INSTITUT FÜR AUTOMATION UND KOMMUNIKATION E.V. MAGDEBURG
AN-INSTITUT DER OTTO-VON-GUERICKE-UNIVERSITÄT MAGDEBURG

KOMMA 2021

Kommunikation in der Automation

TAGUNGSBAND

18.11.2021 | 12. JAHRESKOLLOQUIUM

« KOMMUNIKATION IN DER AUTOMATION »

ULRICH JUMAR, JÜRGEN JASPERNEITE (HRSG.)

IN VERBINDUNG MIT DEM

INDUSTRIAL RADIO DAY | 17.11.2021

INDUSTRIAL RADIO LAB Germany

■ EINE KOOPERATION VON: ifak | inIT | TH OWL

■ UNTERSTÜTZT VON: VDE ITG | GI GESELLSCHAFT FÜR INFORMATIK

Impressum

12. Jahreskolloquium

« KOMMUNIKATION IN DER AUTOMATION»
(KOMMA 2021)

18.11.2021 • Magdeburg

AUFLAGE
70 Exemplare

ISBN 978-3-948749-10-1

DOI: http://dx.doi.org/10.25673/39548

HERAUSGEBER

Ulrich Jumar, Magdeburg

Institut für Automation und
Kommunikation e.V. Magdeburg
An-Institut der Otto-von-
Guericke-Universität Magdeburg
Werner-Heisenberg-Straße 1
39106 Magdeburg

Telefon: +49 (0)391 990140
Fax: +49 (0)391 9901590
Internet: www.ifak.eu

KOMMA 2021
Kommunikation in der Automation

Vorwort

12. Jahreskolloquium Kommunikation in der Automation – KommA 2021

Am 18. November 2021 fand im virtuellen Format am Institut ifak das 12. Jahreskolloquium der Reihe KommA – Kommunikation in der Automation statt. Das abwechselnd von den Instituten inIT, Lemgo und ifak, Magdeburg veranstaltete Kolloquium ist ein bewährtes Forum für Wissenschaft und Industrie zu allen technisch-wissenschaftlichen Fragen rund um die industrielle Kommunikation. Die Veranstaltung wird durch die ITG und die Gesellschaft für Informatik unterstützt. Dem Kolloquium vorgeschaltet war im Jahr 2021 der vom Industrial Radio Lab Germany unter Leitung des ifak organisierte Industrial Radio Day.

Im Fokus des Jahreskolloquiums stehen Kommunikationssysteme – vom Feldbus, über Echtzeit-Ethernet bis zur drahtlosen Kommunikation und der Nutzung von IoT-Technologien. Bei der Systemanalyse und dem Entwurf von Kommunikationssystemen widmen sich die Tagungsbeiträge der formalen Modellierung, der Verifikation und Validierung sowie Interoperabilität, Konformität und Test. Als bedeutsame Aspekte vernetzter eingebetteter Echtzeitsysteme behandelt das KommA-Kolloquium u.a. die Dienstgüte, semantische Interoperabilität, Safety und Security, die Systemintegration und das Engineering. Neben aktuellen Kommunikationstechnologien bilden die vielfältigen Anwendungsbereiche der industriellen Kommunikation einen Schwerpunkt.

Unbenommen vom reichhaltigen Veranstaltungsangebot auf dem Gebiet der automatisierungstechnischen Kommunikation hat das Jahreskolloquium KommA seinen speziellen Platz. Neben den großen internationalen Tagungen wird bewusst der Austausch im kleinen Kreis gepflegt. Bei aller Technologie- und Anwendungsorientierung des Kolloquiums steht die Wissenschaftlichkeit im Fokus. Alle Beitragskurzfassungen werden deshalb von mindestens zwei Vertretern des Programmkomitees begutachtet. Der Verzicht auf parallele Sitzungen befördert den Gedankenaustausch.

Damit trotz der kurzen Tagungsdauer die wichtigsten Themen, Entwicklungen und Trends adressiert werden können, gehören neben den Vorträgen zusätzlich Poster zum Programm. Wie im Fall der Vorträge finden sich auch die Vollmanuskripte der als Poster präsentierten Beiträge im elektronischen Tagungsband.

Prof. Dr. Ulrich Jumar

ifak - Institut für Automation und Kommunikation e.V. an der Otto-von-Guericke-Universität Magdeburg

Prof. Dr. Jürgen Jasperneite

inIT - Institut Industrial IT TH Ostwestfalen-Lippe, Lemgo

Inhaltsverzeichnis

KommA 2021 – Kommunikation in der Automation

http://dx.doi.org/10.25673/39549

MQTT, OPC UA und PROFINET für IIoT

Gunnar Leßmann, Sebastian Schriegel
Phoenix Contact Electronics GmbH, Fraunhofer IOSB-INA
glessmann@phoenixcontact.com, sebastian.schriegel@iosb-ina.fraunhofer.de

Abstract: Das industrielle Internet der Dinge (IIoT) basiert unter anderem auf der Möglichkeit, Daten von Feldgeräten unabhängig von der Prozessteuerung für Analysezwecke nutzen zu können. Allerdings kennen die Feldgeräte selbst üblicherweise ihren Einsatzzweck nicht. Eine entsprechende Semantik liegt in den Feldgeräten also nicht vor. Dieser ist allerdings in der Prozesssteuerung teilweise oder vollständig bekannt. Durch eine geeignete Kombination aus Engineeringumgebung, PROFINET und OPC UA kann eine skalierbare und komfortable Lösung für IIoT ermöglicht werden, mit der Daten-basierte Services einfach auf die relevanten Informationen von Feldgeräten zugreifen, diese verstehen und nutzen können.

1 Motivation

1.1 Einleitung

Das industrielle Internet der Dinge (IIoT) bezieht sich auf vernetzte Sensoren, Instrumente und andere Feldgeräte, die mit den industriellen Anwendungen von Computern, einschließlich Fertigung, Energiemanagement oder allgemein Daten-basierten Services, vernetzt sind. Diese Konnektivität ermöglicht die Datenerfassung, den Austausch und die Analyse und erleichtert so Produktivitäts- und Effizienzsteigerungen sowie andere wirtschaftliche Vorteile [1]. Vorteile entstehen dabei z.B. durch vorausschauende Wartung, Anomalie-Erkennung, Optimierung von Fertigungsprozessen- und Fertigungsorganisation, Asset-Management, Verkürzung von Anlagenstillständen durch ursachengenaue Diagnose oder Energiemanagement. In all diesen Szenarien entsteht der Nutzen durch eine applikationsspezifische Analyse von Daten aus dem Produktionsprozess. Dies wird gemeinhin auch als Data Analytics bezeichnet.

Klassisch können diese Daten aus den Applikationsprogramen der Prozesssteuerung wie z.B. einer PLC gewonnen werden, da hier natürlicherweise die applikationsspezifische Funktion implementiert ist. Allerdings verfügt eine Prozesssteuerung allein nicht über alle Daten, die für die Nutzung der o.a. Mehrwerte erforderlich sind. Inzwischen verfügen auch die eingesetzten intelligenten Feldgeräte über Netzwerkfähigkeit sowie weitergehende Daten über sich selbst und den Fertigungsprozess, die erfasst und mit der Applikation in Kontext gesetzt werden müssen. In der Regel ist die Prozesssteuerung über einen echtzeitfähiges Ethernet System wie z.B. PROFINET mit den Feldgeräten verbunden [12]. PROFINET ermöglicht dabei heute schon die vertikale Kommunikation der Feldgeräte für IIoT Anwendungen diese Eigenschaft wird zukünftig durch Netzwerkkonvergenz auf Basis von Ethernet TSN [13] noch verbessert.

Für die direkte Bereitstellung von Daten aus Geräten ist in vielen IIoT Applikationen bereits MQTT [2] als weitverbreitetes und allgemein bekanntes Protokoll im Einsatz. Allerdings fehlen bei MQTT wichtige Festlegungen, die eine herstellerübergreifende und interoperable Nutzung sicherstellen. Weder die Nutzung der sog. Topics noch Format und Inhalt der Nutzdaten sind für MQTT definiert. Deshalb ist MQTT als Protokoll für eine applikationsübergreifende und interoperable Gewinnung von Daten nicht ausreichend und muss um weitere unabhängige Festlegungen ergänzt werden. Eine hierbei für IIoT weitgehend akzeptierte Technologie ist OPC UA, die in Kombination mit dem Publisher/Subscriber Modell (kurz Pub/Sub) [3] auch die Nutzung mit MQTT ermöglicht.

Aktuell gibt es viele unterschiedliche Aktivitäten in der für OPC UA verantwortlichen Organisation OPC Foundation zur Definition und Standarisierung von domänenspezifischen Informationsmodellen.

Viele dieser Modelle beschreiben naturgemäß eine Teilfunktion wie z.B. das Modell eines einzelnen Roboters oder Sensors/Aktors in der Fertigung. Allerdings müssen diese Modelle für Data Analytics in einer Fertigungszelle oder ganzen Fabrik mit z.B. vielen Robotern und anderen Geräten in einen applikationsspezifischen Kontext gesetzt werden. Darüber hinaus deckt OPC UA aktuell (noch) nicht die notwendige Echtzeitkommunikation zwischen Prozessteuerung und den Feldgeräten ab, so dass hier auch für einen längeren Zeitraum ein optimiertes und etabliertes System wie z.B. PROFINET zum Einsatz kommen wird, da die Einführung von neuen Industriellen Kommunikationslösungen lange Zeiträume erfordert [7] [14].

Inhalt dieses Dokuments ist es daher, ausgehend und aufbauend auf bestehenden Kommunikationstechnologien wie PROFINET, OPC UA und MQTT eine standardisierbare Lösung zu beschreiben, welche die potenziellen

Mehrwerte von IIoT heben kann. Dies wird auch anhand eines Anwendungsbeispiels aus der Automobilfertigung erklärt. Die Anwendung kann aber auf beliebige andere Prozesse übertragen werden.

1.2 Einführung eines Anwendungsbeispiels Klebeprozess

Abbildung 1 zeigt ein Anwendungsbeispiel, bei dem durch die Datenanalyse von Prozessdaten ein Klebeprozess optimiert werden soll. Bei dem Klebeprozess werden zwei oder mehrere Fahrzeugteile miteinander verbunden. Ein Roboter fährt dabei eine Klebevorrichtung an der Klebenaht entlang, während eine Klebesteuerung den vorgeheizten Kleber entsprechend appliziert. Die Steuerung der Klebesteuerung erfolgt über Profinet durch den Roboter. Die Steuerung des Roboters selbst durch eine überlagerte Steuerung (PLC). Die Identifikation des Fahrzeugs wird der Klebesteuerung ebenfalls über Profinet bekannt gemacht.

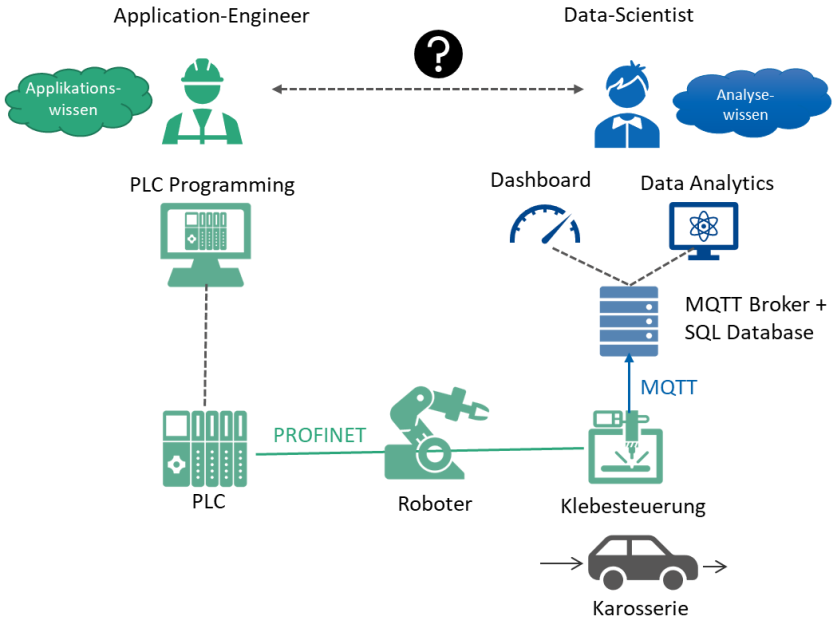

Abbildung 1: Anwendungsbeispiel aus der Automobilfertigung

In diesem Beispiel kann davon ausgegangen werden, dass die Klebesteuerung nicht nur die Identifikation des Fahrzeugs, sondern auch über umfangreiches Wissen der Klebenaht verfügt. Sie kennt z.B. die Temperatur des Klebers, misst den Druck über die Klebezeit und überwacht den Füllstand des Klebematerials. Darüber hinaus kennt sie eigene Typenschilddaten wie Hardware- und Firmwareversion. All diese Informationen sind für Data Analytics interessant, um beispielsweise die folgenden Fragen zu beantworten:

- Mit welchem Temperaur- und Druckverlauf wurden die Nähte eines speziellen Fahrzeugs oder einer Flotte gefertigt?
- Zu welcher Uhrzeit/Arbeitsschicht gibt es üblicherweise Probleme mit der Nachfüllung des Klebers?
- Wie ist die Abhängigkeit zwischen Klebeparametern und Temperaturbedingungen in der Halle im Vergleich zu anderen Fertigungsstandorten.
- Welche Firmwareversion haben alle in der Fabrik eingesetzten Klebesteuerungen?
- Es sind viele weitere Beispiele denkbar.

1.3 Die Rolle und Definition von Data Science

Bemerkenswert ist, dass zur Beantwortung dieser Fragen nicht allein das Automatisierungs- und Prozesswissen, sondern vor allem Methoden und Werkzeuge der Data-Science bekannt sein müssen. Daher werden diese Mehrwerte in der Regel erst durch die Einbeziehung von Experten mit entsprechender Kompetenz ermöglicht. Diese Experten können als Datenwissenschaftler (Data-Scientist) bezeichnet werden. Data Science kann wie folgt definiert werden:

Data Science ist ein interdisziplinäres Wissenschaftsfeld, welches wissenschaftlich fundierte Methoden, Prozesse, Algorithmen und Systeme zur Extraktion von Erkenntnissen, Mustern und Schlüssen sowohl aus strukturierten als auch unstrukturierten Daten ermöglicht [2] [3]

Aus dieser Beschreibung lässt sich erkennen, dass hier nicht der Prozess oder die Automatisierung, sondern die Daten und deren Verhalten im Mittelpunkt stehen. Ob es sich bei den zu analysierenden Daten um die eines Klebeprozesses handelt, kann in erster Näherung ignoriert werden, wenn z.B. über einen längeren Zeitraum die Daten von allen Klebesteuerungen auf Anomalien untersucht werden. Erst bei der Aufdeckung von auffälligen Verhaltensmustern ist ein Rückschluss auf den beteiligten Prozess wichtig und notwendig. Aus Sicht der Data Science steht also die eigentliche Automatisierungsapplikation nicht mehr an erster Stelle. Daher muss das IIoT Szenario aus Sicht der Datenanalyse anders betrachtet werden als aus Sicht der Automatisierungs- und Prozesstechnik.

Eine weitere Eigenschaft der Daten-zentrierten Sichtweise ist die Notwendigkeit einer herstellerübergreifenden Standardisierung von Dateninhalten. Bei der Analyse der Daten werden in der Regel spezielle Normierungen und Optimierungen vorgenommen. Diese hängen von jeweiligen Anwendungsfall und Analyseszenario ab. Im Vergleich zur Automatisierungstechnik ist damit die Notwendigkeit einer tiefen interoperablen Standardisierung der Daten nicht so hoch, aber dennoch hilfreich, wenn es sie gibt. Im o.a. Beispiel steht die Beantwortung der Analysefragen und weniger die Herstellerneutralität der Klebesteuerungen im Fokus.

Prozessgeräte für Data-Science Anwendungen wie z.B. die Klebesteuerung, müssen über notwendige Informationen verfügen, um diese zu ermöglichen. Beispielsweise sei hier genannt:

- Datum und Uhrzeit unter Berücksichtigung der jeweiligen Zeitzone und Genauigkeit
- Eindeutige und menschenlesbare Kennzeichnung des Prozessgerätes im jeweiligen Gültigkeitsbereich wie Maschine, Fertigungszelle, Fertigungslinie, Fertigungshalle, Standort oder Konzern
- Eindeutige Kennzeichnung des Nutzdatums innerhalb aller von dem Gerät bereitgestellten Daten
- Art einer Nutzdateninformation wie z.B. „Prozessinfo“ oder „Störmeldung“
- Typ und Datentyp der im Nutzdatum enthaltenden Informationen
- Versionsinformation des Prozessgerätes

Ferner ist es wichtig, dass die Speicherung der Daten über einen langen Zeitraum und unabhängig von Änderungen im Automatisierungsprozess erfolgt. Es kann a-priori keine Aussage und Entscheidung über Data-Science relevante Fragestellungen getroffen werden, da diese sich erst im Nachhinein ergeben können. Es bedarf also einer gewissen „Vorratsdatenspeicherung“.

Diese Unabhängigkeit vom Prozess ist in einem anderen dargestellten Data-Science Anwendungsfall nicht mehr unbedingt gegeben. In diesem Fall sollen Informationen aus dem Prozess in sog. Dashboards weitgehend „Live“ dargestellt werden. Für Dashboards werden in der Regel die gleichen Daten wie für die Analytics verwendet, allerdings ist hier ein besserer Prozessbezug notwendig.

In den folgenden Darstellungen wird die Farbe „Blau“ zur Kennzeichnung von Data-Science relevanten Inhalten verwendet. „Grün“ gekennzeichnet sind die Automatisierungs- und Prozesstechnisch relevanten Themen.

1.4 Automatisierungs- und Prozesstechnik

Die Disziplin der Automatisierungs- und Prozesstechnik (AT) ist verhältnismäßig alt. Sie wird seit langer Zeit weitgehend unverändert mit Werkzeugen wie PLCs, Motion-Controllern, IEC 61131, und Feldbussen implementiert. Auch innerhalb von Organisationen sind für die AT in der Regel andere Personen und Expertisen notwendig. Hier stehen die optimale Performance, Verfügbarkeit und Qualitätssicherung eines Fertigungsprozesses im Mittelpunkt.

Eine weitere Eigenschaft der Automatisierungstechnik ist, dass sie zeitlich und räumlich unabhängig von der Data-Science zur Anwendung kommen kann. So ist es z.B. ein häufiger Fall, dass erst Jahre nach der Inbetriebnahme einer Maschine oder Anlage zusätzliche Informationen durch Data-Science gewonnen und implementiert werden müssen. Außerdem ist zu dem Zeitpunkt nicht immer sichergestellt, dass noch das Automatisierungs-Know-How oder die entsprechenden Experten verfügbar sind.

Allerdings sind in den Werkzeugen der Automatisierungstechnik in der Regel viele prozessrelevante Informationen enthalten, die auch für Data-Science interessant sein können. In unserem Beispiel sind das z.B. die Steuerungsvariablen für „Start Fügen Links“ und „Stop Fügen Links“. Diesen Variablen sind im Profinet dann die entsprechenden Prozessdaten zugeordnet. Die Automatisierungstechnik kann also für Data-Science wesentliche Informationen über die Applikation bereitstellen.

Hier ist beispielsweise zu nennen:

- Eindeutiger und interpretierbarer Gerätename
- Quelle der Informationen innerhalb eines Gerätes wie z.B. der Steckplatz
- Metainformationen über den Prozess wie die o.a. Steuerungsvariablen

Außerdem erfolgt der Verbindungsaufbau und die Startparametrierung der Geräte im Hochlauf oder nach Gerätetausch üblicherweise durch die PLC/Prozesssteuerung. Nur so kann sichergestellt werden, dass die Automatisierungstechnik die maximale Verfügbarkeit erreicht. Es kann also davon ausgegangen werden, dass bei vorhandener und laufender Prozessteuerung diese Parameter im Gerät vorhanden und aktuell sind. Das ist ein großer Vorteil für die semantische Kopplung von Automatisierungstechnik und Data Analytics.

1.5 Semantische Unabhängigkeit?

Eine wesentliche Erkenntnis ist, dass die Paradigmen, Werkzeuge und Methoden der Data-Science und Automatisierungstechnik deutlich unterschiedlich und sogar weitgehend unabhängig voneinander sind. Allerdings ist das semantische Wissen über die Automatisierung oder den Prozess in der Data-Analytics sehr hilfreich, wenn nicht sogar notwendig. Dies soll wieder anhand des oben angeführten Beispiels erläutert werden.

Ein konstruiertes Beispiel wäre z.B. die Anomalie-Erkennung für das Nachfüllen des Klebers in allen Applikationen eines Fertigungsstandorts. Die Klebesteuerungen senden dazu den aktuellen Füllstand des Klebers mit jeder fertiggestellten Fügestelle in eine Datenbank. Mit Methoden der Datenanalyse können jetzt in dieser Datenbank der zeitliche Verlauf aller Füllstände ausgewertet und Abweichungen identifiziert werden. Hierzu schaut sich der Data Analyst in der Regel das dynamische Verhalten der Werte an, um dann mit weiteren statistischen Methoden Abweichungen vom Normalverhalten zu erkennen. Hierzu ist strenggenommen noch nicht einmal Applikations- oder Prozesswissen über Klebeprozesse notwendig. Dies wird erst wieder relevant, wenn die Anomalien und mögliche Optimierungen mit einem Prozessexperten diskutiert werden sollen. Dabei stellen sich dann konkrete Fragen wie:

- Welcher Klebeprozess hat die größte Abweichung?
- Gibt es Abhängigkeiten von Umweltbedingungen oder Wartungspersonal?
- Wann wurde das beteiligte Gerät zuletzt gewartet?
- …

Hierzu muss jetzt von dem Datenpunkt mit der Anomalie wieder auf das beteiligte Gerät zurückgeschlossen werden. Dazu bedarf es mindestens das Applikationswissen welche konkrete Klebesteuerung die, mit der Anomalie ist. Die Data Analytics allein ist also ohne die Zuordnung zum Prozess weitgehend wertlos.

Weitet man das Beispiel auf die Diagnose für Sensoren aus, die an digitale Eingangsbaugruppen angeschlossen sind, bedarf es für jeden Datenpunkt einer Applikationsinformation, da der Wert eines binären Eingangs keine Aussage über das konkret erfasste Prozesssignal enthält. In einer realen Anwendung können das schon einmal einige 100 bis 1000 Signale sein. Es bedarf also eines Datenaustauschs für Applikationsdaten zwischen der Automatisierungstechnik und der Data-Analytics.

1.6 Anforderungen

Wie Eingangs dargestellt, ist die Verfügbarkeit von Applikationsdaten für Data-Analytics wertvoll. Eine entsprechende Bereitstellung dieser Daten muss folgende Anforderungen erfüllen:

- Automatisch:
  Die Nutzung von Applikationsinformationen für Data-Analytics darf nicht zur mehrfachen und schlimmstenfalls fehlerhaften Eingabe von Daten führen.

- Konsistent:
  Es kann vorkommen, dass der Automatisierungsprozess geändert oder erweitert wird. In diesem Fall muss die Data Analytics immer auf konsistenten Applikationsdaten basieren.
- Zeitlich entkoppelt:
  Neue Data-Analytics Abfragen müssen zeitlich unabhängig von der Automatisierungsapplikation erstellt werden können. So stellt sich die Frage nach speziellen Anomalien z.T. erst Jahre nach der Inbetriebnahme einer Anlage.
- Organisatorisch entkoppelt:
  Oft fällt Applikations-Expertise und Data-Analytics in einem Unternehmen nicht zusammen. Speziell im Data-Science Umfeld gibt es viele IT-Dienstleister, die kein Prozesswissen haben und auch nicht haben müssen. Daher müssen die Schnittstellen zur Data Analytics weitgehend entkoppelt sein.
- Herstellerunabhängig:
  Die Steuerungs- und Programmiersysteme unterschiedlicher Hersteller müssen ihre Applikationsinformationen übergreifend bereitstellen, da es viele Geräte und Steuerungshersteller gibt und die Data Analytics Lösungen unabhängig von speziellen Herstellern sein sollte. Dies ermöglicht völlig neue Geschäftsmodelle für Data Analytics Unternehmen.

Es deutet sich an, dass ein konsistenter und interoperabler Austausch von Applikationsdaten zwischen Automatisierungs- und Data-Analytics ein wesentlicher Erfolgsfaktor für eine optimale Lösung ist.

2 Lösungskonzept

2.1 Einleitung

In den folgenden Kapiteln wird ein Konzept beschrieben, welches auf der Basis der vorhandenen semantischen Informationen der Prozessteuerung in Kombination mit Profinet und MQTT basierend auf OPC UA Pub/Sub eine optimale Data Analytics Lösung für IIoT bereitstellen kann.

2.2 Anwendungsbeispiel

In Abbildung 2 wird das gesamte Lösungskonzept anhand des o.a. Anwendungsbeispiels erläutert.

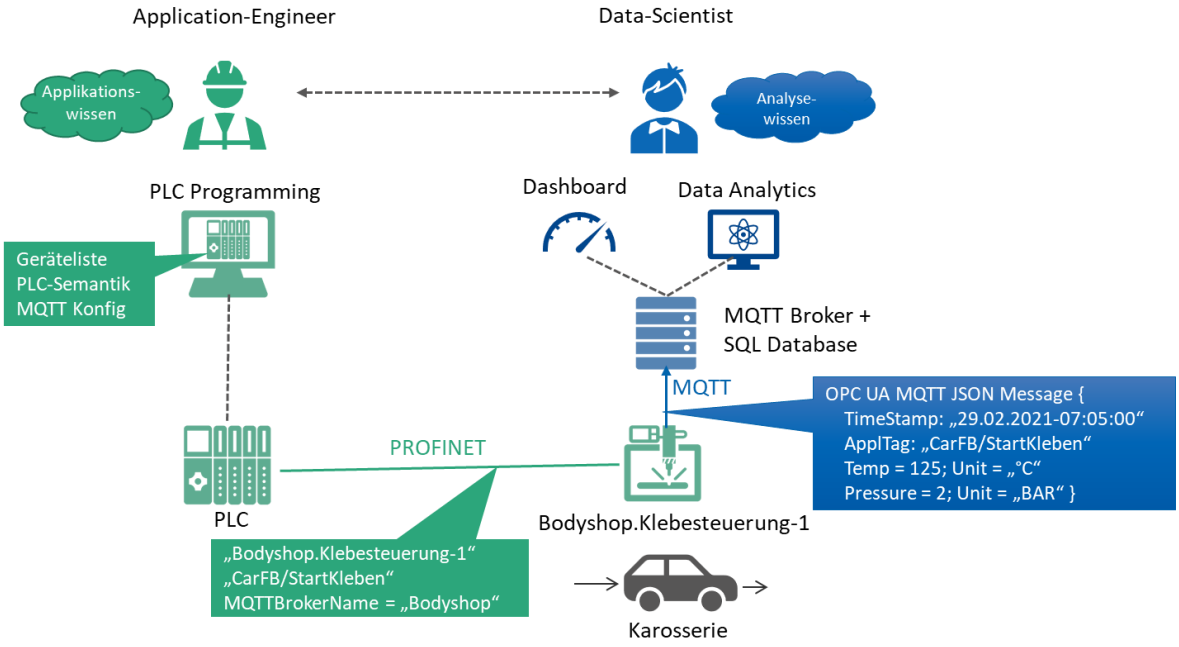

Abbildung 2: Lösungskonzept

Die Idee ist, die in der PLC vorhandene Applikationssemantik auch für Data Analytics einfach nutzbar zu machen. Dies gilt für die Zuordnung zwischen Steuerungsprogramm und Peripherie aber auch für Geräteinformationen wie z.B. den Profinet Gerätenamen und den Stationsaufbau.

Ferner macht man sich Eigenschaften des Profinet Systems zu Nutze, um im Programmiersystem der Steuerung z.B. die MQTT Einstellungen in den Devices zu konfigurieren. Hier sorgt Profinet dann dafür, dass alle Einstellungen incl. Applikationssemantik eingestellt und auch nach einem Gerätetausch wiederhergestellt werden.

Dies wird anhand des Beispiels aus Abbildung 2 noch einmal exemplarisch erläutert:

1. Der Application-Engineer erstellt das Steuerungsprogramm und konfiguriert das Profinet-System. Dabei vergibt er die notwendigen Gerätenamen und ggf. IP-Adressen der Geräte. Aber auch die Zuordnung zwischen Steuerungsvariablen und IO-Punkten wird in der Konfigurationsphase des Steuerungsprogramms vorgenommen. Ferner ist es denkbar, dass im Engineering System der Steuerung auch der Name und weitere Parameter des MQTT Brokers eingestellt werden. Dies bietet den Vorteil, dass nicht bei jedem Gerät einzeln unabhängig noch eine Konfiguration der Brokeradresse notwendig ist.Das Steuerungsprogramm, die Geräteliste und die Zuordnungsliste werden auf die Steuerung geladen.
2. Die Steuerung baut eine Verbindung zu den Feldgeräten auf. Hierbei wird der Profinet Gerätename verwendet. Aber auch die MQTT Brokerparameter und PLC Variablennamen können als „Application Tag“ mit zu den Profinet Devices übertragen werden. Das verwendete MQTT Topic kann ebenfalls im Programmiersystem der Steuerung konfiguriert werden.
3. Das Feldgerät baut eine MQTT Verbindung zum Broker auf. Wenn im Profinet Hochlauf Name und MQTT Parameter gesendet wurden, werden diese beim Verbindungsaufbau verwendet. Grundsätzlich ist aber die Verbindung zum Broker unabhängig von der Profinet Verbindung.
4. Das Feldgerät steuert den Klebeprozess und zeichnet dabei die relevanten Parameter dieses Prozesses auf.
5. Die verwendeten Prozessparameter werden via MQTT an den Broker versendet. Zusätzlich dazu wird das MQTT Paket mit Informationen zum Gerät und der Steuerungsvariablen sowie einem Zeitstempel versehen.
6. Der MQTT Broker nimmt das Paket entgegen und speichert es in einer Datenbank ab. Hierbei ist es hilfreich, wenn allgemeine Informationen wie der Zeitstempel, der Gerätehersteller, der Gerätename und das Application-Tag in eigenen Tabellen oder Spalten der Datenbank abgespeichert werden können. Das vereinfacht die spätere Auswertung.
7. Der Data Scientist soll beispielsweise eine Anomalieerkennung über alle Klebeprozesse eines Standorts umsetzen. Hierzu muss er aus der Datenbank beispielsweise über die Herstellerkennzeichnung von Klebesteuerungen alle relevanten Einträge lesen. Danach wird der Datenbestand mit Data-Science Methoden auf Anomalien untersucht. Über Gerätename und Application-Tags kann im Falle einer Abweichung einfach auf das betroffene Gerät geschlossen und ggf. weitere spezifische Analysen implementiert werden.

Aus dem dargestellten Beispiel wird deutlich, welcher Mehrwert in der Verwendung der Applikationssemantik aus der Steuerung für Data-Analytics Zwecke entsteht. Hierbei ist insbesondere darauf hinzuweisen, dass hierfür kein zusätzlicher Engineeringaufwand entsteht. Ferner stellt die Integration mit dem Steuerungsprogramm eine Konsistenz auch bei zukünftigem Anlangen- oder Maschinenerweiterungen sicher.

All dies ist aber nur hilfreich, wenn es im Rahmen von übergreifenden Standardisierungsaktivitäten betrachtet wird. Diese werden im nächsten Kapitel behandelt.

3 Standardisierung und Überblick zu laufenden Standardisierungsprozessen

3.1 Allgemein

Im Kapitel 1 wurde schon auf die Besonderheiten der Automatisierung- und Data-Science Umgebung eingegangen. In der Data-Science haben die Gewinnung und Auswertung von Daten absolute Priorität. Durch

die flexiblen Werkzeuge und Methoden ist es normal, sich an unterschiedliche Rahmenbedingungen anpassen zu müssen und zu können. Daher ist die Notwendigkeit einer übergreifenden Standardisierung der Analysedaten im Vergleich zur Automatisierung nicht so groß. In der Automatisierungs- und Prozesstechnik gibt es z.B. die Anforderung der herstellerübergreifenden Austauschbarkeit von Geräten. Im Vergleich dazu ist nach einem Geräteaustausch die gleiche und kompatible Bereitstellung von Daten nicht so relevant.

Ferner gibt es Applikationen, wie die oben beschriebene spezielle Klebesteuerung, für die es z.B. Mangels Interesse keine Aktivitäten der Standardisierung geben wird. Dennoch müssen die Daten dieser Applikation gewonnen und ausgewertet werden können.

Wesentlich entscheidender ist, dass gemeinsame herstellerübergreifende Daten einheitlich bereitgestellt werden. Ein gutes Beispiel sind hier beispielsweise Zeitstempel. Unterschiedlich aufgelöste und formatierte Zeitstempel können in einer Data-Analytics Anwendung nicht ausgewertet werden, da ein zeitlicher Zusammenhang zur Nachverfolgung essenziell ist. Gleiches gilt für die Namen und Identifikation der Geräte.

Bei der übergreifenden Festlegung für diese Daten kann die Standardisierung durch Profinet und OPC UA einen signifikanten Beitrag leisten. Dies wird im Folgenden erläutert.

3.2 OPC UA Pub/Sub mit MQTT

Für das dargestellte Beispiel bedarf es noch einer herstellerunabhängigen Festlegung der Nutzung von MQTT. Der MQTT Standard in der verbreiteten Version 3.1.1 [4] trifft in Bezug auf Nutzung von Topics oder der zum Broker transportieren Nutzdaten keine Festlegungen. Diese fehlende Standardisierung hat wesentlich zur Verbreitung von MQTT z.B. in Home-Appliances oder herstellerspezifischen Lösungen beigetragen. Allerdings schließt sich dadurch die Anwendung in einem interoperablen Multi-Vendor System weitgehend aus.

Für die Verwendung in Kombination mit Profinet ist also eine übergreifende Standardisierung, insbesondere der transportierten Nutzdaten, notwendig. In diesem Bereich gibt es aus der IT-Welt bekannte Optionen wie z.B. Googles ProtoBuf [5] Technologie. Allerdings bietet auch OPC UA in Kombination mit dem brokerbasierten Publisher/Subscriber Modell [6] die Möglichkeit der MQTT Nutzung. Da es sich bei OPC UA um einen akzeptierten Standard in der OT handelt, scheint die Nutzung für die Kombination mit Profinet zielführender zu sein.

In der öffentlichen Diskussion und Wahrnehmung wird OPC UA vor allem mit flexiblen Informationsmodellen und dem Zugriff von Clients auf Server in Verbindung gebracht. Eher unbekannt ist die Möglichkeit durch das sog. Publisher/Subscriber Modell [6] in standardisierter Form Informationen an Message-Broker zu versenden. Als Protokolle werden durch den Standard heute schon AMQP und MQTT unterstützt. Wichtig ist hierbei, dass für eine Anwendung von Pub/Sub über MQTT kein OPC UA Server mit dem dazu gehörenden Overhead zwingend ist. In der Praxis reicht also die Verwendung des notwendigen Encodings/Decodings auf der Leitung aus. Dies verbessert die Skalierbarkeit der Gesamtlösung.

Insgesamt bietet die Anwendung von OPC UA Pub/Sub mit MQTT die folgenden Vorteile:

- Akzeptierter Standard mit zunehmender Unterstützung durch viele IoT Anbieter
- Standardisiertes JSON und Binary Encoding
- Standardisierte Datentypen
- Standardisierte Zeitstempel
- Option zur Trennung von Daten und Semantik
- Herstellerübergreifende und herstellerspezifische Informationsmodelle

OPC UA mit Pub/Sub bietet sich also als Standard für die MQTT Nutzung an. Es sei auch noch darauf hingewiesen, dass Pub/Sub auch ohne Profinet genutzt werden kann, da z.B. nicht jedes Feldgerät über ein Profinet Device verfügt. Die Bereitstellung von Applikationsinformationen oder Konfigurationsdaten über Profinet kann daher als eine Option verstanden werden.

3.3 OPC UA Pub/Sub-MQTT für Profinet

Eine optimale Kombination von Profinet und OPC UA im Sinne des beschriebenen Lösungskonzepts kann nur erreicht werden, wenn die beteiligen Nutzerorganisationen OPC Foundation und PROFIBUS/PROFINET International einbezogen sind. Hierzu bietet die OPC Foundation das Model der Companion-Spezifikationen

an. Für Profinet selbst gibt es schon seit 2017 eine sog. Joint Working Group bestehend aus Experten für Profinet und OPC UA, deren Arbeit bereits zwei Companion Spezifikationen hervorgebracht hat:

- OPC UA for PROFINET:
  Abbildung der allgemeinen Profinet Funktionalitäten auf ein OPC UA Informationsmodell [8]
- OPC UA for Energy Management:
  Abbildung der PROFIenergy Eigenschaften auf ein OPC UA Informationsmodell [9].

In dieser Joint WG werden aktuell die beschriebenen Konzepte zur Kombination von Profinet und OPC UA Pub/Sub bewertet und diskutiert.

Aktuell befindet sich die WG in der Phase der Anforderungsanalyse und Diskussion. Anforderungen beziehen sich auf:

- Broker Konfiguration wie z.B. DNS-Name oder IP-Adresse
- Client Identifikation
- MQTT Connection Management
- Zeitstempel und Synchronisation
- Automatisch generierte MQTT Topics und deren Konfiguration
- Standardisierte Message-Typen z.B.
  - Applikationsdiagnose wie z.B. für die oben dargestellte Klebapplikation
  - Profinet Diagnose. Drahtbruch/Kurzschluss/Überlast/..
  - Primary/Backup Umschaltungen bei redundanten Steuerungen
  - Safety-Diagnose für PROFIsafe
  - Energiediagnose und Status für PROFIenergy
  - Antriebsdiagnose für PROFIdrive
  - Versionsinformationen beim Hochlauf für Asset Monitoring

Auf Basis der diskutieren Anforderungen wird im nächsten Schritt eine Abbildung auf die OPC UA Pub/Sub Technologie und deren Standards vorgenommen. Ferner wird identifiziert, welche Erweiterungen im Profinet Standard notwendig sind.

Nach der Diskussion der Anforderungen und Lösungskonzepte muss eine Untersetzung in die Profinet und OPC UA for Profinet Companion-Spezifikation erfolgen. Ziel ist es diese Arbeiten für den nächsten Profinet Maintenance Zyklus bis Ende 2022 abgeschlossen zu haben.

4 Zusammenfassung und Ausblick

Die beschriebene Kombination aus Profinet und OPC UA stellt eine gute Grundlage für IIoT Applikationen dar. Die weite Verbreitung von Profinet in der Automatisierungs- und Prozesstechnik sowie der hohe Reifegrad des gesamten „Ökosystems“ kann in Kombination mit OPC UA zur Unterstützung von Data-Science optimal genutzt werden.

Hierbei ist es nicht notwendig, aber dennoch hilfreich, wenn auch bei der Automatisierungstechnik im Sinne von weitergehenden Standardisierungen z.B. OPC UA FX [UAFX] zur Anwendung kommt. Auch die gleichzeitige Unterstützung eines OPC UA Servers im Feldgerät ist denkbar und nutzbringend.

Im Kern der Lösung steht jedoch auch die (wieder) Verwendung von Applikationssemantik aus der Automatisierungstechnik für Data-Science Disziplinen. Dies vereinfacht die weitgehend unabhängige Implementierung von Analyseanwendungen, ohne zusätzliche Aufwände zur konsistenten Bereitstellung der notwendigen semantischen Informationen.

Durch die Nutzung von OPC UA Pub/Sub in Kombination mit MQTT steht für die Verbreitung eine akzeptierte und geeignete Technologie zur Verfügung.

Beide Technologien können durch die Kombination zukünftig einen wesentlichen Beitrag für neue Anwendungen und Geschäftsmodellen im Zuge von Industrie 4.0 leisten.

5 Literaturverzeichnis

[1] Kagermann, H.; Lukas, W.-D.; Wahlster, W.: Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution. In: VDI-Nachrichten, 2011.

[2] Vasant Dhar: Data Science and Prediction. In: Communications of the ACM, Dezember 2013.
[3] The key word in "Data Science" is not Data. In: it is Science - Simply Statistics.
[4] MQTT Version 3.1.1, Online: Online: https://www.oasis-open.org, 2021.
[5] Protocol Buffers, Google Developers. Online, 2021
[6] OPC Unified Architecture - Part 14:6, Online: https://reference.opcfoundation.org/v104/Core/docs/Part14/, 2021.
[7] Schriegel, Sebastian; Jasperneite, Juergen: A Migration Strategy for Profinet Toward Ethernet TSN-Based Field-Level Communication: An Approach to Accelerate the Adoption of Converged IT/OT Communication. In: IEEE Industrial Electronics Magazine, DOI: 10.1109/MIE.2020.3048925, 2021.
[8] OPC UA for PROFINET Companion Specification. Online: https://www.profibus.com, 2021
[9] OPC UA for Energy Management Companion Specification. Online: https://www.profibus.com, 2021
[10] OPC UA for Factory Automation - OPC Foundation
[11] Sebastian Schriegel: Kompatibilitätsverfahren für Ethernet Time Sensitiv Networks und Profinet-Hardware (am 02.09.2021 angenommene Dissertation), Bielefeld, 2021.
[12] IEC 61158 Version: 2.4MU2, Order No.: 2.712, PROFINET Specification. Online: https://www.profibus.com/download/profinet-specification, 2021.
[13] Biendarra, Alexander; Gamper, Sergej; Friesen, Andrej; Schriegel, Sebastian: Guideline PROFINET over TSN Version 1.3, Profibus International, 2021.
[14] Sebastian Schriegel, Jürgen Jasperneite: Migrationskonzept zur Einführung von Ethernet TSN in die Feldebene. In: at – Automatisierungstechnik, De Gruyter, 2021.

http://dx.doi.org/10.25673/39550

Distributed Asset Management in Industrie 4.0

Nico Braunisch, Frank Hilbert, Martin Wollschlaeger
Institut für Angewandte Informatik - TU Dresden
Professur für Prozesskommunikation

Nöthnitzer Str. 46
01187 Dresden
nico.braunisch@tu-dresden.de
frank.hilbert@tu-dresden.de
martin.wollschlaeger@tu-dresden.de

Abstract: Damit beliebige Assets in Industrie 4.0-Systemen verwendet werden können, müssen deren Informationen, Merkmale und Verhalten digital abgebildet werden. Dies wird über das Konzept der Verwaltungsschale realisiert. In der Umsetzung dieses Konzeptes stößt man jedoch auf mehrere Herausforderungen. Die Verwaltungsschale bildet dabei die verschiedenen Aspekte der Assets in Teilmodellen ab. Die verschiedenen fachlichen Zuständigkeiten der Teilmodelle führt einerseits zu sehr unterschiedlichen Eigenschaften im Hinblick auf die Dynamik dieser Teilmodelle, andererseits können Leistungs- und Verbindungseigenschaften von I4.0 Assets unterschiedlichen Beschränkungen unterliegen. So können die Assets beispielsweise bezüglich der Leistungsfähigkeit und somit der möglichen Operationen als auch hinsichtlich ihrer Erreichbarkeit im Netzwerk beschränkt sein. Daher bietet es sich an, die Verwaltungsschale und deren Services auf unterschiedlichen Systemebenen zu verteilen. Auf diese Weise lassen sich die verschiedenen Anwendungsszenarien eines Assets gezielter adressieren. Dynamische Aspekte können so in Teilmodellen in direkter Nähe des Assets realisiert werden, während aufwendigere Operationen an rechentechnisch mächtigere Systemebenen, wie zum Beispiel die Cloud, ausgelagert werden können. Ebenso können relativ statische Informationen von der Erreichbarkeit des Assets entkoppelt werden, indem diese Informationen von einer zuverlässig erreichbaren Ebene, zum Beispiel einem Gateway, verwaltet werden. In dieser Arbeit wird ein verteilter Ansatz zur Bereitstellung und Ausführung von (pro-)aktiven Verwaltungsschalen für I4.0 Komponenten vorgestellt.

1 Einleitung

Das Management der Assets in Industrie 4.0 (I4.0) wird über die Modellierung der Verwaltungsschalen (engl. Assets Administration Shell, kurz AAS) realisiert [1]. Durch diese wird in Verbindung mit dem Asset die jeweilige I4.0 Komponente gebildet. Betrachtet man zukünftige I4.0 Industrieanlagen als verteilte Systeme, wie in Abbildung 1 dargestellt, im Sinne des Industrial Internet of Things (IIoT) so müssen die Verwaltungsschalen ebenfalls eine aktive Rolle in der Kommunikation spielen. Daher sind sie als (pro-)aktive Verwaltungsschalen auszuführen [2]. In dieser Arbeit wird eine adaptive Laufzeitumgebung für (pro-)aktive Verwaltungsschalen mit variablen Schnittstellen für unterschiedliche Kommunikationswesen vorgestellt. Die Laufzeitumgebung ist dabei plattformübergreifend und modular entsprechend der Service orientierten Architektur von I4.0 umgesetzt [2]. Um die Laufzeitumgebung möglichst flexibel und ressourcenschonend zu gestalten wird auf dezentrale (Micro-)Services auf der Basis von .NET zurückgegriffen.

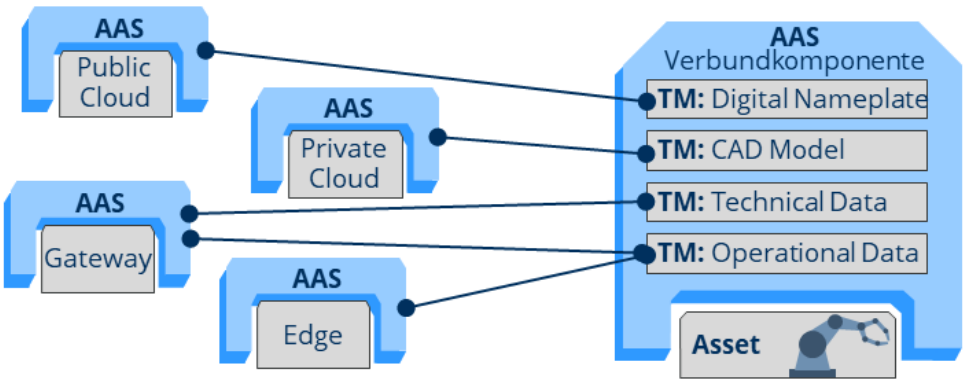

Abbildung 1: Distributed Asset

2 Problemstellung

Die verschiedenen Assets in I4.0 Systemen verfügen allerdings im Allgemeinen über stark variierende Leistungs- und Verbindungseigenschaften. Um die Verwaltungsschalen dennoch zuverlässig erreichbar bereitzustellen, muss ebenfalls die Laufzeitumgebung der Verwaltungsschalen dezentral gestaltet werden und somit ebenfalls, wie in Abbildung 1 dargestellt, die Verwaltungsschalen an unterschiedlichen Orten oder Ebenen im I4.0 System zur Verfügung stehen [3]. Eines der bestimmenden Kriterien für eine (pro-)aktive Verwaltungsschale für I4.0 Komponenten ist die Leistungsfähigkeit der eingesetzten Hardware und Software. Verfügt ein Asset nicht über hinreichende

Rechenleistung für eine (pro-)aktive Verwaltungsschale, so muss die Verwaltungsschale an einer Stelle mit mehr Leistungsfähigkeit ausgeführt werden. In diesem Fall bieten sich Hubs oder Gateways, wie in Abbildung 2 dargestellt, an.
Ein weiteres stark einschränkendes Kriterium ist die verfügbare Konnektivität. Nicht jedes Asset verfügt über eine große Bandbreite und die bestmögliche Übertragungsart. Zudem kann die Verfügbarkeit der Assets im IIoT zeitweise stark eingeschränkt sein. So muss out-of-band Kommunikation und das Handover von Verbindungen berücksichtigt werden. Beide zuvor genannten Einschränkungen können auch durch eine IIoT typische Limitierung im Energieverbrauch zustande kommen, zum Beispiel bei Verwendung batteriebetriebener Assets. In diesem Falle werden Bandbreite und Verfügbarkeit im Energiesparbetrieb genauso gedrosselt wie die Rechenleistung des Assets.

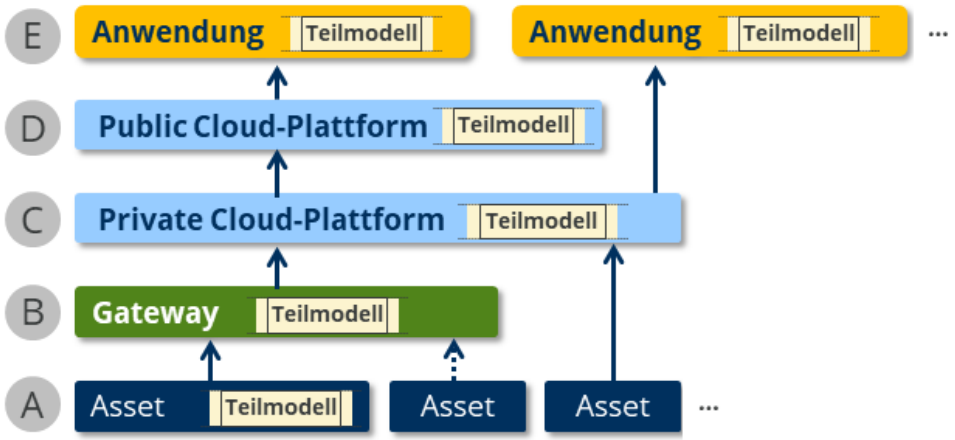

Abbildung 2: Verteilte Verwaltungsschale

Aus den zuvor genannten Gründen müssen Assets je nach gegebener Situation selbsttätig oder durch andere, übergeordnete Komponenten verwaltet werden. Zusätzlich erzeugen die verschiedenen Aspekte der Assets in den Teilmodellen sehr unterschiedliches Verhalten im Hinblick auf die Dynamik und der Kommunikationsparadigmen. So müssen dynamische Aktualwerte kontinuierlich und mit minimaler Latenz abgerufen werden können, wohingegen zum Beispiel Operations und Events in unregelmäßigen Abständen und Informationen zur Identifizierung, Versionsnummer o der Produktdatenblätter relativ selten übertragen werden.
Daher dient eine Verwaltungsschale nicht nur als statische Referenz und Datenquelle für die I4.0 Merkmale. Sie muss darüber hinaus den Zugriff auf Aktualdaten und sich ändernde Relationen erlauben. Dies führt zu verschiedenen Abhängigkeiten der Relationen und Referenzen in den Teilmodellen der Verwaltungsschale. Diese werden bereits in der Modellierung der Schale mitgestaltet. Durch die Verteilung der Verwaltungsschale in Form einer Verbundkomponente (siehe Abbildung 1) und die Verwendung der Bill-Of-Material werden die Relationen und Referenzen der Assets in einem I4.0

System abgebildet [4]. Die Laufzeitumgebung muss diese wiederum auflösen können. Durch den dezentralen Charakter dieser verteilten Verwaltungsschale ergibt sich auch eine Vielfalt in den Schnittstellen der Laufzeitumgebungen. Der Zugriff auf die Verwaltungsschale wird auf unterschiedlichen Systemebenen bereitgestellt Dies wird in Abbildung 2 schematisch dargestellt. Viele der dort benötigten sowie angebotenen Kommunikationsendpunkte genügen je nach Systemebene unterschiedlichen Charakteristika. So können Aktualwerte direkt am Gerät abgerufen werden, während relativ statische Informationen wie Versionsnummern oder Produktdatenblätter hingegen als Datenbank in der Cloud bereitgestellt werden können. Die entstehende Multi-Interface-Problematik muss durch die Laufzeitumgebung aufgelöst werden. Bei der Verteilung kommen die Aspekte der I4.0-Verbundkomponente zum Tragen. Hier sind spezielle Referenzen und Relationen notwendig, um die betreffenden Teilmodelle und deren Merkmale für die unterschiedlich dynamischen Interaktionen abzubilden.
Zur Erfüllung der angeführten Anforderungen stellt der Beitrag das Konzept, Entwurf und Umsetzung einer Laufzeitumgebung für eine verteilte Verwaltungsschale vor. Diese Laufzeitumgebung ist durch eigenständige Kommunikation und kooperative Interaktion in der Lage den Kriterien einer (pro-)aktiven und dezentralen Verwaltungsschale zu genügen. Mit dieser Herangehensweise lassen sich für die verschiedenen Anwendungsszenarien eines I4.0 Systems die Dynamiken der Teilmodelle gezielt adressieren.

3 Anforderungen an Management in Industrie 4.0

In Device Management in Industrial IoT wurden bereits die verschiedenen Anforderungen im Hinblick auf Industrial IoT betrachtet [5]. Für das Asset Management in verteilten System der Industrie 4.0 treffen diese Kriterien ebenfalls zu. Diese Erstrecken sich nach wie vor über die funktionalen Aspekte wie zum Beispiel Überwachung, Diagnose, Wartung und Fehlerbehandlung bis hin zu organisatorischen Aspekten wie zum Beispiel Bereitstellung, Security, Membership und Accounting. Für die Laufzeitumgebung der Verwaltungsschale spiegeln sich besonders für die funktionalen Aspekte die Anforderungen in den Charakteristiken der Schnittstellen wieder. Dabei wird, wie in Abbildung 3 dargestellt, zwischen unären Aufrufen und Streaming unterschieden. Unäre Aufrufe sind besonders im Bereich von Alarmen und Monitoring interessant. Das Streaming Konzept adressiert den kontinuierlichen Abruf von dynamischen Aktualwerten mit minimaler Latenz. Die Aufrufe können wiederum synchron, für feste Funktionsabfolgen wie zum Beispiel im Fall der Logging Aktivierung oder asynchron, zum Beispiel für Updates, erfolgen. In beiden Fällen wird jedoch der Kommunikationsfluss v on d er a ufrufenden V erwaltungsschale (AAS1) g esteuert. E in wesentlich flexiblere A rt d er I nteraktion s tellt d ie K ommunikation ü ber S treams d ar. Hierbei wird zwar von der aufrufenden Verwaltungsschale (AAS1) weiterhin der Kommunikationskanal bereitgestellt, jedoch erfolgt das Senden bzw. Empfangen von Nachrichten beider Kommunikationspartner (AAS1 und AAS2) unabhängig vom der aufrufenden Verwaltungsschale (AAS1). Dies ist besonders im Bereich von Alarmen und Monitoring interessant. Darüber hinaus ist Streaming von Relevanz wenn dynamische Aktualwerte kontinuierlich und mit minimaler Latenz abgerufen werden.

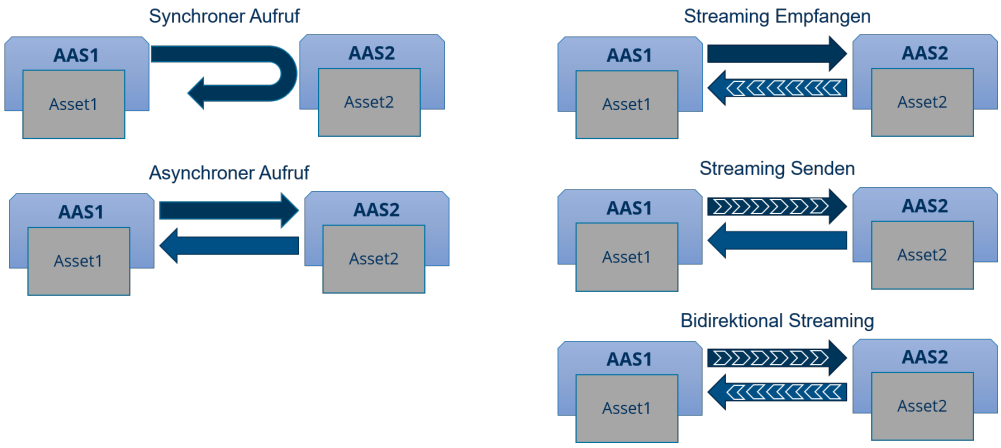

Abbildung 3: Anforderung an API der Laufzeitumgebung

4 Design Distributed Asset Management

Die Verwaltung der Assets im I4.0 System wird über ihre Verwaltungsschale organisiert. Diese stellt die benötigten Dienste im Sinne einer Serviceorientierten Architektur (SOA) bereit. Um die einzelnen Dienste der Verwaltungsschale aktiv nutzen zu können wird eine geeignete Laufzeitumgebung benötigt. Die Umsetzung dieser Dienste stellt jedoch für den Entwickler einen erheblichen Aufwand dar. Um den Mehraufwand bei der Serviceimplementierung möglichst gering zu halten, beruht die hier vorgestellte Umsetzung auf den nun vorgestellten Designprinzipien.
In der SOA bieten die einzelnen Assets ihre Fähigkeiten über Services an und nutzen zur Erfüllung ihrer Aufgaben die Services anderer Assets. Dabei soll die Zuständigkeit beziehungsweise der Anwendungsbereich eines Dienstes möglichst minimal sein. Im vorgestellten Design wird dabei die Umsetzung mittels Microservices eingeführt. Dies ermöglicht es die einzelnen Services unabhängig voneinander zu entwickeln und Abhängigkeiten zur vermeiden. Durch die Modellierung der Eigenschaften der Assets in der Verwaltungsschale werden dabei die Schnittstellen der einzelnen Services klar definiert. Komplexere Eigenschaften und Funktionen können dabei durch eine geeignete Servicekomposition aus anderen Services zusammensetzt werden.
Die Umsetzung der (Micro-)Services ist jedoch weiter stark implementierungsabhängig. Dies bezieht sich sowohl auf die Wahl der Entwicklungsumgebung, Programmiersprache und Ausführungsumgebung. Für die Laufzeitumgebung der Verwaltungsschale ergeben sich dabei noch Abhängigkeiten der jeweiligen Betriebssysteme sowie deren Laufzeitumgebung und Bibliotheken der verwendeten Programmiersprachen. Aus diesem Grund soll die Laufzeitumgebung der Verwaltungsschale plattformübergreifend gestaltet werden. In unserer vorgestellten Laufzeitumgebung wird daher das .NET Fra-

mework als plattform- und programmiersprachenübergreifende Umsetzung verwendet. Weiter können die (Micro-)Services in einer extra Virtualisierungsumgebung wie zum Beispiel Docker oder Kubernetes betrieben werden.
Um den Aufwand für den Entwickler der (Micro-)Services zu minimieren, soll bei der Umsetzung der Service der Großteil des Arbeitsaufwandes durch die automatisierte Generierung von Quellcode beziehungsweise Codegerüsten erfolgen. In der vorgestellten Lösung wird aus diesem Grund eine Interfacebeschreibungssprache (IDL) verwendet, um die Definition der Services, ihrer Endpunkte sowie deren Input- und Out-Parameter zu definierten.
Anschließend können für die einzelnen Services gezielt, entsprechend der gewünschten beziehungsweise benötigten Programmiersprachen, Frameworks und Ausführungsumgebungen, die entsprechenden Servicestubs generiert werden. Diese liefern unabhängig von der Umsetzung jedoch immer die jeweiligen Nachrichtenkanäle mit aufrufbaren Serviceendpunkten, definierten Nachrichten, sowie die dafür benötigten Kommunikationsdienste.

5 Multi API

Um den verschiedenen Charakteristika der unterschiedlichen Systemebenen Rechenschaft zu tragen, müssen die einzelnen (Micro-)Services über unterschiedliche Schnittstellen verfügen. Diese werden, wie in Abbildung 4 schematisch dargestellt, in der jeweiligen Laufzeitumgebung (hellgrüner Kreis) bereitgestellt.

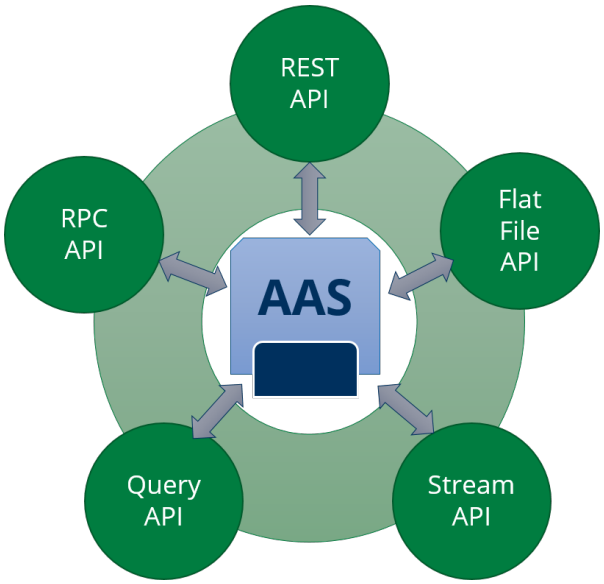

Abbildung 4: Unterschiedliche APIs der Laufzeitumgebung

Wie in Verwaltungsschale im Detail Teil 2 ausgeführt, muss die Laufzeitumgebung eine *REST API* für die Kommunikation mit der Verwaltungsschale bereitstellen [6]. Diese wird vorwiegend für die Kommunikation mit der I4.0 Infrastruktur wie Registry, Directory, Repositories beziehungsweise der jeweiligen Laufzeitumgebungen und Engines der Teilmodelle untereinander verwendet.
Für die Schnittstelle in die passive Verwaltungsschale muss die Laufzeitumgebung eine *Flat File API* bereitstellen. Diese wird verwendet um komplette AASX Pakete [1] oder Modellbestandteile auszutauschen. Der Zugriff auf die Modellbestandteile erfolgte dabei in serializierter Form als JSON oder XML Repräsentation. Weiter werden über die Flat File API die Zugriffe auf Anhänge (Supplementary Files) in der Verwaltungsschale wie Handbücher in PDF Dateien, Engineeringdokumente wie EPLAN oder Bilddateien ermöglicht. Diese API kann ebenfalls als Schnittstelle für assetsspezifische Konfigurationsdateien, wie ISO Images oder INI-Files, genutzt werden.
Für die Abfrage der Eigenschaften und Fähigkeiten der Assets wird in der Laufzeitumgebung eine *Query API* benötigt [7]. Diese kann über eine geeignete Abfragesprache (Query Language) die bestimmten Charakteristika der Assets erfassen und wiedergeben. Für das Auffinden der Assets im I4.0 System anhand eines Directory bzw. Repository wird diese API ebenfalls verwendet.
Für den Zugriff auf die funktionalen Eigenschaften der Assets werden in der Laufzeitumgebung je nach Charakteristik des Zugriffs zwei unterschiedliche APIs benötigt. Für den Aufruf von Funktionen, abgebildet in der AAS durch Operations, wird eine *RPC API* verwendet. Dies ermöglicht es, die Zugriffe auf die Funktionen der Assets auch remote auszuführen. Dabei können die Funktionen sowohl spezifische Eigenschaften am Asset ändern als auch komplexere Operationen ausführen, welche z.B. für die Berechnung oder Akkumulierung von Aktualwerten vor der Übermittlung genutzt werden.
Sollen Werte kontinuierlich erfasst und übermittelt werden, wird eine *Streaming API* verwendet. Für diese wird zwischen dem sendenden und dem empfangenden Asset ein Nachrichtenkanal aufgebaut. Die Übermittlung der Nachrichten über diesen Kanal erfolgt dann kontinuierlich und zeitunabhängig vom Aufruf. Dabei ist besonders auf einen minimalen Overhead der Nachrichten und eine geringe Latenz bei der Übertragung zu achten. Dies ist besonders bei der Übermittlung von Events zwischen den Assets wichtig.

6 Generative Implementierung der (Micro-)Services

Für die Umbesetzung der Laufzeitumgebung wird ein generatives Vorgehen eingeführt, um den Implementierungsaufwand der (Micro-)Services zu minimieren. Die generellen Schritte werden dabei in Abbildung 5 dargestellt.

6.1 Modellierung der Merkmale der Assets

Die Endpunkte der Service müssen als Merkmale der Verwaltungsschale modelliert werden (1). Dies erfolgt beispielsweise über den AAS Package Explorer [8]. Dadurch

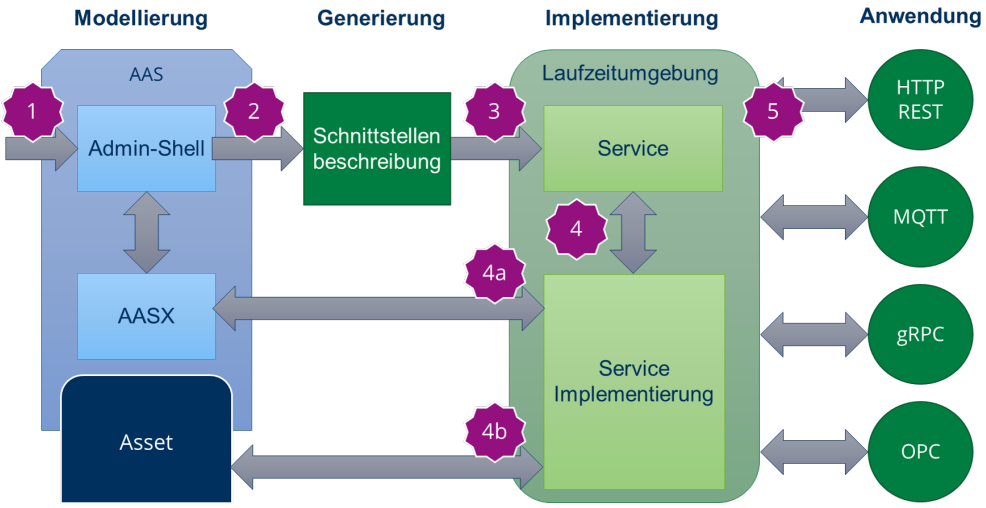

Abbildung 5: Generative Entwicklung der Laufzeitumgebung

werden zugleich die Verwaltungsschale sowie deren verbundenen bzw. eingebetteten Dateien in dem Paketformat der AASX Datei gepackt.

6.2 Generierung der Schnittstellen

Ein weiterer Vorteil in der Verwendung des AAS Package Explorer für die Modellierung der Verwaltungsschale liegt in der Generierung der Schnittstellen für die Endpunkte. Durch die von uns geschaffene Erweiterung des AAS Package Explorer kann die Schnittstellenbeschreibung mittels Protocol Buffers aus der Modellierung direkt exportiert werden (2). Dabei handelt es sich um eine einfache, leichtgewichtige und vielseitige Schnittstellenbeschreibungssprache zur Definition von Nachrichten und Services. Protocol Buffers ist sprachunabhängig und kann für verschiedene Laufzeitumgebungen genutzt werden. Für die aktive Verwaltungsschale werden die verschiedenen Merkmale auf Nachrichten bzw. Serviceaufrufe abgebildet. Diese bilden die Voraussetzung für die Services im Industrie 4.0 System. Für jedes Teilmodell wird dabei ein eigener Service in Protocol Buffers definiert. Für den Anwendungsfall wird sich auf die Abbildung von Dataelements und Operations beschränkt. Dabei werden für jede Property eine Serviceaufrufe für lesenden und schreibenden Zugriff definiert. Die Ausführung der Operations erfolgt direkt über die Serviceaufrufe. Für die Inputparameter der Operation und die Schreibzugriffe auf die Properties werden einzelne Nachrichten definiert. Ebenso werden die Rückgabewerte des Lesezugriffs auf Properties und die Outputparameter der Operations als Nachrichten codiert. Sollten dabei in der Modellierung Referenzen angewendet werden, können diese ebenfalls als eine einzige Nachricht codiert werden. Andernfalls werden dedizierte Nachrichtendefinitionen erzeugt.

6.3 Generierung der Services

Durch die Verwendung der Protocol Buffers als Beschreibungssprache können die Services direkt aus der Schnittstellenbeschreibung generiert werden (3). Dies kann für unterschiedliche Programmiersprachen und Laufzeitumgebungen erfolgen. Daher können verschiedene Teilmodelle in unterschiedlichen Sprachen realisiert werden. Dies ermöglicht es, für jeden Anwendungsfall die geeignetste Sprache und beste Toolchain zu nutzen.

6.4 Implementierung der Services

Leider bleibt die Generierung der Services durch die Schnittstellenbeschreibung auf das Erzeugen von Stubs beschränkt. Daher müssen die funktionalen Aspekte der Assets durch den Entwickler in der, zur Erzeugen der Stubs gewählten Sprache, implementiert werden (4). Dabei können zwei grundlegende Implementierung unterschieden werden. Zum einem werden die Zugriffe auf die passive Schale in der Form von Zugriffen auf die AASX Pakete betrachtet (4a). Dies kann zum Erfassen von festen Werten in Form von Properties oder Abfrage von Elementen der Verwaltungsschale wie topologische Relationen oder externe Referenzen dienen. Zum andrem müssen die Zugriffe auf die funktionalen Aspekte der Assets über die Services abgebildet werden (4b). Aufgrund fehlender Definition der Umsetzung obliegt es in diesem Fall dem Entwickler der Services, diese zu implementieren. Dies kann beispielweise über externe Prozessaufrufe, Lese- bzw. Schreibzugriff auf Programmparameter oder die Ausführung von Steuerungscode erfolgen. In jedem Fall hängt die Umsetzung stark vom jeweiligen Asset ab.

6.5 Service Endpunkte bereitgestellt

Schlussendlich müssen nur noch die Endpunkte erzeugt werden, um die Dienste nutzen zu können (5). Diese können allerdings durch eine geeignete Laufzeitumgebung der Verwaltungsschale selbsttätig erzeugt und bereitgestellt werden.

6.6 Deployment

Für die Anwendung einer SoA in Industrie 4.0 fehlt nur noch das Deployment der I4.0 Komponente. Diese I4.0 Komponente mit (pro-)aktiver Verwaltungsschale müssen die instanziierten Services des jeweiligen Assets bereitstellen. Wie im vorangegangenen Absatz beschrieben, können diese generell in 5 Schritten erstellt werden, wobei die Schritte wesentlich vereinfacht werden, wenn eine günstige Toolchain für das Deployment gewählt wird. Die Laufzeitumgebung der (pro-)aktiven Verwaltungsschale ist das Kernstück der I4.0 Komponente. Diese bildet alle pro- und reaktiven Teile der Verwaltungsschale ab. Die Laufzeitumgebung stellt für die Verwaltungsschale die Ebene dar, welche ein Betriebssystem für eine Applikation darstellt. Somit bildet die Laufzeitumgebung die Grundlage für die Kooperation und Koordination der Verwaltungsschale.

6.7 Bootstrap der Laufzeitumgebung

Bevor die Verwendung der (Micro-)Services beginnen kann, muss zuvor jedoch die Inbetriebnahme der Verwaltungsschale erfolgen. Diese beinhaltet eigenen Schritte um die Kommunikations- und Reaktionsfähigkeit herzustellen. Dadurch wird die Verwaltungsschale mit Hilfe der Laufzeitumgebung zu einer (pro-)aktiven Verwaltungsschale. Zunächst muss aber eine grundlegende Kommunikationsfähigkeit der Laufzeitumgebung hergestellt werden. Diese ist unter Anderem nötig um die Auffindbarkeit im I4.0 System zu gewährleisten oder die Konfiguration der Laufzeitumgebung zu ermöglichen.

7 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde die Problemstellung von Distributed Systems im Hinblick auf Industrie 4.0 erläutert. Dabei wurde im Speziellen auf das Assetmanagement der I4.0 Komponente als Einheit von Asset und Verwaltungsschale eingegangen. Für die in der SOA von I4.0 benötigten Services wurde die Servicegenerierung, -Implementierung sowie -Komposition vorgestellt. Ein besonderes Augenmerk erfordert dabei die Laufzeitumgebung, welche die Managementfunktionalitäten der I4.0 Assets realisiert.
Um diese möglichst flexibel und ressourcenschonend zu gestalten, wird in dem vorgestellten Konzept auf dezentrale (Micro-)Services auf der Basis von .NET zurückgegriffen. Um für den Entwickler dieser Services den Aufwand zu minimieren wird neben der platformübergreifenden und sprachunabhängigen Entwicklung von .NET Microservices ebenfalls die Servicegenerierung aus der modellierten Verwaltungsschale vorgestellt. Dies ermöglicht es, die in der Verwaltungsschale modellierten Merkmale, über die Generierung einer IDL bis hin zu Serviceendpunkten und -Stubs, eine gezielte Entwicklung der Services ohne den wiederholten Implementierungsaufwand für jeden Service erneut absolvieren zu müssen. Zum momentanen Zeitpunkt muss nach der Servicegenerierung noch die konkrete Serviceimplementierung für die (pro-)aktiven Verwaltungsschalen erfolgen. Der Zugriff auf die passiven Elemente der Verwaltungsschale kann über eine geeignete Abbildung ebenfalls generiert werden. Für den Zugriff auf die hardwarenahen Dienste der Assets erfolgt das Mapping momentan noch manuell. Gleiches gilt für die Servicekomposition und die Kooperation des Assets im I4.0 System.
Als ein weiterer entscheidender Punkt für Distributed Systems im Hinblick auf Industrie 4.0 wurde dabei die Bereitstellung unterschiedlicher Endpunkte für die einzelnen I4.0 Services herausgestellt. Im vorgestellten Konzept wurde zur Lösung eine Multi-API präsentiert, welche unterschiedliche Protokolle und Kanäle realisieren kann. Nach dem bisherigen Stand werden dabei Endpunkte über HTTP/REST, gRPC und OPC UA bedient [9]. Für zukünftige Anwendungsfälle wie fähigkeitsbasiertes Engineering [7], digitale Marktplätze und der Erweiterung der Multi-API für Abfragen der Vewaltungsschalen, kann die Laufzeitumgebung zukünftig um eine Query-API mittels GraphQL oder SPARQL ergänzt werden. Jedoch bedarf es dafür eine abgeschlossene Spezifikation der I4.0 API Definition.

Viele der vorgestellten Probleme und vorgeschlagen Lösungen befinden sich noch im Stadium der Entwicklung von Industrie 4.0. Eine Umsetzung der (pro-)aktiven Verwaltungschale als plattform- und implementierungsunabhänge (Micro-)Services stellt jedoch momentan einen der vielversprechendsten Ansätze dar. Die (Micro-)Services verfügen durch ihre lose Kopplung, minimale Zuständigkeit und maximale Vielseitigkeit über die größte Flexibilität, um auf die sich ändernden Anforderungen reagieren zu können.

Danksagung

Dieser Beitrag resultiert aus dem erfolgreich abgeschlossenen Projekt „Funktionsnachweis der Interoperabilität von fluidtechnischen Komponenten am Beispiel von Plug-and-Produce“ (FKM-Nr.: 7046610), das vom Forschungskuratorium Maschinenbau des VDMA gefördert wurde.

Literaturverzeichnis

[1] S. Bader u. a., *Details of the Asset Administration Shell. Part 1 - The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC01)*. Nov. 2020.

[2] T. Miny u. a., *Functional View of the Asset Administration Shell in an Industrie 4.0 System Environment*. Apr. 2021.

[3] S. Bader u. a., *What is the Asset Administration Shell from a technical perspective?* Apr. 2021.

[4] H. Bedenbender u. a., *Beziehungen zwischen I4.0-Komponenten – Verbundkomponenten und intelligente Produktion*. Juni 2017.

[5] N. Braunisch, S. Soler Perez Olaya und M. Wollschlaeger, „Device Management in Industrial IoT“, Nov. 2020.

[6] S. Bader u. a., *Details of the Asset Administration Shell. Part 2 -Interoperability at Runtime - Exchanging Information via Application Programming Interfaces (Version 1.0RC01)*. Nov. 2020.

[7] A. Bayha u. a., *Describing Capabilities of Industrie 4.0 Components*. Nov. 2020.

[8] A. Belyaev u. a., *Diskussionspapier VWS-Referenzmodellierung Exemplarische Modellierung einer fertigungstechnischen Anlage mit AASX Package Explorer auf Basis des VWS-Metamodells*. Apr. 2021. DOI: 10.13140/RG.2.2.17848.47368.

[9] H. Bagci und A. Kara, „A Lightweight and High Performance Remote Procedure Call Framework for Cross Platform Communication“, in *Proceedings of the 11th International Joint Conference on Software Technologies - Volume 1: ICSOFT-EA, (ICSOFT 2016)*, INSTICC, SciTePress, 2016, S. 117–124, ISBN: 978-989-758-194-6. DOI: 10.5220/0005931201170124.

http://dx.doi.org/10.25673/39551

Simulation as a Service für TSN-IA Profile

Dominik Steinmann, Dr. Stephan Höme, Dr. Sven Kerschbaum

Digital Industries, Siemens AG, {dominik.steinmann, sven.kerschbaum, stephan.hoeme}@siemens.com

Keywords: IEC/IEEE 60802 TSN-IA Profile, Profinet TSN, Simulation as a Service, CI/CD, NetDevOps, Converged Industrial Networks, OMNEST/OMNeT++, Network QoS Validation

Abstract: Bereits heute stellt das globale Marktumfeld Produzenten vor enorme Herausforderungen: steigende Produktqualität bei immer kürzeren Time-to-Market-Zyklen. Dies sowie die ebenfalls zunehmenden Einzelstückanfertigungen führen bereits heute zu einer Auflösung der starren Automatisierungspyramide. Die Tage der Automatisierungsinseln und starren Produktionsanlagen sind somit gezählt. Zukünftige Produktionsanlagen werden in einem hohen Maß flexibel und dynamisch sein. Die daraus resultierenden Anforderungen an die digitale Industrie von Morgen erfordern insbesondere auch im Hinblick an die industrielle Kommunikation sowie der zugrundeliegenden Netzwerkinfrastruktur ein erhöhtes Maß an Flexibilität und Automatisierung.

Das aus der SW-Entwicklung bekannte DevOps-Konzept zur Verbesserung der Softwarequalität und der Geschwindigkeit der Softwareauslieferung wird zunehmend um Netzwerkaspekte wie Infrastruktur und Konfiguration erweitert und unter dem Begriff NetDevOps zusammengefasst. Dieser Beitrag beschäftigt sich mit einer möglichen Übertragung des NetDevOps-Ansatzes auf zukünftige IEC/IEEE 60802 TSN-IA Profil basierende Automatisierungsnetze, um applikative Anforderungen an das konvergente Automatisierungsnetz über den gesamten Lebenszyklus hinweg zu validieren, um anschließende automatische Änderungen an Kommunikations- und Netzwerkinfrastrukturparametern zu ermöglichen.

Im Fokus des dargestellten NetDevOps-Ansatzes in Abbildung 1 steht dabei der Network Engineer, der das Netzwerk und dem aus der Applikation resultierenden Netzwerkverkehrs nicht nur auf Echtzeitgarantien von Ankunftszeiten validieren kann, sondern auch auf Mittelwerte, Häufigkeitsverteilung und weiteren Dienstgütekriterien des konvergenten, industriellen Netzes. Betrachtet werden sowohl Eigenschaften des zeitkritischen als auch zeitunkritischen Netzwerkverkehrs wie beispielsweise mit Best Effort Dienstgüte für IIoT und KI-Monitoring- und Diagnoseapplikationen. Dabei wurde eine CI/CD-Pipeline aufgesetzt, die aus einer TSN-Netzwerkplanung und dem zugrundeliegenden Algorithmus und einem auf OMNEST/OMNeT++ basierenden Netzwerk Simulators zur Evaluierung der berechneten Ablaufplanung besteht. So werden vom DevTeam kontinuierlich neue Entwicklungen und Verbesserungen eingespeist. Der Network Engineer nutzt diese CI/CD Pipeline als Simulation as a Service und übergibt dieser eine Netzwerkbeschreibung mit applikativen Anforderungen an das konvergente Automatisierungsnetz.

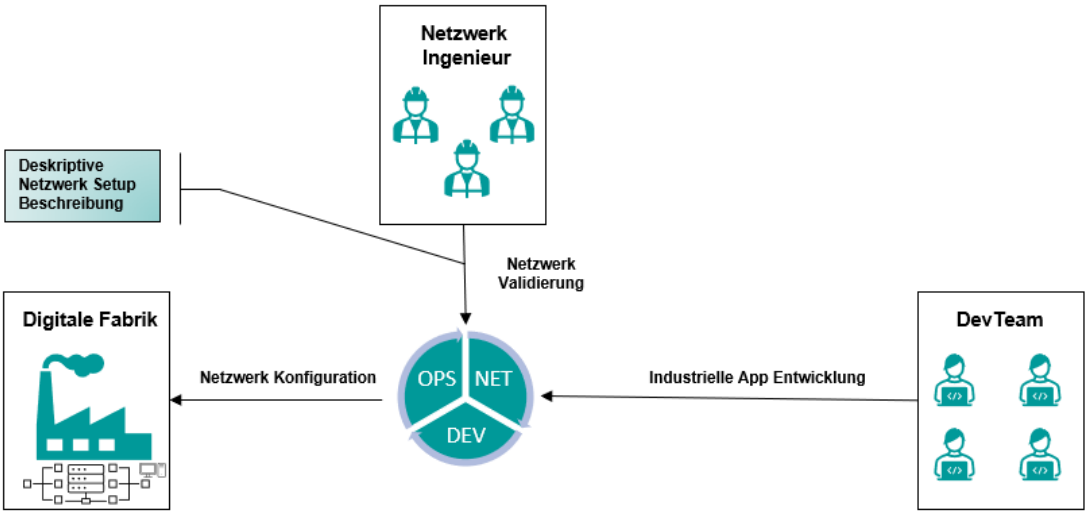

Abbildung 1: Im Fokus des industriellen NetDevOps-Ansatzes für IEC/IEEE 60802 TSN-IA Profile steht der Netzwerk Ingenieur als Anlagenbetreiber, der beispielsweise neue Komponenten in Betrieb genommen hat und den der hinzugefügten Applikation zugrundeliegenden Netzwerkverkehr auf deterministische und Best Effort QoS Eigenschaften im erweiterten, konvergenten Netzwerk der digitalen Fabrik validieren möchte.

Als Beispiel eines IEC/IEEE 60802 TSN-IA Profil basierenden industriellen Automatisierungsnetzes wurde Profinet über TSN herangezogen, da PROFINET eines der führenden Industrial Ethernet-Protokolle ist, die in der Fabrikautomation eingesetzt werden. Der TSN Scheduling Algorithmus der CI/CD Pipeline fungiert als CNC und plant die TSN Streams in Hinsicht auf Datenpfade, Sendereihenfolge und Phasenallokation.

Die Validierung des konvergenten Netzes wird durch eine in OMNEST/OMNeT++ durchgeführte Simulation erweitert, indem neben den Profinet TSN Streams auch Datenströme anderer Applikationen wie z.B. Diagnoseanwendungen, KI, usw. simuliert werden. Profinet TSN Streams haben Garantien (=Worst Case/obere Schranke), die vom CNC errechnet werden, d.h. der Anwender kann sich hierauf verlassen. Interessant sind aber u.U. auch die zu erwartenden Mittelwerte für Anlagenbetreiber, evtl. lassen sich hieraus auch Konfigurationsoptimierungen ableiten wie eines kleineren Aktualisierungsintervalls, das wiederum einen höheren Produktionstakt ermöglicht. Das vom Netzwerk Ingenieur beschriebene Netzwerk mit dem aus der Applikation resultierenden Netzwerkverkehrs wird simuliert und die Simulationsergebnisse anschließend auf die applikativen Anforderungen überprüft. Der Netzwerk Ingenieur erhält die Ergebnisse der Netzwerkvalidierung und kann gegebenenfalls nach seinen Wünschen das Netzwerk verändern, erweitern und/oder optimieren. Nach Rückmeldung vom Netzwerk Ingenieur soll die Netzwerkkonfiguration bei positiver Netzwerkvalidierung durch die CI/CD Pipeline automatisiert stattfinden.

Im Folgenden wird im Kapitel Konzeptübersicht genauer auf das Validierungskonzept der Netzwerkkonfigurationen für Profinet TSN Anwendungen als Beispiel eines IEC/IEEE 60802 TSN-IA Profils eingegangen. Anschließend wird im Kapitel Simulation as a Service die spezifizierten Simulationselemente des Validierungskonzepts näher beleuchtet. Dabei werden zum einen die verwendeten Implementierungen von standardisierten Netzwerkkomponenten und Technologien für ein Profinet TSN Applikationsbeispiel beschrieben, zum anderen die noch fehlenden Profinet über TSN Eigenschaften erläutert. Im Kapitel Vorstellung und Diskussion der Ergebnisse wird ein konkreter Usecase einer möglichen, realitätsnahen Profinet TSN-IA Applikation vorgestellt, anhand derer die Ergebnisse der implementierten Software-Komponenten der CI/CD Pipeline nachvollzogen werden können. Am Ende wird Konzept, Spezifikation und Implementierung des Simulation as a Service für IEC/IEEE 60802 TSN-IA Profile zusammengefasst und ein Ausblick gegeben.

1 Konzeptübersicht

In diesem Kapitel wird eine Übersicht des Konzepts gegeben, dass es zum Ziel hat, applikative Anforderungen an zukünftige konvergente Automatisierungsnetze, die auf dem IEC/IEEE 60802 TSN-IA Profil basieren, über den gesamten Lebenszyklus hinweg zu validieren.

In Abbildung 1 wird der zur Konzepterstellung angewandte NetDevOps-Ansatz veranschaulicht. Dabei steht der Netzwerk Ingenieur als Anlagenbetreiber im Mittelpunkt, da dieser den jeweiligen CI/CD bzw. NetDevOps Zyklus startet und beendet. Werden in der digitalen Fabrik vom Anlagenbetreiber kontinuierlich Produktionsanlagen erweitert oder umgebaut, können sich damit applikative Anforderungen ändern oder es kommen Neue hinzu, die validiert werden müssen. Der Netzwerk Ingenieur speist in Form einer Netzwerkbeschreibungssprache die Änderungen in den Zyklus bei dem Element *Net* ein und startet diesen damit. Diese Beschreibung dient zum einen dem CNC-Algorithmus zur Berechnung der Datenpfade über das TSN Netzwerk und zum anderen der Netzwerksimulation zur Nachbildung der realen Netzwerk Topologie und dem geplanten Traffic Szenarios.

Im Element *Dev* kommen neue oder erweiterte Industrieapplikationen wie der CNC-Pfadplanung oder der Netzwerksimulation hinzu. Die Netzwerksimulation bildet das reale Netzwerk der digitalen Fabrik und das Traffic Szenario nach und prüft die vom CNC geplanten Datenpfade des zeitkritischen Verkehrs zusammen mit dem zeitunkritischen Netzwerkverkehr. Die dem Netzwerk betreffenden Änderungen können somit bereits vor der realen Inbetriebnahme bzw. vor dem Umbau/Erweiterung vor Ort validiert werden und dadurch können unvorhergesehene Netzwerkausfälle vermieden werden. Ein weiterer Use Case wäre die Anbindung neuer IIoT-Applikationen z.B. zur Bereitstellung prozessnaher Daten wie Apps im Edge-Umfeld.

Bei einer validen Konfiguration ist im dritten Element des Zyklus *Ops* eine automatische Änderung an Kommunikations- und Netzwerkinfrastrukturparametern möglich. Bei einer nicht validen, aber auch validen Konfiguration kann der Anlagenbetreiber so lange die Netzwerkkonfiguration verändern, bis eine Netzwerkeinstellung und Setup nach seinen Optimierungskriterien gefunden wurde. Kriterien können die Reduktion von Kosten für die Netzwerkinfrastruktur sein, um beispielsweise eine Überdimensionierung zu vermeiden oder Verringerung der Latenzzeiten, die Einfluss auf die Applikation und damit dem

Produktionstakt haben können. Werden Industrieapplikationen vom DevTeam weiterentwickelt oder optimiert, können durch die Verwendung der Network Simulation as a Service Anlagenbetreiber eine neue oder die aktuelle Netzwerkbeschreibung der digitalen Fabrik auf Auswirkungen prüfen, um weiterhin festzustellen, dass die applikativen Anforderungen erfüllt werden. Vor allem aber kann auch mittels Simulation das Verbesserungspotential im konvergenten Netzwerk aller Verkehrsklassen insbesondere des nicht deterministischen QoS Verkehrs zur Rechtfertigung von Softwareupdates der relevanten Komponenten dargestellt werden. Insgesamt reduziert sich damit Time-to-Market.

Im nächsten Kapitel wird auf die Simulation des geplanten Netzwerks und dem zugrundeliegenden Netzwerkverkehrs eingegangen.

2 Simulation as a Service

In diesem Kapitel wird zunächst das spezifizierte Simulationstool zur Netzwerkvalidierung und Evaluierung des CNC-Pfadplanungsalgorithmus und die verwendeten und erweiterten Simulationskomponenten für Profinet Applikationen als Beispiel von IEC/IEEE 60802 TSN-IA Profile vorgestellt. Danach wird auf die Automatisierung der Netzwerkvalidation eingegangen, indem das gewählte Simulationstool in ein ausführbares Docker Image eingebettet und als solches in eine GitLab CI/CD eingebunden wird. Es wird dargestellt, wie mittels Skripte die Ausführung der Simulation mit anschießender Auswertung der Simulationsergebnisse automatisiert wird.

2.1 OMNeT++ als Simulationstool für Netzwerkszenarien auf Basis von TSN-IA Profile

Als Simulationstool wurde OMNeT++ bzw. die kommerziell bezeichnete Version OMNEST [AB02] gewählt. Dies ist ein Netzwerksimulator, mit dem zu diskreten Zeitpunkten Netzwerkereignisse simuliert werden können. Zu dem Basisframework wurden weitere frei verfügbare OMNeT++ Frameworks hinzugefügt. Das Framework INET [AB01], von dem Version 4.1.2 verwendet wird, liefert dabei Module für grundsätzliche Switching Mechanismen. Das Framework NeSTiNg [AB00] erweitert INET um TSN-fähige Komponenten. Es werden die Features Frame Preemption nach IEEE 802.1Qbu und IEEE 802.3br, VLAN tagging und Strict Priority Scheduling genutzt.

Erweitert wurde NeSTiNg um das selbst entwickelte PNTSN Framework. PN steht dabei für Profinet. Es wurden Netzwerkknoten implementiert, die die im vorangegangenen Abschnitt aufgelisteten Netzwerkeigenschaften unterstützen. Darüber hinaus werden diese nach dem Profinet TSN Knotenmodel hinsichtlich der Verschaltung interner Ports von Endstationen bzw. der CPU, die als Profinet Controller oder Device fungiert und der Bridge bzw. der Switching-Netzwerkkomponente und des entsprechenden Delay Models umgesetzt. Aktuell sind die Endstationen mit einer idealen Clock synchronisiert, was bedeutet, dass auch exakt zu den eingeplanten Zeitpunkten die Frames verarbeitet und weitergeleitet werden. So kann zum Zeitpunkt 0 der Simulationszeit für jeden Ausgangsport eines Knotens das erste Bit aus dem ersten Frame der sortierten Sendeliste auf der Leitung anliegen. In dem für die Simulation umgesetzten PN over TSN Delay Model wurde auch die Option berücksichtigt, dass es auch herstellerspezifische Eigenschaften von ASIC HW geben wird, bei denen die Data Plane für beispielsweise die höchste VLAN Priorität 7 optimiert ist und damit kleinere Bridge Delays von Eingangs- zu Ausgangsport im Verlgleich zu den übrigen Traffic Classes 0 bis 6 möglich sind.

Es wurde der Forwarding Mechanismus Cut-Through des NeSTiNg Frameworks weiterentwickelt. Für PN over TSN wird ein Delayed Cut-Through HW Feature verlangt, was aber bisher in IEEE 802 fehlt. Die Forwarding Verzögerung bei dem Store-and-Forward Mechanismus lässt sich nach der folgenden Formel berechnen:

$$ForwardingDelay = independentDelayMax + dependentDelayMax \times FrameSize$$

Im Gegensatz zur Verzögerungsberechnung bei dem Store-and-Forward Mechanismus, bei dem die gesamte Ethernet Frame Größe in Abhängigkeit der Portgeschwindigkeit zu dem independentDelayMax addiert wird, entspricht die Forwarding Verzögerung bei dem Cut-Through Mechanismus dem independentDelayMax:

$$ForwardingDelay = independentDelayMax$$

Das independentDelayMax wird deshalb als unabhängig bezeichnet, da es nicht von der Ethernet Frame Länge abhängt. Bei Cut-Through wird bei Empfang eines Ingress Ethernet Frames die Destination MAC

Adresse aus dem Ethernet Header gelesen. Zusätzlich wird bei dem implementierten Delayed Cut-Through Mechanimus das PCP Feld des VLAN Headers ausgelesen, da es, wie im Abschnitt zuvor erwähnt wurde, herstellerspezifische Eigenschaften von ASIC HW geben wird, bei denen unterschiedliche VLAN Prioritäten verschiedene Verzögerungen aufweisen können. Nach dem Verarbeiten der Information aus Ethernet und VLAN Header wird das Ethernet Frame direkt an den Egress Port in die jeweilige Queue in Abhängigkeit der VLAN Priorität weitergeleitet. Neben dem Bridge Delay kann die Verweildauer des Ethernet Frames an dieser Bridge durch gleich oder höherpriore Frames verzögert werden, die vor diesem in die Warteschlangen des Ausgangsports eingereiht wurden. Zusätzlich gibt es bei der Anwendung von Frame Preemption eine minimale Framelänge von Preemptable Frames, die von Express Frames nicht mehr unterbrochen werden kann, was zu einer weiteren Verzögerung führen kann. Cut-Through wurde um Delayed Cut-Through erweitert, so dass es nicht wie bisher zu einem Umschalten von Cut-Through auf Store-and-Forward kommt, sondern Ingress Ethernet Frames nach Lesen des Ethernet und VLAN Headers direkt in die Queue des Ziel Egress Ports weitergeleitet und lediglich um die beiden genannten Gründe verzögert werden.

Bevor die OMNeT++ Simulation gestartet werden kann, müssen die in Abbildung 1 vom Netzwerk Ingenieur zur Netzwerkvalidation übergebenen, deskriptiven Netzwerk Setup Beschreibungen in OMNEST/OMNeT++ Konfigurationsdateien transpiliert werden. Die Übersetzungen vor der Simulation werden als Preprocessing bezeichnet. Skripte automatisieren diesen Prozess. Als Beispiel deskriptiver Netzwerk Setup Beschreibungen werden für die Simulation die gleichen in Yaml Syntax codierten Konfigurationsdateien wie für den CNC-Pfadplanungsalgorithmus verwendet. Für diesen gibt es die folgenden Dateien: networks, end-stations, streams, stream-requests, stream-planning-states.

In den networks Dateien wird die Netzwerk Topologie beschrieben. Diese beinhalten eine Liste der in der Topologie verfügbaren Knoten, wie diese miteinander über welche Link Geschwindigkeiten und Kabellänge verbunden sind, welcher Forwarding-Mechanismus z.B. Store-and-Forward oder Delayed Cut-Through jeweils verwendet wird, Delay Werte für den Fall von Frame Preemption und für das Bridge Delay von Eingangs- zu Ausgangsport und die Größe des Queue Speichers für die einzelnen Warteschlangen des Ausgangsports.

TSN Streams können in den streams Dateien festgelegt werden. Hierbei kann der Startknoten konfiguriert werden, an dem dieses Frame verschickt werden soll und der Endknoten, an dem es empfangen werden soll. Weitere Parameter sind die Priorität und damit auch Festlegung der Traffic Klasse. Der Wert high steht beispielsweise für die Stream bzw. Traffic Klasse Stream High, low analog dazu für Stream Low und RT für Stream Real-time. Außerdem kann die Ethernet Frame Länge in Bytes angegeben werden und das Profinet Übetragungsintervall bzw. Stream Zykluszeit.

Über die stream-request Datei erhält der CNC-Pfadplanungsalgorithmus die Informationen, einzelne oder auch mehrere TSN Streams zu bestimmten Zeitpunkten in der Planung hinzuzufügen oder zu entfernen.

In der end-stations Datei kann für jeden Knoten der geplanten Topologie ein Scheduling Cycle bzw. in Profinet Nomenklatur ein Applikationszyklus konfiguriert werden.

Anschließend kann der Netzwerkverkehr über die Informationen aus den networks, streams, end-stations und stream-request Dateien inkrementell geplant werden. Nach jeder einzelnen Berechnung gibt der CNC-Pfadplanungsalgorithmus eine Datei namens stream-planning-states aus. Diese beinhaltet eine Liste aller in streams definierten TSN Streams und liefert den Zustand, ob das Hinzufügen des jeweiligen Streams akzeptiert wurde oder nicht. Abbildung 2 zeigt alle verfügbaren Ausgabewerte des CNC-Pfadplanungsalgorithmus in der stream-planning-states Datei am Beispiel des TSN Streams 1. Im Folgenden wird auf die wichtigsten Parameter eingegangen. Der Pfad gibt den Start- und Endknoten und alle auf den Weg genutzten Knoten an. Der Parameter phases liefert die Information, in welchem Phasenzyklus der Stream am Startknoten allokiert wird. Im Beispiel aus Abbildung 2 wurde für dieses Netzwerkszenario in der end-stations Datei ein Scheduling Cycle bzw. Applikationszyklus von 1 ms konfiguriert. Das Profinet Übetragungsintervall bzw. Stream Zykluszeit des TSN Streams mit id 1 wurde auf 8 ms festgelegt, das heißt einer Reduction Ratio, abgekürzt RR, von 8 in Profinet Nomenklatur. Da der maximale RR Wert 8 über alle TSN Streams ist, die am Knoten 1 geplant werden sollen, wird hier auch von einer Hyperperiode von 8 ms gesprochen, da sich nach allen 8 ms der Plan für das Scheduling wiederholt. Innerhalb dieser Periode gibt es 8 Phasen, in denen nun der TSN Stream mit id 1 eingeplant werden könnte. Der CNC-Pfadplanungsalgorithmus hat diesen Stream in Phase 6 eingeplant. Die Sendereihenfolgeposition des Streams gibt wieder, in welcher Position in der jeweiligen Phase der Stream bei dem Startknoten allokiert wird. Der Parameter worst-case-arrival-time-estimate stellt die Ankunftszeit des Streams am Endknoten im Worst Case zur Verfügung.

```
streams:
  - id: 1
    path: [1, 2, 38]
    phases: [6]
    starting_time: [0.006000087296]
    interferences: [4.6504832e-05]
    travel_time_without_interference: 3.8028e-06
    travel_time_without_interference_per_hop:
      - 1.848e-06
      - 1.6014e-06
      - 3.533999999999999e-07
    total_traveling_time: [5.0220336e-05]
    frame_write_delay_at_send_node: 6.08e-10
    acceptance: true
    worst_case_arrival_times_estimate: [0.0060503076320000001]
    arrival_deadlines: [0.0065000000000000001]
    send_order_positions: [124]
    phase_tree_at_allocation: [7.6000000000000011e-08, 7.6000000000000011e-08, 7.6000000000000011e-08, 7.6000000000000011e-08, 7.6000000000000011e-08,
    7.6000000000000011e-08, 7.5392000000000011e-08, 7.5392000000000011e-08]
    interferer_per_phase:
      - 6: [1019, 1003, 987, 971, 905, 921, 937, 953, 889, 893, 865, 849, 809, 795, 811, 827, 777, 771, 735, 765, 749, 703, 689, 673, 719, 655, 639, 623, 607, 555,
      571, 587, 539, 493, 491, 525, 489, 469, 441, 415, 399, 365, 381, 389, 321, 327, 343, 357, 259, 299, 283, 255, 257, 239, 223, 207, 191, 123, 179, 133, 149,
      63, 79, 95, 111, 29]
    interference_per_phase_per_hop:
      - 6: [4.6504832e-05, 0, 0]
```


Abbildung 2: In diesem Auszug aus der Datei stream-planning-states des CNC-Pfadplanungsalgorithmus werden die Werte für Stream ID 1 gezeigt. Dieser TSN Stream einer beispielhaften Profinet Applikation wird von einem Profinet IO Controller als Node 1 an das Profinet IO Device als Node 38 über die Bridge als Node 2 gesendet.

Skripte automatisieren nun die Übersetzung der networks, end-stations, streams, stream-requests, stream-planning-states Dateien nach der Berechnung des CNC-Pfadplanungsalgorithmus in OMNEST/OMNeT++ benötigte Konfigurationsdateien. Abbildung 3 listet zum einen die zur Ausführung der Simulation notwendigen OMNEST/OMNeT++ Konfigurationsdateien in der linken Spalte auf, zum anderen die für das Transpilieren relevanten Dateien des CNC-Pfadplanungsalgorithmus in der rechten Spalte. In der .ini Datei werden initiale Werte für die eingeplanten Netzwerk Knoten und deren Komponenten gesetzt, in der .ned Datei wird die Topologie beschrieben, die sendListControl.xml Datei enthält für jeden Startknoten die relevanten Informationen für die Ablaufplanung der Datenpfade und schließlich braucht jede Bridge eine Forwarding Database, kurz FDB, um TSN Streams mit bestimmter Ethernet Multicast Adresse an ein oder mehrere Ports zu den jeweiligen Listener der Talker-Listener Beziehung des TSN Streams weiterzuleiten.

OMNeT++ Konfigurationsdateien	**Benötigte Dateien des CNC-Pfadplanungsalgorithmus**
scenario.ini	networks
scenario.ned	networks
sendListControl.xml	streams, stream-planning-state, networks
FDB.xml	networks, stream-planning-states

Abbildung 3: In der linken Spalte sind die für die OMNeT++ Simulation benötigten Konfigurationsdateien und in der rechten Spalte die zur Erstellung nötigen Informationen aus den Schnittstellenparametern des CNC-Pfadplanungsalgorithmus

Nach dem Preprocessing kann nun die OMNEST/OMNeT++ Simulation mit den entsprechenden Konfigurationsdateien aus der linken Spalte in Abbildung 3 ausgeführt werden. Abbildung 4 zeigt im Design-Modus von OMNEST/OMNeT++ einen Topologieausschnitt der transpilierten scenario.ned Datei einer Profinet Applikation als Beispiel eines IEC/IEEE 60802 TSN-IA Profils, das im nächsten Kapitel 3 beschrieben wird. Die nach dem Profinet TSN Knotenmodel implementierten OMNEST/OMNeT++ Netzwerk Knoten werden in der Abbildung 4 als es_id_x bezeichnet, wobei es für endstation steht und id in Kombination mit x auf die in den networks.yaml gleichen Knoten id's hinweisen. Wie bereits vorausgehend erklärt, handelt es sich nicht nur um eine Endstation, es ist zudem ein Switch integriert. Das vom Netzwerk Ingenieur beschriebene Szenario kann neben dem simulierten deterministischen QoS TSN Streams auch auf den Best Effort QoS Traffic untersucht werden. Nach Ende der Simulationszeit lassen sich durch programmierte Signale in OMNEST/OMNeT++ Statistiken erfassen. Es wird eine Datei mit skalaren Werten dieser Signale im Format .sca abgespeichert. Neben Standardsignalstatistiken der Netzwerk Komponenten, die in OMNEST/OMNeT++ nach dem Profinet TSN Knotenmodel umgesetzt wurden, wie zum Beispiel die Anzahl verworfener Pakete packetDropped:count(scalar) oder die Datenrate empfangener Express oder Preemptable Frames und Anzahl der Frame Preemptions im Ethernet MAC Modul wurden die nachfolgenden Signale erweitert. Es wurden Signale zur Statistikerfassung der bytegranularen Speicherauslastung für die einzelnen Queues der Ausgangsports programmiert, weitere Statistikparameter zur Nachverfolgung von Ankunftszeiten der konfigurierten TSN Streams und jeweils die Verzögerungszeiten pro Hop auf dem Datenpfad von Sende- zu Empfangsknoten.

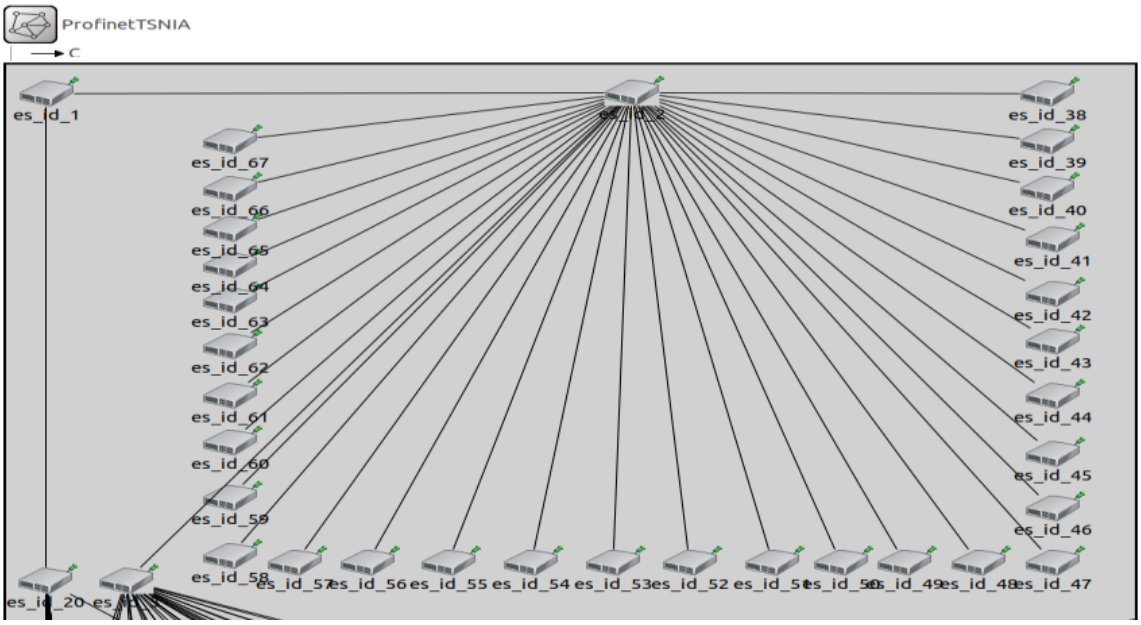

Abbildung 4: Visualisierung eines Topologieausschnitts der transpilierten scenario.ned Datei im OMNEST/OMNeT++ Design Modbus der im nächsten Kapitel 3 vorgestellten Profinet Appliktion als Beispiel eines IEC/IEEE 60802 TSN-IA Profils.

Nach Ende der Simulationsausführung und Ausgabe der Skalarstatistikdatei kann im Postprocessing mit dem im OMNEST/OMNeT++ Basisframework enthalten Tool namens Scavetool die .sca formatierte Skalarstatistikdatei zur leichteren Handhabbarkeit in ein .csv Format übersetzt werden. Anschließend werden die vom CNC-Pfadplanungsalgorithmus berechneten Worst Case Ankunftszeiten auf dem Pfad vom Sende- zum Empfangsknoten pro Hop der akzeptierten TSN Streams mit denen aus der Simulation verglichen. Zusätzlich findet neben der Analyse des deterministischen Traffics auch eine des Best Effort Traffics auf Ankunftszeiten, verworfene Pakete und Kapazitätsauslastung des Queue Buffers statt. Im nächsten Kapitel wird auf die Realisierung einer GitLab CI/CD Pipeline zur automatisierten Validierung und Evaluierung von Netzwerkszenarien nach deterministischen und Best Effort QoS Traffics eingegangen. Nach einem erfolgreichen Durchlauf dieser CI/CD Pipeline in Gitlab können Artifakte heruntergeladen werden. Dabei wird automatisiert zu jedem Szenario ein Netzwerkgraph erstellt und in diesen Artifakten bereitgestellt. Zudem wird nach der Simulation im Postprocessing der Unterschied der Ankunftszeiten der geplanten TSN Streams von Simulation und CNC-Pfadplaungsalgorithmus graphisch visualisiert. In einer Textdatei befinden sich alle relevanten Simulationsergebnisse zu den zuvor aufgezählten Statistikwerten.

2.2 GitLab CI/CD Pipeline zur automatisierten Validierung und Evaluierung von Netzwerkszenarien nach deterministischen und Best Effort QoS Traffics

In diesem Kapitel wird die Umsetzung des Konzepts aus dem ersten Kapitel und die Automatisierung der Simulationsausführung und Nutzen als Simulation as a Service im NetDevOps-Zyklus aus Abbildung 1 präsentiert. Es wird die Validierung des Netzwerkentwurfs des Netzwerk Ingenieurs bzw. Anlagenbetreibers automatisiert, also die Überprüfung, ob sich die Planung des deterministischen kombiniert mit Best Effort Traffics nach den festgelegten und erwarteten QoS Anforderungen verhält.

Über eine GitLab CI/CD Pipeline wird bei Änderung der Netzwerkbeschreibungsdateien oder der Implementierung des CNC-Pfadplanungsalgorithmus der NetDevOps-Zyklus aus Abbildung 1 automatisch gestartet. Die GitLab CI/CD Pipeline besteht aktuell aus vier Stufen, sog. Stages, wie in Abbildung 5 dargestellt. Beschrieben werden diese in der .gitlab-ci.yml Datei.

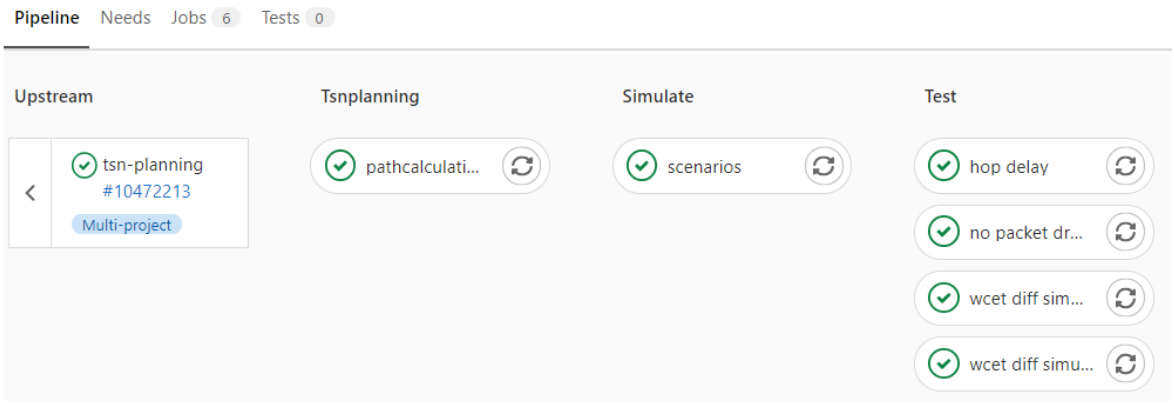

Abbildung 5: CI/CD Pipeline zur automatisierten Validierung der Network QoS (Streams als auch Best-Effort Verkehr)

In Stage Upstream wird bei erfolgreichem Durchlaufen der CI/CD Pipeline des GitLab Repositories tsn-planning, in dem der Code des CNC-Pfadplanungsalgorithmus liegt, die CI/CD Pipeline mit den Stages Tsnplanning, Simulate und Test getriggert. In Stage Tsnplanning kann damit direkt bei Triggern durch Upstream der aktuelle Codestand des CNC-Pfadplanungsalgorithmus verwendet werden und überprüft werden, ob die Änderungen eine Auswirkung zeigen. Wenn der Netzwerk Ingenieur in die Ordnerstruktur aus Abbildung 6 neue Netzwerk Szenario Beschreibungsdateien hinzufügt und in run-scenarios.yaml angibt, welche Szenarien mit welcher Simulationsausführungszeit simuliert werden sollen, kommt der aktuelle zuletzt getaggte Codestand zur Verwendung. Bei dem Tagging Prozess in GitLab wird ein Docker Image erstellt, in dem der CNC-Pfadplanungsalgorithmus des tsn-planning Repositories mit allen notwendigen Softwareabhängigkeiten ausführbar ist. Dieses Docker Image kann aus der Container Registry des tsn-planning Repositories heruntergeladen werden. Die in Abbildung 6 dargestellten Ordner und Dateien werden in dieses Docker Image gemountet. Bei Ausführung des plan-scenarios.py Skriptes in diesem Docker Image werden alle Szenarien aus run-scenarios.yaml und die entsprechenden networks, streams, stream-requests und end-stations Beschreibungen mittels CLI Kommandos zur Berechnung an den CNC-Pfadplanungsalgorithmus übergeben. Im Parameter stream-planning-results wird aktuell nur der Name für die stream-planning-states Datei genutzt. Die berechneten Ergebnisse werden, wie in Kapitel 2.1 beschrieben, in der Datei stream-planning-states ausgegeben und für die nächste Stage Simulate als Artifakt weitergegeben.

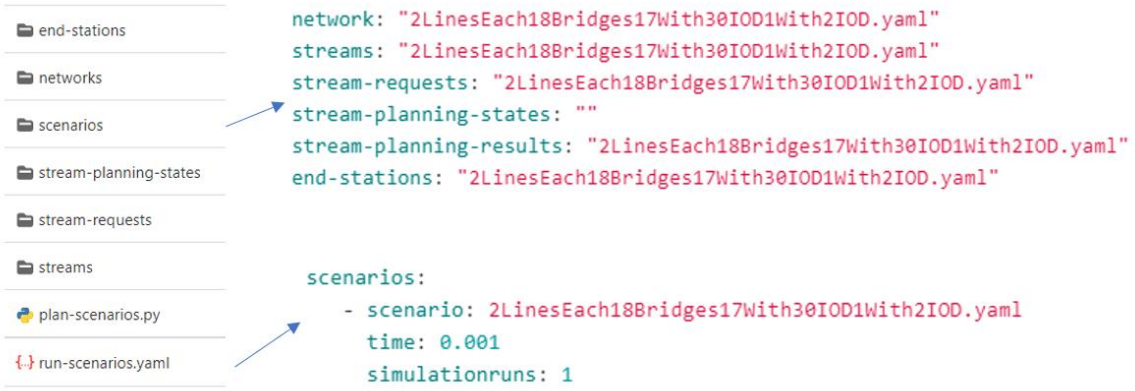

Abbildung 6: Ordnerstruktur des Datenmodels zur Netzwerk Szenario Beschreibung

In Stage Simulate wird ebenso ein Docker Image aus der Container Registry des GitLab Repositories, in dem der Source Code vom Basisframework OMNeT++ bzw. OMNEST mit den erweiterten Frameworks INET, NeSTiNg und PNTSN enthalten ist, heruntergeladen und ausgeführt. Die Artifakte aus der Stage Tsnplanning werden gemountet, so dass die Netzwerk Beschreibungsdateien und berechneten Ergebnisse für das mittels genannter Skripte automatisierte Preprocessing, Simulationsausführung und Postprocessing bereitstehen.

In der letzten Stage Test werden die Ergebnisse des Postprocessings zur vereinfachten Übersicht veranschaulicht.

Im nächsten Kapitel werden die Ergebnisse anhand eines Beispiel Netzwerkszenarios im Detail vorgestellt und diskutiert.

3 Vorstellung und Diskussion der Ergebnisse der CI/CD Pipeline anhand eines exemplarischen Netzwerkszenarios auf Basis eines Profinet TSN-IA Profils

Es wurden verschiedene Netzwerktopologien nach Linien-, Stern- und Baumstruktur mit unterschiedlichen Netzwerkverkehrsszenarien vom Pfadplanungsalgorithmus durchgerechnet und von dem diskreten Netzwerksimulationstools OMNEST/OMNeT++ simuliert, wobei die Topologie aus Abbildung 7 und das im Folgenden präsentierte Netzwerkverkehrsszenario zur Beschreibung und Interpretation der Ergebnisse der CI/CD Pipeline herangezogen wird.

Nach den Anforderungen an die Profinet über TSN Netzwerkdomäne muss das Mengengerüst den Fall abdecken können, dass 1 – 8 Profinet IO Controller und insgesamt 1024 Profinet Komponenten in der gleichen Domäne vorhanden sein können [AB03]. Die Netzwerktopologie aus Abbildung 7 besteht aus einem Profinet IO Controller Node 1, abgekürzt IOC, und 1023 PNIO Devices, abekürzt IOD, die zu diesem eine Profinet Application Relation, abgekürzt AR, aufgebaut haben und Daten austauschen. Dabei sind die IODs auf zwei Linien aufgeteilt. An zwei Ports des IOCs sind jeweils 18 Netzwerkknoten als einfache Bridges verbunden. Davon sind an 17 Bridges jeweils 30 IODs verbunden und an der letzten Bridge der linken Linie die IODs Node 548 und 549 und der rechten Linie das IOD Node 1060. In dem hier veranschaulichten Beispiel werden ausschließlich „Cut-Through“ Bridges bzw. Switches verwendet, die mit 1 Gbit/s Links miteinander verbunden sind. Das Bridge Delay beträgt 8 ns und die Queue Buffer Größe für jede der acht Queues eines Ausgangsports einer Bridge 520.000 Byte. Auf diese acht Queues werden nach Strict Priority die VLAN getaggten Ethernet Frames nach den acht Verkehrsklassen der acht VLAN Prioritäten verteilt.

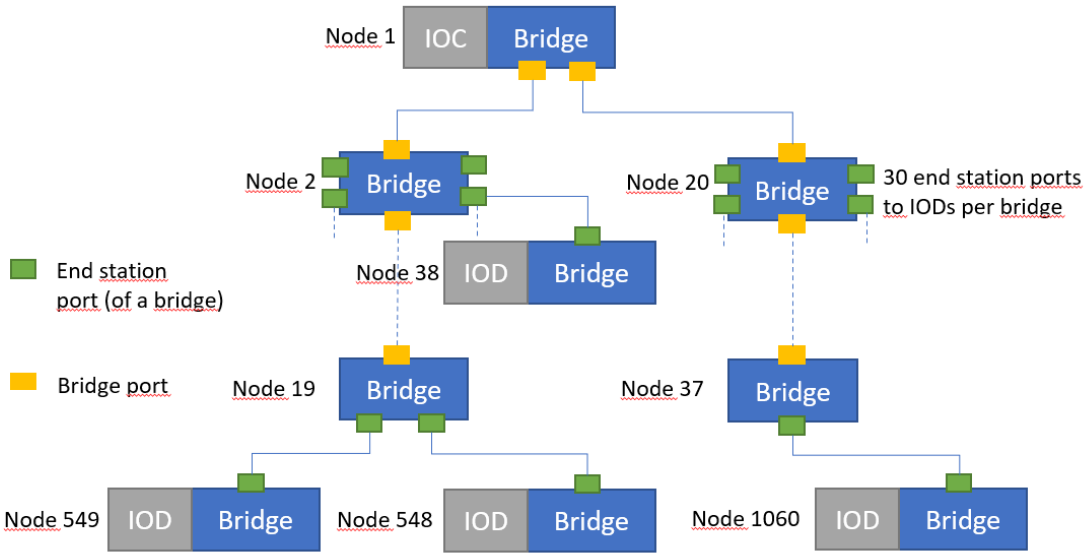

Abbildung 7: Diese Netzwerk Topologie veranschaulicht eine Profinet Applikation als Beispiel eines IEC/IEEE 60802 TSN-IA Profils mit einem Profinet IO Controller als Node 1 und 1023 Profinet IO Devices, zu denen eine Profinet Application Relation besteht.

Für das Profinet über TSN Verkehrsszenario wird zum einen eine isochrone Anwendung simuliert, bei der jeweils ein TSN Stream mit der Verkehrsklasse Stream High von IOC zu jedem der 1023 IODs aus Abbildung 7 und umgekehrt übertragen wird. So ergeben sich insgesamt 2046 TSN Streams mit jeweils einer Stream Zykluszeit von 8 ms. Da die Netzwerk- bzw. Ablaufplanzykluszeit für jeden Node auf 8 ms eingestellt ist, liegt ein RR von 1 vor, das heißt jeder der 2046 TSN Streams kann bei dem Startknoten in der Phase von 8 ms eingeplant werden. Das Übertragungsintervall liegt damit bei 8 ms. Die Deadline ist 50% der Netzwerkzykluszeit, also 500 µs. Kann ein einziger Stream eines Stream-requests inkrementell nicht mehr so eingeplant werden, dass die Ankunftszeit im schlechtesten Fall beim Empfangsknoten subtrahiert der Sendezeit beim Startknoten den Wert von 500 µs übersteigt, dann wird der gesamte Stream-request abgelehnt. In diesem Szenario werden die 2046 TSN Streams jeweils als einzelner Stream-request dem CNC-Pfadplanungsalgorithmus zur inkrementellen Berechnung übergeben. Zu diesem zeitkritischen Verkehr der 2046 TSN Streams mit harten Echtzeitanforderungen wird zusätzlich Verkehr mit Best Effort Dienstgüte für IIoT und KI-Monitoring- und Diagnoseapplikationen simuliert. Dieser Verkehr unterliegt weichen Echtanforderungen, das heißt das bei Überschreitung der Deadline von 500 µs die Ethernet Frames nicht verworfen werden. Somit nimmt für die Applikation lediglich die Nutzerqualität im schlechtesten Fall ab, indem Monitoring- oder Diagnosedaten zeitverzögert am Empfangsknoten zur Verfügung stehen.

Als erstes wird auf die Ergebnisse des zeitkritischen Verkehrs einer isochronen Profinet Applikation als Beispiel eines IEC/IEEE 60802 TSN-IA Profils eingegangen. Abbildung 8 zeigt einen Plot der Ankunftszeiten aller 2046 TSN Streams auf der y-Achse in der Einheit Mikrosekunden, wobei blau für das berechnete Ergebnis des CNC Pfadplanungsalgorithmus steht und rot für das simulierte Ergebnis der OMNEST/OMNeT++ Simulation. Auf der x-Achse wird links der Betrag der größten Differenz zwischen beiden aufgetragen. Nach rechts nimmt diese ab. Die CI/CD Pipeline gibt in der Stage Simulate aus Abbildung 5 diesen Plot mit einer Beschreibungsdatei aus, bei der die Werte der x-Achse als Index Werte für die jeweiligen Flow IDs und der Phase der TSN Streams nachgeschaut werden können. Wie in Abbildung 8 zu sehen ist, ist bei Index 0 der größte Wert für die Ankunftszeit, dieser liegt unterhalb von 400 µs. Daher wurden alle TSN Streams akzeptiert, da die Ergebniswerte die Deadline von 500 µs nicht überschritten haben. Die berechnete Ankunftszeit der 2046 TSN Streams der CNC Pfadplanung für den schlechtesten Fall liegt bei allen über dem Ergebnis der OMNEST/OMNeT++ Simulation. Es wird erwartet, dass die Simulation Ankunftszeiten im Worst Case gleich oder unterhalb der des Algorithmus aufweist, da der Algorithmus den mathematisch, theoretischen Worst Case erfasst, die Simulation aber meist darunter liegt, da es nicht bei jedem Hop zu einer Verzögerung durch Frame Preemption und durch interferierende TSN Streams gleicher Priorität, die zum gleichen Zeitpunkt auf dem Pfad eingespeist wurden, kommt.

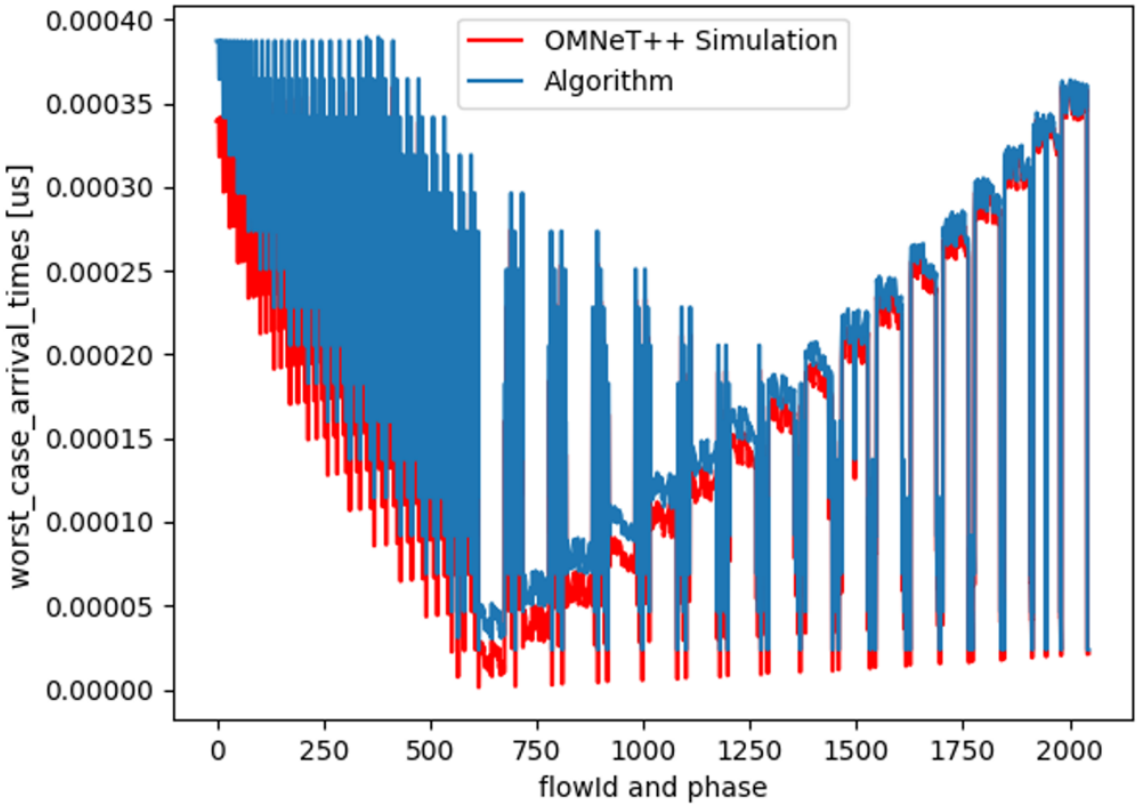

Abbildung 8: Plot der Ankunftszeiten im schlechtesten Fall der 2046 TSN Streams in der Einheit Mikrosekunden auf der y-Achse und nach der größten Differenz zwischen den Ergebnissen aus der Berechnung des CNC Pfadplanungsalgorithmus blau und der OMNEST/OMNeT++ Simulation rot von links nach rechts auf der x-Achse aufgetragen.

Als nächstes wird auf den zusätzlichen Erkenntnisgewinn der OMNEST/OMNeT++ Simulation as a Service bei nicht-zeitkritischen Verkehr eingegangen. Nicht-zeitkritischer Verkehr mit beispielsweise Best Effort Dienstgüte für IIoT und KI-Monitoring- und Diagnoseapplikationen kann nun vom Anlagenbetreiber für die unterschiedlichsten Anlagen und damit Traffic Einspeisung bei den IODs in Abbildung 7 so lange erweitert werden, bis Kapazitätsgrenzen erreicht werden. Die Grenzen können über die aus der CI/CD Pipeline ausgegebenen Beschreibungsdatei an verschiedenen Parametern festgestellt werden. Es wird die Verteilung der Speicherauslastung, also maximaler, minimaler und durchschnittlicher Wert der Queues der Ausgangsports angegeben, über die der Best Effort Traffic weitergeleitet wird. Der Anlagenbetreiber kann über dieses Feedback der Simulation die Trafficlast verteilen, indem beispielsweise die Sendezeitpunkte der Applikationen variiert werden. Wenn bei maximaler Auslastung kein Speicher mehr vorhanden ist, wird dies über einen Parameter direkt ersichtlich, der die Anzahl der verlorenen Pakete wiedergibt. Ein weiterer Parameter ist die maximale, minimale und durchschnittliche Ankunftszeit der Best Effort Frames. Des

Weiteren werden Plots ausgegeben, die die Ankunftszeit und Speicherlastverteilung der Queues als Histogramm darstellen. Werden die Anforderungen an die Applikation nicht in gewünschtem Maße erreicht, kann der Anlagenbetreiber Schnittstellenparameter der Netzwerkbeschreibung aus Abbildung 1 erst einmal ohne Veränderung der Verdrahtung oder gar Hinzufügen neuer Knoten verändern und überprüfen, ob die Anforderungen bereits erfüllt werden. Auf diese Weise wird es über die Simulation as a Service möglich, optimale Applikations- und Netzwerkeinstellungen zu finden, bei denen die Applikationsanforderungen aller Verkehrsklassen aus VLAN, Profinet oder TSN Protokollen erfüllt werden, gleichzeitig aber die Anzahl an Netzwerkknoten und Verbindungen minimal ist, um Beschaffungs-, Installations- und Wartungskosten zu reduzieren.

4 Zusammenfassung und Ausblick

Es wurde zu Beginn das NetDevOps Konzept aus Abbildung 1 vorgestellt, bei dem der OMNEST/OMNeT++ Netzwerksimulator als Simulation as a Service zur Netzwerk Validierung von zeitkritischen kombiniert mit zeitunkritischen Netzwerkverkehrs integriert wurde. Damit lässt sich neben fest geplanten vom CNC vor der Laufzeit berechneten Pfadplanungen für deterministischen Verkehr auch eine Aussage des Verhaltens des übrigen Verkehrs treffen. So kann die minimale, maximale, im Durchschnitt und im Histogramm verteilte Ressourcenauslastung beispielsweise des Queuespeichers der Ausgangsports von Bridges bestimmt werden, so dass es zu keinen Paketverlusten durch Speichermangel kommt und aus der Queuebelegung die Latenz, so dass die Nutzerqualität von Applikationen wie IIoT und KI-Monitoring- und Diagnose ausreichend ist. Der Anlagenbetreiber kann ohne das Risiko, in neue Kommunikations- und Netzwerkinfrastruktur zu investieren, diese bereits mittels der Simulation as a Service virtuell validieren, ob die Anforderungen an Latenz, Datenrate, Paketverluste, usw. erfüllt werden. Zudem lässt sich eine Überdimensionierung der Anlagen vermeiden und damit eine Kostenersparnis von zukünftigen, sehr kostenintensiven Netzwerkkomponenten erreichen. Neben dem in Kapitel 1 beschriebenen Konzepts wird die Komponente Simulation as a Service in Kapitel 2 spezifiziert, wie der NetDevOps Zyklus aus Abbildung 1 mit der Gitlab CI/CD Pipeline aus Abbildung 5 automatisiert wurde. Im letzten Kapitel wurde auf die Ergebnisse der Implementierung dieser Pipeline und die Ausgabedateien der Stage Simulate eingegangen. Hierzu wurde die Profinet Applikation aus Abbildung 7 herangezogen, die auf dem IEC/IEEE 60802 TSN-IA Profil basiert. Es konnte gezeigt werden, dass bei diesem Szenario mit der maximalen Anzahl an Profinet Teilnehmern innerhalb einer Domäne und einem IO Datenaustausch zwischen dem IOC und den IODs die zeitlichen Anforderungen der geplanten TSN Streams in der Simulation erfüllt werden, gleichzeitig aber auch die des zeitunkritischen Best Effort Verkehrs, der an verschiedenen IODs eingespeist wurde.

Mit der CI/CD Pipeline und der Simulation as a Service Komponente mit dem OMNEST/OMNeT++ Netzwerksimulator wurde eine Grundlage geschaffen, auf der nun weiteres Verhalten zur Netzwerkvalidierung untersucht werden kann wie beispielsweise die Analyse verschiedener Implementierungen eines Queue Buffer Managements und auch der unterschiedlichen Granularität der Speicherblöcke der Queuebuffer. Des Weiteren kann das Transpilieren in zukünftige Profinet TSN Konfigurationssyntax und automatisiertes Konfigurieren der realen Netzwerk Komponenten der digitalen Fabrik erweitert werden. Außerdem sind momentan in der OMNEST/OMNeT++ Simulation alle Netzwerk Komponenten mit einer idealen Clock synchronisiert. In Zukunft sollen Effekte durch nicht ideal zeitlich synchronisierte Komponenten und deren Grenzwerte analysiert werden.

5 Literaturverzeichnis

[AB00] https://gitlab.com/ipvs/nesting
[AB01] https://inet.omnetpp.org/Download.html
[AB02] https://omnest.com/
[AB03] https://www.profibus.com/download/profinet-specification

Authors:

Dominik Steinmann ist technischer Experte für die industrielle Kommunikation im Expert House der Siemens AG. Er absolvierte seinen B.Sc. und M.Sc. in Mechatronik an der Friedrich-Alexander-Universität Erlangen-Nürnberg von 2011 bis 2017.

Seine Forschungsschwerpunkte sind industrielle Kommunikationsnetzwerke mit dem Fokus auf QoS und Echtzeit Eigenschaften und Netzwerksimulation.

Dr. Stephan Höme ist Gruppenleiter in der Vorfeldentwicklungs Innovationsabteilung der Siemens AG und arbeitet an industriellen Kommunikationsthemen. Er absolvierte den Dipl.-Ing. und Ph.D. (Dr.-Ing.) am Institut für Automatisierungstechnik an der Universität Magdeburg von 2009 bis 2016.

Seine Forschungsthemen sind industrielle Kommunikationsaskepte (TSN, OPC UA, IPv6), Virtualisierung (Containerisierung, Netzwerkvirtualisierung) und Netzwerksimulation (Digitaler Zwilling).

Dr. Sven Kerschbaum ist Senior Key Expert für Kommunikationssysteme und Steuerungsnetzwerke der Siemens AG. Er absolvierte seinen M.Sc. (Dipl.-Inf. Univ.) und Ph.D. (Dr.-Ing.) am Department für Informatik an der Universität Erlangen von 2007 bis 2017.

Seine Forschungsschwerpunkte sind industrielle Kommunikationsnetzwerke und Ihre Anwendungen mit einem engen Fokus auf Dienstgütekriterien.

http://dx.doi.org/10.25673/39552

Vergleichende IT-Sicherheitsanalyse aktueller Datenplattformen

Sebastian Tebbje, Karl-Heinz Niemann, Björn Nickel,

Fakultät 1
Hochschule Hannover
Ricklinger Stadtweg 120, 30459 Hannover
Sebastian.Tebbje@Hs-Hannover.de
Karl-Heinz.Niemann@Hs-Hannover.de
Björn.Nickel@stud.Hs-Hannover.de

Abstract: Big-Data-Datenplattformen werden immer beliebter, um große Datenmengen bei Bedarf analysieren zu können. Zu den fünf gängigsten Big-Data-Verarbeitungsframeworks gehören Apache Hadoop, Apache Storm, Apache Samza, Apache Spark, und Apache Flink. Zwar unterstützen alle fünf Plattformen die Verarbeitung großer Datenmengen, doch unterscheiden sich diese Frameworks in ihren Anwendungsbereichen und der zugrunde liegenden Architektur. Eine Reihe von Studien hat sich bereits mit dem Vergleich dieser Big-Data-Frameworks befasst, indem sie sie anhand eines bestimmten Leistungsindikators bewertet haben. Die IT-Sicherheit dieser Frameworks wurde dabei jedoch nicht betrachtet. In diesem Beitrag werden zunächst allgemeine Anforderungen und Anforderungen an die IT-Sicherheit der Datenplattformen definiert. Anschließend werden die Datenplattform-Konzepte unter Berücksichtigung der aufgestellten Anforderungen analysiert und gegenübergestellt.

1 Einleitung

Seit dem Aufkommen der digitalen Transformation in fast allen Bereichen sind Daten zu einem der wichtigsten Aktivposten geworden, wenn es darum geht, Wettbewerbsvorteile für Unternehmen zu erlangen. Mit dem technologischen Fortschritt ist die Geschwindigkeit der Datengenerierung weltweit exponentiell angestiegen. Besonders in der Industrie fallen täglich riesige Mengen an heterogenen Daten und Informationen an, die die Verarbeitungskapazität herkömmlicher Datenbanksysteme übersteigen. Um einen Nutzen aus diesen Daten zu ziehen, muss ein alternativer Weg für ihre Verarbeitung gewählt werden [1]. In den letzten Jahren konnten sich Datenplattformen unter dem Begriffen Big Data und Cloud-Technologien am Markt etablieren, um diese Datenflut über verschiedene Unternehmensbereiche zugänglich zu machen. Mit dem Trend zu IoT und Industrie 4.0 müssen diese Daten zukünftig nicht nur innerhalb eines Unternehmens, sondern auch über dessen Unternehmensgrenzen hinweg innerhalb des Wertschöpfungsnetzwerks bereitgestellt werden können. Wertschöpfungsnetzwerke sind eine Organisationsform bestehend aus rechtlich selbständigen und wirtschaftlich operierenden Unternehmen, die über Geschäftsbeziehungen miteinander verbunden sind. Die kooperative Zusammenarbeit ermöglicht es diesen Unternehmen, ihre Planungs-, Produkt- und Prozessdaten durch alle Stufen der Wertschöpfungskette miteinander zu verknüpfen. Im Detail bedeutet dies, durch die technische Anbindung diverser Systeme, Komponenten oder Sensoren mittels passender Hard- und Software, ein gezieltes Sammeln tatsächlich notwendiger Daten zu ermöglichen. Weiterführend besteht die Möglichkeit diese Daten mit eingebundenen Wissensdatenbanken und unter Einbeziehung von KI auszuwerten. Danach werden diese Daten visualisiert und analysiert, in Benachrichtigungen und Aktionen umgesetzt, um gegebenenfalls direkt notwendige Prozesse anstoßen zu können. Beispielweise das Auslösen eines Wartungs- oder Serviceeinsatzes oder einer Teilelieferung. Hersteller, Anwender und weitere Unternehmenspartner können auf derselben Plattform zusammenarbeiten, Daten teilen und ggf. auch direkte Fernzugriffe auf Anlagendaten vornehmen. Die Daten, die von Maschinen im Verlauf der Produktion oder Wartung erzeugt werden, bergen großes Potenzial. Zum einen, um die Effizienz des Produktionsprozesses zu steigern, zum anderen, um die Daten mit Dritten zu tauschen, woraus im Idealfall neue Services oder Produkte entstehen können. Mit der zunehmenden Öffnung der Unternehmensnetzwerke wachsen, neben den Potentialen, auch die Herausforderungen und Risiken in Bezug auf die IT-Sicherheit und die rechtlichen Grundlagen, beispielsweise bei unrechtmäßiger Datennutzung. Die Schwachstellen in der OT-Sicherheit steigen jährlich exponentiell an [2]. Je umfangreicher die erfasste Datenmenge ist, desto höher ist die Wahrscheinlichkeit, dass auch sensible Informationen enthalten sind. Das Offenlegen vertraulicher Informationen könnte dem Unternehmen schaden, weshalb sich die Unternehmen darauf konzentrieren müssen, die Integrität und den Schutz der gesammelten Daten zu wahren. Der Schutz von Daten geht über die Beschränkung des Zugriffs auf bestimmte Ressourcen (Identitäts- und Zugriffskontrolle) hinaus: Es muss auch kontrolliert werden, wie die Daten nach dem Zugriff behandelt werden, was als Data Usage Control [3] bezeichnet wird. Data Usage Control bietet einen gemeinsamen und vertrauenswürdigen Sicherheitsrahmen, um die Einhaltung von Data-Governance-Regeln

und die verantwortungsvolle Nutzung von Unternehmensdaten durch Drittparteien zu gewährleisten und die sichere gemeinsame Nutzung von Daten in Ökosystemen wie Industrie 4.0 zu erleichtern und sicherzustellen. Dies wird über entsprechende Verträge zwischen den Unternehmen realisiert, was nicht Bestandteil dieses Beitrags ist. Aufgrund der Vielzahl von verfügbaren Datenplattformlösungen bedarf es einer Übersicht, die die Funktionsweisen der unterschiedlichen Ansätze gegenüberstellt. Weiterhin müssen die Plattformkandidaten klassifiziert und differenziert werden, da die Ansätze auf unterschiedliche Anwendungsfälle zielen. Des Weiteren sollen die IT-Sicherheitseigenschaften der verschiedenen Implementierungen anhand von zuvor definierten Anforderungen evaluiert werden. In diesem Beitrag werden daher die für Industrie 4.0 relevanten Datenplattformkandidaten bezüglich der allgemeinen- und IT-Sicherheitseigenschaften differenziert und vergleichend gegenübergestellt werden.

2 Forschungsstand

Es gibt bereits eine Reihe von Veröffentlichungen, in denen etablierte Frameworks und Tools zur Erstellung und zum Betrieb von Datenplattformen verglichen und evaluiert werden. Jedoch liegt das Hauptaugenmerk dieser Vergleiche entweder auf einer sehr geringen Auswahl von Frameworks oder es werden ausschließlich die funktionellen Anforderungen an eine Datenplattform berücksichtigt. So werden in [4] zwei Frameworks der Apache Foundation auf ihre Streaming-Performance untersucht. In [5] erfolgt die Untersuchung etwas allgemeiner, es werden vier Frameworks der Apache Foundation anhand von zuvor definierten „Key Performance Indicators“ (KPI) miteinander verglichen. Während [6] primär auf die Skalierbarkeit zweier vergleichbarer Frameworks abzielt. In [7] werden verschiedene Big-Data-Tools auf ihre „graph processing“-Fähigkeiten evaluiert. Außerdem liefert [8] einen Vergleich einer kommerziellen Hadoop Distribution in Kombination mit den Cloud-Service-Provider Amazon Web Services (AWS) und Microsoft Azure. Bei all diesen Veröffentlichungen fällt auf, dass die IT-Sicherheitsanforderungen an eine Datenplattform im Allgemeinen nicht berücksichtigt werden. Dieser Aspekt soll im Weiteren näher betrachtet werden.

3 Klassifizierung von Datenplattformen

Eine Datenplattform stellt eine Komplettlösung für die durchgängige Datenverarbeitung dar. Auf der Plattform, werden Daten aus verschiedenen Datenquellen erfasst, verarbeitet, kontrolliert und für Benutzer oder Datenanwendungen für weiterführende Analysen bereitgestellt. Datenplattformen werden in der Regel für die Verarbeitung sehr großer Datenmengen unterschiedlichen Ursprungs eingesetzt. Eine Datenplattform berücksichtigt möglicherweise jeden einzelnen Datensatz, den ein Unternehmen im Rahmen seiner Geschäftstätigkeit generiert. Dazu zählen sowohl IT-Daten aus dem Bürosektor, als auch Maschinendaten aus der Feldebene. In diesem Beitrag wird zwischen drei verschiedenen Klassen von Datenplattformen unterschieden. Die Klassen werden durch die Verarbeitungsweise (engl. processing) der Daten innerhalb der Plattformen definiert [9].

1) Stapelverarbeitung (batch-only processing):

Bei der Stapelverarbeitung werden Daten zunächst über einen bestimmten Zeitraum gespeichert und blockweise gruppiert. Anschließend werden die Datenblöcke gemeinsam verarbeitet.

2) Streamverarbeitung (stream-only processing):

Bei der Streamverarbeitung werden Daten kontinuierlich erfasst, verteilt und analysiert.

3) Hybridverarbeitung (hybrid processing):

Bei der Hybridverarbeitung kann sowohl Stream-, als auch Stapelverarbeitung betrieben werden.

Der beim Data Warehousing verwendete Schritt des Extrahierens, Transformierens und Ladens (ETL) ist in der Regel ein Batch-Prozess. Zu den wichtigsten Vorteilen der Stapelverarbeitung gehört die Möglichkeit, die Verarbeitung von Aufträgen auf andere Ressourcen mit größerer Kapazität zu verteilen, sowie Computerressourcen unter Programmen und Benutzern aufzuteilen. Während bei der Stapelverarbeitung Daten(-gruppen) verarbeitet werden, die über einen bestimmten Zeitraum gespeichert wurden, und die zu regelmäßig geplanten Zeiten oder nach Bedarf ausgeführt werden, kann der Benutzer bei der Stream-Verarbeitung Daten in Analysetools einspeisen, sobald sie generiert wurden. Dies ermöglicht eine Datenverarbeitung quasi in Echtzeit und sofortige Analyseergebnisse. Zu den Vorteilen gehört die sofortige Erkennung von Bedingungen und Anomalien innerhalb eines sehr kurzen Zeitraums. Die Streamverarbeitung

ist demnach bei Anwendungsfällen, die sehr geringe Verarbeitungslatenzen erfordern, von Vorteil, während die Stapelverarbeitung auf Anwendungsfälle mit sehr großen Datenmengen zielt.

4 Anforderungen an die Datenplattformen

Bei der Wahl eines Frameworks für eine Datenplattform im Kontext von Industrie 4.0 spielen diverse Faktoren eine Rolle. In diesem Abschnitt werden einige grundlegende Anforderungen an ein Framework definiert. In Abschnitt 4.1 werden generelle und in 4.2 sicherheitsspezifische Anforderungen beschrieben.

4.1 Generelle Anforderungen an eine Datenplattform

1) Betriebsmodell:

Beim Betriebsmodell wird zwischen On- und Off-Premise unterschieden. On-Premise stellt den klassischen Weg zur Realisierung einer Datenplattform dar. Dabei erfolgt die Installation auf vorhandener Hardware lokal innerhalb eines Unternehmens. Während Off-Premise die Bereitstellung einer Datenplattform an anderer Stelle, z. B. bei einem Cloud-Anbieter, bedeutet. Möglich ist auch ein hybrider Betrieb der Datenplattform, bei dem vorhandenen unternehmensinterne Ressourcen mit einer oder mehreren Clouds kombiniert werden können. Abhängig vom Betriebsmodell ergeben sich für entsprechende Anwendungsfälle verschiedene Vor- und Nachteile für den Betreiber der Datenplattform. Zu den wichtigsten Fragestellungen dabei gehören die Kosten-Nutzen-Effizienz und die Datenhoheit [10]. Um allen Anwendungsfällen gerecht zu werden, sollte eine Datenplattform in allen Betriebsmodellen betrieben werden können, oder über entsprechende Verbindungsmöglichkeiten verfügen.

2) Skalierbarkeit:

Skalierbarkeit ist die Fähigkeit eines Systems, auf eine zunehmende Last zu reagieren. Es gibt zwei Arten: Skalierung nach oben (vertikal) und Skalierung nach außen (horizontal). Bei der vertikalen Skalierung wird die Hardwarekonfiguration in Bezug auf ihre Leistungsfähigkeit aufgerüstet, während bei der horizontalen Skalierung zusätzliche Hardware-Einheiten hinzugefügt wird. Eine Datenplattform muss in der Lage sein, die Volumenanforderungen des jeweiligen Anwendungsfalls zu erfüllen und sich an das Wachstum der zu bearbeitenden Datenmenge anzupassen.

3) Maschinelles Lernen und Künstliche Intelligenz

Technologische Innovationen, insbesondere in den Bereichen Machine Learning (ML) und künstliche Intelligenz (KI), haben neue Möglichkeiten für Unternehmen jeder Größe geschaffen, von datengestützten Erkenntnissen zu profitieren. Die profitable Nutzung der anfallenden Daten liefert einen wesentlichen Grund für die Erstellung einer Datenplattform, daher muss die Datenplattform über entsprechende Integrationsmöglichkeiten für skalierbare Software-Bibliotheken verfügen, um Algorithmen des maschinellen Lernens ausführen zu können.

4) Echtzeitfähigkeit:

Echtzeitfähigkeit ist für Anwendungen mit zeitkritischen Übertragungsfenstern relevant. In diesem Beitrag wird daher zwischen harter, weicher und keiner Echtzeitfähigkeit unterschieden. Harte Echtzeitfähigkeit bedeutet, dass Deadlines immer eingehalten werden. Bei weicher Echtzeitfähigkeit liegt die durchschnittliche Übertagungszeit innerhalb der Deadline aber vereinzelte Überschreitungen sind tolerabel. Bei keiner Echtzeitfähigkeit sind keinerlei Garantien für die rechtzeitige Verarbeitung der Information gegeben.

5) Resilienz:

Resilienz bezeichnet die Fähigkeit von IT-Systemen, hinsichtlich Störungen und Problemen wie Ausfällen einzelner Komponenten robust zu reagieren und die Anwender weiterhin mit den benötigten Services zu versorgen. Typische Maßnahmen zur Sicherstellung der Resilienz sind redundante, verteilte Systeme und Datensicherungen. Eine Datenplattform bildet die Grundlage für die bereitgestellten Dienste und sollte entsprechend resilient sein. Gleichzeitig wird mit dieser Anforderung auch die Verfügbarkeit adressiert, welches eins der primären Schutzziele der IT-Sicherheit darstellt.

6) Verfügbare Implementierung:

Die Verfügbarkeit von Open-Source und/oder kommerzieller Implementierungen ist eine grundsätzliche Voraussetzung bei der Auswahl geeigneter Lösungen, um den Betreibern anwendungsfallspezifische Konfigurations- und Gestaltungsspielraum zu ermöglichen.

7) Nachrichtenzustellungsgarantie

Garantien für die Zustellung von Nachrichten werden im Falle eines Fehlers benötigt. Die Datensenke bestätigt der Datenquelle den Empfang der Nachricht. Es lassen sich zwei Arten von Garantien unterscheiden: genau einmalige Zustellung und mindestens einmalige Zustellung. Exakte einmalige Zustellung bedeutet, dass die Nachricht weder dupliziert wird noch verloren geht und dem Empfänger genau einmal zugestellt wird. Mindestens eine Zustellung bedeutet, dass mehrere Versuche unternommen werden, die Nachricht zuzustellen, und dass mindestens einer dieser Versuche erfolgreich war. Darüber hinaus kann die Nachricht dupliziert werden, ohne dass sie verloren geht.

8) Datenspeicher und Berechnungsmodus (engl. computation mode)

Der Berechnungsmodus der Datenbank kann in-memory oder disk-based sein. Eine In-Memory-Datenbank speichert alle Daten im Hauptspeicher (RAM) eines Computers. Eine herkömmliche Datenbank ruft die Daten von Festplattenlaufwerken ab. In-Memory-Computing ist schneller, hat aber den Nachteil, dass der Inhalt verloren geht, wenn der Rechner ausfällt. Hierbei handelt es sich um eine nicht funktionale Anforderung.

4.2 Anforderungen an die IT-Sicherheit der Datenplattform

Das grundsätzliche Ziel der IT-Sicherheit ist es, einen Schutz aller Informationen und Prozesse in einem informationstechnischen System zu gewährleisten. Die allgemeinen übergeordneten Schutzziele der IT-Sicherheit sind: Verfügbarkeit, Integrität, Vertraulichkeit, Authentizität und Nichtabstreitbarkeit [11]. In [12] wird eine Übersicht über Normen und Standards zur IT-Sicherheit gegeben und auf dieser Grundlage verschiedene Anforderungen an die IT-Sicherheit definiert. Während unter 4.1 bereits Anforderungen bezüglich der Verfügbarkeit (Resilienz, Nachrichtenzustellungsgarantie) erläutert wurden, werden im Folgenden weitere technische Anforderungen an die IT-Sicherheit einer Datenplattform unter Berücksichtigung der IT-Schutzziele vorgestellt.

1) Authentifizierung

Die Authentizität gewährleistet, dass ein Kommunikationspartner tatsächlich derjenige ist, der er vorgibt zu sein. Besonders bei verteilten Systemen stellt dies eine Herausforderung dar. Eine Datenplattform muss die Fähigkeit haben, alle Nutzer im Netzwerk eindeutig identifizieren und authentifizieren zu können.

2) Autorisierung

Neben der Authentifizierung eines Nutzers, muss es auch eine Möglichkeit zur Autorisierung geben. Eine restriktive Berechtigungsvergabe kann beispielsweise durch Verfahren wie Role Based Access Control (RBAC) oder Discretionary Access Control (DAC) erfolgen.

3) Schutz der Daten vor unautorisiertem Zugriff

Daten können sowohl bei der Übertragung (data-in-transit) als auch im Ruhezustand (data-at-rest) Risiken ausgesetzt sein und müssen in beiden Zuständen geschützt werden. Es gibt verschiedene Verschlüsselungsmethoden um sensible Daten im Sinne der Vertraulichkeit zu schützen. Dabei werden die Daten vor der Übertragung verschlüsselt oder es werden verschlüsselte Verbindungen (wie z.B. HTTPS, FTPS usw.) verwendet. Für den Schutz von Daten im Ruhezustand können die Daten vor der Speicherung oder das Speicherlaufwerk verschlüsselt werden.

4) Audit

Um die Nichtabstreitbarkeit zu gewährleisten, muss die Datenplattform über entsprechende Protokollierungsfunktionen verfügen. Es müssen alle Aktivitäten innerhalb der Datenplattform zu jeder Zeit nachvollziehbar sein. Mit Hilfe von Audits kann sichergestellt werden, dass keine audit-relevanten Ereignisse aufgetreten sind.

5 Übersicht der Datenplattformen

Nachfolgend werden fünf Datenplattformen im Hinblick auf die generellen Anforderungen aus Abschnitt 4.1 beschrieben. Die Auswahl beruht auf den zur Zeit am häufigsten genutzten Frameworks [13]. Alle genannten Kandidaten sind über freie Open-Source-Implementierungen verfügbar und horizontal skalierbar. Ursprünglich für den On-Premise-Betrieb entworfen, sind alle Frameworks auch in der Cloud betreibbar und es bestehen entsprechende Migrationsmöglichkeiten. Die Kandidaten sind den in Abschnitt 3 definierten Klassen zugeordnet.

A) Stapelverarbeitung (batch-only processing):

A.1) Apache Hadoop [14]

Apache Hadoop wurde 2008 als Batch-Processing-Framework definiert, das verteilte Daten über eine Gruppe von Host-Rechnern, die Cluster oder Knoten genannt werden, sammelt und verarbeitet. Hadoop besteht aus drei Kernkomponenten. Die Komponente zum Sammeln von Daten nennt sich Hadoop Distributed File System (HDFS). Die Speicherung der Daten erfolgt dabei auf der Festplatte. HDFS erstellt Replikate den Hosts innerhalb eines Clusters und ist dadurch fehlertolerant. Wenn also ein Rechner im Cluster ausfällt, kann auf die Daten von anderen Rechnern zugegriffen werden, auf denen die gleiche Kopie der Daten erstellt wurde. Die zweite Komponente ist YARN, welche arithmetische Ressourcen für die Auftragsplanung wie CPU und Speicher bereitstellt. Die dritte Komponente ist MapReduce, welches für die Verarbeitung von Daten (Softwareschicht) mit anderen Prozessen verwendet wird. Über die Jahre hat sich ein komplettes Ökosystem mit zahlreichen Erweiterungsmöglichkeiten um Hadoop herum gebildet, darunter auch Apache Mahout. Mahout ist ein Framework zur Erstellung skalierbarer, leistungsstarker Anwendungen für maschinelles Lernen.

B) Streamverarbeitung (stream-only processing):

B.1) Apache Storm [15]

Apache Storm ist ein Framework für die Verarbeitung großer strukturierter und unstrukturierter Daten in Echtzeit. Es ist ausschließlich auf die Stream-Verarbeitung spezialisiert. Storm besitzt zwei Schnittstellen: die normale Storm API und Storm Trident. Erstere unterstützt keine höheren Abstraktionen, sondern lehnt seine API an das reine Stream Processing Model an. Die API bearbeitet jedes Tupel einzeln und nutzt die At-Least-Once-Semantik. Storm Trident bietet eine höhere Abstraktionsebene an und arbeitet mit Micro-Batches. Es stellt eine Exactly-Once-Semantik bereit. Die Streamverarbeitung wird direkt auf den Hosts durchgeführt. Storm überwacht dabei die Datenverarbeitung und wiederholt die Verarbeitung bei Fehlern. Die Verwaltung des Clusters und der Ressourcen erfolgen durch Nimbus. Storm kann im Standalone-Modus oder auf YARN betrieben werden. Anwendungen werden über den Nimbus bereitgestellt und per Thrift, einem Datenaustauschformat, zu den Hosts übertragen. Storm verfügt über eine SAMOA API. Apache SAMOA bietet eine Sammlung verteilter Streaming-Algorithmen für die gängigsten Data-Mining- und Machine-Learning-Aufgaben wie Klassifizierung, Clustering und Regression sowie Programmierabstraktionen zur Entwicklung neuer Algorithmen, die auf verteilten Stream-Processing-Engines (DSPEs) laufen.

B.1) Apache Samza [16]

Apache Samza wurde ursprünglich von LinkedIn entwickelt, um verschiedene Arten von Stream-Processing-Anforderungen zu erfüllen, wie z. B. die Verfolgung von Daten, die Protokollierung von Daten durch Dienste und Dateneingabepipelines für Echtzeitdienste. Samza ist wie Storm ausschließlich auf die Stream-Verarbeitung ausgelegt. Im Gegensatz zu den anderen Frameworks hat es kein eigenes Transport- und Verarbeitungssystem. Es nutzt Apache Kafka als Transportsystem und Apache Hadoop YARN als Verarbeitungssystem. Es verwendet eine In-Memory-Verarbeitungstechnik und bietet die Samza API zur Anwendungsentwicklung und die SAMOA API. Kafka sorgt zusammen mit ZooKeeper für die nötige Fehlertoleranz, indem der Verarbeitungsstand dort gesichert wird. Außerdem wird garantiert, dass die Daten immer mindestens einmal gesendet werden.

C) Hybridverarbeitung (hybrid processing):

C.1) Apache Spark [17]

Apache Spark ging aus einem Forschungsprojekt der University of California, Berkeley hervor. Es war ursprünglich als Ersatz für das Apache Hadoop MapReduce Framework konzipiert. Spark nutzt eine Abstraktion namens Resilient Distributed Dataset (RDD). RDDs sind schreibgeschützte Daten-Partitionen, die auf den Rechenknoten vorliegen. Sie können im Fehlerfall einfach aus den Ausgangsdaten wiederhergestellt werden. Dieses Batch-orientierte Konzept wurde später für Streaming-Daten angepasst. Es nutzt kein natives Streaming, sondern Micro-Batching, das heißt, es sammelt eingehende Daten für einen gewissen Zeitraum und

verarbeitet diese in bestimmten Zeitabständen. Bei jedem Verarbeitungsschritt werden die Operatoren neu auf die Hosts verteilt. Dadurch wird weiche Echtzeit ermöglicht. Es stehen eine umfangreiches API und einige Bibliotheken bereit, darunter eine Machine-Learning-Bibliothek (MLlib). Außerdem kann, genau wie Hadoop, Apache Mahout genutzt werden. Spark bietet mehrere Schnittstellen zum Erstellen von Anwendungen an. Für die Streamverarbeitung ist nur die DStream API relevant. Beim Datentransport werden Checkpoints genutzt, um Fehlertoleranz herzustellen. Als Ablage für Checkpoints wird das verteilte Dateisystem HDFS genutzt. Ein Master verwaltet die Ressourcen und ist beim Starten der Anwendung beteiligt. Mithilfe von ZooKeeper können redundante Master eingesetzt werden, um eine höhere Ausfallsicherheit zu erreichen. Die Clusterverwaltung kann entweder direkt durch Spark im Standalone-Betrieb erfolgen oder durch ein externes Werkzeug wie Hadoop YARN.

C.2) Apache Flink [18]

Apache Flink ist ein Open-Source-Framework, das 2010 entwickelt wurde und sich für die Datenverarbeitung sowohl im Echtzeit- als auch im Batch-Modus bewährt hat. Es verwendet eine In-Memory-Verarbeitungstechnik und bietet eine Reihe von APIs wie Stream Processing API (Datenstrom), Batch-Processing API (Datensatz) und Table API, die für Abfragen verwendet werden. Es verfügt auch über Bibliotheken für maschinelles Lernen (FlinkML) und Graphverarbeitung (Gelly). Flink führt Fehlertoleranzmaßnahmen wie periodisches Sichern von Zwischenständen (Checkpointing) durch. Außerdem wird garantiert, dass die Daten genau einmal gesendet werden. Für die Clusterverwaltung nutzt Flink entweder den JobManager im Standalone-Betrieb, also ohne ein externes Werkzeug, oder beispielsweise Apache Hadoop YARN. Das verteilte Dateisystem HDFS dient der sicheren Ablage von Checkpoints. Der lokale Betriebsmodus erleichtert das Debuggen, da alle Komponenten lokal ausgeführt werden. Zum Starten, Beenden und Verwalten der Anwendungen kann eine CLI oder eine Web-GUI verwendet werden.

6 Sicherheitsanalyse der betrachteten Datenplattformen

1) Authentifizierung

Während Hadoop, Storm, Samza und Flink über keine integrierte Authentifizierungsmechanismen verfügen, bietet Spark die Authentifizierung für Remote Procedure Calls (RPC) und Web UI´s (Servlet Filter) an. Trotzdem ist eine Authentifizierung der Netzwerkteilnehmer in allen Frameworks möglich. Da alle Frameworks über TCP/IP kommunizieren, kann auf der Netzwerkschicht über Transport Layer Security (TLS) eine Authentifizierung erfolgen. Außerdem sind alle Frameworks mit Kerberos kompatibel. Kerberos ist ein TCP/IP-basiertes Authentifizierungssystem das speziell für Authentifizierung auf der Darstellungsschicht in einer verteilten Umgebung konzipiert ist. Durch entsprechende Erweiterungen der Frameworks können weitere Authentifizerungsmöglichkeiten genutzt werden: Für Storm gibt es Plugins für Apache Thrift und SASL, bei Samza kann die Authentifizierung über Kafka erfolgen und Kubernetes liefert Authentifikationsoptionen bei der Verwendung mit Spark. Flink kann theoretisch Erst- oder Drittanbieter-Konnektoren (Kafka, HDFS, Cassandra, Flume, Kinesis usw.) verwenden, die beliebige Authentifizierungsmethoden (Kerberos, SSL/TLS, Benutzername/Passwort usw.) erfordern. Praktisch wird jedoch nur Kerberos für die Authentifizierung sicher unterstützt.

2) Autorisierung

Flink bietet derzeit keinerlei Autorisierungsmöglichkeiten. Hadoop, Samza und Storm nutzen YARN als Ressourcenmanager, welcher umfangreiche Access Control Lists (ACLs) zur Autorisierung unterstützt. ACLs können entweder für Benutzer oder für Gruppen konfiguriert werden. Für Storm kann außerdem das Autorisierungs-Plugin SimpleACLAuthorizer genutzt werden. Die Berechtigungen können dabei an den Benutzernamen oder an das Kerberos-Ticket des Nutzers gebunden werden. Auf diese Weise kann jeder Benutzer mit gültigem Kerberos-Ticket Operationen durchführen, wie z.B. aktivieren, deaktivieren oder auf Clusterinformationen zugreifen. Spark unterstützt die Zugriffskontrolle auf die Web UI, sofern ein Servlet Filter vorhanden ist. Dadurch kann jede Anwendung mit ihren eigenen separaten ACLs konfiguriert werden. Spark unterscheidet dabei jedoch ausschließlich zwischen "Anzeige"-Berechtigungen und "Änderungs"-Berechtigungen. Sofern Spark YARN als Ressourcenmanager nutzt, kann auch darüber die Berechtigungsvergabe zurückgegriffen werden. Bei Hadoop und Storm gibt es die Möglichkeit über Apache Ranger oder Apache Sentry auch rollenbasierte Zugriffsberechtigungen (RBAC) zu erteilen. Sowohl Ranger als auch Sentry bilden ein Framework für die Definition von Richtlinien zur Kontrolle des Zugriffs auf Dateien, Ordner, Datenbanken, Tabellen oder Spalten.

3) Schutz der Daten vor unautorisiertem Zugriff

Da alle Frameworks die Möglichkeit bieten TLS zu nutzen, kann auf der Netzwerkschicht ebenfalls die Ende-zu-Ende Verschlüsselung bei der Datenübertragung durch TLS erfolgen. Oberhalb der Transportschicht implementiert Hadoop´s HDFS eine transparente Ende-zu-Ende-Verschlüsselung. Einmal konfiguriert, werden Daten, die aus speziellen HDFS-Verzeichnissen gelesen und in diese geschrieben werden, transparent ver- und entschlüsselt, ohne dass Änderungen am Code der Benutzeranwendung erforderlich sind. Diese Verschlüsselung erfolgt ebenfalls Ende-zu-Ende, was bedeutet, dass die Daten nur vom Client ver- und entschlüsselt werden können. Außerdem speichert HDFS niemals unverschlüsselte Daten. Apache Storm und Samza verfügen über keine eigenen Verschlüsselungsmethoden. Apache Spark bietet nur die Verschlüsselung von Steuerungsverbindungen (RPC). Zusätzlich unterstützt Spark die Verschlüsselung von temporären Daten, die auf lokale Festplatten geschrieben werden. Für Apache Flink bietet sich ausschließlich TLS an, da es keine eigenen Verschlüsselungsmethoden besitzt.

4) Auditing

Hadoop, Storm und Spark nutzen Apache Log4j als Logging-Framework. Darüber lassen sich sowohl Benutzeraktivitäten als auch Dienstaktivitäten protokollieren. Samza und Flink verwenden die SLF4J-Logging-Schnittstelle. Dadurch kann jedes Logging-Framework, das SLF4J unterstützt, verwendet werden. Standardmäßig wird Log4j 2 als das zugrundeliegende Logging-Framework verwendet. Auf die Log-Dateien kann über die Job-/TaskManager-Seiten der WebUI zugegriffen werden. Der verwendete Resource Provider (z.B. YARN) kann zusätzliche Zugriffsmöglichkeiten bieten.

7 Zusammenfassung und Fazit

In den folgenden Tabellen 1 und 2 sind die Ergebnisse der Evaluierung anhand der aufgestellten Anforderungen zusammenfassend dargestellt.

	Hadoop	**Storm**	**Samza**	**Spark**	**Flink**
Datenverarbeitungsweise	Batch	Stream	Stream	Hybrid	Hybrid
Betriebsmodell	On-/Off-Prem	On-/Off-Prem	On-/Off-Prem	On-/Off-Prem	On-/Off-Prem
Skalierbarkeit	Horizontal	Horizontal	Horizontal	Horizontal	Horizontal
Machine Learning	Mahout	SAMOA API	SAMOA API	SparkMLlib, Mahout	FlinkML
Echtzeit	✗	✓	✓	Weiche Echtzeit	✓
Resilienz	✓	✓	✓	✓	✓
Open Source	✓	✓	✓	✓	✓
Nachrichtenzustellungsgarantie	Exakt eine	Mindestens eine, exakt eine (Trident API)	Mindestens eine	Exakt eine	Exakt eine
Datenspeicher	Disk-based	In memory	In memory	In memory	In memory

Tabelle 1: Übersicht der evaluierten Datenplattformkandidaten

	Hadoop	Storm	Samza	Spark	Flink
Authentifizierungs-Möglichkeiten	Kerberos, TLS	Kerberos, Thrift, SASL, TLS	Kerberos (YARN), SSL+SASL (Kafka), TLS	Kerberos (YARN), TLS, Servlet Filter (WEB UI +RPC)	Kerberos, TLS
Autorisierungs-Möglichkeiten	ACL, RBAC (Sentry, Ranger)	ACL, SimpleACLAuthorizer RBAC (Ranger)	ACL (YARN)	Servlet Filter, ACL (YARN)	✗
Verschlüsselung (in-transit)	HDFS, TLS	(HDFS), TLS	TLS	AES (RPC), TLS	(HDFS), TLS
Verschlüsselung (at-rest)	✓	✗	✗	✓	✗
Auditing	✓	✓	✓	✓	✓

Tabelle 2: Übersicht der IT-Sicherheitsfeatures

In diesem Beitrag wurden die Datenplattformkandidaten Hadoop, Storm, Samza, Spark und Flink anhand verschiedener Anforderungen evaluiert. Die Anforderungen adressieren die allgemeine Funktionalität einer Datenplattform und die vorhanden IT-Sicherheitsfeatures für den sicheren Betrieb. Die Definitionen der Anforderungen beruhen auf kontextspezifische Standards und Normen. Abschließend wurden die Ergebnisse für die einzelnen Kriterien ausgeführt und in Tabellen zusammengefasst dargestellt.

Tabelle 1 zeigt, dass alle Plattformkandidaten die funktionalen Anforderungen weitestgehend erfüllen. Ein Unterschied besteht im Wesentlichen auf der Verarbeitung in Echtzeit, was wiederum vom übergeordneten Verarbeitungsmodus abhängt. Jede dieser Datenplattformen bietet demnach verschiedene Vor- und Nachteile, sodass eine konkrete Auswahl von dem jeweiligen Anwendungsfall abhängig gemacht werden muss. Um möglichst viele Anwendungsfälle abdecken zu können, empfiehlt sich eine Kombination verschiedener Frameworks. In Tabelle 2 werden die Ergebnisse der IT-Sicherheitsanalyse dargestellt. Wobei die Analyse auf den jeweiligen Dokumentationen beruht. Anhand der Spezifikationen erfüllt keine der Datenplattformen alle Anforderungen. Beispielsweise muss bei der Authentifizierung Kerberos oder TLS genutzt werden. Als Autorisierungsmöglichkeiten können lediglich ACLs und Filter genutzt werden, was eine feingranulare restriktive Berechtigungsvergabe im Kontext von Industrie 4.0 erschwert. Hadoop und Strom bieten die Möglichkeit durch Apache Ranger RBAC zu ermöglichen. Daher ist eine finale Auswahl für eine präferierte Plattform nicht möglich. Vielmehr muss man die einzelnen Features bewerten und dann an Hand der Detail-Information eine Auswahl treffen.

8 Literaturverzeichnis

[1] Dumbill, E. 2013. Making Sense of Big Data. *Big data* 1, vol. 1 no. 1, 1–2.

[2] Claroty. 2021. *Claroty Biannual ICS Risk & Vulnerability*. *REPORT: 1H 2021*. https://security.claroty.com/1H-vulnerability-report-2021. Accessed 20 September 2021.

[3] Kelbert, F. and Pretschner, A. 2018. *Data Usage Control for Distributed Systems*.

[4] Samadi, Y., Zbakh, M., and Tadonki, C. 2018. Performance comparison between Hadoop and Spark frameworks using HiBench benchmarks. *Concurrency Computat Pract Exper* 30, 12.

[5] Safaa Alkatheri, Samah Abbas, and Muazzam Ahmed Siddiqui. 2019. A Comparative Study of Big Data Frameworks. *International Journal of Computer Science and Information Security (IJCSIS)* 17, 1.

[6] García-Gil, D., Ramírez-Gallego, S., García, S., and Herrera, F. 2017. A comparison on scalability for batch big data processing on Apache Spark and Apache Flink. *Big Data Anal* 2, 1.

[7] Kaepke, M. and Zukunft, O. A Comparative Evaluation of Big Data Frameworks for Graph Processing, 30–37.
[8] Ahmed Kawser and Kawser Ahmed Pinto. 2020. A Comparative study one of the Hadoop distribution Hortonworks with Amazon Web Service (AWS) and Microsoft Azure (2020).
[9] Bhupender singh thakur and Dr. Kishori Lal Bansal. 2020. Comparative Analysis and Evaluation of Big Data Processing Frameworks and Tools. *Mukt Shabd Journal,* 9, 1338–1348.
[10] Bundesministerium für Wirtschaft und Energie (BMWi). 2019. Sichere unternehmensübergreifende Kommunikation mit OPC UA. Diskussionspapier.
[11] Bundesamt für Sicherheit in der Informationstechnik. 2016. *IT-Grundschutz-Katalog*. https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzDownloads/Download_Archiv/download_node.html.
[12] Tebbje, S. and Niemann, K.-H. 2020. IT-Security-Anforderungen bei firmenübergreifenden Wertschöpfungsketten im Kontext von Industrie 4.0. *Anwendungsorientierte Forschung für die digitale Transformation von KMU - Schriften des Forschungsclusters Industrie 4.0*, 79–93.
[13] Inoubli, W., Aridhi, S., Mezni, H., Maddouri, M., and Mephu Nguifo, E. 2018. An experimental survey on big data frameworks. *Future Generation Computer Systems* 86, 546–564.
[14] Apache Software Foundation. 2005. *Apache Hadoop. Open-source software for reliable, scalabe and distributed computing*. https://hadoop.apache.org/.
[15] Apache Software Foundation. 2011. *Apache Storm. Distributed and fault-tolerant realtime computation*. https://storm.apache.org/.
[16] Apache Software Foundation. 2018. *Apache Samza. A distributed stream processing framework*. http://samza.apache.org/.
[17] Apache Software Foundation. 2014. *Apache Spark. Lightning-fast unified analytics engine*. https://spark.apache.org/.
[18] Apache Software Foundation. 2015. *Apache Flink. Stateful Computations over Data Streams*. https://flink.apache.org/.

Gefördert vom Niedersächsischen Ministerium für Wissenschaft und Kultur unter Fördernummer ZN3489 im Niedersächsischen Vorab der Volkswagen Stiftung und betreut vom Zentrum für digitale Innovationen (ZDIN).

http://dx.doi.org/10.25673/39553

Beurteilung des Störpotenzials für industrielle Funkkommunikation am Beispiel von PROFINET-Kommunikation über Bluetooth 5

Gustavo Cainelli, Lisa Underberg, Lutz Rauchhaupt

Institut für Automation und Kommunikation e.V. Magdeburg

Werner-Heisenberg-Str. 1

39106 Magdeburg

{gustavo.cainelli, lisa.underberg, lutz.rauchhaupt }@ifak.eu

Abstract: Trotz der Entwicklung zuverlässiger Funkkommunikationslösungen für industrielle Automatisierungsanwendungen sind Befürchtungen weit verbreitet, dass die Funkübertragung ungewollt oder gezielt gestört werden kann. Entsprechende Diskussionen bei der Überarbeitung der VDI/VDE-Richtlinie 2185 "Funkgestützte Kommunikation in der Automatisierungstechnik" im VDI/VDE-GMA Fachausschusses 5.21 "Funkgestützte Kommunikation" wurden zum Anlass genommen, das Störpotenzial bei industrieller Funkkommunikation näher zu untersuchen.

Für die erste Testreihe wurde eine Funktechnologie ausgewählt, die aufgrund des Mediumzugriffsmechanismus robust gegen passive Umgebungseinflüsse ist, wie sie im industriellen Umfeld anzutreffen sind: Bluetooth. Untersucht wurden Funkgeräte mit Bluetooth 5, die in der Lage sind, PROFINET-I/O-Kommunikation zu übertragen. Dabei wurden die Entfernung zwischen den Funkgeräten, die Entfernung zwischen Funkgerät und Störer, die Richtung der logischen Verbindung sowie die Störleistung variiert. In dem Beitrag werden die Ergebnisse der Untersuchungen vorgestellt und diskutiert.

Die vorgestellte Testreihe soll als Blaupause für weitere Testreihen zur Untersuchung von Kommunikationssystemen mit WiFi 6 und 5G verwendet werden. Die Untersuchungen sollen dazu beitragen, die Ungewissheit über das Störpotential auf industrielle Funkkommunikation zu verringern. Darüber hinaus sind die Testreihen ein Mittel, um Eigenschaften künftiger Funkkommunikationssysteme wie Elastizität und Resilienz zu erforschen.

1 Einleitung

Mit steigendem Interesse an Funkkommunikationssystemen für industrielle Anwendungen wird das fehlende Vertrauen der Endanwender immer deutlicher. Insbesondere, wenn für den Produktionsprozess kritische Aufgaben ausgeführt werden sollen, sind Endanwender zurückhaltend, eine neue Kommunikationstechnologie einzuführen. Gleichzeitig ist ihnen das Potential eines Funkkommunikationssystems durch dessen Nachrüstbarkeit und Ermöglichung von fahrerlosen Transportsystemen (FTS, engl: automated guided vehicle, AGV) bewusst. Der Test des Zeit- und Fehlerverhaltens aus Anwendungssicht in einer virtuell oder physisch emulierten Umgebung ist in diesem Kontext die Grundlage, das nötige Vertrauen aufzubauen, bevor ein Kommunikationssystem in der operativen Produktion eingesetzt wird.

Die Untersuchung der Leistungsfähigkeit eines Kommunikationssystems aus der Sicht einer Anwendung ist dabei ein Schlüssel, da diese Perspektive die Anforderungen des Endanwenders spiegelt [1]. Aus Sicht einer industriellen Anwendung ist das Kommunikationssystem, egal ob kabellos, kabelgebunden oder hybrid, transparent, solange es ihren Anforderungen genügt. Insbesondere hybride Systeme sind in industriellen Anwendungen relevant, da die Kombination aus bestehenden kabelgebundenen Systemen mit neuen Funksystemen naheliegt [2]. Entsprechend sollte ein Testsystem, das aus Applikationssicht die Performanz ermittelt, auf jedes Kommunikationssystem anwendbar sein.

Die daraus resultierende Trennung von Testsystem und zu testendem System (engl.: system under test, SUT) ist in Abbildung 1 dargestellt. Das Testsystem legt die Rahmenbedingungen für das SUT fest, indem es Einflussgrößen wie Nutzdatenhäufigkeit, Sendeverhalten und Umgebungseigenschaften festlegt. Das Verhalten des SUT erfasst das Testsystem, indem es für die Ermittlung von Kenngrößen wie der Übertragungszeit, der Paketverlustquote und dem Nutzdatendurchsatz relevante Daten misst. Basierend auf diesen Daten führt das Testsystem eine statistische Auswertung durch.

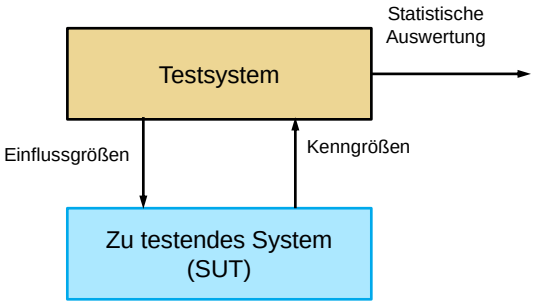

Abbildung 1: Prinzip des Tests von Zeit- und Fehlerverhalten eines zu testenden Systems (engl.: System under test, SUT) mit einem davon unabhängigen Testsystem [2].

Dieses Prinzip des Testens erlaubt die reproduzierbare Durchführung von Tests, sodass beispielsweise unterschiedliche Funkkommunikationssysteme unter gleichen Bedingungen analysiert werden können. Gleichzeitig ist die gezielte Hinzunahme spezifischer Störer, beispielsweise durch Funkgeräte oder einen Signalgenerator möglich.

Dieser Beitrag fasst Grundlagen zu Bluetooth und PROFINET in Abschnitt 2 zusammen. Ebenso wird ein Überblick über Untersuchungen des Zeitverhaltens von Bluetooth-Lösungen gegeben in der Literatur gegeben. Abschnitt 3 beschreibt detailliert den verwendeten Ansatz von Performanztests aus Anwendungssicht und den Testaufbau. Die Ergebnisse werden in Abschnitt 4 analysiert. Abschnitt 5 fasst diesen Beitrag zusammen und gibt einen Ausblick.

2 Technologieüberblick

2.1 Mediumzugriff bei Bluetooth Low Energy

Bluetooth Low Energy (BLE) verwendet ein Polling-Verfahren durch den BLE-Master, bei dem dieser den Mediumzugriff aller BLE Geräte koordiniert, indem er die Zeit in Abschnitte, sogenannte Connection Events (CEs) teilt [3]. Ein CE beginnt mit der Übertragung eines Datenpakets vom Master an eines der BLE-Geräte. Während des CE tauschen Master und Gerät abwechselnd Datenpakete aus, bis alle Nutzdaten übertragen wurden. Anschließend wechselt das BLE-Gerät in einen Standby-Modus, während der Master ein CE mit dem nächsten Gerät initiiert. Die CEs werden in regelmäßigen Abständen generiert und überlappen sich nicht.

2.2 Mediumzugriff bei PROFINET

PROFINET ist in der Lage, zyklisch zeitkritische Nutzdaten zu übertragen und dabei Anforderungen vieler industrieller Applikationen einzuhalten. Der PROFINET-Controller konfiguriert die PROFINET-Geräte und steuert den Datenaustausch. Parameter des Controllers wie z. B. die Zykluszeit können über ein Engineering-Werkzeug vorgegeben werden [4].

2.3 Performanztests von Bluetooth in der Literatur

Dieser Abschnitt gibt einen Überblick über Performanztests von Bluetooth-Systemen, da diese in einem hybriden Netz mit PROFINET den Engpass darstellen.

[5] evaluiert die Leistungsfähigkeit eines BLE Mesh Systems in einer Büroumgebung. Datendurchsatz und Paketfehler werden an einem ausgewählten Endgerät erfasst, welches als Basisstation bezeichnet wird. Die weiteren Endgeräte generieren regelmäßig Datenpakete und schicken diese zur Basisstation. Aus den Ergebnissen schließen die Autoren, dass ein BLE Mesh System weniger gut für Anwendungen der Zustandsüberwachung mit kleinen Sendezeitabständen geeignet sei. Werden die Zeitabstände größer, werde BLE Mesh als Kommunikationstechnologie attraktiver. Die hier verwendete Methode ist nicht geeignet, die Übertragungszeit zu erfassen, da die Basisstation und die Endgeräte nicht synchronisiert sind.

In [6] werden Datendurchsatz und der Received Signal Strength Indicator (RSSI) für BLE 5 mit zwei Kommunikationsgeräten gemessen. Die PHY-Varianten "LE 1M" und "LE 2M", welche sich wesentlich durch die bereitgestellte Übertragungsrate von 1 bzw. 2 Mbit/s auszeichnen. Ein Gerät wurde ortsfest positioniert, während die Position des zweiten Geräts variiert wurde. Die Ergebnisse zeigen, dass BLE 5 eine gute Abdeckung in Innenraumszenarios erreicht. Ein ähnlicher Versuchsaufbau wurde in [7] verwendet, welche das Verhalten von BLE untersucht, während zeitgleich ein ZigBee-System betrieben wird. Ergebnisse einer analytischen sowie der praktischen Evaluation werden präsentiert. Inwiefern die durchgeführten Tests reproduzierbar sind, wird nicht deutlich. Das Zeitverhalten wird weder in [6] noch in [7] untersucht.

In [9] und [10] wird ebenfalls die Leistungsfähigkeit eines Kommunikationssystems analysiert. In [9] steht ein hybrides System aus PROFIBUS DP und einem auf IEEE 802.15.4a-basierenden Funksystem im Fokus. [10] analysiert ein WirelessHART System. Bei diesen Messungen wurde die Firmware der Geräte adaptiert, sodass Datenpakete gezielt generiert, mit Zeitstempeln versehen und evaluiert werden konnten. Diese Messmethode erlaubt potentiell akkurate Messungen des Zeitverhaltens, erfordern jedoch einen Zugriff auf den Quellcode der Geräte oder Protokolle.

Der Blick auf die bisherigen Arbeiten rund um Performanztests insbesondere von Bluetooth zeigt, dass die Analysemöglichkeiten deutlich von der Zugänglichkeit des Quellcodes durch die Wissenschaftlerinnen und Wissenschaftler abhängt.

Wenn kein Zugang zum Quellcode besteht, wird in der Regel auf Paketgeneratoren wie iPerf [11] oder D-ITG [12] zurückgegriffen. Beide Generatoren bauen eine TCP- oder UDP-Verbindung auf, generieren an der Quelle Datenpakete und messen an der Senke den Datendurchsatz sowie die Paketverlustquote. Obwohl iperf und D-ITG etablierte Werkzeuge sind, sind sie für eine bestimmte Gruppe von industriellen Applikationen nicht geeignet. Typischerweise genutzte Schnittstellen wir UART oder digitale Ein- und Ausgänge (Digital I/O) werden nicht unterstützt. Ebenfalls bauen nur wenige Applikationen auf TCP- oder UDP-Verbindungen auf. Das Zeitverhalten eines Kommunikationssystems wird ebenfalls nicht (iperf) oder nicht ausreichend (D-ITG) genau erfasst, da Quelle und Senke nur grob synchronisiert werden können. Die von industriellen Anwendungen geforderte Genauigkeit von < 1 us kann somit nicht erfasst werden.

Vor diesem Hintergrund bleibt der Performanztest von Funkkommunikationslösungen aus Sicht industrieller Applikationen teilweise offen. Das in diesem Beitrag verwendet Testsystem schließt diese Lücken und wird im nächsten Abschnitt näher erläutert. Erste Messungen von PROFINET über Bluetooth sind in [13] publiziert.

3 Performanztest aus Sicht industrieller Anwendungen

Aus der Sicht einer industriellen Anwendung sind die zuvor beschriebenen Eigenschaften wie hohe Genauigkeit wichtig. Um das Zeitverhalten eines Kommunikationssystems entsprechend messen zu können, muss das Testsystem einem umfassenden Konzept folgen und seine Implementierung muss strengen Anforderungen genügen. In diesem Abschnitt wird zunächst ein universell anwendbarer Ansatz für Performanztests vorgestellt. Dann wird dieses Konzept auf das in dieser Arbeit untersuchte SUT angewandt. Außerdem werden die eingesetzten Messgeräte vorgestellt. Es wird beschrieben, welche logischen Kommunikationsverbindungen einem Performanztest unterzogen werden und wie diese Tests systematisch aufgebaut werden können.

3.1 Universelles Testkonzept

Das in dieser Arbeit verwendete Testsystem basiert auf einem universellen Konzept zur Untersuchung des Zeit- und Fehlerverhaltens von Funklösungen, wie in **Fehler! Verweisquelle konnte nicht gefunden werden.** dargestellt. Dieses Konzept enthält alle Komponenten, die erforderlich sind, um bei allen Tests zu jeder Zeit konsistente Ausgangsparameter zu gewährleisten. Das Testsystem emuliert die verteilte Anwendung, indem es Daten von der Quelle zum Ziel überträgt und dabei das SUT für den Transport verwendet. In diesem Fall ist das SUT das auf Bluetooth basierende Kommunikationssystem. Für reproduzierbare Tests muss der Funkkanal der vom SUT genutzt wird bekannt und vom Testsystem kontrollierbar sein.

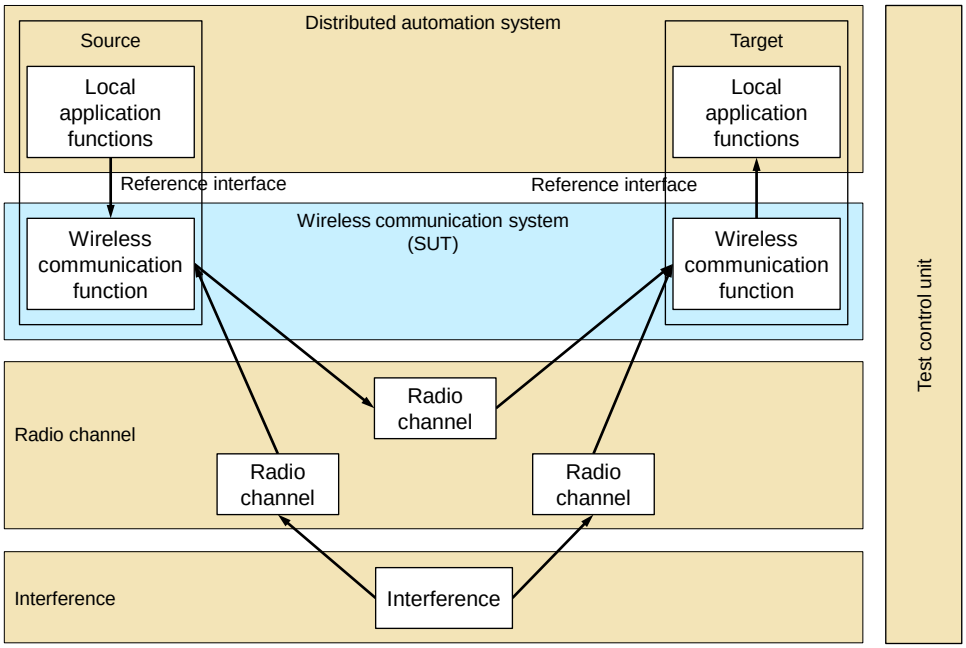

Abbildung 2: Universelles Konzept zur Untersuchung des Zeit- und Fehlerverhaltens von Funklösungen [3].

Wie in [1] beschrieben, können Performanztests entweder in einer realen Umgebung, in einer Referenzumgebung oder in einer Laborumgebung durchgeführt werden. Bei einer realen Umgebung befindet sich die zu untersuchende Funkanwendung in der Fabrik, in der sie tatsächlichen zum Einsatz kommt. Die Referenzumgebung ist eine physikalisch emulierte Umgebung, die einer realen Umgebung ähnelt, z. B. ein fabrikähnliches Gebäude mit der Möglichkeit Reflektionsflächen aufzubauen, um Mehrwegeausbreitung aufgrund von Maschinen oder Wänden nachzubilden. In der Laborumgebung werden ein Kanalemulator und ein Signalgenerator eingesetzt, um die Einflüsse einer realen Umgebung zu emulieren. Der Kanalemulator stellt die passiven Umwelteinflüsse wie Entfernung und Ausbreitungsverhalten dar, und der Signalgenerator fungiert als störender Frequenznutzer, z. B. ein anderes Funkkommunikationssystem. Die Funkgeräte werden in geschirmte Boxen untergebracht, um sie und die Messung vor möglichen Umwelteinflüssen zu schützen, die im Labor vorhanden sein könnten.

Jede Umgebung bringt individuelle Vorteile mit sich. So können in einem Labor größere Entfernungen leicht nachgebildet werden, während eine Referenzumgebung auf ihre tatsächliche räumliche Ausdehnung beschränkt ist.

Das gesamte Testsystem, bestehend aus einem verteiltem Anwendungssystem, dem Funkkanalemulator und dem Signalgenerator zum Erzeugen von Interferenzen, wird durch eine zentrale Managementeinheit konfiguriert, gesteuert und überwacht. Gleichzeitig werden die während des Tests ermittelten Messergebnisse erfasst und für eine anschließende Analyse vorverarbeitet. Dieses universelle Konzept zur Untersuchung des Zeit- und Fehlerverhaltens von Funklösungen ist sowohl für den Test von Standardgeräten oder Prototypen als auch für Simulationen geeignet.

3.2 Testaufbau

Die Topologie des Testsystems und des SUT ist in **Fehler! Verweisquelle konnte nicht gefunden werden.** abgebildet. Das SUT besteht aus einem auf Bluetooth Low Energy (BLE) basierenden Funksystem, einem PROFINET-Controller und drei PROFINET-Geräten. Dies entsprich einer möglichen realen industriellen Anwendung, bei der die Funkgeräte zur Übertragung des PROFINET-Datenverkehrs als Teil eines hybriden drahtgebundenen und drahtlosen Kommunikationsnetzwerks verwendet werden.

Im vorliegenden Fall besteht das Funksystem aus einem BLE-Master und drei BLE-Geräten. Der BLE-Master (BT-M) ist mit dem PROFINET-Controller verbunden und jedes BLE-Gerät (BT-D1, BT-D2 und BT-D3) ist mit jedem der PROFINET-Geräte verbunden. Die physikalischen Verbindungen werden durch gerade Linien dargestellt, die mit Bluetooth, PROFINET oder Digital I/O beschriftet sind.

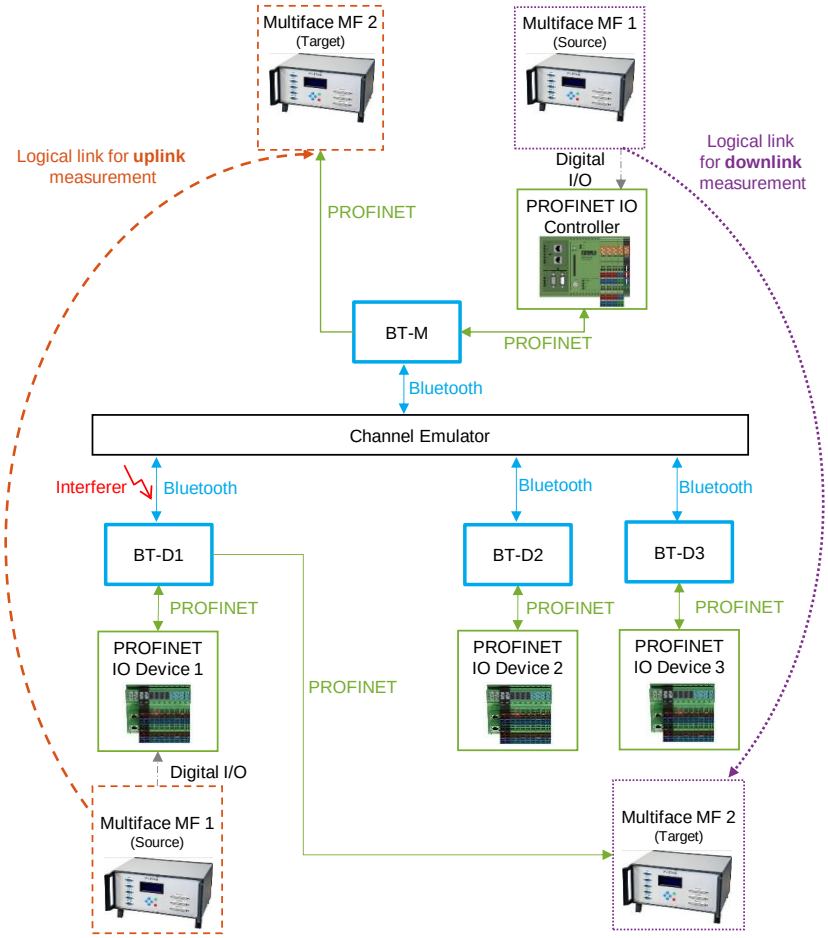

Abbildung 3: Topologie von Testsystem und SUT im Labor (vgl. [13]).

Das Testsystem besteht aus einem Kanalemulator, der die realen Ausbreitungsbedingungen nachbildet, und mehreren Multifaces, Geräten die die Basisparameter liefern und die Leistungsparameter messen. Das vom Kanalemulator verwendete Modell für den Pfadverlust ist das η-Power Law mit $\eta = 2$, was einen Pfadverlust im freien Raum entspricht. Ein Software-Tool berechnet die Dämpfung und steuert den Kanalemulator auf der Grundlage einer bestimmten Entfernung.

Die geschirmten Boxen haben Ethernet-Anschlüsse, zum Anschluss der Funkgeräte an das PROFINET-System oder bzw. an die Multifaces, und HF-Anschlüsse, die zum Anschluss an den Kanalemulator dienen.

Die Multifaces bieten mehrere Schnittstellen, die typischerweise bei industriellen Kommunikationsgeräten verwendet werden, wie z. B. RS-232, digitale E/A und Ethernet. Die Mutifaces wurden vom ifak auf der Basis leistungsfähiger FPGA entwickelt, wodurch sie in der Lage sind, Echtzeitverkehr zu generieren und Zeitparameter mit einer Genauigkeit von < 1 µs [14] zu erfassen. Darüber hinaus kann der Inhalt der zu übertragenden Nachrichten individuell festgelegt werden, wenn eine serielle Schnittstelle verwendet wird. Dies ist in Fällen nützlich, in denen PROFINET-Kommunikation getestet werden sollen, ohne PROFINET-Geräte zu verwenden, die ggf. Einschränkungen mit sich bringen. Das vorgeschlagene System erlaubt es, jedes Byte eines zu sendenden Pakets zu spezifizieren, so dass ein PROFINET-Paket emuliert werden kann.

Die Multifaces sind an einen PC angeschlossen, der als zentrale Managementeinheit fungiert, wie in Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.** beschrieben. Auf dem PC läuft eine vom ifak Magdeburg implementierte Software namens FTTManager, die die Funktionalität einer Teststeuereinheit bereitstellt.

3.3 Logische Verbindungen als Testobjekt

Eine logische Verbindung (LL) stellt eine Kommunikationsbeziehung zwischen zwei Instanzen einer Anwendung dar. Eine einzelne logische Verbindung kann mehrere physische Verbindungen umfassen. Abbildung 3 zeigt die LLs zwischen zwei Multifaces. In diesem Beitrag werden die Ergebnisse des Zeit- und Fehlerverhaltens dieser beiden logische Verbindungen vorgestellt: Eine Uplink-Kommunikation von Multiface (MF)3 zu MF1 über PROFINET-Gerät 1, BT-Gerät 1 und den BT-Master wie auf der linken Seite als gekrümmter gestrichelter Pfeil dargestellt und in umgekehrter Richtung eine Downlink-Kommunikation von Multiface (MF)1 zu MF3 über den, den BT-Master und BT-Gerät 1, wie auf der rechten Seite als gekrümmter gestrichelter Pfeil dargestellt.

Das MF1 erzeugt symmetrische Rechtecksignale entsprechend dem konfigurierten Sendezeitabstand für die logischen Verbindung. Die digitalen Ausgänge von MF1 sind mit den digitalen Eingängen des PROFINET-Controllers und der PROFINET-Geräte verbunden. Das MF3 hat die Zielanwendung implementiert und misst den PROFINET-Verkehr.

Darüber hinaus werden Daten über weitere logische Verbindungen vom PROFINET-Gerät 2 an den PROFINET-Controller und vom PROFINET-Controller an das PROFINET-Gerät 3 gesendet. Das Zeitverhalten dieser logischen Verbindungen ist nicht Gegenstand dieses Beitrages.

3.4 Beschreibung von Testgruppe und Testfällen

Den Performanztests liegt eine sorgfältige und umfassende Spezifikation zugrunde. Sie ist in Testgruppen (TG) gegliedert die jeweils mehrere Testfälle (TC) umfassen. Jeder Testfall ist durch einen konsistenten Satz von Einflussgrößenwerten bestimmt, die durch den Testaufbau realisiert werden. Jede Testgruppe zielt auf die Untersuchung eines bestimmten Aspekts oder eines bestimmten Einflusses auf das SUT ab. Bei den vorliegenden Performanztests steht die Untersuchung des Einflusses von Interferenzen im Fokus, wobei zusätzlich einzelne Einflussgrößen variiert werden.

Folgende Einflussgrößenwerte sind in dieser Testgruppe konstant:

Anzahl logischer Verbindungen

- In dieser Testgruppe waren 4 logische Verbindungen aktiv, jeweils zwei Verbindungen von den Geräten zur Steuerung und zwei von der Steuerung zu den Geräten.

Nutzdatenlänge je Verbindung

- Über jede logische Verbindung wurden 16 Bit Nutzdaten übertragen, die eine fortlaufende Nummer darstellen mit deren Hilfe die erfolgreiche Nachrichtenübertragung geprüft werden kann.

Sendezeitabstand je Verbindung

- Alle 131 ms wird eine neue Nachricht generiert, deren Inhalt eine fortlaufende Nummer darstellt.

PROFINET-Zyklus

- Für die PROFINET-Übertragung beträgt die Zykluszeit 16 ms.

Bluetooth Connection Intervall

- Das Connection Intervall ist die Zeit zwischen dem Beginn von zwei Verbindungsereignissen (CE). Bei BLE ist der Datenaustausch zwischen Master und Slave in aufeinanderfolgenden CEs [4] strukturiert. Das Bluetooth Connection Intervall beträgt 7,5 ms.

Abstand zwischen der Störquelle und dem BT-Gerät 1

- Als Störquelle wird ein Vektorsignalgenerator genutzt, der über den Kanalemulator mit dem BT-Gerät 1 verbunden ist, wobei eine Entfernung von 3 m konfiguriert wurde.

Störsignal

- Das Störsignal mit einem Vektorsignalgenerator gemäß IEEE 802.11g mit 20 MHz Bandbreite im Kanal 7 erzeugt.

Folgende Einflussgrößen wurden variiert und bestimmen die einzelnen Testfälle:

Abstand zwischen den Funkgeräten

- Die Entfernung zwischen BT-Master und BT-Slaves wurde mithilfe des Kanalemulators konfiguriert und beträgt entweder 10 m oder 100 m.

Stördauer (Ton/Toff)

- Das Störsignal liegt entweder kontinuierlich an oder wird mit folgenden Verhältnissen an und ausgeschaltet: 10/5 ms, 10/10 ms, 10/20 ms.

Störleistung

- Die konfigurierte Störleistung beträgt 10 mW oder 100 mW.

Beobachtungsdauer

- Die Ergebnisse basieren auf einer Stichprobe von jeweils 100.000 Nachrichtenübertragungen. Zusätzlich wurden Testfälle mit einer Beobachtungszeit von einer Stunde durchgeführt, bei denen das Störsignal 10 bzw. 20 Minuten deaktiviert, 20 Minuten aktiviert und nochmals 10bzw. 20 Minuten deaktiviert war.

4 Ergebnisse der Untersuchungen

Zur Bewertung des Einflusses von Interferenzen auf die PROFINET-Kommunikation über Bluetooth 5 wird die Aktualisierungszeit nach VDI 2185-4 herangezogen [14]. Die Aktualisierungszeit ist definiert als der Zeitabschnitt von der Übergabe des letzten atomaren Bestandteils der Nutzdaten einer Quelle an der Bezugsschnittstelle eines Ziels bis zur Übergabe des letzten atomaren Bestandteils der unmittelbar nachfolgend übertragenen Nutzdaten der gleichen Quelle. Die Aktualisierungszeit (Update time) entspricht im Idealfall dem Sendezeitabstand. Das heißt, die übertragenen Nutzdaten werden an der Bezugsschnittstelle des Ziels in denselben zeitlichen Abständen übernommen, wie sie an der Bezugsschnittstelle der Quelle übergeben wurden.

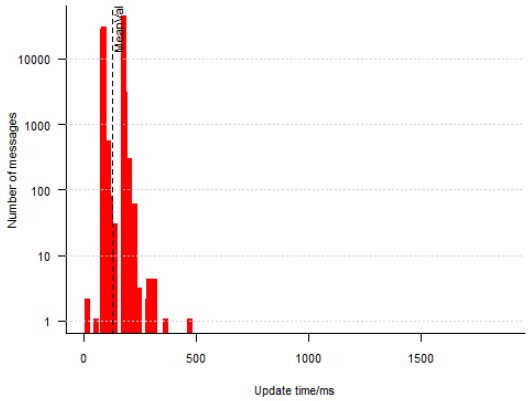

Abbildung 4: TC 01-03-01-01-01: Downlink, Geräteentfernung 10 m, ohne Störung

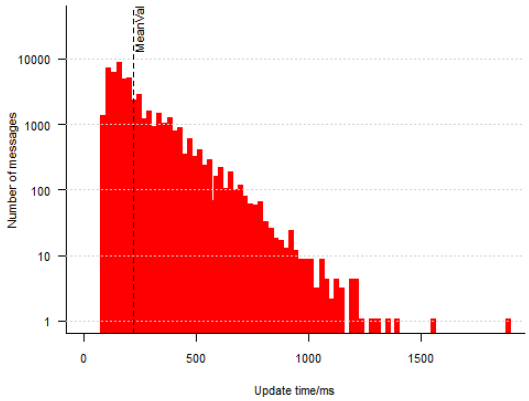

Abbildung 5: TC 01-03-01-01-03: Downlink, Geräteentfernung 10 m, 100 mW kontinuierliche Störung

Abbildung 4a zeigt das Histogramm der Aktualisierungszeit des Downlink für den ungestörten Fall. Die Werteverteilung zeigt eine deutliche Abweichung vom Mittelwert, dem Wert des Sendezeitabstandes von 131 ms. Die Ursache dafür sind der Kommunikationszyklus des PROFINET-Systems von 16 ms und das Connection Interval des Bluetooth Systems von 7,5 ms. Dadurch nimmt die Zeitdifferenz im Normalfall nicht den Wert des Sendezeitabstandes ein. Die Werte der Aktualisierungszeit für die einzelnen Nachrichtenübertragungen sind in Abbildung 6 dargestellt. Bis auf einzelne Ausreißer entsprechen die Werte der Standardabweichung. Die Ausreißer sind erfahrungsgemäß nicht in Störungen der Funkübertragung begründet, sondern im Zeitverhalten des komplexen Prozesses des gesamten Übertragungssystems. Neben den bereits erwähnten Zyklen der Kommunikationssysteme wird das Zeitverhalten durch die Verarbeitungsprozesse der beteiligten Controller beeinflusst. Bei der untersuchten Implementierung kommt dabei zusätzlich ein spezieller Algorithmus zum Einsatz, der eine robuste PROFINET-Übertragung über Funkkommunikationssystem gewährleistet. Dadurch kam es bei den hier vorgestellten Testfällen zu keiner Zeit zu einem Busfehler. Die PROFINET Übertragung wurde trotz der hier vorgestellten Interferenzen nicht gestört. Allerdings hat dieser Algorithmus Auswirkungen auf das Zeitverhalten der Nachrichtenübertragungen.

In Abbildung 5 ist das Histogramm der Übertragungszeit bei einer kontinuierlichen Störung von 100 mW bei 3 m Entfernung des Störers zum Bluetooth Slave und 10 m Entfernung zwischen den Kommunikationspartnern (man beachte die logarithmische Skale für die Anzahl der Übertragungen). Die großen Übertragungszeitwerte erklären sich durch den erwähnten Algorithmus und die Fehlerbewertung des Messsystems. Als fehlerhaft gilt eine Nachricht auch, wenn die Übertragungszeit den Wert des Sendezeitabstandes überschreitet. Es wird in diesem Fall davon ausgegangen, das ein neuer Nachrichtenwert vorliegt und der vorhergehende deshalb als veraltet angesehen werden kann.

Der Einfluss der Interferenz auf die Übertragungszeit der einzelnen Nachrichten ist in Abbildung 7 deutlich zu erkennen. Die eingezeichnete Standardabweichung ist größer im Vergleich zu Abbildung 6.

Bei der Bewertung der Aktualisierungszeitwerte ist zu berücksichtigen, dass sowohl die Steuerung als auch der Bluetooth Master jeweils 4 Verbindungen zu verwalten hat, was zusätzliche Zeitverzögerungen zum Zeitverhalten beiträgt.

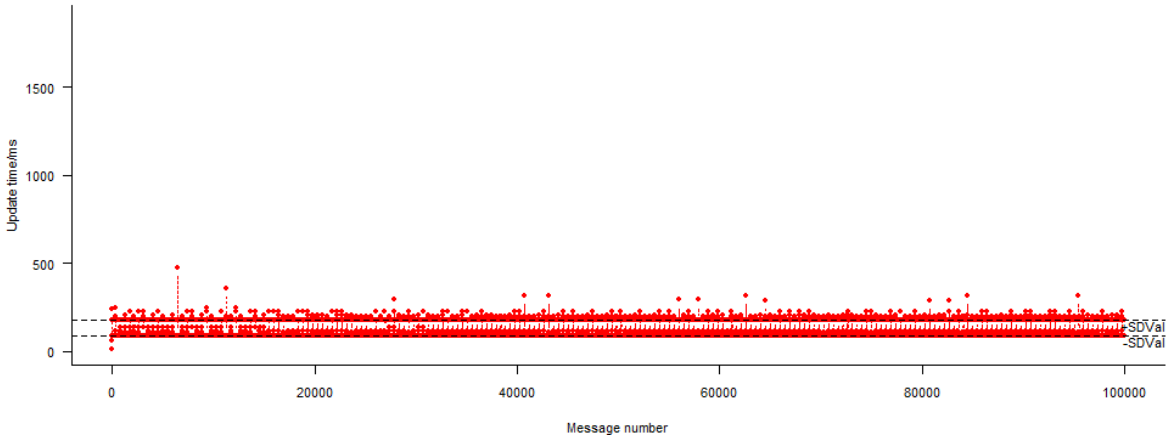

Abbildung 6: TC 01-03-01-01-01: Downlink, Geräteentfernung 10 m, ohne Störung

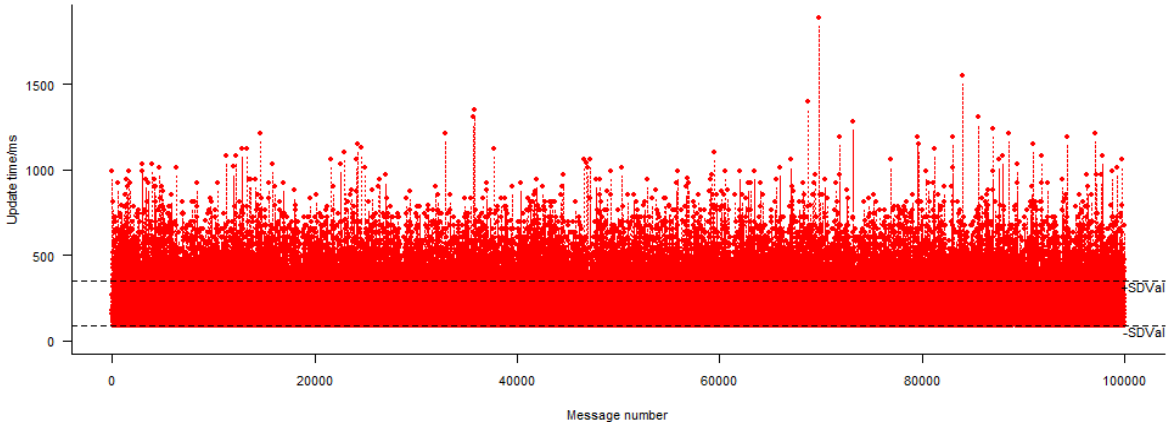

Abbildung 7: TC 01-03-01-01-03: Downlink, Geräteentfernung 10 m, 100 mW kontinuierliche Störung

Trotz kontinuierlichem Störsignal ist nicht zu erkennen, dass das Bluetooth-System die gestörten 20 MHz im Frequenzsprungverfahren ausspart, was ein verbessertes Zeitverhalten erwarten lassen würde.

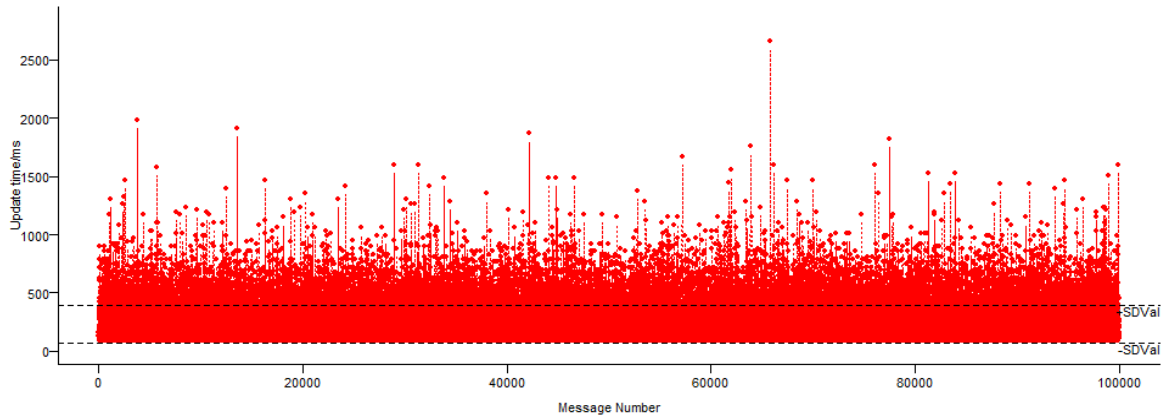

Abbildung 8: TC 01-03-02-03: Downlink, Geräteentfernung 100 m, 100 mW kontinuierliche Störung

So kommt es bei Vergrößerung der Entfernung zwischen den Geräten zu einem noch stärkeren Einfluss der Interferenz. In Abbildung 8 wurde die Skala der Aktualisierungszeit an die Maximalwerte angepasst. Wegen des schlechteren SNIR benötigt es mehr Übertragungen bis eine Nachricht den Kriterien einer korrekten Nachricht entspricht.

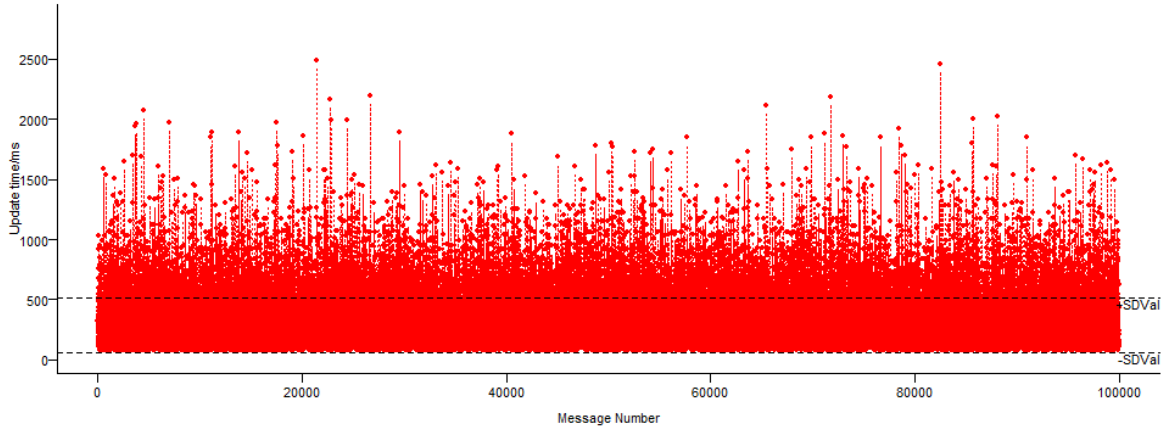

Abbildung 9: TC 01-03-02-03: Uplink, Geräteentfernung 100 m, 100 mW kontinuierliche Störung

In Abbildung 9 sind die Aktualisierungszeiten der Nachrichtenübertragungen des Uplinks dargestellt. Die Standardabweichung ist noch einmal deutlich größer als beim Downlink für denselben Testfall. Dieses Verhalten ist typisch für Sterntopologien und ist auch im ungestörten Fall festzustellen.

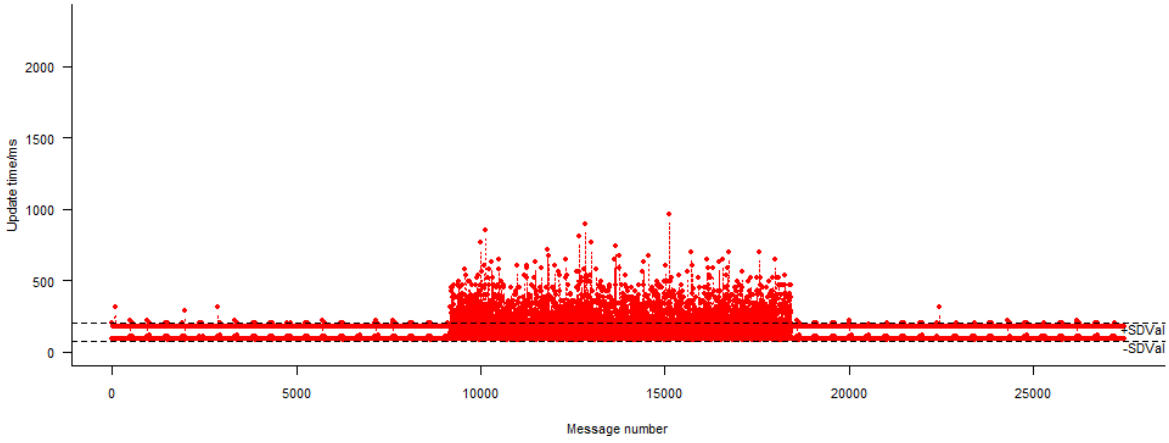

Abbildung 10: TC 01-03-03-01-01: Downlink, Geräteentfernung 10 m, 10 mW kontinuierliche Störung

Abbildung 10 zeigt die Aktualisierungszeiten für einen Testfall, bei dem das Störsignal erst nach 20 Minuten aktiviert und nach weiteren 20 Minuten wieder deaktiviert wurde. Auch in diesem Fall ist nicht zu erkennen,

dass das Bluetooth-System auf den Störer in einer Weise reagiert, die eine Verbesserung des Zeitverhaltens zur Folge hätte.

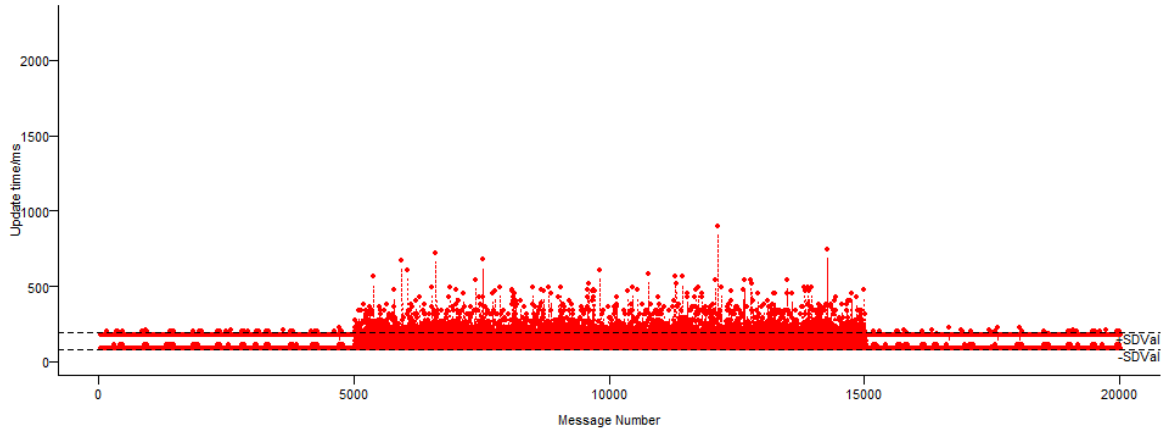

Abbildung 11: TC 01-03-04-02: Downlink, Geräteentfernung 10 m, 10 mW, Ton/Toff: 10/10 ms

Wird die Störung unterbrochen, zeigt sich das auch sofort im Zeitverhalten der Nachrichtenübertragung. Abbildung 11 zeigt die Werte für einen Testfall beim dem nach 10 Minuten ein Störer aktiviert wird, der jedoch nur 10 ms aktiv ist und anschließend für 10 ms deaktiviert wird. Diese zyklische Störung wirkt 20 Minuten auf das SUT bevor die verbleibenden 10 Minuten des Testfalls ungestört bleiben. In Abbildung 12 ist ein Ausschnitt für den Übergang vom ungestörten Fall zum gestörten Fall dargestellt. Die Übertragungszeit steigt plötzlich deutlich unf ist dann durch starke Schwankungen gekennzeichnet.

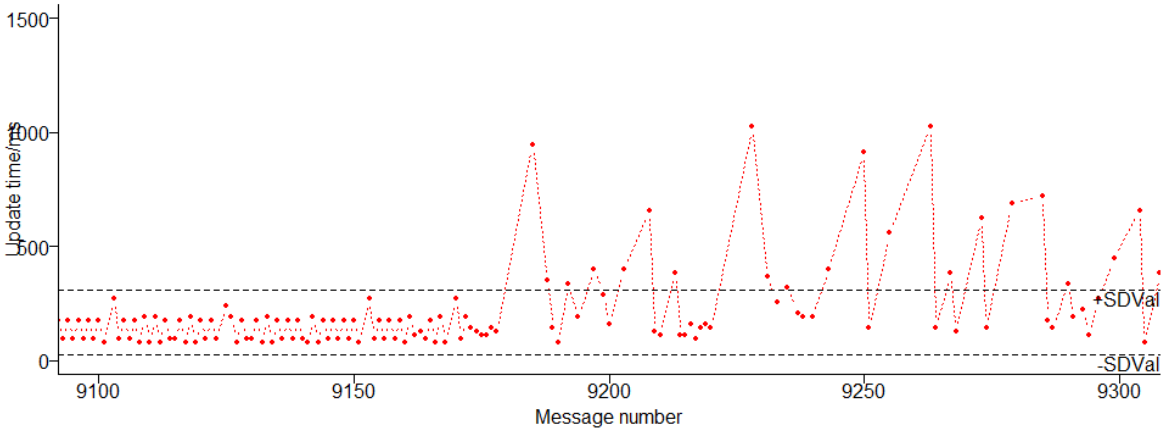

Abbildung 12: TC 01-03-03-02-02: Uplink, Geräteentfernung 10 m, 100 mW, kontinuierlich

5 Zusammenfassung

Eine Übersicht über die statistischen Werte von Übertragungszeit und Aktualisierungszeit gibt Tabelle 1 für die hier dargestellten Testfälle. Die Übertragungszeit für Nachrichtenübertragungen in einem Downlink kann durchaus Werte um 6 ms erreichen. Im Uplink sind es mindestens das dreifache. Allerdings benötigen 95% der Nachrichtenübertragungen über 100 ms und kommen damit bereits in den Bereich des Sendezeitabstandes. Der Einfluss der Interferenzen ist am Perzentil P95 der Übertragungszeit zu erkennen. Deutlicher zeigt sich der Einfluss am Mittelwert der Aktualisierungszeit, der im ungestörten Fall dem Sendezeitabstand von 131 ms entspricht, und an der Standardabweichung der Aktualisierungszeit.

Tabelle 1: Ergebnisse einer Testfallauswahl

Interferenz	Geräteentfernung	Link	TT Min [ms]	TT P95 [ms]	UT Mean [ms]	UT SD [ms]
No	10 m	Down	6	100	131	45
No	10 m	Up	21	123	136	53
10 mW, 10/10 ms	10 m	Down	7	121	145	67
10 mW, 10/10 ms	10 m	Up	27	125	178	123
10 mW, kont.	10 m	Down	6	123	170	92
10 mW, kont.	10 m	Up	18	126	259	202
100 mW, kont.	10 m	Down	6	125	223	130
100 mW, kont.	10 m	Up	19	126	336	279
100 mW, kont.	100 m	Down	6	125	232	161
100 mW, kont.	100 m	Up	25	126	286	231

Wie bereits erwähnt führen beim untersuchten Bluetooth-System Interferenzen nicht zum Abbruch der PROFINET-Kommunikation. Einflüsse auf das Zeitverhalten sind aber deutlich zu erkennen. Neben der Stärke der Störung bezogen auf Leistung und Zeit spielt auch die Richtung der Datenübertragung eine Rolle. Eine Anpassung des Bluetooth-Systems auf die Störung ist im Zeitverhalten nicht zu erkennen.

Die Einflüsse durch Interferenzen können dadurch gemindert werden, dass einzelne zyklische Prozesse besser aufeinander abgestimmt werden. Das Bestreben die Mediumnutzung durch weniger zeitliche Redundanz der Übertragungen zu verringern muss mögliche Interferenzszenarien berücksichtigen. Hier besteht Verbesserungspotenzial durch flexiblere Algorithmen. Die im Bluetooth-Standard spezifizierten Verfahren zur ereignisorientierten Übertragung sollten durch Implementierungen nutzbar gemacht werden.

Die vorgestellte Testreihe kann als Blaupause für weitere Testreihen zur Untersuchung von Kommunikationssystemen verwendet werden. Im Rahmen der Forschungen des Industrial Radio Lab Magdeburg werden Untersuchungen mit WiFi 6 und 5G durchgeführt. Die Untersuchungen sollen dazu beitragen, die Ungewissheit über das Störpotential auf industrielle Funkkommunikation zu verringern. Darüber hinaus sind die Testreihen ein Mittel, um Eigenschaften künftiger Funkkommunikationssysteme wie Elastizität und Resilienz zu erforschen.

6 Literaturverzeichnis

[1] L. Underberg und R. Rauchhaupt, „Performance testing of novel wireless communication networks for industrial automation on the example of 5G," *Conference on Communication in Automation (KommA)*, 2019.

[2] L. Underberg, S. Dietrich, R.Kays und G. Fohler, „Towards hybrid wired-wireless networks in industrial applications," in *IEEE ICPS*, St. Petersburg, Russia, May 2018.

[3] 5G-ACIA, „Performance Testing of 5G Systems for Industrial Automation," White Paper, 2021.

[4] G. Patti, L. Leonardi und L. L. Bello, „A bluetooth low energy real-time protocol for industrial wireless mesh networks," *IEEE IECON*, 10 2016.

[5] X. Wu und L. Xie, „On the wireless extension of PROFINET networks," *IEEE VTS Asia Pacific Wireless Communications Symposium*, 2019.

[6] E. Leon und M. Nabi, „An experimental performance evaluation of bluetooth mesh technology for monitoring applications," *IEEE Wireless Communications and Networking Conference (WCNC)*, 2020.

[7] B. Badihi, M. U. Sheikh, K. Ruttik und R. Jäntti, „On performance evaluation of ble 5 in indoor environment: An experimental study," *IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020.

[8] H. Karvonen, K. Mikhaylov, D. Acharya und M. M. Rahman, „Performance Evaluation of Bluetooth Low Energy Technology Under Interference," *13th EAI International Conference on Body Area Networks* , 2020.

[9] D. Jiang, M. Fei, H. Wang und T. Li, „Wireless network performance test in hybrid wired/wireless network system," *9th World Congress on Intelligent Control and Automation*, 2011.

[10] C. Krots, G. Cainelli, M. Feldman, C. Pereira und I. Müller, „Tool and method for end-to-end reliability analysis of wireless industrial networks," *6th International Embedded Systems Symposium*, 2020.

[11] iperf, „iPerf - The ultimate speed test tool for TCP, UDP and SCTP," [Online]. Available: https://iperf.fr/. [Zugriff am 24 09 2021].

[12] A. Botta, A. Dainotti und A. Pescapè, „A tool for the generation of realistic network workload for emerging networking scenarios,“ *Computer Networks vol. 56, no. 15,* 2021.

[13] G. Cainelli und L. Underberg, „Performance analysis of Bluetooth Low Energy in hybrid network with PROFINET,“ *IEEE ETFA,* 2021.

[14] A. Gnad und L. Rauchhaupt, „Multi-functional interface for test of industrial wireless solutions,“ *Embedded World Conference,* 2008.

[15] VDI/VDE 2185 Blatt 4, „Funkgestützte Kommunikation in der Automatisierungstechnik - Messtechnische Performancebewertung von Funklösungen für industrielle Automatisierungsanwendungen,“ 2019.

http://dx.doi.org/10.25673/39554

Zuverlässige Echtzeit-Funkvernetzung für die Automation – Durchbruch in den Sub-Millisekunden-Bereich

Andreas Frotzscher, Hannes Ellinger, Oliver Haala

Fraunhofer Institut für Integrierte Schaltungen IIS,
Institutsteil Entwurf Adaptiver Systeme EAS
Münchner Straße 16
01087 Dresden
{andreas.frotzscher, hannes.ellinger}@eas.iis.fraunhofer.de

Fraunhofer Institut für Integrierte Schaltungen IIS,Nordostpark 84
90411 Nürnberg
oliver.haala@iis.fraunhofer.de

Abstract: Viele Anwendungen im Bereich der Industrieautomation müssen Daten sehr latenzarm und in Echtzeit übertragen. Insbesondere bei der Anbindung beweglicher Anlagenteile bieten drahtlose Übertragungssysteme deutliche Vorteile gegenüber drahtgebundenen Lösungen. Aktuell verfügbare Funktechnologien können jedoch Anwendungen mit isochronen Zykluszeiten im Sub-Millisekunden-Bereich nicht unterstützen. Für diese Anwendungen wurde am Fraunhofer IIS die UWIN Funktechnologie entwickelt, die sich durch extrem niedrige Übertragungslatenzen und flexibel anpassbare Zykluszeiten im Bereich < 1 ms auszeichnet. Sie ist ausgelegt als Funk-Erweiterung von drahtgebundenen Echtzeit-Feldbussystemen und ermöglicht die latenzarme Übertragung im Nahbereich. Die UWIN Funktechnologie wurde in Laborumgebung verifiziert und wird aktuell in verschiedenen Feldversuchen evaluiert. Dieser Beitrag vergleicht die UWIN Funktechnologie mit den aktuell verfügbaren Funksystemen und stellt die Ergebnisse der Validierungstests in Laborumgebungen sowie der Feldversuche vor.

1 Einleitung und Motivation

Moderne Produktionssysteme sind geprägt von einem hohen Digitalisierungsgrad und Flexibilität der Produktionsprozesse. Sie enthalten häufig eine Vielzahl von beweglichen Subsystemen (z.B. Einarm-Roboter, Schlittenbahnen) und mobilen Systemen (z.B. Autonomous mobile robots, AMR). Diese benötigen dabei meist eine latenzarme Datenanbindung an die übergeordnete Steuerung bzw. Regelung oder an andere Subsysteme. Insbesondere zur Regelung sehr dynamischer Prozesse (z.B. Antriebsregelung, Motion Control, z.B. in Robotikszenarien siehe Abbildung 1) werden isochrone Zykluszeiten im Sub-Millisekunden-Bereich bei einer extrem hohen Übertragungszuverlässigkeit benötigt. In [1] wurden die Anforderungen verschiedenster industrieller Anwendungen untersucht und tabellarisch zusammengestellt (Siehe Tabelle 1).

Abbildung 1: Beispiel einer Roboter-gestützten Prüfanlage (Quelle: Fa. Hiersemann Prozessautomation)

Derzeit werden dafür ausschließlich drahtgebundene Echtzeit-Bussysteme (z.B. ProfiNET IRT, EtherCAT, Sercos III) in Verbindung mit Kabelschleppketten, Drehdurchführungen und Schleifkontakten eingesetzt. Dies beschränkt jedoch sehr die Beweglichkeit der Anlagenteile, benötigt einen relativ großen Bauraum und kann beim Retrofit von bestehenden Anlagen nur mit erheblichem Kosten- und Zeitaufwand eingesetzt werden. Der Datenaustausch zwischen mobilen Subsystemen (z.B. zwischen kollaborierenden AMR's) ist mit drahtgebundenen Bussystemen überhaupt nicht möglich.

Drahtlose Übertragungssysteme können hier deutliche Verbesserungen erzielen, da sie neben der Flexibilität, eine uneingeschränkte Beweglichkeit ermöglichen, den Retrofit von bestehenden Anlagen sehr erleichtern und den Datenaustausch zwischen kollaborierenden mobilen Systemen erst ermöglichen.

	Diagnose & Wartung		**Diskrete Fertigung**		**Lager und Logistik**		
	Generell	**Condi-tion Monito-ring**	**Generell**	**Motion Control**	**Generell**	**AGV**	**Kran-szenario**
Latenz (Sensor → Controller → Aktor)	>20 ms	100 ms	1 ms – 12 ms	250 µs – 1 ms	> 50 ms	15 ms – 20 ms	15 ms – 20 ms
Zuverlässigkeit (Erfolgreiche Übertragung in Latenzanf.)	$1 - 10^{-4}$	$1 - 10^{-5}$	$1 - 10^{-9}$	$1 - 10^{-9}$	$1 - 10^{-2}$	$1 - 10^{-6}$	$1 - 10^{-6}$
Datenrate	kbit/s – Mbit/s	kbit/s	kbit/s – Mbit/s	kbit/s – Mbit/s	kbit/s – Mbit/s	kbit/s – Mbit/s	kbit/s – Mbit/s
Paketgröße	> 200 Byte	1 – 50 Byte	20 – 50 Byte	20 – 50 Byte	< 300 Byte	< 300 Byte	< 300 Byte
Reichweite	< 100 m	100 m – 1 km	< 100 m	< 50 m	< 200 m	~ 2 m	< 100 m

Tabelle 1: Anforderung verschiedener industrieller Anwendungen

2 Stand der Technik

Bisher in der Industrie eingesetzte Funktechnologien (z.B. WLAN, IWLAN, Bluetooth, IO-Link Wireless) können die o.g. Anforderungen im Bereich Motion Control nicht vollständig erfüllen. Ihre Übertragungslatenzen sind zu hoch, sodass sie keine isochronen Zykluszeiten von < 1 ms unterstützen. Gleichzeit werden neue Funktechnologien demnächst zur Verfügung stehen und Verbesserungen hinsichtlich der Latenz und Zuverlässigkeit anbieten. Hierbei ist neben der neuen Mobilfunk-Generation 5G mit Release 16 in ihrer Ausprägung URLLC (engl. Ultra Reliable Low Latency Communication) auch die neue Funktechnologie DECT 2020 New Radio zu nennen. Gleichzeitig wird an einer Erweiterung des WiFi-Standards gearbeitet, der mit IEEE 802.11be kürzere Übertragungslatenzen erreichen soll. Diese Technologien werden nachfolgend aus Sicht des Maschinen- und Anlagenbauers in ihren technischen Parametern und ihren Marktreifegraden verglichen. Hierfür werden 2 Szenarien betrachtet (Siehe Abbildung 2).

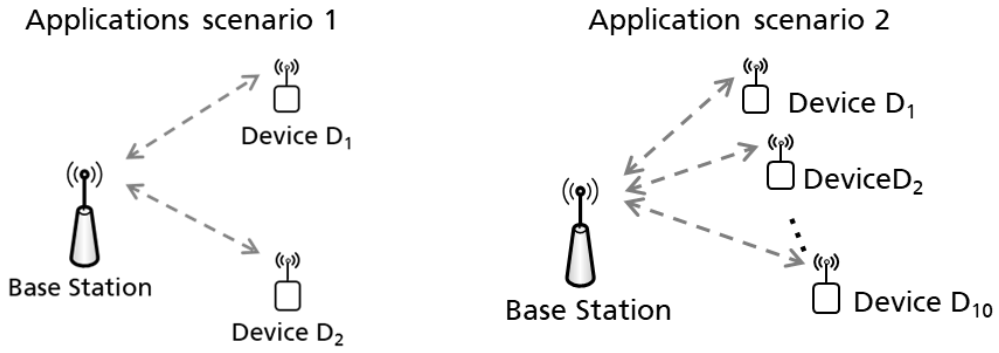

Abbildung 2 Betrachtete Applikationsszenarien für Technologievergleich (Zyklische Übertragung von je 16 Byte in Downlink und Uplink zwischen Base Station und 2 bzw. 10 Field Devices

In beiden Szenarien sollen zyklisch Daten zwischen einer SPS oder Controller mit drahtlos angebundenen Sensoren, Aktoren bzw. sonstigen Peripheriegeräten übertragen werden. Dabei ist die SPS an eine Base Station angeschlossen und die Sensoren, Aktoren etc. an sog. Field Devices angebunden. In beiden Szenarien sollen zyklisch 16 Byte von der Base Station zu den Field Devices (Downlink) und zurück (Uplink) übertragen werden.

Wie oben erwähnt unterschieden sich die betrachteten Funktechnologien in ihrer Marktreife. Während von einigen Technologien bereits Produkte am Markt verfügbar sind, ist bei anderen Technologien der Standardisierungsprozess noch nicht abgeschlossen. Daher soll hier ein Marktreifegrad (engl. Market Readiness Level, MRL) eingeführt werden (), angelehnt an den Technologiereifegrad (Technology Readiness Level, TRL).

MRL	Beschreibung
1 – 4	Technologie noch in Standardisierungsprozess
5	Standardisierung abgeschlossen und verabschiedet
7	Prototypen der Technologie am Markt verfügbar
9	Produkte der Technologie am Markt verfügbar

Tabelle 2: Kategorisierung des Marktreifegrades

Die betrachteten Funktechnologien werden hinsichtlich von 5 Parametern verglichen:

- MRL
- Minimal erreichbare Zykluszeit in dem Applikationsszenario 1 (2 Field Devices)
- Minimal erreichbare Zykluszeit in dem Applikationsszenario 2 (10 Field Devices)
- Paketverlustrate
- Netto-Paketgröße pro Zykluszeit, Verbindung und Downlink/Uplink

Abbildung 3 fasst die Ergebnisse des Technologievergleichs grafisch zusammen und stellt sie mit den Anforderungen aus dem Bereich Motion Control gegenüber.

IO-Link Wireless ist bereits seit einiger Zeit mit Produkten auf dem Markt verfügbar und erreicht unter den verglichenen Funk-Technologien die höchste Zuverlässigkeit (d. h. die niedrigste Paketverlustrate). Die Echtzeitfähigkeit (d. h. die minimal erreichbare Zykluszeit) sowie die Nettonutzdatenmenge pro Field Device sind jedoch vergleichsweise sehr gering. Die Spezifikation von DECT 2020 NR ist Ende 2020 verabschiedet worden. Es bietet gute Echtzeit- und Datennutzlastfähigkeiten, aber seine Zuverlässigkeitseigenschaften sind nicht zufriedenstellend. Die Spezifikation von 5G Rel. 16 ist zu Beginn dieses Jahres verabschiedet worden und bietet eine bessere Echtzeit und die höchste Nettonutzdatenmenge, hat aber Defizite bei der Übertragungszuverlässigkeit.

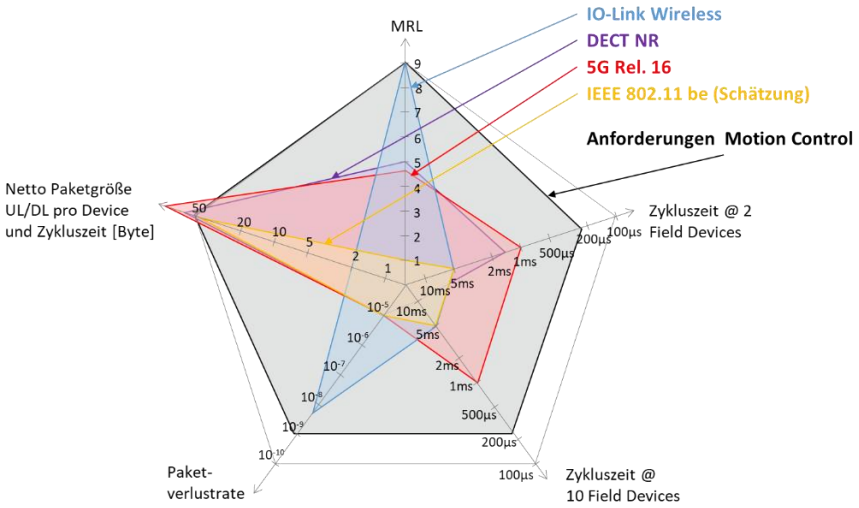

Abbildung 3: Vergleich des Stands der Technik mit den Anforderungen im Bereich Motion Control

Aktuell erarbeitet die IEEE 802.11be Working Group eine Erweiterung des WLAN Standards für niedriglatente Datenübertragungen. Dessen Standardisierung ist jedoch noch in einer sehr frühen Phase und die Leistungsparameter sind noch nicht abgestimmt. Anhand der adressierten Anwendungsfälle ist davon auszugehen, dass Übertragungslatenzen von minimal 5 ms bei einer mittleren Übertragungszuverlässigkeit erreicht werden sollen. Damit schneidet IEEE 802.11be im Vergleich zu den anderen Technologien am schlechtesten ab. Der Vergleich zeigt, dass keine der betrachteten Funktechnologien die Anforderungen von Motion Control-Anwendungen umfänglich erfüllt. Die Kombination von einer sehr niedrigen Übertragungslatenz bei einer gleichzeitig sehr hohen Übertragungszuverlässigkeit kann derzeit keine der untersuchten Funktechnologien bieten.

Generell existieren zur Erhöhung der Übertragungszuverlässigkeit etablierte Datensicherungsverfahren, die oberhalb der Bitübertragungsschicht eingesetzt werden können. So können im Falle eines erkannten Übertragungsfehlers angeforderte Sendewiederholungen (Automatic Repeat reQuest (ARQ)-Protokolle) genutzt werden, um ein Paket mehrfach zu übertragen. Dieser Prozess führt allerdings zu einer Erhöhung der Latenz. Einfacher, aber zeitlich deterministisch können Übertragungen unaufgefordert redundant erfolgen, wobei dies eine feste Reduktion des Datendurchsatzes zur Folge hat [2]. Weiterhin gewährleisten die genannten Verfahren keine Möglichkeiten zur Kompensation eines dauerhaft gestörten Übertragungskanals.

Eine Alternative dazu stellen die Verfahren der Network Coded Cooperation (NCC) dar. Diese vereinen die Vorteile kooperativer Verfahren mit den Vorteilen Netz-basierter Codierverfahren. Die Anwendung von NCC führt somit zu robusten und gleichermaßen effizienten Kommunikationseigenschaften. Es existieren zwar zahlreiche Entwürfe für Protokollschichten mit NCC-Funktionalitäten [3, 4, 5, 6], allerdings wurden nur die wenigsten dieser Entwürfe für URLLC Anwendungsfälle konzipiert. Die einzige bekannte Ausnahme bildet XOR-CoW Protokoll [6]. Allerdings sind die für dieses Protokoll vorgeschlagenen, Netz-basierten Codier- und Decodiervorgänge so simpel, dass das Potential an Zuverlässigkeitserhöhung nicht vollständig ausgeschöpft wird.

3 UWIN - Ultra reliable Wireless Industrial Network

Zur Unterstützung schneller Automatisierungssysteme wurde am Fraunhofer IIS die Echtzeit-Funktechnologie UWIN (Ultra reliable Wireless Industrial Networks) erforscht. Sie ist ausgelegt als eine Drahtlos-Erweiterung von Echtzeitfeldbussystemen zur isochronen Datenübertragung im Nahbereich, z.B. in Fertigungszellen. Sie zeichnet sich durch eine extrem hohe Übertragungszuverlässigkeit und gleichzeitig extrem niedrigen Übertragungslatenzen aus, wodurch isochrone Zykluszeiten von minimal 125 µs zuverlässig unterstützt werden können.

In Analogie zur Topologie eines Feldbus-Systems dient ein Funkmodem als Gateway zur drahtgebundenen Infrastruktur, bzw. zur Regelung. Für die Systemarchitektur ist es naheliegend, dieses Funkmodem als Base Station für ein zelluläres Funksystem zu betreiben. Die Kommunikation der Funkteilnehmer (Field Devices) mit der Regelung wird über die Base Station abgewickelt, zusätzlich wird auch die Zeitsynchronisation der mobilen Field Devices über die Base Station gesteuert.

Abbildung 4 zeigt die Netzwerktopologie zusammen mit dem inneren Aufbau der Funkmodems. Der Kommunikationsstack ist schematisch anhand der Schichten des OSI (Open Systems Interconnection) Modells dargestellt. In der Abbildung sind die untersten Schichten (PHY, MAC und Netzwerkschicht) dargestellt. Zusätzlich enthält der Master das Feldbus-Gateway zur Anbindung an die übergeordnete Regelung, bzw. der Infrastruktur, während die Slaves digitale I/O Schnittstellen zum Anschluss von Sensoren und Aktoren enthalten. Darüber hinaus verfügt der Master über ein Modul zur Koexistenzanalyse, das dem Netzwerkmanagement Informationen über den Zustand des Funksystems und der Funkumgebung liefert.

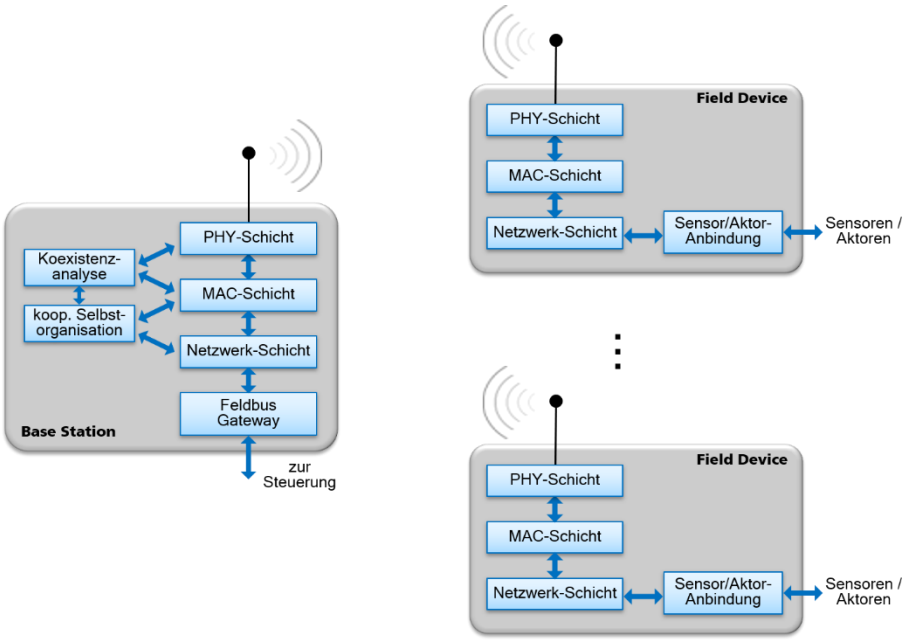

Abbildung 4: Netzwerktopologie und Architektur der Funkknoten

Die Leistungsparameter dieser neuen Funktechnologie sind in Tabelle 3 zusammengefasst.

Tabelle 3 Leistungsparameter des UWIN Echtzeit-Funk Systems

Parameter	Wert	Bemerkung
Zykluszeit	125 µs – 10 ms	Adaptierbar an Anwendungsanforderung
Topologie	Stern	
Anzahl Slaves / System	6 Slaves (@ 125 µs) 80 Slaves (@ 1 ms)	Einstellbar
Payload pro Field Device	max. je 32 Byte in UL & DL	Adaptierbar
Reichweite	> 20 m	
Zuverlässigkeit	Paketverlustrate < 10^{-8}	
Frequenzband	5 GHz (U-NII)	
Signalbandbreite	2 x 20 MHz	

Beim Einsatz von Funksystemen in der Industrieautomation ergeben sich verschiedene Herausforderungen. Zum einen liegen sehr funkunfreundliche Umgebungsbedingungen vor. Diese zeichnen sich durch einen starken und schwer vorhersagbaren Mehrwegeempfang und Interferenzen mit anderen Funksystemen aus. Dies erfordert auf physikalischer Schicht eine robuste Wellenform, mit der sich die extremen und gegensätzlichen Anforderungen Zuverlässigkeit und geringe Latenz im Rahmen der Vorgaben aus Sensor-Aktor-Netzwerken vereinbaren lässt. Durch die Nutzung mehrerer Diversitätsgrade (Raum, Zeit und Frequenz) im Medienzugriff wird die Zuverlässigkeit des Systems erhöht. Ein weiterer wichtiger Baustein zur Garantie der Echtzeitfähigkeit ist der gleichzeitige Betrieb eines Interferenz- und Koexistenzmanagements, um den Zustand des Funksystems überwachen zu können und gleichzeitig benachbarte Funksysteme erkennen zu können.

Der diagnostizierte Zustand des Funksystems kann auch zur kooperativen Selbstorganisation genutzt werden. Ein Aspekt dieser Selbstorganisation besteht darin, besonders störanfällige Kanäle durch die Ausnutzung kooperativer Paketweiterleitungsstrategien und NCC gezielt zu kompensieren. Gleichzeitig können die für

NCC erforderlichen Signalisierungsinformationen interpretiert werden, um den aktuellen Zustand des Funksystems noch genauer zu erfassen. Eine Modellbildung und Simulation des Funksystems erlaubt einerseits eine geeignete Wahl der Freiheitsgrade des NCC Verfahrens hinsichtlich einer angestrebten Erhöhung der Übertragungszuverlässigkeit und ermöglicht andererseits einen unmittelbaren Vergleich zwischen NCC und nicht-kooperativen Ansätzen.

3.1 Modellbildung

In Anlehnung an die bereits vorgestellten Applikationsszenarien (Abbildung 2) erfolgt eine Modellbildung des typischen Anwendungsfalls einer hochdynamischen Regelung. Innerhalb eines Kommunikationszyklus sollen Daten zwischen einer Base Station (BS, Controller) und den Field Devices (FD, Sensor/Aktor-Knoten) ausgetauscht werden. Konkret werden die auszutauschenden Down- bzw. Uplink Pakete mit Groß- bzw. Kleinbuchstaben bezeichnet (Abbildung 5).

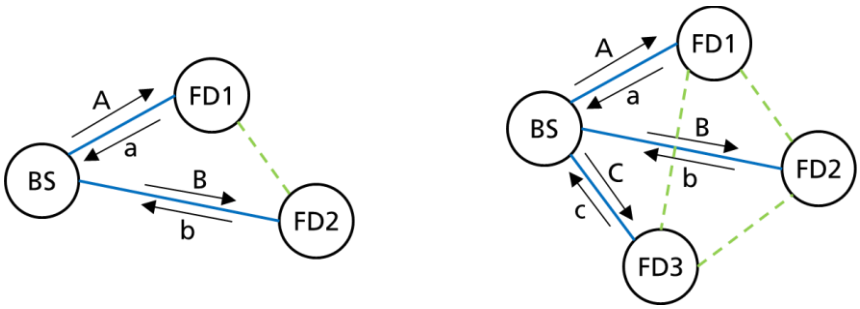

Abbildung 5: Visualisierung der gewünschten Paketübertragungen in einem Kommunikationssystem mit zwei (links) bzw. drei Field Devices (rechts)

Weiterhin wird von einer statischen, vollvermaschten Topologie ausgegangen, d.h. ein ausgesendetes Paket eines Teilnehmers kann sowohl unmittelbar von dem direkt adressierten Teilnehmer (blaue Linien in Abbildung 5) als auch von allen anderen Teilnehmern (grüne Linien in Abbildung 5) empfangen werden, sofern die Kanaleigenschaften zu diesem Zeitpunkt günstig sind. Die Modellierung dieser Eigenschaften erfolgt auf Paketebene, wobei jeder Kanal durch eine Paketfehlerrate parametrisiert wird. Die Paketfehlerrate gibt an, welcher Anteil an Paketaussendungen nicht erfolgreich empfangen werden konnte. Hierbei werden die Kanäle als symmetrisch und die auf unterschiedlichen Kanälen auftretenden Paketfehler als zeitlich unkorreliert angenommen.

Zur Bewertung des Verbesserungspotentials durch den Einsatz von NCC werden drei unterschiedliche Möglichkeiten der Datensicherung untersucht. Bei dem Verzicht auf Maßnahmen zur Datensicherung werden alle zu übertragenden Pakete genau einmal ausgesendet (einfache Übertragung). Eine einfache Möglichkeit einer Datensicherung besteht darin, die Aussendung jeden Datenpakets unaufgefordert zu wiederholen (zweifache Übertragung). Beim Einsatz von NCC werden Pakete entweder nativ (z.B. A) oder kombiniert codiert (A+B) übertragen, wobei ein kombiniertes Paket aus mehreren nativen Paketen gebildet wird (Codiervorgang). Der Empfänger eines einzelnen kombinierten Paketes kann die enthaltenen nativen Pakete nur dann rekonstruieren, wenn er noch weitere geeignete kombiniert codierte oder native Pakete empfängt (Decodiervorgang). Für die konkrete Implementierung des NCC Codecs sei auf [7] verwiesen.

Ein NCC Codierschema umfasst einerseits eine Zuweisung der verfügbaren Zeitschlitze an die Netzteilnehmer und andererseits eine Vorschrift nach welcher die Bildung kombiniert codierte Pakete erfolgen soll. Im Sinne der NCC sind hier zwei Phasen vorgesehen. In einer ersten Phase (Broadcast Phase) übermitteln alle Netzteilnehmer nacheinander native Pakete, wohingegen in der zweiten Phase (Relaying Phase) kombiniert codierte Pakete übermittelt werden. Die im Rahmen dieses Beitrags untersuchten Verfahren der Datensicherung sind in Abbildung 6 gegenübergestellt. Als Metrik zur Bewertung eines Datensicherungsverfahrens wird die Paketverlustrate definiert, wobei ein Paketverlust einen gescheiterten Versuch der Übertragung eines Pakets innerhalb der Zykluszeit beschreibt. Je nach Datensicherungsverfahren wird ein Paketverlust durch einen oder mehrere Paketfehler hervorgerufen.

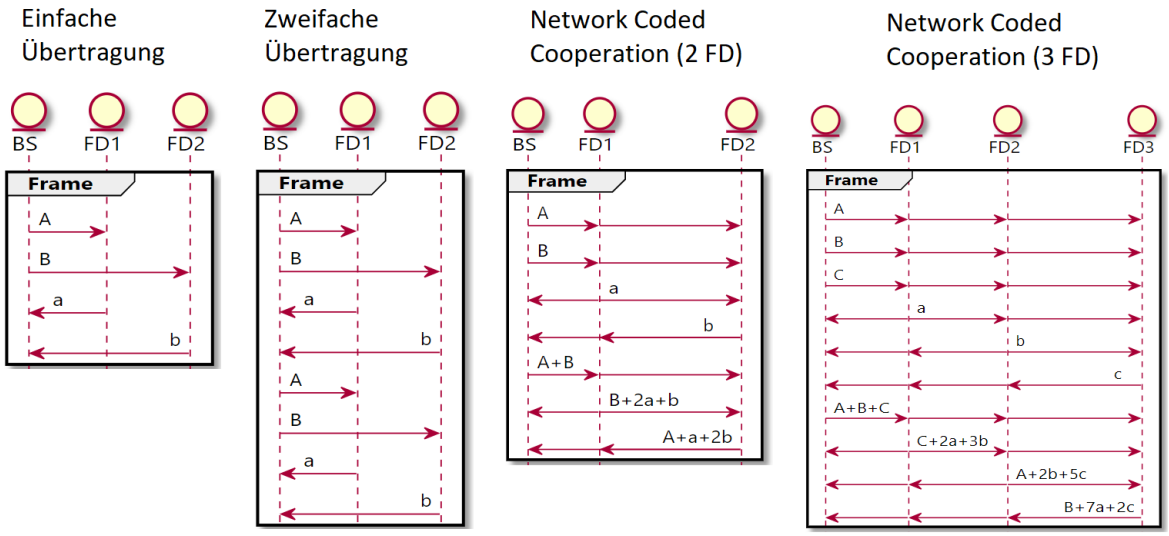

Abbildung 6: Gegenüberstellung unterschiedlicher Datensicherungsverfahren

3.2 Simulationsergebnisse

Bei den durchgeführten Simulationen wird exemplarisch die Paketverlustrate zwischen BS und FD1 ausgewertet und gesondert für den Uplink bzw. Downlink angegeben, falls sich die beiden Werte voneinander unterscheiden. In der Praxis gibt es zwei häufig auftretende Fehlerfälle. Sofern sich weitere Funksysteme in der unmittelbaren Umgebung befinden, können auf allen Kanälen Paketfehler sporadisch auftreten. Weiterhin ist es möglich, dass ein Störobjekt die Kanaleigenschaften zwischen BS und einem FD dauerhaft signifikant verschlechtert. Diese Fehlerfälle werden im Folgenden gesondert untersucht.

Zunächst wird ein Netzwerk mit homogenen Kanaleigenschaften betrachtet, d.h. die Paketfehlerrate auf allen Kanälen ist gleich hoch. Bei dem Verzicht auf ein Datensicherungsverfahren führt jeder Paketfehler unmittelbar zu einem Paketverlust, daher sind Paketfehler- und Paketverlustrate identisch. Wird jedes Datenpaket unaufgefordert wiederholt übertragen, führt dies bereits zu einer signifikanten Reduktion der Paketverlustrate. Der Einsatz von Network Coded Cooperation führt, je nach Datenflussrichtung (Uplink/Downlink), zu leicht unterschiedlichen Paketverlustraten. Allerdings befinden sich diese signifikant unterhalb der Paketverlustrate, die sich durch die wiederholte Übertragung ergibt (Abbildung 7).

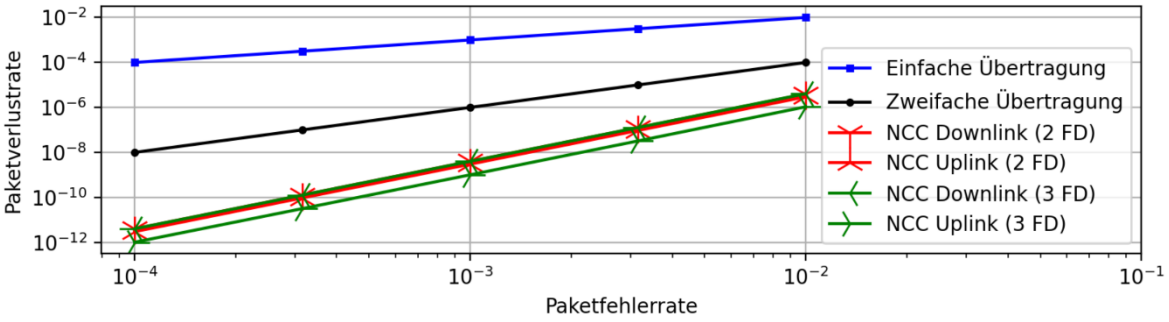

Abbildung 7: Resultierende Paketverlustrate zwischen BS und FD1 für Kanäle mit gleichen Eigenschaften

Weiterhin wird ein Netzwerk mit Kanälen betrachtet, die sich hinsichtlich der Paketfehlerrate unterscheiden. Der Kanal zwischen BS und FD1 ist gestört mit einer Paketfehlerrate zwischen 1 und 100%, wohingegen alle anderen Kanäle eine Paketfehlerrate von 0,01% aufweisen. Bei dem Verzicht auf ein Datensicherungsverfahren entspricht die Paketfehlerrate des gestörten Kanals der resultierenden Paketverlustrate. Auch hier führt eine unaufgefordert wiederholte Übertragung zu einer erheblichen Reduktion der resultierenden Paketverlustrate. Allerdings können im Fall eines Totalausfalls des Kanals keine Daten mehr zwischen BS und FD1 ausgetauscht

werden. Dies ist unter Nutzung von NCC nicht der Fall, da die Daten über redundante Pfade übermittelt werden können (Abbildung 8).

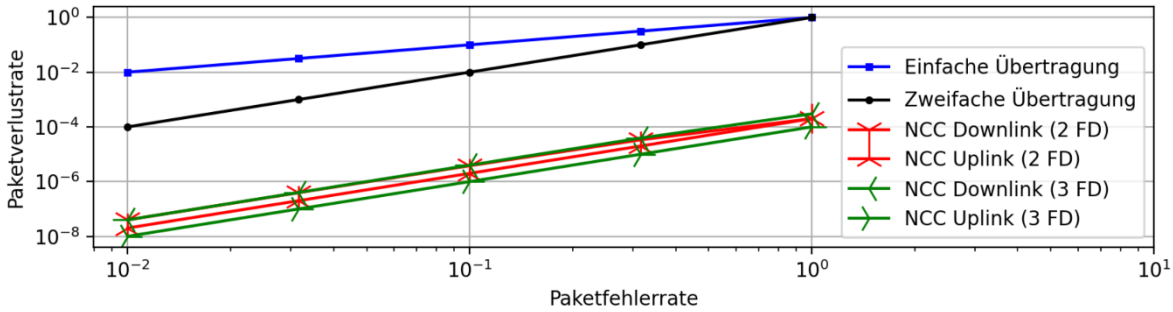

Abbildung 8: Resultierende Paketverlustrate zwischen BS und FD1 bei gestörtem Kanal zwischen BS und FD1

4 Technologie-Evaluierung

Um die UWIN Funktechnologie in Praxisumgebungen erproben zu können, hat das Fraunhofer IIS ein UWIN Evaluation-Kit entwickelt. Dieses besteht aus einer Base Station und zwei Field Devices (siehe Abbildung 9), die periodisch mit einer Zykluszeit von 125 µs Nutzdaten in Downlink und Uplink übertragen. Dabei überwacht die Base Station die Leistungsparameter der einzelnen Verbindungen. Das System arbeitet im lizenzfreien 5GHz U-NII Frequenzband, wobei der Anwender die genutzten Frequenzkanäle selbst bestimmen kann. Über eine Ethernet-Schnittstelle kann der Nutzer auf das Dashboard des Evaluationskits (siehe Abbildung 10) zugreifen. Es stellt verschiedene Leistungsparameter und Metriken zur Übertragungszuverlässigkeit der einzelnen Verbindungen sowohl im Zeitverlauf als auch als Langzeit-Statistiken dar. Mit dem Evaluation Kit soll gemeinsam mit interessierten Partnern die UWIN-Echtzeit-Funktechnologie in konkreten Anwendungsumgebungen erprobt werden, um so Erfahrungen in konkreten Einsatzszenarien und unter Einfluss umgebender Funksysteme zu sammeln. Im Zuge dieser Erprobung werden weiterhin Echtzeit-Kanaleigenschaften erhoben. Diese lassen sich gut verwerten, da sich einerseits die resultierende Übertragungszuverlässigkeit unmittelbar aus diesen ableiten lässt und andererseits eine für diese Anwendungsumgebung realistische Parametrisierung der in Abschnitt 3.1 vorgestellten Modelle ermöglicht wird.

Abbildung 9 Field Device des UWIN Evaluation Kit

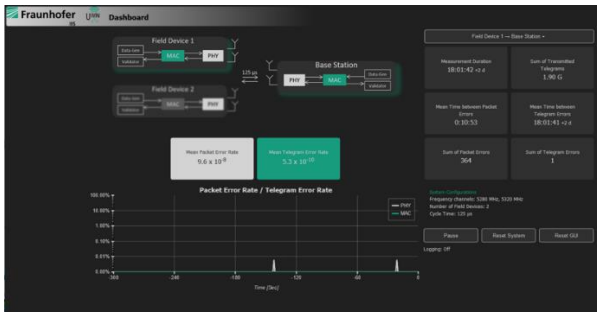

Abbildung 10 Dashboard des UWIN Evaluation Kit

In industriellen Umgebungen werden häufig mehrere Funktechnologien für verschiedene Anwendungen parallel eingesetzt. Um auch in diesen Umgebungen die hohe Übertragungszuverlässigkeit sicherzustellen, wurde insbesondere die Störresistenz gegenüber anderen Funktechnologien von Beginn an in dem Entwicklungsprozess mit betrachtet. Zur Verifikation werden derzeit in einem Funk-Testbed verschiedene Koexistenz-Szenarien aufgebaut und die Leistungsfähigkeit des UWIN Funksystem evaluiert. In der Präsentation werden erste Ergebnisse der Evaluierungstest mit vorgestellt.

5 Zusammenfassung und Ausblick

Kommunikationssysteme, die in der Industrieautomation eingesetzt werden sollen, müssen einerseits latenzarme Kommunikation ermöglichen als auch eine extrem hohe Zuverlässigkeit gewährleisten. Insbesondere im Bereich von Motion Control sind die technischen Anforderungen besonders hoch. Existierende Funkkommunikationssysteme nach dem Stand der Technik können diese Anforderungen für viele Anwendungsfälle nicht vollumfänglich erfüllen.

Die Anforderung einer niedriglatenten, hochrobusten Kommunikation bildete das technologische Design Ziel für die Entwicklung der Echtzeit-Funktechnologie UWIN. Durch die Umsetzung unterschiedlicher Maßnahmen zur Steigerung der Robustheit kann UWIN Paketverlustraten im Bereich von 10^{-8} erreichen und mit isochronen Zykluszeiten von bis zu 125 µs eine Vielzahl anspruchsvoller industrieller Anwendungsfälle (insb. Motion Control) abdecken.

Die Simulationsergebnisse vermitteln einen ersten Eindruck davon, welches zusätzliche Verbesserungspotential die Funktechnologie UWIN durch den Einsatz von NCC entfalten kann. Die Redundanz, die durch die Kombination mehrerer Pakete entsteht, kann besonders effizient genutzt werden. Deshalb ermöglicht bereits das zusätzliche Aussenden weniger kombinierter Pakete eine signifikante Verringerung der Paketverlustraten und die Kompensation eines Totalausfalls eines Kanals. Um das ermittelte Verbesserungspotential zu untermauern sollen die im Zuge der Feldversuche aufgezeichneten Echtzeit-Kanaleigenschaften als Grundlage für weitere Simulationen genutzt werden. Anschließend wird eine Integration von NCC als eine MAC-Erweiterung in der UWIN Funktechnologie angestrebt.

6 Literaturverzeichnis

[1] I. Aktas, A. Bentkus, F. Bonanati, A. Dekorsy, C. Dombrowski, M. Doubrava, A. Golestani, F. Hofmann, M. Heidrich, S. Hiensch, R. Kays, M. Meyer, A. Müller, S. ten Brink, N. Petreska, M. Popovic, L. Rauchhaupt, A. Saad, H. Schotten, C. Wöste und I. Wolff, „Funktechnologien für Industrie 4.0," VDE ITG, Frankfurt am Main, 2017.

[2] R. Prior, „Systematic Network Coding for Packet Loss Concealment in Broadcast Distribution," in *International Conference on Information Networking (ICOIN)*, Kuala Lumpur, 2011.

[3] J. Krigslund, J. Hansen, M. Hundeboll, D. Lucani und F. Fitzek, „CORE: COPE with MORE in Wireless Meshed Networks," in *Vehicular Technology Conference*, Dresden, 2013.

[4] K. Alic und A. Svigelj, „A one-hop opportunistic network coding algorithm for wireless mesh networks," *Wireless Networks*, pp. 1007-1018, 2018.

[5] H. Seferoglu, A. Markopoulou und K. Ramakrishnan, „I2NC: Intra- and inter-session network coding for unicast flows in wireless networks," in *2011 Proceedings IEEE INFOCOM*, Shanghai, 2011.

[6] V. N. Swarmy, *Real-time Communication Systems For Automation Over Wireless: Enabling Future Interactive Tech,* Berkeley: University of California at Berkeley, 2018.

[7] S. Kafaie, Y. P. Chen, O. A. Dobre und M. H. Ahmed, „Network Coding Implementation Details: A Guidance Document," in *22nd Annual Newfoundland Electrical and Computer Engineering Conference (NECEC),* New Foundland, 2013.

[8] A. Frotzscher, U. Wetzker, M. Bauer, M. Rentschler, M. Beyer, S. Elspass und H. Klessig, „Requirements and current solutions of wireless communication in industrial automation," in *IEEE International Conference on Communications Workshops (ICC),* Sydney, 2014.

[9] D. Schulze, A. Gnatt und M. Krätzig, „Anforderungsprofile im ZDKI," 26 Oktober 2016. [Online]. Available: http://www.industrialradio.de/Publications/ZDKI-FG1_AnforderungsProfile_041116.pdf. [Zugriff am 11 April 2018].

[10] „fast automation," [Online]. Available: https://de.fast-zwanzig20.de/industrie/fast-automation/.

[11] The Institute of Electrical and Electronics Engineers (Hrsg.), „IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. IEEE Std. 1588–2002," New York 2002, ISBN 0-7381-3369-8.

http://dx.doi.org/10.25673/39562

Supporting resilience with industrial 5G systems and Industrie 4.0

Gustavo Cainelli, Lutz Rauchhaupt, Lisa Underberg
Institut für Automation und Kommunikation e.V. Magdeburg
IKT

Werner-Heisenberg-Str. 1
39106 Magdeburg
gustavo.cainelli@ifak.eu
lutz.rauchhaupt@ifak.eu
lisa.underberg@ifak.eu

Abstract: In industry 4.0, production systems must be able to adapt to produce products with customer requirements. However, not only this kind of flexibility is needed. The system must also be able to adapt itself to keep working if unforeseen events occur. A system capable of this is called a resilient system. Resilience is the ability not to fail completely if a subsystem fails. This paper aims to introduce the concept of resilience in industrial 5G systems to avoid production downtime. Our approach is based on methods of the Industrie 4.0 concept in order to be able to create digital twins for 5G-networked automation systems. The digital twin, based on the Asset Administration Shell, will allow collaboration between automation and communication systems, adapting the behaviour of the process in case of unforeseen events. For this, the communication system must be considered during the development of the digital twin.

1 Introduction

Mass production, where the quantity of products that will be produced is known, does not require the automation system to have a high level of flexibility. However, one of the aspects of Industry 4.0 (I4.0) is the ability to adapt production to customer needs, without planning all the details of the production system in advance. Furthermore, in I4.0, the production system must be resilient, that is, it must adapt the process behaviour to avoids downtime in production in case of unforeseen events.

An alternative to dealing with the level of adaptability that I4.0 applications require and to make the system resilient are Cyber-Physical Systems (CPS). CPS are systems that integrate physics and computation components in a two way flow of information between them. For each physical device, a Digital Twin (DT) is developed. The DT is a digital representation of the real device. The Digital Twin and the physical part can exchange data for real-time awareness, control and decision making [SSPE16].

As the (wireless) communication system (WCS) is a subsystem of the automation system, it also must be considered during the development of the digital twin. Moreover, the communication system can't be planned and installed for the worst case. If changes are necessary, the WCS and/or its use of the medium should be adapted accordingly. This can only be done when data exchange is possible between the production system and the WCS. For this reason, the WCS must be also considered when modeling the Digital Twin.

The 5G-ACIA [5G-21] and Platform Industrie 4.0 (PI4.0) [Pla20b] have already started the discussion of the Digital Twin of the WCS considering 5G technology. The 5G technology is one of the most prominent communication system that aims to meet the industrial automation requirements. To allow the 5G system negotiate with the automation system, the properties of 5G devices and

5G network should also be considered in the digital model. The goal is bringing the information of the WCS to the digital domain to allow negotiation with the automation system.
This paper aims to introduce the concept of resilience in industrial 5G systems to avoid production downtime. The idea of the proposed paper has its origins in the following considerations:

- The current rigid planning and design of production facilities and their automation will give way to more flexible concepts such as Industrie 4.0 (I4.0) in order to better meet individual customer requirements and use production capacities more efficiently.
- The orientation of mobile networking towards data communication in application areas, starting with the 5th generation, gives further momentum to the desired networking in industrial applications.
- Industrie 4.0 and 5G use the virtualization of properties and functions of system elements to achieve the desired objectives.
- The collaborative use of digitization in industry and mobile networking offers new opportunities for resource-efficient industrial production.

In this context, the proposed paper addresses a way of collaboration between automation and communication. The presented approach is based on methods of the Industrie 4.0 concept in order to be able to create digital twins for 5G-networked automation systems. The digital twin will allow collaboration between automation and communication systems, adapting the behaviour of the process in case of unforeseen events. For our approach, this means that a "communication" sub-model must be available for relevant automation components.
The remainder of this article is organized as follows. Section 2 explain the principle of resilience in the 5G context. Section 3 shows the digital twin approach and the communication submodule. Finally Section 4 gives the summary and the outlook.

2 Resilience in 5G networks for industry

A resilient system should not totally fail if a subsystem fails, or operates abnormally. This means that the system must auto adjust its parameters to keep working. In this way the essential characteristics of a resilient system are [5G-17]:

- Adaptability: It enables dynamic reactions in the event of disturbances and guarantees that minimum functional requirements are met, e.g. by reducing data traffic.
- Regenerability: It should be able to seamlessly return to a stable normal state after the disturbance has ended. It can also transfer functions to other parts of the system if irreversible damage occurs, e.g. by using a base station with a lower but still acceptable quality of service.
- Maturation: It improves its functionality by using experience gained during events, e.g. by having mobile equipment avoid the zones prone to faults due to disturbances.

The principle of resilience is explained with the help of the following figures. The state of the 5G system is expressed by the up state function according to [WKR17]. In Figure 1, the up state function is represented by the light blue graph. If the 5G communication is disturbed, for example by the transport of a large metallic item with a crane through the factory hall, it is possible that not all requirements of the application for the 5G communication are fulfilled anymore. It is expressed by the fact that the up state function falls below the threshold value for the up state. The states of the 5G system represented by the dark blue lines changes from up state to down state. After the disturbance is gone, the communication can continue unaffected and the 5G communication changes to the up state when the corresponding recovery threshold has been exceeded.

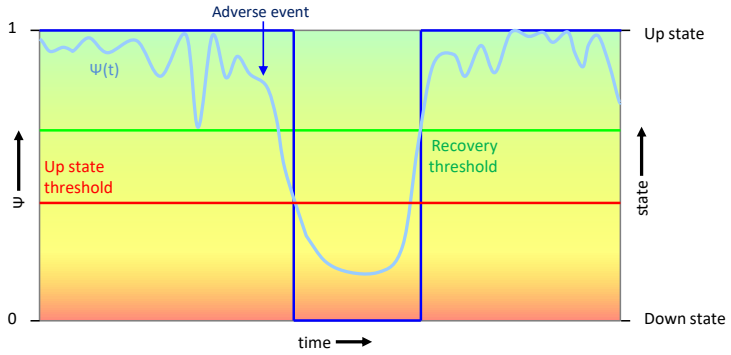

Figure 1: Up state function of a common communication system.

The up state function is determined according to [WKR17] on the one hand from measured variables that are used to specify the performance requirements of the application and on the other hand from measured variables that express the quality and use of the radio medium. In this way, it can be ensured that the application requirements are taken into account. On the other hand, the characterization of the radio medium also allows conclusions to be drawn about expected disturbances. The threshold values result from the requirement values, such as the data traffic to be handled in a certain period of time, and the values describing the conditions for 5G communication, such as the extent of the application.
In Figure 2, application requirements are reduced when an adverse event is detected. This is expressed in the lowering of the threshold for the up state. Practical examples include the termination of connections that are dispensable at that moment or the reduction of the performance of the production process. As illustrated in 2, this can prevent the application from failing due to a communication fault. When the up state function is back in the typical value range, the requirements can also be raised to the typical level again. This behaviour is called elastic response because the system reacts flexibly to adverse events.

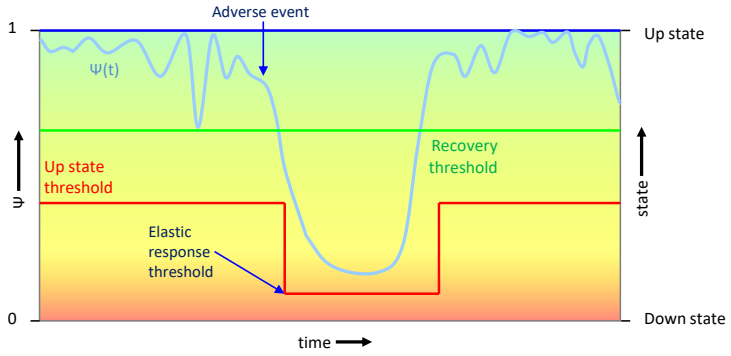

Figure 2: Elastic response of a resilient communication system.

In contrast to an elastic response of a system, a resilient system does not only react to disturbances in a fixed way, but has the ability to adapt through self-regulation. This can significantly increase the dependability of communication networks.
The behaviour of a resilient 5G system is shown in Figure 3. The adaptability is expressed by changing the threshold for the up state. The degree of adaptation depends on the identified event. Regenerability is expressed by lowering the threshold value for regeneration. Normal operation can be achieved more quickly through targeted measures. Maturation is expressed by the change in threshold values and the changed up state function. On the one hand, the values of a worst case no longer have to be assumed, since the response to adverse events can be faster and more targeted. On the other hand, measures gained through experience are expressed in a changed graph of the up state function.

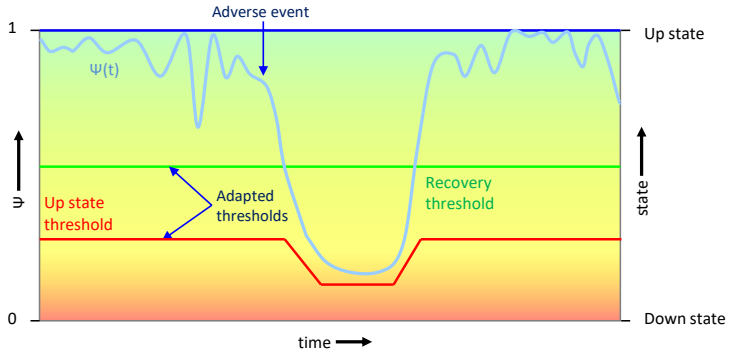

Figure 3: Adapted threshold of a resilient communication system.

3 Digital Twin Approach

3.1 Asset Administration Shell.

As stated before, the CPS can cope with the three main aspects of a resilient system: adaptability, regeneration, and maturation. One emerging category of CPS is the Asset Administration Shell (AAS). The AAS is the implementation of the Digital Twin for Industrie 4.0 from the point of view of the German initiative Platform I4.0 [Pla20b]. It represents the digital part in the Reference Architecture Model Industry 4.0 (RAMI 4.0), since the communication to business layers, [Pla18b]. The AAS contains information that represents characteristics and behaviours of an entity (asset). Assets are components that are valuable to an organisation and it includes devices, machines, documents or even software.

The AASs are key components of I4.0 as they provide all data and functions related to an asset. Through the AAS an asset become an I4.0 component. Industrie 4.0 components connect the physical and digital world robustly. An I4.0 component is formed by the asset (physical part) and the AAS (digital part) [YJL+20]. Each asset is given an AAS which consists of a number of submodels in which all the information and functionalities of a given asset including its features, capabilities, status and measurement data are described [Pla20b]. The AAS should contain information related to the complete asset life cycle (type and instances). The AAS is composed by a passive part and an active part. The passive part are the asset's data which are readable and/or modifiable. The passive part is composed by submodels that describe asset's information. The AAS may incorporate general submodels (e.g. identification) and also specific submodels (e.g. communication). There is no limit to the number of submodels of an AAS, as they are defined according to the level of detail required for the model. Below the submodel level there are the submodel elements like SubmodelCollection and Properties. The submodel elements store specific data related to the submodel. For example, a property, which is a submodel element type, can contain a value that represents a physical variable of the asset. It can be of several types as INT, BOOL or STRING. An example is the transmission power of a 5G device.

On the other hand, the active part consists of procedures and algorithms performed by the asset and the AAS. The active part is composed by methods and can be used for example to read and write properties values in the AAS/asset. Moreover, the active part of an AAS has service-oriented communication capabilities and decision making functionalities [DI20].

An integration element is necessary to connected the asset and the AAS as shown in Figure 4. This component is responsible for translate the data from the assets and update the values in the AAS.

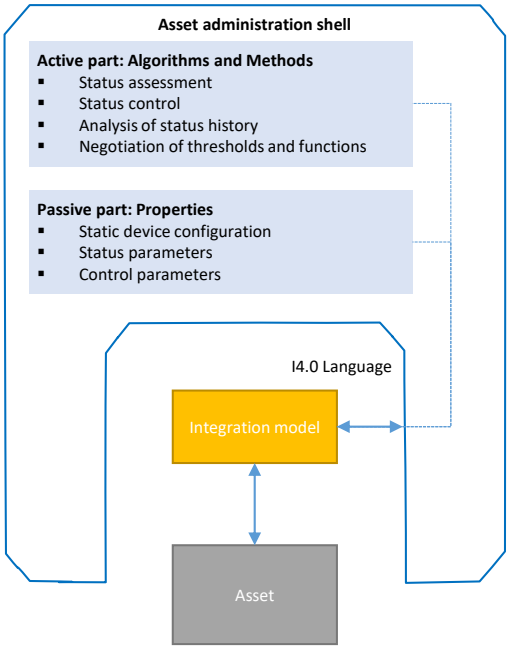

Figure 4: AAS, asset and integration model.

The information stored in the AASs are available to external users (other AASs) through external application program interfaces (APIs) using IIoT communication protocols like MQTT and OPC-UA.[5]. The AASs can communicate with each other to exchange information or to resources negotiation.
As depicted in Figure 5 the AAS of the 5G system can negotiate with the AAS of the automation system concerning the parameters of the up state function. Parameters as the up state threshold, recovery threshold and elastic response can be exchange between the AASs. Moreover, the lessons learned during a disturbance event are also shared between the AAS.

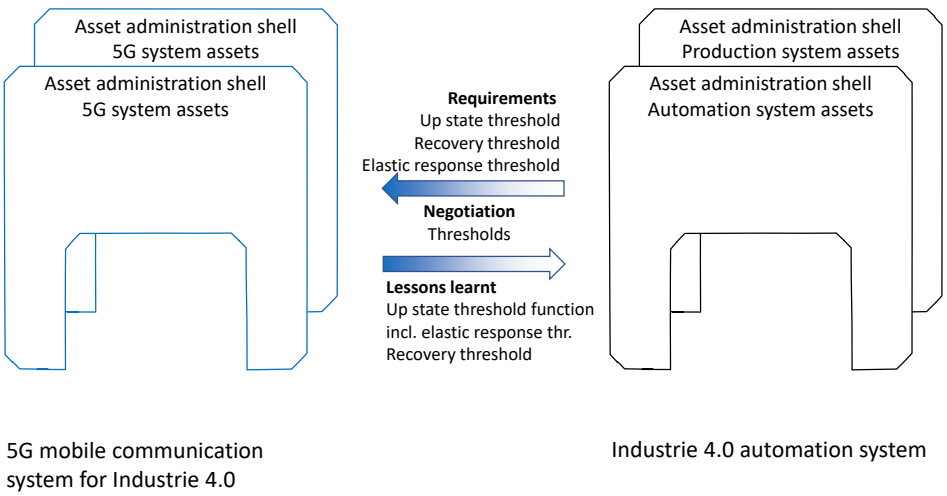

Figure 5: Negotiation between automation system and communication system (5G system).

It is necessary to develop AASs for all parties involved to allow this type of negotiation between the 5G system and the automation system. However, due to the high complexity of the 5G system and possibly the automation system, it is important to determine the necessary components in advance, otherwise, the description of all the existing components in the network may require high

and possibly unnecessary efforts. Therefore, one of the first steps is to determine what are the necessary components of the systems that should be described using AAS. In this work a submodel for 5G device is shown with the submodels that a 5G enabled device can have. Moreover, one main module is described that include the wireless module information.

3.2 Model elements of the industrial 5G digital Twin.

Describing the 5G system as an AAS (DT) is not simple due to the number of different components and also the complexity of each one. The Platform I4.0 proposes describe the 5G system as different AAS entities, e.g. for the 5G automation device (5G-UE AAS), for the Radio Access Network (5G-RAN AAS) and for the Core Network (5G-CN AAS). The 5G-ACIA [5G-21] proposes the use of two AAS called 5G-UE-AAS for 5G automation devices and 5G-NW-AAS for the entire 5G network (RAN and CN). The 5G-UE-AAS represents the endpoint of a 5G link on the device side while the 5G-NW AAS includes the nodes and functions of the 5G RAN and 5G CN. All 5G-UEs are represented by an individual AAS while the User Plane Function (UPF) is part of 5G-NW-AAS. Therefore, as mentioned previously, it is necessary to carefully define which components of the system will be described through the AAS that will allow introducing resilience in an industrial 5G system. Here in this section we describe what are the capabilites of the the active part and what are the submodels of the passive part. Some of the active parts of an AAS should include functions responsible for:

- Continuous monitoring of the network status.
- Regeneration capabilities to avoid or minimize downtimes.
- Adaptive capabilities to adjust the communication system to a state which even with disturbances the applications requirements are met.
- Negotiation of quality of service with the applications.
- Coexistence and interference management between different radio systems.

Figure 4 shows that the active part includes functions related with the up state function and the recovery response. These functions are related with the negotiation between the AAS. The passive part includes static device properties and status parameters. In case of a 5G system, examples of static properties could be data sheet related as supported bandwidth, bit rate and transmit power. Status parameters can be related with connectivity QoS as status of QoS monitoring (off/on) and characteristics parameters measurements (e.g. update time and number of received messages) [5G-21].
To structure the digital information of the asset into distinguishable parts, each AAS' submodels describe a specific asset's aspect. The submodels are formed by submodel elements hierarchically organized. There is a set of submodel elements types defined by the Plattform I4.0 [Pla18a]. These elements are used to represent data or functions related to the asset. A typical submodel element is a property. A property is a variable which has a value and a type as int, float, string, etc. Another kind of submodel element is a range, which has a minimum and a maximum value for the variable.
Figure 6 shows an AAS of an machine that has a 5G device. The figure is an excerpt of the AASX Package Explorer [Pla20a]. This is an open source tool that supports XML and JSON serialization of the AAS's data. Moreover, the tool also provides server generation for OPC UA and export formats for AutomationML. Three submodels are presented: Identification, Localization and Communication. Other possible submodels could be related to machine's functions (bending, drilling, moving, etc...). The focus of this work is on the communication submodel. Currently, within the communication submodel, one of the main submodels is the *WirelessModule*. It refers to the characteristics of the module itself considering datasheet parameters and current measurements. Other submodules like *QoS*, wich stores information related with the quality of service, should also be consider. Moreover, a submodel for the the SIM card, containing information as IMSI, ICCID, and PIN should also be defined.

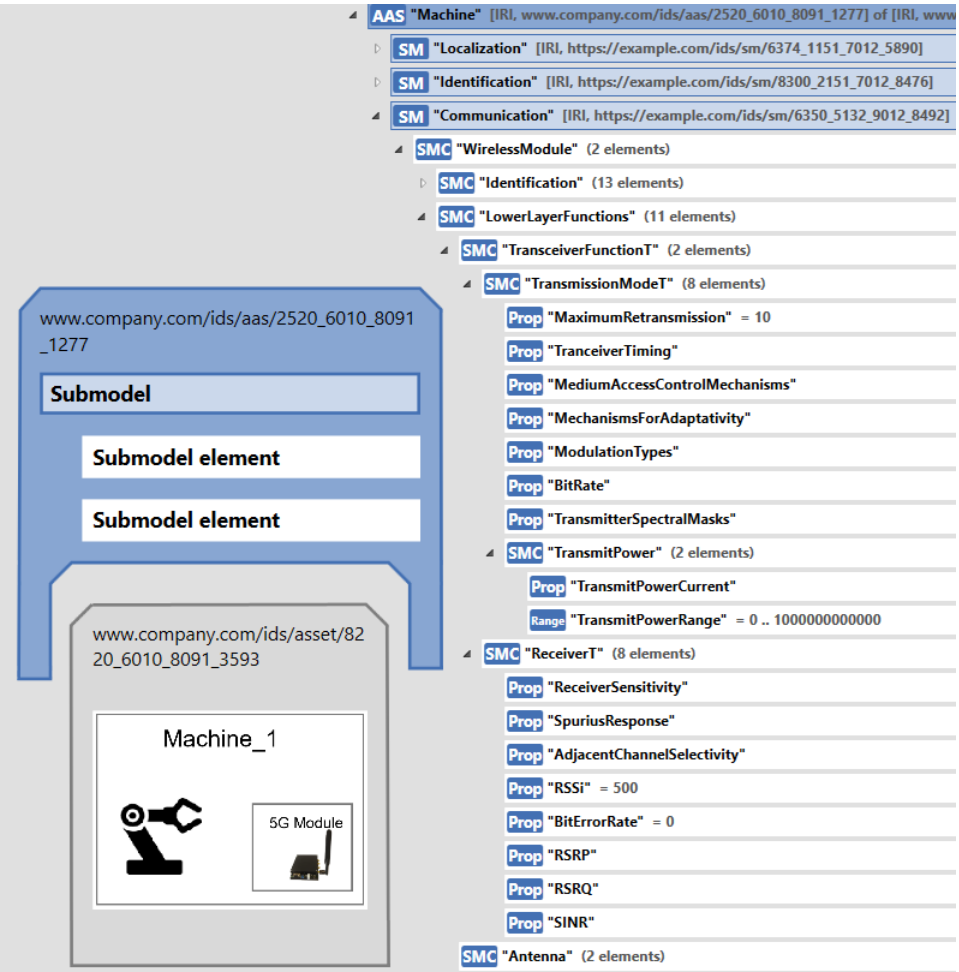

Figure 6: AAS description using the AASExplorer.

The information stored in this submodel can be divided in static and dynamic. Static ones are the capabilities of 5G module as delivered. This data generally can be find in the module's documentation (data sheets). Examples are frequency band and maximum transmit power. Information related with identification (e.g. manufacturer) are also static. The dynamic ones are updated during the operational process. It includes RSSI, RSRP and current output power.
Within the submodel *TransceiverFunction*, two submodels element collection are defined: *Transmiter* and *Receiver*. Under this two elements are properties as bitrate, modulation type and receiver sensitivity. A management system can use information from the AAS related to the wireless signal as RSSI and RSRP to monitor the conditions of the radio channel. The management system can take actions based on these values.

4 Summary and further steps

This work presented a digital twin approach to support resilience in industrial 5G systems. The submodule *WirelessModule* brings information of the wireless module to the digital domain. It includes static information as manufacturer name and dynamic information as RSSI. With these information, the system can be adapted in case of unforeseen events happens increasing the system resilience.
The authors are working in a simulation model using the Simu5G [NSS20] simulator. In this model, the resilience use case presented here is being implemented. This implementation will allow the authors to validate the current approach.
The approaches of the I4.0 platform and the 5G ACIA are to be validated and further developed. The development of the digital twins will follow the asset administration shell concepts. The authors plan to bring the results of the approach presented here into the 5G-ACIA with their

activities.

Bibliography

[5G-17] 5G-ACIA. Resiliente Netze mit Funkzugang (Position Paper). Technical report, 2017. www.shop.vde.com/de/vde-positionspapier-resilientenetze-mit-funkzugang.

[5G-21] 5G-ACIA. Integration of 5G into production networks by means of digital twin (White Paper). Technical report, 2021. www.5g-acia.org/publications/.

[DI20] Andreas Deuter and Sebastian Imort. Plm/alm integration with the asset administration shell. *Procedia Manufacturing*, 52:234–240, 01 2020.

[NSS20] G. Nardini, G. Stea, and D. Sabella. Simu5g: A system-level simulator for 5g networks. In *In Proceedings of the 10th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2020)*, pages 68–80, 2020.

[Pla18a] Plattform I4.0. Details of the asset administration shell - from idea to implementation, 2018.

[Pla18b] Plattform I4.0. Reference architectural model industrie 4.0 (rami4.0) - an introduction, 2018.

[Pla20a] Platform I4.0. Aasx package explorer, 2020.

[Pla20b] Plattform I4.0. Digital twins catalyst reflections from digital transformation world, 2020.

[SSPE16] Greyce N. Schroeder, Charles Steinmetz, Carlos E. Pereira, and Danubia B. Espindola. Digital twin data modeling with automationml and a communication methodology for data exchange. *IFAC-PapersOnLine*, 49(30):12 – 17, 2016. 4th IFAC Symposium on Telematics Applications TA 2016.

[WKR17] Sarah Willmann, Marko Krätzig, and Lutz Rauchhaupt. Methodology for holistic assessment of dependability in wireless automation. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4, 2017.

[YJL+20] Xun Ye, Junhui Jiang, Changdae Lee, Namhyeok Kim, Mengmeng Yu, and Seung Ho Hong. Toward the plug-and-produce capability for industry 4.0: An asset administration shell approach. *IEEE Industrial Electronics Magazine*, 14(4):146–157, 2020.

http://dx.doi.org/10.25673/39563

Timing Influencing Aspects of Industrial Applications

Steven Dietrich, Ludwig Leurs, Maximilian Schüngel
Bosch Rexroth AG
{steven.dietrich, ludwig.leurs, maximilian.schuengel}@boschrexroth.de

Abstract: Industry 4.0 offers not only new opportunities but also additional challenges for industrial applications and their communication systems. This requires a detailed knowledge of corresponding timing requirements to optimally use new technologies such as Time-Sensitive Networking (TSN) and the fifth generation cellular network technology (5G). Therefore, we provide an overview on application related timing aspects and implementation related constraints. Finally, we match application requirements towards network properties, in order to give a more detailed understanding on various timing aspects in industrial communication.

1 Introduction

Industrial applications require a communication network that fully covers their corresponding needs, such as determinism, short transmission latencies and reliability [D+17]. Increasing demands regarding flexibility and mobility require for new technologies, such as Time-Sensitive Networking (TSN) or the fifth generation cellular network technology (5G). These offer novel possibilities of data exchange but may need additional optimization and adaptation concerning the corresponding applications.
However, the data transmission models of today's typical Industrial Ethernet (IE) or fieldbus systems are only defined to the transmission mechanism, cf. [PRO15, ser13, ODV17]. They are very protocol specific and do not fully relate the corresponding manufacturing requirements. This makes it difficult to map the new communication technologies to this transmission models and to benefit from the novel possibilities.
Therefore, the goal of this work is to further analyze and classify timing constraints in industrial networks allowing a more detailed comparison towards the corresponding application requirements. The remainder of this work is structured as follows:
In Sec. 2, we present an overview of related work, followed by an overview of timing constraints we derive from the industrial applications point of view in Sec. 3. In Sec. 4, we analyze current protocol related timing aspects. In Sec. 5, we compare the corresponding timing constraints and relate them towards the data transmission models. Additionally, we derive possibilities for an application specific optimization, which lead to new degrees of freedom for the data transmission that could be used by new communication technologies. Finally, in Sec. 6, we draw the conclusion.

2 Related Work

Timing constraints of industrial communication are related on application based requirements or on transmission based conditions. The latter are derived from the evolutions of industrial communication systems, starting from field bus systems to IE based networks [Sau10]. The use of IE allows a better information exchange across all levels of the automation pyramid. However, the lower layer implementations of the typically used protocols differ in parts very strongly. This results in very different mechanisms for synchronization and the derivation of the corresponding temporal behavior. A comparison and classification of typically used IE protocols is for example presented in [J+07, Dec09].

In order to derive application based timing constraints, applications are typically first divided into different classes. One of the most common classification is the division into the application classes condition monitoring, process automation and factory automation [D+17, GH13, VDI09, ZVE09], which also classifies to different timing aspects. The corresponding data are typically transmitted as cyclic real-time data, acyclic real-time data, and non real-time data [PN09]. This already allows deriving certain requirements towards the real-time capability of the data exchange. Here, a distinction is typically made between non real-time, soft real-time, hard real-time and isochronous real-time capable data transmission [PN09]. More specific transmission requirements are very application related. Some specific application requirements are described for example in [D+17, F+14, VDI09, 5G 20].

Nevertheless, the actual requirements of the application and the implementation-related requirements are still frequently mixed here. Therefore, we will look at these explicitly in individual cases in the following.

3 Application Related Timing Aspects

Industrial applications usually require the exchange of data between a controller, the master, and its sensors and actuator, the slaves. This data exchange is typically referred to a downlink and uplink communication. A communication cycle therefore contains at least the transmission of control data from the master in downlink direction and actual data from the slaves in uplink direction. For synchronized applications, a so-called global sampling point (GSP) typically serves as temporal reference. It usually occurs once per communication cycle with a static relation to the beginning of the communication cycle. It defines when the slaves have to activate their command data and capture their actual data.

The structure of the actual communication cycle and control-loop depends on the application. In [Die21] this is discussed from the controller's point of view. From this perspective, there are four different timing methods possible:

- command data optimized timing,
- actual data optimized timing,
- closed-loop optimized timing, or
- cycle time optimized timing.

In order to generalize the naming, we change the definition here to the network view which interchanges command and atual data view compared to [Die21].

In the command data optimized timing, the communication cycle is designed such

that the device can get the latest command data with minimal latency. Additionally the devices are synchronized to the contoller cycle with an offset to the cycle start, so that all devices can use their command values at the same point in time (GSP). This is useful in applications where the actuators need to follow the command values very fast.
The opposite case is the actual data optimized timing. Here the controller needs the latest actual data from the slaves. Additionally to the communication time the sample points (GSP) offset can be optimized to minimize the overall time between sampling in the devices and usage in the controller. This is especially important for fast processes attached locally at the controller.
In the closed-loop optimized timing both, command and actual data, must always be based on the latest actual data. Therefore, sufficient time for processing must be scheduled for both master and slaves. As a result, the cycle time is longer compared to the other timing variants considering the same setup. Using shorter communication cycles for closed loop will be considered in Sec. 4.
The shortest network cycle time could be achieved with the cycle time optimized timing. Here, neither the command data nor the actual data have to be based on latest generated data. Therefore, it is only possible to react to changed values with at least one cycle offset. Due to the correspondingly shorter cycle time, appropriate control can still be achieved.
For this consideration, it is assumed that all data are equivalent in terms of timing requirements. This is a typical assumption of today's industrial application, which saves overhead for additional signaling channels and guarantees a fast transmission using the existing IE based networks. However, from an application point of view, a further differentiation might be beneficial, since not all data actually have the same timing requirements.
Form the applications point of view, we can distinguish between:

- open-loop control,
- local-loop control, and
- closed-loop control.

In open-loop control, the control action is decoupled from the process output. It is based on the calculation of command data while taking the actual data without high real-time requirements for system state calculation only. Therefore, either the communication cycle needs to be faster than the response time of the process to the control data or the system state is stable in time and delayed transmission does not lead to different behavior. Typical examples are on/off switching of motors, heaters or lights over a certain time, where a sufficient control result could be assumed for the process without a need for feedback. The advantage of using open-loop control is the reduction of control complexity and required components for the control loop. It allows individualized communication cycles that correspond to the calculation of new command data, the transmission and activation of these data at the slaves. Here the command data do not have to be based on latest actual data, which allows a further reduction of the network cycle time. Typical programming might be a sequential flow chart (SFC) using states and transitions and enabling easy diagnostics through missing signals necessary for the next step.
Local loop control is used to decouple fast sub-processes from the application controller. It allows to eliminate communication time counting as dead time in the closed

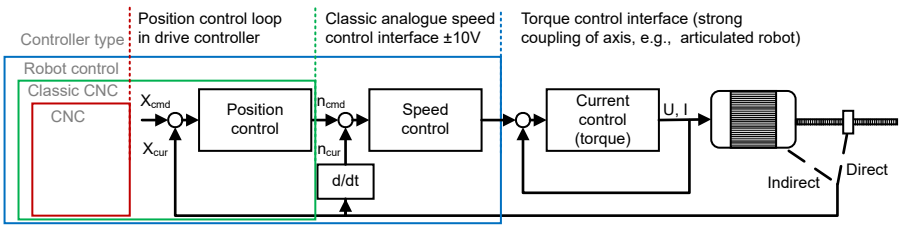

Figure 1: Cascaded control structure.

loop control. Typical examples are drive controller in position set point mode. The command values are target position and start signal plus optional maximum of speed, acceleration and jerk. Motion planner and closed loop control functions are included. In closed loop control the cycle time must be fast enough to capture the upper frequency of the process by fulfilling the Shannon/Nyquist theorem. As we see later in Sec. 4, there are two strategies: one that optimizes the overall loop cycle time and the other that minimizes the network cycle time.

Closed loop control needs synchronization of devices in the system and also of the data transmission. As described before, most applications implement a so called global sampling point (GSP) as temporal reference to activate the command data and capture the actual data. Alignment of the local clocks is either done by additional messages with corresponding synchronization information (e.g. IEEE 1588) or by the encapsulated of these information in the regular cyclic real-time data transmission (e.g. Sercos III). Both might have individual advantages for the timing behavior of the data transmission (cf. Sec. 4).

Based on this classification, it is possible to optimize data transmission for a specific application. One method to represent industrial communication for such an optimization is automata theory. Here, the manufacturing processes or machines are represented as finite automata with a series of states and transitions. A mapping of real-time systems is shown for example in [Pet99].

A main field for synchronized devices is motion control applications. For explanation we consider the control structure of a drive together with its motion controller, computerized numerical control (CNC) or robot control (RC), examplified in Fig. 1.

For Cartesian systems like most machine tools, the axes are independent from each other and can be controlled locally in the drive controller. The CNC interpolates the motion path and sends each position command value to the respective axis. The current position value is only needed for monitoring purpose in the CNC and is not very critical in timely transmission. This constellation can be regarded as open loop control for the CNC and local loop control for the drive controller. Classical CNC systems integrate the position control loop, transfer the speed command value to the speed control loop and need the position feedback value for closed loop operation. As the inner control loops need to be faster, this results to a time critical communication.

For non-cartesian systems like articulated robots, a simple position control loop in the drive does not work because of disturbing momentum of other axes. As the RC

knows the robots kinematic, it is common to use the torque control interface. This leads to even shorter cycle times (typically 125 µs), simplifies the drive controller, but needs a powerful RC. For interoperability of motion controller and drive controller, the interface needs to be standardized. This was done by the fieldbus organizations in regard of the communication aspects by defining appropriate device profiles.
These need to consider timing from the controller application, the network transport and device control loops.

4 Implementation Related Timing Aspects

When looking at different devices profiles for synchronized motion we discover that there are 1- cycle, 2-cycle and even multi-cycles communication models. In this chapter, we describe the background of these models and assign corresponding applications.
Closed loop control always targets to minimize latency inside the loop. In old analogue implementations direct wiring for velocity and feedback position was used. For cost reason serial communication came into industrial controls in the beginning of the 1990^{th}, which helped also to get rid of drift and noise problems but introduced additional latency. The first standard introduced to make the interface between CNC and drives digital was Sercos Interface. A drive controller can be regarded as three cascaded control loops for position, velocity, and torque (cf. Fig. 1). The computer-based NC contained path generation, interpolation, and position control loop. The analogue based drive controller contained velocity and torque (current) control loops. Sercos was designed to interface to any of those control loops in regard of different application requirements. Looking at the communication requirements the outer loops are slower but need higher resolution of the communicated signals. High dynamic machines require fast control loops, so the best would be to have the three loops for position, velocity, and torque running locally in the drive controller without any latency caused by communication. This works perfectly for Cartesian systems like machine tools where there is no cross-dependence between the axis. High performance could be reached by coarse interpolation in the CNC and additional fine interpolation in the drive controller. Supposing that the drive controller can guarantee the required acceleration, the feedback position to the CNC is only needed for monitoring of errors. So, the position interface is just an open loop interface for command position. (Reference to output stream optimization). The cycle time in this case is determined by the minimum cycle time of the controller and the next multiple of control loop cycle time of the drives to keep controller and drives in synchronization, cf. Fig. 2.
For closed loop control it is usual advantageous to choose a 1-cycle model. That would calculate the cycle time by adding the maximum controller calculation time, maximum transport times for sending outputs and inputs including preparing in controller and device. This example is shown in CIP[1] Motion specification, cf. Fig. 2.
The controlling task in the controller would receive the current feedback value, calculate the new command value and send this to the device. The cycle time needs to be integer multiples of the drive control loop time. A common suggestion would be

[1] Common Industrial Protocol

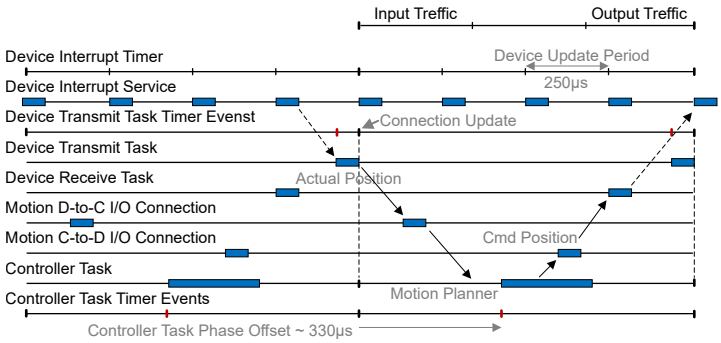

Figure 2: CIP Motion 1-Cycle Timing Model [ODV17].

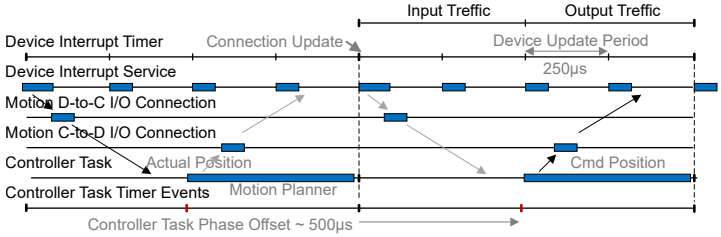

Figure 3: CIP Motion 2-Cycle Timing Model [ODV17].

to divide the cycle time by three for equal parts of calculation time and transport times. This would leave $2/3^{rd}$ of transport and calculation capacity for other purpose supposing full duplex communication.
A 2-cycle timing model would allow a bigger part of the cycle being used for the controller task, cf. Fig. 3.
The drawback is that the command values of the drives are calculated on the feedback values of an earlier cycle. This results in interlaced calculations and requires that the frequency of the signals is below half the frequency of the cycle.
The 3-cycle timing model allows controller calculation and both direction of transport in parallel and spans input, calculation, and output over three cycle. The model is running three interlaced control loops each cycle and therefore could in an extreme case use all resources for computing and transport task, cf. Fig. 4.
The ProfiDrive profile is defined on Profibus/Profinet. As Profibus is a half-duplex system, send and receive directions are not independent and coupled as request and confirmation/response (cf. DX for data exchange in Fig. 5).
A non-optimized multi-cycle timing is shown in Fig. 6.
The inputs can only be used in the next controller cycle and the outputs can only be sent and activated in the following communication cycle. The profile defines three or more communication cycles as the standard for Profidrive. If the Profibus Master is

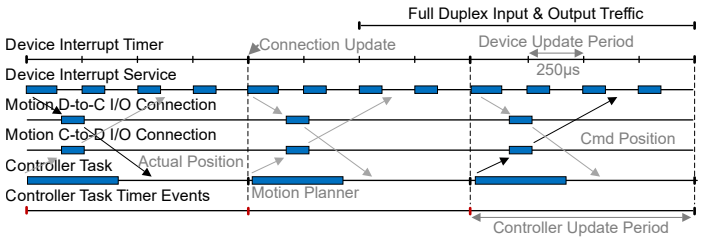

Figure 4: CIP Motion 3-Cycle Timing Model [ODV17].

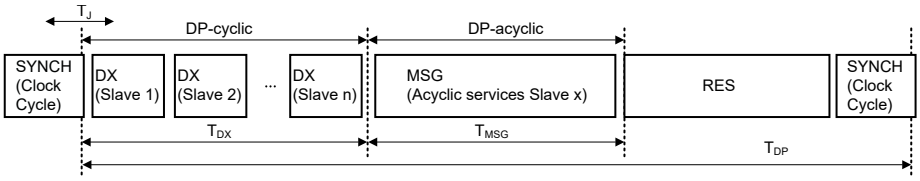

Figure 5: Sequence of an isochronous DP cycle [PRO15].

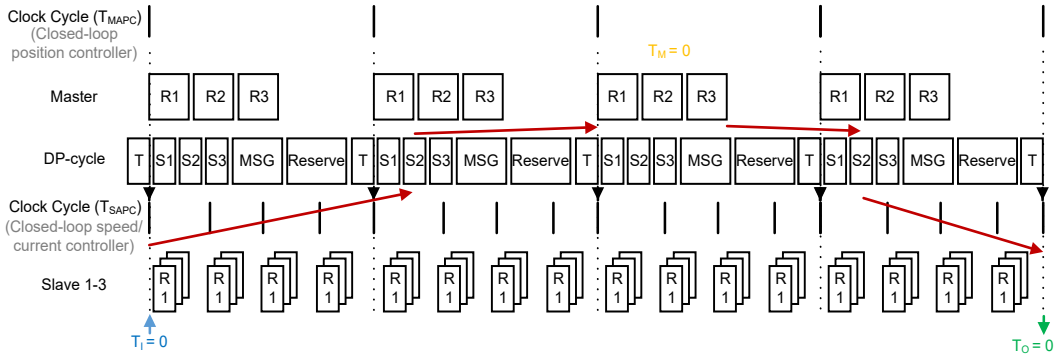

Figure 6: Simple DP cycle [PRO15].

tightly coupled to the control program and the control loops can return their results before the next communication cycle starts, the 2-cycle model is possible, cf. Fig. 7. Profinet is based on Standard Ethernet and uses full-duplex operation, but the 3-cycle model is kept as standard as well. Profinet is working directly on Layer 2 Ethernet frames and also optimizes transport time by dynamic frame packaging (DFP) and cut-through forwarding. The communication cycle time can be lower than in UDP based systems like EtherNet/IP and might compensate for this.
Sercos was the first motion control system and based on TDMA. For synchronization originally a separated telegram was used. With Sercos III this telegram was included

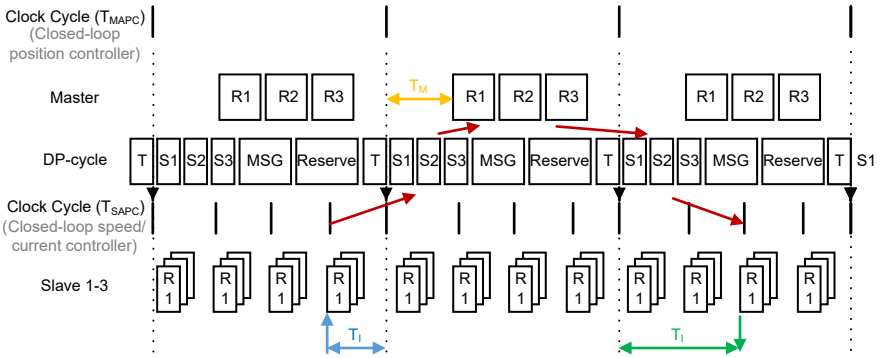

Figure 7: Optimized DP cycle [PRO15].

as a pattern in the first Ethernet frame of the master to device data telegram (MDT0) for data efficiency. The timing defines a real-time channel (RTC) and a channel for universal communication (UCC). Inside the RTC the order of data transport direction can be chosen. This enables optimization of control loops and the latency of command and feedback values. MDT first and AT second enables a 1-cycle timing model. For position control loop locally closed in the drive, the controller internal timing can be aligned to the Sercos timing in both orders of data directions.

5 Matching

After having shown the different applications and their timing requirements together with implementations in the typical fieldbus systems, we want to classify and match the applications to the network properties, as presented in Tab. 1.

Table 1: Matching of application requirements and network properties.

		Applications				
		PLC	Sensor	Drives		CNC/RC
				Position setpoint	Closed-loop control	
Communication	Not synchonized	X[1]	X[1]	X[1]		X[2]
	Synchronized		X[1]		X	X

[1] Optimization possible using asymetric data rate
[2] Only for the integrated PLC

Not synchronized applications like PLC use unsynchronized networks like Profinet RT (Profibus DPV1), Ethernet/IP (DeviceNet). The effect is that at each interface along the data transport an additional buffer is needed and as a worst case a full cycle delay is introduced. The user's approach to reduce the delay and the system reaction time is to set the network cycle time to half of the application cycle time, which generates the requirement for fast network cycle times to the system and device manufacturers. From the device side, most sensors signals are not very time critical and used unsynchronized. Intelligent drives using the setpoint interface include fast closed loop control at local level and can be used with PLC applications for improving precision and simplifying application programming. As devices generate signal of different frequency the data rate could be adjusted by poll/transmit frequency. Especially Sensors are in the first place only transmitting and need an answer only for keep alive of the connection.
Synchronized Applications like CNC/RC need synchronized devices like drives or position feedback sensors in closed loop control mode as well as unsynchronized sensors and actors for their integrated PLC. Modern communication systems based on Ethernet support both synchronized and unsynchronized data transfer in one network. State of the art IE implements this in a more or less proprietary way to each system (Sercos III, EtherCAT, PowerLink, Profinet IRT) or live with higher network cycle time and shift functionality to the device (EtherNet/IP with CIP Sync and CIP Motion). Standardization by IEC/IEEE60802 enable IE to migrate to IEEE802.1 standards like time synchronization (.1AS), quality of service (.1Q) for scheduling (.Qbv), and priorization. Due to synchronization the delay in data transfer can be minimized and buffer resources as well.

6 Summary

In this work, we analyzed and classified timing constraints in industrial networks allowing a more detailed comparison towards the corresponding application requirements. We provided an overview of related work classifying industrial networks and deriving timing requirements. For a specification of application based timing aspects, we compared different timing methods and identified timing constraints of different control structures. We continued with an overview on implementation related timing aspects, investigating typically used Industrial Ethernet (IE) protocols. We compared different cycle models and related them to feasible application control structures. Finally, we concluded our specification with a matching of application requirements towards network properties, allowing a more detailed understanding on various timing constraints in industrial communication.

Bibliography

[5G 20] 5G Alliance for Connected Industries and Automation (5G-ACIA). Key 5G Use Cases and Requirement. Technical report, May 2020.

[D+17] Steven Dietrich et al. Performance indicators and use case analysis for wireless networks in factory automation. In *IEEE International Conference*

on Emerging Technologies And Factory Automation (ETFA'2017). IEEE, September 2017.

[Dec09] Jean-Dominique Decotignie. The many faces of industrial ethernet [past and present]. *IEEE Industrial Electronics Magazine*, 3(1):8–19, 2009.

[Die21] Steven Dietrich. *Methods of Evaluation and Improvement on Cascaded Wired and Wireless Real-Time Communication Networks for Factory Automation*. phdthesis, Brandenburgische Technische Universität Cottbus-Senftenberg, 2021. DOI: https://doi.org/10.26127/BTUOpen-5483.

[F+14] Andreas Frotzscher et al. Requirements and current solutions of wireless communication in industrial automation. In *IEEE International Conference on Communications Workshops (ICC'2014)*. IEEE, 2014.

[GH13] V Çağrı Güngör and Gerhard P Hancke. *Industrial wireless sensor networks: Applications, protocols, and standards*. CRC Press, 2013. ISBN 978-1-1380-7620-4.

[J+07] Juergen Jasperneite et al. Limits of increasing the performance of industrial ethernet protocols. In *IEEE International Conference on Emerging Technologies And Factory Automation (EFTA'2007)*. IEEE, 2007.

[ODV17] ODVA Inc. The CIP Networks Library Volume 9: CIP Motion. Edition 1.1, April 2017.

[Pet99] Paul Pettersson. *Modelling and Verification of Real-Time Systems Using Timed Automata: Theory and Practice*. phdthesis, Uppsala University, February 1999.

[PN09] Carlos E. Pereira and Peter Neumann. *Handbook of Automation*, chapter Industrial Communication Protocols, pages 981–999. Springer Berlin Heidelberg, 2009. ISBN 978-3-540-78831-7.

[PRO15] PROFIBUS Nutzerorganisation e.V. Technical Specification for PROFIBUS and PROFINET. Version 4.2, October 2015. Order No.: 3.172.

[Sau10] Thilo Sauter. The three generations of field-level networks—evolution and compatibility issues. *IEEE Transactions on Industrial Electronics*, 57(11):3585–3595, 2010.

[ser13] sercos international e.V. sercos the automation bus - Communication Specification. Version 1.3.1-1.12, December 2013.

[VDI09] VDI/VDE Society for Measurement and Automatic control (GMA). VDI/VDE 2185 Part 1: Radio based communication in industrial automation. Technical report, 2009.

[ZVE09] ZVEI Automation Division. Coexistence of wireless systems in automation technology. White Paper, 2009.

http://dx.doi.org/10.25673/39568

Investigating the Inter-Domain Forwarding Offset in the Context of Dynamic End-to-End Stream Reservation in Multi-Domain Time Sensitive Networks

Martin Böhm, Diederich Wermser

Research Group Communication Systems
Ostfalia University of Applied Sciences
Salzdahlumer Str. 46/48
38302 Wolfenbüttel
ma.boehm@ostfalia.de
d.wermser@ostfalia.de

Abstract: The constantly growing TSN toolbox offers runtime reconfiguration for dynamic end-to-end stream reservation. While inter-domain communication in TSN has been identified to be required for various use cases, the control plane interactions are not defined yet. This paper presents Multi-Domain Time-Sensitive Networks (MDTSNs) integrating east-westbound communication in the TSN control plane to enable on-demand multi-domain end-to-end bounded-latency stream configuration. An inter domain forwarding offset (IDFO) has been identified to emerge when setting up streams in MDTSN. This paper investigates the IDFO, presents control plane mitigation mechanisms which are implemented and evaluated in a physical MDTSN test environment.

1 Introduction

Inter-domain communication for IEEE 802.1 Time-Sensitive Networking (TSN) has been identified to be required for various use cases gathered by the IEC/IEEE 60802 TSN Profile for Industrial Automation [TSIA] (TSN-IA Profile). E.g., one use case is described [IAUC] where preconfigured machines with tested and approved communication, require communication with other preconfigured machines located in different TSN-domains. So far, the TSN-Toolbox offers features working in single TSN domain i.e., within the domain boundaries. Towards the future of industrial automation networks, infrastructure components will be deployed and changed on demand at runtime [WSJ17]. This requires automation networks to provide dynamic reconfiguration of communication.

The IEEE 802.1Qcc [IEQC] standard offers runtime reconfiguration allowing users to specify stream requirements. The standard introduces three different configuration models for TSN. The *fully centralized* model, shown in Figure 1, which is in focus of this work, introduces a Centralized User Configuration (CUC) which handles end station stream requests and end station TSN feature configuration, while a Centralized Network Configuration (CNC) performs traffic scheduling and network configuration for TSN bridges. Applying the separation between the data plane and the control plane to TSN, as inherited from Software Defined Networking (SDN), the CUC and the CNC are part of the TSN control plane, while end stations and TSN bridges are part of the TSN data plane. As pointed out by Schriegel et al., for the Industrial Internet of Things (IIoT), the information technology (IT) and operational technology (OT) must coalesce. This requires an interface between the networks' control plane [SKJ18]. Such an interface, for the interaction between different TSN domains, has not been addressed by TSN standards yet.

In [BW21] we presented control plane mechanisms for MDTSN integrating an east-westbound protocol in the existing TSN control plane, achieving multi-domain on-demand end-to-end bounded-latency TSN stream configuration in the fully centralized model for unidirectional periodic traffic. In this work, we present a multi-domain use case, where different Manufacturing as a Service (Maas) provider, with machines located in different TSN domains have to cooperate with each other i.e., require streams between different TSN domains. We present control plane mechanisms for the integration of east-westbound communication in the MDTSN control plane. A main challenge in this context is the inter-domain forwarding offset (IDFO) on the MDTSN data plane, which will investigated in detail. Control plane mechanisms to mitigate the IDFO are presented and evaluated in an MDTSN test environment.

The paper is structured as follows: first, Chapter 2 presents TSN in the context of dynamic stream reservation. Use cases for MDTSN are presented in Chapter 3. Related work is presented in Chapter 4. Chapter 5 presents MDTSN and control plane mechanisms for dynamic stream reservation in MDTSN. Chapter 6 investigates the IDFO, as part of MDTSN. Control plane mechanisms to mitigate the IDFO are presented in Chapter 7. These

mechanisms are tested and evaluated in an MDTSN test environment, as presented in Chapter 8. Chapter 9 concludes paper and identifies future work.

2 Time Sensitive Networking

Within IEEE 802.1, enhancements for Ethernet based networks are specified. These enhancements in particular provide real-time capabilities and can be combined in different ways. A bounded latency, low jitter and low packet loss can be achieved by the use time slots with cyclical repetition (IEEE 802.1Qbv [QBV]). This allows to grant specific traffic classes (TCs) exclusive use for the data transmission within a time slot. Traffic is assigned to different queues which open and close for a certain time, called schedule, which are specified Gate Control Lists (GCLs). A stream is an end-to-end connection in TSN, identified by, e.g., MAC addresses. A network-wide time synchronization (IEEE 802.1AS-rev [ASRe]) is required to synchronize GCLs of the devices to properly schedule traffic. The generalized precision time protocol (gPTP), which is based on the precision time protocol (PTP), is specified in the IEEE 802.1AS-rev standard in the context of TSN.

The IEEE 802.1Qcc standard introduces enhancements for the Stream Reservation Protocol (SRP) (IEEE 802.1Qat) for dynamic stream reservation. Besides performance improvements, it presents three different architectural models. Within the *fully distributed* model, application request streams by propagating a stream request directly over the network. In a distributed manner, each TSN bridge along a path configures itself with the communication parameters of the stream request. The second presented model is the *centralized network/distributed user* model. Here, a central entity called Centralized Network Configuration (CNC) is introduced. Stream requests are still propagated over the network, while the first bridge forwards the request to the CNC. For scheduled traffic (IEEE 802.1Qbv), the CNC, with a centralized view on the network, takes care of path finding, traffic scheduling and configures all related TSN bridges affected by the stream request. For more complex use cases, where end stations (talker and listener) also require configuration, the *fully centralized* model, shown in Figure 1, introduces a Centralized User Configuration (CUC). The CUC discovers end stations and their capabilities. Streams are requested directly at the CUC. The CUC communicates the stream requests with the CNC. Furthermore, the CNC provides configurations for the end stations, configured by the CUC. The communication interface between the CUC and the CNC, called user network interface (UNI), is specified by the IEEE while the communication between the end stations and the CUC is application specific. For example, the OPC Foundation specified their PubSub architecture to be compatible with TSN [Fou16]. They also specified a CUC called *PubSub TSN Configuration Broker* (PTCB) [Fou17].

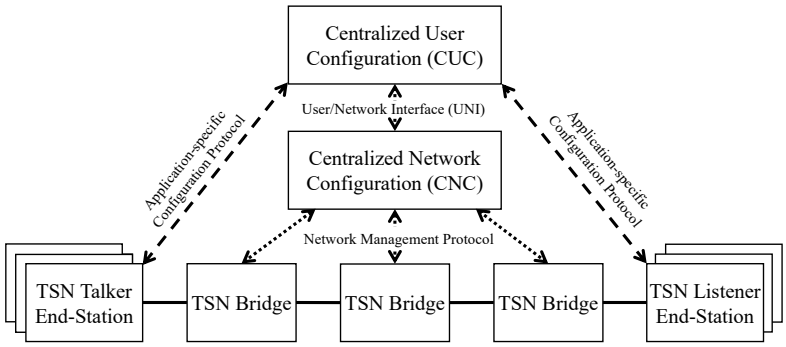

Figure 1: TSN in the fully centralized model

In the TSN-IA Profile an Industrial Automation Management Entity (IA-ME) is presented, which is adapted from the fully centralized model to serve industrial automation use cases and partitions the CUC and the CNC for better understanding. The IA-ME introduces new entities besides the already specified CUC and the CNC. A Topology Discovery Engine (TDE), an Industrial Automation Path Establishment Entity (IA-PE), responsible for path management, and a Best Management Entity Algorithm (BMEA) for the management of multiple IA-ME for e.g., failover, are added to the IA-ME. Within IA-ME, end stations request streams at a Query Stream Server (QSS) which handles the communication with the CUC. The authors of the profile note, that the TDE and the IA-PE could be considered as part of the CNC as well as the QSS could be part of the CUC. This paper assumes the architecture described in IEEE 802.1Qcc.

2.1 Stream Parameter

In a time-sensitive stream request, as described in IEEE 802.1Qcc, there are lots of control plane (communication) parameters involved. The most relevant parameters are further described. Parameters for time awareness are specified within a network cycle, which is a repetitive time interval used in the network for e.g., opening and closing gates of queues. In a stream request from talker to the CUC, the talker specifies the following parameters.

- Source and destination MAC address: MAC address of the talker and the listener of a requested stream.
- Interval: Interval for transmitting data (e.g., 125µs). Usually within predefined traffic classes.
- Maximum frames per interval: Maximum number of frames that will be sent during the interval.
- Maximum frame size: Maximum frame size sent by the talker for the requested stream.
- Earliest Transmit Offset (ETO): Earliest offset within the network cycle where talker is capable of transmitting data.
- Latest Transmit Offset (LTO): Latest offset within the network cycle where the talker is capable of transmitting data.
- Maximum Latency: Maximum latency from talker to listener for a single frame.

After a successful stream request, in the talker configuration, a *Transmit Offset* (also called time aware offset) is specified by the CNC. The value is between the requested ETO and LTO. It specifies a point of time within the network cycle for the transmission of the first frame of the requested stream.

The listener configuration includes an accumulated latency, which specifies a point of time within the network cycle, when the first frame of the configured stream arrives. It is calculated by the CNC by adding up the configured bridge delays and the propagation delays of each bridge along the stream path.

2.2 Industrial Traffic Types

The TSN-IA Profile specifies traffic types for different functionalities. They have different characteristics and requirements on the network. Table 1 depicts a subset of the traffic types. The table shows the name of the traffic type, the periodicity (periodic or sporadic) and the typical period of the application cycle (data transmission interval). Furthermore, it specifies the requirements for the data delivery guarantee. This can be for example a required maximum latency, bandwidth, or no guarantees at all. Some traffic types require the application to be synchronized to the network clock for e.g., scheduled traffic. The table also specifies the typical application data sizes and the criticality.

This work focusses on the most critical traffic types, isochronous and cyclic-synchronous. Isochronous traffic, typically used for control-loop communication, is required to be synchronized to the network clock to reduce jitter, which requires this traffic type to not interfere with other traffic. Messages may be discarded when delivered too late. Cyclic-synchronous traffic has slightly less strict communication requirements compared to isochronous traffic. Its period with slightly larger and it can handle a small jitter.

Table 1: Industrial automation traffic types [IEE, Ind]

Traffic type name	**Periodicity**	**Typical period**	**Data delivery guarantee**	**Synchronized to network**	**Typical application data sizes (Bytes)**	**Criticality**
Isochronous	Periodic	100µs-2ms	~1 application cycle	Yes	Fixed: 30-100	High
Cyclic-Synchronous	Periodic	500µs-1ms	~1 application cycle	Yes	Fixed: 50-1000	High

Cyclic-Asynchronous	Periodic	2ms-20ms	~½ application cycle	No	Fixed: 50-1000	High
Alarms and Events	Sporadic	N/A	100ms-1s	No	Variable: 50-1500	High
…						
Best Effort	Sporadic	N/A	None	No	Variable: 30-1500	High

3 Use Cases for TSN Inter-Domain Communication

In [IAUC], over 35 use cases are specified by the IEC/IEEE 60802 project. Several use cases have been identified, which require inter-domain communication within TSN. E.g., one use case focussing on redundancy is described, where multiple TSN-domains are interconnected via a ring topology. Another use case for machine to machine (M2M) communication and controller to controller (C2C) communication is described for inter-domain communication. Machines are grouped and located in their own TSN domains, because they e.g., run different complex schedules with different network cycles. A communication between machines located in other TSN domains is required. Production cells, which connect the TSN domains of the machines, are also located in their own TSN domains. These production cells are interconnected in a production line TSN domain while the Operation Control HMI is also located in its own TSN domain. New devices, e.g., automatic guided vehicles (AGVs), are plugged/unplugged automatically, requiring communication streams on demand. Technically, the document describes, that the TSN domains share a single OSI layer 2 broadcasting domain.

Figure 2 presents our vision of the future factory, where machines of different MaaS providers have to work together. Within MaaS, manufacturing processes are outsourced to a MaaS provider. To prevent vendor lock-in, when relying on a single MaaS provider, while also benefitting on different specialities and pricing of different MaaS providers, a combination of multiple MaaS provider should be intended. Open standard solutions such as TSN and OPC UA, allow communication between machines of different vendors.

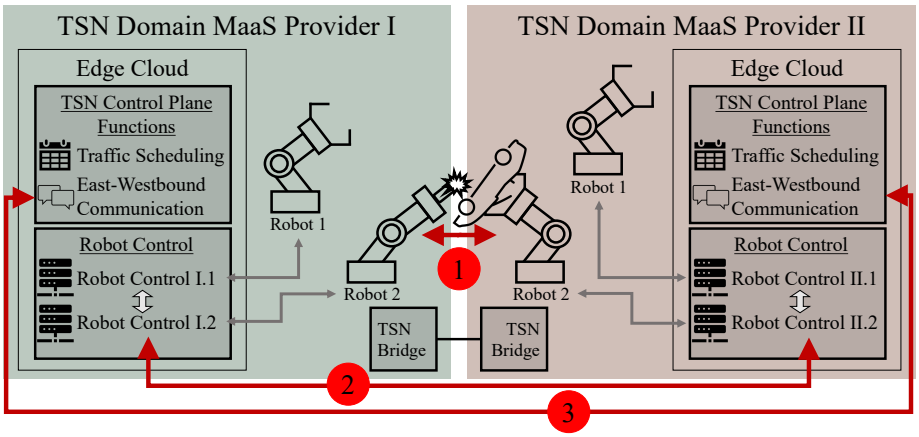

Figure 2: Cooperating robots of Manufacturing as a service (MaaS) providers in interconnected TSN domains

As visualized in Figure 2, each MaaS provider is located in its own TSN domain. A provider specific edge cloud provides TSN control plane function such as traffic scheduling as well as the robot controls. Each MaaS providers solution is part of their business secret and has to be kept confidential. The interaction for the combination of different MaaS providers is highlighted in red color in Figure 2. 1) Robots physically cooperate with each other. 2) Communication with bounded latency between the robot control systems of the providers is required. 3) Dynamic streams are negotiated and configured in the TSN control plane over an east-westbound protocol.

Our developed use case has different requirements compared to the use cases identified by the IEC/IEEE 60802 project. To preserve the internal configuration of the MaaS providers, the TSN domains cannot form an OSI layer 2 broadcast domain. Thus, topology hiding is required. Each multi-domain stream has to be requested and configured separately, as further described in this paper.

4 Related Work

Traffic scheduling, as part of the CNC, is an NP-complete problem. Depending on the size of the network and the streams, the traffic schedule calculation may require hours [CO16]. When streams are scheduled in runtime, the calculation uses the existing configurations to add new streams. Compared to a complete traffic schedule calculation, the runtime schedule calculation is faster as presented in [RP17], where the proposed heuristic scheduling approach is done within seconds. Their scheduling algorithm uses an as-soon-as-possible (ASAP) approach, where streams are scheduled at the earliest point in time, that is feasible. In the worst case, the scheduler destroys the current configuration and reschedules all flows.

In SDN, east-westbound protocols are used for the communication between distributed domain controllers to enable e.g., coordinated flow setups. While there is OpenFlow, the most common standardized southbound protocol for SDN, so far, there has been an expired IETF draft in 2012, called SDNi [SDNi], for an east-westbound protocol in SDN. The sketched protocol proposes e.g., domain-specific policies, reachability information exchange and coordinated flow setups including QoS. While there has been lots of research on east-westbound communication in SDN [ZC18], there is no standardized protocol yet.

In [SKJ18], a distributed control plane for heterogenous TSN is proposed, which requires inter-domain communication. The authors describe the idea of cascaded CNCs which perform stream configurations. Without a coordinated stream setup, which requires communication between the CNCs, the end-to-end latency increases. Mechanisms for the interaction between the CNCs are not covered.

In the IEC/IEEE 60802 project, there have been some contributions about inter-domain communication. In [Lih], an idea for a protocol between CNCs is proposed, called CCP (Config-entity to Config-entity Protocol or CNC to CNC Protocol). The authors point out, that protocol, procedures, and managed objects must be specified for CCP.

5 Multi-Domain Time-Sensitive Networks (MDTSNs)

The audio video bridging (AVB) standard (IEEE 802.1BA [IEBA]) specifies an automatic AVB domain detection mechanism. Domain boundaries are detected by a bridge when a different SRP domain on a network port is detected. This mechanism works for the fully distributed TSN configuration model. In the fully centralized TSN configuration model, TSN domains are formed by connected TSN bridges to their respective CNC.

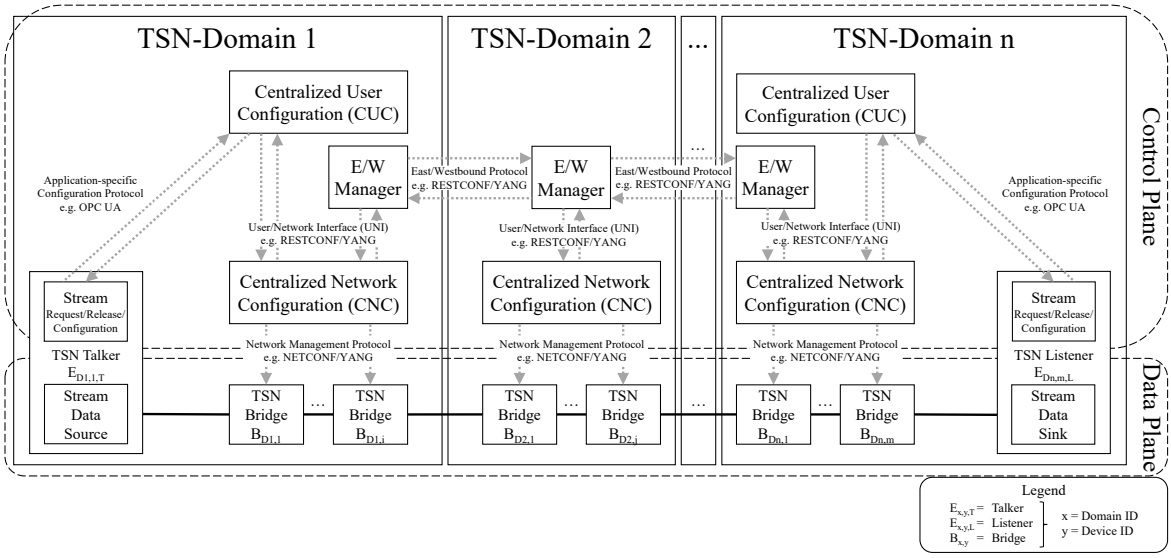

Figure 3: East-westbound communication in the multi-domain time-sensitive networks (MDTSN) control plane

In MDTSN, on-demand end-to-end TSN streams are set up between different TSN domains. Thus, the interaction on the TSN control plane must be defined. Figure 3 shows MDTSN in a horizontal, also called peer-to-peer architecture. Another architecture, not in focus of this work, is the hierarchical architecture, where control plane communication for inter domain communication is handled by a root controller, as presented for SDN in [ZC18]. The horizontal architecture has been chosen because of its advantages with respect to topology hiding. Figure 3 shows an *east/westbound manager* (E/W Manager) added in the TSN control plane enabling

inter domain communication. Compared to SDN, an east-westbound protocol in TSN requires to exchange timing information for dynamic stream configuration.

5.1 Stream Segments

In MDTSN an end-to-end stream consists of multiple stream segments, which are streams provided by each TSN domain as part of the end-to-end stream. We identified three different types of TSN domains which are involved in an MDTSN stream request. The *source domain* is the domain, where the talker end station of a stream request is located. Stream are provided between the end station and the *boundary network port*. Boundary network ports interconnect two TSN domains. The *forwarding domains* are domains, where neither the talker end-station, nor the listener end-station is located. Therefore, the forwarding domain provides a stream between boundary ports. A forwarding domains purpose may only be forwarding i.e., it has no end-stations and no CUC. Within an MDTSN stream, there may be multiple forwarding domains involved. The *destination domain* is the domain, where the listener end-station is located. Stream are provided between a boundary port and the listener end-station. Within the MDTSN stream request, timing information of that stream are forwarded to the next domain.

Due to topology hiding, the CNC, which usually is supposed to configure streams between two end-stations, has no knowledge about the existence of all end-stations involved in an MDTSN stream request. Here, two new parameters are introduced for stream requests: the *ingress MAC address* and the *egress MAC address*. These parameters specify the boundary network port MAC addresses between two adjacent domains. The E/W Manager adds the parameters if required.

5.2 Reachability Information Exchange

Only the CNC has information about the network's topology. Due to topology hiding, this information is limited to the local domain. To enable MDTSN stream requests, the E/W Manager requires reachability information exchange over the east-westbound protocol. Due to topology hiding, this information should be limited. Information required for MDTSN with respect to topology hiding is the topology of interconnected domains, the directly connected boundary network port MAC addresses, a list of local end-stations (available for multi-domain communication) and a list of (disclosed) end-stations of directly and indirectly connected domains.

5.3 MDTSN Stream Request

The sequence diagram in Figure 4 illustrates the stream request procedure for a forwarding domain in MDTSN. It shows the interaction between the previous domain and the next domain of the stream request. (1) An E/W Manager receives a stream continuation requested by a neighboring E/W Manager. (2) Stream parameters are determined by the E/W Manager. This includes validating the existence of the end-stations, the determination of the next domain and its boundary network port. (3) A stream segment reservation request is sent to the local CNC. Note, that this stream request is a reservation request, without a direct configuration. This reduces data plane reconfigurations in case a domain later in the chain of the MDTSN stream request is not able to provide a stream. In (4), the CNC performs traffic scheduling and calculation of network configurations. (5) Scheduling results are provided as a response of the CNC to the E/W Manager. (6) Stream parameters for the next domain are determined including the recalculation of the remaining requested maximum latency, the ingress MAC address, ETO and LTO. (7) A stream continuation request is sent to the E/W Manager of the next domain. (8) Further domains on the path to the listener end-station are trying to provide a stream. (9) When successful, the stream configuration is confirmed to the E/W Manager. (10) The E/W Manager requests the CNC to configure the reserved stream segment. (11) The CNC configures the affected TSN bridges with a new configuration. (12) The successful configuration is confirmed to the E/W Manager. (13) The E/W Manager confirms the successful stream configuration from the listener end-station back to the current domain.

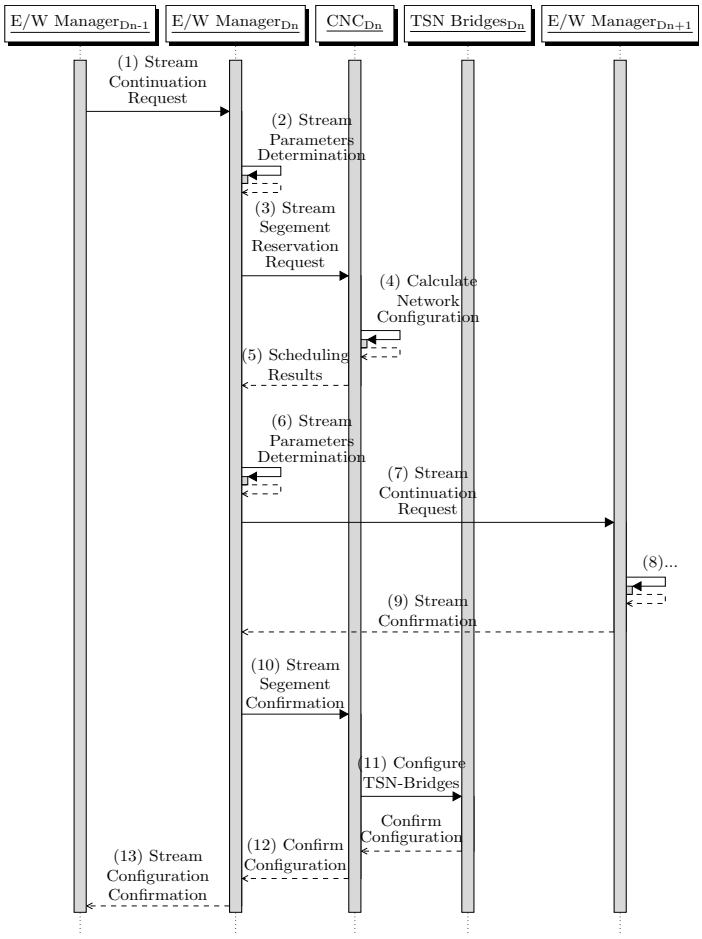

Figure 4: Stream request procedure for a forwarding domain in MDTSN

6 Inter-Domain Forwarding Offset (IDFO)

Figure 5 shows a configured stream on the MDTSN data plane exemplified with three TSN domains. Each domain involved in this stream is responsible for proper scheduling of its stream segment, while notifying the next domain about the periodical arrival time to continue the end-to-end stream setup. Figure 5 depicts, that domain 2 and domain 3 are *wait for forwarding* at the point in time, when messages arrive at its domain, as scheduled, and notified by the previous domain. Depending on the current utilization of the TSN bridges at the point of time of the stream setup, traffic may not be directly scheduled, due to already reserved resources within the network cycle. This idle time leads to an inter-domain forwarding offset (IDFO). When requesting two streams with identical Stream Request Parameter Sets (SRPS) at a different Stream Request Time (SRT), due to different utilizations of the TSN bridges and different sizes of the IDFOs, the streams actually provided end-to-end latency is different. When TSN domains have high traffic loads, this may even lead to unsuccessful stream configurations, where the requested maximum end-to-end latency cannot be achieved. Each domain within an MDTSN stream may have an IDFO for that stream. The size of the IDFOs are not limited and may have the size of multiple network cycles. Depending on the traffic class and the required maximum latency, for cyclic-synchronous traffic, a sum of large IDFOs may work. For isochronous traffic, the required end-to-end latency has the size of one application cycle, which's size is close to the network cycle. Thus, large IDFOs lead to unsuccessful stream requests. Control plane mechanisms to mitigate the size of the IDFOs will further be investigated.

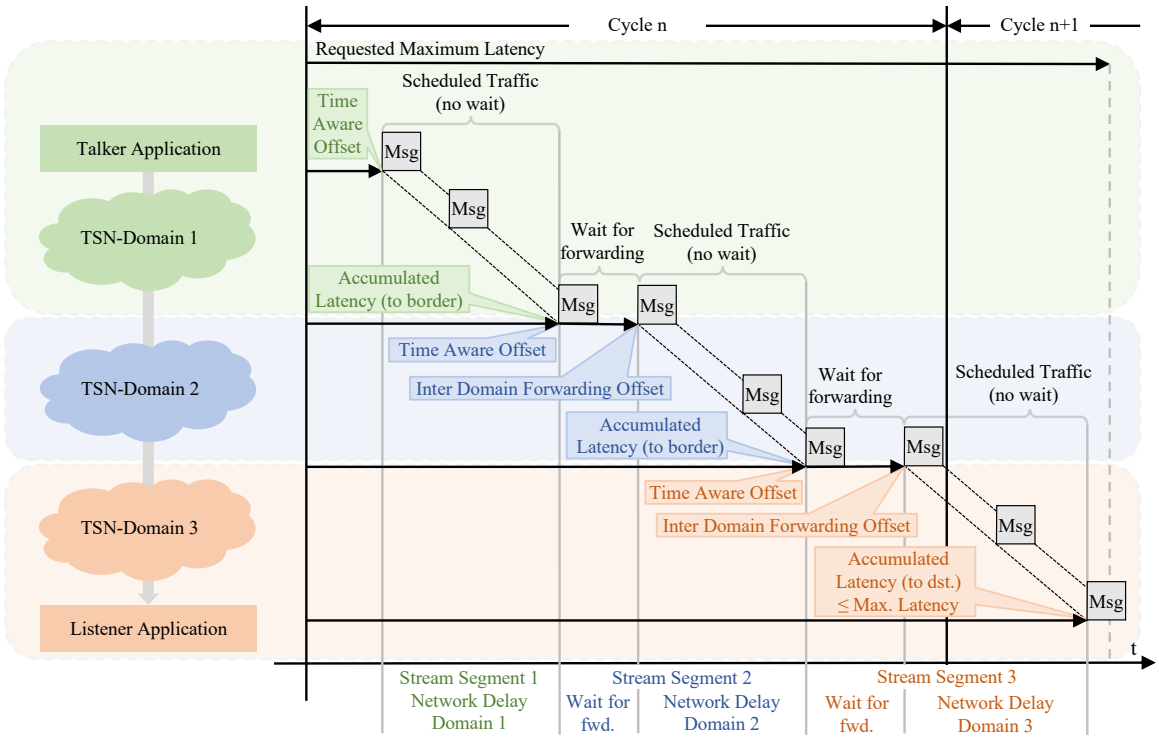

Figure 5: Data plane configuration of an MDTSN stream exemplified with three TSN domains

7 IDFO Mitigations

In MDTSN, each domain provides a stream segment of an end-to-end stream. As presented in the previous chapter, IDFOs emerge in MDTSN, increasing the end-to-end latency. Each IDFO reduces the remaining maximum latency for the next domains. Within the goal of successful stream configurations, the sum of the IDFOs should be minimized. Further, we present different control plane mechanisms to mitigate the IDFO:

1. **Maximum latency sharing**: Sharing the requested maximum latency of a stream request by e.g., diving the requested maximum latency by the number of remaining domains. This mechanism restricts each domain to provide a stream within the reduced maximum latency.
2. **Iterative stream requests**: Whenever a stream request is rejected by a domain, the stream is iteratively requested again by the first domain. Here, the ETO and LTO are restricted for the first domain within the goal to achieve different ETO/LTO for the next domain. Note, that the ETO and LTO increase the flexibility for scheduling, while in MDTSN, only the first domain can make use of this flexibility. For all further domains, the arrival time is fixed, i.e., the ETO and LTO are identical.
3. **Multiple stream configuration offers**: Similar to iterative stream requests, the CNC of the first domain is requested with different ranges of the ETO and LTO to achieve multiple stream configuration offers. Further domains have increased flexibility.

Further, the presented control plane mechanisms are discussed.

1. **Maximum latency sharing**: For scheduling approaches using ASAP scheduling [RP17], where streams are scheduled at the earliest point in time that is feasible, the sharing mechanism forces the scheduler to find a stream reservation with reduced latency. For scheduling approach intending to reduce the end-to-end latency, this mechanism may be ineffective.
2. **Iterative stream requests**: The computational complexity for the schedule calculation rises with the number of existing streams and the size of the network. Nevertheless, due to dynamic stream reservation, adding new streams is less complex than a full schedule calculation [RP17]. Iterative stream requests may lead to longer end-to-end stream configurations when requesting streams repeatedly.
3. **Multiple stream configuration offers**: Providing multiple offers by each domain may lead to a higher flexibility for the schedule calculation for the next domains, increasing the chance of a successful stream configuration. On the other side, this mechanism temporary blocks lots of scheduling

resources. Domain internal stream requests and other MDTSN stream requests, which are parallelly processed, while waiting for active MDTSN stream configurations to finish, have a reduced chance of a successful stream configuration due to less available scheduling resources.

8 Implementation and Evaluation

We developed an MDTSN test environment to prove the viability of the MDTSN control plane mechanisms, focusing on those presented in Chapter 7. All three presented mechanisms, *maximum latency sharing*, *iterative stream requests* and m*ultiple stream configuration offers* have been implemented and tested in the MDTSN test environment.

As a basis for this Proof of Concept (PoC) implementation, single TSN domains, including TSN bridges and end-stations on the data plane as well as CUC and CNC as part of the control plane, are needed. The technical basis implementation partly implements IEEE 802.1AS, IEEE 802.1Qbv and IEEE 802.1Qcc. For the data plane, the TSN bridges TrustNodes of the vendor InnoRoute are chosen, because they already support the required standards.

For the control plane, no implementations are currently available on the market. Therefore, we extended the open source, python based SDN controller Ryu [Ryu] with a *Ryu application* to add functionality for the CUC and the CNC. The CNC supports reservation and configuration of unidirectional streams using exclusive gating i.e., each stream is assigned its own queue. The scheduling algorithm uses the ASAP approach. A GCL is generated as a YANG configuration file for TSN bridges and configured using Netconf. OpenFlow is used to match traffic to queues by the source and destination MAC address of the stream. The interface for the CUC to request and release streams by end stations is implemented as a REST API.

For the east-westbound communication, the E/W Manager has been implemented which communicates with other E/W Managers using a REST API with JSON encoding. It uses the python API to communicate with the CNC. A list of local devices, remote devices and their domain, directly and indirectly connected domains and their boundary network port, as well as contact information of other domain's E/W Managers is set manually in the PoC implementation.

The test environment is similar to the architecture of Figure 3, comprising three TSN domains. Because each domain at least requires an ingress network port and an egress network port on the data plane, a single TSN bridge per domain is sufficient.

8.1 Test scenario

All three control plane mechanisms have been tested in different kinds of scenarios. One scenario has been selected and is simplified shown in Figure 6a. The figure shows the scheduled traffic of the egress network port of each TSN bridge of the MDTSN test environment. Grey boxes show already reserved time slots within the network cycle. The test scenario has been executed for each presented control plane mechanism with a network cycle of 125µs. A cyclic stream is requested with an end-to-end latency of 62,5µs, a PDU size of 100 bytes, an ETO of 0 and an LTO of 125.

Figure 6b shows the stream reservation process of the maximum latency sharing mechanism. The maximum end-to-end latency is divided by the number of remaining domains. The second domain is not able to provide a stream within the required latency. Figure 6c shows the iterative stream request mechanism. It shows that the stream is rejected by domain 3 multiple times while domain 1 is requested with an incremented ETO. In our implementation, the stream has been requested over 60 times within this scenario, until a feasible configuration has been found. Figure 6d shows the multiple stream configuration offer mechanism. We divided the range between the ETO and LTO into 4 parts (0-31,25; 31,25-62,5; …). 2 of the 4 stream offers were successful.

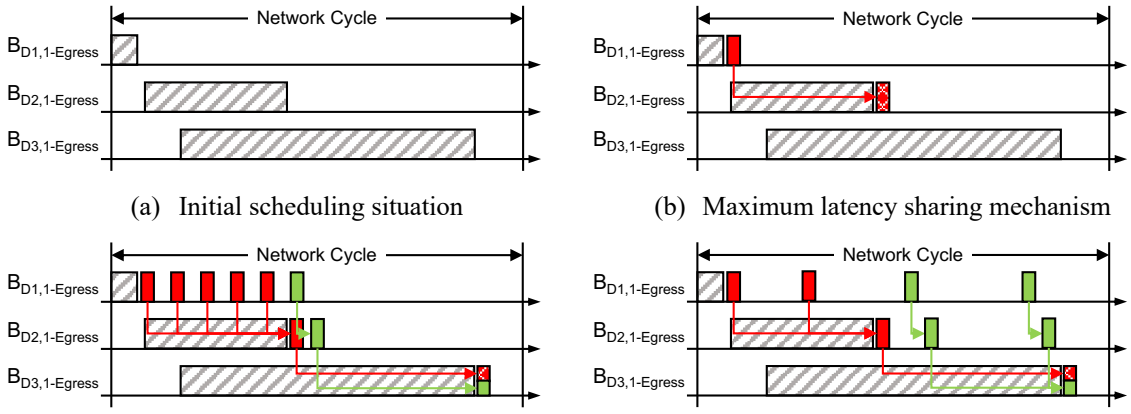

Figure 6: MDTSN stream configuration process of a stream with a requested maximum latency of 62,5µs within a network cycle of 125µs using different control plane mechanisms as presented in Chapter 7

8.2 Evaluation of IDFO Mitigation Approaches

Further the presented and tested control plane mechanisms are discussed focussing on the impact on the IDFO.

1. **Maximum latency sharing**: Within the presented test scenario, this mechanism is not able to provide a stream at all. Nevertheless, in more complex scenarios (larger network with more streams), this mechanism is able to restrict the scheduler, by requesting streams with a lower maximum latency than the requested end-to-end latency. Without the sharing, there may be domains consuming a majority of the requested end-to-end latency. In respect to the IDFO, this mechanism is not effective.
2. **Iterative stream requests**: Within the test scenario, this mechanism is able to find a feasible configuration after over 60 retries. The IDFO in domain 2 has been reduced for this stream. In more complex scenarios, this mechanism may consume lots of control plane computation resources.
3. **Multiple stream configuration offers**: Within the test scenario, this mechanism is able to find multiple stream configurations (2 of 4). The third offer had an IDFO in domain 3 which consumes most of the end-to-end latency, still resulting in a successful stream configuration. The fourth offer resulted in a small IDFO in domain 3. While offers are reserved for a certain stream, these resources are blocked and can not be used for other stream requested arriving in the process of the MDTSN stream request, which may lead other streams to unsuccessful stream configuration.

Some of the presented control plane mechanisms were able to mitigate the size of IDFOs in an MDTSN stream. In the peer-to-peer MDTSN architecture, where the stream configuration is parted into stream segments, control plane mechanism to compensate the missing global end-to-end view while preserving topology hiding, is important. Iterative stream requests and multiple stream configuration offers increase the chance of a successful stream request, while increasing the load on the control plane. The maximum latency sharing mechanism has its benefits and may be used in combination with the other mechanisms.

9 Conclusion and Future Work

This paper presented developed control plane mechanisms for MDTSN to achieve on-demand multi-domain end-to-end bounded-latency stream reservation which are required for inter-domain communication in TSN needed for various use cases. We presented an MDTSN architecture in a peer-to-peer like architecture for the fully centralized TSN configuration model. The control plane mechanisms for the integration of an east-westbound protocol can be implemented without modifying existing TSN standards while considering topology hiding. IDFOs have been identified to emerge within this distributed solution when domains configure stream segments as part of an end-to-end stream. Different control plane mechanisms to mitigate the IDFOs have been presented, discussed, and evaluated in an MDTSN test environment. These mechanisms are able to increase the stream configuration success rate, while the size of emerging IDFOs has been decreased. As a drawback, the mechanisms increase the computational load on the control plane i.e., more east-westbound communication and more traffic scheduling.

Investigating these mechanisms in more complex scenarios while comparing different scheduling approaches should be addressed in the future. Further control plane mechanisms for MDTSN should be developed to support e.g., different network cycles of TSN domains on the data plane. Control plane performance should be analyzed for use cases with frequent setup and release of MDTSN streams.

10 Acknowledgment

This work was funded by the Federal Ministry for Education and Research within the KMU-innovativ program as a part of MONAT (16KIS0782) and the Ministry for Science and Culture of Lower Saxony as a part of the research project SecuRIn (VWZN3224).

11 References

[BW21] Martin Böhm, and Diederich Wermser. Multi-domain time sensitive networks—control plane mechanisms for dynamic inter-domain stream configuration. Electronics 10, no. 20: 2477. 2021 https://doi.org/10.3390/electronics10202477

[CO16] Silviu S Craciunas, Ramon Serna Oliver, Martin Chmelík, and Wilfried Steiner. Scheduling real-time communication in ieee 802.1 qbv time sen-sitive networks. In Proceedings of the 24th International Conference on Real-Time Networks and Systems, pages 183–192, 2016.

[Fou16] OPC Foundation. OPC Unified Architecture Specification, Part 14: Pubsub, Draft 1.04.21. 2016.

[Fou17] OPC Foundation. PubSub TSN Configuration Broker, Whitepaper Version 0.5. 2017.

[IAUC] IEC CD / IEEE 802.1 TSN TG ballot. Use Cases IEC/IEEE 60802. https://www.ieee802.org/1/files/public/docs2018/60802-industrial-use-cases-0918-v13.pdf, accessed on 22.09.2021.

[IEBA] Ieee standard for local and metropolitan area networks–audio video bridging (avb) systems. IEEE Std 802.1BA-2011, pages 1–45, 2011.

[IEQC] Ieee standard for local and metropolitan area networks – bridges and bridged networks – amendment 31: stream reservation protocol (srp) enhancements and performance improvements. IEEE Std 802.1Qcc-2018 (amendment to IEEE Std 802.1Q-2018 as amended by IEEE Std 802.1Qcp-2018), pages 1–208, Oct 2018.

[Ind] Industrial Internet Consortium (IIC). Time sensitive networks for flexible manufacturing testbed Characterization and mapping of converged traffic types. https://iiconsortium.org/pdf/IIC_TSN_Testbed_Char_Mapping_of_Converged_Traffic_Types_Whitepaper_20180328.pdf, accessed on 22.09.2021

[Lih] Lihao Chen. TSN Configuration Interaction. https://www.ieee802.org/1/files/public/docs2019/new-chen-TSN-Configuration-Interaction-0719-v01.pdf, accessed on 22.09.2021.

[QBV] IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks – Amendment 25: Enhancements for Scheduled Traffic. IEEE Std 802.1Qbv-2015, pages 1–57, March 2016.

[RP17] Michael Lander Raagaard, Paul Pop, Marina Gutiérrez, and Wilfried Steiner. Runtime reconfiguration of time-sensitive networking (tsn) schedules for fog computing. In 2017 IEEE Fog World Congress (FWC), pages 1–6. IEEE, 2017.

[Ryu] Ryu SDN Framework Community. Ryu sdn framework. https://ryu-sdn.org/, accessed on 22.09.2021.

[SDNi] H Yin, H Xie, T Tsou, D Lopez, P Aranda, and R Sidi. Sdni: A message exchange protocol for software defined networks (sdns) across multiple domains. IETF draft, work in progress, 2012.

[SKJ18] Sebastian Schriegel, Thomas Kobzan, and Jürgen Jasperneite. Investigation on a distributed sdn control plane architecture for heterogeneous time sensitive networks. In 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), pages 1–10. IEEE, 2018.

[TSIA] IEEE 802.1 Working Group. IEC/IEEE 60802 TSN Profile for Industrial Automation. https://1.ieee802.org/tsn/iec-ieee-60802/, accessed on 22.09.2021.

[WSJ17] Martin Wollschlaeger, Thilo Sauter, and Juergen Jasperneite. The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. IEEE industrial electronics magazine, 11(1):17– 27, 2017.

[ZC18] Yuan Zhang, Lin Cui, Wei Wang, and Yuxiang Zhang. A survey on software defined networking with multiple controllers. Journal of Network and Computer Applications, 103:101–118, 2018.

http://dx.doi.org/10.25673/39569

Secure Usage of Asset Administration Shells - An Overview and Analysis of Best Practises

Andre Bröring, Marco Ehrlich, Henning Trsek, Lukasz Wisniewski
Technische Hochschule Ostwestfalen-Lippe
inIT - Institute Industrial IT

Campusallee 6
32657 Lemgo, Germany
andre.broering@th-owl.de
marco.ehrlich@th-owl.de
henning.trsek@th-owl.de
lukasz.wisniewski@th-owl.de

The Asset Administration Shell (AAS) is a core element for Industrie 4.0. In addition, the security of industrial systems is a permanent topic that could be improved by the AAS and should have a high priority for future developments and implementations of the AAS. This paper evaluates the current threat landscape for Industrial Control Systems (ICS) communicating to the AAS, as well as for IT systems hosting the AAS. The relevance of these threats is evaluated for the AAS and the threats with the highest relevance, namely Basic Web Application Attacks and Malware Infections, are analysed in detail. The recommended countermeasures for these threats are compared with the state of the art of AAS security concepts and result in missing countermeasures and research gaps for an overall security of the AAS.

1 Introduction

The current progress within the developments of the Industrie 4.0 (I4.0) offer a great potential to increase the digitalization and to support new technologies in the area of industrial automation. The upcoming data-driven systems and innovative services are the basis to link virtual and physical processes as a fundamental concept of I4.0 [WSJ17]. This is mostly done by the adaptation of interconnecting approaches from the Information Technology (IT) domain into the environment of Operational Technologies (OT). Ubiquitous connectivity is the general enabler required by the majority of future industrial applications. In general, these trends demand for new possibilities with regard to data acquisition, processing, and utilization, resulting in a higher priority for the intrinsically linked topic of security as well.

The Asset Administration Shell (AAS) as a promising concept in this area is designed to be the future repository for all relevant information related to industrial assets. The AAS can be exchanged as a file, but can also have different communication capabilities to communicate to the represented asset itself and provide information to other services inside organizations or across company boarders [Pla20b]. In a positive scenario, the AAS can help to create future secure industrial production systems, e.g. by providing monitoring information like behaviour anomalies of the represented physical system to detect potential security-related attacks [DP20, EE18]. In a negative scenario, the universal communication capabilities and the overall available information about each asset centralized in the AASs could become a new vulnerability for industrial organizations [HR19]. In this case, a security incident including access to the AASs could have a tremendous impact. Besides manipulations and data leaks, an attacker can use the crucial information about the production system from the AAS for further attacks and exploits that could lead to further loss of intellectual property or production unavailability, like during the shutdown of the Colonial fuel pipeline in the USA in May 2021 [NTE21].
Up to now, a threat analysis for the AAS in all asset life cycle phases covering different scenarios does not exist. On the one hand, the AAS concept has a deep relevance towards the IT domain because of the used technologies. On the other hand, the AAS is specified to function inside the OT domain for industrial automation systems. This creates a tension with regard to the required type of security evaluations, the applicable threats, the general security objectives, and the available countermeasures. The current status of the security-related development of the AAS already covers various concepts and implementations. Nevertheless, an overview of the available functionalities and the subsequent mapping towards present threats and available countermeasures are missing. The results of this work are an evaluation of the security capabilities of the AAS in order to withstand the current threat landscape and the conclusion of open research challenges for future activities in this domain. The remainder of this work is organized as follows: Section 2 further introduces the current state of the art of the AAS, security capabilities of the AAS, and supporting topics. Afterwards, Section 3 presents the current threat landscape endangering AAS implementations and assess the available countermeasures. Section 4 concludes this work by displaying a summary and the outlook containing future work in this research domain.

2 Background

This section gives a short overview of the concepts of the AAS and the state of the art of AAS security from the Plattform I4.0 as well as other research domains.

2.1 The Asset Administration Shell (AAS)

During the life cycle of an asset, all information about the asset can be stored in the AAS. This can be construction data of a machine from engineering tools, values from installed sensors of a machine during operation, or a description of executed process

steps for maintenance tasks. All this data about the assets is stored in a standardized structure and enables the access of data about every asset and data exchange between AASs in a harmonized way with the AAS files or via the APIs [Pla20a]. One asset can have one or multiple AASs referring to each other, which can be stored in different locations and different kinds of storage, such as in an embedded storage on the asset itself, in edge or in cloud storages [Pla17, Pla20b].
The AAS has three different levels of data exchange capabilities and interaction patterns. The first one, called passive AAS, is a simple file containing the AAS and can be exchanged along the whole life cycle of the asset [Pla20a]. The re-active AAS has a standardized communication interface described in a technology-neutral specification and can be realized with an HTTP REST, OPC UA, or MQTT API [Pla20b]. The communication capabilities ensure a seamless communication to the OT networks of a plant containing various assets like Programmable Logic Controllers (PLCs) and Human Machine Interfaces (HMIs), as well as to the IT networks for engineering and life cycle management tools, data-driven services, or Manufacturing Execution Systems (MESs). Independent of the storage location, the AAS may need to have the same communication possibilities to the IT and OT networks, even across company boarders.
The third type of AASs, called pro-active AAS, autonomously and pro-actively interacts with other AASs in peer-to-peer communications using the I4.0 language [Pla21a]. This paper considers only the passive and the re-active AAS, due to simplicity because of the current development status of pro-active AASs.

2.2 Asset Administration Shell Security

The published security concepts for the AAS are considered and summarized in this chapter. Before the Industrial Digital Twin Association (IDTA) was founded, the Plattform I4.0 was the main driver of the AAS and published security requirements, concepts, and approaches in several specifications and discussion papers.
In general, in the current implementations of re-active AASs, a server application, such as the AASX Server, can load the AASX file and provide the information via an HTTP REST or OPC UA API [Ind21]. Hence, the AAS is not communicating by itself, but loaded into an application to enable the communication. As a result, not only the AAS itself, but also the tools and applications interacting with the AAS and providing the API need to be taken into account for the secure usage of AASs [Pla21b].
To share passive AASs with business partners, an AAS provider can share a copy of the file, in that confidential data is missing. If the data provider does not want to create several copies with individual information, or wants to keep some information inside the AAS to share this later, access control comes into play. According to IEC TS 62443-1-1:2009, access control is the "protection of system resources against unauthorized access". The access control management for the AAS includes the management of read access, write access, and write access in a limited value range [Pla17].
In the beginning, Role Based Access Control (RBAC) was proposed for the AAS [Pla17]. However, later, Attribute Based Access Control (ABAC) was defined as the preferred

solution for the AAS in cross-company communication scenarios [Pla19a]. The permission rules for the access control are stored and exchanged within the AASs itself and can be provided by the asset manufacturer. Nevertheless, the receiver of an AAS can also set own access permission rules. Based on the subject attributes, object attributes, and environmental attributes, the access permission rules are evaluated locally to decide if the access is allowed or declined. Prerequisite for the access control are unique identities and an identity management in a secure environment to identify the subjects that want to access the AAS and the objects, such as the whole AAS or submodels, that should be accessed. Details about this secure environment for digital identities are mentioned for further development [Pla17, Pla20a].
A detailed description for an exemplary secure download service including authorization and authentication of a requester as well as the transfer of the AASX files containing custom non-public data from a supplier to the requester, such as an AAS integrator, is described by the Plattform I4.0 [Pla20c]. With a signature of the AAS provider for the AASX package, integrity and confidentiality can be ensured during file transportation [Pla20a]. The security of the AAS after download is not considered.
In the research domain, different ways to share and manage AASs via distributed systems are proposed. First, there is a concept to store the AAS in a distributed file system in combination with a git-based version control and a blockchain-based tamper-proof log [RVPK20]. In a second work, the data itself or a checksum of the data, such as a submodel, is directly shared via a distributed ledger to ensure a high integrity of the data [BBT+21]. On one hand, the distributed solutions prevent single points of failure. In addition, the access for different business partner as well as measures to ensure integrity and confidentiality by encryption are core features. On the other hand, the copies of the data in the distributed network are beyond an owner's control.
In cross-company communication scenarios, the HTTP protocol is recommended and well-established. The integrity, confidentiality, and authenticity of these connections should be ensured on the application layer to avoid an interrupt because of inspecting proxies on the transport layer (TLS proxies) used in many company networks. The availability depends on the physical and the MAC layer, realized as wireless and wired connections [Pla17, Pla20c]. A prototypical implementation of such an HTTP REST server for re-active AASs is already realized in the open-source AASX server [Ind21]. Similar to the secure download service, an authentication server provides access tokens to access objects via the HTTP REST interface of the AASs.
As an alternative to HTTP REST, the Plattform I4.0 provides functionalities for the communication with endpoints outside the organization's networks using OPC UA with direct server-client connections to the assets, or aggregation servers to bundle several OPC UA connections for the cross-company communication [Pla19b]. These methods can be applied to the OPC UA connections of re-active AASs.
Lastly, all accesses to functions and values of the AAS should be logged by the AAS to help to identify manipulations on the AAS. Other security measures, such as data usage control and data origin tracking, are mentioned for further development [Pla17, Pla20a]. Also in future, the security capabilities and especially trustworthiness level of an AAS should be rated on a not-yet-defined scale and delivered by the asset manufacturer [Pla17].

3 Threat Landscape Analysis

After introducing and explaining the fundamental background with regard to the AAS security, this section contains the analysis of the current threat landscape for industrial environments. Security-related threats build up the basis for the strictly needed risk assessment processes later on. The gained insights will be mapped to the current state of the AAS security development in order to check which threats are already considered. In addition, this section includes the comparison of AAS-related threats and the currently available countermeasures. This leads to open challenges aiming at follow-up research directions with regard to the further development of the AAS security concepts.

3.1 Threat Analysis

The current threat landscape endangering and affecting the industrial automation domain is broad and diverse with regard to attack vectors, attackers, and possible compromising approaches [PK18]. As a basis, the "Industrial Control System Security Top 10 Threats and Countermeasures 2019" published by the Federal Office for Information Security (in German BSI) [Bun19] are further investigated in Table 1. Recent incidents show the interconnection of the IT and OT domain, similar to the concept of the AAS. Therefore, the Verizon Incident Classification Patterns [Ver21] that are typical for IT systems are added. Four duplicated threats from these two sources are merged, and the remaining three added to the BSI ICS Top 10. The AAS-specific relevance (Relev.), meaning that the AAS is affected by the threat, is rated between low, medium, and high. The threats with a high relevance have a high effect on the AAS security and are analysed in detail in Section 3.2. The threats with a low relevance can be mitigated on an organization level, as shown below.

The threats *Lost and Stolen Assets* as well as *Infiltration of Malware via Removable Media and External Hardware* have no specific relevance for the AAS, as it is not a hardware component which can be stolen, or to which removable media can be connected to. Further, *Privilege Misuse*, that mainly originates from internal threat actors [Ver21], *Human Error and Sabotage*, and *Social Engineering and Phishing* have no direct relevance, as human interaction is not a key feature of the AAS and the corresponding countermeasures, such as awareness training, are on an organizational policy level. Nevertheless, the mentioned threats can have an impact on the AAS security, as these are all possible entrance points for exploits also affecting AASs.

Technical Malfunctions, Force Majeure & Miscellaneous Errors are general threats that are relevant for the particular software implementations hosting or interacting with the AAS. The *Compromising of Extranet and Cloud Components* is also a threat that is not AAS-specific. Both are out of scope of this paper and the corresponding countermeasures from the BSI [Bun19] can be applied organization-wide. In addition, GAIA-X as a future European cloud infrastructure [GX21] could contribute a secure and sovereign storage infrastructure for the AASs.

Table 1: BSI ICS Top 10 [Bun19] and Verizon Incident Classification Patterns [Ver21]

Threats and Incident Classification Patterns	**Relev.**
Basic Web Application Attacks Malware Infection via Internet and Intranet & System Intrusion	high
(D)DoS Attacks Control Components Connected to the Internet Intrusion via Remote Access Compromising of Smartphones in the Production Environment Technical Malfunctions, Force Majeure & Miscellaneous Errors Compromising of Extranet and Cloud Components	medium
Privilege Misuse Social Engineering and Phishing Human Error and Sabotage Infiltration of Malware via Removable Media and External Hardware Lost and Stolen Assets	low

For *Intrusion via Remote Access*, *Control Components Connected to the Internet*, and *Compromising of Smartphones in the Production Environment*, the AAS could even be part of a security enhancement solution, as it could replace direct connections between the asset and remote access providers, the Internet or smartphones. The AAS can be a standardized interface collecting the data from the assets and providing this to the external endpoints. In this case, the AAS interfaces could be patched more easily than the industrial assets supporting the security measures with more secure interfaces to external endpoints compared to the worse patched ICS interfaces.
In the same manner, the AAS could reduce the impact of *(D)DoS Attacks* targeting the ICS, as the direct accessibility of the ICS can be replaced by the AAS interface. However, the AAS could still be a target for (D)DoS attacks, so that the implementation of countermeasures is recommended, such as the hardening of network access points and communication channels, installation of intrusion detection systems, and redundant connection of components using different protocols [Bun19].

3.2 Detailed Threat Analysis

The remaining threats with a high relevance for the AAS are exemplified here in detail, with an outlook on exiting and missing countermeasures. Table 2 shows an excerpt of the most AAS-relevant countermeasures from the BSI [Bun19] and IEC 62443-4-2. The last column marks, if the corresponding countermeasures are already implemented for the AAS "✓", if at least concepts for the countermeasures do exist "(✓)", or if it is not yet considered "-".

Basic Web Application Attacks A typical way to compromise data from a victim are the *Basic Web Application Attacks*. This is a hacking method and acts as an entrance for further attacks, such as malware or (D)DoS. The Verizon "DBIR 2021

Table 2: Countermeasures from BSI ICS Top 10 and IEC 62443-4-2

Countermeasure	**AAS**
Standardised interfaces to reduce undiscovered ICS vulnerabilities	✓
Identification, authentication, and access control	✓
Limitation of available information	(✓)
Logging, monitoring, and attack detection	(✓)
Provisioning supplier roots of trust	(✓)
Secure communication	(✓)
Network segmentation	-
Antivirus software	-
Periodical backups (data recovery)	-

Data Breach Investigations Report" shows that servers are the most likely affected assets for security breaches in all industries and security incidents in most industries. The main sub-pattern are stolen credentials and brute force attacks [Ver21]. The re-active AAS could be a future target for this kind of attacks.
The AAS itself as a *standardized interface* is already a countermeasure to lower the risk of undiscovered vulnerabilities in APIs by replacing many asset-specific interfaces with the standardized AAS interface. A second countermeasure is the Attribute Based Access Control (ABAC) concept for the AAS including a secure *identification, authentication, and access control* process [Pla20c]. At the same time, this supports the *limitation of available information* by limiting access rights. This could be extended by the partition of an AAS in parts with different risks. As one asset can have several AASs, the information could also be distributed among these AASs. In case of an incident via a Basic Web Application Attack to an AAS connected to the Internet, not all asset data is exploited. More confidential data could be stored in an AAS with limited or no connection to the internet. The AAS concept enables such a partition of asset data, but the best practise to realize this is missing.
The same counts for the idea of *logging and monitoring* of AASs that was proposed early [Pla17]. An overall concept to realize this for the interfaces of re-active AASs is missing. For now, only an approach to log the changes of AAS files was done in [RVPK20], but does not cover the communication interface of re-active AASs. The logging could be complemented with a universal *attack detection* to identify misconducts or attacks.
A missing countermeasure to achieve *secure communication*, in addition to the secure usage of OPC UA [Pla19b] and the proposed security implementation on the application layer for HTTP [Pla20c], is the usage of Virtual Private Networks (VPNs) between AASs, and between AASs and other services.
In addition to the above-mentioned countermeasures, the Top 10 vulnerabilities in API implementations and corresponding countermeasures are listed in detail in the "OWASP API Security Top 10" [The19].

Malware Infection via Internet and Intranet & System Intrusion Especially with recent security incidents, such as the ransomware attack via the IT service provider

Kaseya [Fun21], in mind, malware including ransomware and supply-chain attacks is one of the most relevant threats. A ransomware leading to data loss of all information about the assets could have a huge impact on the owning company. Also, sabotage through manipulated data is possible. The malware can infect an AAS at any party within the supply-chain, resulting in a supply-chain attack. Therefore, the *provisioning supplier roots of trust* similar to the trustworthiness level concept in [Pla17] could increase the security of AASs. To detect malware, an *antivirus software* scanning incoming AAS files should be a minimum standard. The *logging, monitoring, and attack detection* is a second step to identify malware.
In case that the malware is already introduced via an AAS, the further spread can be limited by *network segmentation*. To do so, a suitable and secure network architecture for the AAS including firewalls, VPNs, and network monitoring should be developed to avoid a spread of malware from the Internet via the AAS to the IT and OT networks. Lastly, a *periodical backup* will be helpful to enable a quick recovery after a potential incident. A dynamic selection and rating of data importance as well as an optimization of the backup that is practicable for numerous AASs should be evaluated. For example, not every sensor value needs to be saved forever, whereas some crucial asset properties and submodels should be stored securely without time limitation. In addition, the backup interval could be different for submodels with dynamic data and static properties.

Results The AAS as a standardized interface for industrial assets and the AAS access control, including the identification and authorization, cover already two crucial countermeasures for secure ICSs. However, the detailed investigation of the two threats with the highest relevance for the AAS, as well as the investigation of the corresponding countermeasures yield to some open issues in the state of the art of AAS security.
A concept for a smart partition of asset data in several AASs with different security measures, and the implementations of an overall logging, monitoring, and attack detection system need to be developed. This should be optimized for the whole AAS cross-company life cycle and protect the AAS from unwanted access and manipulations. Further, a future I4.0 network architecture with network segmentation optimized for the AAS that enables a maximum communication capability to different IT and OT networks on one hand, and prevents the spread of malware inside the networks on the other hand, is needed. That applies as well to a backup strategy that is practicable and efficient for numerous AASs and enables a quick recovery after a possible incident.

4 Conclusions and Future Work

For a great acceptance of the AAS and the realization of the positive scenario with the AAS leading to a security enhancement, some characteristics are essential for the further AAS design. This paper shows that not all current ICS threats are relevant for the AAS. Some threats can even be mitigated by the AAS. For some other threats, solutions and countermeasures do already exist and are summarized in this paper.

Nevertheless, the threat analysis for the AAS identified some open research challenges and activities for the implementation of more countermeasures to minimize future security incidents while using the AAS.
In future work, more threats and countermeasures, such as from MITRE ATT&CK® and MITRE Shield, as well as from the IEC 62443-4-2 can be evaluated with regard to the AAS. The data from the future secure AASs can be used, for example, as the basis for an automated safety and security assessment for modular production systems [EBH+20].

Acknowledgement

This contribution was funded within the project AutoS^2 as part of the technology network it's OWL with support from the Ministry of Economic Affairs, Innovation, Digitalization and Energy of the state of North Rhine-Westphalia, Germany.

References

[BBT+21] Andre Bröring, Alexander Belyaev, Henning Trsek, Lukasz Wisniewski, and Christian Diedrich. Secure Asset Administration Shell exchange with Distributed Ledger Technology. In *Shaping a globally secure Industrie 4.0 Ecosystem*. Plattform Industrie 4.0, Berlin, 2021.

[Bun19] Bundesamt für Sicherheit in der Informationstechnik. Industrial Control System Security: Top 10 Bedrohungen und Gegenmaßnahmen: V1.3, 2019.

[DP20] Marietheres Dietz and Gunther Pernul. Unleashing the Digital Twin's Potential for ICS Security. *IEEE Security & Privacy*, 18(4):20–27, 2020.

[EBH+20] Marco Ehrlich, Stefan Benk, Dimitri Harder, Philip Kleen, Henning Trsek, Sebastian Schriegel, and Jürgen Jasperneite. Automatische Bewertung und Überwachung von Safety & Security Eigenschaften – Strukturierung und Ausblick. *Jahreskolloquium Kommunikation in der Automation (KommA)*, 2020.

[EE18] Matthias Eckhart and Andreas Ekelhart. A Specification-based State Replication Approach for Digital Twins. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and Privacy*, pages 36–47, 2018.

[Fun21] Brian Fung. New ransomware attack targets key IT vendor. *https://edition.cnn.com/2021/07/02/tech/ransomware-cybersecurity-attack-kaseya/index.html*, 2021.

[GX21] GAIA-X. What is Gaia-X? *www.data-infrastructure.eu*, 2021.

[HR19] Mark Hearn and Simon Rix. Cybersecurity Considerations for Digital Twin Implementations, 2019.

[Ind21] Industrial Digital Twin Association e.V. admin-shell-io by IDTA: aasx-server. *www.github.com/admin-shell-io*, 2021.

[NTE21] Ellen Nakashima, Yeganeh Torbati, and Will Englund. Ransomware attack leads to shutdown of major U.S. pipeline system. *www.washingtonpost.com/business/2021/05/08/cyber-attack-colonial-pipeline*, 2021.

[PK18] Animesh Pattanayak and Matt Kirkland. Current Cyber Security Challenges in ICS. In *IEEE International Conference on Industrial Internet (ICII)*, 2018.

[Pla17] Plattform Industrie 4.0. Security der Verwaltungsschale: Diskussionspapier, 2017.

[Pla19a] Plattform Industrie 4.0. Access control for Industrie 4.0 components for application by manufacturers, operators and integrators: Discussion Paper, 2019.

[Pla19b] Plattform Industrie 4.0. Secure cross-company communication with OPC UA: Discussion Paper, 2019.

[Pla20a] Plattform Industrie 4.0. Details of the Asset Administration Shell: Part 1 - The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC01): Specification, 2020.

[Pla20b] Plattform Industrie 4.0. Details of the Asset Administration Shell: Part 2 - Interoperability at Runtime – Exchanging Information via Application Programming Interfaces (Version 1.0RC01): Specification, 2020.

[Pla20c] Plattform Industrie 4.0. Secure Download Service: Discussion Paper, 2020.

[Pla21a] Plattform Industrie 4.0. Functional View of the Asset Administration Shell in an Industrie 4.0 System Environment: Discussion Paper, 2021.

[Pla21b] Plattform Industrie 4.0. Was ist die Verwaltungsschale aus technischer Sicht?, 2021.

[RVPK20] Magnus Redeker, Sören Volgmann, Florian Pethig, and Johannes Kalhoff. Towards Data Sovereignty of Asset Administration Shells across Value Added Chains. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Piscataway, NJ, 2020. IEEE.

[The19] The OWASP Foundation Inc. OWASP API Security Top 10 2019: The Ten Most Critical API Security Risks, 2019.

[Ver21] Verizon. DBIR 2021 Data Breach Investigations Report, 2021.

[WSJ17] Martin Wollschlaeger, Thilo Sauter, and Jürgen Jasperneite. The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0. In *IEEE Industrial Electronics Magazine*, 2017.

http://dx.doi.org/10.25673/39570

Integration of Asset Administration Shell and Web of Things

H. K. Pakala*, K. O. Oladipupo*, S. Käbisch, Ch. Diedrich*,**

**Otto von Guericke University Magdeburg, Magdeburg, 39106, Germany*
(e-mail: christian.diedrich@ovgu.de, Harish.Pakala@ovgu.de, kazeem.oladipupo@st.ovgu.de)
***Siemens AG, München, B 80333, Germany (e-mail: Sebastian.kaebisch@siemens.com)*

Abstract:

In the course of the digital transformation of industrial production processes, physical assets are encapsulated with software components, where a set of such computing entities working in unison form and build the cyber-physical systems (CPS). The Platform Industry 4.0 (PI4.0) provides a standardized meta-model in terms of submodels and submodel elements to digitally represent information about an asset. The submodels are used to represent a wide range of information like the asset nameplate, administrative aspects, technical specifications and even the accessibility information. Some of these data have to be acquired from the asset. The W3C Web of Things provides the Thing Description meta-model to represent the accessibility runtime information of these assets.

This paper aims to map the W3C Web of Thing (WoT) Thing Description (TD) to the submodel and the submodel elements of the PI4.0 Asset Administration Shell (AAS). A standard submodel template encapsulating each of the Thing Description classes would be the outcome of this paper and also a plugin module embedded into the AAS package explorer that utilizes this template to automatically generate AAS submodel representation from TD for any given asset. This paper also presents a brief description of a use-case consisting of a Delta Robot which can be accessed via an OPC UA server.

Keywords: automation systems, cyber physical systems, Asset Administration Shell, Web of Things

1. Introduction

Cyber Physical Systems (CPS) is the network of a finite set of computing devices that are encapsulated over physical assets, where the network ensures a bi-directional communication between computing and the physical devices (Alur 2015). The information exchanged between different entities over this network can be channelled, extracted and be used for various tasks like machine learning, data analytics, predictive maintenance and, to control and monitor systems through SCADA softwares. These tasks inturn help in providing an efficient production processes and thereby enabling a self-optimizing and self-managing production system. Internet of Things (IoT) refers to the networking of the physical and the virtual world over the internet (ITU-T Y.4000).

The Platform Industry 4.0 (PI4.0), a network of industry stakeholders, research organizations and like-minded partners is a German government initiative, with the primary aim to integrate CPS with the usage of IOT into the industrial production processes (Henning Kagermann 2013). The PI4.0 network is organized into six different working groups, where each group is associated with specific tasks like creation or adaptation of existing standards, security and legal aspects, creation of new use-cases, education and training.

The group that deals with standardization has coined a new term Asset Administration Shell (AAS), a standardized digital representation of an asset. Accordingly, the group has proposed a meta-model to

represent the information of an asset digitally (Platform Industry 4.0). This meta-model is primarily composed of submodels and submodel elements like property, range, reference element, multi-language property, operation, event, file and blob. The meta-model is a more generalized one and any information about an asset like nameplate, technical data, access related information can be modelled using the submodels (SM) and the available submodel elements (SME). (Platform Industry 4.0)

The concept of W3C Web of Things (WoT) and the Thing Description is another concerted effort of W3C foundation (McCool and Kaebisch 2021). The group provides a standardized meta-model termed as Thing Description (TD) that aims to increase the interoperability and easier integration of runtime data and functions of IoT platforms (Kaebisch et al. 2021). The meta-data has constructs such as properties, actions, events, forms, links and security definitions. The entire meta-data is well organized and each construct is well attributed in the sense of software development technologies. This enables it to include wide range of access related information of an asset. Both the representations AAS and Thing Description are human readable and address both the physical and logical assets in the realm of the cyber physical systems. The Thing Description restricts itself to the aspects of the accessibility, while the AAS meta-model enables to represent wide range of information like digital nameplate or documentations.

In this paper we aim to integrate Thing Description meta-model with asset administrations shell's SM and SME. We consider this aspect as a problem of mapping one meta-model onto another. A set of mapping rules will be introduced which can be used by a model translator for transforming Thing Description instances into AAS submodel Instances. The AAS package explorer is the official modelling tool from the PI4.0, a plugin module will be integrated into this modelling tool as part of this work.

The rest of the paper is organized into six sections, the next section provides background work related to Asset Administration Shell (AAS) and its meta-model, Thing Description and its meta-model. The third section presents a brief summary of past works from different authors in the field of model transformation or model mapping.

2. BACKGROUND WORK

In this section, a brief overview of the AAS meta-model and the TD meta-model is presented.

2.1 Asset Administration and its Meta-Model

The PI4.0 defines an Asset Administration Shell *as a standardized digital representation of an asset where the asset could be a logical or a physical entity (Platform Industry 4.0)* and consequently it proposes a meta-model so as to structure the information about the asset in a standardized manner. This structured information is both human and machine readable. It can be represented either in JSON or XML formats and be wrapped in an AASX package container format.

The meta-model aims to structure the information in terms of a collection of finite number of submodels and each submodel containing a set of elements from the available submodel elements. Each submodel is expected to represent a specific category of information like nameplate, asset access related information, technical document, etc. Figure 1 presents the screen shot from AASX package explorer of the nameplate submodel for delta-robot asset considered for the use-case in this paper.

The UML diagram in Figure 2 shows the shortened version of the AAS meta-model, not all the entities are shown in this diagram, the complete description is available at (Platform Industry 4.0). Figure 2 captures the classes AAS, Asset, SM, DataElement, MultiLanguagePropery (MLP), SME and the information about their respective inherited classes. The complete description of the inherited classes is not represented in this diagram.

The SM and SME classes inherit the HasSemantics class. This class enables to associate specific semantic references to the respective instances. Similarly, both SM and SME are Qualifiable, this is to specify a set of qualifiers, where each qualifier is type-value pair that enables to specify additional information about the specific instance. A semantic reference can also be attributed to a qualifier as it

inherits the HasSemantics class. Additionally, AAS and SM inherit the identifiable class. This class is defined to attribute unique identifiers and version, revision information to the specific instances. The Referable class inherited by the SME class enables to specify idShort attribute to the specific instance which represent an identifier valid in the local instance of an AAS.

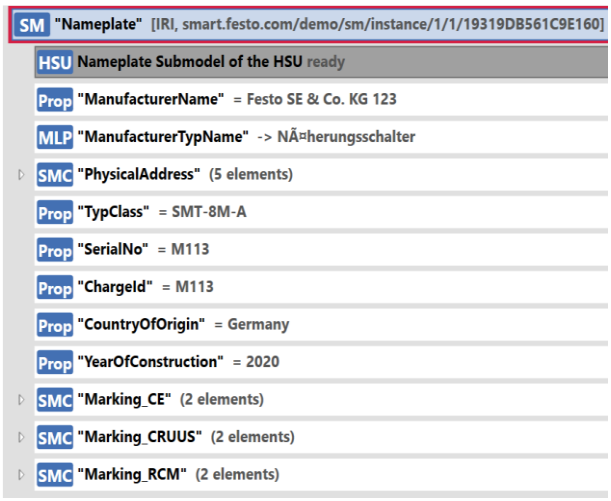

Figure 1 Screenshot of Package Explorer depicting nameplate submodel (Ristin et al. 2021)

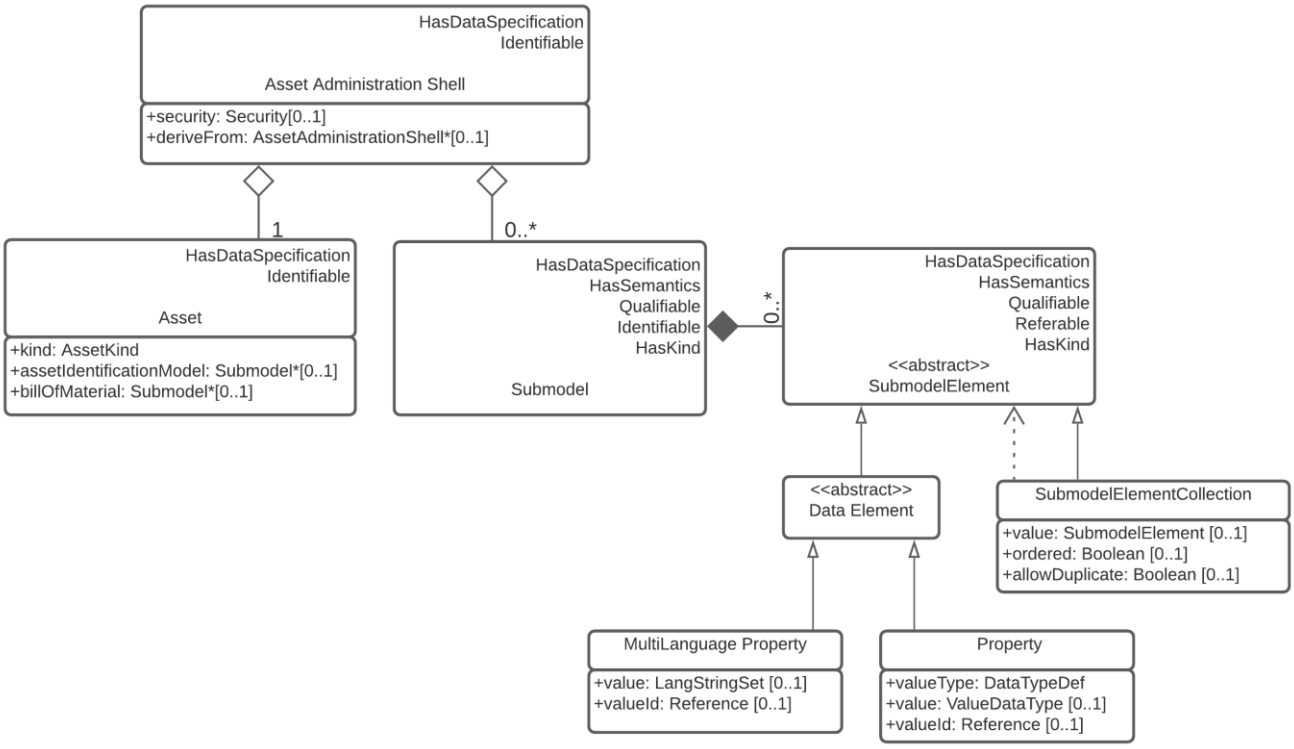

Figure 2 Class diagram representation of AAS Basic meta-model (Platform Industry 4.0)

The submodel can have a finite set of SME, where this SME could be a submodelElementCollection (SMEC) or a dataElement or any other listed SME from the meta-model. The SMEC can also have a finite set of the SME including another SMEC. The MLP (MultiLanguageProperty) data element is used to specify same text in different languages. The other SME like property, reference, range, file, blob,

capability, operation, basic event, entity, relationship element and annotated relationship element are not mentioned in this UML diagram presented in Figure 2, the relevant reasons behind the exclusion is provided in the section 6 of the paper.

2.2 THING DESCRIPTION and TD META-MODEL

The Thing Description (TD) is a recommended JSON-LD based document developed within the W3C Web of Thing (WoT) group (McCool and Kaebisch 2021; Kaebisch et al. 2021). The main idea of a TD is to provide a standardized semantic interface description as an entry point for a physical asset to easily onboard its runtime data and functions as well as enable rapid IoT application development. The WoT working group provides a top meta-model for the TD consisting of well-defined set of classes. Figure 3 presents the top-level meta-model of TD.

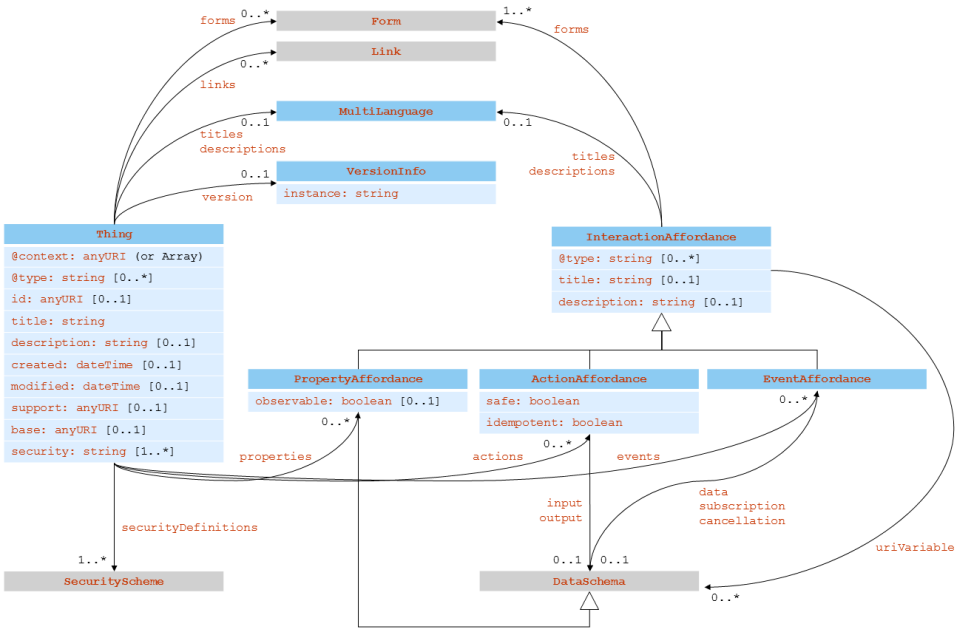

Figure 3 TD meta model (Kaebisch et al. 2021)

The *Thing* class provides the core vocabularies for describing the asset's metadata (e.g., title, ID, etc.) and specifies what kind of properties, actions, and events are exposed. The TD uses different standardized interaction affordances to categorize the capabilities offered by the asset.

1. **Property affordances**: Represent states that an asset exposes. Typically, a property can be read, written to and/or observed. Properties can be also used to define dynamic or static parameters.
2. **Action affordances**: Represent either state manipulations or physical processes that can be invoked.
3. **Events affordances:** Represent notifications (e.g., alarms) and data streams that can be subscribed to.

The *SecuritySchema* class is used to provide metadata about what kind of security modes are used by the asset and what authorization service (if any) needs to be involved in the interaction. The *DataSchema* provides a mechanism for specifying the data model reflected in the properties, actions, and events. In doing so, the *DataSchema* vocabularies rely mainly on JSON Schema terms.

WoT is a protocol agnostic approach and provides a common mechanism to define how specific protocols such as MQTT, HTTP, Modbus or OPC UA can be mapped to the WoT's interaction properties-action-event abstraction. This information is mainly provided by the forms container within a Thing Description. Based on this information, the client knows how to activate each WoT interaction abstraction through a corresponding network-facing interface for a specific protocol on which the asset relies.

3. RELATED WORK

Meta-models define the structure of a model that is a possible representation of an entity in an abstract way (Markus et al. 2013). For example, Programming languages like Java, Python, C are represented by specific meta-models that provide rules and syntax for modelling the programs. There are a vast number of meta-models each related to a specific domain that attempts to solve a modelling problem for that domain. In an interconnected world, interactions between instances or entities modelled using different meta-models is an aspect of concern. The differences in representations made by different meta-models would require to map one meta-model to another and thereby perform model-to-model transformations. In this section, a summary of different research works that address the aspect of model-to-model mapping or transformations is presented.

The Model Transformation Language (MOLA) as described by (Kalnins et al. 2004) provides a structured and descriptive graphical language consisting of flow charts to transform an instance of a source model confirming to a source meta-model into a target model confirming to the target meta-model. The MOLA insists that both source and target meta-models are compliant with MOF 2.0 standard (Overbeek 2006) where this standard provides a framework and set of services for the creation of meta-models and their interoperability.

To bridge the gap between different applications that utilize different meta-models to represent the available information the authors (Agostinho et al. 2010) have proposed an intermediary conceptual meta-model described using UML representation. This meta-model is expected to have features and concepts such that a mapping between this model and any other meta-model is possible. Firstly, the one-to-one mappings are made between conceptual meta-model and all the other available meta-models and later these are used to translate instantiations of any two different meta-models via the intermediary representation.

(Henßen and Schleipen 2014) try to provide a mapping between the information models OPC UA (Stefan-Helmut Leitner 2006) an IEC 62541 standard and the AutomationML (AML) an IEC 62714 standard (Drath et al.). The underlying concept of the OPCUA meta-model is a node class, specializing this class there are a set of classes likes Objects, Variables, references where references provide the relation between the other nodes. The information about any real-world object can be modelled using these node classes and can be hierarchically organized using Folders. The AML on the other hand is a combination of other standards that is specifically designed to model plant engineering information. An AML representation contains a set of InterfaceClassLibs composed of Interface definitions, RoleClassLibs composed of semantic role definitions, SystemUnitClassLibs composed of reusable AML objects and lastly the Instance Hierarchy representing the actual plant description. The authors (Henßen and Schleipen 2014) have mapped these Libraries to OPC UA folder types, and different classes types as OPC UA object types that are organized into the folder types, new reference types HasAMLChild, HasAMLRoleReference, HasAMLInternalLink are created to model the relationships between the AML Objects and ObjectTypes.

(Lelde et al. 2015) propose a unique approach for situations where there is one common target met-model and different source meta-models are expected to be mapped to it in this approach, a new intermediate virtual meta-model that represents the target meta-model from a higher abstraction level and with a better structural foundation is introduced. Fixed set of mappings between the intermediate and the target meta-model (in both the directions) are provided. New mappings with respect to the intermediate meta-model will have to be created for every source meta-model. With the underlying intermediate-to-target meta-

model mappings, it is implicit that a source meta-model instance is transformed to the intermediate meta-model instance.

(Cavalieri et al. 2019) have made an attempt to map AAS (Platform Industry 4.0) meta-model onto the OPC UA meta-model, for complex types such AAS, Submodel, SubmodelElement corresponding OPC UA object types are created. In case of simple types such as version, revision, idShort, value related variables are crated and the HasSemantic entity is mapped to an opcua hierarchical reference type. The collection elements such as submodels, submodelCollections are organized as opcua foldertypes.

(Miny et al. 2020) have proposed custom build model transformation language by extending the features of OCL to translate a source submodel instance to a target submodel instance. A summary of the works presented in the sections indicates that whenever the source and target meta-models follow a standard, a generalized translation framework could be used to translate their corresponding instances. Also, a common higher abstracted meta-model could be used as an intermediate and mapping or translation can be done between this and the actual meta-models. Lastly the custom defined solutions are posed by different authors where the source or the target meta-models do not confirm to exiting standards.

In this paper we propose custom mapping rules between AAS and Thing Description meta-models, the mapping rules are inspired by theory proposed in the (Marco and Yannis 2003). Although the authors attempt to address the aspect of semantic interoperability, the underlying logic seems to be a good fit to the problem addressed in this paper.

4. PROPOSED METHODOLOGY

This section firstly presents a high-level comparison between the two meta-models TD and AAS. Secondly, it proposes new mapping rules for transforming the individual instances from one model representation to another.

The TD meta-model is composed of finite set of classes each represented by a set of attributes, where an attribute could be a variable of simple type like string, float and integer or an object type of another class or a map, list of simple types and of objects. The entire structure is quite complex with circular references where these references could be well suited to accommodate different attributes related to web of things. On the other-hand, even though class diagram representation of the AAS meta-model is much complex, it offers very simple entities like property (of simple type like integer, string, float, anyURI etc), operation that can expect a set of input and output variables, multi-language property, Entity, BasicEvent, Qualifier, Range, File, Blob, Submodel Element Collection, Submodel etc.

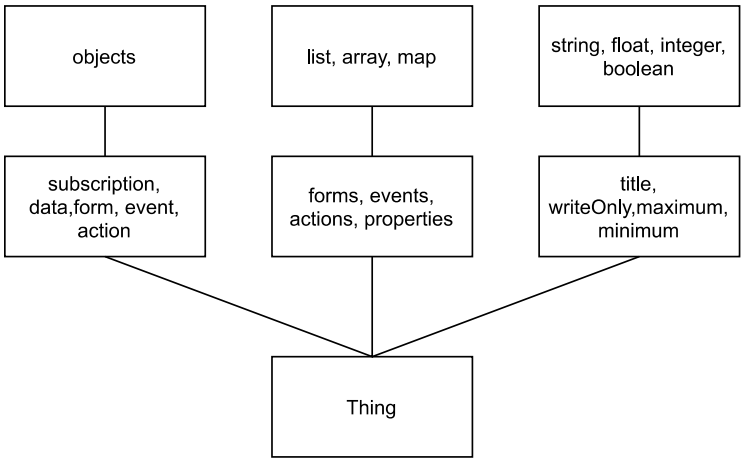

Figure 4 Hierarchical representation of Thing Description Entities

(Marco and Yannis 2003) utilize the concepts of Information-Flow theory to address semantic interoperability between two different systems representing the same domain. In this work firstly, responsibilities that are primitive and common to both the systems are identified. Secondly, these identified responsibilities are mapped to the divisions within the respective systems that address them. Thirdly, these divisions are associated with groups that are composed of a set of divisions. Such a kind hierarchical approach is continued until all the entities among the respective systems are considered. Lastly, the IF logic calculations are performed to map the entities from both the systems. This paper aims to construct such hierarchical structures and then construct mapping logics between TD meta-model and AAS SM and SME.

Figure 4 presents a much simpler and abstract representation of the TD meta-model. This representation is hierarchically structured, at the top- level are the types that any of the TD entity could be, where these are simple types (like string, float, integer and boolean), collection types (like array and map) and complex types like objects. The second level of the representation indicate all the entities from the TD meta-model that are of particular type from the first level. Such a kind of hierarchical representation is similar to the one presented in (Marco and Yannis 2003). In the case of AAS meta-model a similar structure is not contemplated, rather the elements submodel, submodelElementCollection and the qualifiers are only considered for mapping with the elements from TD meta-model.

Figure 5 present mapping rules between TD and AAS. All the entities of the simple type are mapped as qualifiers from AAS meta-model, where the entity name will be the qualifier type and the entity value will be qualifier value. All the complex types including lists, classes and objects are mapped to submodelElementCollection and lastly the ThingDescription docunment or the Thing class is mapped to submodel. For example, the idempotent attribute of the ActionAffordance will be mapped as a qualifier and the action class is mapped to the submodelElementCollection. This ~~is~~ mapping rule is base-line rule for the TD-AAS model transformation proposed as part of this work.

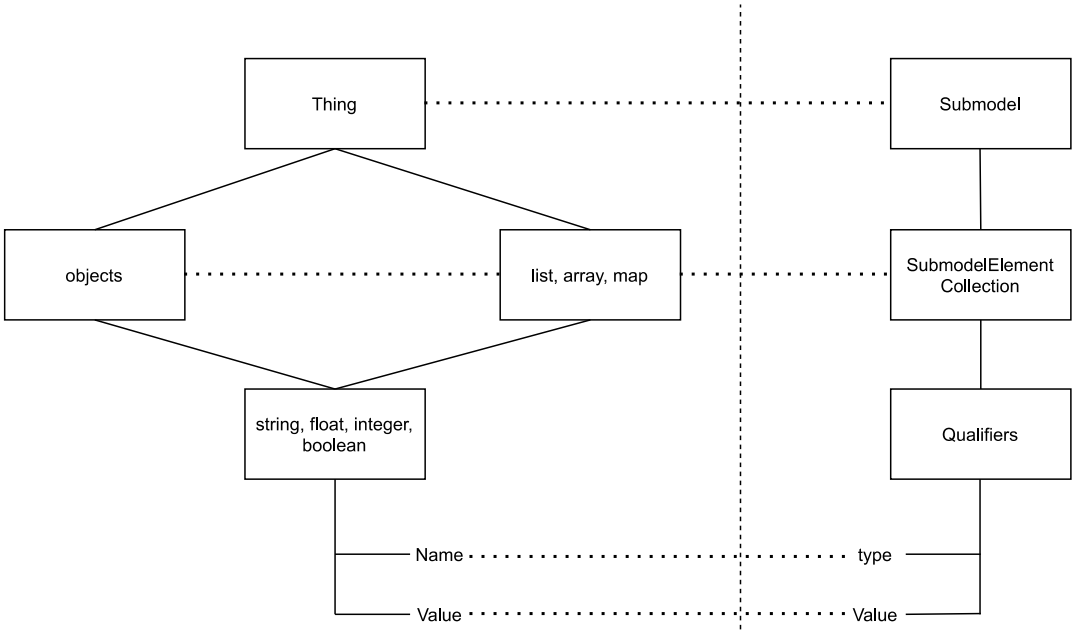

Figure 5 Mapping between Thing Description and AAS meta-models

Few variations compared to this base line mapping rule is associated with to titles, description, versionInfo and id entities. The titles entity is a list of strings, it is mapped to multilanguage property.The descriptions entity is mapped to description attribute of the corresponding submodelElementcollection. The versionInfo attribute of Thing class is mapped to the administration attribute of the submodel, where version is attached to version and model is attached to revision. Lastly the id attribute of the thing class is mapped to the id attribute of identification class related to corresponding submodel.

5. USECASE

In the context of this work, we have modelled Thing Description of the delta robot (DR) which is part of the OVGU pick and place demonstrator (Diedrich et al. 2021). The robot is accessible through an OPC UA server, which is hosted on a PLC and the actual interaction with the robot is done using this PLC. The OPC-UA server instance is modelled as per the robotics OPCUA companion specification. (OPC Foundation 2019). Figure 6 presents the picture of the OVGU demonstrator and a screenshot of the associated OPCUA server instances.

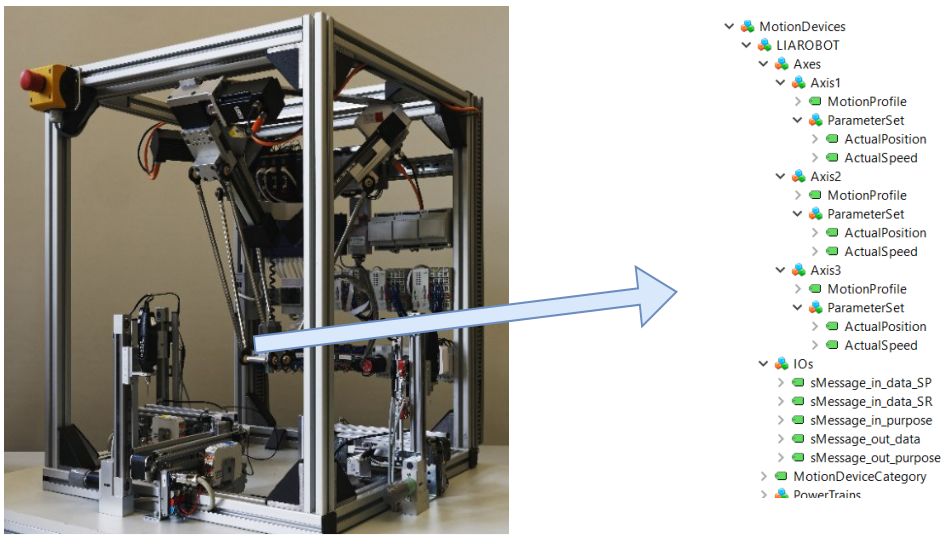

Figure 6 Snaphost of OVGU Demonstrator and Screen shot of OPCUA interface

A Thing Description JSON-LD document is created for this OPCU-UA server instance, here only the variables that are required to access the robot are modelled in the document, the figure 7 presents the screenshot of TD document. Eight variables from the OPCUA server instance are modelled as properties in the TD document and the required security definitions for access to the server are also captured.

```
{   "@context" : ["https://www.w3.org/2019/wot/td/v1"],
    "title": "LIADeltaRobot",
    "base": "opc.tcp://192.168.1.2:4840",
    "description": "The thing description of LIA delta robot.",
    "created" : "2021-09-30T13:22:05+01:00",
    "support": "mailto:kazeem.oladipupo@st.ovgu.de",
    "security": ["nosec_sc"],
    "securityDefinitions": {"nosec_sc": {"scheme": "nosec"}},
    "properties":{
                "Axis1ActualPosition" :{
                "Axis2ActualPosition" :{
                "Axis3ActualPosition" :{
                "sMessage_in_purpose" : {
                "sMessage_in_data_SR": {
                "sMessage_in_data_SP": {
                "sMessage_out_purpose": {
                "sMessage_out_data": {
                        "observable" : false,
                        "type" : "string",
                        "readOnly" : false,
                        "writeonly": false,
                        "description" : "This affordance is used to output the state of the task",
                        "forms" : [{
                            "op" : ["readproperty", "writeproperty"],
                            "href" :
                            "opc.tcp://192.168.1.2:4840/ns=4;s=|var|HX-CP1H16.Application.OPC_UA_Symbols.Motio
                            iceSystem.MotionDevices.LIAROBOT.IOs.sMessage_out_data"
                        }]
```

Figure 7 Snapshot of Thing Description for OPCUA Server Instance

As part of this work a plugin module is added to the AAS package explorer (Ristin et al. 2021), the relevant code is available under the “ovgu/ThingDescription” branch (Pakala 2021). This plugin module implements

the mapping rules presented in the section 4, it consumes a Thing Description document (with an extension .json) and creates the relevant submodel. The figure 8 presents the screenshot of the package explorer and the relevant process for creation of the submodel.

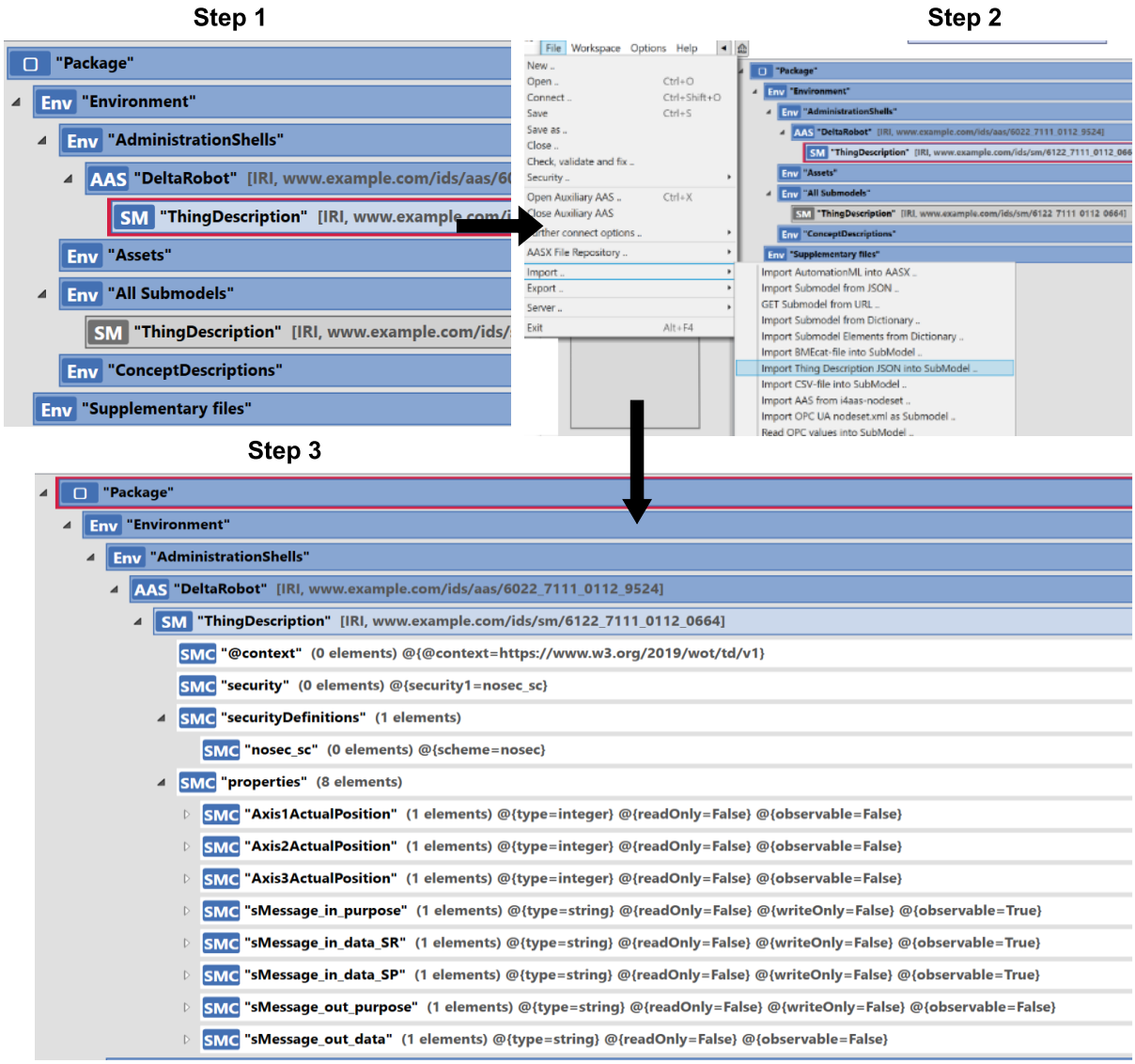

Figure 8 Screenshot of the AASX Package explorer depicting on how to import Thing Description Document

6. DISCUSSION and PROPOSALS

PI4.0 intends to provide a standardised meta-model to digitally model information about an asset. The primary or basic building block of this representation is the submodel and PI4.0 intends to structure all information about an asset like the nameplate, technical data using this submodel. The meta-model additionally provides a set of elements to capture details within the specific category of information. The PI4.0 expects that research and industry bodies come up with standardized submodel templates using the available meta-model elements. It would be easier for manufacturers, suppliers, vendors and other interested partners to utilize these templates and properly organize the information about their assets. This work is such an attempt to create a submodel template for representing access related information of an asset.

The WoT Thing Description (TD) is an ~~better~~ organized and well-represented meta-model used to structure the asset access related information. In this work, instead of working from scratch, we utilize the constructs from the TD meta-model and have created a new submodel template. This process is done by creating a mapping between the two meta-models, thereby custom mapping rules are introduced. Even though the AAS meta-model offers elements like property, reference element, range, MultiLanguage property, operation, basic event and collection only the collection element is used for representing complex entities. The qualifier attribute associated with each of these elements is used to represent the simple types from the TD.

Elements like property, range and operation can be well utilized to represent certain entities from the TD. The property element can be used to represent the simple types and probably operations can be mapped to forms. The main reason, to avoid such mapping rules is to avoid a huge hierarchical structure within the AASX representation and also visual complexity in the package explorer. That is the design and capabilities of the explorer has greatly influenced this work. During the time of this work, the only available modelling tool for the AAS is the package explorer (Marko et al.). Another reason for such a consideration is to avoid complex mapping rules and have an underlying base rule for all the components. A note on the package explorer, operation element can accept any other submodel element as input and output parameters, but the explorer restricts it to property element.

The property element from the AAS meta-model has the feasibility to specify only a single value of a specific type like integer, boolean, string etc. In case this element is re-modelled to represent multiple values, it would greatly influence the mapping rules presented in this paper. An additional suggestion towards the modification of the AAS meta-model would be to add a DateTime class that has the attributes create datetime, modify datetime, last modified and all the elements of the meta-model inherit this class.

7. CONCLUSION

In this paper, we have introduced a submodel template that aims to represent access related information of an asset. For this the WoT TD meta-model is used as a reference. During the course of the research work we have understood that the AAS some online related assets meta-model elements needs to be evolved, existing submodel elements needs to be upgraded and further elements need to be introduced.

8. ACKNOWLEDGEMENTS

The research was carried out in the project “VWS Vernetzt / AAS Networked” under support of the Federal Ministry for Economic Affairs and Energy, FKZ: 01QE1544B.

Publication bibliography

Agostinho, Carlos; Correia, Filipe; Jardim-Goncalves, Ricardo (2010): Interoperability of Complex Business Networks by Language Independent Information Models. In Jerzy Pokojski, Shuichi Fukuda, Józef Salwiński (Eds.): New World Situation: New Directions in Concurrent Engineering. London: Springer London (Advanced Concurrent Engineering), pp. 111–124.

Alur, Rajeev (2015): Principles of cyber-physical systems. Cambridge, Massachusetts, London, England: The MIT Press.

Cavalieri, Salvatore; Mule, Salvatore; Salafia, Marco Giuseppe (2019): OPC UA-based Asset Administration Shell. In : IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society: IEEE.

Diedrich, C.; Belyaev, A.; Schröder, T.; Urban, C.; Werner, T.; Pakala, H. (2021): Modell einer Pick & Place-Anlage basierend auf Verwaltungsschalen. In : Automation 2021: VDI Verlag, pp. 65–74.

Drath, Rainer; Luder, Arndt; Peschke, Jorn; Hundt, Lorenz: AutomationML - the glue for seamless automation engineering. In : 2008 IEEE International Conference on Emerging Technologies and Factory Automation. Factory Automation (ETFA 2008). Hamburg, Germany: IEEE, pp. 616–623.

Henning Kagermann (2013): Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Forschungsunion, acatech.

Henßen, Robert; Schleipen, Miriam (2014): Interoperability between OPC UA and AutomationML. In *Procedia CIRP* 25, pp. 297–304. DOI: 10.1016/j.procir.2014.10.042.

Kaebisch, Sebastian; Kamiya, Takuki; McCool, Michael; Charpenay, Victor (Eds.) (2021): Web of Things (WoT) Thing Description. Available online at https://w3c.github.io/wot-thing-description/, checked on 10/10/2021.

Kalnins, Audris; Barzdins, Janis; Celms, Edgars (2004): Model Transformation Language MOLA. In : Model Driven Architecture: Springer, Berlin, Heidelberg, pp. 62–76. Available online at https://link.springer.com/chapter/10.1007/11538097_5.

Lelde, LACE; Audris, KALNINS; Agris, SOSTAKS (2015): Process DSL Transformation by Mappings Using Virtual Functional Views. Available online at https://www.bjmc.lu.lv/fileadmin/user_upload/lu_portal/projekti/bjmc/contents/3_2_4_lace.pdf.

Marco, Schorlemmer; Yannis, Kalfoglou (2003): Using information-flow theory to enable semantic interoperability. Available online at https://www.icsa.inf.ed.ac.uk/publications/online/0161.pdf.

McCool, Michael; Kaebisch, Sebastian (Eds.) (2021): W3C Web of Things. Available online at https://www.w3.org/WoT/, updated on 10/10/2021.

Miny, Torben; Thies, Michael; Epple, Ulrich; Diedrich, Christian (2020): Model Transformation for Asset Administration Shells. In : IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society: IEEE.

OPC Foundation (Ed.) (2019): Robotics OPCUA Companion Specification. Available online at https://reference.opcfoundation.org/Robotics/, checked on 10/11/2021.

Overbeek, J. F. (2006): Meta Object Facility (MOF): investigation of the state of the art. Available online at http://essay.utwente.nl/57286/.

ITU-T Y.4000, 6/15/2012: Overview of the Internet of things.

Pakala, Harish Kumar (Ed.) (2021): AASX Package Explorer Thing Description Plugin. Available online at https://github.com/admin-shell-io/aasx-package-explorer/tree/ovgu/ThingDescription, checked on 10/11/2021.

Platform Industry 4.0: Details of the Asset Administration Shell. Part 1 - The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC01). In.

Ristin, Marko; Orzelski, Andreas; Hoffmeister, Michael (Eds.) (2021): AASX Package Explorer. Available online at https://github.com/admin-shell-io/aasx-package-explorer, checked on 10/11/2021.

Stefan-Helmut Leitner, Wolfgang Mahnke (2006): OPC UA–service-oriented architecture for industrial applications. Available online at http://cimug.ucaiug.org/kb/knowledge%20base/soa%20for%20industrial%20applications.pdf.

http://dx.doi.org/10.25673/39571

Analyse von Entscheidungsprozessen beim VDI/VDE 2193 Bieterverfahren

Sergej Grunau, Lukasz Wisniewski
Institute Industrial IT - inIT / TH-OWL
Campusallee 6
32657 Lemgo, Germany
sergej.grunau@th-owl.de
lukasz.wisniewski@th-owl.de

Abstract: Das Bieterverfahren nach VDI/VDE 2193 ermöglicht Verhandlungen zwischen Industrie 4.0-Komponenten. Jedoch ist die Umsetzung solcher Industrie 4.0-Komponenten, welche sich durch proaktive Verwaltungsschalen auszeichnen, nicht klar definiert. Besonders die Entscheidungsprozesse im Bieterverfahren, bei denen die Verhandlungsteilnehmer etwa über Angebotszu- oder absage bestimmen sollen, stellen eine Herausforderung in der Umsetzung dar. In dieser Arbeit wird ein konkreter Anwendungsfall aus der auftragsgesteuerten Produktion beschrieben. Anhand des Anwendungsfalls wird analysiert welche Kriterien für die Entscheidungsprozesse notwendig sind und wie diese und die Entscheidungsprozesse in proaktiven Verwaltungsschalen integriert werden können. Dazu zeigen wir eine mögliche Umsetzungsvariante proaktiver Verwaltungsschalen. Diese werden im wesentlichen mit Verwaltungsschalen-Applikationen und reaktiver Verwaltungsschalen realisiert, welche mittels Teilmodellen wie der hier beschriebenen „BiddingprocedureConfig“ miteinander interagieren können.

1 Einleitung

Interoperable I4.0-Komponenten können eine nahtlose Zusammenarbeit von smarten Maschinen und Fabriken innerhalb eines Unternehmens und über die Unternehmensgrenzen hinweg ermöglichen.
Eine Art von Zusammenarbeit sind Verhandlungen zwischen den Maschinen und Produkten um Dienstleistungen auszutauschen die z.B. einen oder mehrere Produktions-

schritte eines Produkts umsetzen sollen. Die Verhandlungen sollen dabei automatisiert ablaufen um eine flexiblere und effizientere Fertigung kundenspezifischer Aufträge zu ermöglichen, so wie in der Auftragsgesteuerten Produktion (siehe [1] [2]). Damit die Verhandlungen automatisiert ablaufen können müssen die Dienstleistungen hinreichend präzise und in einer maschinenlesbaren Form beschrieben sein und die Verhandlungsteilnehmer müssen eine gemeinsame und einheitliche Sprache sprechen. Dabei helfen Modelle und Beschreibungsmittel wie (i) die im Zuge der Plattform Industrie 4.0 standardisierte Verwaltungsschale [3], welche Industrie 4.0-Komponenten merkmalsbasiert beschreibt und (ii) die Sprache für I4.0-Komponenten [4, 5], welche im wesentlichen die Nachrichtenstruktur und Interaktion für ein Ausschreibungsverfahren zwischen I4.0-Komponenten entlang gewisser Regeln und Rollen spezifiziert. Außerdem spielen für die Verhandlungen nicht nur eine präzise Dinenstbeschreibung eine Rolle, es müssen auch Kriterien wie Zeit, Kosten, Aufwand oder andere relevante Charakteristika in Betracht genommen werden. Des weiteren müssen die Erwartungen der Verhandlungsteilnehmer erfüllt sein wenn es um die Umsetzung der geforderten Dienstleistungen geht.
Diese Arbeit konzentriert sich auf die Entscheidungsprozesse beim Bieterverfahren während einer Verhandlung zwischen I4.0-Komponenten. Dazu soll ein konkreten Beispiels aus der auftragsgesteuerte Produktion in einer Industriellen Umgebung betrachtet werden. Das Beispiel beschreibt die automatisierte Verhandlung um einen Lagerplatz zwischen einem Produkt und einem Lager und wird in Kap. 2 beschrieben. Danach werden in Kap. 3 die Entscheidungsprozesse im Interaktionsprotokoll nach dem Bieterverfahren analysiert und im Anschluss eine mögliche Umsetzungvariante für eine I4.0-Umgebung beschriebend die im Wesentlichen mit Verwaltungsschalen-Applikationen und reaktiver Verwaltungsschalen realisiert, welche mittels des Teilmodells „BiddingprocedureConfig“ miteinander interagieren können. Im Anschluss wird die Arbeit zusammengefasst und ein Ausblick geschaffen.

2 Anwendungsfall

Beim Anwendungsfall handelt es sich um die Einlagerung eines Produkts in ein Lager. Für die Einlagerung soll zwischen dem Produkt und dem Lager eine Verhandlung stattfinden. Gegenstand der Verhandlung ist die Dienstleistung, die den Einlagerungsvorgang beschreibt. Das Ziel der Verhandlung für das Produkt ist es unter Berücksichtigung vorher bestimmter Kriterien einen Lagerplatz im Lager zu bekommen. Das Ziel des Lagers ist es einen Preis für ein Angebot anhand der Belegung zu bestimmen und eine Zusage für das abgegebene Angebot zu bekommen.
Dafür sollen eine I4.0-Umgebung umgesetzt werden die folgenden Anforderungen erfüllt:

- Das Produkt und das Lager haben ein proaktives Verhalten und führen die Verhandlungen automatisiert mit dem Bieterverfahren durch.
- Das Produkt und das Lager sollen I4.0 Komponenten sein und sich in einer I4.0 Umgebung befinden.

- Entscheidungen bei der Auswahl des Angebots sollen anhand der vorher bestimmten Kriterien getroffen werden (z. B. beim Erwerb des Produktes beim Verkäufer).

Zusätzlich sollen folgende Rahmenbedingungen eingehalten werden:

- Das Produkt hat eine Verwaltungsschale die bereits mit Informationen angereichert worden sind (z. B. durch den Entwickler oder Verkäufer des Produkts). Diese Informationen sind in Teilmodellen in der VWS modelliert.
- Die Dienstleistung ist als Teilmodell beschrieben und wird von beiden Parteien verstanden.
- Die VWS der Produkte werden als passive Verwaltungsschalen dem Produzenten übergeben und von einer VWS-Serverapplikation als reaktive VWS bereitgestellt.
- Es gibt eine Registry in der die VWS auffindbar sind.
- Die VWS des Lagers wird durch eine eigene VWS-Serverapplikation bereitgestellt und kann Logiken implementiert haben.

3 Analyse von Entscheidungsprozessen

In Abb. 1 wird das Interaktionsprotokoll des Bieterverfahrens nach dem das Produkt und das Lager interagieren dargestellt. Das Produkt nimmt die Rolle des Service Requester (SR) ein und das Lager die Rolle des Service Providers (SP).
Um proaktive VWS für den SP und SR umzusetzen, die das Interaktionsprotokoll nach dem Bieterverfahren umsetzen, sollten diese aus einem aktiven und einem passiven Teil bestehen (siehe [6]). Der aktive Teil einer Proaktiven VWS besteht aus Elementen wie einem Interaktionsmanager, einem Messenger und Algorithmen, die für die Enstscheidungsprozesse während der Verhandlung benötigt werden.
Um die Rahmenbedingungen aus Kap. 2 einzuhalten werden für die Elemente Applikationen und Operationen Implementiert welche die einzelnen Aufgaben während der Verhandlungen übernehmen.
Die Überprüfung der eingegangenen Ausschreibungen und die Berechnung des Preises für ein Angebot auf der SP Seite übernimmt eine Operation, die in der VWS-Serverapplikation des Lagers integriert wird. Sie prüft die Ressourcen über die der SP verfügt und bietet diese, wenn sie nicht belegt sind, als Angebot an.
Für die Entscheidung über die eingegangenen Angebote wird eine Auswahl-Applikation implementiert. Die Auswahl-App wählt eine der eingegangen Angebote anhand der vorgegebenen Kriterien welche in der reaktiven VWS des SR festgelegt wurden aus und versendet eine Bestätigung an den SP.
Nachdem die Bestätigung eingegangen ist belegt eine Operation die Angebotene Ressource und bestätigt die Angebotsannahme.
Die Aufgabe des Interaktionsmanagers und des Messengers übernimmt die Bidding-Applikation. Sie generiert die Nachrichten für die Ausschreibung und für die Angebote

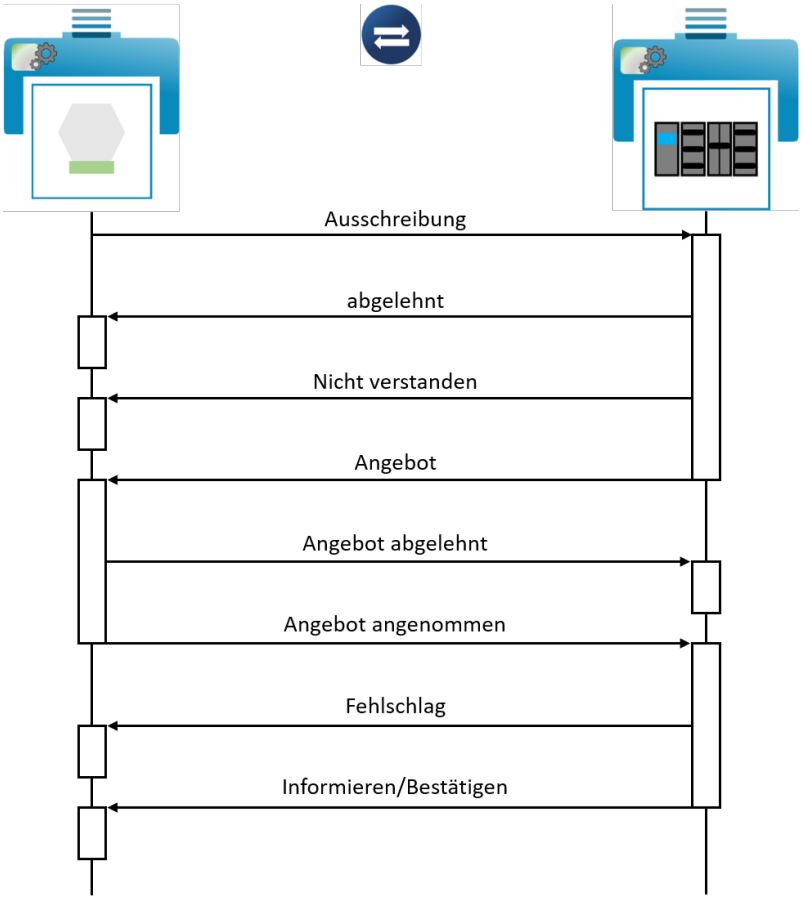

Abbildung 1: Verhandlungen zwischen Produkt und Lager mittels Bieterverfahren nach [4, 2]. Das Produkt (links) nimmt die Rolle des Service Requesters und das Lager die des Service Provider ein.

und tauscht diese mit anderen Verhandlungsteilnehmern aus. Dabei führt sie das Interaktionsprotokoll nach dem Bieterverfahren aus und kann während der Verhandlungen Teilmodelle des Applikationsnutzers (z. B. von der VWS des Produkts) verändern. Zusätzlich kann sie andere VWS-Applikationen und Operationen die oben genannt wurden Aufrufen. Dazu soll ein Teilmodell modelliert werden das mit Teilmodellelementen die Interaktion zwischen den Applikationen für die Entscheidungsfindung ermöglicht.

4 Umsetzung der I4.0 Umgebung

Bei der Umsetzung der I4.0 Umgebung werden zunächst alle Komponenten beschrieben, die an der Verhandlung teilhaben. Dazu gehört wie sie Umgesetzt wurden, welche Funktionalitäten sie haben und welche Aufgaben sie ausführen müssen. Danach wird beschrieben wie die einzelnen Komponenten miteinander interagieren und welche Informationen dabei ausgetauscht werden.

4.1 Komponenten in der I4.0 Umgebung

Die Umgebung in der die Verwaltungsschalen interagieren besteht aus mehreren Komponenten (siehe Abb. 2) die für die Verhandlung nötig sind.
Das Produkt ist der Service Requester (SR) und das Lager ist der Service Provider (SP) bei der Verhandlung um einen Lagerplatz. Die Verwaltungsschalen werden von verschiedenen VWS-Serverapplikationen bereitgestellt.
Die Serverapplikation für die Produkte greift auf AASX-Dateien zu und stellt diese als reaktive Verwaltungsschalen bereit. Nachdem die Verwaltungsschalen der Produkte bereitgestellt sind, werden sie in der Registry registriert. Die Serverapplikation welche die Verwaltungsschalen der Produkte bereitstellt ist der AASX-Server [7].
Die Verwaltungsschalen des Lagers, der Auswahl-App und der Bidding-App werden jeweils von einer eigenen AAS-Serverapplikation bereitgestellt. Anders als beim Produkt wurden diese in Java unter Verwendung des Eclipse BaSyx SDK [8] implementiert, das das VWS-Metamodell umsetzt und Funktionen wie die (De-)Serialisierung bereitstellt. Mit der Implementierung können bei der Umsetzung der VWS z. B. Operationen implementiert werden die Logik beinhalten. Diese Verwaltungsschalen haben ein proaktives Verhalten oder können eins mittels Interaktion mit einer reaktiven VWS generieren.

4.1.1 Bidding-App und Auswahl-App

Die Auswahl- und die Bidding-App haben als Asset eine Applikation und werden fortan VWS-Applikationen genannt. Mit der Interaktion zwischen einer VWS-Applikation mit einer reaktiven VWS wird ein proaktives Verhalten generiert.
Die Bidding-App implementiert Interaktionen von Verwaltungsschalen nach dem Bieterverfahren (siehe [4, 5]), wobei sie beide Seiten bedienen kann: SR und SP. Als SP bietet sie einem SR eine Dienstleistung an, wenn sein zugehöriges Asset fähig und verfügbar ist, um die Anfrage zu erfüllen. Während sie als SR versucht, einen geeigneten SP für die Dienstleistung zu finden, den sein zugehöriges Asset benötigt. Um die Bidding-App auszuführen wurden für den SP und SR jeweils eine Operation, die über die Rest-API der VWS aufgerufen werden kann, integriert. Die Bidding-App nutzt während der Verhandlungen den MQTT-Broker, um die Nachrichten auszutauschen.
Die Auswahl-App ist für die Auswahl eines geeigneten Angebots zuständig. Diese Auswahl ist ein Entscheidungsprozess im Bieterverfahren und wird auf der SR Seite aus-

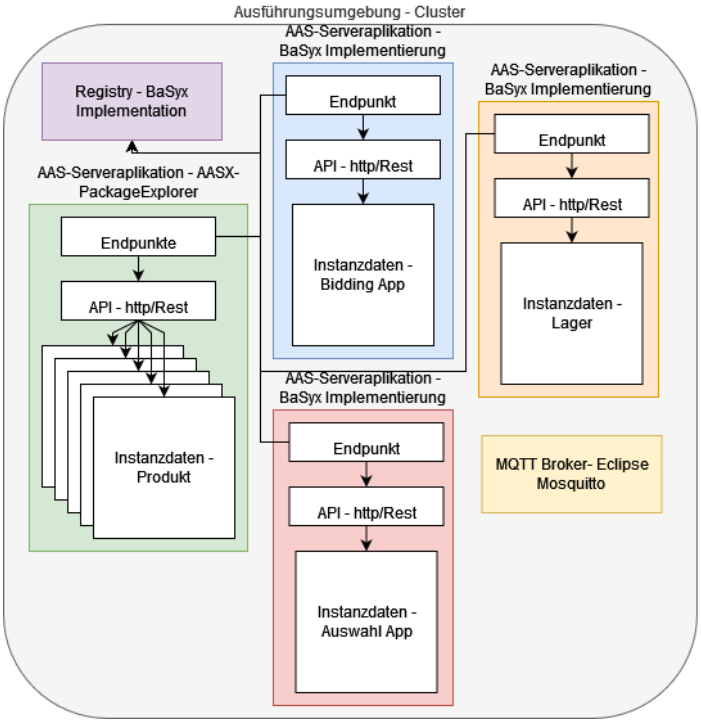

Abbildung 2: Die Abbildung zeigt alle an der Verhandlung beteiligten Komponenten zwischen Produkt (grün) und Lager (orange). Die BiddingApp (blau) und der MQTT-Broker sind für das einhalten des Interaktionsprotokols und den Nachrichtenaustausch zuständig. Alle Komponenten (außer der MQTT-Broker) besitzen eine VWS und sind in der Registry (lila) mit ihren Endpunkten registriert. (frei nach [9])

geführt. Die Auswahl-App hat die Aufgabe aus allen eingegangenen Angeboten nach einem vorgegebenen Kriterium ein geeignetes auszuwählen. Auch die Auswahl-App kann via Operation in der VWS über die Rest-API aufgerufen werden.

4.1.2 Verwaltungsschalen der Produkte

Die Verwaltungsschalen der Produkte sind reaktive Verwaltungsschalen bei denen über eine Rest-API Teilmodelle und Teilmodellelemente gelesen, geschrieben, gelöscht und aktualisiert werden können. Die Verwaltungsschale eines Produktes hat in diversen Teilmodellen beschrieben, welche Anforderungen das Produkt an den Produktionsprozess hat. So ist u. A. auch die Dienstleistung, die für den Einlagervorgang benötigt wird in einem Teilmodell beschrieben. Dieser wird als Interaktionselement für das Bieterverfahren benötigt, um eine Ausschreibung für einen Lagerplatz auszusenden. Weiterhin ist in einem Teilmodell das Kriterium, nach welchem diese Produziert werden soll

beschrieben. Diese Kriterien können Zeit, Preis und Aufwand sein.

4.1.3 Verwaltungsschale des Lagers

Das Lager stellt eine bestimmte Anzahl von Lagerplätzen zur Verfügung und bietet sie als Dienstleistung mit dem Bieterverfahren an. Die Dienstleistung ist in einem Teilmodell beschrieben und gleicht der Dienstbeschreibung der Dienstleistung für den Einlagervorgang in der VWS des Produkts.
Da es sich bei der Umsetzung der VWS-Serverapplikation für die VWS des Lagers um eine Implementierung mit Java unter Verwendung des Eclipse BaSyx SDK handelt sind zusätzlich für den hier beschriebenen Anwendungsfall zwei Operationen integriert. Zum einen der Entscheidungsprozess (CheckAndPricing) für die Überprüfung der eingehenden Nachrichten und der Preisberechnung und zum anderen der Entscheidungsprozess (OccupyAndInform) in dem die VWS des Lagers die Ressourcen belegt und den SR über den Vertragsabschluss informiert. Diese Entscheidungen werden im Bieterverfahren auf der SP Seite ausgeführt.

4.2 Interaktion zwischen den Komponenten

Die VWS des Lagers und des Produkts sind für sich genommen nicht in der Lage mittels Bieterverfahren miteinander Verhandlungen zu führen. Erst nach Aufruf einer der Operationen der Bidding-App und der Auswahl-App können sie Entweder als SP oder SR an Verhandlungen zwischen I4.0-Komponenten teilnehmen. Für den Aufruf über die Rest-API der VWS-Applikation werden jeweils Registry-URL, AAS-ID und die Teilmodell-ID der Dienstbeschreibung benötigt (siehe Abb. 3).

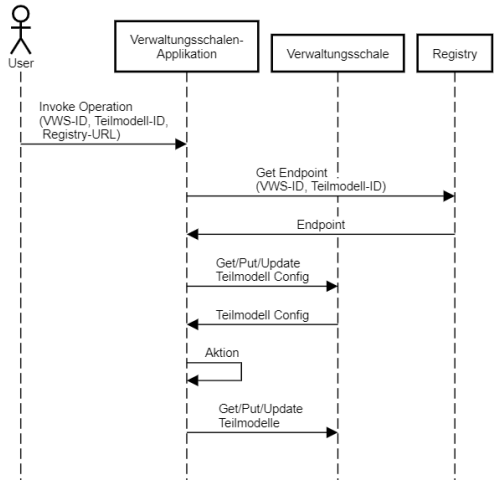

Abbildung 3: Aufruf und Interaktion einer VWS-Applikation mit einer VWS.

Mit diesen Informationen kann sich die VWS-Applikation mit der Verwaltungsschale

des Lagers bzw. des Produktes verbinden und mit ihr interagieren. VWS des Lagers bzw. des Produkts stehen nach Aufruf der Operation mit der VWS-Applikation in einer Server-Client-Beziehung.
Die Interaktion findet wie folgt statt:

1. Zunächst verbindet sich die VWS-Applikation mit der Verwaltungsschale dessen ID als Parameter übergeben wurde. Dies erreicht sie, indem sie den Endpunkt in der Registry anfordert.

2. Dann holt sie sich ein Teilmodell für die Konfiguration der Operation. Z. B. im Fall der Bidding-App das Teilmodell BiddingProcedureConfig welche in Kap. 4.3 näher erläutert wird.

3. Ausführen einer Aktion z. B. des Bieterverfahren bei der Bidding-App oder der Auswahl eines Angebots bei der Auswahl-App.

4. Während der Aktion und danach steht die VWS-Applikation und die VWS in Verbindung und tauschen Daten in Teilmodellen und Teilmodellelementen aus.

Die VWS-Applikation Bidding-App wird sowohl von der Lager als auch von dem Produkt Verwaltungsschalen verwendet. Die Bidding-App generiert die Nachrichten und tauscht diese nach dem Interaktionsprotokoll aus. Sie trifft keine Entscheidungen ob Angebote angenommen werden und berechnet nicht den Preis.
Diese Aufgaben werden von der Auswahl-App auf der SR-Seite und von der in der Verwaltungsschale des Lagers integrierten CheckingAndPricing Operation und der OccupyAndInform Operation auf der SP Seite ausgeführt.

4.3 BiddingProcedureConfig

Das Teilmodell BiddingProcedureConfig (siehe Tab. 1) wird zur Konfiguration der Bidding-App benötigt und wird für die Interaktion mit den VWS-Applikationen verwendet. Dieses Teilmodell muss für das Bieterverfahren in die VWS des Produkts und des Lagers integriert werden.
Im folgenden wird beschrieben wie die Bidding-App, sowohl auf als SR als auch als SP, nach dem Aufruf mit anderen Komponenten interagiert. Diese Beschreibung ergänzt die Interaktion welche in der Abb. 3 durch Aktion abgebildet ist.
Ablauf als SR:

1. Konfiguration des MQTT-Clients mit den Informationen in der Teilmodellelementsammlung „mqttClientConfig“.

2. Generieren und Verschicken des CallForProposal anhand der Teilmodell-ID die während des Aufrufs übergeben wurde. Das Teilmodell beschreibt die Dienstleistung.

3. Warten und Sammeln von eingehenden Angeboten. Diese werden in der Teilmodellelementsammlung „srProposals“ hinterlegt. Die Zeit, die dabei abgewartet wird, ist im Property „srTimeCollect“ festgelegt.

Tabelle 1: Teilmodell BiddingProcedureConfig. (RefEl= ReferenzeElement, SMEC=Teilmodellelementsammlung.)

idShort	**Typ**	**Beschreibung**
mqttClientConfig	SMEC	Konfiguration des MQTT-Client
srTimeCollect	Property	Zeit die für Sammlung
srProposals	SMEC	Sammlung aller eingegangenen Angebote
srSelection	RefEl	Referenz auf Operation der VWS-Applikation Auswahl-App
srCriterion	RefEl	Referenz auf Kriterium die bei der Auswahl berücksichtig werden soll
srSelected	RefEl	Referenz auf das ausgewählte Angebot
spCallForProposals	SMEC	Sammlung aller anfragen auf Angebote
spProposal	RefEl	Referenz auf das Angebot das unterbreitet wird
spCheckAndPricing	RefEl	Referenz auf Operation CheckAndPricing
spOccupyAndInform	RefEl	Referenz auf Operation OccupyAndInform

4. Aufruf der Operation der Auswahl-App, auf die im Referenzelement „srSelection“ verwiesen wird.
5. Das ausgewählte Angebot auf welches „srSelected“ referenziert annehmen.
6. Nach eingegangener Bestätigung den Produktionsschritt als Abgeschlossen markieren.

Ablauf als SP:

1. Konfiguration des MQTT-Clients mit den Informationen in der Teilmodellelementsammlung „mqttClientConfig“.
2. Eingehende Proposals in der Teilmodellelementsammlung „spCallForProposals“ hinterlegen
3. Aufruf der Operation „CheckAndPricing“, auf die im Referenzelement „srSelection“ verwiesen wird.
4. Generieren eines Proposals anhand der Teilmodell-ID die während des Aufrufs übergeben wurde. Das Teilmodell beschreibt die Dienstleistung.
5. Warten auf Angebotsannahme.
6. Aufruf der Operation „spOccupyAndInform“ um den Lagerplatz als reserviert zu kennzeichnen und dem SR das angenommene Angebot zu bestätigen.

5 Zusammenfassung

Es wurde ein konkretes Beispiel aus der auftragsgesteuerten Produktion in einer Industriellen Umgebung betrachtet. Das Beispiel beschrieb die automatisierte Verhandlung um einen Lagerplatz zwischen einem Produkt und einem Lager und wurde in Kap. 2 erläutert. Nachdem in Kap. 3 die Entscheidungsprozesse im Interaktionsprotokoll nach dem Bieterverfahren analysiert wurden, konnte eine mögliche Umsetzungsvariante für eine I4.0-Umgebung beschrieben werden die im Wesentlichen mit Verwaltungsschalen-Applikationen und reaktiver Verwaltungsschalen realisiert wird. Diese interagieren mittels des Teilmodells „BiddingprocedureConfig“ miteinander.

Literaturverzeichnis

[1] Grunau, Sergej and Redeker, Magnus and Göllner, Denis and Wisniewski, Lukasz: The Implementation of Proactive Asset Administration Shells: Evaluation of Possibilities and Realization in an Order Driven Production. Eingereicht in 11. Jahreskolloquium Kommunikation in der Automation (KommA 2020) in Lemgo 2020.

[2] Plattform Industrie 4.0: Anwendungsszenario trifft Praxis: Auftragsgesteuerte Produktion eines individuellen Fahrradlenkers. 2017.

[3] Plattform Industrie 4.0: Details of the Administration shell - Part 1: The exchange of information between partners in the value chain of Industrie 4.0. 2020.

[4] VDI/VDE 2193 Blatt 1: Sprache für I4.0-Komponenten,“ Düsseldorf: VDI, 2019

[5] VDI/VDE 2193 Blatt 2: Sprache für I4.0-Komponenten. Interaktionsprotokoll für Ausschreibungsverfahren. Düsseldorf: VDI, 2019

[6] VWS vernetzt: Specification Testbed „AAS networked “: Proactive AAS - interaction according to the VDI/VDE 2193. Berlin: BMWi, 2020. URL: https://vwsvernetzt.de/projektinhalte/spezifikationen-downloads/

[7] AASX-Server. URL: https://github.com/admin-shell-io/aasx-server, accessed: 01.07.2021

[8] Eclipse BaSyx. URL: https://wiki.eclipse.org/BaSyx, accessed: 01.07.2021

[9] Plattform Industrie 4.0: Was ist die Verwaltungsschale aus technischer Sicht?. 2021.

http://dx.doi.org/10.25673/39575

Anwendungsbeispiele eines offenen Industrial APP Marketplace

Patrick Heidemann, Sascha Heymann, Nissrin Perez, Jan Alsters
Fraunhofer IOSB-INA
Institutsteil für industrielle Automation

Campusallee 1
32657 Lemgo
patrick.heidemann@iosb-ina.fraunhofer.de
sascha.heymann@iosb-ina.fraunhofer.de
nissrin.perez@iosb-ina.fraunhofer.de
jan.alsters@stud.th-owl.de

Abstract: Ein Anwendungsmarktplatz ist eine digitale Vertriebsplattform für Anwendungen (APPs), die meist im mobilen Kontext genutzt werden. Marktplätze wie der App Store von Apple oder Googles Play Store zeigen, wie florierend das Marktplatzgeschäft ist. Diese bieten Entwicklern eine Plattform für den Vertrieb ihrer Anwendungen und erreichen durch ein gemeinsames Ökosystem einen breiten Anwenderkreis. Ein Industrial APP Marketplace, kombiniert mit dem Industrial Internet of Things (IIoT), bietet im industriellen Bereich eine Lösung für eine schnellere Integration und Inbetriebnahme von industriellen Anwendungen in der Produktion aber z.B. auch Lösungen für den SmartCity Bereich. Dieses Paper zeigt zwei Anwendungsbeispiele eines herstellerneutralen Industrial APP Marketplaces, an den besondere Anforderungen an Interoperabilität, Usability und Sicherheit gestellt werden. Bei der Bewertung der Anwendungsbeispiele wird die gesamte Umsetzungskette vom Kauf bis zum Deployment einer APP auf ein Endgerät betrachtet. Bei den Anwendungsbeispielen handelt es sich um einen IIoT Ansatz, der eine direkte Internetverbindung der Endgeräte vorsieht und einen Edge Ansatz, bei dem Endgeräte in einem lokalen Netzwerk nur über ein gesichertes Edge-Gateway mit dem Internet kommunizieren können.

1 Einleitung und Motivation

Während im Konsumentenbereich Marktplätze breite Anwendung finden, i st i m industriellen Bereich das Potenzial noch nicht erschlossen. Dies liegt an der fehlenden Architektur von interoperablen und einfachen Marktplätzen für industrielle Anwendungen [Hey21]. Dabei bietet das IIoT der Industrie neue intelligente Technologien als Lösung. Auch die Nachfrage für leistungsstarke Feldgeräte mit Edge-Computing Funktionen wächst stetig und damit auch die Anzahl der Hardware-Hersteller. Da in diesen Feldgeräten meist heterogene CPUs und eigene maßgeschneiderte Linux-Distributionen oder Echtzeitbetriebssysteme verwendet werden, ist es im Vergleich zu mobilen Geräten für unabhängige Entwickler eine Herausforderung interoperable APPs zu entwickeln. Aus diesem Grund bietet die Containervirtualisierung als fundamentale Abstraktionsschicht die Basis für APPs auf industriellen Geräten. Anwendungen werden in standardisierte Einheiten verpackt, welche als Container bezeichnet werden und zusätzlich zu dem Quellcode der Anwendung auch die Runtime, Systemtools und Bibliotheken beinhalten. Das ermöglicht den Software-Entwicklern APPs im Feld auf modulare Art und Weise plattformunabhängig zu entwickeln [MBP20]. Dies und eine Lösung für einen unkomplizierten Rollout der APPs auf die Feldgeräte sind die beiden grundlegenden Voraussetzungen für einen offenen industriellen APP-Marktplatz [Bar20].

2 Stand der Technik

2.1 Industrial APP Marketplace

Der Industrial APP Marketplace etabliert einen herstellerneutralen Marktplatz für I4.0-Komponenten zum Kauf und Verkauf von industrieller Hardware, Software und Dienstleistungen. Grundsätzlich versteht man unter APPs moderne Software-Anwendungen, die modular aufgebaut sind und ein benutzerfreundliches grafisches Interface haben, im Kontext eines Industrial APP Marketplace aber auch Dienste und Lösungen. Die aus einer Kooperation zwischen OWL Maschinenbau, der SmartFactoryOWL und der Open Industry 4.0 Alliance Community in 2020 entstandene Idee ist mittlerweile ein Zusammenschluss mehrerer Unternehmen und weiterer Akteure, die zum einen gemeinsam an der Plattform weiterentwickeln und zum anderen als APP Entwickler, Hardware Entwickler oder Integrator eigene Services auf dem Industrial APP Marketplace anbieten. Die gemeinsame Schnittmenge dieser drei Rollen und der Rolle des App-Benutzers bilden den Industrial APP Marketplace und damit ein hohes Innovationspotential. [Hey21]

2.2 OI4 Community App-Store

Der *Open Industry 4.0 Alliance Community App Store* ist das erste von der Allianz abgesegnete Projekt. Der App-Store wurde im Rahmen der Hannover Messe 2021 Digital vorgestellt. Die Idee hinter dem abgekürzt bezeichneten OI4 Community App-Store sei, dass das, was für mobile Endgeräte im Consumer-Bereich gut funktioniere, auch ein Modell für den industriellen Markt sein könnte. Auf Edge-Geräten laufen schon heute unterschiedliche Container-Applikationen diverser Hersteller, die häufig miteinander interagieren müssen. Als Non-Profit-Organisation erteilt die Allianz deshalb dem App-Store der Hilscher-Gesellschaft für Systemautomation mbH erstmals das Label „Open Industry 4.0 Alliance Community“. Der OI4 Community App-Store soll den Mitgliedern als eine übergreifende Verkaufsplattform für ihre Apps zur Verfügung gestellt werden, die ohne Investitions- und Unterhaltskosten auskommt. Der Fokus ist hier auf Apps, die auf der mit der Allianz konformen *Open Edge Computing Plattform* laufen. Der OI4 Community App-Store stehe ab Ende des dritten Quartals 2021 für Mitglieder der Open Industry 4.0 Alliance bereit, die dann getestete, Docker-kompatible sowie Hardware-unabhängige industrielle APPs beziehen können. [Man19]

2.3 PLCNext Store

Der PLCnext Store ist eine Online-Handelsplattform, in der Softwarefunktionen für die Steuerungsplattform sowohl von Phoenix Contact als auch von Drittanbietern zum Herunterladen bereitstehen. Das Angebot umfasst Funktionsbausteine, Funktionserweiterungen, Cloud Connectoren bis hin zu weiteren Laufzeitumgebungen wie z.B. Codesys, wodurch die PLCnext Steuerungen von Phoenix Contact auch mit weiteren Programmiersprachen erweitert werden können. Der PLCnext Store steht Software-Entwicklern als Handelsplattform für eigen entwickelte APPs zur Verfügung. Dabei legen die Entwickler den Preis für ihre Software-Lösungen fest, die Lizenzierung und Kaufabwicklung erfolgt über Phoenix Contact. Die Installation kann online oder offline auf ein Gerät übertragen werden. [DB20]

3 Anforderungen

Um einen Industrial APP Marketplace als Vertriebskanal in ein Geschäftsmodell integrieren zu können, muss dieser Marktplatz bestimmte Anforderungen erfüllen und sowohl Kunden, als auch Anbietern einen niederschwelligen Einstieg ermöglichen. Maxim Barbarinow nennt in einem LinkedIn-Artikel[1] zehn Anforderung für einen Industrial APP Marketplace, die in folgender Tabelle gelistet werden:

[1] https://www.linkedin.com/pulse/industrial-iot-application-store-next-big-thing-maxim-babarinow/

	Benennung	Beschreibung
1	Gerätezertifizierung	Zertifizierung zur Kompatibilität zwischen Geräten und Anwendungen
2	Anwendungszertifizierung	Zertifizierung zur Robustheit, Performanz, Sicherheit und Datenschutz der Applikationen
3	Diversität der Endgeräte	Unterstützung einer breiten Palette von Geräten
4	1-Klick Deployment	Remote-Bereitstellung von Anwendungen für Geräteflotten direkt vom Marktplatz aus
5	Anwendungskonfiguration	Unterstützung von Anwendungskonfigurationen
6	Deployment und Anwendungsüberwachung	Statusüberwachung von Gerätefunktionen mit Hilfe von Diagnoseprotokollen
7	Anwendungslizenzierung	Bereitstellung von Lizenzierungen und Urheberschutzrechten für Anwendungen durch Marktplatzbetreiber
8	Geschäftsmodell-Flexibilität	Unterstützung verschiedener Lizenzmodelle und Zahlungsmethoden
9	Seller's Insights	Prognosen von Nachfragen und Trends zur Unterstützung der Verkäufer
10	Multi-Country Support	lokale und mehrsprachige Supportstruktur

Lt. Barbarinow müssen technische Zertifizierungen vorangestellt werden, um die Funktionalität neuer Anwendungen auf vorhandene Geräte sicherzustellen. Die Installation sollte durch offene Prozessarchitekturen, die verschiedene Protokolle unterstützen, sowie durch die Ermöglichung eines 1-Klick Deployments vereinfacht werden. Durch die Unterstützung von Startkonfigurationen und einer Deployment Kontrolle kann sichergestellt werden, dass das Deployment erfolgreich war. Lizenzbedingungen für einen Urheberschutz und für die Abbildung verschiedener Lizenzmodelle und daraus folgender Zahlungsmethoden sollten auf dem Marktplatz geschaffen werden. Interessant sind für die Anbieter zudem Verkaufsanalysen, welche durch die Erhebung und Analyse von entstehenden Daten entstehen. Und zuletzt ist es essentiell, einen breiten Markt zu schaffen, indem die Plattform international mit entsprechenden Sprachen, Währungen, Zahlungsmethoden etc. aufgebaut wird. Neben diesen zehn von Barbarinow gelisteten Anforderungen nennt er weitere Erfolgskriterien, wie eine gute Auffindbarkeit des Marktplatzes, ein positives Nutzererlebnis, eine Partnerverwaltung mit Wiederverkaufsoptionen, Rezensionen und Empfehlungen, Supportangebote und Unterstützung der Finanzbuchhaltung [Bar20]. Auch on-demand Services und mögliche Rollbacks der Anwendungsbereitstellung werden als sinnvolle funktionale Ergänzungen des Marktplatzes angesehen [Bar20]. Eine wichtige Voraussetzung, welche die Open Industry 4.0 Alliance benennt, ist die Nutzung einer offenen auf RAMI 4.0 basierten Architektur mit den Bausteinen 1. Device Connectivity, 2. Edge Operator Cloud, 3. Cloud Central und 4. einem entsprechenden Dienstleistungsangebot. [**?**] Nur durch die Verwendung offener Standardschnittstellen auf Grundlage der Industrie 4.0 Verwaltungsschale kann lt. der Open Industry 4.0 Alliance eine automatisierte Integration von Asssets – das sogenannte „Asset Automatic Onboarding“ realisiert werden. [Man19]

4 Technologien

4.1 Eclipse HawkBit

HawkBit ist ein domänen-unabhängiges Backend-Framework zum Ausrollen von Software-Updates und Firmware auf IoT-Geräten. Dies ist eine open-source Software der Eclipse Foundation und besteht aus einem Update-Server, welcher das Update- und Rollout-Management, Content-Delivery und ein Geräte- und Software-Repository besitzt. Zusätzlich werden drei Schnittstellen in Form von WEB-APIs und ein Webinterface bereitgestellt. Der Software-Rollout kann entweder direkt über die *Management-API* oder über das Webinterface(*Management-UI*) verwaltet werden. Für die direkte Geräteintegration bietet *hawkBit* eine einfache REST-API(*Direct-Device-Integration API*). Eine Indirekte Geräteintegration(andere Konnektivitätsschicht) wird mit der *Device-Management-Federation-API* bereitgestellt. Diese ist erforderlich wenn ein eingeschränktes Gerät keine TLS/HTTP-Verbindung verarbeiten kann, aber ein Standard-Geräteverwaltungs-Protokoll wie OMA-DM oder LWM2M unterstützt. [Fou21]

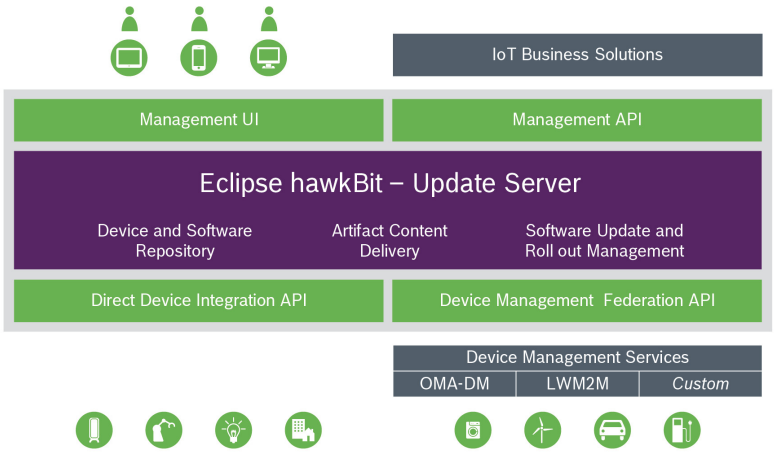

Abbildung 1: HawkBit Architektur

4.2 Docker

Docker ist eine quelloffene Software zur Isolierung von Anwendungen mit Hilfe von Container-Virtualisierung. Docker-Container sind standardisierte Einheiten, welche neben der Applikation auch Bibliotheken, Systemtools und Laufzeit beinhalten. Dadurch ist es möglich, Anwendungen in jeder Umgebung schnell bereitzustellen und zu skalieren. Ein Docker-Image bildet die Basis für einen Container und kann in einem Repository wie DockerHub gespeichert und verwaltet werden. [Edu21]

5 Anwendungsbeispiele

In diesem Kapitel werden zwei verschiedene Ansätze für eine Deployment-Lösung von APPs auf Feldgeräten vorgestellt.

5.1 Grundlagen

5.1.1 Artefakt

Ein Artefakt ist in diesem Kontext eine Datei, welche bei dem Rollout-Prozess auf die Geräte geladen wird. Hierbei handelt es sich nicht um eine Applikation, sondern lediglich um eine Manifest-Datei, welche alle relevanten Information über eine APP enthält. Dazu gehören das Docker Basis-Image, der APP Name, benötigte Ports und weitere Parameter. Als APP wird dann der lauffähige Docker-Container bezeichnet.

5.1.2 hbloader

hbloader ist ein erweiterter HawkBit-Client, welcher in der Programmiersprache Python geschrieben ist und auf dem rauc-hawkBit-Client aufgebaut ist. Der vorkonfigurierte Client meldet sich über die *Direct-Device-Integration-API* am *HawkBit* an und prüft zyklisch über die *Management-API*, ob ein neues Deployment aussteht. Sobald dies der Fall ist, wird das entsprechende Artefakt heruntergeladen und mit Hilfe einer Prüfsumme auf Echtheit geprüft. Anschließend wird ein Docker-Container mit den angegebenen Parametern gestartet.

5.1.3 hbpull

hbpull ist eine Software zur Koppelung einer öffentlich erreichbaren und einer lokalen HawkBit-Instanz. Ein HawkBit-Client meldet sich als Gerät bei der Cloud-HawkBit-Instanz an und überträgt das Artefakt bei einem Rollout an die lokale HawkBit-instanz und legt die APP dort als Software-Modul an.

5.2 IIoT-Ansatz

Bei dem IIoT-Ansatz ist jedes Feldgerät, in diesem Fall eine IoT-Sensorbox, direkt mit dem Internet verbunden.
Nach dem Erwerb einer APP wird das entsprechende Artefakt an die *HawkBit*-Instanz übertragen und dort als Software-Modul angelegt. Über das Management-UI kann das Modul dann in eine Distribution gelegt und diese auf verfügbare Feldgeräte ausgerollt werden. Der *hbloader* installiert dann die APP als Docker-Container auf dem jeweiligen Gerät. Das benötigte Docker-Image wird von einer öffentlich erreichbaren Docker-Registry heruntergeladen.

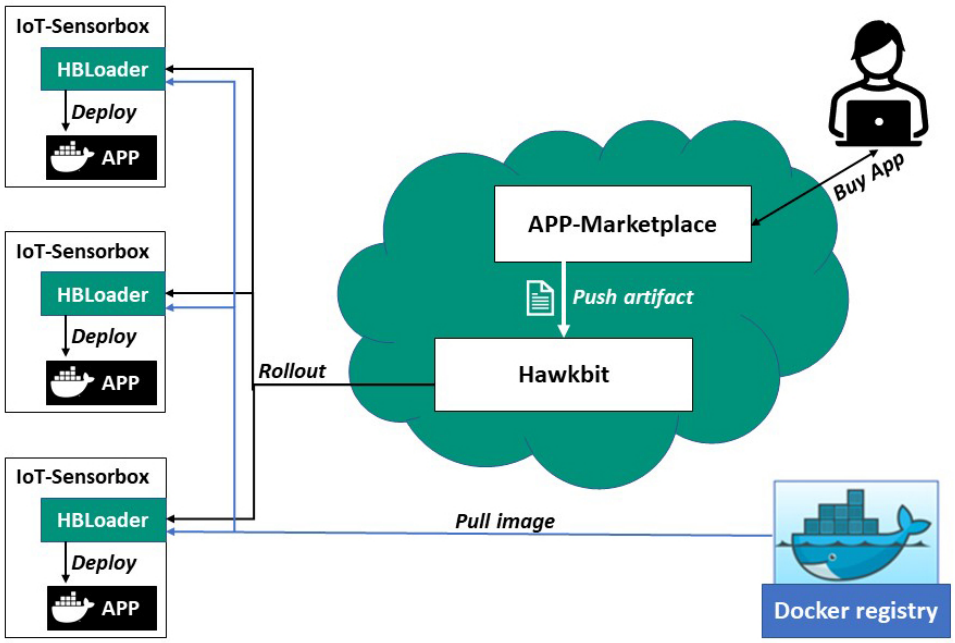

Abbildung 2: APP-Deployment bei dem IIoT-Ansatz

5.3 EDGE-Ansatz

Bei dem Edge-Ansatz sind die Feldgeräte in einem separaten Maschinen-Netz. Ein Edge-Gerät mit zwei Netzwerk-Schnittstellen ist mit diesem und über eine Firewall mit dem Internet verbunden. Auf dem Edge-Gerät läuft eine lokale *HawkBit*-Instanz und eine Docker-Registry.
Nach dem Erwerb einer APP wird diese an das Edge-Gerät übertragen. Dafür ist *hbpull* bei dem im Marketplace integrierten HawkBit als Zielgerät angemeldet und prüft zyklisch auf neue Deployments. Sobald eine neue APP verfügbar ist, wird das Artefakt heruntergeladen, in das lokale *HawkBit* übertragen und dort als Software-Modul angelegt. Von dort aus kann die APP dann über ein *Management-UI* auf die lokalen Feldgeräte ausgerollt werden. Bei der Installation wird das benötigte Docker-Image über die lokalen Docker-Registry heruntergeladen. Diese ist als *Pull-Through-Cache* konfiguriert. Ein Docker-Daemon auf einem der Feldgeräte fordert bei der lokalen Docker-Registry ein Image an. Anschließend wird überprüft, ob das angeforderte Image bereits im lokalen Repository vorhanden ist. Wenn dies der Fall ist, wird das Image direkt an das entsprechende Gerät zurückgegeben. Andererseits wird es von einer öffentlich erreichbaren Registry heruntergeladen und für künftige Anfragen lokal gespeichert.

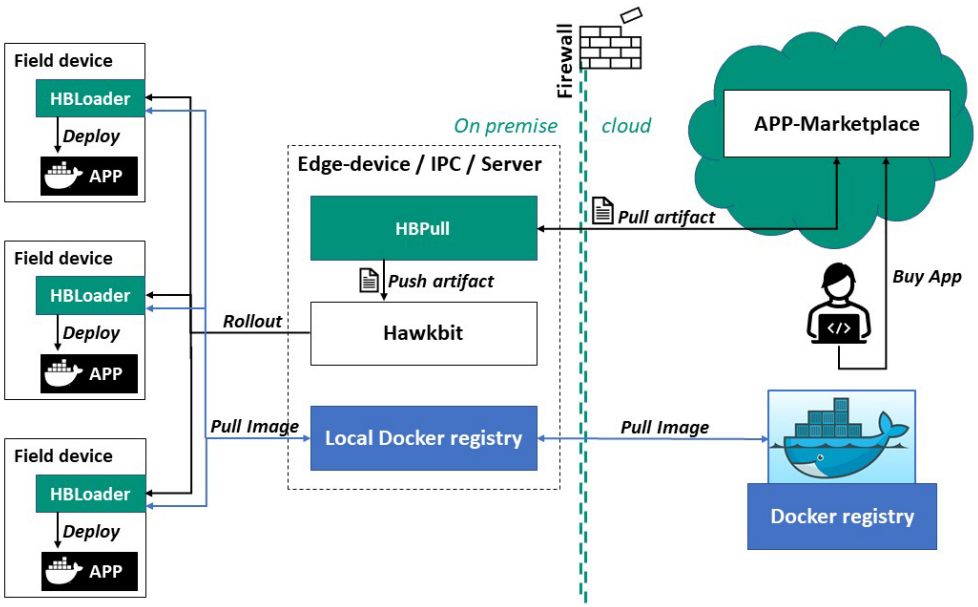

Abbildung 3: APP-Deployment bei dem Edge-Ansatz

6 Geräte- und APP-Sicherheit

Um zu gewährleisten, dass die im Marketplace angebotenen APPs sicher sind, sollten diese einen Prüfprozess durchlaufen. Dabei muss festgestellt werden, ob die bereitgestellte Anwendung robust, performant und sicher ist. Hierfür kann ein Tool wie Nexus-Container[1] verwendet werden. Dieses bietet Sicherheit für den gesamten Lebenszyklus eines Containers, in dem es auf Schwachstellen überprüft und eine Konformitätsprüfung durchführt. Außerdem kann der gesamte Netzwerkverkehr überwacht werden. [Son21] Den Entwicklern müssen zusätzlich Richtlinien und Ressourcen zur Verfügung gestellt werden. [Bar20]

6.1 Lizenzmodelle

Da die Entwicklung von Software ein komplexer, zeitaufwändiger und teurer Prozess ist, führt Software-Piraterie zu großen Einnahmeverlusten für Technologieunternehmen. Um die Nutzung und Weitergabe zu regeln und illegale Kopien zu verhindern, muss der Marktplatz sowohl einen Lizenzierungsmechanismus als auch Lizenzmodelle für Anwendungen unterstützen. [Bar20]
In der folgenden Tabelle sind drei verschiedene Lizenzmodelle aufgelistet:

[1] https://de.sonatype.com/nexus/container

Modell	Beschreibung
Freeware	APPs können kostenlos und unbegrenzt benutzt werden
On Premise	Die Kauflizenz. Für die unbegrenzte Nutzung einer APP wird einmalig gezahlt
Pay per Use	Die Nutzung einer APP wird basierend auf der Nutzung bestimmter Ressourcen abgerechnet

Die Lizenzen könnten entweder per signierte JSON Web Token(JWT[2]) auf die Geräte gebracht werden oder über einen Lizenzserver verwaltet werden.

7 Zusammenfassung

Dieser Beitrag hat zwei Anwendungsbeispiele für einen offenen Industriellen APP Marketplace vorgestellt. Dafür wurden verschiedene Anforderungen betrachtet und basierend darauf zwei Konzepte entwickelt, wie Applikationen auf Feldgeräte ausgerollt und installiert werden können. Bei dem IIoT-Ansatz ist jedes Feldgerät direkt mit dem Internet verbunden und kann über eine Cloud-HawkBit Instanz verwaltet werden. Für den Fall, dass die Feldgeräte aus Sicherheitsgründen nur innerhalb eines lokalen Maschinennetzes erreichbar sind, wird ein zusätzliches Edge-Gerät benötigt. Dieses ist sowohl mit dem internen Maschinennetz als auch über eine Firewall mit dem Internet verbunden.
Die Docker Container Technologie ist ein wichtiger enabler, um einen universellen und offenen Marktplatz zu betreiben. Sie ermöglicht, Software Anwendungen (APPs) anbieterunabhängig auf verschiedensten Plattformen z.B. Steuerungen oder Edge-Gateways zu betreiben. Damit sind Hard- und Software-Lieferant nicht mehr identisch, was wiederum in der industriellen Digitalisierung ganz neue Geschäftsmodelle ermöglicht. Ein weiterer genutzter „quasi“ Standard neben Docker ist *HawkBit* als Rollout Technologie, um Software auf die Feldgeräte, Edge-Koppler, Industrie-PCs, Server und Kubernetes Cluster auszurollen.

[2] https://jwt.io/

Literaturverzeichnis

[Bar20] Maxim Barbarinow. Industrial iot application store: The next big thing?, 2020.

[DB20] Sebastian Gerstl Daniel Buschatzky. Applikationen mit dem plcnext store einfacher und schneller umsetzen, 2020.

[Edu21] IBM Cloud Education. Docker, 2021.

[Fou21] Eclipse Foundation. Eclipse hawkbit, 2021.

[Gö21] Ulrike Götz. Ein app-store für die industrie – eine initiative aus der open industry 4.0 alliance community, 2021.

[Hey21] Sascha Heymann. Industrial app marketplace, 2021.

[Man19] Digital Manufacturing. Was das neue framework der open industry 4.0 alliance kann, 2019.

[MBP20] Jonas Kulawik Maj-Britt Peters. Volkswagen brings additional partners to industrial cloud, 2020.

[mSA15] myfactory Schweiz AG. 4 lizenzmodelle für erp-software im vergleich, 2015.

[Son21] Sonartype. Nexus container integration, 2021.

http://dx.doi.org/10.25673/39572

OEE-Box – Einfacher Einstieg in die Welt der OEE

Jörg Wollert, Marc Gröniger

Institut für angewandte Automation und Mechatronik
Aachener-und-Münchener Allee 1
52074 Aachen
wollert@fh-aachen.de, groeniger@fh-aachen.de

Die Konzeptideen rund um Industrie 4.0 sind der Schlüssel für eine effiziente zukunftsorientierte Fabrikautomatisierung. Mit den Kernzielen horizontaler und vertikaler Integration, sowie einem durchgängigen Engineering sind hohe Anforderungen an die Feldebene, wie auch der Cloudintegration gestellt. Schlüssel für effiziente Wertschöpfungsketten sind Kennzahlen, die in der Feldebene erhoben und auf der Unternehmensebene ausgewertet werden. Eine wesentliche Kennzahl, wenn nicht sogar „die" Kennzahl zur Ermittlung der Produktivität ist die OEE (Overall Equipment Effectiveness). Ein gezieltes OEE-Management ist damit Voraussetzung für die kontinuierliche Verbesserung von Produktionsprozesse. In diesem Beitrag wird ein Konzept und Prototyp vorgestellt, der eine ganzheitliche Erfassung der OEE in intelligenten Sensoren ermöglicht. Eine semantische Beschreibung ermöglicht eine Technologie-invariante Erhebung von OEE-Kennzahlen. Ein Proof-of-Concept für die Digitalisierung eines manuellen Handarbeitsplatz schließt den Beitrag ab.

Einleitung

Die Ideen rund um Industrie 4.0 und dem dazugehörigen Referenzarchitektur–modell RAMI 4.0 beschreiben ein umfassendes Framework für die ganzheitliche Digitalisierung des gesamten Produkt- und Fabrik-Lebenszykluses. Gerade die Themen Datensichtbarkeit und Transparenz der Prozesse bilden die Basis für eine Prozessanalyse und -optimierung. Das Extrahieren von KPIs spielt damit eine wichtige Rolle. Die Abbildung 1 zeigt die Entwicklungsstufen zu Industrie 4.0, so wie sie in der Definition zu RAMI 4.0 durch die arcatec beschrieben sind.

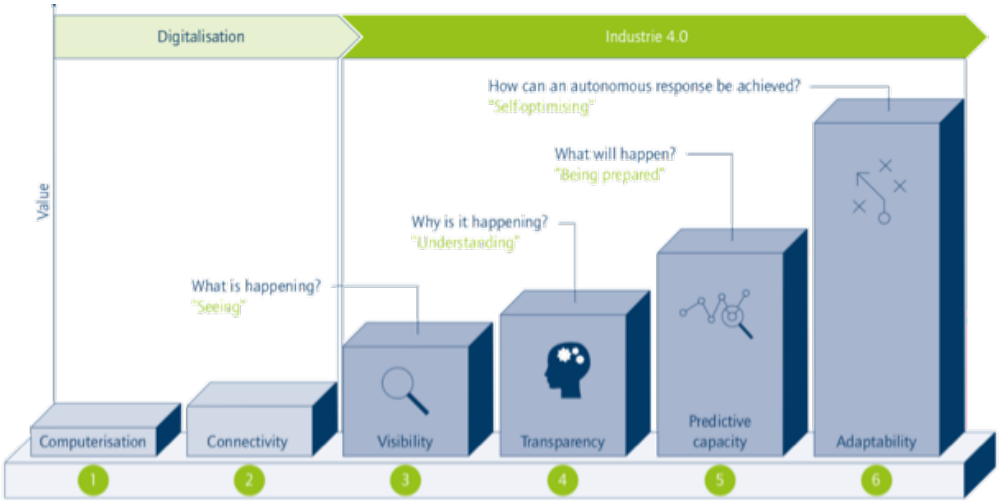

Abb.1: Schritte zu Industrie 4.0 (Quelle: Arcatec)

Computerisierung und Konnektivität sind nur die ersten Schritte zur Digitalisierung. Eine dominante Bedeutung hat die Extraktion sinnvoller Parameter zur Evaluierung und Bewertung von Prozessen und Prozessgüten. Nur auf der Basis dieser KPIs sind optimierende Schritte zur kontinuierlichen Prozessverbesserung und damit zum prädiktiven Handeln möglich.

Das Referenzarchitekturmodell lässt die Art der erhobenen KPIs offen. Diese werden in der Regel selektiv für die jeweiligen Prozesse extrahiert. Aus der Sicht der Betriebswirtschaft bietet die OEE (Operational Equipment Effectiveness) eine gute Basis zur Analyse der Maschinen- und Anlagenleistung. Die folgende Abbildung 2 zeigt die wesentlichen Abhängigkeiten von Prozessorganisation und Erhebung von Prozessparametern. Die Bewertungskriterien von Lean-Production und die dort ermittelten Kennzahlen geben in digitalisierter Form, verbunden mit einer historischen Auswertung, eine ausgezeichnete Basis für Visualisierung und Interpretation von Anlagen- und Fabrikperformance.

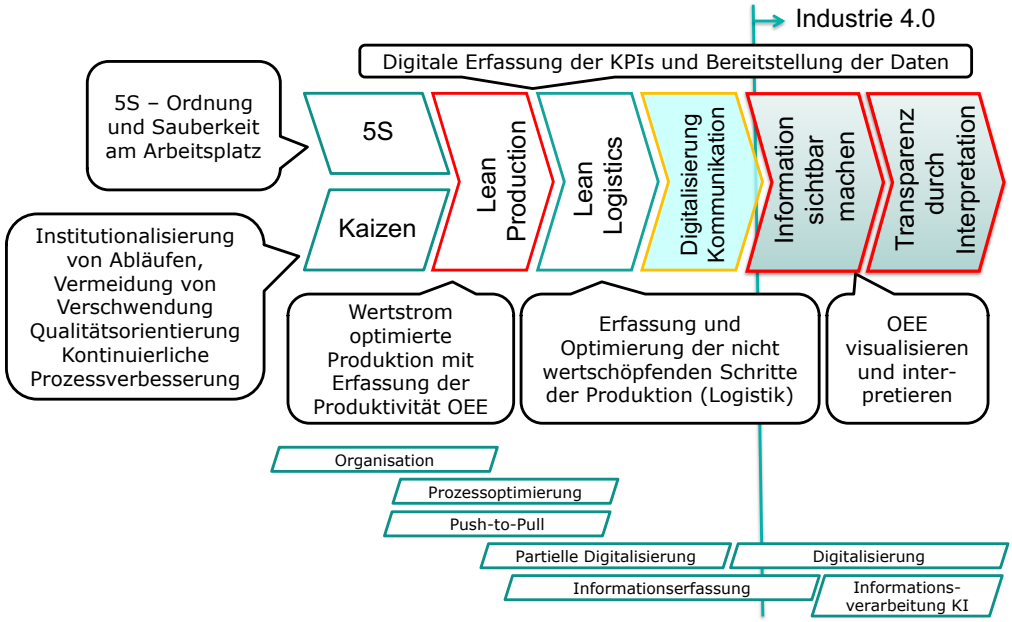

Abb.2: Vom Ordnungsprinzip zur Vollautomatisierung

OEE – Overall Equipment Effectiveness

Die OEE ermöglicht die quantitative Bewertung von Maschinen- und Anlagenleistung. Hierbei erfasst die OEE neben technischen Störungen alle Faktoren, welche die Gesamtperformance beeinflussen. Im Wesentlichen wird die OEE als relative Kennzahl beschrieben, die aus den Teilkennzahlen Verfügbarkeits–grad, Leistungsgrad und Qualitätsgrad besteht.

Der **Verfügbarkeitsgrad** beschreibt das Verhältnis der tatsächlichen Produktionszeit zur geplanten Produktionszeit. Der Verfügbarkeitsgrad ist maximal 1.

Der **Leistungsgrad** bewertet die tatsächlich erreichte Ausbeute an Teilen im Verhältnis zu der theoretisch möglichen Anzahl von Teilen innerhalb der tatsächlichen Produktionszeit.

Der **Qualitätsgrad** bewertet den Anteil der Gutteile im Verhältnis zu den Schlechtteilen.

Eine optimale Anlage bzw. Maschine hat damit eine maximale OEE von 1. Abbildung 3 zeigt neben der OEE (Level3) noch weitere Differenzierungen zur Bewertung der Verfügbarkeit und Performance. Im Besonderen eine Level 4 oder 5 Bewertung ermöglicht eine sehr detaillierte Ausgestaltung der Einflussmaßnahmen.

$$OEE = \frac{Actual\ Loading}{Planned\ Loading} \times \frac{Actual\ Performance}{Max\ Performance} \times \frac{Good\ parts}{All\ parts}$$

Level 2	Level 3 (OEE)	Level 4 (SEMI E10)	Level 5 (Eng)	
DN Downtime 2000	DU Unscheduled Downtime 2200	UDR Repair Time 2200	Machine Failure 2224	Loading losses
			Process Problem 2223	
			Utility Problem 2222	
			Operator Error 2221	
		DUW Wait Time 2210	Waiting for Tech 2214	
			Waiting for Spares 2213	
			Waiting for Operator 2212	
			Waiting for Maintenance 2211	
	SD Scheduled Downtime 2100	SDS Preventive Maintenance Time 2130	Change Lot/Setup 2132	
			EQ Cleaning 2131	
		SDT Productive Test Time 2120	Planned Offline 2122	
			Maintenance 2121	
		SDM Setup Time 2110	Waiting for Setup 2111	
UP Uptime 1000	SB Standby Time 1300	SB Standby Time 1310	No Material 1312	Performance losses
			No Operator 1311	
	EN Engineering Time 1200	EN Engineering Time 1210	Process Engineering 1211	
	PR Productive Time 1100	PR Productive Time 1100	Normal Production 1113	Quality losses
			Rework/ Retest 1112	
			QA Sampling 1111	

Abb. 3: Definition von OEE und SEMI E10

Heute wird die OEE in der Regel durch ERP Systeme ermittelt. Kleinere Unternehmen, mit einer geringen Digitalisierungstiefe, gehören häufig nicht zu den Anwendern der OEE. Die Ermittlung der einzelnen Faktoren dieser Kennzahl ist mit Aufwand und Kosten verbunden, die nicht direkt zu Kosteneinsparungen oder zusätzlichen Produktionskapazitäten führen. Ein Herabsetzen dieser Einstiegshürde, beispielsweise durch einen geeigneten OEE-Sensor, würde mehr Anwendern den Zugang zu einer qualifizierten Prozess- und Anlagenbewertung ermöglichen.

Intelligente Sensoren

In dem vorliegenden Projekt wurde eine generische OEE-Box entwickelt, die als intelligenter Sensor einen niederschwelligen Zugang zur Prozessanalyse ermöglicht. Die digitalen Signale kommen im einfachsten Fall von Tastern, die vom Werker betätigt werden. Mit einer Investition in komplexere Sensorik kann die Erfassung automatisiert werden, was auch im Nachhinein als Upgrade für die OEE-Box erfolgen kann. Beispielsweise kann ein induktiver Sensor die Anwesenheit eines Werkstücks oder eine Objektinspektionskamera die Qualität automatisch erfassen. Soweit kann man eine OEE-Box als komplexen Sensor verstehen.

Die Besonderheit ist eine einfache und durchgängige Integration in den Industrie 4.0 Framework. Heute ist es noch üblich, dass Sensoren an SPSen angebunden werden. Die Vorverarbeitung und Bereitstellung der Daten erfolgt in der Regel OPC-UA. Dieser Weg ist häufig umständlich und erfordert eine Programmierung auf den Steuerungssystemen. In der OEE-Box erfolgt eine Parametrierung des

Sensors auf der Basis einer semantischen Schnittstelle, so dass die Prozessinformation ohne explizite Programmierung vollständig bereitgestellt werden. Darüber hinaus ermöglicht die semantische Beschreibung der OEE-Funktionalität eine einheitliche Verwendung der Informationen der OEE-Box sowohl in der konventionellen OT-SPS-Welt als auch als isolierte Edgecloud-Anbindung. Abbildung 4 zeigt die prinzipielle Anbindung über IO-Link zur OT-Welt und MQTT als klassische Cloud-Anbindung.

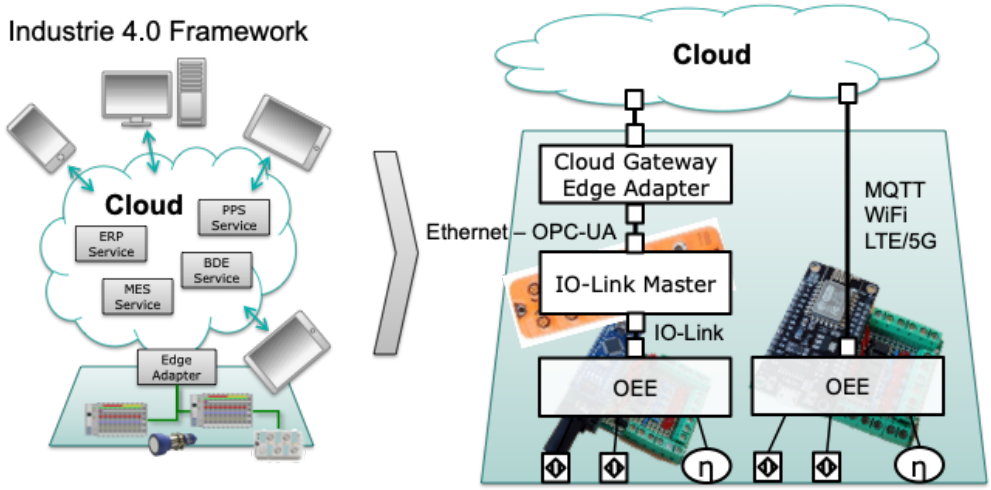

Abb. 4: Anbindung intelligenter Sensoren im I4.0 Kontext

IO-Link bietet mit der IODD (IO-Link Device Description) eine ausgezeichnete Möglichkeit komplexe Devices in einer einheitlichen Semantik zu beschreiben. Durch die inhärente Komplexität der bezogenen Daten der OEE, also individuelle Planzahlen und zeitliche Abhängigkeiten, ist eine Konfiguration der OEE-Box für den spezifischen Kontext notwendig. Ein vergleichbares Verhalten ist auch für die OOE-Box in der MQTT-Cloud-Variante notwendig. Hier wird die identische Semantik auf die JSON-Beschreibung der OEE-Box übertragen. Implementiert wurden die OEE-Boxen auf einen Arduino-Framework für IO-Link Sensoren als auch auf eine Arduino-Node-MCU. Der Basiscode für die OEE-Software und die generelle Semantik können so weitestgehend identisch gehalten werden.

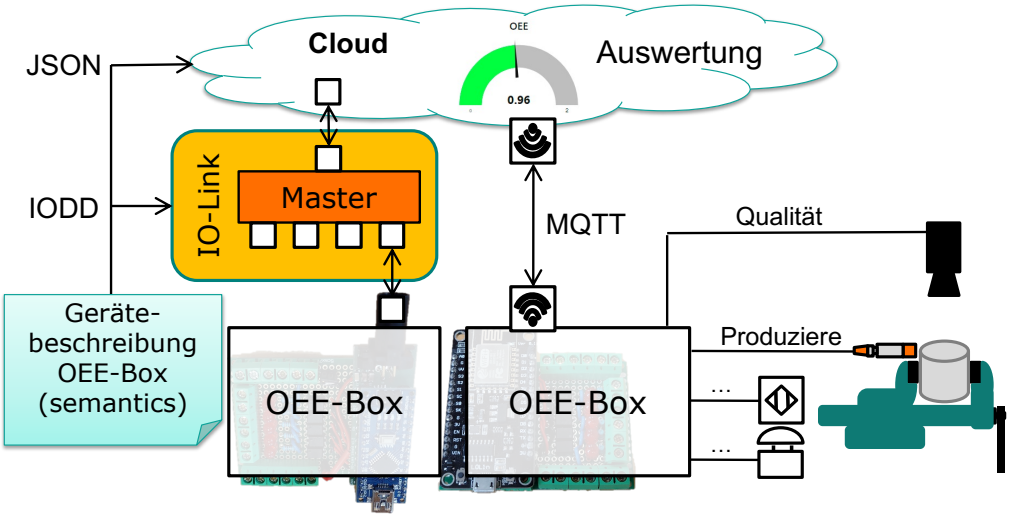

Abb. 5: Die OEE-Boxen verhalten sich identisch in den OT und IT Systemwelten

Proof of Concept

Zur Evaluierung der generischen OEE-Boxen wurden unterschiedliche Fertigungsschritte bei der Montage von Halbzeugen der Firma Item untersucht. In der folgenden Abbildung sind Montagehilfsmittel zur Handgrifffertigung abgebildet. Diese wurden mit geeigneten Sensoren versehen, so dass der gesamte Montageprozess automatisiert erfasst werden konnte.

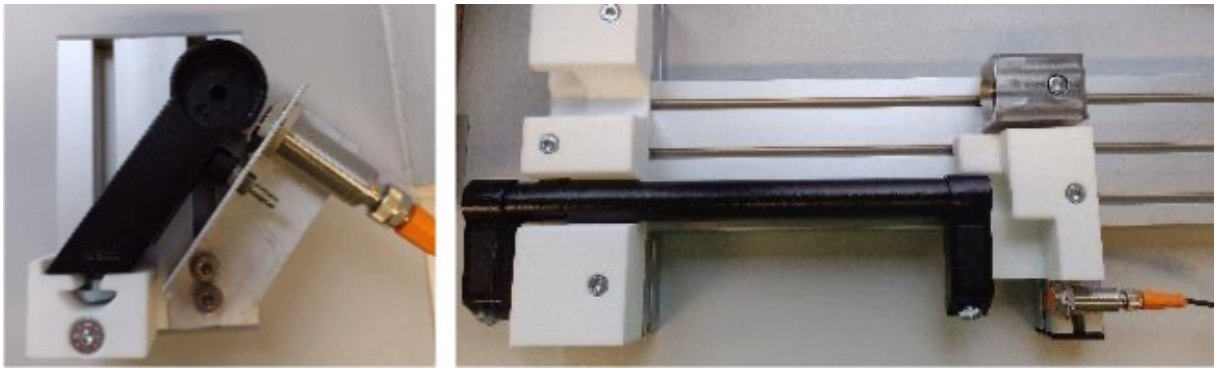

Abb. 6: Montageeinrichtung für die manuelle Handgriffmontage

Es hat sich gezeigt, dass die manuellen Arbeitsschritte mit verhältnismäßig geringem Aufwand automatisiert erfasst werden können. Die Arbeitsbeschreibung reichte in der Regel aus, um prinzipielle Indikatoren zu identifizieren wann ein Arbeitsschritt gestartet oder beendet wurde. Die Umsetzung in beiden Technologievarianten IT (MQTT) und OT (IO-Link und OPC-UA) brachten gleiche Ergebnisse. Der Aufwand ist für jeweils beide Welten identisch gering. Die semantischen Beschreibungen halfen bei der Anbindung der OEE-Boxen.

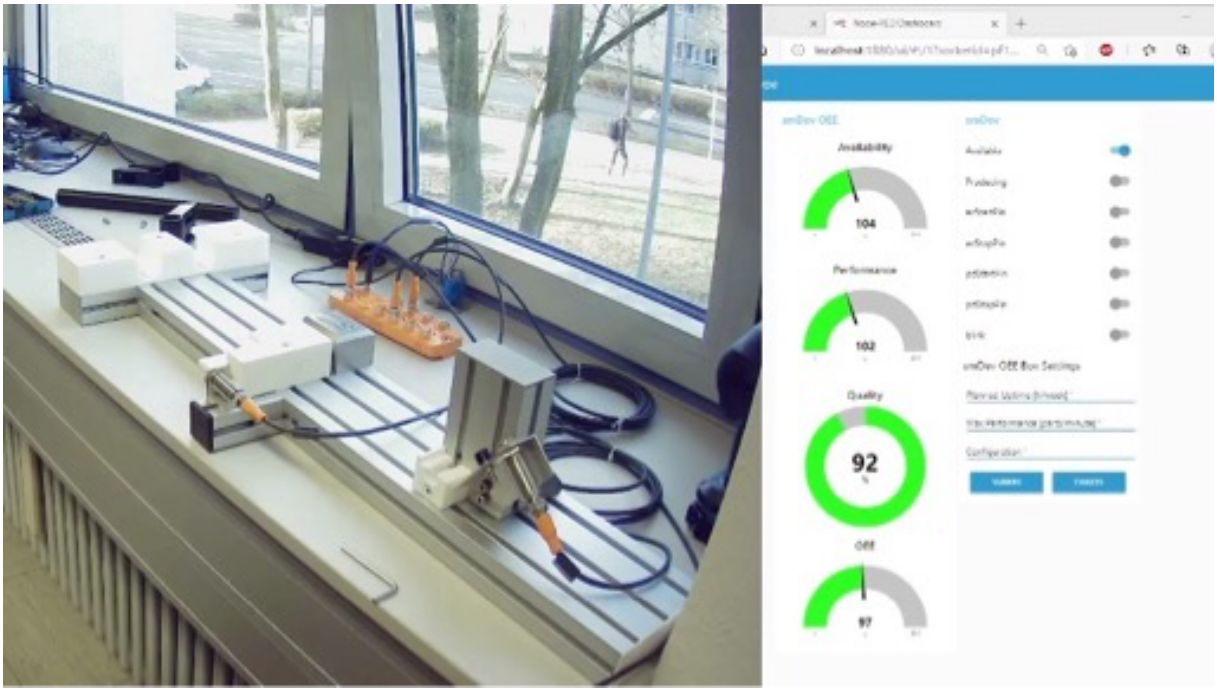

Abb. 6: Manuelle Handgrifffertigung mit OEE-Datenerfassung

Im Netzwerk angeschlossene OEE-Boxen werden automatisch erfasst und können über ein Web-Interface konfiguriert werden. Eine Multiplikation der OEE-Faktoren

mehrerer OEE-Boxen ermöglicht die Bewertung des OEE-Faktors einer Gruppe von Montagesystemen.

Abb. 8: Paralleler Betrieb mehrerer OEE-Boxen

Die OEE-Boxen liefern neben den berechneten OEE-Faktoren zusätzlich den aktuellen Zustand der überwachten Montage sowie Stückzahlen von Gut- und Schlecht-Teilen. Diese Daten wurden zusammen mit den IDs der eingesetzten OEE-Boxen in einer Datenbank gespeichert. Dies ermöglicht die Berechnung der OEE Faktoren über einen beliebigen Zeitraum. So kann die OEE-Kennzahl von verschiedenen, frei wählbaren Zeitabschnitten verglichen werden. Dies ermöglicht eine objektive Bewertung von Änderungen.

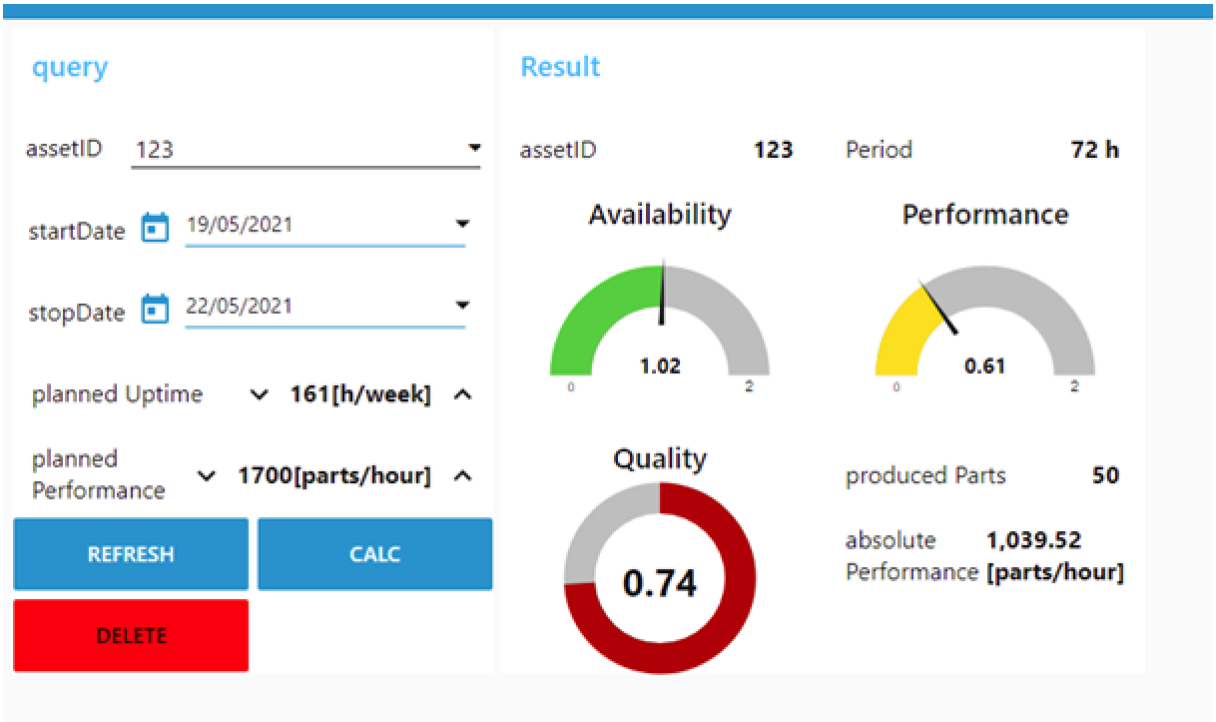

Abb. 7: Auswertung über beliebigen Zeitraum aus Datenbank

Zusammenfassung und Ausblick

Industrie 4.0 lebt von einer intelligenten Erfassung von Prozessparametern zur Optimierung von Prozessen und Abläufen. Die OEE bietet eine ausgezeichnete Möglichkeit Verfügbarkeit und Performance von Anlagen und Maschinen zu bewerten. In dem vorliegenden Projekt konnte eine Systematik für OEE-Boxen entwickelt werden, die auf der Basis einer einheitlichen Prozesssemantik realisiert wurde. Hierbei wurde der Weg IO-Link für die klassische OT-Welt und MQTT für die IT-Welt implementiert. Der Core wurde auf einem Arduino-Framework aufgebaut, so dass sowohl die Systemlogik als auch die Semantik konsistent blieben. Die praktische Umsetzung in der Vorfertigung von Modulen hat gezeigt, dass die gewählte Systematik eine vergleichsweise einfache und konsistente Möglichkeit zur Erfassung von OEE-Paramatern ermöglicht. Der Ansatz ermöglichte das Ableiten eines Produktes für intelligente OEE-Boxen.

---

[1] V. Chavez and J. Wollert, "Arduino based Framework for Rapid Application Development of a Generic IO-Link interface," in Kommunikation in der Automation, 2018.

http://dx.doi.org/10.25673/39573

Funkkommunikation in der Wasserwirtschaft – Anforderungsprofile und Bewertung der Eignung von Low Power Wide Area Networks

Lisa Underberg, Jens Alex, Lutz Rauchhaupt
Institut für Automation und Kommunikation e.V. Magdeburg
IKT und Automation

Werner-Heisenberg-Str. 1
39106 Magdeburg
lisa.underberg@ifak.eu
jens.alex@ifak.eu
lutz.rauchhaupt@ifak.eu

Abstract: Die Branche der Wasserwirtschaft ist traditionell mit für unsere Gesellschaft infrastrukturkritischen Aufgaben betraut. Entsprechend wird bei der Einführung neuer Technologien besondere Vorsicht walten gelassen. Gleichzeitig steigt der Optimierungsdruck auf Akteurinnen und Akteure der Wasserwirtschaft, beispielsweise durch steigende Bevölkerungsdichte bei gleichbleibendem Wassernetz. Vor diesem Hintergrund gewinnen Funkkommunikationssysteme an Relevanz, denn sie sind gut nachrüstbar und je nach Anforderung für einen Batteriebetrieb geeignet. In diesem Beitrag werden Anforderungsprofile an die Funkkommunikation von Anwendungen der Wasserwirtschaft vorgestellt. Die Profile wurden aus Umfragen im Rahmen von deutschlandweiten Workshops mit Akteurinnen und Akteuren der Wasserwirtschaft durchgeführt. Es wurden qualitative und quantitative Anforderungen erfasst. Es wurden zwei Profile extrahiert, die Anwendungen mit ähnlichen Anforderungen an die Funkkommunikation zusammenfassen: Anwendungen der Zustandsüberwachung und Regelungsanwendungen. Die Eignung von Low Power Wide Area Network (LPWAN)-Technologien wird sowohl für die Zustandsüberwachung als auch für Regelungsanwendungen diskutiert. Dabei werden Aspekte wie das Betreibermodell und das genutzte Frequenzspektrum berücksichtigt.

1 Einleitung

Die Branche der Wasserwirtschaft befasst sich mit Themen rund um Trinkwasserver- und Abwasserentsorgung. Traditionell ist diese Branche daher in unserer Gesellschaft mit infrastrukturkritischen Aufgaben betraut und verhält sich entsprechend konservativ, auch wenn es um die Nutzung von Funksystemen für ihre Anwendungen geht. Gleichzeitig steigt der Druck zum Beispiel durch steigende Bevölkerungsdichten bei gleichbleibenden Wassernetzen auf Ver- und Entsorger sowie Verbände Abläufe in ihren Bauwerken und Wassernetzen zu optimieren. Vor diesem Hintergrund steigt das Interesse an Funksystemen aufgrund ihrer einfachen Nachrüstbarkeit aktuell an.
Das ifak richtete Anfang 2021 Workshops für Akteurinnen und Akteure der Branche aus ganz Deutschland aus, während derer zum einen über Funksysteme informiert und zum anderen eine Anforderungserfassung qualitativer sowie quantitativer Aspekte entsprechend [1] durchgeführt wurde. Auf Basis dieser Anforderungserfassung wurden zwei Anforderungsprofile für die Funkkommunikation abgeleitet.

Aus den Anforderungsprofilen wird deutlich, dass Anwendungen der Wasserwirtschaft im Vergleich zu anderen industriellen Anwendungen große zu überbrückende Entfernungen haben, während ihr Sendezeitabstand gleichzeitig vergleichsweise groß ist. Diese Anforderungen passen grob zu Leistungsprofilen von Low Power Wide Area Network (LPWAN)-Technologien. Im Detail ist die Betrachtung einzelner Technologien notwendig, um deren Eignung für die beiden Anforderungsprofile zu bewerten.
In diesem Beitrag werden die Gemeinsamkeiten und Unterschiede von LPWANs hinsichtlich ihrer Auswirkung auf die Eignung für wasserwirtschaftliche Anwendungen beleuchtet. Ein Aspekt ist die Nutzung von lizenziertem Spektrum (z.B. NB-IoT, LTE-M) oder unlizenziertem Spektrum (z.B. LoRaWAN, Mioty, Sigfox, Weightless). Bei ersterem ist der Anwender wirtschaftlich und technologisch von einem Netzbetreiber abhängig, während im unlizenzierten Spektrum zumeist die Wahl zwischen einem Netzbetreiber und dem Aufbau eines eigenen Netzes besteht. Dahingegen gilt im unlizenzierten Spektrum eine strenge Begrenzung der Sendeleistung und -dauer, insbesondere, wenn kein Listen-before-talk (LBT) verwendet wird. Da letzteres für LPWAN-Technologien einen unerwünscht erhöhten Energiebedarf bedeutet, wird in der Regel auf LBT verzichtet. In diesem Spannungsfeld wird dieser Beitrag die Eignung von LPWANs für infrastrukturkritische Aufgaben in der Wasserwirtschaft einordnen.
Der Beitrag ist wie folgt aufbaut. Kapitel 2 beschreibt die aus der Umfrage abgeleiteten Anforderungsprofile sowohl qualitativ als auch quantitativ. Eine allgemeine Übersicht über LPWAN-Technologien wird in Kapitel 3, worauf aufbauend in Kapitel 4 die Eignung einzelner Technologien für wasserwirtschaftliche Anwendungen diskutiert wird. Kapitel 5 fasst zusammen und gibt einen Ausblick.

2 Anforderungsprofile der Wasserwirtschaft

Bereits seit 2007 werden Anforderungsprofile industrieller Anwendungen an Kommunikationssysteme diskutiert, was durch die Erstveröffentlichung von VDI/VDE Richtlinie 2185 Blatt 1 und [2] markiert wird. Seitdem gab es zahlreiche Versuche, Anforderungen industrieller Anwendungen strukturiert zu beschreiben und zu konsolidieren. Einen Überblick zu diesem Prozess gibt [3]. Zuletzt flossen diese Bemühungen in die Überarbeitung der VDI/VDE Richtlinie 2185 Blatt 1 [1], Spezifikationen und Reporte der 3GPP [4, 5, 6] und White Paper der 5G-ACIA [7, 8] ein. Anforderungen der Wasserwirtschaft als spezielle Untergruppe industrieller Anwendungen wurden bisher allerdings nicht erfasst, obwohl diese von den verfügbaren Anforderungsprofilen nicht abgedeckt werden.
Aus den Umfrageergebnissen wurden zwei Anforderungsprofile abgeleitet. Anforderungsprofil A umfasst Anwendungen, in denen das Funksystem zur Zustandsüberwachung eingesetzt werden soll. Hier geht es um das regelmäßige und zuverlässige Erfassen und Übertragen von Messwerten beispielsweise zur Gewässerüberwachung. Im Gegensatz zur reinen Zustandsüberwachung soll in Anwendungen, welche in Profil B erfasst werden, steuernd in einen Prozess eingegriffen werden. Ein Beispiel ist die Regelung eines Mischwasserkanalnetzes.
Die quantitativen Parameter der Anforderungsprofile sind in Tabelle 2 zusammengefasst und in Abb. 1 graphisch dargestellt. Je Anwendungsgruppe ist je ein Anforderungsprofil für den zu erwartenden Maximalwert sowie für den Median angegeben.
Anwendungen der Zustandsüberwachung haben seltener einen Übertragungsbedarf, doch deren Datenpakete sind größer als die der steuernd eingreifenden Anwendungen. Die Anzahl logischer Links und die

Tabelle 1: Quantitative Parameter der Anforderungsprofile der Wasserwirtschaft

	Anforderungsprofil A		Anforderungsprofil B	
	Median	Extremwert	Median	Extremwert
Übertragungszeit	1 h	1 min	1 min	< 1 min
Sendezeitabstand	1 h	15 min	30 s	1 s
Anzahl logischer Links	30	100	10	30
Nutzdatenlänge	500 kB	100 MB	1 kB	10 kB
Distanz	1 km	>30 km	1 km	10 km

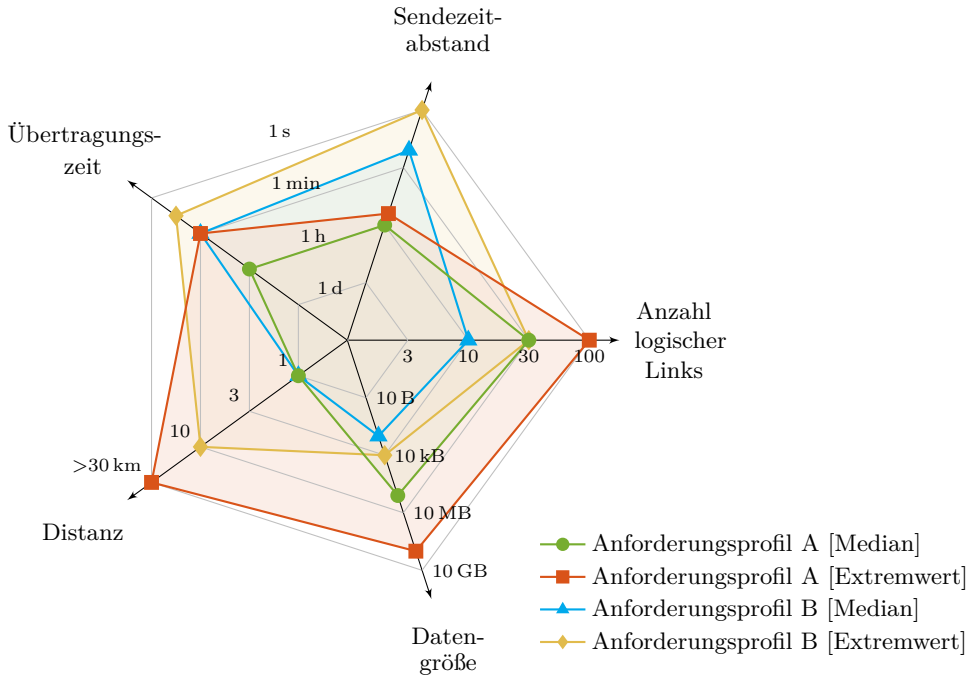

Abbildung 1: Anforderungsprofile A und B der Wasserwirtschaft extrahiert aus Umfrageergebnissen.

zu überbrückende Distanz sind bei der Zustandsüberwachung ebenso größer, während sie längere Übertragungszeiten tolerieren. Insgesamt fällt auf, dass für beide Anwendungsgruppen bereits angenommen wird, das lokal Daten mehrerer Sensoren und Aktoren aggregiert werden, woraus die Nutzdatenlängen zwischen 1 kB und 100 MB resultieren. Die Daten sollen lediglich per Funk an einer zentral erreichbaren Stelle zusammengeführt werden.
Hinsichtlich der qualitativen Anforderungen gleichen sich beide Profile. Ausfallsicherheit beispielsweise durch Redundanz, besonders im Falle von Notlagen wie Großwetterereignissen oder Hochwasser sowie Verschlüsselung und Authentifizierung nach dem Stand der Technik sind von Bedeutung. Sind Daten verschlüsselt, sind Speichern und Übertragen in nicht firmeneigener Infrastruktur unkritisch. Entsprechend der recht unveränderlichen Rahmenbedingungen wie einem begrenzten Stadtgebiet oder einer begrenzten Anzahl an Gewässern ist die Skalierbarkeit der Anwendungen gering bis moderat. Die Anforderungen an die Nutzerfreundlichkeit sind hoch, denn die Daten müssen auch für nicht-Expertinnen und -Experten einfach zugänglich und übersichtlich aufbereitet werden. Im Idealfall ist sogar die Installation und Inbetriebnahme ohne IT- oder Funk-Fachkenntnis möglich. Das Kommunikationssystem sollte zudem wartungsarm sein, also beispielsweise eine lange Batterielaufzeit aufweisen. Insbesondere aufgrund des mehrere Dekaden langen Lebenszyklus' von wasserwirtschaftlichen Anlagen ist eine langfristige Investitionssicherheit und kommerzielle Verfügbarkeit nötig.

3 Übersicht über LPWAN-Technologien

Low Power Wide Area Networks (LPWANs) gewinnen in der Branche der Wasserwirtschaft zunehmend Aufmerksamkeit, da ihre zentralen und namensgebenden Eigenschaften der Überbrückung großer Distanzen (Wide Area) bei gleichzeitig geringem Energiebedarf (Low Power) Kernanforderungen ihrer Anwendungen abdecken.
Neben diesen Eigenschaften weisen LPWAN-Technologien eine ähnlichen grundlegenden Aufbau auf, welcher verallgemeinert in Abb. 2 dargestellt ist. Endgeräte, welche in der Regel Sensoren und Aktoren beinhalten oder mit ihnen verbunden sind, kommunizieren mit zentralen Einheiten des Funksystems, welche beispielsweise technologiespezifisch als Basisstationen oder Gateway bezeichnet werden. Der Anteil des LPWAN-Funknetzes ermöglicht grundsätzlich bidirektionalen Datenaustausch, ist jedoch vornehmlich auf Übertragungen von Endgerät zu zentralem Netz, dem Uplink (UL) ausgelegt. Übertragungen in die

Gegenrichtung, im Downlink (DL), sind oft nachgeordneter Priorität. Beispielsweise können Downlink-Übertragungen häufig nur auf Uplink-Übertragungen folgen, da die Funkgeräte ansonsten zugunsten eines geringen Energiebedarfs nicht erreichbar sind („Sleep mode“). Über Funkstrecken, die andere Technologien nutzen, oder über kabelgebundene Netze werden die Daten an einem zentralen Ort, zum Beispiel einer Cloud oder einem Server, zusammengeführt. Auf die zusammengeführten Daten wird durch ein Auswertungswerkzeug zugegriffen. Dieses Werkzeug kann individuell gestaltet werden oder bereits durch die Technologie vorbereitet sein. LPWAN-Technologien unterscheiden sich darin, welcher Anteil des Netzes in Eigenverantwortung betrieben und welcher Anteil von einem Netzbetreiber übernommen werden kann. Einige Technologien erlauben, dass der Endanwender entscheidet, an welcher Stelle der Übergang zwischen eigenverantwortlich betriebener und von Dritten betriebener Infrastruktur liegt.

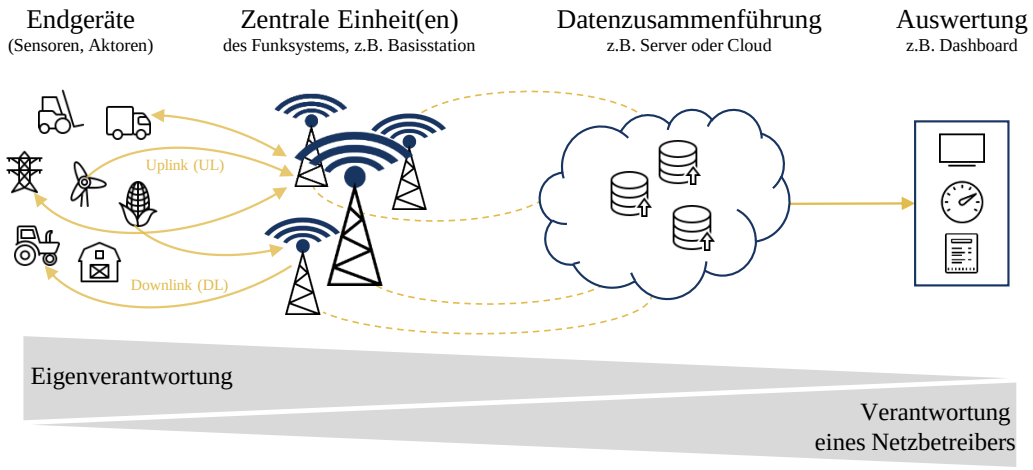

Abbildung 2: Komponenten einer LPWAN-Technologie in verallgemeinerter Darstellung.

Typischerweise nutzen LPWANs das Sub-1 GHz-Frequenzband, da dessen große Wellenlängen eine hohe Reichweite und eine gute Gebäudedurchdringung begünstigen. Je nach Technologie wird ein lizenziertes oder lizenzfreies Frequenzband verwendet. Für die Technologien, die einen lizenzfreien Frequenzbereich nutzen, gilt eine Regulierung nicht nur für die maximale äquivalente, isotrope Strahlungsleistung (engl. „equivalent isotropically radiated power“ (EIRP)), sondern auch für den Kanalzugriff. Falls keine Verfahren wie „Listen before Talk“ oder „Adaptive Frequency Agility“ verwendet werden, gilt eine Beschränkung der maximalen Kanalbelegungsdauer, welche über den „Duty Cycle“ angegeben wird. Tabelle 3 fasst die Regulierung entsprechend EN 300 220-2, Tabelle B.1 [9] zusammen.

Tabelle 2: Sendeleistungs- und Sendezeitbegrenzungen für Nutzung lizenzfreien Spektrums [9]

Band	Frequenzbereich	Sendeleistung (EIRP)	Duty Cycle
K	863 MHz bis 865 MHz	25 mW	$\leq 0{,}1\,\%$
L	865 MHz bis 868 MHz	25 mW	$\leq 1\,\%$
M	868,0 MHz bis 865,6 MHz	25 mW	$\leq 1\,\%$
N	868,7 MHz bis 869,2 MHz	25 mW	$\leq 0{,}1\,\%$
P	869,4 MHz bis 869,65 MHz	500 mW	$\leq 10\,\%$
P	869,7 MHz bis 870 MHz	5 mW	Keine
Q	869,7 MHz bis 870 MHz	25 mW	$\leq 1\,\%$

Für LPWANs kann in der Regel angenommen werden, dass der Duty Cycle angewendet wird, da komplexere Kanalzugriffsverfahren den Energiebedarf erhöhen und damit die Batterielaufzeit verringern.

Entsprechend darf die Sendedauer maximal 10 % betragen. Eine Ausnahme ist Band P, das allerdings durch die kleine Sendeleistung von maximal 5 mW und die damit signifikant kleinere Reichweite für die meisten Anwendungen von LPWANs uninteressant ist.
Eigenschaften verschiedener, aktuell kommerziell verfügbaren LPWAN-Technologien sind in Tabelle 4 zusammengefasst. Während NB-IoT und LTE-M ausschließlich von Netzbetreibern in lizenzierten Frequenzbändern genutzt werden, verwenden LoRaWAN [10], mioty [11], Sigfox [12] und Weightless [13] das lizenzfreie ISM-Band bei 868 MHz. Für Technologien im lizenzfreien Spektrum sind unterschiedliche Betreibermodelle möglich.
Eine technologieinhärente Verschlüsselung bieten alle Technologien außer Sigfox. Bei LoRaWAN und mioty wird AES 128 verwendet. Sigfox verschlüsselt zwar nicht innerhalb der Verarbeitung die Daten, doch es ist möglich, applikationsseitig eine Verschlüsselung zu implementieren. Eine Authentifizierung bieten alle Technologien. In mioty wird ein Cypher-based Message Authentication Code (CMAC) verwendet.
Hinsichtlich der maximal erreichten Datenrate unterscheidet sich LTE-M als Mid-Rate LPWAN-Technologie von den anderen Technologien, welche den Low-Rate-LPWAN-Technologien zugeordnet werden. Die Nutzdatenlänge hängt bei NB-IoT und LTE-M vom Netzbetreiber und dem jeweils geschlossenen Vertrag ab, während sie bei LoRaWAN, mioty und Weightless auf etwa 250 B begrenzt ist. Sigfox beschränkt die Nutzdatenlänge mit maximal 12 B im Uplink besonders stark. Pro Stunde dürfen bei Sigfox außerdem maximal 72 B übertragen werden.
Wegen ihrer Robustheit werden vornehmlich eine Frequenzmodulation oder eine Phasenmodulation kleiner Ordnung verwendet (Gaussian Minimum Shift Keying (GMSK), Gaussian Frequency Shift Keying (GFSK), Quadratur Phase Shift Keying (QPSK), Offset-QPSK (O-QPSK)). Bei mioty wird eine GMSK mit dem in ETSI TS 103 357 [14] beschriebenen Telegram Splitting Ultra Narrow Band (TS-UNB) kombiniert. Einzig LoRaWAN greift auf ein Spreizverfahren (Chirp Spread Spectrum (CSS)) zurück. Wie stark ein Signal gespreizt wird, wird durch den Spreizfaktor (SF) angegeben, welcher bei LoRaWAN zwischen 7 und 12 liegt. Bei allen Technologien ist die Kanalbandbreite entsprechend des Sub-1-GHz-Bandes eher klein.
Die je Technologie erreichbaren Reichweiten werden unterschiedlich angegeben [15, 16], und in der Regel wird zwischen urbaner und ruraler Umgebung unterschieden. Die Reichweite kann entweder über die verwendete Sendeleistung angepasst werden, welche einen Einfluss auf die erreichte Batterielaufzeit hat. Ebenso kann bei LoRaWAN der SF angepasst werden. Ein hoher SF bedeutet eine längere Sendezeit und damit einen höheren Energiebedarf als ein kleiner SF.

Tabelle 3: Eigenschaften von LPWAN-Technologien [16, 17, 18, 19, 20, 15, 10, 11, 12, 13]

	NB-IoT	LTE-M	LoRaWAN	mioty	Sigfox	Weightless
Frequenzband	lizenziert (LTE-Bänder)		Lizenzfrei (Sub-1 GHz-ISM-Band)			
Organisation	3GPP	3GPP	LoRa Alliance	mioty Alliance	Sigfox	Weightless Alliance
Verantwortlichkeit für Infrastrukturgeräte	Netzbetreiber	Netzbetreiber	Netzbetreiber oder in Eigenverantwortung	In Eigenverantwortung	Netzbetreiber oder in Eigenverantwortung	In Eigenverantwortung
Verschlüsselung	LTE-inhärent	LTE-inhärent	AES 128	AES 128	Ohne	Vorhanden
Authentifizierung	LTE-inhärent	LTE-inhärent	Vorhanden	CMAC	Vorhanden	Vorhanden
Datenrate	127 kbit/s DL, 159 kbit/s UL	4 Mbit/s DL, 7 Mbit/s UL	0,3 kbit/s bis 50 kbit/s	0,407 kbit/s	6 Nachrichten je 12 B pro Stunde	0,625 kbit/s bis 100 kbit/s
Länge der Nutzdaten	Abhängig vom Netzbetreiber	Abhängig vom Netzbetreiber	max. 64, 128 oder 235 B abh. vom Spreizfaktor	10 B bis 250 B	0 B bis 12 B UL, 8 B DL	max. 255 B
Modulation	GMSK, QPSK	GMSK	CSS	TS-UNB mit (G)MSK	GFSK (DL), DBPSK (UL)	GMSK, OQPSK
Kanalbrandbreite	180 kHz	5 MHz	125 kHz, 250 kHz oder 500 kHz	100 kHz bis 1500 kHz	100 Hz	unidirektional ggf. 100 Hz, sonst 12,5 kHz
Reichweite	Urban: 1 km, rural: 10 km	<15 km	Urban: 5 km, rural: 20 km	Urban: 5 km, rural: 15 km	Urban: 10 km, rural: 40 km	Urban: 3 km, rural: 5 km

4 Eignung von LPWANs für die Wasserwirtschaft

4.1 Erfüllung quantitativer Anforderungen

Inwiefern die quantitativen Anforderungen der in Kapitel 2 abgeleiteten Profile für die beiden Anwendungsgruppen der Wasserwirtschaft erfüllt werden können, kann je Technologie unterschiedlich gut bewertet werden.
Für NB-IoT und LTE-M hängt die Eignung erheblich von der Netzabdeckung durch Netzbetreiber und deren verfügbaren Vertragsmodellen ab. Entsprechend ist für diese beiden Technologien eine individuelle Überprüfung je konkreter Anwendung notwendig.
Die von Sigfox angebotene Datengröße und der angebotene Sendezeitabstand liegen Größenordnungen von den Anforderungen der Anwendungen entfernt. Sigfox erscheint daher für die betrachteten wasserwirtschaftlichen Anwendungen ungeeignet.
Bei mioty, Weightless und LoRaWAN ist eine genauere Überprüfung nötig. Diese drei Technologien arbeiten im lizenzfreien Band und unterliegen dadurch der Regulierung durch den Duty Cycle. Zur Untersuchung der Implikationen durch den Duty Cycle wird beispielhaft im Folgenden LoRaWAN herausgegriffen, da hier durch offene Spezifikation genaue Informationen zum Übertragungsverhalten verfügbar sind. Es werden Übertragungen im Uplink mit einer Kanalbandbreite von 125 kHz betrachtet.

4.1.1 Implikationen der Sendezeitbegrenzung im lizenzfreien Band

Dadurch, dass in der Regel keine Strategien eines komplexeren Kanalzugriffs verwendet werden, sondern die Endgeräte ihren Ruhemodus nur für eine Übertragung möglichst kurz unterbrechen, kann der Kanalzugriff von LoRaWAN analog zu „Pure Aloha“ [21] modelliert werden. Es wird angenommen, dass die Übertragungen voneinander unabhängig und unsynchronisiert sind sowie zufällig innerhalb einer Stunde auftreten. Somit kann die Generierung einer Übertragung mit einer Poissonverteilung modelliert werden. Entsprechend Tabelle 3 darf ein Endgerät bis zu 3,6 s, 36 s oder 360 s pro Stunde einen Übertragungskanal belegen. Für eine Betrachtung der Obergrenze des Durchsatzes wird angenommen, dass jedes Endgerät die erlaubte Sendezeit vollständig ausnutzt. In diesem Verständnis wird eine Zeitstunde in 1000, 100 oder 10 Übertragungszeiteinheiten eingeteilt. Während der laufenden Übertragung ist es möglich, dass ein weiteres Gerät ebenfalls eine Übertragung generiert und es so zu einer Kollision kommt. In diesem Fall wird angenommen, dass beide Datenpakete verloren gehen.
Basierend auf diesen Annahmen und den im Mittel pro Übertragungszeiteinheit generierten Datenpaketen G kann der Durchsatz S, welcher als Verhältnis erfolgreicher Übertragungen zu insgesamt zur Verfügung stehenden Übertragungszeiteinheiten definiert ist, entsprechend Gleichung (1) berechnet werden.

$$S = G \cdot e^{-2G} \tag{1}$$

Abbildung 3 zeigt den Durchsatz S über der Anzahl von physikalischen Links, die jeweils einen logischen Link abbilden und den Übertragungskanal maximal lange entsprechend des Duty Cycles belegen.

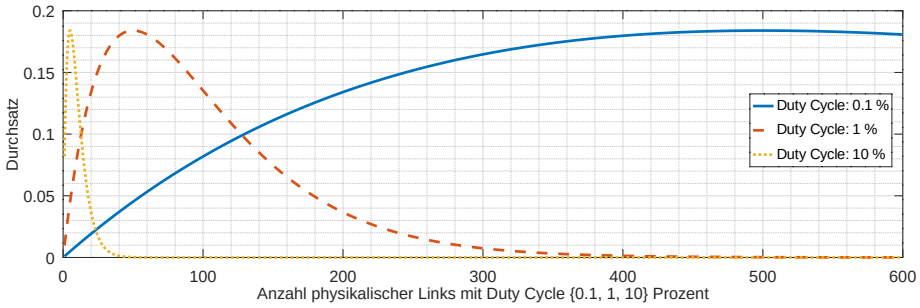

Abbildung 3: Durchsatz über Anzahl von physikalischen Links für drei Duty Cycles.

Der maximale Durchsatz von etwa 18 % wird für 5, 50 oder 500 physikalische Links erreicht. Für weniger physikalische Links ist der angebotene Verkehr zu klein, um den maximalen Durchsatz zu erreichen. Bei

mehr physikalische Links enstehend zunehmend Kollisionen, wodurch der Durchsatz sinkt. Dieser Zusammenhang wird ebenfalls in Abb. 4 deutlich, welche die Kollisionswahrscheinlichkeit zweier Übertragungen zeigt, deren Länge den Duty Cycle jeweils ausnutzt.

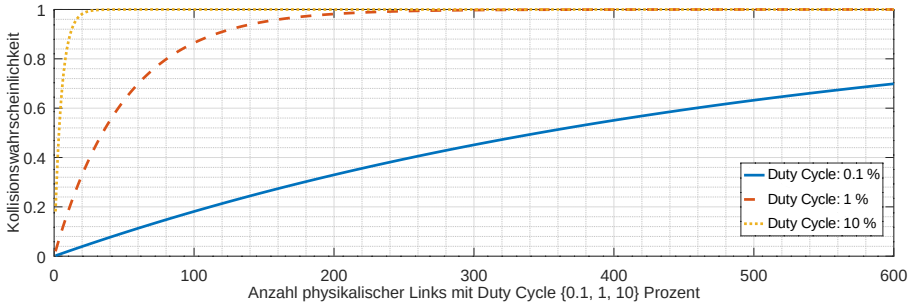

Abbildung 4: Kollisionswahrscheinlichkeit über Anzahl von physikalischen Links für drei Duty Cycles.

Für einen Durchsatz von $S = 0.18$ beträgt die Kollisionswahrscheinlichkeit 63,2 %. Im Umkehrschluss bedeutet dies, dass der Applikation aufgrund eines Paketverlustes nur mit einer Wahrscheinlichkeit von 36,8 % aktuelle Daten zur Verfügung stehen. Mit einer Wahrscheinlichkeit von $0.368^3 = 25.2$ sind die Daten älter als drei Stunden.

4.1.2 Nutzdatenlänge und Sendezeitabstand von LoRaWAN mit 1 % Duty Cycle

Um unter Berücksichtigung des Duty Cycles zu ermitteln, welche Kombination von Nutzdatenlänge und Sendezeitabstand mit LoRaWAN möglich sind, wird die Übertragungsdauer T_{oA} benötigt. Sie wird im Folgenden in Gleichung (2) entsprechend [22] berechnet. Als Präambellänge werden 8 Symbole angenommen. Ein Header und Kanalcodierung mit großer Coderate werden verwendet.

$$T_{\mathrm{oA}} = T_{\mathrm{s}} \cdot \quad 20.25 + \max\left(\left\lceil \frac{8 \cdot l_{\mathrm{usr}} - 4 \cdot \mathrm{SF} + 44}{4\,(\mathrm{SF} - 2 \cdot k)} \right\rceil \cdot 5, 0\right)\right) \tag{2}$$

mit

$$k = \begin{cases} 0 & \mathrm{SF} \in \{7, 8, 9, 10\} \\ 1 & \mathrm{SF} \in \{11, 12\} \end{cases} \tag{3}$$

und

$$T_{\mathrm{s}} = \frac{2^{\mathrm{SF}}}{125\,\mathrm{kHz}}. \tag{4}$$

Die mit einem Duty Cycle von 1 % maximal mögliche Nutzdatenlänge pro Stunde ist für die unterschiedlichen SFs von LoRaWAN in Tabelle 5 dargestellt. Beispielsweise ist für SF = 7 die maximale Nutzdatenlänge auf $l_{\mathrm{usr}} = 222\,\mathrm{B}$ festgelegt, deren Übertragung $T_{\mathrm{oA}} = 369\,\mathrm{ms}$ andauert. Bei dieser Übertragungsdauer können unter Einhaltung des Duty Cycles von 1 % pro Stunde 97 Pakete verschickt werden. Die insgesamt innerhalb einer Stunde übertragbaren Datengröße liegt somit bei $222\,\mathrm{B} \cdot 97 = 21\,534\,\mathrm{B}$.

Tabelle 4: Nutzdatenlänge bei LoRaWAN je physikalischem Link pro Stunde für unterschiedliche SFs

SF	Max. Nutzdatenlänge	Sendedauer	Duty Cycle 1 %
7	222 B	0,369 s	97 Pakete, 21 534 B
8	222 B	0,646 s	55 Pakete, 12 210 B
9	95 B	0,574 s	62 Pakete, 5890 B
10	31 B	0,534 s	67 Pakete, 2077 B
11	31 B	1,151 s	31 Pakete, 961 B
12	31 B	2,138 s	16 Pakete, 496 B

Tabelle 5: Geforderte mittlere Datenrate sowie mittlere von LoRaWAN bereitgestellte Datenrate

Anforderungsprofil	R_m	LoRaWAN SF	R_m $n_\mathrm{PL} = 1$	R_m für $n_\mathrm{PL} = 50$
Profil A Median	33,33 kbit/s	7	47,85 bit/s	2,39 kbit/s
Profil A Extremwert	88 889 kbit/s	8	27,13 bit/s	1,36 kbit/s
Profil B Median	2,67 kbit/s	9	13,09 bit/s	0,65 kbit/s
Profil B Extremwert	2400 kbit/s	10	4,62 bit/s	0,23 kbit/s
		11	2,14 bit/s	0,11 kbit/s
		12	1,1 bit/s	0,06 kbit/s

Mit größerem SF erhöht sich die Sendedauer, wodurch die maximale Nutzdatenlänge kleiner wird. Gleichzeitig wird die Übertragung mit größerem SF robuster. In der Wahl des SF spiegeln sich so die Umgebungsbedingungen sowie die zu überbrückende Distanz zwischen Sender und Empfänger.

4.1.3 Abgleich des LoRaWAN-Leistungsprofils mit den Anforderungsprofilen

In Abschnitt 4.1.1 wird als Sendezeitabstand eine Stunde betrachtet. Mit Blick auf die Anforderungsprofile aus Tabelle 2 und Abb. 1 gilt dieser Sendezeitabstand für nur Profil A [Median]. Um nun die Anforderungsprofile und das Leistungsprofil von LoRaWAN miteinander zu vergleichen, wird im Folgenden die im Mittel benötigte bzw. bereitgestellte Datenrate R_m herangezogen. Wie in Gleichung (5) gezeigt wird sie unter Berücksichtigung der Anzahl der physikalischen Links n_PL, der je Link zu übertragenden Nutzdatenlänge l_usr und dem Sendezeitabstand Δt berechnet.

$$R_\mathrm{m} = \frac{8 \cdot l_\mathrm{usr} \cdot n_\mathrm{PL}}{\Delta t} \tag{5}$$

Die im Mittel geforderte Datenrate ist in Tabelle 6 je Anforderungsprofil genannt. Demgegenüber sind die mit LoRaWAN im Mittel erreichbaren Datenraten bei einem Duty Cycle von 1 % für einen und für 50 physikalische Links (PL) gezeigt. Dabei wird die in Tabelle 5 berechnet Nutzdatenlänge pro Stunde zugrunde gelegt. Die im Mittel geforderte Datenrate liegt zwischen 2,67 kbit/s and 88 889 kbit/s. Dagegen können mit LoRaWAN nur je Gerät 47,85 bit/s erreicht werden. Werden insgesamt 50 physikalische Links genutzt, kann die Datenrate bis zu 2,39 kbit/s betragen. Diese Datenrate reicht auf den ersten Blick fast für das Anforderungsprofil B [Median] aus, doch bleibt zu prüfen, inwiefern Applikationen dieses Profils mit einer Paketverlustrate von mindestens 63 % umgehen.
Insgesamt zeigt der Vergleich von Anforderungs- und Leistungsprofilen, dass LPWANs, die im unlizenzierten Band arbeiten, für die Erfüllung der quantitativen Anforderungen der ermittelten Profile der wasserwirtschaftlichen Anwendungen nicht ausreichen.

4.2 Erfüllung qualitativer Anforderungen

Erweist sich das Leistungsprofil einer Technologie als für ein Anforderungsprofil geeignet, kann die Erfüllung der qualitativen Anforderungen den entscheidenden Einfluss auf die Entscheidung für oder gegen eine Kommunikationslösung nehmen. Zu qualitativen Anforderungen wurden die Aspekte Ausfallsicherheit, beispielsweise durch Redundanz, Verschlüsselung und Authentifizierung, Skalierbarkeit der Funkkommunikationslösung, Nutzerfreundlichkeit, kommerzielle Verfügbarkeit und Investitionssicherheit genannt.
Die Ausfallssicherheit ist ein Parameter, der zum Teil durch das Leistungsprofil abgedeckt werden kann. Ein anderer Anteil betrifft die Wahl des Betreibermodells. Ob es für eine spezifische Anwendung sinnvoll ist, ein eigenes Netz zu betreiben oder auf einen Netzbetreiber zurückzugreifen, kann Einfluss auf die Verfügbarkeit des Netzes haben. Es ist denkbar, dass im Sinne von redundanten, eventuell sogar technologisch voneinander unabhängigen Kommunikationslösungen verschiedene Betreibermodelle parallel genutzt werden. Die Flexibilität kommt den Anforderungen der wasserwirtschaftlichen Anwendungen entgegen.

Verschlüsselung und Authentifizierung sind, außer für Sigfox, technologieseitig gegeben. Der Nutzung öffentlicher Infrastruktur steht damit aus Sicht der wasserwirtschaftlichen Anwendungen nichts entgegen.
Hinsichtlich der Skalierbarkeit zeigt bereits der Abgleich der quantitativen Parameter der Leistungs- und Anforderungsprofile, dass LPWANs nicht ausreichen. Entsprechend sind selbst die moderaten Anforderungen hinsichtlich der Skalierbarkeit eine Herausforderung.
Komplexe Aufgaben entsprechend Anforderungsprofil B brauchen vermutlich aufwändigere Darstellungs- und Auswertungsoptionen als Aufgaben der Zustandsüberwachung entsprechend Anforderungsprofil A, wodurch die Komplexität der Nutzeroberfläche größer sein könnte. Ob die resultierende Nutzerfreundlichkeit für den avisierten Anwendungsfall ausreicht, ist je spezifischem Anwendungsfall zu betrachten.
Die kommerzielle Verfügbarkeit scheint für verschiedene Technologien und verschiedene Komplettlösungen gut, doch ob die angebotenen Geräte tatsächlich ausreichen, ist individuell zu prüfen.
Da das Interesse an LPWANs im Moment einen Schub erfährt, ist bisher nicht abzusehen, ob eine Technologie sich durchsetzt, oder welche Technologien am Markt koexistieren werden. Die Investitionssicherheit muss entsprechend ebenfalls individuell bewertet werden.

5 Zusammenfassung und Ausblick

Auf Basis einer Umfrage, an der bundesweit Akteurinnen und Akteure der Wasserwirtschaft teilnahmen, wurden Anforderungsprofile für zwei Gruppen von Anwendungen der Wasserwirtschaft entwickelt. Anwendungen, welche Aufgaben der Zustandsüberwachung ausführen, werden durch das Anforderungsprofil A repräsentiert, werden Anforderungsprofil B Anwendungen umfasst, die auch steuernd in einen Prozess eingreifen. Aufgrund der großen zu überbrückenden Distanz und den teilweise unzugänglich und abgelegen platzierten Sensoren und Aktoren erscheinen Funkkommunikationstechnologien der Low Power Wide Area Networks (LPWANs) interessant.
Über sieben aktuell kommerziell verfügbare LPWAN-Technologien wurde ein kurzer Überblick gegeben. Potentiell scheinen alle Technologien, außer Sigfox, geeignet zu sein. Doch eine genauere Analyse unter Berücksichtigung der Sendezeitbegrenzung durch den regulatorisch vorgegebenen Duty Cycle zeigt, dass die im Mittel verfügbare Datenrate nicht für die betrachteten Anwendungen ausreicht. Es wurde deutlich, dass sich ein (Funk-)medienzugriff ohne Maßnahmen wie „Listen before talk“ positiv auf den Energieverbrauch auswirkt, allerdings mit Nachteilen eines unkoordinierten Protokolls umgehen muss. Der Betrieb eines Funksystems an den Grenzen des Duty Cycles führt zu erheblichen Paketverlusten von sogar über 60 % beim Maximum des Durchsatzes von 18 %. Selbst, wenn diese Kollisionswahrscheinlichkeit in Kauf genommen werden kann, reicht die bereitgestellte Datenrate nicht aus. Die Rolle qualitativer Anforderungen beispielsweise an das Betreibermodell, die Nutzerfreundlichkeit oder die Investitionssicherheit wurde diskutiert. Allerdings muss individuell festgestellt werden, inwiefern eine konkrete Kommunikationslösung den jeweiligen Anforderungen genügt.
Um für Anwendungen der Wasserwirtschaft eine geeignete Funktechnologien zu identifizieren, kann an verschiedenen Punkten weitergearbeitet werden. Auf Basis des Überblicks zu konkreten LPWAN-Technologien können im nächsten Schritt Akteurinnen und Akteure der Wasserwirtschaft ihr Anforderungsprofil überdenken. Hieraus ergibt sich gegebenenfalls eine Anpassung der Anforderungen, zum Beispiel eine Reduktion der Nutzdatenlänge durch applikationsseitige Vorverarbeitung der Sensordaten (Edge Computing).
Hinsichtlich der Funkkommunikationstechnologien könnten NB-IoT und LTE-M für einen spezifischen Anwendungsfall analysiert werden. Außerdem könnte das Protokoll von Technologien, die das lizenzfreie Band nutzen, angepasst werden, um den Durchsatz zu verbessern. Hier könnte eine Synchronisation der Funkgeräte die Kollisionswahrscheinlichkeit verringern, ohne dass Verfahren wir „Listen before talk“ implementiert werden müssten (analog zu „Slotted Aloha“).

6 Hinweis

Das diesem Beitrag zugrundeliegende Vorhaben „Industrial Radio Lab Germany“ wurde mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) unter dem Förderkennzeichen 16KIS1013 gefördert.

Literaturverzeichnis

[1] VDI/VDE, "Funkgestützte Kommunikation in der Automatisierungstechnik - Anforderungen und Grundlagen," *VDI/VDE Richtlinie 2185 Blatt 1*, 2020.

[2] G. Scheible, D. Dzung, J. Endresen, and J. E. Frey, "Unplugged but connected [design and implementation of a truly wireless real-time sensor/actuator interface]," *IEEE Industrial Electronics Magazine*, vol. 1, no. 2, pp. 25–34, 2007.

[3] L. Underberg and S. Willmann, "Categorization of industrial communication requirements as key to developing application profiles," *IFAC World Congress*, 2020.

[4] 3GPP, "TR 22.804, V16.2.0, Technical Specification Group Services and System Aspects; Study on Communication for Automation in Vertical Domains (Release 16) [Revision 2018-12]," 2017.

[5] 3GPP, "Technical Specification 22.104: Group Services and System Aspects; Service requirements for cyber-physical control applications in vertical domains; Stage 1 (Release 16), V16.1.0," March 2019.

[6] 3GPP, "Technical Specification 22.104: Group Services and System Aspects; Service requirements for the 5G system; Stage 1 (Release 16), V16.7.0," March 2019.

[7] 5G-ACIA, "Key 5G Use Cases and Requirements," *White Paper*, May 2020.

[8] 5G-ACIA, "A 5G Traffic Model for Industrial Use Cases," *White Paper*, November 2019.

[9] ETSI EN 300 220, "Short Range Devices (SRD) operating in the frequency range 25 MHz to 1 000 MHz; Part 2: Harmonised Standard for access to radio spectrum for non specific radio equipment; V3.2.1," 2018.

[10] LoRa Alliance. https://lora-alliance.org/. Accessed: 24 Sep 2021.

[11] mioty Alliance. https://mioty-alliance.com/. Accessed: 24 Sep 2021.

[12] Sigfox S.A. https://www.sigfox.com/en. Accessed: 24 Sep 2021.

[13] Weightless Alliance. https://www.weightless-alliance.org/. Accessed: 24 Sep 2021.

[14] ETSI TS 103 357, "Short Range Devices; Low Throughput Networks (LTN); Protocols for radio interface A; V1.1.1," 2018.

[15] L. Chettri, "A comprehensive survey on internet of things (iot) toward 5g wireless systems," in *IEEE Internet of Things Journal, Bd. 7*, 2018.

[16] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," in *ICT Express, vol. 5, no. 1, pp. 1-7*, March 2019.

[17] Weightless SIG, "Weightless-P System Specification V1.0," 2015.

[18] LoRa Alliance, "LoRaWAN (TM) V1.0.3 Specification," 2018.

[19] S. Popli, R. K. Jha, and S. Jain, "A survey on energy efficient narrowband internet of things (nbiot): Architecture, application and challenges," in *IEEE Access, pp. 16739 - 16776*, November 2018.

[20] ubiik, "Weightless LPWAN Low Power Wide Area Networking Product Catalogue." https://www.european-utility-week.com/___media/libraries/brochures/782C0264-DE2B-F4D2-012E47E2EEF3DB72-document.pdf, 2018. Accessed: 24 Sep 2021.

[21] N. Abramson, "THE ALOHA SYSTEM: another alternative for computer communications," *Fall Joint Computer Conference*, vol. 37, pp. 281–285, 01 1977.

[22] SEMTECH, "SX1272/3/6/7/8 LoRa Modem Design Guide." https://www.rs-online.com/designspark/rel-assets/ds-assets/uploads/knowledge-items/application-notes-for-the-internet-of-things/LoRa%20Design%20Guide.pdf, 2013. Accessed: 24 Sep 2021.

http://dx.doi.org/10.25673/39574

Improvements for Time Synchronization with 5G Transparent Clocks

Tobias Striffler

Technology
Siemens AG
Otto-Hahn-Ring 6
81739 München
tobias.striffler@siemens.com

Prof. Dr.-Ing. Hans D. Schotten

Institute for Wireless Communication and Navigation
TU Kaiserslautern
Paul-Ehrlich-Straße
67663 Kaiserslautern
schotten@eit.uni-kl.de

Abstract: Deterministic, time-sensitive communication over integrated wired and wireless networks is one of the key enablers for Industry 4.0 and future factories, with current efforts in research and standardization focusing on integrating 5G and TSN. However, enabling the required interaction between two separate and different networks also introduces new issues. We focus on the disparity of the 5G System behaving as a single logical TSN Bridge for the sake of this integration. In this paper we investigate how to reduce the time synchronization errors caused by the 5G Systems approach to interacting with the TSN network. An ns-3 based simulation of an integrated 5G – TSN network is used to validate the proposed improvements.

1 Introduction

Part of the vision of Industry 4.0 is the shift towards the "smart factory": Instead of single purpose and static manufacturing, a flexible and mobile factory environment that can be adjusted to react to changing priorities and needs is envisioned. Flexible and mobile production environments (e.g., involving automated guided vehicles or mobile robots) require a communication infrastructure that can provide the necessary capabilities. A communication infrastructure that is capable of deterministic and time sensitive wireless communication without having to completely replace existing hardware is needed. The focus of currently ongoing standardization efforts by the 3GPP have been on the integration of IEEE TSN [1] and 5G [2].
This integration is realized such that the 5G System presents itself as a TSN Bridge to the TSN Network [3]. The advantage is, that the TSN network does not need to be aware of the 5G System. No change to existing procedure is required. However, the differences between a traditional TSN Bridge and a 5G logical TSN Bridge may result in additional errors when synchronizing over a 5G network [4].
In this paper, we start with a problem description in Chapter 2 and following that, propose modifications to how the 5G System interacts with gPTP synchronization messages in order to mitigate the impact of these differences. A simulation, described in Chaper 3, is used to evaluate these modifications in comparison to the standardized approach. This evaluation is shown in Chapter 4. A conclusion to this work is then given in Chapter 5.

1.1 Related Works

A general overview of wireless ultra-reliable, low latency communication (URLLC) and its challenges was given by Mahmood et al. [5]. One of the key use cases for future factories, cooperative mobile robots, is examined by Gundall et al [6]. They propose an integration concept for 5G and TSN networks based on the requirements they derive in their work and validate their concept in a demonstrator. Schüngel et al. [7] investigate the time synchronization performance of an integrated 5G-TSN network according to 3GPP Release 16. The theoretical model they derive is validated with a simulation. However, these works do not consider the specific effects due to the disparity between a TSN Bridge and 5G logical TSN Bridge. We investigated these effects in [4].

Regarding synchronization procedures, the 3GPP assumes the 5G internal synchronization to be achieved via System Information Block (SIB) messages [8]. Synchronization in the TSN network uses (g)PTP [1]. Our contribution in this work is the mitigation of the synchronization error introduced due to the disparity between a TSN bridge and a 5G logical TSN bridge in regard to handling of (g)PTP messages by the 5G System.

2 Problem Description and Proposed Modifications

In this chapter, we will first describe the problem with 5G and TSN integrated networks and then go on to show our proposed approach to mitigating it.

2.1 Problem Description

In [4] we investigated the difference between a traditional TSN Bridge that is integrated as a transparent clock into a TSN network, and a 5G System that is integrated as a logical TSN Bridge (and transparent clock) into a TSN network.

Synchronization in a TSN network is achieved by timestamping of a periodically exchanged synchronization message between the master clock and any clock that is synchronized to it. This synchronization procedure is explained in detail in the IEEE standards documents IEEE 1588 [9] and IEEE 802.1AS [1]. In this comparison, the important part is the calculation of the residence time in a bridge. As shown in Equation 3, the residence time is calculated by subtracting the ingress timestamp from the egress timestamp and multiplying the result with the cumulative rate ratio (CRR) [1].

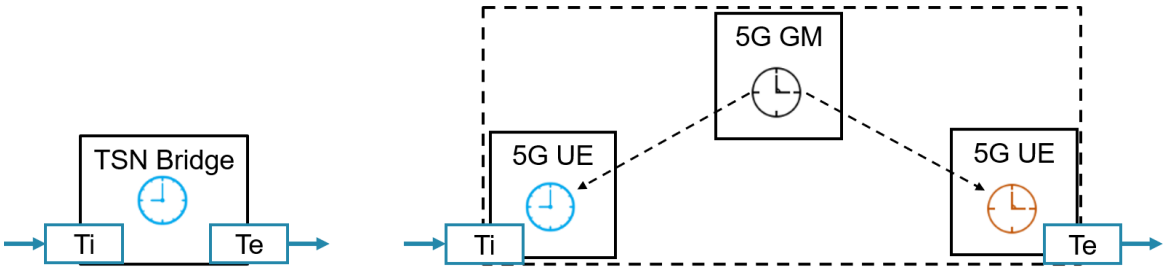

Figure 1: Difference in timestamping setup between TSN Bridge and 5G logical TSN Bridge

This method assumes that both ingress and egress timestamp are made based on the same, continuous time. While a TSN Bridge is a single device with a single clock, a 5G System is not. The ingress and egress timestamps are made by separate devices, with their own clocks, as shown in Figure 1.

Summarizing, there are two key differences between a TSN Bridge and a 5G logical TSN Bridge:

- **Clocks:** The residence calculation does not consider that ingress and egress timestamps are made based on different clocks. The resulting difference in clock drift introduces an additional synchronization error.
- **Time:** TSN Bridges acting as transparent clocks are typically free flowing to avoid time jumps due to the synchronization procedure resulting in offset corrections. That is not possible in the 5G System. Both ingress and egress devices have to be synchronized to the 5G grandmaster (GM). The resulting offset corrections introduce an additional synchronization error. How a requirement for the 5G clocks to provide a monotonously increasing time would impact our proposed modifications is out of scope of this work.

In the following sections we will describe our proposed approach to adjusting the synchronization procedure to mitigate the impact of these issues.

2.2 Clock Drift

From Release 16 onward, the 5G GM, typically the gNodeB (gNB), allows the synchronization of 5G UEs by including reference time information in the periodically distributed SIB messages [8]. In contrast to the PTP procedure in IEEE 1588 [9] and IEEE 802.1AS [1], this procedure does not account for frequency drift between master and slave clocks. The error resulting from this clock drift is kept sufficiently small by using more stable oscillators and choosing an appropriately small synchronization period.

As the goal of our proposed approach is to compensate for the error caused by the difference in clock drift between ingress and egress devices, we need to know said clock drift. Therefore, we first propose a simple method to determine the clock drift between a 5G user equipment (UE) and the 5G GM based on the periodically distributed SIB messages.

The ratio of the frequencies of two clocks can be calculated by comparing the same time interval as measured by each clock. For the 5G System we propose to use the SIB messages to calculate the frequency ratio, based on the reference time stored in the message and the receive time at the 5G UE. The resulting calculation is shown in Equation 1,

$$NRR_{5G} = \frac{T^{i}_{\text{reference,gNB}} - T^{i-1}_{\text{reference,gNB}}}{T^{i}_{\text{receive,UE}} - T^{*i-1}_{\text{receive,UE}}} \quad \text{Eq. 1}$$

where $T^{i}_{\text{reference,gNB}}$ is the reference time from the 5G gNB and $T^{i}_{\text{receive,UE}}$ is the receive time at the UE, both at the i-th instance of the periodic message.

However, as the receiving UE updates its clock offset with every received SIB message, the constant part of its clocks drift is compensated (as the times in Equation 1 are therefore also periodic). Due to these times being linked to the synchronization process, we can adjust the procedure to take this into account: The current receive time, $T^{i}_{receive,UE}$, is taken before the offset is corrected. After the neighbour rate ratio (NRR) and the new offset have been calculated, the receive time is adjusted by the new offset before being stored, as shown in Equation 2, where T^{i}_{offset} is the value of the UE clock offset after the i-th synchronization event:

$$T^{*i}_{\text{receive,UE}} = T^{i}_{\text{receive,UE}} - \left(T^{i}_{\text{offset}} - T^{i-1}_{\text{offset}}\right), \quad \text{Eq. 2}$$

2.3 Residence Time

As stated above, the 5G System is integrated into the TSN network by having it behave like a transparent TSN Bridge. This means, the duration the synchronization message spent in the 5G logical TSN Bridge is measured and stored in the message. This duration is the residence time (see Equation 3), and it is calculated as the difference between the timestamps at the ingress and egress ports of the 5G System, multiplied by the CRR to get the residence time in the TSN GMs time base.

The CRR is the rate ratio between the TSN GM and the current ingress port. As this ratio cannot be calculated directly (unless the current ingress port is a direct neighbour of the TSN GM), the CRR is stored in the synchronization message and updated at every ingress port the message passes. As shown in Equation 4, the CRR at device n is the product of the NRRs between all previous devices along the synchronization path, up to NRR^{n} between the current and preceding device.

$$T_{\text{residence}} = \text{CRR} \cdot (T_{\text{egress}} - T_{\text{ingress}}) \quad \text{Eq. 3}$$

$$\text{CRR}^{n} = \text{CRR}^{n-1} \cdot \text{NRR}^{n} \quad \text{Eq. 4}$$

Figure 1 shows the simplified behaviour of the ingress and egress clocks compared to the 5G GM clock: The clocks drift relative to the 5G GM and relative to each other. This drift has both a constant and a time-variant component. In the 5G System, this drift is corrected by periodically repeating the synchronization procedure and adjusting the respective offset to the 5G GM. In regard to the resulting behaviour, Figure 1 also shows the times of the relevant events.

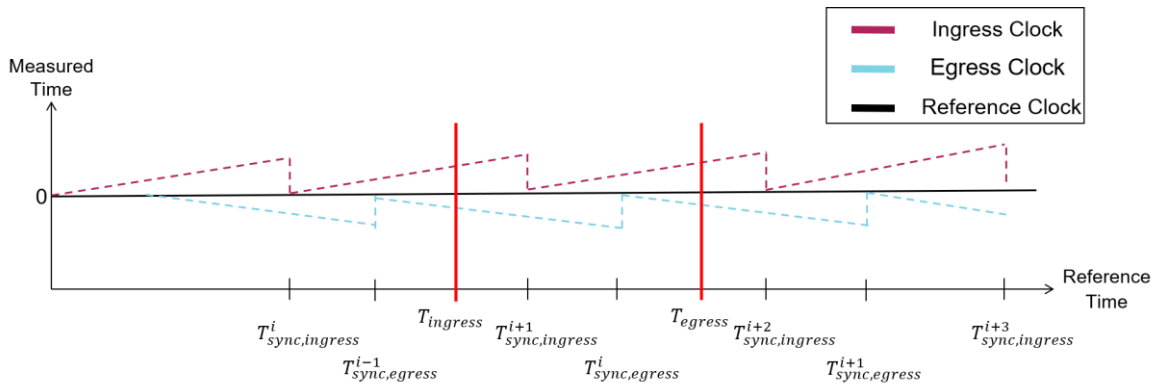

Figure 2: Simplified synchronized clock behavior relative to a reference clock with relevant events

$T^{i}_{\text{sync,ingress}}$ - the time of the i-th synchronization event at the ingress (or egress) UE

T_{ingress} - the ingress (or egress) timestamp used for the residence time calculation

In addition to these (known) times, our proposed approach requires knowledge of the relative clock frequencies of the involved devices:

$\frac{f_{\text{TSN}}}{f_{\text{5G,ingress}}}$ - the rate ratio between the previous TSN device and the ingress UE

$\frac{f_{\text{5G,GM}}}{f_{\text{5G,ingress}}}$ - the rate ratio between the 5G GM and the ingress UE

$\frac{f_{\text{5G,GM}}}{f_{\text{5G,egress}}}$ - the rate ratio between the 5G GM and the egress UE

The rate ratio between the ingress UE and the previous TSN device is known at the ingress UE and calculated via the peer delay mechanism as described in IEEE 802.1AS [9]. The rate ratios between the UEs and the 5G GM is not typically known, here we use the approach we propose in Chapter 2.2 to calculate these ratios.

In general, the approach described here aims to convert both the ingress and egress timestamps to the 5G GMs time base. By having a common understanding of time for both timestamps, the residence time can then be calculated and converted to the TSN GMs time, similar to a TSN Bridge.

First, the ingress timestamp is adjusted before it is stored in the synchronization message, as shown in Equation 5 and Equation 6.

$$T^{\text{corrected}}_{\text{ingress}} = T^{*i}_{\text{sync,ingress}} + \left(T_{\text{ingress}} - T^{*i}_{\text{sync,ingress}}\right) \cdot \frac{f_{\text{5G,GM}}}{f_{\text{5G,ingress}}} \quad \text{Eq. 5}$$

$$T^{*i}_{\text{sync,ingress}} = T^{i}_{\text{sync,ingress}} - \left(T^{i}_{\text{offset,ingress}} - T^{i-1}_{\text{offset,ingress}}\right) \quad \text{Eq. 6}$$

The CRR has to be adjusted in two steps, first at the ingress UE and then again at the egress UE, in order to accurately reflect the two devices of the 5G logical TSN Bridge. At the ingress UE, the rate ratio to the previous TSN bridge and the inverse of the rate ratio to the 5G GM are used to update the CRR (see Equation 7).

$$\text{CRR}^{n}_{\text{temporary}} = \text{CRR}^{n-1} \cdot \frac{f^{n-1}}{f_{\text{5G,ingress}}} \cdot \frac{f_{\text{5G,ingress}}}{f_{\text{5G,GM}}} \quad \text{Eq. 7}$$

where CRR^{n-1} is the CRR after the previous TSN device (stored in the synchronization message).

Second, the egress timestamp is adjusted, and the residence time is calculated according to Equation 8 and Equation 9.

$$T^{\text{corrected}}_{\text{egress}} = T^{*i}_{\text{sync,egress}} + \left(T_{\text{egress}} - T^{*i}_{\text{sync,egress}}\right) \cdot \frac{f_{\text{5G,GM}}}{f_{\text{5G,egress}}} \quad \text{Eq. 8}$$

$$T^{*i}_{\text{sync,egress}} = T^{i}_{\text{sync,egress}} - \left(T^{i}_{\text{offset,egress}} - T^{i-1}_{\text{offset,egress}}\right) \quad \text{Eq. 9}$$

$$T_{\text{residence}} = \text{CRR}^{n}_{\text{temporary}} \cdot \left(T^{\text{corrected}}_{\text{egress}} - T^{\text{corrected}}_{\text{ingress}}\right) \quad \text{Eq. 10}$$

After the residence time calculation in Equation 10, the final CRR is calculated according to Equation 11 and stored in the synchronization message.

$$\text{CRR}^{n} = \text{CRR}^{n}_{\text{temporary}} \cdot \frac{f_{\text{5G,GM}}}{f_{\text{5G,egress}}} \quad \text{Eq. 11}$$

3 Simulation

In this chapter, we give a short overview of the simulation used to test our proposed modifications.

The simulation we use is built in ns-3 [10], and is based on the interactions of the TSN and 5G devices with the PTP synchronization messages. The 5G System part of this simulation is based on the 5G LENA model [11]. The major components to enable these interactions are the clocks, unique to each device, and a timestamping mechanism (as PTP is timestamp-based). More details on this simulation can be found in [4].

3.1 Clocks

Each clock entity is based on the simulation time t provided by ns-3 and therefore capable of perfectly accurate timekeeping. In order to model real-world clocks, errors are added on top of the simulation time. As ns-3 is an event-based simulator, the clock entity updates the time at every event where it is accessed. The resulting time T^i, at the i^{th} event, is calculated based on the difference in simulation time since the i-1st event and the error $c(t^i)$ of this clock (see Equation 12). This error is modelled as the sum of the constant frequency offset f and the time-variant frequency drift rate f' (see Equation 13). If the clock is synchronized to a master clock, the offset T^i_{offset} is added on top in Equation 14.

$$T^i = T^{i-1} + (t^i - t^{i-1}) \cdot (1 + c(t^i)) \quad \text{Eq. 12}$$

$$c(t^i) = f + \int_{t^{i-1}}^{t^i} f' \cdot \sin(t)\, dt \quad \text{Eq. 13}$$

$$T^i_{\text{sync}} = T^i + T^i_{\text{offset}} \quad \text{Eq. 14}$$

3.2 Timestamping

The synchronization procedure implemented here is based on the (g)PTP synchronization described in IEEE 802.1AS [1]. Synchronization messages are timestamped at every ingress and egress port they pass. The timestamp T^i_{TS} is based on the (erroneous) time provided by the local clock entity connected to that port, and an additional error is added for the timestamping event itself (see Equation 15). This time error (TE) is based on the IEC/IEEE 60802 Use Cases document [12] and has both a constant part cTE and a dynamic (random) part dTE.

$$T^i_{\text{TS}} = T^i_{\text{sync}} + cTE + dTE \quad \text{Eq. 15}$$

3.3 Scenario & Procedure

The simulation scenario is shown in Figure 2. The TSN GM synchronizes the TSN Endstation via a line of Bridges and a 5G network (modelled as a logical TSN Bridge). We assume all Bridges to be transparent clocks, meaning they forward the synchronization messages but do not synchronize their own clocks to the TSN GM. Every TSN Bridge periodically performs the Peer Delay procedure as described in IEEE 802.1AS [1] to determine the delay and rate ratio to its neighbours. Only the TSN GM periodically sends out the synchronization message that is forwarded to the TSN Endstation.

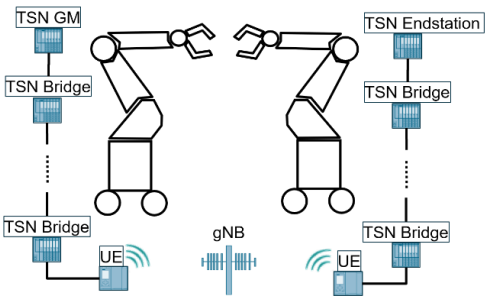

Figure 3: Two cooperative mobile robots as basic scenario for the simulation

4 Results

In this chapter, we look at the performance of our proposed modifications to the 5G-TSN synchronization procedure. First, we detail our choice of parameterization and then discuss the resulting data.

4.1 Setup & Parameters

As described in Chapter 3.3, the simulation models the synchronization procedure according to IEEE 8021AS [1]. The performance of the proposed modifications is based on the achieved synchronization accuracy for a variety of parametrizations. The general simulation parameters are shown in Table 1.

Parameter	Value
Simulation runs	100
Simulation duration	100s
TSN synchronization interval	125 ms [13]
TSN PDelay interval	31.25 ms [13]
5G synchronization interval	[10 ms, 40 ms, 80 ms] [8]
TSN clock frequency offset	50+*U*(-5,5) ppm [13]
TSN clock frequency drift	3 ppm/s (sinusoidal) [13]
TSN cTE	*U*(-10,10) ns [12]
5G cTE	*U*(-275,275) ns [14]
dTE	*U*(-20,20) ns [12]

Table 1: General Simulation Parameters

4.2 5G Clock Drift

First, we look at our proposed approach to determining the clock drift between a 5G UE and the 5G GM. The accuracy of the clock drift calculation depends on the accuracy of the respective timestamps. As shown in Table 1, we assume the 5G synchronization to be accurate within 275 ns according to [14]. As the clock drift calculation is based on the same timestamps as the synchronization itself, we can assume the same accuracy to apply here. Equation 16 gives us a theoretical upper limit, depending on the chosen synchronization interval $T_{sync,interval}$.

$$\max\left(E_{\mathrm{5G,NRR}}\right) = \frac{T_{\mathrm{sync,interval}}}{T_{\mathrm{sync,interval}} + 2 \cdot 275\mathrm{ns}} \qquad \text{Eq. 16}$$

For example, for $T_{\mathrm{sync,interval}} = 10\mathrm{ms}$, that results in a maximum calculation error of 55 ppm. As this maximum error is very unlikely to occur, we additionally take the median of a sliding window of the calculated drift values (similar to common practice in IEEE networks [13]). This reduces outliers and therefore increases the accuracy.

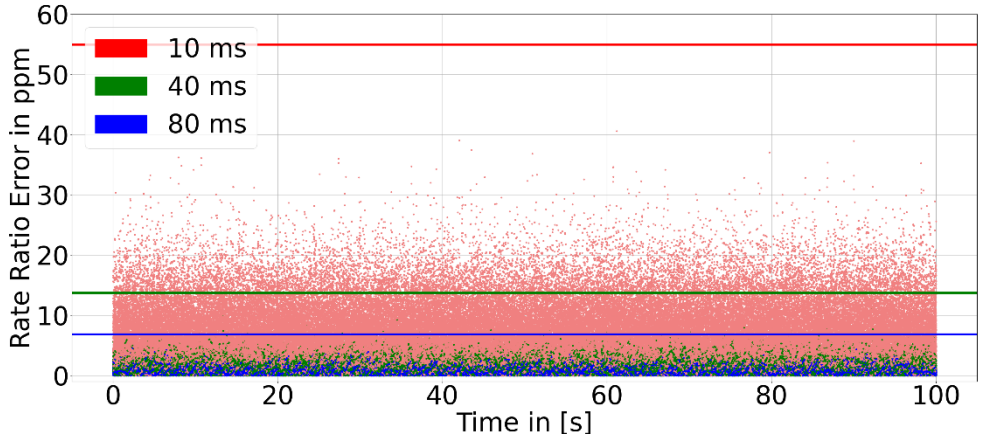

Figure 4: Absolute 5G UE-gNB rate ratio error for different 5G synchronization intervals

Figure 4 shows the achieved accuracy as the difference between the calculated drift and the actual drift applied in the simulation, for different synchronization intervals. The bold lines mark $\max(E_{5G,NRR})$ for the respective synchronization interval. The error is shown to be limited to roughly half the theoretical maximum. While the relative error is quite significant at up to 40 ppm for $T_{sync,interval} = 10ms$, it is limited to 7 ppm and 3 ppm for $T_{sync,interval} = 40ms$ and $T_{sync,interval} = 80ms$, respectively. While that is still far less accurate than the drift determination in a TSN network, it is sufficiently accurate to provide a benefit in our proposed 5G residence time calculation modification.

4.3 5G Residence Time

We evaluate our 5G residence time calculation modification by comparing the calculated residence time to the actual residence time in the simulation. As mentioned in Chapter 2, the residence time calculation in a 5G logical TSN Bridge has two issues:

1. The ingress and egress timestamps are added by separate devices, with possibly different clock drift behaviour and time errors.
2. For the CRR, only the NRR of the ingress device is taken into account. The separate egress device is not.

The impact of the constant and dynamic Time Errors (see Table 1) on the 5G residence time is limited to

$$\max(E_{5G,TE}) = \pm 2 \cdot (cTE_{max} + dTE_{max}) \quad \text{Eq. 17}$$

which results in $\max(E_{5G,TE}) = \pm 590ns$. The impact of the relative clock drift between the ingress and egress UEs depends on the synchronization period of both the 5G and TSN networks and is therefore limited to

$$\max(E_{5G,drift}) = \min(T^{TSN}_{sync,interval}, T^{5G}_{sync,interval}) \cdot f^{5G} \quad \text{Eq. 18}$$

For example, a clock frequency offset of ± 10 ppm and a clock frequency drift rate of ± 3 ppm/s at a $10ms$ 5G synchronization period result in $\max(E_{5G,drift}) = 260ns$.

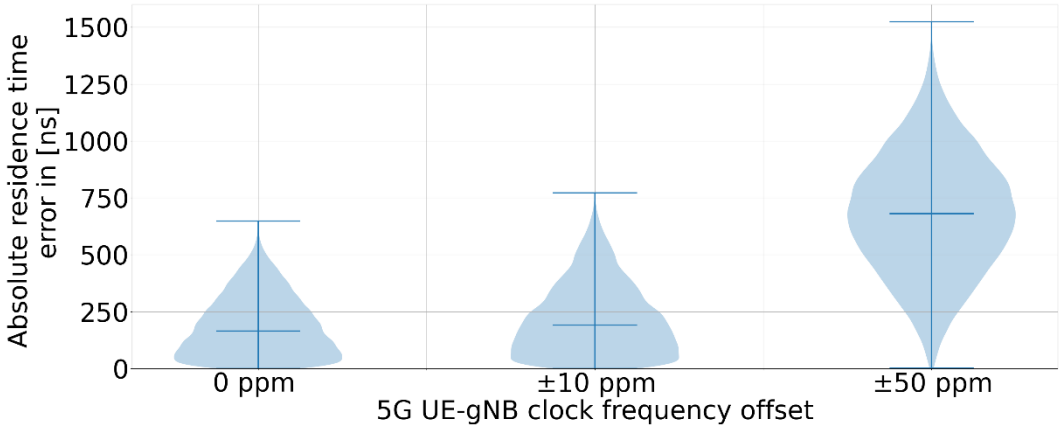

Figure 5: 5G residence time error without modifications for various UE-gNB frequency offsets

Figure 5 now shows the residence time error for the unmodified 5G system with a 5G synchronization interval of 10ms and for different clock frequency offsets between ingress and egress UEs. The violin plots show the full distribution of the data sets, with the extrema marked by horizontal lines.

The behaviour is as expected based on the error components explained above. For clocks that are stable relative to the 5G GM and each other, the impact of ignored egress clock drift is low. However, if the egress clock is allowed to drift relative to the ingress clock (or 5G GM), the resulting error increases significantly.

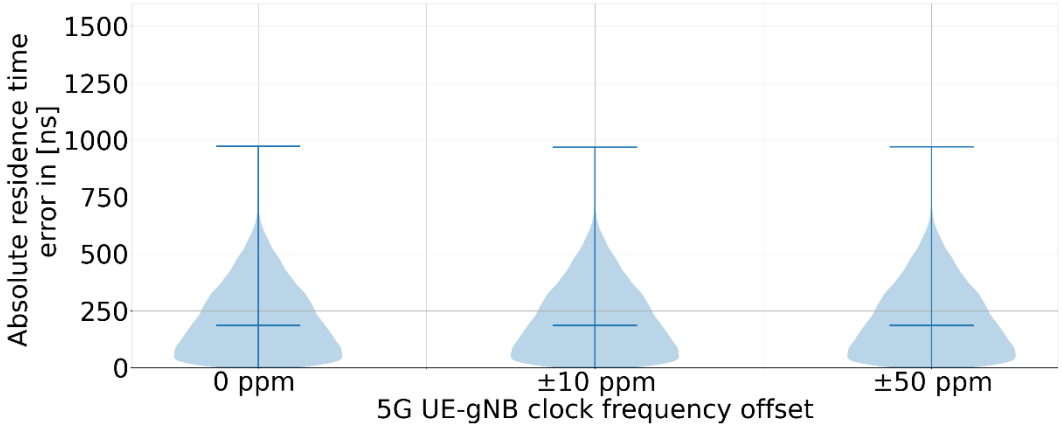

Figure 6: 5G residence time error with modifications for various UE-gNB frequency offsets

In comparison, Figure 6 shows a consistent residence time error independent of the relative clock frequency offset. That is due to the clock frequency offset being compensated by the modified residence time calculation, the residence time error is decoupled from the clock drift. However, for low clock frequency offsets, the error is larger than the unmodified calculation. That is due to the error in determining the rate ratio between the 5G UE and 5G GM, that may exceed the actual rate ratio for very stable oscillators. This may be improved by reducing the uncertainty in the transmit and receive times used to calculate the rate ratio, for example by enabling more precise hardware timestamping at the 5G radio antennas.

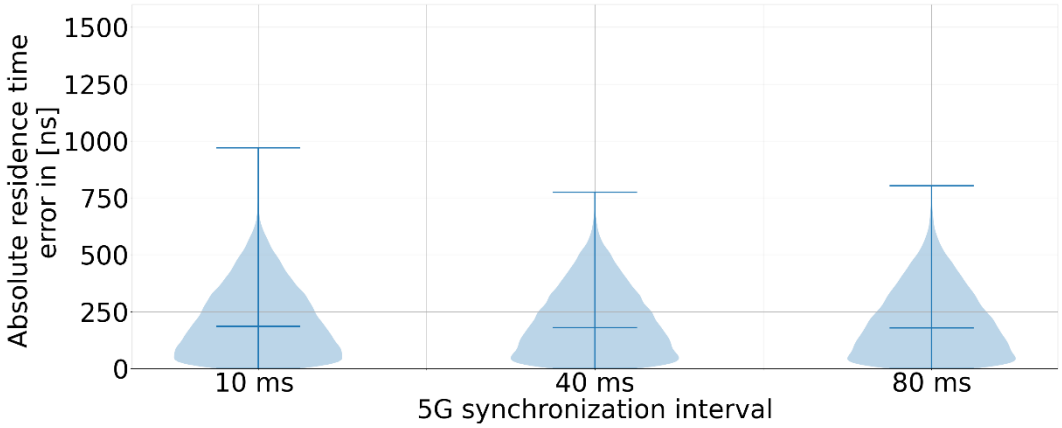

Figure 7: 5G residence time error with modifications for various 5G synchronization intervals

Figure 7 shows the residence time error for different 5G synchronization intervals. We can see that the large error in determining the rate ratio for a short 5G synchronization interval, as shown in Figure 4, results in a large residence time error, as shown in Figure 6. This error is reduced when increasing the 5G synchronization interval, due to the smaller resulting rate ratio error.

We also note that the CRR is used to calculate the residence time in TSN GM time at every TSN Bridge along the path to the synchronizing TSN Endstation. Regarding the CRR calculation, the 5G ingress and egress devices could be considered neighbouring devices. Thus, the error due to not including the 5G egress devices' clock frequency offset relative to the 5G ingress devices clock frequency offset is propagated until the TSN Endstation, resulting in an error for every further residence time calculation.

Considering that the CRR is the product of the NRR of all preceding devices, as shown in Equation 4, the error resulting from removing one device (device n-1 in Equation 19 and 20) from the CRR would be equal to the inverse of the missing NRR.

$$\mathrm{CRR^n} = \frac{f_{GM}}{f^1} \cdot \frac{f^1}{f^2} \cdot \ldots \cdot \frac{f^{n-2}}{f^{n-1}} \frac{f^{n-1}}{f^n} \qquad \text{Eq. 19}$$

$$\mathrm{CRR^n} \cdot \frac{f^{n-1}}{f^{n-2}} = \frac{f_{GM}}{f^1} \cdot \frac{f^1}{f^2} \cdot \ldots \cdot \frac{\cancel{f^{n-2}}}{\cancel{f^{n-1}}} \cdot \frac{\cancel{f^{n-1}}}{\cancel{f^{n-2}}} \cdot \frac{f^{n-1}}{f^n} \qquad \text{Eq. 20}$$

In the case of the 5G egress clock frequency offset not being included, we assume the error to be equal to $\frac{f_{5G,egress}}{f_{5G,ingress}}$, the inverse of the NRR at the 5G egress device. Figure 8 shows the resulting error on the residence time calculation for TSN devices before (here noted as "Baseline") and after the 5G Systems' impact on the stored CRR. The resulting error aligns with our expectation: ± 10ppm frequency offset and ± 3ppm/s frequency drift result in a 26ns error for a 1ms residence time.

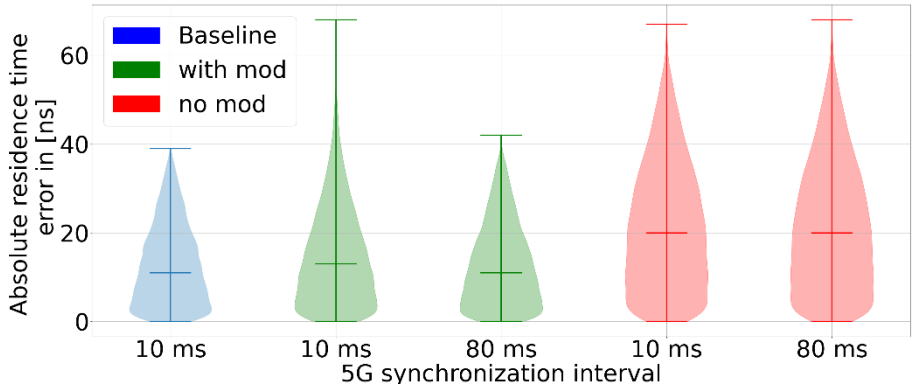

Figure 8: TSN residence time error after the 5G System for $T_{\mathrm{sync}}^{5G} = 10$ms and ± 10ppm 5G clock frequency offset

5 Conclusion

In this work, we propose a modification to the residence time calculation in a 5G logical TSN Bridge in order to compensate for the difference to a traditional transparent TSN Bridge. We show that the performance is comparable for very stable 5G clocks and short 5G synchronization intervals. For longer 5G synchronization intervals and less stable clocks in the 5G UEs, our modified approach to the residence time calculation stays relatively stable, while the unmodified approach fails. As mentioned in the introduction, the flexibility provided by wireless communication is an important part of future factories. More wireless communication devices will be required, therefore one of the relevant factors in deciding whether 5G and TSN will succeed in enabling Industry 4.0 will be the device cost. These modifications may enable the use of cheaper hardware for integrated 5G-TSN networks, though further research is required before use in industrial communication devices is possible.

For future work, a way to determine the clock drift more accurately inside the 5G Network should be investigated, as it was shown to be a limiting factor regarding the achievable accuracy. In addition, we did not consider the impact of requiring the clocks to always provide a monotonously increasing time. The current approach, based on the clocks' offset, will have to be adjusted to the different clock behaviour in such a case. This question will have to be addressed for full viability in industrial use cases.

6 Bibliography

[1] *802.1AS-2020: IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications,* IEEE, 2020.

[2] *TS 23.501 System architecture for the 5G System (5GS),* https://www.3gpp.org/DynaReport/23501.htm: 3GPP, 2021.

[3] T. Striffler, N. Michailow und M. Bahr, „Time-Sensitive Networking in 5th Generation Cellular Networks - Current State and Open Topics," in *IEEE 2nd 5G World Forum (5GWF),* 2019.

[4] T. Striffler und H. D. Schotten, „The 5G Transparent Clock: Synchronization Errors in Integrated 5G-TSN Industrial Networks," in *INDIN 2021: IEEE International Conference on Industrial Informatics,* 2021.

[5] A. Mahmood, M. I. Ashraf, M. Gidlund und J. Torsner, „Over-the-Air Time Synchronization for URLLC: Requirements, Challenges and Possible Enablers," in *15th International Symposium on Wireless Communication Systems (ISWCS),* 2018.

[6] M. Gundall, C. Huber, P. Rost, R. Halfmann und H. D. Schotten, „Integration of 5G with TSN as Prerequisite for a Highly Flexible Future Industrial Automation: Time Synchronization based on IEEE 802.1AS," in *Annual Conference of the IEEE Industrial Electronics Society (IECON-2020),* 2020.

[7] M. Schüngel, S. Dietrich, D. Ginthör, S. P. Chen und M. Kuhn, „Analysis of Time Synchronization for Converged Wired and Wireless Networks," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA),* 2020.

[8] *TS 38.331 NR; Radio Resource Control (RRC) protocol specification (Release 16),* https://www.3gpp.org/DynaReport/38331.htm: 3GPP, 2021.

[9] *1588-2019: IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems,* IEEE, 2019.

[10] *The ns-3 network simulator,* http://www.nsnam.org/, 2021.

[11] N. Patriciello, S. Lagen, B. Bojovic und L. Giupponi, *An E2E simulator for 5G NR networks,* Simulation Modelling Practice and Theory, 2019.

[12] *Use Cases IEC/IEEE 60802,* IEC and IEEE, 2018.

[13] G. M. Garner, *New Simulation Results for Time Error Performance for Transport over an IEC/IEEE 60802 Network Based on Updated Assumptions,* 2020.

[14] *R2-2010837 Reply LS on propagation delay compensation enhancement,* 3GPP, 2021.

http://dx.doi.org/10.25673/39578

Co-configuration of 5G and TSN enabling end-to-end quality of service in industrial communications

Lukas Martenvormfelde[1], Arne Neumann[1], Lukasz Wisniewski[1], and Lukas Schreckenberg[2]

[1] inIT - Institute industrial IT, Technische Hochschule Ostwestfalen-Lippe, Campusallee 6, 32657 Lemgo, Germany
{lukas.martenvormfelde, arne.neumann, lukasz.wisniewski}@th-owl.de
[2] Fraunhofer IOSB-INA, Campusallee 1, 32657 Lemgo, Germany
lukas.schreckenberg@iosb-ina.fraunhofer.de

Abstract. The recent trends in automation require a highly reliable communication in order to enable distributed controls in real-time applications. Time Sensitive Networking and 5G promise to provide the required Quality of Service for the wired and wireless communication, respectively. However, high reliability comes at the expense of a high configuration and engineering effort especially for heterogeneous networks. This paper aims to reduce the barriers of a successful co-configuration 5G and TSN. Therefore, a broad overview over the specifications of both technologies is given with a focus on the individual and joint configuration of both systems. Subsequently, architectural aspects of the co-configuration and a mapping of 5G and TSN parameters are pointed out to enable end-to-end quality of service in industrial communications.

1 Introduction

In recent years, the changes towards Industry 4.0 increased the demands on the underlying communication infrastructure. Particularly, it is required to reliably connect hundreds and thousands of devices some of which have stringent requirements on data rates, error rates or timeliness of the packets. Since it would hardly be possible to cover all aspects in a single communication technology, industrial communication is and will be characterized by combining different network technologies bringing all their beneficial sets of features into a network [16]. The efficient configuration, deployment and maintenance of the resulting heterogeneous networks is considered as a relevant challenge in the area of industrial networks [6]. With respect to demanding applications, several network parameters have to be thoroughly adjusted in order to achieve the required quality of service. Consequently, combining two or more network technologies together is a not straight forward task.

5G and Time Sensitive Networking (TSN) are two communication technologies that gained a lot of interest in industrial communication systems over the

last few years. TSN represents a wired Ethernet based technology defined by the IEEE 802.1 TSN task group providing a set of standards for precise clock synchronization and real-time traffic treatment which is seen as an enabler for time-critical communications [11, 12]. If mobility requirements apply, reliable wireless communication is required in complement to the wired communication technologies, and due to the scarcity of radio spectrum, cognitive radio extensions to unlicensed wireless technologies [3] or the use of licensed systems such as 5G is essential. With 5G, the 3GPP targets enhanced Mobile Broadband (eMBB) as well as massive Machine Type Communication (mMTC) and Ultra Reliable Low Latency Communication (URLLC) in order to fulfil the manifold demands of various applications [7]. Moreover, an integration approach of modelling the 5G system as a bridge in the TSN network has been introduced in research articles [13, 14] and specified by the 3GPP, emphasizing the function of the TSN Translator at User Equipment side and Core Network side of the 5G system.

This paper gives a brief overview over TSN stream types which are defined in IEC/IEEE 60802 and its 5G counterpart, the Quality of Service (QoS) flows, in Section 2. In contrast to many papers that analyse the delays and reliabilities of TSN over 5G [10] or the individual technologies, this paper focuses on aspects of the joint configuration of streams or flows in 5G/TSN network carried out in Section 3. Finally, the work is concluded in Section 4.

2 State of the Art

2.1 Quality of Service Classes in conjunction with Ethernet TSN and usage of TSN mechanisms of the TSN profile 60802

TSN enhances classical Ethernet with mechanisms for clock synchronization and real-time traffic handling by providing a set of several functions. Unfortunately, different implementations and standards use different variations of this functions. For example CC-Link IE TSN uses synchronization and Time Aware Shaper (TAS), but does not use preemption. PROFINET defines synchronization, preemption and TAS for 100 Mbps. In this paper the focus is put on the IEC/IEEE 60802 TSN Profile for Industrial Automation, draft 1.3. [8]. The real-time traffic consists of time-sensitive streams that are defined as a unidirectional stream of data frames which have to be delivered within a bounded time. Streams and network devices are managed in Ethernet TSN Domains. [9]

Due to IEC/IEEE 60802 still being in a draft status, the PROFINET over TSN Guideline, which attempts to anticipate IEC/IEEE 60802, is used as a reference [5]. Three different stream types are defined, which are depicted in table 1 together with common best effort traffic for comparison. The *high stream* is used for cyclic isochronous traffic where the network interface and the application are synchronized. For this stream type, a TSN end station knows when to sent a certain frame. This is called a synchronized network access. Moreover, the routes for the communication are engineered by a central Network Management Engine (NME), also called TSN Domain Management Entity (TDME) and Media Access Control (MAC) address learning is disabled. In contrast the *low stream* type is

used for non-isochronous traffic where the network interface and the application are not synchronized. For the real-time stream type there is no time awareness, but it is still cyclic traffic which has a latency requirement. Furthermore the routes are not engineered but learned through the Spanning Tree algorithm. [5]

Table 1. TSN stream types and best effort traffic adapted from PROFINET over TSN Guideline Version 1.31

Stream Type	**Cyclic /Acyclic**	**Latency Requirement**	**MAC Learning /Engineered**	**Time Awareness**
High (isochronous)	cyclic	yes	engineered	yes
Low (non-isochronous)	cyclic	yes	engineered	yes
Real-Time	cyclic	yes	learning	no
Best Effort (no stream shown for comparison)	acyclic	no	learning	no

The mapping of traffic to a specific stream and stream type is defined in the Ethernet frame header. To realise that, both the destination address and the Virtual Local Area Network (VLAN)-Tag of the frame are used. The tag includes the Priority Code Point (PCP) field in which the stream type is coded as a priority number as well as the VLAN Identifier (VID) field. The VID in combination with the destination address is used as a unique stream path selector. According to IEC/IEEE 60802 draft 1.3 there shall be a middleware with a translation table to translate between different priority types for applications not using the specified PCP or VID. This middleware can also translate to other profiles like 5G [8].

As described in the PROFINET over TSN Guideline [5], there are several mechanisms to ensure the determinism and low latency of the traffic. In order to protect a TSN domain against too much incoming external traffic, e.g. from a different TSN domain, which could block the internal streams, ingress rate limiters can be used. These function of Ethernet TSN bridges enforce a bandwidth limit for the external traffic. Moreover, a new priority and VID is assigned to the incoming traffic when entering the new TSN domain. As the traffic leaves the domain, the previously assigned priority and VID are removed.

Further interdependent mechanisms influence the traffic handling at bridges. When leaving a bridge, the streams are inserted in different queues depending on the assigned priority and are sent only when the permission for that queue is granted by a Transmission Selection Algorithm (TSA). Using only the priority queues is the most basic algorithm which is called strict-priority. However, it can be enhanced with the TAS which adds a time schedule for the different queues and requires common notion of time among the TSN domain members. For instance, the permission for the queue with *high streams* can be always given, while the permission for the *low stream* is only given for a shorter time slot of the communication cycle and the permission for the real-time streams is given for the shortest time slot. The configuration of time aware shaping in a TSN

can be generalised e.g. as a no-wait job shop scheduling problem which has been proven to be NP-hard [4]. Therefore, typically some heuristic approaches are used to schedule such communication. To prevent enlarged latencies of *high streams*, through low priority traffic extending into their time slots, guard bands are used. A guard band is a gap in the transmission as large as the largest possible Ethernet frame. This gap, serving as a buffer, is placed in front of the scheduled time slot of the *high stream*. To utilize the limited available time slots for transmissions more efficiently preemption is introduced in TSN. It is the procedure of splitting a low priority frame into parts and sending them separately. Because of preemption the guard band can be minimized to a size as small as the smallest possible Ethernet frame. Forwarding techniques reduce the latency when a transmission is send over a bridge. Usually the bridge would buffer the whole frame and forwards is afterwards. With the cut-through forwarding method, the incoming frame is directly forwarded to the output port without buffering it. Still a short delay due to the processing in the hardware is existent. In IEC/IEEE 60802 draft 1.3 a delay less than 1 µs is recommended and less than 2 µs is mandatory for a bridge with a specified bandwidth of 1 GB. Cut-through forwarding is only possible, if the port is not blocked by an other transmission.

2.2 5G QoS Flows

The 5G system architecture as depicted in Fig. 1 is described in TS 23.501 [2]. A 5G system is split into a user plane which is visualized by the larger boxes on the bottom of the illustration and a control plane depicted on top of it. A typical user data flow goes from the Data Network (DN), i.e. some external application, through the User Plane Function (UPF) and the Radio Access Network (RAN) to a User Equipment (UE) or vice versa. The control plane consists of 22 different network functions some of which may have multiple instances. A list of all instances of network functions is held in the Network Repository Function (NRF). Two of the most notable and irreplaceable control plane functions are the Access and Mobility Function (AMF) and Session Management Function (SMF) which connect the user plane to the control plane and thus one of their key responsibilities is to pass configuration parameters to the UE, RAN, and UPF, respectively. Furthermore, the Application Functions (AFs) which represent the application interface to the 5G control plane are as dynamic as the related application itself and may start or stop depending on the Operation Technology (OT) requirements. If an AF is trusted by the 5G core, the application can influence parts of the 5G network behavior such as the traffic routing. Moreover, a Network Exposure Function (NEF) exposes network capabilities and statistics to external networks and applications. Thus, an interaction between an OT application and the 5G control plane can be realized over the NEF and the AF.

Furthermore, the 3GPP describes several aspects related to the QoS of the 5G system as well as the integration of 5G with TSN [2]. The specification describes the flow-based QoS model as the finest granularity to differ between the packet flows. Each flow is identified by a unique QoS Flow ID (QFI), and within a PDU session at least one default flow needs to be established. Further QoS

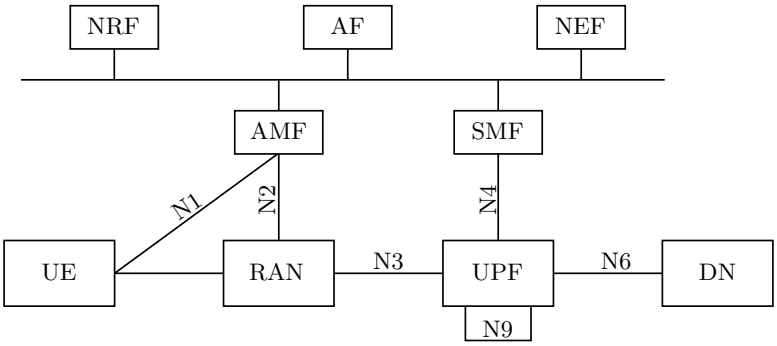

Fig. 1. 5G system architecture

flows may be configured during the PDU session establishment or dynamically assigned by the SMF of the 5G system. Packets that arrive at the 5G system require a mapping to the corresponding QoS flow and finally the radio resources as illustrated in Fig. 2. Therefore, packet detection rules are applied to filter the packets in the UPF or the UE, respectively. The filtering of the packets is supported based on IP or Ethernet headers including the IEEE 802.1Q [9] fields, and the packets mapped to a QoS flow of a PDU session are then forwarded to the RAN with an additional QFI encapsulation. The QFI is used by the Service Data Adaption Protocol (SDAP) layer in the RAN to detect the QoS profile and thus enables the mapping of the flows to the available radio resources, i.e. the Data Radio Bearers (DRBs), but the QFI is not transmitted over the radio interface.

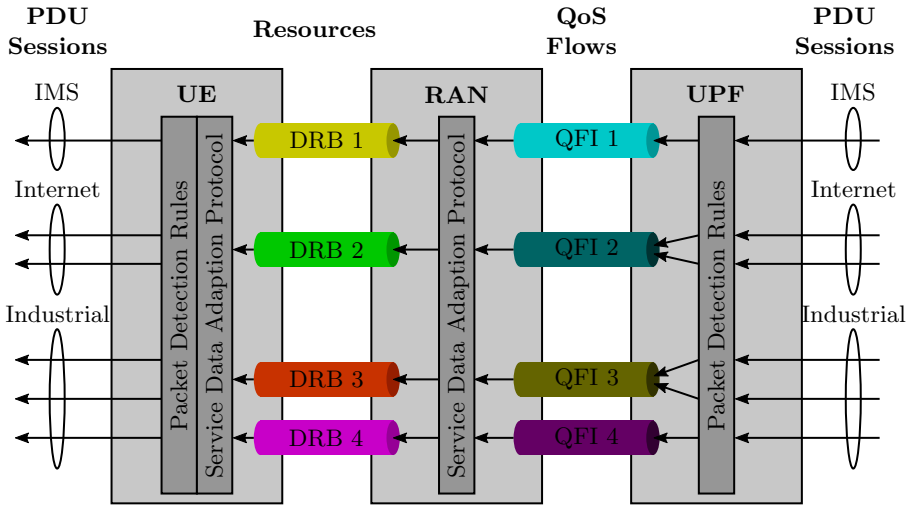

Fig. 2. QoS Flow Model

The flows can be parametrized by the 5G QoS Identifier (5QI) value that refers to various QoS characteristics and has predefined QoS mappings specified by the 3GPP. Furthermore, an Allocation and Retention Priority can be set which is examined by the RAN in order to decide which flows can be preempted in case of resource constraints. In 3GPP networks, emergency flows are assigned to the highest priority and thus they may not be preempted but any other flow may be preempted in order to enable the emergency tasks. Typical preemptable flows concern the user traffic in internet PDU sessions. For industrial purposes, the Allocation and Retention Priority might range somewhere in between and might also be used to distinguish between the importance of the different factory applications. Moreover, the Guaranteed Flow Bit Rate (GFBR) and Maximum Flow Bit Rate (MFBR) of a flow can be set with respect to the required minimal bit rate and the expected maximal bit rate of the initiated flow. Furthermore, the aggregated maximum bit rates for a UE and a PDU session and a maximum tolerable packet loss rate, specified for a QoS flow can be set. The QoS characteristics related to the parametrized 5QI value contain the resource type. This can be of type Guaranteed Bit Rate (GBR) or non-GBR as well as a priority level affecting resource scheduling among different flows and UEs. Contrary to the Allocation and Retention Priority, the priority level of the QoS profile does not abort the entire flow but may prioritize one flow over another. Additionally, a Packet Delay Budget is given to define a maximum delay that the packet forwarding in the 5G system may require. Also a maximum Packet Error Rate for the non-congested system is used in order to provoke changes on the lower layer properties such as the code rate or the modulation scheme. For GBR flows, the averaging window can be specified over which the bit rate shall be guaranteed. If the GBR flow is delay critical, the Maximum Data Burst Volume (MDBV) needs to be provided. All the above mentioned configurations are done by the SMF by signaling either the 5QI value of a predefined QoS profile or each of the parameters.

2.3 5G integration with TSN

With Release 16, the 3rd Generation Partnership Project (3GPP) highlights the integration of 5G in time sensitive communications and defines an architecture for 5G as a TSN bridge in TS 23.501 [2]. The specification is not limited to a specific deployment such as Ethernet TSN. For TSN over 5G, the 5G system is encapsulated, by two TSN-Translator functions on the device side (DS-TT) and network side (NW-TT) in order to enable the 5G integration as shown in Fig. 3. The term TSN system as used in the illustration refers to any TSN network beginning from the smallest possible unit, i.e. a single TSN host. Only one PDU session between the DS-TT and the UPF can be established for a 5G TSN bridge but due to the N9 interface that can connect multiple UPFs, a UE with multiple DS-TTs can be part of multiple logical TSN bridges. The 5G QoS flows for TSN integration are delay-ciritcal GBR flows.

Inside the 5G control plane, a TSN application function is used to connect with the TSN network controller. As mentioned previously, the AF can affect the

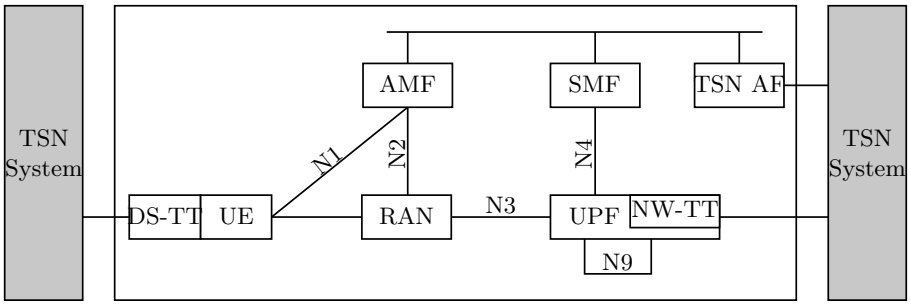

Fig. 3. 5G Bridge Architecture

configuration of the 5G network. In order to obtain the behavior of a transparent bridge as suggested in [10, 13–15], the TSN AF needs to hold information about the delays inside the 5G system and needs to provide forwarding rules to forward layer 2 Ethernet TSN frames over the 5G system operating on layer 3. The standard mainly deals with time synchronization aspects of the 5G integration and specifies that the 5G system appears as a slave to the master clock of the TSN system. Incoming packets get an ingress timestamp by the TSN translator function, and an egress timestamp is created before a packet leaves the 5G system. Moreover, the translator functions shall support hold & forward buffering mechanisms to reduce the jitter of deterministic TSN flows.

TS 23.501 [2] and TS 23.503 [1] further specify details about the management and configuration of the 5G system as a TSN bridge. The interface between the TSN AF and the Centralized Network Configuration (CNC) of the TSN system is used to exchange information about the TSN configuration. For instance, a-priori information about the traffic characteristics such as the periodicity or the frame sizes can be passed from the CNC to the TSN AF. If the characteristics are unknown, the TSN AF can calculate traffic patterns. However, in both cases the TSN AF is required to send Time Sensitive Communication Assistance Information (TSCAI) to the SMF in order to configure the 5G system. In particular, the RAN benefits from the TSCAI which allow semi-persistent scheduling or configured grant transmissions.

3 Co-configuration aspects

Some architectural requirements for self-configuration of the network which are not finally standardized yet or out of scope of the standardization emerge from the specification of the above described technologies. With respect to the joint co-configuration of 5G and TSN, particularly, the connection between the 5G control plane in form of the TSN AF and the CNC of the TSN system is of importance. Since the TSN system is the superior system in the architecture, the CNC is expected to initiate the configuration. However, a fully automatic configuration requires further development and thus the following discussion will

highlight parameters that are of interest rather than the automatic exchange of information. At first, the physical layer configuration of the 5G system has to be considered. Although it is out of scope of this paper, it has to be mentioned that the bandwidth of the 5G system, the duplexing mode and the numerology, e.g. the granularity in time and frequency, are important for the QoS of deterministic traffic and may be tweaked with respect to the desired communication type if accessible. Besides the physical layer properties of 5G which set the boundaries for the communication, many other aspects need to be taken into account in order to map the 5G QoS flows to the TSN streams.

Table 2. Parameter mapping of 5G system to TSN stream

TSN	**5G**	**Description**
Frame Size	GFBR	QoS flow guarantees the resources for the TSN frame
Frame Size	MDBV	No more data than the usual TSN frame has to be transmitted
Periodicity	Averaging Window	Periodicity of the TSN stream is identical to the Averaging Window in which the GBR of the QoS flow is measured

A TSN stream can be mapped to a 5G QoS flow by applying the related Packet Detection Rule at the UPF or UE which can be based on the source and destination addresses as well as the PCP and VID. However, the SDAP mapping of the QFI to the DRB and the scheduling might still lead to suboptimal transmission slots. Obviously, the QoS flow in the 5G system needs to be of type GBR unless it deals with a best effort TSN stream class. Assuming a cyclic TSN stream with constantly large frames or bursts of frames, the 5G QoS flow further needs to guarantee the GFBR within an averaging window which equals the periodicity of the TSN stream. Moreover, since bursts of one or more packets are expected, the MDBV needs to be as large as the amount of data which has to be transmitted within the averaging window as summarized in Table 2. But what will the 5G system make out of this configuration? The GFBR guarantees that enough resources on the DRB are allocated for the transmission of the flow, and the MDBV causes a compact allocation of the resources instead of spread time slots. Thus, all packets or packet fragments of the flow will be transmitted nearly at the same time, but however, this mechanism still leaves plenty of space for decisions to the scheduler of the 5G system which presumably introduces an increased jitter. Nonetheless, the jitter can be reduced by the buffer & hold mechanisms of the 5G TSN translator functions which leads to a trade-off between jitter on the one hand and delay on the other hand with both being undesirable for the TSN stream types *high, low, or real-time*. An easy but inefficient approach might be to reserve more resources than actually required by increasing the GFBR or fractioning the averaging window at the expense

of blocked resources. Moreover, semi-persistent scheduling or configured grants can be exploited in order to reduce the jitter. However, the complexity of the scheduling which is already an NP-hard optimization problem for the scheduled TSN traffic further increases with this approach. Nevertheless, a 5G system will likely reduce the total amount of switches between two TSN end stations.

4 Conclusion

This paper summarized essential QoS related mechanisms of TSN and 5G as a state-of-the-art review of relevant specifications. Moreover, a direction for the joint configuration of 5G and TSN was carried out showing that a significant amount of considerations has to be taken into account. Parameters of both communication domains have been mapped to each other to provide an impression of the ongoing challenges in the joint configuration of heterogenous networks combining 5G and TSN.

In the future, the considerations of this paper have to be implemented in order to validate the concepts and collect the first results. Moreover, the high complexity of configuration and engineering of both systems needs to be reduced. Due to the diversity of different application requirements in modern as well as future factories, it is questionable if the prioritization among the different types is feasible. It has to be studied if a coarser differentiation and planning of not all but the highest priority is sufficient in order to reduce the complexity and ensure the QoS especially for the streams of the highest importance.

5 An author's note on 5G and TSN terminologies

Although both 5G and TSN are communication technologies for industrial real-time communication, the domains are separated enough to do not share a common terminology. The authors of this paper stumbled over several terms and acronyms that have a different meaning in the domains of 5G and TSN and want to share this experience in order to avoid future misunderstanding. For instance, both technologies utilize preemption, but while TSN preempts the transmission of a single Ethernet frame, 5G preempts PDU sessions and QoS flows based on the ARP which does relate to the Allocation and Retention Priority instead of the widely known Address Resolution Protocol.

References

1. 3GPP: Policy and charging control framework for the 5G System (5GS). Tech. Rep. TS 23.503, 3GPP (2021)
2. 3GPP: System architecture for the 5G System (5GS). Tech. Rep. TS 23.501, 3GPP (2021)
3. Chiwewe, T.M., Mbuya, C.F., Hancke, G.P.: Using cognitive radio for interference-resistant industrial wireless sensor networks: An overview. IEEE Transactions on Industrial Informatics **11**(6), 1466–1481 (2015). https://doi.org/10.1109/TII.2015.2491267

4. Dürr, F., Nayak, N.G.: No-wait packet scheduling for ieee time-sensitive networks (tsn). In: Proceedings of the 24th International Conference on Real-Time Networks and Systems. p. 203–212 (2016). https://doi.org/10.1145/2997465.2997494
5. Friesen, A., Schriegel, S., Biendarra, A., Gamper, S.: PROFINET over TSN Guideline Version 1.31. In: Profibus International (PI) (2021)
6. Gaj, P., Scanzio, S., Wisniewski, L.: Guest editorial: Heterogeneous industrial networks of the current and next-generation factories. IEEE Transactions on Industrial Informatics **16**(8), 5539–5542 (2020). https://doi.org/10.1109/TII.2020.2976796
7. Ghosh, A., Maeder, A., Baker, M., Chandramouli, D.: 5g evolution: A view on 5g cellular technology beyond 3gpp release 15. IEEE Access **7**, 127639–127651 (2019). https://doi.org/10.1109/ACCESS.2019.2939938
8. IEC/IEEE: Time-sensitive networking profile for industrial automation. IEC/IEEE 60802, Draft 1.3 pp. 1–104 (2021)
9. IEEE: IEEE standard for local and metropolitan area network–bridges and bridged networks. IEEE Std 802.1Q-Rev Draft 1.0 (Revision of IEEE Std 802.1Q-2018) (2021)
10. Larrañaga, A., Lucas-Estañ, M.C., Martinez, I., Val, I., Gozalvez, J.: Analysis of 5g-tsn integration to support industry 4.0. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). vol. 1, pp. 1111–1114 (2020). https://doi.org/10.1109/ETFA46521.2020.9212141
11. Leßmann, G., Biendarra, A., Schriegel, S.: Vergleich von Ethernet TSN-Nutzungskonzepten. In: 2020 10th Annual Colloquium Communication in Automation (KommA2020) (2020)
12. Lo Bello, L., Steiner, W.: A perspective on ieee time-sensitive networking for industrial communication and automation systems. Proceedings of the IEEE **107**(6), 1094–1120 (2019). https://doi.org/10.1109/JPROC.2019.2905334
13. Mannweiler, C., Gajic, B., Rost, P., Ganesan, R.S., Markwart, C., Halfmann, R., Gebert, J., Wich, A.: Reliable and deterministic mobile communications for industry 4.0: Key challenges and solutions for the integration of the 3gpp 5g system with ieee. In: Mobile Communication - Technologies and Applications; 24. ITG-Symposium. pp. 1–6 (2019)
14. Neumann, A., Wisniewski, L., Ganesan, R.S., Rost, P., Jasperneite, J.: Towards integration of industrial ethernet with 5g mobile networks. In: 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS). pp. 1–4 (2018). https://doi.org/10.1109/WFCS.2018.8402373
15. Neumann, A., Wisniewski, L., Musiol, T., Mannweiler, C., Gajic, B., Ganesan, R.S., Rost, P.: Abstraction models for 5g mobile networks integration into industrial networks and their evaluation. In: Jahreskolloquium Kommunikation in der Automation - KommA 2018. Lemgo, Germany (2018)
16. Wollschlaeger, M., Sauter, T., Jasperneite, J.: The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. IEEE Industrial Electronics Magazine **11**(1), 17–27 (2017). https://doi.org/10.1109/MIE.2017.2649104

http://dx.doi.org/10.25673/39589

Ethernet-APL Bandbreitenerweiterung für industrielle Anwendungen

Harald Müller / Benedikt Spielmann, Jörg Hähniche

Endress+Hauser
Temperature and System Products / Digital Solutions
Obere Wank 1 / Christoph Merian-Ring 4
87484 Nesselwang / 4153 Reinach (CH)
harald.mueller@endress.com
benedikt.spielmann@endress.com
joerg.haehniche@endress.com

Abstract: Ethernet Technologie ist in Prozessanlagen nur in den höheren Ebenen der Automatisierungspyramide im Einsatz, kaum in der Feldebene. Dabei bietet Ethernet viele Vorteile, die vor allem im Zuge der Digitalisierung und Industrie 4.0 genutzt werden könnten. Mit Ethernet-APL liegt nun eine Lösung vor, die den Einsatz von Ethernet mit 10 Mbit/s in der Prozessautomatisierung auch in einer rauen und explosionsgefährdeten Umgebung möglich macht. Auch wenn 10 Mbit/s gegenüber den bisherigen Feldbuslösungen ein Quantensprung in der verfügbaren Übertragungsbandbreite darstellen, zeigt sich in heutigen Betrachtungen der zu erwartenden Netzwerklasten schon bald auch die 10 Mbit/s als ein Engpass in der verfügbaren Bandbreite. In IEEE 802.3 existieren bereits Standards für 2-Leiter Ethernet mit 100 Mbit/s, 1Gbit/s und mehr. Die dort festgelegten maximalen Kabeldistanzen sind allerdings für Anwendungsfälle im Automobilbereich ausgelegt (max. 40m) und genügen somit nicht den Anforderungen anderer Industrien. Für den Nachweis der Realisierbarkeit einer Bandbreitenerweiterung auf 100 Mbit/s bei Beibehaltung der verwendeten Kabelstruktur wurden von Endress+Hauser in Kooperation mit der Fachhochschule Nordwestschweiz diverse Machbarkeitsstudien durchgeführt. Nach einer kurzen Einführung der Notwendigkeiten für eine Bandbreiten-Erweiterung bei Ethernet-APL sollen die Ergebnisse der 100 Mbit/s Untersuchungen im Detail vorgestellt und ein Ausblick auf die weiteren Aktivitäten gezeigt werden.

1 Einführung Ethernet-APL

1.1 Warum es eine neue Technologie im Feld der Prozessautomation benötigt

Die Digitalisierung wird zunehmend auch in der Prozessindustrie vorangetrieben. Prozessanlagen sind danach bestrebt, Produkte mit immer höherer Effizienz und steigender Qualität zu produzieren, Time to Market und Kosten zu reduzieren. Die digitale Transformation kann bei diesen Bestrebungen unterstützen. Neue digitale Technologien ermöglichen die Vernetzung aller Bestandteile einer Prozessanlage, wodurch Daten anlagenweit zentral erfasst, konsolidiert und ausgewertet werden können. Daten, das Gold des digitalen Zeitalters, sind die Basis für eine fortlaufende Prozessoptimierung und Effizienzsteigerung. Die Geräte im Feld der Prozessanlagen sind längst smart genug, detaillierte Informationen über sich selbst und über Prozesse zu liefern. Basierend auf diesen Daten können digitale Services wie beispielsweise Erfassung der installierten Basis, detaillierte Gerätediagnoseinformationen oder vorbeugende Wartungen der Instrumentierung bereitgestellt werden. Problematisch heute ist einzig der fehlende oder der aufwändige Zugang zu diesen Daten. Die aktuell etablierten Technologien im Feld von Prozessanlagen unterstützen zwar bereits einen Fernzugriff auf die Feldebene, für datengetriebene Use Cases ist die verfügbare Bandbreite allerdings nicht ausreichend. Erschwerend kommt hinzu, dass durch verschiedene Technologien entlang der Automatisierungspyramide zusätzliche Hardware, Konvertierung von Protokollen benötigt werden, was die Komplexität des Anlagendesigns erhöht und keine transparente Kommunikation von der Prozessebene bis zum Feldgerät zur Verfügung stellt.

Auf den oberen Ebenden der Automatisierungspyramide ist Ethernet Technologie als Standard für die Datenübertragung etabliert. In Industrien wie der Fabrik- oder Gebäudeautomation wird Ethernet-basierte Kommunikation bereits großflächig eingesetzt, bis zur Instrumentierungsebene. In der Prozessautomation konnte die Ethernet Technologie bisher kaum in der Feldebene zum Einsatz kommen. Gründe dafür liegen in den bestehenden Ethernet Spezifikationen, die den hohen Anforderungen der Prozessindustrie bisher nicht genügen.

1.2 Ethernet-APL

Mit der Einführung des Advanced Physical Layers für Ethernet, kurz Ethernet-APL, sind die Hindernisse für den Einsatz von Ethernet-fähigen Instrumenten, selbst in explosionsgefährdeten Bereichen einer Anlage beseitigt. Ethernet-APL ist ein eigensicherer Physical Layer, ausgelegt für 2-adrige Kabel, für den Einsatz in allen Anwendungen der Prozessautomatisierung. Zu den Charakteristiken von Ethernet-APL gehören eine hohe Kommunikationsgeschwindigkeit, die Möglichkeit der Installation in explosionsgefährdeten Bereichen, die Stromversorgung von Feldinstrumentierung und die Datenübertragung über ein 2-Leiter Kabel sowie die Möglichkeit zur Überbrückung langer Kabeldistanzen.

Die führenden Standardorganisationen FieldComm Group, ODVA, OPC Foundation und PROFIBUS & PROFINET International sowie 12 wichtige Projektpartner aus der Industrie haben in den letzten Jahren im Rahmen des „APL Projekts“ erfolgreich zusammengearbeitet, um diesen neuen Physical Layer für Ethernet Technologie zu entwickeln. Mit der Veröffentlichung der Spezifikationen, Engineering Guidelines und Konformitätstestplänen im Juni 2021, können Endanwender nun Komponenten von führenden Anbietern erwarten – erste Produkte sind bereits offiziell verfügbar.

Ethernet-APL erweitert die Single Pair Ethernet Technologie aus IEEE 802.3cg-2019 (10BASE-T1L) [1], um wichtige Eigenschaften, damit ein Einsatz in Prozessanlagen ermöglicht wird. Die elektrischen Parameter, die ein Ethernet-APL Gerät erfüllen muss, um den eigensicheren Zündschutz zu gewährleisten, sind in der technischen Spezifikation IEC TS 60079-47 definiert [2]. Dieses Konzept wird als 2-WISE bezeichnet (2-Wire Intrinsically Safe Ethernet) und basiert auf dem Fieldbus Intrinsically Safe Concept (FISCO). Durch die Definition von Port Profilen in der APL Port Profile Spezifikation [3] werden funktionale und elektrische Anforderungen mit mehreren Leistungsstufen festgelegt. Dadurch sind unterschiedliche Topologien in APL Netzwerken möglich, wie beispielsweise das weit verbreitete Trunk-and-Spur Konzept mit Kabellängen bis zu 1000m auf den Trunk und bis zu 200m auf dem Spur. Die APL Port Profile Spezifikation beinhaltet zusätzlich Installationsregeln, wie zum Beispiel zugelassene Kabel, Verbindungstechnologie, Schirmung und Erdung. Das Fieldbus Cable Type A nach IEC 61158-2 ist das bevorzugte Kabel für APL Segmente, da es alle notwendigen Kabelanforderungen gemäß 2-WISE für Eigensicherheit erfüllt und auch die genannten Kabeldistanzen unterstützt. Für die Verbindung von APL Komponenten werden in der Spezifikation Klemmverbindungen und M8 / M12 Steckverbinder definiert. Die Ethernet-APL Engineering Guideline unterstützt Endanwender bei dieser neuen Technologie von der Planung bis zur Installation, inkl. Best Practices.

Mit Ethernet-APL wurde ein einheitlicher Physical Layer definiert, der alle im ISO/OSI-Modell darüber liegenden Schichten unterstützt. Damit kann auch jedes bekannte Netzwerkprotokoll und Industrial Ethernet Protokoll, wie z.B. PROFINET auf Ethernet-APL implementiert werden. Durch die Konformitätstests bei den oben genannten Standardisierungsorganisationen wird die Interoperabilität der APL Produkte sichergestellt.

Durch Tests erster APL Geräte in der Praxis wurde der Nutzen der Technologie demonstriert. Mit Ethernet-APL ergeben sich Vorteile entlang aller Lebenszyklusphasen einer Anlage: Engineering Aufwände werden minimiert und weniger komplex. Die Installation von APL Komponenten ist über einfachen Klemmanschluss und Polaritätsunabhängigkeit einfach und fehlersicher. Die Integration von Ethernet-basierten Komponenten ins Prozessleitsystem ist halb-automatisiert und wird je nach Industrial Ethernet Protokoll durch Profile weiter vereinfacht. Die hohe Datenübertragung ermöglicht schnelle Parameter Up- und Downloads. Fehlerlokalisierung im Ethernet-Netzwerk oder in den Geräten ist durch etablierte Tools einfach und auch per Fernzugriff möglich. Dank der durchgängigen Ethernet Technologie wird der Zugriff auf die Daten im Feld vereinfacht, sodass datengetriebene Applikationen, digitale Services im Rahmen der NAMUR Open Architecture [4] ermöglicht werden.

Ethernet-APL verbindet die Vorteile der einfachen, robusten und bekannten 2-Leiter Technologie mit all den Vorteilen der Ethernet Technologie. Durch die hohe Performance und den transparenten Zugriff auf die Daten im Feld wird die digitale Transformation in Prozessanlagen begünstigt und vorangetrieben.

2 Bandbreitenerweiterung 100Mbit/s für Ethernet-APL

Ethernet-APL mit einer Datenübertragung von 10Mbit/s, auf Basis des IEEE Standard: 802.3cg 10BASE-T1L, wird für viele Anwendungsbereiche der Prozessindustrie ausreichen, um datengetriebene Applikationen im Rahmen der Digitalisierung umsetzen zu können. Es gibt jedoch Anwendungen, die noch höhere Datenraten benötigen. Aus diesem Grund beschäftigt sich das APL Projekt mit der Bandbreitenerweiterung

auf 100Mbit/s. Die Umsetzung erfolgt im Rahmen eines neuen IEEE 802.3 Standards für eine 2-Leiter Ethernet Lösung mit 100Mbit/s für lange Kabeldistanzen.

Die technischen Anforderungen an die Architektur leiten sich aus der 10 Mbit/s Ethernet-APL Lösung ab, wobei lediglich die die maximalen Kabellängen aufgrund der höheren Kabeldämpfung bei entsprechender Übertrags Frequenz eingeschränkt sind. Die maximal möglichen Kabellängen werden in der IEEE 802.3 Study Group für das Link-Segment entsprechend spezifiziert. Zurzeit werden hier Kabellängen im Bereich von 300…500 m diskutiert und es zeigt sich, dass mit diesen Kabellängen umfassende Anwendungen in den verschiedenen Industrien abgedeckt werden können. Damit die Kompatibilität mit den anderen IEEE Single Pair Ethernet Standards gegeben ist und insbesondere die Rückwärtskompatibilität mit dem 802.3cg Standard 10BASE-T1L gegeben ist, wird die im IEEE 802.3cg Standard etablierte Auto Negotiation gefordert. Somit können Ethernet-APL Netzwerke Geräte zukünftig mit 10Mbis/s und 100 Mbit/s kommunizieren. Die Festlegung der Übertragungsrate erfolgt in der Start-up Phase mit dem Ziel die max. mögliche Rate entsprechend automatisch auszuwählen, damit eine einfache Inbetriebnahme des gesamten Netzwerks möglich ist.

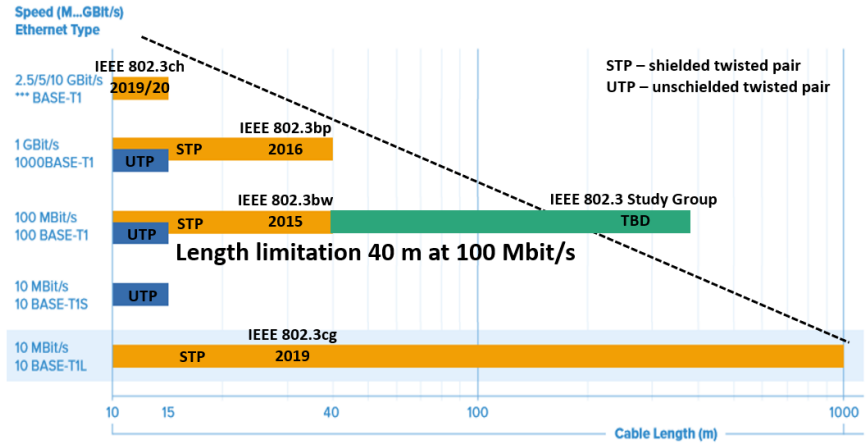

Abbildung 1: Reichweiten und Übertragungsgeschwindigkeiten der aktuellen IEEE802.3 Single Pair Ethernet Standards. Der grüne Balken zeigt die 100 Mbit/s Bandbreitenerweiterung im Rahmen der IEEE 802.3 Study Group

In IEEE 802.3 existieren bereits Standards für 2-Leiter Ethernet mit 100 Mbit/s, 1Gbit/s und mehr, siehe Abbildung 1. Die dort festgelegten maximalen Kabeldistanzen sind allerdings für Anwendungsfälle im Automobilbereich ausgelegt (max. 40m) und genügen somit nicht den industriellen Anforderungen.

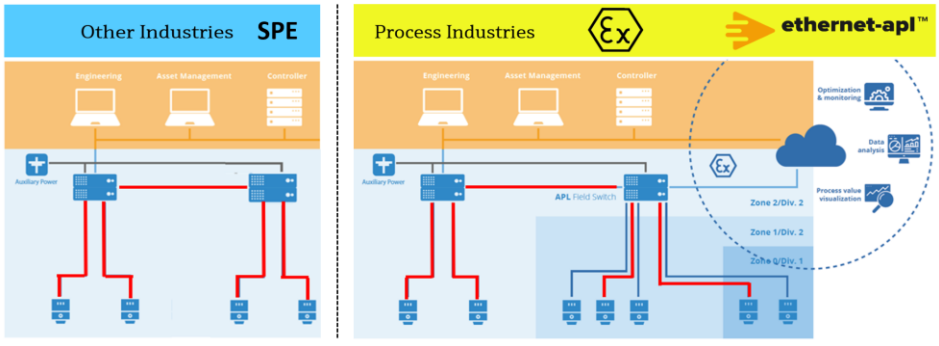

Abbildung 2: In Rot werden mögliche 100 Mbit/s Single-pair-Ethernet Punkt-zu-Punkt Verbindungen dargestellt (Switch-Switch, Switch-Feldgerät). Hierbei wird von der gleichen Architektur wie bei der 10 Mbit/s Lösung ausgegangen.

Im Rahmen des APL Projektes zur Bandbreitenerweiterung auf 100 Mbit/s wurde ein Realisierungsnachweisprojekt von Endress+Hauser in Kooperation mit der Fachhochschule Nordwestschweiz und diversen internen Machbarkeitsstudien bei Endress+Hauser durchgeführt, die gezeigt haben, dass mit einem Fieldbus Cable Type A Kabellängen im Bereich von 300…500 m erreicht werden können.

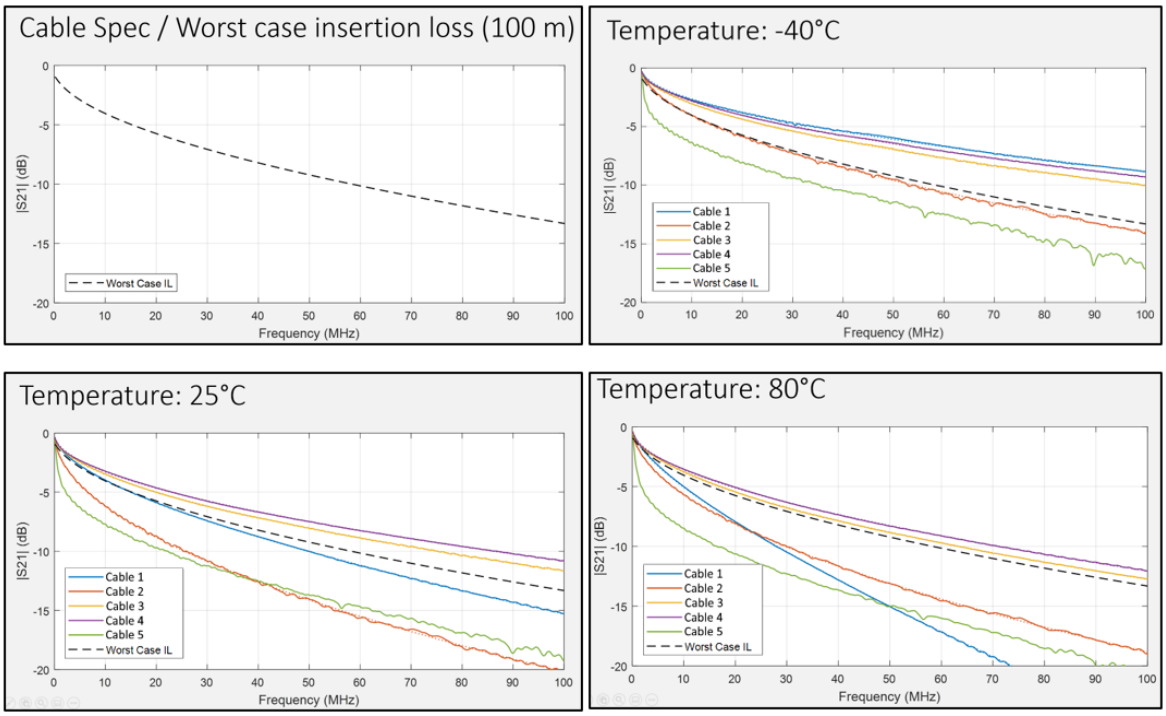

Abbildung 3: Einfügedämpfung gemessen bei unterschiedlichen Temperaturen für diverse Fieldbus Cable Type A

Abbildung 3 zeigt Messergebnisse für die Messung der Einfügedämpfung (Insertion Loss) bei verschiedenen geschirmten twisted-pair Kabeln. Die S21 Kabelübertragungsparameter werden bei verschiedenen Temperaturen im Frequenzbereich bis 100 Mhz dargestellt. Die Kurven zeigen, dass mit einem Fieldbus Cable Type A 100 Mbit/s übertragen werden kann. Die maximal mögliche Kabellänge wird hierbei durch den internen Aufbau des PHY´s und des Signal-to-Noise Abstands beeinflusst.

Aufbauend auf den Kabelmessungen wurde ein Matlab/Simulink worst-case Modell von PHY-Kabel-PHY Strukturen erstellt um theoretisch die Realisierbarkeit einer 100Mbit/s Kommunikation über 200 m (Spurverbindung) bzw. 300…500 m (Trunkverbindung) mit dem Fieldbus Cable Type A nachzuweisen.

Es wurde dabei von einer Signalamplitude von 2 Volt Peak-to-Peak Trunk und 1 Volt Peak-to-Peak Spur ausgegangen. Im Rahmen der theoretischen Untersuchungen mit den simulierten Störquelle zeigten sich die Kodierungsvarianten 3B2T (3 Bit codiert auf jeweils 2 Symbole mit 3 möglichen Zuständen) und 4B3T (4 Bit codiert auf jeweils 3 Symbole mit 3 möglichen Zuständen) den höchsten Signal-zu-Rausch Abstand.

In Abbildung 4 wird der simulierte Verlauf sowie die Verteilung des Signals bei Verwendung eines Fieldbus Cable Type A mit Störsignal (weißen Rauschen) gezeigt. Hierbei wurden optimale Filtereinstellungen verwendet, welche später in der Blind Training Phase ermittelt werden. Die zeitliche Darstellung (links) sowie die Werteverteilung (rechts) wird ohne (grün) und mit (rot) simuliertem Sinus Störsignal dargestellt. Das Störsignal wurde als Sinus mit gleichbleibender Amplitude und zeitlich steigender Frequenz simuliert.

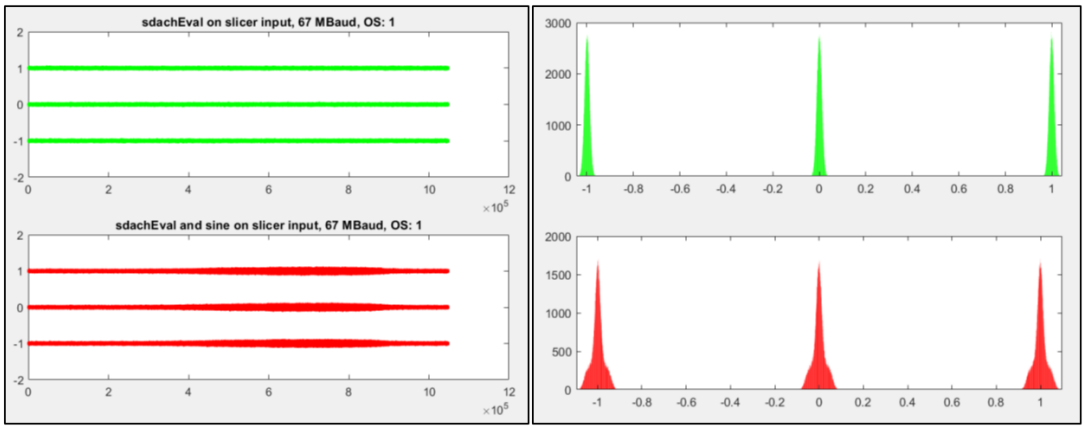

Abbildung 4: Zeitlicher Verlauf und Verteilung von aufbereiteten Signalen vor der Symbolzuordnung, ohne (grün) und mit (rot) simuliertem Störsignal mit 200m Worstcase Fieldbus Cyble Type A

Abbildung 5 zeigt den in Simulink simulierten zeitlichen Verlauf mehrerer Parameter innerhalb eines PHYs bei einem Verbindungsaufbau zweier PHYs über ein 200m worst-case Fieldbus Cable Type A. Dargestellt wird das analog zu digital gewandelte Eingangssignal (orange), das digital aufbereiteten Signal (gelb), der Signal-zu-Rausch Abstand (grün) und der Symbolzuordnungsfehler (rot). Bei der Simulation wurde davon ausgegangen, dass während des Trainings der Signalaufbereitung mit zwei Signalzuständen gearbeitet wird. Nach festgelegter Zeit wurde schließlich auf die PAM-3 Codierung umgeschaltet. Das aufbereitete Signal zeigt eine Signalauflösung mit ausreichender Genauigkeit bereits nach wenigen Millisekunden.

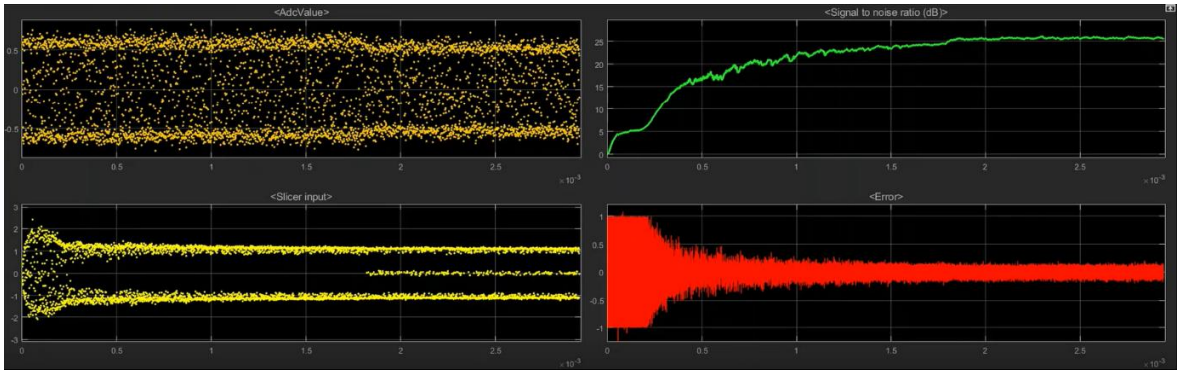

Abbildung 5: Simuliertes Blind Training der adaptiven Filter des PHY Models mit Rohwerten (orange), aufbereiteten Werten (gelb), SNR (grün) und Symbolerkennungsfehler (rot)

Auf Grundlage des simulierten PHY-Kabel-PHY Modells konnte schließlich ein Evaluationsboard mit einem 10/100Mbit/s PHY Prototyp auf FPGA-Basis realisiert werden. Der PHY Prototyp unterstützt dabei sowohl 10Mbit/s Kommunikation über 1000m als auch 100Mbit/s Kommunikation über ein 300m Fieldbus Cable Type A bei einer Signalamplitude von 2Vpp.

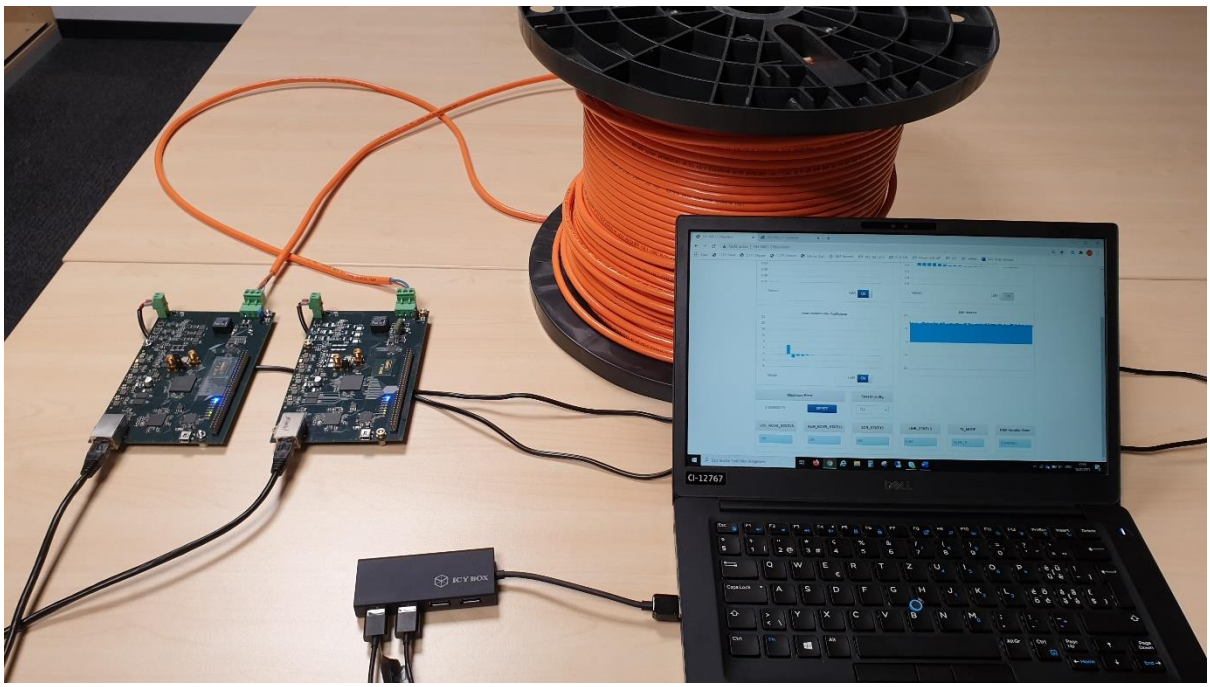

Abbildung 6: Testaufbau mit Evaluationsboards und Fieldbus Cable Type A

Der Physical Medium Attachment Layer (PMA) des Prototyps verwendet eine in Hardware realisierte Hybrid Schaltung sowie einen digitalen Echo Canceller, um das empfangene Echo auszublenden, einen Decision Feedback Equalizer zur Aufbereitung der empfangenen Symbole sowie einen Mueller-Müller Algorithmus zur Taktsynchronisation.

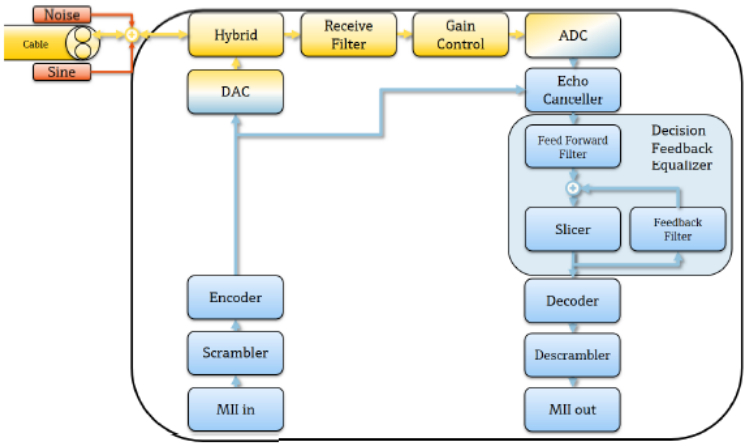

Abbildung 7: Grundstruktur des Transceivers der Simulation

Abbildung 7 zeigt auf Blockebene den Aufbau des 10/100Mbit/s PHY Prototyp auf FPGA-Basis. Die digitalen Bausteine werden als blaue Rechtecke dargestellt, während blaue Pfeile digitale Übergabeparameter anzeigen. Alle blau dargestellten Objekte sollen später auf dem FPGA realisiert werden. Gelbe Rechtecke stellen analoge Schaltungen dar, während gelbe Pfeile Spannungen anzeigen. Der Digital-Analog-Wandler (DAC) sowie der Analog-Digital-Wandler sind als gelb-blaue Rechtecke dargestellt, da diese Komponenten den Übergang zwischen dem digitalen FPGA und dem analogen Frontend bilden. Störungen, die vom Kabel ausgehen, sind dunkelorange dargestellt.

Der Verbindungsaufbau wird mittels Blind Training durchgeführt, wobei beiden PHYs außer der Kodierungsvariante nichts über die gesendeten Signale des jeweiligen Gegenübers bekannt ist. Der Physical Coding Sublayer (PCS) wurde für beide Übertragungsgeschwindigkeiten auf Grundlage des Standards IEEE 802.3cg für 10BASE-T1L aufgebaut. Für die Untersuchung des Prototyps verfügt das FPGA außerdem über ein System-on-Chip mit Webserver, über den sowohl Systemparameter ausgelesen werden als auch Konfigurationen während der Laufzeit eingestellt werden können. Zudem können in Echtzeit jeweils zwei Zwischenparameter aus dem PMA sowie dem PCS auf eine Stiftleiste ausgegeben werden, um die hochfrequenten Berechnungsprozesse analysieren zu können. Das Evaluationsboard ist als 3-Port Ethernet-Switch ausgeführt, wobei jeweils ein Port mit dem 10/100Mbit/s PHY Prototypen, mit einer 100BASE-TX Ethernet Schnittstelle und dem Webserver verbunden ist. Das Evaluationsboard verfügt über zwei Bestückungsvarianten, wobei die eine einen APL Power Source Port mit integrierten Ex-Begrenzungen und die andere einen APL Power Load Port darstellt. So kann die Kommunikation bei laufender Energieübertragung analysiert werden.

Abbildung 8 zeigt einen Ausschnitt aus dem Webserver des Evaluationsboards. Zu sehen sind die aktuellen Filterkoeffizienten des Echo Cancellers sowie des Feed Forward und Feedback Filters, die den Decision Feedback Equalizer bilden. Ausserdem ist die aktuelle Signal-zu-Rausch (SNR) Verhältnisreserve zu sehen, welche den Abstand zur mindestens benötigten SNR darstellt.

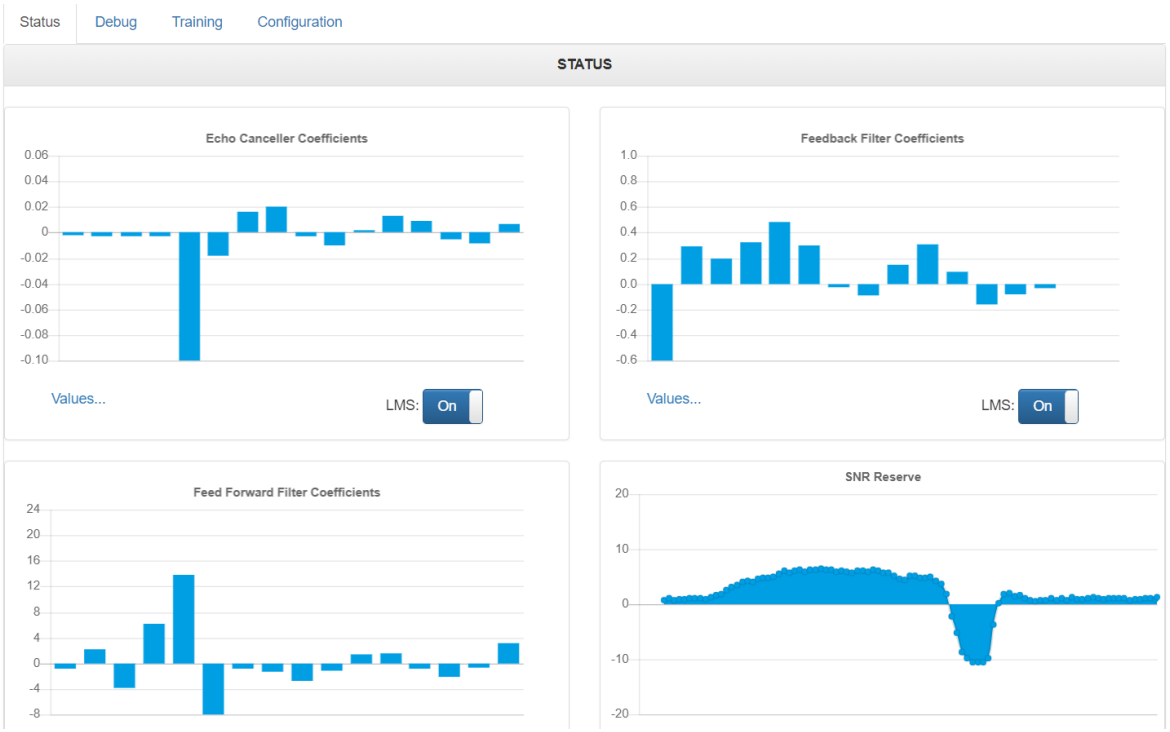

Abbildung 8: Screenshot aus dem Webserver des 10/100Mbit/s APL Evaluationsboards

Abbildung 9: PCB des 10/100Mbit/s APL Evaluationsboards

Abbildung 9 zeigt die zwei Bestückungsvarianten der Evaluationsboards für einen APL Power Source und einen APL Power Load Port.

Das 10/100Mbit/s APL Evaluationsboard wurde im Rahmen der Phase 2 vom Ethernet APL – Projekt von Endresss + Hauser aufgebaut, um die technische Machbarkeit inkl. der geforderten Rückwärtskompatibilität zur Phase 1 – 10 Mbit/s (10BASE-T1L) nachzuweisen. Für die Standardisierung auf IEEE 802.3 werden die Ergebnisse ebenfalls in der aktuellen Study Group und später in der Task Force Group als Nachweis der Machbarkeit verwendet. Die Untersuchungen haben gezeigt, dass über 2 Leiter auch eine Übertragung mit 100 Mbit/s über eine lange Strecke mit dem Feldbus Kabel Typ A möglich ist. 200m Spur-Länge wären kein Problem. In den Untersuchungen war eine fehlerfreie Übertragung über mehr als 300m möglich.

3 Literaturverzeichnis

[1] IEEE Computer Society
IEEE Standard for Ethernet - Amendment 5: Physical Layer Specifications and Management Parameters for 10 Mb/s Operation and Associated Power Delivery over a Single Balanced Pair of Conductors, IEEE 802.3cg-2019, 2019.

[2] IEC- International Electrotechnical Commission
IEC TS 60079-47 Explosive atmospheres – Part 47: Equipment protection by 2-wire intrinsically safe Ethernet concept (2-WISE), 2021.

[3] APL Project
Ethernet-APL Port Profile Specification, Version 1.0, June 2021

[4] NAMUR – Interessengemeinschaft Automatisierungstechnik der Prozessindustrie
NAMUR Empfehlung NE 175: NAMUR Open Architecture, NOA Konzept

http://dx.doi.org/10.25673/39591

Synchronization Requirements of Converged Wired and Wireless Time-Sensitive Networks

Maximilian Schüngel, Steven Dietrich, Shun-Ping Chen†, Michael Kuhn†
Bosch Rexroth AG, †Darmstadt University of Applied Sciences
maximilian.schuengel@boschrexroth.de

Abstract: Following the trend of Industry 4.0, future factory automation will introduce an increasing number of mobile applications to enable the desired highly flexible production. Novel challenges arise for the underlying communication system. This includes the coexistence of real-time and non-real-time communication while supporting mobility. A promising solution is the integration of Time-Sensitive Networking (TSN) and the fifth generation cellular network technology (5G). A key aspect of real-time communication is the establishment of a common sense of time for the aspired fast and precise applications. This work covers industrial use cases of novel TSN/5G networks. An overview on related industrial use cases is given followed by a detailed description of selected uses cases and their communication requirements. This work concludes with a feasibility assessment of the selected use cases, with a strong focus on the synchronization. Possible benefits of novel TSN/5G networks are shown and current shortcomings are discussed.

1 Introduction

A current subject of the industry and academia is the unification of communication technologies to develop a single technology that is able to satisfy the diverse requirements raised by different industrial applications. The integration of Time-Sensitive Networking (TSN) and the fifth generation cellular network technology (5G) is a promising solution. It is currently discussed by the respective standardization groups IEEE and 3GPP. The 5G/TSN convergence is discussed to be transparent such that the integration of 5G into an existing TSN network is seamless. Therefore, the 5G system (5GS) introduces boundary TSN translator devices which are responsible to conceal the 5G complexity from the TSN network [5G-21].
A prerequisite for industrial applications and real-time communication is a tight synchronization in the time domain. The synchronization in TSN networks is established

through the generalized Precision Time Protocol (gPTP) according to IEEE 802.1AS. A major challenge of integrated TSN/5G networks, with respect to traditional TSN networks, is to achieve and maintain tight synchronization between end devices. This is because the convergence of TSN and 5G yields the engagement of heterogeneous synchronization mechanisms [S$^+$21] and because the deployment of radio links introduces uncertainty.
Use cases of integrated TSN/5G networks are heavily discussed and the performance of the individual technologies, i.e. TSN and 5G, is evaluated. However, there are not many studies on the actual End-to-End (E2E) performance of integrated TSN/5G networks. The use cases are often discussed against inconsistent performance indicators. This work studies use cases of factory automation, discusses their E2E requirements which are mapped to a consistent set of performance indicators, and assesses the feasibility of novel TSN/5G networks for such use cases.

2 Performance Indicators of Industrial Communication

Factory automation comprises various different applications that are generally more demanding than other classes of industrial applications. Consequently, the requirements on the underlying communication are more challenging. In order to discuss the feasibility of factory automation applications from a communication perspective, key performance indicators (KPIs) have to be defined.
For both wired and wireless industrial networks KPIs are known from the literature. As main contributor for wired industrial communication the group of IEC 61784 standards provides nine different KPIs in total. The most relevant KPIs for factory automation applications are the delivery time (or latency), the synchronization accuracy, and the real-time (RT) throughput. In scope of this work the synchronization accuracy refers to the maximum time deviation between two independent clocks, i.e. from an E2E perspective. This is different to the synchronization accuracy that describes the maximum time deviation of a clock towards the reference clock as often used in the literature.
Different to wired communication, wireless communication involves uncertainties due to the usage of a shared transmission medium. To discuss the feasibility of factory automation applications in regard to converged wired and wireless networks, i.e. TSN/5G networks, additional KPIs have to be defined in order to account for the wireless communication part. Typical KPIs known from the literature are the availability (or packet loss rate), the number of nodes, the service area, and the node mobility [5G-19, F$^+$14].
The combination of the discussed KPIs for both wired and wireless communication yields the following KPIs for TSN/5G networks:

- Latency
- Synchronization accuracy
- Payload
- Cycle time
- Packet loss rate
- Number of nodes
- Service area
- Mobility

3 Overview of Use Cases in Factory Automation

Factory automation is concerned with discrete production processes that are diverse and include various different applications itself. The manifold applications can be grouped in three different use case areas. The grouping is based on the type of application and the imposed requirements referring to the previously discussed KPIs. The following use case areas can be identified for factory automation [3GP21, 5G-19]: 1) Motion control, 2) Control-to-control, and 3) Mobile robots.

These use case areas are discussed briefly in the following section. A consolidated collection of requirements for the different use case areas and their subordinate use cases is given in Tab. 1. The provided collection of requirements was gathered from related literature and research studies.

Motion control is a specific application of closed-loop control. To control a physical process with high precision, fast control loops are required which exchange target and actual values between controller and actuators/sensors. The communication between the motion controller and both actuators and sensors follows a cyclic pattern with deterministic constraints. Motion control is the most challenging use case area in factory automation and thus imposes the most stringent requirements on the underlying communication [5G-19]. It is associated with the field level of the automation pyramid [Nof09] and relies traditionally on Industrial Ethernet (IE) or legacy fieldbus technologies. Popular motion control use cases are printing machines (control-to-device, C2D), machine tools, packaging machines, and robotic motion control [3GP21, 5G-19, Bro18, D$^+$17, F$^+$14].

Control-to-control use cases refer to applications that involve multiple machines or subsystems. Each machine or subsystem deploys a dedicated controller. Control-to-control use cases apply either to large machines which cluster individual machine functions into different subsystems or to distributed applications which includes multiple individual machines [3GP21]. Since the different machines or subsystems influence a joint process, they are required to perpetually communicate. The communication is cyclic and is exposed to deterministic constraints similar to the motion control use cases but relaxed. Control-to-control use cases naturally refer to the control level and typically rely on IE technologies. Control-to-control uses cases known from the literature are printing machines (control-to-control, C2C) and assembly lines [3GP21, 5G-19].

Mobile robots use cases involve any kind of mobile machinery, e.g. Autonomous Guided Vehicle (AGVs), that are deployed on the factory floor. Mobile robots can be viewed as individual machines that elaborate a process while having degrees of movement. In context of factory automation, mobile robots are predominantly used by means of AGVs for transportation and logistic purposes. Mobile robots implement a dedicated controller for basic functionalities and are typically controlled by a centralized guidance control system which provides missions and routes [5G-19]. Mobile robots continuously communicate with the central guidance control system. They provide their current location with a high precision on the basis which the guidance control system adapts missions and routes. The communication is again based on a cyclic

pattern with deterministic properties but shows relaxed requirements compared to the previous use case areas. Challenging use cases of mobile robots are cooperative carrying applications and video-operate remote control [3GP20, 3GP21, Bro18].

Table 1: Consolidated requirements of factory automation use cases [3GP20, 3GP21, Bro18, D$^+$17, F$^+$14, Kip01]

Use Case Area	Use Case	Latency	Synchronization Accuracy†	Payload	Cycle Time	Packet Loss Rate	Number of Nodes	Service Area	Mobility
1.	Printing Machine (C2D)	2 ms	0.25 µs– 1.25 µs	20 B	2 ms	10^{-4}	100	25×25 m	14.4 km/h– 72 km/h
	Machine Tool	0.5 ms	2 µs	50 B	0.5 ms	10^{-4}	20	15×15 m	72 km/h
	Packaging Machine	1 ms	10 µs	40 B	1 ms	10^{-4}	50	10×5 m	72 km/h
	Robotic Motion Control	0.5 ms	1 µs	50 B	0.5 ms	10^{-4}	50	50×10 m	72 km/h
2.	Printing Machine (C2C)	10 ms	0.25 µs	1 kB	10 ms	10^{-4}	10	100×30 m	stationary
	Assembly Lines	50 ms	2 µs	1 kB	50 ms	10^{-4}	10	100×30 m	stationary
3.	Cooperative Carrying	0.5 ms	500 µs	250 B – 500 B	1 ms	10^{-4}	100	1 km^2	6 km/h – 12 km/h
	Video-operated Remote Control	10 ms	5000 µs	15 kB – 150 kB	10 ms	10^{-4}	100	1 km^2	50 km/h

† end-to-end (E2E)

In the remainder of this section three particular use cases are discussed in detail. The use cases of 1) robotic motion control, 2) printing machines, and 3) cooperative carrying were selected because they impose stringent requirements on the underlying communication with special emphasis on the synchronization accuracy.

3.1 Use Case 1: Robotic Motion Control

Robotic motion control is a major topic of factory automation. Industrial robots elaborate a magnitude of relevant tasks such as assembly, machining, painting, pick and place, or welding. For such tasks, industrial robots have to support a large degree of motion using multiple axes and joints in order to handle complex processes. As discussed in Sec. 3 motion control refers to a closed-loop control process including a controller, actuators and sensors. Sensors may be placed anywhere on the robot and such also onto moving parts. In such cases a physical connection for communication is cost intensive, prone to errors, and typically lacks the support of high data rates [D$^+$17]. To overcome these issues, sensors like a high resolution camera may be connected wirelessly. In this work we additionally assume that the robot represents an enclosed system for modulartiy and flexibility. It includes a controller, a Programmable Logic Controller (PLC), and its connected field devices, i .e. sensors/actuators.
The E2E synchronization requirement of 1 µs between its field devices yields an precision of 20 µm considering a maximum movement speed of 20 m/s. This is really competitive e.g. when compared to pick and place machines which achieve a similar accuracy but typically rely on wired communication. As discussed before, the robot is assumed to be an enclosed system. It may be a subsystem of a large machine or conglomerate of machines such as an assembly line. Therefore, it is assumed that both the robot PLC and the wireless sensors are connected to the 5G RAN using dedicated

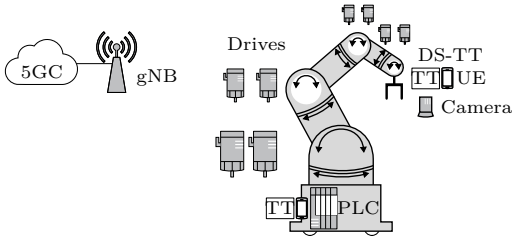

Figure 1: Robotic motion control: Closed-loop motion control that uses wireless sensors for the feedback loop.

User Equipment (UEs) and Device-Sided TSN Translators (DS-TTs). Fig. 1 illustrates an exemplary deployment for the given use case. A local PLC connects field devices via a line/ring network. Additionally, a high resolution camera, serving as exemplary wireless sensor, is mounted on the robot (on a moving segment).

3.2 Use Case 2: Printing Machine

Printing machines are large machines that are used to produce large batch sizes of e.g. newspapers or packaging. They consists of multiple subsystems which cluster different machine functions. For instance flexography or offset printing systems use dedicated printing units that are associated with dedicated colors. The superposition of the different colors yields the final image. Each printing unit represents a subsystem. Thereby printing machines become cross domain. At field level, i .e. within a subsystem, the specific machine functions can be classified as motion control use case. At control level, i.e. between different subsystems, the higher-level application represents a control-to-control use case.
A deviation of 5 µm results in a blurry image with visible artifacts. Assuming that the substrate, e.g. paper, moves with a speed of up to 20 m/s through the printing machine, the different printing units have to be synchronized within less than 0.25 µs E2E [D+17]. In return this means that the printing units have to be synchronized to the reference point with a synchronization accuracy of 0.125 µs. A more sensible substrate speed is 4 m/s which is typically used for offset or flexography printing machines [Kip01]. Such a reduced substrate speed then requires a synchronization accuracy of 0.625 µs of the individual printing units. Printing machines expose a large spatial extent of up to 100 m × 30 m and include up to 10 different subsystems [3GP21]. A subsystem itself may consist of up to 100 devices [3GP20, Bro18]. The typical distance between access points (APs)/radio dots (RDs) in factory deployments is 30 m. Based on that about 3 APs/RDs are required to provide coverage for the printing machine.
As discussed before, a printing machine clusters machine functions into different subsystems. Each subsystem implements a dedicated controller, a PLC, which is responsible to control its associated machine function. On a superordinate level, a printing machine uses a Supervisory PLC (S-PLC) to coordinate the different subsystems. Due to the stringent communication requirements, printing machines traditionally rely on

IE technologies [5G-19]. At field level, i.e. within a subsystem, and at control level, i.e. between subsystems, line networks are established for cost efficiency purposes and a reduced management effort. For redundancy the line networks are typically being closed to ring networks. Fig. 2 shows an exemplary deployment of integrated TSN/5G networks with a printing machine. At control level the wired links between the S-PLC and the PLCs of the different subsystems are replaced by wireless links. The traditional line/ring networks are transformed in a star network which are typical for wireless technologies. The S-PLC is located at a central location and is connected to the 5G Core (5GC) via a Network-Side TSN Translator (NW-TT). The different subsystems implement a UE and DS-TT each. At field level the subsystems still leverage wired links, i.e. TSN. The integration of TSN/5G networks yields many possible benefits. At control level the machine deployment becomes highly flexible with a facilitated reconfigurability. This is supported by the self containment of subsystems which still rely on wired technologies. From a practical perspective this means that subsystems can easily be added/removed, e.g. special effect colors. Deploying the S-PLC in a central location decouples the supervisory control from the actual machine functions enabling a virtualization which promises future proof and scalability [5G-21].

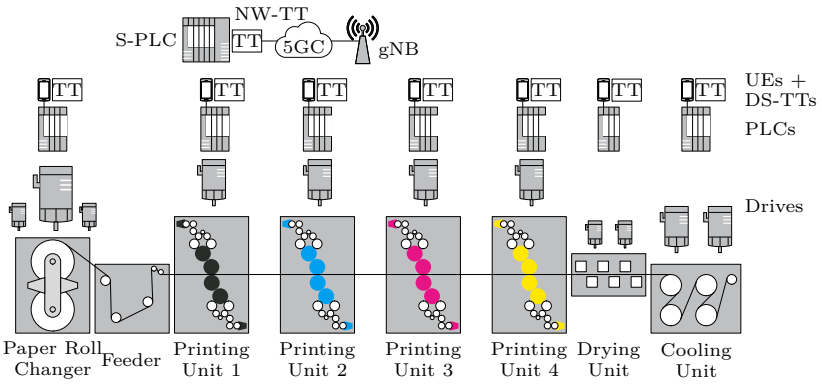

Figure 2: Printing machine: Modular architecture that connects subsystems via 5G.

3.3 Use Case 3: Cooperative Carrying

In factory automation mobile robots are often used for transportation and logistic applications. Special purpose mobile robots for such applications are AGVs. AGVs move along predefined paths through the factory floor. The paths are calculated and provided by central guidance control system. During operation the AGVs continuously monitor their current location using various sensors such as cameras or Radio Frequency Identification (RFID) technology. They provide their current location to the guidance control system on the basis which it can adopt the calculated paths in order to prevent collisions.
Collaborative AGVs move at a speed of 6 km/h to 12 km/h during the cooperative car-

rying operation. A synchronization accuracy of 250 µs to a reference point [3GP21], i.e. the guidance control system, which is derived as 25 % of the cycle time [3GP21], yields a precision of 0.83 mm to 1.66 mm of the collaborative movement. The cooperative carrying operation involves up to 8 AGVs simultaneously. It occurs in a service area of up to 1 km^2 [3GP20, 3GP21] which comprises up to 100 AGVs [3GP21].
AGVs implement a local PLC that is in charge for the actual motion control and safety functions. Within an AGV different systems such as actuators and sensors are connected to the local PLC. For simplicity a line/ring network is assumed as industrial practice. A switched network may also be conceivable. As mentioned before, AGVs require a guidance control system for mission and path planning. The guidance control system is typically located in a central location similar to the S-PLC of a printing machine. Fig. 3 shows a representative cooperative carrying application leveraging integrated TSN/5G networks. Two AGVs carry a difficult work piece that is too large and heavy to be carried by a single AGV. The AGVs local PLCs are connected via a dedicated UE and DS-TT to the 5G RAN. The guidance control system, here represented as S-PLC, is connected to the 5GC via a NW-TT.
The benefits of using integrated TSN/5G networks over other technologies such as Wifi6 lies in the ultra reliability and ultra low latency profile of 5G in conjunction with the RT interface provided with TSN. The short cylce times and the high synchronization accuracy allows novel collaborative processes with high precision. With the spectral efficiency of 5G more AGVs can be deployed in parallels on the factory floor.

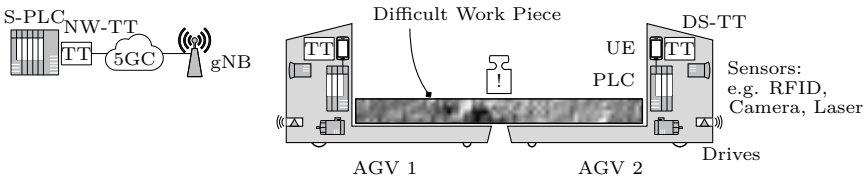

Figure 3: Cooperative carrying: Multiple AGVs collaborate to carry a work piece.

4 Assessment of Use Cases in Factory Automation

As discussed in Sec. 1 integrated TSN/5G networks involve heterogeneous synchronization mechanisms. From an E2E perspective gPTP is the fundamental synchronization mechanism. In PTP based synchronization a grandmaster (GM) clock provides its reference time to the remaining clocks in the network. Our assessment of integrated TSN/5G networks is based on [S+21]. This study researches different GM deployment scenarios in integrated TSN/5G networks and evaluates their E2E performance. The study distinguishes three distinct GM deployment strategies. First, a network-sided GM which is the state-of-the-art approach and specified in Rel. 16. It refers to a GM that is connected to the 5GC. Second, is a device-sided GM which is part of the ongoing Rel. 17 specifications. It refers to a GM that is connected to the 5G RAN. And

third, is a 5G-sided GM which is a novel strategy proposed in [S+21]. It refers to a scenario where the 5G system itself serves as a GM and provides its reference time the both network- and device-sided devices.

4.1 Use Case 1: Robotic Motion Control

As discussed in Sec. 3.1 and shown in Tab. 1, the robot motion control use case requires a stringent E2E synchronization accuracy of 1 µs implicating that device clocks may deviate 0.5 µs from the GM clock. For the robotic motion control use case it is critical that the PLC and the field devices, such as wireless sensors, are tightly synchronized. As discussed before, both devices are connected to the 5G RAN using a dedicated UE and DS-TT. According to [S+21] the 5G-sided GM deployments is capable of synchronizing device-sided device with a worst-case accuracy of 0.44 µs or 0.37 µs for a subcarrier spacing of 60 kHz or 120 kHz respectively, and thus is able to support the robotic motion control requirements from a synchronization perspective. However, the estimations from [S+21] use a rather pessimistic paramterization of the clock properties. When more realistic parameters are used from a practical point of view, more deployment options become feasible. We use relaxed parameterization that is a timestamp granularity of 8 ns, a link propagation delay of 50 ns which is equivalent to a 15 m wired link [G+17]. Then a 5G-sided GM deployment achieves a worst-case synchronization accuracy of 0.46 µs, 0.34 µs, or 0.27 µs for a subcarrier spacing of 30 kHz, 60 kHz, or 120 kHz respectively. In addition the state-of-the-art network-sided GM deployment is now able to support the synchronization requirements with an accuracy of 0.45 µs or 0.38 µs for subcarrier spacing of 60 kHz or 120 kHz respectively.

4.2 Use Case 2: Printing Machine

As discussed in Sec. 3.2 and shown in Tab. 1, printing machines impose among the most stringent communication requirements especially in regard to the synchronization accuracy. As elaborated printing machines require its field devices, i .e. drives within a printing unit, to be synchronized within 0.125 µs to 0.625 µs in reference to the S-PLC in order achieve an E2E synchronization accuracy of 0.25 µs to 1.25 µs.
According to [S+21] a 5G-sided GM deployment is able to achieve an E2E synchronization accuracy between S-PLC and subsystem PLC of 0.62 µs or 0.55 µs for a subcarrier spacing of 60 kHz or 120 kHz respectively. A state-of-the-art network-sided GM, where the S-PLC serves as GM, achieves a E2E synchronization accuracy of 0.62 µs for a subcarrier spacing of 120 kHz. However, this leaves not much head room for synchronizing field devices that are connected to the subsystem PLC. Using a relaxed paramterization as elaborated in Sec. 4.1 creates enough head room for synchronizing field devices with the required precision. Fig. 4 represents the synchronization accuracy of field devices connected to the subsystem PLC over different subcarrier spacings and for different GM deployments. On the left the network-sided GM deployment is shown where the S-PLC serves as GM. On the right the 5G-sided GM deployment is shown where the S-PLC and the field devices are synchronized equally by the 5G system. As expected the 5G-sided GM deployment performs best and is able to synchronize up to

3, 9, or 13 field devices, which are connected in a line network to the subsystem PLC, with the required precision of 0.625 µs. When ring networks are established at field level, then up to 5, 17, or 25 field devices can be supported beyond the PLC.

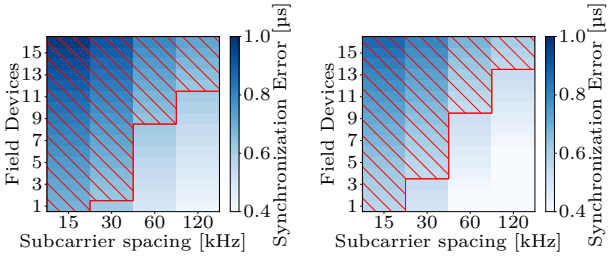

Figure 4: Assessment of E2E synchronization accuracy for printing machines: Network-sided GM (left) and 5G-sided GM (right). The hatched area indicated where the worst-case E2E synchronization error exceeds 0.625 µs.

4.3 Use Case 3: Cooperative Carrying

As discussed in Sec. 3.3 and shown in Tab. 1, the cooperative carrying use case imposes relatively relaxed requirements regarding the synchronization accuracy. The cooperative carrying use case requires the AGVs to be synchronized E2E within 500 µs in order to achieve a precision of 0.83 mm to 1.66 mm. This yields a maximum deviation of 250 µs for the AGV clocks towards the GM clock, e.g. the guidance control system. According to [S+21] the synchronization accuracy requirements imposed by the cooperative use case can easily be achieved with integrated TSN/5G networks even under worst-case parameterization. Using a network-sided GM, e.g. the guidance control system, AGVs can be synchronized with a worst case E2E synchronization accuracy of 0.88 µs, 0.81 µs, 0.69 µs, or 0.62 µs for a subcarrier spacing of 15 kHz, 30 kHz, 60 kHz, or 120 kHz respectively. This means that the speed of movement of the collaborative AGVs can be easily increased at least from a synchronization perspective.

5 Conclusion and Future Work

In this work we derived key performance indicators to classify different use case areas of factory automation. We provided a collection of use cases and their requirements in reference to the key performance indicators. We described a selection of use cases in detail highlighting their networks and application specific synchronization requirements. We compared the use case requirements with a study of heterogeneous synchronization mechanisms in integrated TSN/5G networks and assessed their feasibility.
Both robotic motion control and printing machine use cases impose stringent synchronization accuracy requirements. However, in reference to the emphasized study it becomes evident that both use cases can be implemented using integrated TSN/5G networks when a more sensible parameteriztation is assumed than originally presented

in the study. Important for the implementation of such cases is the grandmaster selection and the 5G subcarrier spacing configuration. The cooperative carrying use case easily can be implemented with TSN/5G networks even using the state-of-the-art network-sided grandmaster deployment and the worst-case parameterization as presented in the reference study.
For our future work we plan to further collect more relevant use cases. Since the assessment of the shown use cases is based solely on a single study from a single reference, we plan to elaborate real-world measurements as soon as Rel. 16 networks and equipment becomes available. First preliminary measurements already confirmed the potential of integrated TSN/5G networks.

Bibliography

[3GP20] 3GPP. Study on Communication for Automation in Vertical domains (CAV) V16.3.0 (Release 16), 2020.

[3GP21] 3GPP. Service Requirements for Cyber-Physical Control Applications in Vertical Domains, 2021.

[5G-19] 5G-ACIA. 5G for Automation in Industry, 2019.

[5G-21] 5G-ACIA. Integration of 5G with Time-Sensitive Networking for Industrial Communications, 2021.

[Bro18] Gabriel Brown. Ultra-Reliable Low-Latency 5G for Industrial Automation, 2018.

[D+17] S. Dietrich et al. Performance indicators and use case analysis for wireless networks in factory automation. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8, 2017.

[F+14] A. Frotzscher et al. Requirements and current solutions of wireless communication in industrial automation. In *2014 IEEE International Conference on Communications Workshops (ICC)*, pages 67–72, 2014.

[G+17] M. Gutiérrez et al. Synchronization Quality of IEEE 802.1AS in Large-Scale Industrial Automation Networks. In *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 273–282, 2017.

[Kip01] Helmut Kipphan. *Handbook of Print Media: Technologies and Production Methods*. Springer, 2001.

[Nof09] Shimon Nof. *Springer Handbook of Automation*. Springer, 2009.

[S+21] Maximilian Schüngel et al. Heterogeneous Synchronization in Converged Wired and Wireless Time-Sensitive Networks. In *2021 17th IEEE International Conference on Factory Communication Systems (WFCS)*, pages 67–74, 2021.

http://dx.doi.org/10.25673/39592

Ethernet-basierte Ultra-Hochgeschwindigkeitskommunikation für eine Regelung dezentraler Einspeiseumrichter von Windenergieanlagen

Holger Flatt[1], Sebastian Schriegel[1], Jan F. Westerkamp[2],
Ingo Mackensen[2], Albrecht Gensior[3] und Jürgen Jasperneite[1]

[1]Fraunhofer IOSB, Institutsteil für industrielle Automation (IOSB-INA), Lemgo
{Holger.Flatt, Sebastian.Schriegel, Juergen.Jasperneite}@iosb-ina.fraunhofer.de

[2]WRD Wobben Research and Development GmbH, Aurich
{Jan.Westerkamp, Ingo.Mackensen}@enercon.de

[3] Fachgebiet Leistungselektronik und Steuerungen in der Elektroenergietechnik, Technische Universität Ilmenau
Albrecht.Gensior@tu-ilmenau.de

Abstract: In diesem Beitrag wird eine Ethernet-Ultra-Hochgeschwindigkeitskommunikation vorgestellt, welche auf der Nutzung von GBit-Ethernet-PHYs in Verbindung mit einer FPGA-Protokollverarbeitung basiert. Hierzu wurde ein mit IEEE 802-Standards kompatibles Protokoll entwickelt, welches es dem Empfänger ermöglicht, Daten bereits vor dem vollständigen Empfang eines Paketes zu nutzen. Die Protokollverarbeitung wurde vollständig in Hardware umgesetzt, welche alle Verarbeitungsschritte zwischen der GBit-Schnittstelle des PHYs und der systeminternen Datenschnittstelle vornimmt. Im Rahmen einer Fallstudie wurde ein FPGA-basierter Systemaufbau zur quantitativen Ermittlung der Kommunikationseigenschaften umgesetzt. In jedem Zyklus sendet dabei eine Zentralsteuerung Echtzeitdaten zur Ansteuerung von IGBT-Umrichter-Brücken und empfängt jeweils Strommesswerte. Für die in der Anwendung relevanten Mengen an zu übertragenden Echtzeitdaten wurden an dem Aufbau Ende-zu-Ende-Latenzen in der Größenordnung < 1 µs sowie ein Jitter < 50 ns gemessen. Aufgrund der Protokollverarbeitung repräsentieren diese Werte die Kommunikationseigenschaften zwischen den späteren Applikationen, welche in der Steuerung und den verteilten IGBT-Umrichterbrücken ausgeführt werden.

1 Einleitung

Die Erhöhung des energetischen Wirkungsgrades ist heute für viele technische Systeme ein primäres Entwurfsziel. Technische Einrichtungen wie Einspeiseumrichter von Windenergieanlagen profitieren von höheren Wirkungsgraden häufig davon, dass durch eine reduzierte Verlustleistung eine Reduktion der Baugrößen und damit auch eine Reduktion von Anschaffungskosten möglich wird.

Da Einspeiseumrichter für Windenergieanlagen von Enercon auf Grund von Skalierbarkeit, Transportierbarkeit und Wartbarkeit heute vielfach modular aufgebaut sind, sind Strommesspunkte und IGBT-Ansteuerungen (IGBT) sowie die zentrale Steuerung der Umrichtereinheiten räumlich verteilt. Insbesondere bei einer zentralen Regelung der Ströme ist eine leistungsfähige Vernetzungslösung mit einer sehr geringen Latenz sowie Jitter erforderlich, da Kommunikationszeiten die Regelungsleistung negativ beeinflussen [CA18]. Werden darüber hinaus Regler für Schaltungsaufgaben eingesetzt, muss die Regelungsaufgabe innerhalb kürzester Zykluszeiten in Echtzeit erfolgen. Dabei werden die Strommesswerte aller Umrichtereinheiten (bis zu acht Stück parallel) ausgewertet und auf dieser Grundlage den IGBT-Halbbrücken aktualisierte Ansteuerungssignale zur Durchführung der Schaltungen übermittelt. Für aktuelle Anwendungen werden in der Literatur Regelungsfrequenzen von bis zu 640 kHz gefordert [Ge20].

Im Rahmen dieses Beitrages wird ein neuer Ansatz einer Ultra-Hochgeschwindigkeitskommunikation vorgestellt. Dieser basiert auf der Nutzung der GBit-Ethernet-Technologie, welche sich durch Kosteneffizienz auszeichnet. Die in einem FPGA umgesetzte Protokollverarbeitung, welche auf IEEE 802-Standards basiert, ermöglicht eine beschleunigte Auswertung von Echtzeitdaten und latenzarme Kopplung von Regler, IGBT-Ansteuerung und Strommesselektronik. Die Lösung ist im Sinne eines Plug-and-Play entworfen, so dass keine manuellen Konfigurationsschritte bei der Inbetriebnahme durchgeführt werden müssen.

Der Beitrag ist wie folgt gegliedert: Abschnitt 2 stellt zunächst die applikativen Anforderungen an die Regelung dezentraler Einspeiseumrichter von Windenergieanlagen dar. Abschnitt 3 thematisiert den neuen Ansatz zur Hochgeschwindigkeitskommunikation und fokussiert insbesondere die beiden Bereiche Protokollentwicklung und die daraus resultierende Hardware-Architektur. In den Abschnitten 4 und 5 werden eine Fallstudie und die

daraus resultierenden Ergebnisse vorgestellt. Abschnitt 6 fasst den Beitrag zusammen und gibt einen Ausblick auf zukünftige Arbeiten.

2 Anforderungen an die Regelung dezentraler Einspeiseumrichter von Windenergieanlagen

Für eine neue Generation von Windenergieanlagen des Unternehmens Enercon soll eine Regelung von acht IGBT-Halbbrücken pro Phase über eine Zentralsteuerung erfolgen, welche wiederum an ein überlagertes System angeschlossen ist. Eine Anforderungsanalyse des Unternehmens Enercon hat ergeben, dass bei einer Regelung eine Kommunikation von Echtzeitdaten zwischen Umrichtereinheiten mit den IGBT-Halbrücken und der Zentralsteuerung in weniger als 1 µs und mit einem Jitter < 100 ns erfolgen muss. Echtzeitdaten sind beispielsweise die Stromdaten der IGBTs sowie die Schaltsignale für die IGBTs. Bei dem geplanten Aufbau, welcher jeweils einzelne Schaltschränke für die Zentralsteuerung und die Umrichter vorsieht, kann mit einer Leitungslänge von 10 m zwischen der Zentralsteuerung und den jeweiligen Umrichtereinheiten gerechnet werden.

Als weitere Anforderungen neben den extrem kleinen Latenzzeiten sind (1) eine zukunftsrobuste Technologielösung durch Nutzung von Standards, (2) Kosten und (3) einfache Inbetriebnahme in den Entwurf der Lösungsarchitektur eingeflossen.

Um Anforderungen an kürzeste Kommunikationszeiten in der Größenordnung von 1 µs, niedrige Hardware-Kosten für Kommunikationskomponenten (z.B. PHYs) sowie Leitungslängen von bis zu 10 m zu ermöglichen, wurde GBit-Ethernet (1000BASE-T) als Kommunikationstechnologie ausgewählt. Eine unidirektionale Übertragung eines minimal großen Paketes (7 Byte Präambel, 1 Byte SFD sowie 64 Datenbytes) benötigt bei einer GBit-Übertragungsrate 576 ns (72 Byte * 8 ns / Byte). Hinzu kommen noch Verzögerungszeiten der PHYs (0,4 µs am Beispiel von [Ma18]), interne Verarbeitungszeiten in der Zentralsteuerung sowie den Umrichtereinheiten zur Paketerzeugung bzw. Analyse. Um eine Ende-zu-Ende-Latenz von 1 µs signifikant zu unterschreiten, sind daher weitere Optimierungsmaßnahmen an der Ethernet-Kommunikation notwendig.

Echtzeit-Kommunikationssysteme ermöglichen häufig Linientopologien, welche eine lineare Abhängigkeit der Verzögerungszeiten von der Anzahl der Geräte aufweisen [Ma19]. Um kürzeste Kommunikationszeiten ermöglichen zu können, ist es erforderlich, die Zentralsteuerung über parallele Punkt-zu-Punkt-Verbindungen direkt mit den Umrichtereinheiten zu koppeln und somit eine Sterntopologie aufzubauen.

Als weitere Anforderungen gilt es, neben den Echtzeitdaten, weitere Daten wie z.B. Temperaturen zu übertragen, Übertragungsfehler zuverlässig zu erkennen sowie Synchronisationsmechanismen für die zyklische Kommunikation zur Verfügung zu stellen.

Ein derartiges Kommunikationssystem, welches eine Zentralsteuerung mit acht Umrichtereinheiten unter den gegebenen Randbedingungen und Anforderungen vernetzen kann, ist bisher nicht bekannt.

3 Neuer Ansatz zur Hochgeschwindigkeitskommunikation

Marktverfügbare und zukünftige Echtzeit-Kommunikationssysteme wie Profinet oder Ethernet TSN-basierte Systeme können den verschiedenen Anforderungen und Eigenschaften spezifischer Maschinen, Anlagen und anderer Applikationen z.B. durch Auswahl der Geräte, Konfigurationsoberflächen oder Plug-and-Play-Mechanismen angepasst werden [Re18]. Anpassbar sind dabei z.B. die Topologie, Adressierung, Anzahl der Geräte oder Datenraten. Eine reine Sterntopologie reicht für die Anwendungen nicht aus [Uc18]. Die Systeme müssen dadurch Bridging, d.h. die Weiterleitung von Frames nutzen, um verschiedene Topologiemöglichkeiten realisieren zu können. Als Anforderung für die Bridging-Latenzen bei Gigabit-Datenrate wird häufig der Wert < 1 µs gefordert [Es20]. Da die Anforderungen der Umrichterapplikation, die in diesem Beitrag beschrieben ist, eine Ende-zu-Ende-Latenz von < 1 µs fordert, können Systeme, die TSN-Bridging verwenden, nicht integriert werden.

Um kürzeste Ende-zu-Ende-Latenzen und einen minimierten Jitter zu ermöglichen, wird ein Kommunikationsansatz auf Basis von IEEE-Ethernet-Technologie und einer reinen Sterntopologie vorgeschlagen, welcher zudem auf einem für den gegebenen Anwendungszweck optimierten Protokoll basiert sowie eine Protokollverarbeitung in Hardware (hier FPGAs) ermöglicht. Abbildung 1 stellt die Gesamtarchitektur bestehend aus Zentralsteuerung und Umrichtereinheiten dar. In den nachfolgenden Abschnitten erfolgt eine Beschreibung der einzelnen Teilthemen.

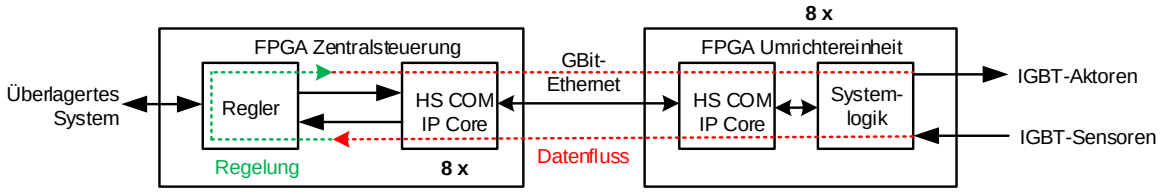

Abbildung 1: Gesamtarchitektur bestehend aus Zentralsteuerung und Umrichtereinheiten

3.1 Regelung und Zeitverhalten

Im Rahmen der Regelung der Ströme eines Umrichtersystems mit Hilfe von IGBT-Halbbrücken, wovon jeweils drei einen Umrichter bilden, ist es erforderlich, Sensordaten der IGBT-Halbbrücken (z.B. Stromwerte) an die Zentralsteuerung zu übertragen. Der Regler berechnet anschließend Ansteuerungssignale für die IGBT-Halbbrücken, welche dann beginnend mit einem neuen Kommunikationszyklus zurück an die Umrichtereinheiten gesendet werden.

Da die Übertragung bei Ethernet in Paketen erfolgt, werden alle Echtzeitdaten jeweils pro Zyklus in einem einzigen Paket versendet. Üblicherweise muss dabei der Empfänger warten, bis das gesamte Paket empfangen ist und kann dann erst die Datenintegrität mit Hilfe des CRCs am Ende des Paketes prüfen. Um die Übertragungslatenz zu reduzieren, werden die Echtzeitdaten am Anfang des Paketes übertragen und mit Hilfe eines weiteren CRCs abgesichert, so dass diese bereits vor Übertragungsende des vollständigen Paketes genutzt werden können. Die Sendezeitpunkte der IGBT-Sensordaten leiten sich aus den Empfangszeitpunkten der IGBT-Aktordaten ab und können für eine Optimierung der Regelungsapplikation zeitlich mit einem Offset verschoben werden. Abbildung 2 visualisiert das beschriebene Zeitverhalten des entwickelten Kommunikationsansatzes zwischen Zentralsteuerung und Umrichtereinheiten.

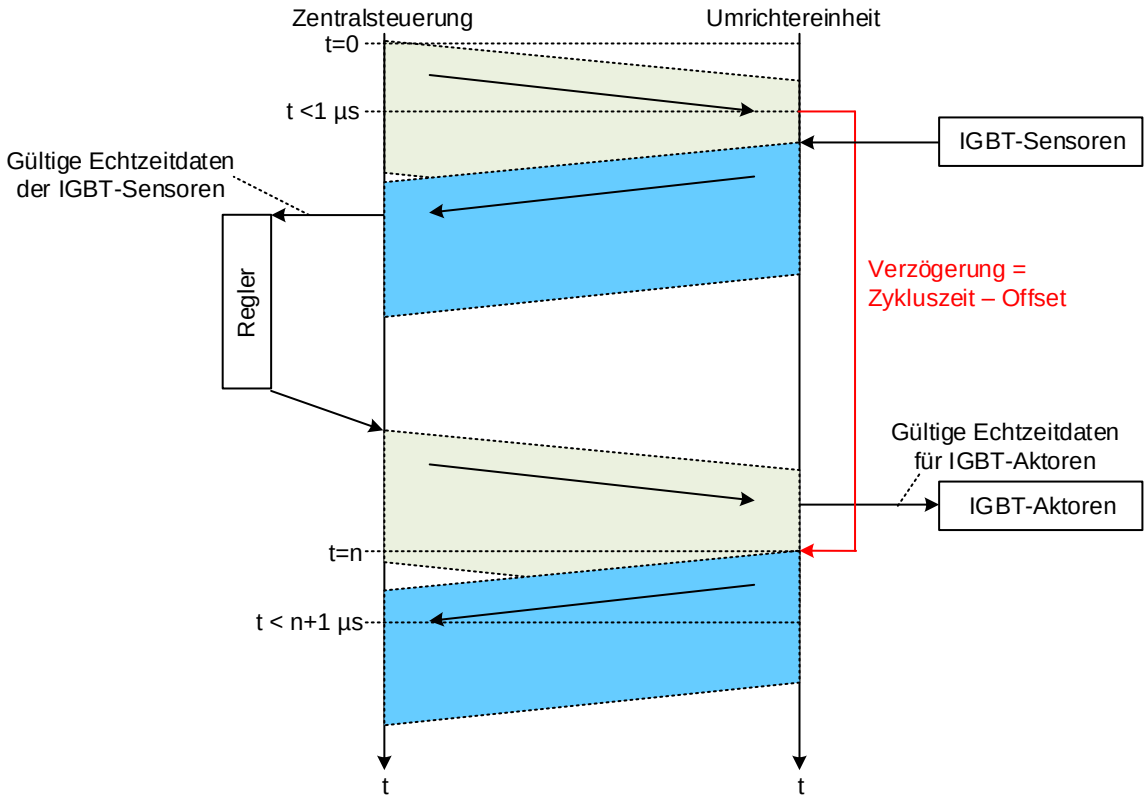

Abbildung 2: Kommunikationsansatz zwischen Zentralsteuerung und Umrichtereinheiten

3.2 Schnittstelle zum überlagerten System

Die von den Umrichtereinheiten gesendeten Messdaten liegen im FPGA der Zentralsteuerung vor und werden von dem dort implementierten Teil des Regelungsalgorithmus verarbeitet. Ein zweiter Teil dieses Algorithmus

ist in einer Echtzeitanwendung implementiert, die von einem der Prozessoren des verwendeten SoC abgearbeitet wird. Über eine Schnittstelle zwischen FPGA und Prozessor (implementiert als AXI-Bus) tauschen beide Teile des Regelungsalgorithmus Daten miteinander aus. Eine detaillierte Beschreibung findet sich in [Ge20].

3.3 Optimiertes Kommunikationsprotokoll

Zur Umsetzung des in Abschnitt 2 vorgestellten Regelungsansatzes wurde ein Ethernet-Layer-2-Protokoll konzipiert (vgl. Tabelle 1), welches die abgeleiteten Anforderungen wie in den folgenden Abschnitten beschrieben ist, umsetzt:

Standardkonformität

Um im Betrieb mit Standard-Werkzeugen Diagnosen der Kommunikation vornehmen zu können, besteht das Protokoll auf den grundlegenden Eigenschaften des IEEE 802.3-Standards. Da insbesondere die Präambel und der Start-of-Frame-Delimiter am Anfang des Paketes, der Standard-Ethernet-CRC am Paketende sowie die zulässigen Paketgrößen von 64-1514 Byte berücksichtigt sind, kann mit üblichen Werkzeugen (z.B. Ethernet-Tap in Verbindung mit Wireshark) eine Aufzeichnung und Diagnose einzelner Pakete ermöglicht werden.

Optimierung auf Effizienz und einfache Möglichkeit zur Umsetzung in Hardware

Mit dem Ziel, eine einfache Umsetzung in Hardware (z.B. FPGA oder ASIC) zu ermöglichen, wird ein einheitliches Paketformat für beide Übertragungsrichtungen genutzt.

Da die Zentralsteuerung über dedizierte Verbindungen mit den Umrichtereinheiten kommuniziert, ist im Normalfall keine Adressierung mit Angabe von Quelle und Ziel erforderlich. Wird aber berücksichtigt, dass in realen Anlagen eine Vertauschung der Leitungen möglich ist, muss hier eine systemseitige Erkennungsmöglichkeit gegeben sein. Gemäß IEEE 802.3 könnten hier per Standard die jeweils 6 Bit großen Felder Source und Destination genutzt werden. Da für die beabsichtigte Umrichtersteuerung max. 9 Geräte im Netzwerk vorhanden sind, ermöglicht ein 1-Bit-Adressfeld mit einer Aufteilung in zwei 4-Bit-Felder eine deutlich ressourceneffizientere Adressierungsmöglichkeit. Im Anschluss an das Adressfeld erfolgt eine Hardware-Versionsnummer, welche eine Erkennung unterschiedlicher Firmware-Stände auf den Geräten ermöglicht.

Latenzoptimierte Übertragung von Echtzeitdaten

Im Anschluss an die Hardware-Versionsnummer können in der Anzahl n parametrisierbare Bytes mit Echtzeitdaten übertragen werden. Bei üblicher Ethernet-Kommunikation wären auf Empfängerseite die empfangenen Echtzeitdaten erst gültig, wenn am Paketende die CRC als gültig erkannt wurde. Um die Echtzeitdaten frühzeitig auf Empfängerseite nutzen zu können, sichert ein 4-Byte-Intermediate CRC die zuvor empfangenen Daten ab. Als weitere Optimierungsmaßnahme wird die Ethernet-Präambel von 7 auf 2 Byte verkürzt.

Zusätzliche Übertragung von Nicht-Echtzeitdaten

Weitere mit der Anzahl m parametrisierbare Bytes an Nichtechtzeitdaten folgen im Anschluss an den Intermediate-CRC. Sofern erforderlich, müssen diese um p Padding Bytes ergänzt werden, um eine minimale Paketgröße gemäß IEEE 802.3 von 64 Byte nicht zu unterschreiten.

Erkennung von Übertragungsfehlern

Ein wesentliches Merkmal, welches vom Protokoll ermöglicht wird, ist die Erkennung von Fehlern. Neben der Erkennung vertauschter Leitungen zwischen den Geräten, unterschiedlichen Firmware-Ständen im Netzwerk sowie CRC-Fehlern muss es auch möglich sein, weitere folgende Fehler zu erkennen:

- Falsche Paketgröße
- Verlorene Pakete
- Link-Unterbrechungen

Falsche Paketgrößen können vom Empfänger einfach erkannt werden, da jedes Paket dieselbe Größe aufweist. Verlorene Pakete werden über einen 2-Byte-Frame Counter erkannt, der nach jedem Frame inkrementiert wird. Link-Unterbrechungen können entweder ebenfalls über den Frame Counter erkannt werden oder durch eine Überwachung der Linkverbindung in Hardware. Weitere mögliche Fehlertypen, wie z.B. falsche Paket-Reihenfolgen und doppeltes Empfangen von Paketen können in der Praxis ausgeschlossen werden, da diese

aufgrund der Punkt-zu-Punktverbindungen nicht möglich sind. Eine Erkennung ist dennoch über den Frame-Counter möglich.

Tabelle 1: Ethernet basiertes Kommunikationsprotokoll für die Regelung dezentraler Einspeiseumrichter von Windenergieanlagen

Feldname	**Größe in Bytes**	**Erläuterung**
Preamble and SFD	3	Verkürzte Ethernet Präambel und SFD
Addressing	1	Nutzung von 2 Nibbles zur Adressierung von Datenquelle und -ziel
Hardware Version	1	Hardware-Versions-Index zur Absicherung unterschiedlicher Firmware-Stände
High Speed Com	n	Nutzdaten für schnelle Echtzeitkommunikation
Intermediate CRC	4	CRC-Absicherung der zuvor übertragenden Inhalte
Low Speed Com	m	Nutzdaten für Nicht-Echtzeitkommunikation
Padding	p	Einfügen von Nullen zur Vergrößerung des Frames auf 64 Byte
Frame Counter	2	Erkennung nicht sequentiell übertragener oder verlorener Frames
Sync Offset	2	Offset zur Anpassung des Zeitpunktes für Datenrücksendung durch Umrichter
Error field	1	Rückübertragung von Fehlercodes an Kommunikationspartner
CRC	4	Standard Ethernet CRC

3.4 IP Core für Ethernet-basierte Ultra-Hochgeschwindigkeitskommunikation

Zur Umsetzung des in Tabelle 1 vorgestellten Protokolls wurde ein FPGA-basierter IP Core entwickelt, welcher den Datenaustausch zwischen der PHY-Schnittstelle (RGMII) und einer systemseitigen internen Schnittstelle (RX Daten, TX Daten) vornimmt. Voraussetzung zur Nutzung ist, dass die komplette PHY-Schnittstelle über den Logikteil des FPGA erreichbar ist (nicht über den CPU Teil im Fall von SoC-FPGAs). Ergänzend wird ein interner Systemtakt (SYSCLK) benötigt, welcher synchron zur systemseitigen Datenschnittstelle ist. Der IP Core ist modular aufgebaut und besteht aus den folgenden Komponenten:

Ein Konverter, welcher mit der RGMII-Schnittstelle verbunden ist, stellt systemseitig eine GMII-Schnittstelle zur Verfügung, da die RGMII-Schnittstelle Daten im Double Data Rate-Betrieb (DDR) verarbeitet und systemseitig ein Single Data Rate-Betrieb erforderlich ist. Angeschlossen an die GMII-Schnittstelle sind jeweils ein RX- und ein TX-MAC Core, welche die MAC-Schicht des Ethernet Layer 2 umsetzen (Präambel und SFD hinzufügen und entfernen sowie CRC am Frameende einfügen bzw. kontrollieren). Die RX- und TX-MAC Cores kommunizieren mit zwei Modulen (Frame Decoder und Frame Encoder), welche das in Tabelle 1 dargestellte Protokoll implementieren. Zur Kommunikation mit den nachgelagerten Einheiten wird hier eine einfache parallele Datenschnittstelle (RX und TX Daten) zur Verfügung gestellt. Die hierfür notwendige Serialisierung bzw. Deserialisierung erfolgen im Encoder bzw. Decoder. Zudem sind alle im Protokoll spezifizierten Funktionen wie z.B. Intermediate CRC, Frame Counter und Fehlerüberprüfung ebenfalls im Encoder und Decoder implementiert. Abbildung 3 stellt die Hardware-Architektur des High Speed Com IP Cores mit den beschriebenen Modulen dar. Grundlegende Eigenschaften, wie z.B. die Menge an Echtzeit- und Nichtechtzeitdaten können separat nach Kommunikationsrichtung parametriert werden, so dass ein flexibler an den jeweiligen Anwendungsfall angepasster Betrieb möglich ist.

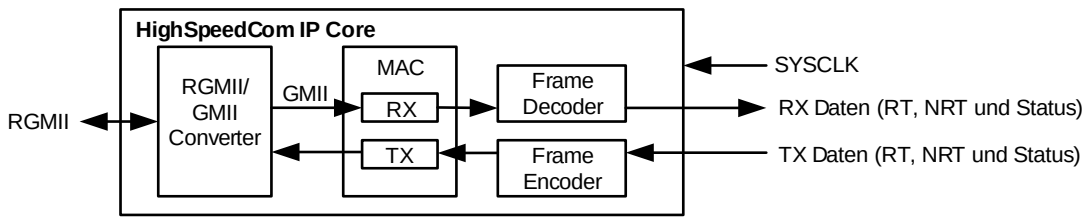

Abbildung 3: Hardware-Architektur des High Speed Com IP Cores

4 Fallstudie für entwickelten Prototyp

Im Rahmen einer Fallstudie wurden zur Evaluation der Kommunikation zwei FPGA-Architekturen entwickelt, welche die Zentralsteuerung und eine Umrichtereinheit repräsentieren. In die FPGA-Architektur der Zentralsteuerung wurde neben dem IP-Core ein zyklischer Datengenerator integriert, der das zeitliche Verhalten des Reglers nachbilden soll. Im FPGA der Umrichtereinheit nimmt ein Test Core die empfangenen Daten entgegen und löst Rückantworten aus. Da die Kommunikation in einem späteren Zielsystem parallel über acht identische IP-Cores auf der Zentralsteuerung stattfindet, ist für die Evaluation der Kommunikationseigenschaften eine FPGA-Architektur mit jeweils einem IP Core in der Zentralsteuerung ausreichend.

Um Latenz und Jitter zu messen, wurde ein Systemaufbau (vgl. Abbildung 4) bestehend aus zwei FPGA-Plattformen der Firma Trenz eingesetzt (TE0720-03 FPGA-Boards in Kombination mit TE0706-03 Basis-Boards) vorgenommen, welche jeweils eine Zentralsteuerung (ZS) und eine Umrichtereinheit (UR) nachstellen. Die FPGA-Boards beinhalten jeweils einen Xilinx Zynq XC7Z020CLG484-1 – FPGA.

Die Boards sind über eine 10 m lange Patch-Leitung verbunden, um eine möglichst reale Einbausituation nachzubilden. Auf beiden Boards werden Trigger-Signale erzeugt und über externe GPIO-Pins ausgegeben. Diese zeigen an, sobald das Senden eines neuen Paktes auf Senderseite veranlasst wird (TX Trigger) bzw. die gültigen Echtzeitdaten auf Empfängerseite dem Test Core zur Verfügung gestellt wurden (RX Trigger). Die Trigger-Signale werden mit einem Oszilloskop zur Messung von Latenz und Jitter erfasst. Abbildung 5 visualisiert das zeitliche Verhalten der Trigger-Signale am Beispiel der Übertragung eines Paketes n.

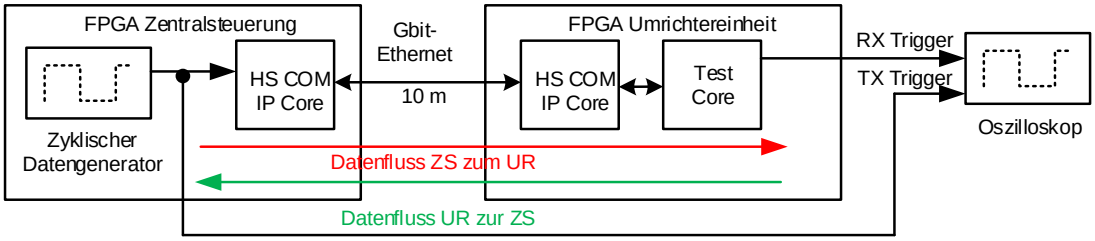

Abbildung 4: Systemaufbau für unidirektionale Messungen von Latenz und Jitter (hier am Beispiel Zentralsteuerung zur Umrichtereinheit)

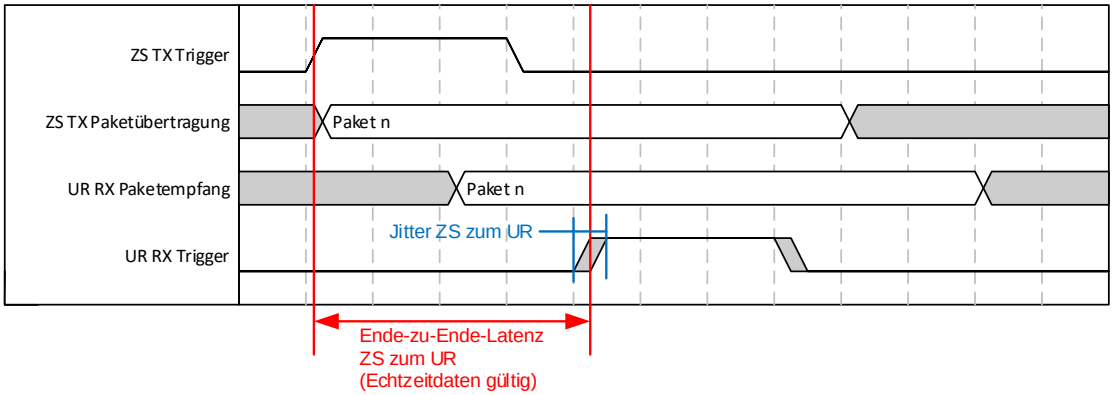

Abbildung 5: Systemaufbau für unidirektionale Messungen von Latenz und Jitter (hier am Beispiel Zentralsteuerung zur Umrichtereinheit)

5 Ergebnisse

5.1 Verkürzung der Ende-zu-Ende-Übertragungslatenz von Echtzeitdaten über einen Gigabit-Ethernet-Link

Um eine schnelle Übertragung von Echtzeitdaten zu ermöglichen, nutzt das in Tabelle 1 vorgestellte Protokoll eine verkürzte Ethernet-Präambel sowie einen Intermediate CRC zur Absicherung der Daten am Anfang des Paketes. Abbildung 6 zeigt einen Vergleich zwischen einer Standard-Übertragung, bei der alle Daten erst nach Empfang des CRC am Paketende gültig sind und der optimierten Übertragung von Echtzeitdaten, die nach

Empfang des Intermediate CRC gültig sind. Für die Berechnung wurden exemplarisch 17 Byte Nichtechtzeitdaten ausgewählt, welche eine Übertragung von Nichtechtzeitdaten im Multiplex ermöglichen (16 Byte Nutzdaten und ein Adressbyte).

Das Ergebnis zeigt, dass bis zur Gültigkeit der Echtzeitdaten je nach Größe bei optimierter Übertragung zwischen 31 und 63 Datenbytes weniger übertragen werden müssen. Bei GBit-Ethernet mit 8 ns pro Byte entspricht das einer Reduktion der Übertragungszeit von 248 – 504 ns.

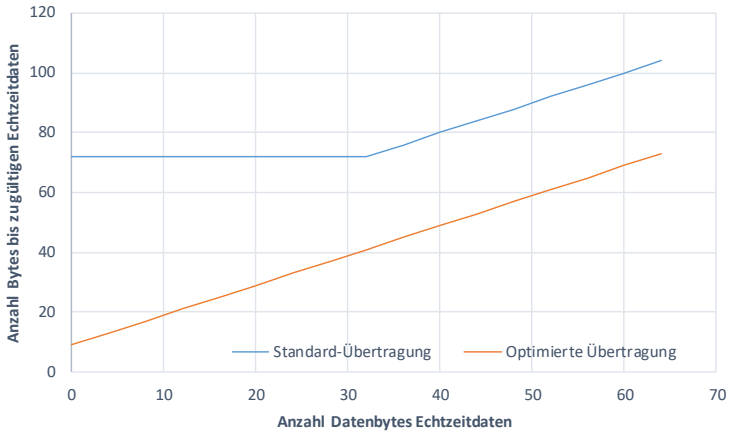

Abbildung 6: Vergleich Standard-Übertragung mit optimierter Übertragung am Beispiel einer Übertragung mit 17 Byte Nicht-Echtzeitdaten für einen Gigabit-Ethernet-Link

5.2 Gemessene Ende-zu-Ende-Latenz und Jitter von Echtzeitdaten

Für Messungen von Latenz und Jitter von Echtzeitdaten wurde der in Abbildung 4 dargestellte Aufbau genutzt. Hierzu wurden eine Konfiguration, welche 13 Byte Echtzeitdaten überträgt, untersucht. Diese Datenmenge ist ausreichend, um beispielsweise 3-phasige Stromdaten mit 16 Bit Auflösung (insg. 6 Byte) sowie zugehörige Statusdaten (7 Byte) zu übertragen.

Das Ergebnis der Messung ist in Abbildung 7 dargestellt. Der linke Teil der Abbildung zeigt den mit einem Oszilloskop erfassten Zeitverlauf der Trigger-Signale für eine unidirektionale Kommunikation. Da die gesamte Übertragungskette Bestandteil der Messung ist, sind die Verzögerungen der beiden beteiligten Ethernet-PHYs und die Verzögerung der 10 m Ethernet-Leitung mit enthalten. Im rechten Teil von Abbildung 7 ist eine Vergrößerung dargestellt, welche eine Bestimmung des Jitters ermöglicht. Es entsteht ein kontinuierlicher Jitter, da die Systemtakte, mit den die beiden Boards operieren, unabhängig voneinander sind. Folgende Ergebnisse wurden ermittelt:

- Ende-zu-Ende-Latenz: ca. 758 ns
- Jitter: ca. 20 ns

Eine Änderung der Datenmenge der Echtzeitdaten hätte eine Latenzänderung um 8 ns/Byte zur Folge sowie eine Änderung der Leitungslänge um ca. 5 ns / m.

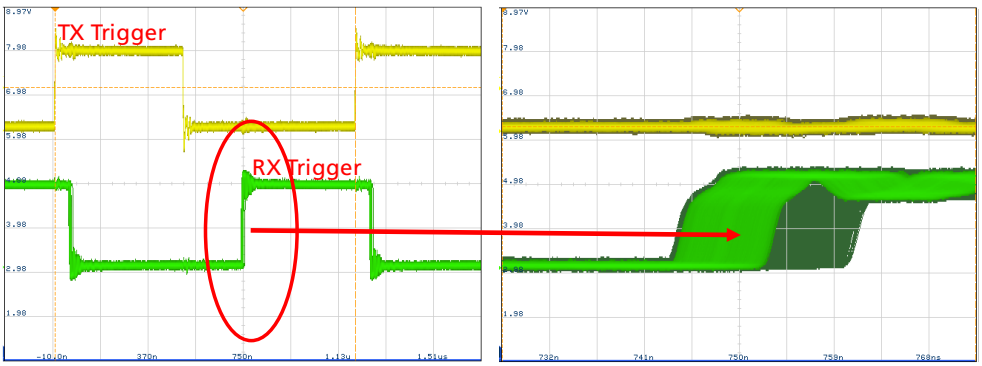

Abbildung 7: Oszilloskop-Screenshot der Messung (hier am Beispiel von 13 Byte Echtzeitdaten)

5.3 FPGA-Ressourcen des IP Cores

Für die Belegung der FPGA-Ressourcen sind primär die Anzahl der belegten CLB-LUTs, CLB-Register und MMCM-Module (Digital Clock Manager) sowie BlockRAMs relevant. Die in Abbildung 4 dargestellte Fallstudie (1 Ethernet-Port, 13 Bytes Echtzeitdaten) belegt zwei MMCM-Module zur Generierung von internen Takten. BlockRAMs werden nur von Xilinx-integrierten Debug-Kernen belegt, die für Entwicklungszwecke verwendet werden. Tabelle 2 listet die Anzahl der CLB-LUTs und CLB-Register auf. Das Ergebnis zeigt, dass der entwickelte IP-Core weniger als 2% der benötigten LUTs und Register belegt. Die Ergebnisse sind für eine spätere Umsetzung einer Umrichtereinheit repräsentativ, da diese ebenfalls über nur einen Ethernet-Port verfügt. Für die Zentralsteuerung muss der IP-Core acht Mal instanziiert werden, so dass in diesem Fall dann ca. 15% der LUTs und 11% der Register belegt werden würden.

Tabelle 2: Belegte CLB-LUTs und –Register der Fallstudie

Modul	**CLB-LUTs, insg. 53200**	**CLB-Register, insg. 106400**
HS Com IP Core	1000	1455
Test Core	109	364
Debug	1523	2996

6 Zusammenfassung und Ausblick

Im Rahmen dieses Beitrages wurde ein neuer Ansatz für eine Ethernet-basierte Ultra-Hochgeschwindigkeitskommunikation für eine Regelung dezentraler Einspeiseumrichter von Windenergieanlagen vorgestellt und evaluiert. Der Ansatz basiert auf einem für die schnelle Übertragung von Echtzeitdaten optimierten Layer-2-Ethernet-Protoll und nutzt GBit-Ethernet als Übertragungsmedium. Die Protokollverarbeitung wird dabei von einem IP Core umgesetzt. Für die in der Anwendung relevanten Mengen an zu übertragenden Echtzeitdaten wurden an einem realen Aufbau Ende-zu-Ende-Kommunikationslatenzen in der Größenordnung < 1 µs sowie ein Jitter < 50 ns gemessen. Der vorgestellte Ansatz ermöglicht im Gegensatz zu bestehenden industriellen Kommunikationssystemen diese kurzen Zykluszeiten, da sowohl Kommunikation als auch Applikation innerhalb eines FPGAs vereint sind und somit zusätzliche Verzögerungszeiten und Jitter entfallen. Eine Nutzung für Regelungsaufgaben insbesondere im Bereich der immer wichtiger werdenden steuerbaren Strom-/ Spannungswandlung in anderen Branchen ist möglich, sofern diese ebenfalls auf einer ähnlichen dezentralen Topologie basieren und ähnlich hohe Anforderungen an die Zykluszeit und den Jitter stellen.

In zukünftigen Arbeiten gilt es, den IP-Core in die FPGA-basierten Zielsysteme der Steuerung und des Umrichters zu integrieren und applikationsseitig mit dem Regler (Zentralsteuerung) bzw. der Applikationslogik als Bindeglied zwischen den Sensoren und Aktoren (Umrichtereinheiten) zu verbinden und die Gesamtsysteme in einem realitätsnahen Umfeld zu evaluieren.

7 Literaturverzeichnis

[CA18] T. P. Corrêa, L. Almeida and E. B. Peña, "Hardware/Software Implementation Factors Influencing Ethernet Latency," 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), 2018, pp. 323-328

[Ge20] Gensior, Albrecht: Approximated sliding-mode control of parallel-connected grid inverters. In: 22nd European Conference on Power Electronics and Applications (EPE'20 ECCE Europe), Lyon, France, 2020

[Ma18] Marvell, “Alaska® 88E1510/88E1518/88E1512/88E1514 Integrated 10/100/1000 Mbps Energy Efficient Ethernet Transceiver”, Datasheet – Public, 2018

[Ma19] X. Ma and C. Zhang, "A Gigabit Real-time Ethernet for Manufacture Automation Control," *2019 6th International Conference on Systems and Informatics (ICSAI)*, 2019, pp. 1040-1045

[Re18] IEC/IEEE 60802 Project Group: Industrial Requirements v12, Online: http://www.ieee802.org/1/files/public/docs2018/60802-industrial-requirements-1218-v12.pdf, 2018.

[Uc18] IEC/IEEE 60802 Project Group: Industrial Use Cases v13, Online: http://www.ieee802.org/1/files/public/docs2018/60802-industrial-use-cases-0918-v13.pdf, 2018.

[Es20] IEC/IEEE 60802 Project Group: IEC/IEEE 60802 Example Selection. Online: http://www.ieee802.org/1/files/public/docs2020/60802-Steindl-et-al-ExampleSelection-0520-v24.pdf, 2020.

[St20] IEC/IEEE 60802 Draft 1.2: Online: http://www.ieee802.org/1/files/private/60802-drafts/d1/60802-d1.pdf, 2020.

Impressum

12. Jahreskolloquium

« KOMMUNIKATION IN DER AUTOMATION »
(KOMMA 2021)

18.11.2021 • Magdeburg

AUFLAGE
70 Exemplare

ISBN 978-3-948749-10-1
DOI: http://dx.doi.org/10.25673/39548

HERAUSGEBER

Ulrich Jumar, Magdeburg
Jürgen Jasperneite, Lemgo

Institut für Automation und
Kommunikation e.V. Magdeburg
An-Institut der Otto-von-
Guericke-Universität Magdeburg
Werner-Heisenberg-Straße 1
39106 Magdeburg

Telefon: +49 (0)391 990140
Fax: +49 (0)391 9901590
Internet: **www.ifak.eu**