Rechnernetze und verteilte Systeme

Schichtunabhängige Konzepte

Kapitel 3

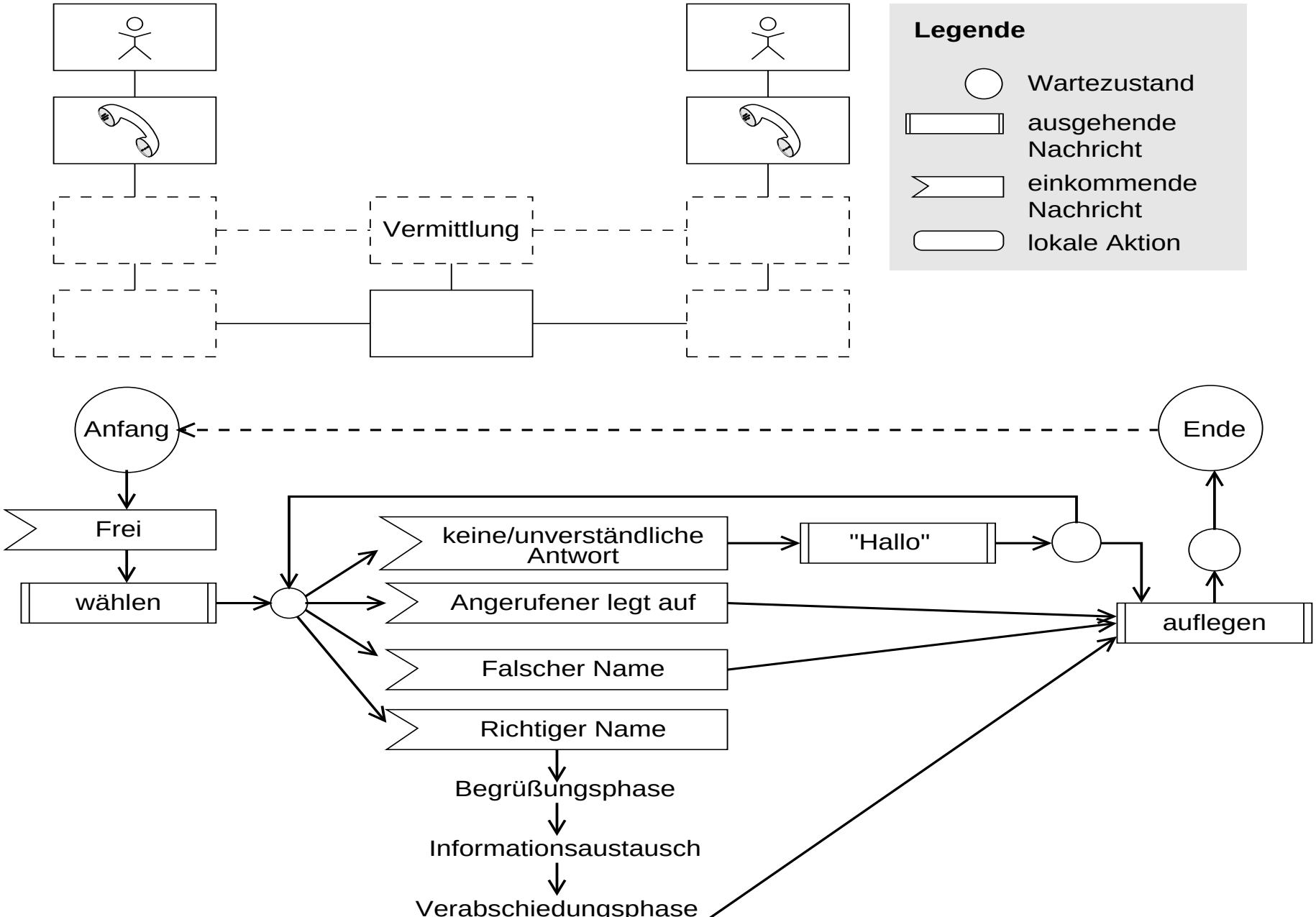
Rechnernetze und verteilte Systeme
Schichtunabhängige Konzepte

Beispiele von Protokollbeschreibungen

Kapitel 3.1

3.1 Beispiele von Protokollbeschreibungen

Telefongespräch als Diagramm (1)



- Kräftige Vereinfachung !
- Nur Anruferseite dargestellt
- Es fehlt u.a.:
  - kein Freizeichen
  - kein Rufzeichen
  - Besetztzeichen
  - Unterbrechung der Verbindung
  - Zeitbegrenzung der Wartezustände
  - zu lange Pause bei Partner
  - Partner legt zuerst auf
  -
- Aufgabe: Ergänze Protokollbeschreibung möglichst vollständig!

3.1 Beispiele von Protokollbeschreibungen

Protokoll Terminal / Rechner an langsamer Analogschnittstelle

MNM TEAM

Aufbau

- T: (hier Mensch) Abheben Handapparat
- V: Aktivieren der Wähleinrichtung, Freizeichen
- T: (Mensch oder Computer) wählen der Nummer des Rechners
- C: Rufsignal erkennen, wenn bereit 1650 Hz-Ton senden
- V: 1650 Hz-Ton übertragen
- T: (Mensch oder Computer) Ton hören, umschalten auf Übertragung. T sendet 980 Hz-Ton
- C: erkennt 980 Hz

Datenphase

- T: sendet 980 Hz binär 1, sendet 1180 Hz binär 0
- C: sendet 1650 Hz binär 1, sendet 1850 Hz binär 0

Abbau

- C: unterdrückt 1650 Hz
- T: erkennt Ende; Warnsignal. Umschalten auf Fernsprechen, auflegen
- V: löst Verbindung
- C: nach kurzer Verzögerung erneut empfangsbereit.

Akustikkoppler

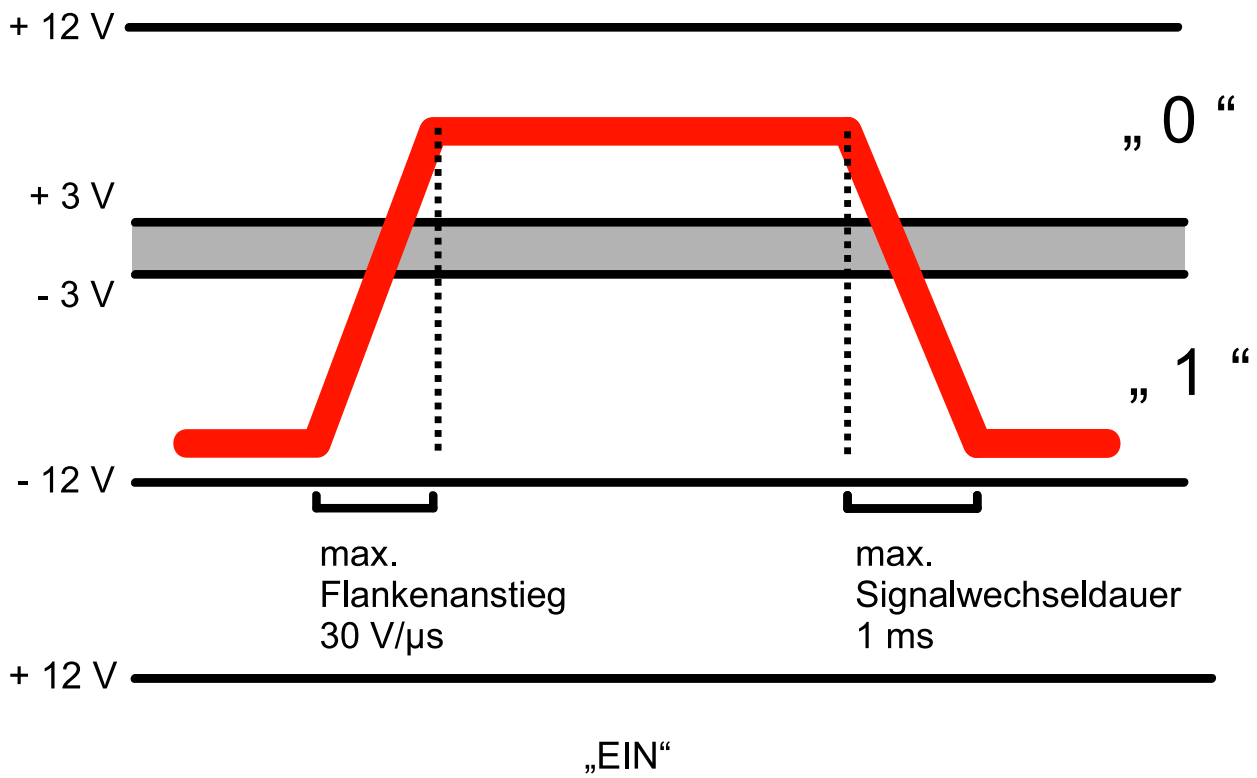
Legende

- T Terminal
- V Vermittlung
- C Computer

3.1 Beispiele von Protokollbeschreibungen

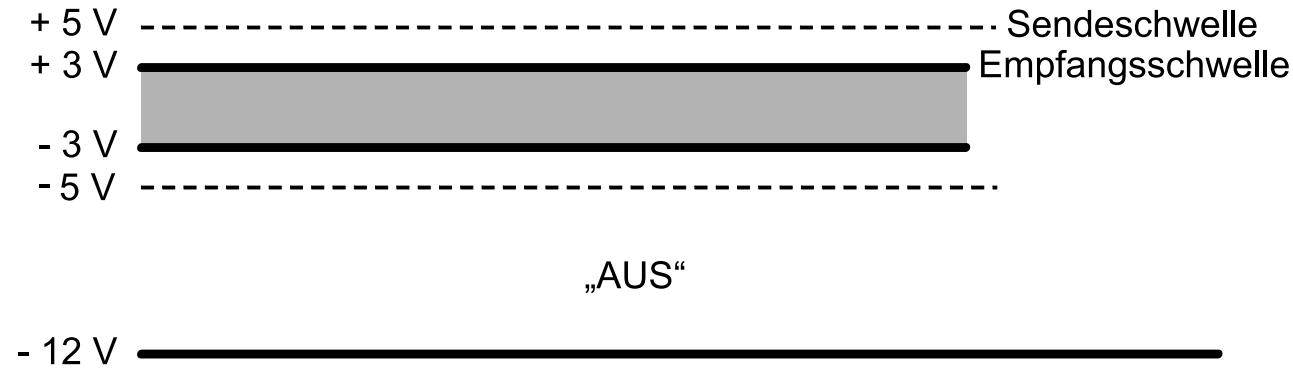
Ausschnitt aus V.24 (Signale)

a) Datenleitungen



„EIN“

b) Meldeleitungen und Steuerleitungen



„AUS“

- 12 V

Datensatz (Beispiel)

PAD	SYN	SOH	*Header*	STX	*Text*	ETX	BCC

- BSC (Binary Synchronous Communications): Zeichenorientierte Leitungsprozedur
- PAD, SYN, SOH, STX, ETX: feste Steuerzeichen aus ASCII
- Header: aus ASCII-Zeichen
- Text: Nutzdaten aus ASCII-Zeichen
- BCC: Prüfzeichen (Block Check Character)

- L2CAP: Erbringt manche Schicht-2-Funktionen
  - Logical Link Control Adaptation Protocol
  - Bildung von Rahmen
  - Auswahl von Protokoll auf der nächsthöheren Schicht
  - Aushandeln von Dienstgüteparametern
- Access code: z.B. zur Unterscheidung zw. Gegenstellen
- Header: 18 Bits
  - Wird drei Mal gesendet; bei Empf. werden Kopien verglichen
  - Mehrheitsbeschluß bei Abweichung

L2CAP Rahmen

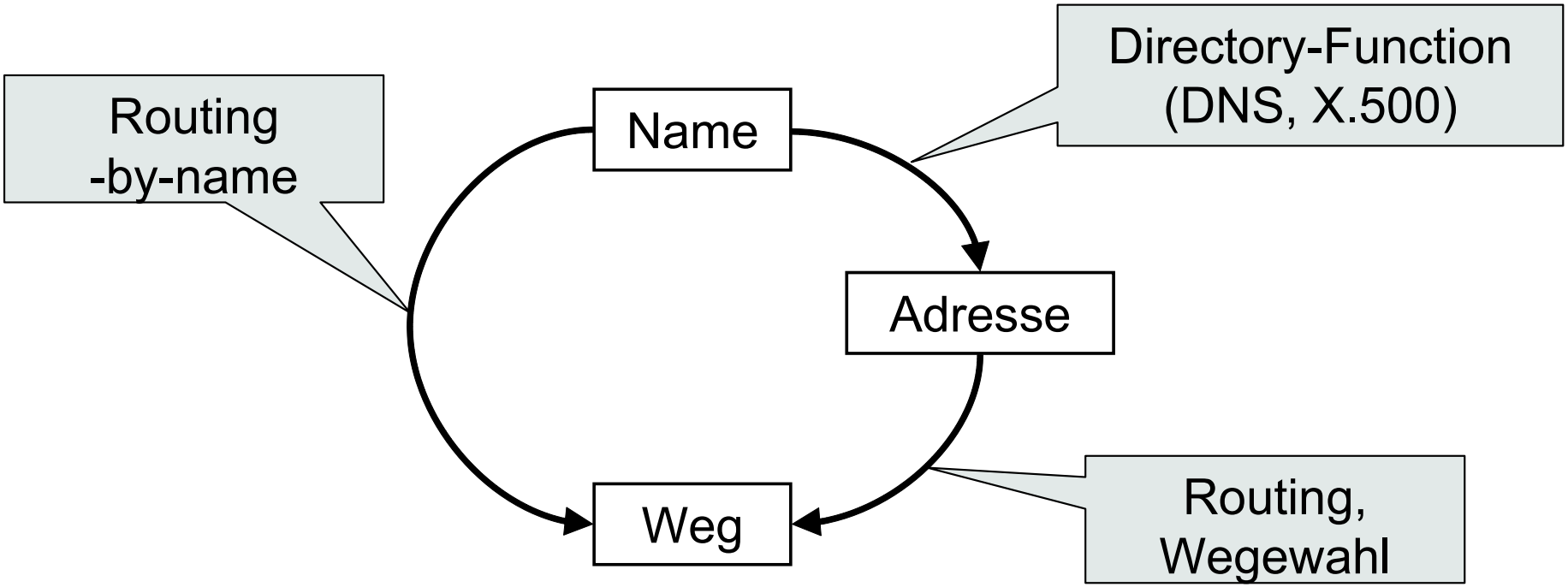
Bits	72	3 * 18 = 54	0-2744
Feld	Access code	Header	Nutzdaten

- Adressierung
- Verbindungsaufbau / Abbau
- Übertragungssteuerung (Quittungen, Timer, Flusssteuerung)
- Fehlerbehandlung und –erkennung (Verfälschung, Duplikate, Verlust)
- (De-)Komposition von Nachrichten
- Splitting / multiplexing
- Arbeitsteilung und Zusammenarbeit mit Protokollen benachbarter Schichten
- Methode der Protokollspezifikation
  - erweiterte endliche Automaten [ESTELLE]
  - Ablaufdiagramme [SDL]
  - Programmiersprachen [ASN.1]
  - Temporale Logik [LOTOS]
  - Petrinetze
- Protokollverifikation, -validierung (Korrektheit, Verklemmungsfreiheit)
- Protokolltests
  - mit Partner: Interoperabilität
  - gegen Standard: conformance test

Rechnernetze und verteilte Systeme
Schichtunabhängige Konzepte

Namen, Adressen, Wege

Kapitel 3.2



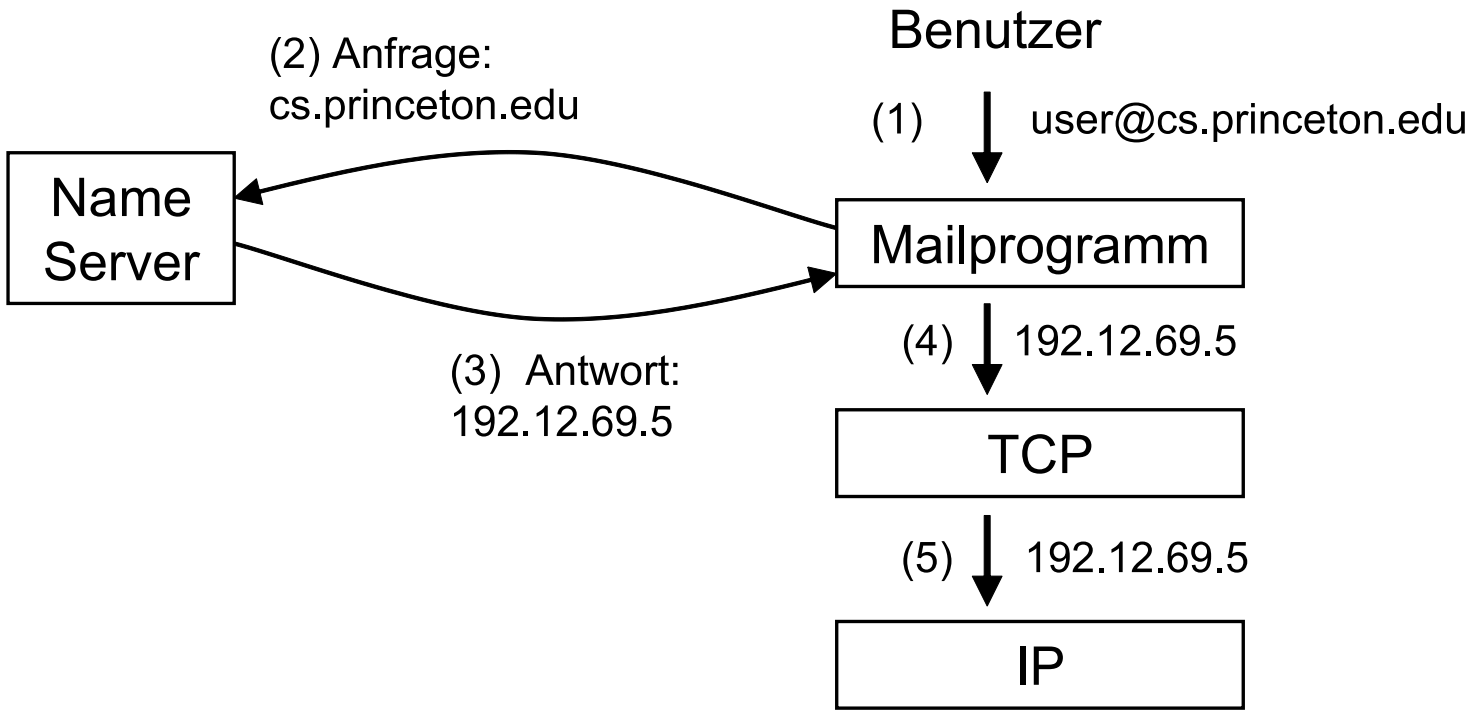
- Name: „Wer”. Bezeichner für Objekt, Instanz, Prozess
- Adresse: „Wo”. Bezeichner für Ort des Objekts
- Weg: „Wo lang”. Bezeichner für Weg zum Objekt

- Namen
  - Bequemlichkeit für Nutzer (Mnemonic)
  - Relokation von Objekten ohne Namensänderung
  - Erzeugung von eindeutigen Namen in verteilter Umgebung
  - mehrere Inkarnationen von logisch identischen Objekten
  - Bedeutung oft kontextabhängig
  - strukturiert oder flach
- Adressen
  - meist in numerischer Form
  - Eindeutigkeit wichtig
  - strukturiert oder flach
- Weg
  - Liste von Namen oder Adressen, die Pfad zum Ziel repräsentieren

- über Zuordnungstabellen, Verzeichnisdienste Beispiele: Personalnr., Versicherungskennzeichen, Telefonnr.
- aus Namensstruktur ableitbar

```
<Person> ::= <Land> <Ort>. <Straße> <Nr> <Stock> <Name>
<Prozessname> ::= <Netz>.<Subnetz>.<Host>.<Prozess ID>
```

- Im globalen Internet geschieht die Namensauflösung über DNS-Hierarchien. Erste Domain-Ebene umfasst edu, com,gov,mil, org, net, Länder (z.B. „de") u.a.



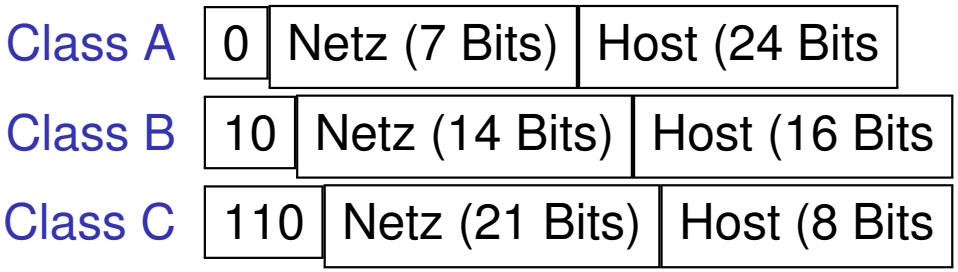
- Struktur
  - uniform global (durchlaufende Nummerierung) Beispiel: Ethernet-Adressen
  - hierarchisch Beispiel: X.121-Adresse, X.400-Adresse, Telefonnr.
- Umfang des Adressraumes
  - groß: permanente Zuordnung, große Header
  - klein: Mehrfachverwendung erfordert Vergabestrategie (z.B. DHCP)
- Codierung
  - variable Namensfelder (→erweiterbar)
  - feste Namensfelder (→einfach)

- 32-Bit-Adressen: 4 Bytes (p.q.r.s), z.B. „208.77.188.166”
- Dezimale Schreibweise: $0 \leq p, q, r, s \leq 255$
- Linker Teil adressiert Netz; rechter Teil adressiert Host.

Netz ID	Host ID

- Besondere Adressen:
  - alles 0 : dieser Host
  - alles 1 : globaler Broadcast
  - Netz ID = 0 : dieses Netz
  - Host ID = 1 : Broadcast im Subnetz
  - 127.0.0.* : Loop-Back-Adresse (Schleifentest)
- Adressvergabe durch NIC (Network Information Center, z.B. DE-NIC); international durch IANA (Internet Assigned Numbers Authority)
- Private Adressen
  - für organisationsinternen Gebrauch reserviert; werden nicht im Internet vermittelt
  - z.B. 192.168.*.*

- Klasse: Vorgabe der Länge der Netz-ID und eines Netzpräfix
- an Präfix kann die Netzklasse abgelesen werden



- Beispiele

Class A 00001010 00000000 00000000 00100000 = 10.0.0.32
Class B 10000010 00000011 00000011 00111100 = 130.3.3.60

Class		Netz ID	Host ID	Subnetze	Hosts
A	$p < 127$	p	q.r.s	126	16777214
B	$128 \leq p \leq 191$	p.q	r.s	16382	65534
C	$192 \leq p \leq 223$	p.q.r	s	2097152	254

- Netz ID bzw. Subnet ID sind Grundlage für Routing-Entscheidung
- schnelle Erkennung der Länge der Netz ID anhand Präfix
  $\Rightarrow$Performanzgewinn

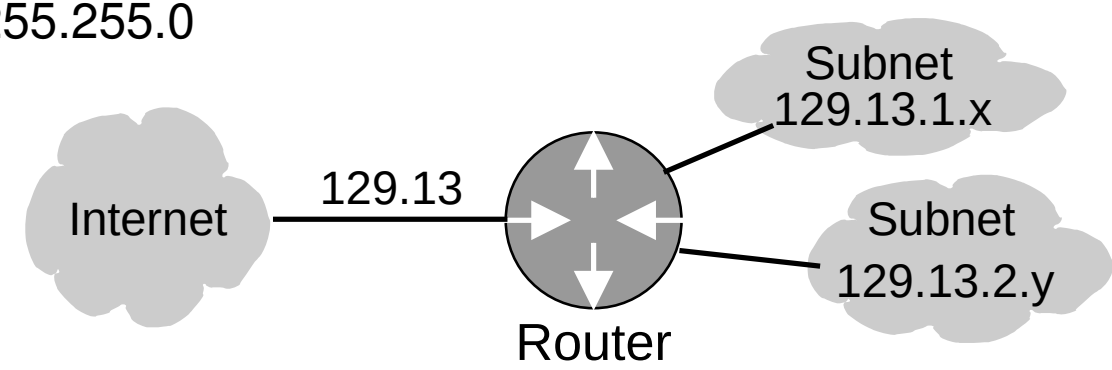
Netz ID	Host ID

Klassenadresse

Netz ID	Subnet ID	Host ID

privates Subnetting

- Adressierungsaspekte im Ziel-Subnetz (Ziel-Router)
  - Subnetz = LAN: Abbildung IP-Adresse auf MAC-Adresse (ARP / RARP)
  - Subnetz = (ISDN, ATM, FR, X.25)
    - Adressermittlung über Tabelle / Directory
    - Verbindungsaufbau
  - Subnetz = Intranet: IP-Adressumsetzung NAT/PAT
- Ausblenden durch Subnet-Masken (bitweises AND)
- Standardmasken
  - A 255.0.0.0
  - B 255.255.0.0
  - C 255.255.255.0



Szenario

Ein Unternehmen plant Verbindung seiner weltweiten Niederlassungen. Es hat 50 Standorte mit etwa 1000 Subnetzen von je etwa 500-800 Endsystemen pro Subnetz.

Diskutieren Sie die Möglichkeiten der Adressvergabe auf der Basis von Class A-Adressen und von Class B- Adressen.

- CIDR: Classless Interdomain Routing
- Struktur: | Network prefix | Subnet ID | Host ID |
- Notation: <network>/<prefix length>
- Beispiel: 192.168.121.0/26
  die ersten 26 Bits bilden Netz ID; der Rest ist Host ID
- Zweck: Grenzen zwischen Netz ID und Host ID werden flexibel
- Bei CIDR müssen im Routingprotokoll die Präfixe in den Routenankündigungen übermittelt werden
- Adresszuweisungen müssen die Netztopologie berücksichtigen (hierarchisches Routing)
- RFC 1517 bis 1520
- Subnetting auch bei CIDR möglich durch Verwendung von Variable Length Subnet Mask

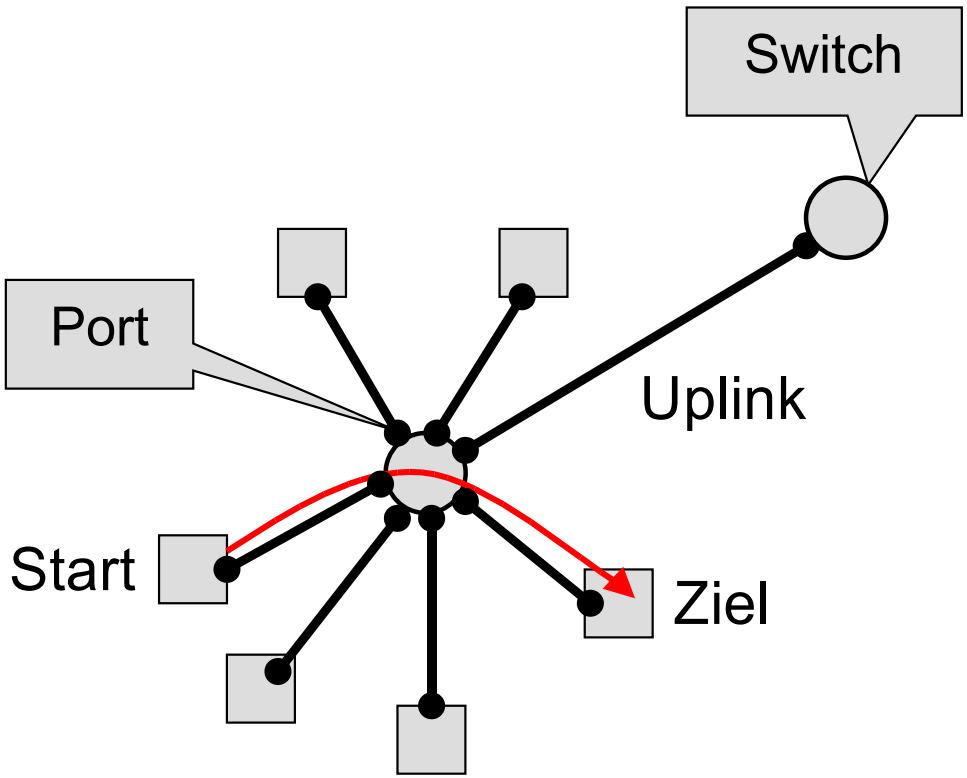
Eine Organisation bekommt den Adressblock 131.42.0.0/16 zugewiesen und benötigt

- 1 Subnetz mit bis zu 32000 Hosts
- 15 Subnetze mit bis zu 2000 Hosts
- 8 Subnetze mit bis zu 250 Hosts

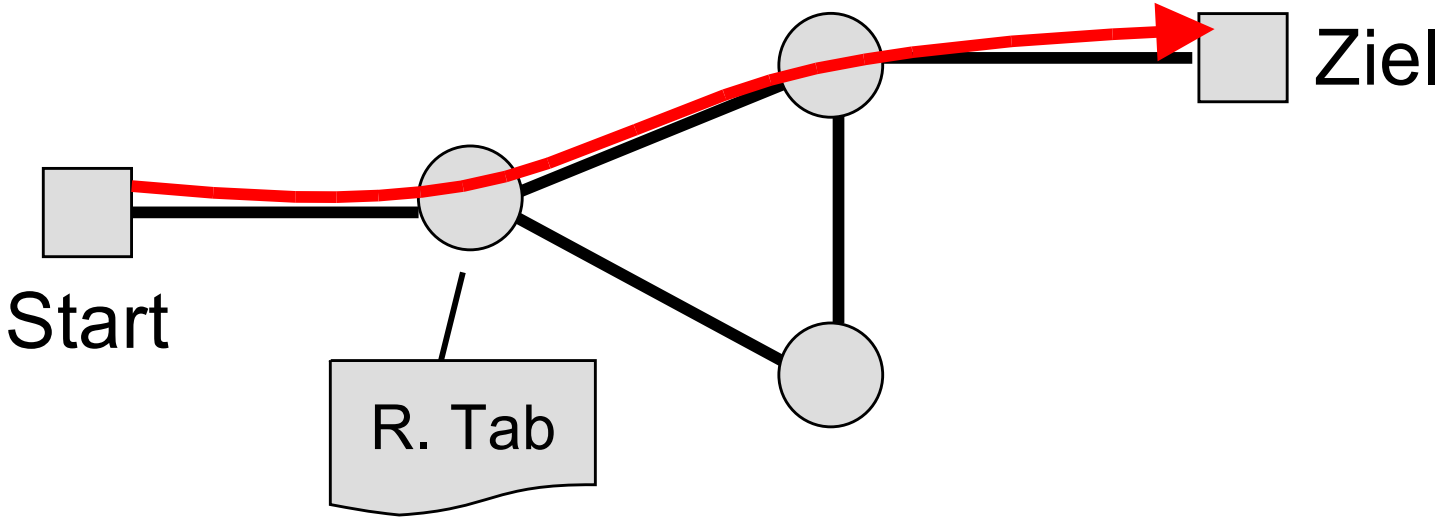
Machen Sie Vorschläge für eine Aufteilung in geeignete Subnetz-Adressen!

- TCP–Nutzer sind durch Sockets gekennzeichnet
- TCP–Verbindung ist gegeben durch (socket 1, socket 2)
- Socket = (IP–Adresse Host, lokale Port-Nr [16 Bits])
- Port-Nr adressiert Anwendungen (d.h. Prozesse, die auf einem Host laufen)
- Vereinbarte (“well-known”) Ports für Standarddienste, z.B.
  - Port = 21 (FTP)
  - Port = 23 (Telnet)
  - Port = 80 (http)
- siehe RFC 1700

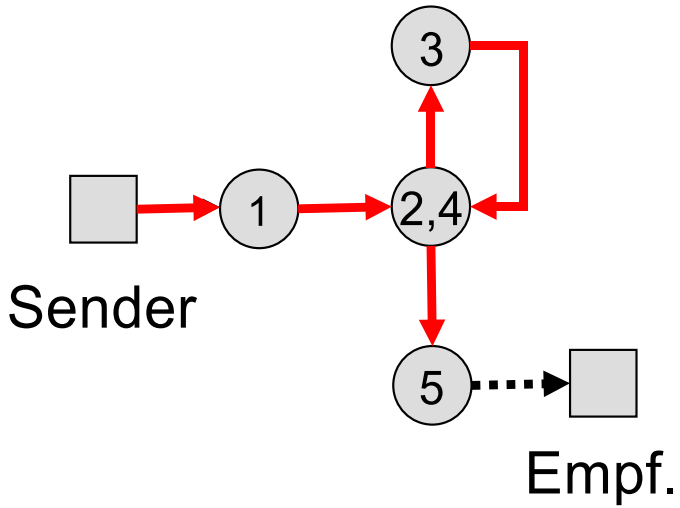
- Ports: Anschlussstellen f. Medium
- Wegewahl: Zuordnung eines Zielport anhand Adresse
- Switch findet Zuordnung
  - anhand von Broadcast
  - anhand von bereits beobachtetem Netzverkehr
- Switch erlernt Zuordnung



- Router trifft Entscheidungen zur Weiterleitung von Paketen
- Weg bestimmt durch
  - Regeln in Routingtabellen
  - Angaben des Senders (selten)
- Routingprotokolle: zum Aushandeln optimierter Wege
  - nach technischen Gesichtspunkten
  - nach organisatorischen/finanziellen Gesichtspunkten



- Wegewahl anhand von DNS
- Oft direkte Zustellung
- Evtl. mehrere Hops



Received: from mailrelay1.lrz-muenchen.de (mailrelay1.lrz-muenchen.de [129.187.254.106])
by **mail.nm.ifi.lmu.de** (8.12.11/8.12.11/Linux MNM 0.1) with ESMTP id l9CAC5v7004532
for <danciu@nm.ifi.lmu.de>; Fri, 12 Oct 2007 16:16:05 +0200
Received: from lxmhs06.lrz-muenchen.de (lxmhs06.lrz-muenchen.de [10.156.6.203]) by **mailrelay1.lrz-muenchen.de** with ESMTP for danciu@nm.ifi.lmu.de; Fri, 12 Oct 2007 16:16:04 +0200
Received: from mailrelay1.lrz-muenchen.de ([10.156.6.201])
by **lxmhs06.lrz-muenchen.de** (lxmhs06.lrz-muenchen.de [10.156.6.203]) (amavisd-new, port 10024)
with ESMTP id mDqRYsUc8VAp for <danciu@nm.ifi.lmu.de>; Fri, 12 Oct 2007 16:16:03 +0200 (CEST)
Received: from mail.gmx.net (mail.gmx.net [213.165.64.20]) by **mailrelay1.lrz-muenchen.de** for danciu@nm.ifi.lmu.de; Fri, 12 Oct 2007 16:16:02 +0200
Received: (qmail invoked by alias); 12 Oct 2007 10:11:01 -0000
Received: from yyyyyyyyyy.dip.t-dialin.net (EHLO ASMCore2Duo) [217.238.48.17x]
by **mail.gmx.net** (mp042) with SMTP; 12 Oct 2007 16:15:01 +0200
From: "NN" <xxxxxx@gmx.de>
To: "'Vitalian A. Danciu'" <danciu@nm.ifi.lmu.de>
Subject: Frage Rechnernetze 1

TEXT

Rechnernetze und verteilte Systeme
Schichtunabhängige Konzepte

Sequenznummern

Kapitel 3.3

3.3 Sequenznummern
Fehlertypen im Netz (1)

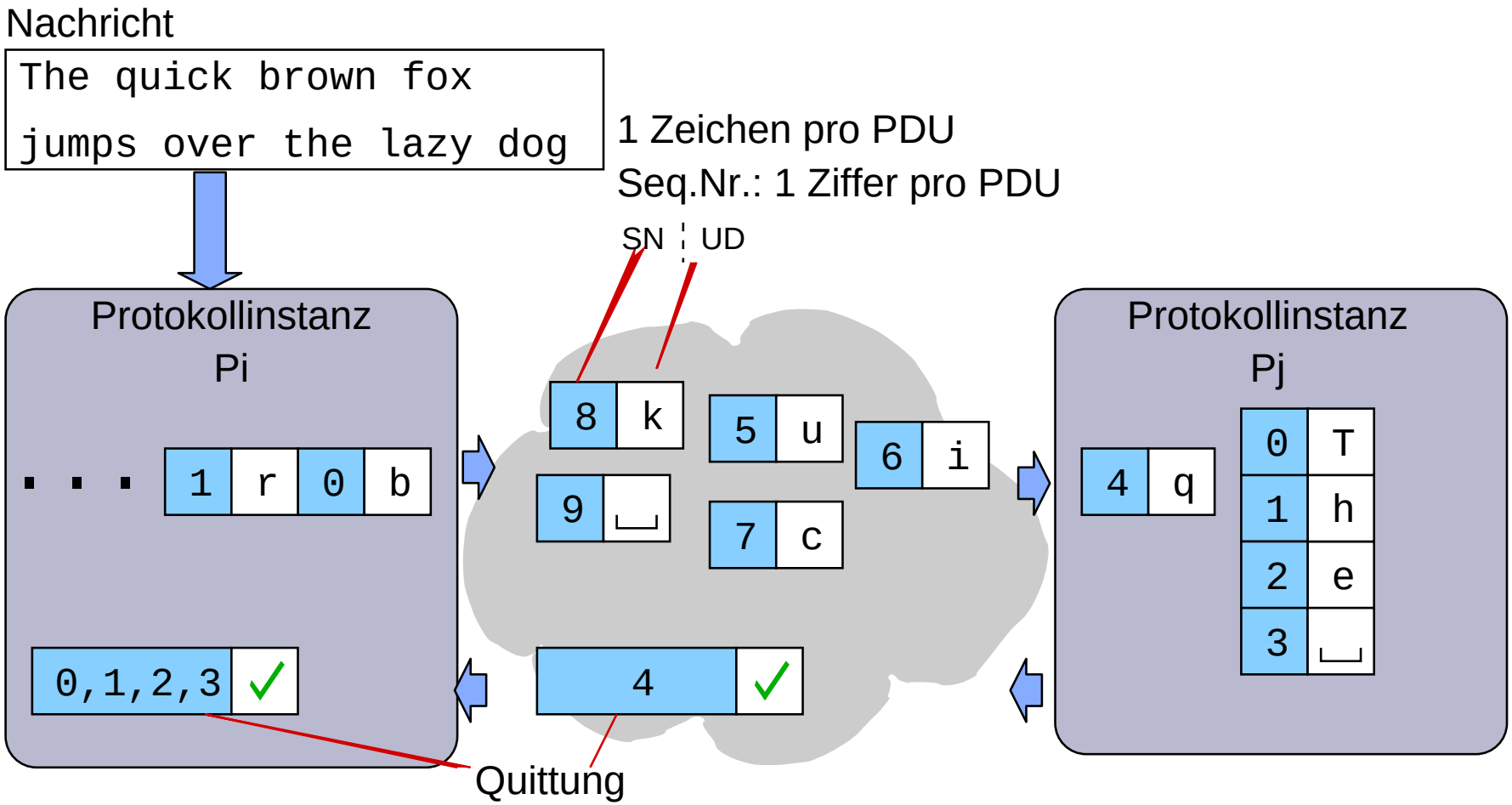
- Verfälschung
  - Gründe: Bits werden verfälscht
    - auf Medium,
    - an Schnittstelle,
    - im Knoten
    - z.B. wegen Crosstalk, Interferenz, Komponentenfehler
  - Entdeckung: Prüfsumme (BCC, CRC)
  - Reaktion:
    - Negative Quittung
    - Time out und Wiederholung
    - Error Correcting Code
- Verlust
  - Gründe:
    - Zerstörung der Adressen
    - Wegwurf durch Netz bei Überlast
    - Statusverlust von Komponenten
  - Entdeckung:
    - keine Quittung
    - Sequenznummern
  - Reaktion: Wiederholung

- Duplikat
  - Gründe:
    - Verlust einer Quittung
    - Benutzen alternativer Wege
    - Restart bei unvollständigem Status
    - zu frühe Wiederholung
  - Entdeckung: Sequenznummern
  - Reaktion: Vernichten
- falsche Reihenfolge
  - Gründe:
    - Wiederholung
    - Alternative Wege
  - Entdeckung: Sequenznummern
  - Reaktion
    - Wegwerfen und Wiederholen
    - Zwischenspeichern und Korrigieren
- Fehlzustellung
  - Grund: Verfälschung Adresse
  - Entdeckung: Prüfsumme
  - Reaktion: Wiederholung

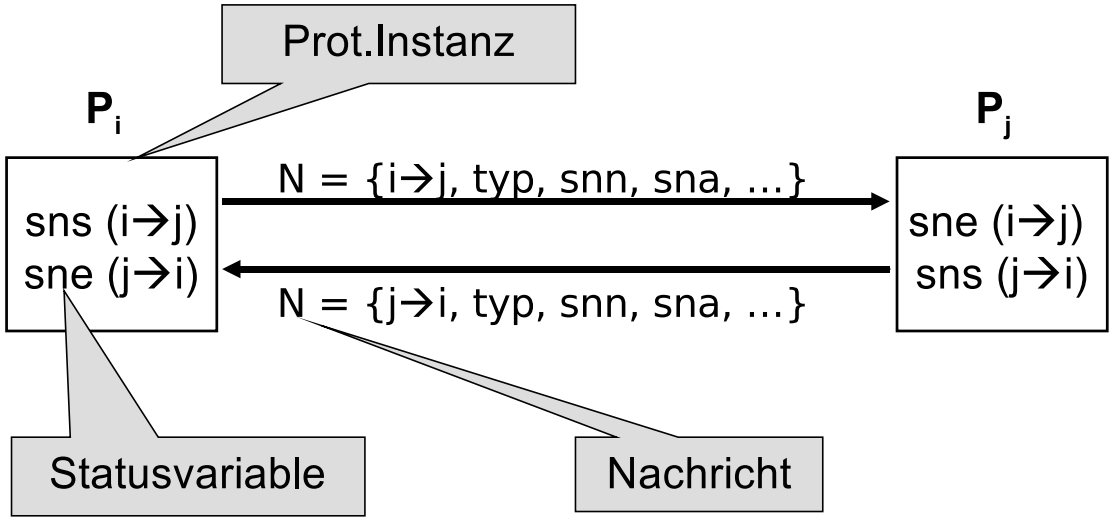
3.3 Sequenznummern

Naiver Ansatz, Motivation

- Ziel: Lösungsbeitrag zu Verlust- u. Duplikatserkennung, Reihenfolgesicherung und Flusssteuerung
- Ansatz: Jeder Partner nummeriert seine Nachrichten fortlaufend und verbindungsbezogen



3.3 Sequenznummern
Instanzvariablen



- N = $\{i \rightarrow j, typ, snn, sna, \ldots, ud\}$ mit
  - i→j: Verbindungs- und Richtungskennung
    - typ ∈ NN, ACK, NAK, SYN, CLS,...
  - snn : Sequenz einer Nachricht
  - sna : Nr der nächsten erwarteten Nachricht
  - sns (i→j): Nr der nächsten zu sendenden Nachricht
  - sne (j→i): Nr der nächsten erwarteten Nachricht
- NN normale Nachricht, ACK pos. Quittung, NAK neg. Quittung, SYN Verbindungswunsch, CLS Verb.-Abbauwunsch

Varianten von Quittungen

- Positive/negative Quittungen für einzelne PDUs
- Sammelquittung
  - einzelne Quittung für mehrere aufeinander folgende PDUs

  → geringere Anzahl Quittungen
- Selektive Quittung
  - für eine Teilmenge bisher gesendeter PDUs
  - ermöglicht selektive Wiederholung

Wiederholung

- Go-Back-N
  - wenn PDU mit snn=N verloren
  - Wiederholung aller PDUs ab snn=N
- Selektive Wiederholung
  - Wiederholung einzelner nicht quittierten PDUs
  - Leistungsvorteil bei großen Sequenznummernräumen (z.B. bei hohem RTD)

N von P_j treffe bei P_i ein

Fall A:

- snn = sne (j→i) ⇒es fehlt keine Nachricht
- P_i führt aus:
  1. sne (j→i) += 1
  2. Quittungen an P_j mit sna = sne (j→i)
  3. Nachrichtenverarbeitung

Fall B:

- snn > sne (j→i) ⇒es fehlen Nachrichten
- P_i führt aus:
  1. neg. Quittungen an P_j mit sna = sne (j→i)
  2. Nachrichten vernichten

Hinweise:

- Sender behält Nachricht bis positiv quittiert
- Erster Ansatz berücksichtigt nicht alle Fehlerfälle

Erster Ansatz berücksichtigt nicht:

- Duplikate
- folgende Verluste:

1. Erste Nachricht geht verloren
   - bekanntes Senderaster →Empfänger moniert
   - Wiederholung →Sender moniert (Timer)
2. Positive Quittung geht verloren
   - Wiederholung Nachricht und
   - positive Quittung bei jeder korrekten Nachricht
3. Negative Quittung geht verloren
   - Wiederholung

Hinweis

positive Quittungen unverzichtbar; negative im Prinzip wohl

Empfängerseite: Unverfälschte N von Pj trifft bei Pi ein

- snn = sne (j→i) ⇒es fehlt keine Nachricht
- P_i führt aus:
  1. sne (j→i) += 1
  2. Quittungen an P_j mit sna = sne (j→i)
  3. Nachrichten verarbeiten
- snn > sne (j→i) ⇒es fehlt Nachricht
  - P_i ignoriert Nachricht
- snn < sne (j→i) ⇒Duplikat
  1. P_i sendet Quittung mit sna = snn (Quittungsduplikat) und
  2. ignoriert Nachricht

Senderseite: Quittung bzgl. Nachricht mit Nr sna trifft ein bei P_j

- P_j hat Nachricht mit Nr sna noch nicht gespeichert ⇒erste angekommene Quittung

→ P_j löscht erfolgreich übertragene Nachricht im Puffer

- P_j hat keine Nachricht mit Nr sna ⇒Quittungsduplikat

→ P_j ignoriert die Nachricht

- In Praxis sind alle Sequenznummernräume **endlich** (modulo M)

$\Rightarrow$ Problem der Eindeutigkeit und Duplikaterkennung.

Beispiel 1: Größe von M abhängig von Senderate und RTD

Sei RTD = 15s und Senderate 200/s, dann werden 12 bits zur Codierung von snn und sna benötigt

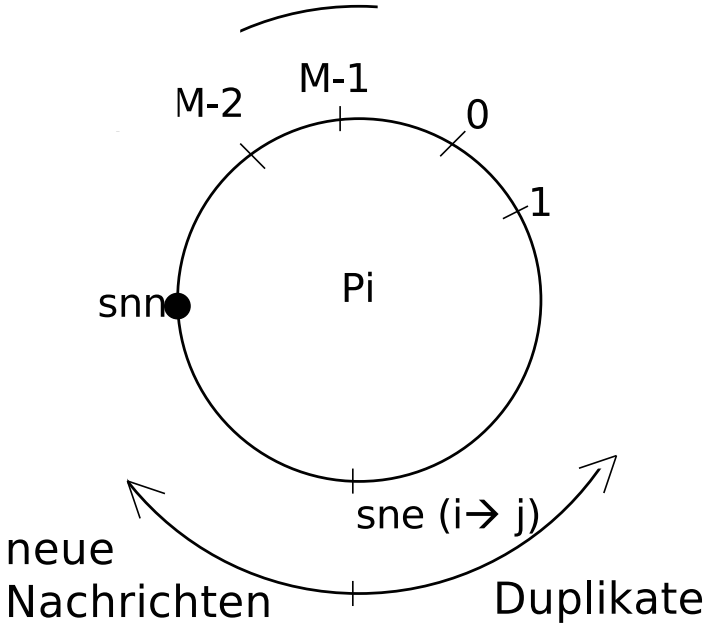
Beispiel 2

- Max. Zeichenrate $\leq$ M $\cdot L_{max}/(RTD + \Delta u)$, wobei
- L_{max} = max. Zeichenanzahl einer Nachricht
- Δu = Uhrendrift, Sicherheitszuschlag

Lösungsansatz zur Eindeutigkeit

- Jede Nachricht enthält Zeitpunkt ihrer Absendung
- Jede Nachricht hat maximale Lebensdauer
- Sender vergibt dieselbe Sequenznr erst nach Ablauf der Lebensdauer

- Problem: Ist snn neu oder Duplikat?
- Ansatz: Fenstertechnik



Erweiterung der Statusvariablen (hier in P_i)

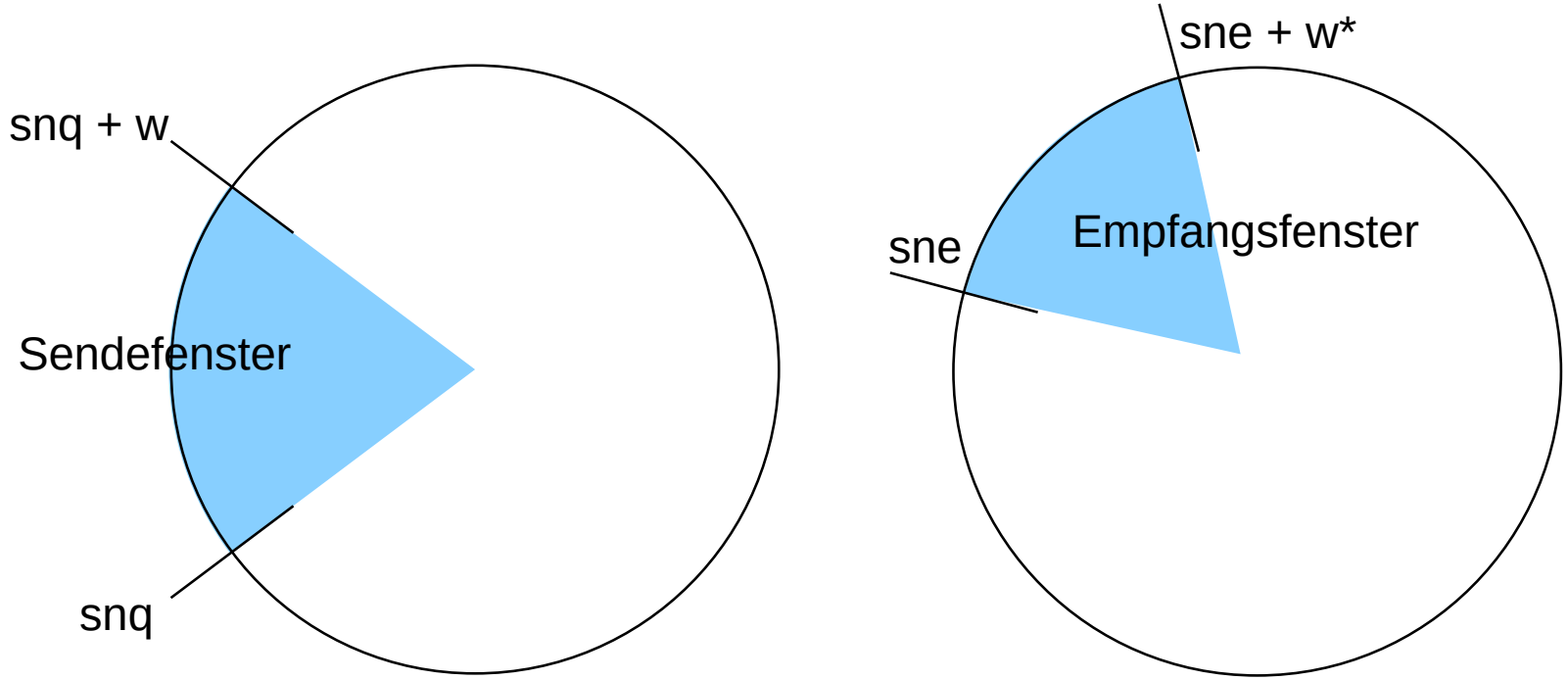
- sns (i→j), sne (j→i) : wie bisher
- snq (i→j): letzte pos. quittierte Nr (wird von P_j mitgeteilt)
- w (i→j): momentan gültiges Sendefenster (bestimmt von P_j)
- w* (j→i): momentan gültiges Empfangsfenster (festgelegt von P_i)

Konventionen

- P_i darf Nachrichten senden im Bereich
  snq (i→j) bis (snq (i→j) + w (i→j)) mod M
- P_j interpretiert Nachrichten im Bereich
  sne (j→i) bis (sne (j→i) + w* (j→i)) mod M als neu,
  sonst als Duplikat

3.3 Sequenznummern

Fenstertechnik: Ablauf



- Lage Sendefenster und Empfangsfenster i.a. verschieden
- $snq \leq sne \leq snq + w$
- w und w* können differieren
- Fenstertechnik impliziert Flusssteuerung!
- Sequenznr i.a. schichtspezifisch eingesetzt (Ausnahme bei zeichenorientierter Nummerierung bei TCP/IP)

Rechnernetze und verteilte Systeme
Schichtunabhängige Konzepte

Erkennung von Bit-Fehlern

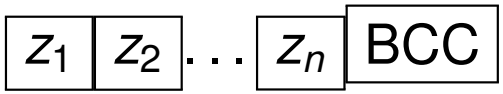
Kapitel 3.4

- Exklusives OR, Summe 0 = gerade Parität, sonst ungerade
- Ein- und Zweibitfehler erkennbar.
- Einbitfehler korrigierbar.
- Weitere Fehlermuster nur bedingt erkennbar

Beispiel ASCII (7-Bit-Code)

zu übertragen (7 Bits)	gerade (8 Bits)	ungerade (8 Bits)
0000000	00000000	00000001
1111111	11111111	11111110

- Nachricht besteht aus Folge von Zeichen aus m Bits
- Übertragen wird um Parität erweiterte Zeichenfolge und Prüfzeichen BCC



	z_1	=	$b_{1,1}$	$b_{1,2}$	$\cdots$	$b_{1,m}$	$b_{1,m+1}$ ← Querparität
	z_2	=	$b_{2,1}$	$b_{2,2}$	$\cdots$	$b_{2,m}$	$b_{2,m+1}$
		$\vdots$					
	z_n	=	$b_{n,1}$	$b_{n,2}$	$\cdots$	$b_{n,m}$	$b_{n,m+1}$
BCC =	z_{n+1}	=	$b_{n+1,1}$	$b_{n+1,2}$	$\cdots$	$b_{n+1,m}$	$b_{n+1,m+1}$

↑ Längs-parität

- $b_{n+1,m+1}$ ist Spezialfall
- Art der Parität muss vereinbart werden

- CRC: Cyclic Redundancy Check
  - Sender und Empfänger vereinbaren Generatorpolynom G vom Grad r
  - Bitfolgen werden als Binärpolynome aufgefasst mit Arithmetik XOR
  - Prüfsequenz wird mittels G berechnet, dann übertragen und geprüft
- Übertragungsfolge U ist Sendefolge S gefolgt von Prüfsequenz Q

$U =$ | Sendefolge (Nutzdaten) | Prüfsequenz (r Bits) |

$U = S \cdot 2^r \oplus Q$

wobei Q der Divisionsrest von $x^r \cdot S/G$ ist.

Beispiele für vereinbarte Generatorpolynome

Bei HDLC: $G = x^{16} + x^{12} + x^5 + 1$

Bei Ethernet:

$G = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

In der Praxis, oft: CRC als rückgekoppelte Schieberegisterschaltung in HW realisiert

Rechenbeispiel: Senderseite

- Sei $G = x^5 + x^4 + x^2 + 1$ ($\equiv 110101$)
- Sei $S = x^9 + x^5 + x^2 + 1$ ($\equiv 1000100101$)

Berechne: $x^5 \cdot S/G = Q = x + 1$; somit U = 1000100101**00011**

Empfängerseite

Empfänger bildet U/G

- Falls Division $\neq 0$, dann Fehler
- Falls Division ohne Rest, dann fehlerfrei
  - oder Fehler ist Vielfaches von G, d.h. nicht erkennbar
- Fehlerhafte Sendung: H = U + F mit F Fehlerpolynom
- Ziel: G so wählen, dass $F \neq 0$ für gängige Fehlermuster

Einbitfehler

$F = 2^k$ (das Bit k+1 ist verfälscht).
F ist nicht teilbar durch alle G mit mehr als einem Term, z.B. $G = x^P + 1$

Zweibitfehler

$F = 2^i + 2^j = (2^{i-j} + 1) \cdot 2^j$
2^j nicht durch $x^P + 1$ teilbar. Auch $2^{i-j} + 1$ darf nicht durch G teilbar sein!
Beispiel: $G = x^{15} + x^{14} + 1$ teilt nicht bis $i - j = 32768$, also werden alle U bis zu dieser Länge geschützt

Bündelfehler

Bündelfehler haben Restfehlerwahrscheinlichkeit 2^{-r}

Beispiel HDLC

- HDLC hat $G = x^{16} + x^{12} + x^5 + 1$, entdeckt
- alle Ein- und Zweibitfehler
- alle Fehler mit ungerader Zahl von Fehlerbits
- alle Bündelfehler der Länge bis 16
- 99,997% der 17er und 99,998% der 18er Bündel

Aufgabe

Zeige, dass die Restfehlerwahrscheinlichkeit für Fehlerbündel der Länge $b > r$ die Größenordnung 2^{-r} hat. r sei Grad des Generatorpolynoms.

- Annahmen:
  - Es werden 24 Prüfbits CRC verwendet
  - Paketlänge sei 1000 Bits
  - Im Mittel werden pro Tag 10^7 Pakete ins Netz gegeben
  - Ein Paket läuft im Mittel über 2,3 Leitungen
  - Die Bitfehlerrate (Leitungsqualität) sei 10^{-6}
  - Wahrscheinlichkeit, dass ein Fehler in einer Nachricht auftritt, sei gleichgroß wie die eines Fehlerbündels
- Aufgabe:
  - Wie viele Pakete pro Jahr werden fälschlicherweise als richtig angesehen?

3.4 Erkennung von Bit-Fehlern

Selbstkorrigierende Codes (1)

- Idee: Codetabelle wird so dünn besetzt, dass Verfälschung zu unzulässiger Codierung führt. Die Fehler- lokalisierung ermöglicht Korrektur.
- Hammingabstand von Codewörtern: Anzahl der verschiedenen Bitpositionen
- Beispiel:
  `1011001` und
  `0011100` haben Abstand 3
- Hammingabstand von Codes: Minimum der Distanzen aller Paare von Codewörtern.

Aufgabe: Code für Korrektur von Einbitfehlern

- Betrachte Codewörter $n = b + r$ Bits
  (b mit Bedeutung, r Redundanz)
- es gibt 2^b Nutzzeichen in n-Bit-Wörtern
- jedes Codewort hat n Nachbarn mit Distanz 1
  - jedes Nutzzeichen lässt sich mit Nachbarn zu n+1 Bitmustern fassen
  - für 1 Bit–Korrektur müssen all diese Bitfolgen für jedes Nutzzeichen disjunkt sein
  - es gibt 2^n n-Bit-Wörter
- $(n+1) \cdot 2^b \leq 2^n \Rightarrow (b + r + 1) \leq 2^r$

Beispiel ASCII

für ASCII gibt b = 7 $\Rightarrow$r = 4
$\Rightarrow$11 Bits erforderlich, um das Nutzzeichen + Redundanz zu repräsentieren
$\Rightarrow$36% Overhead (Vergleich Paritätsbit: 12,5% Overhead)

Beispiel: b = 4 und r = 3, also n = 7

- Prüfbits auf Positionen 1,2 4, Informationsbits auf 3,5,6,7

P	P	I	P	I	I	I

  - Prüfbit 1 sichert ab 1,3,5,7, Prüfbit gerade Parität
  - Prüfbit 2 sichert ab 2,3,6,7
  - Prüfbit 3 sichert ab 4,5,6,7
- Empfänger hat Zähler mit Initialwert 0
  - Bildet Parität über zu sichernde Bits
  - Falls Fehler, wird Prüfbitposition auf Zähler addiert
  - Zähler enthält Bitposition des falschen Bits

- Verfahren ausdehnbar auf Mehrbitfehler
- Bei Distanz $h = 2 \cdot k - 1$ oder $h = 2k$ auf k Bits korrigierbar

Rechnernetze und verteilte Systeme
Schichtunabhängige Konzepte

Verbindungsmanagement

Kapitel 3.5

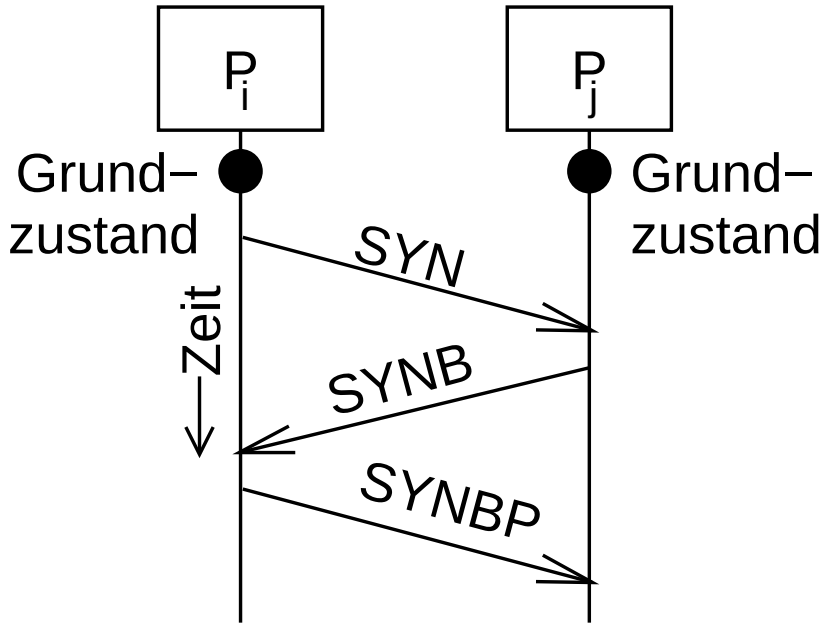
- Verbindungsaufbau (bei verbindungsorientierten Protokollen)
  - Ressourcenreservierung
  - Statusvariable für Verbindungen z.B. für Sequenznummern
  - Timer (für Wiederholung, für Wiedervergabe Sequenznummern, Wiederholungszahl)
  - Protokollprofil, evtl. Aushandlung (*Negotiation*)
- Verbindungsabbau (abrupt, ohne Verlust)
- Multiplexing (upward, downward)

3.5 Verbindungsmanagement

3-way-handshaking (1)

Neue PCI-Typen

- SYN: Aufbauwunsch
- SYNB: Aufbauwunsch–Bestätigung
- SYNBP/N: pos/neg Bestätigung der Bestätigung

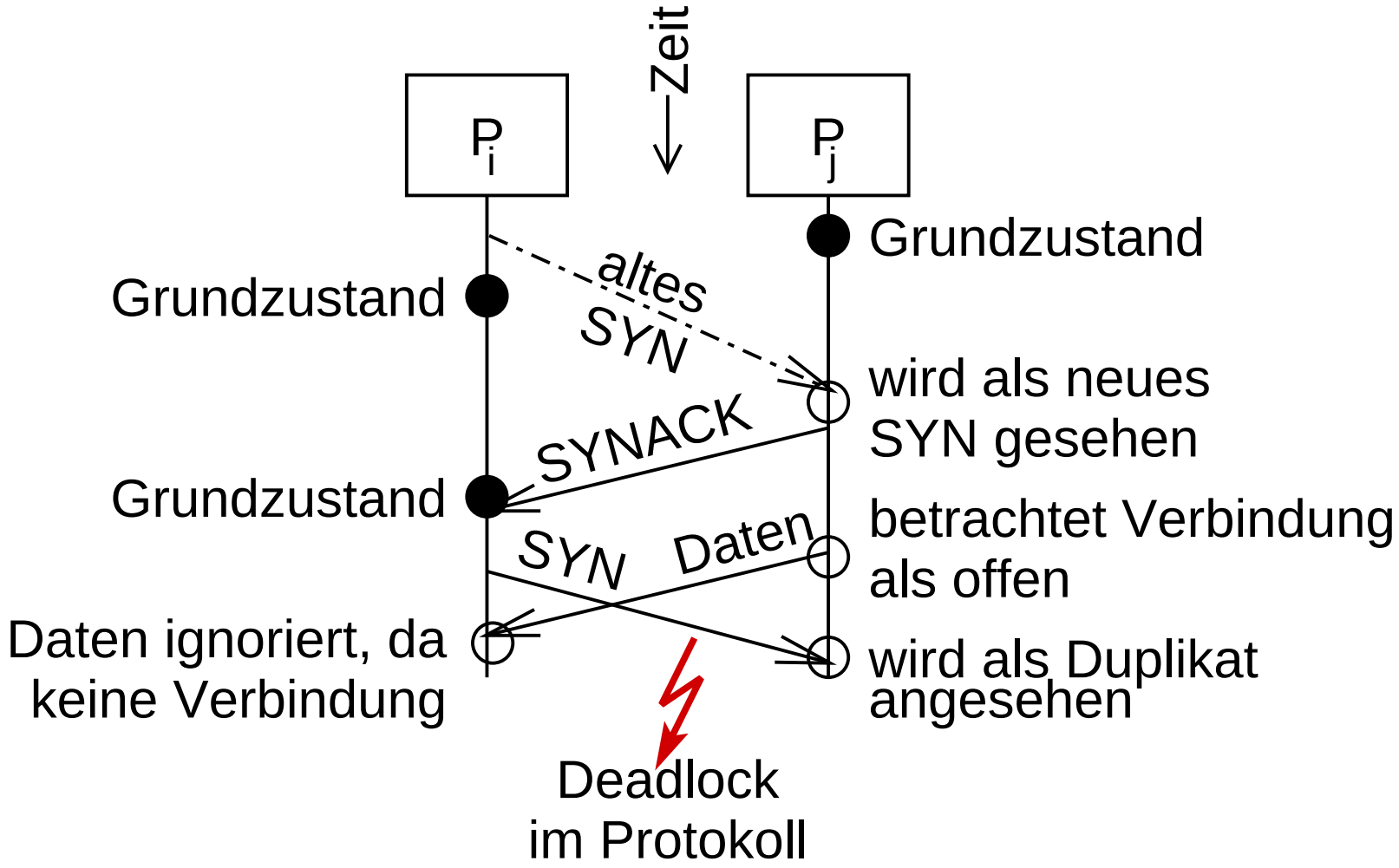


1. P_i sendet SYN und Initialwert für sns (i→j)
2. P_j quittiert mit SYNB und sendet Initialwert für sns (j→i) „halboffene" Verbindung
3. P_i quittiert SYNB, im Fall SYNBP steht komplette Verbindung offen

Aufgabe

Beschreibe ein vollständiges Zustandsübergangsdiagramm für das 3-way-handshaking (einschl. Kollisionsfall).
Frage: Wann reicht 2-way-handshake aus?

- Ein 2-way-handshake (nur SYN, SYNACK) reicht u.U. nicht aus



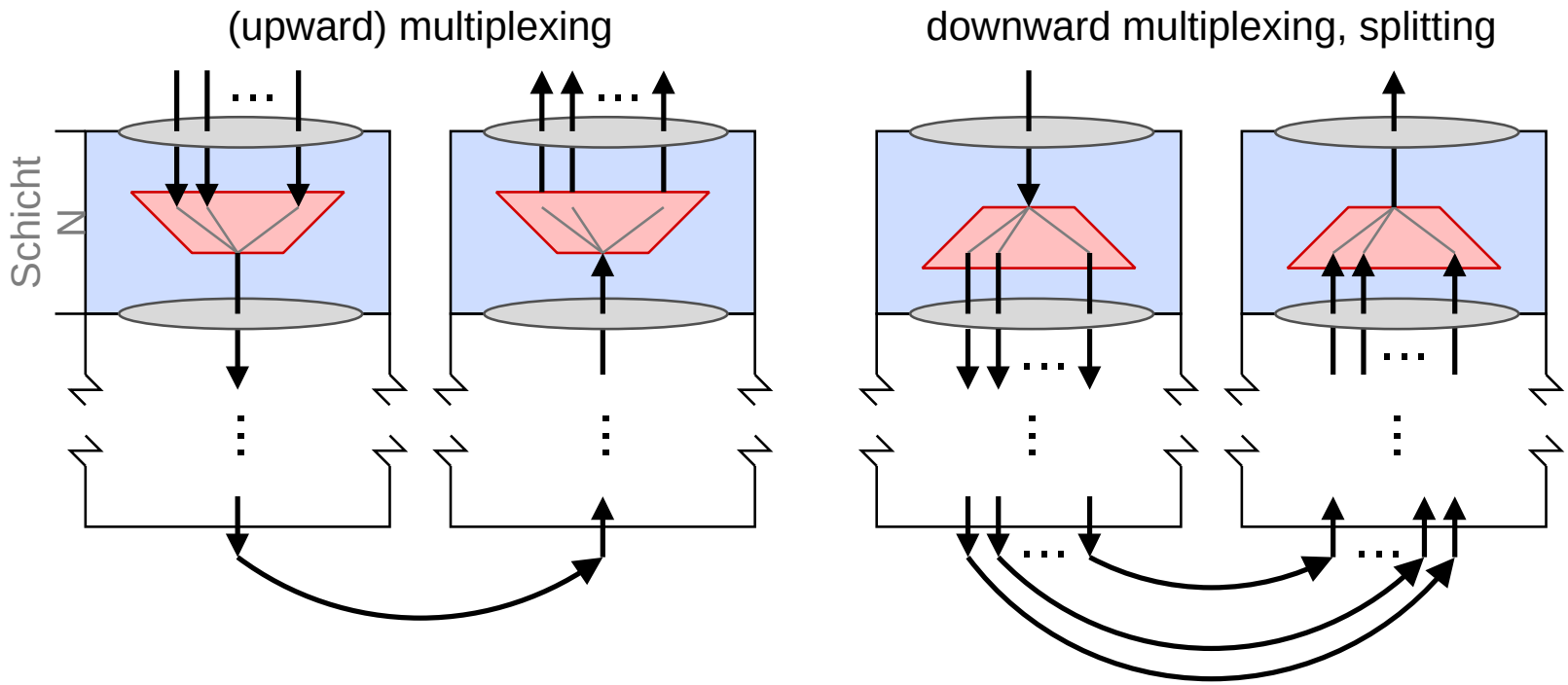
Neue PCI-Typen

- CLS ("close") Abbauwunsch
- CLSB Abbauwunsch-Bestätigung

- CLS und CLSB muss von beiden Partnern gegeben werden, es werden also beide „Halbverbindungen” explizit geschlossen.
- Aufgabe: Entwerfe ein vollständiges Zustandsübergangsdiagramm einschließlich Fehlerfälle!

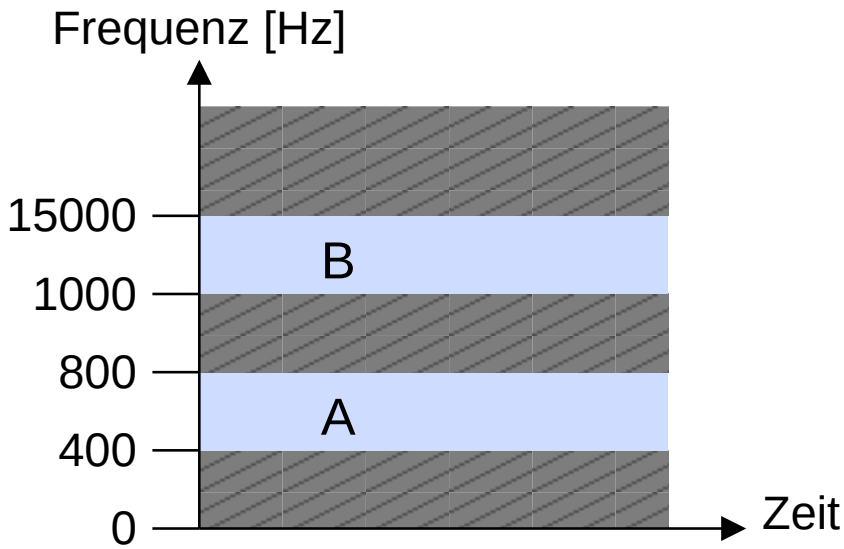
Multiplexing (oft: „Mux")

Abbilden von N-Connections auf (N-1)-Connections in Schicht N.



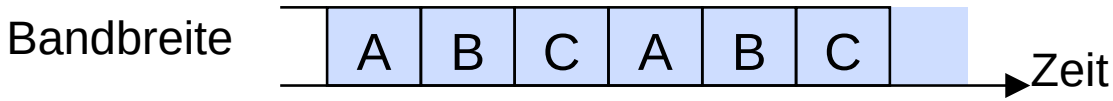
	multiplexing	*splitting*
Motive	viele logische Verbindungen; nur ein Kanal “economy of scale”	mehr Grenzdurchsatz Ausfallsicherheit
Protokoll-funktion	Vereinigen und Aufteilen Pufferung Vergabestrategie erweiterte Formate (PDU)	Aufteilen und Vereinigen Reihenfolgesicherung, Pufferung erweiterte Formate (PDU)

- Frequenzmultiplex (Auf Ebenen 1 bzw. 2) (frequency division multiplex, FDM)
- Frequenzspektrum des Multiplexkanals wird in schmalere Frequenzbänder für Subkanäle aufgeteilt und fest zugeordnet
- z.B. Subkanal A: „0” = 725Hz, „1” = 475 Hz
- B: „0” = 1325 Hz, „1” = 1075 Hz



Festes Zeitmultiplex (time division multiplex, TDM)

Übertragungskapazität des Multiplexkanals wird Subkanälen periodisch für festen Zeitslot zur Verfügung gestellt.



Framing: oft Bit oder Byte Synchronisation über SYNC-Folgen oder gute Uhren; dann kein Puffern oder Mux-Protokoll

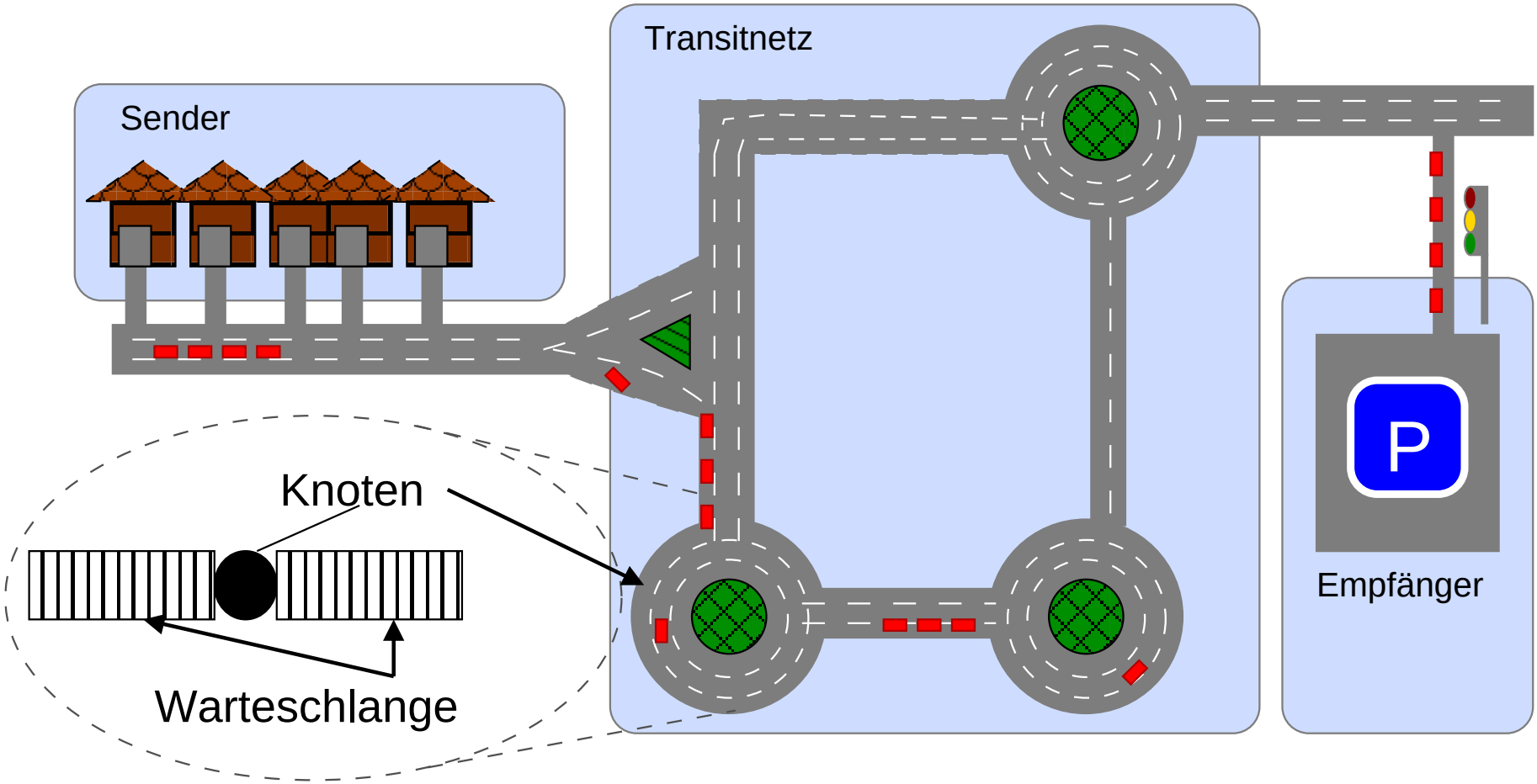
Statistisches Zeitmultiplex

- Kanalzuteilung erfolgt nicht periodisch, sondern verkehrsaufkommen-gesteuert
- keine feste Teilkanalzuordnung
- Mux-Protokoll enthält Steuerinformation zur Sekundärkanalzuordnung
- Pufferung im Mux üblich, Flusssteuertechniken üblich
- Vielfachzugriffsprotokolle der Schicht 2a (MAC-Layer) realisieren stochastisches Zeitmultiplex

Rechnernetze und verteilte Systeme
Schichtunabhängige Konzepte

Fluss und Stau

Kapitel 3.6


Transitnetz
Sender
Knoten
Warteschlange
P
Empfänger

- In technischen Systemen sind Warteschlangen begrenzt

$\Rightarrow$ somit Rückstaumechanismen erforderlich

  - für Verbindung, für Interface
  - für Netz

Flusssteuerung (flow control)

Maßnahmen, die einer Überlastung eines Empfängers durch einen Sender vorbeugen

Staukontrolle (congestion control)

Maßnahmen, die einer Überlastung des Netzes durch eine statistisch relevante Menge von individuellen Komm.-Beziehungen vorbeugen

- virtuelle Kanäle: sicher, aber Puffervergeudung und Aufbauzeit
- Begrenzung des Nachrichtenflusses in das Netz
  - Begrenzung der Anzahl von Verbindungen pro Host/Prozess
  - Begrenzung der Nachrichtenrate (traffic contract, leaky bucket)
- Sicherstellung des Abflusses aus dem Netz
  - garantierte Abnahmerate durch Empfänger
  - Reassembly im Host
- Wegwurf von Paketen bei Überlast (z.B. im Internet)
- Konstante Last im Netz (Taxi-Verfahren, nicht optimal)
- Füllungsregelung pro Verbindung

- Sender verlangt Pufferreservierung vor Sendung, Senden nach Bestätigung →zusätzliche Nachrichten und Verzögerung
- Empfänger stellt Credit über Allokationsnachrichten (z.B. TCP) →Verfahren nicht robust
- Fenstertechnik (mit festem oder dynamischem Fenster) →Problem bei Fensterverkleinerung
- Stop and Go–Technik →Oszillierendes Verhalten, schlecht bei großem RTD
- Zeitrasterabhängiges Senden

Fragen zu Kapitel 3

- Bewertung von flachen und strukturierten Adressräumen?
- Was sind Einflussgrößen für die Größe des Sequenznummernraumes?
- Wozu kann Fenstertechnik eingesetzt werden?
- Wie wird Eindeutigkeit der Sequenznummern sicher gestellt?
- Wie wirken sich zu kleine Fenster aus, wie zu kleine Sequenznummernräume?
- Warum bleibt trotz CRC die Restfehlerwahrscheinlichkeit größer 0?
- Was versteht man unter Hammingabstand?
- Warum wird BCC nicht bei bitorientierten Prozeduren eingesetzt?

Fragen zu Kapitel 3

- Unter welchen Voraussetzungen genügt ein 2-way-handshake?
- Warum ist für einen sicheren Verb.-Aufbau auf einer Transportschicht i.a. ein 3-way-handshake erforderlich?
- Warum sollte i.a. sinnvollerweise ein Verbindungsabbau beidseitig geschehen?
- Ein Protokoll soll splitting beherrschen. Was muss berücksichtigt werden?
- Wo liegt bei FDM und TDM der Unterschied im Hinblick auf die Bandbreite des Multiplexkanals?
- Unterschied zwischen flow control und congestion control?
- Nennen Sie mindestens drei Verfahren der Flusssteuerung.