



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Seminararbeit

Marco Munstermann

Datenaustausch in der Prozessautomatisierung  
Teil 2

## Inhaltsverzeichnis

1	Einleitung .....	1
2	Ethernet in der Prozessautomatisierungstechnik.....	2
2.1	Industrial Ethernet .....	2
2.2	Realtime Ethernet .....	2
3	Applikationsschnittstellen .....	7
3.1	OPC - OLE for Process Control .....	7
3.1.1	OPC Data Access (OPC DA) .....	8
3.1.2	OPC Alarms and Events (OPC AE) .....	8
3.1.3	OPC Historical Data (OPC HDA) .....	9
3.1.4	OPC Data eXchange (OPC DX) .....	9
3.1.5	OPC XML DA .....	9
4	Fragestellung für eine Masterarbeit.....	10
5	Zusammenfassung .....	10
6	Literaturverzeichnis.....	12
6.1	Bibliografie.....	12
6.2	Cybergrafie.....	12
7	Anhang .....	13
7.1	Technische Daten der verschiedenen Realtime-Verfahren .....	13
7.2	Echtzeitvergleich der verschiedenen Realtime-Verfahren .....	14

## Abbildungsverzeichnis

<b>Abbildung 1</b>	Kommunikationspyramide .....	2
<b>Abbildung 2</b>	Zeitschlitzverfahren (TDMA) .....	4
<b>Abbildung 3</b>	Zyklus mit 4 Perioden (Polling) .....	5
<b>Abbildung 4</b>	Aufbau eines Ethernet/IP Netzwerks .....	6
<b>Abbildung 5</b>	Von einer Vielzahl proprietärer Schnittstellen zum OPC-Standard.....	7
<b>Abbildung 6</b>	Verketteter Datenzugriff mittels OPC DX .....	9
<b>Abbildung 7</b>	Klassifizierung der Realtime Ethernet-Verfahren.....	10

## 1 Einleitung

Diese Seminararbeit beschäftigt sich im Wesentlichen mit dem Einsatz von Ethernet auf der Feldbus-Ebene und der Vereinheitlichung von Applikationsschnittstellen. Mit diesen Voraussetzungen kann die vertikale Integration von der Steuerungs- / Regelungsebene bis hinauf in die Managementebene verwirklicht werden. Dieser Arbeit vorangegangen ist bereits ein erster Teil, der einen Einstieg ins Thema gibt und sich dann den seriellen Bussystemen, namentlich ASI und CAN, widmet. Es ist ratsam den vorangegangenen Teil vor diesem zu lesen, da er Begriffe erklärt, die auch in diesem Dokument gebraucht werden.

Ziel dieser Dokumentation ist es, zunächst einen kleinen Überblick über aktuelle Entwicklungen in der Prozessautomatisierung zu geben. Danach werden erkannte Probleme benannt und mögliche Lösungsansätze diskutiert. Sämtliche Betrachtungen sind im Kontext „Seminar Ringvorlesung“ in Hinblick auf die Masterarbeit entstanden.

Die Anfänge der Kommunikation in der Prozessautomatisierung gehen zurück auf die 60-er Jahre. Bereits damals wurden die verteilt arbeitenden SPS'en über einzelne Leitungen verbunden. Hierzu schloss man den digitalen Ausgang einer SPS an den Eingang einer anderen. Auf diese Weise schaffte man bereits sehr früh Echtzeitkommunikation. Im Gegensatz zum industriellen Markt dauerte es auf dem kommerziellen Markt bis Ende der 70-er Jahre bis Ethernet vorgestellt wurde.

Da diese frühe Form der Echtzeitkommunikation insbesondere bei größeren Installationen schnell unhandlich wurde, suchte man hinsichtlich des Verkabelungsaufwandes nach einer einfacheren Lösung: der Feldbus. Bis heute haben sich verschiedenste Implementierungen von Bussystemen auf dem Markt etabliert. Zum einen auf Grund der unterschiedlichen Anforderungen, zum anderen aber auch herstellerbedingt. Viele Hersteller versuchen immer noch „ihr“ proprietäres Kommunikationssystem am Markt zu festigen. Doch inzwischen werden aus Interoperabilitätsgründen offene Bussysteme gewünscht.

Dieses steht ganz im Gegensatz zum kommerziellen Markt. Dort hat sich das Ethernet als einheitlicher und von allen akzeptierter Standard durchgesetzt. Hohe Stückzahlen haben einerseits einen raschen Preisverfall der Interfaces zur Folge, andererseits findet dadurch ein enormer Technologiefortschritt statt. Hersteller von Hard- und Software versprechen sich großen Absatz.

In jüngster Zeit zeichnet sich ab, dass der industrielle Markt auch von diesem technischen Fortschritt profitieren möchte. Idealerweise bietet sich als Basis der Einsatz der gleichen Hardware wie auf dem kommerziellen Markt an. Hinsichtlich der verwendeten Kabel, Stecker und Gehäuse sind verständlicherweise für das raue Industrieumfeld Anpassungen notwendig. Im ersten Abschnitt sollen die verschiedenen am Markt befindlichen Lösungsansätze untersucht werden.

Bestanden dezentrale Steuerungen klassischerweise aus zwei Kommunikationssystemen, so waren die Ebenen nicht nur logisch, sondern auch physikalisch von einander getrennt. Nicht zuletzt durch die Globalisierung sind Unternehmen gezwungen, sich bezüglich ihrer Kommunikation zu öffnen. Die eingesetzte Software muss sich schließlich diesen neuen Herausforderungen stellen. Die Konzepte hierzu gehen auf Überlegungen von General Motors (GM) zurück. GM machte sich bereits 1980 die Vorteile der Objektmodellierung zunutze, wie [MMS] beschreibt.

Aktuell wird diese Idee vom Grundsatz her von der OPC (OLE for Process Control) Foundation unter Verwendung von anderen Technologien verfolgt.

Im zweiten Abschnitt dieser Arbeit wird der „OPC- Gedanke“ vorgestellt.

## 2 Ethernet in der Prozessautomatisierungstechnik

### 2.1 Industrial Ethernet

Der Wunsch des industriellen Marktes vom Fortschritt der Ethernettechnologie zu profitieren, legt nahe auf vorhandene und erprobte Hardware zurückzugreifen. In der aus dem ersten Teil bekannten Kommunikationspyramide (Abbildung 1) wurde bereits deutlich, dass Büro- und Fabrik-Netzwerk heute in der Regel auf Ethernet basieren. Der Einsatz von Ethernet, wie es im Büroumfeld verwendet wird, verbietet sich auf Grund der Einsatzbedingungen in der Industrie (vgl. Teil 1 und [BUSSYS]).

Industrial Ethernet ist seit Ende der 90-er Jahre ein stark diskutiertes Thema. Mit Industrial Ethernet hat man den physikalischen Randbedingungen der Industrie Rechnung getragen. Es entstand ein Kommunikationssystem auf Basis von Ethernet, welches industrietaugliche Hardware verwendet. Hierzu werden spezielle Gehäuse und Kabel eingesetzt. Die Gehäuse müssen bezüglich der IP-Schutzart (IP = International Protection) den Umgebungen entsprechen. Dies umfasst zum einen den Schutz gegen Eindringen von Festkörpern, zum anderen den Schutz gegen das Eindringen von Wasser. Bei den Kabeln kann ein erhöhter Schutz gegen aggressive Stoffe (z.B. Säuren) oder der Schutz gegen elektromagnetische Störungen (EMV) erforderlich sein. Besonders wichtig in diesem Zusammenhang sind adäquate Steckverbindungen. Untersuchungen zu Folge seien in 90% der Fälle Probleme beim Betrieb von Netzwerken im industriellen Einsatz auf die Anschlusstechnik zurückzuführen (siehe hierzu [ETHIND], S. 25).

Im Unterschied zu dem Ethernet aus dem Büroumfeld (RJ-45) gibt es bei Industrial Ethernet keine einheitlichen Steckverbindungen. Jeder Hersteller bietet seinen eigenen Stecker an. Dies schränkt die Kompatibilität von Systemen unterschiedlicher Hersteller ein.

Erschwerend kommt hinzu, dass die Anbieter von Industrial Ethernet Steuerungen / Regelungen in Ermangelung eines Standards ihre proprietären Applikationsprotokolle verwenden.

Es ist festzustellen, dass Industrial Ethernet keineswegs die Lösung aller Probleme darstellt, wie dies so häufig zu hören ist. Industrial Ethernet ist lediglich ein Sammelbegriff proprietärer Kommunikationssysteme, die auf Basis von Ethernet industrietaugliche Hardware verwenden.

### 2.2 Realtime Ethernet

Trotz der Feststellungen aus dem vorangegangenen Kapitel besitzt Ethernet-TCP/IP die potenzielle Eignung auf Feldbus-Ebene eingesetzt zu werden. Vergleicht man die Übertragungsraten, die Stand der Technik auf Feldbusebene sind (z.B. von LIN mit 20 kbit/s bis TTP/C mit 25 Mbit/s), mit denen, die bei Ethernet üblich sind (z.B. von Fast Ethernet IEEE-802.3u mit 100 Mbit/s bis Gigabit-Ethernet IEEE-802.3z mit 1000 Mbit/s), so stellt man fest, dass Ether-

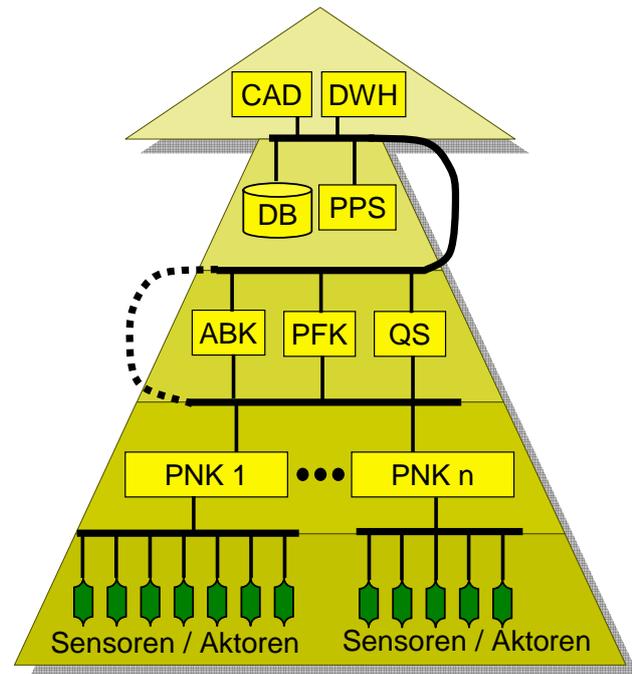


Abbildung 1 Kommunikationspyramide

net deutlich höhere Übertragungsraten zulässt. Mit TCP erhält man eine korrekte Datenübertragung die - wie im ersten Teil gesehen - von großer Bedeutung ist.

Natürlich bleiben Begrenzungen und Einschränkungen erhalten. So ist der Einsatz von Ethernet in explosionsgefährdeten Bereichen nicht möglich, da es im Gegensatz zum Profibus PA nicht eigensicher ist. Des Weiteren werden kommunikationsstörende Knoten nicht wie bei CAN erkannt und gegebenenfalls abgeschaltet. Aus diesem Grund beeinträchtigt ein störender Knoten bis zu seiner Stilllegung das gesamte Netzwerk. Bestimmte Anwendungsgebiete (z.B. Motion Control) stellen hohe zeitliche Anforderungen an das Kommunikationssystem. Diese kann Ethernet aber nicht erfüllen, da es in realen Anwendungen einem Jitter unterliegt, der größer als 1 ms ist. Diese erheblichen Zeitschwankungen entstehen hauptsächlich durch Kollisionen von Nachrichten. Diese werden zwar erkannt (Collision Detection), aber nicht vermieden (Collision Avoidance), wie z.B. bei CAN.

Die Kommunikationspyramide ist in einen unteren Teil mit (harten) Echtzeitanforderungen und in einen oberen Teil ohne Echtzeitanforderungen geteilt. Will man die beiden Segmente mit Ethernet verbinden, so sind spezielle Maßnahmen zu ergreifen, die die Kollisionen verhindern.

Das Bindeglied zwischen den beiden Segmenten soll zwar den Datenaustausch erlauben, muss aber auch dafür sorgen, dass das Echtzeitsegment nicht überlastet wird, damit es seinen zeitlichen Anforderungen gerecht werden kann.

Es folgen nun drei Lösungsansätze mit den jeweiligen Veränderungen gegenüber Ethernet-TCP/IP. Zudem werden die zugehörigen Implementierungen genannt. Die Einteilung der Lösungsansätze orientiert sich an dem OSI-Layer auf dem der Eingriff stattfindet.

#### 1. Lösungsansatz – Eingriff auf dem MAC-Layer

Der erste Lösungsansatz greift bereits auf der OSI-Schicht 2a ein. Anstelle von CSMA/CD sind zwei Ausprägungen des Zugriffsverfahrens zu finden:

- Token passing

*Funktionsprinzip:* Zunächst sind alle Busteilnehmer, zumindest logisch in einem Ring anzuordnen. Im gesamten Netzwerk wird nur ein Telegramm - der Token - versendet. Erreicht der Token einen Teilnehmer, so liest bzw. schreibt derjenige seine Prozessdaten in das Telegramm und reicht es an den nächsten im Ring weiter.

Der Token unterteilt sich in zwei Abschnitte: einen für den Realtime- und einen für den allgemeinen Verkehr. Im vorderen Abschnitt des Tokens (Realtime) befindet sich für jeden Teilnehmer ein Subtelegramm. Darin enthalten sind ein Header und die Prozessdaten des jeweiligen Slaves. Durch die Komprimierung der einzelnen Subtelegramme zu einem größeren Ethernet-Telegramm, wird die Übertragungs-Performance, das Verhältnis von Nutzdatenrate zu Overhead, erhöht. Diese beschreibt die Anreicherung mit Subtelegrammen erhöht die Nutzdatenrate bei gleich bleibendem (Ethernet-)Overhead.

Der hintere Abschnitt des Tokens enthält die allgemeinen Daten. Hier können beispielsweise Ethernet-TCP/IP Telegramme getunnelt werden. Überschreitet ein solches Telegramm die zulässige Größe für einen Zyklus, also die Zeit die benötigt wird bis der Token erneut einen Teilnehmer erreicht, so ist es auf mehrere Zyklen aufzuteilen.

*Implementierung:* Die technische Umsetzung dieses Verfahrens nennt sich „EtherCAT“ und wird weltweit von der EtherCAT Technology Group bzw. dessen über 170 Mitgliedsfirmen vertreten, darunter u. a. ABB, Bosch, NI und Philips.

EtherCAT spezifiziert ein Telegramm, bei dem der Ethertype statt auf IP auf EtherCAT gesetzt wird. Bei EtherCAT erfolgt die gesamte Protokollabarbeitung einschließlich des Tunnelns von Ethernet-TCP/IP Telegrammen in spezieller Hardware (ASICs

oder FPGAs, je nach Hersteller). Dies erlaubt sehr kurze Zykluszeiten. Die Verbindung zwischen dem Realtime- und dem Standard-Ethernet-Segment findet über einen speziellen Switch mit Gateway-Funktionalität statt.

- Time Division Multiple Access (TDMA)

*Funktionsprinzip:* Es handelt sich hierbei um ein Zeitschlitzverfahren. Die zur Verfügung stehende Bandbreite, nicht wie beim Token passing das Telegramm, wird in zwei Zeitbereiche aufgeteilt, wie man Abbildung 2 entnehmen kann, einen für den isochronen Datenverkehr und einen für den asynchronen Datenverkehr.

Im isochronen Zeitbereich werden Realtime-Daten versendet. Hierzu hat jeder Teilnehmer einen eigenen Zeitschlitz (TS = time slot), in dem er seine Prozessdaten verschicken kann. Der zweite Zeitbereich steht den asynchronen Daten zur Verfügung. So können darin beispielsweise Ethernet-UDP/IP Telegramme transportiert werden.

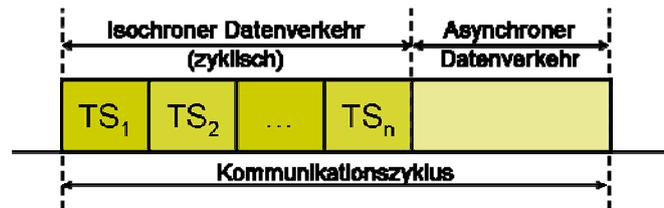


Abbildung 2 Zeitschlitzverfahren (TDMA)

*Implementierungen:* Es existieren zwei technische Umsetzungen dieses Verfahrens: „Profinet v.3“ wird von der PROFIBUS Nutzerorganisation vertreten. Ihr gehören u. a. ABB und Siemens an. Profinet v.3 konforme Teilnehmer werden über spezielle integrierte Switches (zwei oder vier Ports) untereinander verbunden.

„Sercos-III“ ist durch die Interessengemeinschaft SERCOS vertreten. Bei Sercos-III wird bewusst auf Hubs oder Switches verzichtet. Jeder Teilnehmer besitzt zwei Interfaces. Auf diese Art und Weise wird die Topologie auf Linen- und Baumstrukturen eingeschränkt. Es ergeben sich dafür sehr kurze Zykluszeiten. Es ist ein spezieller Gateway von Nöten, um das Realtime-Segment mit der Standard-Ethernet-Welt zu verknüpfen.

Es ist festzustellen, dass bei allen Verfahren, die unter dem 1. Lösungsansatz vorgestellt wurden, spezielle Hardware in den Netzknoten erforderlich ist. Zur Anbindung des Echtzeit-Segments an ein Standard-Ethernet Netzwerk sind zudem spezielle Gateways erforderlich. Diese proprietären Kommunikationssysteme verfehlen den Wunsch der Industrie vom Technologiefortschritt und Preisverfall des kommerziellen Marktes direkt zu profitieren. Bei einer Weiterentwicklung der Ethernet-Technik kann die Hardware nicht (ohne Änderungen) verwendet werden. Dennoch gibt es Anwendungsgebiete (z.B. Robotersteuerungen), deren Anforderungen so hoch sind, dass sie gegenwärtig nur durch proprietäre Systeme erfüllbar sind.

2. Lösungsansatz – Eingriff auf dem LLC- Layer

Beim zweiten Lösungsansatz findet der Eingriff erst auf OSI-Schicht 2b statt. Das CSMA/CD-Verfahren wird beibehalten. Diesem übergeordnet wird ein Zeitschlitzverfahren, welches mittels eines zentralen Pollings realisiert ist.

*Funktionsprinzip:* Es gibt einen Master, der die beteiligten Slaves zyklisch pollt (vgl. Teil 1, ASI). Ein Zyklus ist in vier Zeitbereiche unterteilt. In der Start Period wird ein Start of Cyclic Frame (SOC) vom Master aus mittels Broadcast an alle Slaves verschickt. Die Slaves können sich auf den Beginn des neuen Zykluses synchronisieren. Im anschließenden Zeitbereich, Cyclic Period genannt, wird jeder Slave nacheinander separat durch den Master gepollt (CP = Cyclic Poll).

Das Pollen findet in einem festen Zeitraster statt. Der gepollte Slave erlangt so das Senderecht. Ein Slave antwortet mit einem Broadcast und kann so ohne Umweg alle angeschlossenen Slaves sowie den Master mit Prozessdaten versorgen (CS = Cyclic Send).

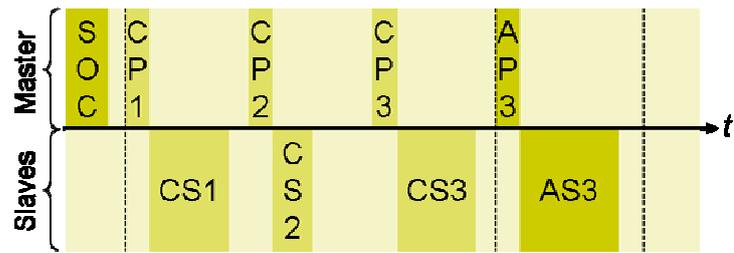


Abbildung 3 Zyklus mit 4 Perioden (Polling)

Zusammen mit seiner Antwort kann der Slave für den nun folgenden Abschnitt „Asynchron Period“ eine Sendeberechtigung beantragen. Dieser Mechanismus ist beispielhaft in

Abbildung 3 dargestellt. Dort beantragt der Slave mit der Nummer drei die Sendeberechtigung für den asynchronen Teil des Zyklusses. Nach dem azyklischen Poll (AP = Acyclic Poll) durch den Master folgt der nicht deterministische Teil der Kommunikation (z.B. über TCP/IP). In diesem Zeitabschnitt darf immer nur ein Slave seine azyklischen Daten (AS = Acyclic Send) versenden.

Die verbleibende Zeit (Idle Period) bis zum Beginn des neuen Zyklusses ist variabel und hängt von der jeweiligen Ausnutzung der zur Verfügung stehenden maximalen Sendezeit in der azyklischen Periode ab. Aus der Sendezeit und der Übertragungsrate leitet sich die Paketlänge ab.

*Implementierung:* Das vorgestellte Verfahren wird in Form von Ethernet-Powerlink(EPL) umgesetzt. Standardisiert wird EPL durch die Ethernet Powerlink Standardization Group (EPSG). Es besteht ein enger Kontakt zwischen der EPSG und der CiA (siehe Teil 1). Ziel dieser Zusammenarbeit ist die Möglichkeit einer einfachen Integration von EPL und CANopen. EPL basiert auf CANopen-Profilen, sodass eine Vielzahl von CANopen-Geräten in einer EPL-Umgebung eingesetzt werden kann (vgl. [IEC-05]). ABB, Atmel und IXXAT sind nur drei der über 120 Mitgliedsfirmen.

EPL spezifiziert Telegramme, bei denen der Ethertype statt auf IP auf Powerlink gesetzt wird. Die Teilnehmer des Echtzeitsegmentes können über Standard-Ethernet Hubs verbunden werden, was durch die Beibehaltung des CSMA/CD-Verfahrens möglich wird. Ein Gateway stellt den Übergang zum Standard-Ethernet Segment her. Er muss seitens des Realtime-Segments den gleichen Kommunikationsstack wie die übrigen Teilnehmer verwenden. Daher ist die Verwendung von Standard-IP Routern nicht möglich. Weil der Gateway als Teilnehmer des Echtzeitsegmentes auftritt, kann er sowohl als Master wie auch als Slave fungieren. In der Rolle des Slaves würde er in der Cyclic Period lediglich die Reservierung für den azyklischen Teil durchführen. Dagegen könnte der Gateway als Master selbstständig entscheiden wann Telegramme aus dem Standard-Ethernet Segment in das Echtzeitsegment gelangen.

Das vorgestellte zentrale Polling, wie es bei EPL zum Einsatz kommt, verwendet Standard-Hardware in den Netzknoten. Die Veränderung gegenüber TCP/IP bezieht sich auf den Kommunikationsstack. Damit erreicht man mit EPL hardwareseitig vom Technologiefortschritt und Preisverfall des kommerziellen Marktes direkt profitieren zu können. EPL ist z.Z. für den Fast Ethernet Standard IEEE 802.3 spezifiziert. In Hinblick auf Gigabit-Ethernet könnten daher Anpassungen notwendig werden.

### 3. Lösungsansatz – Eingriff oberhalb von TCP/IP

Der dritte Lösungsansatz belässt den ISO/OSI Kommunikationsstack einschließlich der Schicht vier unangetastet. Im Gegensatz zu den übrigen Lösungsansätzen ist dieser nicht zyklusbasiert, sondern nachrichtenorientiert bzw. zeitbasiert.

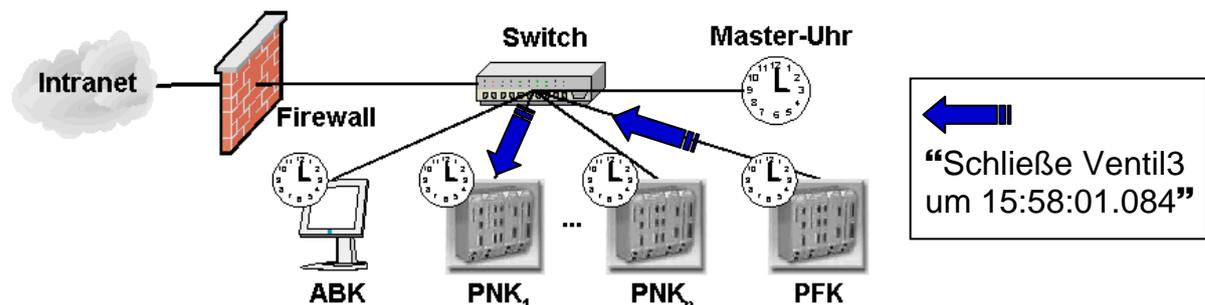
*Funktionsprinzip:* Man macht sich drei aus dem Bereich der Netzwerktechnologie bekannten Verfahren zu nutze. Zum einen ist dies der konsequente Einsatz von Switches (full duplex). Damit eliminiert man Kollisionen auf dem Übertragungsmedium. Mit Hilfe der in den Switches implementierten Store-and-Forward Queues werden Nachrichten, die „gleichzeitig“ an ein und denselben Port zu übertragen sind, serialisiert.

Zum anderen unterscheidet man zeitkritische Daten von zeitunkritischen auf Grund ihrer Priorität (Quality of Service). Auf diese Art und Weise können zeitkritische Daten zeitunkritische in den Queues überholen. Drittens entkoppelt man die Performance des Gesamtsystems von der des Kommunikationssystems indem man jeden Teilnehmer mit einer Echtzeituhr versieht und die Agenda-Technik anwendet.

*Implementierung:* Die Umsetzung dieses Konzeptes findet man unter dem Namen Ethernet/IP (Industrial Protocol) wieder. Ethernet/IP wird durch die Open DeviceNet Vendor Association vertreten, der u.a. ABB, Bosch und Rockwell angehören.

Ethernet/IP spezifiziert nur Standard-Ethernet Telegramme. Realtime-Daten unterscheiden sich von zeitunkritischen Daten auf Grund ihrer höheren Priorität. Anstelle von TCP setzt man auf der Transportschicht UDP ein.

Zur Synchronisierung der Echtzeituhren in den Netzknoten kommt der nach IEEE 1588 normierte Algorithmus zum Einsatz. Auf Anwendungsschicht erhalten alle Steuerbefehle eine Ausführungszeit (Agenda-Technik). Damit wird sichergestellt, dass solange die Nachrichten vor der Ausführungszeit beim Empfänger eintreffen, zeitliche Schwankungen des Kommunikationssystems keinen Einfluss auf das Gesamtsystem haben. Man erreicht wie bei den übrigen Verfahren Jitter von unter einer Millisekunde (siehe Echtzeitvergleich der verschiedenen Realtime-Verfahren).



**Abbildung 4** Aufbau eines Ethernet/IP Netzwerkes

Einen beispielhaften Aufbau eines Ethernet/IP Netzwerkes zeigt Abbildung 4. Hatte ein Steuerbefehl zuvor die Bedeutung: „Schließe Ventil3 jetzt“, gemeint ist mit Eintreffen der Nachricht, so könnte er nun wie in Abbildung 4 dargestellt lauten.

Das Konzept von Ethernet/IP sowohl hardware- als auch softwareseitig nur auf vorhandene Standards zu setzen, verspricht größtes Kosteneinsparungspotential. Die verwendete Hardware ist jahrelang erprobt und in ausreichenden Mengen verfügbar. Selbst die Anbindung an das firmeneigene Intranet, wie in Abbildung 4 dargestellt, kann über einen Standard-Router mit entsprechend konfigurierter Firewall erfolgen. Es werden ausschließlich Standard TCP/IP- und UDP/IP- Protokoll-Stacks verwendet.

Außerdem erreicht man maximale Flexibilität. Je verbreiteter der Standard, desto größer ist schließlich die Herstellerauswahl.

Um eine Aussage wie „Schließe Ventil 3 um 15:58:01.084“ machen zu können, müssen die Steuerbefehle „vorhersehbar“ sein. In reaktiven Systemen ist dies häufig nicht der Fall. Dagegen sind in der Verfahrensautomatisierung, man denke beispielsweise an eine Abfüllanlage, die meisten Ausführungszeiten berechenbar.

Bei dem im Anhang befindlichen Echtzeitvergleichstest resultierten für Ethernet/IP unter Verwendung von CIPSync – einem Anwendungsprotokoll für Ethernet/IP – bei 100 Achsen und einer Übertragungsrate von 100 Mbit/s Reaktionszeiten von deutlich über einer Millisekunde. Damit ist Ethernet/IP für (Fast) Motion Control nicht geeignet. Diese Einschränkung schwächt sich ab mit dem Einsatz von Gigabit-Ethernet.

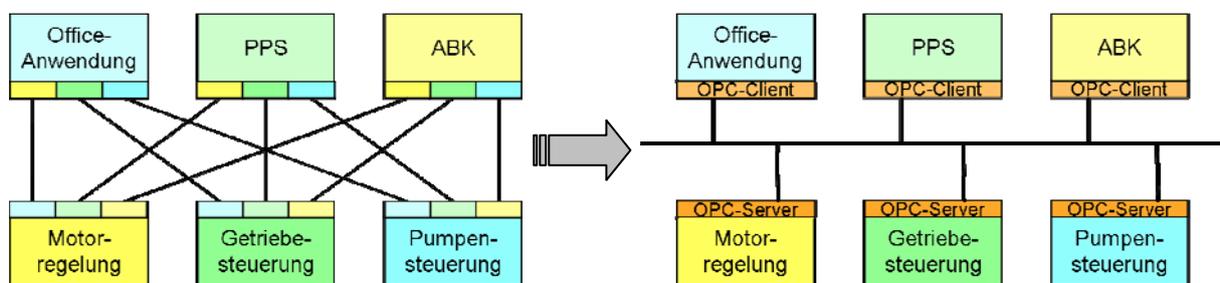
Die Daten aller vorgestellten Realtime-Verfahren finden sich zusammengefasst im Anhang wieder (siehe Technische Daten der verschiedenen Realtime-Verfahren).

### 3 Applikationsschnittstellen

Wenn die in Abbildung 1 gestrichelt dargestellte Verbindung von Fabrik- und Feldbus-Netzwerk mit den im Kapitel Realtime Ethernet vorgestellten Lösungsansätzen technisch realisierbar wird, entsteht der Bedarf einer adäquaten Applikationsschnittstelle für dieses Szenario. Mit Ausnahmen des letzten Lösungsansatzes beschränken sich die Verfahren auf die Kommunikation unterhalb von OSI-Schicht vier. Es fehlt folglich an einer standardisierten Applikationsschnittstelle in der Automatisierungstechnik, um die vertikale Integration über die Prozessleitebene hinweg zu betreiben. Hier setzt OPC an.

#### 3.1 OPC - OLE for Process Control

Der Verwendungszweck von OPC lässt sich an einem einfachen Beispiel erklären (siehe Abbildung 5). Angenommen auf Feldebene existieren drei Geräte: eine Motorregelung, eine Getriebe- und eine Pumpensteuerung. Auf Prozessleitebene stehen dem eine Office-Anwendung, ein Produktionsplanungs- und Steuerungssystem (PPS) und eine Anzeige- und Bedienkomponente (ABK) gegenüber. Diese sind nun vollständig zu einem Gesamtsystem zu integrieren. Im schlimmsten Fall muss für jede Komponente auf einer Ebene ein Treiber pro Komponente auf der anderen Ebene erstellt werden. Somit resultieren in diesem kleinen Beispiel 18 Treiber.



**Abbildung 5** Von einer Vielzahl proprietärer Schnittstellen zum OPC-Standard

Vergleicht man dies mit dem OPC-Ansatz auf der rechten Seite der Abbildung, so wird deutlich welche Vereinfachungen für die Entwicklung eines solchen Systems damit verbunden sind. Die OPC-Clients bzw. -Server müssen vom Kunden (Firmen, die Automatisierungstechnik einsetzen) nicht programmiert sondern nur konfiguriert werden. Daraus resultieren geringere Engineeringkosten des Gesamtsystems.

Auf Grund der Interoperabilität der OPC-Komponenten untereinander werden die Kunden von der bislang herrschenden Herstellerabhängigkeit befreit. Oft sind Entscheidungen für ein bestimmtes Bussystem noch lange Zeit in Form ihrer entsprechenden Konsequenzen merklich.

Wie bereits im Beispiel gesehen, verwendet OPC das Client-Server-Modell. OPC benutzt dazu Microsofts Distributed Component Model (DCOM), welches ehemals OLE (Object Lin-

king and Embedding) hieß (siehe hierzu [MSAUT] oder [OPCGRUND]), als Middleware. Dabei definiert OPC Kommunikationsobjekte, nicht aber die jeweilige Implementierung.

OPC soll die einfache vertikale Integration, wie sie heutzutage verlangt wird, ermöglichen. Von vertikaler Integration verspricht man sich die Optimierung der logistischen Abläufe durch vollständige Kommunikation aller beteiligten Informationssysteme. Dabei fordert man hohe Durchlaufgeschwindigkeiten der Aufträge, Effizienz und Fehlerfreiheit, um größtmögliche Kostenminimierung zu erreichen.

Werden heute Daten vereinzelt noch über den Papierumweg transportiert, so soll mit OPC ein kontinuierlicher maschineller Datenaustausch zwischen den einzelnen Hierarchieebenen der Kommunikationspyramide erreicht werden. Diese Vision drückt sich durch den Ausdruck „vom Office bis zum Sensor“ aus.

Daten aus einem Automatisierungssystem sollen in ein externes PC-System transferiert und dort zur Verfügung gestellt werden. So ist es denkbar einen aktuellen Temperaturwert eines Sensors automatisch in ein Excel-Sheet zu übernehmen, um z.B. ein Report des Herstellungsprozesses anzufertigen.

Die OPC-Foundation definiert eine Reihe von verschiedenen OPC Spezifikationen mit unterschiedlichen Aufgaben. Es folgt ein Überblick der wichtigsten OPC Spezifikationen, deren Verwendung im Anschluss vorgestellt wird:

- OPC Data Access
- OPC Alarms and Events
- OPC Historical Data
- OPC Data eXchange
- OPC Data Access XML

### 3.1.1 OPC Data Access (OPC DA)

Mit OPC DA wird eine standardisierte Schnittstelle zur Prozessdatenkommunikation definiert. Damit ermöglicht man ein bis n Clients den transparenten Zugriff auf einen OPC DA Server, der wiederum Zugriff auf die realen Prozessdaten hat. Hierzu abonniert ein DA-Client die Daten, für die er sich interessiert, beim DA-Server. Dieser sendet dem DA-Client die angeforderten Prozessdaten kontinuierlich zu.

Ein solches Datum ist ein Tripel bestehend aus: Wert, Zeitstempel und Statusinformation. Hinter Wert verbirgt sich dabei beispielsweise bei einem Sensor die eigentliche Messgröße. Der Zeitstempel gibt Auskunft über die Aktualität des Wertes. Die Statusinformation zerfällt in drei Teile: Qualität, Status und Limit. Dabei kann die Qualität „Good“, „Bad“ oder „Uncertain“ sein. Der Status dient im Wesentlichen der genaueren Beschreibung der Qualität. Steht Qualität auf „Bad“ so könnte Status „Not Connected“ die Ursache des Problems erläutern. Limit enthält im Falle eines Sensorfehlers weitere Informationen.

Zusätzlich sind optionale Attribute vorgesehen. Denkbar wäre ein Geräte-Name oder die Standardabweichung eines Sensors.

Ein DA-Client legt im Server (bzw. dessen Speicher) mittels der in dieser Spezifikation festgelegten Methoden Objekte an. Diese bestimmen über die Art und Weise wie die Daten strukturiert sind. Somit kann sich der Client seine „eigene“ Sicht auf die Daten erstellen.

OPC DA ist die am häufigsten implementierte Spezifikation und hat folglich den Charakter des kleinsten gemeinsamen Nenners beim Zusammenschluss von mehreren OPC-kompatiblen Geräten.

### 3.1.2 OPC Alarms and Events (OPC AE)

Eine nach OPC AE spezifizierte Schnittstelle dient dem Übertragen und Quittieren von Ereignissen und Alarmen. Hierzu muss ein AE-Client die Alarme und Ereignisse, die er überwacht

haben möchte, beim AE-Server abonnieren. Im Gegensatz zu der OPC DA Spezifikation werden Ereignisse und Alarmer *diskontinuierlich* gesendet. Der Client legt per Konfiguration des Servers die gültigen Wertebereiche - wiederum in Form von Objekten - fest.

Dabei übernimmt der OPC AE-Server die Überwachung des jeweiligen Datums und generiert ggf. Ereignis-/ Alarmmeldungen. Eine solche Alarmnachricht besteht aus **Ursprung**, **Zeitstempel**, **Meldung**, **Priorität** und weiteren Attributen. Der Bedeutung nach entspricht der Zeitstempel dem von OPC DA. Der Ursprung gibt an wo der Alarm / das Ereignis aufgetreten ist. **Meldung** enthält den entsprechenden Klartext. Über **Priorität** kann die Dringlichkeit des Alarms ermittelt werden.

### 3.1.3 OPC Historical Data (OPC HDA)

In manchen Fällen sind weniger die aktuellen, sondern vielmehr die vergangenen Daten einer bestimmten Zeitspanne von Interesse. Hierzu hat man die OPC HDA spezifiziert. Mittels OPC HDA werden standardisiert historische Prozessdaten übertragen.

Diese werden weder kontinuierlich noch diskontinuierlich, sondern nur auf Anforderung übertragen. Zudem können Aggregate wie z.B. Mittelwerte erstellt werden. Ein OPC HDA-Client kann die Minima oder Maxima einer definierten Zeitspanne anfragen. Die Mechanismen der Konfiguration über die im Server angelegten Objekte entsprechen der Funktionalität nach denen von OPC DA und OPC AE.

Bedenklich scheint die Möglichkeit historische Daten nicht nur lesen, sondern auch einfügen, ersetzen und löschen zu können. Möchte man einen OPC HDA-Server als Protokollinstanz für z.B. die Überwachung einer Kühlkette in der Lebensmittelindustrie nutzen, so darf nur lesender Zugriff auf diese Daten zugelassen werden.

### 3.1.4 OPC Data eXchange (OPC DX)

War es bislang immer zwingend erforderlich strikt nach dem Client-Server-Modell zu modellieren, so bietet OPC DX die Möglichkeit der Server-Server-Kommunikation.

In komplexen Systemen nutzen Server andere Server als Datenquellen. So kann beispielsweise ein OPC DX-Server einen OPC DA-Server als externe Quelle nutzen und dessen Prozessdaten importieren. Die importierten Daten werden daraufhin im eigenen Namensraum angeboten. Vereinfacht ausgedrückt kann man sagen, dass ein OPC DX-Server im Grunde genommen die Summe aus DA-Client und DA-Server ist.

Eine Verkettung von DX-Servern ist erlaubt (siehe Abbildung 6). Je nach Länge der Kette können erhebliche Verzögerungen zustande kommen, sodass ein Auswerten des Zeitstempels unerlässlich ist.

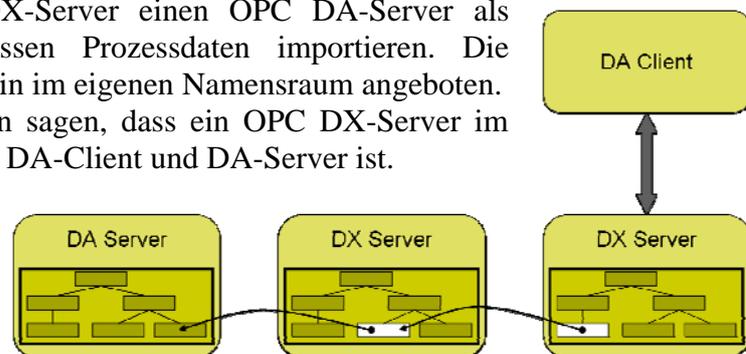


Abbildung 6 Verketteter Datenzugriff mittels OPC DX

### 3.1.5 OPC XML DA

Die OPC XML DA Spezifikation entspricht in seiner Funktionalität der OPC DA Spezifikation. Im Unterschied zu OPC DA basiert OPC XML DA nicht auf DCOM, sondern auf XML und SOAP. Diese Veränderung löst OPC von Microsofts Betriebssystemen, die ansonsten nötig sind um mit DCOM arbeiten zu können, und macht es Plattform unabhängig.

Ein weiterer Vorteil ist die resultierende Möglichkeit den Einsatzbereich auf das Internet zu erweitern. Dies ist kritisch, da die Spezifikation in ihrer jetzigen Form keinerlei Sicherheitsaspekte beinhaltet.

Dem gegenüber steht der Nachteil, dass XML deutlich langsamer (etwa um den Faktor 7, siehe hierzu [OPCGRUND], S. 250) ist als OPC DCOM DA. Eine Ursache hierfür ist die ineffiziente Art wie XML Daten versendet. Eine weitere Ursache ist, dass zum Parsen der XML-Nachrichten zusätzlich Rechenzeit benötigt wird.

Generell kann man sagen dass, durch den Einsatz von OPC die Engineeringkosten gesenkt werden können. Statt wie im obigen Beispiel 18 Treiber schreiben zu müssen / lassen, sind nur noch die OPC Komponenten zu konfigurieren. Damit hat OPC DA die Kunden von der Geräteherstellerabhängigkeit befreit. Seit etwa einem Jahr (18. Dezember 2004) befreit OPC XML DA die Anwender von der Microsoft-Abhängigkeit. Ein wichtiger Schritt, wenn man bedenkt, dass Microsoft ein Unternehmen ist, welches davon lebt kontinuierlich neue Produkte auf den Markt zu bringen. Dies steht im Kontrast zum Investitionsschutz, den die Prozessautomatisierung durch langlebige Produkte sicherstellen muss.

OPC basiert auf TCP/IP, also einem nicht deterministischen Kommunikationsprotokoll. Definierte Antwortzeiten können nicht garantiert werden.

#### 4 Fragestellung für eine Masterarbeit

Im FAUST- Projekt lassen sich die Regelungen (Antriebs-, Lenkregelung), Steuerungen (Kordinierungsrechner) und eine Anzeige- und Bedienkomponente (Leitstand) aus der Kommunikationspyramide wieder finden. Als Feldbus-Netzwerk kommt der CAN-Bus zum Einsatz. Da die Bedienung des Fahrzeuges drahtlos erfolgen sollte, wurde der CAN-Bus über WLAN (IEEE 802.11g) mit der ABK gekoppelt.

Der CAN-Bus ist ein Vertreter der deterministischen Kommunikationssysteme. Dahingegen stammt WLAN aus der Gruppe der nicht-deterministischen Kommunikationssysteme.

Im Projekt lebt man z. Z. mit den Konsequenzen. Treffen die Sollvorgaben vom Leitstand nicht rechtzeitig auf dem Fahrzeug ein, so wird das Fahrzeug in einen sicheren Zustand versetzt. Man könnte aber fragen:

Lassen sich die Echtzeitansätze für Ethernet auch auf WLAN übertragen, um eine deterministische Kommunikation zwischen dem Fahrzeug und dem Leitstand im FAUST- Projekt zu realisieren?

Dies würde eine programmtechnische Umsetzung des gewählten Echtzeitansatzes beinhalten. Im Anschluss könnten die erzielten Ergebnisse analysiert werden.

#### 5 Zusammenfassung

Im ersten Abschnitt wurden die verschiedenen am Markt befindlichen Realtime Ethernet-Verfahren untersucht und mit den jeweiligen Implementierungen in Verbindung gebracht. Profinet v.3, Sercos-III und EtherCAT sind zyklusbasierte Implementierungen, die sowohl hard- als auch softwareseitig proprietär sind, wie sich aus Abbildung 7 entnehmen lässt. Es lassen sich mit ihnen sehr kurze Zykluszeiten erreichen, die für bestimmte Anwendungsgebiete zwingend erforderlich sind. Aufgrund des Eingriffs auf OSI-Schicht 2a kann keine Standard-Ethernet Hardware verwendet werden.

Mit EPL wurde eine Implementierung vorgestellt, die Standard-Hardware im Echtzeitsegment ver-

Software (OSI-Schicht 3 und 4)	Standardkonform	Ethernet/IP
	Proprietär	Profinet v.3 Sercos-III EtherCAT
		Proprietär      Standardkonform
		<b>Hardware</b>

Abbildung 7 Klassifizierung der Realtime Ethernet-Verfahren

wenden kann. Die Anpassung in Hinblick auf Echtzeitfähigkeit findet in der Software statt. Die zuletzt beschriebene Implementierung setzt ganz auf Netzwerk-Standards auf. Das Netz wird nach bestimmten Kriterien konzipiert. Auf Anwendungsschicht wird ein spezielles zeitbasiertes Protokoll eingesetzt.

Zurzeit basieren alle Realtime- Verfahren auf dem Fast Ethernet Standard (IEEE 802.3u). Setzt man stattdessen auf Gigabit-Ethernet (IEEE 802.3z), so lässt sich die Netzleistung um den Faktor zehn steigern. Für die Anwendung im Realtime-Segment heißt dies, dass sich die Reaktionszeiten auf ein Zehntel reduzieren.

Mit OPC wird dann im zweiten Abschnitt eine in der Prozessautomatisierung weit verbreitete Applikationsschnittstelle dargestellt. Mit ihr wird die vertikale Integration von der Management- bis zur Steuerungs- / Regelungsebene möglich. Die Öffnung der firmeneigenen Datenetze zum Internet erfordert zusätzliche Sicherheitsmaßnahmen, um Beeinflussungen des Produktionsprozesses zu verhindern.

Ordnet man die OPC- Geräte einer Ebene der Kommunikationspyramide (Abbildung 1) zu, so findet man sie bereits auf der Feldebene wieder. In Zukunft möchte man jedoch noch weiter nach unten durchgreifen können. Es existiert die Vision vom „OPC- Sensor“. Ein solcher „intelligenter“ Sensor (etwa als OPC DA-Server realisiert) könnte jede Steuerung, die sich für seine Werte interessiert, mit aktuellen Werten versorgen. Soll diese Vision verwirklicht werden, sind allerdings Anpassungen bezüglich der Echtzeit notwendig.

Es stellt sich außerdem die Frage, ob ein solches System preislich konkurrenzfähig etwa zum AS-Interface (siehe Teil 1) sein kann. Die Preise für Speicher und Prozessoren werden zwar weiter fallen, jedoch wird ein so simpel realisiertes System wie ASI immer kostengünstiger sein können.

Die „klassische“ Kommunikationspyramide wird sowohl physikalisch (siehe Ethernet in der Prozessautomatisierungstechnik) als auch logisch (siehe Applikationsschnittstellen) immer weiter aufgeweicht.

## 6 Literaturverzeichnis

### 6.1 Bibliografie

- [BUSSYS] Bussysteme in der Automatisierungs- und Prozesstechnik, Gerhard Schnell (Hrsg.); Vieweg, Heidelberg, 2003
- [DIPL-04] Konzeption und Realisierung einer plattformunabhängigen Fernsteuerung von Anlagen mit OPC über XML Web Services, Birgit Rosenögger, Michael Labicki, Diplomarbeit HAW- Hamburg, 2004
- [ETHIND] Industrieautomation mit Ethernet-TCP/IP und Web-Technologie Frank J. Furrer, Hüthig, Heidelberg, 2003
- [IEC-05] Real-time Ethernet Powerlink (EPL), Publicly Available Specification, IEC PAS 62408, First edition 2005-06
- [IEE-05] IEE Das Automatisierungsmagazin, Hüthig, Heidelberg, 7/8 2005
- [MMS] Winfried Blumann, Alhard Horstmann (Hrsg.), Fertigungsautomatisierung mit MMS, VDI Verlag, Düsseldorf, 1993
- [MSAUT] Thomas Bettermann, Anwendung von Microsoft Softwarestandards in der Automatisierungstechnik, expert verlag, Renningen, 2002
- [OPC-04] OPC Foundation, OPC XML-DA Specification, 2004
- [OPC-98] OPC Foundation, OPC Overview, 1998
- [OPCGRUND] OPC Grundlagen, Implementierung und Anwendung, Frank Iwanitz, Jürgen Lange, Hüthig, Heidelberg, 2005

### 6.2 Cybergrafie

- [BNWISSEN] [www.business-wissen.de](http://www.business-wissen.de)
- [ESUISSE] [www.electrosuisse.ch](http://www.electrosuisse.ch)
- [IEC] [www.iec.ch](http://www.iec.ch)
- [IEE] [www.iee-online.de](http://www.iee-online.de)
- [ITWISSEN] [www.itwissen.info](http://www.itwissen.info)
- [OPCFFOUND] [www.opcfoundation.org](http://www.opcfoundation.org)

## 7 Anhang

### 7.1 Technische Daten der verschiedenen Realtime-Verfahren

Die Tabelle gibt einen kompakten Überblick über die wichtigsten Daten der vorgestellten Realtime-Verfahren. Die Spalten sind in aufsteigender Reihenfolge anhand der Lösungsansatznummer geordnet.

	Profinet v.3	Sercos-III	EtherCAT	EPL	Ethernet/IP
Lösungsansatz	1.	1.	1.	2.	3.
Eingriff auf OSI-Schicht	2a (MAC-Layer)	2a (MAC-Layer)	2a (MAC-Layer)	2b (LLC-Layer)	oberhalb von 4
Verfahrensbasis	zyklusbasiert, isochronen und asynchronen Zyklus	zyklusbasiert, isochronen und asynchronen Zyklus	zyklusbasiert, isochronen und asynchronen Zyklus	zyklusbasiert, isochronen und asynchronen Zyklus	nachrichtenorientiert, zeitbasiertes System
Verfahren	Zeitschlitzverfahren	Zeitschlitzverfahren	Token passing (logischer Ring erforderlich)	CSMA/CD-Verfahren mit übergeordnetem Zeitschlitzverfahren	UDP, QoS, IEEE 1588
Teilnehmer	k.A.	k.A.	Master, Slaves	Master, Slaves	Master, Slaves
Zugriffsverfahren	TDMA	TDMA	Token passing	Polling (zentral)	CSMA/CD
asynchrone Daten	Standard Ethernet-Telegramm	Standard Ethernet-Telegramm	ein Paket enthält allgemeine (getunnelte) und Realtime-Daten, Ethern-type: EtherCAT	Standard Ethernet-Telegramm	niedrige Priorität
synchrone Daten	Ethern-type: Realtime	Ethern-type: Realtime		Ethern-type: Powerlink	hohe Priorität
Hardware im Realtime-Segment	spezielle Switches in den Teilnehmern	2 Interfaces pro Teilnehmer	spezielle Slaves	Standard Ethernet Hub	Standard (Switch mit full duplex & QoS)
Gateway zwischen Realtime-Segment und Standard-Ethernet	k.A.	spezielle Gateways/ PLCs	spezieller Virtual Ethernet Switch oder Slave mit Gateway Funktionalität	spezieller Gateway, bzgl. des Kommunikationsstacks	Standard Router mit Firewall (entsprechend konfiguriert)
Besonderheiten	-	Topologie beschränkt auf Linie und Ring	Slaves schreiben die Prozessdaten direkt ins Telegramm	IEEE 1588 (Zeitsynchronisierung) möglich	-
Organisation	PROFIBUS Nutzerorganisation	Interessengemeinschaft SERCOS	EtherCAT Technology Group	Ethernet Powerlink Standardization Group	Open Device-Net Vendor Association
Hersteller	ABB, Siemens	ABB, Bosch, Rockwell	ABB, Bosch, NI, Philips	ABB, Atmel, IXXAT	ABB, Bosch, Rockwell

## 7.2 Echtzeitvergleich der verschiedenen Realtime-Verfahren

Die Tabelle zeigt die Ergebnisse eines Echtzeitvergleichs mit 100 synchron gesteuerten Achsen. Verglichen werden Reaktionszeit (Zykluszeit) und deren Jitter (Schwankung).

	Profinet v.3	Sercos-III	EtherCAT	EPL	Ethernet/IP CIPSync
Reaktionszeit	< 1 ms	< 0,5 ms	k.A.	< 1 ms	>> 1 ms
Jitter	< 1 ms	< 0,1 ms	k.A.	< 1 ms	< 1 ms
Übertragungsrate	100 Mbit/s	100 Mbit/s	100 Mbit/s	100 Mbit/s	100 Mbit/s