TSN – Time Sensitive Networking

Time-Sensitive Networking (TSN) is currently being developed at the IEEE as a novel technology that offers an entirely new level of determinism in standard IEEE 802.1 and IEEE 802.3 Ethernet networks.

This means that future Ethernet networks will be able to provide:

- Calculable, guaranteed end-to-end latencies
- Highly limited latency fluctuations (jitter)
- Extremely low packet loss

For which applications, however, are these characteristics really relevant and how exactly does TSN achieve this functionality? This White Paper gives an overview of the most important functions provided by TSN and illustrates the advantages of using TSN in demanding industrial networks.

Table of Contents

Real-time Communication Today and in the World of the Industrial IoT

Today, latency guarantees are established as a basic requirement for real-time data transmissions in a number of application scenarios. These include synchronised axles and drives, power generation, transmission and distribution networks as well as the transportation industry. In these fields, the cycle times for the transmission of time-sensitive process data are often significantly below 1 millisecond. To achieve these low cycle times with correspondingly low latency guarantees, real-time communication technologies such as EtherCAT, PROFINET IRT or SERCOS III are currently being used. Although these technologies are based on conventional Ethernet, they commonly incorporate additional mechanisms to provide latency guarantees that, in turn, are often incompatible with each other. As a result, the real-time Ethernet solution market nowadays is severely fragmented and, due to the lack of compatibility, is crippled with regards to future development. Here, TSN has the potential to open up the real-time Ethernet market by establishing a universal physical and data-link layer that is standardised by the IEEE 802, the creators of Ethernet. For customers, this homogenisation will lead to potential cost savings as well as investment security when opting for the implementation of real-time Ethernet.

Besides the above-mentioned applications with "hard" real-time requirements, additional application domains such as process automation can profit from TSN as well. At first, this seems contradictory to the fact that the cycle times in these domains are often significantly larger than, for example, for synchronised drives. For these application scenarios, the benefits of TSN originate in the requirement for guaranteed end-to-end latencies. In current networks, these guarantees are typically approximated by over-provisioning the available bandwidth. In contrast, with TSN, it is possible to eliminate such approximation-based solutions and to tailor both the guaranteed bandwidth as well as the latency exactly to the application requirements. Consequently, TSN permits you to plan and to dimension future automation networks according to their actual bandwidth requirements.

Also, when looking at the future of automation networks, a consistent increase in the significance of TSN is foreseeable. Even today, the field of industrial automation is in a period of transition that is driven by the vision of permitting much more flexible, more intelligent and more dynamic production facilities than is currently possible. Terms that are often associated with this vision are "Industry 4.0 (I4.0)" and "Industrial Internet of Things (IIoT)". They describe intelligent production environments in which production machinery, conveyor systems and workpieces are constantly communicating with each other in order to support an automated and more efficient production process. This is made possible by increased networking of the sensors and actuators that are involved in the production processes. Another factor is the increased integration of the (local) Cloud, where, for example, virtual programmable logic controllers are hosted and interact directly with the production process through the sensors and actuators at the field level. These changes affect the models, on which the development and planning of current automation networks are based. As illustrated in Figure 1, the familiar automation pyramid is expected to transform into an automation pillar in a long-term, continual process. In contrast to the automation pyramid, where real-time requirements for data transmissions were mostly present at the field level, both the field and the connectivity level will need to fulfil low-latency requirements in the case of the automation pillar.

Moreover, another new paradigm is emerging beyond the requirements for calculable and lowest-possible latency and jitter: the increased convergence of the different networks that, today, are still used in parallel within existing production sites. While in current facilities, time-sensitive control data is often transmitted via dedicated networks built only for that particular purpose, it is foreseeable that in the future this control data will be transmitted
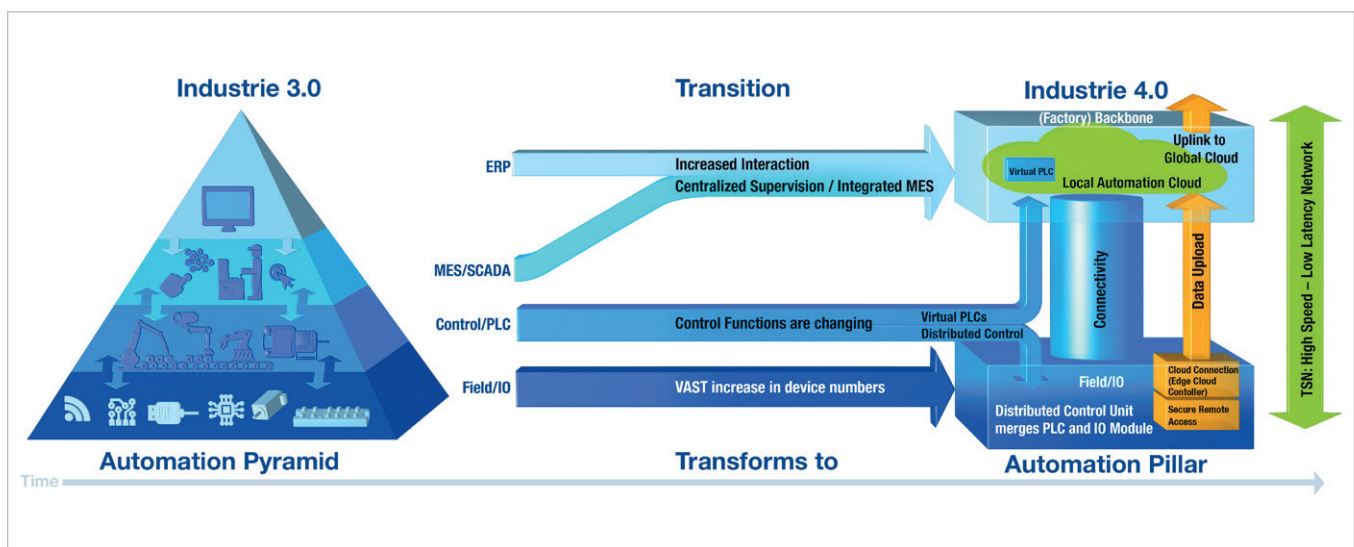

Figure 1: Transformation from the automation pyramid to the automation pillar in future automation networks
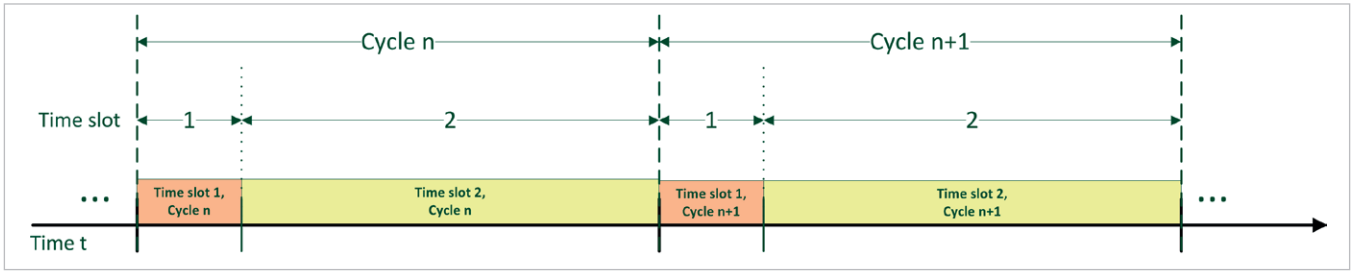
Figure 2: Time division multiplexing permits the reservation of time-slots within a cycle in order to enable the timely transmission of periodic real-time data

in parallel with "Best Effort" data (e.g. configuration and monitoring data) and data with "soft" real-time requirements (e.g. video data from surveillance cameras) over a common network infrastructure. One key characteristic of TSN is to offer a solution for such converging network infrastructures with high demands on bandwidth at the connectivity level and hard as well as soft real-time requirements at the field and connectivity levels. Hence, TSN will play a vital role for demanding and critical applications in the automation networks of the future.

TSN – Mechanisms and Interdependencies

TSN adds a level of determinism to Ethernet-based data communication that is able to meet even the highest demands of modern control networks, for example, in industrial automation and the automobile industry. Even today, it is foreseeable that TSN will reach a broad audience and the target

markets of TSN will likely differ from one another significantly. Thus, for example, deterministic as well as fault-tolerant data transmissions may be a firm requirement in one target market, while in another case, fault-tolerance through redundant transmissions may only be of secondary importance. Therefore, TSN has been conceived as a modular system by which the precise characteristics of the implementation – and the associated hardware and software requirements – can be tailored to fit the individual requirements.

Appropriately so, TSN is not made up of a single standard document, but is a family of standards which have been in development by the IEEE 802.1 TSN Task Group[1] since 2012. By now, these activities have yielded their first results: central mechanisms of the TSN family are already available as standard documents. In order to give an overview of these new technologies, the following sections will address the most important TSN mechanisms and their interdependence.

Prioritisation based on timing with the Time-Aware Scheduler

Until now, it was not possible with Class of Service (CoS) mechanisms such as the IEEE 802.1Q strict priorities to guarantee bounded end-to-end latency of time-sensitive data traffic. Due to queueing effects, an Ethernet frame with low priority that is already in transmission could delay Ethernet frames of even the highest priority (7) at every Ethernet switch along the transmission path. As one of the central components of TSN, the Time-Aware Scheduler (TAS), for the first time, introduces the possibility for prioritising the data transmission of conventional Ethernet frames based on transmission time and thus guaranteeing their forwarding and delivery at a defined point in time.

The fundamental idea of this TSN mechanism, published as Standard IEEE 802.1Qbv-2016[2] in March 2016, is, to utilise TDMA (Time Division Multiple Access) to divide time into discreet segments of equal length, so-called cycles, as illustrated in Figure 2.
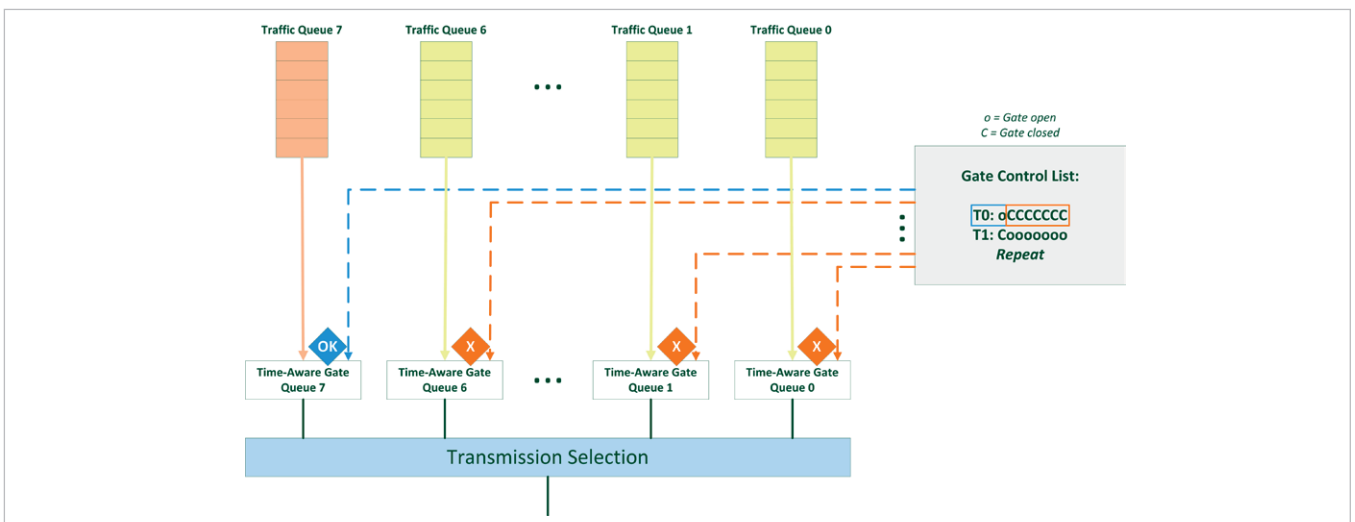

Figure 3: The Time-Aware Scheduler implements time-based prioritisation via the newly-introduced Time-Aware Gates that sit between the CoS queues and the selection of the packets to be sent

This allows dedicated time slots to be provided for the transmission of data packets with real-time requirements within the cycles. With the aid of the Time-Aware Scheduler, the transmission of conventional Best Effort Ethernet traffic can be temporarily interrupted in order to forward time-sensitive data traffic within the reserved time slots for high-priority traffic. The Time-Aware Scheduler thus permits the prioritisation of periodic real-time data (see Time slot 1 in Figure 2) in relation to conventional Best Effort data traffic.

Similar to the strict prioritisation scheme, the Time-Aware Scheduler uses the CoS priorities (PCP — Priority Code Point) that are present in the VLAN tag of the Ethernet header. In this case, all Ethernet frames are processed until they reach the Time-Aware gate queues at the output port. At this point, the Time-Aware Scheduler intervenes in the packet processing, as illustrated in Figure 3. More precisely, with the use of the Time-Aware Scheduler, the selection of the next Ethernet frame to be transmitted is no longer just determined strictly by a linear hierarchy at the queue, but rather the state of the respective gates is also taken into consideration. This state may be either open or closed, based on actual time. Ethernet frames that are waiting for transmission in the associated queues will be considered in the packet

selection, depending on these states. In Figure 3, for example, only the queue with a priority of "7" is processed at this particular point in time.

The Gate Control List determines which traffic queue is permitted to transmit at a specific point in time within the cycle. Besides the states of the Time-Aware Gates, the Gate Control List indicates the length of time for which a specific entry will be active. In the case of the Gate Control List shown on the right side in Figure 3, the list mirrors the cycle that consists of a Best Effort phase, as well as a phase with prioritised data traffic from Figure 2.

The necessity of guard bands and the interruption of Ethernet frames

Due to the very poor predictability of Best Effort traffic patterns, it is generally not foreseeable when a specific Best Effort data packet will need to be processed. As illustrated in Figure 4, for example, the transmission of an Ethernet frame in time slot 2 could be initiated too late. This Ethernet frame would then, despite the use of the Time-Aware Scheduler, extend into the time slot number 1 of the subsequent cycle. This would therefore result in a delayed processing of real-time data and a violation of guaranteed end-to-end latencies.

In order to avoid these situations, besides the transmission barriers between the

time slots, the so-called guard bands have to be introduced in conjunction with the Time-Aware Scheduler. These guard bands suppress the transmission of packets for the duration of a maximum-size Ethernet frame. Thus, the guard bands can prevent the transmission of Best Effort Ethernet frames that would intrude into the subsequent time slot. As illustrated in Figure 4, this prevents delays in processing of real-time data during the transition from a Best Effort phase to a phase with high-priority traffic. But this guard band also inevitably results in undesirable dead times where the network can't be utilised at all and thus, in a waste of bandwidth.

In addition to the explicitly configured guard bands, the Time-Aware Scheduler also permits that the packet length of the next-in-line Ethernet frame is taken into account. The decision whether to transmit now or wait for the next Best-Effort timeslot depends on whether the next frame is short enough to be fully transmitted within the current time slot. But even with this mechanism, situations can occur where there is simply not enough time left in the current timeslot or the frame to be transmitted is too large to fit in the packet. Therefore, even with this mechanism, the dead times that result from the guard bands cannot be entirely prevented.
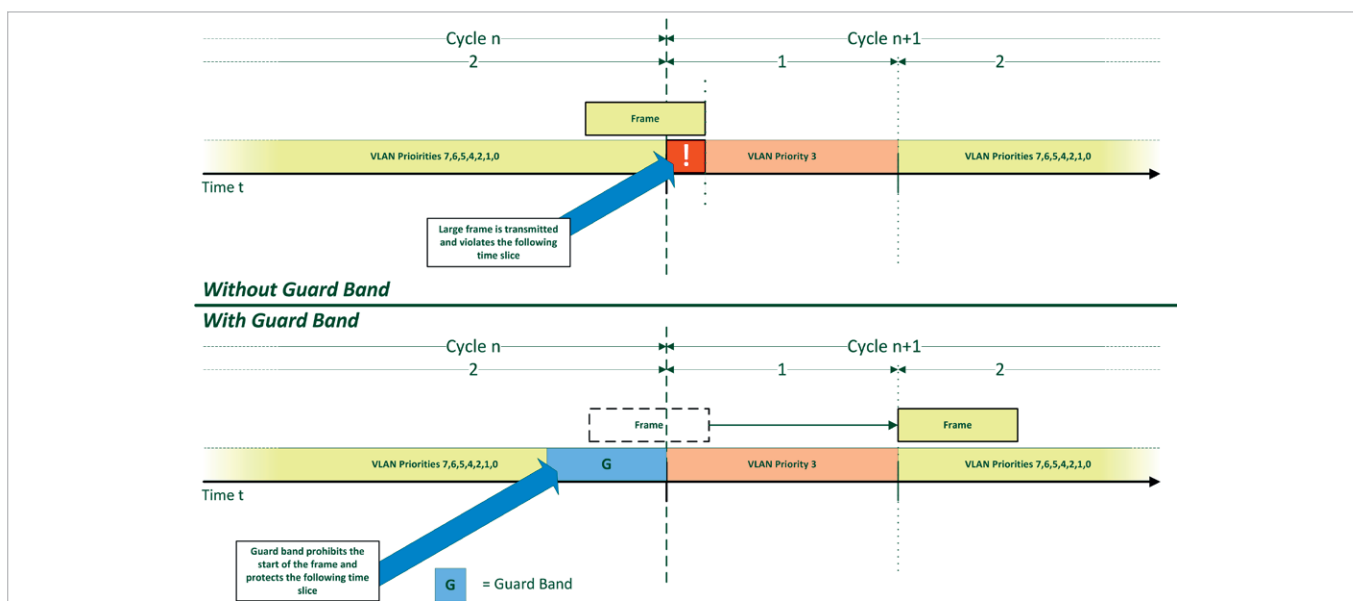

Figure 4: The guard band in TSN prevents Best Effort frames from extending into a time slot that is reserved for real-time data, but it decreases the available bandwidth
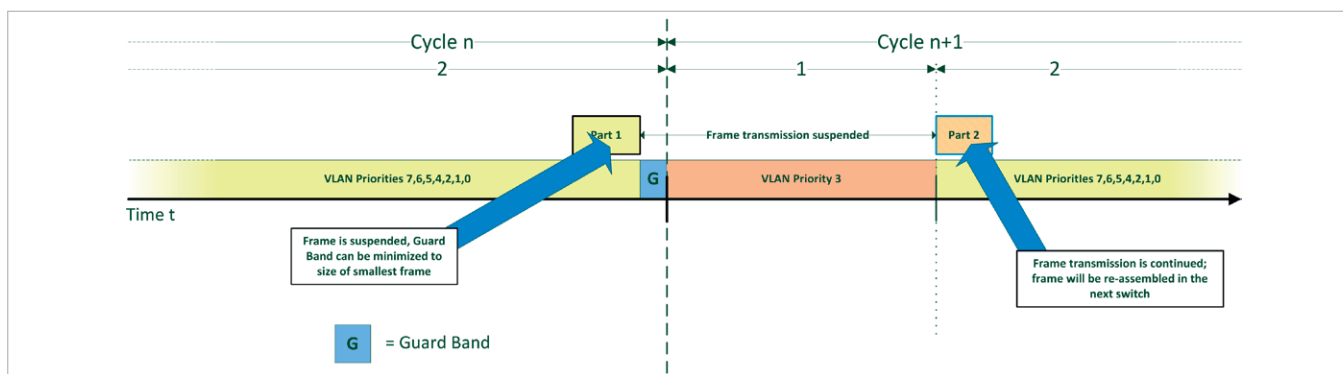
Figure 5: With the method of Ethernet frame pre-emption, the guard band size can be reduced from the maximum size of an Ethernet frame to the size of a partial packet

In order to maximise the usable bandwidth for Best Effort Ethernet frames, the IEEE 802 working group developed a method for Ethernet frame pre-emption (IEEE 802.1Qbu-2016[3], IEEE 802.3br-2016[4]), completed in June 2016. With this method, conventional Ethernet frames can be divided into partial packets ("framelets") of as small as 64 bytes, and each framelet may be transmitted separately. As shown in Figure 5, this permits starting the transmission of a large Ethernet frame, despite insufficient remaining time within the Best Effort phase. The frame can be interrupted at the last 64-byte boundary before the current time slot ends and can then be completed in the next Best Effort phase. Frame pre-emption makes it possible to reduce the guard band to the maximum size of one Ethernet framelet. In the case of a fast Ethernet network, for example, the dead time from each guard band can be reduced to 0.12 ms and thus, a significant improvement of the use of the bandwidth available can be achieved.

Due to the fact that frame pre-emption is a significant intrusion into the normal process of Ethernet frame forwarding and processing, it is necessary for both devices of an Ethernet connection (e.g. two Ethernet switches) to announce their support for this mechanism through the use of the Link Layer Discovery Protocol (LLDP) (IEEE 802.1AB-2016[5]). Only with frame pre-emption support on both ends of the link, the feature can be activated on the corresponding end devices or switch ports. With this, backwards compatibility with existing Ethernet devices is maintained.

Synchronous transmission cycles as a prerequisite

The Time-Aware Scheduler utilises only local configuration data — the data that is available in a particular network device (end device or switch). This configuration data consists of information about the lengths of cycles and time slots, for example. Therefore, besides the Time-Aware Scheduler, close

coordination between the devices in the network is required in order to ensure that the frames match the correct time slots in each switch. This enables the transmission of communication streams that can be transmitted through end-to-end connections, with guaranteed latencies and without queuing times (see Figure 6). This means, in particular, that all network participants must possess a common understanding of time. In particular, all participants must know when a cycle begins and which time slot is active in the cycle. In order to enable this, the use of a protocol for time synchronisation, such as the Precision Time Protocol (PTP) in accordance with IEEE 1588 (IEEE 1588-2008[6]) or the IEEE 1588 Profile IEEE 802.1AS (IEEE 802.1AS-2011[7]) is mandatory.

Both IEEE 1588 as well as IEEE 802.1AS permit the synchronisation of distributed clocks within a network with an accuracy of under 1 µs. Implemented in hardware, timing precision in the range of a few nanoseconds can be achieved (Hirschmann
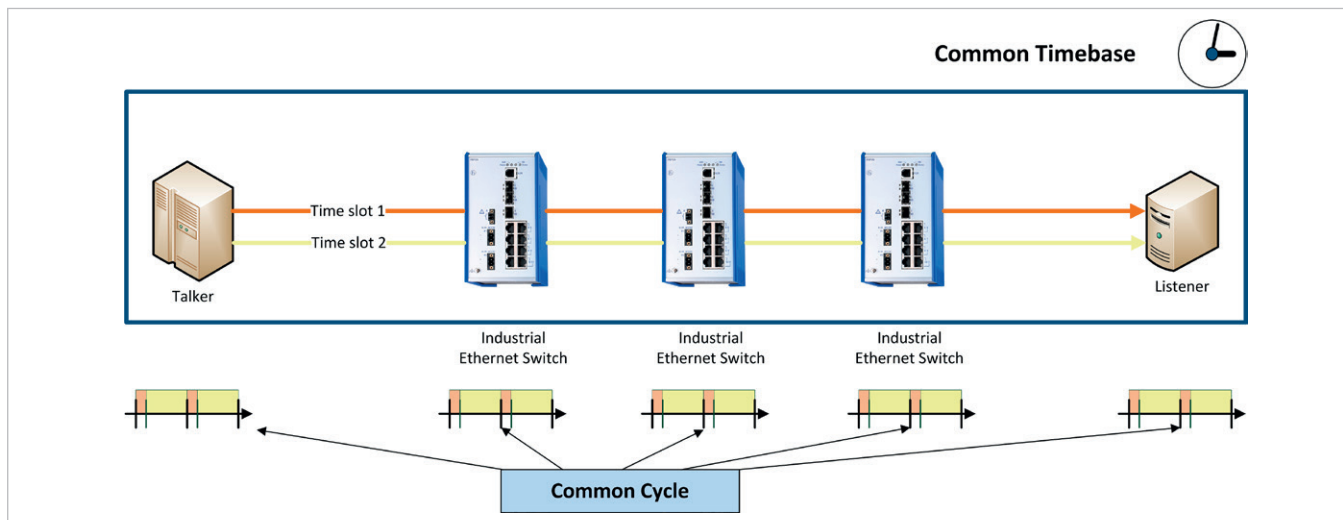

Figure 6: Good time synchronisation is a prerequisite for the TSN Time-Aware Scheduler

PTP Whitepaper[8]). In contrast to the protocols known from IT environments, such as Network Time Protocol (NTP), IEEE 1588 does not necessarily have to utilise a global synchronisation with, for example, an atomic clock. More commonly, the network participant with the most precise, freely running clock is determined with the aid of the Best Master Clock (BMC) algorithm. This device then serves as the reference clock (Grandmaster Clock), against which all remaining network participants are synchronised. For TSN, it is of primary importance that the time is synchronised to all clocks in a network. The actual time of day, on the other hand, plays only a secondary role.

The IEEE 1588 Profile, IEEE 802.1AS, follows the same fundamental synchronisation model as PTP. It was originally developed to limit the large number of configuration options to those parameters that are relevant in local networks (LANs). For example, in case of the transport technology and encapsulation, IEEE 802.1AS is confined to Ethernet transport, while IEEE 1588 provides an additional IPv4 encapsulation scheme for use in wide area networks. As a result of the TSN standardisation process, the existing IEEE 802.1AS profile has been expanded by the addition of parameters from IEEE 1588 that are required for use in automation networks. For example, IEEE 1588 offers support for multiple time domains that can be synchronised in parallel. Accordingly, with IEEE 1588, network participants can be synchronised with a global time reference (as with NTP), as well as a second network time reference. This offers the option to use the global synchronisation for unambiguous event logging, while the network-wide synchronised clock can be used for the Time-Aware Scheduler, since in this case, synchronisation according to global conventions (such as the leap second) is not required. Among other things, this capability will also be included in the next version of this profile with IEEE 802.1AS-Rev[9].

Since the current version of IEEE 1588 was already specified in 2008, this technology for time synchronisation has already been established in many markets and application areas. In some cases, profiles for special applications,

such as the energy market, have been developed and are in use today. In these cases, there is no need to specifically utilise IEEE 802.1AS for time synchronisation – the TSN mechanisms permit the use of any arbitrary mechanism for time synchronisation. Thus, depending on the application area, IEEE 1588 can be used instead of IEEE 802.1AS, with or without a specific profile. IEEE 802, in the future, does not intend to limit this freedom of choice in regard to which protocol needs to be used for time synchronisation. In any case, regardless of the synchronisation protocol that is used, the quality of clock synchronisation that is achieved must be very high in order for all devices in the network to start and end cycles and time slices at the correct points in time.

Traffic Shaping in the case of imprecise transmission timeframes

In application areas such as process automation, periodic control processes are often used that will, for example, result in event-based data transmissions. This can be the case when state transitions need to be communicated or defined boundaries are exceeded with measured variables. Accordingly, the transmission times in these scenarios cannot always be precisely predicted. Even so, clearly defined latency boundaries typically need to be met in order to ensure that the control processes can still act in time with the information that is received. Since the Time-Aware Scheduler is dependent, however, on precise transmission times, the mechanism is not perfectly suited to this kind of traffic model.

In addition to the Time-Aware Scheduler, TSN offers additional prioritisation mechanisms, the so-called Traffic Shapers. These permit the reservation of the maximum bandwidth that is necessary for time-sensitive data transmissions within a defined observation interval (for example 250 μs). The data traffic to be conveyed is subsequently transformed by the respective Traffic Shaper into a type and form that guarantees that certain latency limits can be achieved for time-sensitive data transmissions. One compromise for the flexibility that is gained by using a traffic shaper is, however, lower precision with regard to the achievable latency and

jitter guarantees in comparison to the Time-Aware Scheduler.

In the context of the standardisation activities within the IEEE, there are three different Traffic Shapers that are currently discussed for usage with TSN:

- Credit-Based Shaper (CBS; IEEE 802.1Qav-2009[10])
- Cyclic Queuing and Forwarding (IEEE 802.1Qch-2017[11])
- Asynchronous Traffic Shaping (ATS; IEEE P802.1Qcr[12])

The Credit-Based Shaper was developed in 2009 by the IEEE 802.1 Working Group for the predecessor technology of TSN, Audio/Video Bridging (AVB). As the name indicates, it is primarily targeting audio/video and similar applications. The goal of the Credit-Based Shaper is to ensure provision of the maximum required bandwidth for an audio/video transmission over a time sequence, without a noticeable interruption of the Best Effort data traffic that is simultaneously transmitted. In order to achieve this, the Credit-Based Shaper assigns sending credit to the data streams with reserved bandwidth. The initial value for the sending credit is 0.

As long as the sending credit is in the positive range (≥0), data frames with reserved bandwidth can be transmitted with a higher priority (see for example the transmission of the first AVB frames, marked blue in Figure 7, left side). With each prioritised transmission, the sending credit decreases, until it eventually reaches the negative range. While the sending credit is in the negative range, data frames with reserved bandwidth may no longer be transmitted. Accordingly, Best Effort frames that are in the transmission queue at this time can be processed. If the transmission of data frames with a reserved bandwidth is delayed because of this transmission, the sending credit of the respective data stream increases (see Transmission of the Best Effort frames, marked black in Figure 7). As a result, the delayed Ethernet frames of the prioritised data streams can then be transmitted back to back, following the transmission of the Best Effort frames. This prevents additional delays in the transmission of time-critical frames.

Figure 7: With the Credit-Based Shaper, data streams with reserved bandwidths are handled with higher priority than Best Effort traffic, as long as positive transmission credit is available

Due to its prioritisation characteristics, the Credit-Based Shaper is well-suited for the prioritised transmission of audio/ video data, as it exists, in the video surveillance in production processes or facilities. This is especially true with regard to the small amount of buffering for this data within the receiving end devices. However, it has been shown that the maximum end-to-end latencies of 2 ms and 50 ms respectively, specified by the standard over seven hops, cannot be met for every network topology and every communication pattern[13]. This prevents the use of the Credit-Based Shaper in application fields such as process control, where fixed guarantees regarding the maximum end-to-end latency are required.

For this reason two additional Traffic Shapers are being developed within the IEEE that can guarantee end-to-end latencies without limitation to network topology and communication patterns.

One of these Traffic Shapers is the Cyclic Queuing and Forwarding method that makes use of the mechanisms of the Time-Aware Scheduler. However, when compared to the Time-Aware Scheduler, this Traffic Shaper has significantly reduced requirements concerning the time-precision of the transmission. As shown in Figure 8, the basic concept of the Cyclic Queuing and Forwarding method is to collect the data frames with reserved bandwidth received within a cycle and transmit them as "prioritised" at the start of the next cycle. Thus, the maximum end-to-end latency can be determined precisely through the number of hops on the transmission path and the configured cycle time. With these characteristics, Cyclic Queuing and Forwarding is well-suited to the sporadic data transmission of process automation as described earlier.

Due to the similarities between the mechanisms used by the Time-Aware Scheduler and the Cyclic Queuing and

Forwarding, it will also require the network participants to have a common concept of time and thus a time synchronisation mechanism.

The planned third Traffic Shaper, Asynchronous Traffic Shaping, differs from Cyclic Queuing and Forwarding in that this approach does not require a time synchronisation mechanism. Accordingly, Asynchronous Traffic Shaping will be well-suited to the prioritised transmission of data packets that are needed for the time synchronisation itself. The mechanisms for Asynchronous Traffic Shaping are also in a very early stage of the standardisation process. Therefore, at the time of the creation of this document (4. quarter 2017), no statement can be made as to the precise specification of this Traffic Shaper.
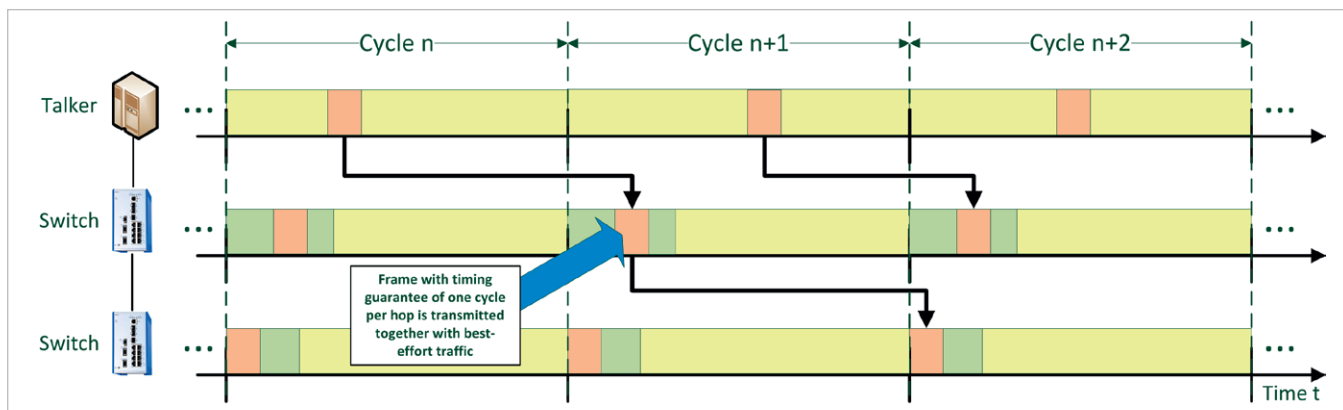

Figure 8: Using Cyclic Queuing and forwarding, data streams with reserved bandwidth are transmitted intermittently by one hop in the direction of the receiver with each cycle

7

Common use of Traffic Shapers and Schedulers

The use of the various Traffic Shapers is always connected to the exclusive assignment of one of the eight CoS priorities from the VLAN tag to a specific shaping/scheduling algorithm. If a device supports the Time-Aware Scheduler – in accordance with IEEE 802.1Qbv, the Cyclic Queueing and Forwarding Traffic Shaper according to IEEE 802.1Qch-2017 and the strict priorities in accordance with IEEE 802.1Q commonly found in almost all Ethernet switches today – the various CoS priorities can be assigned to these scheduling and shaping mechanisms in the device configuration. For example, the priorities 7, 4, 3, 2, 1 and 0 could be assigned to the strict priority mechanism and be used for the transmission of Best Effort traffic. Priority 5 could be assigned to the Cyclic Queueing and Forwarding Shaper and priority 6 to the Time-Aware Scheduler, in order to implement communication with soft and hard real-time requirements. This way, various traffic classes can coexist within the same network and can be prioritised by the appropriate mechanism. The prerequisite for this, however, is that all devices in the network support VLAN tagging in accordance with IEEE 802.1Q and support the scheduling and shaping mechanisms required for processing the data traffic.

Preventing interfering traffic with ingress filtering and policing

In a system in which all participants behave as expected, the TSN standards already described above offer all of the mechanisms required for deterministic data transmission. However, the mechanisms discussed so far require a complete reception of frames, as well as (partial) frame processing in a forwarding switch or receiving end device. As a result, misconfigured devices or malicious network participants can significantly interfere with the operation of TSN mechanisms such as the Time-Aware Scheduler by sending data frames with erroneously assigned CoS priorities or by excessively stressing the resources assigned to them.

In order to counter this, an additional TSN mechanism was developed within the IEEE 802.1 working group that allows discarding data frames that have been erroneously assigned at the time of reception (IEEE P802.1Qci[14]). In addition, this mechanism allows discarding real-time data streams that use more than their reserved bandwidth, thus allowing the policing of streams. Finally, TSN can make use of already existing Layer 2 security mechanisms, such as MACsec (IEEE 802.1AE[15]). This allows ensuring the authenticity of the sender so that only verified Ethernet frames are forwarded. This way, it is possible to handle a multitude of attacks and scenarios with erroneously configured network participants.

Better safe than sorry: Communication Path Redundancy

In addition to such misconfigured or malicious network participants, failure of a network component or cable can also cause interruption of deterministic data transmission. In order to prevent the packet loss resulting from such an interruption, the IEEE is currently developing a redundancy protocol with IEEE P802.1CB[16] that uses mechanisms similar to the already established seamless redundancy mechanisms, High Availability Seamless Redundancy (HSR) and the Parallel Redundancy Protocol (PRP). One goal is to maintain compatibility to HSR and PRP that is specified in IEC 62439-3. IEEE P802.1CB involves static redundancy procedures, in which the redundant transmission paths are permanently active. In the case of a failure, switchover times from one path to another can be avoided.

In order to achieve seamless redundancy with IEEE P802.1CB, the Ethernet frames that need to be transmitted are replicated at the beginning of a redundant transmission path and subsequently forwarded through the network via multiple paths. Usually, the replication occurs either directly on the sending device or, if the end device does not support redundant network connections, such as the one illustrated in Figure 9, at the first network device on the transmission path. When the data arrives at the destination, the first redundant data packet is forwarded in the direction of the application layer. Packet duplicates received after the

Frame is being replicated and transmitted on redundant paths through the network

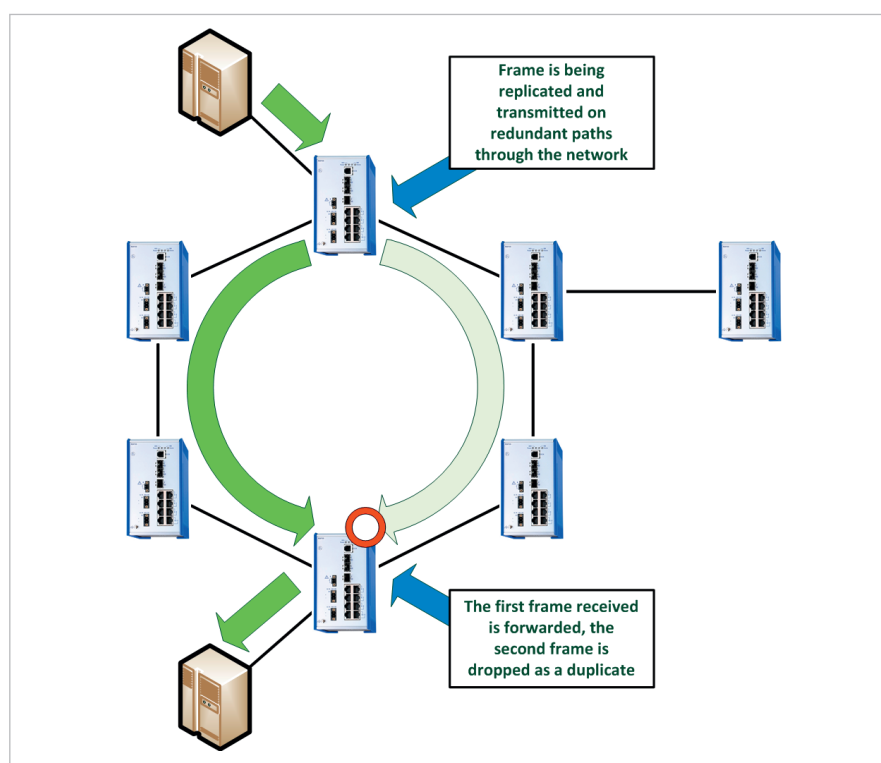**The first frame received is forwarded, the second frame is dropped as a duplicate**

Figure 9: In the case of the seamless redundancy protocol IEEE P802.1CB, Ethernet frames are replicated at the beginning of a redundant transmission path and duplicate packets are discarded later

first packet are recognised via a redundancy field in the Ethernet header and discarded. Thus it is ensured that the redundant data transmission with IEEE P802.1CB is transparent for higher layers in the network stack and do not need to be taken into account.

In comparison to HSR and PRP, the redundancy mechanisms developed in the context of the IEEE P802.1CB offer the advantage that they can be used in any topology. Thus, IEEE P802.1CB is not limited to the otherwise absolutely required ring topology or topologies with completely independent networks. Additionally, IEEE P802.1CB is not restricted to exactly two redundant paths. In order to reduce the probability of packet loss, it is also possible with IEEE P802.1CB to utilise numerous redundant transmission paths. However, in this case, it must be ensured that all redundant paths can support the latency guarantees that are required by the application. The convenient management of requirements and configuration of TSN network paths is thus an important component of a functioning TSN ecosystem consisting of network devices and network management.

Configuration of the complete TSN network

As explained earlier, TSN consists of a series of standards and mechanisms that serve the various requirements of deterministic data transmission. In order to implement these different mechanisms jointly in a network and to be able to parametrise them – independently from the manufacturer – over various network devices, a standardised form of configuration is required in complete TSN networks. This configuration mechanism must permit the use of TSN mechanisms, such as Ethernet frame pre-emption or the activation of redundant data transmission according to the requirements of the applications. In addition, the TSN mechanisms used within a network, such as the Time-Aware Scheduler, must be parametrised and configured, including aspects such as cycle times, CoS priorities and time slots for real-time data.

For configuring TSN, IEEE 802 is currently developing three different models (IEEE 802.1Qcc[17]): a centralised model, a decentralised model and a hybrid approach. Common to all three approaches is that the configuration should be automated to a great extent, in order to ensure that handling of TSN configuration remains manageable. One of the requirements is, that end devices can announce their communication requirements and to automatically configure the relevant network elements according to the announced requirements.

The fundamental configuration process of a TSN network is as follows: First, the TSN mechanisms supported within a network are identified and activated as necessary. Next, the sending device, the so-called talker, announces information about the data stream it wants to transmit. This information includes, in particular, identified characteristics such as the target MAC address and CoS priorities. An end device that is interested in a data stream, the so-called listener, can register for and receive the data packets that are associated with the data stream with the aid of the announced information.

The three planned configuration approaches differ from one another in how the requirements are conveyed and processed. In the centralised approach, talkers and listeners communicate over a direct end-to-end connection with a (logical) central configuration instance, the Centralised Network Configuration (CNC) as illustrated in Figure 10. The CNC calculates the time slot for a new data stream based on the information that is present on the network topology and the already assigned resource reservations then it configures the involved network participants accordingly. Protocols such as OPC UA, for instance, can be used for the connection between the talker or the listener and the CNC. The configuration of the switches can be done through existing device management protocols such as SNMP (Simple Network Management Protocol).
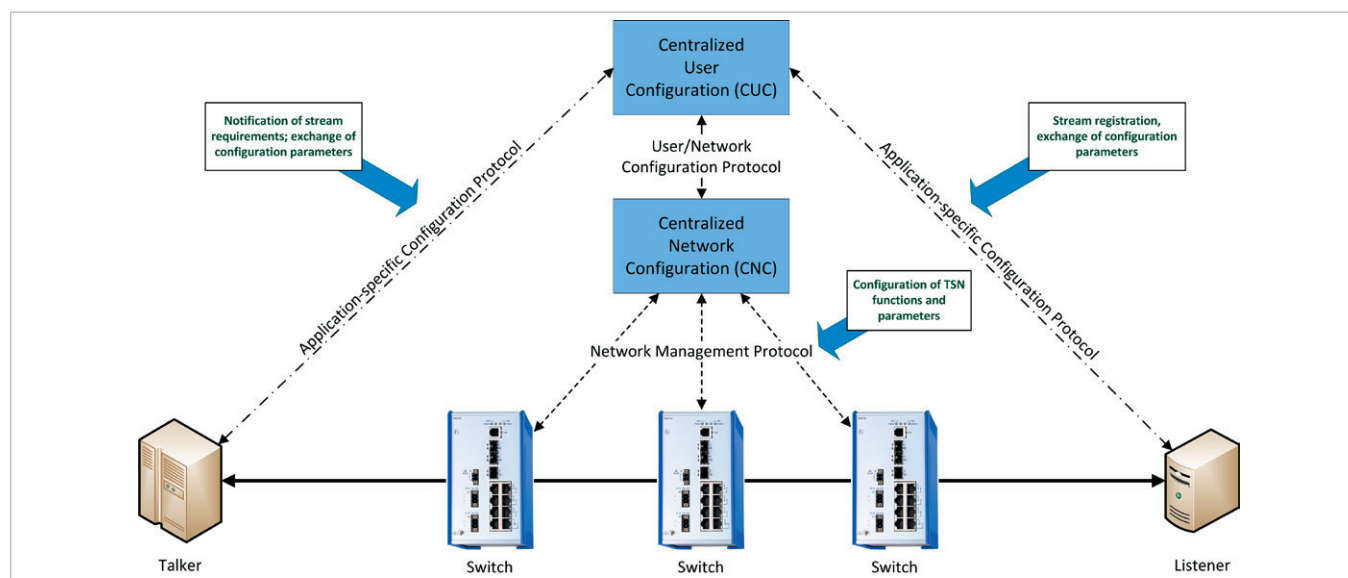

Figure 10: In the centralised TSN configuration approach, the end devices communicate directly with a central configuration instance

With the de-centralised approach, in contrast to the centralised approach, the end device requirements are distributed in the network (see Figure 11). The common configuration of the TSN mechanisms is therefore based on the local information that is present in each device. In this context, IEEE 802 has developed a plan to adapt the Stream Reservation Protocol (SRP) that was developed

for TSN predecessor technology, AVB (Audio- and Video Bridging), to the requirements of TSN.

The hybrid approach represents a unification of the centralised and the decentralised approaches. As with the decentralised approach, the end devices announce their requirements over a decentralised operating protocol. The actual TSN configuration, however,

takes place in a centralised manner, as illustrated in the lower part of Figure 11. An advantage of this method is that end devices only need to support one single configuration protocol, but the network can be managed as centralised or decentralised. However, the expansions of the SRP in IEEE 802 standardisation will be necessary for this approach.
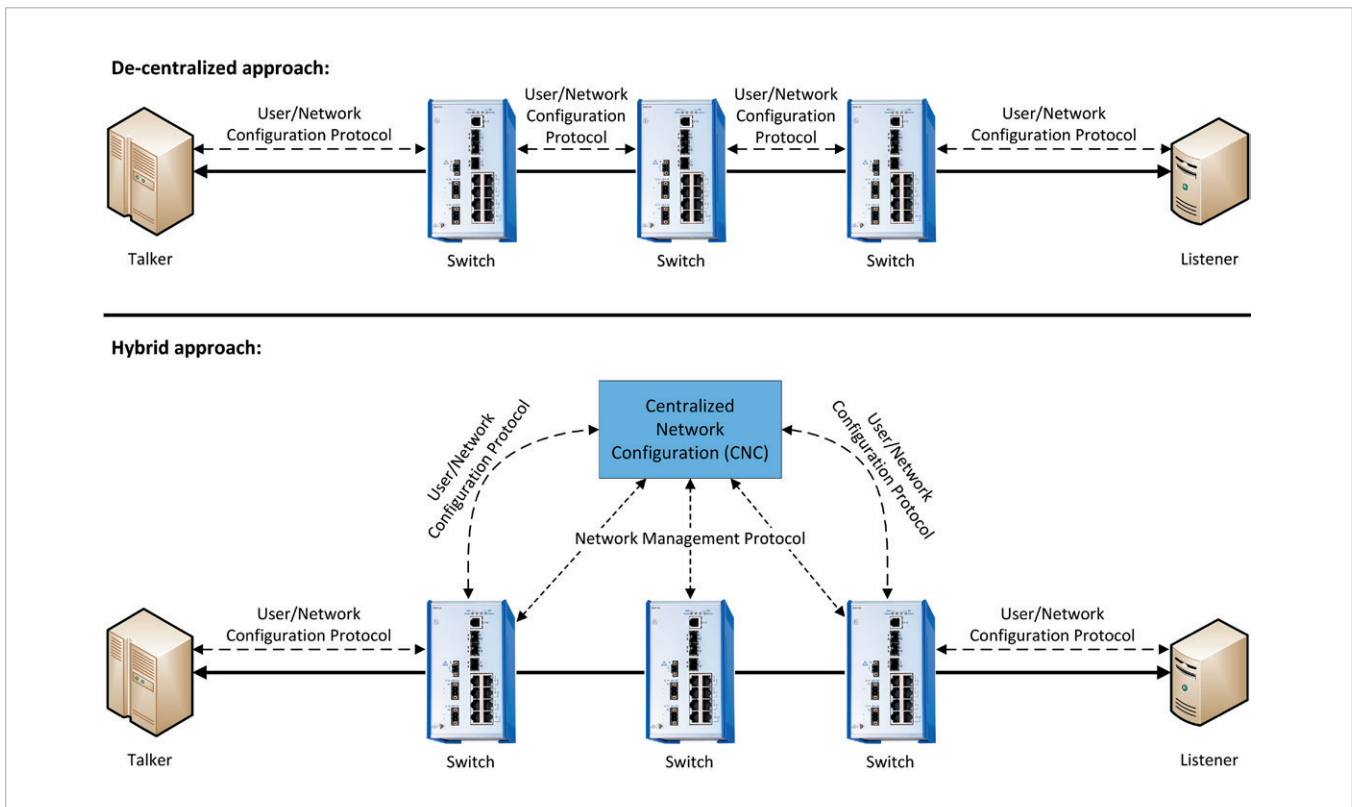

Figure 11: The centralised and hybrid approaches offer a configuration interface to the end devices that is independent of the configuration model

Even though all three of the configuration mechanisms described here are currently still in the standardisation process, it is already possible today to

configure the available TSN mechanisms through standardised interfaces, such as SNMP. This enables the manual engineering of cycle times and the time

slots of the Time-Aware Scheduler by means of a network management tool such as Hirschmann Industrial HiVision (see Figure 12).
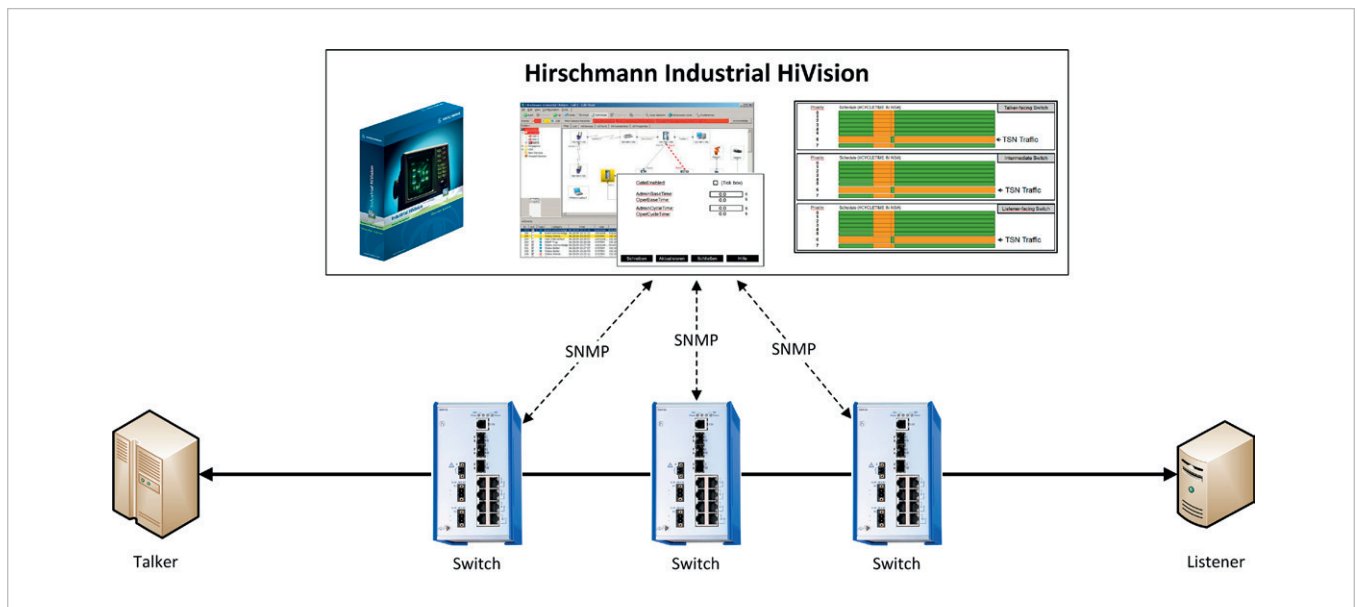
Figure 12: Hirschmann Industrial HiVision makes manual engineering and monitoring of TSN networks possible

Summary and Outlook

With TSN, deterministic data transmission with standardised Ethernet according to IEEE 802.1 and 802.3 is possible for the first time. The operating spectrum of TSN permits its use in various fields of application with, in part, strong differences in requirements for transmission latency, jitter and fault tolerance. The standardisation process in the area of time-sensitive networking is, however, not yet completed and is expected to take a few more years. Accordingly, there are various TSN mechanisms that are currently still in the active standardisation process. It is equally imaginable that additional mechanisms will be added to the already existing TSN family in the future.

Central mechanisms of the TSN protocol family have, however, been completed and have been demonstrated successfully. These mechanisms, such as the Time-Aware Scheduler, can already be integrated in products and their benefits can be used immediately. Equally, through the IEEE 802 standardisation process, backwards compatibility is ensured: TSN networks that are already installed can still be used in the future. Therefore, TSN is no longer a future technology – it's time has come.

References

1. http://www.ieee802.org/1/pages/tsn.html
2. http://ieeexplore.ieee.org/servlet/opac?punumber=7572856
3. http://ieeexplore.ieee.org/servlet/opac?punumber=7553413
4. http://ieeexplore.ieee.org/servlet/opac?punumber=7900319
5. http://ieeexplore.ieee.org/servlet/opac?punumber=7433913
6. http://ieeexplore.ieee.org/servlet/opac?punumber=4579757
7. http://ieeexplore.ieee.org/servlet/opac?punumber=5741896
8. Hirschmann IEEE 1588 White Paper, available at: http://belden.picturepark.com/Website/?Action=viewPdf&AssetId=8050
9. http://www.ieee802.org/1/pages/802.1AS-rev.html
10. http://ieeexplore.ieee.org/servlet/opac?punumber=5375702
11. http://ieeexplore.ieee.org/servlet/opac?punumber=7961301
12. http://www.ieee802.org/1/pages/802.1cr.html
13. http://www.ieee802.org/1/files/public/docs2010/ba-boiger-bridge-latency-calculations.pdf
14. http://ieeexplore.ieee.org/servlet/opac?punumber=8064219
15. http://ieeexplore.ieee.org/servlet/opac?punumber=11085
16. http://www.ieee802.org/1/pages/802.1cb.html
17. http://www.ieee802.org/1/pages/802.1cc.html