IEEE 802.1AS Time Synchronization in a switched Ethernet based In-Car Network

Hyung-Taek Lim, Daniel Herrscher
BMW Group Research and Technology
Munich, Germany
Email: {hyung-taek.lim,daniel.herrscher}@bmw.de

Lars Völker
BMW Group
Munich, Germany
Email: lars.voelker@bmw.de

Martin Johannes Waltl
Technische Universität München
Munich, Germany
Email: martin.waltl@mytum.de

***Abstract*—As of today Ethernet is used in the in-vehicle network mainly for two use cases: connectivity between the head unit and the rear seat entertainment (RSE) as well as faster on-board diagnostics (OBD). With the increasing bandwidth demand in driver assistance and the wish to easier interconnect the driver assistance and infotainment domains additional usage of Ethernet in the vehicle is being examined. The legacy Ethernet does only provide very limited Quality-of-Service (QoS) mechanisms so that demanding real-time in-vehicle applications cannot meet their constraints. The Audio/Video Bridging (AVB) group introduced several IEEE standards to allow audio and video applications with high QoS demands in a switched Ethernet network. Although these mechanisms were not designed for automotive use cases, they are good extensions to switched Ethernet when QoS demands exist. Therefore, an evaluation of AVB for the usage in in-vehicle networks is needed.**

In this work, we focus on a base mechanism of the IEEE 802.1 AVB standard, the IEEE 802.1AS time synchronization protocol and its usage in the in-vehicle network. The evaluation is performed by simulation with the network simulation tool OMNeT++ and we modifed the INET-framework with the IEEE 802.1AS capability for our purpose.

I. Introduction

A current vehicle consists of up to 70 electronic control units (ECUs) with different functionalities interconnected by specific automotive bus systems like CAN, FlexRay, MOST, and LIN [1], [2]. While CAN and FlexRay are designed for closed control loop applications with hard real-time applications but small bandwidth demand, MOST is used for synchronous transport of audio and video streams in the infotainment domain and was designed to satisfy higher bandwidth demands and real-time requirements. Other solutions include specialized sensor busses (e. g. LIN, PSI5, and SENT) and video connections (e. g. analog, LVDS, and APIX). However, the automotive industry currently looks at other communication solutions to reduce the number of different communication technologies used that allow a more flexible architectures. A valid candidate for achieving these goals is Ethernet that is currently only used in two areas within a vehicle: the connectivity between the head unit and the rear seat entertainment system (RSE) as well as the on-board diagnostic (OBD). The breakthrough for more Ethernet usage inside the vehicle is based on a new unshielded 100 Mbit/s physical layer that is able to satisfy the demanding electromagnetic compatibility (EMC) requirements of the vehicle manufactures, while significantly reducing the cable and connector cost [3]. From the technical point of view the legacy switched Ethernet does only provide limited QoS mechanisms which is the weak point in comparison to MOST. At least prioritization mechanism as defined in the IEEE 802.1Q standard [4] can be used to guarantee the highest traffic class [5], [6]. The Audio Video Bridging (AVB) task group added several IEEE standards to achieve the QoS requirements for low latency streaming in Ethernet networks [7]. The following AVB standards exist today:

- IEEE 802.1AS—Timing and synchronization protocol for time-sensitive applications
- IEEE 802.1Qat—Stream reservation protocol
- IEEE 802.1Qav—Forwarding and queuing rules for time-sensitive applications
- IEEE 1722/1733—Layer2/Layer3 transport protocol for time-sensitive applications
- IEEE 802.1BA—Profiles for Audio Video Bridging systems

While low latency audio and video streaming is not the only use case for in-vehicle communications, the mechanisms described can be also used as basis for other applications. One important AVB building block is the high-precision clock synchronization protocol IEEE 802.1AS that lays the basis for the other AVB mechanisms defined. For that reasons, we focus on the IEEE 802.1AS synchronization mechanisms[1] in respect to different clock drifts in the in-vehicle network. Furthermore, we consider a specific in-car scenario with typical applications in a daisy-chain based topology.

This paper is structured as follows: In Section II, we present existing performance studies of the IEEE 802.1AS protocol, other time synchronization protocols, and highlight the differences to our work. Section III describes briefly the IEEE 802.1AS standard. In Section IV our simulation model and the assumed network architecture for the simulation based performance evaluation are presented. Section V concludes our work and sketches our future research path.

II. Related Work

There are prior evaluation works of different network time synchronization protocols such as NTP (Network Time

[1]The current work is based on the IEEE 802.1AS D7.7 Draft standard [8]

978-1-4673-0047-6/11/$26.00 ©2011 IEEE

Protocol), IEEE 1588 Precision Time Protocol (PTP), and IEEE 802.1AS. While NTP does not meet the high accuracy requirement of the clock frequency and crystal oscillator [9], [10], IEEE 1588 PTP and IEEE 802.1AS are able to provide a better performance.

G. Garner et al. analyzed the IEEE 802.1AS protocol in a daisy chain based topology with 7 hops by determining the influence of the endpoint filter bandwidth to the maximum time interval error (MTIE) [11], [12]. The MTIE is a peak-to-peak phase variation for an interval length called observation interval and represents the quantities of jitter and wander. In the work, they showed that the given audio and video streaming applications fulfill all the requirements in terms of the MTIE with the use of 1 MHz filter bandwidth, while the performance is decreased due to the increased filter bandwidth. However, the presented work analyzed the synchronization protocol mainly in a daisy chain based topology with 7 hops by determining the MTIE for video and audio applications. We also verify the daisy chain based topology with 7 hops, but the main difference is that we consider the synchronization accuracy depending on different network load. Furthermore, we analyze an in-car scenario with the typical applications where the number of switches in a network is limited.

K. Sunghwan et al. [13] improved the 802.1AS synchronized time accuracy up to 5% by using single time synchronization frames which considered the propagation delay symmetry and physical coding sublayer (PCS) clock counter. The proposed concept requires a modification of a physical layer (PHY) in an IEEE 802.3 Ethernet standard to invoke the PHY and transmits the synchronization messages to the time-stamp agent.

III. Time Synchronization based on IEEE 802.1AS

The IEEE 802.1AS standard [8] specifies a protocol to synchronize distributed nodes to establish a common reference time in an IEEE 802 network. A single node in a network has the best clock determined by a best master clock algorithm (BMCA, see Sect. III-B) which distributes the clock information to all IEEE 802.1AS capable nodes. The protocol is able to support different network technologies such as IEEE 802.3 Ethernet, EPON, and IEEE 802.11 WLAN but the transport of synchronization messages depends on the network media due to the different time-stamp mechanisms. In this work we want to analyze the IEEE 802.1AS in an Ethernet based in-vehicle network so we focus on the mechanisms of IEEE 802.3 full-duplex point-to-point links.

IEEE 802.1AS uses generalized precision time protocol (gPTP) to send synchronization information based on a PTP profile as specified in an IEEE 1588-2008 standard [14]. After the synchronization process, a gPTP domain is build which consists of only gPTP capable nodes.

A. Best Master Clock Algorithm

The best master clock algorithm (BMCA) selects the best clock of the gPTP domain as grandmaster (GM) and determines a spanning tree for synchronization with the grandmaster as its root. It uses *announce* messages with the port identity vector which identifies time-aware systems that support the IEEE 802.1AS protocol. After the BMCA is finished all ports of the time-aware systems are assigned to one of the following four port states.

- Master port—any port of a time-aware system which is the closest to the root from the view of a subsequent system in a spanning tree. The synchronization messages are transmitted along the spanning tree.
- Slave port—one port of a time-aware system which is the closest to the root of the spanning tree. At this port, synchronization messages are received. The GM does not have assigned slave ports.
- Disabled port—any port of the time-aware system which is not IEEE 802.1AS capable.
- Passive port—any port of the time-aware system which is neither master port or slave port nor a disabled port.

The process of the BMCA is always finished with the creation of spanning tree, where the root of the tree is mostly the grandmaster which distributes the synchronization information. In a case when the root is not a grandmaster, none of the time-aware systems in a gPTP domain is grandmaster capable so that a synchronization process will not start. Figure 1 shows an example result of a gPTP domain after the BMCA was performed, where all ports are assigned to a role. The synchronization tree is the spanning tree and the synchronization information is only transmitted from the master port to the slave port between two time-aware systems.

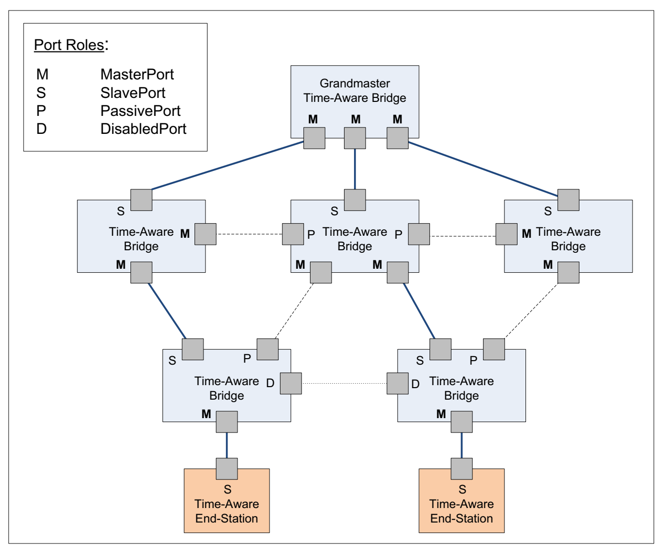


Fig. 1. Resulting spanning tree after performing BMCA

B. Synchronization

After a grandmaster is selected and a synchronization tree is build, the synchronization process is started by the grandmaster. The timing information is only sent from a grandmaster to the directly connected systems and they adjust the synchronization information by adding the propagation delay and the residence time. If the connected system is a time-aware bridge, it forwards the corrected synchronization information along the tree determined by the BMCA. The residence time is the forwarding delay which is needed by a time-aware bridge to transmit a time synchronization to the next subsequent one (see Fig. 3).

A propagation delay is the time taken by a transmitted message between two directly connected time-aware systems (see Fig. 3). This delay is measured in each port of every full-duplex point-to-point link by the peer delay mechanism which is based on a request-response approach (see Fig. 2).

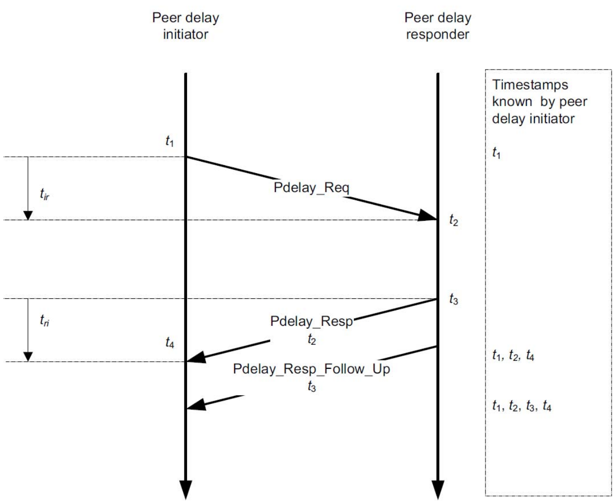


Fig. 2. Propagation delay using peer-delay mechanism over IEEE 802.3 Ethernet [8]

In a first step, the peer delay initiator starts the measurement by sending a *Pdelay_Req* message while the responder replies with a *Pdelay_Resp* message which contains the timestamp t_2. In addition, a *Pdelay_Resp_Follow_Up* is sent from the responder to the initiator to transmit the timestamp t_3. In a final step, the peer delay initiator knows all timestamps and calculates the mean propagation delay with Equation (1), where r represents the clock drift of the responder[2].

$$Pdelay = \frac{(t_4 - t_1) - r \cdot (t_3 - t_2)}{2} \ . \tag{1}$$

The correction of a synchronization information with the residence time and propagation delay is necessary to perform a logical syntonization[3] between the systems in a network. In reality, the local clocks mainly the crystal oscillators of two different systems are never syntonized due to the physical differences like temperature and vibrations. The local clocks have a frequency variation defined as a clock drift which results in different time values between distributed network systems. Each time-aware system also determines the clock drift of all adjacent systems defined as a neighbor rate ratio.

The synchronization information is distributed to the gPTP network by using two successive messages: *Sync* and *Follow_Up*. These messages are sent periodically from each master ports which are received on the slave ports of the time-aware systems. Figure 3 shows a synchronization process over IEEE 802.3 full-duplex point-to-point links between three time-aware systems, where *system A*, *system C* are end nodes and *system B* represents a bridge.

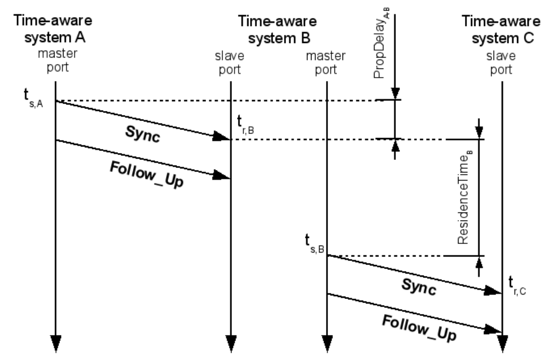


Fig. 3. Synchronization process initiated by grandmaster

A *time-aware system A* sends a synchronization (Sync) messages by time-stamping the information at an egress port at time $t_{s,A}$. *Time-aware system B* receives the message and time-stamps it at an ingress port at time $t_{r,B}$. After a Sync message is sent at a master port to *time-aware system B*, a *Follow_Up* message is transmitted with the following information.

- Precise origin time-stamp from the grandmaster (preciseOriginTimestamp).
- Correction information updated by each time-aware system (correctionField) which includes the propagation delay and the residence time.
- Frequency rate ratio relative to the grandmaster clock (rateRatio).

These information are required to correct the synchronization information in each intermediate and end systems. Finally all time-aware systems are synchronized based on the IEEE 802.1AS protocol and all systems are able to know the accumulated grandmaster rate ratio, where the clock drift between the grandmaster and the own local clock is given.

The IEEE 802.1AS protocol specifies a synchronization accuracy of less or equal than $1\mu s$ between seven or fewer time-aware systems [8] which has to be analyzed in details. We examined the following aspects which could influence the synchronization accuracy.

- Synchronization interval—the default value of the synchronization interval on IEEE 802.3 full-duplex links is 0.125s [8]. The synchronization interval influences the timing error of the time-aware systems. A lower synchronization interval leads to better synchronization accuracy but at the same time higher network load.
- Averaging algorithm of the propagation delay—The propagation delay determined by the peer delay mechanism is not precise due to the limited clock accuracy and time measurement granularity. A clock measures the time in intervals, so that a clock will return a time truncated to the clock accuracy interval. An inaccuracy of a propagation

[2]There is no link asymmetry on IEEE 802.3 full-duplex links.

[3]Syntonization means the synchronization of frequency but not of phase.

delay is occurred only if the last measured values are used for the propagation delay. This effect can be reduced by applying an average filter for successive measured propagation delays. An averaging filter would improve the measurement accuracy and will have less variety over unchanged links. The IEEE 802.1AS standard does not force an averaging computations but recommends the use of an appropriate algorithm.
- Network load—Although a synchronization is performed only between two time-aware systems, the IEEE 802.1AS messages have to be queued in an output queue of each time-aware system. The IEEE 802.1AS standard specifies that frames carrying these messages are neither VLAN-tagged nor priority-tagged [8], but they have an internal priority value for inserting them into the output queue. In case of a high network load, the frames are delayed by a queuing time which can influence the accuracy of a delay measurement and synchronization.

In the following section, we evaluate the influence of these aspects and analyze the applicability of the IEEE 802.1AS in a specific in-car scenario with typical applications.

IV. Performance Analysis of IEEE 802.1AS

A. Requirements and Assumptions

For a performance analysis of the IEEE 802.1AS protocol in a switched Ethernet based in-car network, there are some requirements and assumptions which are listed as follows.

- Static configuration of ECUs/no BMCA—An in-car network is a closed network where all ECUs are configured statically and do not change in their life-time. For this reason, the BMCA is not necessary in a vehicle, so that a node with the best clock in a network (e.g. the head unit) is set manually. The port roles and the grandmaster are assigned by a static configuration.
- No external timing source—The IEEE 802.1AS standard allows to request timing information by an external timing source (ClockSource). In a vehicle, an external timing source is not required, so that the clock master obtains the time from the local clock of a time-aware system.
- No calculations for phase and frequency discontinuity—In a gPTP network, time-aware systems are able to compute the discontinuity of their own clock to the grandmaster. Furthermore, the IEEE 802.1AS protocol measures the phase and frequency discontinuity and with these information the grandmaster in a network is changed. We assume an in-car network with a single grandmaster which is not changed during the time, hence there is no need for this calculations in our simulation model.

B. Metrics

The following metrics are used to analyze the IEEE 802.1AS standard.

- Grandmaster rate ratio—The grandmaster rate ratio is the clock drift from a given system related to the grandmaster determined by an accumulated neighbor rate ratio of uplink ports. It reflects the correct forwarding of a synchronization information and accumulated accuracy of a neighbor rate ratio.
- Propagation delay—The propagation delay is measured to identify the influence of a network load to the peer delay mechanism and the convenience of an averaging filter.
- Synchronization error (time difference to grandmaster)—The synchronization error is determined in each network node by calculating the absolute time difference between the grandmaster and its node. The synchronized time and its error reflect the correct detection of link propagation delay, grandmaster rate ratio and computation of a correction field.

C. Simulation Model

We use the OMNeT++ simulation tool [15] with the INET-framework [16] for the performance analysis. The IEEE 802.1AS architecture divides a time-aware system in a media-dependent and media-independent entity, where all components are placed at Layer-2 (link layer) above the medium access control (MAC) layer. Figure 4 illustrates our architecture in an INET-framework.

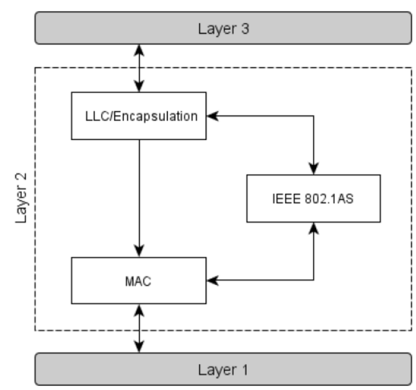


Fig. 4. IEEE 802.1AS simulation model at Layer-2

1) Passive clock: One of the great challenges in a discrete event based simulation environment is an asynchronism of distributed nodes in a network. Usually, all nodes and systems in a network have an unique global simulation time called *simtime* which are synchronized by this time. According to IEEE 802.1AS, each time-aware system has an own free running clock. For this reason, we modified the INET-framework for supporting a local clock of a node so that certain asynchronism between the nodes can be modeled. We choose a passive clock implementation to avoid the influence to the simulation performances. A passive clock operates and determines the time only by a request, where following parameters can be set to configure the asynchronism.

- Clock frequency / tick interval—The speed of a clock is determined by its tick interval. IEEE 802.1AS specifies a clock which requires an tick interval of at least 40ns or 25MHz clock frequency.

- Tick interval asynchronism—In a simulation framework all modules are initialized at simulation start. We use a tick interval asynchronism value to set a phase discontinuity between two clocks.
- Total time offset—When a network node is started up the own local clock will start in a simulation environment. In reality, two devices are not started up exactly at the same time. We set this parameter to have an offset in each clock. This is added when the time is requested.
- Clock drift—A real inaccuracy in a crystal oscillator is occurred by a clock drift of a frequency variance because the clock does not run at an exact speed. Usually, the speed of an oscillator is specified at a typical temperature of $25\,^{\circ}\mathrm{C}$. In a case when an oscillator is operated in an environment which differs from the ideal temperature, a constant drift occurs and this values is set as a parameter. In reality, additional clock drift over a time occurs due to the temperature changes, but this effect is not considered in our simulation model.

2) BMCA configuration file: We do not use the BMCA so the port roles of all time-aware systems in a network and the grandmaster are set by an external XML-configuration file. Listing 1 shows a configuration example of two end-systems connected by a time-aware bridge, where Host_A is the grandmaster.

```
<networkconfiguration>
  <entity name="Host_A">
    <ports>
      <port number="1" role="Master"/>
    </ports>
    <isGrandMaster>true</isGrandMaster>
  </entity>
  <entity name="Host_B">
    <ports>
      <port number="1" role="Slave"/>
    </ports>
    <isGrandMaster>false</isGrandMaster>
  </entity>
  <entity name="Switch">
    <ports>
      <port number="1" role="Slave"/>
      <port number="2" role="Master"/>
    </ports>
    <isGrandMaster>false</isGrandMaster>
  </entity>
</networkconfiguration>
```

Listing 1. BMCA configuration file

D. Performance evaluation

The performance evaluation of the IEEE 802.1AS protocol is divided into two parts. The first part contains an analysis of the peer delay measurement and the synchronization process as specified in the standard, while the latter one analyzes the standard with given in-car applications in a daisy-chain based topology. The given topology of an Ethernet based in-car network with the traffic characteristics is derived from a previous work [6]. We use the following assumptions for the entire simulation based performance evaluation.

- Priority mechanism with two schedulers: The Ethernet frames are classified into four different priorities. The traffic class with the highest priority value is scheduled with a strict priority queuing while other priorities are scheduled with weighted fair queuing (WFQ) using the weight values of 100, 10 and 1 for other priority values.
- Network transmission media: IEEE 802.3 full-duplex point-to-point links with 100 Mbit/s.
- MAC transmission queue size: 100 frames.
- Switch processing time: $3\mu s$.

1) Peer delay measurement: The peer delay mechanism is executed in a media-dependent entity in each port of a time-aware system to determine the propagation delay of a link and a clock drift to the neighbor systems (neighbor rate ratio). We define a simple network with only two time-aware systems Host_A and Host_B connected by a link with a propagation delay of 40ns, where a peer delay interval is set to 1s and a tick interval to 40ns (25 MHz clock). We used two different averaging algorithms to determine the propagation delay. The first approach is an arithmetic average filter which use the last 30 samples for its averaging, while the latter is a smooth average filter. The second filter calculates an average with the last 30 samples of previous average delay computations (see Fig. 5). The results show that the smooth averaging filter is stable against measurement inaccuracy of a clock which depends on a tick interval. Furthermore, the propagation delays are converged to the real propagation delay of 40ns within a range of $\pm$10ns.

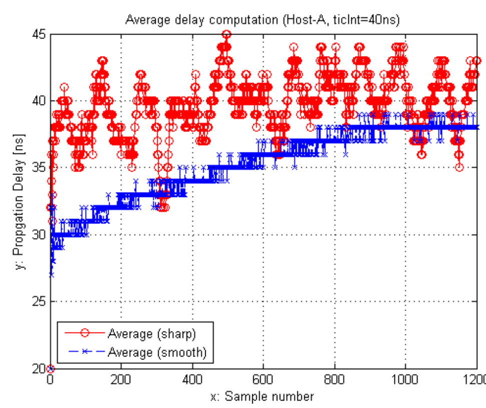
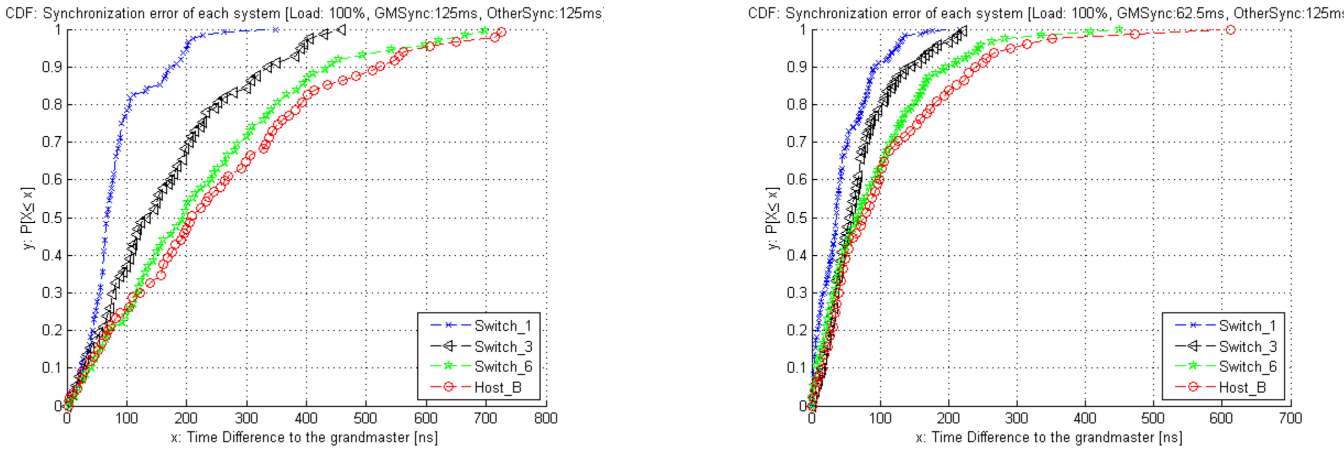


Fig. 5. Comparision of averaging algorithms for a peer delay measurement

2) Synchronization process: We analyze the synchronization process of time-aware systems in a daisy-chain based topology with six switches in a network. The daisy-chain based topology with six switches represents a special synchronization tree for which performance requirements are specified in the IEEE 802.1AS standard. It specifies a maximum synchronization accuracy of 1μs over six or less time-aware systems which are analyzed in this work. Furthermore, we consider additional background traffic in a network to determine the influence of a network load to the synchronization process and its accuracy. In this scenario, we introduce traffic source and sink nodes at each switch. At each side of a link between two time-aware systems additional traffic generators and receivers are connected (see Fig, 6), where the ports of the switches

TABLE I
MAXIMUM TIME DIFFERENCE OF A TIME-AWARE SYSTEM COMPARED TO THE GM-TIME (SYNC-INTERVAL: 125MS / 62.5MS)

Additional Load by Control Data [%]	Synchronization error [ns]						
	Switch_1	Switch_2	Switch_3	Switch_4	Switch_5	Switch_6	Host_B
0	355 / 184	420 / 238	547 / 304	600 / 315	783 / 309	773 / 240	939 / 253
25	344 / 155	442 / 324	695 / 385	611 / 286	780 / 321	733 / 356	835 / 333
50	304 / 160	569 / 407	527 / 226	642 / 276	971 / 296	936 / 160	985 / 305
75	349 / 216	463 / 304	459 / 223	519 / 375	764 / 463	699 / 449	852 / 613



(a) Sync interval of all systems: 125ms

(b) Sync interval of GM: 62.5ms, Other systems: 125ms

Fig. 7. Synchronization error in a high load situation (Network load = 100%)

connected to the traffic generators are set as *DisabledPort* and they do not participate to the synchronization process.

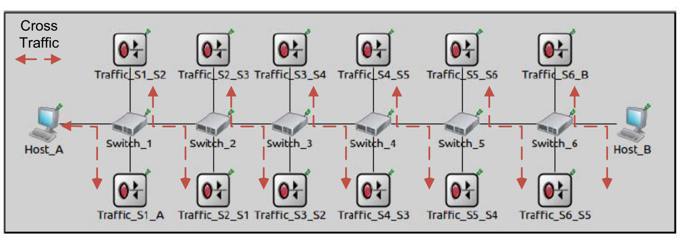


Fig. 6. Synchronization with cross traffic

We use two different traffic classes in a network. The first class contains only audio/video data with a constant bandwidth demand of 25 Mbit/s [6] and the lowest priority. This traffic class is the base traffic in a network whose data are always transmitted. The second traffic class is control data with a payload size of 100 byte which has the highest priority and the same priority value as gPTP messages. The control data frames are transmitted up to a data rate of 75 Mbit/s in a high network load situation where the total network load is increased to 100%. In this situation a delay of IEEE 802.1AS messages in a priority queue is expected. We analyze the synchronization mechanism by determining the influence of a synchronization interval and the network load. The results in Table I shows that the synchronization interval has a significant influence to the synchronization error and its accuracy. A synchronization interval of 62.5ms of a grandmaster improves the accuracy to approximately 50% compare to a case of a 125ms interval. All maximum time differences related to the GM-time are less than the end-to-end performance requirement of 1μs for seven hops in a worst case situation, when high network load exist (see Fig. 7).

The reasons for a stable and accurate result are explained as follows. At first, the grandmaster rate ratio is well estimated which enables a good correction capability at each time-aware system. If a neighbor rate ratio is used for the calculation, the accumulated grandmaster rate ratio is highly accurate to compute the synchronized time. Secondly, the *Sync* and *Follow_Up* messages contain synchronization interval information of an upstream port which is read on receipt and stored at an opposite port. If a time-aware system does not receive three consecutive synchronization messages, it assumes that the grandmaster does not operate accurate and performs the BMCA. In our scenario, we use control data with the same priority values as IEEE 802.1AS messages, so certain delays would occur in a worst case. A control data frame with a payload size of 100 byte has a total transmission size of 138 byte where a Ethernet header, preamble, SFD and IFG are included. In case of a synchronization interval T_{sync} of 62.5ms and a bit transmission time T_{bitTx} of 10ns for Fast-Ethernet, the required number of transmitted control frames to restart a network configuration n_{txCTRL} is calculated with the information of a timeout interval for synchronization messages $T_{syncRxTimeout}$ and a transmission time of a control frame T_{txCTRL} (see Eq. 2). The calculation shows at least 16984 control messages have to be transmitted between two successive synchronization messages to restart the network configuration with the BMCA. This would be in a case when a network load of 100% is only occurred by the highest traffic class which is not a realistic scenario.

$$
\begin{aligned}
n_{txCTRL} &= \frac{T_{syncRxTimeout}}{T_{txCTRL}} \\
&= \frac{n_{synchRxTimeout} \cdot T_{sync}}{l_{CTRL} \cdot T_{bitTx}} \\
&= \frac{3 \cdot 62.5 \cdot 10^{-3} s}{138 \cdot 8bit \cdot 10 \cdot 10^{-9} \frac{s}{bit}} \\
&\approx 16984
\end{aligned} \tag{2}
$$

3) Analysis of a daisy-chain based in-car network: The following network represents a switched Ethernet based in-car network with typical applications and IEEE 802.1AS support (see Fig. 8). It consists of several ECUs, where the head unit operates as a grandmaster and performs the synchronization process. The given daisy-chain based topology with the traffic characteristics of a network are derived from our previous work [6]. The goal of the analysis is to determine if IEEE 802.1AS is suitable for a synchronization and useful in a switched Ethernet based in-car network. In addition, it should identify the influence of a synchronization process to the provided in-car applications. Each node in a network has

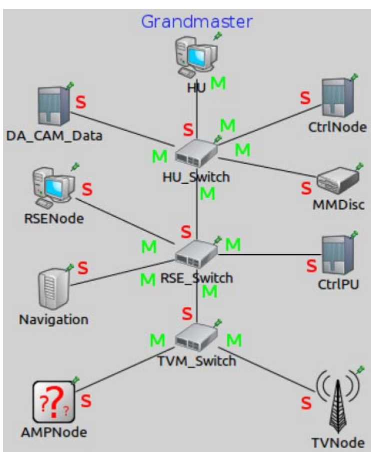


Fig. 8. Switched Ethernet based in-car network with IEEE 802.1AS

an own local clock with certain clock drift relative to the grandmaster which are described in Table II. We assume a cable segment length between two systems of 5m, so that a propagation delay is set to a fixed value of 25ns. Two different synchronization intervals (62.5ms, 125ms) are selected for the GM while a fixed interval of 125ms is set for other time-aware systems in a network.

As in the previous analysis, the synchronization interval of a GM has a significant influence to the synchronization accuracy (see Tab III). The maximum synchronization error is reduced to approximately 50% by using shorter GM synchronization interval. Furthermore, we can see that the provided in-car applications with different priority classes do not influence the synchronization process so that all of the synchronization errors are less than the maximum specifed value of 1μs.

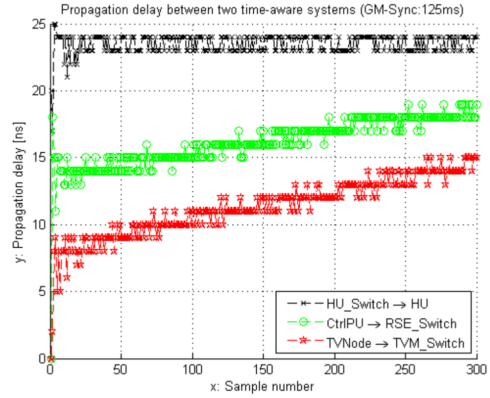


Fig. 9. Propagation delay measurement between two nodes

In case of a propagation delay measurement, the accuracy of approximated propagation delay values strongly depend on the averaging filter due to the time measurement granularity of a local clock which is 40ns and a fixed propagation delay value of 25ns. The measurement of the propagation delay can result in 0 or 40ns so that the accuracy of approximated values depend strongly on the filter. Figure 9 shows a propagation delay measurement between time-aware nodes and switches, where the delays converge to a value of 25ns $\pm$ 10ns. Although the peer delay mechanism suffered from the short propagation path, the grandmaster rate ratio is computed highly accurate within a specified range of $\pm$ 0.1 PPM (see Tab. III).

V. Summary and Future Work

In this work we have presented results of our IEEE 802.1AS simulation models, which we build to analyze and verify the synchronization performance of the IEEE 802.1AS standard in different scenarios. The evaluation of the IEEE 802.1AS standard in different scenarios shows:

- The synchronization process based on IEEE 802.1AS is not influenced in high network load situations. The synchronization error and its accuracy remains below than the specified value of 1μs over seven or fewer hops.
- The grandmaster synchronization interval has a great influence to the performance in terms of accuracy and errors. An accurate and improved synchronization is only achieved when the synchronization intervals of the grandmaster is shorter than all of other time-aware systems. Even if the synchronization interval is very small, the required bandwidth demand of the IEEE 802.1AS protocol is negligible small. An estimated bandwidth demand of 0.027 Mbit/s per link is required in case of the grandmaster synchronization interval of 62.5ms.
- An averaging filter for the propagation delay measurement improves the clock accuracy and the synchronization performance dramatically. The use of an averaging filter is highly recommended.
- In an in-car network, the provided applications do not influence the synchronization process, the performance of IEEE 802.1AS and vice versa.

TABLE II
CLOCK DRIFTS OF EACH NETWORK NODE IN AN IN-CAR NETWORK

Clock drift of a time-aware system [ppm]											
HU	HU Switch	DA_CAM Data	CtrlNode	MMDisc	RSE Switch	RSENode	CtrlPU	Navigation	TVM Switch	AMPNode	TVNode
0	30	-35	-50	10	-15	-15	50	40	20	-50	-5

TABLE III
SYNCHRONIZATION ERROR AND GRANDMASTER RATE RATIO (GM-SYNCINTERVAL: 125MS / 62.5MS)

Metric	HU Switch	DA_CAM Data	CtrlNode	MMDisc	RSE Switch	RSENode	CtrlPU	Navigation	TVM Switch	AMPNode	TVNode
Max. time Difference in [ns]	340 / 191	580 / 255	542 / 249	512 / 319	460 / 333	624 / 326	559 / 282	678 / 323	573 / 301	674 / 475	811 / 445
Avg. GM rate ratio in [ppm]	-29.45 / -29.30	34.44 / 34.47	49.49 / 49.53	-9.60 / -9.50	14.59 / 14.52	14.43 / 14.51	-49.47 / -49.49	-39.49 / -39.52	-19.35 / -19.52	49.55 / 49.55	4.60 / 4.51

The results obtained by the simulations were as predicted by the related work and the IEEE 802.1AS publications. We conclude that our simulation models are close to the real protocols and thus useful in simulating vehicle networks using IEEE 802.1AS. Our next steps include:

- Evaluating scenarios that include nodes and switches without hardware support.
- Evaluate the IEEE 802.1AS protocol with AVB hardware prototypes to further verify our simulation results. The influence of the environment values such as temperature and pressure to clock drifts and synchronization accuracy is determined by using the real world experiments. With these information, we are able to build a statistical model which enables an optimized and increased accuracy of the simulation model.
- Evaluate IEEE 802.1AS in combination with other mechanisms as specified in the AVB standard. We will model and evaluate an AVB network with all the specified protocols with typical provided in-car applications.

The goal of the further evaluation is to reach a deep understanding whether AVB is an appropriate candidate for the use in a in-vehicle network.

ACKNOWLEDGMENT

Some of the research presented here, took place within the project SEIS – Security in Embedded IP–based Systems [17]. SEIS explores the usage of the Internet Protocol (IP) as a common and secure communication basis for electronic control units in vehicles. The project is partially funded by the German Federal Ministry of Education and Research (BMBF) (support codes 01BV0900 – 01BV0917). We would like to thank all SEIS partners directly or indirectly involved in our research.

REFERENCES

[1] T. Nolte, H. Hansson, and L. L. Bello, "Automotive Communications – Past, Current and Future," in *Proceedings of 10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'05)*, Sep. 2005, pp. 985–992 (volume 1).

[2] R. Freymann, "Anforderungen an das Automobil der Zukunft," The 2nd Mobility Forum, Munich. [Online]. Available: http://www.munichnetwork.com/fileadmin/user_upload/konferenzen/mobilitaetsforum-2/071128MUN_Prof_Freymann_Raymond.pdf

[3] R. Bruckmeier, "Ethernet for Automotive Applications," 2010, freescale Technology Forum, Orlando. [Online]. Available: http://www.freescale.com/files/ftf_2010/Americas/WBNR_FTF10_AUT_F0558.pdf

[4] IEEE, "IEEE Standard for Local and Metropolitan Area Networks Virtual Bridged Local Area Networks," *"IEEE Std 802.1Q-2005"*, 2006.

[5] H.-T. Lim, K. Weckemann, and D. Herrscher, "Performance Study of an In-Car Switched Ethernet Network Without Prioritization," in *Nets4Cars/Nets4Trains 2011*, T. S. et al., Ed. Springer-Verlag, 2011.

[6] H.-T. Lim, L. Völker, and D. Herrscher, "Challenges in a Future IP/Ethernet-based In-Car Network for Real-Time Applications," in *Proceedings of the 2011 ACM/EDAC/IEEE Design Automation Conference (DAC11)*, San Diego, USA, Jun. 2011.

[7] IEEE 802.1 AVB Task Group, "IEEE 802.1 Audio/Video Bridging (AVB)." [Online]. Available: http://www.ieee802.org/1/pages/avbridges.html

[8] "IEEE Draft Standard for Local and Metropolitan Area Networks – Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks(IEEE P802.1AS/D7.7)," Nov. 2010.

[9] D. L. Mills, "Improved algorithms for synchronizing computer network clocks," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 245–254, Jun. 1995. [Online]. Available: http://dx.doi.org/10.1109/90.392384

[10] Y.-S. Li, G. Crispieri, and H. Wohlwend, "Using Network Time Protocol (NTP): Introduction and Recommended Practices," 2006. [Online]. Available: http://www.sematech.org/docubase/document/4736aeng.pdf

[11] M. Johas Teener and G. Garner, "Overview and timing performance of IEEE 802.1AS," in *Precision Clock Synchronization for Measurement, Control and Communication, 2008. ISPCS 2008. IEEE International Symposium on*, Sep. 2008, pp. 49 –53.

[12] G. Garner and H. Ryu, "Synchronization of audio/video bridging networks using IEEE 802.1AS," *Communications Magazine, IEEE*, vol. 49, no. 2, pp. 140 –147, Feb. 2011.

[13] S. Kang, J.-G. Ahn, Y.-S. Kwon, J.-H. Eom, and S.-H. Kim, "Timing and Synchronization with Propagation Delay Symmetry and Originated Slave Clock Frequency for Ubiquitous Computing," in *Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on*, Feb. 2010, pp. 764 –769.

[14] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems (IEEE 1588-2008)," Jul. 2008.

[15] "OMNeT++ Simulation Tool, Version 4.0," http://www.omnetpp.org/.

[16] "INET Framework for OMNeT++/OMNEST," http://inet.omnetpp.org/.

[17] eNOVA, "SEIS: Security In Embedded IP-based Systems," http://www.strategiekreis-elektromobilitaet.de/public/projekte/seis.