

Bussysteme

Vorlesungsmitschrieb

des Studiengangs

Informationstechnik

von

Jan Ulses

23. Februar 2015

Dozent:	Jürgen Röthig
E-Mail:	jr@roethig.de
Vorlesungszeitraum:	09.02.15 - 10.05.15
Klausurtermin:	04.05.2015
Autor:	Jan Ulses
Kurs:	TINF12B3
Ausbildungsfirma:	Harman/Becker Automotive Systems GmbH
Studiengangsleiter:	Jürgen Vollmer

Abbildungsverzeichnis

0.1	Stern-Topologie	5
0.2	Ring-Topologie	5
0.3	Bus-Topologie	6
0.4	Maschen-Topologie	6
0.5	Baum-Topologie	6
0.6	Linien-Topologie	7
0.7	Kollisionserkennung CSMA/CD	9

Tabellenverzeichnis

0.1 Einordnung von Technologien in Bussysteme	7
---	---

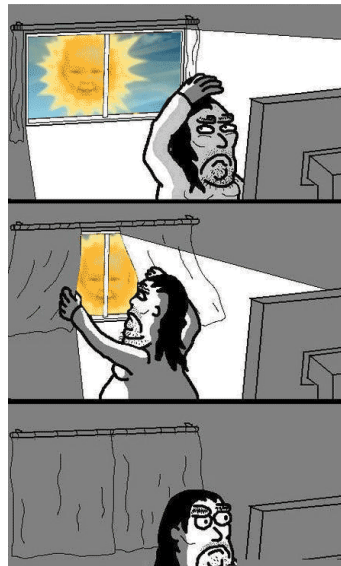
Vorwort

Dieser Vorlesungsmitschrieb ist im 6.Semester des Studiengangs Informationstechnik an der DHBW-Karlsruhe entstanden und wurde mit \LaTeX verfasst. Die Quelldateien sind [hier](#) in einem GIT-Repository zu finden.

Das Fach Bussysteme behandelt das Thema eines Datenbus. Dieser überträgt Daten zwischen Computerbestandteilen innerhalb eines Computers oder zwischen verschiedenen Computern. Anders als bei einem Anschluss, bei dem ein Gerät mit einem anderen Gerät über eine oder mehrere Leitungen verbunden ist, kann ein Bus mehrere Peripheriegeräte über den gleichen Satz von Leitungen miteinander verbinden.

Besonderer Dank geht an meinen Arbeitskollegen Sascha Moser, der hier für Grafiken und Anpassungen des Mitschriebs ausgeholfen hat.

Ein Dank geht auch an die Kommilitonen, welche in jeder Vorlesung ein Vorlesungsmitschrieb auf Google-Docs aktualisiert haben. Dieses konnte wunderbar bei Vorlesungsversäumnissen zum Nachtragen in dieses Dokument genutzt werden und kann [hier](#) eingesehen werden.



Inhaltsverzeichnis

Abbildungsverzeichnis	I
Tabellenverzeichnis	II
Vorwort	III
0.1 Wortherkunft	5
0.2 Topologien von „Computernetzen“	5
0.3 Verwenden alle „Bussysteme“ eine Bustopologie?	6
0.4 Spezielle Aufgaben von Bussystemen	7
0.5 Allgemeine Anforderungen/Aufgaben	7
0.6 Entwicklung eines Medienzugriffsverfahrens am Bsp. CSMA/CD	8
0.7 Exkurs: elektrische Signale	10
0.8 RS232	10

0.1 Wortherkunft

Bus „omnibus“ lat. „für alle“

hier: Kommunikationsmedium „für alle“

klass. „shared medium“: Was einer der Teilnehmer auf dem Medium sendet, hören (potentiell) alle Teilnehmer gleichzeitig mit.

0.2 Topologien von „Computernetzen“

Stern



Abbildung 0.1: Stern-Topologie

Es gibt einen zentralen „Sternverteiler“ in der Mitte und dedizierte Leitungen von diesem zu jedem der Teilnehmer. Üblicherweise ist der Sternverteiler nicht auch ein „normaler Teilnehmer“.

Ring



Abbildung 0.2: Ring-Topologie

Jeder Teilnehmer hat genau einen Vorgänger und einen Nachfolger, mit denen er jeweils verbunden ist.

Bus

Es gibt lediglich eine „Leitung“ als „shared medium“, mit welchem jeder Teilnehmer über eine „Stichleitung“ verbunden ist.

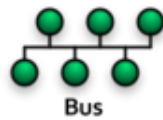


Abbildung 0.3: Bus-Topologie

Maschennetz

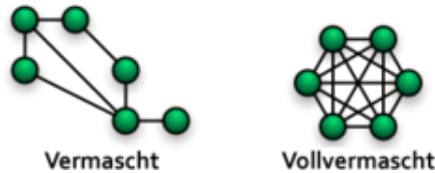


Abbildung 0.4: Maschen-Topologie

Jeder Teilnehmer hat zu beliebig vielen anderen Teilnehmern jeweils eine dedizierte Verbindung. Beim Spezialfall Vollvermascht hat jeder Teilnehmer eine Verbindung zu jedem anderen Teilnehmer.

Baum



Abbildung 0.5: Baum-Topologie

Die Baumtopologie ist eine hierarchische Topologie. Ausgehend von einem „Wurzel Teilnehmer“ gibt es jeweils ein oder mehrere Verbindungen zu Teilnehmern der nächsten Hierarchieebene.

Linie

Jeder Teilnehmer ist mit maximal zwei Teilnehmern verbunden. Es gibt einen Anfang und ein Ende.

0.3 Verwenden alle „Bussysteme“ eine Bustopologie?

Viele der heutigen „Bussysteme“ sind physikalisch keine Busse, sondern nur noch logisch/protokolltechnisch auf einer höheren Ebene.



Abbildung 0.6: Linien-Topologie

Bustopologie	Bustopologie
Ethernet in BNC-Verkabelung	USB: Baum
ProfiBus	MOST: Ring
WLAN	Ethernet in TP-Verkabelung: Baum
IDE (P-Data)	(S-ATA)
PCI	(PCI-Express)
SCSI	(SAS (Seriell Attached SCSI))
	Firewire: vermaschtes Netz mit Einschränkungen

Tabelle 0.1: Einordnung von Technologien in Bussysteme

0.4 Spezielle Aufgaben von Bussystemen

Speziell bei Bussystemen zu lösende Aufgaben:

- Medienzugriff
- Adressierung
- Synchronisierung zwischen Sender und Empfänger

0.5 Allgemeine Anforderungen/Aufgaben

- Übertragung
 - synchron = getaktet Die Übertragung findet in einem festen Zeitraster statt, d.h. zwischen zwei aufeinanderfolgenden Übertragungen vergeht immer dieselbe Zeit (\Rightarrow Takt).
 - Beispiele für synchron:
 - * RS232 (serielle Schnittstelle)
 - Takt: Baudrate, d.h. Schritte pro Sekunde
 - in jedem Schritt wird maximal ein bit übertragen
 - \Rightarrow (bit-)serielle Schnittstelle
 - \Rightarrow bitrate \neq Baudrate
 - Möglichkeiten zu getakteter Übertragung:
 - Taktleitung
 - verbindliche Vereinbarung (hier verwendet) dann aber notwendig: Erstsynchronisation zu Beginn, danach regelmäßige Nachsynchronisation (we-

gen Ungenauigkeit der Taktgeber)
Eine Signalleitung für jede Richtung ist ausreichend
Signale: elektrische Spannung

- * asynchron = bedarfsgesteuert Jede Übertragung beginnt unabhängig vom Zeitpunkt der vorherigen Übertragung, d.h. dann, wenn beide zur Übertragung bereit sind.
- * Beispiele für asynchron:

0.6 Entwicklung eines Medienzugriffsverfahrens am Bsp. CSMA/CD

ALOHA (erstes Bussystem für die Entwicklung des CSMA/CD): Bei Bedarf wird gesendet.

Problem: Zwei oder mehr Sender können gleichzeitig senden, dadurch werden alle Sendungen gestört und müssen verworfen werden.

Aufgrund der nicht zu hohen Anzahl an Teilnehmern und der geringen Datenmenge ist die Kollisionswahrscheinlichkeit gering und das Verfahren funktioniert trotzdem (Kollision muss auf höherer Protokollebene durch zweiten Sendeversuch erkannt und behoben werden).

Slotted ALOHA (Weiterentwicklung des ALOHA): Die Verfügbarkeit des Mediums wird in Zeitschlitze eingeteilt (synchron für alle Teilnehmer), dabei umfasst jede Sendung genau einen Zeitschlitz.

⇒ verringerte Kollisionswahrscheinlichkeit, da Kollisionen nur zu Beginn des Zeitschlitzes und nicht mehr in dessen Verlauf stattfinden können

⇒ nur genau dieser Zeitschlitz ist von der jeweiligen Kollision betroffen.

CSMA (Carrier Sense Multiple Access): Vor dem Beginn der Übertragung wird der Übertragungskanal abgehört, ob bereits eine Sendung stattfindet.

Falls Ja ⇒ späterer Versuch

Falls Nein ⇒ Sendeversuch möglich

Achtung: CSMA ist nur als Weiterentwicklung von “Unslotted” ALOHA möglich Bei Slotted ALOHA ist der Übertragungskanal unmittelbar zu Beginn des Zeitschlitzes immer frei!

⇒ CSMA bringt in diesem Fall keinen Vorteil!

Jetzt: Kollisionen können nur “quasi-gleichzeitig” stattfinden, d.h. innerhalb eines Zeitfensters, welches der Signalverzögerung zwischen den beiden an der Kollision beteiligten Stationen entspricht.

Vorher bei Slotted ALOHA: Kollision immer dann, wenn während des gesamten vorherigen Zeitschlitzes bei zwei oder mehreren Stationen ein Übertragungswunsch entstanden ist.

⇒ CSMA reduziert die Kollisionswahrscheinlichkeit, sofern die Übertragungsverzögerung kürzer als ein Zeitschlitz bei Slotted ALOHA ist.

CSMA/CD (CSMA with Collision Detection): Kollisionserkennung, man hört den Übertragungskanal auch während der eigenen Übertragung ab und vergleicht das gesendete mit dem empfangenen Signal. Ungleichheit bedeutet Kollision.

Sender, welcher die Kollision erkennt bricht die Übertragung ab und sendet stattdessen ein sog. JAM Signal, welches den anderen Stationen (insbesondere den Empfängern) signalisiert, dass eine Kollision stattgefunden hat und die bisherige Sendung verworfen werden muss.

⇒ keine Verringerung der Kollisionswahrscheinlichkeit

⇒ Verkürzung der Zeitdauer in der der Übertragungskanal mit Störung (und damit nutzlos) belegt ist, weil direkt abgebrochen wird



Abbildung 0.7: Kollisionserkennung CSMA/CD

Zur sicheren Kollisionserkennung muss die Mindestlänge eines Paketes auch und gerade beim 1. Sender die doppelte Signalverzögerung aufweisen ⇒ ansonsten bekommt der 1. Sender das JAM Signal des zweiten Senders erst nach dem Ende seiner eigenen Übertragung mit ⇒ zu spät.

Zahlenwert ("klassisches CSMA/CD"):

- Übertragungsverzögerung: $12,5\mu s$
- max. Ausdehnung: $\approx 2500m$
- Übertragungsrate: $10MBit/s$
- Geschwindigkeit in einem Kupferkabel: $200000km/s$

$$v = st \Rightarrow t = sv$$

$$v \approx \frac{2}{3}c \approx 2 * 10^8 \frac{m}{s}$$

$$t = \frac{2500m}{200 * 10^6 \frac{m}{s}} = 12,5s$$

$$r = \frac{m}{t} : \text{Speichermenge(inbit)}$$

$$m = r * t = 125bit \approx 16Byte$$

- in der Realität 42 oder 46 Byte
- Folgerung für Fast Ethernet (100Mbit/s):

keine größere minimale Datenmenge!

sondern kleinere “Kollisionsdomänen”

mehrere Kollisionsdomänen müssen über Switches (“Store and Forward”) und nicht einfach Hubs (Signalverstärker) verbunden werden!

- Folgerung für Gigabit Ethernet (1Gbit/s):

für jede Übertragungsrichtung zwischen jeweils zwei Geräten (Endgerät oder Switch - kein HUB erlaubt) gibt es einen eigenen Übertragungskanal \Rightarrow Vollduplex \Rightarrow keine Kollisionen möglich!

0.7 Exkurs: elektrische Signale

In der Leitung fällt Spannung aufgrund des vorhandenen Leitungswiderstandes $\neq 0$ ab.

$$\frac{V_{Leitung}}{V_{Empfänger}} = \frac{R_{Leitung}}{R_{Empfänger}}$$

\Rightarrow möglichst geringer Leitungswiderstand, möglichst großer Innenwiderstand beim Empfänger Strom:

- + unempfindlich gegen große Abstände
- energieintensiver
- + unempfindlicher gegen EMV-Einstrahlung
- produziert mehr Störungsstrahlung auf andere Leitungen

bei hohen Spannungen: Vor/Nachteile wie bei Strom, zusätzlich:

- Umschalten dauert länger (wegen kapazitiver Wirkung der Leitung)

0.8 RS232

Eine Signalleitung für jede Richtung ausreichend.

Signale: elektrisch

Spannung $0 \hat{=} 12V, 1 \hat{=} -12V$

Ruhepegel: Wenn der Sender nichts zu senden hat, legt er einen ”1Pegel auf die Leitung

Startbit: Mit einem ”0bit für genau eine definierte Schrittzeit (siehe Baudrate) wird die Übertragung ertsynchronisiert”

Datenbit: (zwischen 5 und 8 bit konfigurierbar) Anzahl der bit, welche ein Wort ausmachen. . . maximal 8bit=1Byte

Paritätsbit: (optional, gerade oder ungerade Parität konfigurierbar)

Stoppbits: (ein oder zwei Stoppbits konfigurierbar) Art “künstlicher Ruhepegel”, also “1”, um die Synchronisation durch das nachfolgende Startbit sicher zu ermöglichen.

⇒ während des ”Wortes” (Datenbits+Paritätsbit) ist keine Nachsynchronisation gewährleistet

⇒ Uhr muss während dieser Zeit hinreichende Genauigkeit bieten...

... wie stark darf die Uhr abweichen?

Der Empfänger misst den Pegel genau in der Mitte der von ihm angenommenen Schrittzeit.

Die Abweichung der beiden Uhren voneinander darf max. $\frac{1}{2}$ Bitzeit betragen.

Die max. Abweichung jeder Uhr von der “Normzeit” ist damit nur $\frac{1}{4}$ Bitzeit!

Diese Abweichung darf während des Startbits, der (max) 8 Datenbits und dem (optionalen) Paritätsbit = 10 Bitzeiten nicht auftreten.

$$Abweichung < \frac{\frac{1}{4}}{10} = 2,5\%$$

⇒ die RS232 arbeitet während eines Wortes (von Startbit bis zum Stoppbit) bit-synchron

Das nächste Wort wird mit dem Startbit unabhängig vom vorangegangenen Wort dann begonnen, wenn notwendig

⇒ die RS232 arbeitet Wort-asynchron

Fazit: synchron und asynchron schließen sich gegenseitig nicht aus!

seriell: die einzelnen Informationsbestandteile (z.B. bit) werden nacheinander übertragen

parallel: mehrere Informationsbestandteile (z.B. bit) werden gleichzeitig übertragen

z.B. bit-parallel: 8 Leitungen für 8 bit gleichzeitig, aber mehrere Byte müssen nacheinander übertragen werden, d.h. Byte-seriell

Fazit: serielle und parallele Arbeitsweise schließen sich gegenseitig nicht aus!

Synchronisierung bei der asynchron arbeitenden parallelen Schnittstelle (Centronics-Schnittstelle)
8 Datenleitungen für 8 gleichübertragene Datenbit; d.h. bitparallel und Byte-seriell; aufeinanderfolgende Bytes werden nacheinander und asynchron zueinander übertragen (also nicht in einem festen Zeitraster)

TTL-Pegel, d.h. 15V und 00V

zur Anzeige, ob Daten anliegen, wird eine weitere “Handshake-”Leitung gebraucht ⇒ ”Valid” von Sender zum Empfänger zeigt an, ob gültige Daten anliegen.