



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

# Bachelor

Brian Grunert

Eine Machbarkeitsanalyse für eine  
„Avionics Full Duplex Switched Ethernet“-  
Netzwerkkomponente  
als eingebettetes System



Brian Grunert

Eine Machbarkeitsanalyse für eine  
„Avionics Full Duplex Switched Ethernet“-  
Netzwerkkomponente  
als eingebettetes System

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung  
im Studiengang Technische Informatik  
am Studiendepartment Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Thomas Canzler  
Zweitgutachter : Prof. Dr. rer. nat. Gunter Klemke

Abgegeben am 24. April 2008

**Brian Grunert**

**Thema der Bachelorarbeit**

Eine Machbarkeitsanalyse für eine „Avionics Full Duplex Switched Ethernet“- Netzwerkkomponente als eingebettetes System

**Stichworte**

AFDX, Eingebettete Systeme, Echtzeitprogrammierung, Echtzeitbetriebssystem, Fast Ethernet, Luftfahrt, Luftfahrtelektronik, Mikrokontroller, ARINC 664

**Kurzzusammenfassung**

In der zivilen Luftfahrt wird eine Vielzahl von Kommunikationssystemen eingesetzt. Die Kommunikation dieser Systeme untereinander erhöht die Entwicklungs- und Produktionskosten außerdem den Wartungsaufwand. Um dem entgegen zu wirken, wird von verschiedenen Flugzeugherstellern die Nachfrage nach einem einzigen Standard zur Kommunikation zunehmend relevanter. Die Firma Monogram System GmbH ist stark von dieser Entwicklung betroffen und sucht nach einer Möglichkeit ihre Komponenten gemäß dem AFDX- Standard im Flugzeug zu integrieren. In dieser Arbeit wird ein System beschrieben, das die Anforderungen des AFDX- Standards in seinen Grundzügen umsetzt und eine mögliche Lösung für die Integration der Komponenten darstellt.

**Brian Grunert**

**Title of the paper**

Feasibility study for "Avionics Full Duplex Switched Ethernet"- Networkcomponent based on embedded system

**Keywords**

AFDX, Embedded Systems, Realtime Programming, Realtime Operation Systems, Fast Ethernet, Microcontroller, Avionics, ARINC 664

**Abstract**

In civil aviation different bus systems are used which can not communicate with each other directly. To realise a communication between these systems a high development effort is required. The costs for development, production and maintenance are increasing thereby. The aircraft manufacturer would prefer to use one system for the whole communication. Monogram Systems GmbH as supplier for aircraft manufacturer influenced by this evolution and has to find a solution for their components. This report describes a feasible hardware and software solution for the AFDX- Standard requirements.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>8</b>
<b>Abbildungsverzeichnis</b>	<b>9</b>
<b>1 Einführung</b>	<b>11</b>
<b>2 Kommunikationssysteme</b>	<b>14</b>
2.1 Netzwerkmodell . . . . .	14
2.2 Bussysteme . . . . .	18
2.2.1 Controller Area Network . . . . .	18
<b>3 Ethernet</b>	<b>22</b>
3.1 Einleitung . . . . .	22
3.2 10BaseT . . . . .	23
3.2.1 Übertragungsmedium . . . . .	23
3.2.2 Verbindungstest . . . . .	26
3.2.3 Kodierungsverfahren . . . . .	26
3.2.4 Media Access Control . . . . .	26
3.2.5 Frameformate . . . . .	28
3.3 Fast Ethernet . . . . .	31
3.3.1 Physical Coding Sublayer . . . . .	31
3.3.2 Physical Medium Attachment . . . . .	31
3.3.3 Physical Medium Dependent Sublayer . . . . .	32
3.3.4 4B/5B Kodierungsverfahren . . . . .	32
<b>4 Avionic Full Duplex Switched Ethernet End System</b>	<b>33</b>
4.1 Allgemeine Beschreibung des AFDX- Netzwerkes . . . . .	33
4.2 Einleitung . . . . .	33
4.3 Interoperabilität und Determinismus der Media Access Control Schicht . . . . .	34
4.3.1 Virtueller Link . . . . .	35
4.3.2 Datenfluss/Verkehrs Kontrolle . . . . .	35
4.3.3 Scheduling . . . . .	35
4.3.4 Mac-Beschränkungen . . . . .	36

---

4.3.5	MAC- Adressierung . . . . .	38
4.3.6	Redundanz Konzept . . . . .	39
4.4	Interoperabilität des IP Layers . . . . .	44
4.4.1	Avionik Services . . . . .	44
4.4.2	End System Kommunikationsstack . . . . .	48
4.5	Interoperabilität des Netzwerk Layers . . . . .	49
4.5.1	Adressierung . . . . .	49
4.6	Nachrichtenstruktur . . . . .	54
4.6.1	Implizite- versus Explizite Nachrichten . . . . .	55
4.6.2	Primitive Datentypen . . . . .	56
4.6.3	Datenanordnung . . . . .	56
4.6.4	Spare - Padding . . . . .	56
4.6.5	Functional Data Set . . . . .	57
4.6.6	Globale Nachrichten Struktur . . . . .	59
<b>5</b>	<b>Design</b>	<b>60</b>
5.1	Hardwaredesign . . . . .	60
5.1.1	Dual Core Industrial Controller DCIC 9907 . . . . .	60
5.1.2	Physikalische Eigenschaften . . . . .	62
5.2	Echtzeit Betriebssystem Euros . . . . .	65
5.2.1	Hardwareunabhängigkeit . . . . .	65
5.2.2	Multitaskingfähigkeit . . . . .	66
5.2.3	Euros Objekte . . . . .	66
5.2.4	Clusterkonzept . . . . .	67
5.2.5	Systemzustände . . . . .	67
5.2.6	DCIC 9907 Ethernet Port Treiber . . . . .	67
<b>6</b>	<b>Realisierung</b>	<b>70</b>
6.1	Programm . . . . .	72
6.1.1	Root- Task . . . . .	72
6.1.2	Idle- Task . . . . .	72
6.1.3	VL- Task Sender . . . . .	72
6.1.4	VL- Task Empfänger . . . . .	73
6.1.5	Funktion des VL- Task Sender . . . . .	74
6.1.6	Funktion des VL- Task Empfänger . . . . .	74
<b>7</b>	<b>Performanz</b>	<b>75</b>
7.1	Bandbreite . . . . .	75
7.2	Datenübertragungsrate . . . . .	75
7.3	Übertragungsfrequenz/Datenübertragungsrate . . . . .	75
7.4	Latenz . . . . .	76

---

7.4.1	Technologische Latenz . . . . .	76
7.5	Messaufbau . . . . .	77
7.6	Performanz- Tests . . . . .	77
7.6.1	Test 1 Senden . . . . .	79
7.6.2	Test 2 Senden . . . . .	81
7.6.3	Test 3 Senden und Empfangen . . . . .	82
7.6.4	Test 4 Senden und Empfangen . . . . .	84
<b>8</b>	<b>Zusammenfassung</b>	<b>85</b>
<b>9</b>	<b>Ausblick</b>	<b>87</b>
	<b>Literaturverzeichnis</b>	<b>88</b>
	<b>Glossar</b>	<b>89</b>

# Tabellenverzeichnis

3.1	PIN-Belegung des MDI bei 10BaseT . . . . .	24
3.2	Kenngößen des 10BaseT-Netzwerkes . . . . .	25
4.1	TFTP RFC . . . . .	47
4.2	Switch Forwarding . . . . .	51
4.3	Primitive Datentypen . . . . .	56



# Abbildungsverzeichnis

1.1	CAN System mit Anbindung an das AFDX- Netzwerk . . . . .	12
1.2	Anbindungs des AFDX- End Systems an das Netzwerk . . . . .	13
2.1	OSI-Referenzmodell . . . . .	14
2.2	OSI Virtuelle Kommunikation . . . . .	15
2.3	OSI Kommunikationswege . . . . .	16
2.4	CAN Arbitrierung . . . . .	20
3.1	Crossover Twisted-Pair-Kabel . . . . .	24
3.2	1:1 Twisted-Pair-Kabel . . . . .	25
3.3	Data Terminal Equipment . . . . .	27
3.4	Ethernet Frame . . . . .	29
3.5	Präambel . . . . .	30
4.1	Netzwerk Hierarchie . . . . .	34
4.2	Integritäts- und Redundanz Prüfung . . . . .	39
4.3	Fehlerhaft übertragenes Frame . . . . .	40
4.4	Verlust eines Frame . . . . .	40
4.5	Reset des Senders . . . . .	41
4.6	Fehler im Switch . . . . .	41
4.7	Speicherüberlauf . . . . .	46
4.8	Sub- Virtuelle Links mit FIFO Queue . . . . .	47
4.9	Minimales AFDX Frame . . . . .	49
4.10	Maximales AFDX Frame . . . . .	50
4.11	Adressierung im End System <a href="#">ARINC (2005)</a> . . . . .	50
4.12	Physikalischer Aufbau . . . . .	51
4.13	IP Quelladresse . . . . .	53
4.14	IP Zieladresse . . . . .	54
4.15	Nachrichtenstruktur . . . . .	57
4.16	Functional Status Set . . . . .	58
4.17	Data Set . . . . .	59
4.18	Global Data Set . . . . .	59

---

5.1	Evaluationboard . . . . .	61
5.2	DCIC 9907 Blockdiagramm <a href="#">DualCore (2006)</a> . . . . .	63
5.3	Betriebssystem Komponenten <a href="#">EUROS (2006)</a> . . . . .	66
7.1	Testaufbau Datenrate . . . . .	78
7.2	Dauertest . . . . .	79

# 1 Einführung

Die Machbarkeitsanalyse findet im Rahmen einer möglichen Produkt Erweiterung bei der Firma Monogram Systems GmbH statt. Die Arbeit soll eine Aussage darüber treffen inwieweit sich eingebettete Systeme zur Kommunikation über das AFDX- Netzwerk eignen. Zu diesem Zweck werden die auf den Markt existierenden möglichen Konfigurationen beschrieben und verglichen, um ein, für die Anforderungen geeignetes System aufzubauen, und über das AFDX- Netzwerk kommunizieren zu lassen.

Die möglichen Konfigurationen der Hardware müssen im Bereich der eingebetteten Systeme angesiedelt sein, da neben der Kommunikation mit dem AFDX- Netzwerk auch Aufgaben zur Steuerung und Auswertung von Aktoren und Sensoren zu erfüllen sind. Die eingebetteten Systeme sollen im Bereich der zivilen Luftfahrt zum Einsatz kommen, und unterliegen damit den hier geforderten strengen Richtlinien der Zertifizierung.

Im Fokus der Analyse steht dabei, ob die, für ein AFDX- End System nach ARINC Spezifikation 664 Part 7, erforderlichen Bedingungen unter Verwendung eines eingebetteten Systems erfüllt werden können.

Die in der zivilen Luftfahrt angewandten Kommunikationssysteme sind vielfältig. Neben CAN, TTP und RS-485, als reine Bussystem, existiert auch ein Netzwerksystem, das AFDX- Netzwerk. Das AFDX- Netzwerk wird bereits als Backbone- System von verschiedenen Flugzeugherstellern erfolgreich eingesetzt.

Die folgenden Grafiken sollen einen Überblick über den aktuellen Stand der Kommunikation geben und den Unterschied zum AFDX- Netzwerk verdeutlichen.

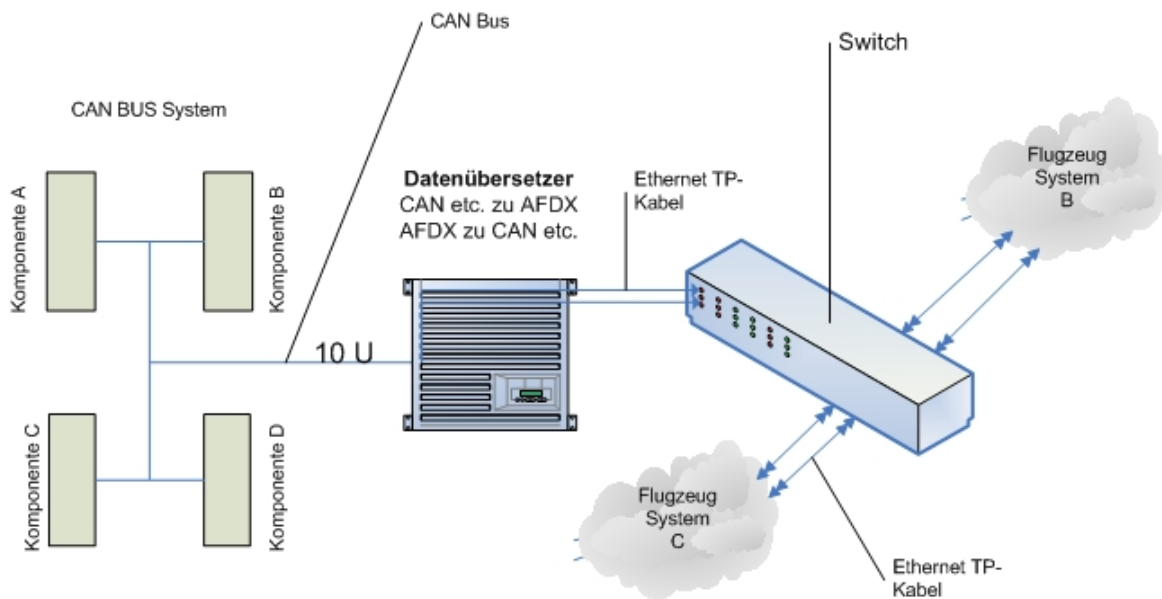


Abbildung 1.1: CAN System mit Anbindung an das AFDX- Netzwerk

Ein Grund für den Einsatz von AFDX ist der erhebliche Geschwindigkeitsvorteil gegenüber dem Bussystemen. Um diesen Vorteil für andere Systeme im Flugzeug zu nutzen, zum Beispiel das Wasser- und Abwassersystem, muss evaluiert werden in wieweit es möglich ist AFDX konforme Kommunikation auf Mikrocontroller Ebene umzusetzen.

Der große Vorteil dieses Verfahrens ist, dass Sensoren und Aktoren direkt in das AFDX-Netzwerk eingebunden werden können, ohne dass zusätzliche Hardwarekomponenten benötigt werden, die die Umsetzung auf andere Bussysteme übernehmen. Ein weiterer Vorteil ist die Redundanz des AFDX- Netzwerkes, welches die Ausfallsicherheit der Kommunikationsschnittstelle erhöht.

Um eine fundierte Aussage über das Verhalten eines eingebetteten Systems treffen zu können wird das nach der ARINC 664 Spezifikation in seinen Grundzügen implementiert. Hierzu ist es erforderlich eine geeignete Hardware, die mit den gegebenen Eigenschaften der AFDX- Spezifikation konform geht, zu finden.

Der erste Ansatz besteht darin einen ASIC zu erwerben, der bereits den AFDX- Stack enthält. Von diesem Ansatz muss Abstand genommen werden, da die Firmen diesen ASIC nur zur eigenen Verwendung herstellen, und er nicht vertrieben wird. Der Erwerb der Intellectual Property ist ebenfalls nicht möglich, da auch die Rechte nicht zum Verkauf freigegeben sind.

Nachdem hier keine zeitnahe Lösung zu erwarten ist, wird die Gewichtung auf den Sektor der Mikrocontroller gelegt. Es gilt einen Controller zu bestimmen der die Möglichkeit bietet

zwei Ethernet- Schnittstellen an zu sprechen. Für eine Marktanalyse wurde die Embedded World aufgesucht, der direkte Kontakt zu den einzelnen Ausstellern zeigt schnell das Produkte dieser Art nicht nachgefragt sind. Es wurden von verschiedenen Herstellern angeboten eine Zusammenarbeit anzustreben, um eine Lösung für das Problem zu finden. Das würde den Rahmen der Arbeit überschreiten.

Weitere Markanalysen ergaben das die Firma Dualcore Semiconductors einen Core mit dem entsprechenden Design anbieten.

Das SoC entspricht den Anforderungen und wird für die Umsetzung der Arbeit eingesetzt. Ein weiterer Vorteil ist das neben dem Controller auch ein Evaluationboard angeboten wird, dass die Integration des System erheblich beschleunigt.

Das End System wird wie in der folgenden Grafik zu sehen, in das AFDX- Netzwerk eingebunden. Wie deutlich zu erkennen ist, kann hier auf zusätzliche Geräte zur Datenübersetzung verzichtet werden.

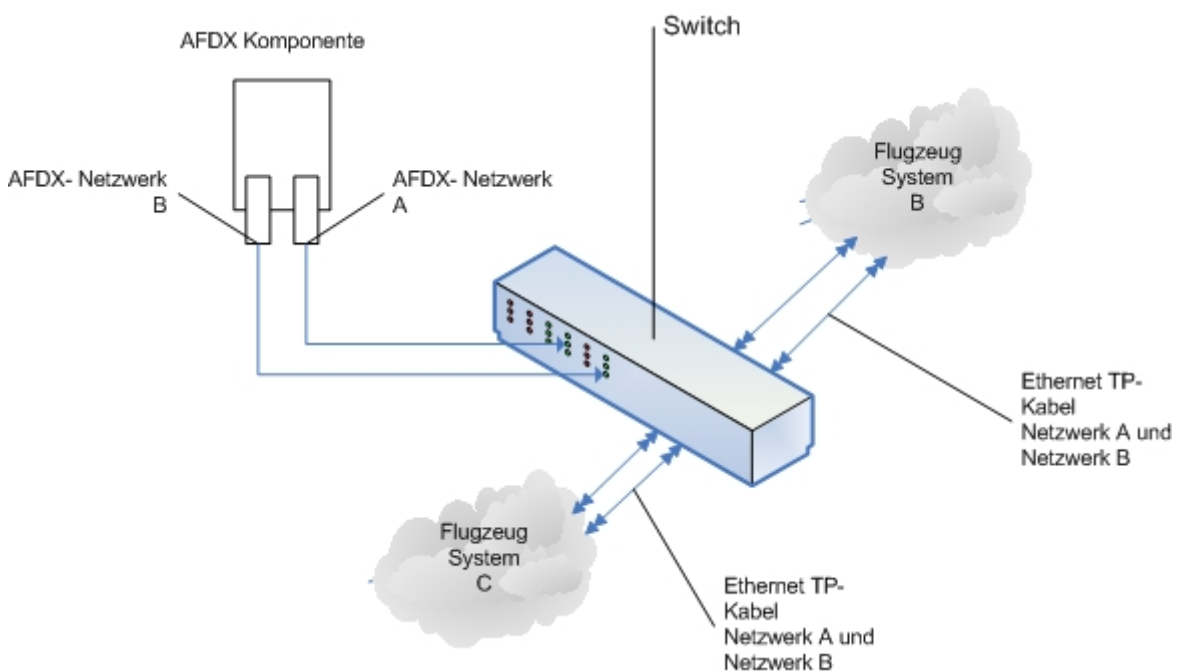


Abbildung 1.2: Anbindung des AFDX- End Systems an das Netzwerk

# 2 Kommunikationssysteme

## 2.1 Netzwerkmodell

Um die Kommunikation innerhalb eines Standards im Netzwerk zu gewährleisten wird häufig das OSI- Referenzmodell (Open- Systems- Interconnection- Referenzmodell) heran gezogen. Bei diesem handelt es sich um ein in sieben Schichten (Layer) unterteiltes Netzwerkmodell, bei der jede Schicht einen bestimmte Dienste zur Verfügung stellt. Die Dienste leisten einen, für den Ablauf der Kommunikation funktionellen Beitrag. Der Dienst der durch die jeweilige Schicht erbracht wird kann durch die anderen Schichten im OSI- Referenzmodell (Abbildung 2.1) erweitert werden(Rech (2002)).

Anwendungsschicht	7
Darstellungsschicht	6
Kommunikationssteuerungsschicht	5
Transportschicht	4
Vermittlungsschicht	3
Sicherungsschicht	2
Bitübertragungsschicht	1

Abbildung 2.1: OSI-Referenzmodell

Die Dienste unterliegen definierten Regeln und werden Protokolle genannt. Werden nun zwischen zwei Netzwerkteilnehmern Informationen ausgetauscht, erfolgt die Kommunikation immer auf Schichten der gleichen Ebene. Da diese aber nicht direkt miteinander kommunizieren, nennt man diese Virtuelle Kommunikation. Das wieder führt dazu das in der gleichen Ebene auf Sender- und Empfängerseite gleiche Protokolle verwendet werden müssen.

Die Schichten, die auf einer Ebene liegen werden als Peers bezeichnet und dem entsprechend werden die auf ihnen ausgeführten Protokolle, Peer-Protokolle genannt. Damit die

Informationen auf beiden Seiten verstanden werden, wird ein einheitliches Format benötigt. Dieses Format nennt sich Protocol Data Unit (PDU) und ist in zwei Teile aufgeteilt, den Protokollinformationen (Protocol Control Information, PCI) die somit den Header repräsentieren und der Dateneinheit (Service Data Unit, SDU). Die Umsetzung des Dienstes in einer Schicht erfolgt über eine oder mehrere Arbeitseinheiten (Entities). Der Zugriff erfolgt über Service Access Points (SAP), die eine Art Adresse für Funktionsaufrufe darstellt (Rech (2002)).

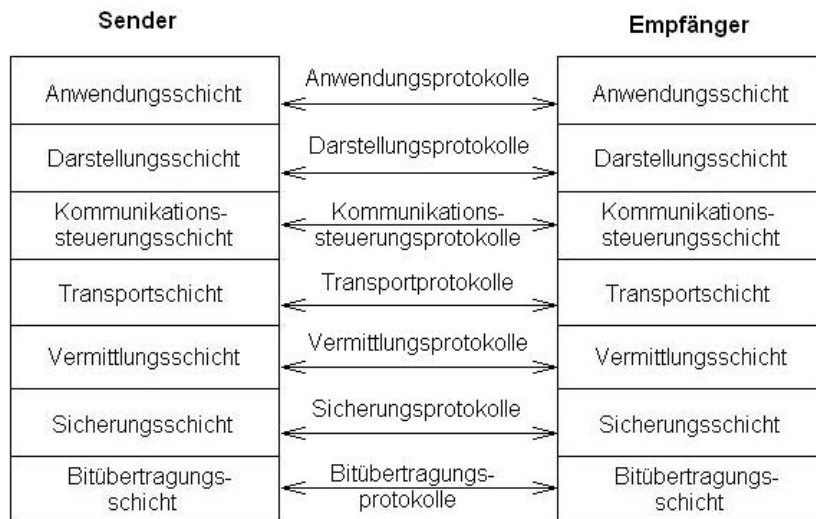


Abbildung 2.2: OSI Virtuelle Kommunikation

Die zu übertragenden Daten werden vertikal durch das OSI-Referenzmodell gereicht. Beim Sender von oben nach unten und beim Empfänger in der umgekehrten Richtung. Die physikalische Übertragung findet horizontal statt, von Bitübertragungsschicht zu Bitübertragungsschicht. Dadurch bleibt das Prinzip der Virtuellen Kommunikation bewahrt, da alle Schichten erreicht werden.

- Bitübertragungsschicht

Die unterste Schicht im OSI-Referenzmodell ist die Bitübertragungsschicht (engl. Physical Layer), sie dient zur physikalischen Übertragung der Daten. Der Dienst überträgt dabei einen physikalischen Bitstrom von einem Sender zu einem oder mehreren Empfängern. Diese ungesicherte, physikalische Verbindung wird dabei vom Übertragungsmedium bei Bedarf auf- oder abgebaut. Da sich über die Bitübertragungsschicht das Übertragungsmedium definiert, sind in dieser Schicht die Größe des Datensignals, die Bit- oder Bytesynchronisation, die Kodierungs- und Modulationsverfahren hinterlegt. Des Weiteren werden hier auch das Interface, der physikalische Anschluss und die Verbindungselemente definiert.

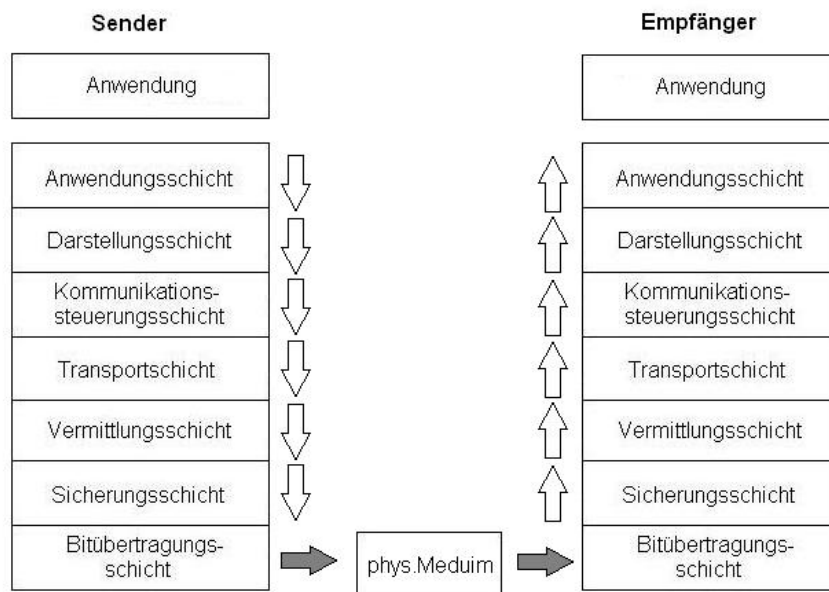


Abbildung 2.3: OSI Kommunikationswege

- Sicherungsschicht

Die zweite Schicht des OSI- Referenzmodell, die Sicherungsschicht(engl. Data Link Layer) hat die Aufgabe einen fehlerfreien Austausch der Daten zu gewährleisten. Um das zu realisieren werden die Daten zu Datenpaketen(Frames) zusammengefasst. Die Fehlererkennung erfolgt dabei über Prüfsummen, die dem Datenpaket angehängt werden. Das erfordert ein einheitliches Format der Datenpakete. In der Sicherungsschicht wird außerdem auch der Zugriff auf das Übertragungsmedium über ein entsprechendes Zugriffsverfahren realisiert. Es findet eine Unterteilung in zwei Bereiche statt, wobei sich der Untere um die Art der verwendeten Netzwerktechnologie kümmert, und der Obere, der von allen IEEE- Netzwerktechnologien gleichermaßen verwendet wird, den Datenaustausch vereinfacht.

- Vermittlungsschicht

Die Vermittlungsschicht(engl. Network Layer) im OSI- Referenzmodell dient dem eigentlichen Datentransfer. Der Dienst der hier zur Verfügung gestellt wird, ist die Wegfindung im Netzwerk. Die Wegfindung(Routing) ermöglicht Kommunikation über die Netzwerkgrenzen hinaus. Dazu wird in dieser Schicht die PDU mit Ziel- und Quelladressen versehen, über das Routing erfolgen kann. Die dritte Ebene ist von den ersten beiden unabhängig, das ermöglicht zum einen die Wegfindung über mehrere logisch strukturierte Netzwerke und zum anderen die Bildung von hierarchisch untergliederten Teilnetzwerken.



- Transportschicht

Die vierte Schicht, die Transportschicht(engl. Transport Layer), kann als Vermittler zwischen den über ihr und unter ihr liegenden Schichten betrachtet werden. Wenn Programme Daten miteinander austauschen sorgt die Transportschicht dafür, dass die Daten der Vermittlungsschicht an das richtige Programm weitergeleitet werden. Sie sorgt für einen zuverlässigen Austausch von Daten und dass mehrere Dienste oder Anwendung dieselben Transportmechanismen nutzen können. Außerdem erfolgt hier die Aufteilung der Daten in Segmente(Fragmentierung) wie sie von der Vermittlungsschicht weitergeleitet werden können. Auch das Zusammensetzen der Daten zu logischen Einheiten (Defragmentierung) erfolgt innerhalb der Transportschicht. Dabei wird auch die Reihenfolge der Fragmente berücksichtigt.

- Kommunikationssteuerungsschicht

Die Kommunikationssteuerungsschicht ist die fünfte der sieben Schichten im OSI- Referenzmodell und wird auch Sitzungsschicht genannt. Der Dienst den sie bereitstellt ist Anwendungsorientiert und sorgt zum einen für den Verbindungsauf- und abbau und zum anderen für die Darstellung der Übertragungsdaten, in einer unabhängigen Form, für die darüber liegenden Schichten. Der Anmeldevorgang wird auch als Session bezeichnet. Nachdem der Aufbau einer Sitzung statt gefunden hat, ist die Grundlage für eine virtuellen Verbindung zwischen zwei Prozessen abgeschlossen. Dienste der Sicherungsschicht sind der Aufbau, die Überwachung, als auch die Beendigung einer Sitzung.

- Darstellungsschicht

Die sechste Schicht im OSI- Referenzmodell ist die Darstellungsschicht (engl. Presentation Layer). Zwischen Kommunikationssteuerungsschicht und Anwendungsschicht ist sie für die Kommunikation und das weiterreichen der Daten verantwortlich. Zusätzlich stellt sie noch bestimmte Dienste die im Netzwerk benötigt werden bereit. Dazu gehören Informationen über Schnittstellen zu bestimmten Netzwerkressourcen, wie Speichermedien oder Drucker. Es wird festgelegt, in welcher Form dem Anwender die Daten präsentiert werden, und ob dabei Verschlüsselung- oder Komprimierungsverfahren zum Einsatz kommen.

- Anwendungsschicht

Die oberste Schicht, die Anwendungsschicht(engl. Application Layer) stellt die Verbindung zwischen Anwender und Anwendungsprozessen über das Netzwerk her. Hier sind Protokolle angesiedelt die speziell an die Anwendung angepasst sind. Ein Beispiel ist das File Transfer Protocol(FTP) mit dem Dateien übertragen werden können.

## 2.2 Bussysteme

Bussysteme kommen in verschiedensten Bereichen der Industrie zum Einsatz. Sie unterscheiden sich in vielen Eigenschaften voneinander. Angefangen bei den klassischen Feldbussen über Sensor- Aktor- Busse der untersten Geräteebene bis hin zu Bussystemen in Fahrzeugen aller Art.

Der Fokus dieser Arbeit liegt speziell auf Bussystemen die in der zivilen Luftfahrt angewendet werden. Das Controller Area Network (CAN) wird dabei genauer beschrieben, da dieses durch das AFDX- Netzwerk ersetzt werden soll. Beschrieben werden die geographischen, die zeitlichen und die technischen Eigenschaften der einzelnen Systeme, um die Unterschiede zum AFDX- Netzwerk zu verdeutlichen.

### 2.2.1 Controller Area Network

Das am Ende der achtziger Jahre von der Firma Bosch entwickelte Controller Area Network(CAN) sollte Zuverlässigkeit, geringe Kosten, ein effektives Protokoll und Multi-Master-Fähigkeit zum Ziel haben.

Das von der Firma Intel entwickelte CAN-Protokoll wurde von der Internationale Organisation für Normung(ISO) unter der ISO 11898 festgelegt.

Mit der Gründung der CAN in Automation e.V. (CiA) im Jahr 1992 wurde eine Basis für die Förderung von CAN-basierten Anwendungen und Anwender geschaffen. Die CiA bündelt außerdem Normierungs-, Marketing- und Entwicklungstätigkeiten.

Die Anforderung an ein Bussystem in der Automobilindustrie und in der Luftfahrt ähnelt sich stark. Das ist auch ein Grund, warum sich CAN auch in der Luftfahrt durchsetzen konnte.

#### 2.2.1.1 Physikalische Struktur

Die einzelnen Buskomponenten werden über eine Linienarchitektur, die aus einer verdrehten Zweidrahtleitung besteht miteinander verbunden. Die Ausdehnung des Netzwerkes wird stark durch Übertragungsrate bestimmt und liegt zwischen 40m bei 1MBit/s und 1000m bei 50kBit/s.

### 2.2.1.2 Protokolleigenschaften

Das CAN ist ein Multi-Master-Netzwerk. Nachrichten werden als Broadcast übertragen und unterscheiden sich in ihrer Priorität. Dabei bildet das Arbitrierungssystem die Grundlage des CAN-Protokolls.

Nachrichten können von jedem Busteilnehmer(Knoten) gesendet werden, die einzige Bedingung die dabei erfüllt sein muss, ist ein freier Bus. Der Busteilnehmer beginnt mit der Übertragung des Telegrammheaders, der einem speziell der Nachricht zugeordneten Identifier enthält. Die Arbitrierung wird auf den Identifier angewandt, wenn zwei Knoten gleichzeitig mit der Übertragung beginnen, während des Arbitrierungsvorgang, werden rezessive- Bits(1) des einen Knoten von dominante- Bits(0) des anderen Knoten überschrieben. Alle Knoten des Systems hören, auch während des Sendens, den Bus auf dominante- Bits ab und beenden ihren Sendevorgang sobald ein solches dominantes- Bit empfangen wird. Das bedeutet, Nachrichten mit der höchsten Priorität, dem niedrigsten Identifier, gewinnt die Arbitrierung und sendet ohne Zeitverlust ihre Nachrichten. Nachdem ein Knoten die Arbitrierung verloren hat wird erneut versucht auf den Bus zuzugreifen.

Die in Abbildung 2.4 gezeigten Knoten sind als Master 1 bis Masster 4 bezeichnet. Versucht Master 1 unter dem Objekt- Identifier 34 (obj-id) eine Nachricht an den Bus zu übergeben und gleichzeitig auch Master 4 mit der Objekt- Id 12, wird für Master 1 die Bitfolge 000 0011 0100 und für Master 4 die Bitfolge 000 0001 0010 übertragen. Da rezessive Bit(1) von dominanten Bits (0) überschrieben werden erhält Master 4 den Buszugriff.

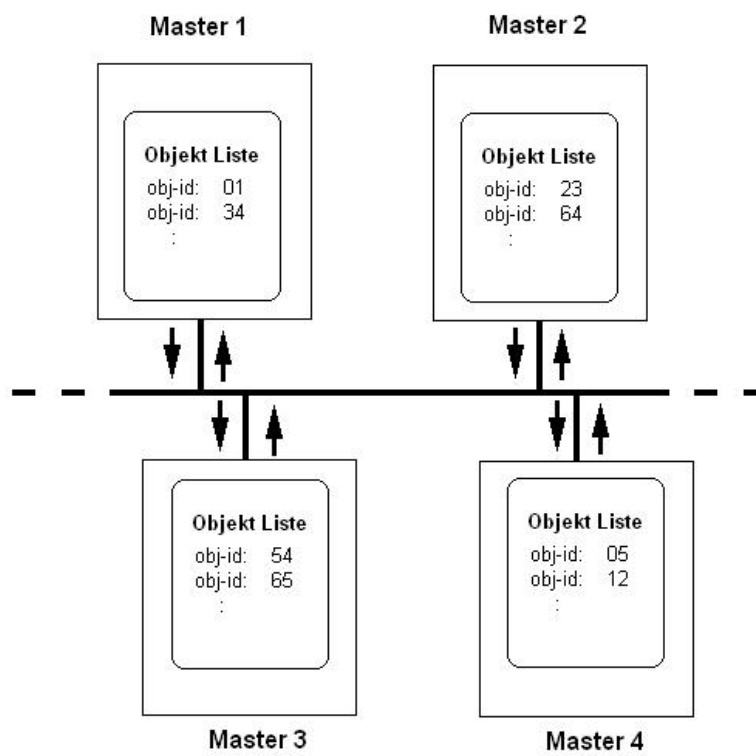


Abbildung 2.4: CAN Arbitrierung

Die maximale Verzögerungszeit für die höchstprioritäre Nachricht beträgt die Zeit, die benötigt wird, die aktuell zu übertragene Nachricht zu senden. Das bedeutet es lässt eine Aussage über deren maximale Latenzzeit treffen. Problematisch ist das bei sehr hoher Buslast wobei Nachrichten niedriger Priorität am Buszugriff gehindert werden.

Die Grundlage von CAN bildet das Arbitrierungssystem, damit es greifen kann muss die maximale Signallaufzeit im Netzwerk deutlich kleiner sein als eine Bitzeit, da jeder Knoten innerhalb dieser Bitzeit reagieren muss. Dies führt auch zu dem engen Zusammenhang zwischen maximaler Netzwerkausdehnung und der Übertragungsrates. Die Signaldämpfung ist nicht so gewichtig wie bei anderen Kommunikationssystemen.

Das hohe Maß welches CAN an Übertragungssicherheit zur Verfügung stellt ist bereits Hardwareseitig implementiert. Dazu zählt neben der Arbitrierung auch das Filtern von Nachrichten, die Fehlererkennung und -beschränkung und die Empfangsbestätigung. Die Fehlererkennung wird bei CAN über fünf Mechanismen realisiert, die Bitlevel-Überwachung, Cycling Redundancy Check(CRC), Acknowledgement- Check, Bit Stuffing und durch die Nachrichtenüberwachung. Des Weiteren verfügt jeder Knoten über eine Selbstüberwachung, die ihn bei bestimmten Fehlerraten selbsttätig in verschiedene Fehlermodi setzen. Diese Modi, error active, error passive und bus- off unterscheiden sich dabei in der Art der Fehlerbehandlung und werden nacheinander beginnend bei error active angesprochen.

## 3 Ethernet

Die in diesem Kapitel aufgeführten Punkte stellen nur einen Auszug aus der Ethernet-Welt dar und beschränken sich dabei auf die in der Luftfahrt verwendeten Technologien.

### 3.1 Einleitung

Die bestimmende Netzwerktechnologie im Bereich der Local Area Networks ist das Ethernet. Die in seiner ursprünglichen Form von Robert Metcalfe, zur Anbindung von Druckern, entwickelt wurde, arbeitet mit einer Datenrate von 2,94 MBit/s.

Mit der Gründung der DIX-Gruppe (DEC, Intel und Xerox) im Jahr 1979 wurde die Entwicklung des Ethernets vorangetrieben. Mit dem Ziel die Ethernet-Technologie als Standard für Local Area Network durchzusetzen. Im Jahr 1980 wurde dieser firmenspezifische Standard, der heute als Ethernet-Version 1 bekannt ist, und eine Datenrate von 10-MBit/s unterstützt, veröffentlicht.

Der eigentliche Ethernet-Standard wurde 1983 vom amerikanischen Normungsinstitut IEEE (Institute of Electrical and Electronics Engineers) unter der Bezeichnung IEEE 802.3 veröffentlicht. Der beim AFDX-Protokoll angewandte 10BaseT-Standard, auch bekannt als IEEE 802.3i, erschien im Jahr 1990. Ein wesentlicher Unterschied bestand in der Art des Übertragungsmediums. Es wurde zum ersten Mal anstelle eines Koaxialkabels ein Unshielded-Twisted-Pair-Kabel (UTP-Kabel) verwendet.

Im Jahr 1995 erfolgte dann die Erweiterung um den Fast-Ethernet-Standard 100Base-X, der die Übertragungsrate auf 100MBit/s anhebt. Dieser wird zurzeit in der zivilen Luftfahrt erfolgreich angewendet und dient als Backbone-System bei verschiedenen Flugzeugherstellern.

Aus Gründen der Vollständigkeit soll an dieser Stelle auch der Gigabit-Ethernet-Standard erwähnt werden. Dieser wird im Folgenden nicht weiter betrachtet, stellt aber den nächsten Schritt der Übertragungstechnologien in der zivilen Luftfahrt dar. Standardisiert wurde er 1998 durch die IEEE 802.3z. Der Standard bezieht sich auf die Verwendung von Glasfaser als Übertragungsmedium. Die IEEE 802.3ab-Gruppe hingegen veröffentlichte 1999 den Standard auf Basis der bekannten UTP-Technologie ([Rech \(2002\)](#)).

## 3.2 10BaseT

Bei der Einführung der 10BaseT Variante(IEEE-802.3i) wurde das Koaxialkabel durch ein verdrehtes Kupferkabel(Twisted- Pair- Kabel) ersetzt. Der Grund für die Erweiterung des IEEE-802.3 Standards war die beim Koaxialkabel hohe Störanfälligkeit. Durch das veränderte Übertragungsmedium, änderte sich auch die physikalische Struktur im Netzwerk grundlegend. Da es sich bei der 10BaseT Variante um eine Sternförmige Netzwerkarchitektur handelt, sind alle Stationen die dem Netzwerk angehören durch Punkt- zu- Punkt- Verbindungen über einen zentralen Punkt miteinander verbunden. Bei diesem zentralen Punkt handelt es sich entweder um einen Hub oder einen Switch.

Der Verkabelungsaufwand erhöht sich bei dieser Netzwerkarchitektur. Er bringt aber auch gleichzeitig einen großen Vorteil mit sich. Denn sollte es zu einem Kabelbruch kommen ist nicht das gesamte Netzwerk sondern nur eine Station betroffen.

### 3.2.1 Übertragungsmedium

Das als Übertragungsmedium eingesetzte Twisted- Pair- Kabel(TP- Kabel) ist ungeschirmt mit einer Impedanz von 100 Ohm, das entspricht der Kategorie 3 (Cat 3) oder höher(Rech (2002)). Bei der Kategorie 3 handelt es sich um ein verbessertes Telefonkabel, das aus acht Adern besteht und von denen jeweils zwei als Paar miteinander verdreht sind. Genutzt werden bei der 10BaseT Variante nur zwei Aderpaare, eines zum senden und das andere zum empfangen.

Durch die Verwendung von Komplementärsignalen auf den beiden Aderpaaren ist es dem Empfänger möglich eingestreute Störsignale zu filtern. Ein weiterer Effekt des Komplementärsignals ist es, dass elektromagnetische Abstrahlung deutlich reduziert wird. Aufgrund dieser Reduzierung konnte ein ungeschirmtes TP- Kabel den Restriktionen der Elektromagnetischen Verträglichkeit überhaupt genüge leisten. Damit diese positiven Eigenschaften zum tragen kommen, bedarf es einem hohen Qualitätsstandard bei der Herstellung der TP- Kabel. Die Länge der einzelnen Adern des Aderpaares sollten gleich sein und sie sollten möglichst homogen miteinander verdreht sein.

Die Dämpfung für ein TP- Kabel des 10BaseT-Standards beträgt 11,5dB über einen Frequenzbereich von 5Mhz - 10Mhz und einer Länge von 100m. Dabei sind alle Faktoren berücksichtigt die die Maximaldämpfung beinhalten kann. Die Segmentlänge beträgt 100m und als Steckverbinder werden ausschließlich RJ-45-Buchsen und -Stecker verwendet.

Das Media Dependent Interface(MDI) bei einer 10BaseT-Komponente wird durch die RJ-45-Buchse dargestellt. Wie bereits beschrieben werden dabei zwei Aderpaare für die Signal-

übertragung verwendet. Für den Datenausgang handelt es sich um TD+ und TD- und für den Dateneingang um RD+ und RD-. Die genaue Pinbelegung betrachtet die Tabelle 3.1.

PIN-Nummer	Signal
1	TD+ (Datenausgang)
2	TD- (Datenausgang)
3	RD+ (Dateneingang)
4	keine Verwendung bei 10BaseT
5	keine Verwendung bei 10BaseT
6	RD- (Dateneingang)
7	keine Verwendung bei 10BaseT
8	keine Verwendung bei 10BaseT

Tabelle 3.1: PIN-Belegung des MDI bei 10BaseT

Die direkte Verbindung von zwei 10BaseT-Komponenten ist möglich, erfordert aber eine spezielle Form des TP- Kabels, es wird als Crossover- Kabel bezeichnet. Dabei werden die entsprechenden PINs mit einander gekreuzt(siehe Bild3.1).

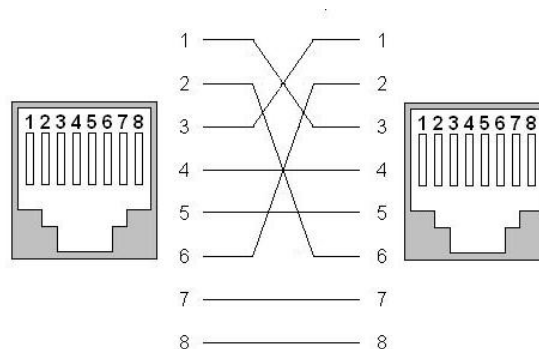


Abbildung 3.1: Crossover Twisted-Pair-Kabel

Bei der Vernetzung mehrerer Stationen über einen zentralen Verteiler (Switch oder Hub) wird die Crossoverfunktion innerhalb der Multi Access Unit(MAU) ausgeführt. Bei diesen handelt es sich um MDI- X- Port. Der Vorteil ist, dass sich alle Stationen im Netzwerk über einen Switch mit einem 1:1-TP-Kabel, wie in Abbildung 3.2 dargestellt verbinden lassen.

Die typischen Kenngrößen für ein 10BaseT-Netzwerk sind in der folgenden Tabelle 3.2 aufgeführt:



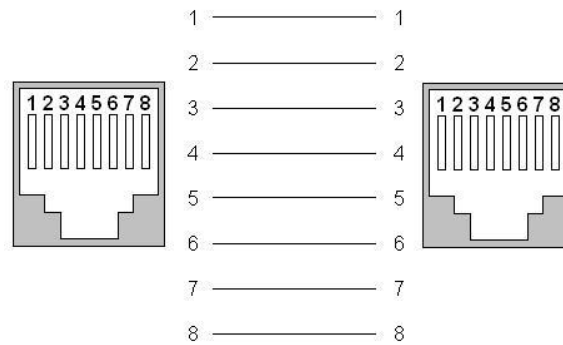


Abbildung 3.2: 1:1 Twisted-Pair-Kabel

Kenngröße	10BaseT
Topologie	Stern
Signalisierungstechnik	Basisband
Kodierungsverfahren	Manchester
Kabeltyp	Twisted-Pair-Kabel
Impedanz	100 Ohm
Maximale Segmentlänge	100 m
Maximale Ausdehnung	500 m
Maximale Anzahl an Stationen pro Segment	2
Anschluss	Direkt über RJ-45-Stecker

Tabelle 3.2: Kenngrößen des 10BaseT-Netzwerkes

### 3.2.2 Verbindungstest

Im 10BaseT-Netzwerk werden neu angeschlossene Stationen mittels eines Leitungsverbindungstest(Link Integrity Test) darauf hin getestet ob die gegenüberliegende Komponente vorhanden ist und ob sie korrekt arbeitet.

Um dies abzu prüfen wird ein 1MHz Signal(Link-Test-Signal) an die gegenüberliegende Multi Access Unit(MAU) gesendet. Das Link-Test-Signal wird nur dann ausgesendet, wenn keine Datenkommunikation stattfindet. Das Aussenden des Link-Test-Signal wiederholt sich dabei periodisch, solange es nicht von Datenpaketen unterbrochen wird. Sollte eine Unterbrechung stattfinden, so wird spätestens 16 +/- 8 ms nach dem Datenpaket erneut ein Link-Test-Signal gesendet.

Eine Verbindungsunterbrechung wird dann angenommen, wenn empfangsseitig für 50 bis 150 ms kein Link-Test-Signal erkannt wurde. Erst wenn 2 bis 10 Link-Test-Signale als korrekt erkannt wurden, wird die Verbindung als funktionsfähig eingestuft.

Da das Link-Test-Signal mit einer Frequenz von 1MHz arbeitet, ist der Test für das Ethernet, dessen Frequenz 10-mal höher ist, nur bedingt Aussagefähig. Bei schlechter Qualität der Kabel kann das Signal bei 1MHz korrekt empfangen werden und bei 10MHz nicht mehr.

### 3.2.3 Kodierungsverfahren

Das Manchester-Kodierungsverfahren wird bei allen 10MBit-Ethernet-Standards angewendet und legt fest wie Daten innerhalb des Ethernet-Standards zu kodieren sind. Da es aber nicht ausreicht eine logische Null oder logische Eins darzustellen, sondern auch die Taktinformationen notwendig sind um es den einzelnen Stationen im Netzwerk zu ermöglichen sich zu synchronisieren, müsste eine eigene Leitung für das Taktsignal verlegt werden. Aus diesem Grund werden bei der Manchester-Kodierung die Taktinformationen zusammen mit den Dateninformationen übertragen. Die Taktinformationen befinden sich dabei in der Mitte einer Bitperiode und erlauben es der Empfangsstation sich mit dem eingehenden Signal zu synchronisieren und die Dateninformationen heraus zu filtern.

### 3.2.4 Media Access Control

Die Media Access Control- Ebene (MAC- Schicht) stellt den funktionell entscheidenden Teil im Ethernet-Standard dar. Aufgrund der Definitionen und Beschreibung der MAC- Schicht, wird das Zielgerichtete Übertragen von Daten innerhalb des Netzwerkes ermöglicht. In der MAC- Ebene wird das Zugriffsverfahren mit Kollisionserkennung als auch das Prozedere

nach auftreten einer Kollision beschrieben. Außerdem sind auf ihr die Definitionen der Frame-Formate und das Adressierungsschema hinterlegt.

Alle Funktionen die auf der MAC- Ebene definiert werden, sind unabhängig vom darunter liegendem Übertragungsmedium. Welches ein weiter Vorteil des Ethernet-Standards darstellt. Denn das Übertragungsmedium ist von dem Zugriffsverfahren und Datenformat entkoppelt.

Verfügen Ethernet-Komponenten über MAC- Funktionen, stellen also einen adressierbaren unabhängigen Endpunkt dar, werden sie auch als Data Terminal Equipment(DTE) bezeichnet. Der Endpunkt ist nicht der Übergang vom Übertragungsmedium zum Interface, sondern die obere Schicht innerhalb der MAC- Schicht, bei dem die Daten an höhere Schichten übergeben werden.

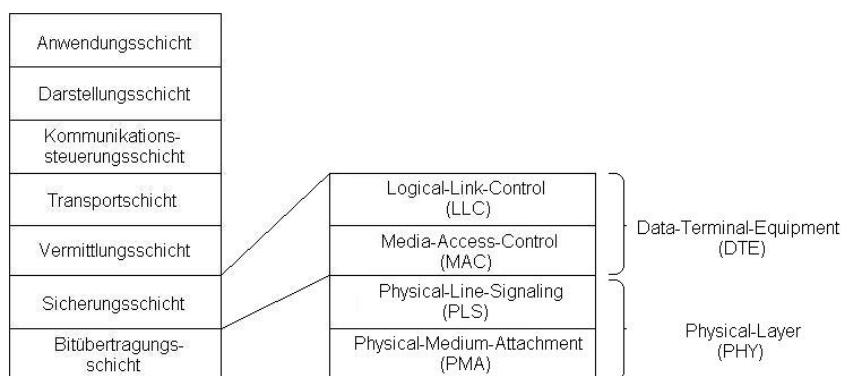


Abbildung 3.3: Data Terminal Equipment

### 3.2.4.1 Zugriffsverfahren

Als eines der ersten Topologien kann die Multipunktverbindung genannt werden, bei dieser Topologie teilen sich mehrere Teilnehmer ein Übertragungsmedium zur gleichen Zeit. Da es nicht vorgesehen ist das mehrere Teilnehmer zur gleichen Zeit senden, da es durch gleichzeitiges senden zu Kollisionen kommen würde, musste ein geeignetes Zugriffsverfahren entwickelt werden welches die Kommunikation mehrerer Teilenehmer ermöglicht.

Dieses Verfahren ist beim Ethernet das Carrier Sense Multiple Access/ Collision Detection (CSMA/CD), das beim Ethernet-Standard auf der MAC- Ebene definiert ist. Das Verfahren

lässt sich dabei in drei Teile zerlegen die jeweils wichtige Funktionen für den Kommunikationsaustausch übernehmen.

Die Carrier Sense Funktion ermöglicht es einer Station das Übertragungsmedium abzuhören und festzustellen ob es belegt oder frei ist. Sollte das Übertragungsmedium belegt sein, das ist der Fall wenn eine Station bereits Daten überträgt, dann versucht diese Station zu einem späteren Zeitpunkt erneut Zugriff auf das Medium zu erlangen. Ist das Medium frei, so beginnt die Station mit der Übertragung. Es besteht zu diesem Zeitpunkt die Gefahr, dass mehrere Stationen mit der Übertragung beginnen. Die Funktion verhindert nicht den gleichzeitigen Zugriff mehrerer Stationen auf das Medium.

Der Multiple Access erlaubt mehreren Stationen den zeitgleichen Zugriff auf das Medium. Wenn zwei oder mehr Station das Übertragungsmedium gleichzeitig nutzen, wird während der Übertragung von jeder dieser Stationen überprüft, ob eine Kollision aufgetreten ist. Der Mechanismus der Collision Detection überprüft dabei, ob die Daten vollständig übermittelt wurden oder eine Kollision aufgetreten ist. Bei dem erkennen einer Kollision wird der Übertragungsvorgang von der sendenden Station abgebrochen. Ein erneuter Sendeversuch wird nach einer von einem Zufallsgenerator ermittelten Zeit initialisiert. Dieser Zufallsgenerator wird genutzt, um zu verhindern, dass mehrere Stationen gleichzeitig mit der Datenübertragung beginnen. Der zur Ermittlung der Wartezeit verwendete Algorithmus nennt sich Backoff-Prozess.

#### **3.2.4.2 Halb- oder Vollduplex**

Der Unterschied zwischen den beiden Übertragungstechniken liegt in der Richtung der Datenübermittlung. Im Halbduplex-Modus findet die Kommunikation immer in eine Richtung statt. Dies war in den Anfängen des Ethernet-Standards durch die Art des Übertragungsmediums nicht anders möglich. Da der gesamte Nachrichtenaustausch über ein Koaxialkabel erfolgte.

Im Vollduplex- Modus ist die Datenübertragung zeitgleich in beide Richtungen möglich. Durch die Einführung von TP- Kabel und Lichtwellenleiter konnte die Restriktion der einseitig gerichteten Kommunikation aufgehoben werden. Voraussetzung für den Vollduplex-Modus ist, dass es für den Empfänger als auch für den Sender von Daten getrennte Kanäle existieren. Des Weiteren kann auf Zugriffsverfahren, wie CSMA/CD verzichtet werden.

#### **3.2.5 Frameformate**

Um im Ethernet Daten gerichtet durch das Netzwerk zu übertragen, müssen diese in ein einheitliches Format gebracht werden. Die Form wird als Frame oder Datenpaket bezeichnet.

Ein Frame setzt sich dabei aus mehreren Bitfeldern zusammen. Beginnen mit der Präambel, die zur Synchronisation mit der Empfängerstation dient, der Hardware-Ziel- und Quelladresse, dem Längen- oder Typenfeld, dem Nutzdaten und dem CRC- Feld. Die Umsetzung des Frames erfolgt dabei auf MAC- Ebene.

Die Festlegung des Frameformats erfolgte durch die DIX-Gruppe und wurde mit der Veröffentlichung des Ethernet- Standards durch die IEEE leicht geändert. Die Änderung beschränkt sich auf den Aufbau der Präambel und darauf ob das dritte Feld als Typen oder Längenfeld genutzt wird.

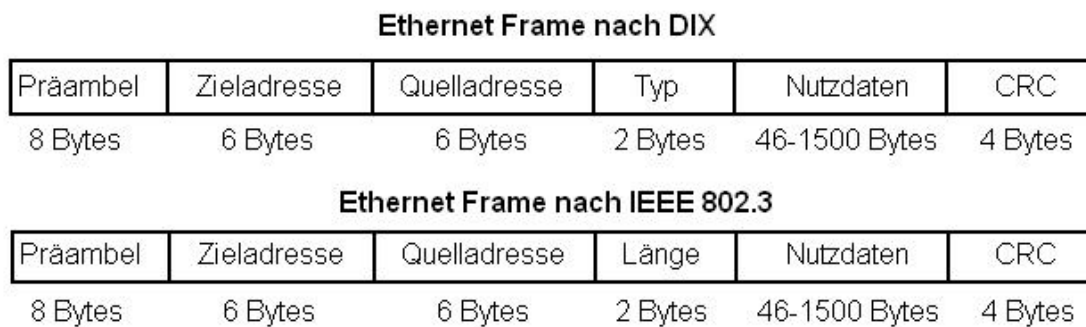


Abbildung 3.4: Ethernet Frame

### 3.2.5.1 Adressenformate

Ethernet-Komponenten verfügen über eine eindeutige Hardwareadresse, auch MAC- Adresse genannt. Diese ist weltweit eindeutig identifizierbar, 6 Bytes lang und dient zur Adressierung der Frames innerhalb der MAC- Schicht. Damit die einzelnen Komponenten im Netzwerk die Daten interpretieren können, ist das Verfahren welches zur Anwendung kommt von entscheidender Bedeutung. Unterschieden wird zwischen der kanonischen und der nonkanonischen Darstellung der Adresse.

Bei der kanonischen Darstellung wird das Least Significant Bit(LSB) eines Bytes zuerst dargestellt und wird beim Ethernet-Standard angewandt. Die nonkanonische Darstellung, die zum Beispiel beim Fibre Distributed Data Interface(FDDI) angewandt wird, stellt zuerst das Most Significant Bit(MSB) eines Bytes dar.

Der Bitstrom, 0000 0011 für die beiden Verfahren wird Hexadezimal wie folgt interpretiert:

1. Kanonische Darstellung 0C
2. Nonkanonische Darstellung 03

Die beiden ersten Bit im ersten Byte der MAC- Adresse erfüllen eine bestimmte Funktion. Das MSB oder auch Individual (0)/Group(1) Bit entscheidet ob es sich bei der MAC- Adresse um eine Unicast- oder Multicastadresse handelt. Das darauf folgende Bit, Global(0)/Lokal(1)-Bit, bestimmt ob es sich um eine Global Administered Address(GAA) oder Locally Administered Address(LAA) handelt. GAAs werden von den Herstellern der Ethernet-Komponenten vergeben und dürfen weltweit nur einmal vorhanden sein. Die LAAs werden innerhalb eines Netzwerks von Netzwerkadministratoren vergeben und überschreiben die vom Hersteller vergebene Adresse. Besondere Beachtung gilt bei der kanonischen Darstellung bei einer Aussage über die Art der Adresse, um Fehler bei der Interpretation zu vermeiden.

### 3.2.5.2 Präambel nach DIX und IEEE 802.3

Die Präambel dient zur Synchronisation mit dem Empfänger und verhindert den Verlust von Datenbits am Anfang des Frames. Die Funktionen unterscheiden sich bei der DIX-Gruppe und IEEE Komitee nicht. Der Aufbau der Präambel weichen leicht voneinander ab. Der Aufbau der Bytes nach DIX und IEEE 802.3 ist identisch. Die ersten acht Bytes unterscheiden sich in der Kombination der Nullen und Einsen nicht(siehe Abbildung 3.5), jedoch werden bei der IEEE die ersten 7 Bytes als Präambel bezeichnet und das darauf folgende Byte als Start-of- Frame Delimiter (SFD). Die Unterteilung der Präambel dient zur eindeutigen Differenzierung zwischen der eigentlichen Synchronisation und der Kennzeichnung des Frameanfangs. Um diesen Unterschied zu verdeutlichen wurde bei der IEEE der SFD als eigenständiges Byte hervorgehoben.

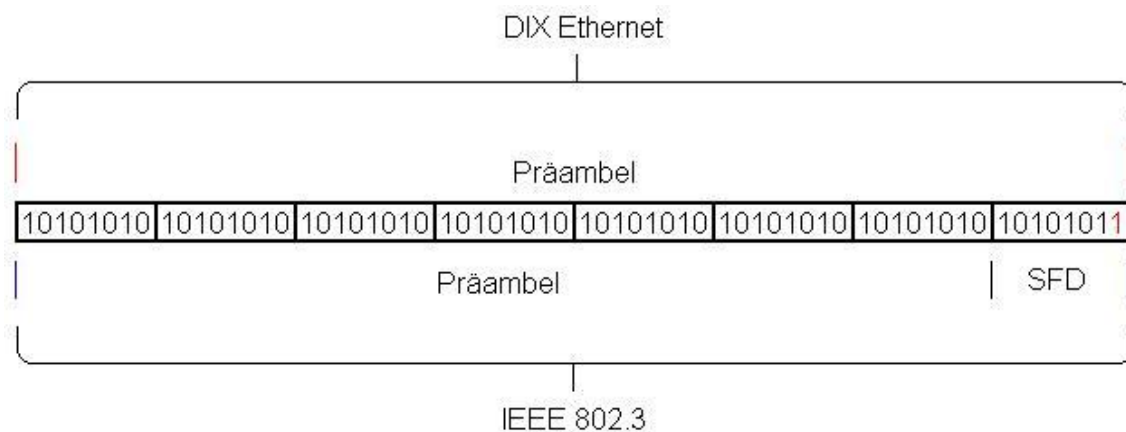


Abbildung 3.5: Präambel

### 3.3 Fast Ethernet

Das von der IEEE-802.3u-Gruppe entwickelte Fast Ethernet, wurde 1995 veröffentlicht und steigerte die Datenrate um den Faktor 10 auf 100MBit/s. Auf Grund dieser Performanzsteigerung und der Kompatibilität der unterschiedlichen Ethernet-Komponenten untereinander konnte sich der Fast Ethernet-Standard gegenüber anderen etablieren.

Die Erweiterung betreffen die physikalische Schicht, bei der es zwei unterschiedliche Varianten zu unterscheiden gilt. Die erste ist der 100Base-TX Standard, bei dem das Übertragungsmedium ein TP- Kabel der Kategorie 5 ist, und der 100BaseFX Standard, dessen Übertragungsmedium ist ein Glasfaserkabel. Neben diesen wurden noch zwei weitere mit in den Standard aufgenommen, die sich aber nicht am Markt durchsetzen konnten.

Die Unterschiede zum 10Base-T Standard liegen in der physikalischen Schicht. Beim Fast Ethernet wurde der PLS(Physical Signaling Service) durch den PCS(Physical Coding Sublayer) ersetzt. Damit setzt sich die physikalische Schicht aus dem PMD(Physical Medium Dependent), PMA (Physical Medium Attachment) und dem PCS zusammen.

Außerdem wurde das AUI(Attachment Unit Interface) des 10MBit-Ethernet durch das MII(Media Independent Interface)ersetzt, dass zusammen mit dem Reconciliation Layer die Verbindung von der physikalische Schicht zur MAC- Schicht herstellt. Damit ermöglicht, wurde eine für die Mac-Schicht immer gleiche Darstellung der Daten und die Adaptierung von 10MBit beziehungsweise 100MBit Transceivern.

Das MII überträgt seine Daten jeweils in einer Breite von 4Bit in jede Richtung, mit einer Frequenz von 25MHz bei einer Übertragungsrate von 100MBit/s. Diese Dateneinheit von 4Bit wird dabei als Nibble bezeichnet. Somit wird ein Datenbyte in zwei Nibble geteilt und vom MII übertragen. Des Weiteren können über das MII Managementdaten, die Informationen zum Senden und Empfangen liefern, ausgetauscht werden.

#### 3.3.1 Physical Coding Sublayer

Die Aufgabe des PCS ist, die Realisierung der 4B/5-Kodierung, Generierung des Carrier Sense und der Kollisionserkennung für die MAC- Schicht und die Formatierung der Daten zur Übertragung eines seriellen Stroms als binde Glied zwischen PMA und MII.

#### 3.3.2 Physical Medium Attachment

Hier werden Kontrollsignale generiert die eine Aussage über die Verfügbarkeit des PMD zulassen und die Synchronisation für die Auto- Negotiation- Funktion vornehmen. Außerdem

ist die Schicht für die Rückgewinnung des Taktes, aus den eingehenden NRZI- kodierten Daten, zuständig.

### 3.3.3 Physical Medium Dependent Sublayer

Als unterste Schicht in der physikalischen Schicht, stellt der PMD die Verbindung zum Übertragungsmedium her, das können TP- Kabel als auch Glasfaserkabel sein.

### 3.3.4 4B/5B Kodierungsverfahren

Die Einführung der 4B/5B-Kodierung war nötig um das TP- Kabel auch weiterhin als Übertragungsmedium nutzen zu können. Der Grund dafür liegt bei der Manchester-Kodierung, die eine Übertragungsfrequenz von 100MHz vorausgesetzt hätte und zu massiven Problemen bei der Verwendung von Kupferkabeln führen würden.

Als Grundlage wird die NRZI- Kodierung verwendet, die zum Zeitpunkt der Entwicklung des Fast Ethernet Standards bereits beim Fiber Distributed Data Interface (FDDI) angewandt wurde. Dabei wird eine logische Eins durch einen Flankenwechsel von Null auf Eins und eine logische Null durch fehlenden Flankenwechsel dargestellt. Das Problem was daraus ergibt ist, dass es bei fehlenden Pegelwechsel nur mangelhafte Informationen über den Takt gibt und der Gleichstromanteil zu groß wäre. Deshalb wird beim Fast Ethernet auf die 4B/5B-Kodierung zurückgegriffen. Aus dem Datennibble wird eine 5Bit lange Codegruppe erzeugt. Die bewirkt, dass das Auftreten von zu langen Null-Sequenzen verhindert wird und somit die Taktinformationen im Datenstrom erhalten bleiben. Ein weiterer Vorteil des Verfahrens ist die Gewinnung zusätzlicher Bitkombinationen, die eine Aussage über die Qualität der Verbindung zulassen.

Die neu gewonnen 5B-Symbole werden auf verschiedene Weise genutzt, zum einen dienen sie der Link-Erkennung und zum anderen Synchronisation gegenüberliegender Stationen. Außerdem wird vor jedem Frame ein SFD(Start- of- Frame Delimiter) und am Ende jedes Frames ein ESD(End- of- Stream Delimiter) angefügt.



# 4 Avionic Full Duplex Switched Ethernet End System

In diesem Kapitel werden die Unterschiede zum Fast Ethernet Standard erklärt. Die gesamten Informationen in diesem Kapitel basieren auf der ARINC Spezifikation 664 Part 7 ([ARINC \(2005\)](#)).

## 4.1 Allgemeine Beschreibung des AFDX- Netzwerkes

Avionics Full Duplex Switched Ethernet kurz AFDX beschreibt ein deterministisches Modell eines Netzwerksystem basierend auf dem IEEE 802.3 & IP Standard.

Um zu einem deterministisches Netzwerk zu gelangen, bedarf es einer Netzwerkhierarchie wie in in Abbildung 4.1 dargestellt. Dabei wird das IEEE 802.3 & IP Konforme- Netzwerk in Sub- Netzwerke unterteilt.

Diese „Spezifischen Netzwerke“ sind erlaubte Vereinfachungen, des Komplexen IEEE 802.3 & IP Protokollstacks, um die Analyse der Flugsicherheit und die Zertifizierung zu erleichtern.

Die „Deterministischen Netzwerke“ wiederum, sind ein Bestandteil der „Spezifischen Netzwerke“. Die wichtigste Eigenschaft dieser Netze ist der Determinismus. Diese Eigenschaft muss mathematisch als auch durch Tests beweisbar sein.

## 4.2 Einleitung

Das AFDX- Netzwerk unterscheidet zwei Hardwarekomponenten auf der die gesamte Kommunikation basiert. Das für diese Arbeit ausführlich beschriebene End System und den Switch. Auf die Eigenschaften des Switch wird nicht näher eingegangen, er wird nur der Vollständigkeit halber erwähnt.

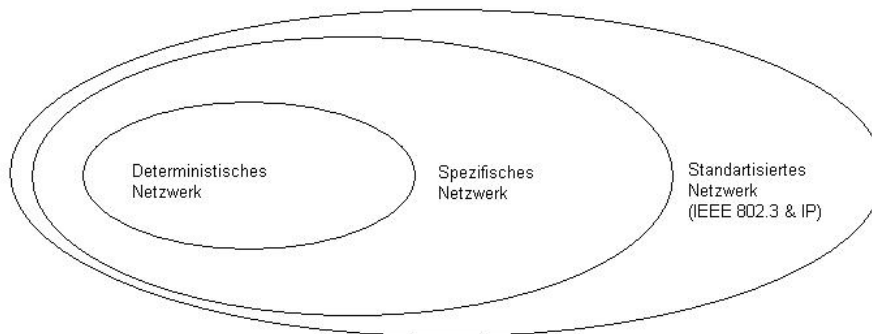


Abbildung 4.1: Netzwerk Hierarchie

Die Aufgabe eines End Systems ist die Bereitstellung von Diensten, die den sicheren und zuverlässigen Austausch von Informationen zwischen verteilten Softwarekomponenten garantieren.

Die Methode des „Quality of Service“ bietet eine Möglichkeit den Datentransfer zu kategorisieren, um sicher zustellen dass bestimmte Daten beziehungsweise Informationen zu jeder Zeit ungehindert durch das Netzwerk gelangen, unabhängig von sich konkurrierenden Nachfragen.

Im Bereich der Luftfahrt, speziell bei den in der Luftfahrt verwendeten Netzwerken, gibt es keinen Anlass zwischen verschiedenen Klassen oder Kategorien von Daten beziehungsweise Informationen zu unterscheiden. Denn jede Übertragungsnachfrage muss beantwortet werden. Jede Anfrage wird dabei nach einer garantierten maximalen Zeit beantwortet. Aufgrund dieser Anforderung wird beim AFDX von nur einer Service Klasse ausgegangen, dem „Garantierten Service“.

Der „Garantierten Service“ soll stabil, mathematisch beweisbar sein und einen Datenstrom in seiner End- to- End Laufzeit begrenzen. Durch die begrenzte Laufzeit lässt sich die Datenübertragungsrate auf der Sicherungsschicht (Link Level) garantieren.

Damit unterstützt der „Garantierte Service“ folgende Eigenschaften, einen nach oben begrenzte Laufzeit und eine konstante Datenübertragungsrate zwischen einem sendenden Knoten und einem oder mehr empfangenden Knoten.

### 4.3 Interoperabilität und Determinismus der Media Access Control Schicht

### 4.3.1 Virtueller Link

Diese virtuelle Verbindung ist ein Kommunikationsobjekt, das eine logische unidirektionale Verbindung von einem End System zu einem oder mehreren End Systemen definiert, und der eine feste Bandbreite zugewiesen wird. Die Art der Nutzung des Virtuellen Links(VL) ist dabei zu vernachlässigen und darf in keiner Weise die Bandbreite anderer Virtueller Links beeinflussen.

Jede virtuelle Verbindung eines End Systems(ES) wartet bis die nach gefragten Daten von der entsprechenden Partition geliefert werden, von der sie benötigt werden. Das gilt sowohl für das senden als auch für das empfangen von Daten.

Der Prozess des Virtuellen Links beschreibt eine Datenflusskontrolle. Der Mechanismus reguliert den Datenfluss der verschiedenen Quellen eines End Systems. Das wird genutzt, um einzelne Partitionen von der Netzwerkschicht abzugrenzen. Daher darf ein VL nicht von mehr als einer Quelle genutzt werden.

### 4.3.2 Datenfluss/Verkehrs Kontrolle

Ein End System wird durch zwei Parameter bestimmt, der Bandwidth Allocation Gap(BAG) und dem Jitter. Die BAG repräsentiert das kleinste mögliche Zeitintervall zwischen zwei aufeinander folgende Pakete die zu einem Datenstrom gehören.

Es werden grundsätzlich keine Frames übertragen, auch dann nicht wenn ein VL verfügbar ist.

Um die BAG für jeden VL zu garantieren, wird die Übertragung der Datenströme reguliert.

### 4.3.3 Scheduling

Bei einem sendenden End System mit mehreren Virtuellen Links wird das Scheduling über einen Scheduler Multiplexer mit vor gestellten Verkehrsregulator realisiert. Dieser Verkehrsregulator übergibt dem Scheduler die Datenströme ohne Jitter und sorgt dafür, dass jeweils nur ein Paket je BAG übertragen werden kann. Die Ausgabe vom Scheduler erfolgt innerhalb der BAG und unterliegt einer maximalen Jitter, die nicht überschritten werden darf. Werden Pakete direkt aufeinander folgend gesendet und innerhalb der Grenzen des maximalen Jitter, wird die Datenübertragungsrate voll ausgenutzt.

Das ES steuert somit die Übertragung der Daten auf Basis der Virtuellen Links. Der Verkehrsregulator, dem alle Virtuellen Links in einem End System zugehören, reguliert mit der Traffic Shaping Function(Verkehrs Umformungsfunktion) die Ausgabe zum Scheduler, dass

die Basis für die deterministische Analyse darstellt. Das erfordert exakte Kenntnis der Datenübertragungscharakteristika.

Damit entspricht die maximal nutzbare Datenübertragungsrate ( $D_{max}$ ) dem Verhältnis von maximaler Virtueller Link Paketgröße ( $L_{max}$ ) zur Bandwidth Allocation Gap.

$$D_{max} = \frac{L_{max}}{BAG}$$

#### 4.3.4 Mac-Beschränkungen

Um den Verlust von Frames zu vermeiden, die durch das unerwartete Signalstörungen auftreten können, und um die Interframe Gap (IFG) zu gewährleisten, muss das End System einige Eigenschaften bereitstellen.

Das Empfangen der Frames mit der vollen Übertragungsrate des Mediums, und diese Frames der richtigen Partition mit der gleichen Übertragungsrate zur Verfügung zu stellen. Es ist außerdem erforderlich, dass der MAC- Layer, aufeinander folgende Frames in der richtigen Reihenfolge übermittelt.

Für den kürzesten Frame würde dies folgendes bedeuten:

$$64\text{Bytes}(\text{Frame}) + 12\text{Bytes}(\text{IFG}) + 7\text{Bytes}(\text{Preamble}) + 1\text{Byte}(\text{SFD}) = 84\text{Bytes}$$

Diese 84 Byte stellen die minimale Länge einer Nachricht dar. Soll diese bei 100Mbps/s übertragen werden ergibt sich daraus folgende Übertragungszeit.

$$\frac{84\text{Bytes}}{12500000\text{Bytes/s}} = 0,00000672\text{s}$$

Je Frame ergibt sich also eine Übertragungszeit von

$$6,72\mu\text{s}$$

Damit ergibt sich für den MAC- Layer.

$$\frac{12500000\text{Bytes/s}}{84\text{Bytes}} = 148809\text{Frames}$$

#### 4.3.4.1 Jitter

Der Jitter im AFDX Netzwerk beschreibt die Verzögerung die beim senden eines Frames während seines Zeitfensters auftritt. Diese Zeiten dürfen ein gewisses Maximum nicht überschreiten.

Die folgenden Formeln beschreiben das Intervall der Verzögerungszeit. Die in der ersten Formel heran gezogenen  $40\mu s$  sind das technologische Minimum,  $Nbw$  ist die Bandbreite des Übertragungsmediums und  $L_{max}$  gibt die Anzahl der Octets wieder. Die 20 Bytes setzen sich aus Präambel, Start-of-Frame Delimiter(SFD) und Interframe Gap(IFG) zusammen.

$$\left\{ \begin{array}{l} max\_jitter \leq 40\mu s + \frac{\sum_{i \in \{setofVLs\}} (20Bytes + L_{max}Bytes) * 8Bits/Bytes}{Nbwbit/s} \\ max\_jitter \leq 500\mu s \end{array} \right.$$

Die sich aus der ersten Formel ergebenden Werte beschreiben ein ideales Netzwerk ohne Burst und Fragmentierung der Frames. In der zweiten Gleichung wird ein Maximum von  $500\mu s$  bestimmt, die Zeit muss eingehalten werden um den Determinismus für das gesamte Netzwerk garantieren zu können.

Die oben genannten Werte sind von fundamentaler Bedeutung um den Determinismus im AFDX- Netzwerk zu demonstrieren.

Die sich aus den Formeln ergebenden Zeiten haben Einfluss auf die gesamte Verzögerung des sendenden End Systems. Da auch hier zu unterscheiden ist, wie sich das Netzwerk verhält, werden 2 Situationen betrachtet.

Der erste Fall beschreibt die Verzögerung bei gleichen Packet größen, die nicht fragmentiert werden. Die maximale Verzögerung eines Virtuellen Links, der unter Berücksichtigung dieser Eigenschaften Nachrichten sendet, wird durch die Bandwidth Allocation Gap(BAG) , Jitter ( $max\_jitter$ ) und der technologischen Verzögerung(TLT) während des Sendens ermittelt.

$$max\_latenz_i \leq BAG_i + max\_jitter + TLT$$

Für den zweiten Fall wird davon ausgegangen das die Host Partition Pakete unterschiedlicher Länge sendet. Die Nachrichten können auch die maximale Größe von 1522 Bytes überschreiten und werden fragmentiert, wodurch es zu einer weiteren Verzögerung kommt. Die maximale Verzögerung( $max\_latency_i$ ) muss in diesem Fall um die Anzahl der Pakete( $p$ ) erweitert werden. Dadurch ergibt sich folgende Gleichung:

$$max\_latenz_i(frame_p) \leq p * BAG_i + max\_jitter + TLT$$

### 4.3.5 MAC- Adressierung

Ein Virtueller Link wird allein über die MAC- Zieladresse identifiziert. Die MAC- Zieladresse im AFDX- Frame versteht sich als Gruppen- und Lokale Adresse und stellt somit eine Multicast- Adresse dar. Die MAC- Quelladresse eines AFDX- Frame wird als Unicast- Adresse verstanden, die zur Identifizierung des Ethernet- Interface genutzt wird.

#### 4.3.5.1 MAC Zieladresse

Diese 48 Bit lange Multicastadresse setzt sich aus 2 Feldern zusammen, einem „Constant Field“ und dem „Virtual Link Identifier“.

Das „Constant Field“ ist für jedes End System im AFDX- Netzwerk gleich. Das Least Significant Bit(LSB) des ersten Bytes identifiziert die Gruppen Adresse(GAA) und ist immer 1. Die Gruppen Adresse wird genutzt, um den Datenaustausch zwischen den End Systemen zu realisieren. Das darauf folgende Bit, des ersten Bytes identifiziert die Lokale Adresse(LAA), auch diese ist immer 1.

Der „Virtual Link Identifier“ ist 16 Bit lang, kann frei gewählt werden, und unterliegt keinen Restriktionen.

#### 4.3.5.2 MAC Quelladresse

Die Unterteilung der MAC- Quelladresse ist IEEE 802.3 konform und enthält 2 Adresstypen, die Individuelle und die Lokale Adresse. Die Struktur des Ethernet- MAC- Frames wird durch 4 Felder definiert.

Einem 24 Bit langen „Constant Field“, dessen LSB im ersten Byte die Individuelle Adresse kennzeichnet und die immer 0 ist. Das erste Bit, nachdem LSB, im ersten Byte identifiziert die lokale Adressen deren Wert immer 1 ist.

Das „User\_Defined\_ID Field“ ist 16 Bit lang und wird verwendet, um einem Host im AFDX- Netzwerk eine IP- Adresse, die einmalig vergeben wird, zuweisen zu können.

Die „Interface\_ID“ ist 3 Bit lang und spiegelt das Interface wieder welches im Redundanten AFDX- Netzwerk benutzt wird.

Am Ende der MAC- Quelladresse steht ein „Constant Field“ dessen 5 Bits alle 0 sind.

### 4.3.6 Redundanz Konzept

End Systeme kommunizieren über unabhängige und redundante Netzwerke, sodass der Datenstrom gegen Ausfall einzelner Netzwerk- Komponenten geschützt ist. Der Vorteil besteht darin, dass selbst bei Verlust eines Netzwerkes, die End Systeme weiterhin miteinander kommunizieren können.

Eine Partition stellt dem End System einige Daten bereit, die dann dem Kommunikationsstack übergeben werden. Hier wird dem Frame eine Sequenznummer hinzugefügt, diese Sequenznummer wird bei jedem nachfolgenden Frame inkrementiert. Zweck dieser Sequenz ist es, die auf der Empfängerseite eingehenden Frames wieder in die richtige Reihenfolge zu bringen und Duplikate zu entfernen, um sie danach an die Empfangspartition zu übergeben.

Bei dieser Methode bemerkt die Partition nichts von der darunter liegenden Netzwerkredundanz, das die Implementation einer Schnittstelle zwischen Kommunikationsstack und Partition vereinfacht.

Beim Standardverfahren wird jedes Frame über beide Netzwerke gesendet. Nach Empfang analysiert ein Algorithmus des Kommunikationsstack die eingehenden Frames und wählt das erste richtige aus. Das bedeutet, das erste Frame das empfangen wird, wobei es nicht von Bedeutung ist vom welchen der beiden Netzwerke, und die richtige Sequenznummer aufweist gilt als akzeptiert. Diese Frames werden dann an die Empfangspartition übermittelt. Frames mit gleicher Sequenznummer werden verworfen.

Die Integritäts- Prüfung findet vor der Redundanz Prüfung statt, erkannt und entfernt werden invalide Frames. Die redundanten Frames werden ausschließlich bei der Redundanz Prüfung erkannt und entfernt.

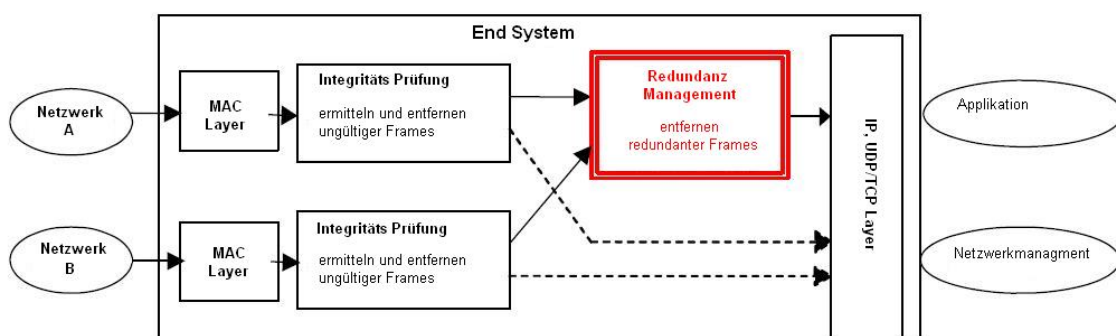


Abbildung 4.2: Integritäts- und Redundanz Prüfung

In den Abbildungen 4.3, 4.4, 4.5, 4.6 wird gezeigt, wie sich die Redundanz Prüfung bei unterschiedlichen Netzwerk Ereignissen verhält.

### 1. Fehlerhaft übertragenes Frame

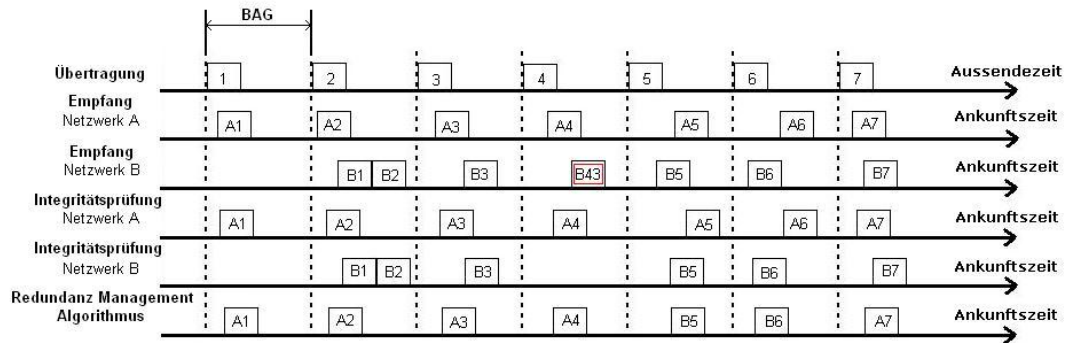


Abbildung 4.3: Fehlerhaft übertragenes Frame

### 2. Verlust eines Frame

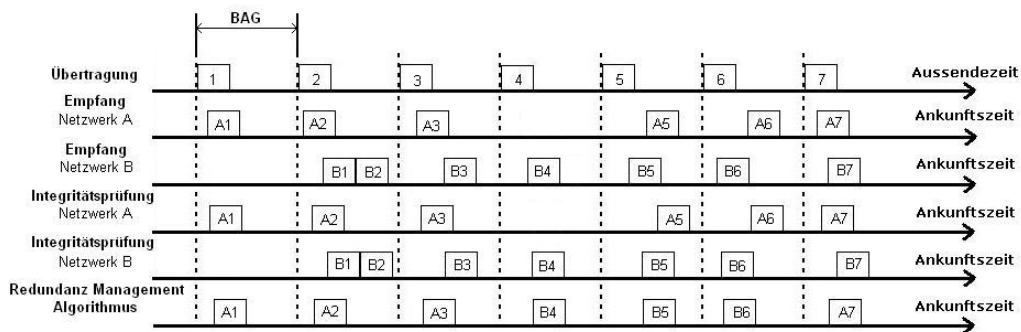


Abbildung 4.4: Verlust eines Frame



### 3. Reset des Senders

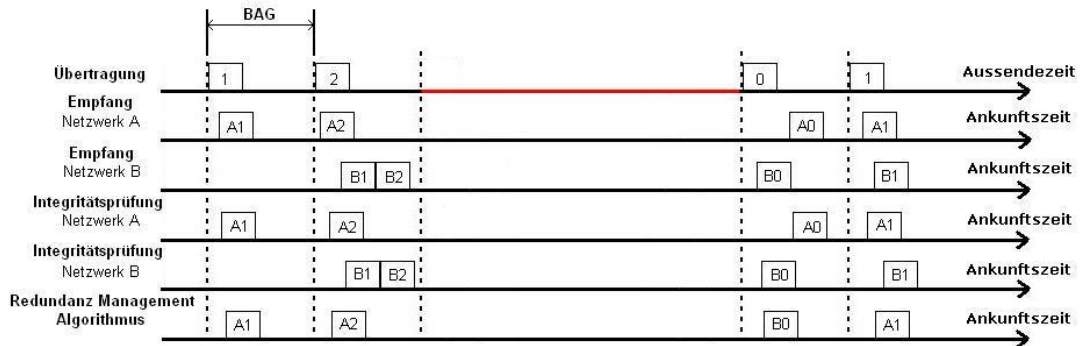


Abbildung 4.5: Reset des Senders

### 4. Fehler im Switch

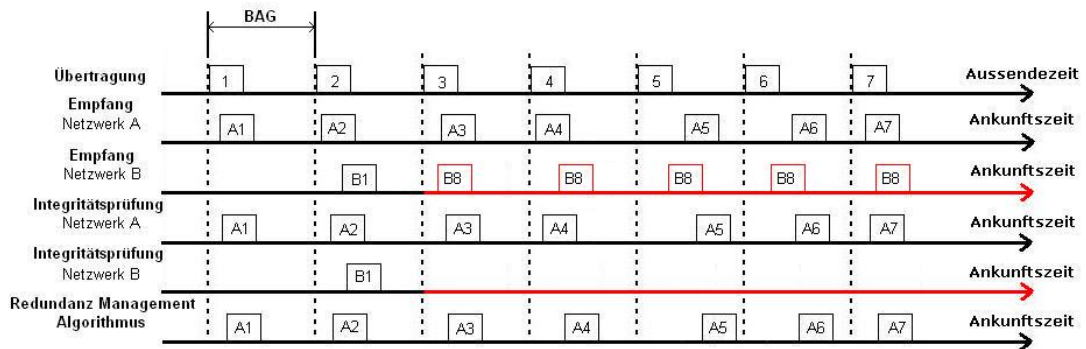


Abbildung 4.6: Fehler im Switch

#### 4.3.6.1 Sequenznummern des sendenden End Systems

Sobald das End System Daten über einen Virtuellen Link überträgt wird dem Frame eine Sequenznummer hinzugefügt. Die Sequenznummer ist ein 8 Bit langes Feld innerhalb des Ethernet- Frame. Die Länge des Feldes genügt dem Anspruch redundante Frames zu erkennen. Das Octet kann somit zwischen  $2^8$  Werten unterscheiden, das ist ausreichend, um den Worst Case Fall , der beim übermitteln der Frames auftreten kann, zu genügen. Der maximale Offset der Sequenznummer ist abhängig von der maximalen Netzwerk- Übertragungszeit(SkewMax) und der Bandbreitenzuordnung (BAG).

Die Werte für SkewMax und BAG wurden so gewählt das sie dem Worst Case entsprechen.

SkewMax = 5 ms , BAG = 1 ms

$$\text{int}[SkewMax/BAG] + 2 = 7$$

Für jeden Virtuellen Link gilt, das der Initialwert der Sequenznummer 0 ist. Der Wert ist für die Initialisierung vorbehalten und wird somit nur beim Neustart des End Systems erneut verwendet. Die Frame Sequenznummer wird um 1 inkrementiert und kann bis auf maximal 255 erhöht werden. Das darauf folgende Frame erhält als Sequenznummer die 1. Die Inkrementierung um 1 , ermöglicht es fehlende Frames leicht zu entdecken und einen Fehler in der Sequenz zu bemerken.

Die Frame Sequenznummer ist wie bereits erwähnt ein Teil des MAC- Payloads und ist vor dem MAC- CRC- Feld zu finden. Die redundanten Kopien der Frames werden mit einer maximalen Verzögerung von 0,5ms gesendet, das ist wichtig um den Algorithmus auf Empfänger möglichst einfach zu halten.

#### 4.3.6.2 Sequenznummern des empfangenden End Systems

- Integritätsprüfung

Befindet sich das Netzwerk in einem Fehlerfreien Zustand, wird die Integritätsprüfung kein Frame verwerfen und sie an den Redundanz Manager weiterleiten. Für den Fall eines Fehlers in der Sequenz, werden die betreffenden Frames während der Integritätsprüfung verworfen und das Netzwerkmanagement benachrichtigt.

Die Sequenznummer eines Frames müssen um gültig zu sein, im Intervall von  $[PSN + 1, PSN + 2]$  liegen. Die vorhergehende Sequenznummer (PSN) beschreibt den

Wert der im vorher empfangen Frame enthaltenen Sequenznummer. Der „+“Operator ist nötig um den Umbruch von 255 auf 1 zu ermöglichen.

Die Funktion erhöht die integrative Robustheit des Systems, da sie Frames die verzögert das End System empfängt entfernt, einzelne Fehlerhafte Frames löscht und damit den Einfluss von Switchen die Störung aufweisen reduziert.

Außerdem akzeptiert die Integritätsprüfung folgende Sonderfälle:

- Empfangs- Sequenznummer ist 0
- das erste Frame nach einem Reset

Frames die nicht den oben genannten Bedingungen entsprechen werden verworfen.

- Redundanz Management

Es müssen Rahmenbedingungen geschaffen werden, um die Funktion des Redundanz Managements zu sichern. Wichtigster Aspekt ist, dass das Netzwerk entsprechend seiner Konventionen arbeitet.

Die Konfiguration des Redundanz Managements basiert auf dem SkewMax Parameter, der Parameter gibt an wie viel Zeit maximal vergehen darf bis 2 redundante Frames über das Netzwerk übertragen wurden. Die Zeit ist abhängig von der vorliegenden Netzwerk-Topologie und muss aus diesem Grund, für jedes neues Netzwerk neu bestimmt werden.

Der SkewMax Parameter wird bei der Konfiguration der Virtuellen Links festgelegt. Dabei werden zwei Arten von Virtuellen Links wie folgt definiert.

- Redundante Virtuelle Links(RVL): 2 identische Frames werden über beide Netzwerke übertragen.
- Nicht Redundante Virtuelle Links(NRVL): Frames werden entweder über das eine oder andere Netzwerk übertragen

Bezogen auf die oben genannten Virtueller Link Definitionen, leitet das End System die empfangenen Daten in folgender Art und Weise weiter:

- RVL bei aktiviertem Redundanz Manager: eins der beiden Frames wird an die Partition weitergeleitet
- RVL bei deaktiviertem Redundanz Manager: beide Frames werden an die Partition weitergeleitet
- NRVL: Frame wird ohne Prüfung an die Partition weitergeleitet, Redundanz Manager erhält keine Kopie des Frames und greift somit nicht

Der Redundanz Management Algorithmus entscheidet einzig auf Grund der Redundanz Sequenznummer ob ein Frame abgelehnt oder angenommen wird. Der Redundanz Manager stellt sicher, dass die weitergeleiteten Frames in aufsteigender Reihenfolge bei der Partition eintreffen. Da der Redundanz Manager die Frames nicht neu ordnet, kann es beim weiter leiten an die Partition zum Verlust einzelner Frames, innerhalb der Sequenz, kommen. Der Grund hierfür ist, dass der SkewMax Wert nach Erhalt eines Frames überschritten werden kann. Wenn es zu dieser Situation kommt, akzeptiert der Redundanz Manager das Frame unabhängig von dessen Sequenznummer.

## 4.4 Interoperabilität des IP Layers

Die Kommunikation zwischen der Vermittlungsschicht und der darüber liegenden, unterliegt keinen besonderen Beschränkungen. Es gibt 2 Konventionen auf die bei der Umsetzung zu achten ist.

Übermittelt ein AFDX- Port Daten an einen Applikationsport, so erfolgt das in einer 1-N Beziehung. Ein AFDX- Port kann mit einem oder mehreren Applikationsports in Beziehung stehen. Die einzige Ausnahme ist das sich 2 AFDX- Ports nicht einen Applikationsport teilen.

Findet die Kommunikation in die andere Richtung statt, also vom Applikationsport zum AFDX- Port, ist dies eine 1-1 Beziehung. Auch hier ist darauf zu achten, dass sich nicht 2 Applikationsports einen AFDX- Port teilen.

### 4.4.1 Avionik Services

Bei einem AFDX- End System gibt es 2 verschieden Möglichkeiten Daten von einer Partition zu übermitteln. Da die Kommunikation ausschließlich über Ports erfolgt, werden diese in 2 Klassen eingeteilt, in die Kommunikationsports und die Service Access Point(SAP)- Ports.

#### 4.4.1.1 Kommunikationsports

Es werden 2 verschieden Arten von Services von den Kommunikationsports unterstützt, Sampling und Queueing. Als Transportprotokoll wird User Datagram Protocol(UDP) verwendet, um die Kommunikation möglichst effizient zu halten.

- Avionik Sampling Service

- Übertragung

Die Sampling Ports unterstützen keine IP- Fragmentierung was zur Folge hat, dass jede Nachricht, die über Sampling Ports übermittelt wird, maximal die Größe des Payloads die im dazu gehörigen Virtuellen Link festgelegt wurde haben darf. Der Sampling Service basiert auf einer unidirektionalen Multicast Verbindung, um Daten von einem Sender an einen oder mehrere Empfänger zu übertragen. Diese Verbindungslose- und Quittierungsfreie- Kommunikation ermöglicht es den Sampling Service einfach zu halten. Es ist nicht notwendig eine Flusskontrolle oder eine Fehleranalyse bei der Übermittlung zu implementieren, diese werden durch die Virtuellen Links verwaltet.

- Empfang

Die letzte Nachricht die von einem Sampling Port empfangen wurde, kann von verschiedenen Applikationsport gelesen werden. Um die gespeicherte Nachricht vom Sampling Port zu lesen, muss der Partition der Port bekannt gemacht werden. Sollten neue Nachrichten im Speicher des Sampling Port bereit liegen, wird das den Partitionen, die mit dem Port verbunden sind, mitgeteilt.

- Avionik Queueing Service

Der Queueing Service ist verbindungslos und es bedarf daher keiner Quittierung. Er wird genutzt, um Nachrichten unterschiedlicher Länge über ein und denselben Kommunikationsport zu übertragen. Um eine Sequenz von Nachrichten zu garantieren werden diese in einem FIFO-Buffer hinterlegt, dies gilt für das Senden als auch für das Empfangen. Da jede Instanz des Queueing Service 8k Octets an Applikationsdaten verarbeiten können muss, ist auch der Umgang mit IP- Fragmenten erforderlich.

- Übertragung

Bei der Fragmentierung müssen die Pakete gemäß ihrer Reihenfolge an das AFDX- Netzwerk übergeben werden. Im Fall eines Speicherüberlaufs wird der Partition, die es betrifft, ein Fehler signalisiert und das Frame verworfen.

- Empfang

Der Partition werden nur Daten übergeben, die vollständig in der Queue hinterlegt worden. Das bedeutet im Fall von Fragmentierung werden die Daten erst dann der Partition zur Verfügung gestellt, wenn alle Teile der Nachricht im FIFO-Buffer bereit liegen. Für den Fall eines Speicherüberlaufs wird auch hier das Frame verworfen.

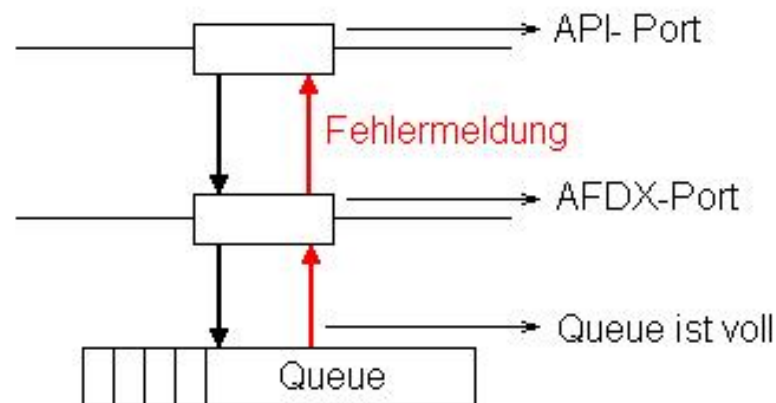


Abbildung 4.7: Speicherüberlauf

#### 4.4.1.2 SAP-Port

- Service zum IEEE 802.3 Netzwerk

Ein End System mit einem SAP- Port weist folgende Eigenschaften auf:

- SAP- Ports können für die Kommunikation im AFDX- Netzwerk genutzt werden
- Zugang zum IEEE- Netzwerk könne über Gateways oder Router realisiert werden
- jede Instanz ist in der Lage 8k Octets an Daten zu verarbeiten
- optional ist es möglich den IP- Layer direkt zu erreichen, das wird über eine TCP - Verbindung, eines speziell konfigurierten SAP- Port, realisiert

Das End System kann sich auf Anfrage aus dem IEEE- Netzwerk, die entsprechende IP- Nummer und Port- Nummer zugänglich machen, um eine Kommunikation aufzubauen.

- Fehler Management

Der SAP- Port wird auf der Empfängerseite überwacht. Für den Fall eines Speicherüberlaufs, wird die Partition, die die Nachricht erhalten sollte, informiert und das Frame verworfen.

- Datei Übertragungsservice

Zur Übertragung von Dateien wird das Trivial File Transfer Protokoll(TFTP) verwendet. Jede Instanz des Service muss 8 Octets an Daten verarbeiten können und unterliegt Request for Comments (RFC) Definitionen die in der Tabelle 4.1 aufgeführt sind.

RFC	Titel	Kategorie
783	TFTP Protokoll (Revision 2)	Standard, aktualisiert in RFC 1350
1123	Anforderungen für Internet Hosts	Standard
1350	TFTP Protokoll (Revision 2)	Standard
2347	TFTP Optionen	Standard, aktualisiert die RFC 1350
2348	TFTP Blockgröße	Standard, aktualisiert die RFC 1350
2349	TFTP Timeout Intervall und Transfergröße Optionen	Standard, aktualisiert die RFC 1350
1785	TFTP Optionen Negotiationsanalyse	Information, aktualisiert die RFC 1350

Tabelle 4.1: TFTP RFC

#### 4.4.1.3 Sub- Virtuelle Links

Ein Virtueller Link kann sich aus mehreren Sub-Virtuellen Links zusammensetzen, wie in Abbildung 4.8 dargestellt. In diesem Fall setzt sich der Virtuelle Link ausschließlich aus Sub-Virtuellen Links zusammen. Jeder dieser Sub-Virtuellen Links verarbeitet seine Informationen über einen eignen First In-First Out(FIFO)-Buffer. Die einzeln FIFO-Buffer der Sub-Virtuellen Links werden unter Verwendung des Round Robin Verfahrens ausgelesen und im dem Haupt- FIFO- Buffer des Virtuellen Link gespeichert.

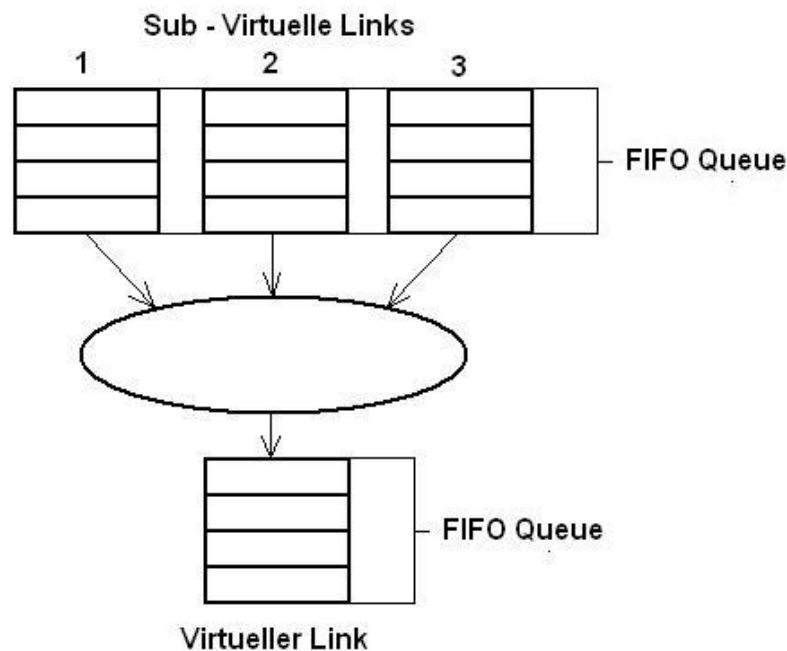


Abbildung 4.8: Sub- Virtuelle Links mit FIFO Queue

Angewendet wird das Round Robin Verfahren auf der MAC- Frame Basis. Es ist darauf zu achten, das fragmentierte Pakete die Priorität beim Zugriff auf die Sub-Virtuellen Links FIFO-Buffer erhalten. Der Virtuelle Link FIFO-Buffer kann maximal 4 Sub-Virtuelle Links FIFO-Buffer verwalten. Ausgelesen werden diese FIFO-Buffer nur dann, wenn sich Frames in ihm befinden. Dabei wird jeweils nur ein Frame je Bandwidth Allocation Gap(BAG) an den Haupt-FIFO- Buffer übergeben. Sobald das Frame übertragen wurde, wird die Sequenz unterbrochen bis die BAG endet, anschließend startet die Sequenz erneut, dann mit dem nächsten Sub- FIFO- Buffer.

Da ein Sub- Virtueller Link immer an einem Virtuellen Link gebunden ist, kann dessen Sub-Virtuellen Link FIFO- Buffer auch immer nur von dem dazugehörigen Virtuellen Link- Buffer gelesen werden.

Die IP- Fragmentation wird im IP- Layer vollzogen, dadurch kann verhindert werden, das Sampling Nachrichten durch lange Queueing Nachrichten verzögert werden. Das Round Robin Verfahren kann durch die Verwendung von mehreren Sub- Virtuellen Links sehr effizient zum Einsatz gebracht werden, da jedem Sub-Virtuelle Link nacheinander Fragmente vom IP- Layer in seine Queue einfügen werden. Das setzt voraus, dass sich die Sub-Virtuellen Links unterhalb des IP- Layer befinden.

## **4.4.2 End System Kommunikationsstack**

### **4.4.2.1 MAC Profil**

Die Sicherungsschicht eines End Systems basiert auf einer Voll- Duplex- Ethernet Verbindung wie sie im IEEE 802.3 Standard definiert ist. Damit müssen alle vom End System generierten Frames mit diesem Standard konform sein. Für den Fall, dass es bei Verbindung zur Bitübertragungsschicht zu Fehlern kommt, wird die Übertragung nicht abgebrochen. Das verhindert, das Frames die beim auftreten des Verbindungsverlustes bereits im Speicher lagen nach diesem nicht mit einer zu hohen Verzögerung gesendet werden. Außerdem wird auch die Stabilität des gesamten Netzwerkes verbessert, da sich Fehler im Switch nicht auf End Systeme ausweiten.

AFDX- Frames werden an die höher liegen Schichten weitergeleitet, wenn die Frame Sequenz Nummer und die Cyclic Redundancy Check(CRC) als richtig erkannt wurden.

### **4.4.2.2 IP Profil**

Die Paketstruktur entspricht der von IPv4 (Internet Protocol Version 4), dessen Paketlänge gemäß Spezifikation zwischen 21 - 1500 Bytes liegen kann. Im AFDX- Netzwerk liegt sie



zwischen 21 - 1499 Bytes, zurückführen lässt sich das auf die Sequenznummer für das Redundanz Management.

## 4.5 Interoperabilität des Netzwerk Layers

### 4.5.1 Adressierung

#### 4.5.1.1 Einleitung

Ein Datenfluss im AFDX- Netzwerk wird durch UDP/TCP Ziel- und Quelladresse, MAC Ziel- und Quelladresse und die physikalische Ethernetverbindung vom empfangenden End System identifiziert.

Das filtern von Frames im empfangenden End System wird der Art vollzogen, das nur als korrekt identifizierte Frames an den entsprechenden Kommunikations- bzw. SAP- Port weiter geleitet werden. Ob ein Frame gültig ist oder nicht, ergibt sich aus der Analyse des Quintupel.

#### 4.5.1.2 Struktur eines AFDX- Frames ohne Fragmentation

Die Struktur eines AFDX- Frames ist in der Abbildung 4.9 und in der Abbildung 4.10 zu sehen. Dargestellt ist die minimale als auch die maximale Framegröße.

Präambel	Start Frame Trennzeichen	Zieladresse	Quelladresse	Typ	IP Struktur	UDP Struktur	AFDX Nutzdaten	Padding	Sequenznummer	Frame Test Sequenz	Inter Frame Gap
7 Bytes	1 Byte	6 Bytes	6 Bytes	2 Bytes	20 Bytes	8 Bytes	1- 17 Bytes	0- 16 Bytes	1 Byte	4 Bytes	12 Bytes
64 Bytes											

Abbildung 4.9: Minimales AFDX Frame

Wie in der Abbildung 4.11 gut zu erkennen, kommuniziert End System 1 mit End System 2 und End System 3, über den Virtuellen Link 3. Partition 1 von End System 1 hat Zugriff auf den Virtuellen Link 1 und Partition 2 vom gleichen End System auf den Virtuellen Link 2 und den Virtuellen Link 3. Die Tabelle 4.2 zeigt die Adresstabellen für jedes End System.

Das Demultiplexing wird auf Empfängerseite über die Zieladressen von MAC, IP, und UDP realisiert.

Präambel	Start Frame Trennzeichen	Zieladresse	Quelladresse	Typ	IP Struktur	UDP Struktur	AFDX Nutzdaten	Sequenznummer	Frame Test Sequenz	Inter Frame Gap
7 Bytes	1 Byte	6 Bytes	6 Bytes	2 Bytes	20 Bytes	8 Bytes	1471 Bytes	1 Byte	4 Bytes	12 Bytes
1518 Bytes										

Abbildung 4.10: Maximales AFDX Frame

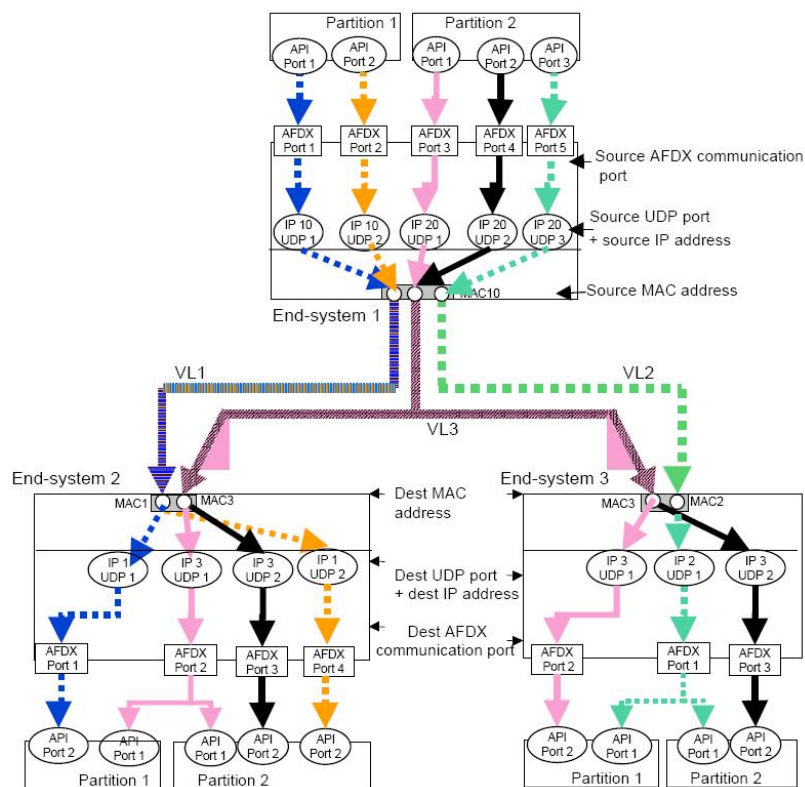


Abbildung 4.11: Adressierung im End System [ARINC \(2005\)](#)

Die Topologie für das beschriebene Beispiel ist in der Abbildung 4.12 dargestellt.

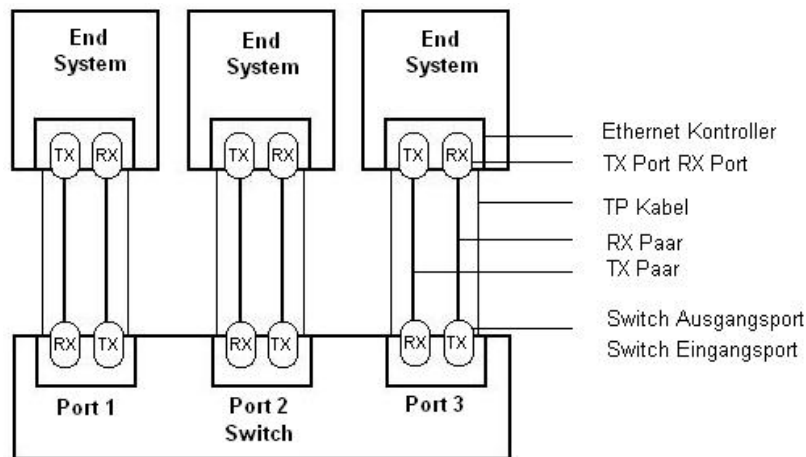


Abbildung 4.12: Physikalischer Aufbau

Zu sehen ist ein sternförmiges Netzwerk, alle End Systeme sind über einen Switch miteinander verbunden. Das Forwarding geschieht unter Verwendung der MAC- Adresse, die sowohl Unicast- oder Multicast Adressen sein kann.

Eingangsport	MAC Zielfeld der empfangenen Frames	Ausgabeport
1	MAC 1 (VL1)	2
1	MAC 2 (VL2)	3
1	MAC 3 (VL3)	2 und 3

Tabelle 4.2: Switch Forwarding

#### 4.5.1.3 Identifizierung der Ende - zu- Ende Kommunikation

Die Peer zu Peer Verbindungen werden durch die UDP-, IP- Quelladressen, Mac- und IP- Zieladressen und den UDP- Zielport identifiziert. Ausgewertet wird dabei jedes Frame welches den Empfänger erreicht. Dieses Verfahren garantiert im AFDX- Netzwerk die Identifikation jeder übertragenen Nachricht. Für jede IP Ziel- und Quelladresse können mehrere UDP/TCP- Ziel oder Quellports existieren.

- Intra AFDX- Kommunikation

Als Intra AFDX- Kommunikation wird Kommunikation innerhalb des AFDX- Netzwerkes bezeichnet. Die grundlegendste Eigenschaft der Intra AFDX- Kommunikation ist, das

für jede Nachricht die Adressierung fest vorgegeben ist. Zu berücksichtigen sind dabei 2 Arten, die Unidirektionale und die Bidirektionale Kommunikation.

Die Basis der Unidirektionale Kommunikation sind die AFDX- Kommunikationsports, die auf UDP- Ports basieren und entweder senden oder empfangen können. Sie stellen als Service den Queueing- oder Sampling Dienst bereit.

Bei der Bidirektionalen Kommunikation kann die Umsetzung durch 2 verschiedene Arten realisiert werden, wobei die dabei zur Anwendung gebrachten Protokolle oberhalb der UDP/TCP- Schicht zu finden sind, als Beispiel sei hier TFTP genannt.

Die erste Möglichkeit wäre, die über SAP- Ports die mit UDP/TCP- Ports verbunden sind zu nutzen. Dabei kann ein SAP- Port Sender als auch Empfänger von Nachrichten sein. Um die Bidirektionale Kommunikation zu erhalten, sind 2 SAP- Ports notwendig, einer der sendet und einer der empfängt. Die Kommunikation für ein- und ausgehende Verbindungen wird mit Quadrupeln realisiert. Die SAP- Ports müssen konform zu den verwendeten Internet Protokollen arbeiten. Das heißt für TFTP muss zum Beispiel Port 69 verwendet werden und kein anderer.

Die Zweite Möglichkeit ist die Verwendung der konventionellen AFDX- Kommunikationsports. Um die Bidirektionalität zu gewährleisten, ist es nötig im End System, 2 AFDX- Kommunikationsports zu definieren, einen Senderport und einen Empfängerport.

Bei der Bidirektionalen Verbindung ist es egal welche der beiden Möglichkeiten benutzt wird, darauf zu achten ist, dass die Empfängerports im Queueing Modus arbeiten.

- Extra AFDX- Kommunikation

Die Extra AFDX- Kommunikation beschreibt die Kommunikation zwischen dem AFDX- Netzwerk und einem IEEE 802.3 konformen Netzwerk. Auch zwischen diesen beiden Netzwerken werden beide Methoden der Kommunikation, die Uni- und Bidirektionale Kommunikation, verwendet. Die Adressierung erfolgt ebenfalls über statisch definierte Quadrupel.

Bei der Unidirektionalen Verbindung überträgt das End System Nachrichten an das IEEE 802.3 Netzwerk über die konventionellen AFDX- Kommunikationsports mittels einer UDP- Verbindung. Dazu werden in einer End System Konfigurationstabelle die Ziel Identifikation und der dazugehörige Port eingetragen.

Bei der Bidirektionalen Verbindung können verschiedene Protokolle die UDP/TCP verwenden benutzt werden. Die Kommunikation entspricht der Intra AFDX- Kommunikation für Bidirektionale Kommunikation.

#### 4.5.1.4 IP Adressierungsformat

Das hier beschriebene Format bezieht sich speziell auf das im AFDX- Netzwerk verwendete und wird als solches auch nur innerhalb dieses Netzwerkes verwendet. Für Frames die über die Grenzen des AFDX- Netzwerkes gesendet werden, muss ein Mechanismus die IP- Adresse für das IEEE konforme Netzwerk umsetzen.

- IP- Quelladressen

Die IP- Quelladresse dient zur Identifikation der sendenden Partition, des jeweiligen End Systems. Die IP- Adresse ist eine Klasse A Adresse, die sich aus den Feldern, wie in Abbildung 4.13 dargestellt, zusammensetzt.

IP Unicast Adressformat (Quell - oder Zieladresse) 32 Bit				
Klasse A 1 - Bit	Private IP Adresse 7- Bit	Benutzer definierte ID 16 - Bit	Partition ID 8 - Bit	
0	000 1010	nnnn nnnn nnnn nnnn	Spare Feld 000	n nnnn

Abbildung 4.13: IP Quelladresse

Das Feld User Defined ID ist 16 Bit groß und Identifiziert den Host und kann frei gewählt werden.

Die Partitions- ID unterteilt sich in 2 weitere Felder und dient zur Identifikation der Partition. Das Spare Feld wird erst dann verwendet wenn die Anzahl der Partitionen über 32 steigt. Ist das nicht der Fall wird es auf Null gesetzt.

- IP- Zieladresse

Die IP- Zieladresse wird vom End System verwendet, um IP- Pakete zu einem oder mehreren Ziel End Systemen zu senden. Sie kann dabei eine Unicast- als auch eine Multicast Adresse sein. Das Adressformat ist in Abbildung 4.14 dargestellt.

#### 4.5.1.5 AFDX Kommunikationsport, SAP und UDP/TCP Adressierungsformat

Für die Intra- und Extra Kommunikation existieren 2 Schnittstellen zwischen End System und Partition, die AFDX Kommunikationsports und die SAP Ports.

Die AFDX Kommunikationsport lassen sich durch folgende Eigenschaften beschreiben:

IP Adressierungsformat 32 Bit		
Klasse D 4 - Bit	IP Multicast ID 28 - Bit	
	Konstantes Feld 12 - Bit	Virtuelle Link ID 16 - Bit
1110	0000 1110 0000	nnnn nnnn nnnn nnnn

Abbildung 4.14: IP Zieladresse

- Unidirektional Zugriffs für das Senden und das Empfangen
- Sampling und Queueing
- während des Sendens und Empfangens existiert zwischen Sender und Empfänger nur eine Verbindung, die durch einen Quintupel definiert ist, dieser Quintupel setzt sich aus UDP- Quellport, IP Quellport, MAC Zieladresse, IP Zieladresse und UDP Ziel Port zusammen
- während des Sendens ist einem ADFX Kommunikationsport maximal eine Partition zugeordnet
- während des Empfangs können mehrere Partitionen auf einen AFDX Kommunikationsport zugreifen
- der Pfad für den Austausch von Nachrichten ist durch die Konfigurationstabelle vorgegeben

Der SAP Port zeichnet sich durch folgende Eigenschaften aus:

- ein SAP Port wird auf ein UDP Port abgebildet
- Unidirektional Zugriff, entweder senden oder empfangen
- die Möglichkeit 2 SAP Ports zu verbinden, für Bidirektionale Verbindungen
- beim senden wird eine feste Konfiguration eines Quadrupels verwendet

## 4.6 Nachrichtenstruktur

Nachrichten sind die Nutzdaten, die über das Netzwerk übertragen werden, betrachtet werden diese aus der Sicht der Anwendungsschicht des OSI- Referenzmodells.

### 4.6.1 Implizite- versus Explizite Nachrichten

Nachrichtenstrukturen lassen sich in Implizite- oder Explizite Nachrichten einteilen. Explizite Nachrichten beinhalten Informationen die es dem Empfänger erlauben die Nachricht zu dekodieren. Diese zusätzlichen Informationen sind Identifier, die angeben wie welche Struktur in ihr kodiert ist und werden als Overhead bezeichnet.

Implizite Nachrichten enthalten keinen Overhead, es werden nur Nutzdaten übertragen und Informationen wie diese zu interpretieren sind. Diese Art Nachricht nutzt die Bandbreite eines Netzwerkes effizienter.

Eine AFDX- Applikation erfährt das Nachrichtenformat, der Impliziten Nachrichten, über spezielle Portnummern auf der sie Nachrichten empfängt. Angewandt wird hierbei das Konzept des „Well Known Service“, welches bereits seit Jahren im Internet genutzt wird.

Typische AFDX- Implementationen benutzen zwei Arten von Portnummern:

- Die Nummern der Transportschicht (UDP - TCP)
- Die Nummern der ARINC 653 Kommunikationsschicht

Die UDP/TCP Portnummern, die im OSI- Referenzmodell der Transportschicht zugeordnet sind, werden von der IANA(Internet Assigned Numbers Authority) verwaltet und in drei Gruppen unterteilt:

- Zugeordnete Portnummern 0 - 1023
- Registrierte Portnummern 1024 - 49151
- Dynamische/Private Portnummern 49152 - 65535

Da es sich bei einem AFDX- Netzwerk um ein statisch konfiguriertes und nach außen geschlossenes Netzwerk handelt, sind die von der IANA unterteilten Gruppen nicht zwingend zu berücksichtigen und können somit von 1- 65535 frei gewählt werden. Wenn es sich bei den Diensten um bekannte Services handelt, sollten aus Gründen der Transparenz auch die entsprechenden Portnummer gewählt werden. Als Beispiel soll das TFTP dienen, welches auf Port 69 definiert ist.

Das AFDX- Netzwerk ist wie bereits erwähnt ein geschlossenes System. Es besteht die Möglichkeit über Gateways oder Layer 3 Router eine Verbindung mit anderen Netzwerken aufzubauen.

Wird ein Gateway verwendet erfolgt die Kommunikation über „Preassigned Portnumbers“. Diese verhindern den möglichen Adresskonflikt der einzelnen Netzwerke. Für den Fall das ein Layer 3 Router für die Interaktion zwischen den Netzwerken eingesetzt wird, ist darauf zu achten, das die „Assigned/Registered“ Portnummern nicht verwendet werden.

### 4.6.2 Primitive Datentypen

Jedes Datenelement ist ein vordefinierter, Netzwerk unabhängiger, Typ der auf einer Sprachunabhängigen IDL(Interface Description Language) basiert. Diese Datentypen haben eine definierte Bit und Byte Ordnung, die es Anwendungen ermöglicht Daten zu senden und zu empfangen unabhängig von der der Bit oder Byte Ordnung des Host-Prozessors.

Folgenden Datentypen sind für den Bereich der Luftfahrt im ADN(Aircraft Data Networks) definiert.

Daten Element Typ	Länge in Bit	Wertebereich
Signed_32 Integer	32	$(-2^{31} \dots 2^{31} - 1)$
Signed_64 Integer	64	$(-2^{63} \dots 2^{63} - 1)$
Float_32	32	$(+ - (2^{-1}/2^{22})2^{-126} \dots 127)$
Float_64	64	$(+ - (2^{-1}/2^{51})2^{-1022} \dots 1023)$
Boolean	32	nicht Anwendbar
String	-	nicht Anwendbar
Opak Daten	-	nicht Anwendbar

Tabelle 4.3: Primitive Datentypen

### 4.6.3 Datenanordnung

Die Datenanordnung gibt an, wie Daten im physikalischen Speicher abzulegen sind, das erhöht die Effizienz bei speichern und lesen, wodurch erheblich Zeit während der Programmausführung gespart werden kann.

Wenn die Datenelemente in einer Nachricht geordnet sind, ist es für die Anwendung einfacher die Anordnung der Daten im Speicher sicher zustellen.

Datenelemente werden angeordnet genannt, wenn die Adresse des Elementes im Speicher durch die Länge des Datenelementes in Bytes teilen lässt.

### 4.6.4 Spare - Padding

Spare Felder in einer Nachricht sind Felder für Daten, die eventuell für eine spätere Erweiterung reserviert werden. Mit dem Vorteil das nachträgliche Ergänzungen keinen Einfluss auf die Anwendung haben.



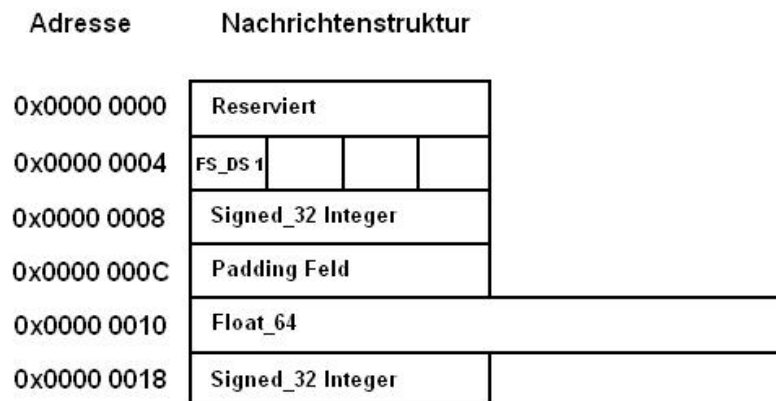


Abbildung 4.15: Nachrichtenstruktur

Padding wird verwendet um die Datenanordnung zu realisieren.

Beide können genutzt werden, um neue Parameter in eine Nachricht einzupflegen.

#### 4.6.5 Functional Data Set

Das FDS(Functional Data Set) gibt an, wie primitive Daten innerhalb einer Nachricht gruppiert sind, sie repräsentieren dabei Flugzeugdaten.

Alle Daten die nicht Protokoll basiert sind, werden in solchen FDS formatiert bevor sie gesendet werden.

Das FDS besteht aus zwei verschiedenen Feldern:

- Dem Functional Status Set (FSS)
- Dem Data Set(DS)

##### 4.6.5.1 Functional Status Set

Das FSS (Functional Status Set) ist ein 32 Bit großes Feld und enthält 4, 8Bit große Status Felder. Das erste gibt die Korrektheit und den Status des ersten DS(Data Set) wieder. Sollte keine weiteren DS im FDS vorgesehen sein, werden die übrigen 24Bit mit Nullen gefüllt. Für den Fall das weitere vorgesehen sind, ist das entsprechende FSS in das FDS mit aufzunehmen. Sollten alle vier Status Felder im FSS und die dazugehörigen DS genutzt werden,

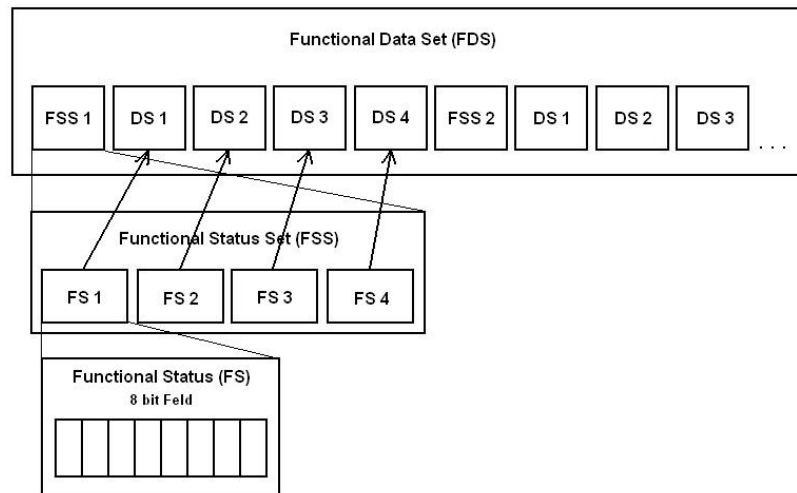


Abbildung 4.16: Functional Status Set

besteht die Möglichkeit weitere FSS in das FDS aufzunehmen. Die Anzahl der FDS ist einzig durch die maximale Payload Größe beschränkt.

Die Anzahl der primitiven Datentypen innerhalb des DS ist nicht begrenzt, das bedeutet, dass mehr als einer dieser Typen im DS untergebracht werden kann. Ein DS wird immer von einem FSS repräsentiert. Ist ein primitiver Datentyp des DS ungültig ist das gesamte DS als ungültig zu setzen. Aus diesem Grund ist es von Vorteil die Daten eines Sensor oder Aktors in einem DS zusammen zufassen, sollte ein Teil des DS ungültig sein ist die Wahrscheinlichkeit gegeben das auch der andere Teil ungültig ist.

Luffahrtsysteme müssen den Status der empfangenen Daten kennen, mit Hilfe der Functional Status-Bytes ist diese Identifikation für jedes FDS gewährleistet.

Das FSS ist eine wie folgt definierte Byte Enumeration.

Diese Statusbytes sind priorisiert, für den Fall das mehrere dieser Bedingungen eintreffen. Die Priorisierung erfolgt immer von der kleinsten (1) zur höchsten (4) wie in Tabelle.. dargestellt.

#### 4.6.5.2 Data Set

Wie bereits erwähnt können DS mehr als einen primitiven Datentypen enthalten. Es ist auch möglich Spare Felder am Ende dieser DS einzufügen um das ergänzen von späteren Entwicklungen oder Produkterweiterungen zu vereinfachen. Diese Spare Felder sollten

mit Rücksicht auf die Netzwerklast gut gewählt sein. Daten innerhalb der DS, sind hintereinander angeordnet und das ohne Bytes zwischen ihnen frei zu lassen. Das Padding der Daten Anordnung ist davon nicht betroffen. Gefüllt werden die Spare Felder mit Nullen.

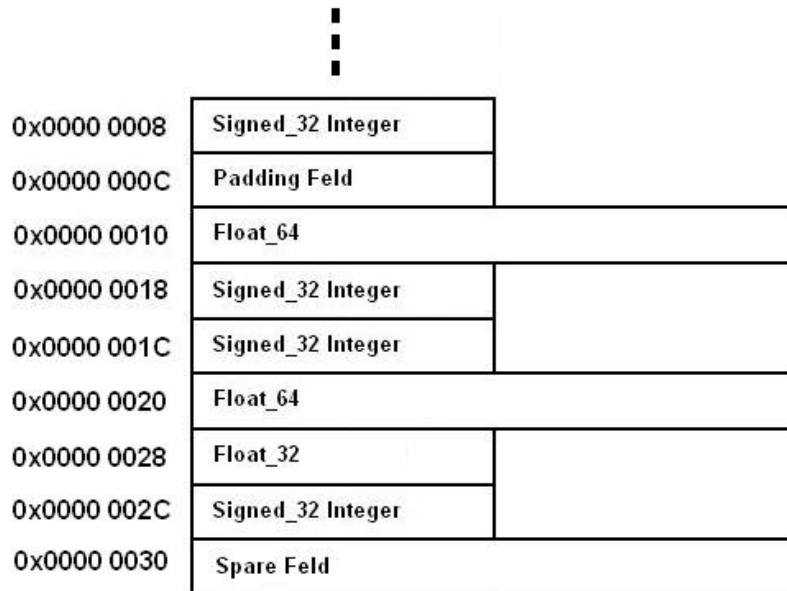


Abbildung 4.17: Data Set

#### 4.6.6 Globale Nachrichten Struktur

Die „Global Aircraft Data Network Message“ setzt sich aus dem zuvor beschriebenen Feldern zusammen. Zusätzlich wird am Ende der Nachricht ein „Global Spare“ Feld angehängt dieses ist aber nur optional. Das Feld sollte groß genug sein um ein FDS aufnehmen zu können um spätere Entwicklungen leichter zu implementieren. Dieses Feld ist mit Nullen zu füllen.

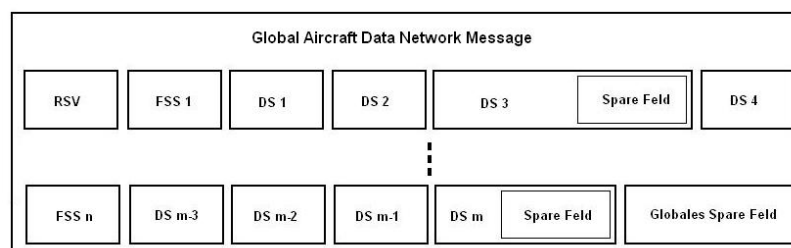


Abbildung 4.18: Global Data Set

# 5 Design

## 5.1 Hardwaredesign

Die hier beschriebenen Zusammenhänge basieren auf der zum Evaluationboard mitgelieferten Dokumentation. [EUROS \(2005a\)](#) [DualCore \(2006\)](#)

### 5.1.1 Dual Core Industrial Controller DCIC 9907

Der Controller basiert auf der ARM 9 Architektur und kann für die Software Entwicklung zusammen mit einem Evaluationboard bestellt werden. In diesem Kapitel wird auf die Eigenschaften der Hardware eingegangen. Einen ersten Überblick soll die folgende Grafik geben die das Evaluationboard zeigt. Das dargestellte Board entspricht dem zur Entwicklung verwendeten.

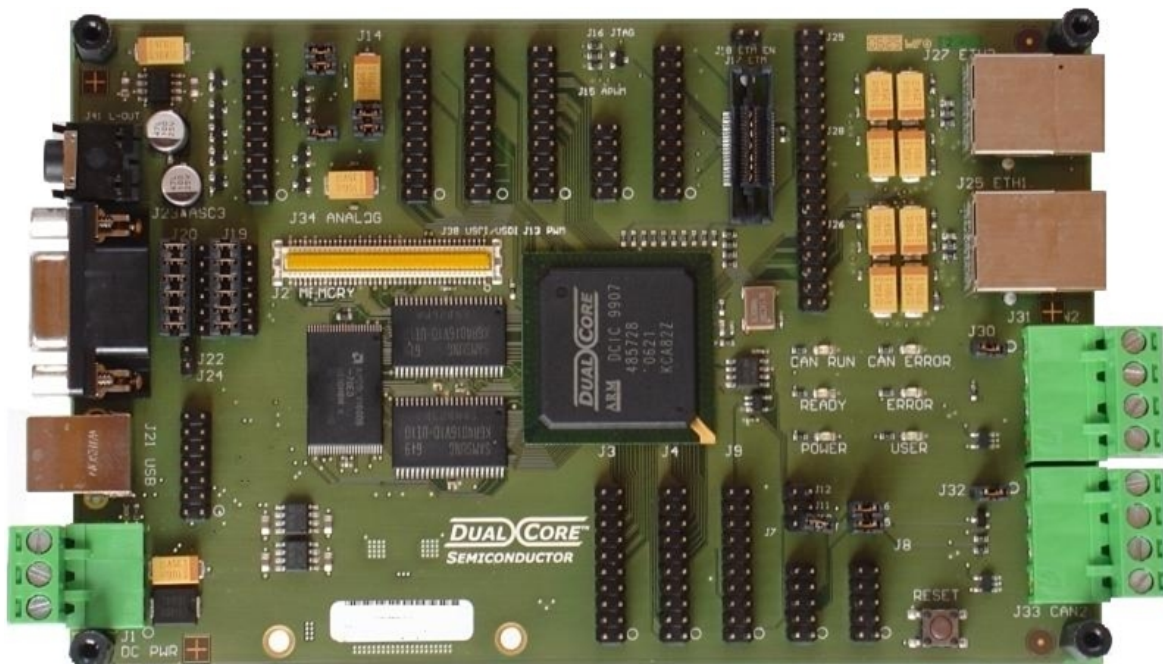


Abbildung 5.1: Evaluationboard

### 5.1.1.1 Systemarchitektur

Das DCIC 9907 Evaluationboard ist ein System- on- Chip Design System. Als Prozessor wurde ein Dualcore ARM 946E-S verwendet. Ein Kern fungiert dabei als Host- der andere als Control- Prozessor. Die beiden Kerne unterscheiden sich in der Größe des "Tightly Coupled Memorys". Der Host- Prozessor unterstützt zusätzlich die ARM "Realtime Trace Technology" mit einem zusätzlichen 8 Bit Port "Embedded Trace Macrocell".

Der ARM946E-S ist eine synthetisierte Macrozelle die einen breiten Bereich der eingebetteten Systeme abdeckt.

Der Control- Prozessor führt den gesamten Programmcode aus, die schnellsten "Control loops" beginnen dabei im „Tightly Coupled Memory“.

Der Host-Prozessor steuert die verschiedenen Kommunikationskanäle unterstützt wird dies durch das Realtime Betriebssystem der Firma Euro Embedded Systems.

Beide Kerne enthalten den ARM vSTE Instruktionssatz und unterstützen eine erweiterte 16x32 Bit Multiplikatorfähige "Bingle cycle" MAC Operationen.

Die Architektur des Dualcore ARM Prozessor basiert auf der AHB- Interconnection Matrix, diese ermöglicht den parallelen Zugriff zwischen mehreren AHB Master und Slave Modulen im System.

Im DCIC9907 ist eine 6 Master / 4 Slave Interconnection Matrix implementiert. Beide ARM-Kerne, die Ethernetchnittstellen oder die DMA- Master haben Zugang zu einem der 4 Slave Kanäle.

Für hohen Datendurchsatz sind zwei SDRAM Blöcke integriert und ein 96 Kilo Byte großer Wait State Speicher der den Datenfluss des Host-Prozessors unterstützt. Zum schnellen Datenaustausch zwischen Host- und Control- Prozessor steht ein 1 Kilobyte großer Speicherblock zur Verfügung. Das hat den Vorteil, dass der Control- Prozessor von möglichen Schreibzyklen des Host -Prozessors entkoppelt ist.

Die Kommunikationsschnittstelle unterstützt eine Vielzahl von IP- Kommunikationsblöcken, dazu wurden die Ethernetkontroller mit Hardwareerweiterungen versehen. Diese unterstützen die ARM - Prozessor - Algorithmen und Supervisor- Funktionen die es dem System erlauben in Echtzeit zu arbeiten.

### **5.1.2 Physikalische Eigenschaften**

Die interne Spannungsversorgung arbeitet mit 1,8 VDC die externe mit 3,3 VDC und alle Input/Outputpins sind TTL kompatibel. Der Temperaturbereich ist von -40°C bis +85°C zertifiziert.

Die Bauform des DCIC9907 ist BGA(Ball Grid Array) mit 304 Pins und einem Ball Abstand von 1,27 mm.

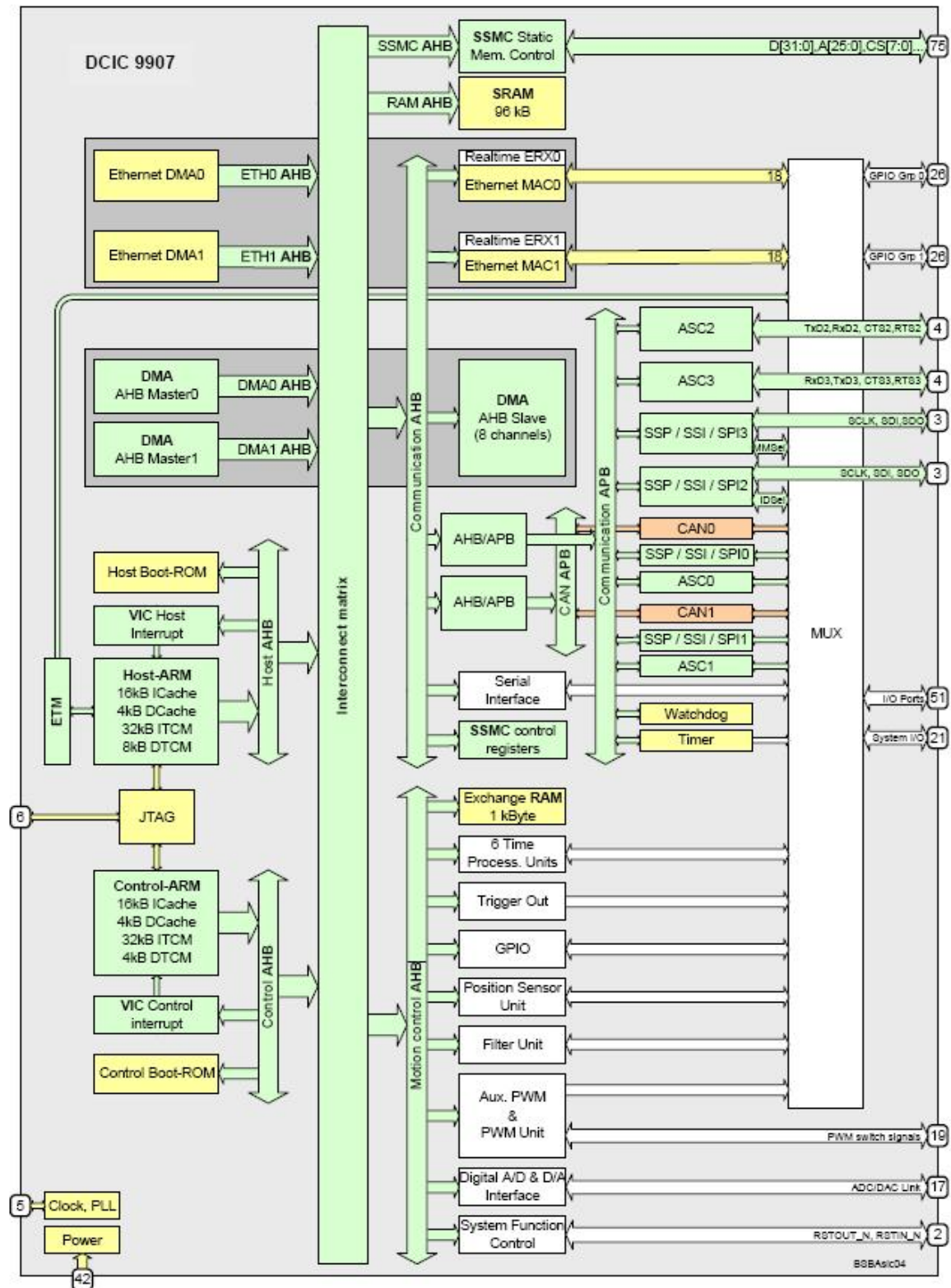


Abbildung 5.2: DCIC 9907 Blockdiagramm DualCore (2006)

### 5.1.2.1 Host CPU

Der Host- Prozess ist für die Ausführung der „main()“ Applikations- Task des Echtzeit- Betriebssystems optimiert. Dabei wurde der Speicher, die Debug- Möglichkeiten und die Boot- Sequenz der beiden CPU- Kerne des DCIC 9907 verbessert.

Der Host- Kern enthält 16kbyte Instruktions- Cache und 4kbyte Daten- Cache außer diesen sind 32kbyte „Tightly- Coupled- Instruction- Memory“ (ITCM) und 8 kbyte „Tightly- Coupled- Data- Memory“ (DTCM) vorhanden.

Nach einem Reset bietet der Host- ARM die Möglichkeit von externen- oder von On- Chip Speicher zu starten.

Der Host- ARM arbeitet mit einer Taktfrequenz von 128MHz.

### 5.1.2.2 Control CPU

Der Control- Prozess wurde für die Ausführung von sich oft wiederholenden Schleifen optimiert. Das Ziel ist die Ausführung unabhängigen synchronisierten Task bei unterschiedlichen Frequenzen. Der gesamte Code der liegt dabei komplett im „Tightly- Coupled- Memory“ des Control- ARM.

Der Control- Kern benötigt nicht Zwingender weise Instruktions- oder Daten- Cache er ist in derselben Größe wie beim Host- ARM vorhanden außerdem enthält er 32kbyte ITCM und 4kbyte DTCM. Der ITCM sollte groß genug sein um die komplette Kontroll- Schleife des Programms enthalten, was in der Selben Performanzsteigerung resultiert wie der Einsatz von Instruktions- Cache.

Der Reset des Control- ARMs wird über den Host- ARM gesteuert, über ein spezielles System- Control- Register. Der Control- ARM hat nur die Möglichkeit dieses Register wieder zurückzusetzen.

Der Control- ARM arbeitet mit einer Taktfrequenz von 128MHz.

### 5.1.2.3 Ethernet Kommunikationsschnittstelle

Das Evaluationboard stellt neben dem Ethernet Standard nach IEEE 802.3 zusätzlich zwei Echtzeit Ethernet Versionen zur Verfügung, das Powerlink und Precision Time Protocol(PTP).

Unterstützt wird der TCP/IP Standard, die Datenrate von 10/100 MBit/s über zwei getrennte Ethernet MAC. Erweitert sind diese um jeweils eine Ethernet Realtime Extension(ERX).



Jedes Ethernet Modul(ENET) ist direkt mit der Interconnection Matrix verbunden und agiert auf dem Advanced High- Performance Bus(AHB) als Direct Memory Access(DMA) Master. Das Media Independent Interface(MII) beider Ethernet Module ist mit dem MUX verbunden. Es besteht die Möglichkeit beide ENET über spezial Register mit dem High- Speed- Interface zu verbinden.

#### 5.1.2.4 Ausstattung

Die Ethernetschnittstelle ist mit folgenden Komponenten ausgestattet:

- Media Independent Interface(MII)
- Reduced Media Independent Interface(RMII)
- Advanced High-performance Bus (AHB) Schnittstelle
- IEEE 802.3 CSMA/CD Technologie
- Halb/Voll Duplex Übertragung
- 10/100 MBit/s Datenraten
- MAC-Tx FIFO mit einer Länge von 2048 x 8 Bit
- MAC-Rx FIFO mit einer Länge von 512 x 35 Bit
- unterstützt short/long-packet-mode
- Cycling Redundance Check (CRC) generieren und testen

## 5.2 Echtzeit Betriebssystem Euros

EUROS ist ein Echtzeit-Betriebssystem das eine Vielzahl von CPU- Architekturen unterstützt. Es eignet sich für leistungsschwächere als auch für High- End- Baugruppen. Völlig unabhängig ist auch über welche Schnittstellen das System verfügt, da es durch den Modularen Aufbau des Betriebssystems, diese individuell mit eingebunden werden können.

### 5.2.1 Hardwareunabhängigkeit

Der Einsatz von EUROS ist Hardwareunabhängig, die Applikation wird in C/C++ entwickelt und greift auf EUROS- Schnittstellen zu, da diese Schnittstellen für alle Architekturen gleich aufgebaut sind.

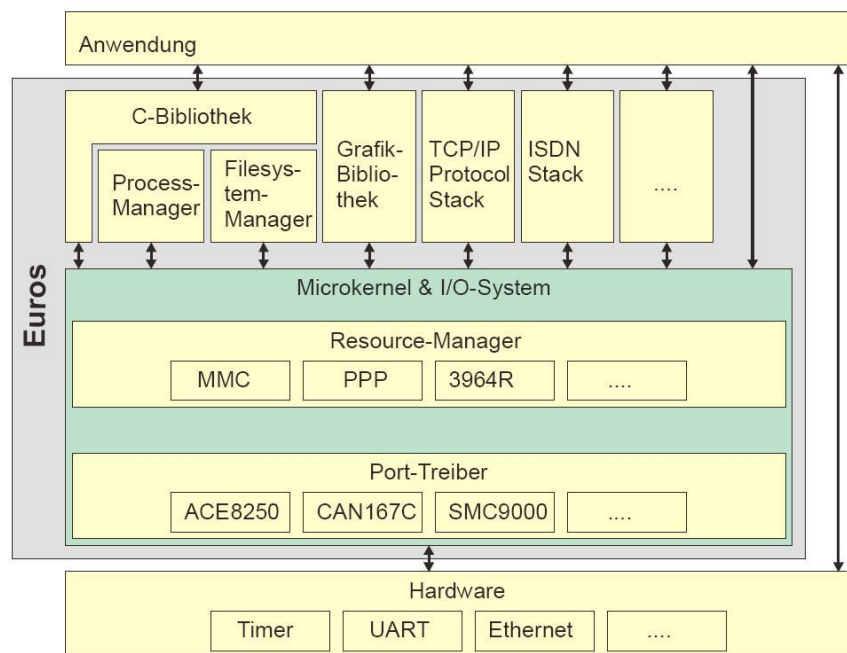


Abbildung 5.3: Betriebssystem Komponenten [EUROS \(2006\)](#)

## 5.2.2 Multitaskingfähigkeit

Das Betriebssystem ist Multitaskingfähig, mehrere Aufgaben laufen parallel ab. In dem zu entwickelten System ist das tatsächlich der Fall, da zwei CPU- Kerne parallel arbeiten.

Für Tasks die zur selben Zeit auf einem der beiden CPU- Kerne ausgeführt wird gilt, dass nur einer zurzeit die CPU- Zeit für sich beanspruchen kann. Um mehrere Tasks gleichzeitig auszuführen erhält jede Task vom Betriebssystem- Scheduler Rechenzeit. Dieses basiert auf einem prioritätsgeteuerten Round Robin Verfahren.

## 5.2.3 Euros Objekte

Die Verwaltung der Betriebsmittel wird bei EUROS über Objekte die zur Laufzeit erzeugt oder gelöscht werden geregelt. Die Objekte sind von einem bestimmten Typ, die die Funktionsweise dieser festlegen. Objekte gleichen Typs werden bei EUROS hierarchisch angeordnet. Die Wurzel dieser Baumstruktur stellt dabei das Root- Objekt dar welches beim Start des Systems erzeugt wird. Für jeden Typ von Objekten steht ein spezieller Satz von Diensten zur Verfügung.

## 5.2.4 Clusterkonzept

Aufgrund der hierarchischen Anordnung der Objekte ist die logische Strukturierung von zusammen gehörigen Funktionen möglich. Dies ermöglicht einzelne Funktionsaufrufe auf einen kompletten Teil- Baum, ein Cluster anzuwenden. Das Clusterkonzept bringt folgende Vorteile mit sich:

- Kompakter Code Mit einem Funktionsaufruf können mehrere Objekte aufgerufen werden
- Kapselung von untergeordneten Objekten möglich Task muss nicht jedes untergeordnete Objekt kennen um es anzusprechen
- Warten an mehreren Objekten gleichzeitig möglich

## 5.2.5 Systemzustände

Für Echtzeitsysteme ist die Einhaltung einer definierten Reaktionszeit auf ein externes Ereignis garantiert, diese liegen üblicher weise im Nanosekundenbereich. EUROS arbeitet um diese Anforderungen garantieren zu können mit den vier folgenden Systemzuständen.

- Interrupt Zustand
- Nested Zustand
- System Zustand
- Anwendungs- Zustand

## 5.2.6 DCIC 9907 Ethernet Port Treiber

Die Informationen zu diesem Kapitel basieren der Dokumentation von EUROS

### 5.2.6.1 Technische Eigenschaften

Die DCIC 9907 Ethernet Port Treiber unterstützen Bitrate von 10 MBit/s oder 100 MBit/s und die Schnittstellen arbeiten je nach Bitrate nach 10Base-T oder 100Base-TX Standard. Die physikalische Schnittstelle, der PHY arbeite mit Media Independent Interface (MII) oder Reduced Media Independent Interface (RMII). Die besondere Eigenschaft des DCIC 9907 Ethernet Port Treiber ist, dass er zwei Kanäle unterstützt.

### 5.2.6.2 Konfigurationsmöglichkeiten

Die DCIC 9907 Ethernet Port Treiber unterstützen Bitrate von 10 MBit/s oder 100 MBit/s und die Schnittstellen arbeiten je nach Bitrate nach 10Base-T oder 100Base-TX Standard. Die physikalische Schnittstelle, der PHY arbeite mit Media Independent Interface (MII) oder Reduced Media Independent Interface (RMII). Die besondere Eigenschaft des DCIC 9907 Ethernet Port Treiber ist, dass er zwei Kanäle unterstützt.

Die Erzeugung eines Kanals erfolgt über eine Datenstruktur vom Typ `tloConfig` die von der Anwendung bereitgestellt wird. Eine entsprechend Variable von dem Typ `tloConfig` ist bereits der Treiberbibliothek vorhanden. Die Struktur besteht aus vier der folgenden Komponenten:

- ChannelNumber
- MacAddress
- PhyAddress
- Flags

Die Channel Number enthält die Nummer des Kanals auf dem gesendet und empfangen wird. Die MacAddress ist ein 6 Byte großes Array in der die MAC- Adresse in Netzwerk- Byte-Ordnung für den Kanal hinterlegt ist. In der PhyAddress ist die Adresse der physikalischen Schnittstelle gespeichert. Der Inhalt der Komponente Flags legt fest wie die Steuerung des Kanals und des Treibers zu vollziehen ist. Dabei können folgende Flags gesetzt werden:

- DCIC9907\_ETH\_MII
- DCIC9907\_ETH\_RMII
- DCIC9907\_ETH\_10
- DCIC9907\_ETH\_100
- DCIC9907\_ETH\_DUPLEX
- DCIC9907\_ETH\_AUTONEG

Die Flags DCIC9907\_ETH\_MII und DCIC9907\_ETH\_RMII legen fest welche physikalische Schnittstellen benutzt wird.

Die Geschwindigkeit wird mit den Flags DCIC9907\_ETH\_10, DCIC9907\_ETH\_100 eingestellt dieser Wert lässt sich dann auch nicht mehr ändern.

Ob die Ethernet-Schnittstelle im Halb- oder Vollduplex- Modus arbeitet wird über das Flag `DCIC9907_ETH_DUPLEX` entschieden, sollte es nicht gesetzt sein ist automatisch der Halbduplex- Modus gesetzt.

Sollte die Autonegotiation zur automatischen aushandeln der Geschwindigkeit mit dem Flag `DCIC9907_ETH_AUTONEG` eingestellt sein werden die Flags `DCIC9907_ETH_10`, `DCIC9907_ETH_100` und `DCIC9907_ETH_DUPLEX` ignoriert.

In der Treiberbibliothek ist bereits eine Variable `Dcic9907EthConfig` vom Typ `tloConfig` enthalten, die zur Erzeugung von Kanälen geeignet ist.

## 6 Realisierung

Nachdem die Hardware für das System ausgewählt war, galt es eine Software zu entwickeln mit der eine Aussage über die Machbarkeit zu treffen ist.

Die Software muss aus diesem Grund die wichtigsten Eigenschaften des AFDX- Standards aufgreifen, um zu einer fundierten Aussage zu gelangen.

An dieser Stelle ist darauf hinzuweisen, dass nicht alle wichtigen Eigenschaften des AFDX- Standards umgesetzt werden konnten. Um welche speziellen Eigenschaften es sich dabei handelt wird im weiteren Verlauf des Kapitels erklärt.

Zunächst werden die Eigenschaften des Systems diskutiert, die als wichtig erachtet sind und in der Software umgesetzt werden.

Die Kommunikation der einzelnen Komponenten im AFDX- Netzwerk basiert auf der Verwendung von Virtuellen Links. Unter Verwendung der ihnen zugewiesenen Eigenschaften ist es möglich, den Determinismus des AFDX- Netzwerkes zu garantieren.

Der als Zielsystem ausgewählte Sensor, der aus Gründen der Geheimhaltung nicht weiter beschrieben werden darf, muss über 5 Virtuelle Links verfügen können. Für den Empfang von Nachrichten sind dabei drei VLs eingehende Nachrichten vorgesehen und zwei VLs für ausgehende Nachrichten. Jedem dieser VLs wird eine MAC- Adresse zugewiesen, über den er eindeutig identifiziert werden kann. Die von Euros verwendeten Ethernet- Treiber verfügen nicht über die Möglichkeit eine Ethernet- Schnittstelle mit mehr als einer MAC- Adresse zu versehen.

Aus diesem Grund wurde der Quellcode der Ethernet- Treiber von der Firma Euros Embedded Systems GmbH zur Verfügung gestellt, um die gewünschten Anpassungen vornehmen zu können. Diese wurden nicht vorgenommen. Die Anpassung von Treibern stellt ein tiefen Eingriff in das System dar. Geringe Veränderungen können zu Seiteneffekten führen, die ohne nähere Untersuchung ein hohes Maß an Fehleranfälligkeit mit sich bringen. Um eine einwandfrei Funktion der angepassten Treiber garantieren zu können bedarf es außerdem einer erneuten Verifizierung. Aufgrund dieser Gegebenheiten wurde davon Abstand genommen.

Um die Ethernet- Treiber testen zu können muss der Netzwerkmanager des TCP/IP- Stack die Verwaltung mehrerer MAC- Adressen pro Ethernet- Adapter handhaben können. Das

kann der Netzwerkmanager nicht leisten und dem entsprechend müssen zuerst hier die Voraussetzungen geschaffen werden, um danach die Ethernet- Treiber anzupassen.

Um das Problem der MAC- Adressen zu umgehen und dennoch eine Aussage über die Verwendung der vorliegenden Hardware treffen zu können, wurde das Programm so konzipiert das verschiedene Setups geladen werden können. Die Setups spiegeln dabei verschiedene Szenarien wieder, die im Kapitel Performanz 7 ausführlich besprochen und getestet werden.

Das End System hat die Aufgabe, Daten zu versenden oder zu empfangen. Das Übertragen von Daten erfolgt AFDX- Konform. Das bedeutet, die zu übertragen Frames 4.5.1.2 entsprechen denen des AFDX- Standards. Für das Empfangen von Daten müssen die Frames auf Integrität und Redundanz geprüft werden, das kann nur bedingt umgesetzt werden. Denn ohne die Möglichkeit mehrere MAC- Adressen an einen Ethernet PHY zu binden ist eine Umsetzung des Redundanzmanagements nicht sinnvoll.

Die Nachrichtenstruktur der Frames entspricht genau der in Kapitel Nachrichtenstruktur 4.6 beschrieben und konnte exakt nachgebildet werden.

Das System entspricht damit den folgenden Gegebenheiten:

- Anzahl der gleichzeitig genutzten PHYs 2
  - 2 PHYs senden über jeweils eine Netzwerk-Adresse
  - 2 PHYs empfangen über jeweils eine Netzwerk-Adresse
  - 1 PHYs senden über eine Netzwerk-Adresse und 1 PHYs empfängt über eine Netzwerk-Adresse
- Anzahl der TASKs
  - Root- Task
  - Idle- Task
  - Anzahl der Virtuellen Links
    - \* 3 VL- Task empfangen Nachrichten im AFDX- Nachrichtenformat
    - \* 2 VL- Task senden Nachrichten im AFDX- Nachrichtenformat

## 6.1 Programm

Das entwickelte Programm wurde, wie unter Kapitel Realisierung 6 beschrieben umgesetzt. Der Quellcode des Programms kann nicht veröffentlicht werden da er Eigentum der Firma Monogram Systems GmbH ist. Um einen Überblick der Umsetzung geben zu können, wird das Design des Programms in seinen Grundzügen beschrieben.

Für die abgedruckten Quellcodeauszüge hat die Monogram Systems GmbH ihre Genehmigung erteilt.

### 6.1.1 Root- Task

Die Root Task hat die Aufgabe das Test Programm zu starten. In ihr werden alle nötigen Initialisierungen für das Programm vorgenommen.

Zuerst werden die Treiber für die Ethernet-Ports gestartet und die Task zum Senden und Empfangen der Nachrichten erzeugt.

Danach wird das Netzwerkinterface an die Netzwerkkomponente gebunden und die Eigenschaften, die das Netzwerkinterface beschreiben gesetzt, und mit diesen gestartet.

Nach erfolgreichem Start wird die Task für das Senden und Empfangen gestartet.

### 6.1.2 Idle- Task

Die Idle- Task wird vom Betriebssystem erzeugt und gestartet und soll hier nur der Vollständigkeit halber erwähnt werden.

### 6.1.3 VL- Task Sender

Die VL- Task Sender in dem die ausgehenden Nachrichten verwaltet werden.

Hier wird die Funktion `start_AFDX_client_A()` aufgerufen, in der alle weitere Operationen die für das Senden von Nachrichten notwendig sind, abgearbeitet.

Für jede dieser Task wird eine eigene Funktion angestoßen um die speziellen Eigenschaften jedes Virtuellen Link zu setzen.

Insgesamt werden zwei dieser Task gestartet, das entspricht der Anzahl vom Sensor ausgehenden AFDX- Nachrichten, die einen separaten Virtuellen Link benötigen.



```
void VL_out_X (int Arg, void* pArg)
{

/*Starten des Clients Funktion aus AFDX_Client.c*/
start_AFDX_client_X();
TaskSleep(1000|MS);

TaskTerminate();
}
```

Das X steht für die Virtuelle Link Nummer die gleichzeitig die Anzahl der sendenden Task wieder spiegelt.

#### 6.1.4 VL- Task Empfänger

Die VL- Task Empfänger in dem die eingehenden Nachrichten verwaltet werden.

Hier wird die Funktion start\_AFDX\_serv() aufgerufen, in der alle weitere Operationen die für das Empfangen von Nachrichten notwendig sind, abgearbeitet.

Für diese Task wird eine eigene Funktion angestoßen, um die speziellen Eigenschaften jedes Virtuellen Link zu simulieren.

Insgesamt werden drei dieser Task gestartet, das entspricht der Anzahl vom Sensor eingehenden AFDX- Nachrichten die über diese Virtuellen Links empfangen werden.

```
void VL_in_X (int Arg, void* pArg)
{

/*Starten des Servers Funktion aus AFDX_Server.c*/
start_AFDX_serv();
TaskSleep(1000|MS);

Panic("Connection closed VL_2");
//soclose(ServSock);

TaskTerminate();
}
```

Das X steht für die Virtuelle Link Nummer die gleichzeitig die Anzahl der empfangenden Task wieder spiegelt.

### 6.1.5 Funktion des VL- Task Sender

Die Funktionalität des sendenden Virtuellen Links wird in einer Routine realisiert.

Es werden die nötigen Speicher reserviert, um die ausgehenden Nachrichten zu Speichern. Die Funktionen des TCP/IP Stack zum anlegen der Socket, an die entsprechend Netzwerkschnittstellen, werden aufgerufen.

Die Nachrichten können in verschiedenen Geschwindigkeiten übertragen werden. Die Geschwindigkeit richtet sich nach der Anzahl der Virtuellen Links. Da auch hier das Fehlen der zweiten MAC- Adressen zum tragen kommt, kann der zweite Virtuelle Link auf die gleiche MAC- Adresse angewandt werden.

### 6.1.6 Funktion des VL- Task Empfänger

Die Funktionalität des empfangenden Virtuellen Links wird in einer Routine realisiert.

Es werden die nötigen Speicher reserviert, um die eingehenden Nachrichten zu Speichern. Die Funktionen des TCP/IP Stack zum anlegen und binden des Socket, an die entsprechende Netzwerkschnittstellen, werden aufgerufen.

Es wird ständig auf das Eintreffen neuer Nachrichten gewartet.

Die Nachrichten werden einem Integritätscheck unterzogen und die Nachrichtenstruktur wird auf Richtigkeit geprüft.

Nach abgeschlossener Integritätsprüfung sollte der Redundanzmanager die eintreffenden Pakete untersuchen und Duplikate entfernen bevor sie an die Applikationsschicht weitergeleitet werden. Das kann aufgrund der fehlenden redundant arbeitenden zweiten Ethernet-Schnittstelle nicht realisiert werden und muss bei erfolgreicher Anpassung des Netzwerkmanagers und des Treibers ergänzt werden.

Das Fehlen der zusätzlichen zwei weiteren VLs wird umgangen, in dem alle drei empfangenden VLs die gleiche MAC- Adresse nutzen. Dadurch kann zumindest eine annähernd gleiche Prozessorlast angenommen werden.

# 7 Performanz

Das Ziel ist es, das AFDX- Netzwerk in einem deterministischen Rahmen zu nutzen. Dieser Determinismus lässt sich durch die Aufnahme von Messungen am End System nachweisen. Der Vorteil dieser Methode ist die Vereinfachung der Zertifizierung des End Systems.

Der Nachweis über die Performanz des End Systems, wird durch zwei voneinander getrennte Messungen erbracht. Die Trennung wird vorgenommen, um Empfangsverzögerung und Sendeverzögerung unabhängig voneinander zu bewerten.

## 7.1 Bandbreite

Unter der Bandbreite eines Übertragungssystems wird der Bereich der Frequenzen verstanden, der ohne wesentliche Beeinflussung übertragen werden kann. Denn mit steigender Frequenz steigt auch die Dämpfung die das Datensignal in seiner Höhe beeinflusst. Ideal ist es, wenn die Bandbreite des Übertragungsmediums größer ist als die Bandbreite des Datensignals. ([Rech \(2002\)](#))

## 7.2 Datenübertragungsrate

Die Datenrate oder Datenübertragungsrate gibt an, wie viel Bits während einer Zeiteinheit übertragen werden können.

## 7.3 Übertragungsfrequenz/Datenübertragungsrate

Da die Begrifflichkeiten in der Praxis häufig vertauscht oder synonym verwendet werden, wird der Unterschied nochmals hervorgehoben.

Grundsätzlich ist die Datenübertragungsrate nicht mit der Übertragungsfrequenz gleich zu setzen. Die Datenübertragungsrate gibt die Anzahl der Bits an, die innerhalb eines Zeitintervalls übertragen werden können. Die Übertragungsfrequenz ist davon abhängig welches Kodierungsverfahren bei der Datenübertragung angewendet werden. Das Kodierungsverfahren beschreibt wie Informationen dargestellt werden und sorgt dafür dass die auftretende Frequenz bei der Übertragung möglichst klein gering ausfällt. Denn die Übertragungsfrequenz bestimmt die Dämpfung des zu übertragenden Signals und somit dessen Reichweite.

## 7.4 Latenz

Die zur Ermittlung benötigten Werte bei der Verzögerung, werden zwischen den Kommunikations- Diensten und der Physikalischen Schicht eines End Systems bestimmt.

Die Messung für die Sendeverzögerung beginnt, wenn das letzte Bit einer Nachricht von der Host Partition an den Kommunikationsdienst übertragen wurde, und endet sobald das letzte Bit des Ethernetframes von der Physikalische Schicht übermittelt wurde.

Die Empfangsverzögerung wird bestimmt, wenn das letzte Bit des Ethernetframes von der Physikalische Schicht an den MAC Layer übergeben wird und endet sobald das letzte Bit der Daten der Host Partition zur Verfügung steht.

Die Rahmenbedingung für diese Messungen setzen leere Buffer und keine IP- Fragmente voraus. Des Weiteren müssen Konflikte bezüglich des Netzwerkzugriffes, die Messergebnisse verfälschen, vermieden werden.

Die Gesamtverzögerung eines End Systems setzt sich aus der technologischen Verzögerung und der Konfigurations- Verzögerung zusammen.

### 7.4.1 Technologische Latenz

Die Technologische Verzögerung definiert sich als Zeit die von der Aufnahme der Übertragung bis zum Abschluss vergeht.

#### 7.4.1.1 Übermittlung

Die Verzögerungszeit die bei der Übertragung im End System auftritt, wird durch zwei Messung bestimmt. Die erste Messung wird vorgenommen, wenn das letzte Bit der sendenden Partition für den Kommunikationsservice bereit steht. Die zweite Messung wenn das letzte Bit dieses Paketes vom physikalischen Medium übertragen wurde.

Die Verzögerungszeit innerhalb des End Systems ist auf maximal  $150\mu s$  + Paket Verzögerung begrenzt.

## 7.5 Messaufbau

Um die AFDX- Netzwerkumgebung nachzustellen wurde ein Netzwerk, das aus dem entwickelten End System, einem Switch und einem Rechner auf dem Wireshark installiert ist, aufgebaut. Der genaue Aufbau ist grafisch unter dem Abschnitt Performanz- Tests [7.6](#) dargestellt.

Für alle Untersuchungen wurden die folgenden Hardwarekomponenten verwendet:

- Evaluationboard DCIC9907 der Firma Dualcore Semiconductor
- Netzwerk Switch NETGEAR FS105 IS 5-PORT SWITCH 10/100 Mbit
- Notebook mit 10/100 Mbit Netzwerkkarte
- Standard TP-Kabel der Kategorie 5

## 7.6 Performanz- Tests

Der Test 1 und Test 2 soll Daten liefern die eine Aussage über die zu erreichende Datenrate zulassen und zeigen in welchem Maß Paketverluste auftreten. Die Funktionen die der AFDX- Standard vorgibt werden bis auf einige Ausnahme vernachlässigt. Es werden nur die zur Übertragung benötigten Prozesse gestartet, auf alle anderen wurde verzichtet. Dem zu Grunde liegt das zu testende Verhalten der Ethernet- Schnittstelle. Um diese Ergebnisse nicht zu verfälschen wurde auf zusätzliche Prozesse oder Routinen verzichtet. Die Konfiguration für das End System ist unter den einzelnen Kapiteln für die Test zu entnehmen.

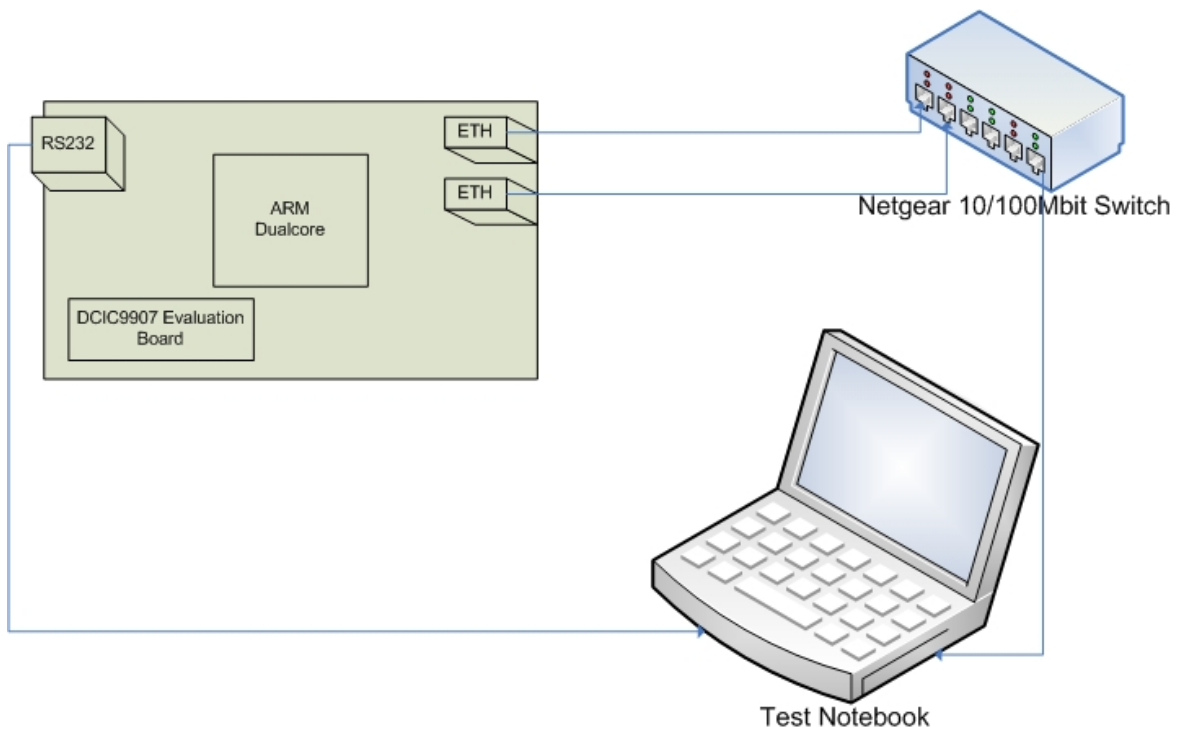


Abbildung 7.1: Testaufbau Datenrate

Der Test 3 und Test 4 sind als Dauertests zu verstehen. Sie dienen zur Überprüfung der Stabilität des End Systems. Die Nachrichtenstruktur als auch auftretende Paketverluste werden einer Prüfung unterzogen. Zu bemerken ist, dass das Evaluationboard aufgrund einer Lizenzbeschränkung auf eine maximale Programmausführung von 60 Minuten begrenzt ist. Um die Robustheit des End Systems nachzuweisen müssen, diese ohne dass die Zeit der Ausführung begrenzt ist wiederholt werden.

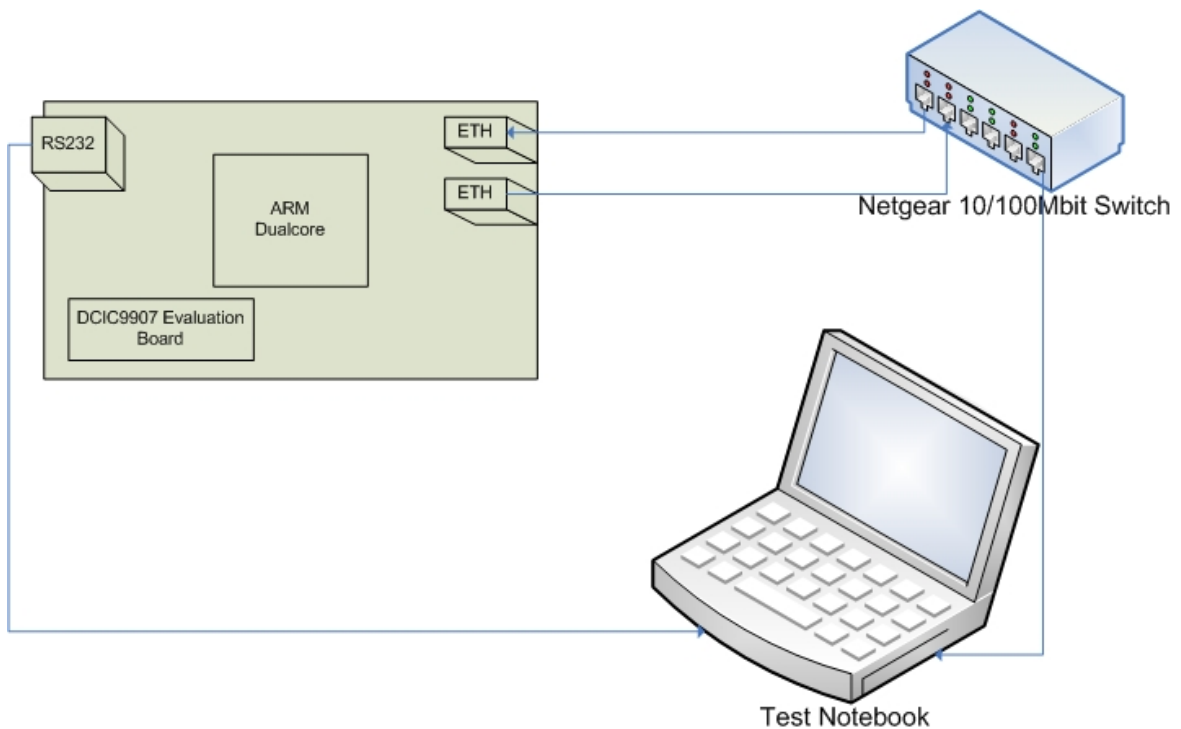


Abbildung 7.2: Dauertest

### 7.6.1 Test 1 Senden

Das System wird mit einem VL der Daten an das AFDX- Netzwerk überträgt gestartet. Das entspricht nicht dem gewünschten System und die Ergebnisse werden benötigt um Rückschlüsse auf die maximale Systemlast bei vollständiger Konfiguration zu erlauben.

Die Konfiguration der Software ist der folgenden Aufzählung zu entnehmen:

- 1 Virtueller Link zum senden
- 1 Ethernet Port
- Tasks die ausgeführt werden
  - Idle- Task
  - Root- Task
  - VI- Task Sender
- Kein BAG konfiguriert
- Framegröße 64 Byte und 1518 Byte

Die BAG wurde für diesen Test ausgeschaltet, um die maximale Übertragungsrate zu gewährleisten. Die BAG würde diese begrenzen und eine Aussage über die maximale Übertragungsrate verhindern.

Die AFDX- Paketlänge wurde auf das Minimum von 17 Byte eingestellt. Damit entspricht der auf dem Netzwerk übertragene Frame einer Länge von 84 Byte, die sich wie in Kapitel MAC-Beschränkungen 4.3.4 beschrieben, zusammen setzt.

### 7.6.1.1 Ergebnis

Die mit Wireshark aufgezeichneten Pakete entsprechen alle dem geforderten Aufbau 4.5.1.2. Das lässt den Rückschluss zu, dass die Frames nicht manipuliert wurden. Des Weiteren entsprach auch die Nachrichtenstruktur 4.6 der beschrieben.

Es konnten keine Paketverluste nachgewiesen werden.

Die folgenden Daten basieren auf der Messung von Frames der Länge von 64 Byte.

Übertragen wurden bei diesem Test 12781 Pakete in der Sekunde. Das ergibt pro Frame eine minimale Übertragungszeit von  $79\mu s$ .

Die wie folgt bestimmt wurde:

$$\Pi_{12781Frames/s}^{84Byte} = 1073604Byte/s$$

$$\frac{84Byte}{1073604Byte/s} = 0,000078241s$$

Je Frame ergibt sich also eine Übertragungszeit von

$$78,24\mu s$$

Der Wert für die errechnete Zeit unterscheidet sich um Faktor 10 von dem zu erwarteten Wert. Um eine genaue Aussage über die Ursache dieses Verhaltens treffen zu können, sind weitere Untersuchungen notwendig.

Die folgenden Daten basieren auf der Messung von Frames der Länge von 1518 Byte.

Übertragen wurden bei diesem Test 4166 Paket in der Sekunde. Das ergibt pro Frame eine minimale Übertragungszeit von  $240\mu s$ .

Die wie folgt bestimmt wurde:

$$\Pi_{4166Frames/s}^{1538Byte} = 6407308Byte/s$$



$$\frac{1538 \text{ Byte}}{6407308 \text{ Byte/s}} = 0,000240038 \text{ s}$$

Je Frame ergibt sich also eine Übertragungszeit von

$$240,04 \mu\text{s}$$

Der Wert für die errechnete Zeit unterscheidet sich um Faktor 2 von dem zu erwarteten Wert. Auch hier sind weitere Untersuchungen nötig, um eine genaue Ursache für das Verhalten zu bestimmen.

### 7.6.2 Test 2 Senden

Das System wird mit zwei VLs gestartet, die Daten an das AFDX- Netzwerk übertragen. Dabei wird jedem der VLs eine Ethernet- Schnittstelle zugewiesen.

Die Konfiguration der Software ist der folgenden Aufzählung zu entnehmen:

- 2 Virtueller Link zum senden
- 2 Ethernet Port
- Tasks die ausgeführt werden
  - Idle- Task
  - Root- Task
  - 2 VI- Task Sender
- Kein BAG konfiguriert
- Framegröße 64 Byte

Die BAG wurde für diesen Test ausgeschaltet, um die maximale Übertragungsrate zu gewährleisten. Die BAG würde diese begrenzen und eine Aussage über die maximale Übertragungsrate verfälschen.

Die AFDX- Paketlänge wurde auf das Minimum von 17 Byte eingestellt. Damit entspricht der auf dem Netzwerk übertragene Frame einer Länge von 84 Byte, die sich wie in Kapitel MAC-Beschränkungen [4.3.4](#) beschrieben, zusammen setzt.

### 7.6.2.1 Ergebnis

Die Ergebnisse entsprechen denen des Test 1 7.6.1. Abweichungen sind nur im Zeitverhalten festzustellen. Auch hier kam es nicht zu Paketverlusten.

Übertragen wurden bei diesem Test 10645 Paket in der Sekunde. Das ergibt pro Frame eine minimale Übertragungszeit von  $94\mu s$ .

Die wie folgt bestimmt wurde:

$$\prod_{10645Frames/s}^{84Byte} = 894180Byte/s$$

$$\frac{84Byte}{894180Byte/s} = 0,000093941s$$

Je Frame ergibt sich also eine Übertragungszeit von

$$93,94\mu s$$

Der Wert für die errechnete Zeit unterscheidet sich auch hier, um Faktor 10 von dem erwarteten Wert. Weitere Test zur bestimmen der Ursache sind hier zu vollziehen.

### 7.6.3 Test 3 Senden und Empfangen

Das System wird mit zwei VLs gestartet, wobei ein VL Daten über einen Switch an den anderen VL überträgt. Dabei wird jedem der VLs eine Ethernet- Schnittstelle zugewiesen.

Die Konfiguration der Software ist der folgenden Aufzählung zu entnehmen:

- 1 Virtueller Link zum senden
- 1 Virtueller Link zum empfangen
- 2 Ethernet Port
- Tasks die ausgeführt werden
  - Idle- Task
  - Root- Task
  - 1 VI- Task Sender
  - 1 VI- Task Empfänger

- BAG konfiguriert
- Framegröße 258 Byte

Die BAG wurde für diesen Test auf 110 *ms* gesetzt, das bedeutet, der sendende Task hat die Möglichkeit seine Nachricht einmal in 110 *ms* zu senden.

Die AFDX- Paketlänge wurde auf das Minimum von 192 Byte eingestellt. Damit entspricht der auf dem Netzwerk übertragene Frame einer Länge von 258 Byte.

### 7.6.3.1 Ergebnis

Die Auswertung der Log- Dateien hat ergeben, dass es zu keinen Paketverlusten kam. Die Nachrichtenstruktur wurde bei allen gesendeten Paketen wie beschrieben eingehalten. Damit ist davon auszugehen, dass es für die Dauer des Tests zu keinen Manipulationen während der Übertragung gekommen ist.

Übertragen wurden bei diesem Test 32727 Pakete in der Sekunde.

Die wie folgt bestimmt wurde:

$$\Pi_{32727Frames/h}^{258Byte} = 8443566Byte/h$$

Um die Zeit für ein Frame zu erhalten muss die Anzahl der übertragenen Frames pro Sekunde bestimmt werden.

$$\frac{32727Frames/h}{3600} = 9,091Frames/s$$

Anschließend wird die Zeit bestimmt, die der VL benötigt hat das Frame zu übertragen.

$$\frac{1s}{9,091Frames/s} = 0,10999s/Frame$$

Das bedeutet das alle 105.35 *ms* ein Frame übertragen wurde.

### 7.6.4 Test 4 Senden und Empfangen

Der Test 4 entspricht bis auf die AFDX- Payloadgröße dem Test 3. Die AFDX- Payloadgröße wurde hier auf das Maximum von 1471 Byte gesetzt. Das entspricht dem Worstcase- Fall. Alle Pakete die diese Payloadgröße überschreiten werden fragmentiert und das wird nicht vom End System unterstützt.

Die Ergebnisse sind mit denen des Test 3 identisch und werden deshalb nicht erneut beschrieben [7.6.3](#).

## 8 Zusammenfassung

Ziel dieser Arbeit war es ein eingebettetes System, das nach ARINC 664 Standard kommuniziert, zu entwickeln. Ausserdem galt es den Nachweis zu erbringen, dass dieses eingebettete System den geforderten Performanzanforderungen genügt.

Dazu wurde der von ARINC veröffentlichte AFDX- Standard 664 P7 analysiert, um eine Gewichtung für ein geeignetes eingebettetes System zur Implementation der Software zu finden.

Das DCIC 9907 Evaluationboard der Firma Dualcore Semiconductors traf die zu erfüllenden Performanz- und Peripherieanforderungen. In Verbindung mit dem dazugehörigen Evaluationboard und dem Echtzeitbetriebssystem war es möglich eine entsprechende Umgebung für das eingebettete System zu realisieren.

Umzusetzen waren die Funktionalitäten der Virtuellen Links die über die MAC- Adressen eine scheinbar physikalische Verbindung zu anderen Komponenten im AFDX- Netzwerk herstellen. Des Weiteren wurde die Integritätsprüfung, das Redundanzmanagement sowie die Nachrichtenstruktur auf Softwareebene realisiert.

Während des Entwurfs der Software stellte sich heraus, dass der zur Umsetzung verwendete Ethernet- Treiber nicht über die gewünschte Eigenschaft verfügt, mehrere MAC- Adressen an einen Ethernet- Port zu binden. Die Firma Euros Embedded Systems GmbH war bereit den Quellcode der Treiber zur Anpassung der benötigten Funktionalität zur Verfügung zu stellen. Eine Verifikation der Treiber nach Änderung des Quellcodes wäre nicht im Rahmen dieser Arbeit realisierbar gewesen.

Wie vermutet konnte auch der Netzwerkmanager des verwendeten TCP/IP- Stack nicht mehr als eine MAC- Adresse pro Ethernet- Adapter verwalten. Die Anpassung des Netzwerkmanagers in Bezug auf die Verwaltung der MAC- Adressen ist eine Grundvoraussetzung für den Erfolgreichen Einsatz der Hardware. Die Arbeiten am Netzwerkmanager waren in der kürze der Zeit nicht zu realisieren.

Die Software ermöglicht Frames AFDX- Konform zu versenden. Die Funktion der Virtuellen Links wurde vollständig implementiert, auch die auf die eventuell spätere Integration der MAC- Adressen wurde geachtet. Die Integritätsprüfung wurde vollständig umgesetzt und

getestet. Das Redundanzmanagement wurde nicht realisiert, da es zum Zeitpunkt der Performanzprüfungen keine Möglichkeit gab redundante Pakete auf beiden Ethernet-Adaptern zu empfangen.

Nachdem die Software für das System entwickelt war, folgten die Performanztests 7.6. An dieser Stelle ist darauf hinzuweisen das die Ergebnisse zu dem Test 1 (7.6.1) und Test 2 (7.6.2) nicht den erwarteten entsprachen. Die erwartete Datenrate wurde nicht erreicht. Der Test 3 und der Test 4 (7.6.3) konnten die erwarteten Ergebnisse erfüllen.

Grundsätzlich ist die Umsetzung des AFDX-Standard auf das vorliegende System positiv zu bewerten und kann zu einer Kostengünstigen Lösung der Anbindung der Komponenten an das AFDX-Netzwerk beitragen.

## 9 Ausblick

Für eine kommerzielle Anwendung des Systems im Bereich der zivilen Luftfahrt sind Anpassungen am Ethernet- Treiber und am Netzwerkmanager nötig.

Die Erkenntnisse, die in der Arbeit gesammelt wurden, werden durch eine weiter bei Monogram Systems GmbH durchgeführte Diplomarbeit aufgegriffen und näher untersucht.

# Literaturverzeichnis

[IEEE ] : <http://ieeexplore.ieee.org/>

[CAN 1991] : *CAN Specification*. 1991

[ARINC 2005] : *AIRCRAFT DATA NETWORK PART 7 AVIONICS FULL DUPLEX SWITCHED ETHERNET (AFDX) NETWORK*. 2005

[DualCore 2006] DUALCORE: *Dual Core Industrial Controller DCIC9907 Data Book 0.95*. DualCore Semiconductor GmbH, 2006

[EUROS 2002] EUROS: *EUROSplus Driver Implementation Manual*. EUROS Embedded Systems GmbH, 2002

[EUROS 2005a] EUROS: *DualCore DCIC9907 Ethernet Port-Treiber*. EUROS Embedded Systems GmbH, 2005

[EUROS 2005b] EUROS: *EUROSplus Referenzhandbuch*. EUROS Embedded Systems GmbH, 2005

[EUROS 2005c] EUROS: *EUROSplus TCP/IP Networking*. EUROS Embedded Systems GmbH, 2005

[EUROS 2006] EUROS: *EUROSplus Programmier-Handbuch*. EUROS Embedded Systems GmbH, 2006

[Gruhler 2001] GRUHLER, Gerhard: *Feldbusse und Geräte-Kommunikationssysteme*. Franzis, 2001. – ISBN 3-7723-5745-8

[Rech 2002] RECH, Jörg: *Ethernet Technologien und Protokolle für die Computervernetzung*. Heise, 2002. – ISBN 3-88229-186-9

[Scholz 2005] SCHOLZ, Peter: *Softwareentwicklung eingebetteter Systeme*. Springer, 2005. – ISBN 3-540-23405-5



# Glossar

**ADN** Aircraft Data Networks

**AFDX** Avionics Full Duplex Switched Ethernet

**AHB** ARM High Performance Bus

**ARINC** AERONAUTICAL RADIO, INC.

**ARM** Advanced RISC Machine

**AUI** Attachment Unit Interface

**BAG** Bandwidth Allocation Gap

**BGA** Ball Grid Array

**CAN** Controller Area Network

**Cat** Bezeichnet die Kategorie der Kabel die im Ethernet- Netzwerken verwendet werden

**CiA** CAN in Automation e.V.

**CPU** Central Processing Unit

**CRC** Cycling Redundancy Check

**CSMA/CD** Carrier Sense Multiple Access/ Collision Detection

**DCIC** Dual Core Industrial Controller

**DIX** Eine Gruppe die 1979 von DEC, Intel und Xerox gegründet wurde und Entwicklungen im Bereich des Ethernets betrieb

**DMA** Direct Memory Access

- DS** Data Set
- DTCM** Tightly Coupled Data Memory
- DTE** Data Terminal Equipment
- ENET** Ethernet Modul
- ERX** Ethernet Realtime Extension
- ES** End System
- ESD** End- of- Stream Delimiter
- ETH** Ethernet
- EUROS** Enhanced Universal Realtime Operating System
- FDDI** Fibre Distributed Data Interface
- FDS** Functional Data Set
- FIFO** First In First Out
- FSS** Functional Status Set
- FTP** File Transfer Protocol
- GAA** Global Administered Address
- IANA** Internet Assigned Numbers Authority
- ID** Identifier
- IDL** Interface Description Language
- IEEE** Institute of Electrical and Electronical Engineers
- IFG** Interframe Gap
- IP** Internet Protocol
- IPv4** Internet Protocol Version 4

- 
- ISO** International Standards Organisation
  - ITCM** Tightly Coupled Instruction Memory
  - LAA** Locally Administered Address
  - LAN** Local Area Network
  - LSB** Least Significant Bit
  - MAC** Media Access Control
  - MAU** Multi Access Unit
  - MDI** Media Dependent Interface
  - MII** Media Independent Interface
  - MSB** Most Significant Bit
  - MUX** Multiplexer
  - Nbw** Network bandwidth
  - NRVL** Nicht Redundant Virtueller Link
  - NRZI** Non Return to Zero Invers
  - OSI** Open- Systems- Interconnection Referenzmodell
  - PCI** Protocol Control Information
  - PCS** Physical Coding Sublayer
  - PDU** Protocol Data Unit
  - PHY** Physical Layer Control
  - PLS** Physical Signaling Service
  - PMA** Physical Medium Attachment
  - PMD** Physical Medium Dependent

- 
- PSN** Pre Sequence Number
- PTP** Precision Time Protocol
- RD** Receive Data
- RFC** Request for Comments
- RMI** Reduced Media Independent Interface
- RS-485** Recommended Standard 485
- RVL** Redundant Virtueller Link
- Rx** Receiver
- SAP** Service Access Point
- SDRAM** Synchronous Dynamic Random Access Memory
- SDU** Service Data Unit
- SFD** Start- of- Frame Delimiter
- SoC** System-on-Chip
- TCP** Transmission Control Protocol
- TD** Transmit Data
- TFTP** Trivial File Transfer Protokoll
- TLT** Technological Latency
- TP- Kabel** Twisted- Pair- Kabel
- TTL** Transistor Transistor Logic
- TTP** Time Triggered Protocol
- Tx** Transmitter
- UDP** User Datagram Protocol

**UTP** Unshielded- Twisted- Pair- Kabel

**VDC** Volts of Direct Current

**VL** Virtueller Link

# Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 24. April 2008

Ort, Datum

Unterschrift