A Comparison of Time Synchronization in AVB and FlexRay in-vehicle networks

Helge Zinner, Josef Noebauer*, Jochen Seitz† and Thomas Waas‡
*Continental Automotive GmbH,
†Ilmenau University of Technology,
‡University of Applied Sciences Regensburg

***Abstract*—Ethernet gains more and more interest by the automotive industry which uses already a couple of different in-vehicle networks. Together with an demand for the integration of the IP protocol and a steadily increasing bandwidth requirements, Ethernet based networks are researched for the usage in this area to cope with future needs [1]. Especially Ethernet AVB promises an interesting networking solution for multiple application areas. In addition to the continuously rising communication demands of different industry's AVB provides different enhancements for Quality of Service (QoS) within Ethernet based networks. Timing sensitive applications, e.g. synchronous multimedia transport or real time control loops, require the predictable behavior that is addressed by this standard. The authors built up a QoS-based gateway interconnecting FlexRay and AVB. Most automotive network technologies used within automotive are time-triggered since many applications require a timely handling. Therefore the time synchronization methods are an essential part and the evaluation starts with the comparison of both time synchronization protocols. Since AVB is not originally designed to be used in in-vehicle networks yet, it has also to be proven if the protocol is able to fulfill the demands for automotive networks. The authors focus mainly on the average speed of time message distribution since vehicle network communication have to be established very fast. The comparison of the time accuracy shall shown if it is possible to keep the timing constraints of FlexRay when embedding those frames within Ethernet frames and to built up a global common time base. The results show that 802.1AS together with certain requirements to the AVB environment can provide almost similar timing guarantees in comparison to FlexRay which was proved by a demonstrator system.**

***Index Terms*—IEEE 802.3 Standards, Ethernet AVB, Automotive electronics, FlexRay**

I. Introduction

A public funded project SEIS [1] has already started to research the usage of the IP protocol within automotive networks. The use of more consumer standards in in-vehicle networks like IP and Ethernet, is motivated by different needs. In order to meet goals related to cost reduction, integration of consumer electronics, increasing reliability of components and experience in tools and development we concentrated on an IEEE based Ethernet standard. It is foreseen that the requirements regarding real time capabilities are still increasing in the future. Driven mainly by costs the complexity of the communication network increases with the requirements for bandwidth, safety and security. Sensor applications are mainly communicate over a single wire bus technology (Local Interconnect Bus - LIN) whereas high bandwidth multimedia is transported with the fiber optical based technology MOST (Media Oriented Systems Transport). The most widest spread in-vehicle network is the CAN (Controller Area Network) which is mainly applied for control systems mainly within powertrain and driving assistance. For dynamic systems controlling the stability of the vehicle the FlexRay network is used.

The gateway (the device which provides a gateway functionality) is the only device in current vehicle architectures which provides a network communication between the different communication protocols. This ECU is equipped with almost all communication controllers which are available in the vehicle. Hence the gateway is one of the most important components for distributed applications.

Current vehicle networking communication requirements are separated in two main categories, time-triggered and event-triggered. This results in the different requirements for the transport of real-time, non-real time and multimedia data. The time-triggered approach is followed by the FlexRay, which is the newest and most promising automotive network. This technology can provide predictable delays and jitters independent from the overall bus load. FlexRay was introduced in 2007 for the BMW X5 as suspension network. Nowadays it will be also used in more critical areas like engine control systems. Due to increasing bandwidth demandsfor control applications which cannot be transported by a single CAN network, many vehicles are equipped with multiple CAN networks to cope with the increasing bandwidth requirements in the powertrain network. The networking with multiple lower price CANs is thrown into question. Why not just a single network with a higher data rate?

An already ongoing unification and simplification of the vehicle networks requires a near term high bandwidth network for all subsystems. This has advantages for sensor data fusion and eases distributed applications. Vehicle architectures have nowadays already multiple separated FlexRay clusters since the data rate of 10 MBit/s per cluster is often not sufficient. Hence another application scenario could be the interconnection of two FlexRay clusters providing distributed real time applications.

Applications with strict real time requirements are typically not crossing the borders of a single network today. The gateways between the different networks do not provide a heterogeneous time synchronization strict real-time guarantees

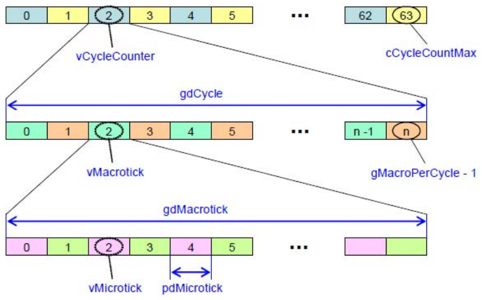

Fig. 1. FlexRay Timing Hierarchy [2]

can not be given along the whole path thought the vehicle networks. Each of the time-triggered networking technologies (LIN, FlexRay, MOST) supports a different approach. The challenge is to provide a common time base for the time sensitive applications and ECUs. Hence this paper evaluates the use of AVB for providing a heterogeneous time synchronization. This enables communication between current networks by maintaining guaranteed QoS which is not possible yet but might be required in future.

State Of The Art

In comparison to event driven systems, the control of the message process is in responsibility of the communication system itself, not in the control of the application. The time-triggered system has a defined schedule of their message sending cycles. This ensures a predictable behavior of the communication system. But cooperating electronic control units must have a global common notion of time in order to work in a distributed network. "Common notion of time" means that the clock speed difference is within a specified maximum limit. To guarantee the accuracy of controlling it is necessary to continuously synchronize the clocks of all networked components.

FlexRay

The FlexRay [2] communication protocol was invented as high-bandwidth communication network for in-vehicle real-time data transfer. FlexRay is a time-triggered network, hence the bus write access is scheduled by a time-division multiple access (TDMA) scheme, which means that bus write accesses are only allowed in exactly specified time intervals. In terms of FlexRay those intervals are called slots. A node of the network can only write messages in slots which are assigned to its ID and this has to be configured at the design of the network and cannot be changed during runtime. The slots can be either in a static or dynamic segment. Those segments form a communication cycle together with the symbol window segment and the network idle time (NIT) segment. The static segment is used for static periodical messages where each message has a predefined sending slot. In comparison to the dynamic segment the message length is fixed. The dynamic segment follows a more flexible approach where messages are sent with respect to their priority. The symbol window is used to transmit special control messages. The last part of the cycle marks the NIT which is used for clock synchronization by expanding or shortening this segment. The FlexRay communication is based on a hierarchy timing level as depicted in Fig. 1. The communication is tailored into 64 cycles. A cycle (gdCycle) is divided into an integer number of macroticks. Macroticks are divided into an integer number of microticks (gdMacrotick) which is the smallest timing unit in the FlexRay bus, whereas the microtick length (pdMicrotick) is referred to as interval of time derived directly from the crystal.

FlexRay uses a distributed clock synchronization mechanism in which each node individually synchronizes itself to the cluster. It follows a fault tolerant approach, because it is not based on a single timing master. Each node reconstructs the global time base. The predefined schedule defines some *sync nodes* (if a cluster has less than three nodes, each node must be a sync node) which are responsible for sending timing information in the static part of the FlexRay cycle. All nodes derive their time-triggered schedule from these *sync frames*. Every node shall measure (measurement phase) and store the time differences between the expected and the observed arrival times of all sync frames. The accuracy is based on the microtick level as visible in Fig. 1. This is done for two successive cycles for offset correction and rate correction, as depicted in Fig. 2. The nodes store the values in a table and use the fault-tolerant midpoint algorithm [3] to calculate the rate and offset correction with those recorded values. The microtick is not affected by the clock synchronization mechanisms, and is thus a node-local concept. Different nodes can have microticks of different duration due to the different characteristics of the local crystals. But rather the numbers of microticks per macrotick will be adjusted then. The correction of the time offset is then done by changing the NIT that concludes each communication cycle.

Ethernet AVB

Providing QoS for time sensitive applications in asynchronous Ethernet networks was hardly possible in a cost effective way with current Ethernet standards. Hence, the IEEE 802 working group defined the new AVB standard for providing quality of service. AVB extends the standard Ethernet with three additional substandards. IEEE 802.1Qav [4] utilizes methods described in IEEE 802.1Q to separate time-critical and non

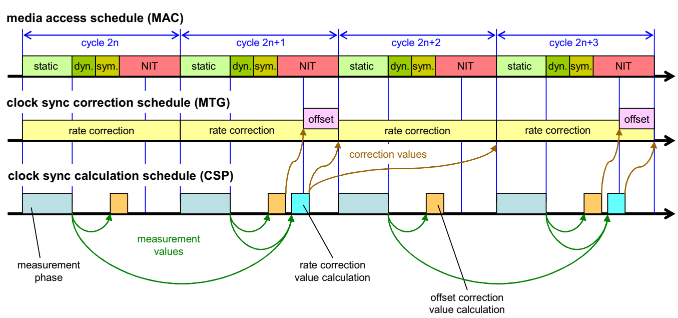

Fig. 2. FlexRay Clock Synchronization [2]

time-critical traffic into different traffic classes. Egress port buffers are separated into different queues, each allocated to a specific class. This ensures a separation of low priority traffic from high priority traffic. Moreover, all egress ports have a credit-based shaping mechanism to prevent bursty behavior. IEEE 802.1Qat [5] is an extension of IEEE 802.1Q and defines a protocol to signal reservation requests and reserve resources for media streams. It provides the necessary QoS by allocating buffers within switches. IEEE 802.1AS [6] describes the time synchronization and is responsible for the precise time synchronization of the network nodes to a global time. It is based on the Precision Time Protocol (PTP) described in IEEE 1588 [7], a clock synchronization protocol for networked measurement and control systems. IEEE 802.1AS is used to synchronize distributed local clocks referred as slave clocks with a reference clock.

Clock Synchronization

As defined by the IEEE 802.1AS protocol there can be only one best clock in an AVB network which is the grandmaster. The Best Master Clock Algorithm (BMCA) determines the so called grandmaster of an AVB network and provides the reference clock for the networked devices. The system clock of the slave devices is synchronized to this reference clock. It is necessary that the rate and offset calculation is performed regularly in order to provide a low jitter in the range of nanoseconds. Since the frequency of the crystal is fixed to a certain value, the offset and drift correction to the reference clock can be performed in software by the IEEE 802.1AS synchronization algorithm. Contrary to FlexRay, special timing messages (PTP frames) are sent to exchange timing information.

The precision and accuracy of the clock synchronization is a measure of the network's real time capabilities. The accuracy of networked clock synchronization depends on several factors. The quality of the crystal, the granularity of the timestamps and their sending interval, the propagation delay and the timestamping point. Synchronization protocols have a long history in the area of the packet-oriented Ethernet protocol. The software based network time protocol (NTP) is one of the famous protocols which still allows accuracy in the magnitude of milliseconds. The timestamping used by NTP is referred to as software timestamping. This is done at the application level. The accuracy of the timestamping will be better the closer to the physical media the timestamp is generated. This is due to the asynchronous mode of operation and the variable delay in PHY (physical layer), MII (Media Independent Interface) and MAC (Media Access Control). Hardware timestamping can be done at the MII/MAC or directly in the PHY. The latter allows accuracy in the magnitude of sub-nanoseconds. 802.1AS requires hardware supported timestamping. Hence this implies that standard transceivers cannot be used in an 802.1AS environment because they do not support the timestamping on this level. The specified worst-case accuracy is less than one microsecond but can have a significant higher accuracy as shown in [8]–[10].

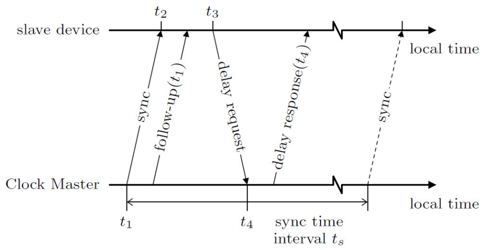

Fig. 3. PTP synchronization

The synchronization uses the PTP algorithm as depicted in figure 3. This algorithm consists of two parts. The master port sends a synchronization (*sync*) message to the slave port in a specified sync interval. The point in time when this message left the device (timestamp point) is denoted with t1 and is immediately forwarded within a follow-up message. When the slave port receives the sync message, a timestamp t2 is generated. The sync and follow-up messages are used for the offset correction. After that, a so called delay request message is sent at the time t3 from the slave to the master device and the master responds with the receiving timestamp t4 to the slave. This second part of the algorithm calculates the propagation delay. Since the clocks in the network can drift independently and continuously, the synchronization algorithm is executed periodically every ts microseconds. The requirements of 802.1AS ensure that any two timeaware systems (i.e., seven or fewer hops) will be synchronized within $1\,\mu s$ peak-to-peak of each other during steady-state operation.

Clock Accuracy FlexRay/AVB

FlexRay's node clock is typically driven by a nominal frequency f of 40 MHz for a data rate of 10 MBit/s. With a maximum clock frequency deviation of *fo* 3000 ppm (parts per million) a frequency drift of 120 kHz per clock period can occur, according to Equation (1). See also (Annex A of [2]).

$$\Delta f = \frac{f * fo}{10^6} \tag{1}$$

When using a 40 MHz crystal, one clock period has a length of $25\,ns$. This results in a time drift per period of $75\,ps$. The cycle length of FlexRay depends on the configuration. $16\,\mu s$ to $16\,ms$ are the lower and upper limits for the cycle length. Since FlexRay nodes will be synchronized for two successive cycles, this results in a maximum drift between two nodes of $96\,ns$ for the minimum or $96\,\mu s$ for the maximum cycle time. Typically the FlexRay is configured with a cycle time of $5\,ms$ or 5000 macroticks, which results in a maximum node drift of $15\,\mu s$. The timestamp resolution of FlexRay is based on the microtick resolution. This means, the synchronization is based on the microtick level of $25\,ns$ and cannot be more accurate. This causes an additional jitter of $25\,ns$.

AVB requires 100 MBit/s full duplex and faster and will not work for 10 MBit/s. Since the automotive industry is mainly focused on 100 MBit/s, this research is not focused on other variants.

The IEEE 802.3 standard has a defined interface between the PHY and Media Access Control layer, the Media Independent Interface for 100 Base-Tx. This interface can be monitored and whenever a frame of interest is seen, a timestamp can be created and stored. In case of MII this interface is clocked with a nominal frequency f of 25 MHz which is generated using a crystal. With respect to the specification in Annex B of [6], crystals used in AVB environment require a maximum drift rate *fo* of $200\,ppm$. This results in a frequency drift of 5 kHz according to Equation (1).

The change in period between the maximum (25002500 Hz) and the minimum 24997500 Hz frequency is $8\,ps$ while one clock period has a length of $40\,ns$. This means that two clocks could move away from each other for $8\,ps$ per $40\,ns$. This results in 3125000 clock periods per $125\,ms$ synchronization interval.(Synchronization messages can be sent in a sync frequency interval of $7.8125\,ms$ through $64\,s$. In order to provide also low cost AVB devices, the minimum sync frequency interval which has to be supported is set to $125\,ms$.) Multiplied with the change of period, this result in a maximum of $25\,\mu s$ drift between two clock periods. The timestamp resolution of 802.1AS is based on a 48 bit value that expresses the seconds and 32 bit value for the nanosecond resolution. But since the Ethernet is clocked with 25 MHz, the clock signal will have a jitter of +/- $20\,ns$. If the inherent jitter does not satisfy the application requirements, an external precision Phase-Locked Loop (PLL) must be employed to filter it out.

Network Startup

The requirements of the automotive industry require typically a network start below $100\,ms$. This time specifies an upper limit when a network communication should be possible. These are requirements which can typically not be fulfilled with consumer protocols since they have a different focus. Therefore this sections gives a detailed view of the network startup of FlexRay and AVB.

FlexRay defines two types of nodes for the startup, coldstart nodes and non coldstart nodes. A coldstart node is able to start a cluster by sending special startup frames (i.e. sync frames). In each cluster consisting of at least three nodes, at least three nodes shall be configured to be coldstart (see 7.2.1.1 of [2]). The startup is initiated with a collision avoidance symbol (CAS) sent by one of these nodes. All other nodes will then monitor the FlexRay channel. Then the first coldstart node will send sync frames for four cycles. Those four frames need to be monitored by the 2nd coldstarter to synchronize to. This coldstart node will also start to transmit sync frames. A non coldstarter then needs to monitor those startup frames of the coldstart nodes and can start sending four cycles later. Hence at least 8 cycles are needed for the bus initialization. With respect to the allowed cycle length this process can take between $128\,\mu s$ and $128\,ms$. Since FlexRay is typically used with a schedule of $5\,ms$ this cluster startup will take at least $40\,ms$.

AVB uses a Master-Slave clock synchronization hierarchy with one best clock, referred to as grandmaster. The Best Master Clock Algorithm (BMCA) of IEEE 802.1AS determines this master clock of an AVB network and provides the reference clock for the networked devices. The AVB nodes will send announce messages to their neighbors, which include information about the quality of their own local clock. Each receiver compares the clock information from the announce message with their clock and the parameter of their neighbor clocks. The algorithm constructs a time synchronization spanning tree with the grandmaster clock as the root node. As part of constructing the tree, each port of each time-aware system is assigned a port role. 802.1AS defines four different port roles. The role port changes its role to *MasterPort* if it is closer to the best clock than the other port connected to this port. *SlavePort* state is selected when the port is closest to the best clock than any other port of the node. The port state *DisabledPort* is chosen by ports which cannot execute the IEEE 802.1AS protocol completely. The role port *PassivePort* is dedicated to ports which are not part of the former three states. Announce messages are sent each second by default from master port to slave port, also when the network is already synchronized. Thus it is possible that a better clock (node) can later join the network. Additionally a failure of the grandmaster clock can so be compensated and a new grandmaster can be determined. In order to synchronize all nodes, the information about the best clock has to be given to all nodes. The time until the grandmaster is known by all nodes is influenced by the number of hops between the grandmaster and the longest path in the network, the propagation time of the announce message and possible interfering traffic in switches. The size of an announce message can be expressed as message size S_{an} + type length value (tlv) S_{tl} which for traces the route of an Announce message through the timing system for each hop (h). We assume a short announce message with a length of 102 byte (S_{an} = 94 byte + S_{tl} = 8 byte). Since all hops have to be informed of the best clock, the number of hops determines the maximum distribution time. Since all hops have to be informed of the best clock, the number of hops determines the maximum distribution time. The announce message distribution time depends on the number of hops which can is expressed Equation (2) whereas the number of hops is limited to seven by the IEEE 802.1AS standard.

$$T_{an} = (S_{an} + h(S_{tl}) * 8)/D_{et} \tag{2}$$

As part of constructing the tree (T_{st}) several BMCA state machines have to be passed which can be expressed in(T_{pr}). PortAnnounceInformation, PortRoleSelection and PortAnnounceTransmit state machine have to be passed. Together with the announce message distribution this results in the minimum time information forwarding time for each node. Equation (3) allows a calculation of the overall spanning tree creation time in dependence of the number of hops (h).

$$T_{st} = h(T_{an} + T_{pr}) \tag{3}$$

Under the assumption of an AVB network with maximum 7 hops we calculate a sync message distribution from end to end. The whole sync message delay is made up of the residence time T_{rs} (time interval between the receipt of a sync message and the sending of the next sync message on another port), specified by maximum $10\,ms$ and the sync transmission time of $5.76\,\mu s$ for a 60 byte Layer 2 Ethernet message. Since the follow-up message will be sent immediately after the sync message, the delay between those messages is not taken into account. The follow-up transmission time for a Layer 2 message size of 90 byte is $8.16\,\mu s$.

The minimum total time of the network startup (T_{ns}) is about the announcement of the best clock and the distribution of timing information, by sending of sync (T_{sy}) and follow-up (T_{fw}) messages. As soon as a port changes its role to MasterPort it starts sending sync messages immediately. So timing information can be propagated before the whole synchronization spanning tree is established. Therefore T_{ns} is only delayed by T_{an}. So the minimum time for distributing time information is about $60\,ms$ as expressed in Equation (4). Adjusting the frequency of an crystal (e.g., using a PLL) is slow and prone to gain peaking effects. The message distribution is negligible small compared to the frequency adjustment which is in magnitude of seconds (depends on implementation).

$$T_{ns} = h(T_{sy} + T_{fw}) + (h-1)T_{rs} + T_{an} \tag{4}$$

In the worst case an additional delay in the magnitude of microseconds can occur for the sync interval of $125\,\mu s$ by interfering traffic (assuming one maximum size interfering frame in the queue), maximum size announce messages and a time delay which shall prevent the network from bursts sync frames. See also [11].

Failure of Timing master

At least three nodes shall be configured to be sync nodes in a **FlexRay** cluster of more than two nodes. If one of these nodes would have a malfunction the network has still a synchronized schedule. A FlexRay cluster with one remaining sync node would cause the communication to stop.

If the grandmaster of the **AVB** network fails, the network will identify a new grandmaster with the use of the BMCA algorithm. The minimum total time of the grandmaster change time (T_{gm}) is calculated by the detection of a grandmaster error (T_{ts}), announce message delay (T_{an}) and sync message delay (T_{sy}). The absence of sync messages indicates a failure of the grandmaster. The default timeout to detect this error is specified by three missing sync frames. By assuming the minimum sync interval of $125\,\mu s$, which has to be supported by all AVB nodes (see section 6.9 of [12]), a reaction can be done after $375\,\mu s$ at the earliest. The sync receipt timeout interval (T_{ts}) is specified by three missing sync frames. Under the assumption the next best clock is far off, (max. distance of 7 hops) the new grandmaster has to be informed respectively determined. In addition to the sync receipt timeout time the announce message delay (minimum $7\,\mu s$) must be added per port. So the minimum total time of the grandmaster change is about 0.44 seconds, as expressed in Equation (5). This does not include any frequency adjustments.

$$T_{gm} = T_{ts} + (h-1)(T_{an} + T_{rs}) + (h) * T_{sy} \tag{5}$$

In the worst case an additional delay in the magnitude of microseconds can occur for the sync interval of $125\,\mu s$ by interfering traffic (assuming one maximum size interfering frame in the queue), maximum size announce messages and a time delay which shall prevent the network from bursts sync frames.

SYNCHRONIZATION BETWEEN FLEXRAY AND IEEE 802.1AS

Commonly multiple independent FlexRay networks are installed in a vehicle. But even if the clusters are based on the same periodic timing, called schedule, they will have a timing offset and drift against each other for the same reasons (crystals) as described above. In order to exchange data between two FlexRay networks they should follow the same schedule. Therefore one challenge is the time synchronization between multiple clusters which is a well known problem and will be addressed in the next revision of the FlexRay specification already. A similar but additional challenge is the synchronization of multiple clusters that are not physically attached to one common ECU but are only connected via network. The possibility to use a backbone network for interconnecting two different FlexRay clusters can be shown by using the AVB time synchronization. A system for proof this was built up by the authors. To achieve time synchronization between multiple clusters they have to adapt to each other. This could be done by e.g. nominating one master and adjustment of the others to this master cluster. An alternative approach utilizing the inherent time synchronization of the AVB backbone is to derive the timing of the clusters from the global AVB timing. The current FlexRay specification allows the following time synchronization method of independent clusters. A single FlexRay node can accomplish a clock correction as specified in [2]. A number of microticks per cycle can be added to or subtracted from the NIT to carry out a clock correction. This will extend or shorten one cycle and modifies the start time of the next cycle. Since FlexRay nodes obtain their time synchronization information from the expected arrival date of their slots, the frequency of the network will so be changed.

The system consists of a QoS gateway that was connected to a computer based residual bus simulation with simulated ECU's. The QoS gateway supports also the AVB standard and and first synchronizes to the 802.1AS time. Then the FlexRay timing was synchronized to the Ethernet timing by implementing the external FlexRay clock correction (Sync-Nodes). The frequency rate correction could be gradually adapted in a range of seconds. Without synchronization the observed drift was measured to be $1.75\,\mu s$ per 10ms. After the synchronization there was no drift anymore. However the algorithm of the FlexRay clock correction might need some optimization in order to reduce the remaining jitter of +/- 1 μs.

Outlook and Conclusion

802.1AS has a fixed sending cycle and is not influenced by any application sending cycle. The time synchronization messages will only be send from a host to next host. This ensures, that almost no waiting delay is added within the switch queues. Sync messages are timestamped on egress port of the sender and ingress port of the receiver. The sending time is transported in the following *follow_up* message. The timestamp shall be the time, relative to the local clock at which the message timestamp point passes the timestamping point marking the boundary between the time-aware system and the network media. The definition of the timestamping point, allows transmission delays to be measured in such a way (at such a low layer) that they appear fixed and symmetrical to 802.1AS even though the MAC client might otherwise observe substantial asymmetry and transmission variation [6].
The capability of the time synchronization of 802.1AS has already been proven since it is derived from IEEE1588 [7] for industrial applications. The same clock quality as within FlexRay can be achieved since the key factors influencing the time accuracy and precision like crystal, timestamping point or sending frequency are well defined and reasonably chosen within AVB. In comparison to FlexRay, Ethernet is a packet oriented network and needs an additional protocol for distributing the common time base. FlexRay just monitors the arrival time of incoming messages. So AVB synchronization is done with the help of the 802.1AS protocol which causes an negligible small overhead of much less than 1 MBit/s (depending on sending intervals) per link. Due to the switched architecture of Ethernet interfering traffic can influence the propagation delay of the PTP messages. With regard to the switch configuration, it has to be taken into account that the 802.1A messages are not treated with the highest priority. They are even not marked with any 802.1p priority value.

Currently the authors investigate the data transport from a FlexRay cluster to an AVB network and vice versa. A proposal for the implementation of a time synchronization was already suggested and also proven with the validation system. Time synchronization between these networks was identified as a prerequisite. For a full FlexRay / AVB gateway further work needs to be done. The simulation of the FlexRay network will be exchanged by real ECUs in a next step. Furthermore the backbone architecture, synchronization of multiple FlexRay's and efficient data routing will be investigated and implemented.

It would be useful to have just one single network in the vehicle. Due to multiple applications with completely different requirements, several networks have been established in the past. The authors believe that it is useful that Ethernet replaces multiple existing automotive solutions rather than presenting just a new additional network. Nowadays FlexRay is still not very well established in the automotive industry although it is a known standard for several years. The authors built up a prototype system to demonstrate interworking between FlexRay and AVB. One purpose is to propose a mid-term migration concept where both technologies can co-exist. Another purpose is to show that the transport of time-critical data within distributed control applications is possible using an Ethernet AVB network and FlexRay could be fully replaced.As of today FlexRay is not used in the way it was designed. Fault tolerance is one of the major key aspects for the usage of this technology but never used! Although the name AVB leads to the assumption, that Audio-Video-Bridging is only suitable for multimedia data.

References

[1] "Sicherheit in eingebetteten IP-basierten Systemen (SEIS)," http://www.strategiekreis-elektromobilitaet.de.

[2] FlexRay Cooperation, "FlexRay Communications System Protocol, Specification, Version 2.1," 2008.

[3] J. L. Welch and N. A. Lynch, "A New Fault-Tolerant Algorithm for Clock Synchronization," Information and Computation, vol. 77, no. 1, pp. 1-36.

[4] "IEEE 802.1Qav. Virtual Bridged Local Area Networks Amendment 11: Forwarding and Queuing Enhancements for Time-Sensitive Streams."

[5] "IEEE 802.1Qat. Virtual Bridged Local Area Networks Amendment 9: Stream Reservation Protocol (SRP)."

[6] "IEEE 802.1AS. Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks."

[7] "IEEE Std 1588, Standard for A Precision Clock Synchronization Protocol for Networked Measurement and Control Systems."

[8] H. Zinner, A. Kern, T. Streichert, J. Noebauer and J. Teich, "Accuracy of Ethernet AVB Time Synchronization Under Varying Temperature Conditions for Automotive Networks," in *Proc. DAC2011*. ACM, June 2011.

[9] G. Garner, A. Gelter, and M. Teener, "New simulation and test results for ieee 802.1as timing performance," in *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*. IEEE, October 2009, pp. 1–7.

[10] M. J. Teener and G. Garner, "Overview and timing performance of ieee802.1as," in *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*. IEEE, September 2008, pp. 49–53.

[11] G. M. Garner, "Analysis of Grandmaster Change Time in an 802.1AS Network (Revision 1)," September.

[12] "IEEE 802.1BA. IEEE Standard for Local and Metropolitan Area Networks—Audio Video Bridging (AVB) Systems."