



Praxisbericht: Implementierung von TSN-Endpunkten im industriellen Umfeld

Marvin Büchter, Sebastian Wolf

Entwicklung Kommunikationstechnik
Weidmüller Interface GmbH & Co KG
Klingenbergstraße 16
32758 Detmold

marvin.buechter@weidmueller.com

sebastian.wolf@weidmueller.com

Zusammenfassung. Dieser Beitrag beschreibt die praktische Umsetzung eines industriellen Anwendungsfalls in einer heterogenen TSN-Domäne und legt hierbei einen Schwerpunkt auf die Integration von TSN in industrielle Netzwerk-Endgeräte nach dem Automationsprofil der Avnu Alliance. Er stellt heraus, welche technischen Anforderungen TSN-Endpunkte erfüllen sollten und wie diese aktuell mit am Markt verfügbaren Lösungen umgesetzt werden können.

1 Einleitung

Mit fortschreitender IEEE-Standardisierung von Ethernet TSN werden immer mehr vorwettbewerbliche TSN-Lösungen und -Demonstratoren auch für den industriellen Markt sichtbar. Dabei konzentrieren sich die Hersteller von industrietauglichen TSN-Chip-Lösungen derzeit vorrangig auf Infrastruktur-Komponenten.

Dieser Beitrag beschreibt die praktische Umsetzung eines industriellen Anwendungsfalls in einer heterogenen TSN-Domäne und legt hierbei einen Schwerpunkt auf die Integration von TSN in industrielle Netzwerk-Endgeräte nach dem in [GA00] beschriebenen Avnu Alliance Automationsprofil. Er stellt heraus, welche technischen Anforderungen TSN-Endpunkte erfüllen sollten und wie diese aktuell mit am Markt verfügbaren Lösungen umgesetzt werden können.

Dabei gliedert sich der Beitrag wie folgt: Zunächst wird ein industrieller Anwendungsfall vorgestellt, aus dem die Problemstellungen und Anforderungen an einen industriellen TSN-Endpunkt abgeleitet werden. Anschließend wird ein kurzer Überblick über den Stand der TSN-Automatisierungsprofile gegeben. Ausgehend von den Anforderungen des Avnu Alliance Automationsprofiles werden die aktuellen am Markt befindlichen technischen Lösungen dargestellt. Nachfolgend wird die Implementierung einer der Lösungen kurz beschrieben und im Anschluss messtechnisch verifiziert und ausgewertet.

Kenntnisse von Ethernet, Ethernet TSN, FPGAs sowie dem Linux-Betriebssystem werden vorausgesetzt.

2 Ein industrieller Anwendungsfall

Ein Internet of Things (IoT)-Koppler koppelt Feld-nahe IO-Module mit Sensoren und Aktoren an eine übergeordnete Verarbeitungseinheit an. Die Verarbeitungseinheit könnte ein lokaler Edge-PC, eine lokale Cloud oder eine über das Internet erreichbare Cloud-Infrastruktur sein. Sie kennzeichnet sich durch hohe und skalierbare Rechen- und Speicherkapazitäten, die maschinennah so nicht zur Verfügung stehen. Für den Transport der Maschinendaten kommen leichtgewichtige IoT-Protokolle wie MQTT oder AMQP zum Einsatz.

Die Daten werden durch die Verarbeitungseinheit zwischengespeichert und nachfolgend analysiert. Aus diesen Analysen können z. B. Erkenntnisse über die Qualitätsentwicklung der Produktion, die Vorhersage eines möglichen Problems oder einer Energiepitze abgeleitet werden. Die Reaktion auf diese Ergebnisse und damit die Rückwirkung in den Steuerungsprozess der Anlage erfolgt derzeit häufig mit einem variierenden größeren zeitlichen Versatz im Vergleich zum Zeitpunkt, an dem die Maschinendaten zur Analyse vorlagen. Für die Reaktion auf die Vorhersage eines möglichen Ausfalls, der erst in einigen Stunden auftreten wird, reicht dies völlig aus.

Um aber z. B. in schnellen Produktionsprozessen eine durchweg hohe Qualität der erzeugten Güter zu erreichen und damit die Ausschussrate zu senken, sind kontinuierliche Analysen mit direkten schnellen Rückkopplungen in den Produktionsprozess notwendig. Hier werden Maschinendaten nicht zwischen-gespeichert, sondern direkt im Datenstrom analysiert. Diese Art der Analyse wird oft mit dem Stichwort Stream Analytics verknüpft. Dafür ist es notwendig, dass Maschinendaten auch aus verschiedenen Quellen möglichst zeitsynchron und in gleichbleibenden zeitlichen Abständen der Verarbeitungseinheit zur Verfügung gestellt werden. Die Rückkopplungen auf den Steuerungsprozess der Maschinen sollten ebenso schnell und über das selbe Kommunikationsnetzwerk möglich sein.

Aus diesem Anwendungsfall lassen sich drei Probleme ableiten:

- Problem 1: Daten müssen in zeitlich konstanten Intervallen vom IoT-Koppler gesendet werden.
- Problem 2: Datenströme mehrerer IoT-Koppler müssen zeitlich synchronisiert werden.
- Problem 3: Sowohl die IoT-Datenströme als auch die deterministische Rückkopplung durch die Verarbeitungseinheit sollen über das gleiche Netzwerk ermöglicht werden.

Für die Probleme 1 und 2 stellt TSN eine Lösung mit 802.1AS [IE07] und 802.1Qbv [IE09] bereit. Für Problem 3 bietet TSN die Möglichkeit über eine konvergent genutzte Ethernet-TSN-Infrastruktur mehrere Datenströme mit zeitlich konstanten Bedingungen zu übertragen. Ein zusätzliches Netzwerk für die Kontroll-Ströme ist damit nicht erforderlich.

Ziel war es nun, den IoT-Koppler als TSN-Endpunkt zur Teilnahme an einer TSN-Domäne um die grundlegenden für den o. g. Anwendungsfall notwendigen TSN-Mechanismen zu erweitern.

3 TSN-Automatisierungsprofile

Es gibt eine ganze Reihe von Projekten, welche sich mit der Definition eines TSN-Profiles für die industrielle Automatisierungstechnik beschäftigen. Dabei geht es darum, Anwendungsfälle, Anforderungen, Referenz-Modelle für Netzwerke und -Teilnehmer zu definieren sowie Elemente und die zugehörige Konfiguration aus dem Baukasten der IEEE TSN-Standards für diese Anwendungsfälle festzulegen.

Eines der am weitesten inhaltlich fortgeschrittenen Profile stammt von der Avnu Alliance und ist in der Veröffentlichung „Theory of Operation for TSN-enabled Systems“ [GA00] beschrieben. Aufgrund des hohen Reifegrades im Hinblick auf TSN-Endpunkt-Architekturen wurde dieses Profil als Grundlage und Referenz für die Implementierung des TSN-Endpunktes verwendet.

Weitere Projekte, die sich mit der Definition von TSN-Profilen im Automatisierungsumfeld beschäftigen, sind nachfolgend kurz referenziert. Sie werden hier nicht weiter betrachtet.

- Profinet TSN Profile [PI05]
- EtherCAT TSN Communication Profile [WE01]
- Ethernet TSN Nano Profil [SKJ02]
- IEC/IEEE 60802 [II03]
- OPC UA TSN [BBS04]

4 TSN-Endpunkt-Architektur aus dem Profil der Avnu Alliance

Das Whitepaper mit dem Titel „Theory of Operation for TSN-enabled Systems“ der Avnu Alliance [GA00] beschreibt eine generische Systemarchitektur zur heterogenen Nutzung einer TSN-Infrastruktur. Die Avnu strebt an, die Interoperabilität der TSN-Geräte verschiedener Hersteller durch Profilspezifikationen und Zertifizierungsverfahren sicher zu stellen. Um eine Basis für Kommunikationssysteme der Industrie zu bilden, konzentriert sich die Architektur auf drei Mechanismen: Zeitsynchronisation, Quality of Service (QoS) durch Nutzung von zeitlich geplantem Datenverkehr und eine komplett zentralisierte Netzwerk-Konfiguration.

In der beschriebenen Architektur benötigt ein TSN-Endpunkt sechs TSN-spezifische Elemente: CUC Interface, Time-Sensitive Stream Object, Network Interface, End Station Configuration State Machine, Time Synchronisation und Topology Discovery. Diese Elemente und ihre Zusammenhänge sind in [Abbildung 1](#) dargestellt und werden im Folgenden erläutert.

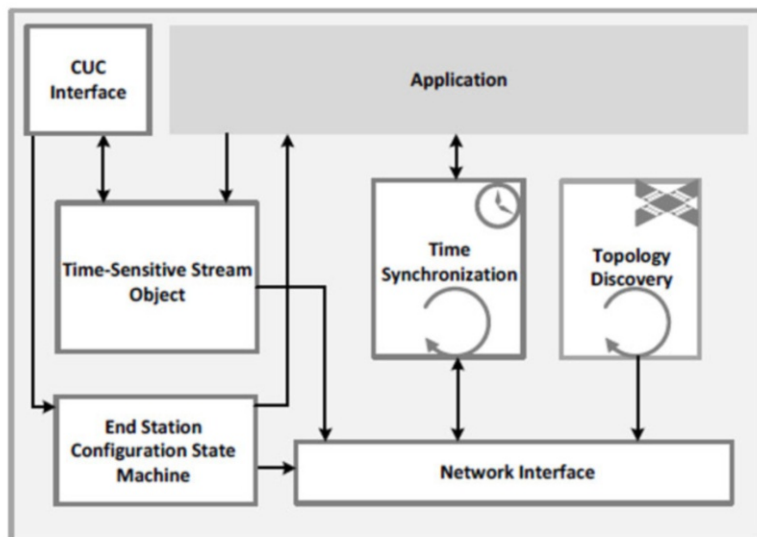


Abb. 1. TSN-Endpunktarchitektur der Avnu Alliance (Quelle: [GA00])

4.1 CUC Interface

Die Systemarchitektur nutzt eine komplett zentralisierte Netzwerkkonfiguration. Demnach existiert eine logische Einheit mit der Bezeichnung Centralized User Configuration (CUC) zur Applikationskonfiguration. Um mit der CUC zu kommunizieren, wird im TSN-Endpunkt ein CUC-Interface benötigt. Über das CUC-Interface werden die Stream-Anforderungen des TSN-Endpunktes der CUC mitgeteilt. Die CUC ermittelt mit der CNC zusammen einen Kommunikationspfad durch die TSN-Domäne. Die relevanten TSN-Parameter werden anschließend dem Endpunkt und den Netzwerkinfrastruktur-Komponenten mitgeteilt.

Zur Kommunikation zwischen CUC und TSN-Endpunkt wird von der IEEE kein Protokoll vorgegeben. Es wird so ermöglicht, dass bestehende Protokolle zur Applikationskonfiguration um die Konfiguration der TSN-Streams erweitert werden können.

4.2 Time-Sensitive Stream Object

Für jeden logischen TSN-Talker und -Listener auf einem TSN-Endpunkt wird ein Time-Sensitive Stream Object angelegt. Dieses Objekt definiert alle für einen zyklischen Datenstrom relevanten Parameter. Die Parameter werden von der Applikation und vom CUC-Interface im Verlauf der TSN-Endpunkt-Konfiguration mit der End Station Configuration State Machine, festgelegt. Nachdem die Parameter des TSN-Datenstroms gesetzt wurden, wird das Netzwerkinterface konfiguriert und die Übermittlung des Datenstroms wird gestartet.

4.3 Network Interface

Das Netzwerkinterface der TSN-Endpoint-Architektur besteht wie in [Abbildung 2](#) zu sehen aus drei Teilen: einer Stream Translation, einer Time-Sensitive Queue und einer Best Effort Queue. Über die Best Effort Queue wird der konventionelle Datenverkehr, welcher zeitlich unbestimmt gesendet und empfangen wird, verwaltet. Die Frames, welche für den TSN-Datenstrom bestimmt sind, werden, wie nachfolgend beschrieben, über die Stream Translation in TSN-Frames übersetzt und anschließend von der Time-Sensitive Queue zu einem bestimmten Zeitpunkt im Kommunikationszyklus gesendet.

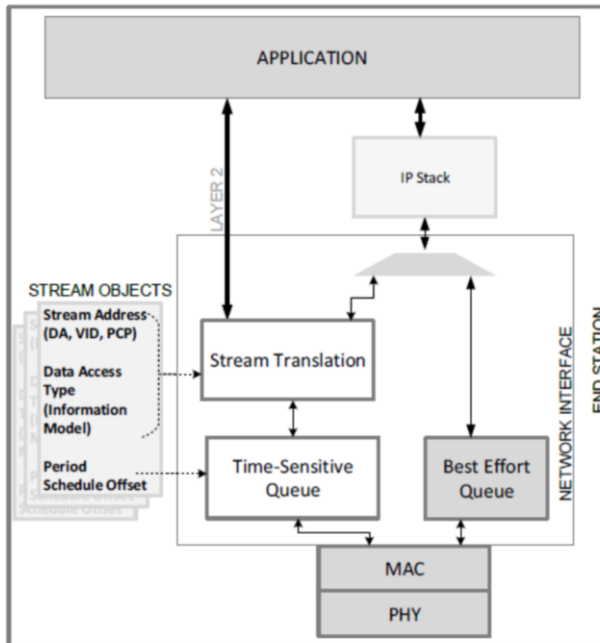


Abb. 2. Netzwerkinterface der Avnu TSN-Endpoint-Architektur (Quelle: [GA00])

Stream Translation.

Zur Anbindung der Applikation an das Netzwerkinterface sind zwei Varianten aufgeführt. In einer Variante werden von der Applikation Layer-2-Frames bereitgestellt und in der zweiten Variante durchlaufen die Daten der Applikation einen IP-Stack und werden in IP-Frames gekapselt. Für beide Varianten wird jeweils ein Verfahren zur Übersetzung der Daten in TSN-Streams genannt.

Das Verfahren zur Übersetzung von Layer-2-Frames in TSN-Frames entspricht der Active Destination MAC and VLAN Stream Identification aus dem Standard IEEE 802.1CB [IE08]. Es wird die Destination MAC Address, der VLAN Identifier und die Priorität geändert, sodass der Frame zu dem TSN-Datenstrom mit diesen Parametern gehört.

Das Verfahren zur Übersetzung von IP-Paketen entspricht einer Kombination aus der IP Stream Identification und der Active Destination MAC and VLAN Stream Identification aus dem Standard IEEE 802.1CB. Diese Kombination wurde auch im Entwurf des Standards IEEE P802.1Qcc [IE06] zur Implementierung in einem TSN-Talker vorgeschlagen. Ob ein Frame zu einem TSN-Datenstrom gehört, wird anhand der Parameter des IP-Headers geprüft. Anschließend wird die Destination MAC Address, der VLAN Identifier und die Priorität des Frames geändert, sodass der Frame zu dem TSN-Datenstrom mit diesen Parametern gehört.

Time-Sensitive Queue.

Die Time-Sensitive Queue steuert das Senden der TSN-Frames in Abhängigkeit der im Time-Sensitive Stream Object konfigurierten Parameter. Da für jeden logischen TSN-Talker ein individuelles Time-Sensitive Stream Object mit individueller Zykluszeit und Sendezeitpunkt existiert, entspricht dieses Modell dem Per-Stream Scheduling aus dem Entwurf des Standards IEEE P802.1Qcc.

4.4 End Station Configuration State Machine

Die End Station Configuration State Machine beschreibt die vier Zustände Init, Configuration, Ready und Running. Diese Zustände werden durchlaufen, um einen Endpunkt im Zusammenspiel mit einem CUC als TSN-Talker oder TSN-Listener zu konfigurieren.

4.5 Time Synchronisation

Innerhalb der TSN-Domäne gibt es eine gemeinsame Zeitbasis. Mit der gemeinsamen Zeitbasis wird der geplante Netzwerkverkehr in einer TSN-Domäne koordiniert und die Synchronisation verteilter Applikationen ermöglicht. Die Architektur der Avnu Alliance nutzt das Generalized Precision Time Protocol (gPTP)-Profil des Standards IEEE 802.1AS [IE07] zur Synchronisation der gemeinsamen Zeitbasis innerhalb der Domäne und weist auf die Weiterentwicklung des Standards im Projekt P802.1AS-Rev der IEEE TSN Task Group hin.

4.6 Topology Discovery

Damit die Datenströme innerhalb der TSN-Domäne durch eine CNC sinnvoll geplant werden können, benötigt diese eine Übersicht über die gesamte Netzwerktopologie. Per Link Layer Discovery Protocol (LLDP) teilt jeder Teilnehmer einer TSN-Domäne seinem Nachbarn seine Identität mit. Jeder Switch weiß also, welcher direkte Nachbar über welchen Port an dem Gerät angeschlossen ist. Diese Informationen kann die CNC von jedem Switch der Domäne abrufen und daraus eine Übersicht über die gesamte Netzwerktopologie erstellen. Ein reiner TSN-Endpunkt muss demnach nicht mit der CNC kommunizieren, er muss lediglich dem direkten Nachbarswitch aktiv seine Identität per LLDP mitteilen.

4.7 Technische Anforderungen an TSN-Endpunkte

In der in diesem Beitrag beschriebenen Umsetzung liegt der Fokus auf den folgenden drei Elementen der Avnu-TSN-Endpunkt-Architektur: Time Synchronisation, Stream Translation und Time-Sensitive Queue.

Time Synchronisation: Der TSN-Endpunkt sollte eine PTP-Hardwareuhr inkl. Softwarestack mit Unterstützung des gPTP-Profiles vom Standard IEEE 802.1AS implementieren. Die Hardwareuhr sollte einen Jitter im niedrigen zweistelligen Nanosekundenbereich aufweisen.

Stream Translation: Der TSN-Endpunkt sollte eine Einheit umsetzen, welche Datenstrom-Übersetzung durch aktive Datenstrom-Identifikation unterstützt. Datenströme sollten anhand Layer-2- oder Layer-3-IP-Parameter identifiziert und die Felder Destination-MAC-Address, VLAN Identifier und Priorität eines Ethernet-Frames passend modifiziert werden können.

Time-Sensitive Queue: Der TSN-Endpunkt sollte eine Einheit für das zeitgesteuerte Einspeisen der Frames eines TSN-Datenstroms umsetzen. Diese Einheit sollte einen TAS (Time-Aware Shaper) mit Per-Stream Scheduling ([Abbildung 1](#)) ermöglichen, welcher Frames zu einem bestimmten Offset im Kommunikationszyklus mit geringem Jitter einspeisen kann. Für zukünftige Anwendungen sollten mehrere voneinander isolierte TSN-Datenströme konfiguriert werden können und die Anzahl der unterstützten Datenströme skalierbar sein.

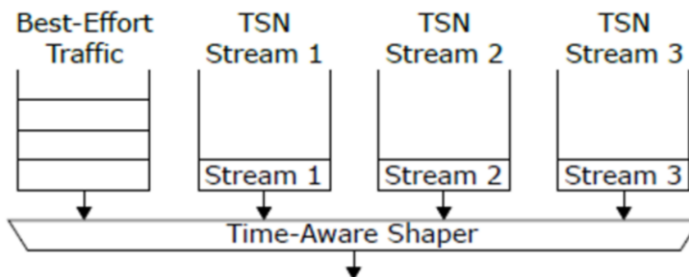


Abb. 3. Prinzip eines Per Stream Scheduling Verfahrens

Integrierter Switch bei 2-Port-Endgerät: Besitzt das Endpunkt-Gerät zwei Ethernet-Ports ist es sinnvoll einen integrierten TSN-Switch mit umzusetzen. Geräte mit zwei externen Ports werden in der Industrie gerne genutzt, um eine Linientopologie aufzubauen. Zwei oder mehr externe Ports sind bei einem TSN-Endpunkt-Gerät außerdem sinnvoll, wenn mit dem System das Redundanzverfahren Frame Replication and Elimination for Reliability (FRER) nach Standard IEEE 802.1CB genutzt werden soll. Denn so können die Pakete schon im TSN-Endpunkt-Gerät vervielfältigt und in verschiedene Netzwerk-Pfade eingespeist werden.

5 Verfügbare Lösungen und Umsetzung

Viele Hersteller am Markt adressieren Infrastruktur-Komponenten und konzentrieren sich vor allem auf Switch-Lösungen. Es sind Lösungen in Form von Application-Specific Integrated Circuits (ASICs) und in Form von Intellectual Property (IP)-Cores für FPGAs verfügbar. Als ASIC wurden Lösungen von Analog Devices, Broadcom, Hilscher, NXP und Renesas betrachtet. Als IP-Core wurden Lösungen von TTTech und Xilinx betrachtet.

Als kritische Anforderung stellte sich die Time-Sensitive Queue insbesondere mit Ihrem geforderten Merkmal „Per-Stream Scheduling“ heraus. Dieses wird aktuell nur bei den Lösungen von Hilscher und Xilinx als hardwaregestützte Funktion angeboten. Für ein zukunftssicheres Design, welches in der Lage ist, mehr als nur einen logischen Time-Aware TSN-Talker zu implementieren wird der Per-Stream Scheduling Mechanismus benötigt. Diese beiden Lösungen werden im Folgenden näher beschrieben und verglichen.

5.1 Hilscher NetX 51

Hilscher bietet mit dem NetX eine Netzwerkcontroller Familie an, die durch ein Firmwareupdate um TSN-Mechanismen erweitert werden kann [HI10]. Um ein bestehendes Produkt um TSN zu erweitern, kann ein NetX-Baustein als Netzwerkcontroller für eine auf einer separaten Host-CPU befindlichen Applikation dienen. Je nach Anwendung und insbesondere bei Neuentwicklungen kann ein NetX in Form einer Komplettlösung als Applikationscontroller inkl. Netzwerkcontroller eingesetzt werden.

In TSN-Demonstrationen und als Angebot zur Evaluation wird von Hilscher aktuell der NetX 51 Baustein verwendet. An allgemeinen Netzwerkfunktionen besitzt dieser einen integrierten 3-Port Switch, zwei integrierte MACs und zwei integrierte PHYs. Die Übertragungsrate des NetX 51 ist aktuell auf 100 MBit/s begrenzt. Mit dem NetX 4000 ist jedoch auch ein 4-Port Netzwerkcontroller mit Gigabit- und TSN-Support angekündigt.

In der aktuellen TSN-Firmware sind folgende TSN-Funktionen implementiert:

- IEEE 802.1AS Timing and Synchronisation
- Time-Sensitive Queue: Zeitgesteuertes Einspeisen der Frames eines TSN-Datenstroms.
- IEEE 802.1Qbv (Switch): Enhancements for Scheduled Traffic

Für die Zeitsynchronisation nach dem Standard IEEE 802.1AS steht eine integrierte PTP-Hardwareuhr zur Verfügung. Zur Regelung der Uhr wird auf dem Evaluierungsboard der Open Source Softwarestack vom LinuxPTP Project [SF11] genutzt.

In der Time-Sensitive Queue dieser Implementierung können vier Datenströme konfiguriert werden. Laut Hersteller ist die Anzahl der Warteschlangen sowie die Anzahl der konfigurierbaren Datenströme nicht durch die Hardware begrenzt und kann auf Kundenwunsch skaliert werden. Eine Stream Translation Einheit zur Übersetzung zwischen regulären Ethernet-Frames und TSN-Datenstrom-Frames ist aktuell nicht

vorhanden. Die Übersetzung müsste entweder vom Anwender selbst als eigene Einheit oder als Teil der Applikation implementiert werden.

5.2 Xilinx 100M/1G TSN Subsystem IP

Mit dieser IP bietet Xilinx die Möglichkeit, ausgewählte SoCs um TSN zu erweitern. Zu den unterstützten Produktfamilien zählen die Zynq-7000 und Zynq Ultra-Scale+ MPSoC SoCs von Xilinx [XI12]. Die IP kann wahlweise als TSN-Endpunkt oder als TSN-Endpunkt mit 3-Port Switch konfiguriert werden. Die Anbindung an eine bzw. zwei externe PHYs kann wahlweise über ein Gigabit Media Independent Interface (GMII)- oder ein RGMII-Interface erfolgen.

Aktuell kann die IP zur Implementierung folgender TSN-Funktionen genutzt werden:

- IEEE 1588, 802.1AS und 802.1AS-rev: Timing and Synchronization
- Stream-Translation: Übersetzung zwischen regulären Ethernet-Frames und TSN-Datenstrom-Frames.
- Time-Sensitive Queue: Zeitgesteuertes Einspeisen der Frames eines TSN-Datenstroms.
- IEEE 802.1Qbv (Switch): Enhancements for Scheduled Traffic
- IEEE 802.1Qci (Switch): Per-Stream Filtering and Policing
- IEEE 802.1CB (Switch): Frame Replication and Elimination for Reliability

Für die Zeitsynchronisation nach dem Standard IEEE 802.1AS stellt diese Lösung eine PTP-Hardwareuhr im FPGA zur Verfügung. Zur Regelung der Uhr wird auch bei dieser Lösung der Open Source Softwarestack vom LinuxPTP Project empfohlen. Die Time-Sensitive Queue wird durch eine separate IP von Xilinx namens Time-Aware-DMA realisiert. Je nach IP-Konfiguration kann diese 4 bis 256 TSN-Streams bedienen und bildet für jeden Stream eine eigene Warteschlange ab.

Zur Stream Translation gibt es bei der Lösung von Xilinx zwei Möglichkeiten. Die erste Möglichkeit ist die Übersetzung im TSN-Switch. Hierzu wird der Adress-Lookup Mechanismus genutzt, um während dem Suchvorgang des Ziel-Ports zeitgleich den Frame zu übersetzen. Diese Art der Übersetzung entspricht der Active Destination MAC and VLAN Stream Identification des Standards IEEE 802.1CB.

Die Zweite Möglichkeit ist die Übersetzung im IP-Stack des Betriebssystems. Hierzu bietet Xilinx für Linux ein Kernelmodul an, welches jedes IP-Paket, das den IP-Stack des Linux-Systems durchläuft, analysiert und bei dem Zutreffen bestimmter IP-Parameter übersetzt. Diese Art der Übersetzung entspricht einer Kombination aus der IP Stream Identification und der Active Destination MAC and VLAN Stream Identification des Standards IEEE 802.1CB.

5.3 Vergleich der Lösungen

Nach der Betrachtung der beiden relevanten Lösungen werden diese in [Tabelle 1](#) anhand der geforderten TSN-Endpunkt-Anforderungen verglichen. Der NetX-Netzwerkcontroller von Hilscher sowie die TSN Subsystem IP von Xilinx eignen sich gleichwertig als Grundlage zur Implementierung eines TSN-Endpunktes in Anlehnung an die Avnu TSN-Endpunkt-Architektur. Für die Umsetzung wurde die Lösung von Xilinx gewählt, da der IoT-Koppler auf einem Xilinx-Zynq 7020 SoC aufbaut und die IP dort gut integriert werden konnte.

Tabelle 1. Vergleich der Lösungen von Hilscher und Xilinx

Anforderung	Hilscher NetX51	Xilinx TSN Subsystem IP
Time Synchronisation 802.1AS, Hardware-Zeitstempel	vorhanden	vorhanden
Stream Translation	nicht vorhanden	vorhanden (Übersetzung anhand von IP- Parametern möglich)
Time-Sensitive Queue für TSN-Streams mit Per-Stream Scheduling	≥ 4 TSN-Streams	4 – 256 TSN-Streams
TAS nach 802.1Qbv für TSN und Best Effort Traffic	≥ 2 Queues	2 oder 3 Queues

5.4 Umsetzung

Die Xilinx-IP wurde in den FPGA des Xilinx SoC 7020 integriert und an eine der beiden internen CPUs angeschlossen. Weiterhin wurden die Xilinx-Treiber und Software-Tools in das Linux-System des IoT-Kopplers integriert.

Die **Time-Synchronisation** erfolgte auf Basis einer PTP-Hardwareuhr. Mit dem Xilinx-Treiber für die PTP-Hardwareuhr wurde diese in das PTP Hardware Clock (PHC)-Subsystem des Linux-Betriebssystems integriert. Dort kann die PTP-Hardwareuhr mittels grundlegender Funktionen aus dem Linux-Betriebssystem bedient und geregelt werden. Zur Regelung der PTP-Hardwareuhr wurde der Open Source Softwarestack vom LinuxPTP Projekt genutzt [SF11].

Die **Stream-Translation** erfolgte im von Xilinx bereitgestellten Kernel-Modul.

Die **Time Sensitive Queue** und der TAS mit Per Stream Scheduling wurden im IP-Core umgesetzt und mittels diverser Interrupts und DMAs an die SoC-CPU angebunden.

6 Auswertung

Um die Funktion der integrierten TSN-Mechanismen beurteilen zu können, wurden Messungen bezüglich der Zeitsynchronisation, der Übersetzung von Datenströmen und der Sendemechanismen durchgeführt.

6.1 Messung der Zeitsynchronisation

Zur Messung der Zeitsynchronisation wurde ein Takt der PTP-Hardwareuhr über IO-Pins des IoT-Kopplers ausgegeben. Zwei IoT-Koppler wurden direkt verbunden, wobei jeder Koppler einmal als PTP-Grandmaster und -PTP-Slave eingestellt wurde. Die Takte der Koppler wurden auf ein externes Oszilloskop geführt, um die Abweichung zwischen den verschiedenen synchronisierten PTP-Hardwareuhren der Koppler messen zu können.

Im Ergebnis ließ sich erkennen, dass die Synchronisation in der Rolle als PTP-Grandmaster, sowie als PTP-Slave, funktioniert und die zeitliche Abweichung als mathematische Standardabweichung ermittelt 26,46 ns betrug.

6.2 Prüfung der Übersetzung von Datenströmen

Zur Überprüfung der Stream Translation wurden UDP-Pakete im Linux System eines als TSN-Talker fungierenden IoT-Kopplers mit dem Tool netcat generiert. Bei einem als TSN-Listener fungierenden Koppler wurden alle eintreffenden Pakete mit dem Tool tcpdump mitgeschnitten. Nachdem die Vermittlung der Pakete sichergestellt wurde, wird im Talker die Stream Translation für diese IP-Pakete konfiguriert. Anschließend wurde anhand eines weiteren Mitschnitts vom Listener geprüft, ob die Parameter des Ethernet-Frames korrekt modifiziert wurden.

Bei Funktionsprüfungen mit verschiedenen Konfigurationen der IP-Parameter sind keine Fehler aufgetreten. Während der Prüfung ist jedoch aufgefallen, dass das verwendete Kernelmodul zur Datenstrom-Übersetzung nur in Send-Richtung arbeitet, eine Rückübersetzung beim Listener kann hiermit nicht realisiert werden. Wird der TSN-Switch des IP-Cores verwendet, kann mit ihm eine Übersetzung in Send- und Empfangsrichtung auf ISO/OSI Schicht-2 umgesetzt werden.

6.3 Prüfung der Sendemechanismen

Time Sensitive Queue.

Zur Prüfung der Sendemechanismen werden die Sendezeitpunkte eines Ethernet-Frames benötigt. Hierzu wurde das Custom Hardware-Timestamp Feature des Xilinx IP-Cores verwendet. Beim Senden oder Empfangen eines Ethernet-Frames kann mit dieser Funktion die aktuelle PTP-Zeit per Hardware an eine vorbestimmte Stelle des Frames geschrieben werden. Die hierzu verwendeten Zeitpunkte werden jeweils beim Durchlaufen des Start Frame Delimiter (SFD) in der programmierbaren Logik direkt

vor dem PHY aufgenommen (siehe [Abbildung 4](#)). Die Auswertung dieser Zeitstempel wurde mit dem Programm Wireshark durchgeführt.

Der Xilinx IP Core besitzt einen TAS zur Isolierung von Best-Effort-, AVB- und TSN-Datenverkehr. Bei diesem Mechanismus muss geprüft werden, ob der TSN-Datenverkehr von anderem, von dem Endpunkt erzeugten, Datenverkehr optimal isoliert wird.

Um die ordnungsgemäße TAS-Funktion zu prüfen, werden von einer Testapplikation mehr Pakete generiert, als in einem geplanten Zeitschlitz übertragen werden können. Anhand von Sende- und Empfangszeitstempeln kann überprüft werden, ob Frames nur in dem für sie vorgesehenen Zeitschlitz vermittelt werden.

Die TAS-Konfiguration war dabei wie folgt: das Gate der Warteschlange für zeitlich geplanten Datenverkehr wird abhängig von der PTP-Netzwerkzeit in einem TAS-Zyklus mit einer Zykluszeit von 1ms zu Beginn eines jeden Zyklus für $100\mu\text{s}$ geöffnet (siehe [Abbildung 5 \(a\)](#)). Von einer Applikation werden bewusst mit einer längeren Zykluszeit von 1s jeweils 12 Frames mit einer Größe von 150 Byte in die Warteschlange für zeitlich geplanten Datenverkehr eingefügt. Ein Frame mit einer Größe von 150 Byte benötigt bei einer Übertragungsrate von 100 MBit/s etwa $13,6\mu\text{s}$ zur Übertragung. Sieben solcher Frames benötigen $7 \times 13,6\mu\text{s} = 95,2\mu\text{s}$ zur Übertragung, das heißt, dass Frame 8 bis 12 erst in dem $100\mu\text{s}$ Zeitschlitz des nächsten TAS-Zyklus übertragen werden können.

Wie erwartet und in [Abbildung 5 \(b\)](#) dargestellt wurden sieben Frames in dem ersten TAS-Zyklus und die restlichen fünf Frames in dem zweiten TAS-Zyklus vermittelt. Es wurden keine Frames vermittelt, welche zeitlich über ihr Zeitfenster hinausragen. Die Isolierung der verschiedenen Arten von Datenverkehr war damit erfolgreich verifiziert.

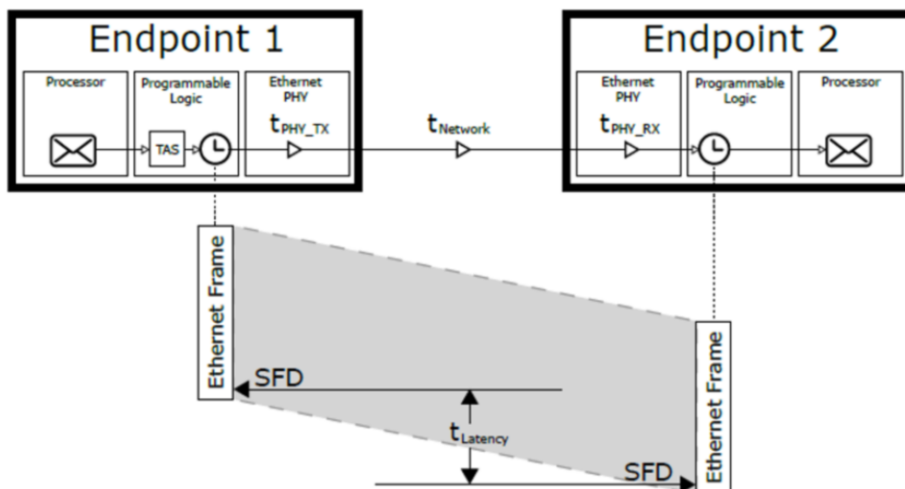


Abb. 4. Aufnahme der Sende- und Empfangszeitpunkte

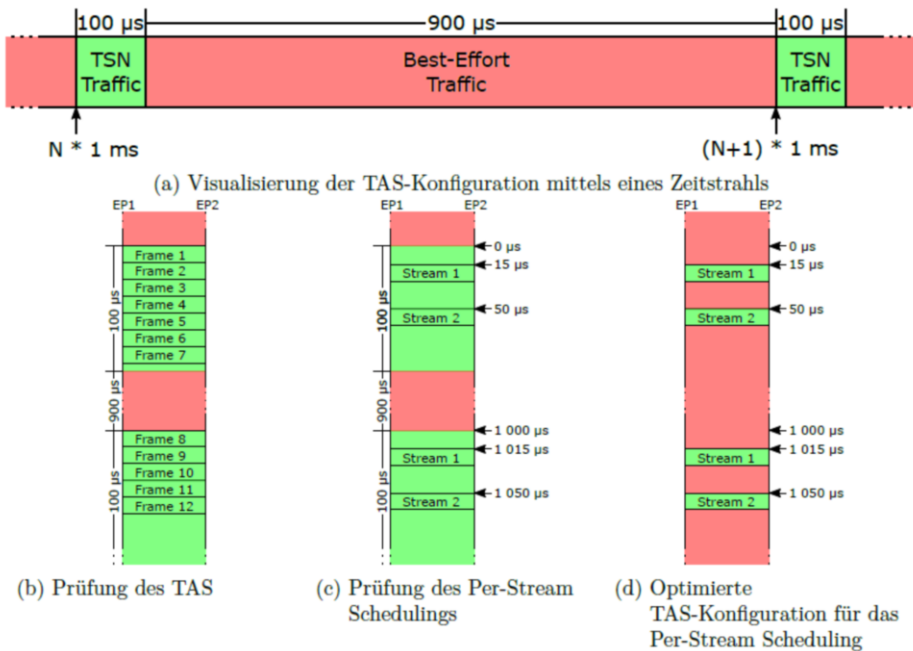


Abb. 5. Prüfung der Sendemechanismen

Per-Stream Scheduling.

Der Xilinx-IP-Core implementiert auch den Per-Stream Scheduling Mechanismus. In diesem Modell besitzt jeder TSN-Datenstrom eine eigene logische Warteschlange und es werden nur eine vorbestimmte Anzahl an Frames zu einem vorbestimmten Offset des TAS-Zyklus in den Sendepfad des TSN-Datenverkehrs eingespeist. Somit wird erreicht, dass sich die TSN-Datenströme untereinander nicht beeinflussen können.

Außerdem sollten die TSN-Frames nun mit einem sehr geringen Jitter zu einem konfigurierten Zeitpunkt von dem TSN-Endpunkt in die TSN-Domäne eingespeist werden. Dieser geringe Jitter wird im Standard IEEE 802.1Qcc gefordert, damit bei der Planung der TSN-Datenströme, z. B. durch eine CNC, möglichst geringe Toleranzen berücksichtigt werden müssen.

Bei diesem Mechanismus muss überprüft werden, ob die Sendezeitpunkte der Frames eines Datenstroms den konfigurierten Zeitpunkten entsprechen und ob der Jitter der gesendeten Frames tatsächlich minimiert wird.

Um diesen Sendemechanismus zu prüfen, werden zwei TSN-Streams konfiguriert. Das heißt, dass in einem TAS-Zyklus ein Frame von Datenstrom 1 mit einem TAS-Offset von 15µs und ein Frame von Datenstrom 2 mit einem TAS-Offset von 50µs gesendet werden (siehe [Abbildung 5 \(c\)](#)).

Anschließend werden im TSN-Talker mit der Applikation packETH im Millisekundentakt Frames für beide Streams erzeugt. Mit der Applikation tcpdump werden in 24

Sekunden 24000 Pakete von jedem Stream bei dem TSN-Listener mitgeschnitten. Anhand der eingefügten Zeitstempel in den Frames werden die Sende- und Empfangszeitpunkte untersucht.

Im Ergebnis wurden die beiden TSN-Streams sehr gut voneinander isoliert. Im Mittel wurde die Übermittlung jedoch etwa $2,3\mu\text{s}$ vor dem geplanten Zeitpunkt begonnen und besitzt eine Standardabweichung von unter 50ns.

Um dieses verbleibende Problem zu lösen, kann eine optimierte Konfiguration des TAS angewandt werden. Dabei muss der TAS so konfiguriert werden, dass das Gate für den zeitlich geplanten TSN-Datenverkehr immer nur zum exakten Sendezeitpunkt eines TSN-Datenstroms geöffnet wird und auch nur so lange geöffnet bleibt, wie ein zu sendender Frame maximal für die Übertragung benötigt (siehe [Abbildung 5 \(d\)](#)).

Im Ergebnis lässt sich festhalten, dass mit der Xilinx TSN-IP die Avnu TSN-Endpunkt-Architektur in einem industriellen IoT-Koppler sehr gut umgesetzt werden konnte.

Literatur

- [GA00] Gardiner, E.: Theory of Operation for TSN-enabled Systems, Avnu Alliance™ Best Practices, 2017, <https://avnu.org/knowledgebase/theory-of-operation>, Stand 03.10.2018.
- [WE01] Weber, K.: EtherCAT and TSN - Best Practices for Industrial Ethernet System Architectures, EtherCAT Technology Group, 2018
- [SKJ02] Schriegel, S.; Kobzan, T.; Jasperneite, J.: Investigation on a distributed SDN control plane architecture for heterogeneous time sensitive networks, In: 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), 2018, S. 1–10
- [II03] IEC/IEEE: 60802 TSN Profile for Industrial Automation, <https://1.ieee802.org/tsn/iec-ieee-60802-tsn-profile-for-industrial-automation>, Stand 17.08.2018
- [BBS04] Bruckner, D.; Blair, R.; Stanica, M-P.; Ademaj, A.; Skeffington, W.; Kutscher, D.; Schriegel, S.; Wilmes, R.; Wachswender, K.; Leurs, L.; Seewal, M.; R., Hummen; Liu, E-C.; Ravikumar, S.: OPC UA TSN - A new Solution for Industrial Communication, Shaper Group, 2017
- [PI05] PROFIBUS & PROFINET International: Integration von TSN in PROFINET schreitet voran, <https://www.profibus.com/newsroom/news/integrationvon-tsn-in-profinet-schreitet-voran>, Stand: 17.08.2018
- [IE06] IEEE P802.1Qcc/D2.3: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements, New York: IEEE, 2018
- [IE07] IEEE 802.1AS-2011: Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks, New York: IEEE, 2011
- [IE08] IEEE 802.1CB-2017: Frame Replication and Elimination for Reliability, New York: IEEE, 2017
- [IE09] IEEE 802.1Qbv-2015: Enhancements for Scheduled Traffic. New York: IEEE, 2015

- [IE10] IEEE 802.1Qbu-2016: Frame Preemption. New York: IEEE, 2016
- [HI10] Hilscher: Die Zukunft der Industriellen Kommunikation - TSN als Firmware Update, <https://www.hilscher.com/de/news/die-zukunft-der-industriellen-kommunikation-tsn-als-firmware-update/>, Stand: 14.08.2018
- [SF11] SOURCEFORGE: The Linux PTP Project. <http://linuxptp.sourceforge.net>, Stand: 27.05.2018
- [XI12] XILINX: Xilinx 100M/1G TSN Subsystem, <https://www.xilinx.com/products/intellectual-property/1gtsn.html>, Stand: 26.05.2018

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.

