

# Rechnernetze und verteilte Systeme

## Vermittlungsschicht

### Kapitel 7

# Rechnernetze und verteilte Systeme

## Vermittlungsschicht

### Grundlagen

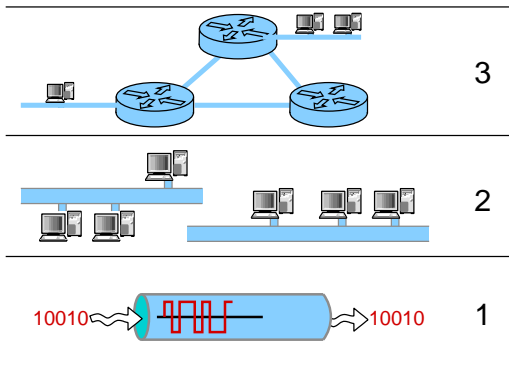
#### Kapitel 7.1

- Pfadschaltung (Vermittlung) zwischen zwei Endsystemen unter Berücksichtigung von Transitsystemen und Transitnetzen auf Basis eines netzglobalen Adressraumes
- Wegewahl (Routing)
- Dienstgüte verhandeln
- Bereitstellung eines verbindungslosen oder verbindungsorientierten Netzdienstes

*Hinweis:* Wegewahl und Vermittlung können auch technologieabhängig auf anderen Schichten vorkommen, so z.B. auf Ebene 7 (E-Mail-Relays), Ebene 2a (Bridges im MAC-Layer), Zellvermittlung bei ATM

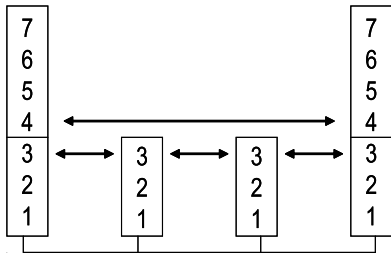
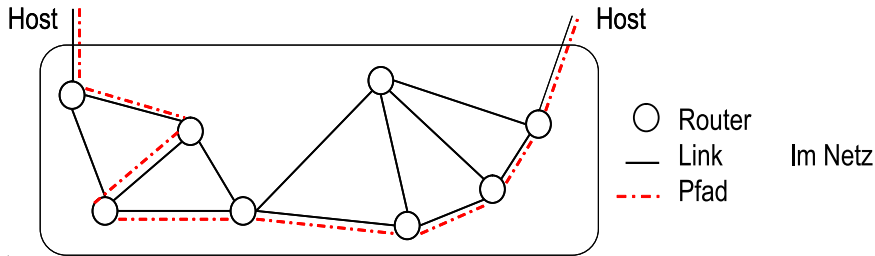
# 7.1 Grundlagen

## Netzbildung



# 7.1 Grundlagen

## Pfadvermittlung



in OSI-Architektur

# 7.1 Grundlagen

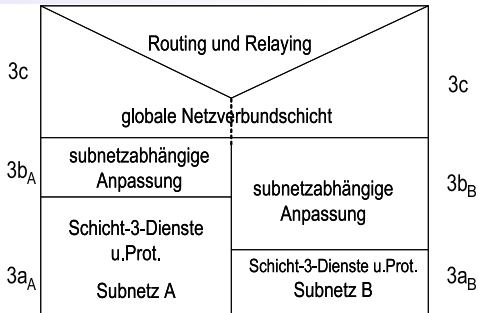
## QoS-Parameter Schicht 3

- Verbindungsaufbauwahrscheinlichkeit (Blockierwahrscheinlichkeit)
- Verbindungsaufbauzeit
- Durchsatz der Schicht-3-Verbindung
- Nachrichtenübertragungszeit
- Schwankung in der Übertragungszeit (Jitter)
- Restfehlerrate

# 7.1 Grundlagen

Interworking Unit (IWU)

Struktur einer IWU  
(Router, Gateway)  
zwischen Subnetz A  
und B

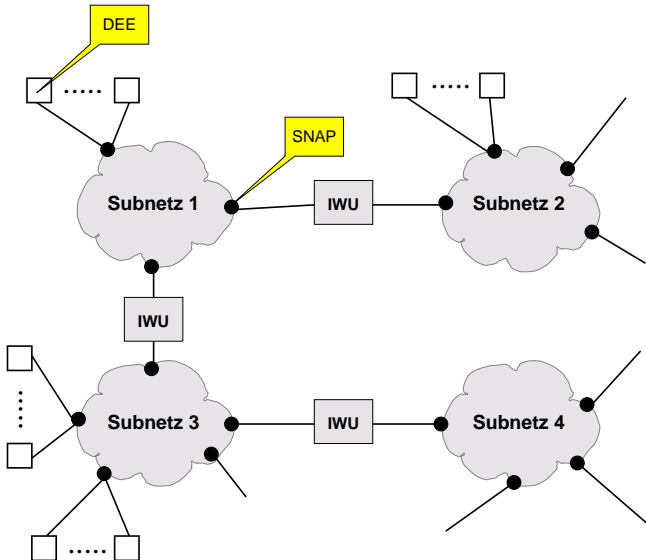


## Aufgaben einer IWU

- Adressierung über (Sub-)Netzgrenzen hinweg; Adressabbildung
- Anpassung Nachrichten bzgl. PDU-Struktur, Länge u.ä.
- Abbildung von Diensten (VO, VL) und Dienstgüteparametern
- Abbilden von Protokollparametern (z.B. Fenster, Timer)
- Anpassen von Fehlerbehandlungs- u. Meldemechanismen
- Globale Wegewahl

# 7.1 Grundlagen

## Subnetze





# Rechnernetze und verteilte Systeme

## Vermittlungsschicht

### Vermittlungsverfahren

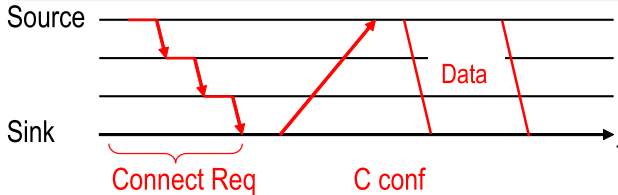
#### Kapitel 7.2

# 7.2 Vermittlungsverfahren

## Leitungsvermittlung

### Leitungsvermittlung, Durchschaltung, Circuit Switch

- Vor Übertragung Aufbau des Pfades von Sender zu Empfänger
- Dedizierter Kanal für die gesamte Datensphase
- typisch ist gleicher Grenzdurchsatz für die Links
- typisch gekoppelt mit VO-Dienst
- Beispiel: POTS, ISDN-B-Kanäle

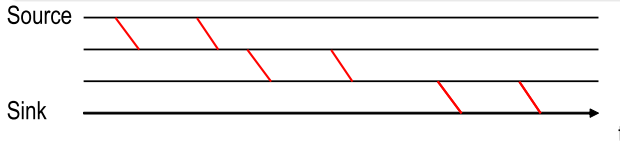


# 7.2 Vermittlungsverfahren

## Nachrichtenvermittlung

### Nachrichtenvermittlung, Message Switching, Store and forward, Speichervermittlung

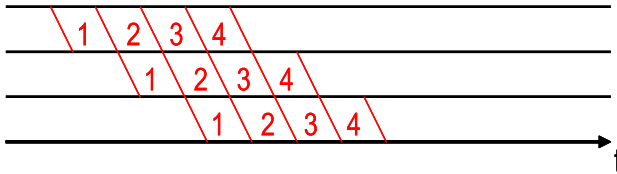
- Nachricht wird als Ganzes versendet
- Jeder Knoten auf Weg speichert Nachricht und sendet dann auf nächstes Streckenstück
- Streckenstücke sind i.a. nicht homogen
- Schwankendes Processing und Queuing Delay
- Beispiel: Email, IP



## 7.2 Vermittlungsverfahren

Paketvermittlung, Packet switching

- Zerlegung der Nachrichten in etwa gleichlange Pakete.
- Senden der Nachrichten paketweise nach Prinzip Message Switching
- ⇒ Eine Art “Pipelining Effekt”, bessere Leitungsauslastung
- ⇒ Reihenfolgeproblem
- VO und VL möglich
- Beispiel: X.25, ATM - Zellvermittlung



## 7.2 Vermittlungsverfahren

Vermittlung: Fragen

Bewerten Sie die 3 Vermittlungsverfahren in Hinblick auf...

- Verzögerung bei der Pfadherstellung
- Durchsatz für lange Nachrichten
- Antwortzeiten für kurze Nachrichten
- Schwankung in Durchsatz und Antwortzeit
- Möglichkeit der Code-, Format- u. Ratenanpassung bei Übertragung
- Laufwegermittlung und Ü-Steuerung
- Reihenfolgesicherung
- Eignung für Sprachübertragung

Welches ist die "beste" Technik?

## 7.2 Vermittlungsverfahren

Vergleich Paket- u. Nachrichtenvermittlung

- $L_D$  Nutzdatenlänge zerlegt in  $p$  Pakete
- $L_K$  Headerlänge (sei gleich bei beiden Verfahren),
- $\delta = \frac{L_D}{L_K}$  Verhältnis Nutzdaten- zu Headerlänge
- $T^N, T^P$ : Zeiten für Nachrichten-/Paketvermittlung
- Weg bestehe aus  $n$  Leitungen (seien gleich in Länge und Rate  $C$ )

$$\text{Dann : } T^N(n) = n \cdot \frac{L_D + L_K}{C}, \text{ ferner : } T^P(1) = \frac{\frac{L_D}{p} + L_K}{C}$$

$$\text{Somit : } T^P(n) = n \cdot T^P(1) + (p - 1) \cdot T^P(1) = \frac{L_D + p \cdot L_K}{C} \cdot \frac{n + p - 1}{p}$$

$$\text{Verbesserung : } \alpha(\delta, n, p) = \frac{T^P(n)}{T^N(n)} = \frac{\delta + p}{\delta + 1} \cdot \frac{n + p - 1}{n \cdot p}$$

Gewinn wächst mit größerem  $\delta$  und  $n$ , wird optimal bei

$$p = \sqrt{\delta \cdot (n + 1)}$$

# Rechnernetze und verteilte Systeme

## Vermittlungsschicht

### Wegewahl, Routing

#### Kapitel 7.3

# 7.3 Wegewahl, Routing

## Begriffe und Probleme

### Begriffe

- Routingalgorithmen beschreiben Wegewahlverfahren
- Verfahrensauswahl und -ausprägung hängt ab von der Routingstrategie (Policy), also von den *Zielfunktionen*(„Kostenfunktionen“), z.B.
  - geringe Ü-Kosten,
  - geringe Ü-Zeiten,
  - gute Leitungsauslastung,
  - großer Durchsatz
- Verfahren soll sein
  - einfach (Alg.-Komplexität, Netzoverhead),
  - adaptiv (Last, Topologie),
  - robust (bei Fehlern), fair
- Grundlage ist Routing-Tabelle

### Probleme

- Zielkonflikte
- Beschreibung der Topologie
  - wie beschrieben (Leitungen, Knoten, Kosten)
  - vollständig / partiell
- Berechnung
  - wo (zentral / dezentral)
  - welche Info wird vom Algorithmus benötigt
  - wie wird Info bereitgestellt
  - welche Ereignisse stoßen Berechnung an
  - wann wird neuer Weg aktiviert



# 7.3 Wegewahl, Routing

## Routing-Tabelle

- realisiert Vorgabe zur Weitergabe von Nachrichten
- bestimmt die „Entscheidungsfindung“ in IWUs/Routern
- Default Route: spezieller Eintrag; wird benutzt, wenn kein anderer (spezifischerer) Eintrag passt.

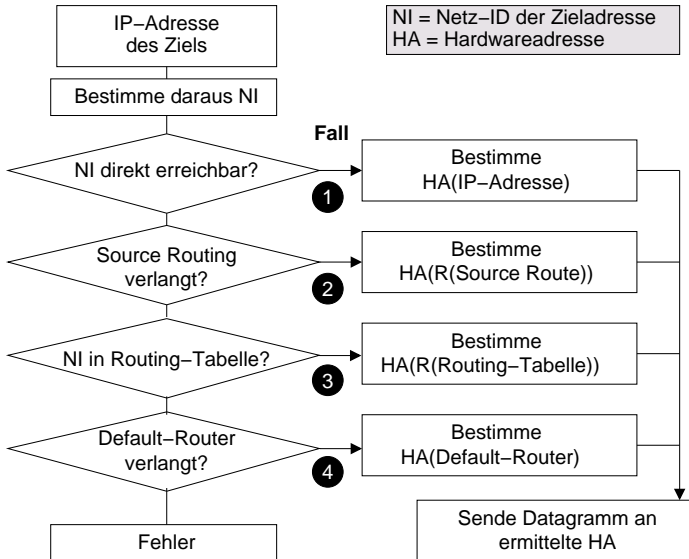
Felder eines Tabelleneintrags (eine Zeile):

- Ziel (Netz oder Host)
- Gateway (d.h. Router)
- Netzmaske (Angabe zur Bestimmung der Netzadresse)
- Metrik (Kostenwert)
- Schnittstelle/Iface (Zielleitung)

Auf Unix-Derivaten (z.B. Linux): `$ route -n`

# 7.3 Wegewahl, Routing

Wegewahlentscheidung im Router — Ablauf



# 7.3 Wegewahl, Routing

Weiteres

- Zu unterscheiden:
  - Wegewahl (route discovery) und
  - Weitergabe (forwarding)

## Routing...

- abhängig von Kommunikationsbeziehung
- meistens Unicast für 1:1 - Beziehung
- Routing für Multicast (Gruppenkommunikation, 1:n, m:n) wird zunehmend wichtiger (conferencing, Videosever, CSCW<sup>a</sup>)
- Routing für Broadcast muss für NBMA<sup>b</sup>-Netze emuliert werden
- Routing vereinfacht für spezielle Topologien (Stern, Bus, Ring, Baum, vollständige Vermaschung)

---

<sup>a</sup>CSCW = computer supported cooperative work

<sup>b</sup>NBMA = non-broadcast multiple access

## 7.3 Wegewahl, Routing

### Wegkosten

- $K$  Knotenmenge,  $L$  Leitungsmenge
- $A(j) = \{i \in K \mid (j, i) \in L\}$  Nachbarknoten zu  $j$
- $W_{qz,f}$ :  $f$ -ter Weg von Quelle  $q$  zum Ziel  $z$
- $D_{qz}(W_{qz,f})$ : Kosten für  $f$ -ten Weg von  $q$  und  $z$

### Optimalitätseigenschaft der Wege

- Falls

$$D_{qz}(W_{qz,f}) = \sum_{(i,j) \in W_{qz,f}} D_{ij}(W_{ij})$$

(d.h. Wegkosten sind Summenkosten über Teilwege)

- dann

$$D_{qz}^{opt} = D_{qx}^{opt} + D_{xz}^{opt}$$

für beliebiges  $x$  auf optimalem Weg

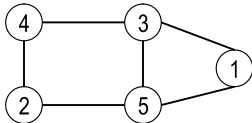
- Das heisst:

- jeder Teilweg optimal
- in  $x$  kann weiterer Weg optimal gesucht werden

# 7.3 Wegewahl, Routing

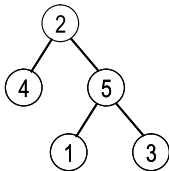
Quell-Senken-Baum (1): Prinzip

Netz

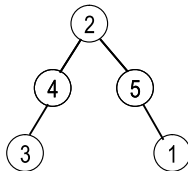


$$\forall(j,i):D_{ji}=10$$

QSB



oder



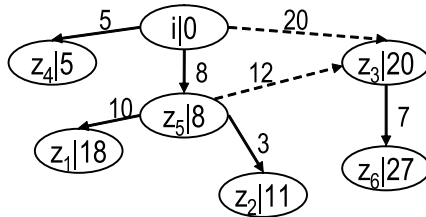
- in der Regel mehrere QSB zu einem gegebenen Netz
- Ziel Routingalgorithmen: QSB mit optimalen Wegen aufbauen und benutzen

## 7.3 Wegewahl, Routing

Quell-Senken-Baum (2): Wegetafel

Wegetafel Knoten i	Ziel	Nachbar	Kosten
$z_1$	$z_5$		18
$z_2$	$z_5$		11
$z_3$	$z_3, z_5$		20
$z_4$	$z_4$		5
$z_5$	$z_5$		8
$z_6$	$z_3, z_5$		27

Quell-Senken-Baum



# 7.3 Wegewahl, Routing

## Shortest Delay First Algorithmus (1)

### Idee

- Schrittweiser Aufbau des QSB von Wurzel her
- Alle noch nicht in QSB enthaltenen Wege müssen über Nachbarn von QSB Knoten führen (Kandidatenmenge)
- Aus Kandidatenmenge  $H$  wird Knoten mit minimalen Kosten ab Wurzel gewählt. Er kommt in QSB.

### Bezeichnung

- $A(j)$ ,  $D_{jk}$ ,  $q$  wie gehabt
- $H = \{j \in K \mid j \in A(k), \text{ wobei } k \text{ in QSB und } j \text{ nicht in QSB}\}$
- $D_{qj}^*$  momentane Kosten von  $q$  nach  $j$  während Algorithmus
- $V_j$  momentaner Vaterknoten von  $j$  zur Erreichung von  $D_{qj}^*$
- nächster Knoten ( $H$ ) liefert Knoten  $r$  mit  $D_{qx}^* \leq D_{qr}$  für alle  $r$  aus  $H$

Algorithmus läuft in jedem Knoten (hier in  $q$ )

## 7.3 Wegewahl, Routing

### Shortest Delay First Algorithmus (2)

#### Wurzel

$QSB := q;$

$H := \{ \};$

$D_{qq}^{opt} := 0$

$x := q$

$x$  bezeichnet zuletzt in  
QSB eingetragenen  
Knoten

**while** true **do**

*// Vermessen aller Nachbarn von  $x$*

**for all**  $j \in A(x)$  **do**

$d = D_{qx}^{opt} + D_{xj}$

**if**  $j \notin QSB \wedge j \notin H$  **then**

$H := H \cup \{j\}; D_{qj}^* = d; V_j := x;$

**end if**

**if**  $j \in H \wedge d < D_{qj}^*$  **then**

*// Es gibt besseren Weg, umhängen*

$D_{qj}^* = d; V_j := x;$

**else if**  $H = \{ \}$  **then**

**return;** *// Ende Algorithmus;*

**end if**

$x :=$  nächster Knoten ( $H$ );

$x$  in QSB aufnehmen als Sohn von  $V_x$  und

in  $H$  löschen;

$D_{qx}^{opt} := D_{qx}^*;$

**end for**

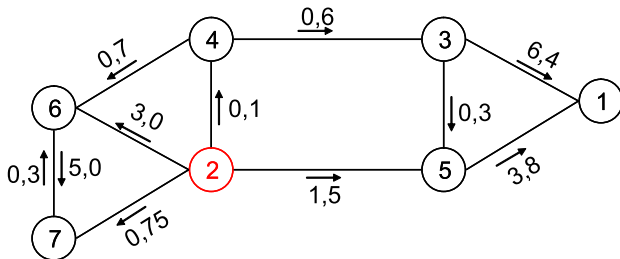
**end while**



## 7.3 Wegewahl, Routing

### Shortest Delay First Algorithmus (3)

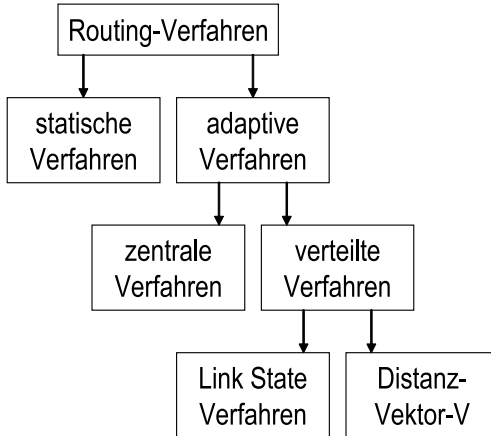
#### Beispiel



Aufgabe: Berechne QSB für  $q=2$

# 7.3 Wegewahl, Routing

## Routing-Verfahren



### • Weitere Verfahren

- isoliertes Verfahren, nicht adaptiv: Flooding
- isoliertes Verfahren, lastabhängig: Hot Potato

# 7.3 Wegewahl, Routing

Static Routing, Directory Routing

## Idee

Routingtabelle wird *einmal* aufgrund festgelegter Routing-Metriken erstellt

Ziel $z$	Nachbarknoten auf dem Weg $j \rightarrow z$					
	1. Wahl		2. Wahl		3. Wahl	
	Nr.	Gewicht	Nr.	Gewicht	Nr.	Gewicht
	1	0,63	6	0,21	4	0,16

- Berechnen Zufallszahl  $x$  aus  $[0,00 \text{ bis } 0,99]$ 
  - $x \leq 0,63 \rightarrow$  nimm 1. Wahl
  - $0,63 < x \leq 0,84 \rightarrow$  nimm 2. Wahl
  - $x > 0,84 \rightarrow$  nimm 3. Wahl
- Grenzfälle:
  - nur eine feste Wahl
  - Alternativen nur bei defekter 1. Wahl (backup trunk)
- Wertung: Verfahren einfach, jedoch nicht adaptiv

# 7.3 Wegewahl, Routing

## Routing durch Flooding

- Idee: Jede eintreffende Nachricht wird an alle Nachbarknoten geschickt
- Problem: es entstehen beliebig oft Kopien
- Maßnahmen:
  - nicht zurück an sendenden Knoten
  - Lebensdauerbegrenzung durch Timer oder Hop Count
  - Nachrichtenmerkmal speichern
- Wertung:
  - Verfahren erzeugt Zusatzlast, aber sehr einfach
  - Verfahren extrem robust (z.B. Milit.Netze)
  - kann benutzt werden für schnelle multiple update von DB
  - enthält immer optimalen Weg (Metrik gegen andere Verfahren)

# 7.3 Wegewahl, Routing

Hot Potato

- isoliertes Verfahren
- ankommendes Paket wird als hot potato betrachtet und auf abgehende Leitung mit kürzester Warteschlange gelegt
- Evtl. erhebliche Umwege, falls gewählte Leitung nicht optimal zum Ziel führt
- empfindlich gegen Überlast

## 7.3 Wegewahl, Routing

Adaptives Verfahren nur mit Nachbarkennntnis (1)

- In allen Knoten  $j$  wird für jedes Ziel Wegetabelle geführt

$z$	$a^{opt}(j, z)$	$D_{jz}^{opt}$

- Jeder Knoten  $j$  erhält periodisch oder ereignisgesteuert von jedem Nachbarn  $D_{kz}^{opt}$  für  $k \in A(j)$  und alle  $z$  (Übertragungsvektor)
- Berechne  $D_{jz}(W_{kz}) = D_{jz} + D_{kz}^{opt}$  für alle  $k \in A(j)$ 
  - $D_{jz}$  : lokal
  - $D_{kz}^{opt}$  : durch Austausch
- Bestimme  $a^{opt}(j, z)$ , d.h. optimaler Nachbar
- Aktualisiere Wegewahltabelle

# 7.3 Wegewahl, Routing

Adaptives Verfahren nur mit Nachbarkennntnis (2)

- Beispiel: Distanz-Vektor-Alg. (Bellman-Ford-Alg)
- wird im Routing Information Protocol (RIP) im Internet benutzt
- Merkmale:
  - jeder Knoten hat Routingtabelle (Adresse, Distanz),
  - Update geschieht periodisch
  - jede Kante ist mit Gewicht belegt
  - Distanz ist Summe der Gewichte auf dem zum Ziel
- Initialisierung mit Eintrag, . . .
  - dessen Ziel dem lokalen Knoten entspricht,
  - dessen Next-Hop nicht angegeben ist und
  - dessen Distanz auf 0 gesetzt ist

## 7.3 Wegewahl, Routing

### Distanz-Vektor-Algorithmus

```
while true do
  warte auf nächste Routing Nachricht vom Nachbarn N;
  for all Eintrag in Nachricht do
    Z sei Ziel im Eintrag; D sei Distanz;
    Berechne:  $C := D + \text{Gewicht der Kante des Eintreffens von NN}$ ;
    //Prüfe und aktualisiere die lokale Routingtabelle:
    if es gibt keine Route zu Z then
      ergänze Eintrag mit Ziel Z, NextHop N, Distanz C;
    else if es gibt eine Route mit NextHop N then
      ersetze Distanz in existierender Route durch C;
    else if es gibt eine Route mit  $D > C$  then
      ändere NextHop auf N und Distanz auf C;
    end if
  end for
end while
```



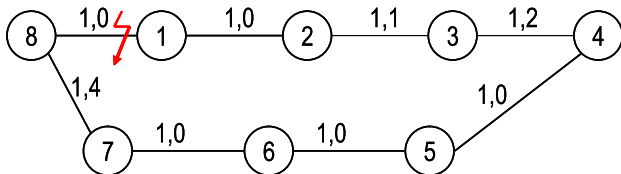
## 7.3 Wegewahl, Routing

Adaptives Verfahren nur mit Nachbarkennntnis (3)

### Wertung

- Verfahren sehr einfach, Rechenaufwand gering
- Nur partielle Topologie muss bekannt sein
- Zusätzliche Netzbelastung
- Wegewahlinfo breitet sich nur langsam aus, kann zu Inkonsistenzen führen!

### Beispiel 1 (Aufgabe)

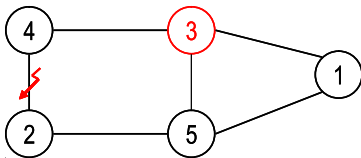


$W_{18}$  falle aus zum Zeitpunkt  $t_1 < x < t_2$ . Berechne die ersten Schritte zu den Zeitpunkten  $t_1, \dots, t_8$

## 7.3 Wegewahl, Routing

Adaptives Verfahren nur mit Nachbarkennntnis (4)

### Beispiel 2 (Aufgabe)



$D_{jk} = 1$  für alle  $j, k$   
periodisch,  $t_i = t_0 + i\Delta t$

Initialisiere  $D_{jz}^{opt}$  mit  $N + 1$  ( $N=5$ )

Annahme:  $L_{24}$  fällt aus in  $[t_x, t_{x+1}[$ , werde in  $[t_y, t_{y+1}[$  wieder repariert.  
Betrachte Wegetabelle in Knoten 3!

# 7.3 Wegewahl, Routing

Gemeinsam adaptive Wegberechnung (1)

## Ziel

- möglichst alle Knoten sollen aktuelle Wegewahlinfo haben.
- langsame Ausbreitung von Anpassereignissen soll vermieden werden

## Verfahren

- Basis ist SDF-Algorithmus
- jeder Knoten kennt Topologie und Bewertung
- jeder Knoten enthält von jedem anderen Knoten  $j$  dessen lokalen Verzögerungsvektor  $[k, D_{jk}]$  für alle  $k \in A(j)$
- Verteilung Vektoren z.B. durch Flooding

# 7.3 Wegewahl, Routing

Gemeinsam adaptive Wegberechnung (2): Beispiel

## Link-State-Routing, SPF-Routing

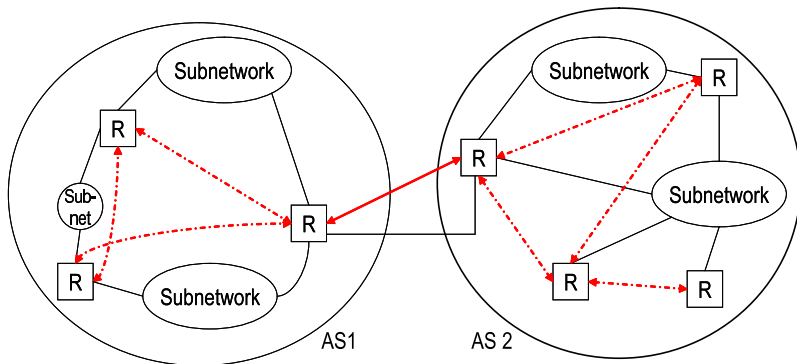
- wird im Internet im Routingverfahren OSPF verwendet
- jeder Router versucht seine Nachbarn kennenzulernen
- jeder Router bildet ein Link State Packet (LSP) mit Namen der Nachbarn und Gewichten der zugehörigen Links
- LSP werden an alle Rechner geschickt, jeder Rechner speichert die zuletzt erhaltenen LSP aller anderen Router. Damit kennt jeder Router die vollständige Netztopologie
- Berechne QSB nach SDF-Verfahren (Dijkstra)

# 7.3 Wegewahl, Routing

Routing im Internet: Autonome Systeme (1)

## Routing-Hierarchie

- 1 Autonome Systeme (AS): administrativ selbstständige Netze
- 2 Subnetze innerhalb eines AS



----- Interior Router Prot, z.B. RIP, OSPF

—— Exterior Router Prot, z.B. BGP

# 7.3 Wegewahl, Routing

Routing im Internet: Autonome Systeme (2)

## Autonomes System (AS)

- Eine Menge von Routern unter der Kontrolle *einer* administrativen Domäne (AS ist *Routing Domain*)
  - AS sind genehmigungspflichtig (durch IANA/ICANN)
  - Nutzung eines **Interior Gateway Protocol** (IGP) im Inneren des AS
  - Nutzung eines **Exterior Gateway Protocol** (EGP) zwischen AS
  - AS hat eindeutige numerische ID (z.B. LRZ/MWN: AS 12816)
  - Abkommen zwischen AS (ggf. mit Zahlungen verbunden)
    - **Peering**: Routen **in/aus** Netz eines fremden AS
    - **Transit**: Routen **durch/über** das Netz eines fremden AS
    - Technische Realisierung durch **Network Exchange**
- 
- IGP (intra-AS)
    - RIP (Routing Information Protocol): Distanz-Vektor-Algorithmus
    - OSPF (Open Shortest Path First): Link-State-Routing
  - EGP (inter-AS)
    - BGP (Border Gateway Protocol): ähnlich Distanz-Vektor-Verfahren

# 7.3 Wegewahl, Routing

## Border Gateway Protocol

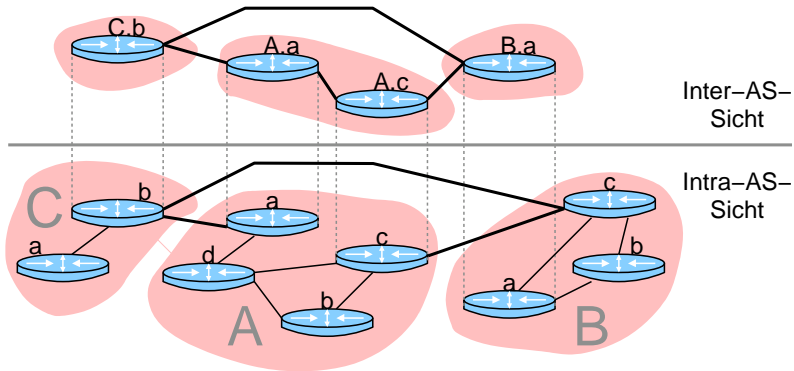
EGP-Kriterien für Wegewahl: finanziell, organisatorisch, eher nicht-technisch

### BGP: Arbeitsweise ähnlich Distanz-Vektor-Verfahren

- statt Distanzangabe: Angabe von **Pfad** als Sequenz der zu durchlaufenden AS
  - keine Metrik: Pfade realisieren **policy-basiertes** Routing
- bestimmte Pfade können ausgeschlossen oder uninteressant gemacht werden.
- Aufgaben: Neighbour acquisition, neighbour reachability, network reachability
  - TCP-basiert, Port 179 (Session: ständige Verbindung zw. "BGP-speakers")
  - RFC 4271, mit CIDR-Unterstützung; RFCs 1772-1774

# 7.3 Wegewahl, Routing

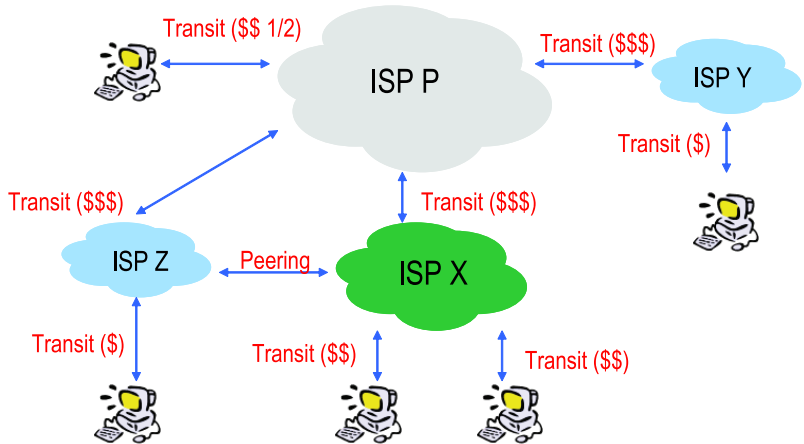
## Routing im Internet (3): Intra-AS-Routing





# 7.3 Wegewahl, Routing

## Routing im Internet (4): Transit vs. Peering



# 7.3 Wegewahl, Routing

Fragen zu Wegewahl/Routing

- Warum war es sinnvoll, im Internet RIP durch OSPF abzulösen?
- Was ist der Kernalgorithmus bei Link State Routing?
- In welchen Fällen ist Flooding ein sinnvolles Verfahren?
- Nennen Sie Kostenfunktionen, die dem Optimalitätskriterium genügen?
- Was versteht man unter autonomen Systemen?
- Neuere BGP-Versionen unterstützen CIDR. Was bedeutet diese Aussage?

# Rechnernetze und verteilte Systeme

## Vermittlungsschicht

### Internetprotocol IP

#### Kapitel 7.4

- Verbindungsloser Dienst, zwei Dienstprimitive SEND, DELIVER
- Parameter:
  - Source address, destination address: IP-Adresse Sender-/ Empfänger-Host
  - Protocol: ID des IP users (SAP-Adresse), z.B. 1 ICMP, 2 IGMP, 6 TCP, 8 EGP, 9 UDP, 46 RSVP, 89 OSPF
  - type of service (ToS) indicator
    - Priorität (8 Stufen)
    - Reliability (2 Stufen)
    - Delay (normal/low)
    - Thruput (normal/high)
  - identifier<sup>1</sup>: eindeutige Kennung für Reassembly und Error Reporting
  - "don't fragment" id<sup>1</sup>
  - time to live (TTL)<sup>1</sup>, gemessen in *hops*
  - data length (in Bytes)
  - option length, option data, data

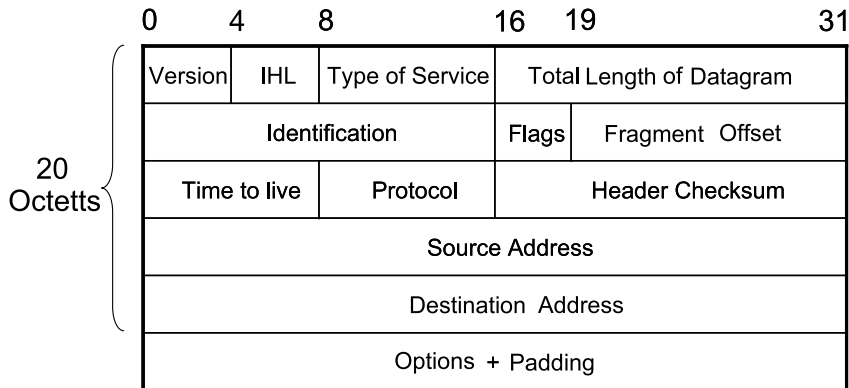
---

<sup>1</sup> entfällt bei DELIVER, options gestattet Sonderwünsche, z.B. Security label, Source Routing, Route Recording, stream id, timestamping

# 7.4 Internetprotocol IP

## IP-Header-Format

Das IP-Protokoll ist datagramm-orientiert



# 7.4 Internetprotocol IP

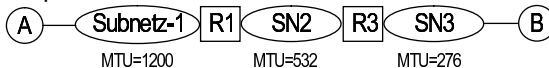
## Header Parameter

- Version: Version des IP-Protocols, derzeit 4
- IHL: Internet Header Length in 32-Bit-Worten (Minimum= 5), d.h. kleinster Header = 20 Oktetten
- Length: in Bytes
- Flags:
  - M “More bit” (für Fragmentierung und Wiederherstellung (*Reassembly*)
    - M = 0 folgt nichts (letztes Teilpaket)
    - M = 1 (folgen weitere Teilpakete)
  - DF “Don’t fragment bit”
- Fragment Offset: benutzt für Reassembly, zählt in Anzahl von 64-bit-Einheiten für den Datenanteil in Bezug auf Datenanfang
- Padding: Auffüllen des Headers auf Vielfaches von 32-bit-Einheiten
- Data:
  - variabel lang
  - Vielfaches von 8 bit
  - max. Datagramm-Länge (inkl. Header): 64KByte = 65535 Oktetts

# 7.4 Internetprotocol IP

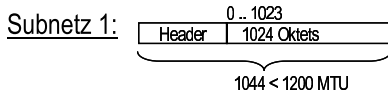
## Fragmentierung

### Beispiel

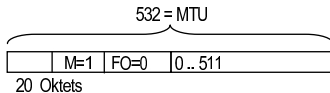
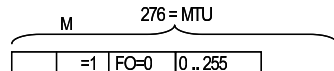


M: More Data Bit  
(in Flags)  
FO: Fragment Offset

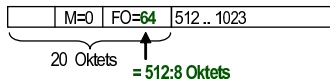
Aufgabe: Nachricht mit 1044 Oktets von A nach B  
IP-Header  $\geq 20$  Oktets



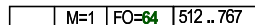
### Subnetz 1:



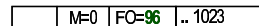
### Subnetz 2:



FO=256:8



FO=512:8



FO=768:8

# 7.4 Internetprotocol IP

## IP-Unterstützende Protokolle

- ICMP (Internet Control Message Protocol), RFC 792
  - Benachrichtigungsprotokoll, wirkt unterstützend für Fehler- und Mgmtbehandlung, nötige IP-VL-Dienst
  - Destination unreachable, time exceeded, Parameterproblem (Syntaxfehler im IP-Header), source quench (bei Router- Überlast), Redirect (Routing-Korrektur), Echo, Timestamp, usw.
  - Basis für traceroute
- DHCP (Dynamic Host Configuration Protocol)  
Dynamische temporäre Zuweisung von IP-Adressen
- DNS (Domain Name Service)
  - Abbildung von Namen auf IP-Adressen
  - Interaktion zwischen Resolver (Client) und Server (Directory)
- ARP (Address Resolution Protocol), RFC 826  
Abbildung von IP-Adressen auf MAC-Adressen



# 7.4 Internetprotocol IP

## Virtuelle Private Netze (VPN)

**Idee:** transparentes Zusammenschalten mehrerer (privater) Netze zu einem einzigen (dem virtuellen Netz)

- private Netze über öffentliche Netze verbunden
- oft: Einsatz kryptografischer Verfahren

**Nutzungsszenarien**

- Aggregation der Netze mehrerer Standorte (einer Organisation) zu einem einzigen Netz
- Anbindung nomadischer Nutzer (z.B. reisende Mitarbeiter)
- Allgemein: Verschattung der physischen Netztopologie

# 7.4 Internetprotocol IP

## Fragen zu Kapitel 7.4

- Welches Vermittlungsverfahren wird im Internet auf Ebene 3 angewendet?
- Unterstützt IP einen VO- oder VL-Dienst?
- Wer sind im Internet Dienstanutzer von IP?
- Warum muss IP Fragmentierung unterstützen?
- Nennen Sie (außer Adressen) mindestens 3 wichtige IP-Headerinformationen.
- Wozu braucht man ICMP?
- Wo braucht man ARP?
- Wozu dient das Protocol-Field im IPv4-Header?

# Rechnernetze und verteilte Systeme

## Vermittlungsschicht

### IPv6

#### Kapitel 7.5

### Motivation

- Adressknappheit bei IPv4 bei wachsendem Bedarf an Adressen
- Neue Anforderungen: QoS, Mobilität ...

### Neue und ausgebauten Funktionalität im Vergleich zu IPv4

- erweiterter Adressraum (128 statt 32 bit)
- Erweiterungen für QoS auch bzgl. Streams wie Sprache, Video
- dynamische Adresszuweisung (Mobilitätsunterstützung)
- Ressource Allocation (Reservierungsprotokolle)
- Security capabilities (privacy, authentication)
- hop-by-hop-options (Routeranweisungen)

### Status

- IPv6 wächst; Implementierungen in gängigen BS umgesetzt
- Vorreiter China, Japan. Grund u.a. mangel an IPv4-Adressblöcken

- 128-Bit Unicast-, Multicast- und *Anycast*-Adressen
  - Keine Netzklassen
  - Netzschnittstellen (*Interfaces*) erhalten Adressen (nicht Hosts)
  - Anycast: Adressierung eines beliebigen Hosts einer Gruppe (→ Lastverteilung, Ausfallsicherheit)

### Schreibweise von Adressen: hexadezimal $x:x:x:x:x:x:x:x$

- $x$  entsprechen 16 Bit, z.B. `1080:0:0:0:8:800:200C:417A`
- Verkürzt: zusammenfassen von Nullen durch '::', z.B. `1080::8:800:200C:417A` (nur ein mal pro Adresse möglich)
- Präfixschreibweise (wie CIDR): `12AB::CD30:0:0:0:0/60`

### • Spezielle Adressen

- Unspecified Address ("Abwesenheit einer Adresse"): `0:0:0:0:0:0:0:0`
- Loopback Adresse: `0:0:0:0:0:0:0:1`

### Aggregatable Global Unicast Addresses

- Felder einer Adresse

FP Format Prefix

TLA ID Top-Level Aggregation Identifier

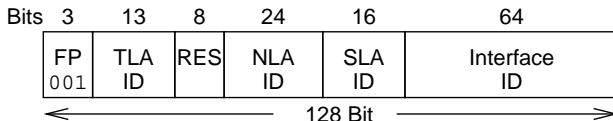
NLA ID Next-Level Aggregation Identifier

SLA ID Site-Level Identifier

RES reserviert

- Dreistufige Adresshierarchie

- Public Topology (z.B. Transitnetze)  
dargestellt durch erste 48 Bits (insbesondere TLA, NLA)
- Site Topology (standortbezogen) → SLA-Feld
- Interface Identifier

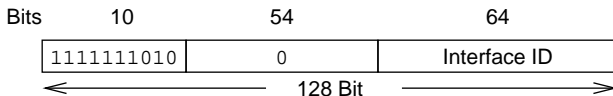


# 7.5 IPv6

Lokale Unicast-Adressen: "Local-Use IPv6 Unicast Addresses"

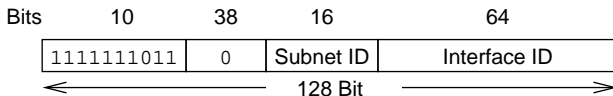
## Link-local

- Bezug nur auf einen bestimmten Link
- Einsatz z.B. für Autokonfiguration



## Site-local

- Geltungsbereich umfasst einen Standort/eine Site
- ähnlich IPv4 privaten Adressen
- sollten nicht außerhalb des Standorts geroutet werden



# 7.5 IPv6

## IPv6-Adressen auf Basis von Schicht-2-Adressen

**Idee:** Nutzung gerätegebundener Adressen bei der Bildung von IPv6-Adressen

- Anwendung nur auf *Interface Identifier* (64 Bits)
- Extended Unique Identifier (EUI-64) (Definition der IEEE)
- z.B. anwendbar mit Ethernet, Token Ring, FDDI ...
- Beispiel: IEEE 802, 48 Bit → Abbildung auf 64 Bits erforderlich

**00-23** Kennung des Herstellers mit Flag-Bits

6 universal/local (1 = globaler Geltungsbereich)

7 individual/group

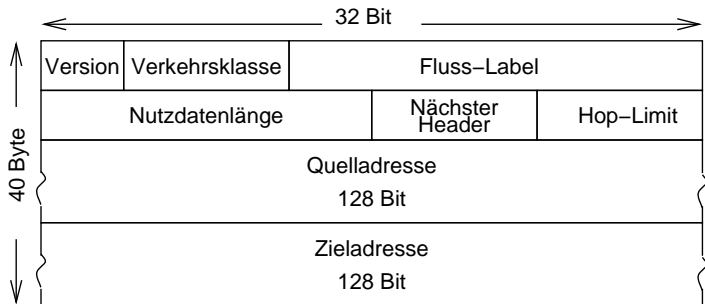
**24-39** 11111111 111111110

**40-63** spezifisch für Geräteinstanz



# 7.5 IPv6

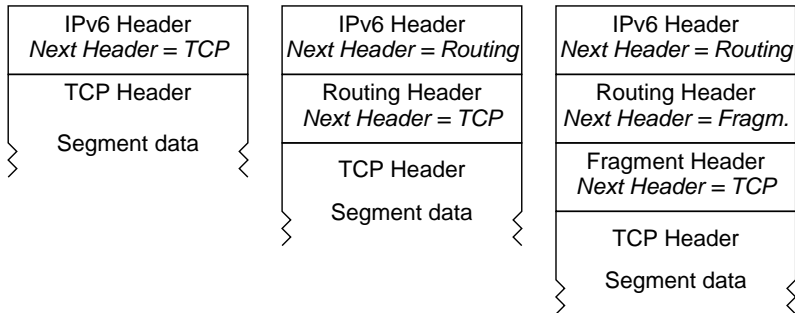
## Header



- Feste Headerlänge von 40 Byte → Header-Längenfeld entfällt
- Keine Header-Prüfsumme → keine Neuberechnung bei jedem Hop
- *Verkehrsklasse* entspricht Type-of-Service Feld bei IPv4
- *Fluss-Label* Zuordnung von Paketströmen/-flüssen
- *Nächster Header* entspricht Protokollfeld in IPv4-Header  
kann alternativ auf *Extension Header* verweisen
- *Hop-Limit* entspricht TTL-Feld bei IPv4

# 7.5 IPv6

## Extension header: Verkettung



- “Nächster-Header”-Feld gibt Typ des folgenden Headers an
- Anzahl (i.d.R. eins) und Reihenfolge der Extension Header vorgegeben → effiziente Abarbeitung
- IPv6-Spezifikation (RFC 2460) fordert mindestens:
  - Hop-by-Hop Options, Routing, Fragment, Destination Options, Authentication, Encapsulating Security Payload
- Zusätzliche Header-Typen möglich, z.B. für Mobilitätsunterstützung

### Hop-by-Hop Options Header

- Zweck: Angabe von Optionen, die jeder Knoten auf dem Weg berücksichtigen soll. (z.B. Padding)
- Felder
  - *Next Header* : Angabe des folgenden Headers (wie bekannt)
  - *Hdr Ext Len* : Länge dieses Headers (Anzahl Oktetts - 1)
  - *Options* : type-length-value (TLV) kodierte Optionen

### Routing Header

- Zweck: Angabe von Zwischenstationen
- Felder
  - *Next Header, Hdr Ext Len*: wie bekannt
  - *Routing Type*: identifiziert Routing-Header Variante (default: 0)
  - *Segments Left*: Anzahl verbliebene Zwischenstationen. Wenn 0 erreicht wird, wird der Routing Header ignoriert und der nächste Header verarbeitet
  - *type-specific data*: abhängig von Routing Type

### Fragment Header

- Zweck: Senden eines IP-Pakets, das größer ist als die Max. Transfer Unit (MTU)
- Felder
  - *Fragment Offset*: Offset der Daten, die diesem Header folgen
  - *M flag*: wie IPv4 *more flag* (0 = letztes Fragment, 1 sonst)
  - *Identification*: eindeutige Kennung des *IP-Pakets*, das fragmentiert wird
- Anders als bei IPv4: Fragmentierung nur bei DEE, *nicht* im Transitsystem
- *nicht-fragmentierbarer* Teil
  - IPv6-Header + alle Header, die Ende-zu-Ende Relevanz haben
- *fragmentierbarer* Teil: Rest
- Fragment-Pakete haben die Form:
  - 1 nicht-fragmentierbarer Teil (wie oben)
  - 2 Fragment Header
  - 3 Inhalte des ersten Fragments

### Betriebliche Aufgabenstellung

- Ersetzung der IPv4-Technologie; möglichst kosteneffizient
- ⇒ Koexistenz von IPv4 und IPv6 während Übergangsphase

### Prinzipielle Ansätze

- Dual-Stack: Knoten besitzen sowohl IPv4- als auch IPv6-Implementierung
- Tunneling
  - Kapselung von Paketen im tunnelnden Protokoll
  - IPv6-Verkehr über IPv4-Netz bzw. IPv4-Verkehr über IPv6-Netz

⇒ Management des Tunnels: Erstellung, Umgang mit Fragmenten
- Übersetzung
  - Protokollübersetzung in Übergangsknoten/Router
  - in Programmbibliotheken (APIs zur Netzprogrammierung)

# Rechnernetze und verteilte Systeme

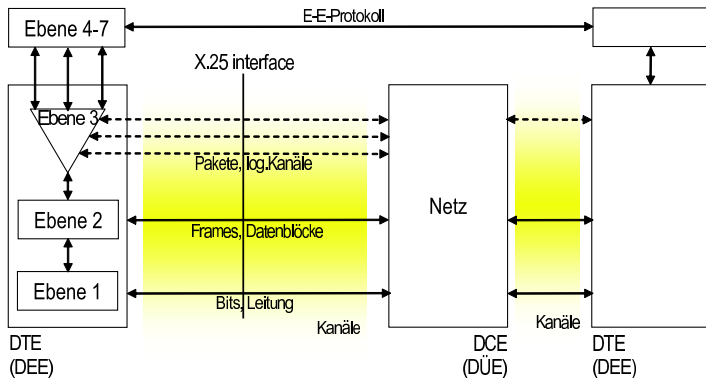
## Vermittlungsschicht

X.25

Kapitel 7.6

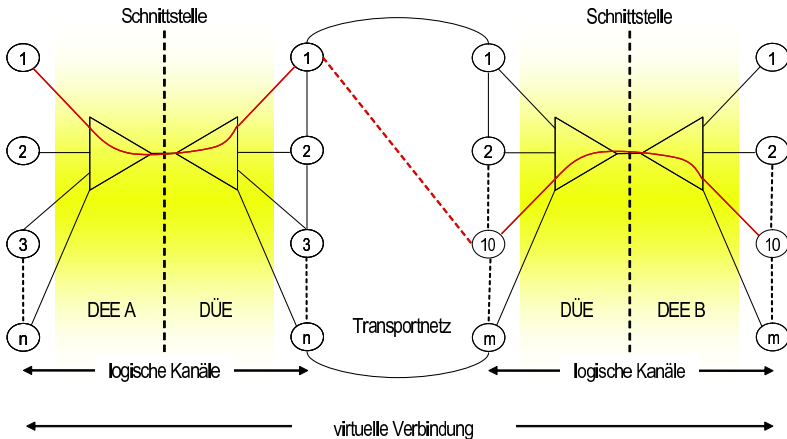
- X.25 ist von ITU als Netzzugangsprotokoll spezifiziert. Ist Basis für DatexP

- Ebene 1 = X.25 level 1 = X.21
- Ebene 2 = X.25 level 2 = HDLC LAP B
- Ebene 3 = X.25 level 3 = Paketebene (PLP)
- PLP ist Multiplexprotokoll, bietet VO/VL-Dienst, Paketvermittlung



# 7.6 X.25

## Logische Kanäle, logische Verbindungen





Virtuelle Verbindung	Logischer Kanal
Eine virtuelle Verbindung stellt eine (Ende-zu-Ende-) Verbindung zweier DEE über das Transportnetz dar.	Ein logischer Kanal stellt eine <i>lokale</i> Einrichtung zwischen DEE und DÜE dar.
Zur Abwicklung können verschiedene logische Kanäle benutzt werden.	Nur <i>eine</i> virtuelle Verbindung wird zu ihrer Abwicklung einem logischen Kanal zugeordnet.
Eine virtuelle <i>Wählverbindung</i> existiert nur zwischen Verbindungsherstellung und Verbindungsauflösung (bzw. Auf-/Abbau). Eine <i>feste</i> virtuelle Verbindung gilt als ständig existierend.	Ein logischer Kanal ist immer existent und entweder einer virtuellen Verbindung zugeordnet oder frei (ready state).

- Es werden 16 log. Kanalgruppen zu je 256 Kanälen unterstützt
- Anzahl und Art der Kanäle werden bei Subskription festgelegt, d.h. für PVC, DG, sx-Verbindung, dx-Verb. (SVC)
- Merkmale bei Einrichtung:
  - Gebührenübernahme
  - Durchsatzklasse
  - Closed user group
  - Subadressen (Nst-Anlage)
  - Blockierung best. Rufe
  - Fenstergrößen
- Merkmale bei Verb.-Aufbau: z.B. Gebührenübernahme

DTE → DCE	DCE → DTE	DG	PVC	VC
Verbindungsaufbau/abbau				
Call request	Incomming Call			X
Call accept	Call connected			X
Clear request	Clear indication			X
Clear confirmation	Clear confirmation			X
Daten und Interrupts				
Data	Data		X	X
Interrupt	Interrupt		X	X
Interrupt Confirm	Interrupt Confirm		X	X
Datagram	Datagram	X		
	Datagram Service Signal	X		

DTE → DCE	DCE → DTE	DG	PVC	VC
Flußsteuerung und Reset				
RR	RR	X	X	X
RNR	RNR	X	X	X
Reset request	Reset indication	X	X	X
Reset confirm	Reset confirm	X	X	X
Restart				
Restart request	Restart indication	X	X	X
Restart confirm	Restart confirm	X	X	X
Diagnostic				
	Diagnostic	X	X	X

} betrifft eine  
log. Verb.

} betrifft alle  
virt. Verb.

# 7.6 X.25

## Paket-Formate

8	7	6	5	4	3	2	1
0	0	0	1	log. Kanal-Gr			
log. Kanal-Nr							
0	0	0	0	1	0	1	1
Source Addr				Dest. Addr			
Length				Length			
Addresses							
				0	0	0	0
0	0	Fac. Field Length					
Facilities < 40B							
User Data < 128 B							
bei Fast attach							

Call Request

8	7	6	5	4	3	2	1
D	Q	0	1	log. Kanal-Gr			
log. Kanal-Nr							
P(R)		M		P(S)		0	
User Data < 128 B							

Data Packet