

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 Automatisierunginseln 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.

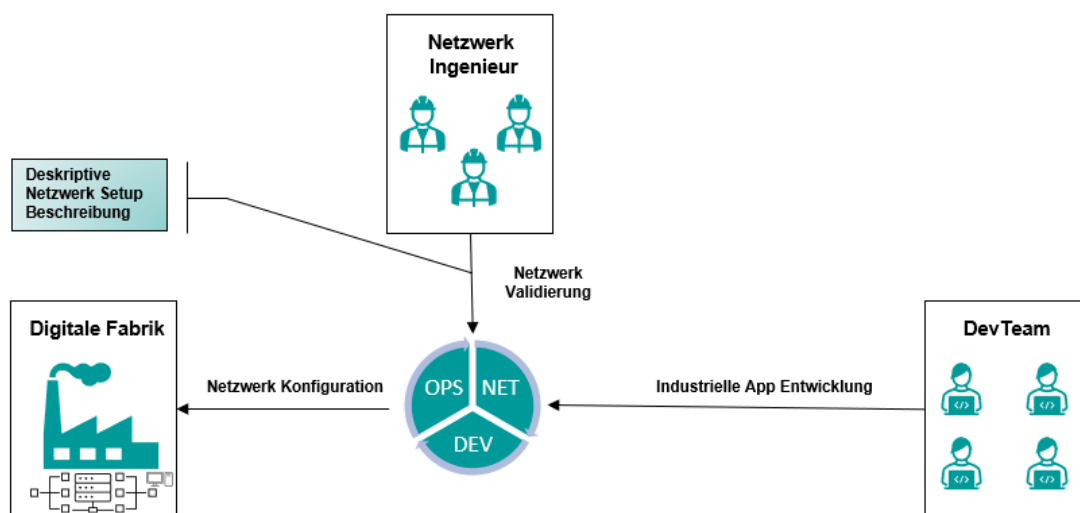


Abbildung 1: Im Fokus des industriellen NetDevOps-Ansatzes für IEC/IEEE 60802 TSN-IA Profile steht der Netzwerk Ingenieur als Anlagenbetreiber, der beispielsweise neue Komponenten in Betrieb genommen hat und den der hinzugefügten Applikation zugrundeliegenden Netzwerkverkehr auf deterministische und 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 anschließender 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 Knotenmodell 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 herstellereigenspezifische 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 Vergleich 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:

$$\textit{ForwardingDelay} = \textit{independentDelayMax} + \textit{dependentDelayMax} \times \textit{FrameSize}$$

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:

$$\textit{ForwardingDelay} = \textit{independentDelayMax}$$

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 Mechanismus das PCP Feld des VLAN Headers ausgelesen, da es, wie im Abschnitt zuvor erwähnt wurde, herstellereigenspezifische 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 höherprioritäre 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 OMNeT/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 Übertragungsintervall 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 Phasenzklus 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 Übertragungsintervall 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:
  - 1.849e-06
  - 1.6014e-06
  - 3.5339999999999999e-07
  total_traveling_time: [5.0220336e-05]
  frame_write_delay_at_send_node: 6.08e-10
  acceptance: true
  worst_case_arrival_times_estimate: [0.006050307632000001]
  arrival_deadlines: [0.0065000000000000001]
  send_order_positions: [124]
  phase_tree_at_allocation: [7.60000000000000011e-08, 7.60000000000000011e-08, 7.60000000000000011e-08, 7.60000000000000011e-08, 7.60000000000000011e-08,
7.60000000000000011e-08, 7.53920000000000011e-08, 7.53920000000000011e-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 transpilieren 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 Knotenmodell 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 Standardstatistiken der Netzwerk Komponenten, die in OMNEST/OMNeT++ nach dem Profinet TSN Knotenmodell 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.

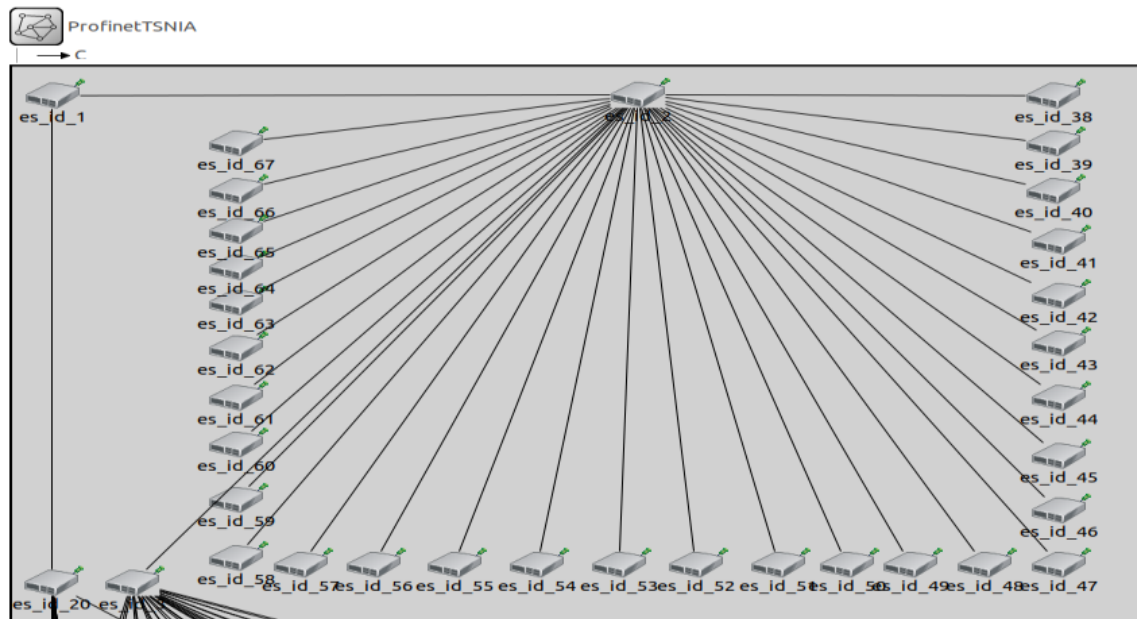


Abbildung 4: Visualisierung eines Topologieausschnitts der transpilierten scenario.ned Datei im OMNEST/OMNeT++ Design Modbus der im nächsten Kapitel 3 vorgestellten Profinet Applikation als Beispiel eines IEC/IEEE 60802 TSN-IA Profils.

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 Sendezum 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 Artefakte heruntergeladen werden. Dabei wird automatisiert zu jedem Szenario ein Netzwerkgraph erstellt und in diesen Artefakten bereitgestellt. Zudem wird nach der Simulation im Postprocessing der Unterschied der Ankunftszeiten der geplanten TSN Streams von Simulation und CNC-Pfadplanungsalgorithmus 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.

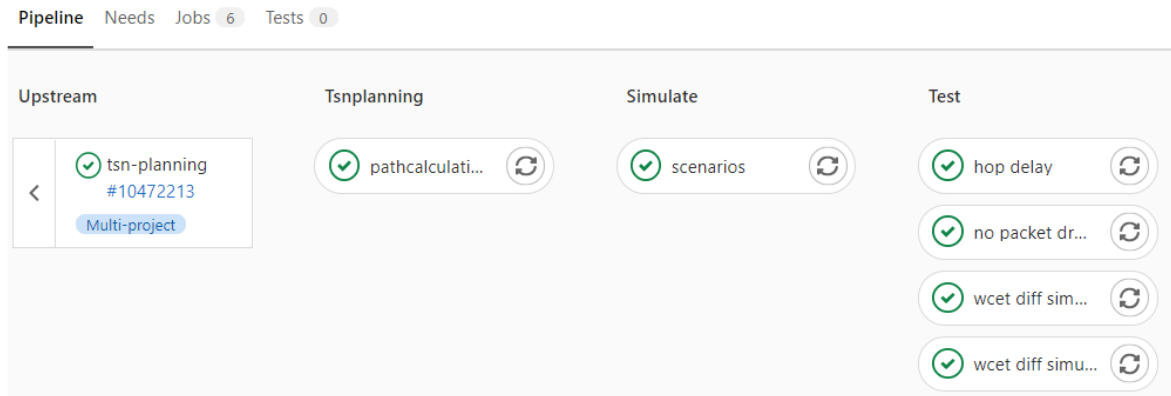


Abbildung 5: CI/CD Pipeline zur automatisierten Validierung der Network QoS (Streams als auch Best-Effort Verkehr)

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 Artefakt weitergegeben.

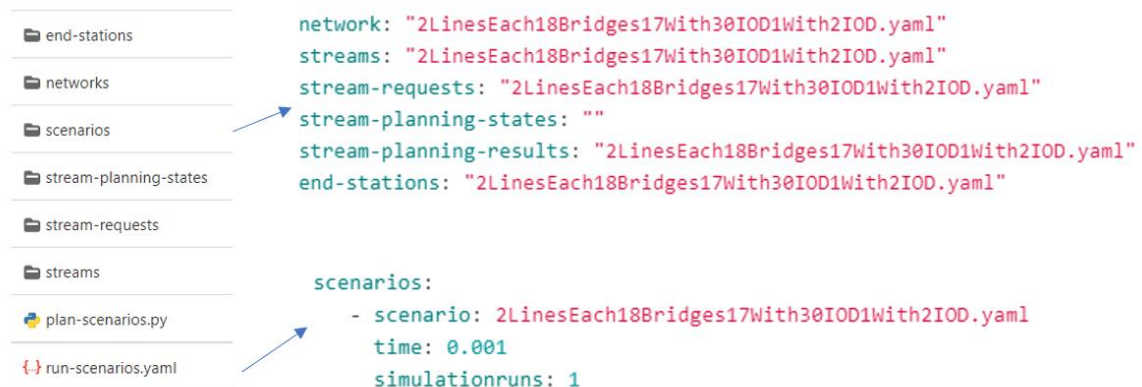


Abbildung 6: Ordnerstruktur des Datenmodells zur Netzwerk Szenario Beschreibung

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 Artefakte 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, abgekürzt 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.

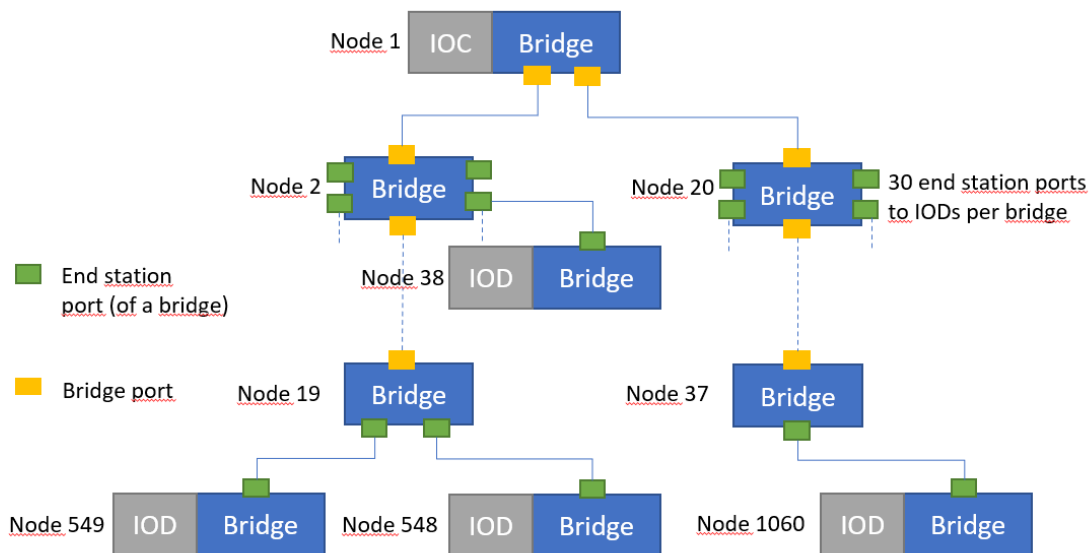


Abbildung 7: Diese Netzwerk Topologie veranschaulicht eine Profinet Applikation als Beispiel eines IEC/IEEE 60802 TSN-IA Profils mit einem Profinet IO Controller als Node 1 und 1023 Profinet IO Devices, zu denen eine Profinet Application Relation besteht.

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 μ s. 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 μ s 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 Echtzeitanforderungen, das heißt das bei Überschreitung der Deadline von 500 μ s 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 $400 \mu\text{s}$. Daher wurden alle TSN Streams akzeptiert, da die Ergebniswerte die Deadline von $500 \mu\text{s}$ 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.

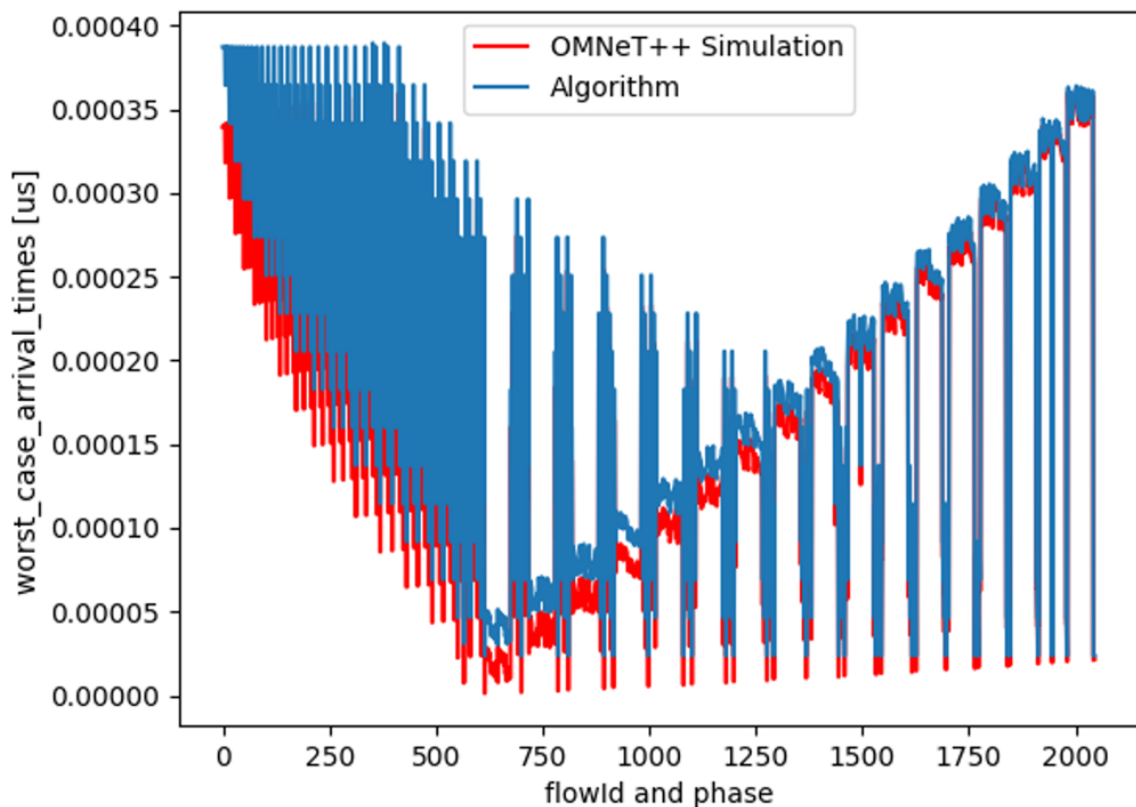


Abbildung 8: Plot der Ankunftszeiten im schlechtesten Fall der 2046 TSN Streams in der Einheit Mikrosekunden auf der y-Achse und nach der größten Differenz zwischen den Ergebnissen aus der Berechnung des CNC Pfadplanungsalgorithmus blau und der OMNEST/OMNeT++ Simulation rot von links nach rechts auf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

