5 Protocols for Automotive Ethernet

One of the reasons for the automotive industry to adopt Ethernet-based communication as an in-vehicle networking system is the chance for synergies, i.e., the possibility of reusing protocols that have been developed and tested in other industries. Across the various protocol layers for the various applications it therefore needs to be carefully investigated whether to adopt, adapt, or to add protocols. Figure 5.1 gives an example overview of a typical protocol stack. This chapter discusses four areas that require special care: Audio Video Bridging (AVB) and its successor Time-Sensitive Networking (TSN, see Section 5.1), Virtual LANs (VLANs) and switch configuration in the context of security (see Section 5.2), the Internet Protocol (IP; see Section 5.3); and what is needed in terms of command and control (see Section 5.4).

Note that the described solutions make no claim to be complete; it might well be possible to use other protocols with the Automotive Ethernet PHY transceivers. However, the solutions described in this section describe a solution that works and that can be adopted by those wanting to deploy Automotive Ethernet.

5.1 Quality of Service (QoS), Audio Video Bridging (AVB), and Time-Sensitive Networking (TSN)

Ethernet as such, i.e., the PHY and MAC layers as defined by IEEE 802.3 at the time, provide best-effort communication only. Introducing switches improved the determinism of each individual link, since the various connected units no longer needed to contend for the same medium at potentially the same time and in case of collisions had to go into random, i.e., nondeterministic, back-off periods. However, in a switched network, data of different sources with different destinations might still have to be sent over the same link at the same time. It is therefore on Layer 2 in the switch – often also referred to as a (multiport) bridge[1] – that Quality of Service (QoS) requirements can effectively be supported. Today, it is mainly at IEEE 802.1 that the respective protocols and procedures are being standardized.

This book uses the term QoS for requirements and solutions that influence the flow of data such that it can be received at a defined quality [2]. These can vary significantly depending on the use case and focus area (see, e.g., [3] [4]). It is therefore important to start with some background information on the origin of the Audio Video Bridging

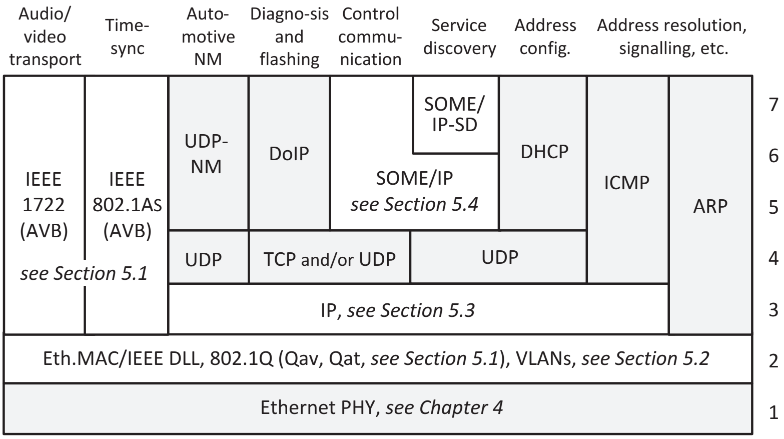

Figure 5.1 Protocol overview for Automotive Ethernet [1].

(AVB) standardization activity (Section 5.1.1) and to highlight the differences between the originally envisioned audio and video use cases and their deployment in automotive (Section 5.1.2). In return, this allows describing how each QoS protocol for audio and video applications can best be used in in-vehicle networking (Section 5.1.3). However, even if audio video entertainment provides the origin for the series of IEEE standards, this is not conclusive. Section 5.1.4 describes efforts around standardizing protocols for more safety critical applications in an Ethernet network. These efforts are called Time-Sensitive Networking (TSN).

5.1.1 How Audio Video Bridging (AVB) Came to Ethernet

In July 2004, the IEEE 802.3 group accepted a Call For Interest (CFI) on “Residential Ethernet” in order to investigate the use of Ethernet for time-sensitive Audio and Video (AV) applications [1]. Apparently, more than a year previously, discussions on the need for more Consumer Electronics (CE) centric Ethernet networking had started simultaneously in different groups of industry players, who then aligned the standardization in the IEEE [5] [6].

At the time, the Internet in combination with audio – and later video – compression formats was drastically and irreversibly changing the consumer behavior in respect to music consumption. Even if the “share it with all for free” Napster platform had only lived from May 1999 to February 2001 [7],[2] it initiated a change: The PC/laptop/mobile device replaced the home hi-fi and CD collection as the center for consumer entertainment. These new devices are able to serve at the same time as storage, rendering device, synthesizer, sound mixer, and media server, and the PCs and laptops – at least up till now [8] – always had an Ethernet interface. The CFI on Residential Ethernet addressed the specific quality requirements of AV transmission in an Ethernet LAN, and thus broadened the market potential of Ethernet into the consumer space.

Next to being widely deployed in PCs and laptops, Ethernet offered [5] plug & play, large data rates, and network management in terms of neighbor discovery, virtual network support, and traffic prioritization. Nevertheless, even with priority, there was/is neither timing guarantee nor a reference time to which the receiver can relate. Buffering data can help to overcome jitter up to a certain point. AV applications like VoIP or IP-TV rely on buffering in order to improve the quality. However, finding the correct buffering size in such a situation is no easy task: Buffers that are too small bear the risk of buffer overflow, dropped packets, and quality degradation, while buffers that are too large are costly and introduce additional latency. So, Ethernet did not support endpoint synchronization, timing support, bounded latency support, and bandwidth allocation [5]. The goal of the IEEE 802.3 effort was thus to develop mechanisms for better supporting AV applications in an Ethernet network by providing the appropriate mechanisms.

Very quickly after the respective study group had been set up by IEEE 802.3, it became apparent that the proposed solutions were better suited for standardization in IEEE 802.1 [9], and by the end of November 2005 the effort was officially moved [10]. In September 2011 (plus a latecomer in August 2013), the following set of standards associated with first-generation Audio Video Bridging (AVB, AVBgen1) were completed (see also Section 5.1.3):

- IEEE 802.1Qav, “Forwarding and Queuing Enhancements for Time-Sensitive Streams” (traffic shaping), 5 January 2010[3]
- IEEE 802.1Qat, “Stream Reservation Protocol,” 30 September 2010
- IEEE 802.1AS, “Timing and Synchronization for Time-Sensitive Applications,” 30 March 2011
- IEEE 1733, “Protocol for Time-Sensitive Applications in Local Area Networks” (AVB adaptation of RTP), 25 April 2011
- IEEE 1722, “Transport Protocol for Time-Sensitive Applications in a Bridged Local Area Network” (layer 2 transport protocol), 4 March 2016
- IEEE 802.1BA, “Audio Video Bridging (AVB) System” (overall system configuration, profiles), 30 September 2011
- IEEE 1722.1, “Device Discovery, Connection Management, and Control Protocol for 1722^{TM} Based Devices” (control mechanisms and service discovery), 23 August 2013

These standards leave the implementer with a variety of options. In 2008, a number of companies participating in the AVB standardization at IEEE thus started working on the formation of an industry alliance that would market AVB and aid interoperability [11]. The AVnu Alliance that was launched in consequence in August 2009 [12] now offers the respective certification programs. Concerning the proliferation of AVB in the market, AVB enabled silicon started to emerge in 2012 (see, e.g., [13] [14]), which from an automotive perspective was just right. With finishing the AVB standards listed above, the IEEE started directly with standardizing AVGgen2 in order to be able to support more time-critical applications (see also Section 5.1.4). As these do not longer comprise audio and video applications only, the AVB effort was officially renamed Time-Sensitive Networking (TSN)in November 2012 [15].

5.1.2 The Audio Video Bridging (AVB) Use Cases

Before going into detail of specific applications, some general remarks on fundamental differences between the quality requirements of AV consumption (including speech) and those traditional Ethernet was designed for:

- While most data applications require every single packet to arrive intact, the occasional **packet loss** can go unnoticed by the user in the case of AV transmissions. That not all information is needed, is emphasized as such by the fact that many AV compression formats are “lossy,” meaning that not all information can be recovered with the decompression. MP3 and MPEG are well-known examples. Even though it is in general more critical to lose a packet of compressed data than a packet of uncompressed data, an additional and occasional loss of a packet containing compressed data is not necessarily perceived as a quality degradation [16].
- The situation reverses in the case of **delays**. In general, delays in the milliseconds range or even of a few seconds not discussed when it concerns a file transfer or building up a website. AV applications, in contrast, have very stringent timing requirements [17]:
  - The **absolute delay** must be small in the case of live streaming data. A musician, e.g., tolerates only a 10 ms delay between initiating a sound and expecting to hear it [18] [19].
  - The data must be **synchronized**. In a home (or concert hall) sound and image might travel different paths with different delays. For quality replay, the delay between sound and picture replay needs to be smaller than ±80 ms (“lipsynch” requirement) [20]. Standard home stereo sound needs synchronization between the different streams of less than ±1 ms [18]. For high-end surround sound the synchronization requirement reduces to less than ±10 μs [20], or ±1 μs in the professional environment [9].
  - No matter where the AV content is stored or replayed, there should be **no noticeable jitter**. Sudden interruptions or delay variations in AV streams can occur in case competing traffic is taking up too much data rate. Buffers can only partially compensate for this. For some applications the absolute delay is simply limited and in general the larger the buffer the higher the costs.

5.1.2.1 In Homes/Consumer Devices

Since the introduction of audio and video (AV) compression formats, AV streams are turned into strings of data packets that can be stored on and replayed from various consumer devices like PCs, laptops, tablets, phones, memory-sticks, portable music players, etc. While the traditional home entertainment consisted of units with a clear media-to-function relation (record player, tape recorder, CD-player, amplifier, etc.) and one-way, analog communication between them, the transformation of entertainment data into packets allows for/requires bidirectional networking between units, which Ethernet inherently supports. The general observations made above apply to all AV applications.

Additional requirements, e.g., on timing in consumer devices come from gaming applications, which require a response time of less than 50 ms for human activity and less than ±80 ms difference between video animation and audio. Other home related AV applications with again different requirements are home surveillance and health care [5].

In contrast to the professional audio and automotive use cases, requirements that prevail in consumer applications are (a) the needed support of ad hoc/plug & play capability (no IT administrator) and (b) the necessity of lowest cost [5]. In consequence, aspects like discovery (which is addressed on higher layers by UPnP[4] and DLNA[5]), self-configuration, and a high level of compatibility and interoperability are very important [11]. Furthermore, in a home, media might be shared over a variety of networks that include IEEE 802.11 WLAN/WiFi and Coordinated Shared Networks[6] (CSN) [21], as well as Ethernet.

5.1.2.2 In Professional Audio

Typical application areas for networked professional audio equipment are concerts/live shows, recording studios, conference centers, theme parks, houses of worship, art installations, or any other place where live sound is used professionally [17] [22]. This emphasizes one of the fundamental differences to the consumer use case: In professional audio, good quality perception is a core purpose. A network deployed for professional audio has to be absolutely reliable, with no audio defects, video dropouts, or other artifacts [17]. Furthermore, the timing requirements are very stringent: As has been said, for musicians the delay, e.g., between the microphone and the earphone of an artist needs to be smaller than 10 ms [18] [19]. Allowing 8 ms for processing means that the network delay cannot exceed 2 ms [23]. Professional audio also has very stringent requirements on speaker synchronization, which needs to be within a few microseconds [9].

As with all industrial products the use of a new technology/concept needs to provide for direct or indirect cost savings (or for new functionalities that are expected to result in monetary advantages, see also Figure 3.9. The starting point of professional audio networks is a setup that comprises a huge amount of high-quality, single-purpose, unidirectional, analog, or even digital audio cabling using different technologies. Furthermore, the same extensive wiring is used for the respective video infrastructure, and yet another lot of cabling for control (of amplifiers and loudspeakers), which might use an Ethernet infrastructure [9] [17] [22] [24] [25]. This is not only expensive in respect to the wiring, but also difficult to maintain, and invites the development of proprietary solutions on higher layers, which seem to have prevailed for a long time [5]. Being able to handle such a setup requires very specialized know-how [17].

Thus, the attraction to be able to use a single network, i.e., the Ethernet infrastructure, for all data that need to be networked in the professional AV applications, is obviously large. In pre-AVB times, this was too cumbersome [9] [17]. So when AVB activities started, it is only natural that these were supported by professional audio companies from the start [18], in the IEEE [5] as well as when establishing the AVnu Alliance [11].

An important difference to the residential/consumer and automotive uses of AVB is that the extent of the professional audio network can be significantly larger, in meters as well as in number of nodes. However, in contrast to the consumer use case, the professional AVB network can be expected to be professionally set up and controlled.

5.1.2.3 In Cars

Ever since Ethernet started being discussed for automotive use, the Quality of Service (QoS) capabilities of Ethernet and the potential of the AVB solutions have been investigated (see also Section 3.2). With Ethernet coming from the IT and CE industries, Ethernet was first considered for "similar" in-vehicle infotainment applications. So, while today, Ethernet is naturally being discussed also for in-vehicle control applications (see Section 5.1.4) the focus at the beginning was on enabling AV applications and by the time the AVnu Alliance was set up in 2009, automotive applications were identified as one of the target areas for AVB [26].[7]

In-vehicle AV consumption was not one of the original use cases addressed with the standardization of AVB at IEEE. Nevertheless, Infotainment is an important quality element for vehicle users; after all, the stringent automotive Electromagnetic Compatibility (EMC) requirements (see also Section 4.1) have also been installed to ensure unblemished audio consumption while driving. Nevertheless, in the hierarchy of applications inside vehicles, infotainment is always secondary in relation to driving and safety. This is the most important difference to the consumer and professional audio use cases discussed before and its consequences impact the in-vehicle use of AVB (see the following sections). Furthermore, automotive has an additional timing constraint: The AV system needs to be fully operational within two seconds of power on [27]. Neither in the consumer domain, nor in professional audio does such a (stringent) start-up requirement exist. As in the professional audio domain, the in-vehicle AV network is professionally set up beforehand, even if various car models exist and the exact layout additionally depends on the options the customer selects.

Naturally, also the automotive audio use case itself differs from the ones that can be expected in homes or even the professional environment. The high-quality expectations from car customers and the complexity of handling the various different audio sources in vehicles had once even led to the development of a new in-vehicle networking technology (MOST, see Section 2.2.4.1). An example of audio use cases and their hierarchy inside the vehicle is presented in Table 5.1. As can be seen, a significant amount of the complexity is not handled at the network interface but is organized on higher layers. These can be based on the same principles that MOST handles the functionalities or use a GENIVI-based implementation (see Section 3.5.3), which in return also supports MOST. From an automotive perspective, it is important to keep the separation between application specific requirements and QoS functions the network can provide based on AVB. The separation of the ISO/OSI layering model should be maintained.

Furthermore, costs and resources distinguish the use cases. The AVB functionalities require hardware capabilities in the Ethernet semiconductors and processing power from a separated microcontroller (μC) or from the switch. In professional audio it can be assumed that all processing resources needed will be provided. After all, audio function and quality is their prime concern. In the CE industry costs, in principle, need to be low, though the resources available and customer expectations are likely to vary significantly,

Table 5.1 Example audio hierarchy in an automotive audio network

Layer	Functional block	Features and functions
High	Human–Machine Interface (HMI)	Customer interface for volume control, source changes, additional control interfaces (e.g., changes of volumes for audio interrupts sources like jingles and alarms).
Mid	Audio management system	Fixed system behavior: controls the mixing stages in the audio sink by special control commands. An example is the audio output in case a navigation audio guidance message or park control beep occurs at the same time as the driver is listening to the radio or making a phone call via the in-built hands-free system. The solutions here are generally car manufacturer specific.
Low	Audio network interface	Network resource management: responsible for the availability of the requested bandwidth. The source needs to know when to allocate bandwidth and the sink knows when and how to connect to the source data.

depending on the monetary value spent on the CE device. However, a laptop or even a tablet will have much greater resources available than a typical ECU inside a car that is optimized for cost, space, and processing power.

While costs are important, the question of where and how to implement the AVB functions in an automotive network has more facets. If the AVB functions are embedded on a microcontroller that is integrated into the switch, which supplier will provide the software for it, the Tier 1 or the semiconductor supplier? If the AVB software comes from the semiconductor supplier, but the Tier 1 is responsible for the function of the ECU, who, if there is a malfunction, can diagnose it? Who is responsible? In case resources of the ECU's main purpose μC are used, how can it be ensured that the network functionality is never impaired by other ECU functions, especially during start-up or reboots? As a first approach, [28] proposed to use only an absolute minimum of AVB features for the automotive networks and with this initiated an important discussion on automotive AVB. But the result was nonstandard compliant, and was therefore developed further [27]. As a guideline, it is helpful to store as much (initial) configuration data as possible in some form of digital memory, in order to achieve independence between the ECU and networking functions, especially during start-up. For details, see Section 5.1.3.

5.1.2.4 Direct Comparison of Use Areas

The description in the previous sections showed that the three use cases have very different requirements. What they have in common is that all would like to realize high-quality AV streaming in a (mainly) Ethernet-based network. Additionally, the use and the network are restricted to a certain purpose, size, and physical location, even if a concert hall network can have significantly larger dimensions than a LAN inside a family home or an "Automotive Area Network" (AAN) inside a car. All three use cases can live with

Table 5.2 Comparison of the requirements and properties of the different AVB application areas

Criteria	Home/consumer devices	Professional audio	Car AV
AV application scenarios	Multiple source/sink AV replay in the home, home surveillance [5]	Recording studios, concerts/live shows, conference centers, theme parks, houses of worship, art installations [17] [22]	Simultaneous audio streams of different priority, synchronous replay of AV, camera data for driver assistance
Importance of AV quality	Expectations are likely to correlate with the price paid for the equipment	**The core purpose**	Entertainment and comfort are important under normal circumstances but **driving (safety) is requirement No. 1**
Variability and planability of network setup	Ad hoc, **plug and play**, no IT admin [5], requires self-configuration, service discovery, etc. [21]	Setup can change but will be carefully planned from event to event	Known number of predefinable variations per car model (limited plug and play from passengers)
Network technologies used	Ethernet, WiFi/WLAN, Coordinated Shared Networks (CSNs) [21]	Ethernet	Ethernet
Service rejection (not enough data rate)	Acceptable	Not acceptable	Not acceptable
Synchroniza-tion accuracy required	Stereo synch ∼ 1 ms/10 µs [18] [20]	∼**1 µs** [9]	Like for consumer devices
Maximum network delays	50 ms [19]	2 ms application delay [23]	80 ms lipsynch [19]
Start-up requirements	None	None	**AV system fully operational within 2 s** [27]
Link length	<200 m [5]	Can be long, but <1 km expected	≤15 m [29], 3.5 m average [30]
Available processing resources	Depends, larger on computers, smaller on mobile devices	As large as needed	Generally shared with other functions, rather small
Costs	Very low costs [5]	Function before costs, savings in harness	Needs to be competitive (see Section 3.4.2)

the maximum delay guaranteed for 100 Mbps Ethernet AVB traffic of 2 ms over 7 hops on ISO/OSI layer 2.[8]

Table 5.2 directly compares the main properties and requirements of the AVB use cases. The main difference is the low-cost, multivendor plug & play requirement of the consumer use case, in contrast to the very high-quality requirements of the professional audio use case, or the secondary nature of AV in automotive and its stringent start-up requirement.

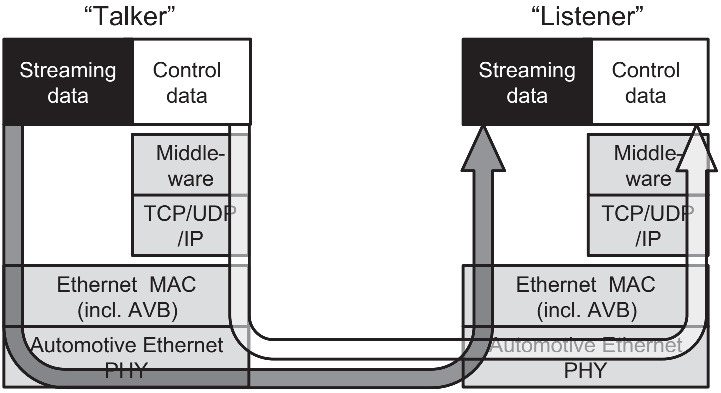

Figure 5.2 IEEE 1722 streaming data and application control data in automotive.

5.1.3 The AVB Protocols and Their Use in Automotive

The AVB specifications facilitate QoS guarantees for streaming data within an "AVB cloud," i.e., a group of networked devices all supporting the core AVB functions either in the role of forwarding switches (which in the IEEE context are referred to as bridges, see introduction to Section 5.1) and/or as end nodes. The basic QoS requirements are that the streams can be rendered in synch with each other, that network delays are not noticeable in the application, and that the network resources are available for as long as the application needs them [9]. AVB distinguishes between "Talkers" that are the source of the streaming data, "Listeners" that are the consumers of those streams, and the AVB capable switches in between. The implementation of AVB requires that the underlying Ethernet network runs at least at 100 Mbps full duplex, that the Ethernet payload does not exceed the maximum size of 1500 bytes, and that the flow control/pause frames (see Section 1.2.1) are disabled. The following subsections describe the AVB mechanisms in more detail.

5.1.3.1 IEEE 1722: Transport

The IEEE 1722 [31] specification describes the transport for AV data. It leverages concepts from the IEC 61883 standards on digital interfaces of consumer AV equipment and thus FireWire/IEEE 1394 [24]. The key property of IEEE 1722 is that it identifies Ethernet packets carrying AV content on layer 2 and not on higher layers. This allows bypassing higher layer protocols (see Figure 5.2), thus reducing the processing time needed and making the latency more predictable.

In principle, IEEE 1722 allows transporting two types of content: streaming data and data for controlling IEEE 1722. Figure 5.3 depicts the respective packet structures: an Ethernet frame/packet carrying an IEEE 1722 packet and its content, which has its own header for the packet and potentially even a header attached to every AV content unit. The Ethernet packet has to include the otherwise optional IEEE 802.1Q header. The priority information (as defined in IEEE 802.1Q and used in IEEE 802.1Qat/SRP) encapsulated within is essential for the functioning of the AVB QoS concept. Using

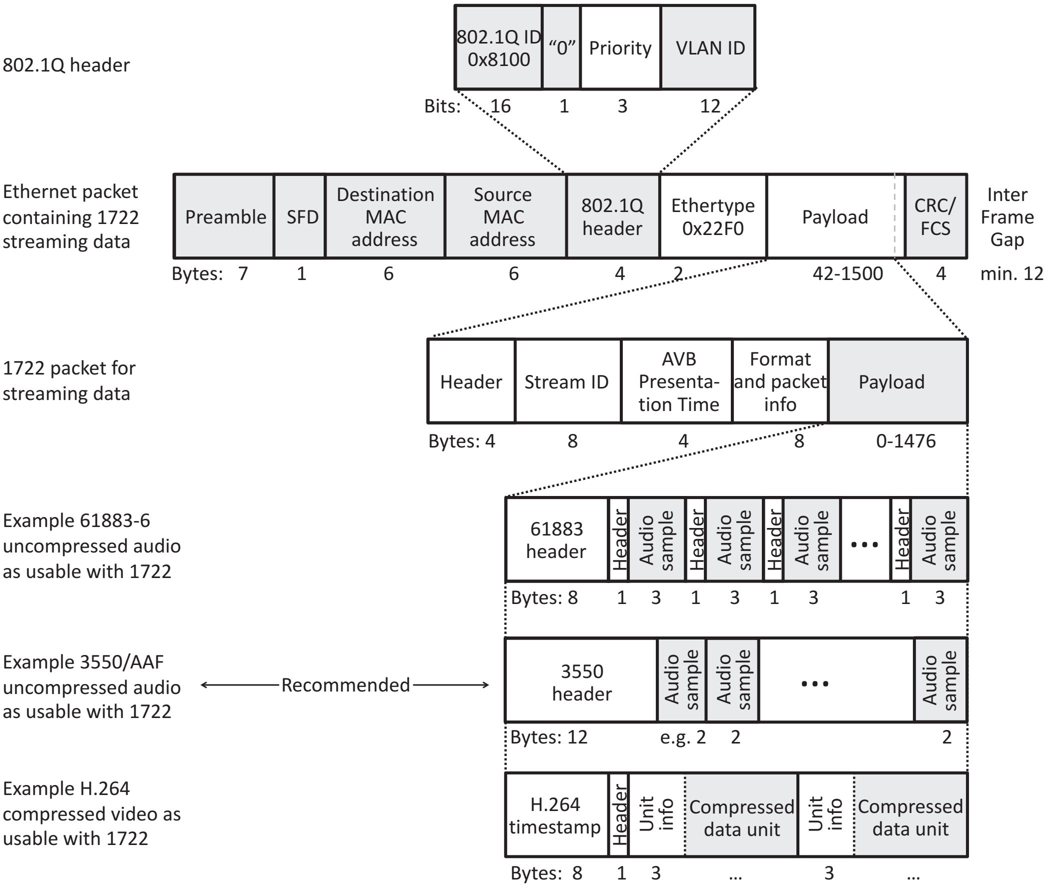

Figure 5.3 IEEE 1722 packet format with example 1722 payloads.

VLANs (see also Section 5.2.2.1) is in principle orthogonal to AVB and independent. However, to be able to receive streams Listeners must be members of the right Talker's VLAN [31]. The IEEE 1722 Ethertype is 0x22F0. If the IEEE 1722 packet is shorter than the required 42 bytes minimum length, the Ethernet MAC will pad the packet automatically, like for any other protocol transmitted via Ethernet.

An IEEE 1722 streaming packet consists of a header, the stream ID, the "Presentation Time," payload information and the payload itself (see Figure 5.3). The header defines what type/format of AV data to expect. It also includes the sequence number in order to allow Listeners to identify missing packets. The stream ID unambiguously defines a specific data stream and is derived from the Talker's MAC address (see Section 5.1.3.3). The field for payload information is directly related to the format of the data inside the payload.

A very important part of AVBTP 1722 is the "Presentation Time." It defines the time a received packet should be presented to the Listener application, i.e., when it should leave layer 2 at the receiver. The Talker sets the Presentation Time depending

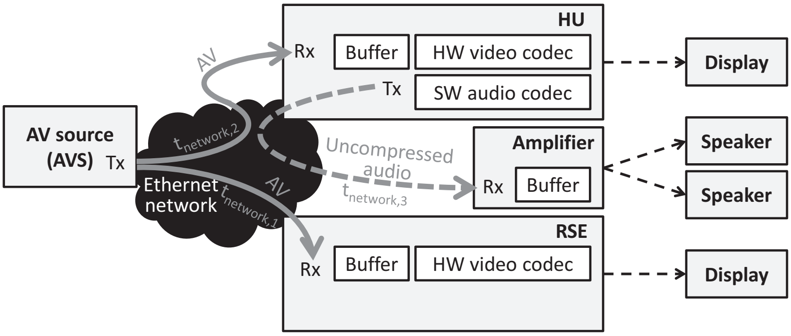

Figure 5.4 Different audio and video paths in an Automotive Area Network (AAN) that are a challenge for achieving lipsynch.

on the time the packet left the application buffer inside the Talker and the expected worst-case duration the packet needs in the cloud ("Max Transit Time"). The default value for the Max Transit Time in the case of the highest priority streaming "Class A" traffic is 2 ms; in the case of the next lower priority, "Class B" traffic, it is 50 ms. The standard allows this value to be different/negotiated, but it does not define how this should be done. Standard plug & play equipment can thus be expected to use the standardized default value(s). The Presentation Time is represented in nanoseconds (ns) as the remainder when dividing the absolute time by $2^{32} - 1$ ns. The concept of the Presentation Time is a good example of the close interrelationship between the AVB standards, as the Presentation Time can only work in a previously synchronized network (see Section 5.1.3.2 for IEEE 802.1AS). Once the synchronization has been established the Presentation Time can be used as feedback and correction for the synchronization. The Max Transit Time is also one of the values that determine the required buffer size of the Listeners (see also Section 5.1.3.3).

The IEEE 1722 provides an important mechanism also for QoS in Automotive Ethernet. In terms of its use, the following considerations are important:

1. **Supported data formats:** The IEEE 1722–2011 specification covered mainly FireWire/ISO 61883 headers[9] but not, e.g., the formats discussed in the automotive industry for the camera use cases: MJPEG and H.264 (see also ISO 17215). This was changed with the IEEE 1722–2016 release [31]. Should yet more formats be needed, the payloads defined for the Real-time Transport Protocol (RTP) in IETF RFC 3550 [32] can be used with IEEE 1722 without modification.
2. **Use of the Presentation Time:** Figure 5.4 shows an example in-vehicle network consisting of an Audio Video Source (AVS), a Head Unit (HU), a Rear Seat Entertainment (RSE), and an Amplifier. The goal is to have a lip synchronous replay of the content on two different displays and two speakers. The Presentation Time as defined in IEEE 1722 defines the time the data shall be presented to the system beyond layer 2. In the example this means the time the data are passed on to the AV codecs, and not, as would be desirable, the time of playing the data on the displays and speakers. The example scenario is even more critical, as after decoding the uncompressed

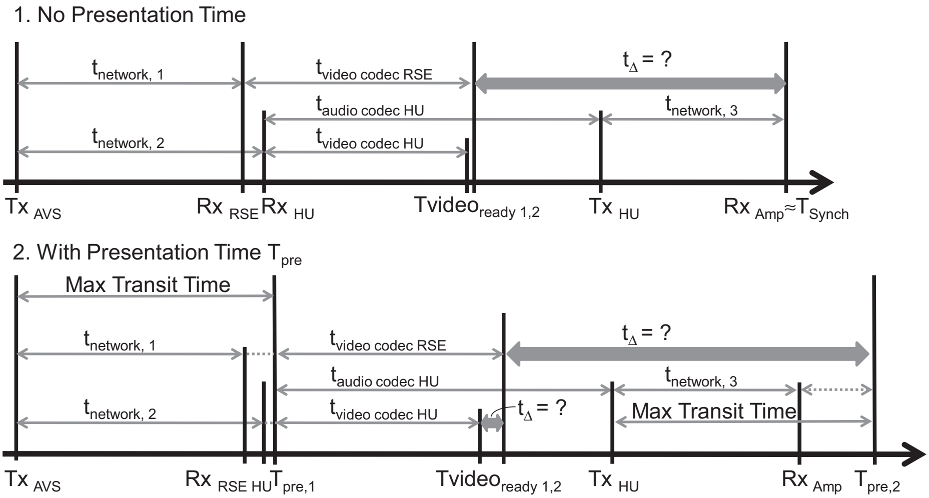

Figure 5.5 Timing behavior with and without Presentation Time T_{pre}, assuming T_{pre} is derived from the maximum delay in the network Max Transit Time.

data is reinserted into the network. This transmission is completely independent from the first and not considered in the original Presentation Time provided by the AVS. The Presentation Time as originally defined is thus not sufficient for ensuring synchronous replay in this scenario. But, even if only the HU and RSE video replay was considered, the Presentation Time would only help, if the processing delay caused by the video decoding in the two units differed only marginally.

Figure 5.5 depicts the consequences for the timing behavior. In the upper half of Figure 5.5, the Presentation Time is not used and the codecs start processing the data the moment they receive it (Rx_{RSE}, Rx_{HU}). In the example, the audio codec is realized in software and slower (without this being decisive). In the end neither the two image streams are ready at the same time ($Tv_{ready,1}$, $Tv_{ready,2}$), nor is the audio, which arrives significantly later at the speakers (Rx_{Amp}). The lower half of the picture shows the same scenario under the assumption that the maximum delay possible, the Max Transit Time, is used to derive the Presentation Time ($T_{pre,1}$, $T_{pre,2}$). As can be seen, this does not help to improve the synchronization between any of the output files.

Instead of a Presentation Time that defines when to present the data to the application beyond layer 2, it would thus be desirable to have a time available that defines when to present the AV information to the customer [27].One approach would be to set the Max Transit Time to a value other than the default values of the standard. The standard, in principle, allows for this. However, it needs to be assured that all used units support the use of a different value.[10]

3 **Dynamic versus static use:** IEEE 1722 expects either locally administered unicast addresses or the use of multicast addresses, which can be statically or dynamically allocated. In order to support the dynamic allocation of the multicast addresses the

IEEE 1722 specification defines a MAC Address Acquisition Protocol (MAAP) [31]. Within a specifically reserved range of multicast addresses the MAAP can dynamically establish which addresses can be used for a new stream, while defending the address from other uses once it has been selected. As has been described with the use cases in Section 5.1.2, the automotive scenario is not particularly dynamic. Once a car model has been designed, the number of different Automotive Ethernet network topologies for this car is limited. At the same time, start-up is critical and therefore all dynamic negotiations disadvantageous. A static preconfiguration is thus also preferred for the IEEE 1722 address allocation.

5.1.3.2 IEEE 802.1AS: Precision Time Protocol (PTP)-Based Synchronization

The main purpose of IEEE 802.1AS [33] is to synchronize all nodes in an AVB cloud to a common reference time. The standard mandates a precision of ±500 ns for two end nodes that have fewer than 7 AVB nodes in between, which means that direct neighbors have to synchronize with nanosecond precision [18] [24]. IEEE 802.1AS – also referred to as the "generalized PTP" (gPTP) – is a simplified extension of the IEEE 1588 specification [34], which had been started at the end of the 1990s and was first completed in 2002 [35] and updated in 2008 (PTPv2). The main difference between IEEE 1588 and IEEE 802.1AS is that gPTP assumes that all communication between time-aware systems is done using IEEE 802 MAC PDUs and addressing only, while IEEE 1588 supports various layer 2 and layer 3–4 communication methods.

Like in most time synchronization approaches, one node in an IEEE 802.1AS network functions as "Grandmaster," to whose clock all other clocks synchronize. Which unit needs to be the Grandmaster is not standardized. Ideally, the Grandmaster is the node with the best suited clock in the cloud. The standard consequently addresses the following two topics: (a) how to select the Grandmaster and (b) how to correctly synchronize to its time throughout the AVB network.

The Grandmaster can be pre- or autoselected with help of the Best Master Clock Algorithm (BMCA). The BMCA is a distributed algorithm: every Grandmaster capable node receiving a respective "announce" message compares the information of the current best Grandmaster with its own clock related quality values. If the eight differently rated values of its own clock yield a better result than that of the current Grandmaster announced in the message, the unit having done the comparison announces itself as the new best Grandmaster. The process is repeated until the truly best Grandmaster in the network has been found. The "announce" messages are sent cyclically and the Grandmaster can change anytime the Grandmaster selection data changes, like the actual Grandmaster leaving the network, a unit having access to a better clock,[11] or a new Grandmaster suitable unit joining.

As a side effect, the BMCA also determines the "clock spanning tree," i.e., the paths on which synchronization related messages pass through the network. This is done by labeling the all ports in the AVB network as follows:

- "Slave port" (the port on which the last message from the Grandmaster was received)
- "Master port" (ports on which the message was passed on)

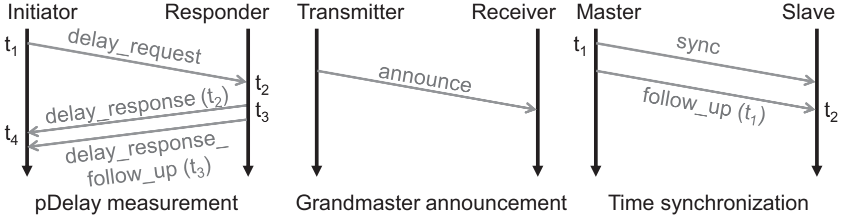

Figure 5.6 Flow charts of major IEEE 802.1AS functions.

- "Disabled port" (ports that connect to non-1AS capable nodes)
- "Passive port" (ports that lead to redundant paths in the AVB cloud).

Normally, an ECU without a switch has one port only. Unless this ECU provides the Grandmaster – then the one port is Master – such nodes have one Slave port only.

In order to achieve synchronization, every unit needs to know the delays caused by the propagation of messages in the network. IEEE 802.1AS thus defines so-called "pDelay" measurements, in which every node in the AVB cloud learns the propagation delays between itself and its direct AVB neighbors to which packets might be sent. Additionally, the pDelay measurement is also suited to determine whether a direct neighbor is actually 1AS capable. An important tool for the pDelay determination is time-stamping: The IEEE 802.1AS Ethertype (0x88F7) triggers sampling of the local clock at the ingress and egress respectively of the PTP message from the MAC [24]. To achieve the aspired nanosecond precision in the time-stamping, it is necessary to implement the time-stamping in hardware instead of software [36]. The pDelay measurements are cyclically repeated.

Last, but not least, the time to synchronize to needs to be made known in the network. This is done in two steps with "sync" and "follow_up" messages along the clock spanning tree in the network. Figure 5.6 gives an overview of the mechanisms.

IEEE 802.1AS messages use a specific multicast MAC address (01-80-C2-00-00-0E) [33]. This address enables units to exchange information between direct neighbors (only). In consequence, it is not foreseen that such IEEE 802.1AS Ethernet packets include the optional IEEE 802.1Q header, as a VLAN information would potentially collide with the purpose of the multicast address used.

For automotive use, it suggests itself to preselect the IEEE 802.1AS Grandmaster (and the clock spanning tree) and to choose as the Grandmaster an ECU every car is equipped with. It may sometimes be the case that an optional ECU would actually provide the better clock, e.g., if the option is equipped with GPS capabilities (mind though that GPS might be problematic in garages and that the clocks used, e.g., in a FlexRay node are also of high quality and well suited). Nevertheless, dynamic selection of the Grandmaster would not only unnecessarily strain the start-up time, it would also lead to more effort in the qualification and testing of the network. More variants (of, e.g., who the Grandmaster is) create more possibilities for malfunctions and robustness is essential in automotive.

The layout of in-vehicle networks is predesigned, known, and in principle does not change during use. Some links and ECUs might be disabled when not in use (see partial networking in Section 6.3.3), like a surround view system that is initialized with the reverse gear and switched off when the velocity of the car has exceeded a certain limit. Nevertheless, the link lengths and the location in the network do not change from one time the car is used to another. Thus, also the pDelay values will not change (much) every time the car is started. In order to speed up the start-up further it is thus proposed to learn and store the last pDelay values.

5.1.3.3 IEEE 802.1Qat: Stream Reservation

The IEEE 802.1Qat Stream Reservation Protocol (SRP) [37] allows allocating bandwidth for individual application and traffic streams within the AVB cloud. The main idea is that Talkers announce the availability of streaming data to all units in the AVB cloud. If Listeners would like to receive the stream, they also announce this. In consequence, all switches the data has to pass through in order to get from the Talker to the Listeners, evaluate the availability of the needed bandwidth. If available, the specific streaming bandwidth is guaranteed. If not available, the reservation request is declined. By default a maximum of 75% of the available bandwidth can be reserved, though system designers can, with care, increase or decrease this number depending on the actual requirements. If a setup allows, e.g., up to 50% of bandwidth to be reserved for Class A traffic, but actually only 30% have been reserved, Class B traffic can reserve the remaining 20% should the available bandwidth in Class B not be sufficient.

Also IEEE 802.1Qat uses special Ethertypes: 0x22EA is used for the actual reservation with the “Multiple Stream Reservation Protocol” (MSRP); 0x88F5 (MVRP) and 0x88F6 (MMRP) identify control packets for necessary information and registration associated with it.[12]

Without limitation, important information needed for the stream reservation is a unique stream ID, which is generated from the Talker’s MAC address and a 16 bit number the Talker assigns to the stream, and quality data about the stream itself. This includes the traffic class, the frame rate, and the length of every packet sent. Table 5.3 gives examples for the bandwidth needed for uncompressed stereo audio data for the defined traffic Classes A and B and a new traffic Class C[13]. This new traffic class has been generated in order to meet today’s DSPs and DMAs typical process block rate of 32 or 64 audio samples at either 44.1 kHz or 48 kHz [38]. As can be seen, transmitting audio samples with the originally defined IEC 61883–6 packet format consumes significantly more bandwidth than using the newer AVTP Audio Format (AAF). The AVnu automotive profile thus recommends the use of the AAF and does not support IEC 61883–6 [38]. Table 5.3 also shows that the higher the frequency of packets the larger the bandwidth consumed.

The use of SRP in automotive poses three main challenges:

1 Table 5.3 shows that the average number of bytes streamed in a Class A packet is very small (for the assumed simple stereo audio use case) and packets are sent at short intervals. Having significantly more overhead than payload is not only a waste

Table 5.3 Examples of the required bandwidth for uncompressed, stereo audio streams in case of different traffic classes and sample rates in an IEC61883–6 and an AAF packet format

Class	Frequency of packets (kHz)	Time between packets (ms)	Number of audio samples per packet (stereo)		Date rate for IEC 61883–6 (Mbps)		Date rate for AAF (Mbps)	
			44.1 kHz	48 kHz	44.1 kHz	48 kHz	44.1 kHz	48 kHz
A	8	0.125	12	12	7.04	7.04	5.76	5.76
B	4	0.25	23	24	4.93	5.06	3.58	3.65
C (64 samples)	0.75/0.689	1.333/1.451	128	128	3.16	3.44	1.78	1.93

Note: Traffic Class C is not part of IEEE 802.1Qat [37]. The class with 64 samples at 44.1 kHz or 48 kHz has been published in [38]. The calculation includes 16-bit sampling, 30 bytes overhead for the Ethernet packet, 24 bytes overhead for the 1722 packet, 8 bytes + 1 byte overhead per sample for IEC 61883 or 12 bytes, and none per sample overhead for AAF.

of bandwidth, it also results in an unnecessarily high processing load in all involved devices. Seen from this perspective, it is more advantageous to use longer packets; ideally they use the complete maximum MAC frame size [27]. To somewhat improve the ratio, it is thus advantageous to introduce new traffic classes, like the one identified with "C" in Table 5.3.

2 A denial of an IEEE 802.1Qat reservation request inside a car is not acceptable. It would be disastrous if, e.g., a driver assist function using camera data failed, because a rear seat passenger was watching a High-Definition (HD) video. Even rejecting an audio stream from a passenger's mobile device is critical, as it would likely be perceived as a malfunction of the car [27]. It is therefore essential that the AVB network and the expected transmission rates, including the ones from consumer devices, are carefully planned upfront and that these considerations are reflected in the network design (see also Chapter 6).

3 Last, but not least, like the dynamic selection of the Grandmaster, a dynamic reservation of streams at system start-up potentially takes too long. Inside cars, the applications using the Ethernet network as well as their data rate requirements are known; this includes the multimedia applications passengers might bring in. After all, every car only seats a certain number of passengers who each can use only a limited number of devices. It is thus in principle possible to envision a static reservation of streams. However, the information on the reserved streams needs to be provided to all nodes involved. One possible solution is to prestore data that can be accessed with every start-up. There are two ways to generate the prestored data. One is to run the SRP protocol once as part of the network setup during the manufacturing process and to store the outcome. The other is to separately design the data in the development process and store different tables for every car/option combination in every AVB ECU.

5.1.3.4 IEEE 802.1Qav: Forwarding, Queuing, and Traffic Shaping

The idea of IEEE 802.1Qav [37] [39] is to improve the actual quality of all AV transmissions in the AVB network by ensuring that the packets of each individual stream are evenly distributed over time. Even if, e.g., a stream cannot use more than the assigned 10% of the available bandwidth, it makes a significant difference for (the delays of) the rest of the streams (and the network as such), whether this stream uses all bandwidth for 1 minute and then sends nothing for 9 minutes or whether it sends something for 0.1 ms and then nothing for 0.9 ms.

To achieve this is relatively straightforward at the source of streams, the Talkers. The traffic classes assigned with the streams determine the frequency of the packets sent; every 125 μs with Class A, every 250 μs with Class B, and every 1 ms with Class C. It just needs to be assured that the Talkers do indeed send the packets at this rate. For live streams like camera or microphone data or for CD audio the data is generated continuously at the application data rate anyway, so all that needs to be done is to package the data at the assigned packet frequency. In other cases of prestored streaming data, this can be quite different. The transmitting node will likely send as much data as possible at once, with the idea that the receiver will buffer the data until it is played. This unnecessarily strains the bandwidth available in the network at the switches and causes the risk of dropped packets owing to buffer overflow. Larger buffers can mitigate this risk, but increase the costs of the receiver [9] and the latency. So, in the case of prestored streaming data, pacing the output of the Talker can make an essential difference to the AV quality in the network.

In the switches, the situation is more complex. A switch potentially needs to handle AV prioritized streams of multiple Talkers and traffic that passes through from connected non-AVB units. The issue with the latter is the following: Within an AVB cloud, the Stream Reservation (SR) traffic classes are matched to some of the eight quality values provided with IEEE 802.1p; by default Class A traffic has priority 3 and Class B traffic has priority 2. Traffic that passes through the AVB cloud from connected non-AVB units might use the same priority values. These priority values then need to be changed when the traffic enters and restored when it exits the AVB cloud.

A switch only evaluates the Ethernet header to decide through which output port(s) the data is sent and into which priority queue of the output port(s) the data need to go. All data with the same priority go into the same queue, which generally works on a First In First Out (FIFO) basis, independent from the source of the data. SR traffic has priority over non-SR traffic. Only the traffic of the SR Class queues can be paced/shaped.

The functioning of the AVB credit-based shaper is visualized in Figure 5.7. If SR traffic enters the priority queue of a port currently busy it collects credit at an “idle” rate, which is equal to the overall bandwidth currently reserved for the respective SR traffic class. As soon as the port is no longer occupied and the credit is equal to or larger than 0, the priority packet is transmitted. During transmission the credit reduces at a “send” rate, which equals the transmission rate of the link minus the idle rate. If at the end of the transmission the credit is still equal to or larger than 0 more packets from the

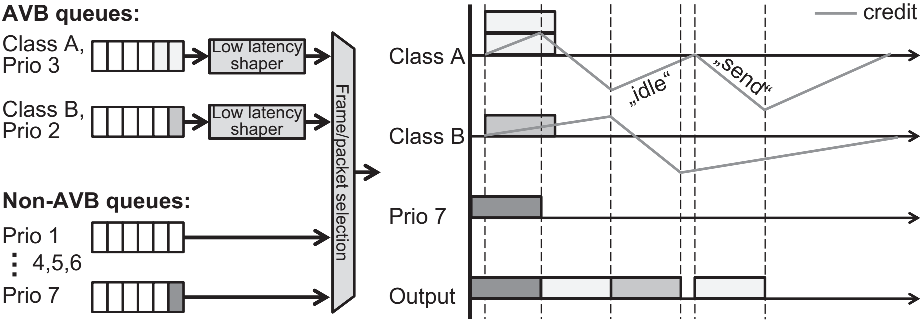

Figure 5.7 Principle of credit-based shaping and queuing with IEEE 802.1Qav.

same queue can be transmitted. If at the end of the transmission the credit is still equal to or larger than 0 and the priority queue is empty, the credit is set to 0. When the credit for a queue is negative, no packet from that queue can be transmitted until the credit has again increased at the idle rate back to 0 or higher.

The main concern for automotive with IEEE 802.1Qav is that the immediate reception of safety critical control data is more important than the AV quality. To simply use the highest priority SR Class A for control data is not a solution as the traffic shaping might actually delay the transmission of a safety critical message [40] [41]. The use of the highest priority non-AVB traffic class for critical control data would guarantee an average throughput of 20% or more (depending on how much has been reserved for the AVB queues). However, AVB-queue packets within their reserved bandwidth always pass first, and the critical control data might be delayed. There is no simple solution for this. Thus, a significant amount of effort of the actual TSN standardization addresses the requirements for safety critical control traffic (see Section 5.1.4).

Apart from the principal concern addressed above, the advantages of shaping as such have been discussed at length for automotive use cases. It is not the issue to pace the Talker output; this is a low-effort implementation. The concern is in the switches, as not every shaping algorithm is suitable for the use cases that require support. If multiple streams pass a switch, a different algorithm might be optimum for each. The solutions thus require careful design.

Figure 5.8 summarizes the AVB elements proposed in Sections 5.1.3.1 to 5.1.3.4 for use in Automotive Ethernet.

5.1.3.5 Other First-Generation AVB Protocols

The following specifications are part of AVB (gen1) but at the time of writing were of minor relevance for automotive implementations:

- The Real-time Transport Protocol (RTP) describes how to transport audio and video (AV) streams over layer 3/IP-based networks. The standard **IEEE 1733–2011** [42] describes how to map the RTP time with the IEEE 802.1AS Presentation Time [9]

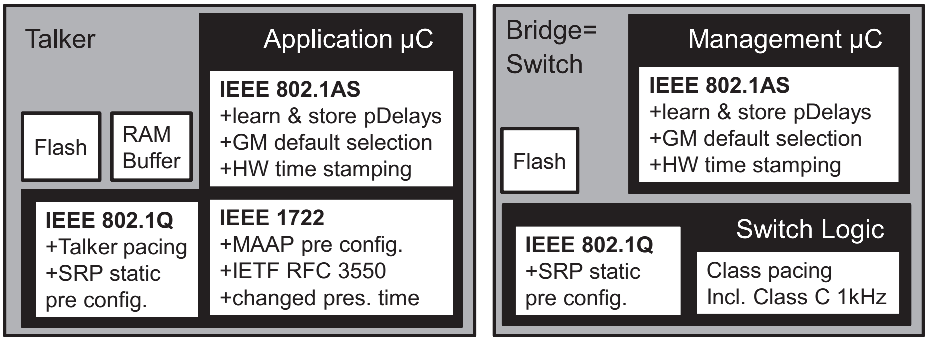

Figure 5.8 Proposed elements of an automotive AVB implementation for automotive AV ECU(s) either in the function of a Talker (Listener) or within the switch semiconductor.

and thus how to let layer 3 AV data profit from layer 2 AVB mechanisms [43]. This increases the flexibility on the technologies used within an AVB network.

- As the previous sections indicated, AVB supports various use cases and ways on how to set up the AVB network. **IEEE 802.1BA-2011** [44] defines profiles and default configurations in order to support easy handling especially where expert network knowledge cannot be expected.
- **IEEE 1722.1–2013** [45] defines typical middleware functionalities for discovering and handling devices and services for IEEE 1722-based systems. From an automotive perspective, the tasks that middleware has to fulfill are more complex than and somewhat different from what is expected when connecting AV multimedia devices. IEEE 1722.1 is therefore of less interest. Table 5.1 showed the hierarchy and different levels of coordination needed when implementing AV applications in cars. This is quite different from any consumer system. Furthermore, the middleware should cover not only AV services but be usable throughout all in-vehicle Ethernet ECUs, independent from their use case, size, operating system, etc. A functioning solution for an automotive middleware that can be used in an Ethernet-based network is thus discussed in detail in Section 5.4.

5.1.4 Time-Sensitive Networking (TSN) for Safety Critical Control Data

Quality of service for Ethernet-based AV applications in the car is important. After all a very high percentage of drivers use some form of audio entertainment while driving [46]. Additionally, new driver assist functions like automated stop and go in traffic jams increase the interest in in-vehicle video entertainment. Nevertheless, the most important requirement in a car is safety. When Ethernet is used for safety critical applications – and with the prospect of autonomous driving it is likely that the high data rates Ethernet provides will be used for such applications [47] – the respective communication needs to have higher priority than any other data in the network and the arrival of the data needs to be guaranteed and that within a specific time.[14] Note that this implicitly requires a more refined distinction between different traffic classes on layer 2: not only between AV and data but also, e.g., between (time-sensitive) control traffic and other data.

An Audio Video Bridging (AVB) Ethernet network as originally developed for AV applications of "consumer devices" and "professional audio" was/is not suitable to (additionally) accommodate the time-critical control traffic in automotive (or industrial automation networks for that matter). With the completion of the AVBgen1 standards the scope of the QoS effort within IEEE 802.1 was thus broadened from AV data only to more types of data with QoS requirements and more stringent requirements. The project was renamed "Time-Sensitive Networking" (TSN) in November 2012 in order to reflect the new scope [15].

The TSN standards cater for quite a variety of requirements and degrees to which these requirements can be fulfilled. After all, the potential use cases for TSN multiply with the inclusion of control traffic and the development of new specifications. Even so (or potentially because of it), in 2016 the automotive industry was more hesitant to start any efforts to specify a respective profile on how exactly to use TSN for safety critical applications. In contrast, for the AV use automotive efforts had started quite early and resulted in a respective specification published by AVnu [38]. The following therefore just gives an overview on the different aspects the new TSN specifications address and from which the car manufacturers can choose (see, e.g., [48] for combination options). They are clustered in focus areas as proposed by [48] [49] [50].

1 TSN needed to support more use cases and with that **more data types**. It has been mentioned in Section 5.1.3.1 that the 2016 version of the IEEE 1722 specification incorporated more AV formats common in the automotive (and potentially other) industries [31]. Moreover, the new IEEE 1722 specification supports encrypted frame formats, UDP/IP encapsulation, and the tunneling of typical automotive in-vehicle networking messages from technologies like LIN, CAN (FD), FlexRay, and MOST. It also allows to distribute additional, application-dependent clock and event information, which can be useful for some use case.

2 One major concern for time-critical applications are **small latencies**. One TSN goal thus sets the maximum latency achievable to 100 μs over 5 hops. In AVBgen1 the latency goal was 7 hops in 2 ms. Concepts like the AVBgen1 credit-based shaper support the AVBgen1 requirements, but cause too much delay for TSN.

  A very basic method for reducing delays in switches is to implement cut-through switching. Cut-through switching is a proprietary method that allows an incoming packet to be sent out before the packet has been completely received. This can, in principle, be done as soon as the destination address has been recognized. However, even with cut-through switching, any incoming packet has to wait for packets currently being sent out on the same egress port to be completed, even if the priority of the currently egress blocking packet is lower. For a 100 Mbps Ethernet channel this might cause a delay of up to about 122 μs, for a 1 Gbps Ethernet channel this might take up to about 12.2 μs, at every switch on the way.[15] The following methodologies have been specified in order to reduce delays in such situations:

- The IEEE 802.3 specification on Interspersing Express Traffic (IET)\IEEE 802.3br [51] defines how in the PHY the transmission of a long, low priority frame can be intercepted and how to intersperse high priority time-sensitive traffic instead. The required "**preemption**" methodologies on layer 2 are filed under IEEE standards numbering 802.1Qbu [52]. IET and preemption can be deployed without any higher level organization in the network, as long as both ends of a link agree (e.g., with help of the Link Layer Discovery Protocol (LLDP)).

  The minimum fragment size is 64 bytes and fragments must be reassembled to its original packet before the packet can be passed onto other links in the network. The MAC merge sublayer adds a 60 bytes CRC. This means that any fragment shorter than 64 + 60 = 124 bytes can be preempted. The worst-case delay in case of IET and preemption is thus 123 bytes, when a packet of such length was just started. Note that IET and preemption as such cannot guarantee any specific latency or delay values. They simply provide a methodology to reduce latencies for certain traffic in mixed traffic environment with long low priority frames.[16]
- **"Time-Aware Shaping" (TAS)**/IEEE 802.1Qbv restrains the original Ethernet best-effort idea yet more. It can be used in engineered networks, when time-critical information is sent at regular intervals [50]. It basically introduces a circuit-switched/TDM channel into the otherwise packet-based communication; a methodology also used in the Industrial Ethernet technology Profinet or TTEthernet [53]. With TAS, traffic with lower priority is blocked during preprogrammed, regular time windows so that the high priority control streams are not delayed by traffic with lower priorities [54]. In combination with preemption, the length of the guardband necessary before the reserved control data time window can be reduced. With TAS and cut-through switching a minimal switch latency of 1 μs can be guaranteed regardless of frame size [50].

  At the time of writing IEEE 802.1Qch was being standardized with the goal of making delays more deterministic and better determinable by emulating a cyclic transmission behavior [55].

3 For safety critical control and fail-safe operation systems, it is not only important that data arrive with small delays under normal circumstances. It is also important that they arrive at all in case of unforeseen disruptions. Two concepts are being supported by the TSN standardization efforts for this: **ingress filtering and policing** and **redundancy**, i.e., frame replication and elimination for reliability. Ingress policing prevents faulty Talkers (e.g., sensors) or switches from disrupting bandwidth and latency guarantees of other streams in network [56], when these faulty units send more data than had been assigned to them at a specific point in time. Ingress policing prevents flooding of switches at their entrance [47]. It is proposed as a fundamental mechanism to make an Ethernet network more dependable [48]. At the time of writing it was being specified in IEEE 802.1Qci [57].

For redundancy three concepts have been standardized in TSN:

- First of all, a network topology has to be designed such that **alternative paths** exist. Then central knowledge about these paths and their status has to be available. IEEE 801.1Qca then describes how to provide an alternative path in case the currently used path fails. IEEE 801.1Qca defines how to setup, modify and tear down the respective TSN streams [58].
- IEEE 802.1CB enables "**seamless redundancy**." Critical packets are duplicated and sent on alternative paths in the network. In the unit where the different copies merge back onto the same path, the duplicate arriving later is removed from the network. An added sequence number ensures that the receiver can put the packets in the right order, even if they have arrived out of order. The concept is similar to what is being done in AFDX. At the time of writing IEEE 802.1CB was being completed [59].
- Another important aspect in a TSN network is the availability of a Grandmaster clock. It therefore makes sense to ensure the availability of a Grandmaster clock also with redundancy concepts. The new IEEE 802.1AS revision provides that a single Grandmaster can transmit duplicates of its clock on alternative routes, that multiple time domains can exist, and that a redundant time master is possible [50]. In case the actual Grandmaster ceases to exist this allows for an "instant" switch over to the new Grandmaster clock. In very large networks, in which the selection of a new Grandmaster can take up to a second, this is a valuable amendment. Another element discussed for the IEEE 802.1AS extension IEEE 802.1AS-Rev is speeding up the timing and reducing the processing in a TSN network by (re)introducing a one-step clock synchronization as had originally been available in IEEE 1588 but had been omitted for IEEE 802.1AS [60].

4 Last, but not least, TSN allows for **better scalability** in the form of reduced management traffic for reservation and configuration. The respective enhanced SRP is specified in IEEE 802.1Qcc. Depending on the actual network this can further optimize the time and processing effort needed for providing QoS in an Ethernet-based network. The methodologies discussed include the use of preconfigured systems (which is opportune for Automotive Ethernet), configurable SR classes (see also Section 5.1.3.3), handling new reservations, and efficiently supporting TAS, preemption, and redundancy. At the time of writing IEEE 802.1Qcc was still being completed [61].

As can be seen, TSN offers a variety of specifications and within each specification more choices for supporting the transmission of safety critical control data within an Ethernet-based in-vehicle network. The designer of the network can choose from the TSN specifications like from a toolbox, depending on the exact requirements that need to be fulfilled [48]. The good part is, the specifications are more independent from each other than their sheer number and volume implies. Strong dependencies exist between IET and preemption (IEEE 801.1Qbu and 802.3br) and TAS requires a synchronized time (IEEE 802.1Qbv and AS). Frame preemption improves TAS (IEEE 802.1Qbu, Qbv, and 802.3.br), and redundant paths require configuration (IEEE 802.1Qca, CB).

Table 5.4 Overview of AVB and TSN specifications as provided by IEEE in respect to automotive use

	Transport	Time synch	Stream reservation	QoS	Safety (seamless redundancy)	Security
AVB (AVBgen1)	1722–2011	802.1AS-2011	802.1Qat	802.1Qav		
TSN (AVBgen2)	1722–2016	802.1AS-rev	802.1Qat 802.1Qcc 802.1Qca	802.1Qav 802.1Qbu& 802.3br (802.1Qbv) (802.1Qch) (802.1Qcv)	802.1CB 802.1Qca	802.1Qci

Note: Additional useful standards that were available prior to the AVB effort used are IEEE 1588 and 802.1Q and p.

Else, designers can make individual choices. Table 5.4 gives an overview on the AVB and TSN specification and their relation to each other.

5.2 Security and Virtual LANs (VLANs)

In the Digital Age cyber security has become a major concern. Ever since the first PC virus in the 1980s, the number of threats and attacks has continued to grow with significant (financial) impact [62]. With the increasing digitalization and increasing connectivity of cars, it is an understandable concern that also cars might become the target of hacks. Next to mere inconvenience (car cannot be used), intervention on privacy (use is monitored), and monetary losses (car was stolen, repair is needed), the potential of hacks into cars has another dimension: personal safety. A hack that succeeds in tampering with the basic driving functions like acceleration, breaking, and steering might well put lives at risk. It has been shown that the threat is real (see, e.g., [63] [64]). Attacks therefore have to be prevented as effectively as possible.

Considering the potential consequences and the deviousness of its source – after all security is needed in order to protect against the malicious criminal energy of other humans – the topic is vast (see [65] for reading recommendations). This section therefore focuses on the aspects relevant for and special to Automotive Ethernet. Section 5.2.1 structures the automotive security topic in order to be able to place security in Ethernet in the overall security context. Section 5.2.2 emphasizes on the role of switches configuration and VLANs.

5.2.1 Security in Automotive

Security in automotive first of all requires explicit consideration and an analysis of the security threats and attack surfaces on system level. A comprehensive protection

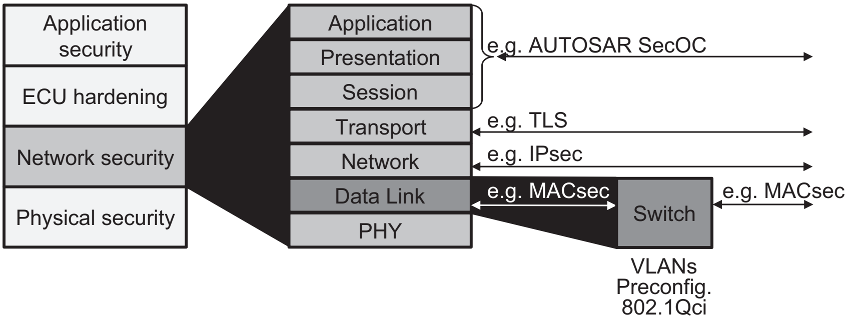

Figure 5.9 Layered automotive security approach (see, e.g., [68] [66]).

strategy is needed in order to minimize the risk of a security attack. Effective security implementations thus pursue a layered approach. The basic idea is that no implementation is perfect. Vulnerability is caused by software bugs, configuration errors, weak network design, or alike. But, if there are various layers of security and an attacker has overcome one, he still has to tackle several others. This approach is standard in the IT industry and opportune also for security in the car industry [66].

Before discussing a layered security structure in more detail, it is necessary to emphasize that (again) the circumstances in the IT industry vary from those found in the car. In IT the network is generally plug & play and unique per location. IT networks can be huge with manifold resources and frequent updates. Cars, in contrast, have a fixed topology of limited size and resources (memory, computing power, . . .). Each model is designed once and built often, with a long product life cycle and limited opportunities for (security software) updates. If one car is hacked, all cars of the same model are potential targets. Furthermore, attack patterns can change over time and become more effective. A data encryption method that is considered secure now and a car that is well protected today, might be successfully attacked in 10 years, when the same car model is still on the road. The processing power by then is likely to more easily hack longer keys that would have been too time consuming to hack today. Last but not least, an IT network is generally not restarted various times per day, and when it is restarted, it does not need to be fully operational within two seconds. A car does (see also Table 5.2). So, security methods adopted inside the car can neither use the same resources nor need as much time to complete, as IT security methods. This has to be taken into consideration when adopting security standards from the IT industry.

Figure 5.9 shows an example of what a layered security approach in automotive can look like (see left column). To start with the physical accessibility of the vehicle electronics is secured. This comprises very basic hardware measures like making it difficult to access ECUs or wiring (from the outside). Furthermore, a vehicle can be designed such that if wiring might be accessed, the communication on those wires does not connect into the vehicle network but ends at the ECU the wire is connected to. The physical security can also include architectural choices like limiting the number of ECUs with off-board communication.

The next level represents the security on the networking level, which again can be divided into layers, i.e., measures effective on each ISO/OSI layer. The depiction in Figure 5.9 includes examples of existing authentication and encryption protocols that have been developed by the IT or automotive industry and that implementers can choose from. The following gives a brief overview only as the emphasis of this section is on security measures inherent in the Ethernet technology, that have nothing to do with cryptography (see Section 5.2).

- The AUTOSAR SECure On-board Communication (**AUTOSAR SecOC)** has been developed in order to provide a resource-efficient and practical security mechanism that seamlessly integrates into the AUTOSAR communication [67] and that, being on AUTOSAR level, can be used with all networking technologies supported in AUTOSAR (CAN (FD), FlexRay, Ethernet, LIN). It provides end-to-end authentication and integrity based on message authentication codes and freshness values (counter or timestamp). For efficiency in computation and bandwidth consumption it assumes symmetric keys [68], though asymmetric keys are not precluded.
- A typical security attack on TCP level would be a TCP sequence prediction attack. Knowing the packet sequence allows to send counterfeit packets, which can potentially harm the receiver. A protocol used to prevent such attacks is the **Transport Layer Security (TLS)** protocol (previously known as Secure Sockets Layer (SSL)). This protocol ensures privacy and data integrity between two communicating applications like HTTP, IMAP, SMTP etc., by providing encryption and authentication mechanisms [69]. The protocol supports a number of different methods for encryption, cryptographic key exchange, and authentication, which are negotiated between client and server. TLS 1.2 was specified in the RFC 5246 in 2008 [70], with TLS 1.3 being work in progress in 2016 [69]. TLS is used with TCP and does not work with UDP.
- The basic concern in an IP network is that every router an IP packet passes through can read the packet and even change its content. It is also possible that one node could send a packet pretending to be another node, by using that node's address in the origination address field (called IP-spoofing). In this context the Internet Protocol SECurity (**IPsec**) was developed. Its goal is to ensure end-to-end privacy, authenticity, and integrity across the, in principle, not secure Internet. IPsec uses various mechanisms to achieve this like encryption and the addition of a header element containing a message authentication code; all directly integrated on layer 3 of the ISO/OSI layering model and thus transparent for the applications [71]. IPsec was developed in conjunction with IPv6, but can be and is used with IPv4 as well. It was standardized by the IETF in a number of RFCs (see [72] for an overview). IPsec covers more corner cases than MACsec (see below). IPsec AH can be used if authentication is of interest only and can be implemented without much effort.
- A typical attack on the MAC layer is ARP spoofing. The Address Resolution Protocol (ARP) resolves IP addresses into MAC addresses. In an attack, the attacker sends a message with the IP address of a masqueraded host, but with its own MAC address. The receiving node caches the falsified IP and MAC address combination.

> The attacker is thus in a position to intercept, to manipulate or interrupt the communication and can start other attacks such as flooding/denial of service and paralyze the complete network. The IEEE 802.1 thus standardized IEEE 802.1AE,[17] whose latest amendment, 802.1AEbw, was released in 2013 [73] and which is generally referred to as MACsec. MACsec offers P2P encryption and authentication between directly connected nodes. It is performed for every hop also protecting the VLAN tag and not end-to-end like IPsec or SecOC. Especially its authentication algorithm is of interest for the automotive industry, as it provides a good level of integrity in the single infrastructure with mixed security domains that the in-vehicle network represents. However, to be usable in automotive, MACsec requires, like many of the TSN features discussed in Section 5.1.4, hardware support in controllers and switches and thus adds costs to the semiconductors. At the time of writing the discussion in the automotive industry on methods to use for Ethernet-based communication had not be concluded [74].

Cryptography is obviously very important for network security. There is a variety of algorithms available for different purposes like key exchange, peer authentication, message authentication, message encryption etc. Their details are not decisive in the scope of this book and will therefore not be discussed (see [75] for suggestions on respective publications). Crucial in automotive is their implementation. It needs to be effective and fast. In comparison with other in-vehicle networking technologies, the implementations need to cope with significantly higher data rates when used with Ethernet. Pure software-based implementations are not efficient enough, neither in terms of processing time nor in the use of resources. Hardware support is therefore needed, which is typically provided with help of a dedicated Hardware Security Module (HSM). An HSM efficiently executes cryptographic functions and securely stores cryptographic keys [76]. In consequence, the implementation of crypto-algorithms offers opportunities for suppliers to differentiate their products and is not discussed further here.

Once the communication in the network has been secured, the ECU itself, i.e., its software and electronics, need to be protected. This concerns the ECU implementation, the processors chosen, the partitioning of software on the processor or onto different processors and alike. Also, ports, currently not active can be deactivated in order to protect the ECU better. On the highest level, the application can comprise, e.g., plausibility checks and data use policies additional to yet more authentication and encryption. An application can be made to accept expected data only or to accept certain data, like control messages, only in specific application states. Anomalies can be detected if, e.g., cyclic messages are received more often than defined or if sensor data contains undefined information.

Naturally, a layered security approach makes sense also in case of other in-vehicle networking technologies. The approach as such is not Ethernet specific. However, this section shows that Ethernet-based communication in automotive benefits from the 15+ years head start, the IT industry has in security compared with the automotive industry [77]. The notion that introducing Automotive Ethernet weakens the security in cars is thus incorrect.

Note that at the time of writing, the industry had not only not yet converged on algorithms to use for security with Automotive Ethernet but there also was no respective industry-wide standardization activity. Various efforts existed that discussed other, specific aspects of security in automotive. One of the outcomes has been discussed above, the AUTOSAR secOC [67]. Another outcome was produced by the US based Society of Automotive Engineers (SAE), which was working on recommended practices for in-vehicle cybersecurity [78]. The focus of the resulting SAE J3061 is on the integration of cyber security in automotive processes and not on specific protection mechanisms. Furthermore, members of the German Association of the Automotive Industry (VDA) had initiated the ISO 21434 standardization project in order to cover procedural aspects of providing automotive security [79] and in the Japanese car industry, JASPAR had set up a security group in their organization [80].

5.2.2 Ethernet-Specific Security Aspects

In an Ethernet network there are two properties of importance that can have impact on the security: (a) communication can be broadcast communication and (b) there is no default control in an Ethernet network on how much traffic a network participant can sent [81] [74]. Broadcast happens with all unrestricted broadcast and multicast messages, but also when the addressee of a unicast message is not (yet) known. So broadcast can happen any time. For the ECUs, in theory, they should be designed such that they do not send too much traffic. However, next to design errors or simple malfunctioning that might cause a unit to send too much data, this can be just the outcome of a malicious attack security measures want to prevent.

Both, too much broadcast and too much traffic being sent by even only one participant, can flood the network on MAC level and result in a denial of service or other malfunctions of the network. Additionally, broadcasted messages as such can be listened to anywhere in the network. The following thus investigates the means available on switch level that support these two tasks: 1. Stop too much traffic from coming into a switch and 2. Stop too much traffic from going out (see also Table 5.5).

Section 5.1 discussed different methods available with AVB/TSN. Noteworthy in the context of security are the Stream Reservation Protocol (SRP, see Section 5.1.3.3) and ingress filtering and policing/IEEE 802.1Qci (see also Section 5.1.4). As the overview in Table 5.5 shows, ingress policing is the only really suitable means to prevent too much data at the ingress of a switch. On the other hand, the SRP has some but only small influence on the traffic at the exit ports, because the SRP switches impose transmit limits at class level and not for individual streams, senders, or receivers. The following two subsection explain the effects that can be achieved when using VLANs and when using the switch configuration accordingly. Last but not least, Section 5.2.2.3 will briefly cover the topic of key management.

5.2.2.1 VLANs

One way to structure data within an Ethernet network is the differentiation between infotainment, control, and best-effort traffic, as described in Section 5.1. Another way to

Table 5.5 Overview of means to mitigate security threats in the Ethernet switch (without authentication or encapsulation)

	AVB/TSN	VLANs	Switch configuration
Stop too much traffic from coming in	**(++)** Ingress policing, IEEE 802.1Qci	**(+)** VLAN filtering can, e.g., drop packets with no, unknown, or unsupported VLAN ID	
Stop too much traffic from going out	**(+)** SRP limits the outgoing traffic per class and port	**(++)** limit VLAN traffic to respective VLAN **(+)** VLAN broadcast zones **(+)** Only forward packets with VLAN tag **(0)** add VLAN tag to packets without	**(++)** (semi)static ARP and MAC address forwarding tables **(++)** multicast filtering, define rules for packets with unknown addresses

Note: For authentication and encapsulation, see Section 5.2.1.

structure the data is by assigning data to different Virtual LANs (VLANs) as defined in IEEE 802.1Q. A VLAN describes a virtual Ethernet segment in the Ethernet network, in which all participants are identified by the same VLAN ID (see also Figure 5.3 or Figure 1.5 in Section 1.2.1). This means that a VLAN enabled switch will pass on packets with a VLAN ID only between units of the same VLAN and not to others, even if units of other VLANs are physically connected to the same switch or if the message is a broadcast message. VLANs were developed in the context of significantly larger IT networks, in order to be able to handle and segment them, but they are useful also in the context of Automotive Ethernet.

In respect to security, VLANs can thus be used to isolate traffic and to reduce broadcast zones. Depending on the design, the isolation can be between critical/uncritical traffic, internal/external traffic or it can isolate the traffic flows of different application areas or security zones. An example of separating different VLAN traffic flows is shown in Figure 5.10 and described further below. VLANs can also be used to perform some ingress policing and drop all packets that are not part of the VLANs the switch supports. If packets arrive without VLAN tag, the switch can drop the packet or it can add a tag based on available packet information like port, protocol, fixed header fields etc. [74]. To manipulate this would require hacking into the switch configuration interface, which often is a host controller or µC. Consequently, in case of VLANS it is the µC that would need to be hacked, which should in any case receive considerable security protection.

Note that next to improving the security, a smart use of VLANs can simplify various challenges. Examples are:

- **Data logging and testing:** VLANs give flexibility in relating ECUs to network segments, independent from the physical location of the units. This will have an increasing importance for data logging and analysis in growing Automotive Ethernet networks.

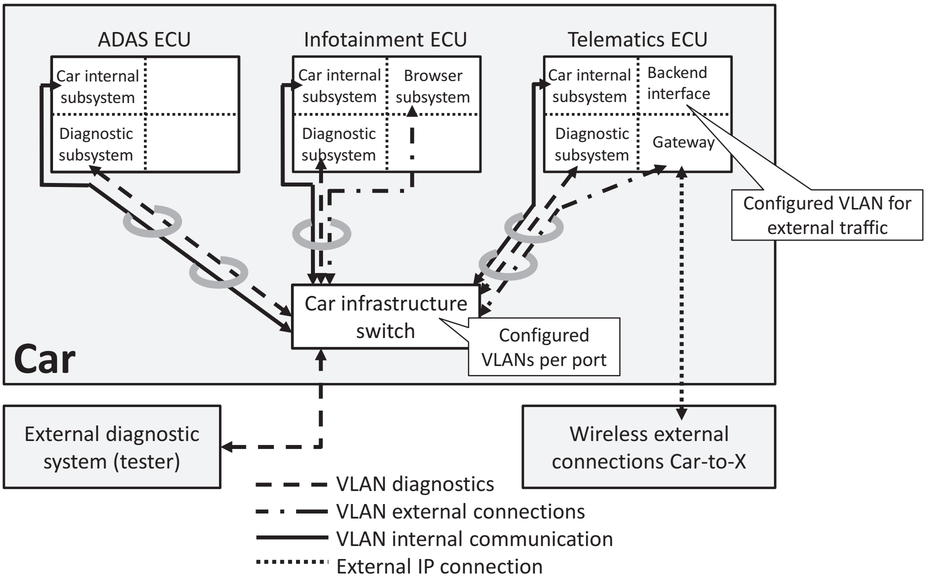

Figure 5.10 Example use of VLANs including software compartmentalization.

- **Performance:** A certain communication can be assigned to a specific VLAN and this VLAN can be prioritized within the switches.

Figure 5.10 shows an example of what traffic isolation in an in-vehicle Ethernet network could look like. In the example an ECU called "Car infrastructure switch" has been selected as the major unit to perform traffic isolation. This unit can physically located anywhere inside the car. The VLAN filtering rules, defined during the development process, are applied per port. The depiction also shows where adding or removing a VLAN tag makes sense: for both external interfaces. When, e.g., diagnosing the car via the standardized diagnostic interface, the diagnostic traffic simply receives the respective diagnostics (VLAN) tag and is then distributed in the diagnostics VLAN inside the car. Within the car, this traffic is always seen as "diagnostics traffic" and never as "car internal traffic," which makes effective isolation quite easy and results in a quite efficient firewall. The external tester is unaffected.

The situation is the same for the other external interfaces most modern cars will have, e.g., for connecting to the Internet. In this case, it is possible to work identically. Traffic that enters or leaves the car via one of the many radio interfaces can be tagged as coming from outside, while the tag is removed when data from the car leave via the same interface. Inside the ECU such tagged traffic is handled in an isolated area only. This ensures that a browser application has no access to car internal data, even if it passes along the same wires. The strict separation of traffic is crucial and car manufacturers should ensure this with help of a respective development processes.

VLANs are unprecedented in cars. Their implementation offers a powerful tool to the designers of the Automotive Ethernet network and will also provide competitive advantages for those who do it well. It is therefore unlikely that the automotive industry will standardize how to implement VLANs on a general basis. However, for specific use cases, like the automotive camera interface (ISO 17215), the use of VLANs is addressed.

5.2.2.2 Other Switch Configuration Possibilities

The main task of a switch is to look at the address fields of a packet received and to forward it to the exit port(s) behind which the destination address can be found. In order to be able to do this, the switch maintains a forwarding table. This table is being filled by the switch remembering from which port packets with which source MAC address comes from, so if the same source address is found later in the destination MAC address field of a different packet, the switch knows behind which port the unit can be found. If a destination address is not yet known, the switch forwards the packets to all ports (i.e., broadcasts it). A potential attacker could thus flood a network with broadcasted messages by continuously sending packets with destination addresses unknown to the switch.

Such attacks can be prevented if the MAC address learning of the switch is limited to an initial starting/setup phase using the first packets only or if the address learning is switched off completely and instead a preconfigured static forwarding table is used. This includes the possibility that the forwarding table is learned once with the first start in the factory and that the resulting forwarding table is stored and used after that. The same applies for the ARP tables that match MAC and IP addresses. In the static network of a car these are viable and advisable proceedings. Naturally, the learning can also simply be limited based on number of entries, address range, and frequency of change [81].

Additional filtering rules can be defined for multicast messages. Without filtering rules, multicast messages are also simply broadcasted in the network. Rules that match specific multicast addresses to a fixed set of destination addresses can ensure that the switch forwards multicast messages only to those. Furthermore, it should be defined upfront what to do in case multi- or unicast messages are received with unknown destination addresses. Such messages can be dropped, or likewise forwarded to (a) specific address(es) only.

References [81] [74] propose the use of Access Control Lists (ACLs) to precisely configure packet forwarding in a switch. ACLs have originally been developed for IP routers for applying forwarding rules on layer three, but they can also be found in switches. An ACL is a list of match-action pairs that can be applied to VLANs or ports and typically permit or deny transmission based on the bitwise match of the Ethernet or higher layer protocol headers (e.g., IP, UDP, or TCP). Table 5.5 gives an overview on the different, Ethernet inherent means and their effectiveness to mitigate threats of too much traffic entering or leaving the switch.

Naturally, the switch can be used to enforce authentication such that switch ports are deactivated and will only be activated for normal communication after the connected node has been successfully authenticated. The authentication of the network nodes need to be done by the switch firmware or the microcontroller directly connected to the switch to guarantee a fast start-up.

5.2.2.3 Efficient Key Management

As a basic principle of cryptography the same key should not be used for different functions and devices and a key should not be used for a long period of time [82]. Additionally, data authentication, key exchange, and data encryption each requires a different key. If the key for data encryption is compromised, for example, the other keys are not affected and a new key can be assigned via the existing encrypted key exchange. To limit the amount of data for which a key is used, every key is assigned a specific lifetime for which it is valid, depending on its use. Moreover, communication can be divided into separate groups with separate connection keys according to the respective vehicle domain or other functional aspects.

The distribution of the keys to the ECUs in the vehicle is a very complex task. The usual methods employed in the IT world, such as the Internet Key Exchange Protocol IKEv2 (IETF 4306) and X.509 certificates (IETF 5280), are not suitable for in-vehicle key management. Those methods would consume too many resources and an online connection to a Certification Authority Server would be required. This can neither be guaranteed at all times nor is it fast enough. Instead, one ECU in the vehicle needs to assume the role of a key master which distributes the keys to the other ECUs.

A potential solution is the following: The key exchange is realized by means of symmetric encryption and is triggered through a diagnosis request, an elapsed period of time or from a service backend server outside the vehicle. In this setup the key master is the only ECU that communicates with the service backend server regarding key management. It uses asymmetric cryptography methods for this purpose.

5.3 The Internet Protocol (IP)

The Internet Protocol (IP) is the fundamental protocol associated with enabling and using the Internet. However, it is only one of several protocols that represent the “Internet protocol suite” or “TCP/IP protocol suite,” a combination of protocols that enables vendor and operating system–independent communication between networking enabled electronic devices [83].[18] The first version of the protocol(s) was published in 1974 as RFC 675, “Specification of Internet Transmission Control Protocol” (TCP) [84]. In 1981, with the fourth version, TCP and IP were, for the first time, separately described in RFC 793 [85] and RFC 791 [86]. The User Datagram Protocol (UDP) was standardized in 1980 (RFC 768 [87]) and is also part of the TCP/IP protocol suite. Today, most networking uses the TCP/IP protocol suite. Thus, cars have to facilitate it if they are to be handled as nodes in the (worldwide) network. Cars also have to support it as part of the Ethernet-based communication in in-vehicle networking. In short, the TCP/IP protocol suite is a fundamental part of AANs.

In general, there are no specific automotive requirements when implementing the respective protocols. After all, the possibility of reusing standard compliant implementations of protocols like TCP, UDP, and IP is one of the reasons for doing Ethernet-based communication in the first place. One aspect that has to be taken into account when implementing the TCP/IP protocol suite is the footprint of the software. Small ECUs,

Table 5.6 Original IPv4 addressing classes

Class	Range class	No. networks	Hosts per network	Range private	Amount private	Purpose
A	0.0.0.0 to 127.255.255.255	126	~16.78 Mio	10.0.0.0 to 10.255.255.255	~16.78 Mio	Unicast
B	128.0.0.0 to 191.255.255.255	16.384	65.534	172.16.0.0 to 172.31.255.255	~1.05 Mio	Unicast
C	192.0.0.0 to 223.255.255.255	~2.1 Mio	254	192.168.0.0 to 192.168.255.255	~0.066 Mio	Unicast
D	224.0.0.0 to 239.255.255.255	n/a	~221 Mio	n/a	n/a	Multicast addresses
E	240.0.0.0 to 255.255.255.255	n/a	~315 Mio	n/a	n/a	Reserved for experiments

Note: The use of A, B, and C has been obliterated with CIDR and, of course, IPv6.

like cameras integrated into the side mirror, have little processing power available in optimized DSPs or µCs. It is therefore important to pay attention to the implementation of the software [88]. The skillful implementation of the TCP/IP stack gives a competitive advantage to those capable. However, it is not the topic of this book.

The portion of the TCP/IP protocol suite that does leave some structural choices in automotive is the use of IP. The core functions of IP are addressing, i.e., identifying and locating units (called "hosts" in IP), and routing packets from a source address to a destination across, well anything, from within a small network to across multiple networks and around the world. In the public Internet, IP addresses thus have to be globally unique. To ensure this, Regional Internet Registries distribute the IP addresses, while the Internet Assigned Numbers Authority (IANA) publishes the availability of addresses. Additionally, there are IP address ranges available for closed/private networks (see, e.g., Table 5.6). Anybody can use these, as long as the units using them have no access to the global network. Examples for closed networks are factory floors, on which robots communicate (only) with each other or, in parts, cars (see Section 5.3.1 for more details).

An Internet Protocol version 4 (IPv4) address consists of 32 bits, which originally only identified the network and the host in the traffic classes shown in Table 5.6. It became evident quickly that the original concept was not sustainable and various methods were developed either to make routing more efficient and/or to make the address range stretch further. Examples are subnetting, Variable Length Subnet Mask (VLSM), and Classless Inter-Domain Routing (CIDR) [89].[19] Despite these efforts, the predominantly used IPv4 has run out of addresses [90] and everyone designing networks using IP today – which includes the automotive industry – has to integrate the use of IPv6 (see Section 5.3.2). IPv6, first published in 1995 [91], uses 128 instead of 32 bits for the address, so there is hope that the address space will last longer.

Last, but not least, the question of security is often discussed in the context of IP. Because IP is the connecting element in the worldwide network, it is seen as an entry point also for undesirable elements. Section 5.2.1 provides some basic considerations for security in automotive, including a brief description on IPsec, a security protocol to use on layer three.

5.3.1 Dynamic versus Static Addressing

Modern IT systems need to be very flexible. They have a changing number of nodes and routes in the network. One of the key elements to support this is the dynamic setting of IP addresses. Deploying Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) servers is state of the art. Automotive Ethernet networks are entirely different. An in-vehicle network is an almost closed system. Even if the number of active nodes in an in-vehicle network may vary (for more details see Sections 5.4.4 and 6.3.3), the maximum number of nodes is more known and more limited upfront. Furthermore, in contrast to IT networks, cars might be started and parked several times a day (see also Section 6.3.1). Fast start-up is therefore very important, which makes a static IP configuration the natural choice. However, there are some use cases in which dynamic IP addressing makes sense in cars, too.

The following possibilities exist to assign IP addresses inside cars. The list shows how Automotive Ethernet adds, with IP addressing (and VLANs, see Section 5.2.2.1), another design dimension to the electronics development inside cars.

- **Static:** The IP addresses are assigned during the development and every ECU of the same class, e.g., Head Units (HU), always receives the same address, independent from the car they are built into. As the same address is obviously used repeatedly between cars, it is selected from a specific address range. The private address pool listed in Table 5.6 would be a designated source for it, but does not have to be. In the case of static addressing it is absolutely important that never two ECUs of the same class are built into one vehicle.
- **Pseudo-dynamic (branding):** In this case the ECU is delivered without IP address, but receives a then static IP address during the assembly. Consequently, after the address has been assigned, it cannot be changed anymore. This process is needed in case the same part is assembled multiple times inside a car. The cameras of the surround view system provide an example. Exactly the same camera is placed at different locations inside the car. So, for assembly and also for repair, the cameras are delivered without IP address. This reduces costs for logistics and storage. This branding procedure is standardized with the automotive camera interface of ISO 17215 (Part 4).
- **Dynamic:** This is required, when the vehicle or parts of the vehicle communicate with external components/the world outside the vehicle, e.g., the diagnostic tester (see ISO 14300) or the Internet. In this case, it is no longer possible to use addresses from the car internal address space. Instead, the ECUs directly connecting the car to the outside world, "port ECUs," use the IP address(es) an externally located DHCP server assigns to them. One possibility to connect other ECUs inside the car via those "port ECUs" is to implement a dynamic/static address translation with a Network

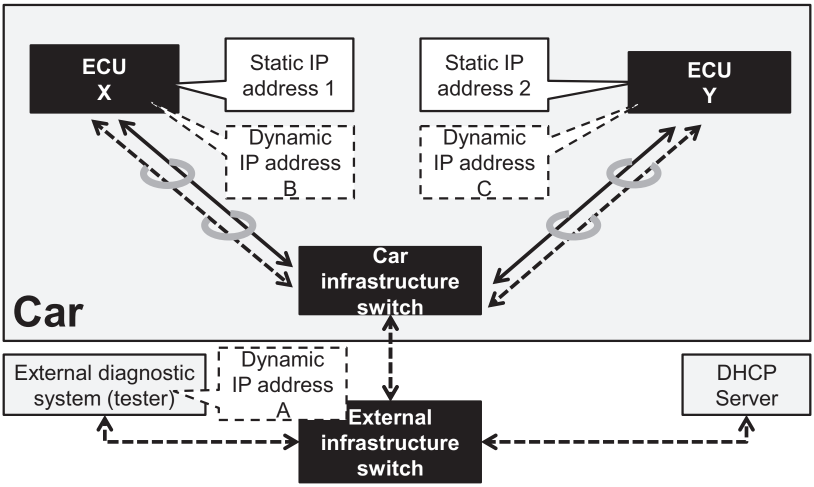

Figure 5.11 Example use case for multiple IP addresses (solid line for regular vehicle internal traffic, dashed line for diagnostic traffic).

Address Translation (NAT) in the port ECUs. Another is to request more temporary IP addresses from the external DHCP server.

- **Multiple:** In this case, an ECU accommodates and uses several IP addresses. This is the case if ECUs want to use various address spaces. One example is the diagnostics use case (see Figure 5.11). In this case the car internal network uses static IP addresses. The external tester cannot know the internal address structure, and the internal network cannot participate in the external communication. For the time of the testing, additional IP addresses are thus assigned by the DHCP functionality of the external network that the tester is part of. Figure 5.11 shows an example of the use of multiple IP addresses. As can be seen, where appropriate, separation of traffic can be achieved also with multiple IP addresses assigned to the communication partners on layer 3, and not only with VLANs on layer 2. In the example shown, only the dynamic addresses B and C are needed and they are thus assigned only when the diagnostic system is connected and would like to communicate with the internal components directly.

5.3.2 IPv4 versus IPv6

With the address space of IPv4 running out [90], migration scenarios from IPv4 to IPv6 are often discussed [92]. However, this is not really a concern for Automotive Ethernet nor does the automotive industry face challenges that are different from those of other use cases. A significant amount of automotive communication is internal to the vehicle only and it is the explicit intent that there is no interface to the outside world.

This communication can continue to use static IPv4 addressing, if this is desired, e.g., because the complexity is smaller than for IPv6. For the communication between car and external world, this is of course different. The car has to integrate into the network the outside world defines and the respective components naturally need to support IPv6 to be future-proof. With the multiple addressing scheme an Automotive Ethernet network can support both.

5.4 Middleware and SOME/IP

5.4.1 Definition of "Middleware"

This section starts with the disambiguation of the term "middleware." The term originates in the development of complex software systems and addresses all functions that are needed for a "service" to allow data exchange between otherwise decoupled software components. This data exchange often passes through a network and it is the task of the middleware to ensure that the network used is transparent to the software components exchanging the data. As is shown in Figure 5.1, middleware ("SOME/IP") cooperates at the higher layers of the ISO/OSI layer model. It organizes the transport of complex data (messaging) and moderates function calls (Remote Procedure Calls, RPCs) between the software components.

One of the disadvantages of using a middleware is its size and load. However, with the increasing size and performance of software systems this is mitigated. Furthermore, it becomes ever more important to be able to handle complex software systems comfortably and to improve their quality; with middleware being a suitable tool for it. The advantages of using modern middleware are the improvement of distributed development on different software modules and the much better testability of the modules.

The amount of software in cars today can be huge [93] and is still increasing along with the distribution of functions and systems inside cars (see also Chapter 7 for an outlook). These distributed functions can use various processes within one ECU but they might be spread also over various processes in different ECUs. With the thus increasing complexity, simply placing messages onto the network under the assumption that the correct function will receive them is no longer sufficient. RPCs are required to control the distributed functions and the correct methods to initiate this. Additionally, different ECUs might use different software architectures (and Operating Systems, OSs). This means that middleware also has the important role of bridging between Portable Operating System Interface (POSIX) capable Unix-like operating systems such as Linux or QNX and AUTOSAR systems, which are all used in automotive.

5.4.2 The History of SOME/IP

When starting the development of Automotive Ethernet, the intention was to reuse one of the many middleware solutions available; preferably one following an open source licensing model. Various approaches were scanned and some solutions, such as Etch [94] [95] [96] or Google Protocol Buffers [97] (serialization only) for middleware or Bonjour [98] for Service Discovery (SD), were investigated more closely. In principle, both solutions could have been modified to fit the small processing capacities available. However, two issues remained unsolved:

- **AUTOSAR** provides many software modules, which incorporate some of the middleware functions and are configured with the help of a separate tool chain. In order to avoid **incompatibilities**, the reuse of existing (IT) middleware solutions would have required either bypassing the AUTOSAR modules or ensuring the use of the same data types and the partitioning of the existing middleware solutions such that they could have been integrated in AUTOSAR.
- **The licensing of the existing (IT)** middleware solutions was **not** quite **as needed**. Although the licensing of the respective implementations of the investigated solutions was open, the essential patents needed for adapting the solutions were not. Instead, those patents were protected and owned by large IT companies, with unknown consequences.

While, in theory, it would have been possible to make one of the technically suitable solutions usable with AUTOSAR, this was not possible in combination with the licensing issue. It was thus decided to develop a new solution. To reduce the risk of running into licensing issues with the new solution, the IPR situation was taken into consideration, while at the same time the new solution was being published as state-of-the-art technology. Naturally, the solution was developed to be directly usable with AUTOSAR systems. The Scalable service-Oriented MiddlewarE over IP (SOME/IP) specification has thus been an integral part of AUTOSAR since AUTOSAR version 4.1. Additionally, SOME/IP is provided as a GENIVI library. More public information is available on [99].

5.4.3 SOME/IP Features

SOME/IP was designed to support the following features needed for the automotive use cases:

- Service-based communication approach
- Small footprint
- Compatibility with AUTOSAR (no other middleware is AUTOSAR compliant)
- Scalability for the use on very small to very large platforms
- Flexibility in respect to different operating systems used in automotive like AUTOSAR, OSEK, QNX, and Linux

5.4.3.1 The Header Format

Figure 5.12 shows the SOME/IP header. The individual elements are explained in the text below.

- **Message ID:** The first 16 bits of the Message ID identify the service (Service ID) used. The service provides the overall structure for the middleware communication. An example of a service could be "*CD_Player*" (the complete example is given in the notes[20]). Each service needs to have a unique Service ID, which the system integrator

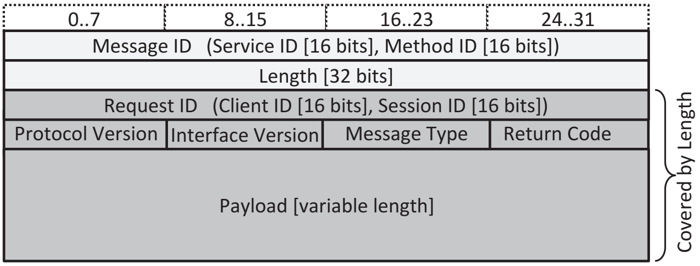
0..7
8..15
16..23
24..31
Message ID (Service ID [16 bits], Method ID [16 bits])
Length [32 bits]
Request ID (Client ID [16 bits], Session ID [16 bits])
Protocol Version
Interface Version
Message Type
Return Code
Payload [variable length]
Covered by Length

Figure 5.12 Header format for SOME/IP.

assigns. A service can consist of a set of methods, events, and fields, which are identified in the Method ID. The 16 bits for the Method ID represent the other half of the Message ID. An example of a method could be "track_number.set." In comparison, CAN provides only a small subset of what is possible with service-based communication. However, the idea behind the SOME/IP message IDs is similar to that of CAN message IDs (see Section 2.2.2.2). It is therefore possible to treat the SOME/IP message IDs with the same process structure, which just needs to be enhanced/adopted for SOME/IP.

- **Length:** The length field uses 32 bits to specify the number of bytes including the payload, some header information, and the Request/Client ID (see Figure 5.12).
- **Request ID:** The Request ID allows a client to differentiate between multiple calls of the same method. The first 16 bits of the Request ID are called Client ID and identify a specific client. For example, if a user would like to set the track in the CD-player (server) from the Head Unit (Client A), this would have a different Client ID than if a user of the Rear Seat Entertainment (RSE) (Client B) would like to set the track in the same CD-player. The second 16 bits of the Request ID represent the Session ID. If, e.g., Client A sends the message to set the track in the CD-player multiple times, each of these messages receives a different Session ID. When generating a response message, the server always has to copy the Request ID from the request to the response message. This allows the client to map a response to the correct request. The Request ID is an inheritance from AUTOSAR's Client/Server communication.
- **Protocol Version:** An 8 bit field which identifies the SOME/IP protocol version. At the time of writing, SOME/IP has the version 1.
- **Interface Version:** These 8 bits identify the major-version of the service interface. The interface definition and version numbering is up to the designer. In case additions are made and new versions are defined, this field in the header allows the automatic detection of version incompatibilities in the design.
- **Message Type:** This field differentiates between the different possible types of messages. With SOME/IP version 1.0 the values shown in Table 5.7 were defined.
- **Return Code:** The 8 bits of the Return Code signal whether a request was successfully processed.
- **Payload:** The payload field contains the parameters of the SOME/IP message. In the case of the example, this might be "10," if that represents the value the track should

Table 5.7 Important SOME/IP message types

Value	Name	Purpose
0x00	REQUEST	Request expecting a response (even void)
0x01	REQUEST_NO_RETURN	Fire and forget request
0x02	NOTIFICATION	Request for a notification (i.e., a subscription for an event call back or a field value) expecting no response
0x80	RESPONSE	The response message (for a REQUEST or as a result of a NOTIFICATION)
0x81	ERROR	In case a RESPONSE cannot be delivered because of an error

be set to. The size of the SOME/IP payload field depends on the transport protocol used. For UDP, the SOME/IP payload can contain 0 to 1400 bytes. The decision to limit the payload length to 1400 bytes was taken in order to allow for future changes to the protocol stack, like using IPv6 or adding security protocols. Since TCP supports the segmentation of payloads, larger payload sizes are automatically supported. With the SOME/IP Transport Protocol (TP) segmentation larger payload sizes are also supported for UDP. The serialization of parameters, i.e., the order of the values in the payload and in which order to place the least to most significant bits, is also specified in SOME/IP.

5.4.3.2 The Service Concept and the Supported RPC Mechanisms

SOME/IP defines a service by its Service Interface, i.e., the activities of client and server, based on the defined communication principles. In this, a Service Interface is comparable to a MOST FBlock (see also Section 2.2.4.2). Figure 5.13 gives an overview

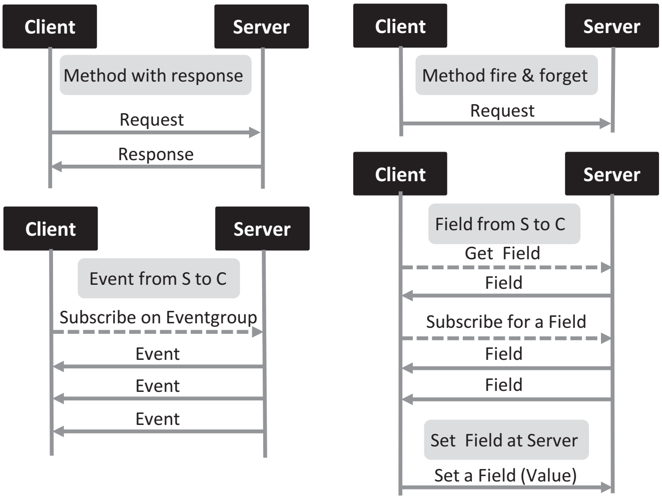

Figure 5.13 Communication principles supported by SOME/IP. C = client; S = server.

of the different communication principles supported by SOME/IP. A Service Interface may include (a) methods with response (request/response) or without response (fire & forget); (b) events, i.e., a message from the server to the client when something happens; (c) fields, which get, set, or notify of a property or status; or (d) event groups, which are logical groups of events and fields used for publish/subscribe handling. Figure 5.13 visualizes these basic communication principles.

- **Request/Response:** Describes a method with Request and Response messages. The Request is a message from the client to the server calling a method. The Response is a message from the server to the client transporting the result of the method invocation.
- **Fire & Forget:** Describes a method with just Request messages. Like in the Request/Response case the client invokes a method at the server. In contrast to the Request/Response case, the client does not expect a Response.
- **Events:** In this case the server sends messages with specific information to the client, either cyclically or when there is a change (event). Prior to this, the client will have told the server that it wants to receive the information, i.e., will have "subscribed." As the server expects no response from the client, this could be also seen as a "Fire & Forget" communication principle from the server side. The Event messages are similar to regular CAN messages.
- **Fields:** Fields represent "properties" that can be accessed remotely. The communication principles for "get" Fields are in line with Events. In addition, Fields can be "set" by the client. Furthermore, the "properties" are available at all time that system is alive, while the Event is only valid within the time this event is happening. Thus, a "property" can be seen as a kind of software variable that can be addressed from an external interface. Fields are similar to "Properties" in MOST.

5.4.4 Service Discovery (SD)

Automotive Ethernet is often just seen as a new in-vehicle networking technology that simply allows for higher data rates. However, there are a few more important differences. In the context of the communication methods, the key difference is that Ethernet-based communication provides for service-orientation. Until the introduction of Automotive Ethernet, MOST was the only in-vehicle networking technology supporting a service-based approach, and therefore service-oriented communication was only deployed in the infotainment domains of those car manufacturers using MOST. That MOST/infotainment uses a service-based approach is not by accident though. High-end infotainment systems were the first to need the more complex interfaces that service-orientation provides. Examples include complex data types, access to databases, the transmission of lists, etc. Also the use of Remote Procedure Calls (RPCs) was first required in infotainment and is thus also supported by MOST.

In the rest of the in-vehicle domains the "CAN-approach" dominated, i.e., information is put onto the channel and it depends on the mechanisms implemented in the receiver if and how that information is processed. However, especially in innovative user domains like the driver assist domain, the "CAN-approach" is becoming less suitable to cover the communication requirements. Additionally, with a data payload of 8 bytes, CAN does not allow for extensive header information, which limits the adoption of new concepts like RPC or SD. With Automotive Ethernet being used in the driver assist domain and more service-orientation diffusing into other in-vehicle areas, the authors are convinced that a general shift to service-oriented communication is necessary to meet the in-vehicle communication requirements of the future.

Service-based communication is one of the key properties and key advantages of SOME/IP. Also important is the addressing. Communication is not only via broadcast but uses unicast as well. With unicast, it makes sense to address communication partners only if they are really available.

With SOME/IP-SD, SOME/IP also supports a mechanism that determines which and if a service is available or not. Service Discovery (SD) as such, however, is still a controversial subject in the automotive industry. The main counterargument is that the in-vehicle network and the availability of functions are not dynamic enough to justify the use of SD. The following list reflects a number of situations in which the seemingly static network is faced with increasing dynamics, but for which SD provides a solution:

- **Dynamics during start-up:** One of the more complex tasks in the system design of cars is the start-up. Each ECU in a car can show a different start-up behavior. Some ECUs will start-up quickly; others will be slower. Some ECUs start even if the voltage drops down to 3.5 V; for others a start-up voltage of 8 V is not sufficient. This means that during the start-up, functions are available at different points in time. Without SD, a hard limit needs to be set that defines the point in time at which all functions are expected to be available. It needs to be defined according to the function or ECU needing the longest start-up time. With the above mentioned different option/combinations, the time limit would either be different for every vehicle or always the longest. In contrast, with SD available, every function/ECU can announce its availability when ready and, in general, user functions can be available earlier. This significantly simplifies the process. During start-up, SD has another advantageous side effect in a switched Ethernet network: The switches can learn the addressing tables directly with the SD messages.
- **Dynamics because of customer variants:** Most car manufacturers offer options for their customers to choose from when buying a car. As a rule of thumb, the bigger and more premium a car is, the greater the number of options that can be selected. A large number of options means an even larger number of combinations of options, so in consequence many car manufacturers build individual cars according to the specific customer's requirements. Without SD, each ECU needs to be statically configured with respect to the availability of functions of other ECUs in the car. With SD available, ECUs can establish on their own which other functions/ECUs are available in a vehicle, without requiring any option combination–specific preconfiguration. This is significantly less error prone. Therefore, the more complex the car, the larger the advantages of SD are.
- **Dynamics in the event of failures:** In a network that functions with Fire & Forget messages only, it is not always directly evident when a communication partner has

failed. An ECU not detecting any related messages on the link might simply assume that a certain event has not happened or that a value has not changed. In contrast, with SD active in the background, an ECU will know immediately when a service/another ECU is no longer providing a certain functionality. Failures are thus detected better and the respective failure modes can be activated within a certain timeframe.

- Additionally, SD can be used with individually **adjustable "Time-To-Live" (TTL) values**, which indicate how long an entry is valid. A user of that value expects an update once the TTL has expired. If it fails to arrive, the user can also conclude on the faulty behavior of the communication partner and can start specific error processing. This improves the stability of a network, but of course cannot replace messages in case cyclic data is missing for a safety critical application. Cyclic messages with "Application Cyclic Redundancy Checks" (CRCs) are normally used for an end-to-end safety application.
- **Dynamics in the case of partial networking for energy efficiency:** Because of the increasing size of the in-vehicle network and the increasing number of ECUs, energy efficiency is ever more important. As is pointed out in Section 6.3.3, it is of interest to fully power only those ECUs that are needed at a particular moment in time. This needs to take different scenarios into account. A customer might want to finish a call via the built-in hands-free system, despite having arrived at their destination and parked the car. The car should then be smart enough to deactivate all ECUs on the network that are not needed, like the engine control or drive train. This example shows that, with partial networking, the in-vehicle network can be expected to change dynamically. In the changing environment, active ECUs have to know which functions are still available and which are not. Without SD this can be realized with timeouts. Like in the start-up scenario, however, this makes the system slower than with SD. With SD the knowledge of availability is more immediate.

The more complex an in-vehicle network becomes, the better it is to have service-based communication and SD available. Without service-based communication, the complexity of an Automotive Ethernet network is much higher, as explained above. When implementing SD in a network, there are two principal approaches, a centralized and a decentralized one:

- In the **centralized approach**, one ECU monitors and maintains the service information of the network. Each participant sends its respective information just to this one ECU and each participant requests the respective information just from this one ECU.
- In the **decentralized approach** all communication partners apply the following rules: Each communication partner offers its available services to all other units via multicast or broadcast and each communication partner requests available services from all other units via multicast or broadcast. If two communication partners find each other, they can establish a respective one-to-one communication. Important advantages of the decentralized approach are: minimal start-up delays, which then mainly depend on the start-up time of the physical network; no specific third-party ECU is needed; and there are multiple sources of data, meaning that no single component has to

handle all the data but the load and risk in the event of an error are distributed. In other words, there exists no single point of failure.

Notes

1 The terms bridge, switch, and router are not always unambiguously used. This book shall use the terms similarly to what is being described in [100]. All terms – also including hubs and repeaters – refer to units in a communication network that pass on data. What differs is how and on what basis they do this.The use of the term **router** is comparably consistent. A router forwards data packets on the basis of ISO/OSI layer 3 addresses, which in today's networks are IP addresses. "Routing" describes a functionality. The term in itself does not say whether a router is a standalone box or a function integrated into a microcontroller. Generally, routers are used for large-scale communication: to pass on data between different Wide Area Networks (WANs), between different Local Area Networks (LANs), and also between LANs and WANs. In vehicles, routing is used in the flash and diagnostic automotive use case described in Section 3.1. In this use case, the central gateway holds the router that connects the car's internal "LAN," or so-called Automotive Area Network (AAN), to the outside world (see also Section 5.3 for the use of IP in an Ethernet-based communication in automotive). Routers can also be used to pass on data inside LANs, though this can generally be done more efficiently via what in this book shall be referred to as "switches."

The use of the term **switch** is not so consistent. In this book, a switch forwards packets in an Ethernet network based on the ISO/OSI layer 2 addresses, i.e., the hardware/MAC address provided in the Ethernet packet (see Section 1.2.1). A switch is directly related to the Ethernet technology and, at least in in-vehicle networking, a new concept. The impact of this concept on the in-vehicle EE architecture and topology choices is severe and subject of a separate chapter (Chapter 6). The switching function is generally realized in hardware in a special switch semiconductor. The semiconductor can be "switch only" (which is rare), a "switch with integrated PHYs" (which is most common) or a "switch integrated into a System on Chip" (SoC). If this book talks about switches, the default meaning is the (part of the) semiconductor that provides the respective function. In the IT industry, the term "switch" often refers to a standalone networking product, which actually has quite a market (see Table 1.3). This switch is a box with a number of RJ-45 sockets that allows the connection of various devices via Ethernet and that will direct the traffic between them (based on the layer 2 address). In Chapter 6, this book will also consider a separate ECU containing the switching function, which is then a "switch box" or "standalone switch." Sometimes the function of a switch is extended to layer 3 as a "layer 3 switch," which somewhat blurs the distinction and can be confusing. In this book, layer 3 forwarding is "routing" and layer 2 forwarding is "switching." Note that the IEEE 802.1 specifications never utilize the term "switch" as used here, but only "bridge."Following [101], a **bridge** also passes on traffic based on layer 2, but can do so not only at the MAC level between Ethernet links, but also on the Link Layer Control) level. This means, it can handle different MAC control algorithms and can thus bridge traffic between different IEEE technologies without needing the IP address. In the IEEE 802.1 terminology this distinction makes sense; for Automotive Ethernet it is of minor relevance. Occasionally the term "bridging" is also used on layer 3; after all, various different technologies can be bridged with the help of IP addressing. When using the term bridge in this book, it will always be used in combination with the layer it bases its functionality on – like "layer 2 bridge" – in order to avoid confusion.

Last, but not least, there are repeaters and hubs. Both function on layer 1, i.e., they have no intelligence that allows them to forward a packet based on addressing. A **repeater** is a simple one-to-one device that amplifies the signal to increase the range. Repeaters can be of interest in automotive, too, when, e.g., Ethernet PHYs designed for 10 or 15 m links are being used in trucks or busses. **Hubs**, in contrast, have lost their relevance in Ethernet networks in general, and will also not be considered in automotive. Hubs take the input data from one connected device and broadcast them to all attached other devices. This makes the link a shared link and contradicts the P2P/switched Ethernet network approach that has replaced CSMA/CD Ethernet a while back.

2 Since then Napster has been relaunched more than once. Latest, in June 2016, the music streaming service Rhapsody rebranded itself as Napster [102]. This shows that the memory of the revolutionary change Napster initiated in music consumption around the turn of the century, is still attributed with sufficient marketing potential 15 years later; even if user have to face the change from “for free” to “add-free” [100].

3 IEEE 802.1 Qat “Stream Reservation” as well as the IEEE 802.1 Qav “Traffic Shaping” were both incorporated in the IEEE 802.1Q revision of 2011 [37].

4 Universal Plug and Play (UPnP) describes a set of protocols that allow for the vendor-independent, distributed media management, discovery, and control in an IP-based network that consists of consumer devices like computers, printers, Internet gateways, audio rendering units, mobile devices, etc. The first version of UPnP was published as ISO/IEC 29341 in December 2008 [103] and updated/extended in 2011. The UPnP forum, which drives and markets the developments, was founded in October 1999. Since January 2016, the Open Connectivity Foundation (OCF) has taken over the assets from the UPnP Forum [104].

5 The Digital Living Network Alliance (DLNA), founded in June 2003, has the goal of ensuring the interoperability between applications of networked consumer devices that involve images, AV data [105]. For this, the DLNA provides design guidelines based on higher layer standards like UPnP and certification programs. The guidelines are based on standards like UPnP for media management, discovery, and control. At the time of writing the DLNA homepage listed about 1500 DLNA certified products.

6 Coordinated Shared Network (CSN) is a generic term for a network in which the media is shared on a contention-free, time-multiplexed basis. The network access in a CSN is coordinated by one unit designated or elected as the network coordinator, which also might be the interface to, e.g., an Ethernet LAN. CSN technologies are technologies used in the home environment – Multimedia over Coax (MoCa), Homeplug (Inhouse Powerline Communication), and Ultra Wide Band (UWB)/IEEE 802.15.4a are given as examples – and were thus considered necessary to integrate into the AVB concept [106] [34] [5]. CSNs play no role in automotive.

7 At the time, summer 2009, this was actually very progressive. Automotive Ethernet had not yet taken off. One car manufacturer, BMW, used Ethernet for flash updates and for a private link between HU and RSE (see Sections 3.1 and 3.2), but the interest of everyone else was very moderate. The bus system for automotive infotainment (AV) was MOST or the transmission was analog or LVDS, also at BMW. It can be assumed that the involvement of Harman and Broadcom in starting the AVnu Alliance spurred the inclusion of automotive. Harman had just provided the HU/RSE system for BMW and together with Broadcom BMW made promising progress toward the first use of 100 Mbps BroadR-Reach Ethernet.

8 The seven hops 2 ms requirement has two explanations. One is historical, the other was derived from the harshest, i.e., smallest, maximum network delay requirement in the professional auto domain. A musician needs to hear the response to his/her action within 10 ms; 8 ms of these are needed for DSP delays and the delay of the sound traveling, e.g., between speaker monitor and the musician. This leaves a delay of 2 ms for the network. These 2 ms were split into a realistic number of hops, leaving some margin and taking into consideration that in the worst case a 100 Mbps AVB packet needs to wait at every switch behind a best-effort 1500 bytes packet for about 122 μs [23]. Seven is also the historical limit for the layer 2 network hop count, which is reused in AVB. This seven-hop limit was established early on in Ethernet development. While IEEE 802.3 was working on repeaters, DEC developed

the first layer 2 bridge. The upper layer protocol DEC used for the bridge was time sensitive and did not support more than seven hops. A daisy chain of seven hops for a layer 2 bridged system seemed a reasonable worst case, and even though the DEC upper layer protocol is no longer used and technology has advanced to fast hardware supported switching, the number has stayed [6].

9 Reference [31] includes the IEC 61883-2,4,6,7,8 formats (i.e., Standard Definition Digital Video Cassette Recorder (SD-DVCR), MPEG2-Transport Stream of compressed video, uncompressed digital audio and music, satellite TV MPEG ITU-R BO.1294, and digital video data ITU-R BT.601), IIDC 1394-based uncompressed industrial camera, and formats that will be defined by the Musical Instrument Digital Interface (MIDI) manufacturers' association [107].

10 The 2016 IEEE 1722 version includes a Clock Reference Format (CRF), which allows for the distribution of event timing information within a system. This is of particular interest for Driver ASsist (DAS) functions, e.g., to be able to combine four camera pictures correctly in a surround view image. It would also allow to be able to determine the exact distance to an obstacle on the road from camera images. For autonomous driving cameras represent an important source of redundant information in addition to other sensors like radar, ultrasound, etc.

11 It sounds unusual, but especially in automotive this can happen easily. When, e.g., a car is parked in a garage where it has no GPS reception, the clock of the navigation unit might be inferior to the clock of another ECU. When the car then leaves the garage and GPS becomes available, the clock of the navigation system might suddenly be superior.

12 The Multiple VLAN Registration Protocol (MVRP) provides for the registration into the correct VLAN(s) and Multiple MAC Registration Protocol (MMRP) provides for the registration and announcement of the multicast addresses.

13 Note that when writing the first edition of this book, it was anticipated that Class C traffic would be specified to have a packet frequency of 1 kHz. This has not materialized. It was seen as more fitting to provide a traffic class based on a number of audio samples well aligned with processing capabilities, which would then have a varying packet frequency, depending on whether it was used with 44.1 kHz audio sampling or 48 kHz [38].

14 In the automotive industry, FlexRay was developed exactly to fulfill the requirements of safety critical applications (see also Section 2.2.5.2) and it might well still be used for that purpose. However, this book is about Ethernet-based communication and thus discusses the possibilities an Ethernet system provides in the same context.

15 As was introduced in Section 1.2.2, the synchronization accuracy in industrial applications is in the range of 1 μs [108]. Obviously, if a 12.2 μs packet needs to complete the transmission first, there is no chance to meet the requirement and use Ethernet for this application.

16 IEEE 802.3 limits the Ethernet frame size to 1500 bytes payload. However, a concept called "jumbo frames" exists, which is used in various nonstandardized variations and allows payloads of up to 9000 bytes payload [109]. These packets intend to increase throughput by reducing overhead. The potential delay owing to packets blocking the egress port, however, is increased almost 6-fold. Today, this is one of the reasons not to use jumbo frames. Once IET and preemption found their way into products, the authors expect that also jumbo frames will be used more frequently, also in automotive.

17 IEEE 802.1 standardized a number of security related protocols, whose reuse for Automotive Ethernet is worth investigating, e.g., IEEE 802.1x is very widely implemented for key management [74]. It defines Ethernet encapsulation for the Extensible Authentication Protocol (EAP), which in return is a framework for the exchange of authentication messages. Another standard of interest is IEEE 802.1AR, the Secure Device Identity Standard, which was first published in 2009 and updated in 2015. It defines the device identity and cryptography to be used by the device and the operation within EAP-TLS/802.1x. It assumes hardware support for efficient operation.

18 Other protocols of the TCP/IP protocol suite are the Address Resolution Protocol (ARP) and the Reverse Address Resolution Protocol (RARP), which translate layer 3 IP addresses to layer 2 Ethernet MAC addresses and vice versa (RFC 826, 1982 [110]). Furthermore, the Internet Control Message Protocol (ICMP) sends error messages or relays query messages (RFC 792, 1981 [111]). The Internet Group Management Protocol (IGMP) establishes IP multicast group membership for IPv4 (RFC 3376, 2002 [112]) on layer 2 MAC addresses. When using static multicast addresses, it is not necessary to provide IGMP for an Automotive Ethernet application. Last, but not least, the User Datagram Protocol (UDP) (RFC 768, 1980 [87]) is part of the TCP/IP protocol suite. See also Figure 5.1.

19 Subnetting allows the division of a single Class A, B, or C network into smaller networks. After all, a Class A network with up to 16.78 Mio units in one network, or even 65.534 in a Class B network, is a lot of units. Means to simplify routing and network design are thus appealing. Subnetting was first specified 1985 in RFC 950 [113]. Variable Length Subnet Mask (VLSM) allows a subnetted network to use more than one subnet mask and thus to use the assigned address space more efficiently. It was first addressed 1987 in RFC 1009 [114]. Classless Inter-Domain Routing (CIDR) eliminated the concept of Class A, B, and C addresses. With CIDR, the number of nodes in a network was no longer restricted to either 16.78 Mio (Class A), 65.534 (Class B), or 254 (Class C), but was selectable arbitrarily. CIDR also allowed the reduction of the number of entries in routing tables. It is said that, without this, the Internet would not have sustained [114]. CIDR was specified in 1993 RFC 1517, 1518, 1519, and 1520 [89]. With IPv6, neither VLSM nor CIDR are needed.

20 The service *CD_Player* is used as an example to explain the basic features of SOME/IP and RPCs. Every service has to be defined during the development process by its service interface. This is normally done with an Interface Description Language (IDL) and could look as follows:

```
Service CD_Player
{
track_number              // Field
{unsigned int track;      // the track number
set (track);              // Method for setting the track (uses a request/response method)()
get ();                   // Method for getting the actual track number played
}
tray.eject ();            // Event that is triggered if the eject button is pressed
Boolean tray_state;       // Status OPEN or CLOSED when tray is open or closed
                             respectively
tray_state: open_tray     // Method that is used for open the tray, the return value of this
   ();                       Method is
// the tray_state.
}
```

Following the above service interface definition, "A client would like to change the track to track number 10" would cause the command *CD_Player.track_number.set(10)* to be sent from, e.g., the Head Unit (HU, client) to the CD-player (server). The method of the service is *track_number.set*, the payload value is *10*, and the communication principle typically used for this would be a request/response method, meaning that a response for the command set is expected.

In the next case "A client would like to open the tray of the CD-Player and would like to know when the job is done." When using the above description, the command from any client (e.g., the Head Unit) to the CD-player would be *CD_Player.open_tray()*. In this example the client expects a response in the form of an acknowledgment. This is thus an example of the request & response communication principle. When the client receives *CD_Player.open_tray() = = OPEN*, it knows that the command has been successfully completed.

There is more than one way to achieve the same result. The key, whichever way is chosen, in service-based communication was clearly defined upfront with respective data types, the data structures, and the methods and communication principles used. In the above example the client could also send a read ("get field") command to receive information on the CD-player tray status. Or it could have subscribed to the CD-player, asking the CD-player to automatically inform the client every time the status of the tray changes ("event"). The respective command would be *Subscribe.CD_Player.Eject()*. In the event of the tray opening the CD-player would send *CD_Player.Eject()* to all subscribed clients, which then would be a event from the server to the clients.

This example emphasizes the possibilities service-based communication offers in contrast to the CAN-like communication principle of fire & forget and simple messages only. The serialization of SOME/IP ensures that the information fits into the existing packet format like all other traffic. The content, however, ensures a type of contract for a service to be fulfilled between the communication parties.

References

[1] L. Völker, "One for All: Interoperability from AUTOSAR to GENIVI," in *1st Ethernet&IP@Automotive Technology Day*, Munich, 2011.

[2] ITWissen, "QoS (Quality of Service)," continuously updated. [Online]. Available: www.itwissen.info/definition/lexikon/quality-of-service-QoS-Dienstguete.html. [Accessed 15 August 2016].

[3] ITU, "Definitions of Terms Related to Quality of Service," ITU, Geneva, 2008.

[4] ITU-T, "Network Performance Objectives for IP-Based Services," ITU, Geneva, 2011.

[5] IEEE 802.3, "Residential Ethernet, IEEE 802.3 Call for Interest," July 2004. [Online]. Available: http://grouper.ieee.org/groups/802/3/re_study/public/200407/cfi_0704_1.pdf. [Accessed 12 October 2013].

[6] Y. Kim, *E-mail correspondence*, 2013.

[7] T. Lamont, "Napster: The Day the Music Was Set Free," 24 February 2013. [Online]. Available: www.theguardian.com/music/2013/feb/24/napster-music-free-file-sharing. [Accessed 30 October 2013].

[8] Gartner, "Gartner Says Declining Worldwide PC Shipments in Fourth Quarter of 2012 Signal Structural Shift of PC Market," 14 January 2013. [Online]. Available: www.gartner.com/newsroom/id/2301715. [Accessed 9 September 2013].

[9] M. Johas Teener, "No-excuses Audio/Video Networking: the Technology Behind AVnu," 24 August 2009. [Online]. Available: http://avnu.org/wp-content/uploads/2014/05/No-excuses-Audio-Video-Networking-v2.pdf. [Accessed 20 August 2016].

[10] IEEE 802.3, "Website for IEEE 802.3 Residential Ethernet Study Group," 21 November 2005. [Online]. Available: http://grouper.ieee.org/groups/802/3/re_study/. [Accessed 11 October 2013].

[11] R. Kreifeld, *E-mail correspondence*, 2013.

[12] Business Wire, "AVnu Alliance Launches to Advance Quality of Experience for Networked Audio and Video," 25 August 2009. [Online]. Available: www.businesswire.com/news/google/20090825005929/en. [Accessed 8 October 2013].

[13] Marvell, “Marvell Announces Industry-First Audio Video Bridging Family of SoCs with Integrated Switching, CPU and Endpoint Functionality,” 8 May 2012. [Online]. Available: www.marvell.com/company/news/pressDetail.do?releaseID=2296. [Accessed 16 August 2016].

[14] J. Urban, “Debunking Some Myths about AVB,” 17 May 2013. [Online]. Available: http://blog.biamp.com/debunking-some-myths-about-avb/. [Accessed 20 August 2016].

[15] IEEE 802.1, “802.1 Plenary – 11/2012 San Antonio Closing Slides,” November 2012. [Online]. Available: www.ieee802.org/1/files/public/minutes/2012–11-closing-plenary-slides.pdf. [Accessed 30 October 2013].

[16] E. Hellerud, “Transmission of High Quality Audio over IP Networks,” April 2009. [Online]. Available: www.diva-portal.org/smash/get/diva2:277816/FULLTEXT02. [Accessed 1 November 2013].

[17] R. Kreifeld, “AVB for Professional A/V Use,” 30 July 2009. [Online]. Available: http://avnu.org/wp-content/uploads/2014/05/AVnu-Pro__White-Paper.pdf. [Accessed 20 October 2013].

[18] H. Kaltheuner, “Das Universalnetz, Ethernet AVB: Echtzeitfähig und Streaming-tauglich,” *c’t*, no. 13, pp. 176–81, June 2013.

[19] M. Johas Teener, “Residential Ethernet Objectives, Requirements and Possible Solutions,” 9 May 2009. [Online]. Available: www.ieee802.org/1/files/public/docs2005/liaison-mikejt-rese-objectives-requirements-0505.pdf. [Accessed 17 October 2013].

[20] R. Steinmetz, “Human Perception of Jitter and Media Synchronization,” *IEEE Journal on Selected Areas in Communication*, vol. 14, no. 1, pp. 61–72, January 1996.

[21] K. Stanton, “AVB for Home/Consumer Electronics Use,” 11 August 2009. [Online]. Available: http://avnu.org/wp-content/uploads/2014/05/AVB-for-Home-Consumer-Electronics-Use__White-Paper.pdf. [Accessed 24 October 2013].

[22] F. Held, “Digitale Audio-, Video- & Licht-Kommunikationsprotokolle in der Veranstaltungstechnik,” 12 October 2012. [Online]. Available: https://entropia.de/wiki/images/b/b8/Avb_vortrag_felix.pdf. [Accessed 24 October 2013].

[23] M. Johas Teener, *E-mail correspondence*, 2013.

[24] R. Boatright, “Understanding New Audio Video Bridging Standards,” 10 May 2009. [Online]. Available: www.embedded.com/design/connectivity/4027005/Understanding-IEEE-s-new-audio-video-bridging-standards-item-1. [Accessed 24 October 2013].

[25] M. Schettke, “AVB Audionetzwerke in der Praxis: Betriebssicherheit und deren automatisierte Überprüfung,” 10 June 2014. [Online]. Available: http://schettke.com/files/Diplomarbeit_AVB-Praxis_Schettke_2014.pdf. [Accessed 10 August 2016].

[26] R. Kreifeld, “AVB for Automotive Use,” 20 July 2009. [Online]. Available: only updated version available http://avnu.org/wp-content/uploads/2014/05/2014–11–20_AVnu-Automotive-White-Paper_Final_Approved.pdf. [Accessed 20 October 2013].

[27] M. Kicherer and T. Königseder, “BMW Proposal for an AVB Gen 2 Automotive Profile,” BMW White Paper, München, 2013.

[28] K. Matheus, M. Kicherer, and T. Königseder, “Audio/Video Transmission in Cars using Ethernet,” BMW White Paper, Munich, 2010.

[29] T. Hogenmüller et al., “Use Cases & Requirements for IEEE 802.3 RTPGE Ethernet,” May 2012. [Online]. Available: http://grouper.ieee.org/groups/802/3/RTPGE/public/may12/hogenmuller_01_0512.pdf. [Accessed 30 May 2012].

[30] K. Matheus et al., “1 Pair or 2 Pairs for RTPGE: Impact on System Other Than the PHY Part 1: Weight & Space,” January 2013. [Online]. Available: www.ieee802.org/3/bp/public/jan13/matheus_3bp_01_0113.pdf. [Accessed 18 January 2013].

[31] IEEE Computer Society, "IEEE 1722–2016: Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks," IEEE, New York, 2016.

[32] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," July 2003. [Online]. Available: http://tools.ietf.org/html/rfc3550. [Accessed 15 November 2013].

[33] IEEE Computer Society, "IEEE 802.1AS-2011: Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks," IEEE, New York, 2011.

[34] IEEE Computer Society, "IEEE 1588–2008: Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," IEEE, New York, 2008.

[35] Hirschmann, "Precision Clock Synchronization, the Standard IEEE 1588," 2008. [Online]. Available: www.belden.com/docs/upload/precision_clock_synchronization_wp .pdf. [Accessed 28 October 2013].

[36] EndRun Technologies, "Precision Time Protocol (PTP/IEEE-1588)," 2013. [Online]. Available: www.endruntechnologies.com/pdf/PTP-1588.pdf. [Accessed 8 November 2013].

[37] IEEE Computer Society, "IEEE 802.1Q-2011: Media Access Control (MAC) Bridges and Virtual Bridge Local Area Networks," IEEE, New York, 2011.

[38] G. Bechtel, M. K. Ben Gale, and D. Olsen, "Automotive Ethernet AVB Functional and Interoperability Specification, Revision 1.4," AVnu, Beaverton, 2015.

[39] IEEE Computer Society, "IEEE 802.1Qav:2009 Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams," IEEE, New York, 2009.

[40] S. Stein and R. Racu, "Ethernet Quality of Service @ Volkswagen," in *2nd Ethernet&IP@Automotive Technology Day*, Regensburg, 2012.

[41] J. Diemer, D. Thiele, and R. Ernst, "Formal Worst-Case Timing Analysis of Ethernet Topologies with Strict-Priority and AVB Switching," in *7th IEEE International Symposium on Industrial Embedded Systems (SIES12)*, Karlsruhe, 2012.

[42] IEEE Computer Society, "IEEE 1733–2011: Layer 3 Transport Protocol for Time-Sensitive Applications in Local Area Networks," IEEE, New York, 2011.

[43] J. Damori, "Are Layer 2 or Layer 3 Protocols Better? Yes," 31 May 2013. [Online]. Available: http://blog.biamp.com/are-layer-2-or-layer-3-protocols-better-yes/. [Accessed 20 October 2013].

[44] IEEE Computer Society, "IEEE 802.1BA-2011: Audio Video Bridging (AVB) Systems," IEEE, New York, 2011.

[45] IEEE Computer Society, "IEEE 1722.1–2013: Device Discovery, Connection Management, and Control Protocol for IEEE 1722 Based Devices," IEEE, New York, 2013.

[46] J. Lane, "Digital Audio Listening In Car Is Increasing," 11 October 2010. [Online]. Available: http://audio4cast.com/2010/10/11/digital-audio-listening-in-car-is-increasing/. [Accessed 12 November 2013].

[47] M. Jochim and M. Osella, "The Need for IEEE Standardized Ethernet Mechanisms for Active Safety Applications," in *3rd Ethernet&IP@Automotive Technology Day*, Leinfelden-Echterdingen, 2013.

[48] D. Zebralla, "Requirements on Future in-Vehicle Architectures for Automotive Ethernet," in *Automotive Ethernet Congress*, München, 2016.

[49] M. Johas Teener, "IEEE 802 Time-Sensitive Networking: Extending Beyond AVB," in *3rd Ethernet & IP @ Automotive Technology Day*, Leinfelden-Echterdingen, 2013.

[50] D. Pannell, "IEEE TSN Standards Overview & Update," in *5th Ethernet&IP@Automotive Technology Day*, Yokohama, 2015.

[51] IEEE-SA, "IEEE 802.3br-2016 Amendment: Specification and Management Parameters for Interspersing Express Traffic," IEEE, New York, 2016.

[52] IEEE Computer Society, "IEEE 802.1Qbu: Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks – Amendment: Frame Preemption," IEEE, New York, 2016.

[53] Wikipedia, "TTEthernet," 31 March 2016. [Online]. Available: http://en.wikipedia.org/wiki/TTEthernet. [Accessed 20 August 2016].

[54] IEEE Computer Society, "IEEE 802.1Qbv: Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Enhancements for Scheduled Traffic," IEEE, New York, 2016.

[55] IEEE 802.1, "Website for .1Qch – Cyclic Queuing and Forwarding," continuously updated. [Online]. Available: www.ieee802.org/1/pages/802.1ch.html. [Accessed 20 August 2016].

[56] M. Jochim, "Ingress Policing," 10–14 November 2013. [Online]. Available: www.ieee802.org/1/files/public/docs2013/tsn-jochim-ingress-policing-1113-v2.pdf. [Accessed 20 August 2016].

[57] IEEE 802.1, "Website for: 802.1Qci – Per-Stream Filtering and Policing," continuously updated. [Online]. Available: www.ieee802.org/1/pages/802.1ci.html. [Accessed 3 September 2016].

[58] IEEE Computer Society, "IEEE 802.1Qca: Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Path Control and Reservation," IEEE, New York, 2016.

[59] IEEE 802.1, "Website for 802.1CB – Frame Replication and Elimination for Reliability," continuously updated. [Online]. Available: www.ieee802.org/1/pages/802.1cb.html. [Accessed 20 August 2016].

[60] IEEE 802.1, "Website for 802.1AS-Rev – Timing and Synchronization for Time-Sensitive Applications," continuously updated. [Online]. Available: www.ieee802.org/1/pages/802.1AS-rev.html. [Accessed 20 August 2016].

[61] IEEE 802.1, "Website for 802.1Qcc – Stream Reservation Protocol (SRP) Enhancements and Performance Improvements," continuously updated. [Online]. Available: www.ieee802.org/1/pages/802.1cc.html. [Accessed 20 August 2016].

[62] R. Moussalli, "Dealing with security threats in the digital age: Part 1," 19 August 2015. [Online]. Available: http://tatacommunications-newworld.com/2015/08/dealing-with-security-threats-in-the-digital-age-part-1/. [Accessed 29 August 2016].

[63] C. Miller and C. Valasek, "Remote Exploitation of an Unaltered Passenger Vehicle," 10 August 2015. [Online]. Available: http://illmatics.com/Remote%20Car%20Hacking.pdf. [Accessed 5 October 2016].

[64] A. Greenberg, "Remote Exploitation of an Unaltered Passenger Vehicle," 21 July 2015. [Online]. Available: www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/. [Accessed 5 October 2016].

[65] SANS Institute, "The Best Security Books to have in your library," SANS Institute, 2014. [Online]. Available: www.sans.edu/research/book-reviews/article/security-books-best. [Accessed 2 February 2014].

[66] S. Singer, "IP based Communication in Vehicles – learnings from the IT Industry," in *Automotive Ethernet Congress*, Munich, 2016.

[67] AUTOSAR, "Specification of Module Secure Onboard Communication, Release 4.2.2," undated. [Online]. Available: www.autosar.org/fileadmin/files/releases/4-2/software-architecture/safety-and-security/standard/AUTOSAR_SWS_SecureOnboardCommunication.pdf. [Accessed 29 August 2016].

[68] R. Pallierer and M. Ziehensack, “Secure Ethernet for Autonomous Driving,” in *Automotive Ethernet Congress*, München, 2016.

[69] Wikipedia, “Transport Layer Security,” 29 August 2016. [Online]. Available: https://en.wikipedia.org/wiki/Transport_Layer_Security. [Accessed 30 August 2016].

[70] T. Dierks and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2,” August 2008. [Online]. Available: https://tools.ietf.org/html/rfc5246. [Accessed 30 August 2016].

[71] M. Lindner, “Security Architecture for IPsec,” 2007. [Online]. Available: www.ict.tuwien.ac.at/lva/384.081/infobase/L97-IPsec_v4-7.pdf. [Accessed 2 February 2014].

[72] Wikipedia, “IPsec,” 20 July 2016. [Online]. Available: http://en.wikipedia.org/wiki/Ipsec. [Accessed 20 August 2016].

[73] IEEE Computer Society, “IEEE 802.1AEbw-2013: Media Access Control (MAC) Security Amendment 2: Extended Packet Numbering,” IEEE, New York, 2013.

[74] Y. Kim, “Ethernet Security,” in *Automotive Ethernet Congress*, Munich, 2016.

[75] Wikipedia, “Books on Cryptography,” 18 August 2016. [Online]. Available: https://en.wikipedia.org/wiki/Books_on_cryptography. [Accessed 31 August 2016].

[76] L. Apvrille, R. El Khayari, O. Henniger, Y. Roudier, H. Schweppe, H. Seudié, B. Weyl, and M. Wolf, “Secure Automotive on-Board Electronics Network Architecture,” 30 May 2010. [Online]. Available: www.evita-project.org/Publications/AEHR10.pdf. [Accessed 8 October 2016].

[77] H. G. Molter, “Introduction to Security,” in *Automotive Ethernet Congress*, Munich, 2016.

[78] SAE, “Cybersecurity Guidebook for Cyber-Physical Vehicle Systems, J3061,” continuously updated. [Online]. Available: http://standards.sae.org/wip/j3061/. [Accessed 2 September 2016].

[79] ISO, “ISO/AWI 21434 Road Vehicles – Automotive Security Engineering,” 2016. [Online]. Available: www.iso.org/iso/catalogue_detail.htm?csnumber=70918&utm_source=ISO&utm_medium=RSS&utm_campaign=Catalogue. [Accessed 18 November 2016].

[80] H. Goto, “In-Vehicle Ethernet Technology Promoted by JASPAR and Industry Trends,” in *4th Nikkei Electronics/Automotive Seminar*, Tokyo, 2016.

[81] B. Gale, “Ethernet Security in the Car,” in *IEEE-SA (4th) Ethernet&IP@Automotive Technology Day*, Detroit, 2014.

[82] D. Kleidermacher and M. Kleidermacher, *Embedded Systems Security – Practical Methods for Safe and Secure Software and Systems Development*, Oxford: Newnes, 2012.

[83] W. R. Stevens, *TCP/IP Illustrated*, vol. 1, *The Protocols*, Reading, MA: Addison Wesley Longman, 1994.

[84] Wikipedia, “Internet Protocol Suite,” 19 August 2016. [Online]. Available: http://en.wikipedia.org/wiki/Internet_protocol_suite. [Accessed 20 August 2016].

[85] Information Sciences Institute University of Southern California, “Transmission Control Protocol,” September 1981. [Online]. Available: http://tools.ietf.org/html/rfc793. [Accessed 4 August 2013].

[86] Information Sciences Institute University of Southern California, “Internet Protocol,” September 1981. [Online]. Available: http://tools.ietf.org/html/rfc791. [Accessed 4 August 2013].

[87] J. Postel, “User Datagram Protocol,” 29 August 1980. [Online]. Available: http://tools.ietf.org/html/rfc768. [Accessed 4 August 2013].

[88] M. Kessler, “Ethernet in Small ECUs, Challenges and Chances,” in *3rd Ethernet&IP@Automotive Technology Day*, Leinfelden-Echterdingen, 2013.

[89] 3Com, “Understanding IP Addressing, Everything You Ever Wanted to Know,” 2001. [Online]. Available: http://pages.di.unipi.it/ricci/501302.pdf. [Accessed 20 August 2016].

[90] Number Resource Organization, “Free Pool of IPv4 Address Space Depleted,” 3 February 2011. [Online]. Available: www.nro.net/news/ipv4-free-pool-depleted. [Accessed 23 November 2013].

[91] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6),” December 1995. [Online]. Available: http://tools.ietf.org/html/rfc1883. [Accessed 23 November 2013].

[92] C. Bao, C. Huitema, M. Bagnulo, M. Boucadair, and X. Li, “IPv6 Addressing of IPv4/IPv6 Translators,” October 2010. [Online]. Available: http://tools.ietf.org/html/rfc6052. [Accessed 18 February 2014].

[93] A. Busnelli, “Car Software: 100M Lines of Code and Counting,” 26 June 2014. [Online]. Available: www.linkedin.com/pulse/20140626152045–3625632-car-software-100m-lines-of-code-and-counting. [Accessed 18 August 2016].

[94] The Apache Software Foundation, “Apache Etch,” 2013. [Online]. Available: http://etch.apache.org/. [Accessed 19 February 2014].

[95] A. Bouard, J. Schanda, D. Herrscher, and C. Eckert, “Automotive Proxy-Based Security Architecture for CE Device Integration,” in *Mobile Wireless Middleware, Operating Systems, and Applications*, Heidelberg: Springer, 2013, pp. 62–76.

[96] K. Weckemann, F. Satzger, L. Stolz, D. Herrscher, and C. Linnhoff-Popien, “Lessons from a Minimal Middleware for IP-Based in-Car Communication,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Alcala de Henares, 2012.

[97] Google Developers, “Protocol Buffers,” 2 April 2012. [Online]. Available: https://developers.google.com/protocol-buffers/. [Accessed 30 January 2014].

[98] Apple, “Bonjour for Developers,” 2014. [Online]. Available: https://developer.apple.com/bonjour/. [Accessed 30 January 2014].

[99] L. Völker, “Scalable service-Oriented Middleware over IP (SOME/IP),” 2013. [Online]. Available: http://some-ip.com. [Accessed 30 January 2014].

[100] Napster, “Company Info,” continuously updated. [Online]. Available: http://us.napster.com//about?from=rhapsody. [Accessed 15 August 2016].

[101] T. Hümmler, “Router, Switches, Hub und Co. Ratgeber: Was ist im Netzwerk?,” 26 October 2013. [Online]. Available: www.tecchannel.de/netzwerk/lan/2038788/router_repeater_switch_hub_bridge_was_ist_was_im_netzwerk/. [Accessed 18 October 2013].

[102] B. Popper, “Rhapsody Rebrands Itself as Napster because ¯_(ツ)_/¯,” *The Verge*, 14 June 2016. [Online]. Available: www.theverge.com/2016/6/14/11936974/rhapsody-rebrands-as-napster. [Accessed 15 August 2016].

[103] ISO/IEC, “ISO/IEC 29341- (1–13)-1:2008: Information Technology – UPnP Device Architecture,” ISO, Geneva, 2008.

[104] Open Connectivity Forum (OCF), “About UPnP,” 1 January 2016. [Online]. Available: https://openconnectivity.org/upnp. [Accessed 23 August 2016].

[105] DLNA, “About DLNA,” 2013. [Online]. Available: www.dlna.org/dlna-for-industry/about-dlna. [Accessed 1 November 2013].

[106] P. Klein, “Support for Coordinated Shared Network in 802.1AVB,” January 2008. [Online]. Available: www.ieee802.org/1/files/public/docs2008/av-phkl-csn-0108-v1.pdf. [Accessed 1 November 2013].

[107] Musical Instrument Digital Interface Manufacturers Association (MIDI), “IEEE Ethernet AVB, AVB and MIDI,” 2013. [Online]. Available: www.midi.org/techspecs/avbtp.php. [Accessed 2 November 2013, no longer available].

[108] M. Felser and T. Sauter, "Standardization of Industrial Ethernet – the Next Battlefield?," in *Proceedings of the IEEE 5th International Workshop on Factory Communication Systems*, Vienna 22 September 2004.

[109] Wikipedia, "Jumbo frame," 16 September 2016. [Online]. Available: https://en.wikipedia.org/wiki/Jumbo_frame. [Accessed 5 October 2016].

[110] D. C. Plummer, "An Ethernet Address Resolution Protocol," November 1982. [Online]. Available: http://tools.ietf.org/html/rfc826. [Accessed 21 November 2013].

[111] J. Postel, "Internet Control Message Protocol," September 1981. [Online]. Available: http://tools.ietf.org/html/rfc792. [Accessed 21 November 2013].

[112] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, "Internet Group Management Protocol, Version 3," October 2002. [Online]. Available: http://tools.ietf.org/html/rfc3376. [Accessed 21 November 2013].

[113] J. Mogul and J. Postel, "Internet Standard Subnetting Procedure," August 1985. [Online]. Available: www.ietf.org/rfc/rfc950.txt. [Accessed 23 November 2013].

[114] R. Braden and J. Postel, "Requirements for Internet Gateways," June 1987. [Online]. Available: www.ietf.org/rfc/rfc1009.txt. [Accessed 23 November 2013].