Echtzeitsysteme

Kommunikation

26. Januar & 2. Februar 2009

Überblick

Kommunikationslatenzen

Protokolllatenz (engl. *protocol latency*)

Protokolllatenz ist das Zeitintervall einer Nachrichtentransaktion zwischen zwei (entfernten) Netzwerkschnittstellen

- ab Übergabe an der Netzwerkschnittstelle des Sendeknotens
- bis Ablieferung an der Netzwerkschnittstelle des Empfangsknotens

Beständigkeit (engl. *permanence*) der Nachricht an beiden Schnittstellen

- für ein konsistentes Verhalten des verteilten Systems als Ganzes

Latenzschwankung (engl. *latency jitter*)

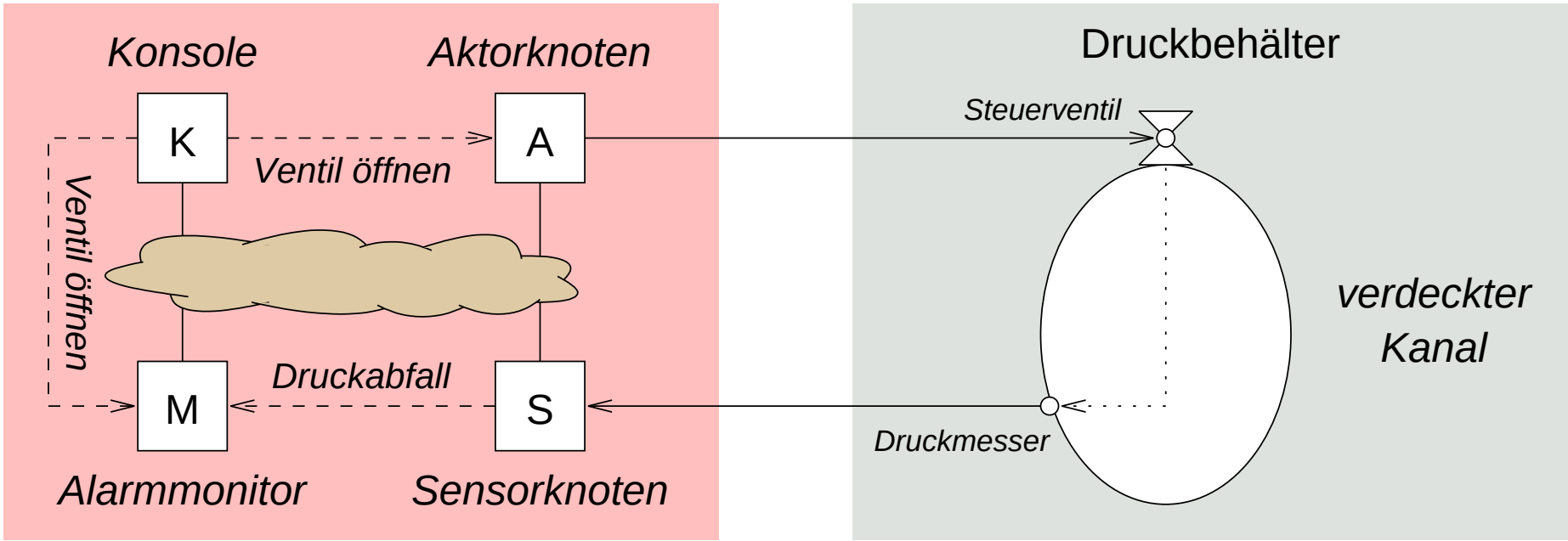
- voraussagbare Latenz (kl. Maximum), minimale Schwankungen

Gruppenruf (engl. *multicast*), keine Punkt-zu-Punkt-Verbindungen

- Simultanzustellung derselben Nachricht an mehrere Empfänger
  - ein Echtzeitabbild „gleichzeitig“ mehreren Tasks anbieten
- kleines und bekanntes Zustellungsintervall (über alle Empfänger)

Beständigkeit von Nachrichten

Relation von empfangenen Nachrichten zu vorausgegangenen gesendeten Nachrichten



M überwacht den Druck im Druckbehälter $\mapsto$ *kontrolliertes Objekt*

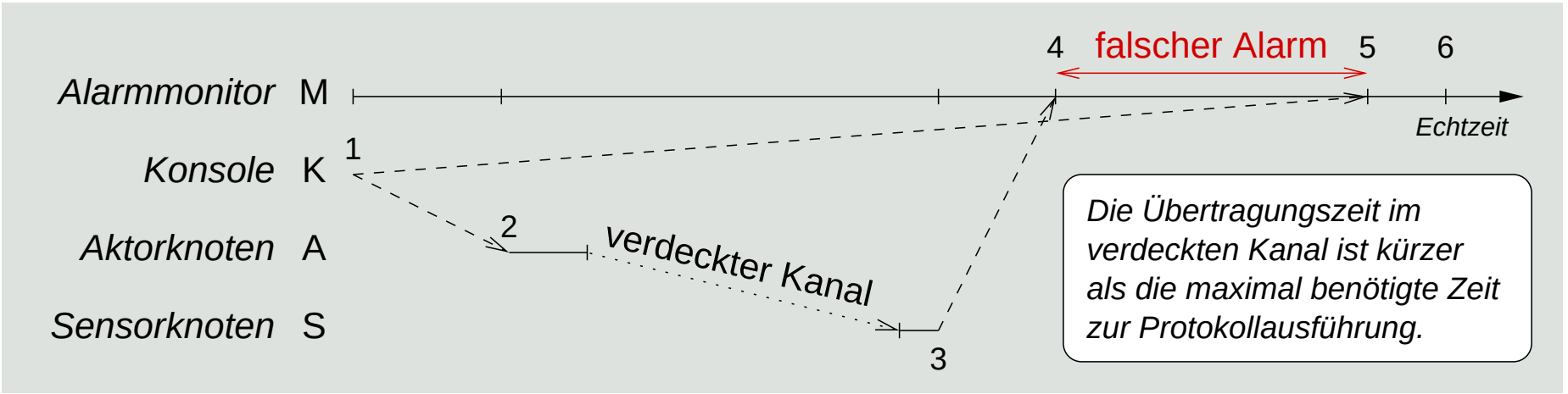
- empfängt eine Nachricht von S, wenn sich der Druck ändert
- schlägt an bei jeder plötzlichen, unersichtlichen Druckänderung

K betätigt den Mechanismus zum Öffnen des Druckventils $\mapsto$ *Operateur*

- sendet eine Nachricht an A zur Betätigung des Steuerventils und an M zur Information, damit dieser keinen Alarm auslöst

Beständigkeit von Nachrichten (Forts.)

Problem verdeckter Kanäle im kontrollierten Objekt



1. Versenden der Anweisung/Information zum Öffnen des Druckventils
2. Empfang der Anweisung durch A, der das Druckventil öffnet
3. Versenden des von S abgelesenen Druckmesswertes an M
4. Empfang des Druckmesswertes durch M $\mapsto$ *Druckabfall* $\rightsquigarrow$ Alarm
5. verspäteter Empfang der Information zum Öffnen des Druckventils
6. eine (Druckmesswert-) Nachricht von S an M hat erst jetzt Bestand

M muss Aktionen verzögern, bis eine Nachricht von S an ihn Bestand hat!

Nachrichten und Aktionen darauf

Aktionsverzögerung bis zur Nachrichtenpermanenz

Aktionsverzögerung (engl. *action delay*)

- ist definiert als Zeitintervall $[t_s, t_p]$ auf der Echtzeitachse:
  - t_s Startzeitpunkt der Übertragung einer bestimmten Nachricht M_i
  - t_p Zeitpunkt, ab dem Nachricht M_i beim Empfänger Bestand hat
- der Empfänger muss jede Aktion zur Nachricht M_i aufschieben...
  - bis die Aktionsverzögerung abgelaufen ist
  - um ein inkorrektes Systemverhalten zu vermeiden

unwiderrufliche Aktion (engl. *irrevocable action*)

- eine Aktion, die nicht mehr rückgängig gemacht werden kann und einen bleibenden Effekt auf die Umgebung verursacht
  - die Betätigung der Schussvorrichtung einer Waffe
  - die Auslösung des Schleudersitzes des Aston Martin DB 5
- solche Aktion darf nur nach Ablauf der Aktionsverzögerung starten

Dauer der Aktionsverzögerung

Schwankungen im Kommunikationssystem und Zeitbewusstsein des Empfängers

Annahme: die Protokolllatenzen sind den Empfängern bekannt, nämlich d_{min} (minimale Verzögerung) und d_{max} (maximale Verzögerung)

- ein außenstehender Betrachter überblickt alle signifikanten Ereignisse

globale Zeit $\mapsto$ Uhrensynchronisation (endliche Genauigkeit)

- Sendezeit t_s global eindeutig und Bestandteil einer Nachricht M_i
  - jede Nachricht wird vom Sender mit einem Zeitstempel versehen
  - der Empfänger muss $d_{max} - t_s$ Zeiten ab Empfang von M_i warten
- M_i hat Bestand zum Zeitpunkt $t_p = t_s + d_{max} + \delta_g$

lokale Zeit $\mapsto$ unterschiedliche Uhrzeiten bei Sender und Empfänger

- Empfänger weiß nicht, wann eine Nachricht M_i gesendet wurde
  - er muss mindestens $d_{max} - d_{min}$ Zeiten ab Empfang von M_i warten
  - auch dann, wenn M_i bereits d_{max} Zeiten unterwegs war
- M_i hat Bestand zum Zeitpunkt $t_p = t_s + 2d_{max} - d_{min} + \delta_l$

☞ δ_g bzw. δ_l bezieht sich auf die Auflösung der Zeitbasis [1, S. 110]

Unterstützung für Zusammensetzbarkeit
Architektonische Gesichtspunkte

Abkapselung des Zeitverhaltens von Knoten (engl. *temporal isolation*)

- Brandmauer (engl. *firewall*) gegen Steuerfehlerausbreitung
  - isolierte Prüfung der zeitl. Randbedingung von Anwendungssoftware
- autonome Kontrolle im Kommunikationssystem
  - von Anwendungssoftware unabhängige Implementierung/Validierung
- zeitgesteuerte Kommunikation an der Netzwerkschnittstelle

Verpflichtungen der Klienten nachkommen $\mapsto$ Schutz des Anbieters

- Überlastung der Anbieter (engl. *server*) vermeiden
  - zuviele oder unkoordinierte Anforderungsnachrichten unterbinden
- Flusskontrolle der Dienstanforderungen der Klienten
  - Klienten helfen, ihre zeitlichen Verpflichtungen erfüllen zu können
- Anbietern ermöglichen, ihre Termine einhalten zu können

Flexibilität

Unterstützung verschiedener Systemkonfigurationen

Evolution von Systemkonfigurationen und den damit verbundenen (statischen) Änderungen im Zeitablauf fördern

- Beispiel: Kundenwünsche am Automobil...
  - Kunde $_A$ $\mapsto$ Schiebedach, Navigationssystem, Sitzheizung
  - Kunde $_B$ $\mapsto$ Klimaanlage, Diebstahlsicherung
  - Kunde $_C$ $\mapsto$ Allradantrieb, Luftfederung, Anhängerkupplung
  - $\vdots$
  - jede Option realisiert durch ein oder mehrere vernetzte Steuergeräte
- ggf. sehr viele (sinnvolle) Kombinationen sind zu ermöglichen...
  - ohne bestehende Knoten/Teilsysteme erneut testen zu müssen
- begrenzender Faktor sollte lediglich die Netzwerkbandbreite sein

...allerdings ist oft auch Flexibilität im dynamischen Sinne gefordert:

- sporadische Nachrichten behandeln, bei minimaler Verzögerung

Fehlererkennung
Kommunikationsfehler

Dienstleistungen müssen vorhersagbar (engl. *predictable*) und verlässlich (engl. *dependable*) erbracht werden

- Fehler bei der Nachrichtenübertragung sind so zu korrigieren, dass eine Erhöhung von Schwankungen in der Protokolllatenz ausbleibt
- dass ein Fehler nicht korrigierbar war, ist (mit minimaler Latenz) allen Kommunikationspartnern mitzuteilen

Erkennung von Nachrichtenverlust beim bzw. durch den Empfänger ist von besonderer Bedeutung

- Beispiel: ein Knoten, an dem ein Steuerventil angeschlossen ist und der Steuerkommandos von anderen Knoten empfängt
  - das Steuerventil muss auch im Fehlerfall korrekt bedient werden
  - Verlust einer Nachricht $\leadsto$ Verlust eines Steuerkommandos
- der Verlust ist mit niedriger Fehlererkennungslatenz anzuzeigen

Fehlererkennung (Forts.)

Ausfallverwaltung (engl. *blackout management*)

EMI (elektromagnetische Interferenz), d.h., Signale oder Emissionen, die im freien Raum abgestrahlt oder entlang von Hochspannungs- und Signalleitungen oder -einrichtungen geführt werden

- kann eine korrelierte Verstümmelung von Nachrichten bewirken
  - Verlust/Verfälschung einer größeren Ansammlung von Nachrichten
  - Beeinträchtigung der Funktion ggf. mehrerer Sender/Empfänger
- ein Ausfall (engl. *blackout*) im Kommunikationssystem ist möglich
  - Ausfallzeiten bewegen sich normalerweise im Millisekundenbreich
  - sie können aber auch von kürzerer Dauer sein...
- das KS muss diesem Phänomen gegenüber resistent sein
  - es muss einen *Blackout* erkennen und überstehen
  - es muss nach einem *Blackout* wie gewohnt weiterarbeiten können

☞ dito. sind Knotenausfälle zu erkennen und dem Ensemble anzuzeigen

Zuverlässigkeit durch verteilte Aufgaben

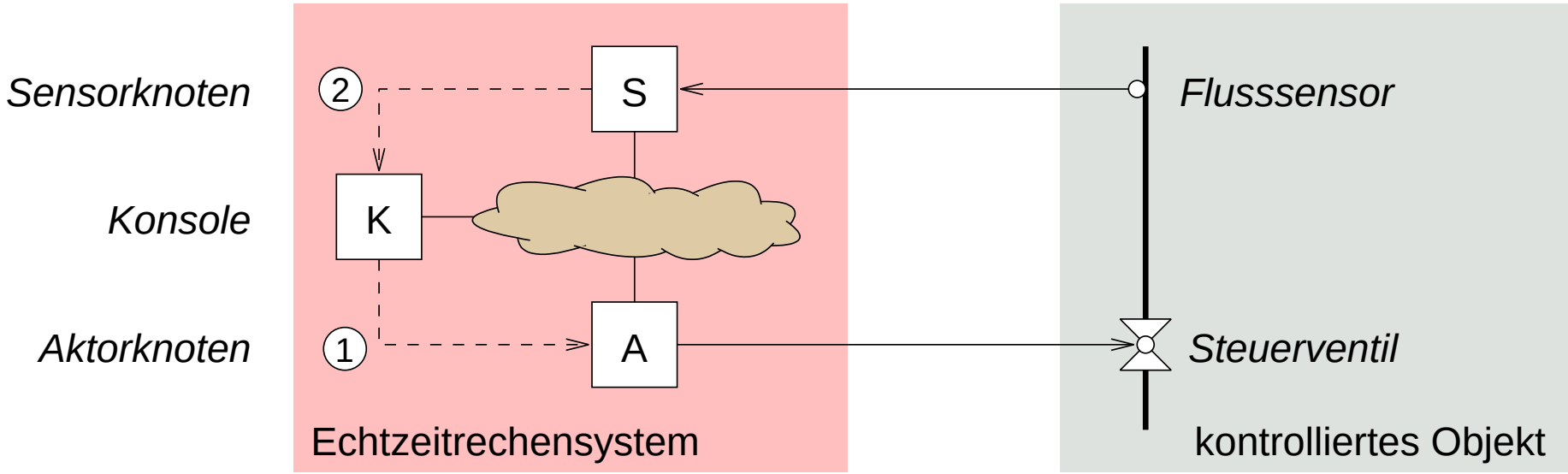
End-zu-End-Bestätigung (engl. *end-to-end acknowledgement*)

Grundregel: traue niemals einem Aktuator (bzw. Aktor $\mapsto$ Stellglied)...

- die Bestätigung einer Steuernachricht sollte von einem Knoten kommen, der diese Nachricht nicht empfangen und verarbeitet hat
  - unabhängiger Sensorknoten überwacht den Effekt des Aktorknotens
  - der Sensor signalisiert dem Auftraggeber Erfolg/Misserfolg der Aktion
- an der Schnittstelle zwischen Echtzeitrechensystem und kontrolliertem Objekt ist ein End-zu-End-Protokoll gefordert
  - bloße Empfangsbestätigung vom Empfängerknoten ist unzureichend
    - sie sagt nur aus, dass eine Nachricht angekommen ist
    - die in der Nachricht kodierte Aktion steht jedoch ggf. noch aus
    - trotz Empfang ist nicht garantiert, dass die Aktion jemals stattfindet
  - eine „Ausführungsbestätigung“ der Steuerungsaktion ist erforderlich
- mit einem Prozedurfernaufruf [2] vergleichbar, der von einen Knoten angenommen und ausgeführt und einem anderen beantwortet wird

End-zu-End-Bestätigung von Steuerungsaktionen

Steuerung und Überwachung durch unabhängige Funktionseinheiten



Der am Flusssensor gemessene, durch den Sensorknoten S zur Konsole K übermittelte Wert ② ist die End-zu-End-Bestätigung der zuvor von K über den Aktorknoten A an das Steuerventil versandten Nachricht ①.

Three Mile Island

28. März 1979, Kernschmelze im Reaktorblock 2 ([1, S. 148] und [3])

Perhaps the single most important and damaging failure in the relative long chain of failures during this accident was that of the Pressure Operated Relief Valve (PORV) on the pressurizer. The PORV did not close; yet its monitoring light was signaling green (meaning closed).

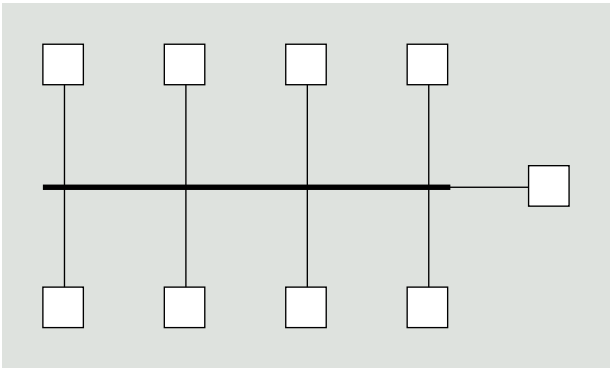
Annahme $\mapsto$ Eingang der Empfangsbestätigung des Steuersignals zum Schließen des Ventils impliziert, dass das Ventil geschlossen wurde

- eine elektromechanische Störung im Ventil führte jedoch dazu, dass diese Implikaton nicht unter allen Bedingungen galt
- (mechanische) Überprüfung der Funktion war nicht vorgesehen

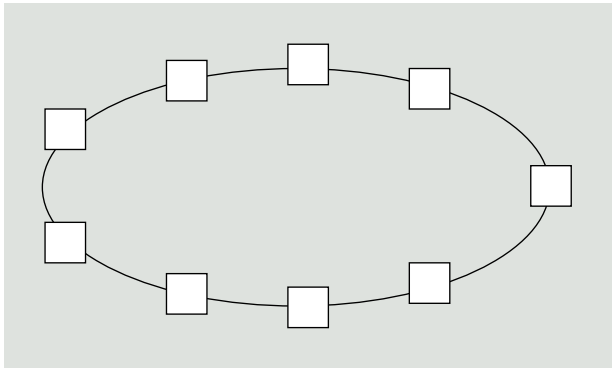
Vertrauen ist gut, Kontrolle ist besser.

Netzwerktopologie

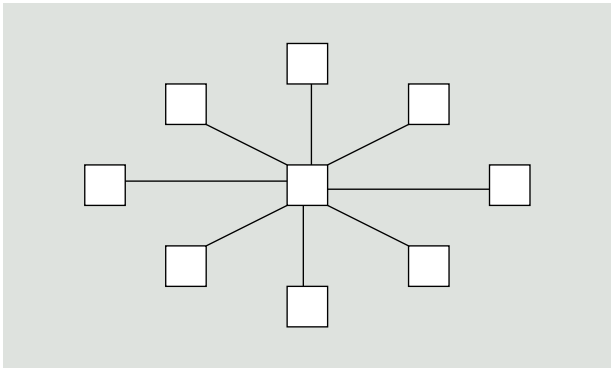
Physikalische Struktur des Kommunikationssystems



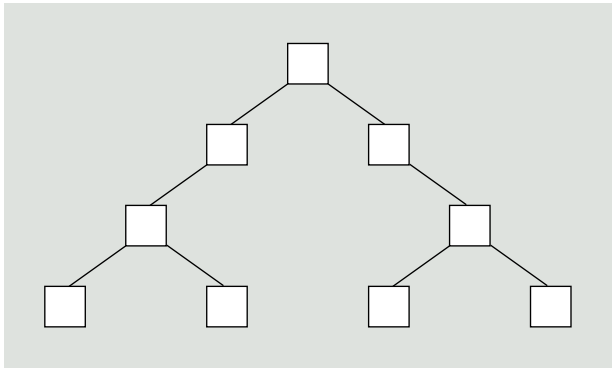
Bus (engl. *bus*), passive Topologie
zentrales Medium



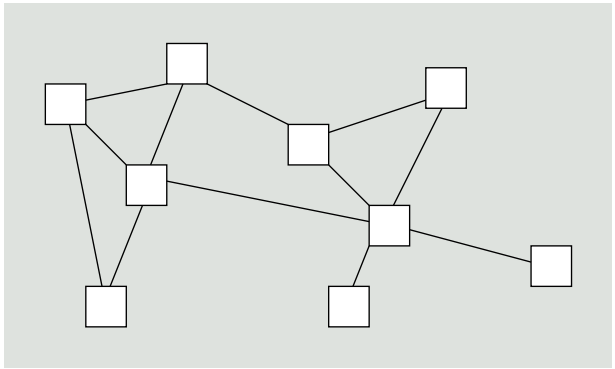
Ring (engl. *ring*), aktive Topologie
verteilte Steuerung



Stern (engl. *star*)
zentrale Verteilung



Baum (engl. *tree*)
dezentrale Steuerung



Masche (engl. *mesh*)
$\rightsquigarrow$ Internet

Netzwerktopologie (Forts.)

Bus *versus* Ring — in Echtzeitkommunikationssystemen verbreitete Techniken

Bus $\mapsto$ ein Hauptkabel, an dem alle Teilnehmer über spezielle T-Stücke (z.B. BNC, engl. *bayonet navy connectory*) angeschlossen sind

- Teilnehmer hören „gleichzeitig“, was auf dem Bus geschieht
  - Simultanzustellung von Nachrichten $\rightsquigarrow$ *Multicast*
- Zugriffsverfahren koordinieren „gleichzeitige“ Sendevorgänge
  - Bus ist gemeinsames und teil- bzw. unteilbares Betriebsmittel
- unabhängig von Knotenfunktionen; Problem: Kabelbruch

Ring $\mapsto$ Zweipunktverbindungen zwischen zwei Teilnehmern

- Nachrichten werden bis zum Bestimmungsort weitergeleitet
  - jeder Teilnehmer agiert als Zwischenverstärker (engl. *repeater*)
- Adressierung sorgt für überschneidungsfreies Senden
  - der Ring ist ein abschnittweise teilbares Betriebsmittel
- abhängig von Knotenfunktionen; Problem: Teilnehmerausfall
  - Fehlerfall: ggf. Umschaltung der Drehrichtung des Arbeitswegs

Physikalische Abschottung

Knoten als kleinste ersetzbare Einheit (engl. *smallest replaceable unit*, SRU)

Fehlertoleranz durch aktive Redundanz bei der eine Gruppe von SRUs eine fehlertolerante Einheit (engl. *fault-tolerant unit*, FTU) bildet

- bes. kritisch ist ein Gleichtaktfehler (engl. *common-mode failure*)
  - bei Systemen, deren Gerätschaften nicht redundant ausgelegt sind[1]
  - wenn kein korrektes Ergebnis mehr abgeleitet werden kann
  - hervorgerufen durch ggf. nur ein einzelnes physikalisches Ereignis
- SRUs derselben FTU sind physikalisch getrennt anzuordnen
  - z.B.: elektr[on]ische Lenkung (engl. *steer-by-wire*) beim Automobil
    - SRUs dieser sicherheitskritischen Funktion müssen eine FTU bilden
    - unfallbedingte Schäden dürfen keinen Systemausfall hervorrufen
    - die SRUs dieser FTU müssen auf verschiedene Einbauorte verteilt sein
  - das Kommunikationssystem muss spontan Ausweichrouten anbieten
- Systemrekonfigurierung ist eine aperiodische/sporadische Aufgabe

[1] Besonders heimtückische Variante: wenn redundante Kopien desselben (Software-) Prozesses unter identischen Bedingungen scheitern $\rightsquigarrow$ *multi-version programming*.

Regelung des Datenflusses

Aufgabe der Netzwerkschicht des ISO OSI Referenzmodells [4]

Steuerung der Geschwindigkeit des Informationsflusses zwischen Sender Empfänger, so dass der Empfänger mit dem Sender Schritt halten kann

- Sender werden veranlasst, nur soviele Nachrichten zu übertragen, wie der Empfänger auch aufnehmen kann
- Empfänger bestimmen maximale Kommunikationsgeschwindigkeit

Zweck der Maßnahme ist es, eine Überschreitung der Aufnahmekapazität des Empfängers zu vermeiden und diesen nicht zu überlasten

- ereignisgesteuerte Systeme sind besonders von Überlast bedroht:
  - Nachrichtenversand/-empfang verursacht Unterbrechungen
  - Pufferplatz für zu sendende/empfangende Nachrichten ist begrenzt
  - Nachrichten werden von einzuplanenden/-lastenden Jobs verarbeitet
- die Steuerung des Informationsflusses geschieht explizit oder implizit

Explizite Flusskontrolle

Voraussetzung — die jedoch oft übersehen wird — ist, dass sich ein Sender im Kontrollbereich eines Empfängers befindet

- ein Empfänger kann Gegendruck (engl. *back pressure*) auf den Sender ausüben, indem er die Übertragungsrate kontrolliert
  - Flusskontrolle durch Gegendruck (engl. *back-pressure flow control*)
- der Gegendruck des Empfängers äußerst sich dadurch, dass beim Sender die Übertragung weiterer Daten hinausgezögert wird
  - empfangene Nachrichten werden ohne weitere Behandlung verworfen
  - Empfangsbestätigungen werden bewusst und gezielt zurückgehalten
- das weitere Vorankommen des Senders hängt ab vom Zustand und vom Verhalten des Empfängers

Protokolle mit 1-zu-1-Synchronisation zwischen Sender und Empfänger bilden die Grundlage für Ereignisnachrichten

- Maximierung der Bandbreitenausnutzung ist nebensächlich (in EZS)

Explizite Flusskontrolle (Forts.)

Bedeutung (für Echtzeitsysteme) haben Protokolle, die nach dem Schema „sende und warte“ (engl. *send and wait*, auch *stop and wait*) arbeiten

PAR (engl. *positive acknowledgement and retransmission*)

senderseitige Schritte $\rightsquigarrow$ Fehlermaskierung

- die Quelle sendet ein Paket, startet einen Zeitgeber und erwartet eine Empfangsbestätigung, bevor ein neues Paket gesendet wird
- bleibt die Empfangsbestätigung aus, läuft der Zeitgeber ab und das Paket wird wiederholt gesendet
- ist die maximale Anzahl von Wiederholungen (desselben Pakets) erreicht, wird der Sendevorgang abgebrochen $\mapsto$ *Exception*

empfangsseitige Schritte $\rightsquigarrow$ Duplikatunterdrückung

- nimmt die Senke ein eingetroffenes Paket an, sendet sie eine Empfangsbestätigung an die Quelle zurück
- gleicht die Laufnummer des Pakets der des von derselben Quelle zuletzt empfangenen Pakets, wird das Paket verworfen

Explizite Flusskontrolle (Forts.)

Aktionsverzögerung auf Basis von PAR

Beispiel *Token*-Bus: die maximale *Token*-Umlaufzeit sei 10 ms und die Transportzeit für eine Nachricht liegt bei 1 ms

(a) System mit lokaler Zeit, Uhrauflösung ist vernachlässigbar ($\delta_l = 0$)

- Zeitabschaltung (engl. *timeout*) ist auf mindestens 22 ms zu setzen
  - *Token*-Verlust $\rightsquigarrow$ einmalige Umschaltung der Drehrichtung
  - die Umschaltung erfolgt nach spätestens 11 ms
- minimale Verzögerung $d_{min} = 1\,\text{ms}$
  - der Sender hat das *Token*, wenn eine Nachricht bereitgestellt wird
- maximale Verzögerung $d_{max} = 55\,\text{ms}$
  - Abbruch nach maximal drei Wiederholungen (66 ms)
  - dritte Wiederholung, *Token* gerade vorbei: $2 \times 22\,\text{ms} + 10\,\text{ms} + 1\,\text{ms}$
- Aktionsverzögerung $2d_{max} - d_{min} = 109\,\text{ms}$
  - die meisten Nachrichten werden in einer Runde (11 ms) empfangen
  - sie sind nach Empfang 98 ms zurückzuhalten, um Bestand zu haben

(b) System mit globaler Zeit, die Auflösung der Uhr sei $g = 100\,\mu\text{s}$

- Aktionsverzögerung $d_{max} + \delta_g = d_{max} + 2g = 55.2\,\text{ms}$

Explizite Flusskontrolle (Forts.)

Gefahr vor Überlast durch „Flattern“ (engl. *thrashing*)

Wiederholungen von Nachrichten bei Zeitüberschreitung (engl. *timeout*)

- Ursache kann sein, dass das Kommunikationssystem die gegebene Last kaum noch bzw. nicht mehr bewältigen kann
  - anfällig sind Systeme, deren Normallast nahe der Maximallast liegt
  - Wiederholungen wegen Übertragungsfehler sind dann bes. kritisch
  - ein abrupter Leistungsabfall (Durchsatz) kann die Folge sein
- Überlast erhöht das Risiko von Zeitüberschreitungen, woraufhin zusätzliche Last anfällt...
  - die die bereits vorhandene Überlast weiter ansteigen lässt
  - die Zeitüberschreitungen dadurch noch wahrscheinlicher macht
- ebenso abrupt, wie die Überlastsituation aufgetreten ist, wird sie auch wieder verschwinden
  - ggf. muss nur eine einzige Kommunikation erfolgreich abschließen

Thrashing ist unbedingt zu vermeiden ($\mapsto$ *rare-event situation*) und d.h.:

(a) kontinuierliche Überwachung der Betriebsmittelanforderungen

(b) Flusskontrolle durch Gegendruck, bei beobachtetem Leistungsabfall

Implizite Flusskontrolle

Voraussetzung ist globale Zeit (engl. *global time*)

Sender und Empfänger treffen vorher (z.B. beim Systemstart) eine Übereinkunft über die Sendezeitpunkte von Nachrichten

- der Sender verpflichtet sich, Nachrichten nur zu den vereinbarten Zeitpunkten zum Empfänger zu versenden
- der Empfänger verpflichtet sich, alle Nachrichten des Senders zu empfangen, solange dieser seine Verpflichtung einhält

unidirektionale Kommunikation $\mapsto$ Bestätigungen für eingetroffene Nachrichten entfallen, Fehlererkennung ist Aufgabe des Empfängers:

- er weiß, wann eine erwartete Nachricht nicht mehr eintreffen kann
- globale Zeit erlaubt ihm, den Zeitpunkt auf $t_s + d_{max}$ festzulegen
  - für jeden ihn betreffenden Sendezeitpunkt t_s

Fehlertoleranz (aktive Redundanz) durch *Multicast* ist gut umsetzbar

- gleichzeitige Übertragung von k Kopien derselben Nachricht
- bevorzugt über mehrere Kanäle, soweit verfügbar und möglich

Gegenüberstellung

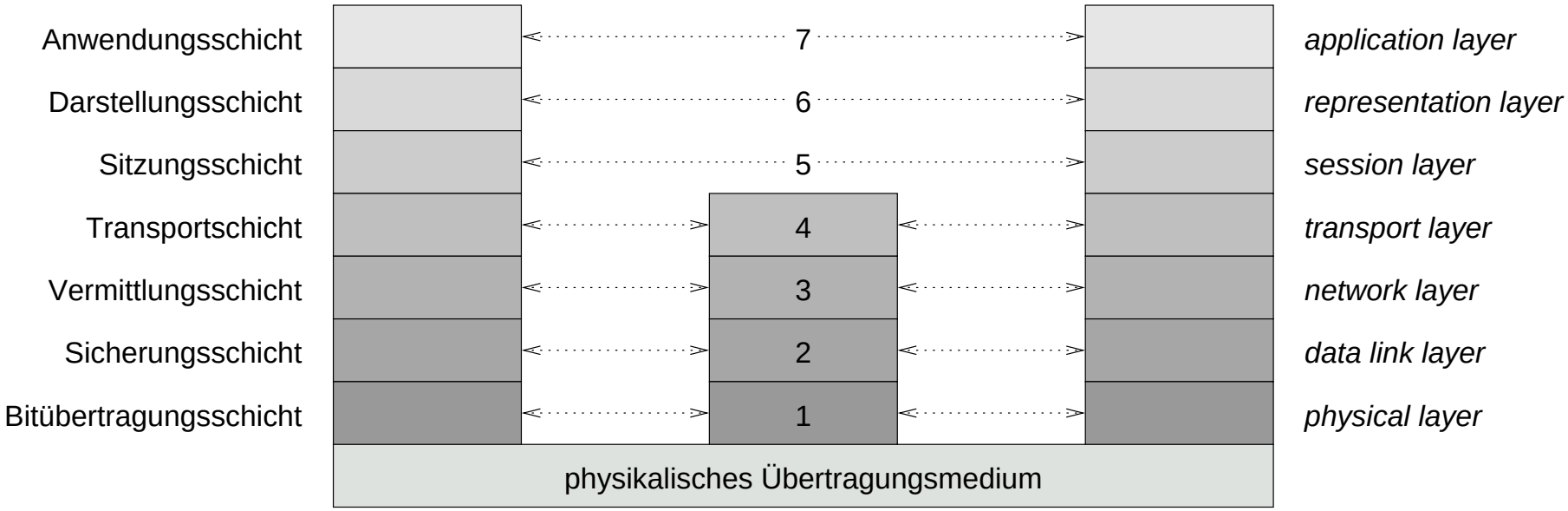
Vor dem Hintergrund *Hard Real-Time System* (HRTS, [1, S. 153])

Charakteristik	Flusskontrolle		HRTS
	explizit	implizit	
Steuersignal	Der Empfänger muss in der Lage sein, die Sendeereignisse des Senders steuern zu können.	Die Signale werden bei Fortschreiten der Echtzeit mit konstanter Rate generiert.	Der Empfänger kann die Ereignisse im Kontrollbereich des Senders nicht völlig kontrollieren.
Fehlererkennung	Sender	Empfänger	Empfänger
Thrashing	anfällig	nicht anfällig	ist zu vermeiden
Multicast	schwer	einfach	gefordert

Flusskontrolle macht die Prozessschnittstelle zwischen kontrolliertem Objekt und Echtzeitrechensystem besonders kritisch

- nicht alle Ereignisse, die im kontrollierten Objekt anfallen, werden im Kontrollbereich des Echtzeitrechensystems liegen
- ein Alarmschauer kann die Folge sein, wenn mehr Ereignisse im kontrollierten Objekt anfallen, als im Entwurf angenommen wurde

Schichtenmodell
ISO OSI Referenzmodell [4]



1 – 4 $\mapsto$ transportorientierte Schichten

- Nachrichten segmentiert, in eingerahmten Paketen übertragen

5 – 7 $\mapsto$ anwendungsorientierte Schichten

- Daten (kanonisch) über logische Verbindungen austauschen

Schichtenmodell (Forts.)

ISO OSI Referenzmodell „*considered harmful*“

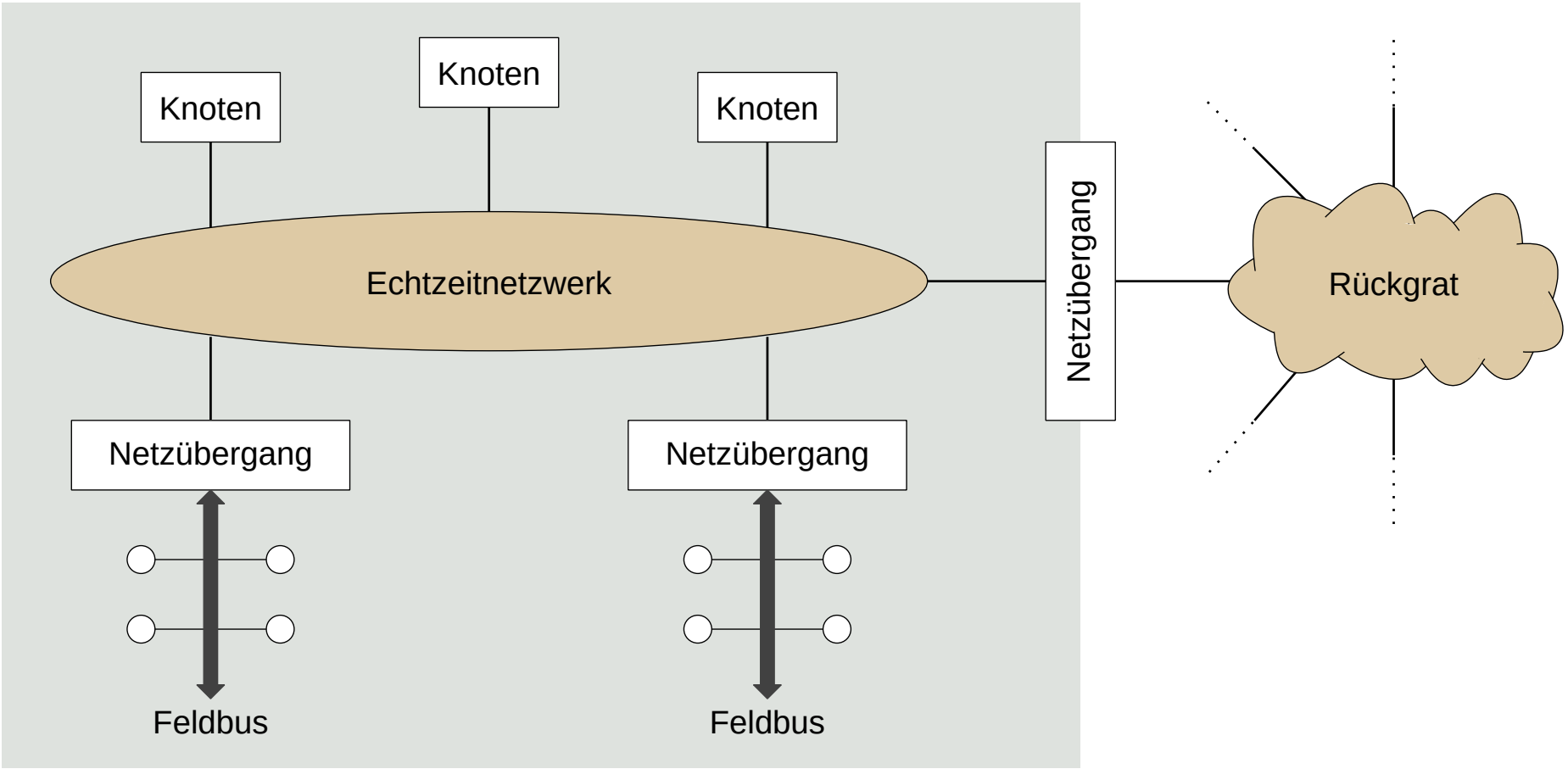
OSI (engl. *open systems interconnection*, 1982) $\mapsto$ Referenzarchitektur zur herstellerunabhänigigen Vernetzung von Rechensystemen

- oft jedoch (falsch) verstanden als Implementierungsarchitektur
  - ein Stapel PAR-ähnlicher Protokolle ist verschachtelt abzuarbeiten, um Daten auf Anwendungsebene auszutauschen
  - Folge sind hohe Latenzschwankungen und geringe Dateneffizienz
- hinter vielen OSI-Protokollen verbergen sich folgende Annahmen:
  - Kommunikation verläuft über Punkt-zu-Punkt-Verbindungen
  - Nachrichten sind ereignisgesteuert
  - Protokolle arbeiten nach PAR und mit expliziter Flusskontrolle
  - Echtzeitperformanz ist keine zentrale Frage
- Annahmen, die nicht mit Anforderungen von HRTS vereinbar sind

☞ quer zu den Schichten ausgerichteter Entwurf (*cross-layer design*)

Netzwerkföderation

Organisation von Echtzeitnetzen [1, S. 156]



Netzwerkföderation (Forts.)

Typen von Echtzeitnetzen

Echtzeitnetzwerk (engl. *real-time network*) Kern einer Gerätegruppe

- zuverlässige und zeitlich vorhersagbare Nachrichtenübertragung
  - insb. periodische Zustandsnachrichten mit impliziter Flusskontrolle
- Unterstützung für Fehlertoleranz: replizierte Knoten/Kanäle
  - Mitgliedsdienst (engl. *membership service*), Knotenausfallerkennung
- Uhrensynchronisation mit Auflösung im Mikrosekundenbereich

Feldbus (engl. *field bus*) Anschluss von Sensoren und Aktoren

- Netz von Mikrocontrollern (μC $\mapsto$ Sensor und/oder Aktor)
- periodisch übertragene, kurze Nachrichten mit Zustandsdaten
- strikte Echtzeitanforderungen an Latenz und Latenzschwankungen
  - zieht meist präzise Uhrensynchronisation auf Busebene nach sich

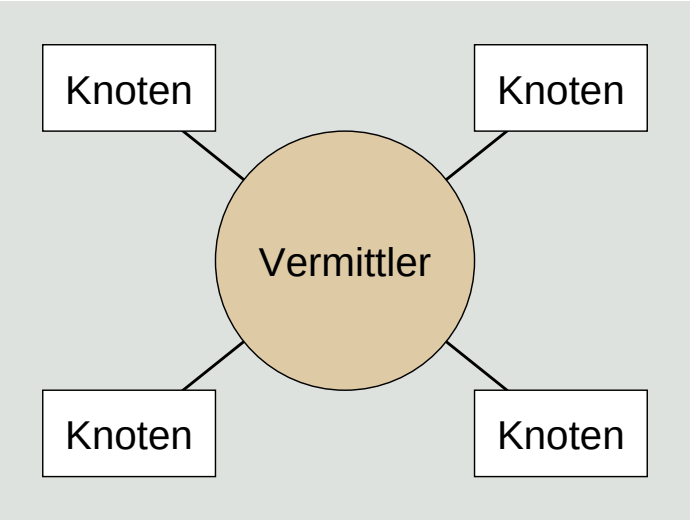
Rückgrat (engl. *backbone network*) Verbindung zur „Außenwelt“

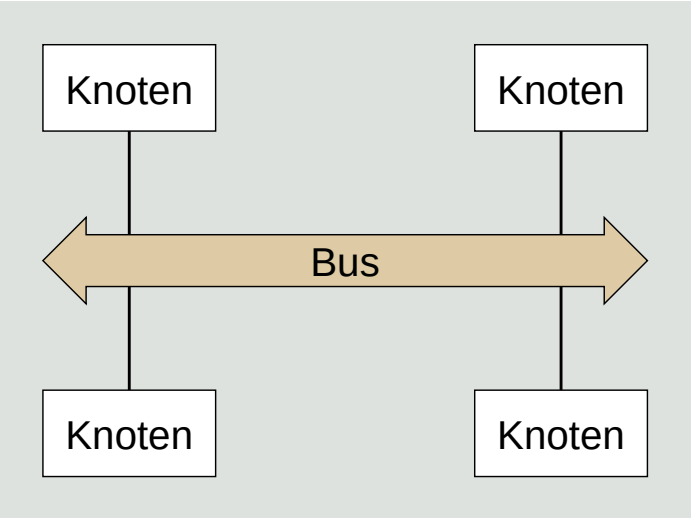
- Austausch zeitunkritischer Daten mit anderen Rechensystemen

Kommunikationsmedium

Wiederverwendbares Betriebsmittel

Kommunikation zwischen Knoten ist nur über ein ihnen gemeinsames Betriebsmittel möglich, das den Austausch von Informationen erlaubt





Vermittler $\mapsto$ Rechensystem

- Punkt-zu-Punkt-Verbindung
- *shared-memory* Prozessor

Bus $\mapsto$ Leitungssystem

- Zugriffskontrolle
- *shared media*

Kommunikationskanal

Wiederverwendbares unteilbares Betriebsmittel

Zugriffskontrolle für die exklusive Vergabe eines Übertragungskanals leistet das Netzzugangsprotokoll (engl. *media-access protocol*)

- wichtige Charakteristiken des Übertragungskanals dabei sind:
  - Bandbreite
  - Ausbreitungsverzögerung
  - Bitlänge
- als Verfahren kommen allgemein zum Einsatz:
  - CSMA (CD/CA)
  - *Token*
  - *Minislotting*
  - *Master/Slave*
  - TDMA

Kollisionen durch gleichzeitiges Senden mindern allg. den Durchsatz und machen insbesondere Kommunikation nicht deterministisch

- dadurch ggf. entstehende Latenz(schwankung)en sind zu begrenzen

Übertragungskanal
Datenübertragungsrate

Bandbreite (engl. *bandwidth*)

- gibt an, innerhalb welcher Frequenzspanne Informationseinheiten (z.B. Bits) parallel übertragen werden können
  - Bandbreite ist nicht gleich Datenrate (Shannon-Harley-Gesetz)
  - neben der binären ist z.B. die ternäre Kodierung möglich

  Datenrate (engl. *data rate*) bezeichnet die Anzahl von Bits, die den Kanal pro Zeiteinheit durchlaufen können
- ist abhängig von...
  - physikalische Eigenschaften des Kanals
  - Beeinflussung durch die Umgebung
- Beispiel: Beschränkungen im Autmobil wegen EMI

  10 Kbit/s einzelner Draht (engl. *single wire*)
  1 Mbit/s nicht abgeschirmter verdrillter Draht (engl. *twisted pair*)

Übertragungskanal (Forts.)

Signallaufzeit und Speicherkapazität

Ausbreitungsverzögerung (engl. *propagation delay*)

- Zeitintervall, das ein Bit zum Durchlaufen des Kanals benötigt
- abhängig von Kanallänge und Wellengeschwindigkeit im Kanal:
  $\approx 3 \cdot 10^8\,\mathrm{m/s}$ Licht im Vakuum
  $\approx 2 \cdot 10^8\,\mathrm{m/s}$ Licht im Kabel $\approx 2/3$ Licht im Vakuum
- die Bitgeschwindigkeit liegt bei $\approx 200\,\mathrm{m}/\mu\mathrm{s}$
  - m.a.W.: das Signal benötigt $\approx 5\,\mu\mathrm{s}$ über ein Kabel von 1 km Länge

Bitlänge (engl. *bit length*)

- Anzahl von Bits, die den Übertragungskanal in der durch die Ausbreitungsverzögerung def. Zeitspanne durchlaufen können
- Beispiel: Bandbreite $= 100\,\mathrm{Mbit/s}$, Kanallänge $= 200\,\mathrm{m}$
  - $10^8/(2 \cdot 10^8) \cdot 200 = 100$

Übertragungskanal (Forts.)
Protokolleffizienz (von Bussen)

Dateneffizienz (engl. *data efficiency*)

- zwischen jeweils zwei aufeinanderfolgenden Nachrichten ist ein Mindestabstand einzuhalten $\mapsto$ Kollisionsvermeidung
  - gleicht dem Zeitintervall von mind. einer Ausbreitungsverzögerung
- Konsequenz daraus ist, dass die Dateneffizienz des Kanals (Busses) duch das Zugriffsprotokoll (zusätzlich) begrenzt ist
  - sei l_{msg} die Nachrichtenlänge und l_{bit} die Bitlänge
  - dann gilt als obere Grenze: Dateneffizienz $< l_{msg}/(l_{msg} + l_{bit})$
- z.B.: Bandbreite $= 100\,\text{Mbit/s}$, Kanallänge $= 1\,\text{km}$, $l_{msg} = 100$
  - $l_{bit} \approx 10^8/(2 \cdot 10^8) \cdot 100 = 500$
  - Dateneffizienz $< 100/(100 + 500) = 0.1\overline{6} < 16.7\,\%$

CSMA
(engl. *carrier sense multiple access*)

Klasse von verteilt arbeitenden Zugriffsverfahren, die keine zentrale Kontrolle (beim Schreiben) erfordern

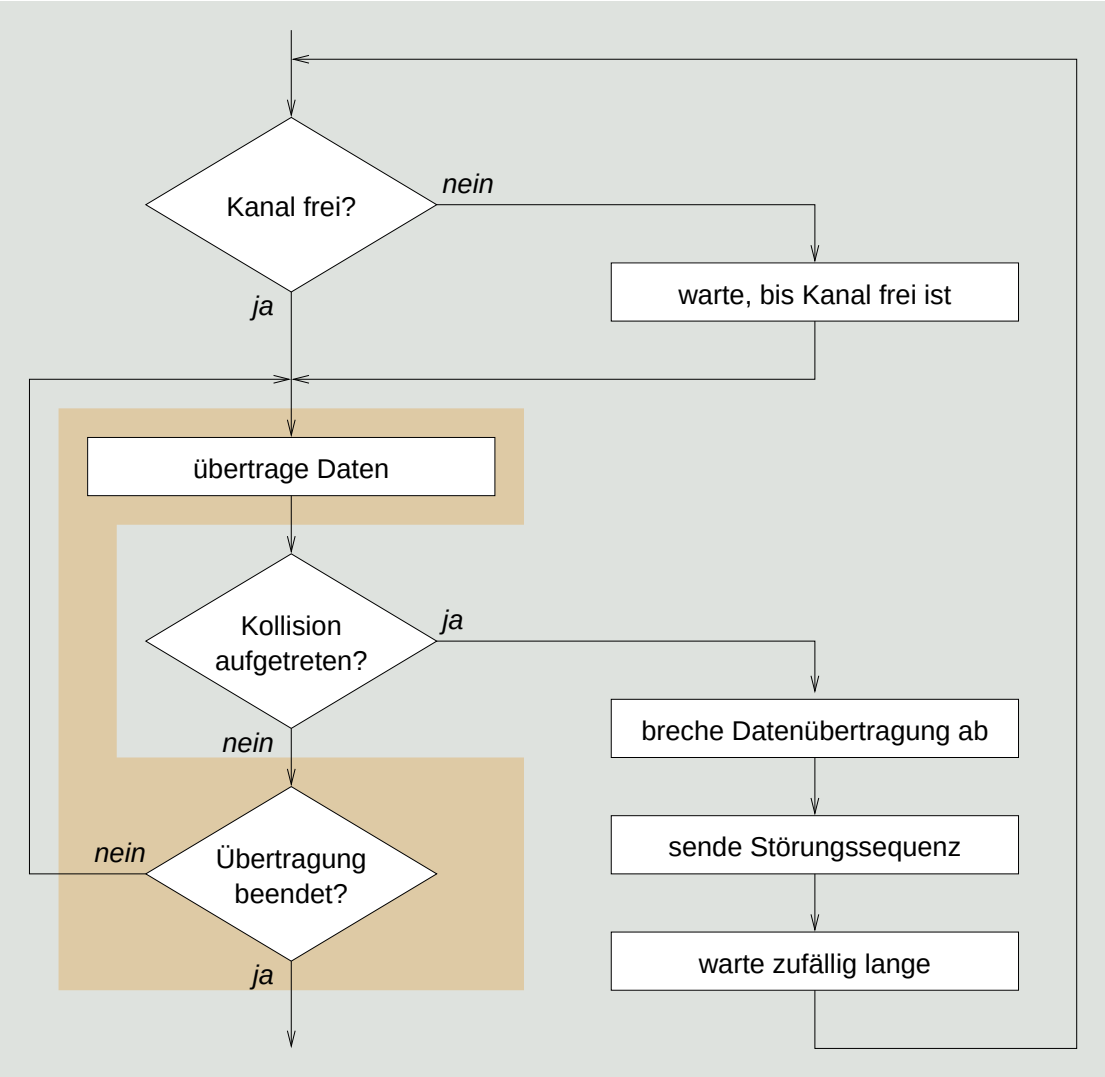
- jeder Teilnehmer hört beim Schreiben gleichzeitig den Bus auf kollidierende Schreibzugriffe ab $\mapsto$ *carrier sense*

CD (engl. *collision detection*)
- im Konfliktfall werden die Schreibzugriffe zurückgenommen
- jeder betroffene Teilnehmer wartet zufällig lang
- nach den Wartephasen werden die Schreibzugriffe wiederholt

CA (engl. *collision avoidance*) z.B. durch Bitarbitrierung
- Nachrichten gehen eindeutige Identifikationen/Prioritäten voran
- im Konfliktfall geben „Verlierer“ ihre Schreibzugriffe auf
- nach endlich vielen Schritten/Takten bleibt ein Gewinner übrig

CSMA/CD

Standardprotokoll für Halbduplexbetrieb



Ethernet (Xerox [5])

- lokales Netzwerk
  - quittierungsfrei
- Varianten:
  - *thick/thin*
  - *fast/gigabit*
- Probleme:
  - Wartezeiten
  - Fairness

CSMA/CD (Forts.)

Verringerung der Wahrscheinlichkeit von Kollisionen

LON (engl. *local operating network*, [6]) Gebäudeautomatisierung

- Busteilnehmer (Knoten): Neuron-Chip mit drei 8-Bit Prozessoren

  network CPU $\mapsto$ Schichten 2–6 des OSI-Referenzmodells

  media-access CPU $\mapsto$ Netzzugangsprotokoll (Schicht 1)

  application CPU $\mapsto$ Anwendung, Knotenfunktion
- Knoten greifen nach zufällig langer Verzögerung auf den Bus zu
  - am Anfang von (regulären) Übertragungen
  - während Übertragungswiederholungen als Folge von Kollisionen
  - nach Rücknahme des Trägers (engl. *carrier*) des voherigen Transfers
- die Größe des „Wartefensters“ ist eine Funktion der Kanallast
  - speziell ausgelegt, um die Wahrscheinlichkeit von Kollisionen bei hoher Last zu minimieren $\mapsto$ p-persistentes CSMA
  - Flusskontrolle kommt durch stochastischen Gegendruck zustande (engl. *stochastic back-pressure flow control*)

CSMA/CA

Kollisionensvermeidung durch Arbitrierung: Bitarbitrierung

CAN (engl. *control area network*, [7]) Automobilindustrie

- jeder gesendeten Nachrichten geht ein Nachrichtentyp voran:
  - beschreibt den Inhalt einer Nachricht, verwendet zur Arbitrierung
  - je kleiner der Wert des Typs, desto höher die Priorität der Nachricht
- die Übertragung beginnt mit dem höchstwertigsten Bit des Typs
  - sendet ein Knoten eine 1 (rezessives Bit), empfängt er aber eine 0 (dominantes Bit), so bricht er seine Übertragung ab
  - erneuter Sendeversuch, nachdem Busruhe erkannt wurde: 11 Bitzeiten auf Ruhepotential (rezessiver Buspegel)
- Beispiel: vier Knoten senden gleichzeitig...

Nachrichtentyp 1431 $\mapsto$ ~~1 0 1 1 0 0 1 0 1 1 1~~
Nachrichtentyp 1337 $\mapsto$ ~~1 0 1 0 0 1 1 1 0 0 1~~
Nachrichtentyp 1335 $\mapsto$ ~~1 0 1 0 0 1 1 0 1 1 1~~
Nachrichtentyp 1332 $\mapsto$ 1 0 1 0 0 1 1 0 1 0 0
Bus 1 0 1 0 0 1 1 0 1 0 0 $\rightsquigarrow$ 1332 hat Vorrang

Token

Vergabe von Übertragungsrechten

Knoten im Besitz eines Übertragungsrechtes ($\mapsto$ *Token*) können den Bus zum Schreiben verwenden

- das Bussystem ist als Ring ausgelegt (engl. *token-ring bus*)
  - das *Token* umläuft den Ring von Knoten zu Knoten: aktive Topologie
- das Antwortverhalten bestimmt sich durch zwei Zeitparameter:
  THT (engl. *token-hold time*)
    - längste Zeit, für die ein Knoten den *Token* halten darf
  TRT (engl. *token-rotation time*)
    - längste Zeit, die ein *Token* zum kompletten Umlauf benötigt
- ernstes Problem: Ausfall des Knotens, der den *Token* besitzt
  - Verlust des *Token* wird durch Zeitüberschreitung festgestellt
  - einer der anderen Knoten (welcher?) erzeugt einen neuen *Token*

Profibus (engl. *process field bus*, [8]) Prozessautomatisierung

Minislotting

Zeitkontrolliertes Zugangsverfahren

Zeit wird in eine Folge von „Minischlitzen“ (engl. *mini slot*) unterteilt, jeder länger als die Ausbreitungsverzögerung des Kanals

- jedem Knoten ist eine eindeutige Anzahl von *Minislots* zugeordnet
  - zu verstreichende Zeit der Ruhe (auf dem Kanal) vor dem Schreiben

ARINC 629 (*Aeronautical Radio Incorporated*, [9]) Flugzeugindustrie

- ein Warteraumprotokoll ähnlich zum „Bäckereialgorithmus“ [10]
  - in einem ersten Zeitintervall finden sich Prozesse, die schreiben wollen, in einen (verteilten) Warteraum ein
  - im nachfolgenden Zeitintervall ($\mapsto$ Epoche) können alle wartenden Prozesse schreiben, bevor neue den Warteraum betreten dürfen
- kein (böswilliger) Knoten kann den Bus monopolisieren
  - Busschutz (engl. *bus protection*)

Minislotting (Forts.)

ARINC 629 — Zeitparameter zur Kontrolle des Zugangs zum Übertragungsmedium

SG (engl. *synchronization gap*)

- kontrolliert Zutritt zum Warteraum: identisch für alle Knoten

TG (engl. *termination gap*)

- kontrolliert den Buszugriff: unterschiedlich für jeden Knoten

TI (engl. *transmit interval*)

- verhindert Monopolisierung des Busses: identisch für alle Knoten

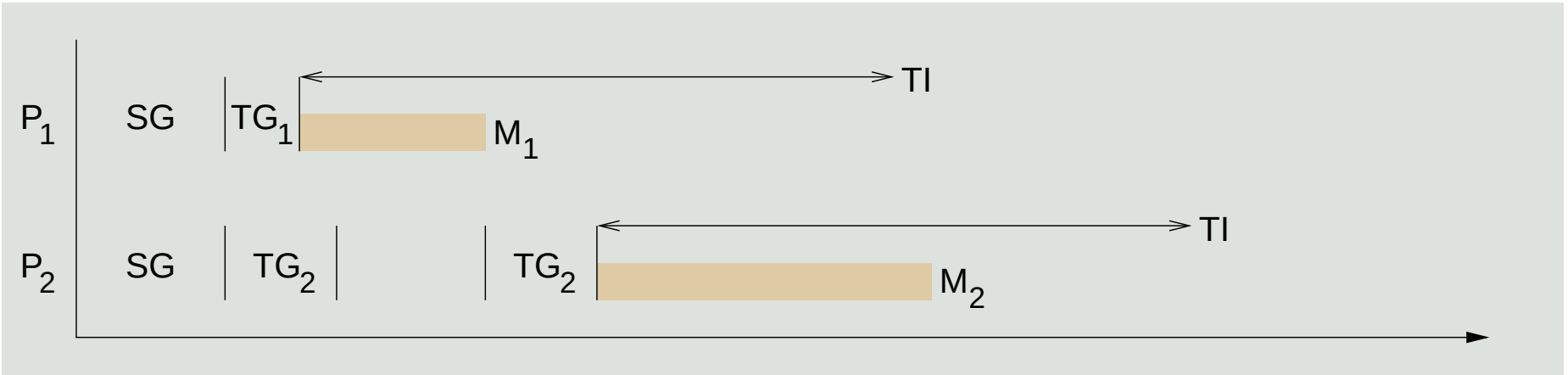
Relationen, die zwischen den Zeitparametern (*Timeouts*) definiert sind:

- $SG > \max(TG_i)$, für alle Knoten bzw. Prozesse i
- $TI > SG$

Minislotting (Forts.)

ARINC 629 — Protokollverlauf

Prozesse P_1 und P_2 wollen gleichzeitig senden, $TG_1 < TG_2$:



1. P_1 und P_2 warten SG Zeiten Busruhe ab, betreten den Warteraum
2. jeder Prozess i wartet zusätzlich noch seine TG_i Zeiten Busruhe ab
3. wegen $TG_1 < TG_2$ sendet P_1 zuerst seine Nachricht M_1
4. P_2 erkennt Verkehr auf den Bus; wartet, bis M_1 übertragen wurde
5. P_2 wartet TG_2 Zeiten Busruhe ab und sendet seine Nachricht M_2
6. P_1 und P_2 können frühestens nach TI Zeiten erneut senden

Master/Slave

Zentraler *Master* kontrolliert Buszugriffe

FIP (*factory instrumentation protocol*, [11]) Prozessautomatisierung

- arbeitet nach dem Modell „Produzent-Verteiler-Konsument“
  - Produzent $\mapsto$ pro Transaktion genau ein Sender
  - Verteiler $\mapsto$ Schiedsrichter (engl. *bus arbitrator*, BA)
  - Konsument $\mapsto$ pro Transaktion ggf. mehrere Empfänger
- Transaktionen verlaufen periodisch und in vier Schritten ab:
  1. BA gibt einen „Variablennamen“ per Sammelaufruf bekannt
  2. Produzent und Konsumenten erkennen die aufgerufene Variable
  3. der Produzent gibt den Variablenwert an die Konsumenten ab
  4. die Konsumenten nehmen den Variablenwert an, sofern benötigt
- Variablennamen sind systemweit eindeutige Bezeichner für...
  - *Boolean*, *Integer*, *Bitstring*, *Bytestring*, *General Time*, Verbünde
- freie Zeit kann für sporadische Daten genutzt werden
  - abfragen (engl. *polling*) der Knoten durch den BA

LIN (*local interconnect network*, [12]) Automobilindustrie

TDMA

(engl. *time division multiple access*)

Übertragungsrechte werden durch Voranschreiten der Echtzeit vergeben:

- Voraussetzung: fehlertolerante globale Zeitbasis in allen Knoten
- statische Aufteilung der gesamten Kanalkapazität in Zeitschlitze
  - jeder Knoten (Busteilnehmer) hat einen eindeutigen Sendeschlitz
  - TDMA Runde $\mapsto$ Sequenz von Sendeschlitzen einer Knotengruppe
    - in jeder Runde kann ein Knoten eine Nachricht übertragen
    - ist nichts zu versenden, bleibt ein Rahmen (engl. *frame*) leer
  - Runden wiederholen sich $\mapsto$ Gruppentakt (engl. *cluster cycle*)
    - Sequenz verschiedener TDMA-Runden
  - die Gruppentaktlänge bestimmt die Periodizität des TDMA-Systems

TTP (engl. *time-triggered protocol* [13])

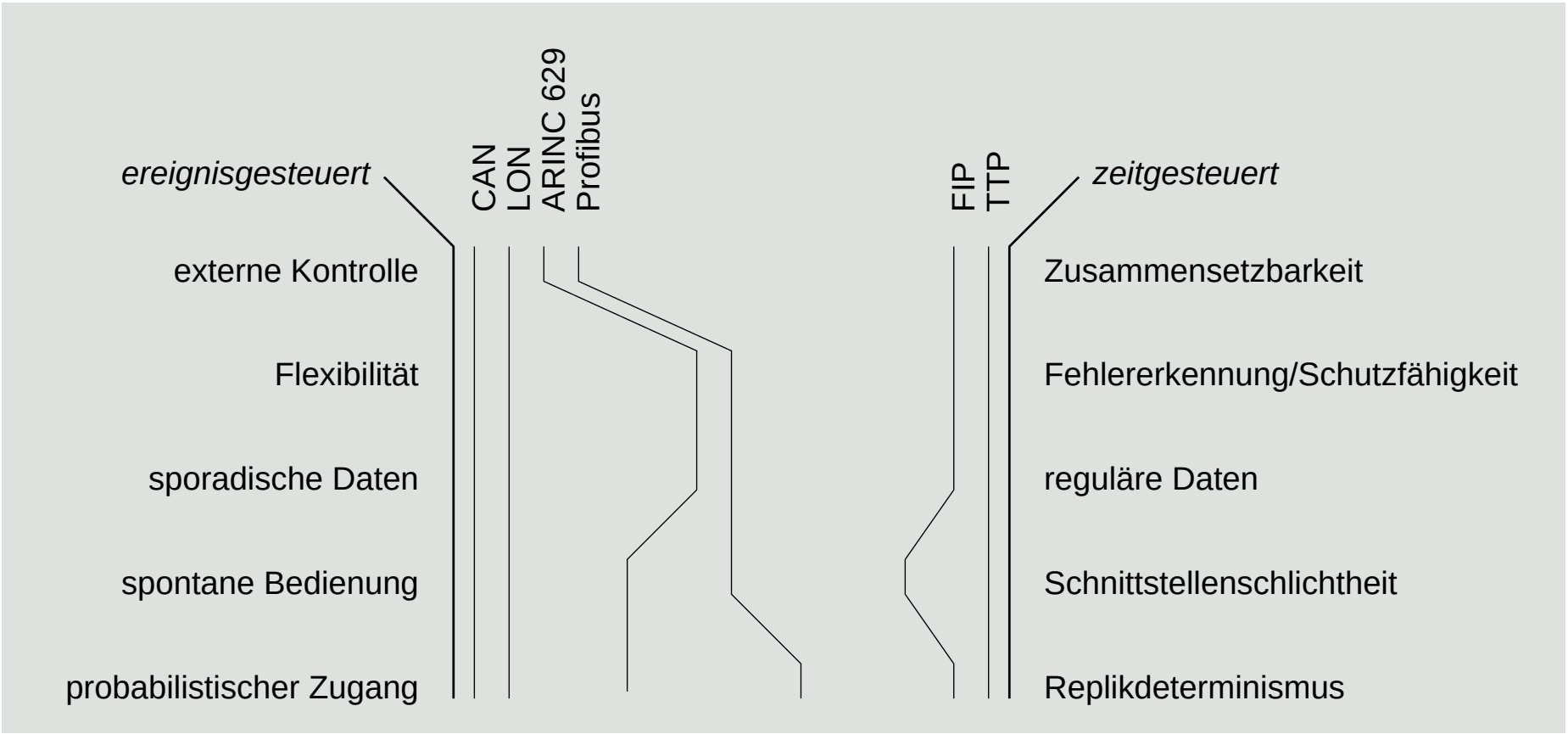
- Varianten, die in der Automobilindustrie Verwendung finden:

  byteflight ([14], Sicherheits- und Informationsbussystem: SI-BUS)

  FlexRay ([15], zeit- *und* ereignisgesteuerter Bus)

Vergleich

Entwurfsentscheidungen [1, S. 164]



Flexibilität
sofortige Antwort
sporadische Daten

versus

Zusammensetzbarkeit
Fehlererkennung
reguläre Daten

Resümee

Anforderungen an Echtzeitkommunikation

- Protokolllatenz, Fehlererkennung; physikalische Struktur
- Unterstützung für Zusammensetzbarkeit, Flexibilität

Flusskontrolle zur Vermeidung von Überlast

- explizit, implizit; Gegenüberstellung (HRTS)
- *send and wait* (PAR), Aktionsverzögerung, *Thrashing*

Netzwerkarchitekturen „*OSI considered harmful*“?

- ISO OSI 7-Schichtenmodell, Architektur von Echtzeitnetzwerken
- Echtzeitnetzwerk, Feldbus, Rückgratnetz

Netzzugangsprotokolle zur Vergabe des Betriebsmittels „Bus“

- Kommunikationsmedium, Charakteristiken des Übertragungskanals
- CSMA (CD/CA), *Token*, *Minislotting*, *Master/Slave*, TDMA

Literaturverzeichnis

[1] Hermann Kopetz.
Real-Time Systems: Design Principles for Distributed Embedded Applications.
Kluwer Academic Publishers, 1997.

[2] Bruce Jay Nelson.
Remote Procedure Call.
PhD thesis, Department of Comupter Science, Carnegie-Mellon University, Pittsburg, PA, USA, May 1981.
Technical Report CMU–81–119.

[3] Scott Johnson.
Inside TMI: Minute by minute.
`http://kd4dcy.net/tmi`, 2005.

Literaturverzeichnis (Forts.)

[4] International Organization for Standardization.
Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model.
ISO/IEC 7498-1. ISO, 1994.

[5] Robert M. Metcalfe and David R. Boogs.
Ethernet: Distributed packet switching for local computer networks.
Communications of the ACM, 19(5):395–404, July 1976.

[6] Echelon Corporation.
Enhanced media access control with LonTalk protocol.
Engineering Bulletin 005-0001-01C, January 1995.

[7] International Organization for Standardization.
Road vehicles — Control area network (CAN) — Parts 1–4.
ISO 11898. ISO, 2003.

Literaturverzeichnis (Forts.)

[8] Deutsches Institut für Normung.
Der Profibus.
DIN 19245. Beuth-Verlag, Berlin, Köln, 1991.

[9] Neil C. Audsley and Alan Grigg.
Timing analysis of the ARINC 629 databus for real-time applications.
In *Proceedings of the ERA Avionics Conference and Exhibition*, pages 10.1.1–10.1.11, Heathrow, UK, November 20–21, 1996.

[10] Leslie Lamport.
A new solution of Dijkstra's concurrent programming problem.
Communications of the ACM, 8(7):453–455, 1974.

[11] Philippe Leterrier.
The FIP protocol.
Technical report, WorldFIP Europe, Nancy, France, 1992.

Literaturverzeichnis (Forts.)

[12] J. Will Specks and Antal Rajnák.
LIN—protocol, development tools, and software interfaces for local interconnect networks in vehicles.
In *Proceedings of 9th International Conference on Electronic Systems for Vehicles*, Baden-Baden, Germany, October 5/6, 2000.

[13] Hermann Kopetz and Günter Grünsteidl.
TTP—a time-triggered protocol for fault-tolerant real-time systems.
In *Proceedings of the Twenty-Third Annual International Symposium on Fault-Tolerant Computing (FTCS-23)*, pages 524–533, Toulouse, France, June 22–24, 1993. IEEE.

[14] Josef Berwanger, Martin Peller, and Robert Grießbach.
byteflight — a new protocol for safety critical applications.
In *Proceedings of the 28th FISITA World Automotive Congress*, Seoul, Korea, June 12–15, 2000.

Literaturverzeichnis (Forts.)

[15] FlexRay Consortium.
Flexray communication systems.
Protocol Specification Version 2.1, Revision A, December 2005.