

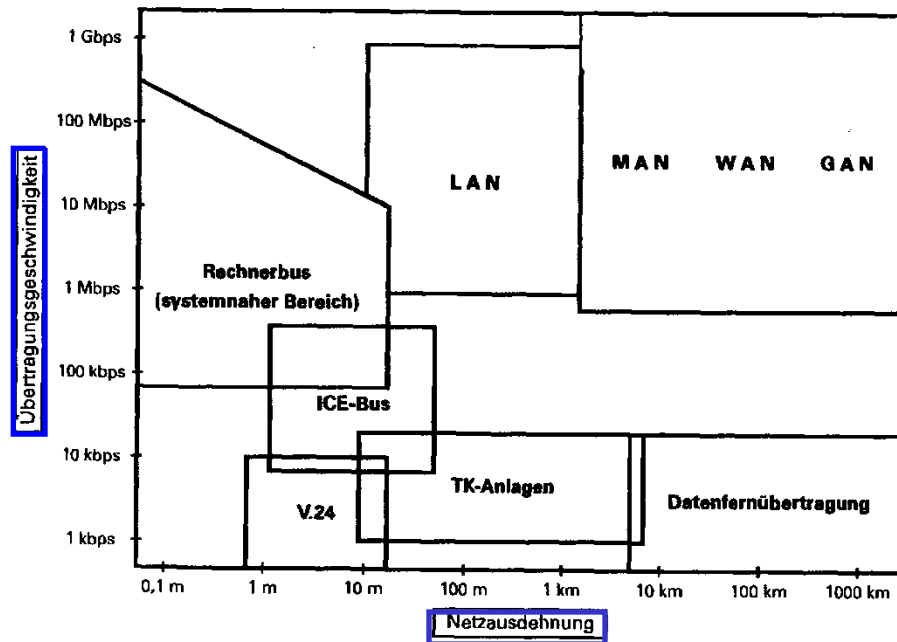
# Kapitel 6      Kommunikation in Weitverkehrsnetzen und im globalen Internet

6.1	Übersicht über und Klassifikationen für (über)regionale Kommunikationsnetze	2
6.2	Vermittlungstechniken: Durchschalte- und Zwischenspeicherungstechniken	14
6.3	Wegeermittlung durch ein vermaschtes Kommunikationssystem	33
6.4	Namensgebung und Adressierung	54
6.5	Interkonnektion von Netzen	69
6.6	Schmalband-ISDN	77
6.7	ATM-Protokolle und –Netze	84
6.8	Das globale Internet	95

# 6. Kommunikation in Weitverkehrsnetzen und im globalen Internet

## 6.1 Übersicht über und Klassifikationen für (über-)regionale Kommunikationsnetze

- Einordnung von Kommunikationsnetzen nach Übertragungsgeschwindigkeit und Netzausdehnung, vgl. [Göb 99] :



**Abb. L.1-1** Einordnung von Kommunikationsnetzen nach Übertragungsgeschwindigkeit und Netzausdehnung

➤ Weitere **Kommunikationsnetz-Klassifikationen** mit Relevanz für nicht-lokale Netze

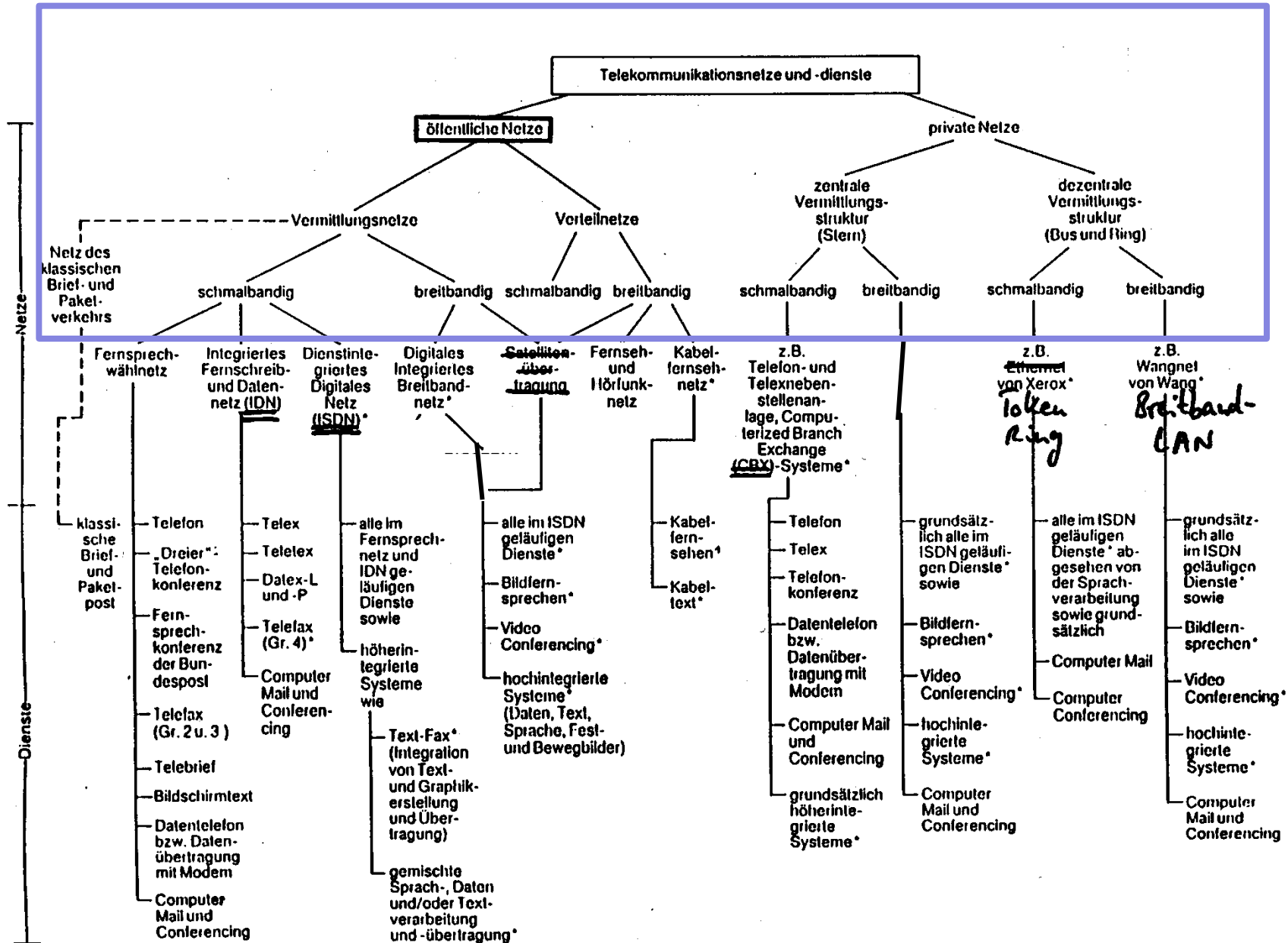


❖ **Verteil-** *versus* **Vermittlungsnetze** :

- **Verteilnetz**: zahlreichen Empfängern wird seitens eines (oder einiger weniger) Sender dieselbe Information unidirektional bereitgestellt, z.B. TV, Rundfunk, ...
- **Vermittlungsnetz**: zwischen zwei oder einer (kleineren) Gruppe von Kommunikationspartnern wird Information bidirektional ausgetauscht, z.B. Telefon, Fax, Videokonferenz, ...

❖ **private** *versus* **öffentliche Kommunikationsnetze** :

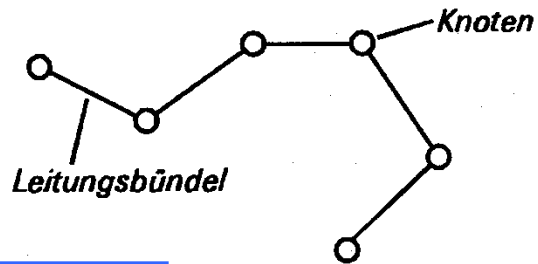
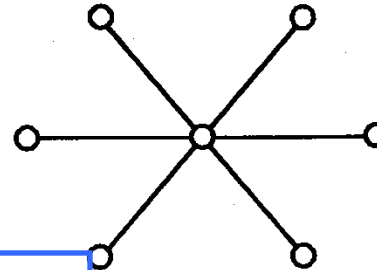
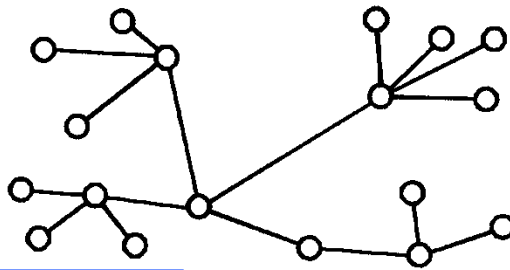
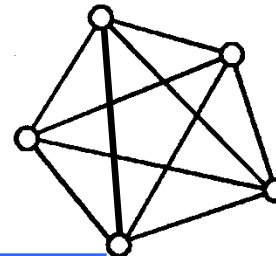
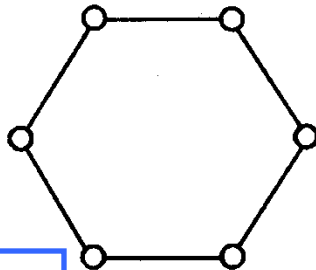
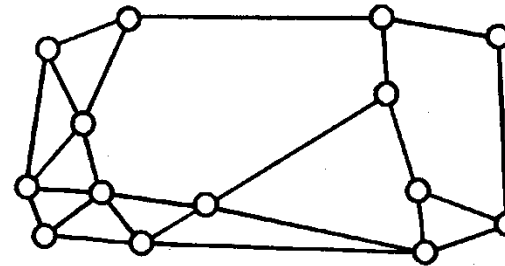
- **privates Netz**: Privatperson oder Unternehmen als Netzbetreiber
- **öffentliches Netz**: öffentlicher Anbieter von Netzdiensten, z.B. Telekom, Internet Service Provider (ISP), ...



### Zusammenhänge zwischen Diensten und Netzen in der Telekommunikation

(Fokus auf zurückliegende Dekade und auf leitungsgebundenen Kommunikationsnetzen)

## Häufige **Topologieformen** bei (über-) regionalen Netzen:

**Linienetz****Sternnetz****Strahlennetz****Maschennetz****Ringnetz****Verbundnetz**

## Bem. zu **Topologien**:



- **Linienetz**: besitzt Ähnlichkeit zu Bus, vgl. DQDB bei MANs; evtl. redundante Übertragungswege (Leitungsausfälle !)
- **Sternnetz**: evtl. auch als Grundform einer Baumstruktur (z.B. viele Teilnehmeranschlüsse an Ortsvermittlung *oder* zahlreiche Ortsvermittlungs- an Fernvermittlungsstellen angeschlossen)
- **Strahlennetz**: auch hier Nähe zu hierarchischer Netzstruktur (s.u.)
- **Maschennetz**: in nicht-lokalem Bereich nur relevant bei rel. geringen Knotenzahlen  
→ auch als “*vollständige Vermaschung*“ bezeichnet
- **Ringnetz**: z.B. bei FDDI-Ringen in MANs
- **Verbundnetz**: *die* dominierende Topologieform bei WANs und GANs.  
→ auch als “*irreguläre Vermaschung*“ bezeichnet

# Bewertung der Topologien

➤ **Bewertungskriterien** für Wahl einer Netzform, insbesondere

- geographische Verteilung (u.a. der Netzbenutzer);
- Parameter der zu vermittelnden Nachrichten (Größe, Verkehrsmatrix, etc);
- Unempfindlichkeit gegenüber Störungen, Ausfällen;
- Art der benutzten Vermittlungstechnik in den Knoten.



➤ **Vor-/Nachteile** obiger Topologieformen:

- **Liniennetz:**

- + : rel. geringe Anzahl von Leitungen (hier immer im Sinne von *Leitungsbündeln* !) erforderlich
- : bei Ausfall von Knoten bzw. Leitungen → kein Ersatz

- **Sternnetz:**

- + : rel. geringe Anzahl von Leitungen, übersichtliche Netzgestaltung, Routing trivial
- : Ausfall von Leitung → betroffener peripherer Knoten isoliert; Ausfall zentr. Vermittlungsknoten → “rien ne va plus !“

➤ **Vor-/Nachteile** obiger Topologieformen (Fortsetzung):

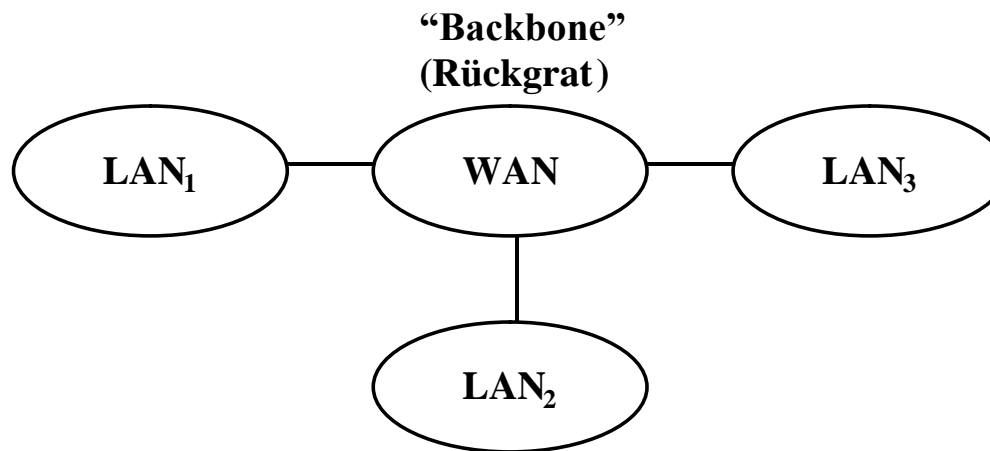
- **Strahlennetz**, vgl. Stern
- **Maschennetz** : (auch: **vollständige Vermaschung**)
  - + : Ausfall von Knoten → genau dieser Knoten nicht mehr Sender und Empfänger; Ausfall von Leitung → Verkehrsumleitung über Ersatzwege
  - : hoher Verkabelungsaufwand, evtl. geringe Leitungsauslastung, aufwändige Zusatzverkabelung bei Hinzunahme von Knoten
- **Verbundnetz** : (auch: **irreguläre Vermaschung**)
  - + : Netzengpässe durch zusätzliche Leitungen gezielt eliminierbar, evtl. auch gezielte, bedarfsorientierte Wahl von Leitungskapazitäten (Datenraten), Ausfalltoleranz und hohe Effizienz
  - : hohe Komplexität (u.a. aufwändige Wegeermittlung, Fluss-/Sättigungskontrolle, ...), anspruchsvolle Netzkonfigurierung



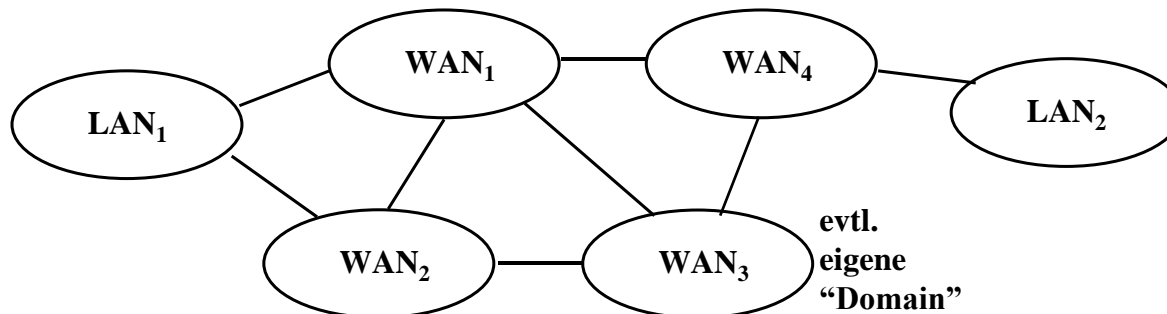
# Zur Strukturierung der Topologien sehr großer Netze:

## ➤ Netz aus Netzen:

- *LAN-WAN-Kopplung*, z.B.

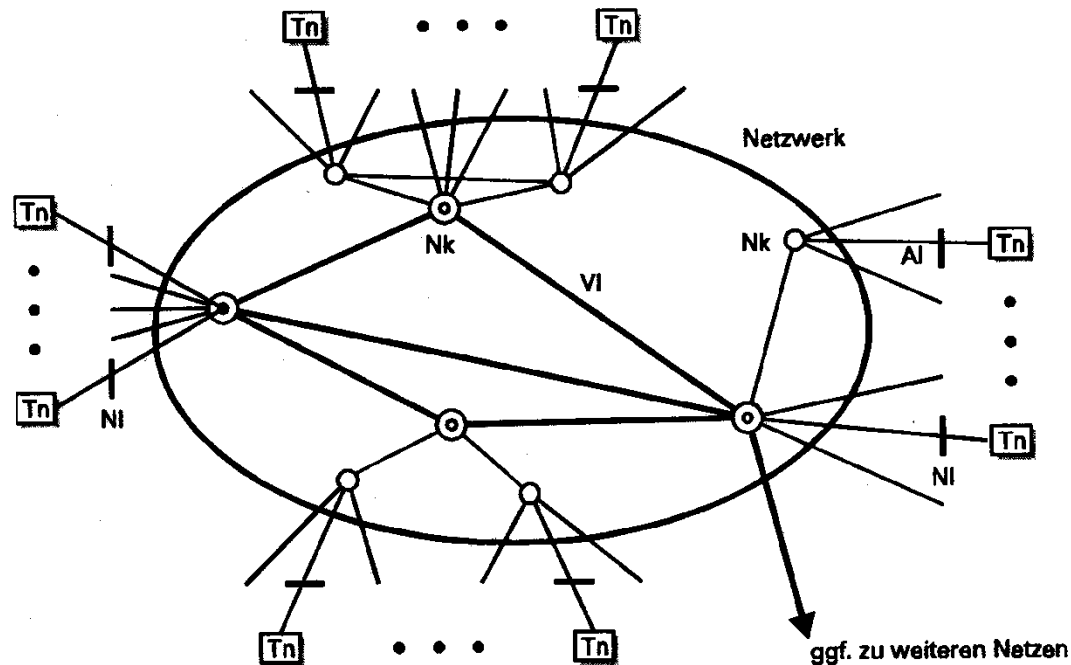


- *Netze als Transitnetze*, z.B.



# Zur Strukturierung der Topologien sehr großer Netze (Forts.):

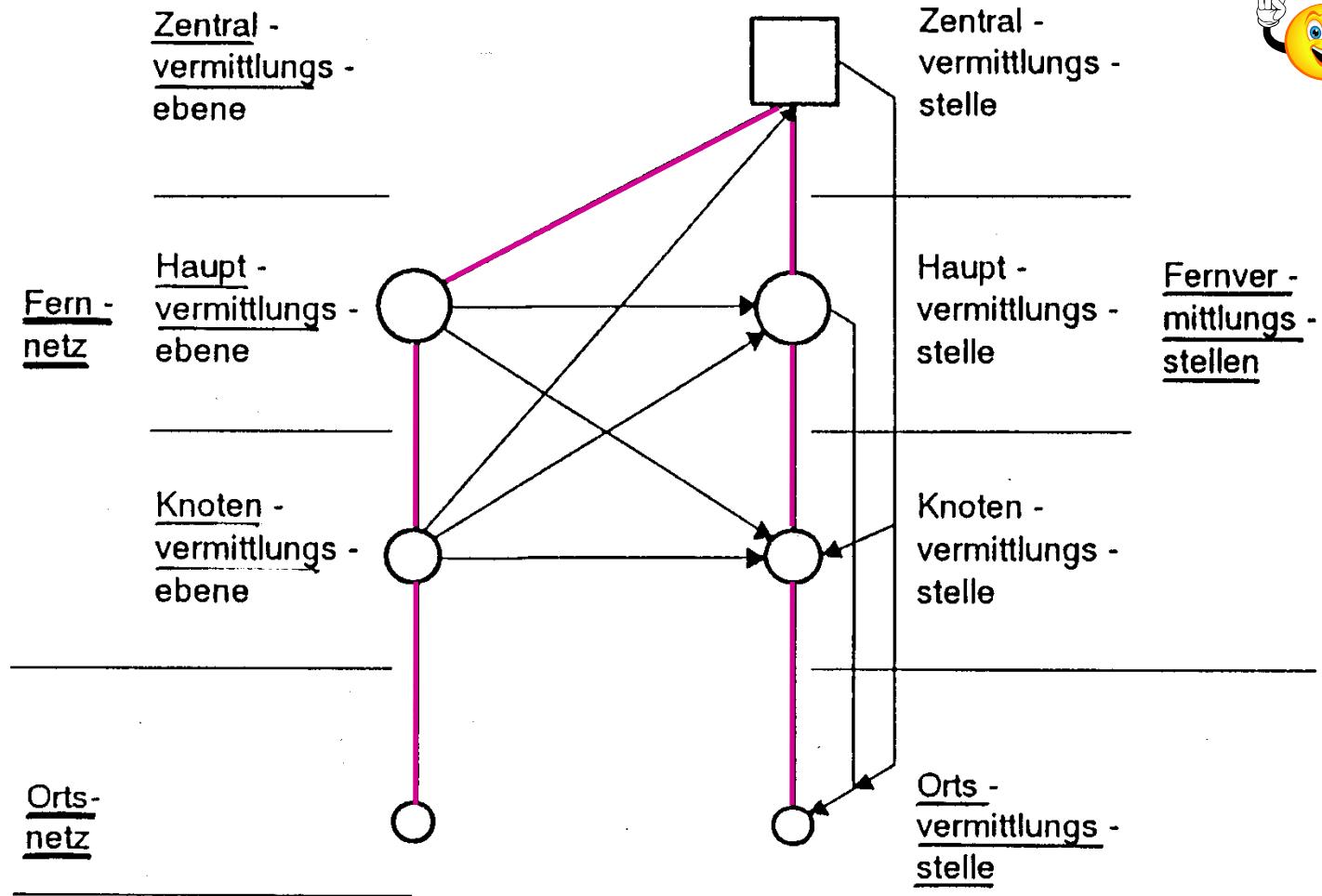
## ➤ Hierarch. Aufbau eines Vermittlungsnetzes:



Notation:

- **NI** : Network Interface
- **Nk** : Netzknoten unterschiedlicher Hierarchiestufen
- **Al** : Anschlussleitung
- **VI** : Verbindungsleitung





**Bild 3.7.** Struktur des Telefonnetzes/64-kbit/s-ISDN der Deutschen Telekom

— Letztweg (Kennzahlenweg),  $\Rightarrow$  Querweg

(aus: Bocker [Boc 97])



## **Betreiberhierarchie** für öffentliche Kommunikationsnetze

- Anbieter von Übertragungsdiensten (sog. “*Carriers*“):  
→ z.B. Anbieter von optischen Ü-Strecken, Satellitenverbindungen, Funkstrecken, u.a.m.
- Anbieter von elementaren Datenübertragungs- oder Sprachübertragungsdiensten (Wählan-  
schlüsse)  
→ z.B. X.25-Dienst über DATEX-P oder Zugang zu DATEX-L-Netz
- Anbieter von Internet-Zugangsdiensten  
→ z.B. ISP (Internet Service Provider) wie Arcor, HanseNet / ALICE (O<sub>2</sub>), T-Com /  
T-Online, DFN-Verein, ...
- Anbieter von höherwertigen Individualkommunikationsdiensten über dienstintegrierte Netze  
→ z.B. Zugang zu ISDN-Netz
- Anbieter von Informations-, Auskunfts-, Suchdiensten u.ä.  
→ z.B. Fachinformationsdienste, Suchmaschinen im Internet,  
“*Content Provider*“

*Nota bene:* Betreiberhierarchien können auch andere Ausprägungen besitzen als im obigen Beispiel !

## Beispiel einer konkreten Betreiberhierarchie:

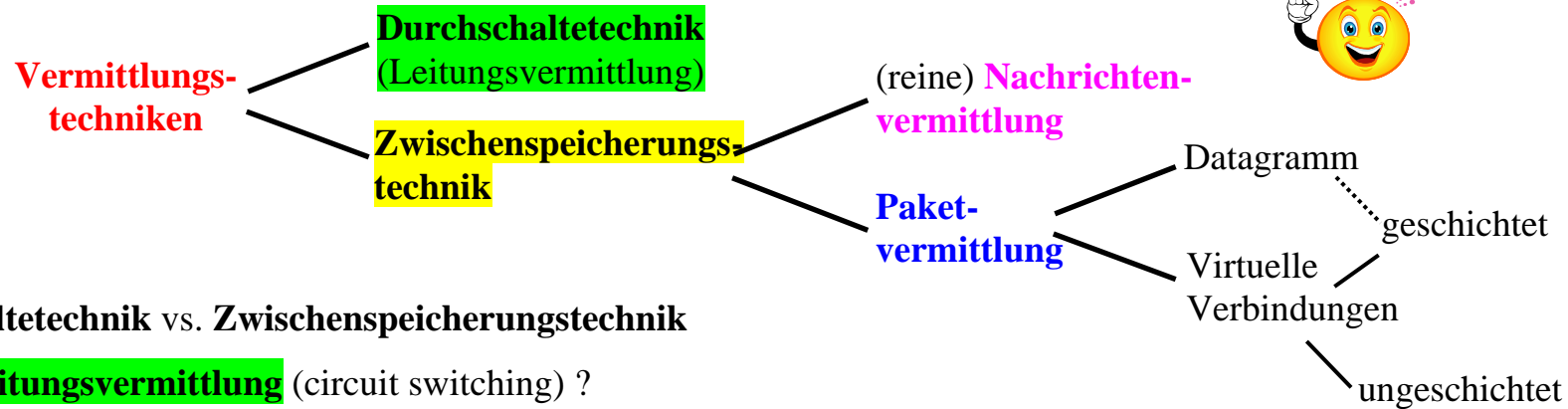


⇒ Konsequenzen:

- erschwertes Netzmanagement,
- Konkurrenzkampf (→ Preisvorteile ! dynamischer Markt !),
- “Black Box“-Sicht auf Netze für manche Dienstanbieter

## 6.2. Vermittlungstechniken : Durchschalte- und Zwischenspeicherungstechniken

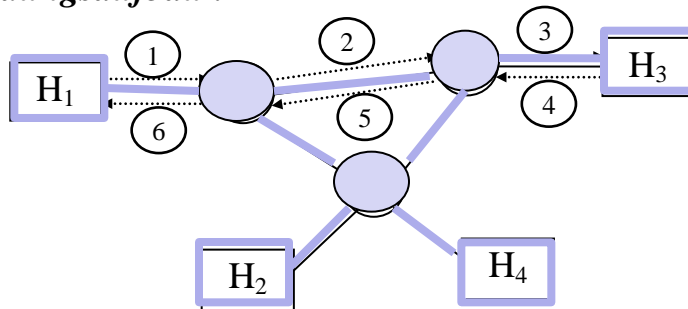
Klassifikation :



### (a) Durchschaltetechnik vs. Zwischenspeicherungstechnik

- Prinzip **Leitungsvermittlung** (circuit switching) ?  
(Ursprung : Fernsprech-Wählnetze)

Verbindungsaufbau :



Notation :

$H_i$  = Arbeitsrechner/Endsystem



= Vermittlungsrechner

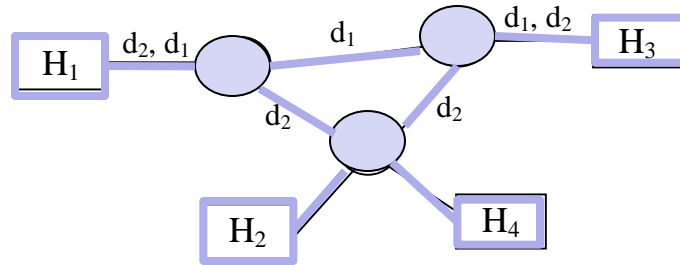
(Darstellung in diesem

Kap. zur einfacheren Unterscheidung verwendet)

⇒ anschließend fest durchgeschaltete, dedizierte physikal. Verbindung für Dauer der Kommunikation  
(statische Reservierung von Betriebsmitteln)

## (a) Durchschaltetechnik vs. Zwischenspeicherungstechnik (Forts.):

- Prinzip **Zwischenspeicherungstechnik** (store-and-forward) ?  
(Ursprung : wiss. Rechnernetze, z.B. ARPAnet; später: Internet)



$d_1, d_2$  : Dateneinheiten

⇒ Zwischenspeicherung von Dateneinheiten in VR; Weiterleitung falls Leitung frei  
(*dynam. Allokation* von Betriebsmitteln)

### Beurteilung Durchschalte-/Zwischenspeicherungstechnik :

#### \* mögl. *Vorteile* der Durchschaltetechnik :

- weitgehend konstante Verzögerungen bei DÜ (Wartezeiten eliminiert)
- vereinfachte Wegeermittlung (Routing nur bei Leitungsdurchschaltung)
- allg. : weniger Komplexität

#### \* mögl. *Nachteile* der Durchschaltetechnik :

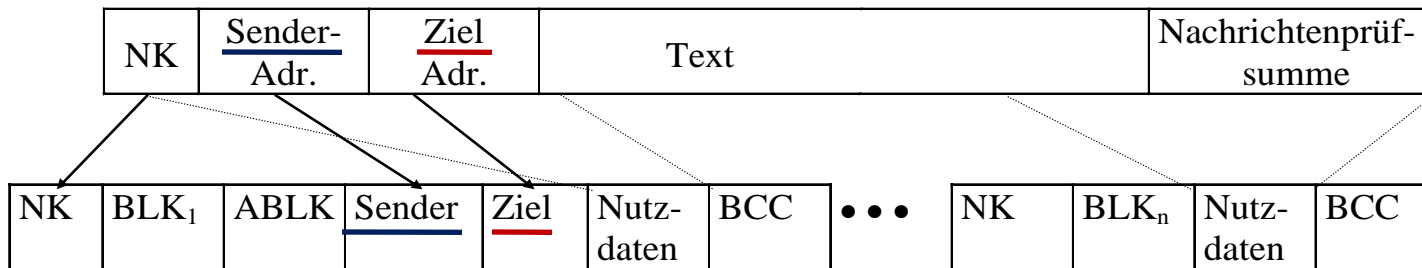
- Verbindungsaufbau relativ lange; lohnt evtl. nicht bei kurzer Verbindungsdauer (Bsp.: Transaktionsorientierter Verkehr)
- Verschwendung von Ressourcen durch statische Reservierung (Schwankungen der Verkehrslast!)



## (b) reine Nachrichtenvermittlung vs. Paketvermittlung

- Prinzip **Nachrichtenvermittlung** (*message switching*) ?  
 ⇒ Interpretiere  $d_i$  als Nachrichten (variabler Länge) bei Zwischenspeicherungstechnik.  
 Vorgehensweise beim Datenaustausch :
  - $d_i$  evtl. fragmentiert in zusammengehörige Blöcke bei DÜ über Leitung  
 (Adressinfo nur im 1. Block der Nachricht)
  - zusammengehörige Blöcke reassembliert zu  $d_i$  in jedem VR

Formate der Dateneinheiten bei reiner Nachrichtenvermittlung :



NK ≡ Nachrichten-Kennung  
 BLK ≡ Block-Nr.

ABLK ≡ Anzahl der bei Fragmentierung  
 entstandenen Blöcke

BCC ≡ Blockprüfzeichen (Block Check Character)

- Prinzip **Paketvermittlung** (*packet switching*) ?  
 ⇒ Interpretiere  $d_i$  als Pakete (mit a priori vorgegebener Maximallänge, z.B.  $I_{max}$  [kb]),  
 wobei Paket ≡ Teil der ursprüngl. Nachricht; Pakete besitzen jeweils vollständige Adressinfo;  
 Paketreassemblierung erst am „Ausgangsknoten“ des Kommunikationssystems

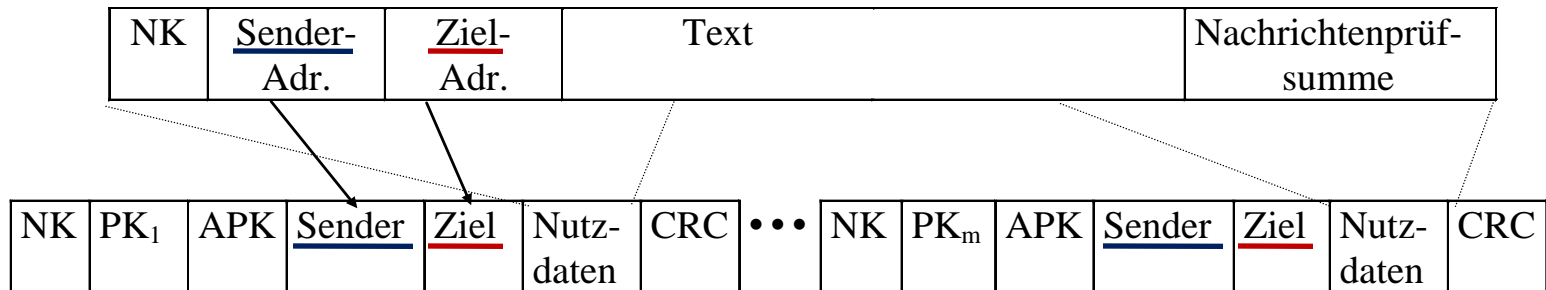




- Prinzip **Paketvermittlung** (packet switching) ? [Fortsetzung]

⇒ Interpretiere  $d_i$  als Pakete (mit a priori vorgegebener Maximallänge, z.B.  $I_{max}$  [kb]),  
wobei Paket  $\equiv$  Teil der ursprüngl. Nachricht; Pakete besitzen jeweils vollständige Adressinfo;  
Paketreassemblierung erst am „Ausgangsknoten“ des Kommunikationssystems oder im Endsystem

Formate der Dateneinheiten bei Paketvermittlung :



PK  $\equiv$  Paket-Nr.

APK  $\equiv$  Anzahl der bei Fragmentierung entstandenen Pakete

**Beurteilung** Nachrichten-/Paketvermittlung :

- \* mögl. *Vorteile* der Nachrichtenverm. :
  - nur Routingentscheidungen pro Nachrichten
- \* mögl. *Nachteile* der Nachrichtenverm. :
  - evtl. erheb. Verzögerung kurzer Nachrichten durch längere
  - umfangreicher Pufferplatz (*variabler* Größe) in VR benötigt

## (c) Varianten der **Paketvermittlung** — Datagramm vs. Virtuelle Verbindungen

- Prinzip **Datagramm** (datagram) ?

einfach(st)e Paketvermittlungstechnik mit freilaufenden Paketen zwischen Sender und Empfänger, ohne sämtl. / die meisten der Eigenschaften virtueller Verbindungen (s.u.)

→ insbesondere keine Erhaltung der ursprünglichen Paketreihenfolge



- Prinzip **Virtuelle Verbindungen** (virtual circuit) ?

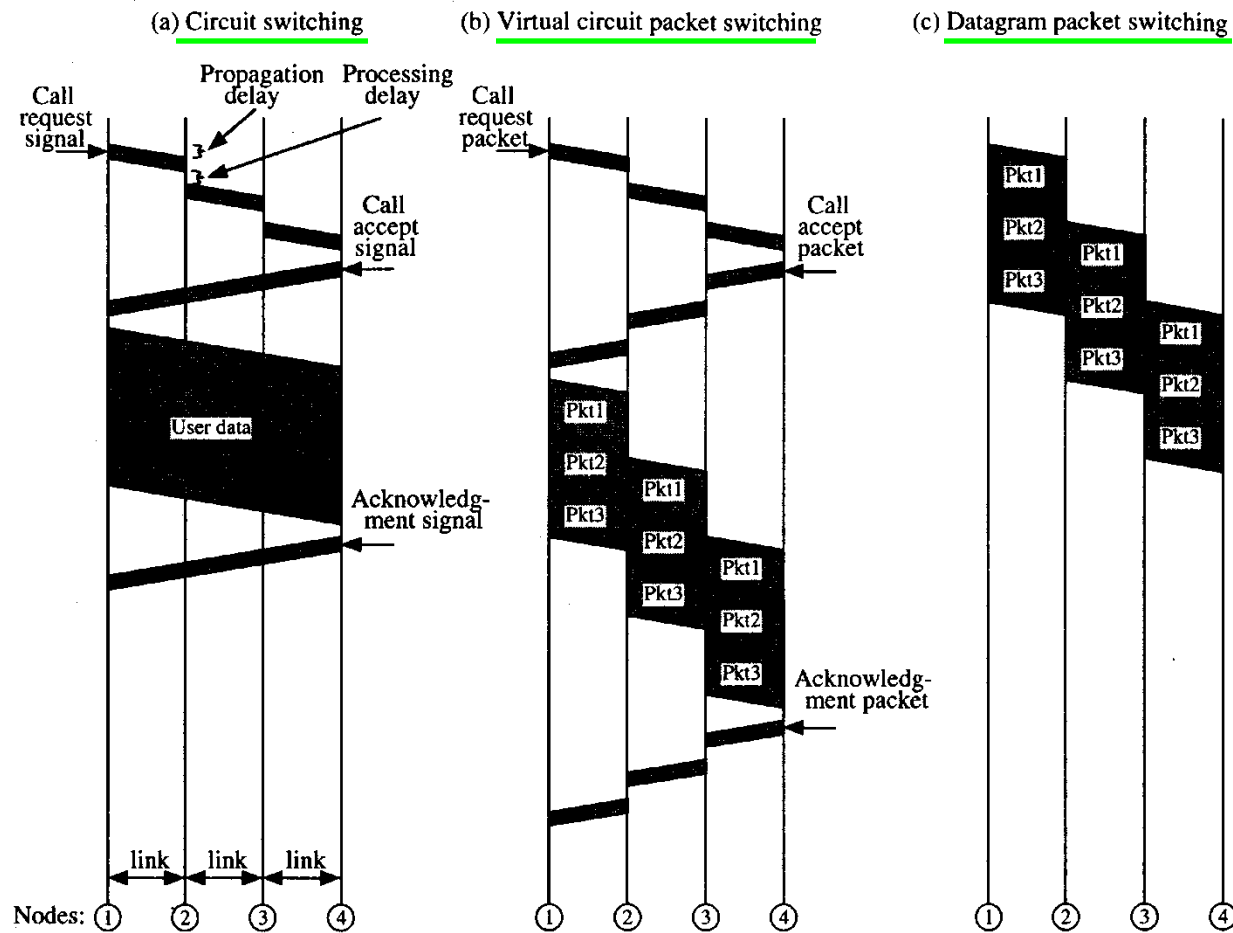
**Virtuelle Verbindung**  $\equiv$  Assoziation (temporär etablierte Kommunikationsbeziehung) zwischen je zwei kommunizierenden Einheiten mit

- Erhaltung/Wiederherstellung der ursprünglichen Reihenfolge der übertragenen Pakete vor Übergabe an Empfänger
- Vermeidung von Duplikaten, Verlusten von Paketen
- Korrektur von Übertragungsfehlern
- Flusskontrolle bezogen auf virtuelle Verbindungen

*ergo* : Bei Datagramm-Technik i.a. Kommunikationsfunktionen (z.B. Fluss-/Fehlerkontrolle) in den Endknoten anzusiedeln, mit teilweise erheblichem Aufwand

# Durchschaltetechnik vs. Verbindungsorientierte/-lose Paketvermittlung

## ➤ Zeitliche Abläufe:



## ➤ Wesentliche Charakteristika:



<b>Circuit switching</b>	<b>Datagram packet switching</b>	<b>Virtual-circuit packet switching</b>
Dedicated transmission path	No dedicated path	No dedicated path
Continuous transmission of data	Transmission of packets	Transmission of packets
Fast enough for interactive	Fast enough for interactive	Fast enough for interactive
Messages are not stored	Packets may be stored until delivered	Packets stored until delivered
The path is established for entire conversation	Route established for each packet	Route established for entire conversation
Call setup delay; negligible transmission delay	Packet transmission delay	Call setup delay; packet transmission delay
Busy signal if called party busy	Sender may be notified if packet not delivered	Sender notified of connection denial
Overload may block call setup; no delay for established calls	Overload increases packet delay	Overload may block call setup; increases packet delay
Electromechanical or computerized switching	Small switching nodes	Small switching nodes
User responsible for message loss protection	Network may be responsible for individual packets	Network may be responsible for packet sequences
Usually no speed or code conversion	Speed and code conversion	Speed and code conversion
Fixed bandwidth	Dynamic use of bandwidth	Dynamic use of bandwidth
No overhead bits after call setup	Overhead bits in each packet	Overhead bits in each packet

Durch Mischformen zwischen Datagramm-Technik (DG) und der Benutzung virtueller Verbindungen (VCs) resultieren folgende

## Varianten der Paketvermittlung :



- I. **reines Datagramm** → **UDP/IP**
  - DG : zwischen kommunizierenden Prozessen
  - VCs : nirgendwo
- II. **geschichtete VCs auf Datagramm-Basis**, vgl. (1) (a) s.u. → **TCP/IP**
  - DG : zwischen ARn bzw. Datenendeinrichtungen
  - VCs : nur bekannt in ARn
- III. **Realisierung der VCs in „Randknoten“** des Vermittlungsnetzes (VN), vgl. (1) (b) s.u.
  - DG : im „Innern“ des VN
  - VCs : bekannt für jeweilige „Randknoten“ des VN
- IV. **ungeschichtete virtuelle Verbindungen**, vgl. (2) s.u.
  - DG : nirgendwo
  - VCs : zwischen kommuniz. Prozessen (durchgehend bekannt)

## Implementierungsmöglichkeiten für virtuelle Verbindungen (VCs)

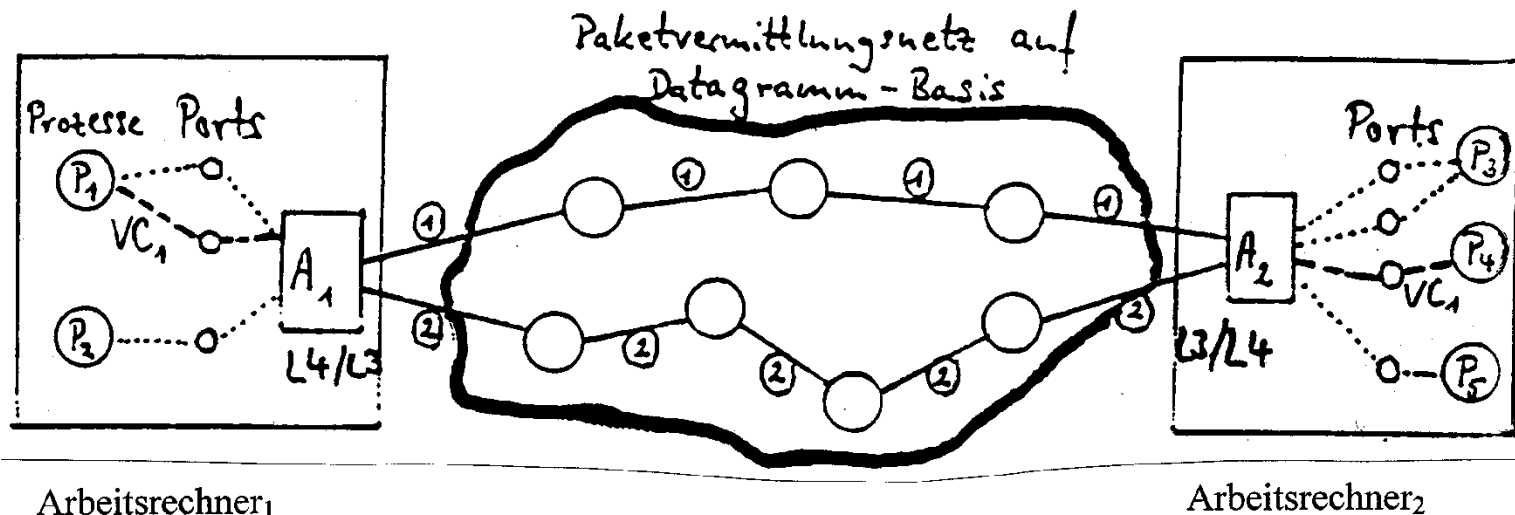
### (1) „geschichtete“ ( *layered* ) virtuelle Verbindungen auf Datagramm-Basis :

#### (a) Realisierung der virtuellen Verbindungen in den **Arbeitsrechnern**

(durch die Instanzen  $A_1$  und  $A_2$  in der geg. Abb., s.u., L3/L4 = OSI-Schichten 3/4);  
VCs transparent für das Vermittlungsnetz.



- Datagramm-Schnittstelle vom Arbeitsrechner zum Vermittlungsnetz
- Bereitstellung von VCs für die kommunizierenden Prozesse  $P_i$  in den Arbeitsrechnern.



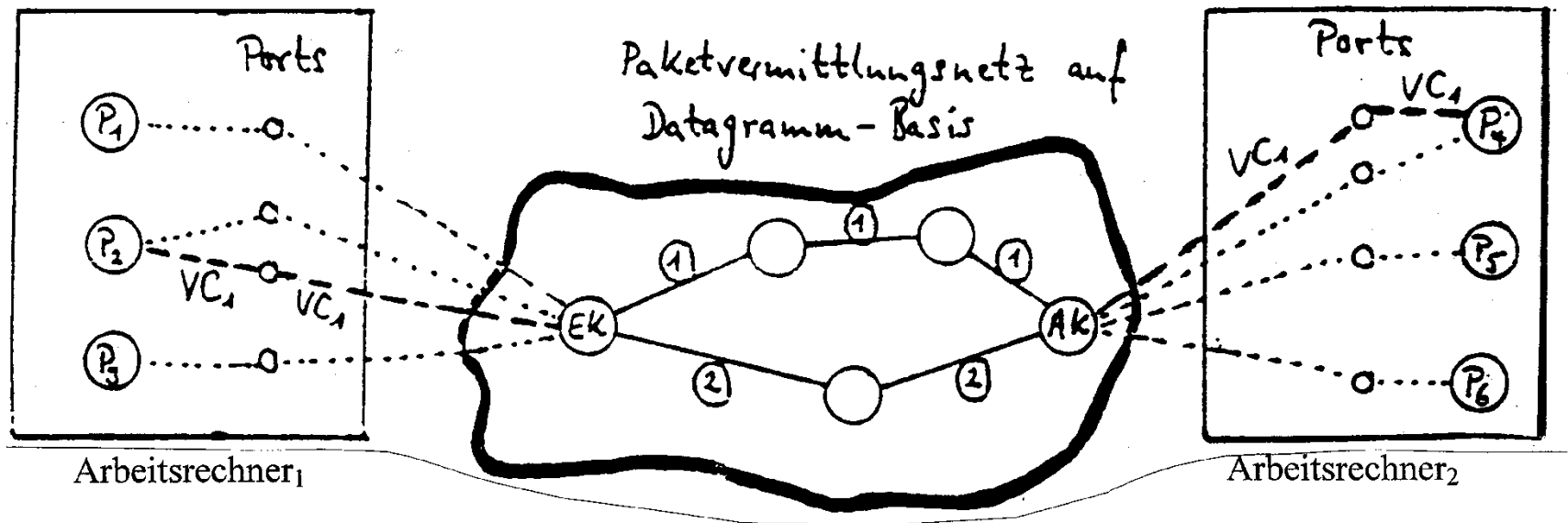
Transport von Paketen einer virtuellen Verbindung über alternative Wege zwischen den Arbeitsrechnern (z.B. Wege ①, ② in obiger Abb.) möglich.

→ vgl. TCP/IP

(b) Realisierung der virtuellen Verbindungen in **Eingangs-** (EK) und **Ausgangsknoten** (AK) des Vermittlungsnetzes; VCs nur für EK und AK, jedoch nicht im „Netzzinnern“, sichtbar.

– Schnittstelle zum Vermittlungsnetz :

Bereitstellung von VCs durch das Netz und evtl. zusätzlich direkte Zugriffsmöglichkeit auf die Datagramm-Dienste des Netzes.



→ vgl. X.25 : histor. TRANSPAC-Lösung (Frankr.)

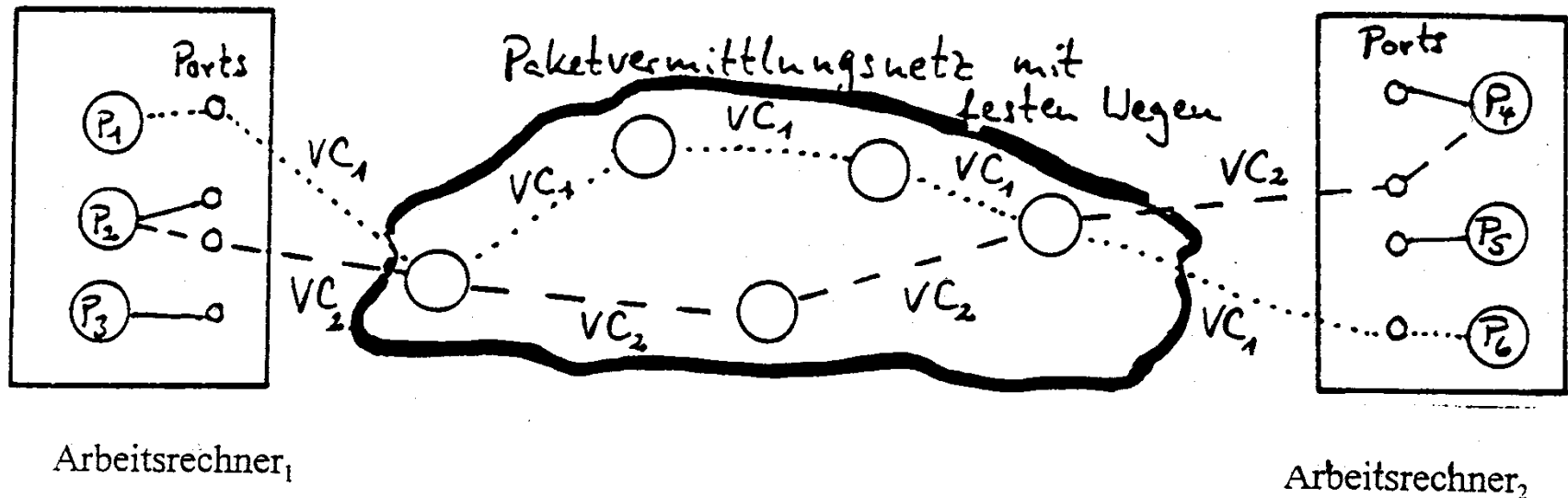


## (2) „ungeschichtete“ ( *unlayered* ) virtuelle Verbindungen auf der Basis fester Wege

Realisierung von virtuellen Verbindungen durch Benutzung genau eines Weges im Vermittlungsnetz für sämtl. (innerhalb der betreffenden VC) zu übertragenden Pakete; virtuelle Verbindungen für **sämtliche Knoten** des Vermittlungsnetzes **sichtbar**.

→ vgl. **X.25** : histor. DATAPAC-Lösung (Kanada); **X.75**  
auch : **ATM** (virtual circuits)

- Schnittstelle zum Vermittlungsnetz : Bereitstellung von VCs durch das Netz
- Wahl des festen Weges für eine VC bei Verbindungsaufbau (durch zentrale Instanz oder dezentralisiert)
- evtl. Reservierung von Betriebsmitteln für eine VC.





## Vorteile der **Datagramm**-Technik :

- günstiges Verhalten bei Anwendungen mit beträchtlichen Fluktuationen im Verkehrsaufkommen (z.B. bei Büroautomatisierung, bei Datenbankabfragen/-updates) oder bei Anwendungen ohne Bedarf nach vernachlässigbarer Fehlerrate (z.B. Übertragung digitalisierter Sprache);
- Anwendungen mit Mehrpunkt-Verbindungen zwischen kommunizierenden Prozessen (*“Multicast“*) einfacher zu unterstützen als mit Hilfe von Punkt-zu-Punkt-VCs;
- Möglichkeit der Benutzung alternativer Wege;
- einfachere Schnittstelle und weniger aufwändige Implementierung der Vermittlungsfunktionen als bei VCs;
- einfachere Kopplung autonomer Netze mit Datagramm-Vermittlung als solcher mit VCs.



## Vorteile der **virtuellen Verbindungen** :

- Reduktion des Umfangs zu übertragender Daten durch die Verringerung der Kontrollinformation für Pakete bei VCs gegenüber Datagramm-Technik;
- Anwendungen mit Punkt-zu-Punkt-Verbindungen zwischen kommunizierenden Prozessen (z.B. Terminaldialog) in natürlicher Weise auf VCs abbildbar;
- vereinfachte Fluss- und Sättigungskontrolle durchführbar (Vermeidung bestimmter bei Datagramm-Technik möglicher Verklemmungssituationen, wie z.B. *„store-and-forward lock-up“*; Flusskontrolle beziehbar auf VCs).

## Zu Vermittlungstechniken :

**Zellenvermittlung** (*cell switching*), vgl. **ATM** (**A**synchronous **T**ransfer **M**ode)

$\Rightarrow \approx$  Paketvermittlung



aber : 1) Pakete klein (53 Byte : 48 Byte Nutzdaten  
+ 5 Byte Kontrollinfo.)

2) Pakete konstante Länge (nicht nur Maximallänge)

1)  $\wedge$  2)  $\Rightarrow$  *schnelle Paketvermittlung*, da :

- konstante Länge vereinfacht Interpretation  
(da auch Zellenaufbau identisch für alle Zellen)  
und Betriebsmittelverwaltung (für Speicherung  $\wedge$  Übertragung)
- kleine Pakete implizieren kleine Übertragungszeiten,  
z.B.  $v_D = 10 \text{ Gb/s} = 10^{10} \text{ b/s}$

Zellengröße  $\approx 4 \cdot 10^2 \text{ b}$ , d.h. für Übertragungszeit  $t_{üz}$  von  
1 Zelle gilt :

$$t_{üz} = L/v_D \approx 10^{-10} \cdot 4 \cdot 10^2 \text{ s} = 4 \cdot 10^{-8} \text{ s} = 4 \cdot 10^{-2} \mu\text{s} = 0,04 \mu\text{s} = 40 \text{ ns}$$

*Wie Zellen erhalten ?*

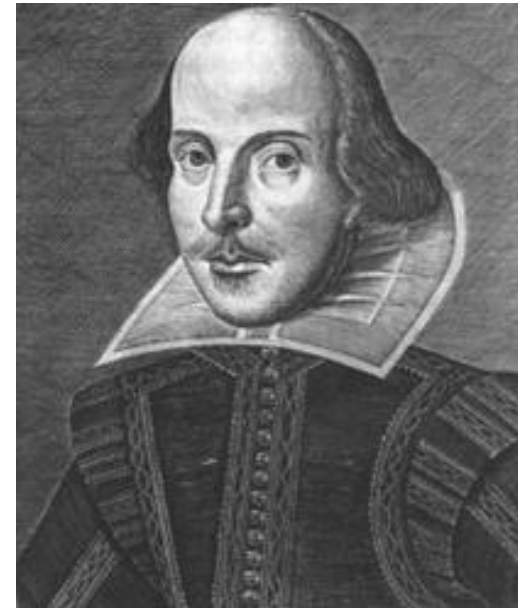
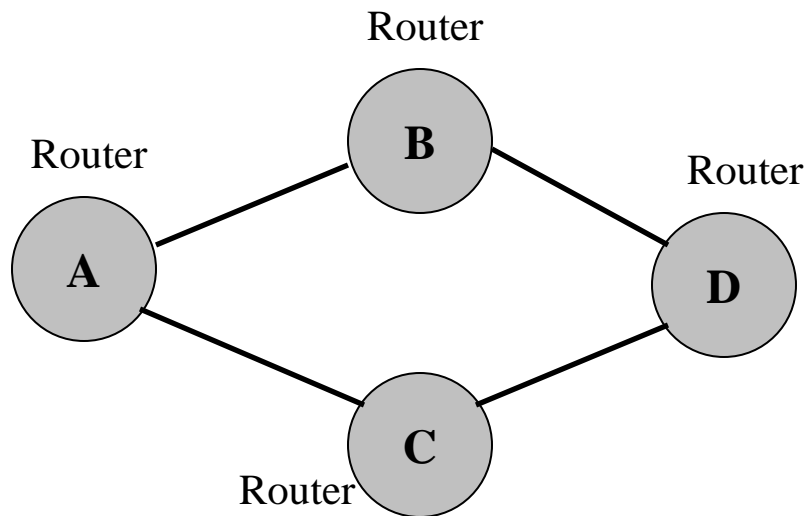
- a) Fragmentierung/Reassemblierung von Paketen (L3) oder Übertragungsblöcken (L2);  
*nota bene* : letzte Zelle auffüllen („dummy“)
- b) Verkettung kleinerer Dateneinheiten (z.B. PCM-Abtastwerte von je 1 Byte Länge)

Durchsatzoverhead  $d_{ovh}$  signifikant :  $d_{ovh} = \frac{5}{53} \approx 10\%$  (!!)

*aber* : was kümmert uns Durchsatz in Hochgeschwindigkeitsnetzen (?!)

# Das Problem der **Wegeermittlung**/des **Routing** durch ein vermaschtes Kommunikationssystem

[„Uraltproblem“ → vgl. William Shakespeare]



Ausgangssituation : A sendet mit Ziel D

→ ... die „alte Shakespeare-Frage“ : **To B or not to B ?**

## "Wer routet so spät durch Nacht und Wind..." (Teil I)

(Autor unbekannt)

Frei nach Johann Wolfgang von Goethe:

*Der Erlkönig (EDV-Version)*

Wer routet so spät durch Nacht und Wind?

Es ist der Router, er routet geschwind!

Bald routet er hier, bald routet er dort

Jedoch die Pakete, sie kommen nicht fort.

Sie sammeln und drängeln sich, warten recht lange  
in einer zu niedrig priorisierten Schlange.

Die Schlangen sind voll, der Router im Stress,  
da meldet sich vorlaut der Routingprozess  
und ruft "All Ihr Päckchen, Ihr sorgt Euch zu viel,  
nicht der IP-Host, nein, der Weg ist das Ziel!"

Es komme gar bald einem jeden zu Gute  
eine sorgsam geplante und loopfreie Route.

Des Netzes verschlungene Topologie  
entwirr' ich mit Dijkstras Zeremonie.

Der Lohn, eine herrliche Routingtabelle,  
dort steh'n sogar Routen zu Himmel und Hölle.

## **"Wer routet so spät durch Nacht und Wind..." (Teil II)**

Vergiftet der Rückweg, das Blickfeld gespalten,  
mit RIP wird die Welt nur zum Narren gehalten.  
Doch OSPF durchsucht schnell und bequem  
Mein ganz und gar autonomes System.  
Für kunstvolle Routen, das vergesst bitte nie,  
benötigt man Kenntnis der Topologie.  
Zu Überraschungs- und Managementzwecken  
durchsuch' ich mit RMON die hintersten Ecken.  
Kein Winkel des Netzes bleibt vor mir verborgen,  
mit SNMP kann ich alles besorgen.  
Wohlan nun, Ihr Päckchen, die Reise beginnt,  
Mit jeder Station Eure Lebenszeit rinnt.  
Doch halt, Ihr Päckchen, bevor ich's vergesse:  
Besorgt euch mit NAT eine neue Adresse!"  
"Mein Router, mein Router, was wird mir so bang!  
Der Weg durch das WAN ist gefährlich und lang."  
"Mein Päckchen, mein Päckchen, so fürchte Dich nicht,  
denn über Dich wacht eine Sicherungsschicht."

## **"Wer routet so spät durch Nacht und Wind..." (Teil III)**

"Mein Router, mein Router, was wird mir so flau!

Dort draußen am LAN-Port, da wartet die MAU!"

"Mein Päckchen, mein Päckchen Dir droht nicht der Tod,

denn über Dich wacht ja der Manchester-Code.

Doch halte dich fern von der flammenden Mauer.

Die sorgt selbst bei mir noch für ängstliche Schauer."

"Mein Router, mein Router, wie glänzt dort voll Tücke

der schmale und schlüpfrige Weg auf der Brücke."

"Oh weh! Das Netz ist mit Broadcasts geflutet.

Ach hätt' ich doch niemals zur Brücke geroutet!

Mein Päckchen, den Kopf hoch, Du musst nicht verzagen,

an Dich wird sich niemals ein Bitfehler wagen."

Schnell wie der Wind geht die Reise nun weiter

durch helle und funkelnde Lichtwellenleiter.

"Mein Päckchen, mein Päckchen, willst Du mit mir gehen?

Die Wunder des Frame-Relay-Netzes ansehen?"

Mein Router, mein Router, ja hörst Du denn nicht,

was die WAN-Wolke lockend mir leise verspricht?"

## **"Wer routet so spät durch Nacht und Wind..." (Teil IV)**

"Glaub mir, mein Päckchen, im LAN, da entgeht

Dir sowieso Lebens- und Dienstqualität.

Reise nur weiter ganz ruhig und sacht

Quer durchs ATM-Netz mit FRF.8 ."

"Mein Router, mein Router, man hat mich verführt,  
zerlegt, verschaltet und rekombiniert!"

"Mein Päckchen, das macht nichts, nun sparen wir viel,  
ein VPN-Tunnel, der bringt Dich ans Ziel.

DiffServ und TOS-Feld, merk' Dir die Worte,  
die öffnen zu jedem Router die Pforte."

Finster der Tunnel, die Bandbreite knapp,  
wie schön war die Backplane im eigenen Hub.

Am Ende des Tunnels: Das Päckchen ist weg,  
vernichtet vom Cyclic Redundancy Check.

*Quelle:* [www.unterhaltungsspiele.com](http://www.unterhaltungsspiele.com)

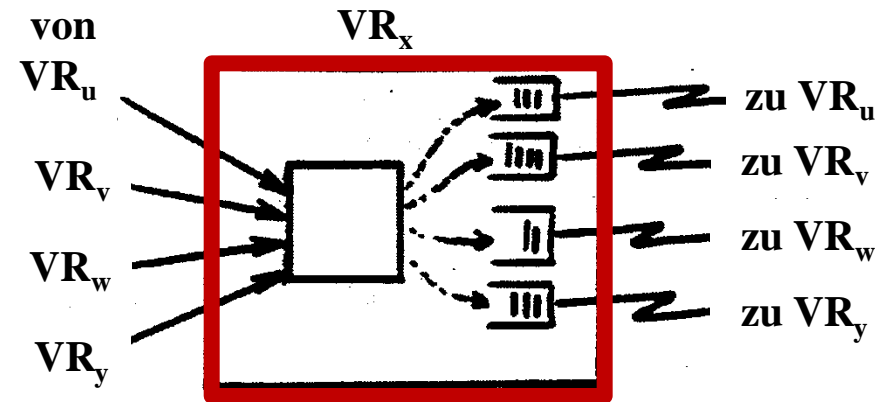
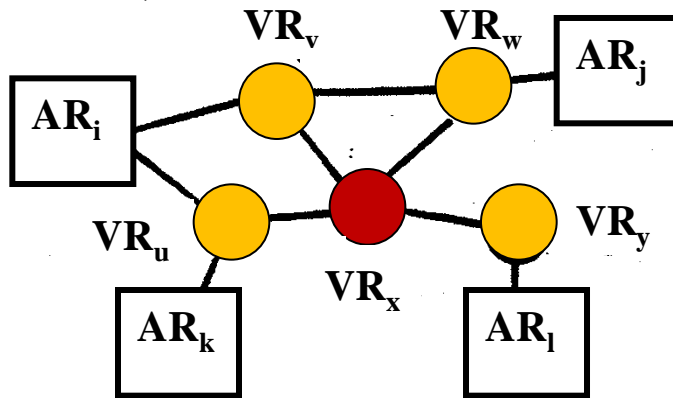


# 6.3 Wegeermittlung durch ein vermaschtes Kommunikationssystem

## 6.3.1 Ziele, Beispiele, Klassifikation

Beispiele einer Topologie eines  
Rechnernetzes:

vereinfachtes Warteschlangenmodell  
des Vermittlungsrechners  $VR_x$ :



**Wegeermittlung (Routing) :**

Zuordnung einer jeden empfangenen und weiterzuleitenden Dateneinheit zu einer Ausgangswarteschlange (vgl. obiges Warteschlangenmodell).

**Pfad (path) :**

Eine Menge hintereinanderliegender Übertragungsleitungen, die einen gegebenen Sendeknoten S mit einem Empfangsknoten E verbinden.

Beispiel (vgl. obige Topologie) :

Pfad zwischen  $AR_i$  und  $AR_j$  gebildet durch die Leitungen  $AR_i \rightarrow VR_v$ ,  
 $VR_v \rightarrow VR_w$  sowie  $VR_w \rightarrow AR_j$  .

**Routing auf einfachem Pfad (single path) :**

Benutzung eines einzigen Pfades zwischen S und E, um eine Menge von Daten von einem Sendeknoten S zu einem Empfangsknoten E zu leiten.

**Routing auf multiplen Pfaden (multiple paths) :**

Benutzung mehrerer zwischen S und E existierender Pfade, um Daten von S nach E zu leiten.

Vorteil bei Routing sämtlicher Dateneinheiten einer (geschichteten) virtuellen Verbindung über einen einfachen Pfad :

- Beibehaltung der ursprünglichen Sequenz der Dateneinheiten auf ihrem Weg vom Sender zum Empfänger (keine aufwändige Umordnung der Daten bei Empfänger notwendig)

- Ziele der Wegeermittlung :

- optimale Anpassung der Datenwege an Stöorzustände
- Lastverteilungen (insbesondere wichtig bei lokaler Überlastung von Knoten und Leitungen im Kommunikationssystem KSS)

- Routing im Vermittlungsknoten  $VR_x$  : i.a. Wahl der **kürzesten Pfade** (*shortest path*) zwischen  $VR_x$  und dem Empfänger der weiterzuleitenden Dateneinheit.

- „shortest path“ - Bestimmung in Anbetracht einer Kostenfunktion

→ mögliche **Kostenfunktionen** :

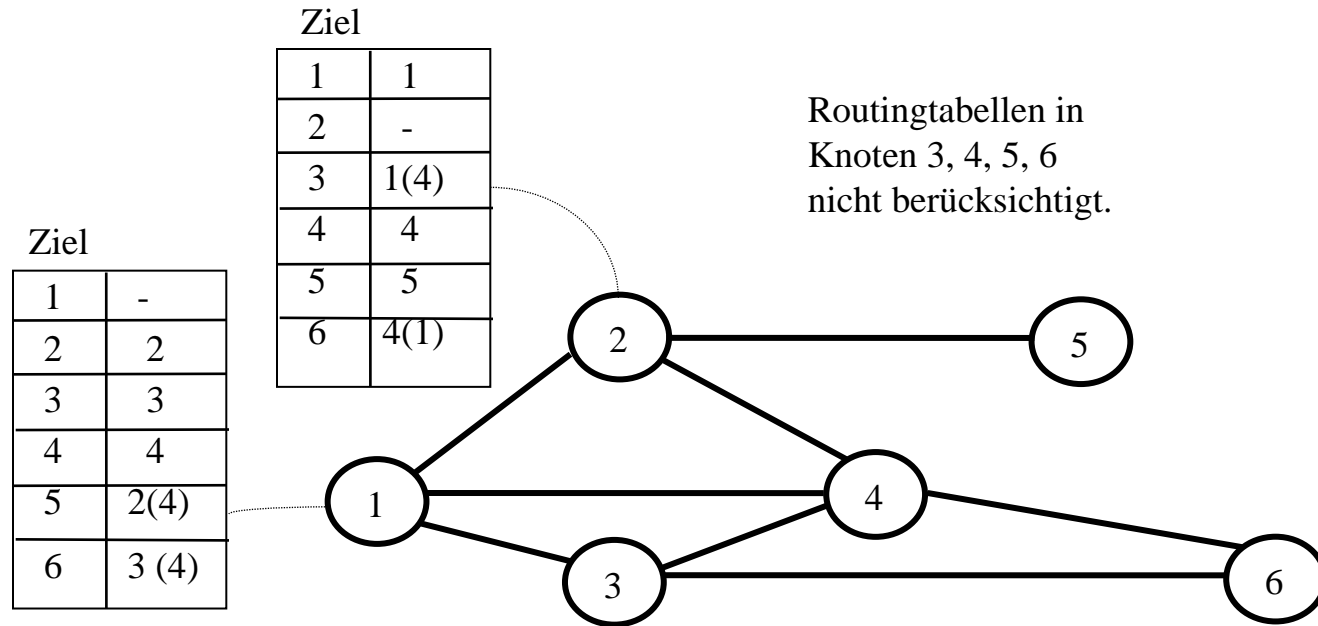
- Kosten eines Pfades =  $\Sigma$  Kosten der auf Pfad befindlichen Leitungen;  
Leitungskosten z.B. umgekehrt proportional zu Leitungskapazität oder geprägt durch aktuelle Auslastung und Fehlerrate der Leitung
- Kosten eines Pfades = Anzahl auf dem Pfad liegender Knoten (*number of hops*)
- Kosten eines Pfades abhängig von aktuell beobachteter Verzögerung beim Transport von Dateneinheiten über diesen Pfad



Bemerkung : gewählte Kostenfunktion evtl. abhängig vom Typ des zu transportierenden Verkehrs (Dialog-, Dateitransferanwendung)

- Beispiel eines einfachen Wegeermittlungsalgorithmus :

Knoten mit Wegeermittlungsinformation (in sog. Routingtabellen)



Aufbau der Routingtabellen im Beispiel : für jedes Ziel → Angabe des nächsten Knotens auf optimalem Pfad (in Klammern : Knoten für Ersatz-Pfade)

**Vorgehensweise** : bei Erhalt einer weiterzuleitenden Dateneinheit

→ Bestimmung des nächsten Knotens anhand der Routingtabelle (bei Ausfall der Leitung zu diesem Knoten : angegebener Ersatzpfad zu benutzen)

## **Ziele** bei Wegeermittlung :

- Vermeidung von „Ping-Pong-Effekten“
- Verweilzeitminimierung
- Anpassung an Topologieänderungen
- kurze Speicherbelegungen in VRn

## **Ebenen des Routing**

(d.h. Routing auf welchen Schichten der Protokollhierarchie ?)

- Routing innerhalb der Datensicherungsschicht (z.B. in Multihop-ad-hoc-Netzen)
- Routing auf Vermittlungs-/Netzwerkschicht in einem Vermittlungsnetz
- Netzübergreifendes Routing auf Vermittlungs-/Netzwerkschicht bei Kopplung verschiedener Vermittlungsnetze (z.B. Routing durch Transitnetze)
- Routing innerhalb der Anwendungsschicht (z.B. in Peer-to-Peer-Netzen)

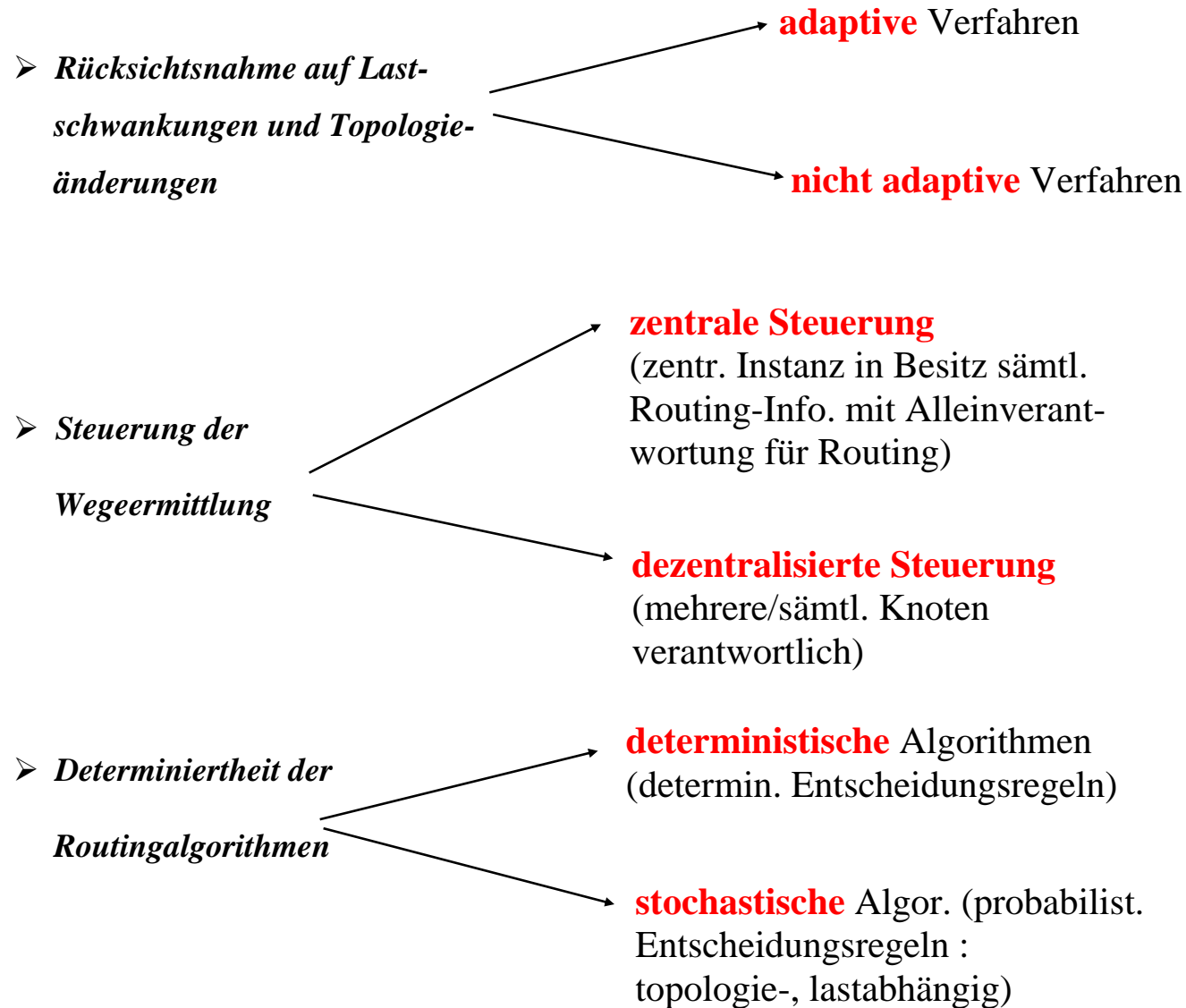


### ***Nota bene:***

Wegeermittlung u.a. trivial bei folgenden Topologien:

Punkt-zu-Punkt, Bus, Broadcastsysteme, Ring, Stern, vollständige Vermaschung, ...

## Kriterien für Klassifikation von Routing-Algorithmen :



## Aktionen zur Realisierung adaptiver Routing-Verfahren



- Messung Kommunikationssystem (KSS)-Zustand
- Verteilung neuer KSS-Zustandsinfo an Zentrale bzw. mehrere Knoten
- Berechnung des aktuellen KSS-Zustands durch Zentrale bzw. mehrere Knoten
- Berechnung der „optimalen“ Wege
- nur bei zentraler Steuerung :  
Bekanntgabe der optimalen Wege für andere Knoten
- update Routingtabellen

### *Qualität* von Routing-Algorithmen *abhängig von*

- Implementierungsaufwand
- Overhead an zusätzlicher Kontrollinfo
- Verfügbarkeit der Instanz(en) zur Steuerung der Wegeermittlung und/oder KSS-Vermessung
- Reaktionsfähigkeit auf Lastschwankungen und Topologieänderungen
- Wahrscheinlichkeit für Fehlersituationen (Deadlocks, Zyklen,...)

---

**Performance Criteria**

Number of hops  
Cost  
Delay  
Throughput

**Decision Time**

Packet (datagram)  
Session (virtual circuit)

**Decision Place**

Each node (distributed)  
Central node (centralized)  
Originating node (source)

**Network Information Source**

None  
Local  
Adjacent node  
Nodes along route  
All nodes

**Network Information Update Timing**

Continuous  
Periodic  
Major load change  
Topology change



## 6.3.2 Nicht adaptive Routing-Verfahren

Klassifikation nicht adaptiver Routing-Verfahren :

- (1) **zufälliges Routing** (random)
- (2) **„überflutendes“ Routing** (flooding)
  - (a) sämtlicher Leitungen
  - (b) ausgewählter Leitungen
- (3) **festes Routing** (fixed)
  - (a) einfache Pfade
  - (b) doppelte Pfade
    - Lastaufsplittung
    - ein alternativer Pfad
  - (c) multiple Pfade



## Grobbeschreibung der Verfahren

### (1) **Zufälliges Routing**

- nächster Knoten zufällig gewählt ohne Berücksichtigung des Netzzustandes
- Ausgangsleitungen in einem Knoten mit entweder identischen oder unterschiedlichen Wahrscheinlichkeiten für ihre Benutzung
- Vermeidung von Zyklen (z.B. durch Verbot eines wiederholten Durchgangs durch denselben Knoten)

### (2) **Überflutendes Routing**

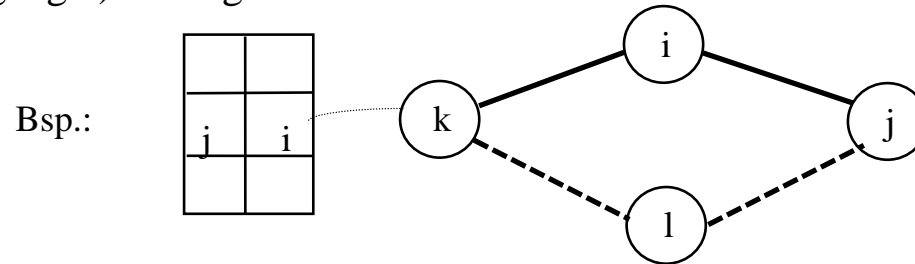
- Absenden einer erhaltenen Dateneinheit an sämtliche direkt benachbarte Knoten (ausgenommen der Knoten, von denen Dateneinheit erhalten wurde)
- Auftreten von Zyklen zu vermeiden
- beträchtliches zusätzliches Verkehrsaufkommen wesentlicher Nachteil



### (3) festes Routing

#### (a) einfache Pfade

- jedes in Knoten  $k$  bekannte Ziel (z.B. Knoten  $j$ ) von  $k$  über genau eine (Ausgangs-)Leitung zu erreichen



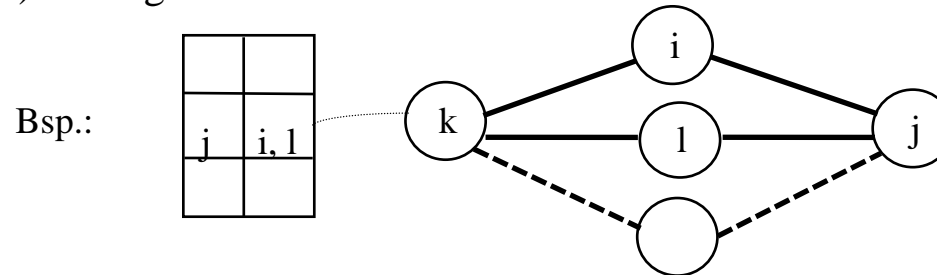
- evtl. existierende alternative Pfade von Knoten  $k$  nach  $j$  bleiben auch bei Leitungsausfall unberücksichtigt
- Vermeidung von Zyklen gewährleistet
- Nachteile :
  - bei Ausfall von Knoten/Leitungen keine Möglichkeit, Ersatz-Pfade für die vom Ausfall betroffenen Pfade zu definieren
  - keine Möglichkeit der Anpassung an Lastschwankungen



### (3) festes Routing (Fortsetzung):

#### (b) doppelte Pfade

- jedes in Knoten  $k$  bekannte Ziel (Knoten  $j$ ) von  $k$  über genau zwei (Ausgangs-)Leitungen zu erreichen



- Wahl der für konkrete Übertragung benutzten Ausgangsleitung deterministisch oder zufällig (Wahl des ersten Pfades mit Wahrscheinlichkeit  $p$  und der Alternative mit Wahrscheinlichkeit  $1-p$ )
- fällt Knoten  $i$  aus und dennoch wird weiterhin Weg über  $i$  gewählt  
→ Verfahren bleibt nicht adaptiv im strengen Sinne;  
Verfahren wird hybrid, wenn bei Ausfall von  $i$  nur noch Alternativweg Berücksichtigung findet
- Verfahren in Praxis recht gut brauchbar (insbesondere bei permanent starker Belastung mit geringen Änderungen des Verkehrs, z.B. bezogen auf Verkehrsmatrix)

#### (c) multiple Pfade

- entsprechende Verallgemeinerung von (b) :  
i.a. Existenz von  $\geq 2$  Alternativen für manche oder gar für alle Pfade



## 6.3.3. Adaptive Routing-Verfahren

Klassifikation **adaptiver Routing-Verfahren** :



- (1) **lokales/isoliertes Routing** (lokal)
  - (a) „hot potato“ - Routing
  - (b) lokale Abschätzung der Übertragungsverzögerungen
- (2) **verteiltes Routing** (distributed)
  - (a) Nachbarknoten 1. Ordnung
  - (b) Nachbarknoten 1.+2. Ordnung
  - (c) sämtliche Knoten
- (3) **Delta-Routing**
- (4) **Zentralisiertes Routing** (centralized)

Kriterien für Klassifikation :

- Menge und Lage der Knoten zur Sammlung und Verarbeitung der Status-information bzgl. KSS-Zustand
- die Art der Steuerung der Wegeermittlung :  
zentral/dezentral



## Grobbeschreibung der Verfahren

### (1) **Lokales Routing**

- Erfassung der Statusinformation bzgl. KSS-Zustand durch jeden Knoten selbst (ohne Zusatzinformation benachbarter Knoten, daher auch *isoliertes Routing*)

#### (a) **„hot potato“ Routing**

- erfasste Statusinformation : Warteschlangenbelegung für die (Ausgangs-)Leitungen
- abzusendende Dateneinheit wird Ausgangsleitung mit aktuell geringster Warteschlangenbelegung zugeordnet (ohne Rücksicht auf das adressierte Ziel)
- geringe Dauer für Speicherbelegungen erzielt, jedoch „übles“ Leistungsverhalten und daher nicht praktikabel

noch (1) **Lokales Routing** :

(b) **lokale Abschätzung der Übertragungsverzögerungen**

- in Knoten  $k$  : Zuordnung einer abzusendenden Dateneinheit mit Ziel  $j$  dergestalt, dass (Ausgangs-)Leitung zu demjenigen Knoten  $i$  gewählt wird, für dessen Index  $i$  gilt :

$$T_k(i, j) = \min_{m \in M} T_k(m, j)$$

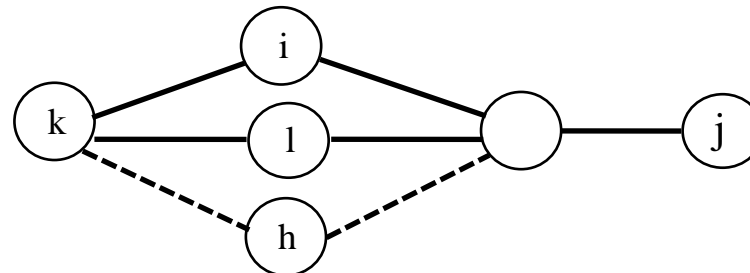
Notation :

$M$  = Menge der direkt benachbarten Knoten-Nummern, über die Dateneinheiten des Knotens  $j$  erhalten werden können,

*sowie*

$T_k(m, j)$  = beobachtete Verzögerung von, im Knoten  $k$  über den Nachbarknoten  $m$ , ankommenden Dateneinheiten  
(Absender : Knoten  $j$ )

Bsp.:



$$M = \{i, l, h\}$$

noch (1) **Lokales Routing** :

(b) **lokale Abschätzung der Übertragungsverzögerungen** (Fortsetzung)

- Annahme des Verfahrens :  
Verzögerung auf einem Pfad in eine Richtung  $\approx$  Verzögerung auf Pfad in entgegengesetzter Richtung
- update in Knoten  $k$  für  $T_k(m, j)$  sobald von Nachbarknoten  $m$  eine Dateneinheit mit Absender  $j$  erhalten wird  
→ **Nachteile** :
  - insbesondere bei langem Pfad von  $j$  nach  $k$  :  
Abschätzung veraltet
  - regelmäßiger Verkehr von  $j$  nach  $k$  Voraussetzung für regelmäßige updates
- Beobachtung der Übertragungsverzögerungen von Dateneinheiten  
z.B. auf der Basis von Zeitstempeln möglich, die beim Verlassen des Sendeknotens zugewiesen werden



## (2) Verteiltes Routing

### (a) Nachbarknoten 1. Ordnung



- Definition der Menge der **Nachbarknoten  $n$ -ter Ordnung** für einen Knoten  $k$  :  
Menge der Knoten, die über  $n - 1$  zwischenliegende Knoten von  $k$  aus erreicht werden bzw. diejenigen Knoten, die  $n$  “Hops“ entfernt sind.  
→ *Nachbarknoten 1. Ordnung* = Menge der Knoten, zu denen direkt eine Leitung führt (auch : direkt benachbarte Knoten)
- Routing - Information (bzgl. beobachtetem Zustand des KSS) nur zwischen direkt benachbarten Knoten ausgetauscht (z.B. periodisch oder ereignisgesteuert)
- Routing - Entscheidungen : auch hier basierend auf Schätzungen (eigene und die der direkt benachbarten Knoten) der zu erwartenden Übertragungsverzögerungen bei Wahl von Ausgangsleitung  $m$ , um Ziel  $j$  zu erreichen
- Implementierung dieses Verfahrens als ursprünglicher Routing-Algorithmus des ARPA-Netzes (periodischer Austausch von Routing-Information nach 2/3 sec.)

(2) **Verteiltes Routing** (Fortsetzung):

(b) **Nachbarknoten 1. + 2. Ordnung**

- entsprechend (2) (a)
- mehr Overhead und Komplexität, aber i.a. als Lohn bessere Abschätzungen

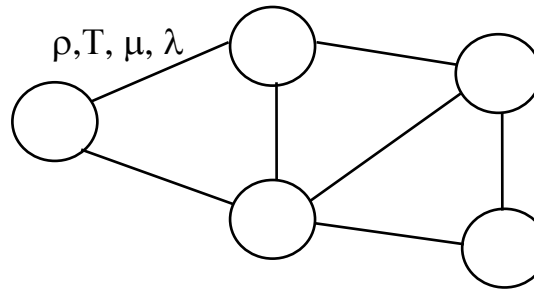
(c) **sämtliche Knoten**

- entsprechend (2) (b) mit Nachteilen Overhead und Komplexität in verschärfter Form

# Verteiltes Routing

(sämtliche Knoten)

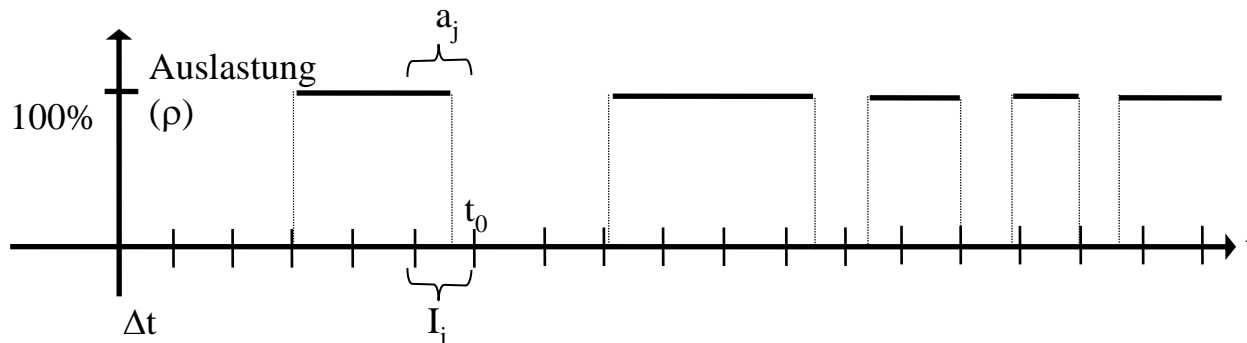
## (1) Was wird vermessen ?



$\rho$ : Auslastung  
 $T$ : Verzögerungs-/Verweilzeit  
 $\mu$ : Bedienrate (hier: Paketlänge)  
 $\lambda$ : Ankunftsrate

→ Leitungsauslastung  
(Knotenauslastung)

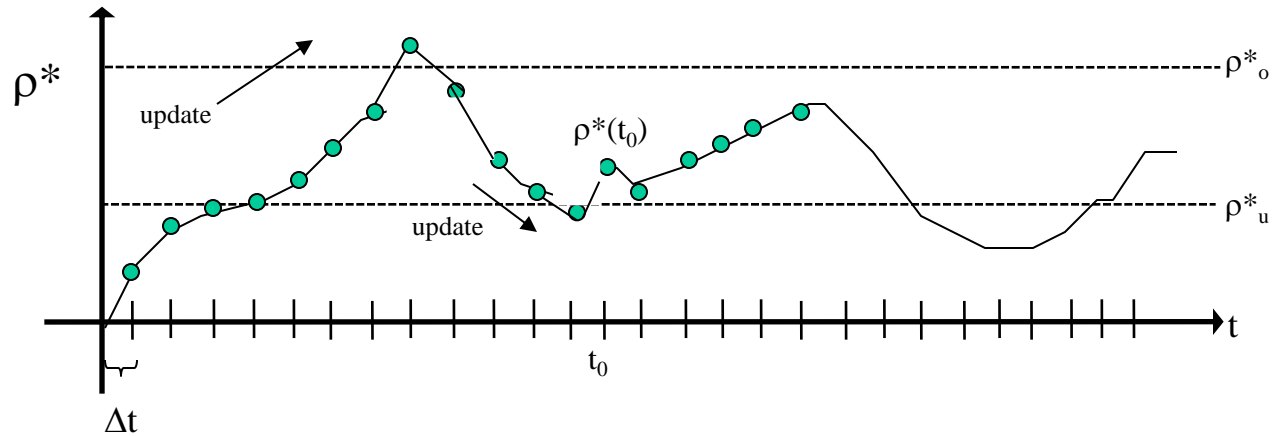
## (2) Wie wird vermessen ?



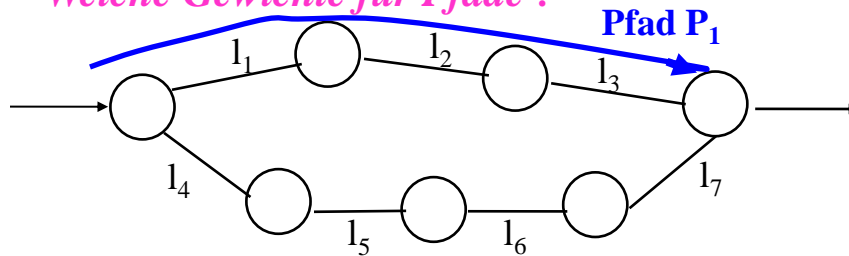
→ jüngste Messungen erhalten stärkeres Gewicht als „veraltete“ Messungen,  
 vgl.  $\rho^*(t_0)$  als geschätzte Momentanauslastung zu  $t = t_0$ ,  $c_i \equiv$  „Gewichte“,  $c_i \geq 0$  :

Sei  $a_j \equiv$  (gemessene) Auslastung in  $I_j \Rightarrow \rho^*(t_0) = c_0 a_j + c_1 a_{j-1} + \dots + c_k a_{j-k}$

**Hinweis:** zu Auslastungsschätzungen und Schwellwertsystemen, vgl. auch Paper von B.E. Wolfinger, J. Wolf & G. Le Grand in : *Internat. Journal of Communication Systems*, Wiley, Vol. 18, Issue 4, May 2005, pp. 373-394 **sowie** Dissertation Dr. Jürgen Wolf (FB Informatik, Uni HH 2007)

(3) *Wann update ?*

→ Schwellwerte einführen („thresholding“), z.B.  $\rho_o^*$ ,  $\rho_u^*$  für oberen/unteren Schwellwert

(4) *Welche Gewichte für Pfade ?*

$$g(P_1) = g(l_1) + g(l_2) + g(l_3)$$

mit

$$g(l_i) \equiv 1/d_i \cdot 1/(1 - \rho_i^*)^2$$

wobei  $d_i \equiv$  Datenrate von Link  $l_i$

### (3) **Delta-Routing**

- Mischform zwischen zentraler und dezentraler Steuerung
- Zentrale Instanz sammelt Statusinformation bzgl. Zustand des gesamten KSS; Transfer dieser Information an die Knoten (evtl. relativ selten)
- Verwendung der Statusinformation der zentralen Instanz durch die Knoten zur Wegeermittlung unter zusätzlicher Berücksichtigung von Beobachtungen der lokalen Knotenumgebung

### (4) **Zentralisiertes Routing**

- mögliche Verfahren in dieser Klasse ergeben sich direkt aus der Definition der zentralen Steuerung
- keine gegenseitige Abstimmung der Routing-Entscheidungen der einzelnen Knoten erforderlich, jedoch evtl. Verfügbarkeitsprobleme (Ausfall der zentralen Instanz) und mehr produzierter Overhead an zusätzl. zu transportierenden Dateneinheiten als z.B. bei lokalem, adaptivem Routing und manchen Verfahren des Delta-Routing.

# 6.4 Namensgebung und Adressierung

## 6.4.1. Begriffe und Beispiele

- Def. **Name** : Bezeichnung (Kette von Symbolen) mit Bezug/Verweis auf ein Objekt.
- Def. **Adresse** : Angabe zur Lokalisierung eines Objektes.



Eigenschaften von Namen:

- feste/variable Länge
- mögliche Symbole: Bits, Zeichen
- häufige Konventionen bei Verwendung von Zeichen als Symbole:
  - alphanumer., alphabet., numer., spez. Zeichen

➤ Beispiele für mit Namen zu versehende Objekte in Rechnernetzen:

- Dateien
- Prozesse (Anwendungen ~ , System ~ )
- Speicherbereiche (z.B. Mailboxes)
- periphere Geräte, Netzkarten u.ä.

➤ Beispiele für Objektbenennungen:

- Rechner: TKRN182, rzdspc1, ...
  - Ethernet-Karten: → weltweit eindeutig
  - E-Mail-Benutzer: → evtl. Alias
  - logische Verbindung: → dynam. Benennung
  - Dateien: Dateiname
  - Rechnerbenutzer: "User-ID"
- u.v.a.m.

## 6.4.2 Namensgebung

➤ Beispiel für Bedarf an Namen bei Kommunikation:

- Interprozesskommunikation, z.B.:

SEND (data = MSG1, destin = PROC\_XYZ)

- Zugriff auf Betriebsmittel, z.B.:

REQUEST (resource = NETPRINTER, printfile = ...)

GET\_FILE ( filename = NET1.NODE\_X.USER\_XY.PROJECT\_REPORT.DOC )

u.ä.



➤ Zur Eindeutigkeit von Namen:

- **benutzer**eindeutig
- **rechner**eindeutig (vgl. “lokale Eindeutigkeit“, s.u.)
- **netz**eindeutig
- **domain**eindeutig (evtl. bei mehreren gekoppelten Rechnernetzen)
- **weltweite**eindeutig



→ Namensvergabe beeinflusst von angestrebtem Gültigkeitsbereich,  
vgl. unterschiedliche Zuständigkeiten für Namensvergabe, wie z.B. ICANN,  
öffentlicher Netzbetreiber, ISP, Rechenzentrumsleiter, Einzelbenutzer

nota bene:            *Alias-Bildung* bei Namensvergabe möglich

➤ Bereitstellung netzweit eindeutiger Namen (**n-eindeutige Namen**) für kommunizierende Prozesse

→ bereits vorhanden: feste Namenskonventionen für Knoten,  
d.h. lokal eindeutige Namen (**l-eindeutige Namen**)  
für die Prozesse in ihren “lokalen” Knoten

ergo: n-eindeutige Namen erzeugbar durch Überlagerung einer netzweiten Namenskonvention über die Namenskonventionen der Knoten

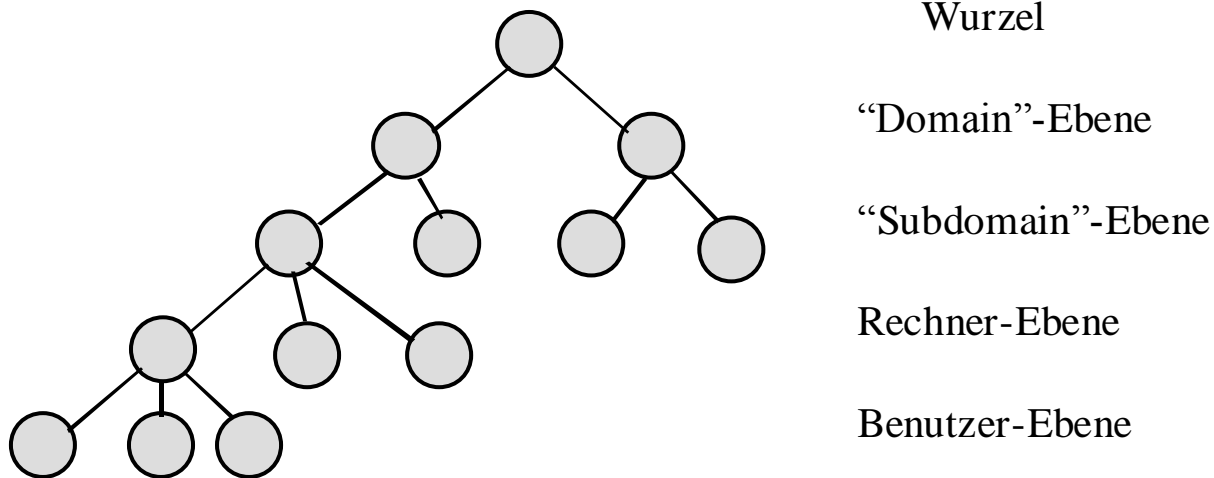
## Varianten der Namensgebung:

### I. **Hierarchische Verknüpfung** (“hierarchical concatenation“)

→ Ausgangspunkt:

hierarchische Zerlegung eines Netzes, z.B. in:

a) “Domain“, “Subdomain“, Rechner, Benutzer



Beispiel: [Müller@PC1.informatik.uni-xy.de](mailto:Müller@PC1.informatik.uni-xy.de)

siehe auch: “Internet Domain Naming“



b) Zentralvermittlung, Haupt ~, Knoten ~, Orts ~ und Fernsprechteilnehmer

Beispiel: 0049 . 4 . 101 . 1234 . . .

D HH PI . . .

➤ **Grobbeurteilung** der “**hierarchical concatenation**“-Methode:

- ✚ :

  - vereinfachte Namensvergabe (Ebenen-bezogen), aufwandsarm (vgl. Tel.-Nr.)
  - ggf. vereinfachte Abbildung “Name → Adresse“
- ✚ :

  - heterogene Namensgebung der Knoten problematisch
  - evtl. erschwerte Eingliederung von Knoten mit neuer Art lokaler Namensgebung

## II. **Dynamische Namenszuteilung** durch Knoten (“allocation“)



→ Ausgangspunkt:

- Sei  $N$  die Menge der zu vergebenden (z.B. netzweit eindeutigen) Namen.
- Sei  $P = \{P_1, \dots, P_k\}$  die Menge der für die Namensverwaltung zuständigen Prozesse;

für  $P$  gilt: jeder Prozess  $P \in P$  erhält a priori einen n-eind. Namen  $n_p \in N$  (statische Zuordnung) und in jedem Knoten existiert mindestens ein  $P \in P$ .

Bezeichne  $N_p \subset N$  die an  $P$  fest vergebenen Namen.

➤ Beschreibung der dynamischen Verwaltung der Namensmenge  $N \setminus N_p$ :

$N \setminus N_p$  wird auf sämtliche Knoten fest aufgeteilt (→ **Partitionierung**)

- **Annahme:** Prozess  $Q_1$  in Knoten  $i$  wünscht Kommunikation mit Prozess  $Q_2$  in Knoten  $j$  (mit l-eind. Namen  $l_j$ )
- **Aktionsfolge:**
  1.  $Q_1$  adressiert Namensverwaltungsprozess  $P \in P$  in Knoten  $j$  (Mitteilung: Wunsch nach Kommunikation mit Prozess  $Q_2$  namens  $l_j$ )
  2.  $P$  ermittelt bislang noch unbelegten n-eind. Namen  $n_q \in N \setminus N_p$  und nimmt **dynamische Zuordnung** von  $n_q$  zu  $l_j$  vor
  3.  $P$  teilt Name  $n_q$  an  $Q_1$  mit
  4.  $Q_1$  kommuniziert mit  $Q_2$  (Adressierung von  $Q_2$  durch  $Q_1$  unter Verwendung von  $n_q$ )
  5. nach Ende der Kommunikation zwischen  $Q_1$  und  $Q_2$ :  
Auflösung der dynamischen Zuordnung von  $n_q$  zu  $l_j$   
(Auflösung durch  $P$ )

➤ **Grobbeurteilung** der “allocation“-Methode:

- Namensverwaltung stark zentralisiert
- Menge der pro Knoten dynam. verwalteten n-eind. Namen relativ klein
- starre Zuordnung von Prozessen zu Knoten notwendig
- Initiator einer Kommunikation benötigt umfangreiche Information bzgl. Kommunikationspartner
- komplexe Implementierung
- Auflösung einer Zuordnung “n-eind. Name  $\leftrightarrow$  Objekt  $x$ “ sämtlichen Kommunikationspartnern von  $x$  mitzuteilen

### III. **Statische Namenszuteilung** (n-eind. Namen) an kommunizierende Objekte



⇒ Abb. netzeindeutiger auf lokal eindeutige Namen (“**mapping**“)

- Netz verwaltet a priori definierte Menge  $N$  von n-eind. Namen durch:

Statische Zuordnung von  $N$  zu der Menge derjenigen Prozesse (bzw. kommunizierenden Objekte), auf die von irgendeinem Knoten aus zugegriffen werden kann.

- Bei Kommunikation zwischen Prozessen  $Q_1$  (auf Knoten  $i$ ) und  $Q_2$  (auf Knoten  $j$ )  
→  $Q_1$  kennt n-eind. Namen von  $Q_2$ ;  
in Knoten  $j$  erfolgt Abb. (mapping) von n-eind. Namen von  $Q_2$  auf l-eind. Namen von  $Q_2$ .

*nota bene:* Analogie Mobilfunknetze (  $N$  = Gesamtmenge der Tel.-Nr. )

➤ Einige **Vorteile** der “**mapping**“ Methode:

- lokale Namen durch jeden Knoten beliebig wählbar
- Verlagerung eines Objektes auf anderen Knoten bzw. Migration eines angesprochenen Netzbenutzers in andere Funkzelle möglich  
→ Neuordnung von Betriebsmitteln sowie Mobilität von Benutzern unterstützt
- Verwendung desselben 1-eindeutigen Namens durch verschiedene Knoten unproblematisch, da 1-eind. Namen knotenextern nicht sichtbar.

## 6.4.3 Adressierung

### ➤ **Einflussgrößen** für Adressierung

- Art des betroffenen Dienstes (1:1, 1:n, n:m)
- Topologie
- geographische Verteilung der
  - Systemkomponenten
  - Benutzer

### ➤ **Generelle Möglichkeit der Adressierung** von Objekten:

- direkte Verwendung des Objekt-Namens
- Adressierung nach Inhalt/Wert eines Objektes
- Angabe des Absenders (nur 1 Empfänger existiert)
- Name einer adressierten Gruppe
- Angabe des Weges zum angesprochenen Objekt
- Angabe einer Beziehung des angesprochenen Objektes zum Absender (z.B.  $\text{adr}(\text{destination}) := \text{own\_adr} + 1$ )



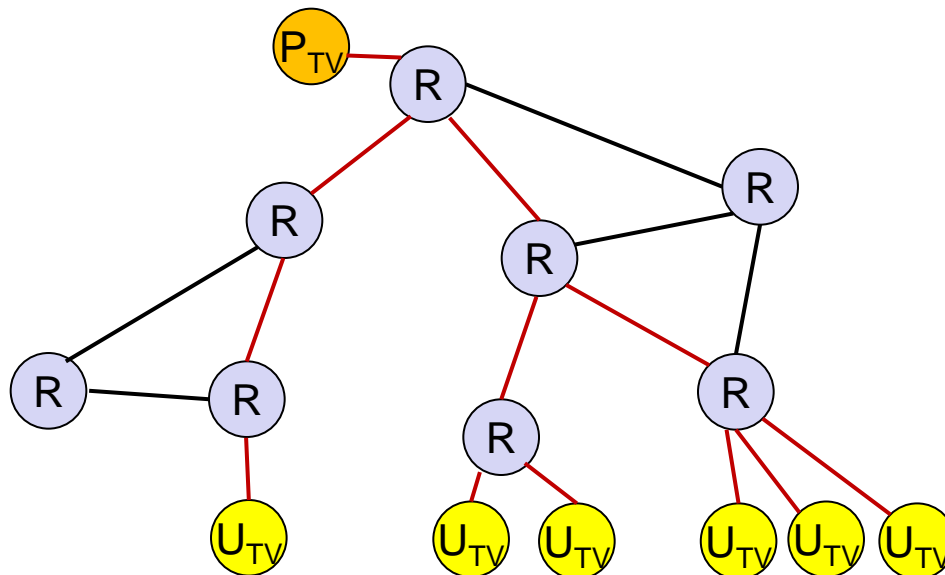
➤ Beispiele für **multiple Adressierung** ( $\geq 2$  Objekte):

- explizite Angabe eines Mehrfachziels (z.B. Auflistung)
- Gruppenangabe (z.B. Klassen von Objekten adressiert)
- gleichzeitige Adressierung sämtlicher möglicher Kommunikationspartner (z.B. “broadcast“)



nota bene: - “**multicast**“ → an mehrere (z.B. “Multicast“-Gruppe)  
 - “**broadcast**“ → an alle (z.B. innerhalb eines Subnetzes)

**Beispiel für Multicast bei IPTV:**



*Notation:*

$P_{TV}$ : Anbieter (“Provider“) eines gegebenen TV-Programms bzw. TV-Kanals

$U_{TV}$ : Benutzer, der genau das gegebene TV-Programm betrachten möchte

R: Vermittlungsrechner / Router

“Multicast-Tree“ / “Multicast-Baum“:

durch die dunkelroten Leitungen aufgespannter Baum (kann sich dynam. verändern bei Änderung der “Multicast“-Gruppe, d.h. der Menge der  $U_{TV}$ ’s)

*Bem:* Jedes angebotene Programm (TV-Kanal) führt zu einer eigenen “Multicast“-Gruppe

## Von dem Namen zur Adresse

### ➤ Beispiele:

- “Gelbe Seiten“ (“Yellow Pages“)
- “Name Server“ in (z.B. lokalen) Rechnernetzen

### ➤ Def. **Namensverwalter** (name server) :

Geg. Namensräume  $R_1, R_2$  (Menge von Namen).

Ein **Namensverwalter** ist eine Instanz in einem verteilten System oder Rechnernetz, die (ggf. gemeinsam mit anderen Namensverwaltern) eine Abb. von  $R_1 \rightarrow R_2$  realisiert.  $\square$



⇒ Namensverwalter erbringt sog. **Namensverwaltungsdienst** (name service).

- Bsp. 1.:  $R_1 = N = \text{Menge n-eind. Namen (z.B. zur Benennung von Objekten)}$   
 $R_2 = \{\text{adr (obj)} \mid \text{obj ist Objekt eines verteilten Systems}\}$
- Bsp. 2.:  $R_1 = \langle \text{Name, Wohnort, Straße, Haus-Nr.} \rangle$  für Personenmenge  
 $R_2 = \text{Menge von Tel.-Nr.}$

➤ **Realisierungsvarianten** für Namensverwaltungsdienste:

- Anfragen an “Name Server“, “Location Manager“ o.ä., z.B.
  - Wo ist gegenwärtiger Aufenthaltsort von Mobilfunknetzbenutzer (mit folgendem n-eind. Namen ...) ?
  - DNS-Anfragen bei DNS-Servern (Domain Name System)
  - Zuordnung IP-Adresse ↔ ATM-Adresse, vgl. z.B. ATMARP-Server (ARP für Address Resolution Protocol)
  - an welchem Rechner befindet sich Benutzer mit n-eind. Namen XYZ ? ... oder nur: in welchem Netz befindet sich Benutzer XYZ ?  
(⇒ endgültige Zuordnung: “Name → Adr.“ erst dort)
- Namensverzeichnis: **zentral** oder **verteilt**  
... und sofern verteilt → **repliziert** oder **partitioniert**

➤ Anfragen an “Name Server“:

- gezielt
- im “Broadcast“-Verfahren
- bei genau einem bevorzugten Server (z.B. “home directory“ oder dem für das eigene Netz zuständigen Server) → evtl. automatische Unteranfragen, sofern Info nicht direkt verfügbar

## Von der Adresse zu der Route / dem Pfad

⇒ zuständig für Abbildung “Adresse → Pfad” :  
die **Routing-Komponente(n)**

Lösung in der Regel:

Adresse oder nur Teil davon wird herangezogen,  
um sich möglichst

*schnell, kostengünstig, sicher, o.ä.*

der adressierten Netzkomponente zu nähern.

→ **Varianten** der **Routenzuordnung** zu gegebener **Adresse**:

- **statische Routenzuordnung**:  
evtl. bereits durch Sender erledigt (vgl. z.B. “*Source Routing*“)  
•  
•  
•
- **flexible, dynamische Routenfindung**:  
evtl. verteilte Entscheidungsfindung d.h. Abb. “Adresse → Pfad” wird erst  
auf dem Weg zu der adressierten Komponente sukzessive vervollständigt  
(ggf. abhängig von Netzzustand)

*nota bene:* zu Details vgl. Wegeermittlungsalgorithmen in Abschn. 6.3

# 6.5 Interkonnektion von Netzen

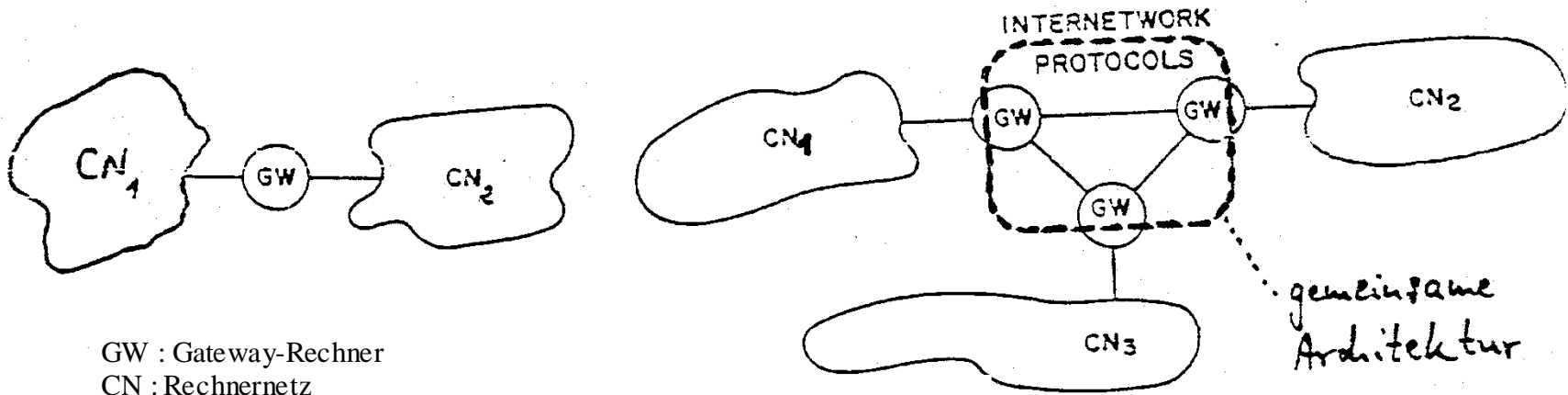
## **Gründe für Rechnernetzkopplungen:**

- netzübergreifende Kommunikation
- Zugriff auf Dienste weiterer Netze

## **Probleme :**

- Heterogenität von Netzarchitekturen
- Netz- / Datensicherheit
- Effizienzeinbuße
- Accounting
- Übergreifendes Netzmanagement schwierig
- Fehlersuche höchst aufwändig
- Evtl. unklare Zuständigkeiten / Verantwortlichkeiten

## “Direkte” Kopplung oder “Indirekte” Kopplung ?



### Aufgaben von Gateway-Rechnern :

→ Konvertierung elementarer Protokollfunktionen, z.B.

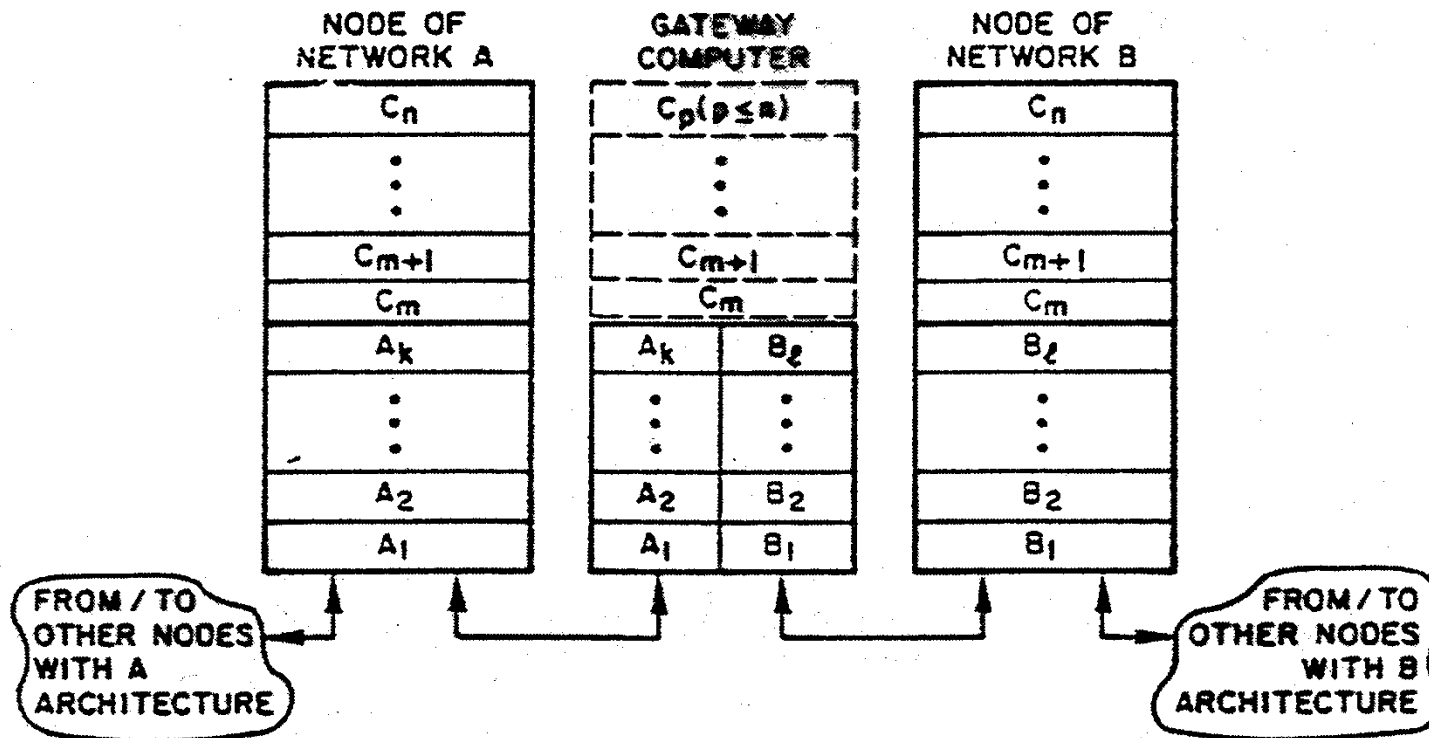
- Adressierung
- Routing (zwischen GWs, RNen)
- Fehlerkontrolle/-anzeige
- Fragmentierung/ Reassemblierung (von DEen)
- Flusskontrolle (zwischen RNen)
- Accounting
- Datenschutz, z.B. erhöhte Netzsicherheit durch GW als “Firewall“



## Verfeinerte Betrachtung der **direkten Kopplung**



Generisches Architekturmodell :



Sonderfälle :

(a) *keine C-Schichten*

falls  $k = l = 7 \rightarrow$  Application Level Gateway.

(b) *C-Schichten exist.  $\wedge A_k / B_l \geq$  Vermittlungsschicht :*

(b<sub>1</sub>)  $k = l = 4 \rightarrow$  **Kopplung auf Transportebene;**  
**Dienste oberhalb TS kompatibel**

(b<sub>2</sub>)  $k = l = 3 \rightarrow$  Kopplung auf Vermittlungsebene;  
vgl. Internetworking Sublayer von L3 (bei ISO, ECMA).

(c) *C-Schichten exist.  $\wedge A_k / B_l \leq$  Leitungsschicht :*

(C1)  $k = l = 2 \rightarrow$  GW  $\hat{=}$  BRIDGE (Zwischenspeicherungsfunktion)

(C2)  $k = l = 1 \rightarrow$  GW  $\hat{=}$  REPEATER

(C3)  $k = l = 0 \rightarrow$  gemeinsame Architektur (C).

Varianten für  $C_m, \dots, C_p$  :

I.  $n = p$  : GW ist Host

II.  $m = p = 3$  : gemeins. Vermittlungsschicht; keine Anwendungen in GW

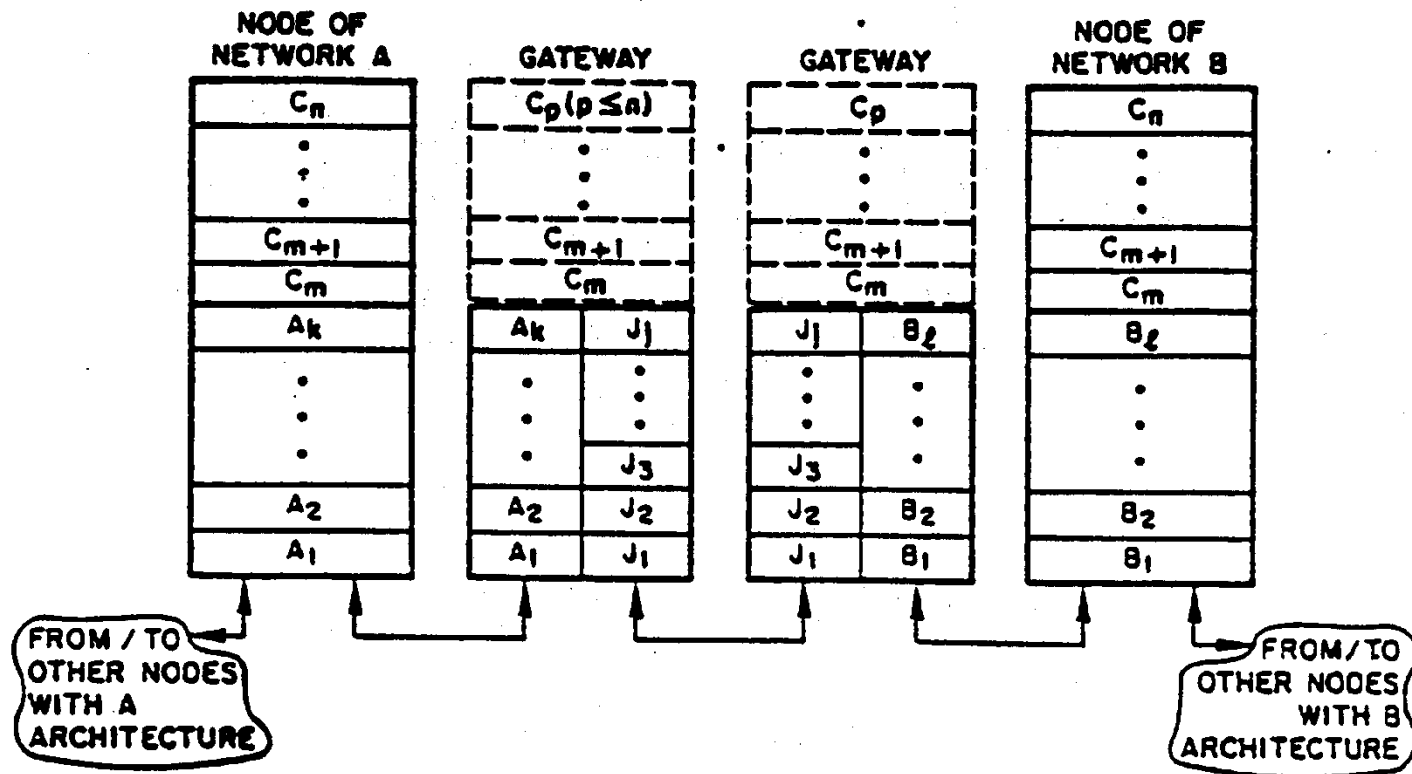
III.  $p = 0$  : GW mit 2 „Gesichtern“; GW nur für Kopplung



## Verfeinerte Betrachtung der **indirekten Kopplung**



Generisches Architekturmodell :



## Beispiele für Zwischenarchitektur (J) :

- TCP / IP- basierte Protokollhierarchie (heutzutage *die* Standardlösung für dieses Problem)
- postalisches Vermittlungsnetz, z.B.  $j = 3 \wedge J_1, \dots, J_3 \equiv \text{X.25}$
- Standard-Architekturmodell (OSI), z.B.  $j = 7 \wedge \text{ISO-Standardis.}$
- SNA, z.B. DECnet-SNA-Gateways (in frühen Netzen)
- 
- 
- 
- etc.

Aufwand für Interkonnektion von  $n$  Architekturen :

- direkte Kopplung  $\longrightarrow n(n-1)/2$  Abbildungen
- indirekte Kopplung  $\longrightarrow n$  Abb.

## Dienst- versus Protokollkopplung

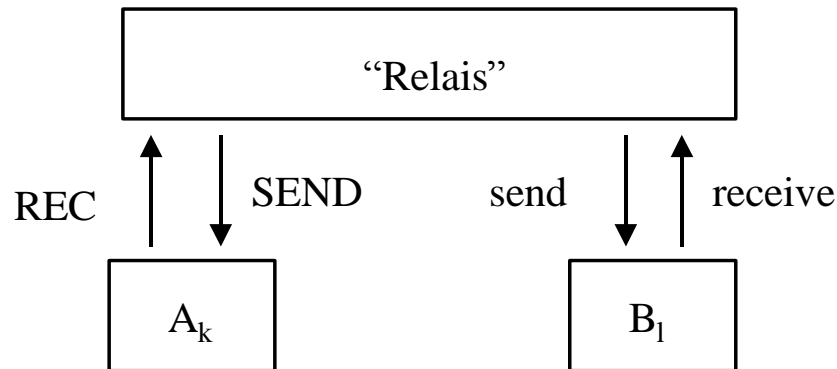


### a) **Dienstkopplung:**

→ Kopplung auf Dienstebene

(insbesondere bei Existenz weitgehend identischer Dienste in beiden zu kopplenden Netzen)

- Bsp.:
  - Dateitransfers in Netz A und B (bei Kopplung von A und B)
  - verbindungsorientierte Paketvermittlung sowohl in Netz A als auch in B
- *Abläufe in Gateway:*



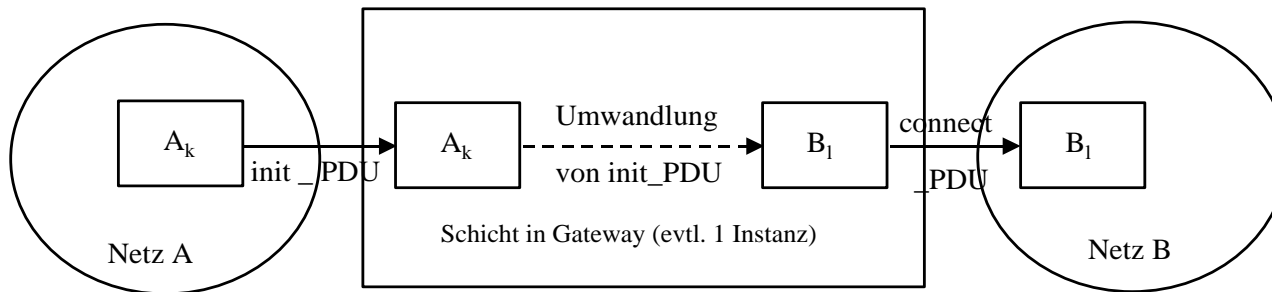
**nota bene:** Dienstprimitive *“receive”* impliziert *“SEND”* und *“REC”*  
 impliziert anschließendes *“send”*  
 (z.B. für Empfangen und Senden einer Datei)



## b) **Protokollkopplung:**

→ Kopplung auf Protokollebene  
 (insbesondere bei Existenz weitgehend identischer Protokolle  
 in beiden zu koppelnden Netzen, d.h. ähnliche Protokollkontrollinfo PCI  
 wird ausgetauscht)

### • *Abläufe in Gateway:*



*nota bene:*

**Erhalt der Dateneinheit "*init\_PDU*" impliziert Absenden von "*connect\_PDU*" jeweils für Verbindungsaufbau mit einander entsprechender PCI**

# 6.6 Schmalband-ISDN

**ISDN** = Integrated Services Digital Network

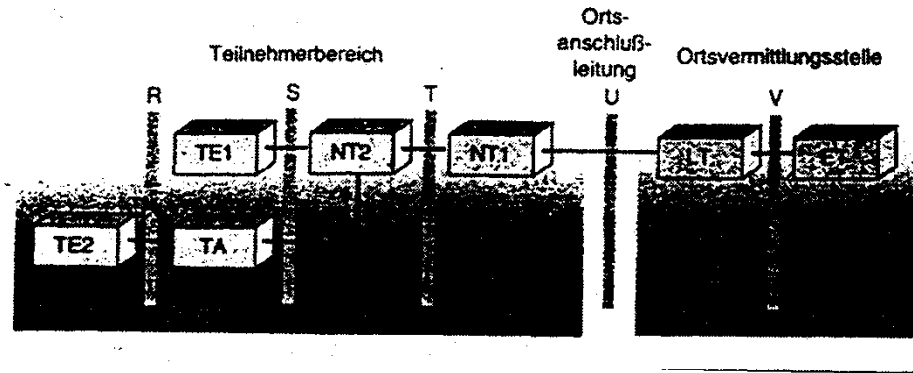
**Tabelle 2.1. Bestehende und mögliche Dienste im ISDN mit Bitraten bis zu 64 kbit/s ( $n \times 64$  kbit/s)**

Dienstklassen	64-kbit/s-ISDN-Dienste über B-Kanäle (64-kbit/s)	64-kbit/s-ISDN-Dienste über den D-Kanal	Bestehende Dienste aus dem Fernsprechnetz	Dienste für Endgeräte des Integrierten Text- und Datennetzes (IDN)
<b>Dialogdienste</b>	<b>Übermittlungsdienste:</b> Leitungsvermittelte Übermittlungsdienste • 64 kbit/s trans • 3,1 kHz Audio • Sprache • 7 kHz Audio Paketvermittelte Übermittlungsdienste Frame Relay-Übermittlungsdienste  <b>Teledienste:</b> ISDN-Fernsprechen, -Fernsprechkonferenz • Telefonie 3,1 kHz • Telefonie 7 kHz ISDN-Teletex ISDN-Telefax (Gruppe 4) ISDN-Fernskizzieren (Telewriting) ISDN-Festbildübermittlung ISDN-Bildtelefon, -konferenz <sup>1)</sup> ISDN-Sicherheitsdienste ISDN-Fernwirkdienste	ISDN-Datenübermittlung (paketvermittelt) ISDN-Sicherheitsdienste ISDN-Fernwirkdienste	Fernsprechen Telefax (Gruppe 4/3) Datenübertragung mit V.-Schnittstellen (parallel, seriell) Sicherheitsdienste Fernwirkdienste	Für Datenübertragungsgeräte mit Schnittstellen X.21 und X.25
<b>Speicherdienste</b>	Voice Mail Text Mail z.B. Telebox-400-IPM Fax Mail			
<b>Abrufdienste</b>	ISDN-Bildschirmtext			
<b>Verteilungsdienste</b>	Datenverteilung Sprachverteilung Festbildverteilung		Bildschirmtext	

<sup>1)</sup> Mit niedrigerer Bildauflösung im Vergleich zum bestehenden Fernsehen

(aus: [Boc 97])

# ISDN : Schnittstellen und Protokolle



Bezugspunkt	Schnittstelle für Basisanschluss	Schnittstelle für Multiplexanschluss
R	z. B. V.24, X.21 (vgl. Abschn. 2.11)	—
S	$S_0$	—
T	$S_0$	$S_2$
U	$U_{K0}$	$U_2 (=PCM 30)$
V	—	$V_2$

Mögliche Schnittstellen am zentralen Bezugspunkt S :

(1)  **$S_0$ -Schnittstelle** (Basisanschluss für ISDN-Teilnehmer) mit Kanalstruktur :

$B+B+D_{16} + YY$  wobei

$B \equiv$  **B-Kanal** à 64 Kbit/s (Basiskanal)

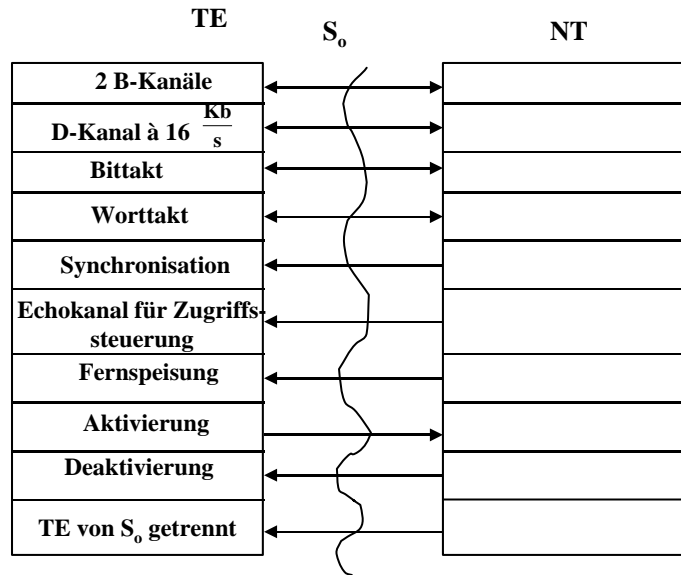
$D_{16} \equiv$  **D-Kanal** à 16 Kbit/s (Steuerkanal)

$YY \equiv$  **Kanal für Synchronis., Überwachung etc.** (48 Kbit/s)

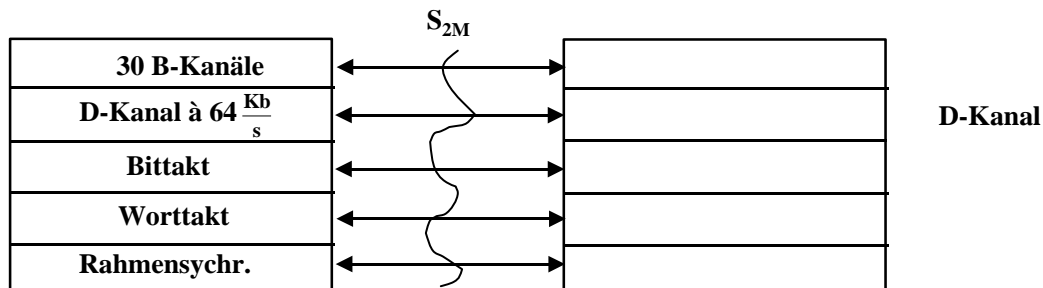
→ Nettobitrate : **144 Kbit/s** (Bruttobitrate: **192 Kbit/s**)



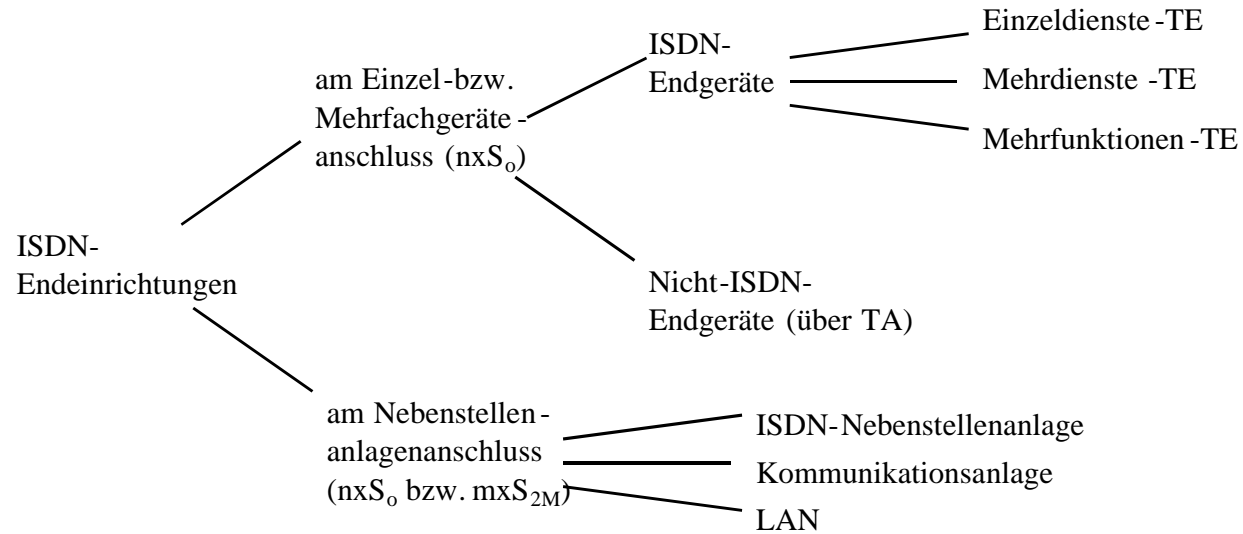
## Funktion der $S_o$ -Schnittstelle :



- (2)  **$S_{2M}$  – Schnittstelle** (z.B. Primär-Multiplexanschluss für ISDN-Nebenstellenanlage) mit Kanalstruktur:  $30 \times B + D_{64} + XX$  wobei
- $D_{64} \equiv$  **D-Kanal** à 64 Kbit/s
- $XX \equiv$  **Kanal für Synchronis , Überwachung** etc. (64 Kbit/s)
- Nettobitrate : **1984 Kbit/s** (Bruttobitrate: **2 Mbit/s**)



## ***Klassifikation der ISDN-Endeinrichtungen :***



## *Topologien für Anschluss von TEs an NT :*

Punkt-zu-Punkt, Stern (NT  $\cong$  Zentrum), Bus, Ring



## ➤ Kommunikation im ISDN :

- **Verbindungsaufbau**

- über D-Kanal (!)
- Signalisierung im Netz (→ spez. Signalisierungsprotokolle, “out-of-band“-Signalling)
- Durchschalten einer Verbindung (→ Leitungsvermittlungsprinzip)

- **Datenaustausch**

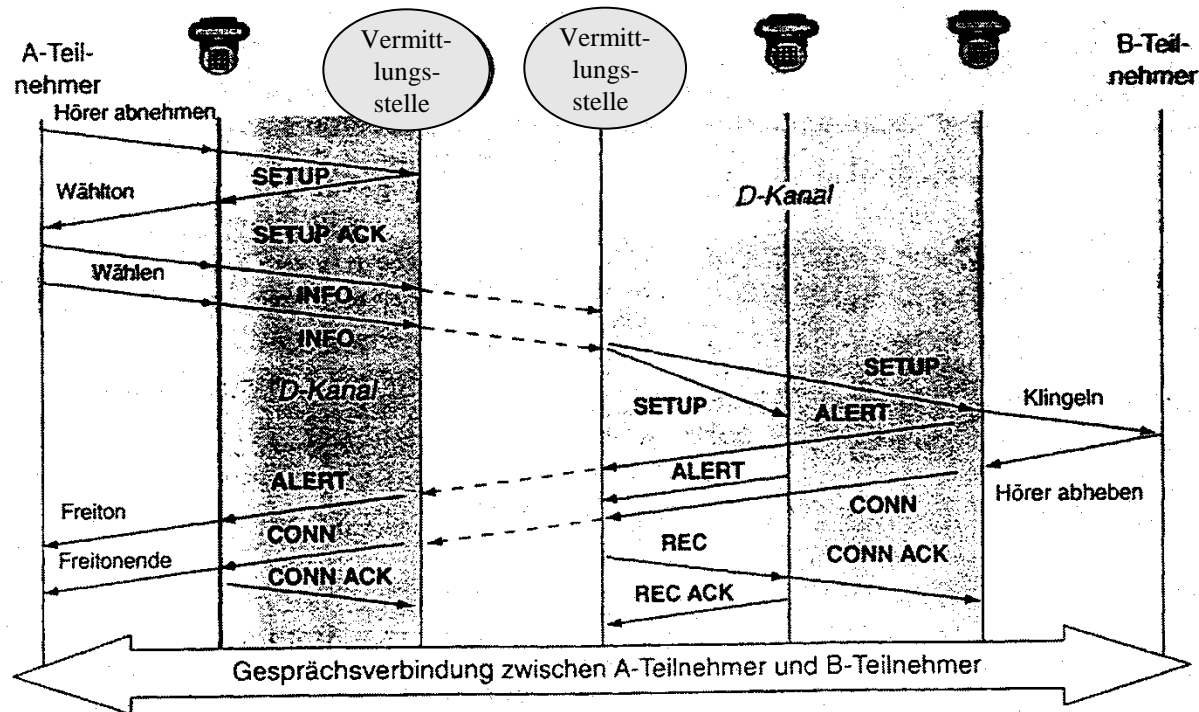
- über B-Kanal (transparent)
- über D-Kanal (evtl. Berücks. von spez. Protokoll)
- mögl. Dienstschtaltung

- **Verbindungsauflösung**

z.B. auf Initiative von TE oder netzseitig



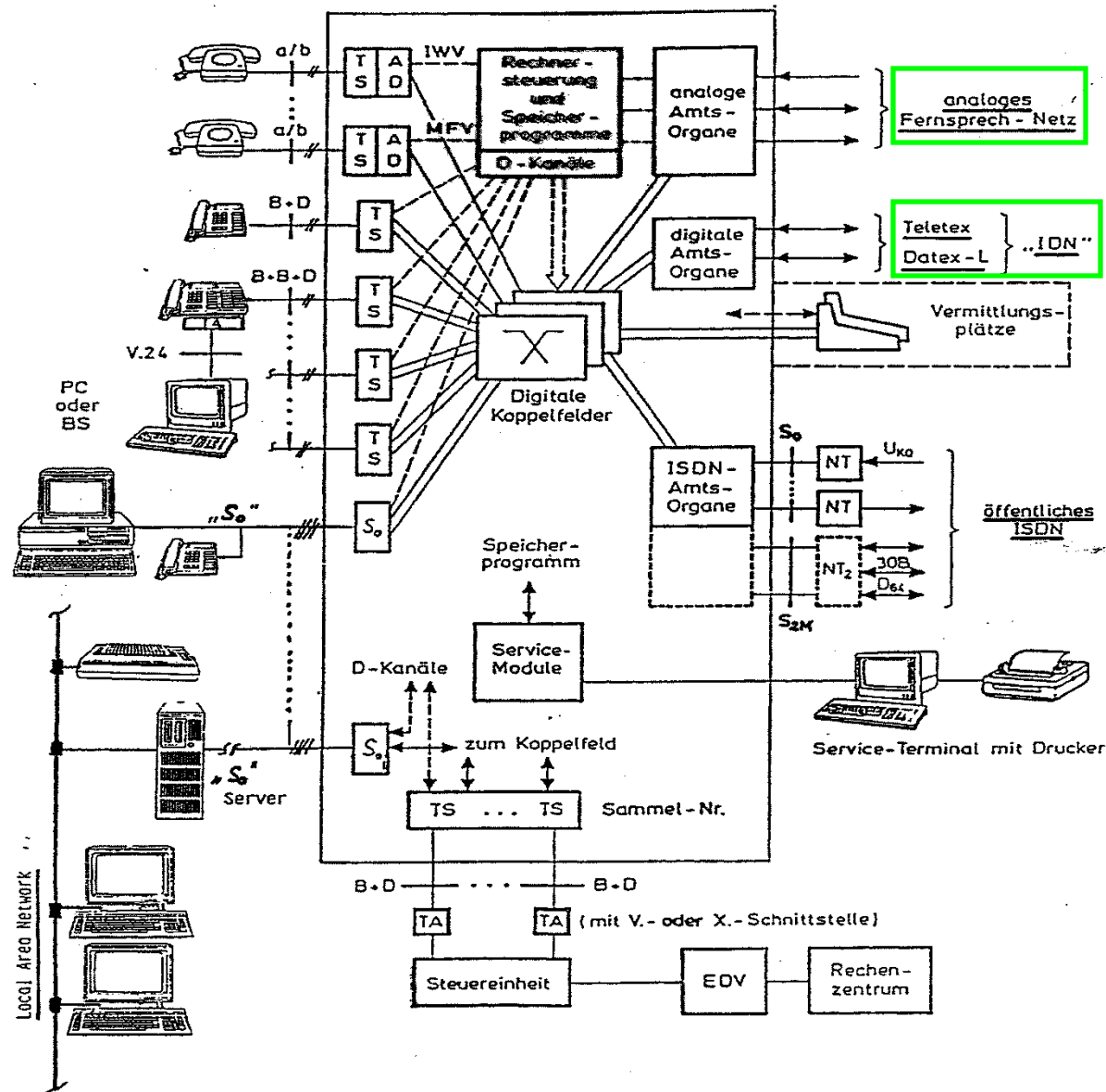
➤ **Protokollablauf für den Aufbau einer leitungsvermittelten Verbindung im S-ISDN :**



➤ **Interkonnektion von ISDN mit existierenden Netzen :**

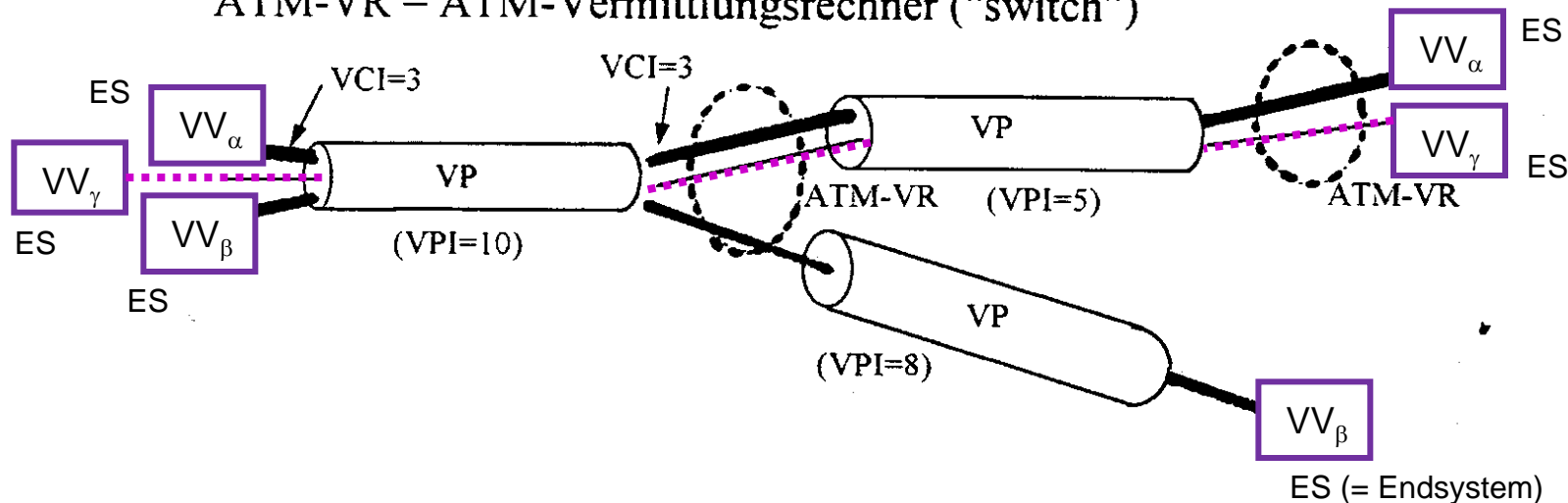
- relativ einfach mit leitungsvermittelten Netzen (z.B. DATEX-L, Fernsprechnet)
- aufwendig mit paketvermittelten Netzen (z.B. DATEX-P, IP-Netzen auf Basis von Datagrammtechnik), vgl. Abschn. 6.5

# ISDN-fähige, digitale Telekommunikations-Anlage:



## 6.7 ATM-Protokolle und -Netze

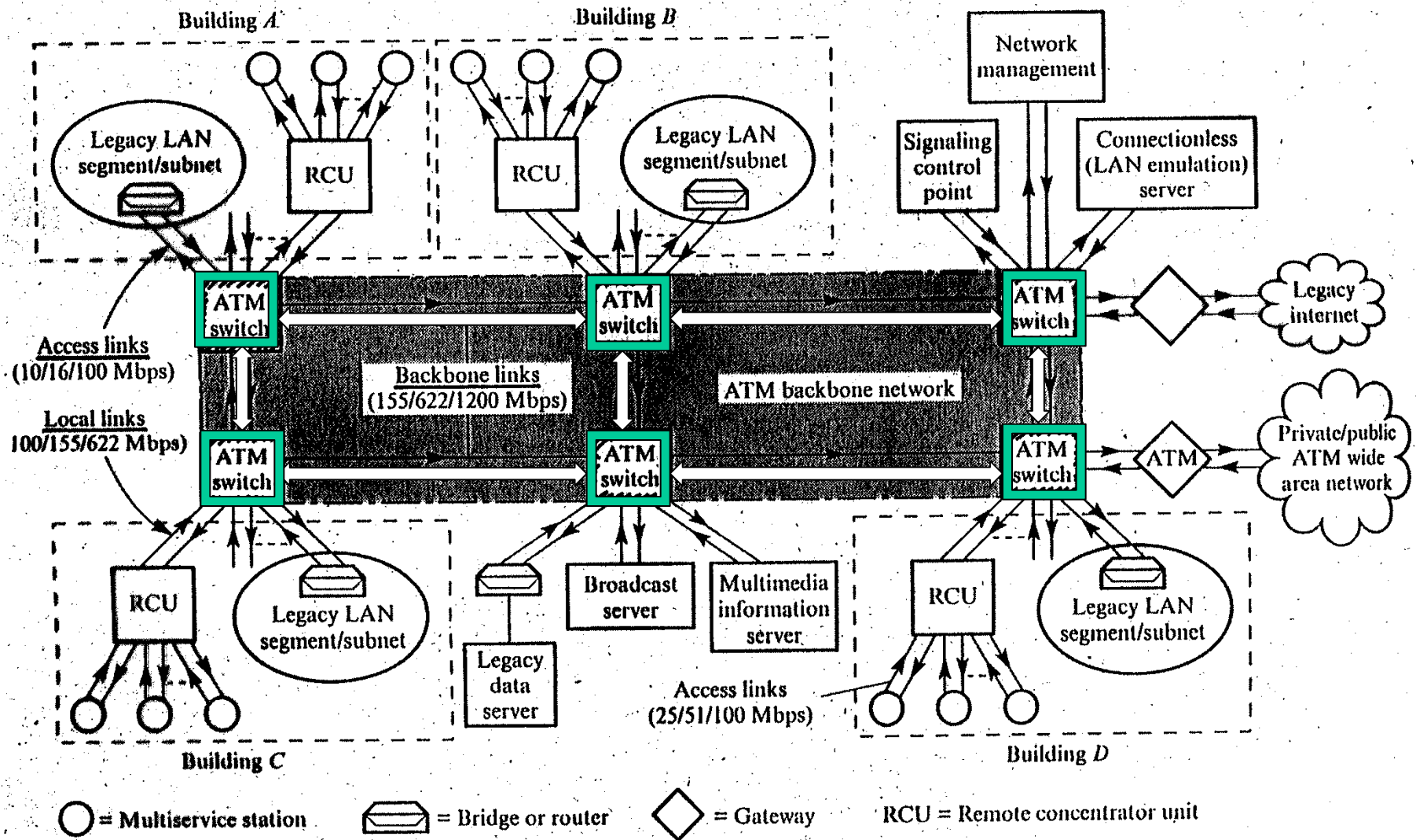
- ATM-Techn.: durch CCITT/ITU standardisierte Lösungsvariante für B-ISDN (zu Vorteilen der Dienstintegration: vgl. S-ISDN)  
nota bene: weitere Richtlinien für ATM-Systeme durch ATM-Forum (nicht-kommerz., > 100 Mitglieder weltweit)
- Vermittlungstechnik: Zellenvermittlung vgl. Abschn. 6.2  
→ realisiert schnelle Paketvermittlung
- Austausch von ATM-Zellen (cells) über virtuelle Verbindungen (ungeschichtet, vgl. Abschn. 6.2 )  
→ Identifikation einer virtuellen Verbindung VV durch Tupel (VCI, VPI);  
VCI = virtual channel identifier; VPI = virtual path identifier  
ATM-VR = ATM-Vermittlungsrechner ("switch")



## *ATM-Protokolle und -Netze*, allg. Eigenschaften (Forts.):

- bei Verbindungsaufbau:  
 Spezifikation von Dienstqualitätsanforderungen (Quality-of-Service, QoS) seitens Netzbenutzer  
 → Netz realisiert QoS-Garantien
  - deterministisch
  - stochastisch
- Bsp. für QoS-Garantien:
  - konstanter Durchsatz  $d_0$  übertragen (constant bit rate, vgl. Leitungsvermittlung)
  - Zellenverlustrate  $\leq \varepsilon_0$  → in welchen Intervallen überprüft?
  - Verbindungsaufbauzeit  $\leq \tau_0$  (evtl. nur mit Wahrscheinlichkeit  $p < 1$ ) etc.
- ⇒ QoS-Garantien erfordern:
  - Betriebsmittelreservierung (z.B. Leitungs-, Vermittlungskapazitäten)
  - Kontrolle des Benutzerverhaltens (z.B. ratenbasierte Flußkontrolle für zu übertragende Info)

# ATM-LAN: Beispielkonfiguration



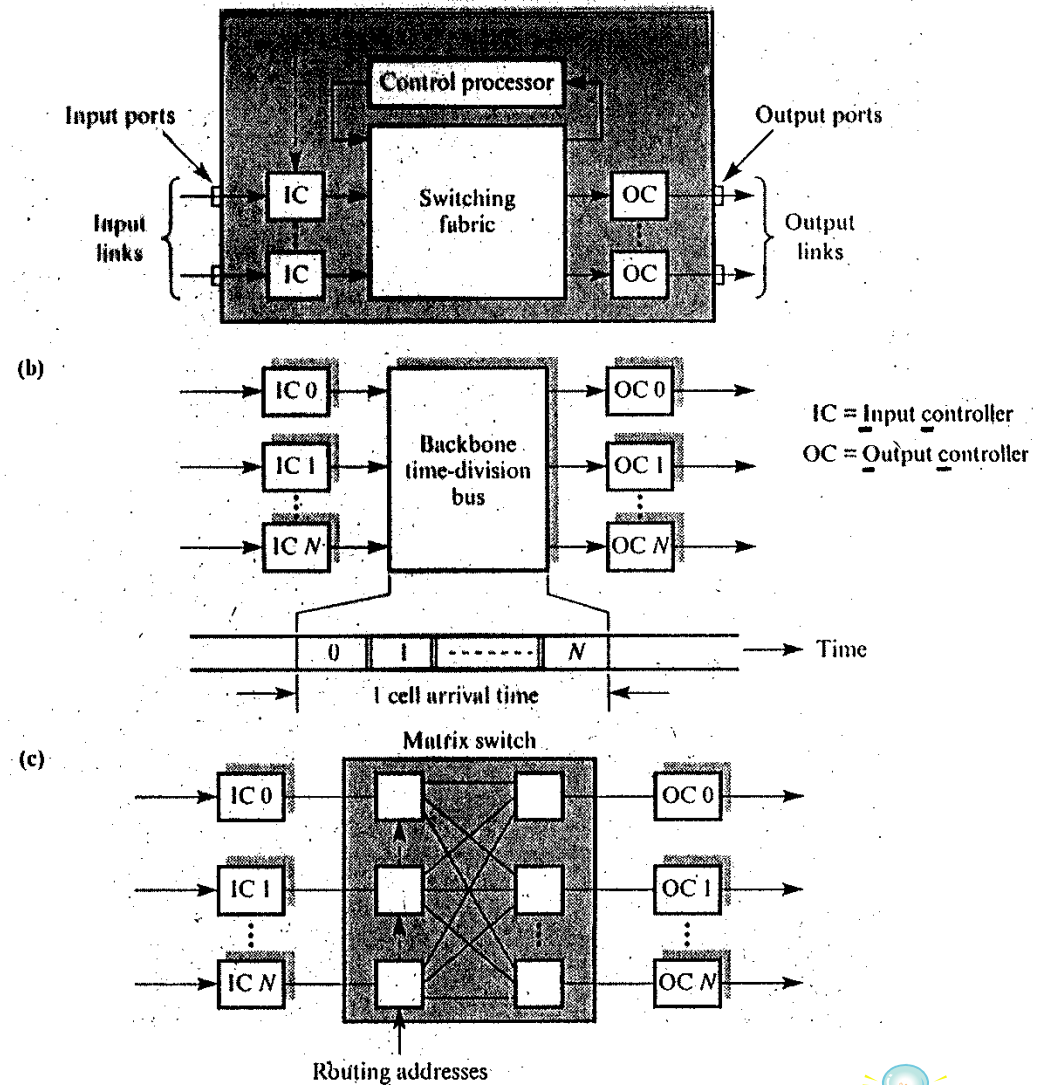


### ATM Switch Architecture

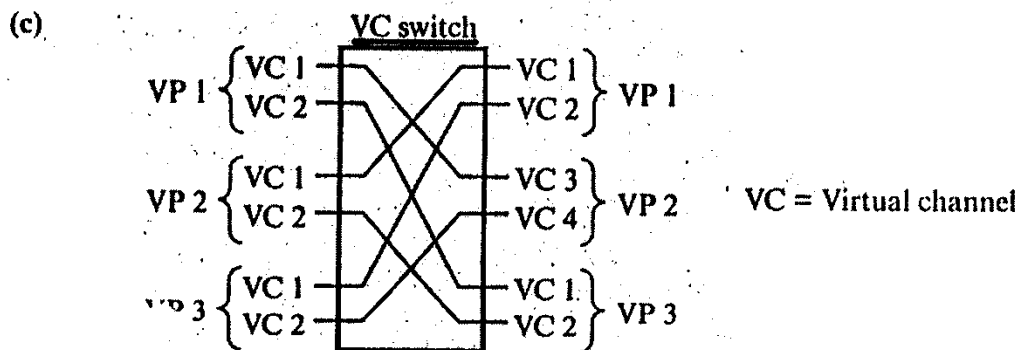
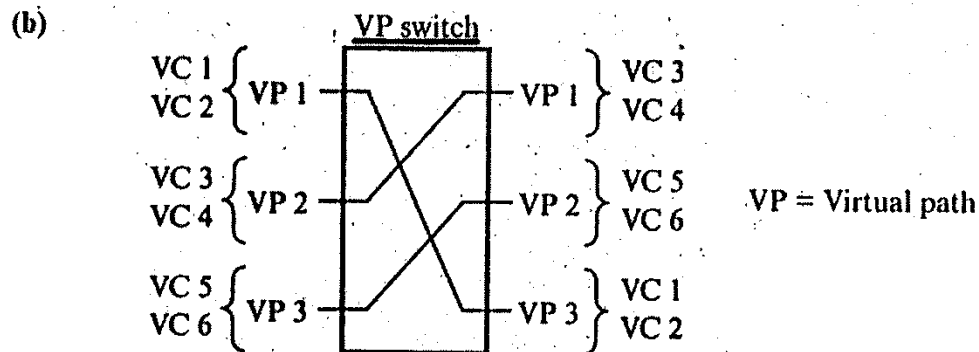
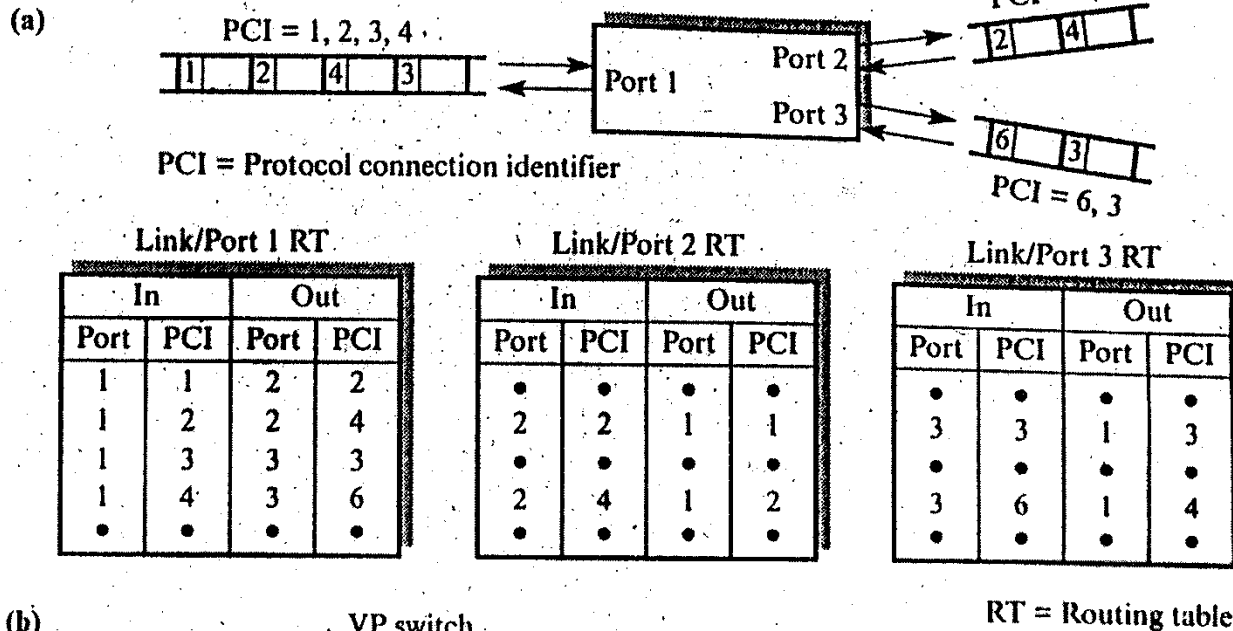
→ vgl. auch Abschn. 3.3

(a) general structure;  
(b) time-division bus  
schematic; (c) fully  
connected matrix  
switch.

(aus: [Hal 96])







## Cell switching principles

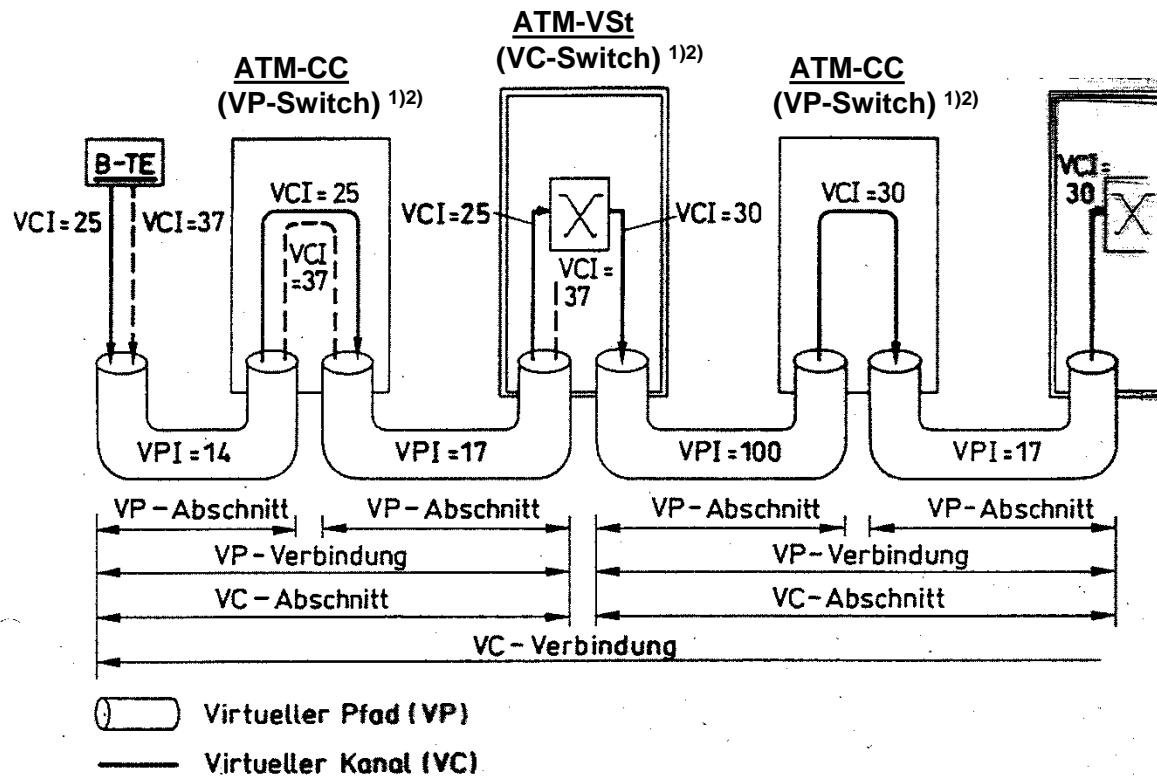
(a) Routing schematic

(b) VP routing

(c) VC routing

(aus: [Hal 96])





**Bild 6.9. ATM-Crossconnect und ATM-Vermittlungsstelle**

- VCI virtuelle Kanalkennung (virtual channel identifier)
- VPI virtuelle Pfadkennung (virtual path identifier)
- VC virtueller Kanal (Virtual Channel)
- VP virtueller Pfad (Virtual Path)
- CC Crossconnect
- VSt Vermittlungsstelle
- B-TE Breitband-Endeinrichtung
- 1) Bestandteil eines Teilnehmeranschlußnetzes
  - 2) Virtuelles Verbindungsleitungsnetz als flexible Netzinfrastruktur
  - 3) Steuerung per TMN (Telecommunication Management Network)
  - 4) Steuerung per Signalisierung

(aus: [Boc 97])

**Virtuelle Kanäle  
und virtuelle Pfade  
in ATM-Netzen**

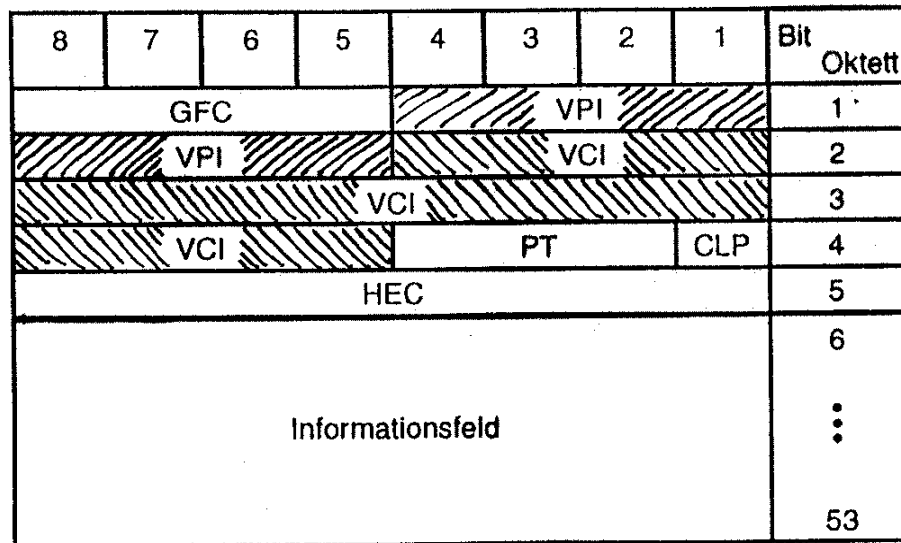


## Zellenaufbau

### Zellkopf

In Bild 4.17 ist auch der detaillierte Aufbau des Zellkopfes an der Benutzer-Netz-Schnittstelle (UNI) dargestellt. Der Zellkopf besteht aus folgenden Feldern:

- Generic Flow Control (GFC)
- Virtual Path Identifier (VPI)
- Virtual Channel Identifier (VCI)
- Payload Type (PT)
- Cell Loss Priority (CLP) → 1 = niedrige Prio.; 0 = hohe Prio.
- Header Error Control (HEC) → CRC,  $P(X) = X^8 + X^2 + X$



Header (5 Byte)

"Payload" (48 Byte)



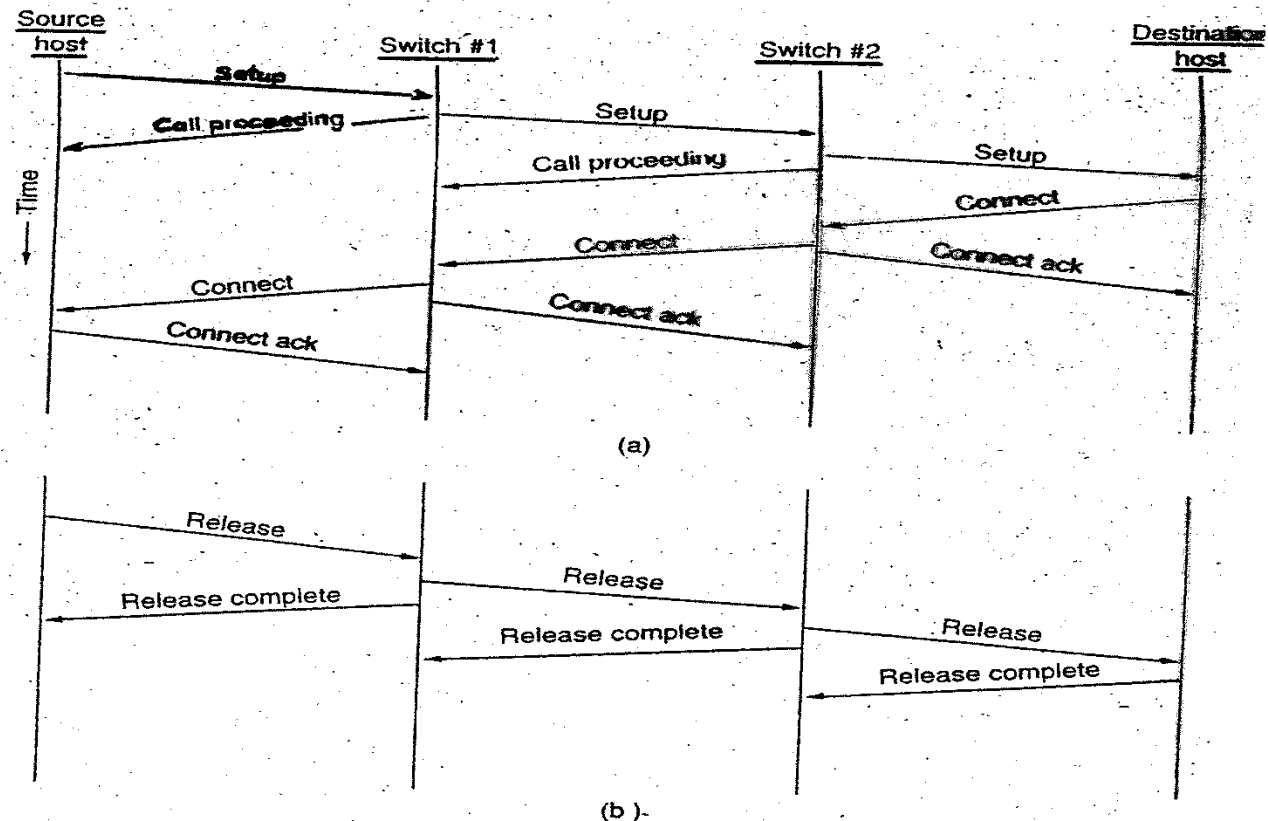
*Nota bene:* 48 Byte als **Kompromiss** aus  
64 Byte (Wunsch USA)  
und 32 Byte (Wunsch Europa)

→ das ist Standardisierung ! 😊

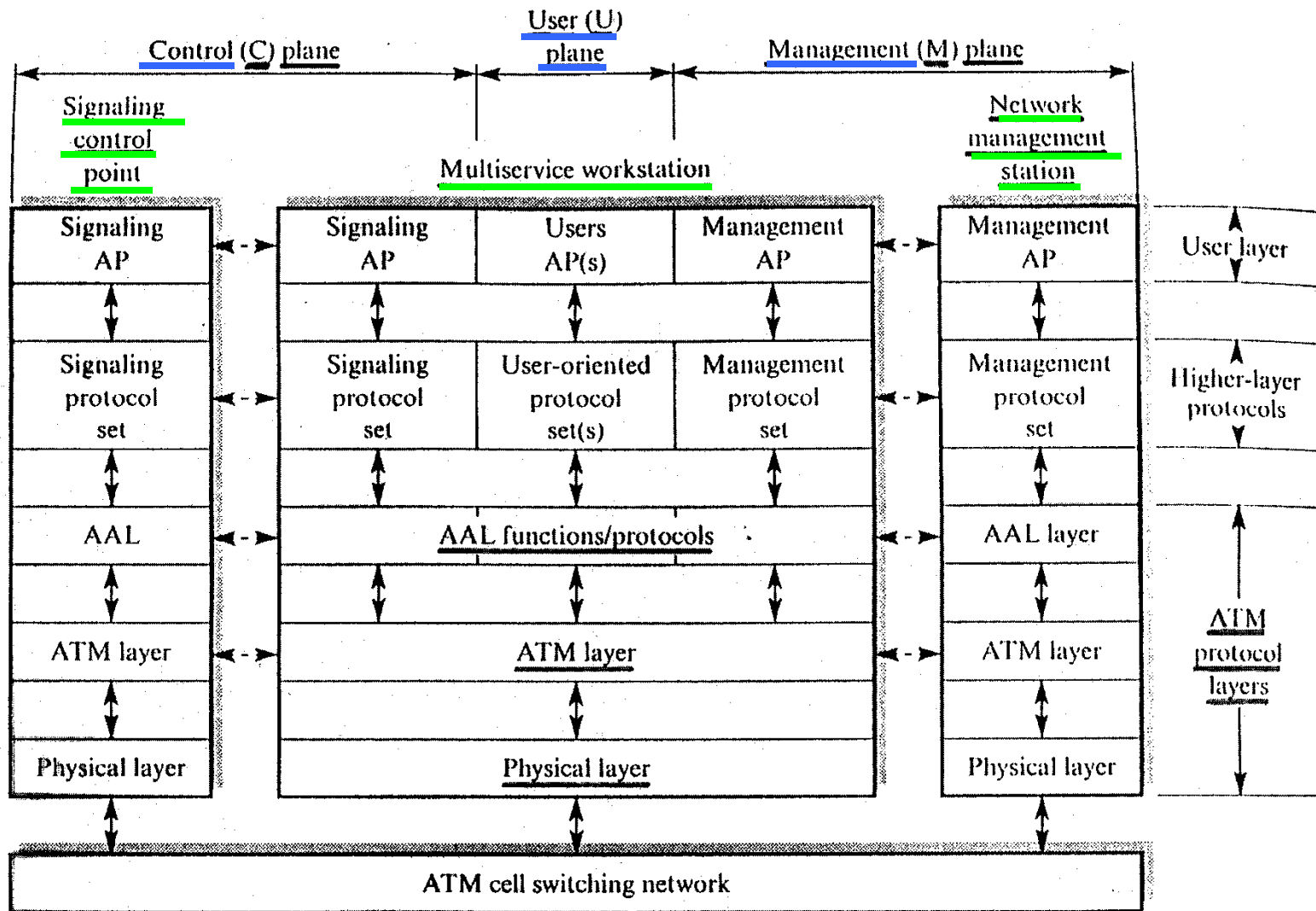
**Bild 4.17.** Struktur einer Zelle (aus: [Boc 97])

# Verbindungsauf- / abbau in ATM-Netzen

Message	Meaning when sent by host	Meaning when sent by network
SETUP	Please establish a circuit	Incoming call
CALL PROCEEDING	I saw the incoming call	Your call request will be attempted
CONNECT	I accept the incoming call	Your call request was accepted
CONNECT ACK	Thanks for accepting	Thanks for making the call
RELEASE	Please terminate the call	The other side has had enough
RELEASE COMPLETE	Ack for RELEASE	Ack for RELEASE



# ATM – Protokollhierarchie



AP = Application process  
AAL = ATM adaptation layer

ATM = Asynchronous transfer mode

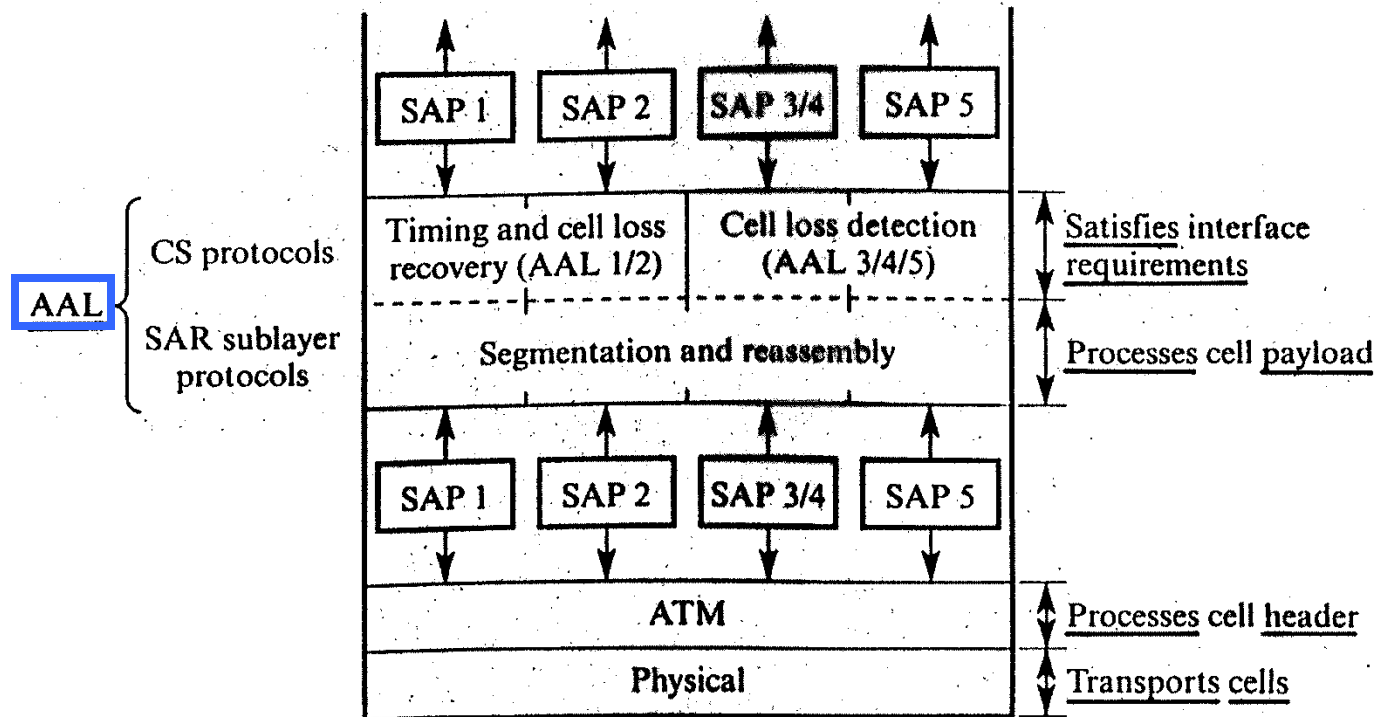


# ATM Adaptation Layers (AALs)

## (a) AAL-Charakteristika:

Service type			
AAL 1	AAL 2	AAL 3/4	AAL 5
Timing relationship		Yes	
Bit rate		No	
Constant / CBR		Variable / VBR : variable bit rate	
Mode		Connection-oriented	
		Connectionless	

## (b)



CS = Convergence sublayer

SAR = Segmentation and reassembly



Class	Description	Example
CBR	Constant bit rate	T1 circuit
RT-VBR	Variable bit rate: <b>real time</b>	<b>Real-time</b> videoconferencing
NRT-VBR	Variable bit rate: <b>non-real time</b>	<b>Multimedia</b> email
ABR	Available bit rate	Browsing the Web
UBR	Unspecified bit rate	Background file transfer

Fig. 5-69. The ATM service categories.



Service characteristic	CBR	RT-VBR	NRT-VBR	ABR	UBR
Bandwidth guarantee	Yes	Yes	Yes	Optional	No
Suitable for real-time traffic	Yes	Yes	No	No	No
Suitable for bursty traffic	No	No	Yes	Yes	Yes
Feedback about congestion	No	No	No	Yes	No

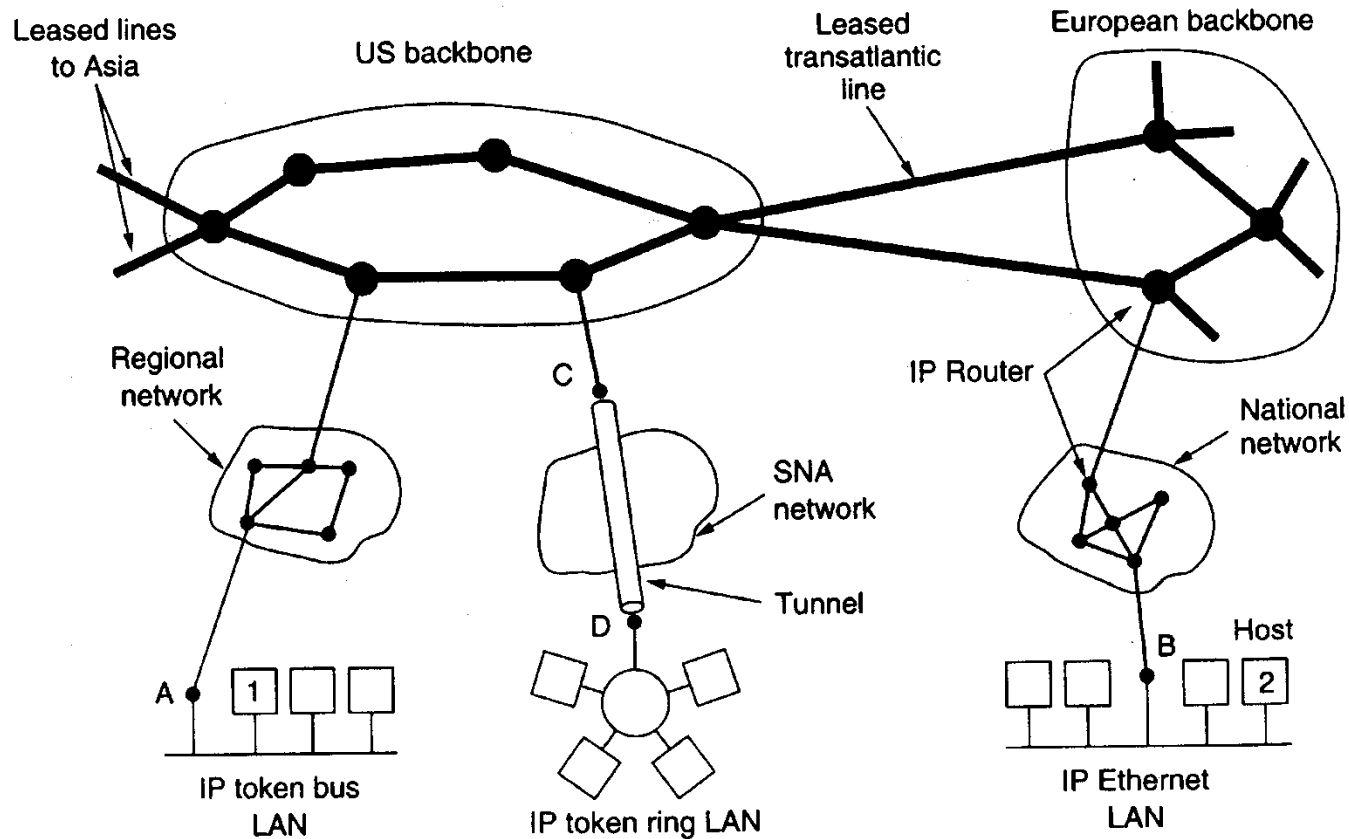
Fig. 5-70. Characteristics of the ATM service categories.

Parameter	Acronym	Meaning
Peak cell rate	PCR	Maximum rate at which cells will be sent
Sustained cell rate	SCR	The long-term average cell rate
Minimum cell rate	MCR	The minimum acceptable cell rate
Cell delay variation tolerance	CDVT	The maximum acceptable cell jitter
Cell loss ratio	CLR	Fraction of cells lost or delivered too late
Cell transfer delay	CTD	How long delivery takes (mean and maximum)
Cell delay variation	CDV	The variance in cell delivery times
Cell error rate	CER	Fraction of cells delivered without error
Severely-errored cell block ratio	SECBR	Fraction of blocks garbled
Cell misinsertion rate	CMR	Fraction of cells delivered to wrong destination

Fig. 5-71. Some of the quality of service parameters.

# 6.8 Das globale Internet

➤ Das Internet als **Interkonnektion von Rechnernetzen**



Typen von (Rechnernetz-) Knoten im Internet:

- Endsysteme/ “Hosts“ (Arbeitsrechner)
- Vermittlungsrechner: (IP-) “Router“, “Gateway“-Rechner, “Bridges“

➤ **Netzkopplungen im Internet:**

• **Erleichterungen**

- universelles, standardisiertes (seitens IETF) **IP**-Protokoll
- Datagrammdienst vereinfachend für Kopplungen

• **Schwierigkeiten**

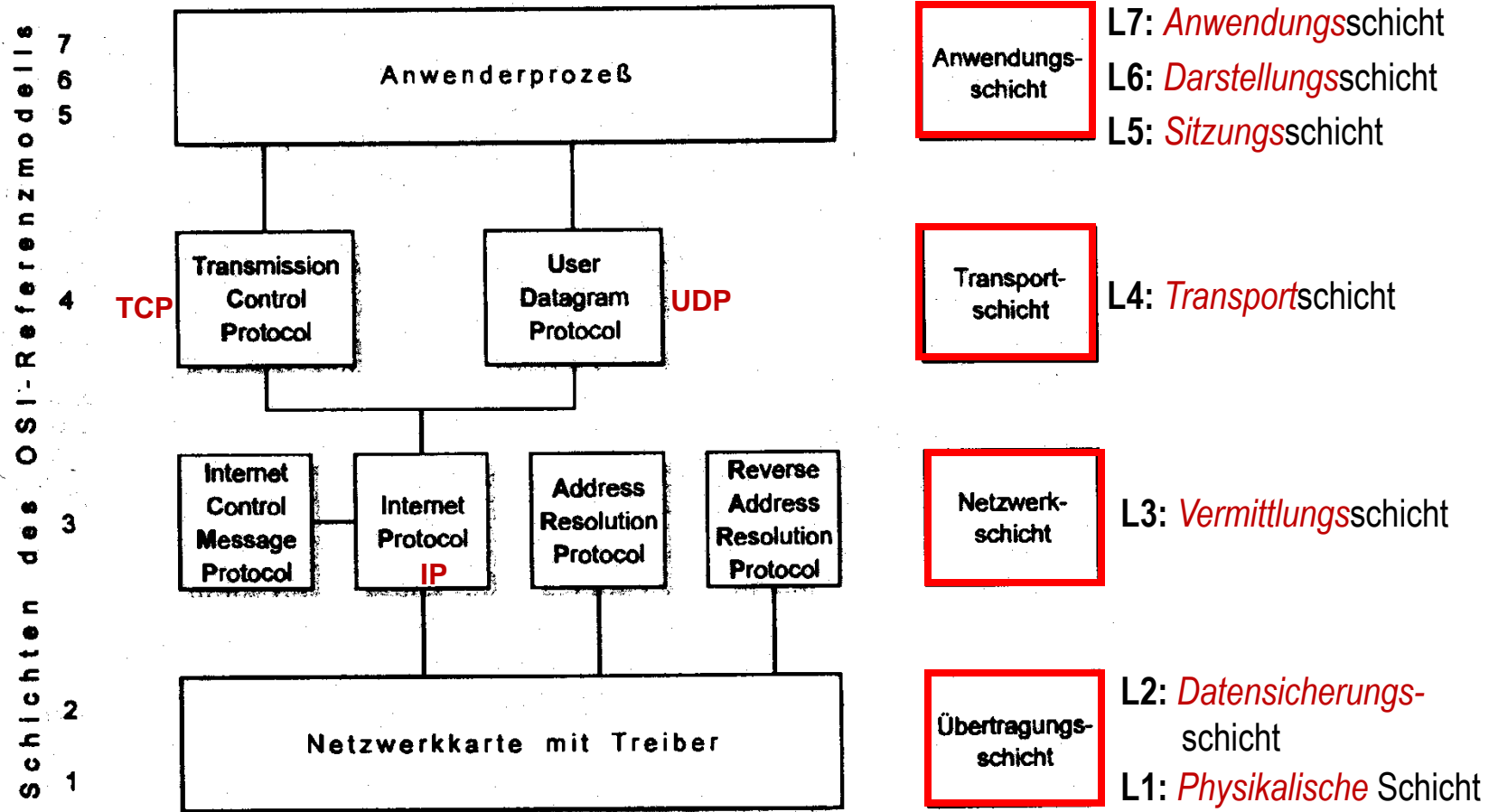
- verschiedene Netzprovider (→ Netzmanagement ?!)
- Probleme der Netzsicherheit
- Kopplung mit verbindungsorientierten Netzen  
(sowohl paket- als auch leitungsvermittelte)
- IPv4 (Version 4) parallel zu IPv6

*nota bene:* vgl. auch allg. Diskussion zu “Interkonnektion von Netzen“  
(siehe Abschn. 6.5)



➤ Die **Internet – Protokollhierarchie** (Grobstruktur)

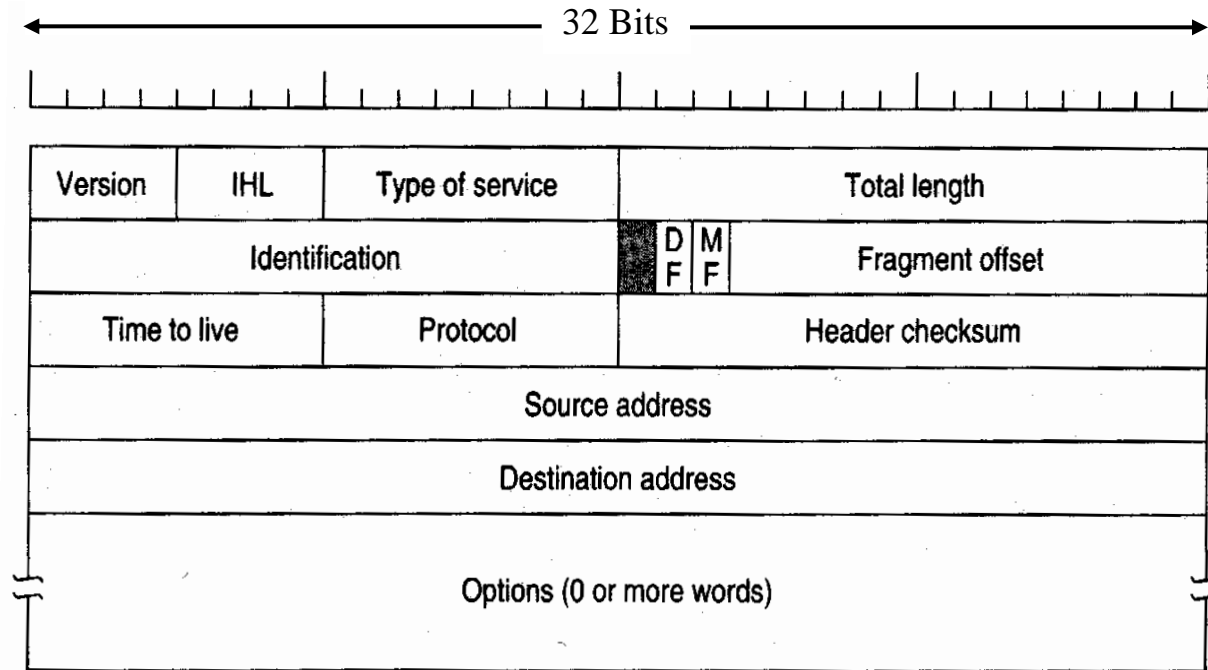
**ISO / OSI-Bezeichnungen:**



*nota bene:* Zu einigen der Internet-Protokolle, insbesondere IP und TCP, vgl. Grundlagen der Systemsoftware (GSS)



- Zur **Syntax** und **Semantik** der **IP-Protokolldateneinheiten**  
(auch: “IP-Datagramme“):



wobei:

- **Version:** Version-Nr.
- **IHL:** IP-Headerlänge (in 32-Bit-Worten)
- **Type of Service:** Typ der DÜ (z.B. hohe Zuverlässigkeits- vs. Echtzeitanforderungen)
- **Total length:** Gesamtlänge des IP-Datagramms ( $\leq 64\text{kByte}$ )
- **Identification, ..., Fragment offset:** für Fragmentierung/Reassemblierung
- **Time to live:** Begrenzung der Paket-Lebensdauer
- **Protocol:** (global eindeutige) Nummer des benutzten Transportprotokolls
- **Header checksum:** Prüfsumme für Header
- **Source/Destination address:** IP-Adresse (bitte NICHT: „die IP“ !) von Sender/Empfänger



# Der “**Best Effort**” - Charakter des Internet

→ zur Erinnerung:



Variationsbreite der Vermittlungstechniken:

“**Standleitung**” ... **S-ISDN** ... **ATM** ... **X.25** ... “**Datagrammorientierte Paketvermittlung**”

sehr **statische**

**Reservierung der  
Netzbetriebsmittel;**

i.a. schlechte BM-  
Auslastung/-nutzung

**flexible, lastadaptive  
Betriebsmittelnutzung;**

i.a. effizienter, aus-  
falltoleranter, ...

➤ Dienstleistung im traditionellen Internet:

“**Best Effort**” ≡ nach bestem Bestreben (“man tut was man kann“ ... → evtl. nicht viel !)

- “Best Effort“ - Charakter von Netzen i.d.R. inakzeptabel für Echtzeitkommunikation (Details, vgl. Kap. 8), da Mindestqualität des Netzes unabdingbar für (noch) akzeptable Qualität der anwendungsorientierten Dienste (z.B. bei Audio-/Videokommunikation)

→ geforderte Netzqualitäten, u.a.

- **limitierte Paketverlustrate**,
- **limitierte Verzögerungsschwankungen** bei Ende-zu-Ende-Paketübertragungen,
- **“garantierter“ Minstdurchsatz** (auch über kurze Beobachtungsintervalle),
- . . .

- Maßnahmen zur Verbesserung der Dienstgüte (QoS) im Internet:

- **“dumb fat pipe“** → **Überdimensionierung** (wenig komplex, aber evtl. teuer)
- **a priori Reservierung von Ressourcen** → z.B. **Integrated Services Architecture** (ISA; auch: **IntServ**) und **RSVP**, aber: evtl. Blockierungszustände für anfragende Benutzer, hohe Komplexität (u.a. viel Zustandsinfo zu führen)
- **Priorisierung von Übertragungen** → z.B. **DiffServ**, aber: evtl. Engpass auf höchster Prioritätsebene



## Der **“Black Box“ –Charakter** des Internet

### ➤ Gründe für die limitierte Beobachtbarkeit des Internet:

- zahlreiche Betreiber → u.a. Konkurrenzkampf, Sicherheitsanforderungen, ... zwingen zu “Abschottung“ des Netzes
- eine enorm hohe Anzahl von Benutzern und Endsystemen existiert → eingeschränkte Möglichkeiten der Datenfassung
- höchst dynamisches Benutzerverhalten (ergo: keine Stationarität im Netzverhalten zu erwarten)
- stark limitierte Möglichkeiten zu Experimenten unter eng kontrollierten experimentellen Randbedingungen, u.a.m.

### ➤ **Vor-/Nachteile** des “Black Box“ –Charakters, u.a.:

- + :
  - Netzsicherheit (bedingt) verbessert
  - Entscheidungen weniger komplex wegen des im allg. erzwungenen Verzichts auf Einbeziehung von globaler Zustandsinfo bzgl. des Netzes
  - FuE möglich, um Limitationen bzgl. (legitimer (!)) Beobachtbarkeit zu entschärfen 😊
- :
  - Verhaltensprognosen schwierig bis unmöglich
  - Accounting sehr problematisch und allgemein Netzmanagement erschwert
  - Reaktion auf gegenwärtigen Netzzustand nur bedingt möglich (→ unscharfe Zustandsinfo)

➤ **Möglichkeiten und Grenzen für Messungen** trotz “Black Box“ –Charakter:

- Messungen an anwendungsnahen Schnittstellen in zugreifbaren Endsystemen relativ problemlos → z.B. Lastmessungen
- Charakterisierung des Netzverhaltens aus der Sicht von gezielt etablierten Kommunikationsverbindungen (z.B. über TCP)
- langfristige Beobachtungen des zufällig vorgefundenen Netzverhaltens (z.B. umfangreiche Trace-Statistiken für ausgewählte Benutzer oder Endsysteme)

# Sicherheit im Internet

## ➤ Angriffsmöglichkeiten



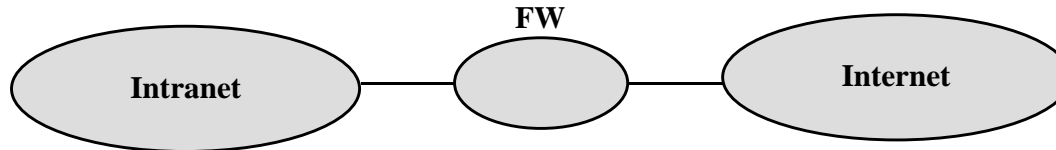
### ⇒ Beispiele

- Modifikation von Adressen (z.B. sendeseitige IP-Adresse)  
→ Vortäuschung eines falschen Absenders
- Abhören von Passwörtern (z.B. bei Telnet-Dienst evtl. unverschlüsselt übertragen) → spätere unbefugte Passwort-Benutzung
- “**Denial-of-Service**“ (DoS)- Angriffe → unbefugte Nutzung eines Dienstes (bzw. von Ressourcen) seitens zahlreicher Benutzer; evtl. “verteilter Angriff“ von zahlreichen Endsystemen aus (z.B. Angriffe auf Server-Rechner)
- absichtliche Vernichtung von IP-Datagrammen auf Weg von Sender zu Empfänger → im allg. nicht als Angriff sichtbar (da protokollkonform)

u.v.a.m.

➤ **Spezielle Schutzmechanismen** (u.a. für das Internet):

- **“Firewall“** (FW) :

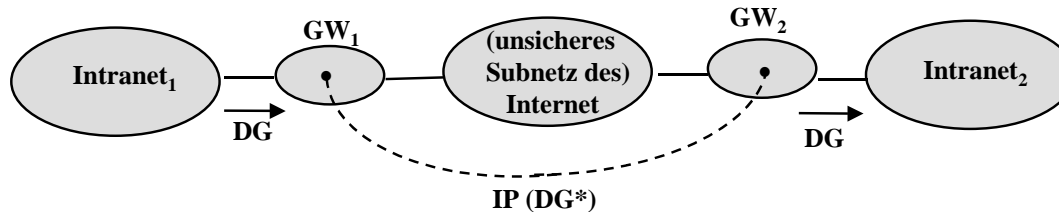


→ u.a. Überprüfung von Adressen, von geforderten Diensten, ...  
seitens FW

- **Verschlüsselung:**

→ grundsätzlich auf unterschiedlichen Ebenen der Protokollhierarchie  
(unterschiedlicher Aufwand und resultierender Grad an Sicherheit)

- **“Tunneling“**:



→  $GW_1$  verschlüsselt Datagramm  $DG$  (als  $DG^*$ ) und bettet  $DG^*$  in IP-Paket ein;  
 $GW_2$  entpackt  $IP(DG^*)$  und entschlüsselt  $DG^*$ .



*nota bene:* Weitere Details zu Netzsicherheit im allg., vgl. Kap. 12 und einschlägige Lehrveranstaltungen (u.a. zu IT- & Network Security).



## Einige Internet - Trends

- **optische Kommunikationsinfrastrukturen** (incl. optische Vermittlungsrechner) ... und dann doch die “*dumb fat pipes*“ ?!
- **Mobilkommunikation** → z.B. *Mobile IP*:  
u.a. Vergabe statischer IP-Adressen an mobile Endsysteme problematisch (nota bene: hierarchische Adressierung bei IP)
- **Echtzeitanwendungen** → u.a. *VoIP – Voice over IP* / Internet-Telefonie; *IPTV – Internetfernsehen*; Medienkommunikation wie bei Videokonferenzen, u.v.a.m.
- **IPv4 → IPv6** :  
insbesondere “riesige“ Adressfelder (128 bit), genauer:  
«Das neue Format stellt 340 Sextillionen Adressen bereit - das sind 600 Billionen Adressen auf jeden Quadratmillimeter der Erdoberfläche», laut Verband der deutschen Internetwirtschaft *eco*.  
Zudem: mehr Sicherheit sowie bessere QoS.  
Kein abrupter Übergang von IPv4 zu IPv6 (“sanfte Migration“):  
zunächst “IPv6-Inseln“, später “IPv4-Inseln“.