. As much as 40% of the bandwidth available for the SR Class is reserved to Class A, while the remaining 35% is reserved to Class B. ST messages are sent at regular time intervals and the

Stations	**Max_skew** (μs)	Δ
$node_1 \Longleftrightarrow B_3$	3.43	$\Delta_{B_3} = (1 + 3.43)$
$node_3 \Longleftrightarrow B_4$	4.062	$\Delta_{B_3} = (1 + 4.06)$
$B_4 \Longleftrightarrow B_3$	0.625	$\Delta_{B_3} = (1 + 0.63)$
$B_3 \Longleftrightarrow B_2$	3.437	$\Delta_{B_3} = (1 + 3.44)$
$B_2 \Longleftrightarrow B_1$	2.812	$\Delta_{B_3} = (1 + 2.81)$
$B_1 \Longleftrightarrow B_0$	2.812	$\Delta_{B_3} = (1 + 1.87)$

Table 8.3: *Maximum skew and Δ values*

cycle time, i.e., the least common multiple of their periods, is equal to 500 μs in the considered case. The ST_window is sized as $2 \times \Delta$. This value is off-line calculated for all the nodes and bridges traversed by ST frames (i.e., B_4, B_3, B_2, B_1, and $node_6$) using equation 8.2, where the maximum synchronization error (max_sync_error), according to the findings in [25], is 1μs and the maximum skew (max_skew) is obtained from the drift values in Table 8.2 applying the formula 8.11

Table 8.3 shows the maximum skews between consecutive nodes and the Δ values for each station crossed by an ST flow, obtained from 8.2.

$$max\_skew = drift\_difference \times sync\_interval \tag{8.11}$$

The drift is considered a fixed value because, as show in [68], after each synchronization the skew starts growing again but, due to the very small drift variations, it grows in a quasi-linear way. Each ST_Window is centred on the *expectedReceiveTime* instant obtained from equation 8.3. Here the t_{BRIDGE} is assumed equal to 10 μs, as in [25], and the time to transmit one frame out of the bridge or of the Talker node is calculated according to the frame size, also considering the 22 bytes overhead per frame. Time checks on the minimal distance are used to implement TABS. From the stream model in Table 8.1, it is possible to calculate the amount of bandwidth reserved for the guard band, that depends on both the transmission period of the scheduled traffic (500 μs), and the maximum frame size among all the registered streams (256 bytes for AVB SR Class flows). We calculate the guard band, as in eq. 8.12

$$GuardBand = \frac{(256 + 22) \times 8}{500 \times 10^{-6}} = 4.096 Mbit/s \tag{8.12}$$

As there are two different nodes that send ST traffic, offset scheduling is applied to ensure that the ST_Windows of the two ST flows at the common bridge they traverse, i.e. B_3, are one after the other, to avoid bandwidth waste. We accomplish this adding a suitable offset (22 μ s in our case) to the ST flow generated by $node_4$.

The AVB simulation This simulation refers to the AVB standard. Here, the time-sensitive traffic is mapped on the Stream Reservation classes, as shown in Table II. SR Classes A and B are given priority 7 and 6, respectively. Following the SRP, 75% of the total bandwidth was reserved for SR traffic (40% for Class A and 35% for Class B). All the SR traffic is handled by CBS.

The TTE simulation Table 8.1 shows the mapping used in our TTE simulations. Scheduled traffic was mapped on the TT class and

Talker/ Type	Latency [μs]						Absolute Jitter [μs]		
	AVB_ST		*AVB*		*TTE*		*AVB_ST*	*AVB*	TTE
	Avg	**Max**	**Avg**	**Max**	**Avg**	**Max**			
$node_1$	55	55	55	55	51	53	0	0	6
$node_2$	121	121	132	145	110	115	0	24	10
$node_3$ - NST	154	187	159	174	154	157	32	14	8
$node_3$	103	103	141	165	122	122	0	30	0
$node_4$ - NST	145	145	121	121	143	150	0	0	10
$node_4$	89	89	149	149	100	100	0	0	0
$node_5$	90	95	94	94	97	100	7	0	6

Table 8.4: *Mean and Maximum latency and absolute jitter, for each simulation*

was given the highest priority, i.e. level 1. The other streams were mapped on the RC class with level 2 and 3, respectively. Strict Priority scheduling is used between the switch queues, FIFO within the same queue.

8.6 Results and Conclusion

Table 8.4 shows that the simulation results for the ST traffic flows obtained by AVB_ST outperform the ones obtained for the same traffic by standard AVB. In particular, for ST traffic flows we not only observe a lower latency with AVB_ST than with standard AVB, but we also see that the mean and maximum latency with our approach are equal, thus the jitter is zero. This is because, in AVB_ST approach, ST traffic does not experience queuing delays, thanks to offset scheduling and TABS, is provided with preferential service while crossing the network and does not undergo shaping, so the latency is low and predictable. Conversely, with standard AVB, the ST originated from $node_3$ experiences higher and non-constant latency. This is due to the interference from other traffic flows in the Class A, (i.e., those originating from the same node, from $node_2$ and $node_4$) and to the additional delay due to CBFQ. The AVB_ST results are comparable with TTE results for both the ST flows from $node_3$ and $node_4$, and in both cases a null jitter is found. The ST flow from $node_4$ hows a lower latency than the $node_3$ ST flow, as the latter has one more bridge to cross (i.e., B_4) to reach the destination. In all these simulations, $node_1$ flow experienced the lowest latency, as this flow is the only one generated from $node_1$ and has to cross only one bridge to reach the listener. In AVB simulations, the latency for this flow is constant as, according to our settings, the frames sent from $node_1$ reach the listener without waiting in the bridge queue (as the Class A queue of B_1 is empty when $node_1$ frames go through it). The second lower latency value is found for the AVB stream coming from $node_5$, although such a flow is mapped on the lowest priority in all the simulations, as this flow is the only one generated at $node_5$ and has to cross only two bridges, thus experiencing a limited interference from other flows. The maximum latency for the non-ST flow generated by node is lower with standard AVB than with AVB_ST. This is natural, as the latter downgrades the flow, while standard AVB serves it in the highest priority class, i.e. SR Class A. However, the latency increase, for this non-ST flow, in AVB_ST is not dramatic, and as far as the mean latency is concerned, the three protocols offer comparable values. The maximum latency for the $node_3$ non-ST flow is lower with TTE, that serves this flow in the RC class with the second highest priority, and also the jitter obtained by TTE is the lowest one. This is because TTE does not apply CBS. We underline that the non-ST flow from $node_3$ is the one that experiences the highest latency with AVB_ST, as this flow not only suffers from the interference due to the ST traffic class and the TABS mechanism, but is also affected by all the other SR-Class A flows, as this flow has to cross five hops (and so, all the bridges) to reach the listener.

Concluding,the AVB_ST approach proposed in [66] to introduce support for scheduled traffic in AVB networks proved to be very beneficial. Thanks to the offset-based scheduling, the temporal isolation provided to the ST traffic class through the TABS mechanism, and strict priority handling, ST traffic obtained low and predictable latency values, without significantly affecting SR traffic.

CHAPTER

NINE

SCHEDULED TRAFFIC OVER IEEE AUDIO VIDEO BRIDGING: A PHASE-BASED APPROACH

In this chapter the AVB_P approach is presented.

9.1 The AVB_P approach

The AVB_P approach described in this paper is organized in cycles., as shown in Fig. 9.1. Each cycle consist of four phases, each one identified with a color: red, green, orange, and yellow. Each node decides to send a different frame depending on the current phase, according to the following rules.

- During the Red phase, only ST frame transmissions are allowed.
- During the Green phase, only AVB SR Class frames can be trans-

mitted. First, the AVB Class A queue is inspected, searching for an AVB Class A frame to send. Once that the Class A queue is emptied, the queue of Class B frames is inspected. If, during the transmission of an SR Class B frame, an SR Class A frame arrives, the latter will be delayed by the transmission of the Class B frame; once it is completed, the Class A frame will be sent if the green phase is not expired.

- During the Orange phase, SR traffic and best-effort traffic are allowed. However, transmission is ruled by priority, so if there is any SR frame enqueued, the latter will be sent before any best-effort frame.

- During the yellow phase, no transmissions can start, as this phase is only used to complete any ongoing transmission. The yellow phase is therefore a safety margin to make sure that, when a new cycle starts, there will not be ongoing transmissions from the previous cycle that could delay the sending time of an ST frame.

Sizing the phases The duration of each phase is defined off-line (as explained in the following) based on the traffic model that characterizes the network.

The SR Class A and Class B frames undergo a credit-based fair queuing mechanism, while best-effort and ST traffic are handled by strict priority. However, in the AVB_P approach, the portTrasmissionRate (PTR) parameter of the IEEE AVB standard mentioned in section 2.5 has to be revisited, as a portion of the total bandwidth is now reserved for the transmission of the ST frames and another portion is left unused due to the yellow phase. As a result, the bandwidth reservable for the SR Classes ($B_{SRClass}$) is computed as in Formula (9.1)

$$B_{SRClass} = PTR - B_{STClass} - B_{yellowPhase} \tag{9.1}$$

where $B_{STClass}$ is the bandwidth reserved for the transmission of ST frames, while the $B_{yellowPhase}$ is the bandwidth unused for the sake of avoiding interference on ST frames. The latter is calculated as in formula (9.2), considering the worst-case scenario in which the interfering frame for the ST traffic is assumed to be the longest frame that can be transmitted on the network.

$$B_{yellowPhase} = \frac{size_{longestFrame}}{cycleTime} \tag{9.2}$$

The cycle time is the interval between the start time of two consecutive red phases, as shown in Fig. 9.1, and is equal to the shortest period among all the periods of the flows that are mapped on the ST Class.

In order to maintain the compatibility with the IEEE 802.1Q networks, AVB_P approach does not require any change to the AVB frame structure. The AVB_P mechanism is simply based on counters of the microseconds elapsed during a cycle since the start of each red phase. There is one counter for each network node, i.e., end station or bridge, and counters are configured off-line. The duration of each cycle time is defined off-line too. The counter is set to zero at the beginning of the red phase and the maximum value that can be reached by the counter is equal to an entire cycle time. Whenever this values is reached, the counter is reset to zero. To know the current phase when the queues are inspected to find a new frame, the counter value is compared with the range of each phase. Phase duration, and the relevant temporal ranges, are static and configured off-line. The time synchronization obtained through the 802.1AS standard, as described in 2.5, guarantees that nodes have a common notion of time, therefore counters are aligned.

The duration of the red phase depends the ST traffic that has to cross the network. As the scheduling of the ST traffic is planned off-line, the length of the red phase is the sum between the maximum time needed to transmit the ST frames from the source to the destination and the Δ, defined in [66] as the sum of the synchronization error (1μs for IEEE 802.1AS) and the maximum possible skew between two consecutive nodes. In principle, the length of the red phase should also consider the possible delay caused by the interference due to other ST frames (as only ST frames might interfere with other ST frames). However, in the AVB_P approach such an interference is avoided adopting offset scheduling [69].

The duration of the yellow phase is computed as the time needed for transmitting the longest frame over a single link in the network. This choice ensures that no frame transmission started during the green phase would interfere with the transmission of an ST frame in the red phase of the next cycle. The duration of the green phase is computed as in Equation (9.3):

$$greenP = GR \times (cycleTime - redP - yellowP) \tag{9.3}$$

where redP and yellowP are the duration of the red phase and of the yellow one, respectively, and GR is a percentage that is heuristically set based on the amount of bandwidth to be reserved to the

SR traffic. GR is 100% if there is not best effort traffic, otherwise it depends on the amount of best effort traffic.

The duration of the orange phase is computed as in Equation (9.4):

$$orangeP = cycleTime - redP - yellowP - greenP \tag{9.4}$$

The admission control for a new SR Class A or Class B flow is made based on both the deltaBandwidth(N) (defined in section 2.5) and the sum of the duration of the green phase and the orange phase. The maximum amount of bytes, for each SR traffic Class, that can be sent across a bridge port during sum of the duration of the green and the orange phases is calculated as in Equation (9.5)

$$MaxByte_N = SRClassBandwidth_N \times (greenP + orangeP) \tag{9.5}$$

where SRClassBandwidth is the amount of bandwidth reserved for given traffic class and is defined as in Formula (9.6)

$$SRClassBandwidth = \frac{deltaBandwidth(N)}{100} \times B_{SRClass} \tag{9.6}$$

where deltaBandwidth(N) is the percentage of bandwidth to reserve for the SR Class traffic and $B_{SRClass}$ is the bandwidth reservable for the SR Class as defined in Equation (9.1).

For each requesting flow belonging to a given SR traffic class, the bridge compares the number of bytes that the flow requires to send with the total amount of bytes that can be transmitted for that specific traffic class. The new traffic flow is admitted if and only if such a sum is lower than the $MaxByte_N$ of the traffic class N.

Conclusions Thanks to the combination of offset- based scheduling and the temporal isolation provided by the different transmission

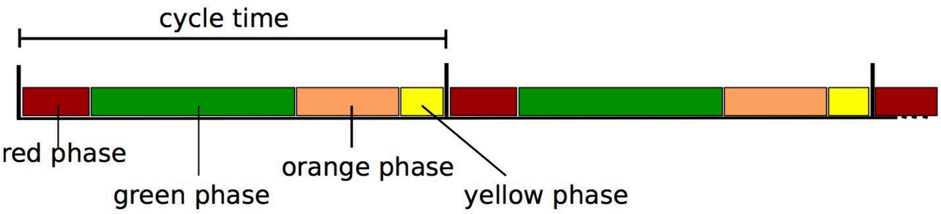

Figure 9.1: *Cycle time and phases in the AVB_P approach approach*

phases, the ST flows is able to obtain low and deterministic latency values. Thanks to the careful sizing of the different phases, this result for ST flows can be achieved without significantly affecting the SR traffic, sometimes even improving its performance as compared with the one obtained with standard AVB.

CHAPTER

TEN

CONCLUSIONS

This dissertation presented a deep study of the IEEE Audio Video Bridging (AVB) protocol and proposed novel solutions that improve the IEEE Audio Video Bridging (AVB) standard in order to provide support to high-priority scheduled traffic.

Chapter 4 explored the AVB capabilities in a realistic automotive scenario. The study is made through a simulative assessment and the results confirmed AVB as a promising candidate for in-car communication.

Then two comparative assessments, i.e., between AVB and Time-Triggered Ethernet and between AVB and Media Oriented System Transport, are presented in Chapter 5 and Chapter 6 respectively. The results of simulations in Chapter 5, in terms of latency and jitter, proved that the best results, for both AVB and TTE, are obtained using the single star topology and both, AVB and TTE, are able to fullfill the requirements of ADAS applications.

The results in Chapter 6 demonstrate that AVB performs better than MOST for camera-based ADAS traffic, as the maximum latency values experienced by the video frames sent by the cameras are significantly lower in AVB than in MOST.

Chapter 8 presented AVB_ST, an extension of AVB able to provide support to time-driven real-time traffic on AVB networks. AVB_ST proved to be very beneficial and, thanks to the offset-based scheduling, the temporal isolation provided to the ST traffic class through the TABS mechanism, and strict priority handling, ST traffic obtained low and predictable latency values, without significantly affecting SR traffic.

In Chapter 9 another novel solution for providing support to scheduled traffic on AVB networks, for both automotive and industrial applications, was presented: AVB_P. AVB_P exploit a phase-based schedule thus simplifying the procedure for allowing temporal isolation for the scheduled traffic. It does not require TABSs, but a simple counter and makes easier the management of the ST transmissions merging all the ST_Windows in a single large interval.

Given these conclusions, future work will deal with further refinements of both the approaches here presented. Moreover, the AVB_P performance will be investigated through a comprehensive evaluation in different scenarios and under varying workloads. Following the demands of automotive industry, AVB should be able to deal with message preemption and for this reason, solutions about packet segmentation will be investigated. Moreover, in order to make AVB be a backbone for hybrid, more complex, networks for industrial scenarios, it would be important to investigate wireless-like solutions of AVB, able to cooperate with wireless and wired networks at the same time.

BIBLIOGRAPHY

[1] R. Bruckmeier, "Ethernet for automotive applications," in *Freescale Technology Forum*, Orlando, Jun. 2010.

[2] A. Chini and M. Tazebay, "802.3bp rtpge mode conversion measurements for automotive link segments," in *IEEE 802.3bp RTPGE Task Force Meeting*, August 2012.

[3] L. Lo Bello, "Novel trends in automotive networks: A perspective on ethernet and the ieee audio video bridging," in *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*, pp. 1–8, 16-19 Sept 2014, Barcelona, Spain.

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

[5] IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005), *IEEE Standard for Local and Metropolitan Area Networks*

- *Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams*, Jan 2009.

[6] IEEE Std 802.1AS-2011, *IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks*, March 2011.

[7] I. Lab., "Automotive ethernet: An overview," tech. rep., White Paper, May 2014.

[8] *Road vehicles – Local Interconnect Network (LIN) – Part 7: Electrical Physical Layer (EPL) conformance test specification.*

[9] T. Nolte, H. Hansson, and L. Lo Bello, "Automotive communications-past, current and future," in *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on*, vol. 1, pp. 8 pp.–992, Sept Catania, Italy, 19-22 Sept. 2005.

[10] *Road vehicles – Low-speed serial data communication – Part 2: Low-speed controller area network (CAN).*

[11] TTA-Group, *Time Triggered Ethernet-Spezification*, 2010.

[12] IEEE Standard 802.3, *Carrier Sense Multiple Access with Collision Detect on (CSMA/CD) Access Method and Physical Layer Specifications.*, 2000.

[13] K. Steinhammer, P. Grillinger, A. Ademaj, and H. Kopetz, "A time-triggered ethernet (tte) switch," in *Design, Automation and*

Test in Europe, 2006. DATE '06. Proceedings, vol. 1, pp. 1–6, March 2006.

[14] Rev. 2.4, OASIS Silicon Systems,, *MOST Media Oriented System Transport*, 2005.

[15] A. Grzemba, *The Automotive Multimedia Network, from MOST25 to MOST150,*, franzis ed., 2011.

[16] J. Noebauer and H. Zinner, "Payload efficiency and network considerations of most and ethernet," *Special issue MOST - Elektronik automotive*, vol. 1, pp. 14–17, Mar 2011.

[17] IEEE 802.1ASbt-2011 - Draft, *Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks*, Sept 2011.

[18] IEEE 802.1Qca–2012 - Draft, *Standard for Local and Metropolitan Area Networks-Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Path Control and Reservation*, Dec 2012.

[19] IEEE 802.1CB-2013 - Draft, *Standard for Local and Metropolitan Area Networks- Frame Replication and Elimination for Reliability*, Mar 2013.

[20] IEEE 802.1Qcc–2013 - Draft, *Standard for Local and Metropolitan Area Networks-Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements*, Oct 2013.

[21] IEEE 802.1Qbv - Draft, *Standard for Local and Metropolitan Area Networks- Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Enhancements for Scheduled Traffic*, Oct 2014.

[22] D. Pannel, "Overview of ieee 802.1qbv time aware shaping," in *Joint IEEE-SA and ITU Workshop on Ethernet*, Geneva, Switzerland, 2013.

[23] IEEE Std 1588-2008, *Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, 2008.

[24] R. Scheiterer, C. Na, D. Obradovic, and G. Steindl, "Synchronization performance of the precision time protocol in industrial automation networks," *Instrumentation and Measurement, IEEE Transactions on*, vol. 58, pp. 1849–1857, June 2009.

[25] H.-T. Lim, D. Herrscher, L. Volker, and M. Waltl, "Ieee 802.1as time synchronization in a switched ethernet based in-car network," in *Vehicular Networking Conference (VNC), 2011 IEEE*, pp. 147–154, Amsterdam, Netherlands, 14-16 Nov. 2011.

[26] H. Zinner, J. Noebauer, J. Seitz, and T. Waas, "A comparison of time synchronization in avb and flexray in-vehicle networks," in *Intelligent Solutions in Embedded Systems (WISES), 2011 Proceedings of the Ninth Workshop on*, pp. 67–72, July 2011.

[27] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert, "Trends in automotive communication systems," *Proceedings of the IEEE*, pp. 1204–1223, June 2005.

[28] B. Sostawa, "Seamless integration of video camera systems in adas," *Elektronik automotive special issue - MOST*, vol. 5, no. 1, pp. 28–30, May 2014.

[29] J. Diemer, J. Rox, R. Ernst, F. Chen, K. Kremer, and K. Richter, "Exploring the worst-case timing of ethernet avb for industrial applications," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, pp. 3182–3187, 25-28 Oct. 2012, Montreal, Canada.

[30] A. Shrikant, "Full ethernet avb stack for adas and infotainment endpoints," in *Excelfore, 4th Ethernet Technology Day,*, 23-24 Oct. 2014, Detroit, MI.

[31] S. Singer, "Solutions, highlights and challenges of ethernet avb solutions," in *FreeScale Technology Forum*, May 2014.

[32] R. Bruckmeier, "Ethernet for automotive applications," in *FreeScale Technology Forum*, Orlando, June 2010.

[33] M. Johas Teener, A. 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, pp. 2339–2354, Nov 2013.

[34] C. Hammerschmidt, "Ethernet to gain ground in automotive applications," in *Bosch predicts. EETimes Europe.*, Feb. 2011.

[35] J. Noebauer, "Is ethernet the rising star for in-vehicle networks?," in *Invited Talk at the 16th IEEE Conference on Emerging Technologies and Factory Automation (ETFA),*, France, Sept. 2011.

[36] L. Lo Bello, "The case for ethernet in automotive communications," *SIGBED Rev.*, vol. 8, pp. 7–15, Dec. 2011.

[37] G. Alderisi, G. Iannizzotto, and L. Lo Bello, "Towards ieee 802.1 ethernet avb for advanced driver assistance systems: A preliminary assessment," in *Emerging Technologies Factory Automation (ETFA), 2012 IEEE 17th Conference on*, pp. 1–4, Sept Krakov, Poland, 17-21 Sept. 2012.

[38] G. Alderisi, A. Caltabiano, G. Vasta, G. Iannizzotto, T. Steinbach, and L. Lo Bello, "Simulative assessments of ieee 802.1 ethernet avb and time-triggered ethernet for advanced driver assistance systems and in-car infotainment," in *Vehicular Networking Conference (VNC), 2012 IEEE*, pp. 187–194, Seoul, South Korea, 13-16 Nov. 2012.

[39] H. Zinner, J. Noebauer, T. Gallner, J. Seitz, and T. Waas, "Application and realization of gateways between conventional automotive and ip/ethernet-based networks," in *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pp. 1–6, June San Diego, CA, 5-9 June 2011.

[40] H.-T. Lim, K. Weckemann, and D. Herrscher, "Performance study of an in-car switched ethernet network without prioritization," in *Communication Technologies for Vehicles* (T. Strang, A. Festag, A. Vinel, R. Mehmood, C. Rico Garcia, and M. Röckl, eds.), vol. 6596, pp. 165–175, Springer Berlin Heidelberg, June 2011.

[41] H.-T. Lim, L. Volker, and D. Herrscher, "Challenges in a future ip/ethernet-based in-car network for real-time applications," in *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pp. 7–12, June San Diego, CA, 5-9 June 2011.

[42] M. Rahmani, R. Steffen, K. Tappayuthpijarn, E. Steinbach, and G. Giordano, "Performance analysis of different network topologies for in-vehicle audio and video communication," in *Telecommunication Networking Workshop on QoS in Multiservice IP Networks, 2008. IT-NEWS 2008. 4th International*, pp. 179–184, Venezia, Italy, 13-15 Feb. 2008.

[43] R. Daoud, H. Amer, H. Elsayed, and Y. Sallez, "Ethernet-based car control network," in *Electrical and Computer Engineering, 2006. CCECE '06. Canadian Conference on*, pp. 1031–1034, Ontario, Canada, May 2006,.

[44] T. Steinbach, H.-T. Lim, F. Korf, T. Schmidt, D. Herrscher, and A. Wolisz, "Tomorrow's in-car interconnect? a competitive evaluation of ieee 802.1 avb and time-triggered ethernet (as6802)," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, pp. 1–5, Quebec City, Canada, 3-6 Sept. 2012.

[45] H.-T. Lim, B. Krebs, L. Volker, and P. Zahrer, "Performance evaluation of the inter-domain communication in a switched ethernet based in-car network," in *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, pp. 101–108, Bonn, Germany, 4-7 Ott. 2011.

[46] R. Cummings, K. Richter, R. Ernst, J. Diemer, and G. A., "Exploring use of ethernet for in-vehicle control applications: Afdx, ttethernet, ethercat, and avb," *SAE Int. J. Passeng. Cars - Electron. Electr. Syst.*, vol. 5, no. 1, pp. 72–88, 2012.

[47] IEEE Std 802.1QBA-2011, *IEEE Standard for Local and metropolitan area networks-Audio Video Bridging (AVB) Systems*, Sept 2011.

[48] C. Boiger, "Class a latency issues," in *IEEE 802.1 Interim Meeting*, Kaua'i, HI, USA, Jan. 2011. [Online]: http://www.ieee802.org/1/files/public/docs2011/baboiger- class- a- latency- issues-0111.pdf.

[49] N. Balbierer, T. Waas, J. Noebauer, and J. Seitz, "Energy consumption of ethernet compared to automotive bus networks," in *Intelligent Solutions in Embedded Systems (WISES), 2011 Proceedings of the Ninth Workshop on*, pp. 61–66, July 2011.

[50] A. Camek, C. Buckl, P. Correia, and A. Knoll, "An automotive side-view system based on ethernet and ip," in *Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on*, pp. 238–243, March 2012.

[51] G. Alderisi, G. Iannizzotto, G. Patti, and L. Lo Bello, "Prioritization-based bandwidth allocation for most networks," in *Emerging Technologies Factory Automation (ETFA), 2013 IEEE 18th Conference on*, pp. 1–4, Sept 2013.

[52] W. Hintermaier and E. Steinbach, "A novel real-time video data scheduling approach for driver assistance services," in *Intelli-*

gent Vehicles Symposium (IV), 2011 IEEE, pp. 1–6, June Kongresshaus Baden-Baden, Germany, 5-7 June 2011.

[53] Y. G. Lee and B. C. Song, "An intra-frame rate control algorithm for ultralow delay h.264/advanced video coding (avc)," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, pp. 747–752, May 2009.

[54] OMNeT++ Community, OMNeT++ 4.0, *[Online]. Available: http: //www.omnetpp.org*.

[55] INET Framework for OMNeT++ 4.0, *[Online]. Available: http: //www.omnetpp.org*.

[56] M. Green, ""how long does it take to stop?" methodological analysis of driver perception-brake times," *Transportation Human Factors*, vol. 2, no. 3, pp. 195–216, 2000.

[57] T. Steinbach, H. D. Kenfack, F. Korf, and T. C. Schmidt, "An extension of the omnet++ inet framework for simulating real-time ethernet with high accuracy," in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, SIMUTools '11, (ICST, Brussels, Belgium, Belgium), pp. 375–382, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011.

[58] T. Steinbach, H. Dieumo Kenfack, F. Korf, and T. C. Schmidt, "An Extension of the OMNeT++ INET Framework for Simulating Real-time Ethernet with High Accuracy," in *Proceedings of the 4th International ICST Conference on Simulation Tools and*

Techniques, (ICST, Brussels, Belgium, Belgium), pp. 375–382, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011.

[59] J. Diemer, J. Rox, and R. Ernst, "Modelling of ethernet avb networks for worst-case timing analysis," in *7th Conference on Mathematical Modelling*, Vienna, Austria, Feb. 2012.

[60] D. Pannell, "Avb - generation 2 latency improvement options," in *802.1 TSN face to face meeting*, Atlanta, March 2011.

[61] J. Imtiaz, J. Jasperneite, and S. Schriegel., "A proposal to integrate process data communication to ieee802.1 audio video bridging (avb)," in *In proc. of the 16th IEEE International Conf. on Emerging Techonologies and Factory Automation (ETFA)*, Toulouse, France, Sep 2011.

[62] J. Imtiaz, J. Jasperneite, and L. Han, "A performance study of ethernet audio video bridging (avb) for industrial real-time communication," in *Emerging Technologies Factory Automation, 2009. ETFA 2009. IEEE Conference on*, pp. 1–8, Sept 2009.

[63] J. Jasperneite, J. Imtiaz, M. Schumacher, and K. Weber, "A proposal for a generic real-time ethernet system," *Industrial Informatics, IEEE Transactions on*, vol. 5, pp. 75–85, May 2009.

[64] J. Imtiaz, J. Jasperneite, and K. Weber, "Approaches to reduce the latency for high priority traffic in ieee 802.1 avb networks," in *Factory Communication Systems (WFCS), 2012 9th IEEE International Workshop on*, pp. 161–164, May 2012.

[65] E. Heidinger, F. Geyer, S. Schneele, and M. Paulitsch, “A performance study of audio video bridging in aeronautic ethernet networks,” in *Industrial Embedded Systems (SIES), 2012 7th IEEE International Symposium on*, pp. 67–75, June 2012.

[66] G. Alderisi, G. Patti, and L. Lo Bello, “Introducing support for scheduled traffic over ieee audio video bridging networks,” in *Emerging Technologies Factory Automation (ETFA), 2013 IEEE 18th Conference on*, pp. 1–9, Sept 2013.

[67] K. Tindell, “Adding time-offsets to schedulability analysis,” in *Rep. YCS221, Dept. of Computer Science*, York University 1994.

[68] L. Lo Bello, A. Raucea, G. Patti, and O. Mirabella, “L-ptp: A novel clock synchronization protocol for powerline networks,” in *Emerging Technologies Factory Automation (ETFA), 2012 IEEE 17th Conference on*, pp. 1–4, Sept 2012.

[69] K. Tindell, “Adding time-offsets to schedulability analysis,” tech. rep., Dept. of Computer Science,York University, 1994.