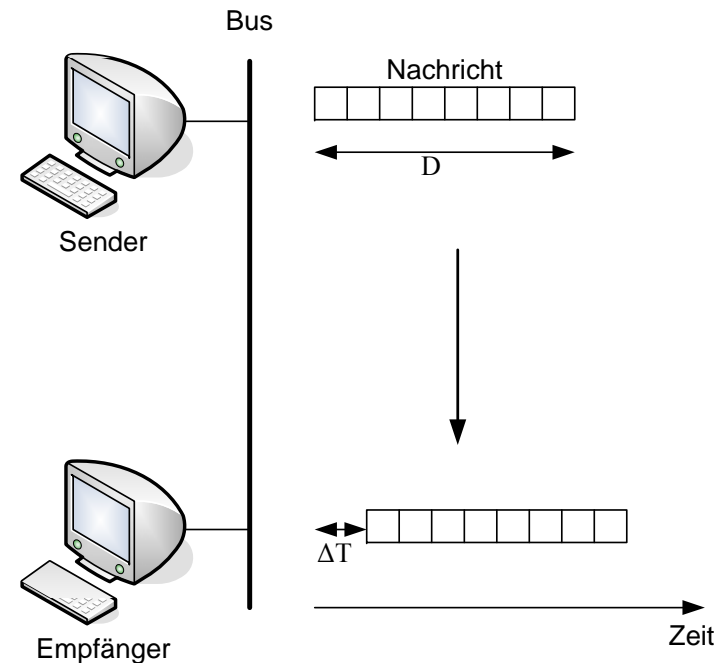


Kollisionen

- Um Kollisionen rechtzeitig zu erkennen muss die Signallaufzeit ΔT deutlich kleiner als die Nachrichtenübertragungsdauer D sein.
- Das Störsignal (JAM) wird geschickt um alle anderen Nachrichten auf die Kollision aufmerksam zu machen \Rightarrow Verkürzung der Zeit zur Kollisionserkennung
- Würden die Rechner nach einer Kollision nicht eine zufällige Zeit warten, käme es sofort zu einer erneuten Kollision.
- Lösung im Ethernet: Die Sender wählen eine zufällige Zahl d aus dem Intervall $[0 \dots 2^i]$, mit i = Anzahl der bisherigen Kollisionen (Backoff-Verfahren).
 \Rightarrow Mit ansteigendem i wird eine Kollision immer unwahrscheinlicher.
 \Rightarrow Bei $i = 16$ wird die Übertragung abgebrochen und ein Systemfehler vermutet.



TCP vs. UDP

- TCP (Transmission Control Protocol) ist ein zuverlässiges, verbindungsorientiertes Transportprotokoll:
 - Vor der Übertragung der Daten wird zunächst eine Verbindung zwischen Sender und Empfänger aufgebaut (Handshake).
 - Datenverluste werden erkannt und automatisch behoben durch Neuversenden des entsprechenden Datenpakets.
 - ⇒ Aufgrund von unvorhersehbaren Verzögerungen (Backoff-Verfahren) und hohem Overhead ist TCP nicht für den Einsatz in Echtzeitsystemen geeignet.
 - Weiteres Problem: Slow Start der Congestion Control Strategie von TCP/IP ⇒ zu Beginn der Übertragung wird nicht die volle Bandbreite ausgenutzt
- UDP (User Datagram Protocol) ist ein minimales, verbindungsloses Netzprotokoll:
 - Verwendung vor allem bei Anwendungen mit kleinen Datenpaketen (Overhead zum Verbindungsaufbau entfällt)
 - UDP ist nicht-zuverlässig: Pakete können verloren gehen und in unterschiedlicher Reihenfolge beim Empfänger ankommen.
 - ⇒ Einsatz in weichen Echtzeitsystemen, in denen der Verlust einzelner Nachrichten toleriert werden kann (z.B. Multimedia-Protokollen wie z.B. VoIP, VoD) möglich.

RTP, RTSP: Motivation

- Problem von UDP/IP in Multimediasystemen:
 - keine Möglichkeit zur Synchronisation
 - verschiedene Multimediaströme können kollidieren (z.B. in VoD)
 - Qualitätskontrolle ist wünschenswert

⇒ in Multimediasystemen werden zusätzliche Protokolle (RTP, RTCP) verwendet.
- Multimediaverbindung mit RTP/RTCP
 - Zur Übertragung der **Steuerungsnachrichten** (in der Regel nicht zeitkritisch) werden zuverlässige Protokolle eingesetzt (z.B. TCP/IP)
 - Zur **Datenübertragung** wird ein **RTP (Real-Time Transport Protocol)**-Kanal eingesetzt.
 - Jeder RTP-Kanal wird mit einem **RTCP (Real-Time Control Protocol)**-Kanal zur Überwachung der Qualität verknüpft.
 - RTP/RTCP setzen in der Regel auf UDP/IP auf und sind End-zu-End-Protokolle

RTP, RTCP

- RTP:
 - Multicasting
 - Bestimmung des Datenformats (PT)
 - Zeitgebend durch Zeitstempel, die Berechnung des Jitters wird dadurch möglich
 - Möglichkeit zur Ordnung der Pakete und zum Erkennen von verlorenen Paketen durch Sequenznummer

Byte 0				Byte 1				Byte 2				Byte 3																			
Bit 0	1	2	3	4	5	6	7	Bit 0	1	2	3	4	5	6	7	Bit 0	1	2	3	4	5	6	7	Bit 0	1	2	3	4	5	6	7
V=2	P	X	CC	M	PT			sequence number																							
timestamp (in sample rate units)																															
synchronization source (SSRC) identifier																															
contributing source (CSRC) identifiers (optional)																															
Header Extension (optional)																															

RTP Header

- RTCP:
 - Überwachung der Qualität der Datenkanal: versandte Daten/Pakete, verlorene Pakete, Jitter, Round trip delay
 - Unterschiedliche Pakete stehen zur Verfügung: Sender report, receiver report, source description und anwendungsspezifische Pakete

Zusammenfassung Ethernet

- Ethernet ist aufgrund des CSMA/CD Zugriffsverfahrens für harte Echtzeitsysteme nicht geeignet:
 - unbestimmte Verzögerungen durch Backoff-Verfahren
 - keine Priorisierung von Nachrichten möglich
- Aufgrund der starken Verbreitung (\Rightarrow niedrige Kosten, gute Unterstützung) wird Ethernet dennoch häufig in Echtzeitsystemen eingesetzt:
 - Durch Verwendung von echtzeitfähigen Protokollen in weichen Echtzeitsystemen (z.B. Multimedialkontrolle).
 - Durch Verringerung der Kollisionswahrscheinlichkeit durch Aufteilung des Netzes in verschiedene Kollisionsdomänen (z.B. switched ethernet).
- Mittlerweile werden auch diverse Implementierungen von Real-Time Ethernet eingesetzt, allerdings gibt es noch keinen allgemein anerkannten Standard (siehe Zusammenfassung/Trends).



Echtzeitfähige Kommunikation

Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA*)

Vertreter: CAN

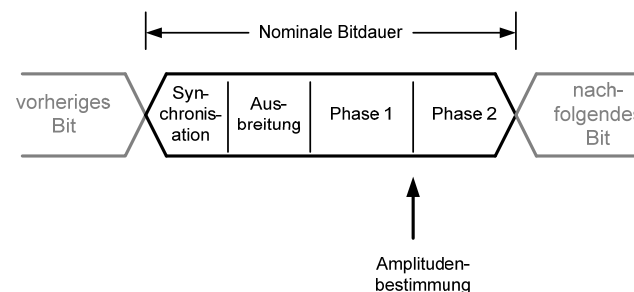
Teilweise wird die hier vorgestellte Methode auch CSMA/CR (Collision Resolution) genannt.

CAN-Protokoll

- Grundidee von Collision Avoidance:
 - Kollisionen werden rechtzeitig erkannt, bevor Nachrichten unbrauchbar werden
 - Wichtigere Nachrichten werden bevorzugt \Rightarrow Priorisierung der Nachrichten
- Daten:
 - CAN (Controller Area Network) wurde 1981 von Intel und Bosch entwickelt.
 - Einsatzbereich: vor allem Automobilbereich, Automatisierungstechnik
 - Datenübertragungsraten von bis zu 1Mbit/s, Reichweite 1km
 - Implementierung der Schichten 1,2 und 7 des ISO/OSI-Modells

CAN: Schicht 1

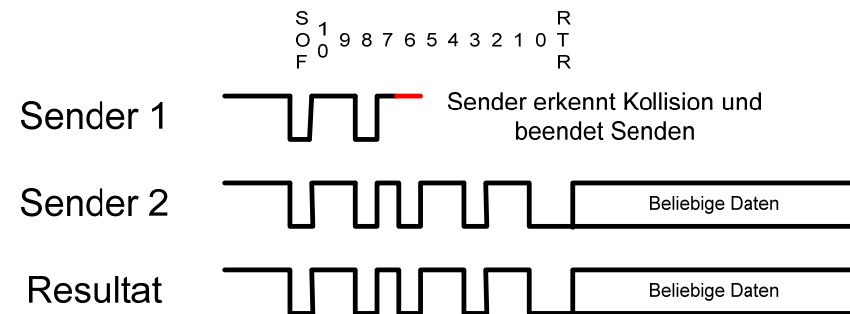
- Busmedium:
 - Kupfer oder Glasfaser
 - Empfehlung Twisted Pair: Möglichkeit zur differentiellen Übertragung (robuster gegenüber Störungen)
- Codierung: NRZ-L (Non-Return-to-Zero-Level)
 - Problem mit NRZ-L: lange Sequenzen monotone Sequenzen von 0 oder 1 können zu Problemen bei der Synchronisation führen, in CAN wird deshalb nach fünf gleichen Bits ein inverses Bit eingefügt (**Bitstuffing**)
- Daten werden **bitsynchron** übertragen:
 - ⇒ Datenübertragungsrate und maximale Kabellänge sind miteinander verknüpft.
 - Konfigurationsmöglichkeiten:
 - 1 MBit/s, maximale Länge: 40m
 - 500 kBit/s, maximale Länge: 100m
 - 125 kBit/s, maximale Länge: 500m
 - Maximale Teilnehmerzahl: 32-128



http://www.port.de/pdf/CAN_Bit_Timing.pdf

CAN: Schicht 2

- Realisierung eines CSMA/CA-Verfahrens:
 - Bei der Übertragung wirken Bits je nach Wert entweder **dominant** (typischerweise 0) oder **rezessiv** (1).
 - Dominante Bits überschreiben rezessive Bits, falls sie gleichzeitig gesendet werden.
 - Jedem Nachrichtentyp (z.B. Sensorwert, Kontrollnachricht) wird ein Identifikator zugewiesen, der die Wichtigkeit des Typs festlegt.
 - Jeder Identifikator sollte nur einem Sender zugewiesen werden.
 - Wie bei Ethernet wartet der Sender bis der Kanal frei ist und startet dann die Versendung der Nachricht.



- Beim gleichzeitigen Senden zweier Nachrichten, dominiert der Identifikator des wichtigeren Nachrichtentyps, den Sender der unwichtigeren Nachricht beendet das Senden.
- ⇒ Verzögerung von hochpriorigen Nachrichten auf die maximale Nachrichtenlänge begrenzt (in Übertragung befindliche Nachrichten werden nicht unterbrochen)

CAN: Framearten

- Datenframe:
 - Versand von maximal 64bit Daten
- Remoteframe:
 - Verwendung zur Anforderung von Daten
 - Wie Datenframe, nur RTR-Feld auf 1 gesetzt
- Fehlerframe:
 - Signalisierung von erkannten Fehlerbedingungen
- Überlastframe:
 - Zwangspause zwischen Remoteframe und Datenframe

Länge in Bit	1	11	1	1	1	4	0..64	15	1	1	1	7	3
Zweck	Start of frame	Identifier (Extended CAN 27bit)	Remote Transmission Bit	Identifier Extension Bit	reserviert	Datenlängenfeld	Datenfeld	CRC-Prüfsumme	CRC Delimiter	Bestätigungsslot	Bestätigungsdelimiter	End of Frame	Intermission

CAN: Schicht 7

- Im Gegensatz zu Schicht 1 und 2 ist die Schicht 7 nicht in einer internationalen Norm spezifiziert.
- Es existieren jedoch diverse Implementierungen (z.B. CANOpen) für Dienste der Schichten 3-7 zur Realisierung von:
 - Flusskontrolle
 - Geräteadressierung
 - Übertragung größerer Datenmengen
 - Grunddienste für Anwendungen (Request, Indication, Response, Confirmation)
- Zudem gibt es Versuche eine Norm CAL (CAN Application Layer) einzuführen.
- Ziele:
 - Einheitliche Sprache zur Entwicklung von verteilten Anwendungen
 - Ermöglichung der Interaktion von CAN-Modulen unterschiedlicher Hersteller



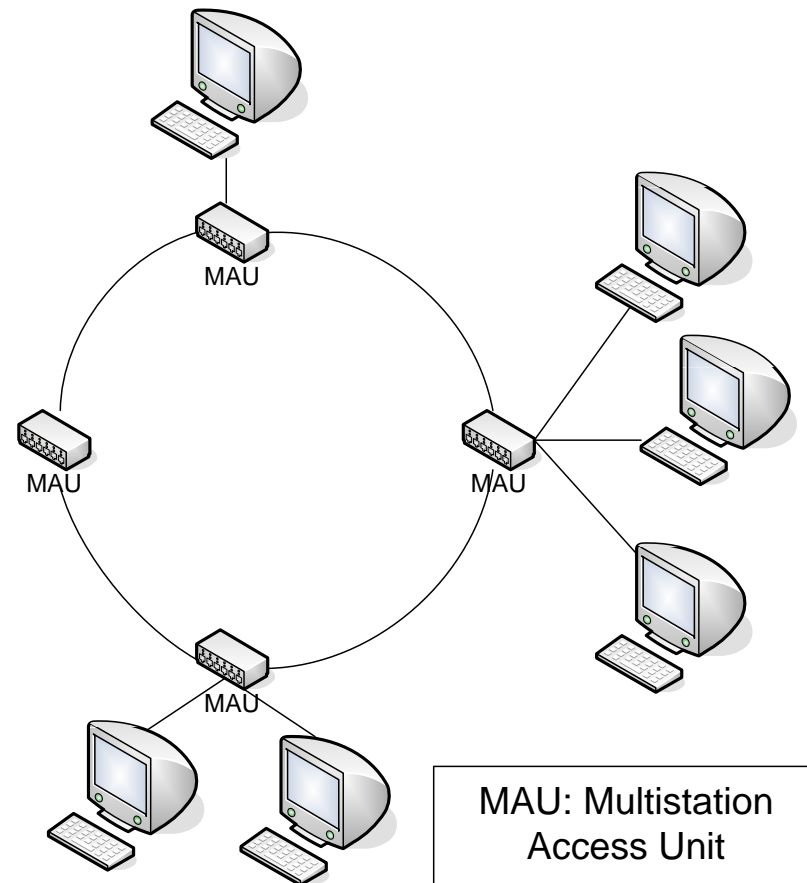
Echtzeitfähige Kommunikation

Tokenbasierte Verfahren

Vertreter: Token Ring

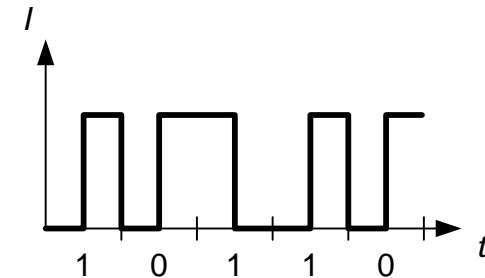
Tokenbasierte Verfahren

- Nachteil von CSMA/CA: Begrenzung der Datenrate und der Netzlänge durch Bitsynchronität
- Tokenbasierter Ansatz: Eine Einheit darf nur dann senden, wenn sie eine Berechtigung (Token) besitzt.
- Die Berechtigung wird zumeist zyklisch weitergegeben \Rightarrow Token Ring.
- Die Berechtigung / das Token ist dabei eine spezielle Bitsequenz.



Token Ring: Schicht 1

- Token Ring wird im Standard IEEE 802.5 spezifiziert.
- Erreichbare Geschwindigkeiten: 4 bzw. 16 MBit/s
⇒ aufgrund der Kollisionsfreiheit mit den effektiven Datenübertragungsraten von 10 bzw. 100 MBit/s Ethernet vergleichbar
- Codierung:
 - differentieller Manchester-Code
 - somit selbstsynchronisierend
- Topologie:
 - Ring
 - aufgrund der möglichen Verwendung von MAUs auch sternförmige Verkabelung möglich



Differentieller Manchester-Code

Token Ring: Zugriffsverfahren

1. Die Station, die das Token besitzt, darf Daten versenden.
2. Das Datenpaket wird von Station zu Station übertragen.
3. Die einzelnen Stationen empfangen die Daten und regenerieren sie zur Weitersendung an den nächsten Nachbarn.
4. Der Empfänger einer Nachricht kopiert die Nachricht und leitet die Nachricht mit dem gesetzten C-Bit (siehe Nachrichtenaufbau) zur Empfangsbestätigung weiter.
5. Empfängt der Sender seine eigene Nachricht, so entfernt er diese aus dem Netz.
6. Nach Ende der Übertragung wird auch das Token weitergesendet (maximale Token-Wartezeit wird vorher definiert, Standardwert: 10ms)
7. Im 16 MBit/s Modus wird das Token direkt im Anschluß an das Nachrichtenpaket versendet (**early release**) \Rightarrow es können sich gleichzeitig mehrere Token im Netz befinden

Token Ring: Prioritäten

- Token Ring unterstützt Prioritäten:
 - Insgesamt gibt es 8 Prioritätsstufen (3 Bit)
 - Jeder Station wird eine Priorität zugewiesen.
 - Der Datenrahmen besitzt ebenfalls einen Speicherplatz für die Priorität.
 - Eine Station kann in die Priorität in dem Prioritätsfeld von Nachrichten vormerken, allerdings darf die Priorität nur erhöht werden.
 - Stationen dürfen Tokens nur dann annehmen, wenn ihre Priorität mindestens so hoch ist, wie die Priorität des Tokens.
 - Applet zum Ablauf:
<http://www.nt.fh-koeln.de/vogt/mm/tokenring/tokenring.html>

Token Ring: Token Paket

- Das Token besteht aus:
 - Startsequenz (1 Byte, JK0JK000)
 - J, K: Codeverletzungen entsprechend Manchester-Code (kein Übergang in Taktmitte)
 - Zugriffskontrolle (1 Byte, PPPTMRRR)
 - P: Zugriffspriorität
 - T: Tokenbit (0: freies Token, 1: Daten)
 - M: Monitorbit
 - R: Reservierungspriorität
 - Endsequenz (1 Byte, JK1JK1IE)
 - I: Zwischenrahmenbit (0: letztes Paket, 1: weitere Pakete folgen)
 - E: Fehlerbit (0: fehlerfrei, 1: Fehler entdeckt)

Token Ring: Tokenrahmen

- Der Datenrahmen besteht aus:
 - Startsequenz wie Token
 - Zugriffskontrolle wie Token
 - Rahmenkontrolle (1 Byte, FFrrZZZZ)
 - FF: Paketart (00: Protokollsteuerpaket, 01: Paket mit Anwenderdaten)
 - rr: reserviert für zukünftige Anwendungen
 - ZZZZ: Informationen zur Paketpufferung
 - Zieladresse (6 Byte): Adresse eines spezifischen Geräts oder Multicast-Adresse
 - Quelladresse (6 Byte)
 - Routing Informationen (0-30 Bytes): optional
 - Daten
 - Prüfsumme FCS (4 Byte): Berechnung auf Basis der Daten zwischen Start- und Endsequenz
 - Endsequenz wie Token
 - Paketstatus (1 Byte ACrrACrr)
 - A: Paket wurde vom Empfänger als an in adressiert erkannt
 - C: Paket wurde vom Empfänger erfolgreich empfangen

Token Ring: Monitor

- Für den fehlerfreien Ablauf des Protokolls existiert im Token Ring ein Monitor.
- Aufgaben:
 - Entfernung von fehlerhaften Rahmen
 - Neugenerierung eines Tokens bei Verlust des Tokens (nach Ablauf einer Kontrollzeit)
 - Entfernung endlos kreisender Nachrichten bei Ausfall der Senderstation (Markierung der Nachricht beim Passieren des Monitors, Löschen der Nachricht beim 2. Passieren)
 - Signalisierung der Existenz des Monitors (durch Active Monitor Present Nachricht)

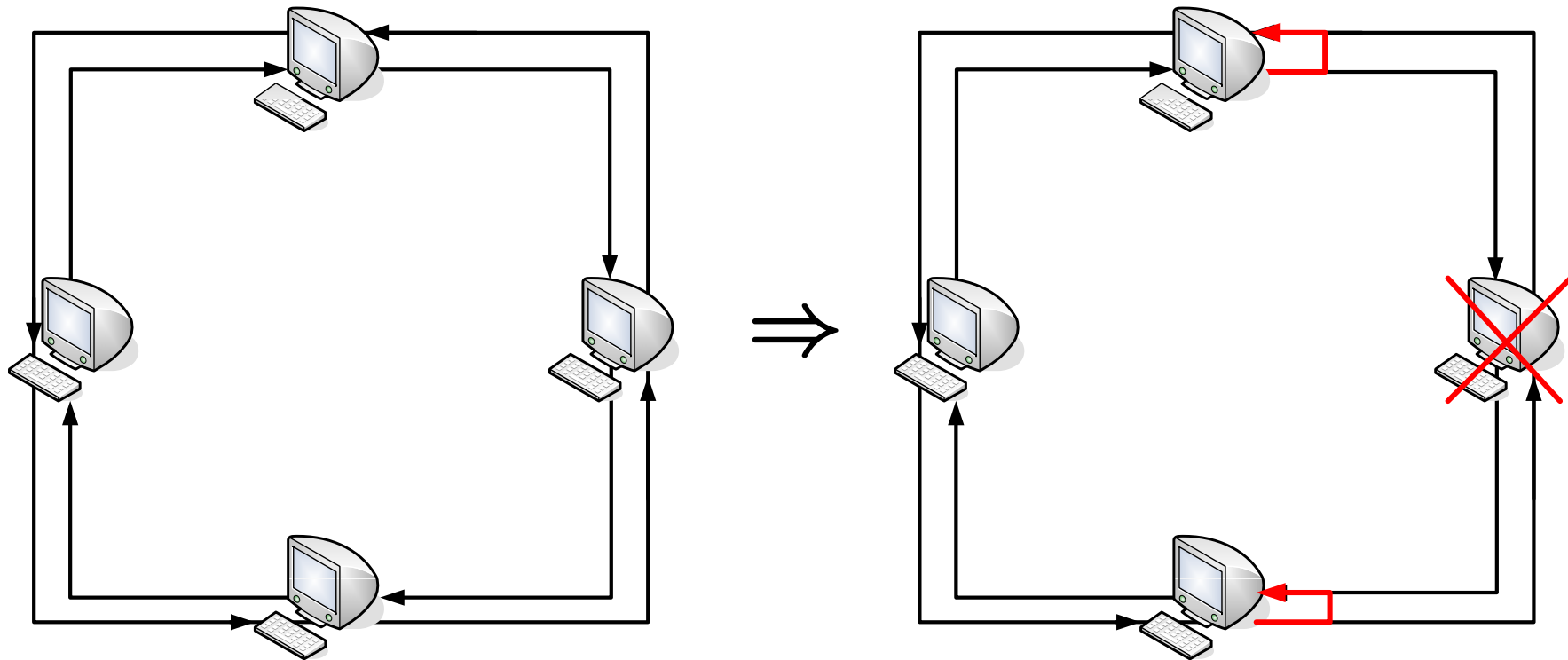
Token Ring: Initialisierung / Rekonfigurierung

- Bei der Initialisierung bzw. dem Ablauf des Standby Monitor Timer (Mechanismus zur Tolerierung des Ausfalls des Monitors)
 1. Senden eines Claim Token Paketes
 2. Überprüfung, ob weitere Pakete die Station passieren
 3. Falls nein \Rightarrow Station wird zum Monitor
 4. Generierung eines Tokens
 5. Jede Station überprüft mittels des Duplicate Adress Test Paketes, ob die eigene Adresse bereits im Netzwerk vorhanden ist.
- Der Ausfall einer Station kann durch das Netzwerk erkannt werden und evtl. durch Überbrückung kompensiert werden.

FDDI

- Fiber Distributed Data Interface (FDDI) ist eine Weiterentwicklung von Token Ring
- Medium: Glasfaserkabel
- doppelter gegenläufiger Ring (aktiver Ring, Reservering) mit Token-Mechanismus
- Datenrate: 100 MBit/s, 1000 MBit/s
- Codierung: 4B5B (wie in FastEthernet)
- maximal 1000 Einheiten
- Ringlänge: max. 200 km
- Maximaler Abstand zwischen zwei Einheiten: 2 km
- Fehlertoleranz (maximal eine Station)
- Nachrichten können hintereinander gelegt werden (early release)
- Weitere Entwicklungen FDDI-2

Fehlerkonfiguration in FDDI



MAP / Token Bus

- **MAP: Manufacturing Automation Protocol** (Entwicklung ab 1982 von General Motors)
- Einsatz hauptsächlich im Produktionsbereich
- Schicht 1: anstelle von Ring-Topologie nun beliebige Topologie durch den Einsatz von Bridges, Gateways und Routern
- Medienzugriffsverfahren:
 - Token Bus, spezifiziert in IEEE 802.4
 - ähnlich Token-Ring, die benachbarte Station zur Weiterleitung des Tokens wird anhand einer Adresse bestimmt.
- In MAP werden zudem alle sieben Schichten des ISO/OSI-Modells spezifiziert.
- Aufgrund des Umfangs und der Komplexität konnte sich MAP nicht durchsetzen.
- Maximale Übertragungsrate: 10 MBit/s



Echtzeitfähige Kommunikation

Zeitgesteuerte Verfahren

Vertreter: TTP

Zugriffsverfahren: TDMA

- **TDMA (Time Division Multiple Access)** bezeichnet ein Verfahren, bei dem der Zugriff auf das Medium in Zeitscheiben (slots) eingeteilt wird.
- Die Zeitscheiben werden für jeweils einen Sender zur Verfügung gestellt.
- Vorteile:
 - Kollisionen sind per Design ausgeschlossen
 - Einzelnen Sendern kann eine Bandbreite garantiert werden.
 - Das zeitliche Verhalten ist vollkommen deterministisch.
 - Synchronisationsalgorithmen können direkt im Protokoll spezifiziert und durch Hardware implementiert werden.
- Nachteil:
 - keine dynamische Zuteilung bei reinem TDMA-Verfahren möglich
- Bekannte Vertreter: TTP, Flexray (kombiniert zeitgesteuert und dynamische Kommunikation)

Einführung TTP

- Entstanden an der TU Wien (SpinOff TTTech)
- TTP steht für Time Triggered Protocol
- TTP ist geeignet für harte Echtzeitsysteme:
 - verteilter, fehlertoleranter Uhrensynchronisationsalgorithmus (Einheit: 1 μ s), toleriert beliebige Einzelfehler.
 - Zwei redundante Kommunikationskanäle \Rightarrow Fehlersicherheit
 - Einheiten werden durch Guards geschützt (Vermeidung eines babbling idiots).
 - Kommunikationsschema wird in Form einer **MEDL (Message Descriptor List)** a priori festgelegt und auf die Einheiten heruntergeladen.
- Einsatz unter anderem im Airbus A380

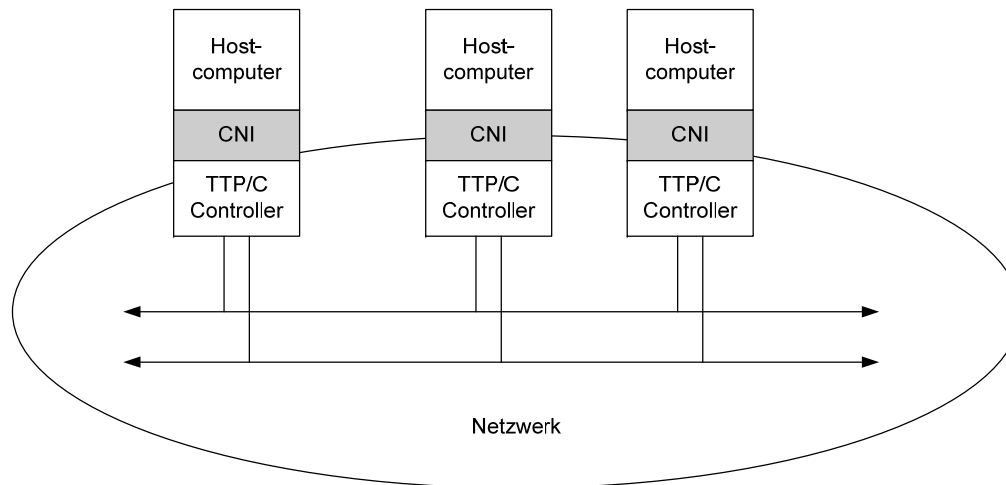


Echtzeitfähige Kommunikation

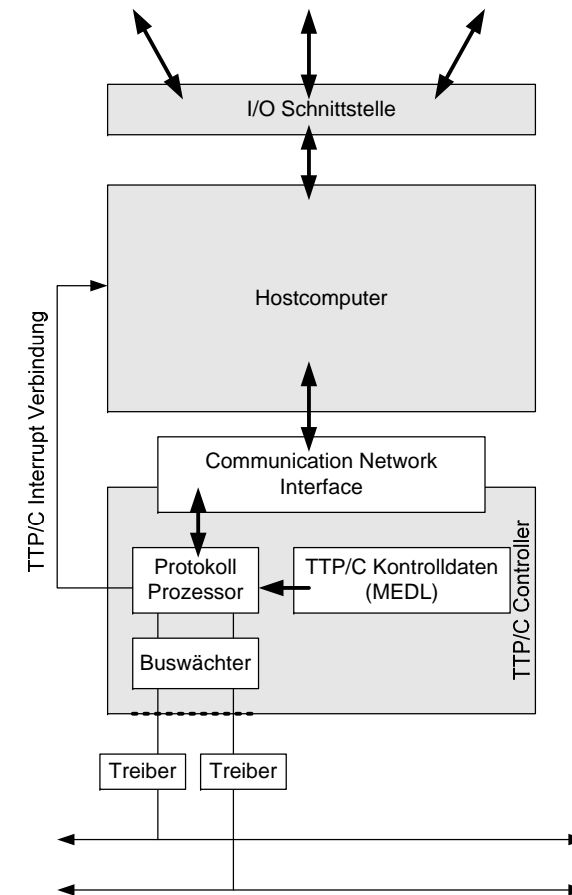
Zeitgesteuerte Verfahren

Vertreter: TTP

TTP-Architektur

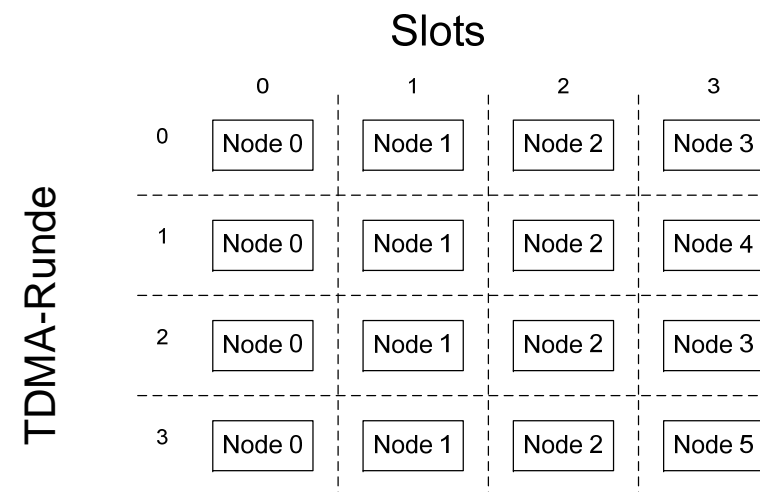


- Erläuterung:
 - Hostcomputer: Ausführung der eigentlichen Anwendung
 - CNI: Gemeinsamer Speicherbereich von Hostcomputer und TTP/C-Kontroller
 - Unterbrechungsverbindung: zur Übermittlung von Ticks der globalen Uhr und außergewöhnlicher Ereignisse an den Hostcomputer
 - MEDL: Speicherplatz für Kontrolldaten



TTP: Arbeitsprinzip

- Die Controller arbeiten autonom vom Hostcomputer (notwendige Daten sind in MEDL enthalten)
 - für jede zu empfangende und sendende Nachricht: Zeitpunkt und Speicherort in der CNI
 - zusätzliche Informationen zur Ausführung des Protokolls
- In jeder TDMA-Runde sendet ein Knoten genau einmal
 - Unterscheidung zwischen
 - reellen Knoten: Knoten mit eigenem Sendeschlitz
 - virtuelle Knoten: mehrere Knoten teilen sich einen Sendeschlitz
 - Die Länge der Sendeschlitze kann sich dabei unterscheiden, für einen Knoten ist die Länge immer gleich
⇒ TDMA-Runde dauert immer gleich lang



Protokolldienste

- Das Protokoll bietet:
 - Vorhersagbare und kleine, nach oben begrenzte Verzögerungen aller Nachrichten
 - Zeitliche Kapselung der Subsysteme
 - Schnelle Fehlerentdeckung beim Senden und Empfangen
 - Implizite Nachrichtenbestätigung durch Gruppenkommunikation
 - Unterstützung von Redundanz (Knoten, Kanäle) für fehlertolerante Systeme
 - Unterstützung von Clustermoduswechseln
 - Fehlertoleranter, verteilter Uhrensynchronisationsalgorithmus ohne zusätzliche Kosten
 - Hohe Effizienz wegen kleinem Protokollaufwand
 - Passive Knoten können mithören, aber keine Daten versenden.
 - Schattenknoten sind passive redundante Knoten, die im Fehlerfall eine fehlerhafte Komponente ersetzen können.

Fehlerhypothese

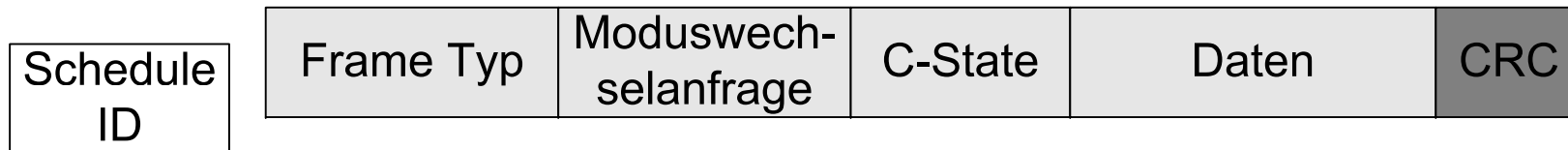
- Interne physikalische Fehler:
 - Erkennung einerseits durch das Protokoll, sowie Verhinderung eine babbling idiots durch Guards.
- Externe physikalische Fehler:
 - Durch redundante Kanäle können diese Fehler toleriert werden.
- Designfehler des TTP/C Controllers:
 - Es wird von einem fehlerfreien Design ausgegangen.
- Designfehler Hostcomputer:
 - Protokollablauf kann nicht beeinflusst werden, allerdings können inkorrekte Daten erzeugt werden.
- Permanente Slightly-Off-Specification-Fehler:
 - können durch erweiterte Guards toleriert werden.
- Regionale Fehler (Zerstören der Netzwerkverbindungen eines Knotens):
 - Folgen können durch Ring- und Sternarchitektur minimiert werden.

Zustandsüberwachung

- Das Protokoll bietet Möglichkeiten, dass Netzwerk zu analysieren und fehlerbehaftete Knoten zu erkennen.
- Der Zustand des Netzwerkes wird dabei im Kontrollerzustand (C-State) gespeichert.
- Der C-State enthält:
 - die globale Zeit der nächsten Übertragung
 - das aktuelle Fenster im Clusterzyklus
 - den aktuellen, aktiven Clustermodus
 - einen eventuell ausstehenden Moduswechsel
 - den Status aller Knoten im Cluster
- Das Protokoll bietet einen Votieralgorithmus zur Überprüfung des eigenen Zustands an.
- Ein Knoten ist korrekt, wenn er in seinem Fenster eine korrekte Nachricht versendet hat.
- Knoten können sich durch die Übernahme der Zeit und der Schedulingposition integrieren, sobald ein integrierender Rechner eine korrekte Nachricht sendet, erkennen in die anderen Knoten an.

Datenpakete in TTP

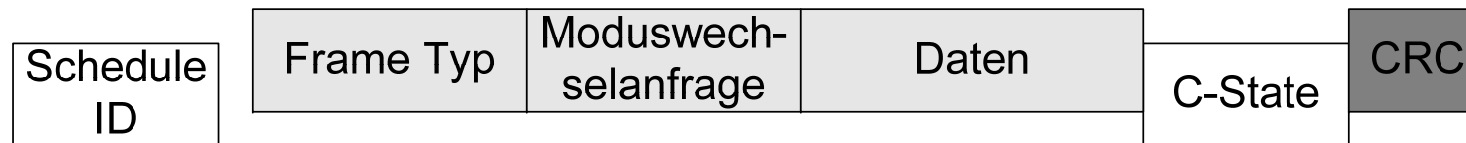
- Paket mit explizitem C-State



- Kaltstartpaket



- Paket mit implizitem C-State



In Frame enthalten, in CRC eingerechnet	Nicht in Frame enthalten, in CRC eingerechnet	Berechneter CRC
---	---	-----------------

TTP: Clusterstart

- Der Start erfolgt in drei Schritten:
 1. Initialisierung des Hostcomputers und des Controllers
 2. Suche nach Frame mit expliziten C-State und Integration
 3. a) Falls kein Frame empfangen wird, werden die Bedingungen für einen Kaltstart geprüft:
 - Host hat sein Lebenszeichen aktualisiert
 - Das Kaltstart Flag in der MEDL ist gesetzt
 - die maximale Anzahl der erlaubten Kaltstarts wurde noch nicht erreichtSind die Bedingungen erfüllt, sendet der Knoten ein Kaltstartframe.
 3. b) Falls Frame empfangen wird: Versuch zur Integration

TTP: Sicherheitsdienste / Synchronisation

- Sicherheitsdienste:
 - Korrektheit: Alle Knoten werden über die Korrektheit der anderen Knoten mit einer Verzögerung von etwa einer Runde informiert.
 - Cliquentdeckung: Es werden die Anzahl der übereinstimmenden und entgegengesetzten Knoten gezählt. Falls mehr entgegengesetzte Knoten gezählt werden, so wird ein Cliquentfehler angenommen.
 - Host/Kontroller Lebenszeichen: der Hostcomputer muss seine Lebendigkeit dem Kontroller regelmäßig zeigen. Sonst wechselt der Kontroller in den passiven Zustand.
- Synchronisation:
 - In regelmäßigen Abständen wird die Uhrensynchronisation durchgeführt.
 - Es werden die Unterschiede der lokalen Uhr zu ausgewählten (stabilen) Uhren (mind.4) anderer Rechner anhand den Sendezeiten gemessen.
 - Die beiden extremen Werte werden gestrichen und vom Rest der Mittelwert gebildet.
 - Die Rechner einigen sich auf einen Zeitpunkt für die Uhrenkorrektur.



Echtzeitfähige Kommunikation

Zusammenfassung

Zusammenfassung

- Die Eignung eines Kommunikationsmediums für die Anwendung in Echtzeitsystemen ist vor allem durch das Medienzugriffsverfahren bestimmt.
- Die maximale Wartezeit ist bei
 - CSMA/CD: unbegrenzt und nicht deterministisch (\Rightarrow keine Eignung für Echtzeitsysteme)
 - CSMA/CA, tokenbasierten Verfahren: begrenzt, aber nicht deterministisch (abhängig von anderen Nachrichten)
 - zeitgesteuerten Verfahren: begrenzt und deterministisch.
- Die Priorisierung der Nachrichten wird von CSMA/CA und tokenbasierten Verfahren unterstützt.
- Nachteil der zeitgesteuerten Verfahren ist die mangelnde Flexibilität (keine dynamischen Nachrichten möglich).
- Trotz diverser Nachteile geht der Trend hin zum Ethernet.

Trends: Real-Time Ethernet

- Es existieren verschiedene Ansätze
 - Beispiel: Ethercat von Beckhoff
 - Die Nachrichten entsprechen dem Standardnachrichtenformat von Ethernet
 - Pakete werden von einem Master initiiert und werden von den Teilnehmern jeweils weitergeleitet.
 - Jeder Knoten entnimmt die für ihn bestimmten Daten und kann eigene Daten anfügen.
 - Die Bearbeitung erfolgt on-the-fly, dadurch kann die Verzögerung minimiert werden.
 - Beispiel: Profinet von Siemens
 - Drei verschiedene Protokollstufen (TCP/IP – Reaktionszeit im Bereich von 100ms, Real-time Protocol - bis 10ms, Isochronous Real-Time - unter 1ms)
 - Profinet IRT benutzt vorher bekannte, reservierte Zeitschlitze zur Übertragung von echtzeitkritischen Daten, in der übrigen Zeit wird das Standard-Ethernet Protokoll ausgeführt

Klausurfragen

- Klausur Wintersemester 07/08 (4 Punkte = 4min)
 - Erläutern Sie kurz die wesentlichen Unterschiede zwischen TokenRing, TokenBus und Ethercat in Bezug auf Topologie und Mediumszugriffverfahren.
- Wiederholungsfragen:
 1. Was ist der Unterschied zwischen dominanten und rezessiven Bits.
 2. Nennen Sie zwei Mechanismen zur Bitsynchronisierung und erklären Sie diese.
 3. Was ist der Unterschied zwischen CSMA/CD und CSMA/CA?
 4. Erläutern Sie zwei verschiedene Ansätze um Ethernet echtzeitfähig zu machen.
 5. Beurteilen Sie die Kommunikationsprotokolle Ethernet, CAN, TTP nach Ihrer Echtzeitfähigkeit und gehen Sie vor allem auf die Möglichkeit zur Vorhersage der maximalen Nachrichtenlatenz ein.



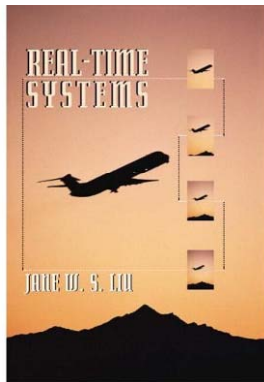
Kapitel 5

Echtzeitbetriebssysteme

Inhalt

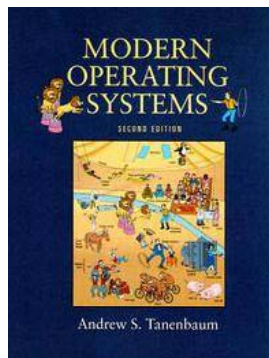
- Grundlagen
- Betrachtung diverser Betriebssysteme:
 - Domänenspezifische Betriebssysteme:
 - OSEK
 - TinyOS
 - Klassische Echtzeitbetriebssysteme
 - QNX
 - VxWorks
 - PikeOS
 - Linux- / Windows-Echtzeitvarianten
 - RTLinux/RTAI
 - Linux Kernel 2.6
 - Windows CE

Literatur



Jane W. S. Liu, Real-Time
Systems, 2000

Dieter Zöbel, Wolfgang Albrecht:
Echtzeitsysteme: Grundlagen und
Techniken, 1995



Andrew S. Tanenbaum: Modern
Operating Systems, 2001

Arnd Heursch et al.: Time-critical tasks in Linux 2.6, 2004

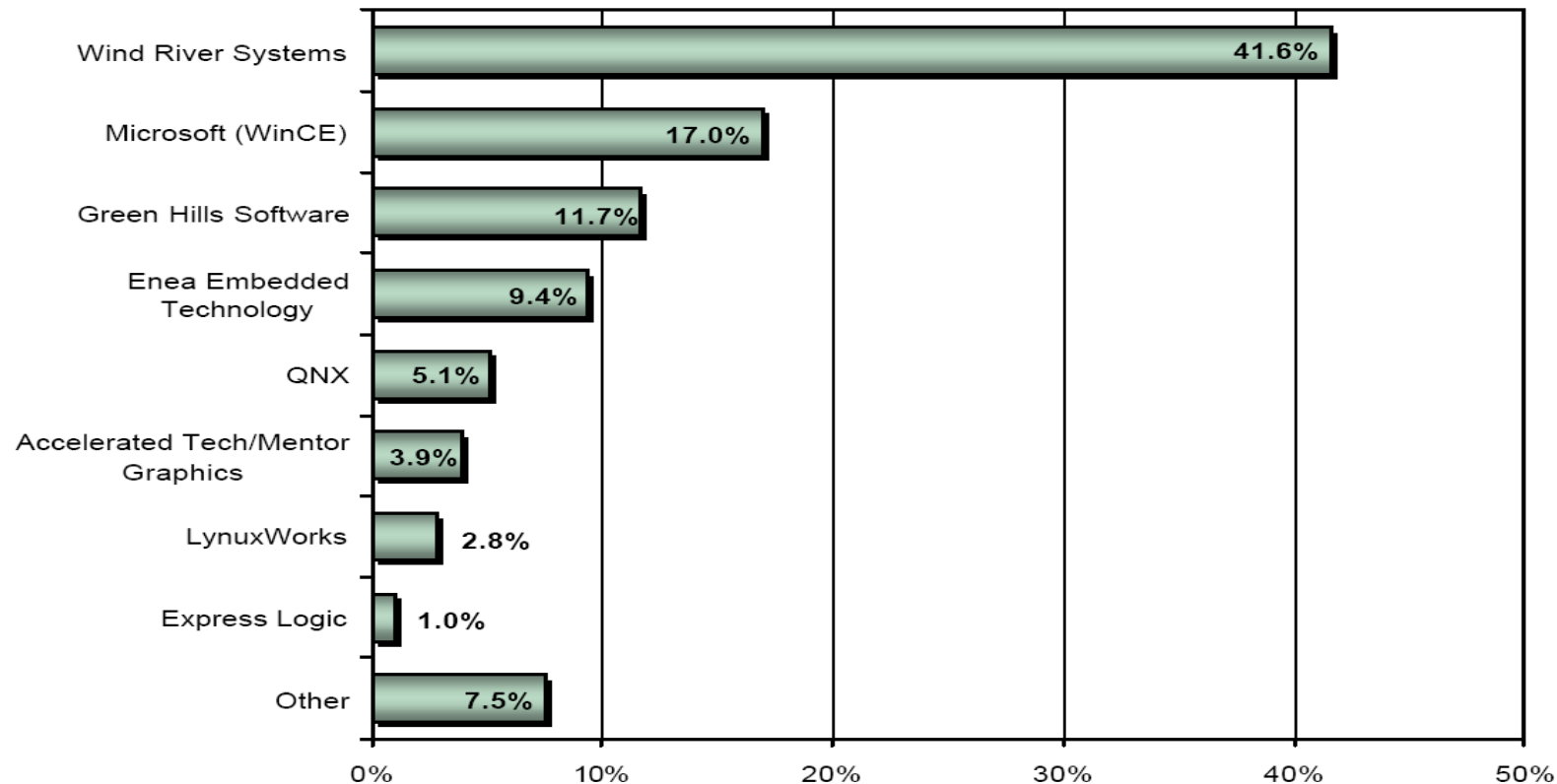
http://inf3-www.informatik.unibw-muenchen.de/research/linux/hannover/automation_conf04.pdf



Interessante Links

- <http://www.mnis.fr/en/support/doc/rtos/>
- <http://aeolean.com/html/RealTimeLinux/RealTimeLinuxReport-2.0.0.pdf>
- <http://www.osek-vdx.org/>
- <http://www.qnx.com/>
- <http://www.windriver.de>
- <http://www.fsmlabs.com>
- <http://www.rtai.org>
- <http://www.tinyos.net/>

Marktaufteilung (Stand 2004)



Marktanteil am Umsatz, Gesamtvolumen 493 Mio. Dollar, Quelle: The Embedded Software Strategic Market Intelligence Program 2005

Anforderungen an Echtzeitbetriebssysteme

- Echtzeitbetriebssysteme unterliegen anderen Anforderungen als Standardbetriebssysteme:
 - stabiler Betrieb rund um die Uhr
 - definierte Reaktionszeiten
 - parallele Prozesse
 - Unterstützung von Mehrprozessorsystemen
 - schneller Prozesswechsel (geringer Prozesskontext)
 - echtzeitfähige Unterbrechensbehandlung
 - echtzeitfähiges Scheduling
 - echtzeitfähige Prozesskommunikation
 - umfangreiche Zeitdienste (absolute, relative Uhren, Weckdienste)
 - einfaches Speichermanagement

Fortsetzung

- Unterstützung bei der Ein- und Ausgabe
 - vielfältigste Peripherie
 - direkter Zugriff auf Hardware-Adressen und -Register durch den Benutzer
 - Treiber in Benutzerprozessen möglichst schnell und einfach zu implementieren
 - dynamisches Binden an den Systemkern
 - direkte Nutzung DMA
 - keine mehrfachen Puffer: direkt vom Benutzerpuffer auf das Gerät
- Einfachste Dateistrukturen
- Protokoll für Feldbus oder LAN-Bus, möglichst hardwareunterstützt
- Aufteilung der Betriebssystemfunktionalität in optionale Komponenten (Skalierbarkeit)



Echtzeitbetriebssysteme

Kriterien zur Beurteilung

Beurteilung von Echtzeitbetriebssystemen

- Folgende Aspekte werden wir genauer betrachten:
 - Schedulingverfahren
 - Prozessmanagement
 - Speicherbedarf (Footprint)
 - Garantierte Reaktionszeiten

Schedulingverfahren

- Fragestellung:
 - Welche Konzepte sind für das Scheduling von Prozessen verfügbar?
 - Gibt es Verfahren für periodische Prozesse?
 - Wie wird dem Problem der Prioritätsinversion begegnet?
 - Wann kann eine Ausführung unterbrochen werden?

Arten von Betriebssystemen

- Betriebssysteme werden in drei Klassen unterteilt:
 - Betriebssysteme mit **kooperativen Scheduling**: es können verschiedene Prozesse parallel ausgeführt werden. Der Dispatcher kann aber einem Prozess den Prozessor nicht entziehen, vielmehr ist das Betriebssystem auf die Kooperation der Prozesse angewiesen (z.B. Windows 95/98/ME)
 - Betriebssysteme mit **präemptiven Scheduling**: einem laufenden Prozess kann der Prozessor entzogen werden, falls sich der Prozess im Userspace befindet. (z.B. Linux, Windows 2000/XP)
 - **Präemptible Betriebssysteme**: der Prozessor kann dem laufenden Prozess jederzeit entzogen werden, auch wenn sich dieser im Kernelkontext ausgeführt wird.

⇒ Echtzeitsysteme müssen präemptibel sein.