

Modellbasierter Entwurfsassistent zur Auslegung spezifischer Architektur- und Konfigurationseigenschaften von Kommunikationsnetzen mit Echtzeitanforderungen

Gunnar Leßmann

Phoenix Contact Electronics GmbH
Dringenauer Straße 30, 31812 Bad Pyrmont
{glessmann@phoenixcontact.com}

Daniel Schneider, Holger Flatt, Sebastian Schriegel, Jürgen Jasperneite
Fraunhofer-Anwendungszentrum Industrial Automation (IOSB-INA)
Langenbruch 6, 32657 Lemgo

{daniel.schneider, holger.flatt, sebastian.schriegel, juergen.jasperneite@iosb-ina.fraunhofer.de}

Abstract: Industrielle Netzwerke werden immer komplexer durch den Einsatz neuer Technologien wie Time Sensitive Networking (TSN). Die Anforderungen werden zudem verschärft durch sich ändernde gesetzliche Vorgaben wie z.B. bei Smart Grids. Planer solcher Netzwerke haben häufig spezifische Echtzeitanforderungen, Datenraten und Redundanzanforderungen zu beachten, deren Realisierung u.a. abhängig sind von der Leistungsfähigkeit und der Topologie des Netzwerkes. Designalternativen sollten daher quantitativ bewertbar sein, um individuelle Planungsaspekte erweiterbar sein und eine generelle Anwendbarkeit aufweisen. Da diese Kriterien aktuell von keinem verfügbaren Planungswerkzeug erfüllt werden, wurde in diesem Beitrag mit Hilfe von Matlab Simulink® ein generisches und skalierbares Modell von Netzwerkkomponenten erstellt. Diese Komponenten können parametrisiert und in einer anwendungsnahen Topologie kombiniert werden. Leistungsqualifizierende Parameter wie die Echtzeitfähigkeit, Auslastung der Komponenten, Ausfallsicherheit und die Topologie wurden zur Bewertung von Designszenarien eines beispielhaften Windparks herangezogen. Eine Erweiterung des Modells ermöglicht es darüberhinausgehend die Auswirkungen von zukünftigen Technologien, wie z.B. TSN zu analysieren.

1 Einleitung

Die vollständige Vernetzung fast aller unserer Lebensbereiche ist in der aktiven Umsetzung, um das Potential unserer persönlichen und gemeinsamen globalen Ressourcen effektiver einzusetzen. Ob in der intelligenten Produktion (Smart Factory, Industrie 4.0), dem intelligenten Wohnen (Smart Home) oder den intelligenten Energienetzen (Smart Grid), alle intelligenten Anwendungen basieren auf vernetzten Systemen, was auch als Internet of Things bezeichnet wird. Dabei birgt die Vernetzung von technischen Systemen spezifische Anforderungen bzgl. Verfügbarkeit, Echtzeitfähigkeit und Sicherheit. So muss heute z.B. ein Windpark auf Fernsteuersignale (z.B. Einspeiseleitung P , Blindleistungsanteil S , $\cos \varphi$) von Netzleitstellen unterhalb einer Sekunde reagieren [HS11]. Ein Windpark stellt ein verteiltes System mit redundanten Parksteuerungen dar, welche einzelne Windenergieanlagen steuern und regeln. Die Auslegung entsprechender Windparknetze ist komplex, da sich die Vielzahl an gegenseitig beeinflussenden Parametern eine Konzeption erschweren. Anlagenplaner stehen vor der Herausforderung basierend auf Erfahrungswissen oder manuellen Berechnungen solche Netzwerke zu konzipieren.

Diese Parameter erfordern ein effizientes Planungswerkzeug, um den Komplexitätsgrad für den Menschen deutlich zu reduzieren. Eine hohe Vielfalt von Software-Werkzeugen für die Analyse bereits existierender Netzwerke ist am Markt verfügbar, genau wie für das Zeichnen und Dokumentieren von Netzwerken. Für den Entwurf von Kommunikationsnetzen mit spez. Echtzeitanforderungen wiederum existieren keine offen anwendbaren Lösungen. Einzellösungen erlauben unterschiedliche Office-Standardkomponenten (Switches, Hubs, PCs, Router, Server, Drucker) frei zu kombinieren und eine exemplarische Netzwerkauslastung zu simulieren [SW11]. Hochkonfigurierbare und generische Simulationstools, wie z.B. OMNet++, haben den Nachteil einen deutlich zu hohen Detaillierungsgrad aufzuweisen, was die Modellerstellung dementsprechend aufwändig gestaltet [Qu12]. Weiterhin fehlt den genannten Werkzeugen eine Anbindung an funktionale Simulationen (wie z.B. die Auslegung von Reglern, Regelstrategien), wie es für die Auslegung eines Windparks notwendig sein kann. Für den Anlagenplaner sind daher beide Varianten nicht hilfreich, wenn es um die Entwicklung echtzeitfähiger Windparknetzwerke geht. Der Beitrag stellt einen neuen Ansatz vor, der die quantitative Bewertung unterschiedlicher Lösungen und Lastszenarien in einer Simulationsumgebung ermöglicht. Das Verhalten von Komponenten sowie die Abhängigkeiten der

Netzwerkkomponenten können spezifisch für den Einsatz von Echtzeitanwendungen dargestellt und analysiert werden.

In Abschnitt 2 wird der aktuelle Stand der Technik zu Netzwerkplanungswerkzeugen dargestellt. Die Umsetzung und Modellierung eines Netzwerkes zur Analyse wird in Abschnitt 3 erläutert. Abschnitt 4 zeigt anhand eines Beispiels Simulationsergebnisse und leitet daraus verbesserte Netzwerkdesigns ab. In Kapitel 5 wird die Arbeit zusammengefasst und ein Ausblick auf zukünftige Arbeiten gegeben.

2 Stand der Technik

Die Planung von Netzwerken mit vielen Netzwerkknoten, deren Eigenschaften sich gegenseitig beeinflussen, stellt eine komplexe Aufgabe dar, die den Menschen überfordert [Sa09]. An Netzwerke (wie z.B. Windparkvernetzungen) werden leistungsqualifizierende Anforderungen gestellt, wie z.B.: a) Latenz von Signalen, b) Jitter (Streuung der Latenz von Signalen), c) Datenrate pro Zeitintervall und d) Ausfallsicherheit. Um diese Anforderungen zu erfüllen, sollten Netzwerkkomponenten ausgewählt, geeignet vernetzt und konfiguriert werden. Spezifische Topologien, die z.B. über Redundanzmechanismen zur Absicherung der Verfügbarkeit verfügen (z.B. Hiper-Ring, doppelte Leitungsführung, redundante Back-Up-Controller) können erhebliche Auswirkungen auf das Kommunikationsverhalten haben (z.B. eine Verdopplung des Datenverkehrs in Ringen durch HSR). Eine Netzwerkarchitektur wird weiterhin wesentlich durch die angeschlossenen Endgeräte z.B. Steuerungen beeinflusst. Neben der Auslastung der verfügbaren Datenraten sind auch die Leistungsgrenzen der Geräte zu betrachten, wenn spezifische Anwendungen auf diesen ausgeführt werden. Die Auslastungen der CPUs von speicherprogrammierbaren Steuerungen hängt unter anderem davon ab, wie viele Kommunikationsverbindungen zu unterlagerten Geräte existieren und mit welcher Zykluszeit die Kommunikation abläuft. Um die Latenzen und den Jitter zu klassifizieren, sind neben dem Medientyp auch die Distanz zwischen den Geräten zu beachten (bei Windenergieanlagen bis zu 50km). Die zur Verfügung stehende Datenrate (10/100 MBit/s, 1/10/.. GBit/s) sowie deren Auslastung des Mediums hat wiederum Auswirkung auf die Laufzeit und die Stabilität der Weiterleitung von Steuerdaten (ab einer bestimmten Grenzzahl von Datenpaketen überlasten Switches und leiten nicht mehr alle Frames weiter). Zusammenfassend wurden in Tabelle 1 Kriterien aufgelistet, die beim Design von großen Netzwerken Gegenstand intensiver Untersuchungen sein sollten.

Tabelle 1: Anforderungen bei der Netzwerkplanung mit Echtzeitanforderungen

Nr.	Name	Beschreibung
1	Echtzeitanforderungen	Anwendungsanforderungen an Latenz und Jitter, Verarbeitungs-/ Zykluszeiten von Netzwerkgeräten
2	Auslastung der Netzwerkgeräte	Auslastung der Datenrate, CPU Last, Verarbeitbarkeit
3	Ausfallsicherheit	Notwendigkeit von Redundanzmechanismen
4	Topologie	Physikalische Anordnung von Netzwerkgeräten

Um die Qualität und die Realisierbarkeit eines Netzwerkdesigns zu beurteilen, sind Vergleichsmöglichkeiten und quantitative Analysen ein sinnvolles Hilfsmittel für den Netzwerkplaner. Eine hohe Vielfalt von Software-Werkzeugen für die Analyse bereits existierender Netzwerke ist am Markt verfügbar, genau wie für das Zeichnen und Dokumentieren von Netzwerken. Einzellösungen erlauben unterschiedliche Office-Standardkomponenten mit exemplarischen Netzwerkauslastungen zu simulieren [SW11], die aber nicht ausreichend sind, um die oben genannten Anforderungen zu erfüllen. Es wurde aber auch ein spezialisiertes Werkzeug zur Spezifikationsunterstützung von Echtzeitanforderungen bei Software-Systeme entwickelt [Ba05], welches aber spezialisiert auf die Anwendung und nicht die Architektur oder Hardwareerfordernisse ist. Um die Anwendbarkeit des Werkzeugs zu erhöhen, sollte ein Netzsimulator dem Benutzer ermöglichen, eine Netzwerktopologie darzustellen, Szenarien zu definieren und Angaben zu den Knoten im Netzwerk zu treffen. Hierzu werden nachfolgend drei Werkzeuge betrachtet, die potentielle Lösungsformen für das industrielle Umfeld darstellen:

- Ähnlich wie die Simulatoren NS2 und NS3, ist OMNeT++ eine nicht proprietäre Simulationsumgebung, die einen komponentenbasierten Netzwerk-Simulator mit graphischem Nutzerinterface darstellt (Haupteinsatzgebiet sind die Kommunikationsnetze) [6]. Vorteile dieser Simulationsumgebung sind die bereits zahlreich vorhandenen Bibliotheken, die das Verhalten von Standard Netzwerkkomponenten modellieren. Individuelle Modell-, oder Szenarienerweiterungen können in C++ nachträglich eingefügt werden. Generell kann demgegenüber aber festgestellt werden,

dass hochkonfigurierbare und generische Simulationstools, wie OMNet++ den Nachteil haben einen deutlich zu hohen Detaillierungsgrad aufzuweisen, was eine komplexe Modellerstellung zur Folge hat [Qu12]. Weiterhin fehlt den genannten Werkzeugen eine Anbindung an funktionalen Simulationen (wie z.B. die Auslegung von Reglern, Regelstrategien), wie es für die Auslegung eines Windparks notwendig sein kann.

- Das Tool Sinetplan des Unternehmens Siemens ist dafür entwickelt worden PROFINET-Netzwerke vor der Inbetriebnahme zu analysieren und Schwachstellen im Netzwerkdesign zu beheben, aber auch eine Auslastungsüberwachung im Betrieb bereitzustellen. Verfügbare, modellierte PROFINET-basierte Netzwerkressourcen können so in einem Netzwerk geplant werden. Diese können abhängig von der Steuerungslogik auf unterschiedliche Leistungsparameter hin umfangreich untersucht werden. Nachteilig ist, dass die Lösung auf PROFINET zugeschnitten ist und eigene Geräte aufwändig (über Beschreibungsdateien) einzubinden sind. Individuelle Analysen sind in dem Werkzeug nicht vorgesehen.
- Demgegenüber ist der Simulator Matlab Simulink® ein Produkt der Firma Mathworks. Simulink® ermöglicht im Vergleich zu OMNeT++ eine generische, aber nicht spezialisierte, hierarchische Modellierung mit Hilfe grafischer Blöcke. Zusätzliche, komplexere Schaltblöcke können von Matlab aus selbst erstellt oder von Mathworks zugekauft werden. Für die numerische Simulation stehen verschiedene Lösungsverfahren zur Verfügung. Nachteilig an Simulink® ist, dass keinerlei standardisierte Bibliotheken für echtzeitfähige Netzwerkressourcen verfügbar sind.

Für den Anlagenplaner sind daher alle Varianten nur bedingt hilfreich, wenn es um die Entwicklung großer Netzwerke mit Echtzeitanforderungen geht. Da das Werkzeug Matlab Simulink® einerseits eine hohe Modellierungsfreiheit bietet und andererseits die Modellierung einfach skaliert werden kann, wird im Rahmen des Beitrags ein modellbasierter Entwurfsassistent für Netzwerke mit Echtzeitanforderungen mit diesem Werkzeug vorgestellt. Der Aufwand, um grundständige Funktionen von Netzwerkkomponenten abzubilden, wird dabei in Kauf genommen. Dabei wird das Tool Matlab Simulink® in Version 2016a verwendet, was neben der Netzwerkplanung auch eine Reglerauslegung für die Steuer- und Regelaufgaben innerhalb des Netzwerkes ermöglicht.

3 Modellierung

Die Qualität eines guten Modells besteht in der vereinfachten / reduzierten Abbildung der meist komplexen Realität, die es beschreiben soll. Die Methode der Greybox-Modellierung [So03] soll hier angewandt werden, um einerseits wichtige Grundfunktionen im Detail abzubilden, andererseits das Verhalten eines Objektes nur auf sein äußeres Verhalten hin zu betrachten. Für die Simulation soll ein Modell erarbeitet werden, welches in der Lage ist, ein Netzwerk hinsichtlich der in Tabelle 1 definierten Parameter zu analysieren. Speziell für das Beispiel von Windparks ist es notwendig eine Zeitauflösung von mindestens 1ms zu gewährleisten, um die Echtzeitfähigkeit vor allem anhand des Parameter Latenz (s.a. [HS11]) bewerten zu können. Die folgenden Eigenschaften, die ein reales Netzwerk umfassen, können wiederum vernachlässigt werden, da sie für die Betrachtung der genannten Parameter nicht unmittelbar relevant sind:

- Die einzelnen Daten müssen nicht dargestellt werden, wenn die Datenraten Betrachtungsgegenstand der Modellierung sind. Daraus resultiert, dass ebenfalls keine Fehler bei der Datenübertragung betrachtet werden (wie z.B. CRC, Over-/Undersize Frames, Alignment).
- Das lastkritischste Szenario stellt die Many-To-One-Kommunikation in einem Netzwerk (oder Many-To-Many) dar [Ee04]. Hieraus resultiert, dass ein Routing von Datenpaketen von Sendern zu unterschiedlichen Empfängern nicht betrachtet werden muss. Weiterhin wird nur eine Kommunikationsrichtung betrachtet: Die Receive- und Transmit- Kommunikationsverbindungen (RX/TX) sind bei Ethernetverbindungen entkoppelt und im Voll-Duplex-Verfahren geregelt. Zur Vereinfachung wird nur die kritische Richtung der Datenübertragung betrachtet.

Mit diesen Grundannahmen wird nachfolgend die Modellierung von vier Netzwerkkomponenten beschrieben, mit denen unterschiedliche Lastszenarien dargestellt werden können. Bei der nachfolgenden Beschreibung werden grundsätzliche Kenntnisse von Matlab und Simulink® vorausgesetzt.

3.1 Sender

Um einen Sender von Daten zu modellieren, müssen die folgenden Anforderungen erfüllt werden: Das Senden von Daten wird in zyklische und azyklische Kommunikation unterschieden: Zyklische Daten werden in einem fest definierten Zeitintervall mit konstantem Informationsgehalt versendet. Azyklische Kommunikation tritt dagegen einerseits zeitlich unregelmäßig auf und ist andererseits auch in der Menge an

übertragenen Daten stark inhomogen (z.B. das Auslösen eines Alarms) [En07]. Diese Daten werden in der Regel mit einer höheren Priorität als die zyklischen Daten übertragen. Die Priorisierung durch VLAN-Tags ist ein Mechanismus für die Übertragung von wichtigen Datenpaketen - bei Auslastungsproblemen werden höher priorisierte Frames bevorzugt und somit mit geringerer Latenz durchgeleitet.

Modelliert wird dieses Verhalten durch zwei „Timeseries“-Objekte. Hierbei wird das zyklische Datenvolumen in festgelegten Zeitintervallen zur Simulationszeit definiert. Azyklische Kommunikation kann dagegen in einem konfigurierbaren Zeitrahmen in pseudozufallsverteilten Zeitintervallen definiert werden. Die Priorität der Daten kann für beide Kommunikationstypen einmalig festgelegt werden und wird den jeweilig definierten Datenvolumina zugeordnet.

3.2 Leitung

Die generierten Daten des Senders müssen danach über ein Medium zum Empfänger übertragen werden. Um die physikalische Übertragung dieser Daten zu modellieren, müssen die folgenden Anforderungen erfüllt werden: Hierbei spielen im Wesentlichen zwei Effekte eine Rolle: Einerseits entsteht je nach Länge und Medienart der Leitung eine Transportverzögerung. Andererseits ist das Datenvolumen, welches pro Zeitpunkt übertragen wird, unter Umständen kleiner, da z.B. das Medium belegt ist oder unterschiedliche Datenraten unterstützt werden. In diesem Fall werden die Daten fragmentiert und konkateniert, d.h. in Datenpakete zerteilt und nacheinander abgesendet. Diese Auslastung des Mediums wird mit der Netzlast umschrieben: Netzlast ist definiert als Szenario, in dem eine äußere Last einen Prozentsatz $X\%$ der verfügbaren Netzwerk Datenrate nutzt. Diese Last kann zufällig verteilt sein innerhalb eines Millisekundenintervalls. Bezeichnet wird die mittlere Last in diesem Intervall. Hierdurch werden 100MBit/s zu 12,5kByte/ms und 1GBit/s zu 125kByte/ms (Beispiel: 30% Netzlast in einem 100MBit/s Netzwerk führen zu 3,75kByte Rohdatenverkehr pro ms) [PR15]. Die mittlere Lastbeaufschlagung durch Datenpulse/Bursts ist dabei hauptsächlich limitiert durch die Taktfrequenz auf der Leitung (Maximum durch Shannon-Hartley-Gesetz begrenzt) [Pro03].

Modelliert wird dieses Verhalten durch Standardelemente von Simulink® und individuellen Funktionsblöcken. Die Transportverzögerung kann durch einen Block „Transport Delay“ dargestellt werden. Anhand eines Parameters kann die Verzögerung eingestellt werden. Die Netzlast wurde demgegenüber durch eine individuelle Matlab basierte Funktion modelliert. Hierbei wird zunächst überprüft, wie hoch die Auslastung des Mediums bereits ist. Neue Daten werden, falls eine vollständige Auslastung bereits erreicht wurde, fragmentiert und zum nächst möglichen späteren Simulationszeitpunkt übermittelt. Alternativ ist es möglich, bereits hier eine priorisierte Datenübertragung mit unterschiedlichen Queuingmechanismen zu benutzen (s.a. Abschnitt 3.3 Switch) oder die überschüssigen Daten zu verwerfen und nicht zu übertragen. Diese maximal zulässige Netzlast sowie das Verhalten des Endknotens können frei konfiguriert werden.

3.3 Switch

Werden von mehreren Sendern Daten über eine Leitung übertragen, müssen sich diese das gemeinsame Medium Kabel teilen. Bei der Um- und Weiterleitung von Daten treten Latenzen auf, die bei der Verarbeitung hervorgerufen werden. Um die Datenübertragung mehrerer Sender zu modellieren, müssen die folgenden Anforderungen erfüllt werden: Das Weiterleitungsverhalten eines Switches wurde bereits in einer Studie [Ja07] experimentell ermittelt und konnte in zwei unterschiedliche Varianten unterschieden werden (vgl. Abbildung 1). Das „Strictly Priority Queueing“ beschreibt ein Verhalten, bei dem bei maximaler Last ausschließlich die höchstpriorären Frames übertragen werden. Demgegenüber werden beim „Weighted Fair Queueing“ alle Frames aus den Queues übertragen, allerdings wird hochpriorären Daten mehr Bandbreite zur Verfügung gestellt.

Dieses Verhalten wurde jeweils modelliert und durch strikt lineares Verhalten bis zur Lastgrenze angenähert. Nach dieser Lastphase wird die priorisierte Verarbeitung der Daten je nach Strategie gemäß der Darstellung durch eine Finite-State-Machine modelliert. Die jeweilige Queuing-Strategie kann im Modell für jede Komponente individuell eingestellt werden.

Neben der Priorisierung von Daten stellt die Weiterleitezeit vom Eingangsport zum Ausgangsport eine wesentliche Eigenschaft eines Switches dar. Diese Zeit hängt von mehreren Parametern ab (Linkspeed, Traffic Pattern, Port Numbers, ..), vor allem aber von der Paketgröße. Weiterhin wird außerdem entweder ein Cut-Through oder Store-and-Forward Verfahren als Switching Strategie angewandt. In einem Whitepaper wurden diese Eigenschaften miteinander verglichen und die Auswirkung auf die Latenz der

Datenweiterleitung gemessen (siehe Abbildung 2) [Ci12]. Diese Arbeit basiert auf 10Gbit Switches mit entsprechend hohen Taktraten (GMII 500MHz).

Überträgt man dieses Verhalten auf die niedriger getaktete 1Gbit Leitung steigt die Latenz um den Faktor 15,6. Diese Latenz ist bezogen auf die 1ms-Skalierung des Modells vernachlässigbar klein. Da in diesem Modell nicht die Framegrößen modelliert wurden, ist dieser Mechanismus außerdem nicht direkt abbildbar. Um Switching Einflüsse dennoch annähernd betrachten zu können, wird nachfolgend eine einstellbare, aber über die Simulationszeit konstante, Verzögerung von 0,1ms angenommen. Diese Annäherung kann solange als zulässig angesehen werden bis die Skalierung des Modells nicht unter 1ms fällt.

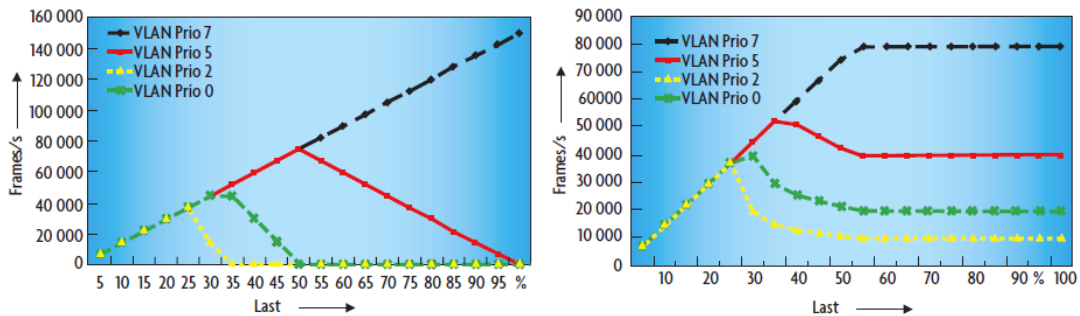


Abbildung 1: Switching Strategien: links- Strictly Priority Queueing, rechts- Weighted Fair Queueing [Ja07]

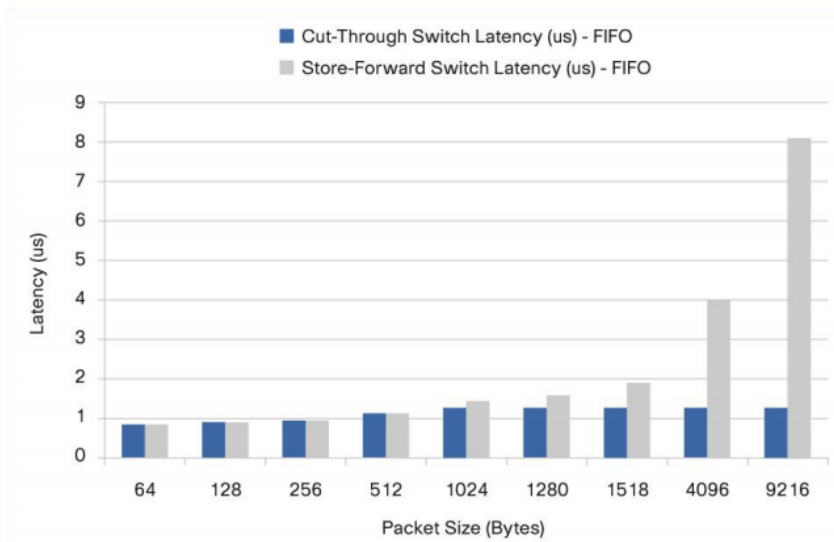


Abbildung 2: Gemessene Latenz im Vergleich zwischen Cut-Through und Store-And-Forward Switching Strategien bei 10Gbit/s [Ci12]

3.4 Steuerung

Um eine Steuerung (Controller) zu modellieren, müssen die folgenden Anforderungen erfüllt werden: Der Controller enthält eine Logik, die Eingangsdaten verarbeitet und Ausgangsdaten generiert. Dabei ist die Last der CPU des Controllers in Abhängigkeit der angeschlossenen Geräte und der jeweiligen Aktualisierungszeit zu beachten. In praktischen Aufbauten sind Switches den Controllern vorgeschaltet, die nur die spezifischen Frames zum Controller weiterleiten. Würde dies nicht beachtet, ergäbe sich nach der Grundannahme ohne ein Routing eine deutlich zu hohe Lastberechnung.

Modelliert wird dieses Verhalten durch die Verwendung von Lastkurven, die im Vorfeld bei unterschiedlichen Komponenten und Lastfällen ausgemessen wurden. Es werden hierzu im Modell die Anzahl unterschiedlicher Signale gezählt und in Abhängigkeit von den jeweiligen Aktualisierungszyklen in einer Finite-State-Machine ausgewertet, in der die Linearisierungen der Lastkurven hinterlegt sind. Die Zykluszeiten werden dabei nicht im Modul gemessen, sondern den ursprünglichen Parametrierungen der Sender entnommen (s.a. Abschnitt 3.1 Sender). Es kann außerdem eine maximale CPU-Last eingestellt werden, ab der der Controller überlastet ist und keine Frames mehr verarbeitet. Die Datenübertragung bricht dann für diesen spezifischen Controller simuliert ab, solange bis die Last wieder unter die vorgegebene Grenze fällt.

Zur korrekten Berechnung der Last ist außerdem ein Routing von Signalen notwendig. Daher wurden individuelle Routing-Funktionen dem Modell hinzugefügt, welches spezifische Frames, die von der hierarchisch übergelagerten Instanz kommen filtert und zum Controller leitet.

Wie bereits erläutert sendet ein Controller auch Daten an angeschlossene Geräte. Hierbei wurde das Modell des Senders im Controller erweitert, so dass die generierten Daten vervielfacht werden für gleichartige Gerätefamilien. So können auf einfache Weise auch große Mengen gleichartiger angeschlossener Geräte angesprochen werden. Individuelle Geräte können durch weitere Sender ins Controller Modell integriert werden.

Die Verarbeitungszeit eines Controllers ist für die Analyse der Echtzeit von erheblicher Bedeutung für die Modellierung. Zu beachten ist, dass Controller in Zyklen arbeiten. Hierbei wird angenommen, dass es auf der Applikationsebene einen Prozess-Zyklus (Process-Cycle) gibt, der Daten verarbeitet. Ein IO-Zyklus wiederum empfängt und versendet Daten. Eine Logik (auch Task genannt) verarbeitet Daten innerhalb eines Zeitintervalls Process-Cycle (z.B. eine Regelung). Je nach Länge der einzelnen Tasks werden alte Daten mehrfach übertragen, da die Zyklen nicht miteinander synchronisiert laufen. Zu berücksichtigen ist, wie diese Zyklen zueinanderstehen und welche Auswirkung das auf die Durchleitezeit neuer Daten hat [Vo11]. In Abbildung 3 ist beispielhaft ein IO-Zyklus (IO-Cycle) dargestellt, bei dem neue Daten unmittelbar nach Beginn eines IO-Zyklus am Gerät ankommen. Die neuen Daten werden somit im zweiten IO-Zyklus übernommen. Diese neuen Daten sind wiederum unmittelbar nach Beginn eines Prozesszyklus verarbeitet. Dies bedeutet demnach, dass die neuen Daten erst im zweiten Prozesszyklus verarbeitet werden. Nachdem die Daten verarbeitet wurden, benötigt es genau wie auf der Empfangsseite zwei IO-Zyklen, um die Daten zu versenden. Dies führt zu einer Worst-Case-Betrachtung bei der Datenverarbeitung in Controllern, die in Abbildung 3 dargestellt ist.

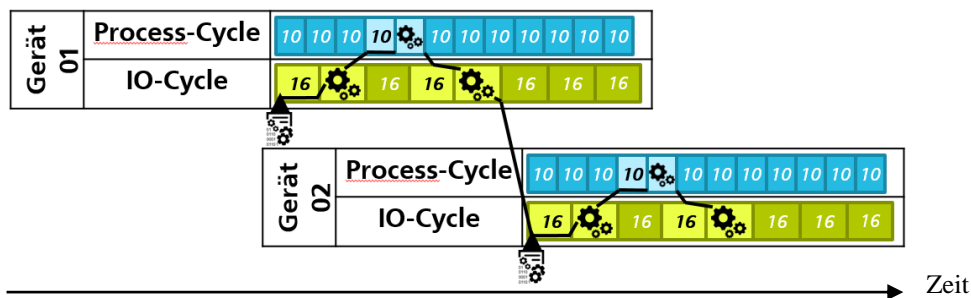


Abbildung 3: Worst-Case-Betrachtung für Durchleite- und Verarbeitungszeiten neuer Daten bezogen auf freilaufende Prozess und IO-Zyklen in einem Controller

Wird eine Synchronisation umgesetzt, können die Geräte derart betrieben werden, dass der IO-Zyklus neue Daten aus dem Prozess unmittelbar und ohne Zeitverzögerung übernehmen kann.

Modelliert wurde dieses Verhalten durch einstellbare Process- und IO-Zyklen, die eine Gesamtverzögerung pro Controller ergibt, die durch einen Block „Transport Delay“ umgesetzt wird. Hierbei kann weiterhin eingestellt werden, ob die Zyklen synchronisiert oder vollständig freilaufend sind. Eine Zeitplanung mehrerer Geräte, wie sie durch Zeitsynchronisationsmechanismen wie IEEE1588 ermöglicht werden, können hier ebenfalls durch eine einstellbare Abstimmung der Sender in Ihrer Auswirkung analysiert werden.

4 Fallstudie

4.1 Netzwerktopologie

Mit diesen vier Elementen lassen sich unterschiedliche Topologien aufbauen und modellieren. Es lassen sich so die Netzwerkkomponenten z.B. von Windparks im Verbund simulieren. Hier wurde eine Topologie gewählt, bei der zentrale Steuerungen zur Gesamtregelung (ZSPS) in einem Ring angeordnet sind. Diese Steuerungen werden vom Netzbetreiber (NB) mit Informationen zur Netzleistung versorgt. Wiederum regelt jede Steuerung eine Untermenge von Windenergieanlagensteuerungen (WSPS) (im Ring angeordnet), an die jeweils ein Hauptumrichter (HU) angeschlossen ist. Diese Topologie ist in Abbildung 4 dargestellt.

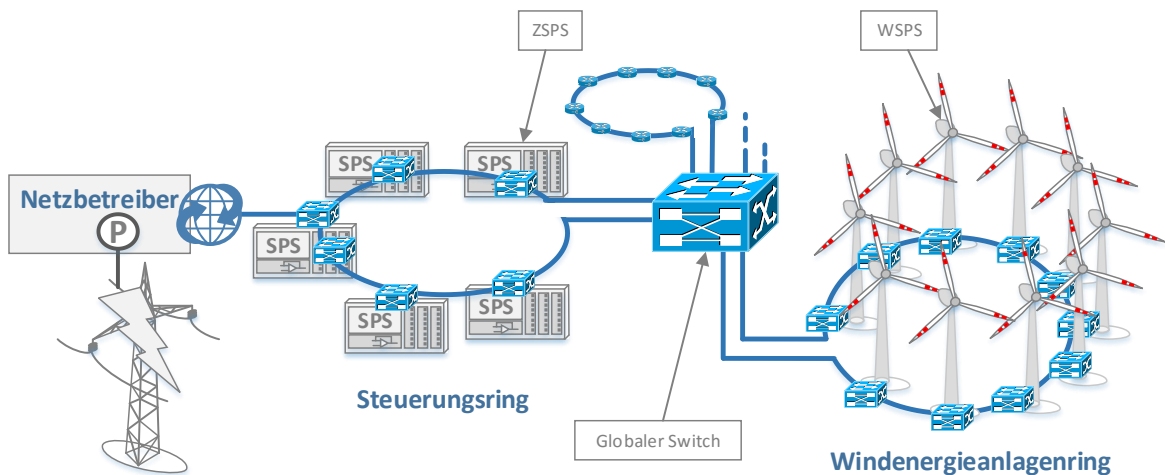


Abbildung 4: Zu untersuchende Topologie vom Netzbetreiber bis zur Windenergieanlage

Relevant für die Modellierung ist hier der kritische Pfad. Dazu werden die Ringe geöffnet und zu einer Linie ausgerollt. Um das kritische Steuersignal darstellen zu können, wird ein Sender modelliert, der zyklisch den ZSPS Daten übermittelt. Eine ZSPS wird als Controller modelliert, der in diesem Fall aber zusätzlich mit den WSPS kommuniziert. In diesem Steuerungsring aus ZSPS befindet sich ein globaler Switch, der den Datenverkehr an mehrere Ringe zu WSPS aufteilt. Zur Vereinfachung wurde nur ein Switch und eine Leitung modelliert, welches stellvertretend für die Verarbeitungseigenschaften von 10 WSPS parametrisiert wurde. Die Daten empfangende WSPS wurde als Controller modelliert, dem noch ein Hauptumrichter nachgelagert ist. Die HU stellt in Simulink® einen „Scope“ dar, der Datenströme historisch loggt und visualisiert.

4.2 Analyseergebnisse

Die in Abbildung 5 und 6 dargestellte Übersicht eines betrachteten Windpark-Szenarios soll nachfolgend analysiert werden. Die Überprüfung der Echtzeiteigenschaften einer zeitkritischen Datenübertragung vom Netzbetreiber bis zur HU ist von zentraler Bedeutung für die Bewertung eines Netzwerkdesigns (genannt CEP). In Abbildung 5 wird dieses anhand von zwei Datenaufnahmen durch ein Logging direkt am Sender und am Empfänger dargestellt. Der zeitliche Unterschied von 98ms zwischen den blau dargestellten Peaks (kritisches Datensignal) zeigt die Latenz innerhalb des Netzwerkes auf. Treibender Faktor für die vorliegende Latenz stellen vor allem die Verarbeitungszeiten in den ZSPS und in den WSPS dar. Zu beachten ist auch die Form des Signals, das beim HU ankommt. In geringem Maße weicht dieses vom Ursprungssignal ab. Diese Änderung entsteht, da an einer Stelle im kritischen Pfad eine vollständige Auslastung vorliegt und den Frames priorisiert nur noch Teile der Bandbreite zur Verfügung gestellt werden. Übertragungswege mit kritischen Bandbreitenauslastungen können unter anderem durch die Veränderung der Topologie verbessert werden. In Abbildung 6 wird daher die Bandbreite des Steuerungsringes, der im GBit-Betrieb arbeitet, dargestellt. Bei 5 ZSPS wird die Bandbreite in Spitzen bis zu 100% ausgelastet, was deutlich über der empfohlenen Grenze von 1/3 der Bandbreite liegt [PR10]. Eine Kommunikation könnte hier Störungen unterliegen. Neben topologischen Veränderungen kann auch der Einfluss einer zeitlichen Ablaufplanung betrachtet werden. Konkret bedeutet diese Aussage, welche Bandbreitenauslastung zu erwarten ist, wenn die Controller zu aufeinander abgestimmten Zeiten Daten senden. In Abbildung 7 ist dieser beschriebene Fall dargestellt. Die maximale Auslastung liegt hier nur noch bei ca. 24% und damit innerhalb der empfohlenen Auslastung. Kommunikationsschwierigkeiten sind hier nicht mehr zu erwarten.

Die Auslastung der CPUs aller Controller ist eine zweite wichtige Kenngröße die nachfolgend betrachtet werden soll. Die weiteren Simulationsergebnisse zeigen, dass die ZSPS zu 74,1% CPU Auslastung in Überlast arbeiten, die Auslastung der WSPS mit 15,6% nur sehr gering ausgelastet ist.

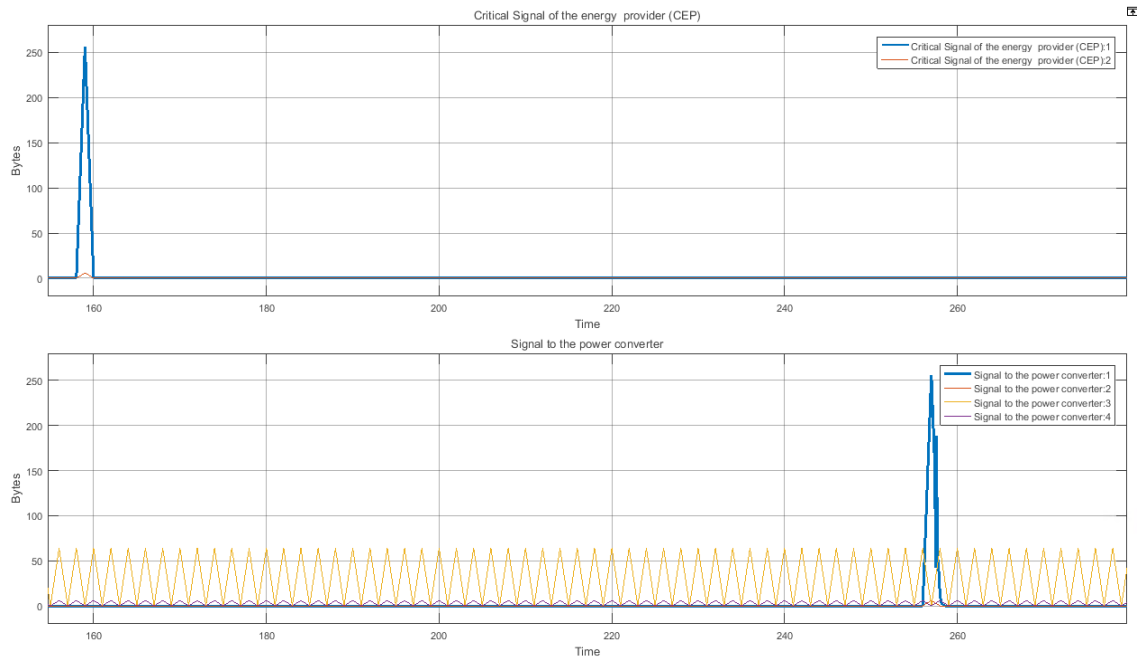


Abbildung 5: Signallaufzeit eines kritischen Datensignals vom Netzbetreiber bis zur HU

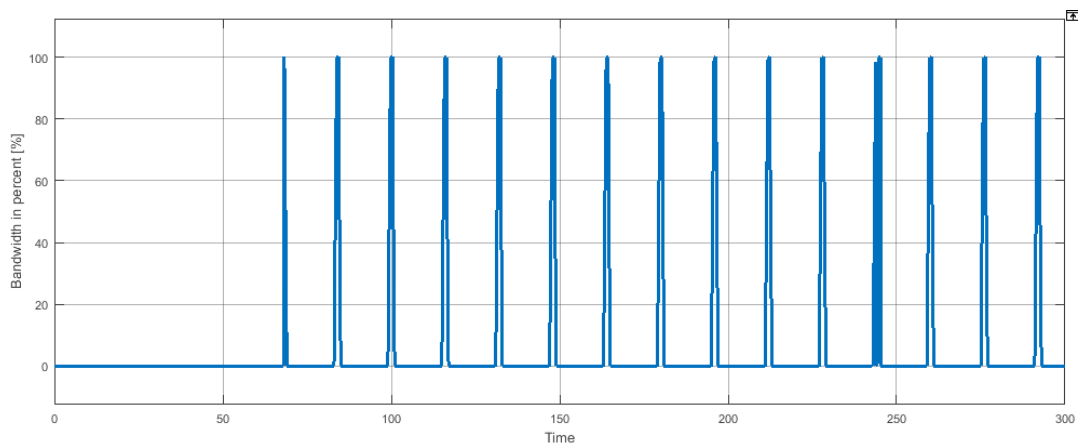


Abbildung 6. Bandbreitenauslastung im Steuerungsring im Fall bei dem alle ZSPS gleichzeitig Daten senden

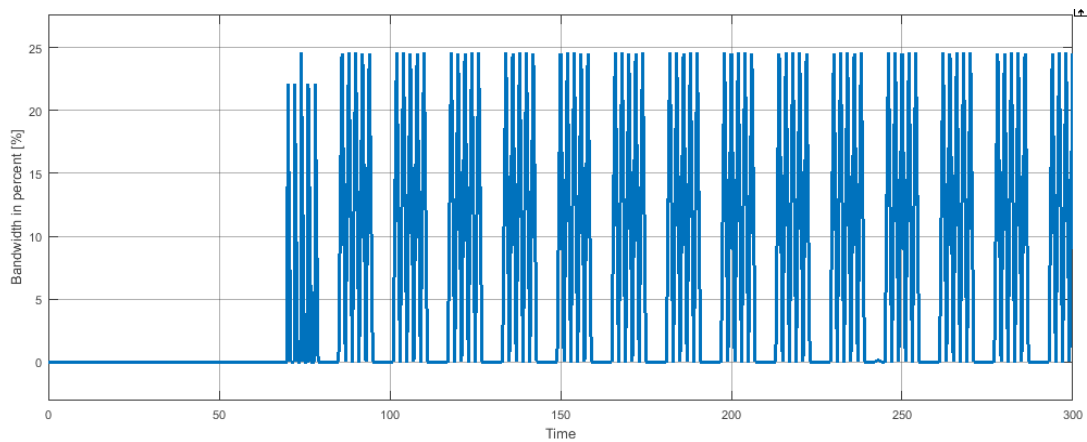


Abbildung 7: Bandbreitenauslastung im Steuerungsring im Fall bei dem die ZSPS zeitlich geplant Daten versenden

Es sollte untersucht werden, ob man mittels eines Zeitschlitzverfahrens, wie es in der Echtzeit-Ethernet-Lösung Profinet IRT verwendet wird, ein verbessertes Kommunikationsverhalten im Steuerungsring ermöglichen kann. Die Datenrate müsste dazu auf 100MBit/s reduziert werden, was bei einer

Synchronisation der Sendezeitpunkte aber unproblematisch ist. Die Nutzdaten des IRT-Rings müssten dann an den jeweiligen Windenergieanlagenring weitergeleitet werden. Zu berücksichtigen ist, dass in diesem Fall die Anzahl von Geräten auf 64 begrenzt wäre und so eine umfassende Anwendbarkeit nicht möglich ist.

5 Zusammenfassung und Ausblick

In der vorliegenden Studie wurde am Beispiel von Windparks dargestellt, warum es notwendig ist, Netzwerke mit Echtzeitanforderungen anhand von Kennzahlen bewertbar auszulegen und Designalternativen zu vergleichen. Der Stand der Technik wurde hinsichtlich geeigneter Methoden zur Unterstützung des Entwicklungsprozesses dieser komplexen Netzwerke analysiert. Die identifizierten Werkzeuge bieten entweder eingeschränkt anwendbare oder nur aufwändig erweiterbare Lösungen an. Mit der hier entwickelten Lösung einer generischen Matlab Simulink® Modellierung ist es möglich diese Nachteile zu kompensieren. Zur quantitativen Bewertung unterschiedlicher Lösungen wurde ein Modell erstellt, das es ermöglicht das Kommunikationsverhalten sowie die Abhängigkeiten der Netzwerkkomponenten untereinander darzustellen und effizient zu analysieren. Es wurde das äußere Verhalten von Netzwerkkomponenten und in Einzelfällen auch die inneren Mechanismen modelliert. Analysen eines hiermit simulierten Windparks geben Aufschluss über die Leistungsfähigkeit des erstellten Planungswerkzeugs. Es konnte gezeigt werden, dass gesetzliche Rahmenanforderungen an Smart-Grids nur mit leistungsfähigen Controllern in bestimmten Topologien einzuhalten sind. Insgesamt konnte mit der Studie durch eine Simulation eine belastbare Aussage über die Lastsituation, sowie mögliche Lösungen bei der Kommunikation innerhalb eines Windparks gegeben werden.

Mittels dieses Werkzeuges ist es möglich miteinander kommunizierende, komplexe und verteilte Systeme zu entwickeln und bewerten. Neue Technologien wie Time Sensitive Networking (TSN) [Sch16], Software Defined Networking (SDN) und Network Function Virtualization (NFV) sind Beispiele für weitere Einsatzfelder dieses Werkzeugs. Die technologischen Vor- und Nachteile dieser neuen Kommunikationslösungen könnten mit Hilfe dieses Werkzeugs modelliert und abgeschätzt werden. Wie stark die Echtzeitfähigkeit einer Anlage durch die Einführung von TSN verbessert wird, könnte am konkreten Anwendungsbeispiel überprüft werden. Werden Netzwerkfunktionen durch eine Cloud konfiguriert und verwaltet, ergeben sich neue und komplexe Anforderungen an bestehende und zu planende Netzwerke. Mittels des hier vorgestellten modellbasierten Entwurfsassistenten zur Auslegung von Netzwerken mit Echtzeitanforderungen könnten mit einer aufbauenden Modellierung auch Zukunftstechnologien auf Ihre Anwendbarkeit und Auswirkung hin bewertet werden.

6 Literaturverzeichnis

- [Ba05] Bauer, A.; Romberg, J.; Schätz, B.: "Integrierte Entwicklung von Automotive-Software mit AutoFOCUS", Inf. - Forsch. Entw., vol. 19, no. 4, pp. 194-205, 2005.
- [Ci12] Cisco Systems Inc., „Understanding Switch Latency,“ June 2012. [Online]. Available: http://www.cisco.com/c/en/us/products/collateral/switches/nexus-3000-series-switches/white_paper_c11-661939.pdf
- [Ee04] Ee, C.T.; Bajcsy, R.: „Congestion control and fairness for many-to-one routing in sensor networks“, SenSys '04 Proceedings of the 2nd international conference on Embedded networked sensor systems, Pages 148-161, 03.11.2004
- [En07] Enste, U.; Müller, J.: „Datenkommunikation in der Prozessindustrie – Darstellung und anwendungsorientierte Analyse“, ISBN 978-3-8356-3116-8, Oldenbourg Industrieverlag GmbH, 2007, München
- [HS11] Hau, M., Shan, M.: „Windparkregelung zur Netzintegration“. IWES, 2011. Kasseler Symposium Energie-Systemtechnik, Kassel, 2011. S. 109-120.
- [Ja07] Jasperneite, J.; Gamper, S., „Echtzeit-Betrieb im Ethernet - Industrial Ethernet-Switches auf dem Prüfstand“, Elektronik, pp. 60-65, 16/2007
- [Le16] Lehnhöfer, E., „Permanent online – Windparks dauerhaft verschlüsseln und intelligent überwachen“, *messtec drives Automation*, März 2016
- [PR15] PROFIBUS Nutzerorganisation: „PROFINET IO Security Level 1“, *Guideline for PROFINET*, June 2015.
- [PR10] Profibus Nutzerorganisation e.V., „PROFINET Design Guideline - Profibus,“ 11 2010.
- [Pro03] Proakis, J.G., Salehi, M.: “Communication Systems Engineering”, ISBN 0130950076, Prentice Hall; 28. April 2003
- [Sa09] Sandom, C.; Harvey, R.S.: „Human Factors for Engineers“, Institution of Engineering and Technology, ISBN 978-0-86341-329-2, London, 2009

- [Sch16] Schneider, D.; Flatt, H.: „Time Sensitive Networking (TSN)“. In: Deutsche Forschungsgesellschaft für Automatisierung und Mikroelektronik e.V. (DFAM), Online voraus. <http://www.dfam.de/service-studien-handbuecher.php>, November 2016
- [So03] Sohlberg, B.: “Grey box modelling for model predictive control of a heating process”, Journal of Process Control, Volume 13, Issue 3, Pages 225–238, April 2003
- [SW11] Spiceworks, Inc.;“Network Mapping Software”, Patent US 20110145056 A1, 16 Juni 2011, <http://www.spiceworks.com/free-network-mapping-software/>
- [Vo11] Vogel-Heuser, B.; Feldmann, S.; Werner, T.; Diedrich, C.: „Modeling network architecture and time behavior of Distributed Control Systems in industrial plant automation”, IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society, Melbourne, VIC, 2011, pp. 2232-2237.
- [Qu12] Qutaiba I. Ali (2012). Simulation Framework of Wireless Sensor Network (WSN) Using MATLAB/SIMULINK Software, MATLAB - A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 2, Prof. Vasilios Katsikis (Ed.), InTech