5 Weitverkehrsnetze und Routing

5.1 Das Prinzip der Paketvermittlung

5.2 Virtuelle Verbindungen oder Datagramme?

5.3 Wegewahl (Routing) für Punkt-zu-Punkt-Netze

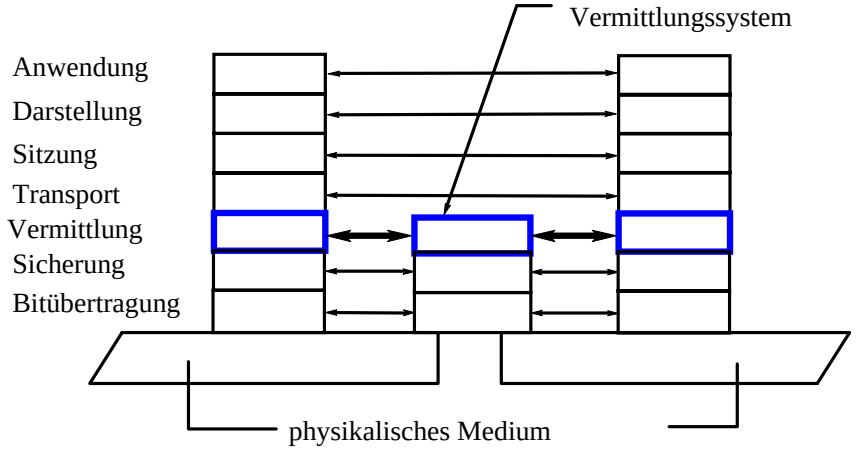
5.4 Wegewahl (Routing) für Multicast-Netze

5.5 Überlastkontrolle in der Vermittlungsschicht

5.6 Beispiele: IP, X.25, ATM

5.1 Das Prinzip der Paketvermittlung

Die Vermittlungsschicht im Referenzmodell



ISO-Definition für die Vermittlungsschicht

Die Vermittlungsschicht stellt die Fähigkeit bereit, Netzverbindungen zwischen offenen Systemen aufzubauen, zu betreiben und abzubauen.

Die Vermittlungsschicht bietet den Transportinstanzen **Unabhängigkeit von Wegewahl- und Vermittlungsentscheidungen**, die mit dem Aufbau und Betrieb einer Netzverbindung verbunden sind.

Aufgaben der Vermittlungsschicht

- Wegewahl und Vermittlung
- Verbindungsaufbau und -abbau
- Multiplexen von Netzverbindungen
- Segmentierung
- Fehlererkennung/Fehlerbehebung (Ende-zu-Ende)
- Sicherstellung der Paketreihenfolge
- Flußkontrolle (Ende-zu-Ende)

Eine möglichst große Vielfalt von Netzkonfigurationen soll unterstützt werden.

5.2 Virtuelle Verbindungen oder Datagramme?

Virtuelle Verbindung
Der Weg durch das Netz wird beim Aufbau der virtuellen Verbindung ausgewählt, d.h. für jede neue virtuelle Verbindung findet in jedem Netzknoten nur einmal eine Wegewahlentscheidung statt. Der ganze über diese virtuelle Verbindung fließende Verkehr nimmt denselben Weg durch das Netz.

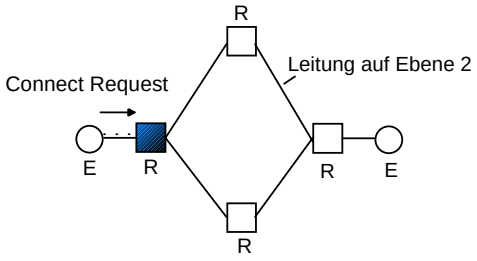
Datagramm
Die Zieladresse bestimmt in jedem Netzknoten auf dem Pfad die ausgehende Leitung. Für jedes Datagramm wird in jedem Knoten erneut eine Wegewahlentscheidung getroffen.

Virtuelle Verbindung oder Datagrammdienst?

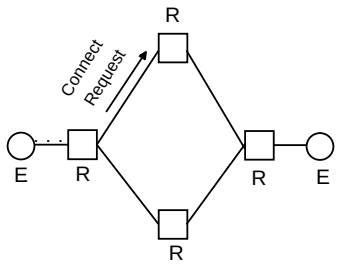
Virtuelle Verbindung

- "perfekter" Kanal durch das Netz
  - Ordnung der Nachrichten (Sicherstellung der Reihenfolge)
  - Fehlerüberwachung (verlorene und duplizierte Pakete)
  - Flußkontrolle
- Phasen
  - Verbindungsaufbau
  - Datenübertragung
  - Verbindungsabbau
- Vorteile
  - Niedriger Mehraufwand für die Adressierung während der Datenübertragung
  - Keine Neusortierung oder Fehlerüberwachung im Endsystem nötig (Transportschicht)

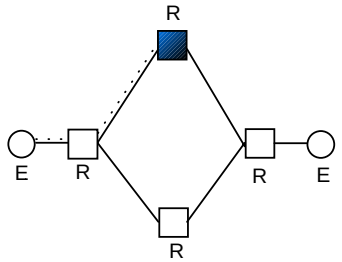
Beispiel: Aufbau einer virtuellen Verbindung



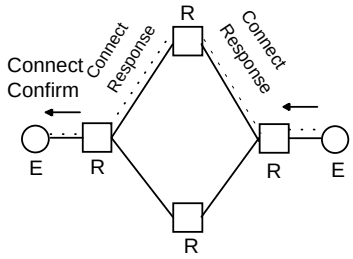
a) Festlegen des Weges



b) Aufbauphase der 1. Teilstrecke



c) Virtueller Verbindungsabschnitt existiert, Festlegung der Wegefortsetzung

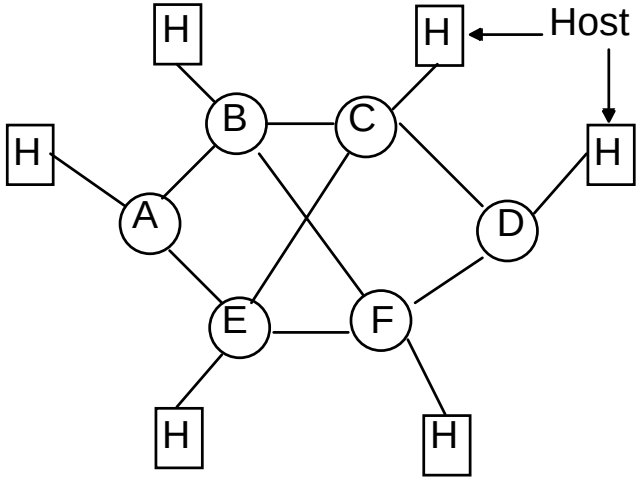


d) nach weiteren Schritten virtuelle Verbindung fertigestellt

Implementierung von virtuellen Verbindungen innerhalb des Netzes

In jedem Netzknoten werden Tabellen mit Zustandsinformationen über bestehende virtuelle Verbindungen verwaltet.

(a) Beispiel-Subnetz:

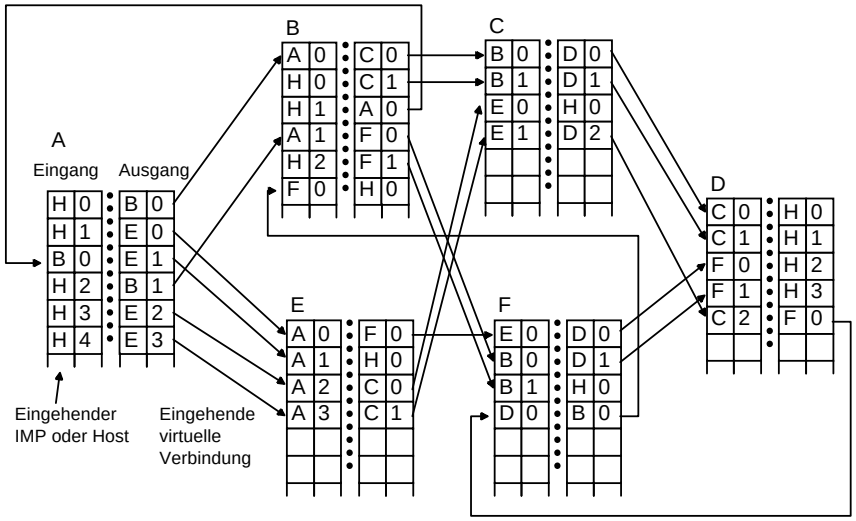


(b) Acht virtuelle Verbindungen durch das Subnetz:

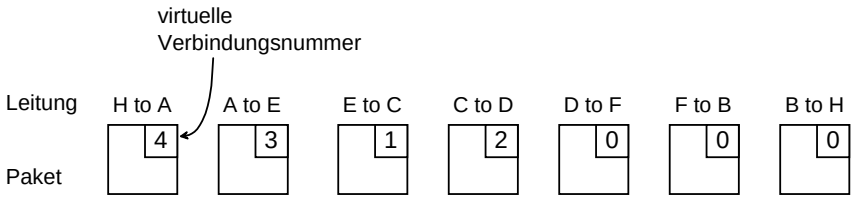
Ausgehend von A	Ausgehend von B
0 – ABCD	0 – BCD
1 – AEFD	1 – BAE
2 – ABDF	2 – BF
3 – AEC	
4 – AECDFB	

Zustandsinformation in den Netzknoten

(c) Router-Tabellen für die virtuellen Verbindungen in (b)



(d) die virtuelle Verbindungsnummer (Kanalnummer) kann von Teilstrecke zu Teilstrecke verschieden sein. Hier die Beispielverbindung 4 - AECDFB



Datagramm

Jedes Paket (Datagramm) wird als isolierte Einheit betrachtet.

- Volle Zieladresse in jedem Paket
- Pakete können außerhalb der Reihenfolge eintreffen
- Keine Fehlerüberwachung, keine Flußkontrolle

Vorteile

- Primitiver als virtuelle Verbindungen, daher einfacher zu implementieren
- Kein Verbindungsaufbau und -abbau, deshalb niedriger Overhead für kurzlebige Verbindungen
- flexibler und zuverlässiger
- besser geeignet für heterogene Subnetze

5.3 Wegewahl (Routing) für Punkt-zu-Punkt-Netze

Besondere Netztopologien

Wegfall des Wegewahlproblems auf Broadcast-Medien, z.B. in einem Segment eines Busses oder Rings (lokale Netze): hier ist keine Wegewahl erforderlich, da jede Nachricht alle Empfänger erreicht.

5.3.1 Routing-Algorithmen

Aufgabe

Leitwegbestimmung für Pakete durch das Netzwerk vom Quellsystem zum Zielsystem

Der **Leitwegbestimmungsalgorithmus** eines Vermittlungsrechners (Knotens) entscheidet, auf welcher Ausgangsleitung ein eingegangenes Paket weitergeleitet wird.

- Datagramm: individuelle Entscheidung für jedes Paket
- Virtuelle Verbindung: Leitwegbestimmung nur beim Verbindungsaufbau

Wünschenswerte Eigenschaften eines Algorithmus

- Korrekt
- Einfach
- Robust (Rechner- oder Leitungsausfälle)
- Stabil (Gleichmäßige Ergebnisse)
- Fair
- Optimal

Algorithmen für die Leitwegbestimmung

Optimierungskriterien

- Durchschnittliche Paketverzögerung
- Gesamtdurchsatz

Zielkonflikt, daher gebräuchlich:

- Minimierung der Teilstrecken (hops) pro Paket
  - reduziert Verzögerung
  - vermindert benötigte Bandbreite
  - steigert Durchsatz

Leitwegbestimmung

Klassifikation der Verfahren

1. Statische (nicht-adaptive) Verfahren

- keine Berücksichtigung des aktuellen Netzzustands
- gehen von Mittelwerten aus
- Leitweg zwischen i und j wird für alle i, j vor der Inbetriebnahme des Netzwerks bestimmt
- keine Änderung während des Betriebs (statisches Routing)

2. Adaptive Verfahren

- Entscheidungen basieren auf aktuellem Netzzustand
- Messungen/Schätzungen der Topologie und des Verkehrsaufkommens
- Weitere Unterteilung in
  - Zentralisierte Verfahren
  - Isolierte Verfahren
  - Verteilte Verfahren

Statische Leitwegbestimmung

Beim statischen Routing ist die gesamte Topologie des Netzes einer zentralen Stelle bekannt. Sie berechnet die optimalen Pfade für jedes Paar (i,j) von Knoten, erstellt daraus die Routing-Tabellen für die einzelnen Knoten und versendet diese.

Sinnvoll, wenn das Netz relativ klein und relativ statisch ist.

Mehrfach-Leitwegbestimmung (multipath routing)
Benutzung alternativer Leitwege zwischen jedem Knotenpaar (i,j)

- Häufigkeit der Nutzung abhängig von der Güte der Alternative
- Höherer Durchsatz durch Verteilung des Datenverkehrs auf mehrere Pfade
- Höhere Zuverlässigkeit, da der Ausfall eines Links nicht so schnell zur Unerreichbarkeit von Knoten führt

Realisierung

- Jeder Knoten enthält Routing - Tabelle mit je einer Spalte für jeden möglichen Zielknoten

Z	A1	G1	A2	G2		An	Gn

Z ... Ziel
Ai ... i-beste Ausgangsleitung
Gi... Gewicht für Ai

- Gi bestimmt die Wahrscheinlichkeit, mit der Ai benutzt wird: $\left(\sum_{i=1}^{n} G_i = 1\right)$

Auswahl der Alternativen

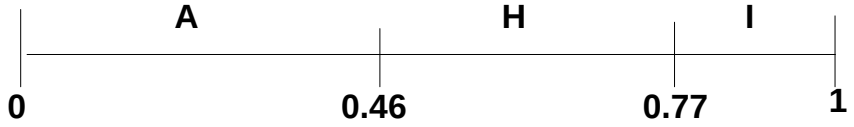
```
Generieren einer Zufallszahl z (0 <= z <= 1)

Wähle A1, falls 0  <= z <= G1

Wähle A2, falls G1 <= z <  G1 + G2


Wähle An, falls G1 + G2 + .. + G(n-1) <= z < 1
```

Beispiel : Ziel B, Quelle J



Statische Leitwegbestimmung, Beispiel

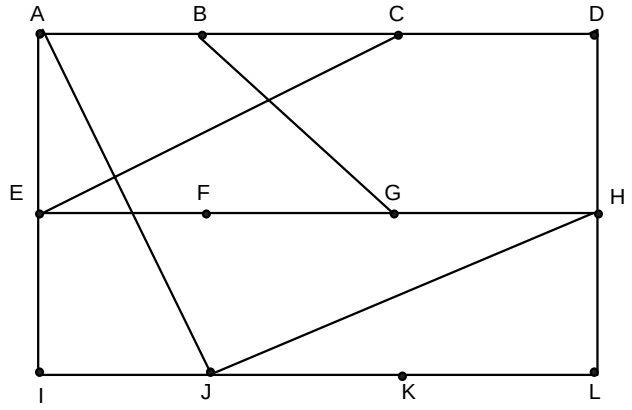


Tabelle des Knotens J mit Leitwegalternativen

Ziel	Erste	Wahl	Zweite	Wahl	Dritte	Wahl
A	A	0.63	I	0.21	H	0.16
B	A	0.46	H	0.31	I	0.23
C	A	0.34	I	0.33	H	0.33
D	H	0.50	A	0.25	I	0.25
E	A	0.40	I	0.40	H	0.20
F	A	0.34	H	0.33	I	0.33
G	H	0.46	A	0.31	K	0.23
H	H	0.63	K	0.21	A	0.16
I	I	0.65	A	0.22	H	0.13
-						
K	K	0.67	H	0.22	A	0.11
L	K	0.42	H	0.42	A	0.16

Bestimmung der Leitwegtabellen

Statisches Verfahren

- Tabellen werden vom Netzwerkoperator zentral erstellt
- Tabellen werden vor Inbetriebnahme der Knoten geladen und dann nicht mehr verändert

Eigenschaften

- einfach
- gute Ergebnisse bei relativ konstanter Topologie und konstantem Verkehr

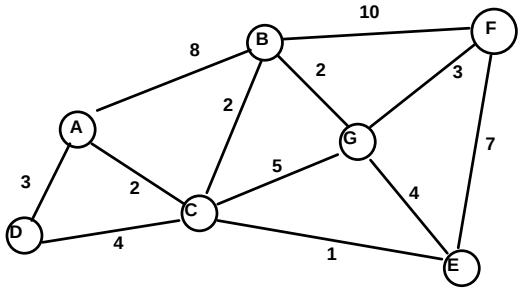
aber:

- schlecht bei stark variierendem Verkehrsaufkommen und bei Topologieänderungen
- schlecht bei großen Netzen (skaliert nicht)

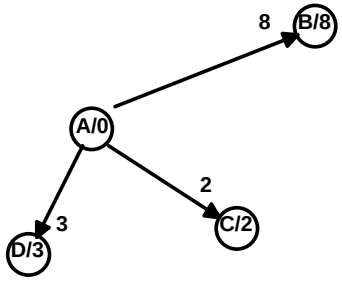
In der Praxis noch immer sehr häufig benutzt!

Leitwegbestimmung mit dem Algorithmus "kürzeste Wege"

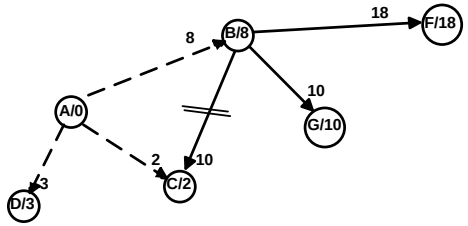
Graphenalgorithmus



a) Netzwerktopologie mit gewichteten Verbindungen

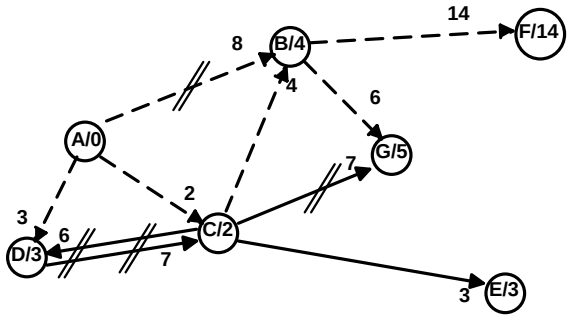


b) Markierung von Knoten A aus

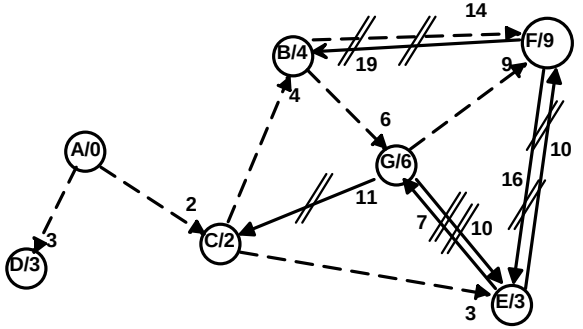


c) Markierung von Knoten B aus

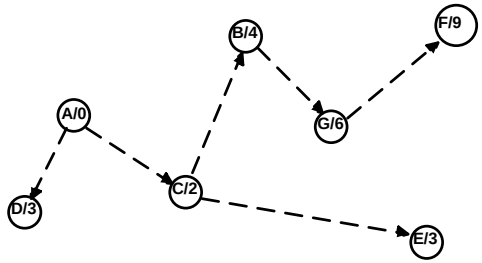
Kürzeste Wege (2)



d) Markierung von Knoten C, D aus



e) Markierung von Knoten B, G, E, F aus



f) Keine neuen Markierungen mehr möglich. Baum zeigt kürzeste Wege von A aus zu allen Knoten

Zentralisierte adaptive Leitwegbestimmung

Prinzip

- Im Netz: RCC (Routing Control Center, Leitwegsteuerzentrum)
- Jeder Knoten sendet periodisch Status-Information zum RCC
  - Liste der verfügbaren Nachbarn
  - aktuelle Warteschlangenlängen
  - Auslastung der Leitungen, etc.
- RCC sammelt Informationen und berechnet optimalen Pfad für jedes Knotenpaar, berechnet die einzelnen Leitwegtabellen und verteilt sie an die Knoten

Beispiel TYMNET

- paketvermitteltes Netz in den USA
- ca. 1000 Knoten
- verwendet virtuelle Verbindungen
- RCC-gesteuert

Zentralisierte adaptive Leitwegbestimmung

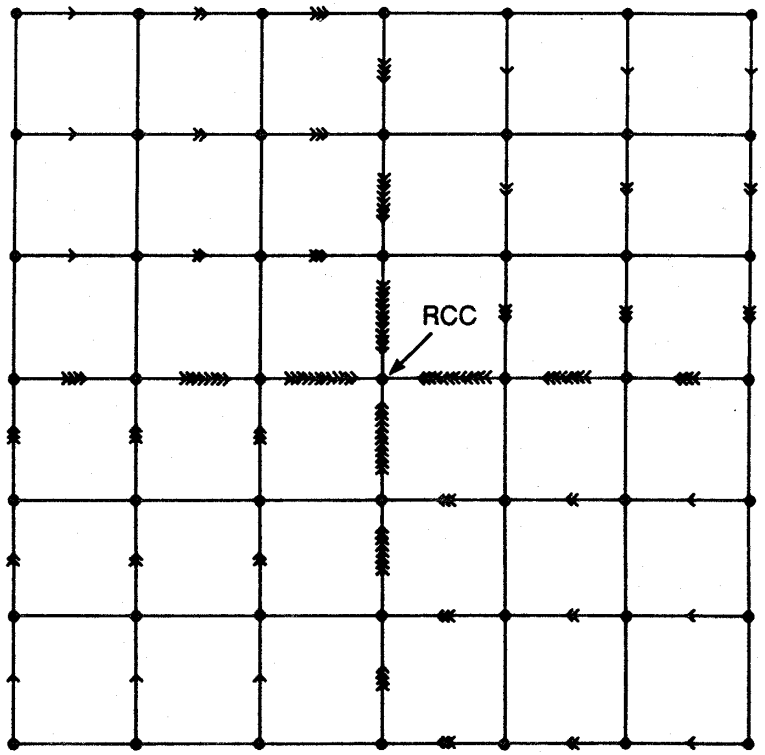
Eigenschaften

- RCC hat vollständige Information ⇒ die Entscheidungen sind optimal
- Die Einzelknoten sind von der Leitwegberechnung befreit

Aber:

- Berechnung muß oft durchgeführt werden (ca. jede Minute oder öfter)
- Verkehrskonzentration in der Nähe des RCC ("performance bottleneck")
- geringe Robustheit ("single point of failure")
- keine korrekte Entscheidung bei Netzpartitionierung
- Knoten erhalten Tabellen zu unterschiedlichen Zeiten ⇒ Inkonsistenzen sind möglich

Zentralisierte Leitwegbestimmung



Isolierte adaptive Leitwegbestimmung

Prinzip

- Kein Austausch von Routing-Information zwischen Knoten
- Entscheidungen basieren ausschließlich auf lokalen Informationen

Beispiele für Verfahren

- Backward Learning (Baran)
- Flooding
- Delta-Routing (Rudin, 1976)

Algorithmus "Backward Learning"

- Knoten "lernt" von eintreffenden Paketen
  Paket (..., Q, Z, ...)
  Q = Quell–Knoten
  Z = Teilstreckenzähler (hop counter)

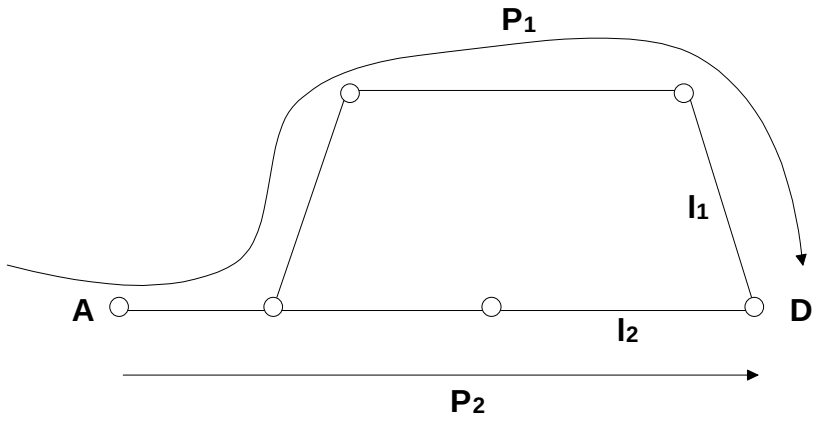
  Paket wird auf Leitung L empfangen
  ⇒ Q ist über L in Z Teilstrecken erreichbar

- Leitwegtabelle im Knoten:
  L-Tabelle: Jeder Eintrag is ein Tripel aus
  (Zielknoten, Ausgangsleitung, Z_{min})

- Aktualisierung der Leitwegtabelle
  Knoten empfängt Paket (..., Q, Z, ...) auf L

```
if not      ( Q in L-Tabelle )
then        Add(Q,L,Z)
else        if Z < Zmin
            then Update(Q,L,Z)
```


Backward Learning: Beispiel




```
P1( ..., A, 4,...)   →   Add (A, l1, 4)
P2( ..., A, 3,...)   →   Update (A, l2, 3)
```

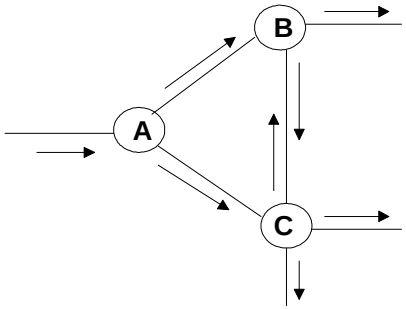
Problem
Algorithmus registriert keine Verschlechterungen

Lösung
periodisches Löschen der Leitwegtabellen (neue Lernperiode)
Löschzeitpunkte kritisch:

- zu häufig: Netz ist überwiegend in der Lernphase
- zu selten: zu langsame Reaktion auf Verschlechterungen

Flooding-Algorithmus

Ein empfangenes Paket wird auf allen Leitungen weitergeleitet außer auf derjenigen, auf der es angekommen ist.



Problem: Unendliche Anzahl von Duplikaten
Begrenzung des Prozesses: Streckenzähler ("hop counter") im Paketkopf

- Initialisierung mit dem Durchmesser des Netzes = längstem Pfad im Netz (worst case)
- Wird auf jeder Teilstrecke um 1 decrementiert
- Duplikate erhalten den Streckenzähler des Originals
- Zähler = 0: Paket wird vom Router weggeworfen

Eigenschaften

- sehr robust, sehr einfach, aber
- große Anzahl von Duplikaten, große Netzbelastung
  $\Rightarrow$ Einsatz nur für sehr spezielle Anwendungen

Algorithmus "Delta-Routing" (1)

Prinzip
Kombination von isoliertem und zentralisiertem Verfahren.

- Jeder Knoten berechnet periodisch die "Kosten" seiner Leitungen und sendet diese zum RCC (Kosten = Funktion aus Verzögerung, Warteschlangenlänge, ...)
- RCC berechnet
  - die k besten Pfade von Knoten i nach Knoten j (für alle i, j)
  - Liste der zum besten Pfad "äquivalenten" Pfade

$$c_{ij}^{n} - c_{ij}^{1} < \delta \quad \text{mit}$$

c_{ij}^{m} = Gesamtkosten des m-besten Pfads

- RCC sendet jedem Knoten für jedes mögliche Ziel eine Liste von äquivalenten Pfaden
- Jeder Knoten darf zwischen den *äquivalenten* Pfaden frei wählen, z.B.
  - zufällig
  - die Leitung mit den aktuell geringsten Kosten

Algorithmus "Delta Routing" (2)

Wahl von δ Verschieben der Autorität zwischen Knoten und RCC

$\delta \Rightarrow 0$: RCC trifft Entscheidung
$\delta \rightarrow \infty$: Knoten trifft Entscheidung

Bei geeigneter Wahl von δ bessere Leistung als bei rein isolierten oder zentralisierten Verfahren.

Verteilte Leitwegbestimmung

Prinzip

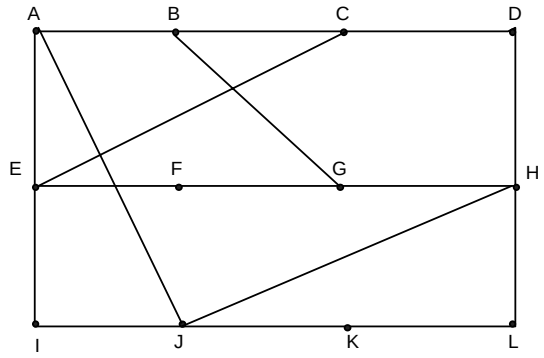
Die Knoten tauschen mit ihren Nachbarn Leitweginformationen aus:

- Jeder Knoten kennt "Entfernung" zu jedem Nachbarn
  - Anzahl der Teilstrecken (= 1)
  - Verzögerungszeit (Echo-Pakete)
  - Warteschlangenlänge, etc.
- Jeder Knoten sendet periodisch seinen Nachbarn eine Liste mit seinen geschätzten Entfernungen zu jedem Ziel
- X empfängt Liste E vom Nachbarn Y
  Entfernung (X, Y) = e
  Entfernung (Y, Z) = E(Z)
  $\Rightarrow$ Entfernung(X, Y) über Y : E(Z) + e

Die Tabelle mit den einem Knoten bekannten Distanzen heißt Distanzvektor. Das Verfahren heißt deshalb auch **"distance vector routing".**

Verteilte Leitwegbestimmung

Beispiel



Rechte Spalte: nach dem Eintreffen der Distanzvektoren neu geschätzte Verzögerung von J aus

	A	I	H	K		
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	-
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA Verzögerung=8	JI Verzögerung=10	JH Verzögerung=12	JK Verzögerung=6		

Hierarchische Leitwegbestimmung

Die Größe der Routing-Tabellen ist proportional zur Größe des Netzwerks:

- großer Speicherbedarf
- viel CPU-Zeit zum Durchsuchen der Tabellen
- viel Bandbreite zum Austausch von Routinginformation.

Hierarchische Leitwegbestimmung ab einer bestimmten Netzgröße notwendig:

- Knoten werden in Regionen gruppiert
- Jeder Knoten kennt
  - Details seiner Region
  - Leitweg zu allen anderen Regionen

Nachteil : nicht immer optimale Entscheidungen möglich

Beispiel

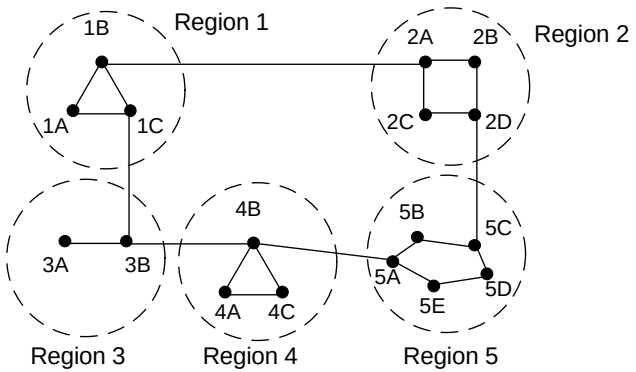


Tabelle für 1A

Ziel	Leitung	Teilstrecken
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchische Tabelle für 1A

Ziel	Leitung	Teilstrecken
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

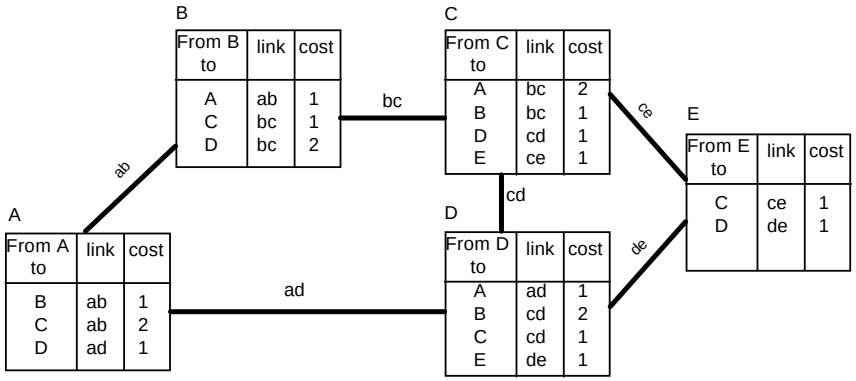
5.3.2 Routing im Internet

Distance Vector Routing

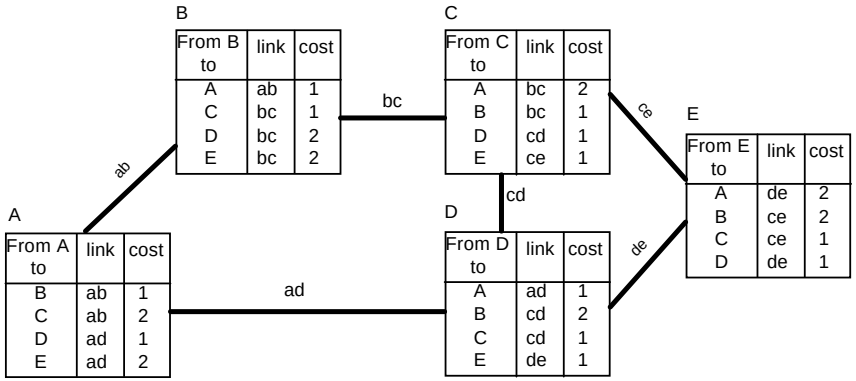
Das heute am meisten verwendete Verfahren im Internet ist ein adaptives verteiltes Verfahren auf der Basis von Distanzvektoren (distance vector routing). Das eingesetzte Protokoll heißt **RIP** (Routing Information Protocol).

Alle Internet-Router tauschen periodisch RIP-Nachrichten aus und aktualisieren ihre Routing-Tabellen beim Eintreffen von RIP-Nachrichten von ihren Nachbarn.

Beispiel für Routing mit Distanzvektoren



(a) E ist gerade hinzugekommen



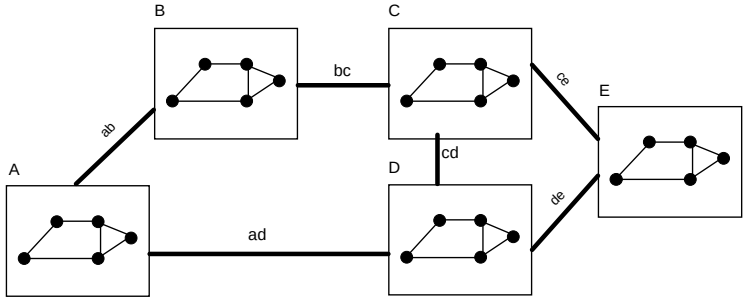
(b) Nach einer Runde von DVRP-Nachrichten

OSPF-Routing

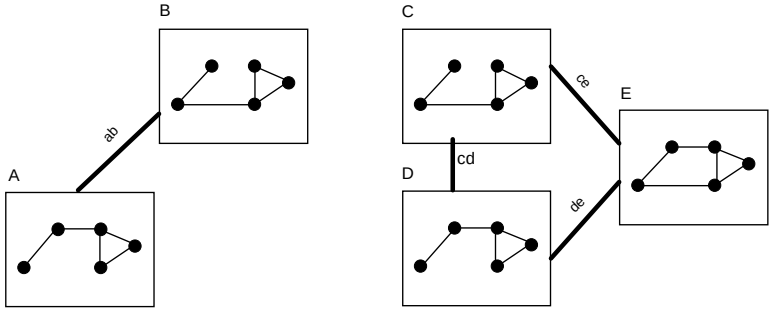
Ein zweiter wichtiger Routing-Algorithmus im Internet ist OSPF (Open Shortest Path First). Die Idee ist, daß alle Knoten jederzeit die gesamte Netztopologie kennen und lokal alle optimalen Pfade berechnen können. Wenn sich die Topologie ändert, tauschen die Knoten Änderungsnachrichten aus. Jeder Knoten unterhält lokal eine Datenbank über die gesamte Topologie.

Auf der Basis der vollen Topologie werden die optimalen Pfade zu allen anderen Knoten mit dem Algorithmus von Dijkstra (Shortest Path First = SPF) berechnet. Im Internet-Slang heißt der Algorithmus deshalb auch Open Shortest Path First (OSPF).

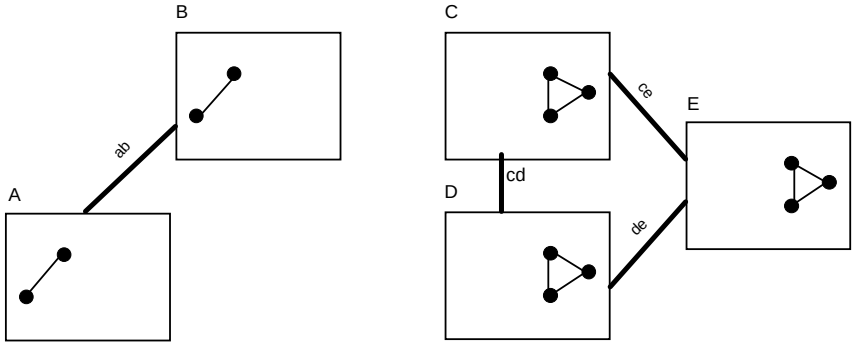
Beispiel für OSPF-Routing



(a) Netzwerk im stabilen Zustand



(b) Die Links bc und ad sind ausgefallen



(c) Nach einer Runde von OSPF-Nachrichten

5.4 Wegewahl (Routing) für Multicast-Netze

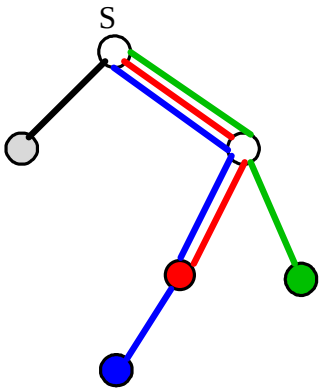
Warum ist Multicast wichtig für Multimedia?

- Multimedia-Anwendungen erfordern meist eine 1:n - Kommunikation.
- Beispiele:
  - Videokonferenz
  - Tele-Kooperation (CSCW) mit gemeinsamen Arbeitsbereich
  - near-Video-on-Demand
  - Verteil-Kommunikation (Broadcast)
- Digitale Video- und Audioströme haben sehr hohe Datenraten (• 1,5 MBit/s)
- Realisierung durch n einzelne Verbindungen würde die meisten Netze überlasten.

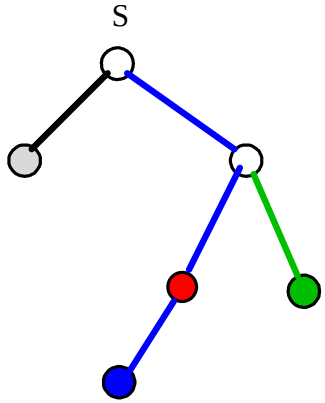
Motivation für Multicast

Mehr „Intelligenz“ im Netz verringert:

- die Last bei den Sendern
- die Last auf den Teilstrecken



n end-to-end connections



multicast-connection

Anforderungen an Multicast für Multimedia

- Unterstützung von isochronen Datenströmen mit **garantierter Dienstgüte**
  - maximale Ende-zu-Ende-Verzögerung (delay)
  - maximale Varianz in der Verzögerung (delay jitter)
  - maximale Fehlerrate (error rate)
    für eine vereinbarte Verkehrslast (Vertragsmodell)
- Erfordert eine Reservierung von Ressourcen in allen Links und Knoten im Netz
  - Bandbreite
  - CPU-Leistung
  - Pufferplatz
  - "schedulability"
- Erfordert Formate und Protokolle für eine Gruppenadressierung
- Erfordert neue Algorithmen für die Fehlerkorrektur (z.B. FEC oder Reliable Multicast)
- Erfordert Algorithmen für dynamisches Hinzufügen und Löschen von Teilnehmern

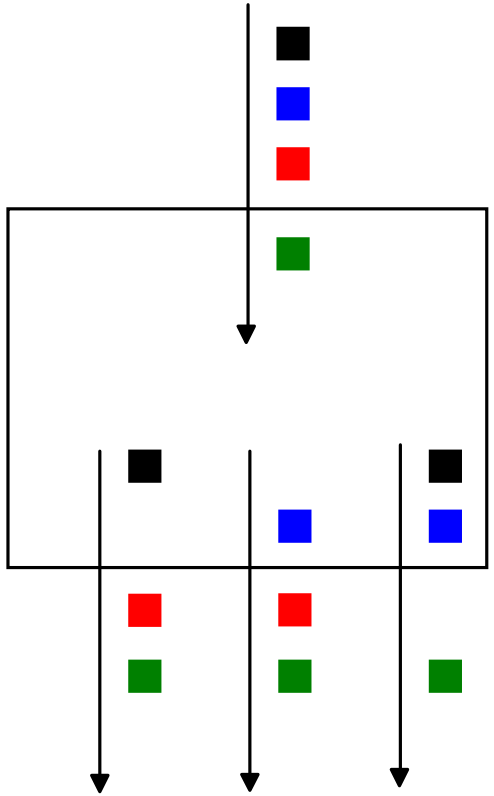
Multicast in LANs

Ethernet, Token Ring, FDDI usw:

- Die Topologie hat Broadcast-Eigenschaft
- Die Schicht-2-Adressen nach IEEE 802.2 erlauben die Einrichtung von Gruppenadressen für Multicast
- Aber: Ab Schicht 3 wurden viele Jahre lang nur Einzeladressen unterstützt! Deshalb wird die Muticast-Fähigkeit der LANs nicht ausgenutzt!

Multicast in der Netzwerkschicht

- Prinzip: Duplizierung von Paketen so "tief unten" im Multicast-Baum wie möglich
- Erfordert ein Multicast-Adressierungsschema in Schicht 3 und mehr "Intelligenz" in den Schicht 3 - Vermittlungsstellen (Routern)
- verbindungslos oder verbindungsorientiert?



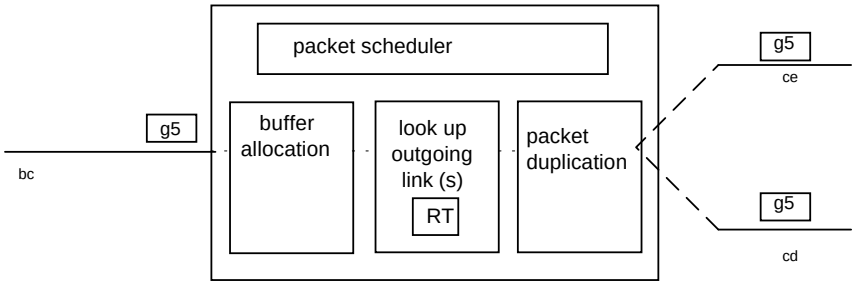
erweiterter Router

Router mit Multicast-Erweiterung

RT Routing Table

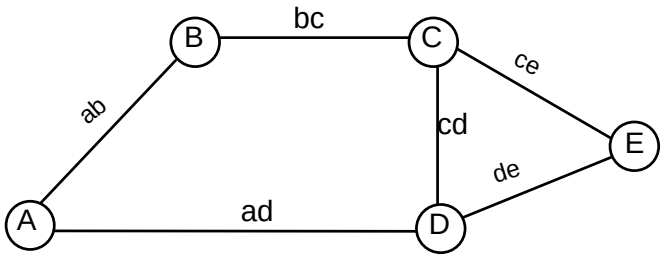
From C to	link	cost
g5	{ce, cd}	

Router at node C

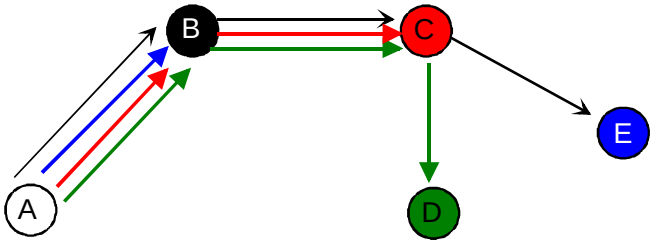


g5 = IP address of group 5

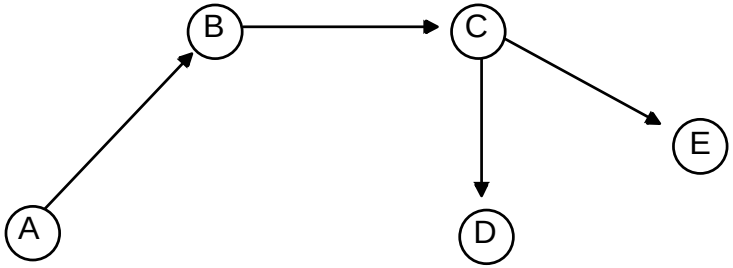
Beispiel-Topologie



Multicast in unserem Beispiel



(a) vier einzelne Verbindungen



(b) eine Multicast-Verbindung

Routing-Algorithmen für Multicast

Multicast Routing ist bisher nur im Internet in Schicht 3 realisiert worden (Multicast-IP). Die eingesetzten Algorithmen sind Erweiterungen der klassischen Routing-Algorithmen; sie sind mit diesen kompatibel.

Multicast im Internet ist empfängerorientiert. Für eine Multicast Session wird zunächst eine IP-Gruppenadresse vereinbart. Der Sender beginnt, an diese Adresse zu senden. Jeder Knoten im Internet kann entscheiden, ob er in eine existierende Gruppe aufgenommen werden möchte.

Reverse Path Broadcasting

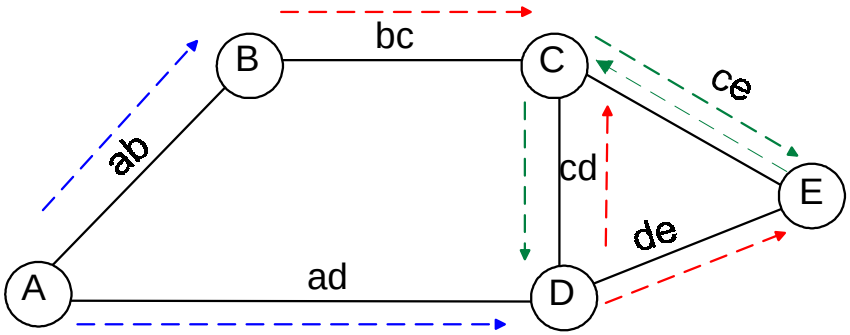
Eine einfache Möglichkeit zur Realisierung von Multicast wäre die Verwendung von **Flooding** als Routing-Algorithmus; das Verfahren wird hier auch als Broadcasting bezeichnet.

Effizienter ist der Reverse Path Broadcasting-Algorithmus (RPB). Er nutzt die Tatsache aus, daß jeder Knoten seinen kürzesten Pfad zum Sender aus der Routing Tabelle kennt! Man bezeichnet diese Pfade als **Reverse Paths**.

Die Idee ist nun, daß ein Knoten nur diejenigen Pakete an seine Nachbarn weitergibt, die auf dem kürzesten Pfad vom Sender angekommen sind. Dieses Verfahren generiert wesentlich weniger Pakete als reines Broadcasting.

Beispiel für Reverse Path Broadcasting
(noch unvollständig)

Für unsere Beispieltopologie arbeitet der (bisher noch unvollständige) RPB-Algorithmus wie folgt:

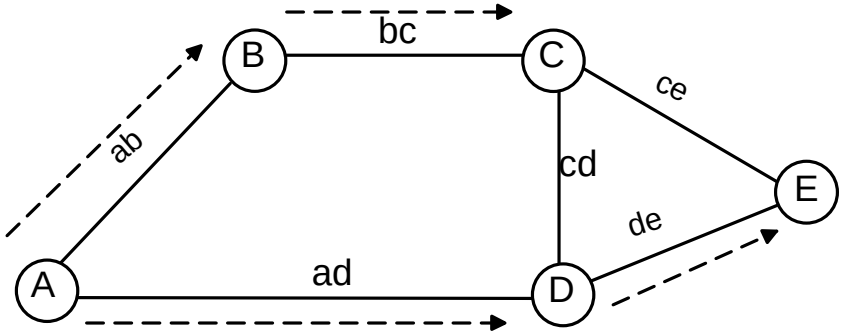


Wie wir sehen entstehen noch immer überflüssige Pakete: die Knoten D und E erhalten jedes Paket zweimal, Knoten C sogar dreimal.

Beispiel für Reverse Path Broadcasting
(vollständiger Algorithmus)

Wenn jeder Knoten seinen Nachbarn etwas Zusatzinformation mitteilt, kann RPB weitere überflüssige Pakete verhindern. Die Zusatzinformation besteht in der Benennung des eigenen kürzesten Pfades zum Sender. In unserem Beispiel informiert E seine Nachbarn C und D darüber, daß *de* auf seinem kürzesten Pfad zu A liegt.

Den Paketfluß für den vollständigen RPB-Algorithmus zeigt die untenstehende Abbildung.



Truncated Reverse Path Broadcasting (TRPB)

Beschränkt die Auslieferung der Daten auf diejenigen Subnetzwerke, die Gruppenmitglieder enthalten. Als Subnetzwerke werden nur LANs betrachtet, die an Blättern des Routing-Baumes hängen.

Dazu wurde ein einfaches Protokoll definiert, mit dem Router die Hosts in ihrem LAN befragen können, ob sie an den Paketen einer bestimmten Gruppe interessiert sind (IGMP: Internet Group Management Protocol). Wenn ein Router in seinem LAN keinen interessierten Host vorfindet, wird er in Zukunft Pakete mit dieser Gruppenadresse nicht mehr auf sein LAN geben.

Vorteil

- Vermeidet überflüssige Pakete in den Blatt-LANs

Nachteil

- Eliminiert nur Subnetzwerke, verringert nicht den Datenverkehr innerhalb des Baumes

Reverse Path Multicasting (RPM)

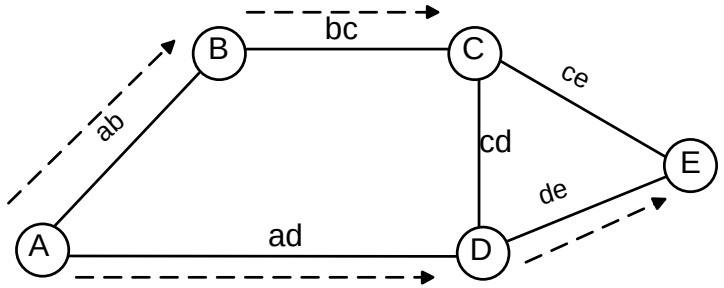
Der TRPB-Algorithmus etabliert Pfade zu allen Routern im Netz, ob sie Mitglied der Gruppe sein wollen oder nicht. Es ist offensichtlich sinnvoll, in der Datenphase einer Session den Routing-Baum so zurückzuschneiden, daß Pakete nur noch dorthin weitergeleitet werden, wo sie wirklich gebraucht werden.

Dies geschieht durch die Generierung von **prune messages**. Diese wandern im Baum von den Blättern zur Wurzel hin und teilen den Knoten der jeweils höheren Ebene mit, daß es weiter unten im Baum keine Empfänger mehr gibt. So wird aus dem Broadcast-Baum ein Multicast-Baum. Das Verfahren wird als Reverse Path Multicasting (RPM) bezeichnet. Im Internet werden die "prune messages" von den Routern generiert und weitergeleitet.

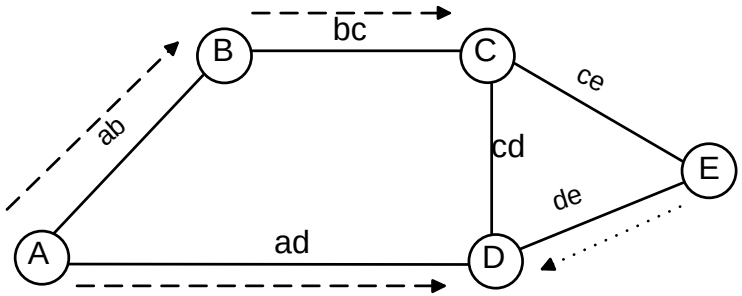
Algorithmus Pruning

- Ein Router, der als Kind-Links nur Blatt-Links ohne Gruppenmitglieder besitzt, sendet einen Non-Membership-Report (NMR) an den übergeordneten Router, d.h. an den vorhergehenden Router im Multi-cast-Baum.
- Router, die von allen untergeordneten Routern NMRs empfangen haben, senden ebenfalls einen NMR an den übergeordneten Router.
- NMRs enthalten eine Zeitangabe, nach der das Pruning wieder aufgehoben werden soll.
- NMRs können auch aufgehoben werden, wenn ein neues Gruppenmitglied an einem Link aktiv wird

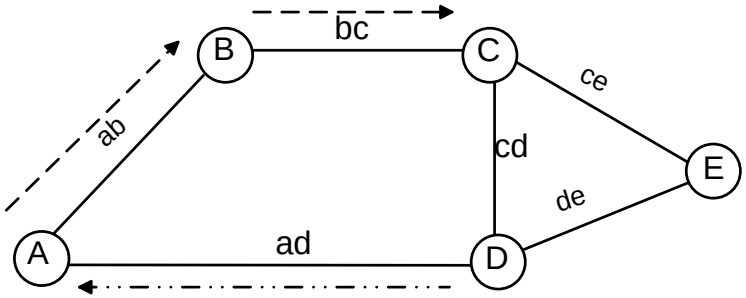
Beispiel für Reverse Path Multicasting



(a) Baum in der anfänglichen RPB Phase



(b) E sendet eine "prune message"



(c) D sendet eine "prune message"

Vor- und Nachteile von RPM

Vorteil

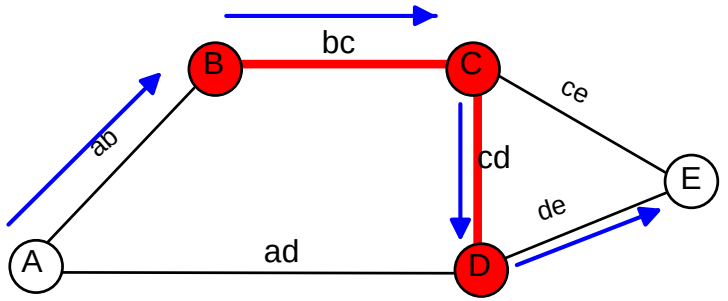
- Reduzierung des Datenverkehrs im Vergleich zu TRPB

Nachteile

- Periodischer Versand der Daten an **alle** Router weiterhin nötig, damit sie „es sich anders überlegen“ können
- Statusinformation in jedem Knoten für jede Gruppe und jeden Sender nötig

Kernbäume (Core-Based Trees)

Alle bisher dargestellten Verfahren haben den Nachteil, daß pro (Sender, Gruppe)-Paar ein eigener Multicast-Baum aufgebaut und verwaltet werden muß. Diesen Nachteil vermeiden die Kernbäume (Core-Based-Trees). Es wird nur ein Baum pro Gruppe eingerichtet. Jeder Sender sendet zum Baum hin. Die Nachrichten werden entlang des Baumes transportiert und erreichen von hier aus die Blätter. Ein Beispiel zeigt die untenstehende Abbildung.



rot = Kernbaum

Multicast-Routing im Internet

Im Internet wird Multicast im **MBone** (Multicast Backbone) erprobt.

Die erste, experimentelle Generation des Multicast-IP-Protokolls verwendete TRPB. Dies führte zu einer erheblichen Mehrbelastung aller IP-Router durch Multicast-Pakete. Später wurde "tree pruning" hinzugefügt, also der Algorithmus RPM implementiert.

Die Knoten informieren einander über ihre kürzesten Pfade zum jeweiligen Sender durch ein modifiziertes Distance-Vector-Routing-Protocol, das als Distance-Vector-Multicast-Routing-Protocol (**DVMRP**) bezeichnet wird. Dieses Protokoll ist heute am weitesten verbreitet.

Zur Zeit arbeiten mehrere Forschungsgruppen intensiv an einem neuen Protokoll, das **Protocol-Independent Multicast (PIM)** heißt. Es beruht auf der Idee des Kernbaums (Core-Based-Tree). Es befindet sich zur Zeit in der Erprobung.

QoS-Based Routing

Multicast-Routing für IP ist ein aktuelles Forschungsthema. Noch weitgehend ungelöst ist das Problem eines Routings unter Einbeziehung von Dienstgüteanforderungen (“**QoS-based routing**“).

Multicast-Ausblick: IP Version 6

Die Multicast-Fähigkeit wird in das IP-Protokoll integriert werden.

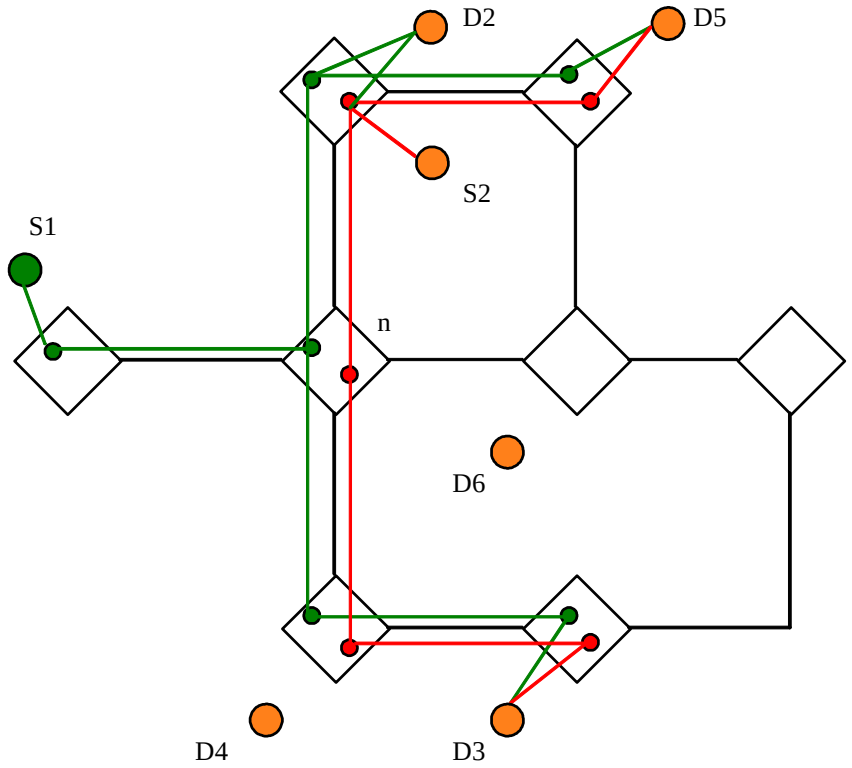
Alle IP-Router werden Gruppenadressen interpretieren können und Multicast-Routing beherrschen.

Das IGMP-Protokoll wird in das klassische Internet Control Message Protocol (ICMP) integriert werden.

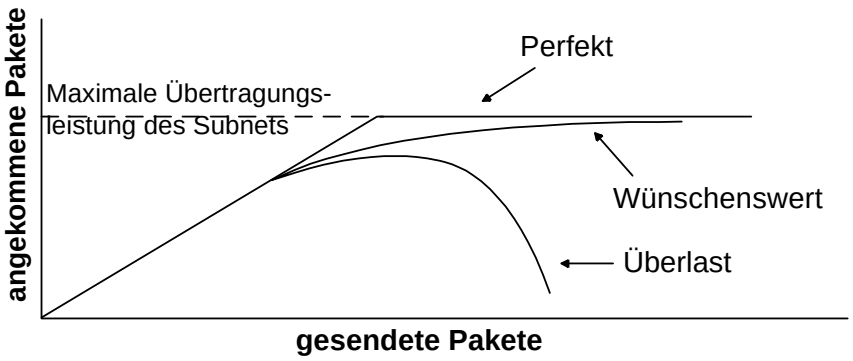
IP-Router werden Pakete nach frei definierbaren Prioritäten schedulen können. Sie werden Paketfilter enthalten, die beispielsweise für Layered Multicast eingesetzt werden können.

Die Unterstützung von QoS durch die Reservierung von Ressourcen in den Routern wird heftig diskutiert, aber es gibt noch keinen Konsens. Die „flow labels“ in den Headern der IP-Pakete ermöglichen zumindest die Zuordnung von Paketen zu einem Strom und die Verwaltung von „soft state“ in den Routern.

Dynamic Join and Leave mit QoS-Garantie



5.5 Überlastkontrolle in der Vermittlungsschicht



Gründe für Überlast

- Knoten zu langsam für Routing-Algorithmen
- Ankommender Verkehr überfordert Ausgangsleitungen

Überlastung tendiert dazu, sich selbst zu verstärken

Beispiel: Knoten wirft wegen Überlastung Paket weg

- Paket muß erneut gesendet werden
  (zusätzlicher Verbrauch an Bandbreite)
- Sender kann Puffer nicht freigeben
  (zusätzliches Binden von Ressourcen)

Besonders kritisch in Datagramm-Netzen!

Strategie 1: Pufferreservierung

Prinzip

- Voraussetzung: Virtuelle Verbindungen
- Reservierung der benötigten Puffer beim Verbindungsaufbau
- Falls nicht genügend Puffer vorhanden:
  - alternativen Pfad wählen, oder
  - Verbindungswunsch abweisen

Beispiel 1:

Bei Verwendung des Stop-and-Wait-Protokolls zur Flußkontrolle: ein Puffer pro Knoten und Verbindung (simplex)

Beispiel 2:

Bei Verwendung des Sliding-Window-Protokolls zur Flußkontrolle

- w Puffer pro Knoten und Simplex-Verbindung
  (w = Fenstergröße)

S1: Pufferreservierung

Eigenschaften

- Keine Überlastung möglich

aber

- die Puffer bleiben verbindungsbezogen reserviert, auch wenn zeitweise keine Daten übertragen werden.

Daher meist nur bei Anwendungen eingesetzt, wo garantierte geringe Verzögerung und hohe Bandbreite erforderlich sind, z.B. bei der digitalen Sprachübertragung über paketvermittelte Netze.

Strategie 2: Wegwerfen von Paketen

Prinzip

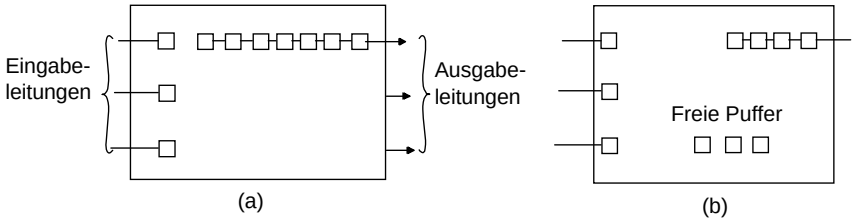
- Keine Reservierung von Ressourcen
- Ankommendes Paket wird weggeworfen, wenn es nicht gepuffert werden kann

Datagramm: Keine weiteren Vorkehrungen notwendig

Verbindungsorientierter, zuverlässiger Dienst: Puffern jedes Pakets beim Sender, bis der Empfang quittiert ist

S2: Wegwerfen von Paketen

Maximale Anzahl von Puffern pro Ausgangsleitung



Eine "unfaire" Beeinträchtigung fremder Paketströme kann dadurch verringert werden, daß für die Paketanzahl in der Ausgabeschlange einer Ausgangsleitung eine Obergrenze definiert wird.

Aber dann: Verwerfen von Paketen trotz freier Puffer möglich

S2: Wegwerfen von Paketen

Eigenschaften

- sehr einfach

aber

- wiederholt übertragene Pakete verschwenden Bandbreite

Paket muß 1 / (1 - p) mal gesendet werden, bevor es akzeptiert wird (p = Wahrscheinlichkeit, daß Paket verworfen wird)

Kleine, einfache Optimierung :
Zunächst Wegwerfen von Paketen, die noch nicht weit gekommen sind (Streckenzähler auswerten)

Strategie 3: Isarithmische Überlastkontrolle

Prinzip

Begrenzung der Anzahl von Paketen im Netz durch Vergabe von "Permits"

- Menge von "Permits" im Netz
- Zum Senden wird "Permit" benötigt
  - Senden: „Permit" wird zerstört
  - Empfangen: „Permit“ wird generiert

Probleme

- Teile des Netzes können überlastet werden, während andere Teile unterbelastet sind
- Gleichmäßige Verteilung der Permits schwierig
- Zusätzliche Bandbreite für Permit-Transfer
- Schlecht bei Übertragung großer Datenmengen (z.B. Dateitransfer)
- Endgültiger Verlust von Permits durch Fehler im Netz schwer zu erkennen

Strategie 4: Flußkontrolle mißbrauchen

Prinzip

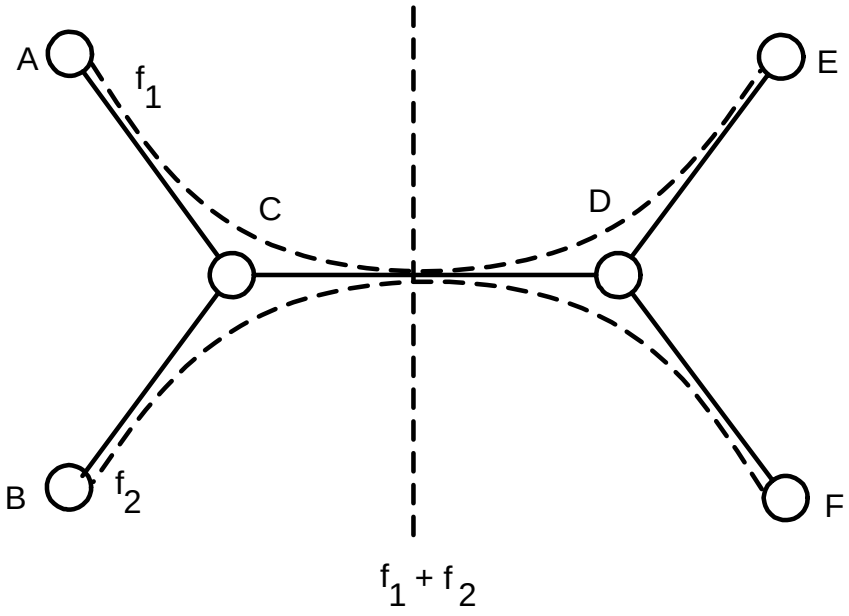
Flußkontrolle zur Überlastvermeidung "mißbrauchen"

- Flußkontrolle eigentlich definiert zwischen Paaren von Endsystemen
- Das Netz darf aber nun in den inneren Knoten **von sich aus** die Fenstergröße des Sliding-Window-Protokolls verändern
- Implementiert zum Beispiel in Schicht 3 von SNA (IBM)

Nachteile der Überlastkontrolle durch Flußkontrolle

Im Sinne der Architektur unsauber, wenn die Flußkontrolle in Schicht 4 gemacht wird. Denn Schicht 3 muß nun den Flußkontrollparameter im Paketheader verändern.

Mehrere Paketflüsse führen über einen gemeinsamen Link der Schicht 3. Wie kann die Flußreduzierung **fair** erfolgen?



Strategie 5: "Choke"-Pakete

Prinzip

Netzmanagement-Pakete drosseln den Verkehr bei Überlast

- Jede Ausgangsleitung eines Routers ist mit einer Variablen $u (0 \le u \le 1)$ versehen, die die aktuelle Auslastung angibt
- *u* > Grenzwert: Leitung geht in den Zustand "Warnung"
- Wenn die Ausgangsleitung für ein Paket im Zustand "Warnung" ist, sendet der Router für jedes eintreffende Paket ein "Choke"-Paket an die Quelle
- Quelle empfängt Choke-Paket: Reduzierung des Datenverkehrs zu dem betreffenden Ziel

Variante

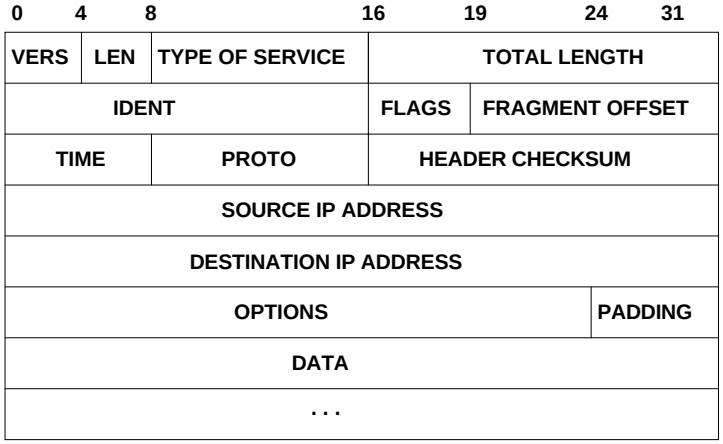
Es gibt mehrere Grenzwerte für *u*, die zu unterschiedlich harten Warnungen führen und den Sender zu unterschiedlichen Reduzierungen des Datenstroms veranlassen.

5.6 Beispiele: IP, X.25, ATM

5.6.1 IP (Internet Protocol)

- Ein Datagramm-Protokoll (verbindungslos)
- Ein Host-zu-Host-Protokoll
- Handhabt die Fragmentierung großer Pakete: große Dienst-Datagramme können in kleinere Protokoll-Datagramme „fragmentiert“ werden.

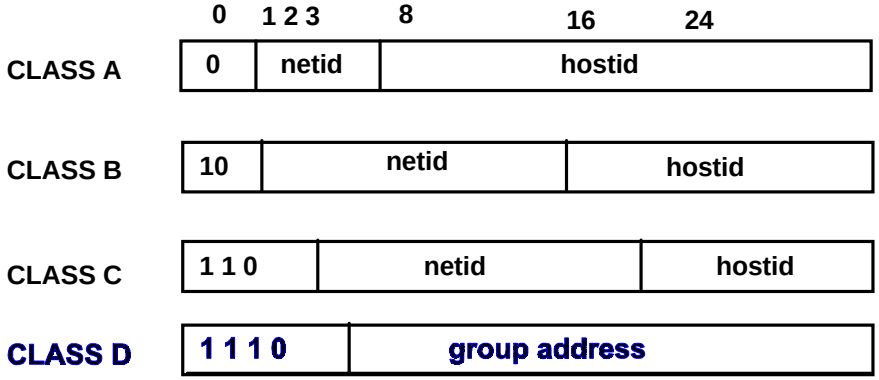
Format von IP-Datagrammen



VERS	Protokollversion
LEN	Länge des Headers (Wörter)
TYPE OF SERVICE	QoS (Priorität und D/T/R)
TOTAL LENGTH	Länge incl. Daten in Bytes
IDENT	Identität des Datagramms
FLAGS	"nicht fragmentieren/letztes Fragment"
FRAGMENT OFFSET	Offset dieses Teils
TIME	Lebensdauer in Sekunden ("time to live")
PROTO	Type des höheren Protokolls
HEADER CHECKSUM	EXOR der Header-Wörter
SOURCE ADDRESS	IP-Adresse des Quell-Hosts
DEST ADDRESS	IP-Adresse des Ziel-Hosts
OPTIONS	Kommandocode für Netzmanagementdatagramme
PADDING	Auffüllen auf Wortgrenze
DATA	Nutzdatenfeld

Adressierung im Internet

Die IP-Adresse ist eine hierarchische Adresse mit Netz- und Hostidentifikationsnummer (netid und hostid). Es gibt drei Formate für Subnetze unterschiedlicher Größe sowie ein Format für Multicast:



Gebräuchlich ist seltsamerweise eine dezimale Schreibweise mit einer Zahl pro Byte. Beispiel:

10.0.0.0	für Arpanet
128.10.0.0	für ein großes Ethernet-LAN
192.5.48.0	für ein kleines Ring-LAN

(hostid = 0 bezeichnet ein Netz aus einem Host)

Beispiel: Uni Mannheim (Auszug)

```
127.0.0.1 localhost
134.155.48.96 pi4 pi4.informatik.uni-mannheim.de
pi4d01
#
# Host Database
#
# If the yellow pages is running, this file is only
consulted when booting
#
# These lines added by the Sun Setup Program from ser-
ver pi3s01
#
#  Einteilung Fakultaet Mathematik + Informatik Subnet
134.155.48.xx
#  LS   I          0 - 1f   pi1
#  LS  II         20 - 3f   pi2
#  LS III         40 - 5f   pi3
#  LS  IV         60 - 7f   pi4
#  POOL + Fak.    80 - 9f   fmi
#
134.155.48.109     herodot  pi4r01  rs6000  rs6000-320
134.155.48.110     pi4t01   ts1   eps-4 lantronix  #
terminal-server PI IV
134.155.48.111     pi4t02   ts2
134.155.48.112     pi4p01
134.155.48.113     pi4p02
134.155.48.114     pi4p03   thales
134.155.48.115     pi4p04   euklid
134.155.48.116     pi4p05
134.155.48.117     pi4p06   archimedes
134.155.48.118     pi4p07   diogenes
134.155.48.119     pi4p08   pythagoras
```

Adreßauflösung im LAN

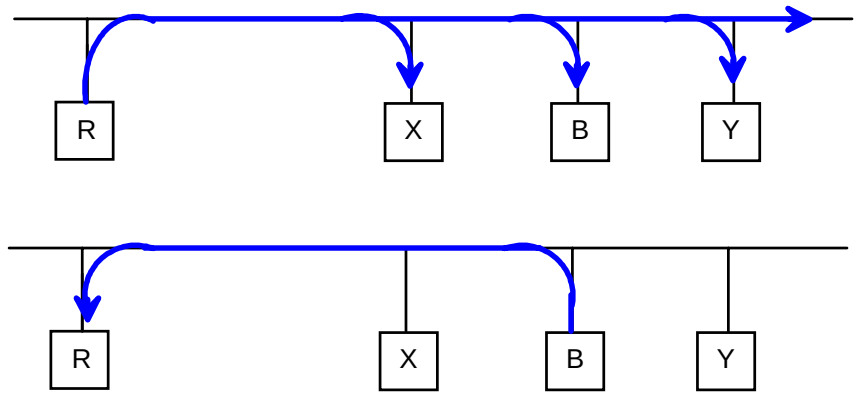
Problem:
Wie soll die Abbildung der Internet-Adresse (IP-Adresse) eines Rechners auf die physikalische Stationsadresse im LAN (IEEE 802-Adresse) erfolgen?

1) Wenn die physikalische Adresse (Stationsadresse) durch den Benutzer ausgewählt werden kann, wähle für den Hostid-Teil der INTERNET-Adresse die physikalische Adresse
2) Wenn die physikalische Adresse vorkonfiguriert („fest verdrahtet“) ist, unterhalte eine Abbildungstabelle (z.B. im Router) und/oder benütze das **Address Resolution Protocol ARP**

Address Resolution Protocol ARP

Protokoll im Router

1) Sende mittels Broadcast auf dem LAN ein ARP-Request-Paket, welches die physikalische und die Internet-Adresse des Senders und die Internet-Adresse des gesuchten Empfängers enthält.
2) Warte auf die Antwort des Empfängers durch ein ARP-Reply-Paket, welches seine physikalische Adresse enthält.
3) Unterhalte einen Cache aus (I,P)-Adreßpaaren für spätere Anfragen.
4) **Verbesserung:** Der Empfänger des ARP-Requests speichert das (I,P)-Paar des Senders auch in seinem Cache.



5.6.2 X.25

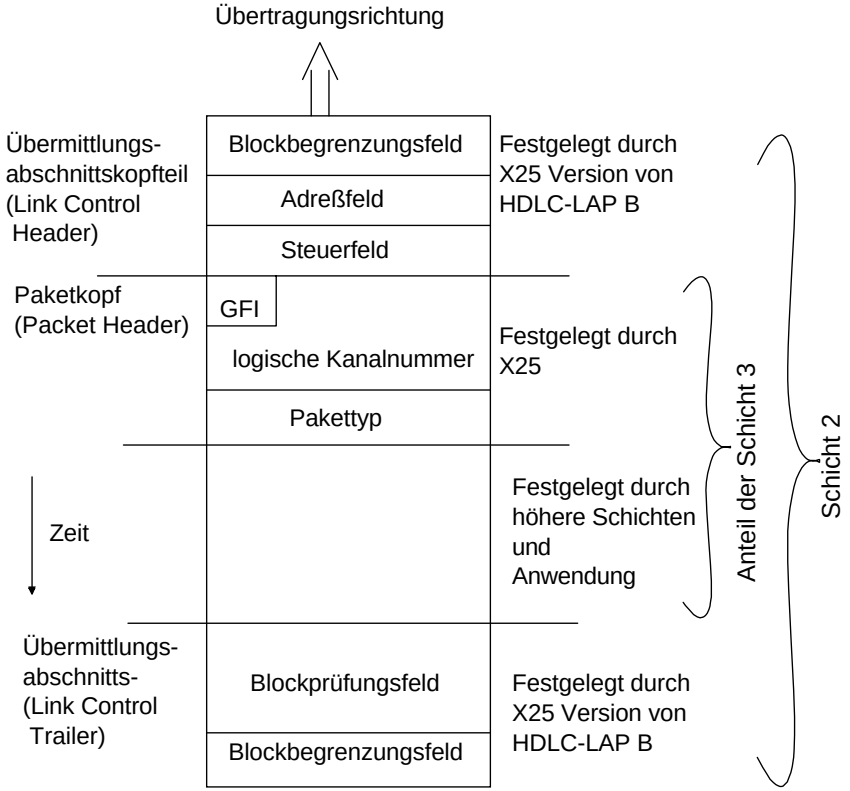
Der wichtigste internationale Standard für paketvermittelte Weitverkehrsnetze.

Ein Standard des CCITT (jetzt ITU-T).

- Einsatz zur Vermittlung in **öffentlichen** Paketvermittlungsnetzen
- Umfaßt die Schichten 1 bis 3 des Referenzmodells
- Blockbegrenzung: Synchronisation (Wiederaufsetzen des Empfängers)
- Adreßfeld: Ziel-/Herkunftsadresse des Pakets
- Steuerfeld: Unterscheidung Daten-/Steuerpakete; Sequenznummern für Paketreihenfolge
- GFI: Kennung des Paketformats (General Format Identifier)
- Logische Kanalnummer: Unterscheidung verschiedener Verbindungen an einem Zugangspunkt
- Pakettypen: Auf-/Abbau der Verbindung, Daten, Interrupts, Flußsteuerung

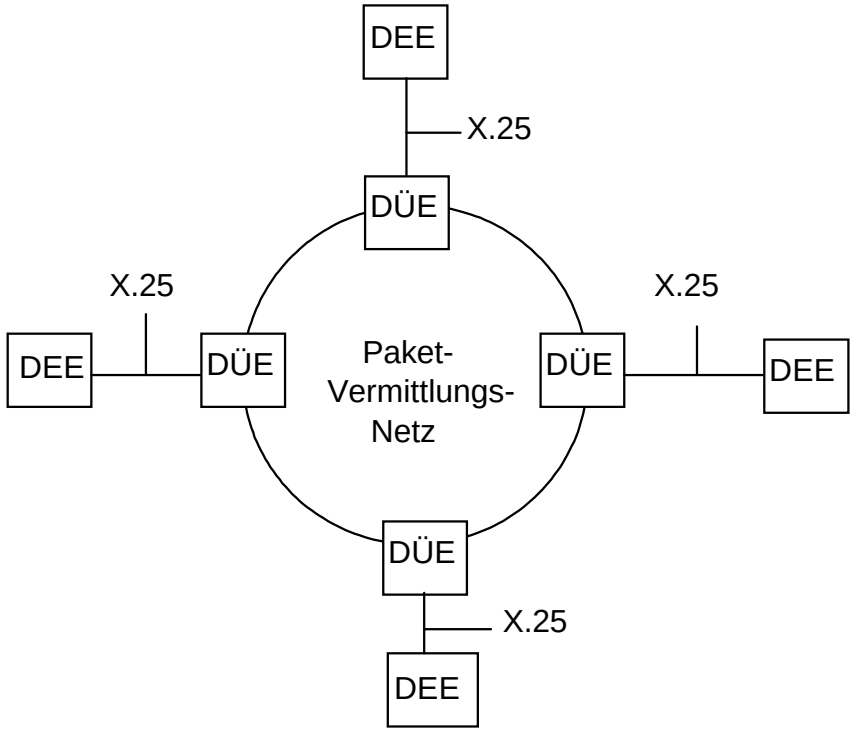
Paketstruktur von X.25

Beispiel: Datenpaket

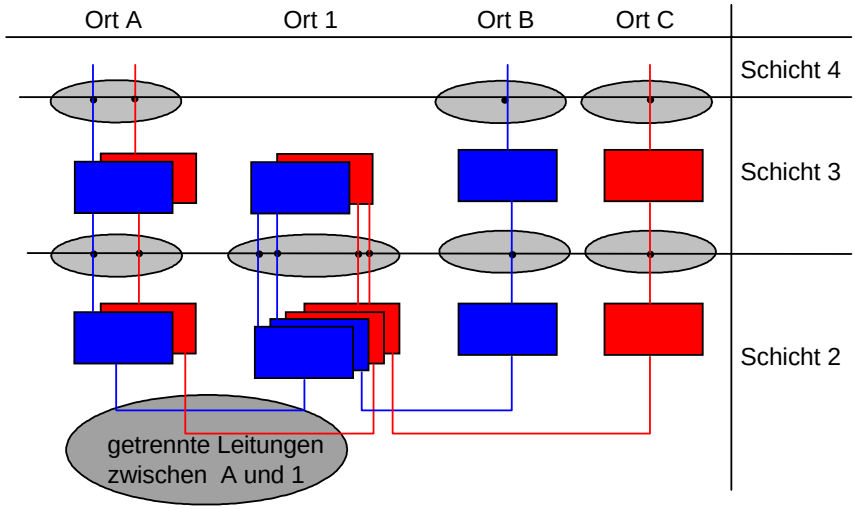


Einordnung von X.25

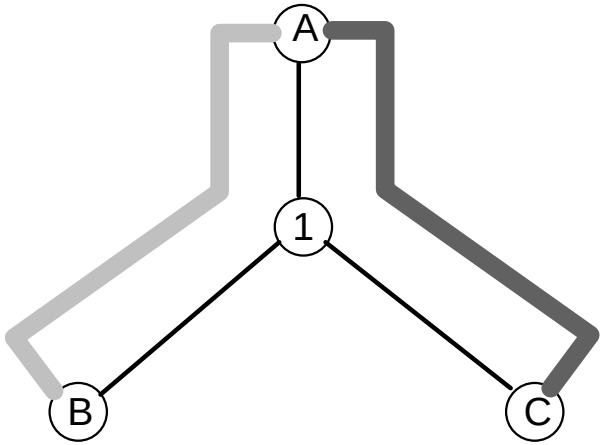
Standardisierung einer **Schnittstelle** zwischen einem privaten Endgerät (DEE=Datenendeinrichtung) und dem öffentlichen Paketvermittlungsnetz (DÜE=Datenübertragungseinrichtung).



Beispiel: Vermittlung ohne Multiplexen



Multiplexen: Zwei Schicht 3-Verbindungen über eine Schicht 2-Verbindung

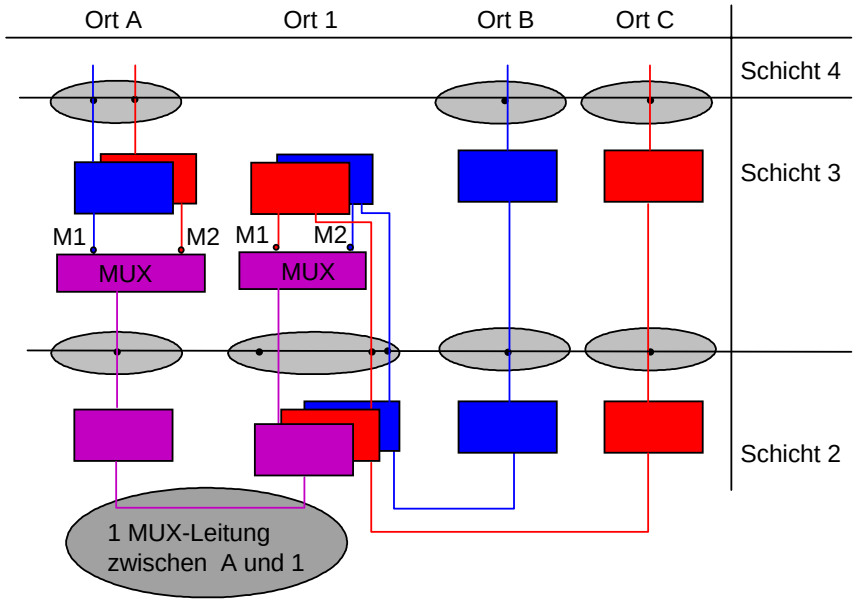


Schicht 1- und Schicht 2-Verbindungen: A1, B1, C1

Schicht 3-Verbindung AC

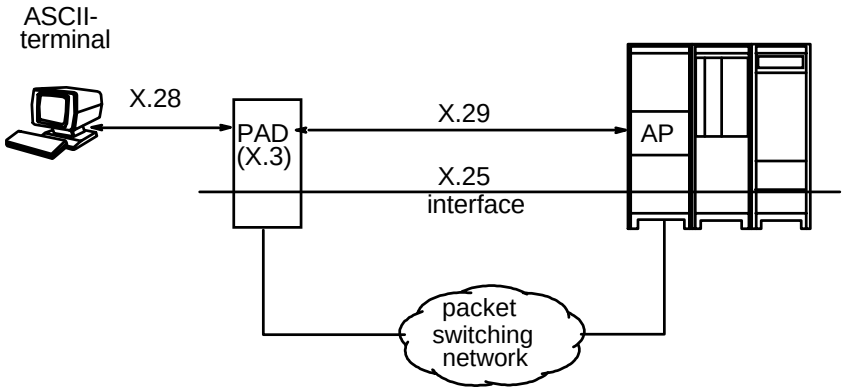
Schicht 3-Verbindung AB

Beispiel: Vermittlung mit Multiplexen



PAD (Packet Assembly/Disassembly)

- Übertragung von zeichenorientierten Terminal-Datenströmen über X.25
- CCITT Standards X.3 / X.28 / X.29 ("Triple X")



PAD = Packet Assembly/Disassembly Facility
AP = Application Program

5.6.3 ATM (Asynchronous Transfer Mode)

Grundlagen

- Eine schnelle Paketvermittlungstechnik für Zellen fester Größe
- Basiert auf asynchronem (statistischem) Zeitmultiplexing; daher der Name ATM
- Verbindungsorientiert; unterscheidet virtuelle Pfade und virtuelle Verbindungen
- Implementierung der Vermittlungsrechner soll zwecks Erreichung hoher Zellraten möglichst weitgehend in Hardware möglich sein
- Verzicht auf Fehlererkennung, Flußkontrolle usw. in der Zellvermittlungsschicht
- Soll ein breites Spektrum verschiedener Datenraten und ein breites Spektrum verschiedener Anwendungsanforderungen befriedigen

Übertragungsmodi

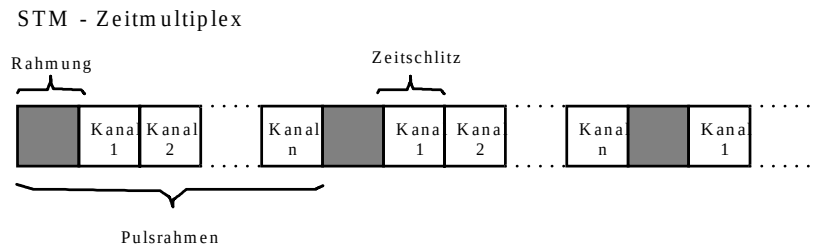
STM (Synchronous Transfer Mode)

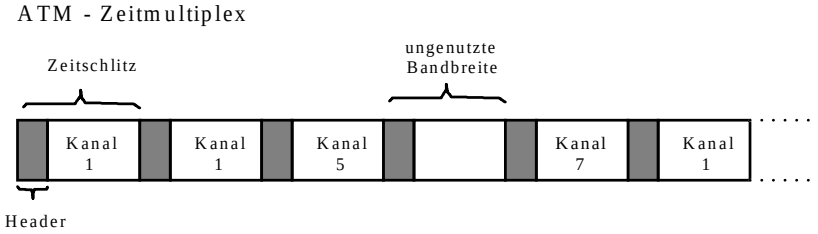
- Multiplexen von physikalischen Verbindungen mit synchronem Zeitmultiplexing
- Prinzip der Kanalvermittlung (z.B. mit dem "*Time Slot Interchange"*-Verfahren*)*
- Gut für Verkehrsströme mit konstanten Bitraten
- Höhere Bitraten durch Schalten von mehreren parallelen Kanälen möglich

ATM (Asynchronous Transfer Mode)

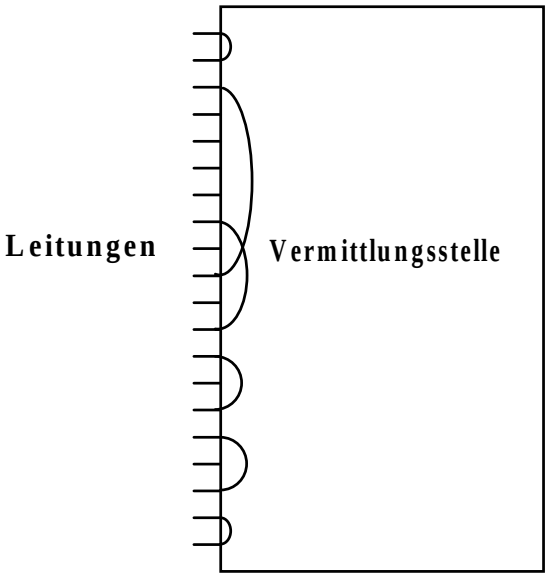
- Multiplexen physikalischer Verbindungen mit asynchronem Zeitmultiplexing
- Paketweise Vermittlung von Zellen fester Größe *("Schnelle Paketvermittlung")*
- "Leichtgewichtige" Protokolle: Keine Flußkontrolle, keine Fehlerkorrektur
- Integration von Verkehren mit verschiedensten Zellraten möglich

Synchrones und asynchrones Zeitmultiplexing





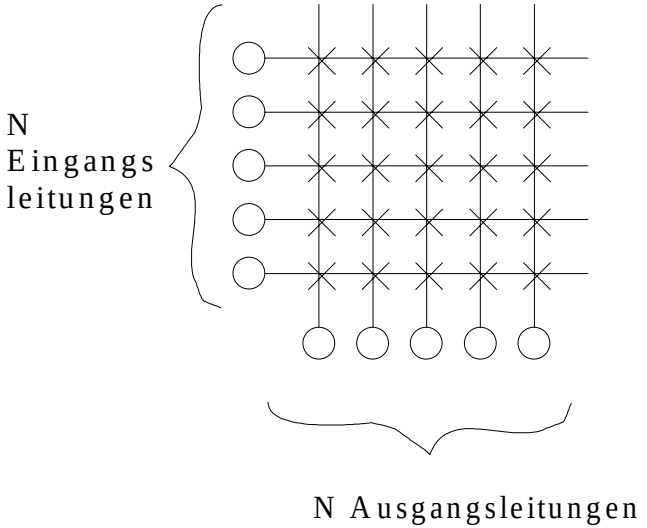
Vermittlungstechnik



Funktion einer Vermittlungsstelle (Switch), abstrakt

Raumvielfache (Space-Division Switch)

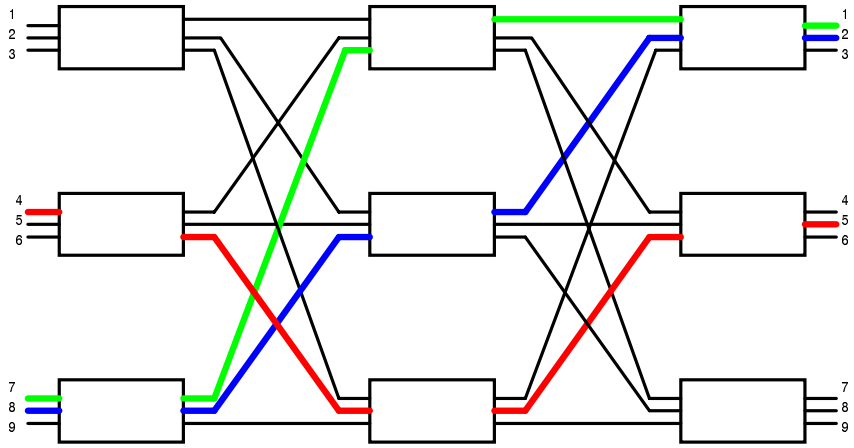
Prinzip



Nachteile einer Implementierung als Matrix

- Anzahl der Verbindungspunkte (crosspoints) wächst mit N^2
- Defekter Verbindungspunkt macht eine bestimmte Verbindung unmöglich
- Schlechte Auslastung der Verbindungspunkte (maximal N aus N^2 in Gebrauch)

Mehrstufige Raumvielfache (multi-stage space division switches)



Vorteile

- Geringere Zahl an Verbindungspunkten
- Mehrere alternative Pfade zur Verbindung eines Eingangs mit einem Ausgang; dadurch höhere Zuverlässigkeit

Nachteile

- Blockierung: keine Verbindungsmöglichkeit zwischen Eingang und Ausgang. Im obigen Beispiel: Eingang 9 kann mit Ausgang 4 oder 6 nicht verbunden werden!

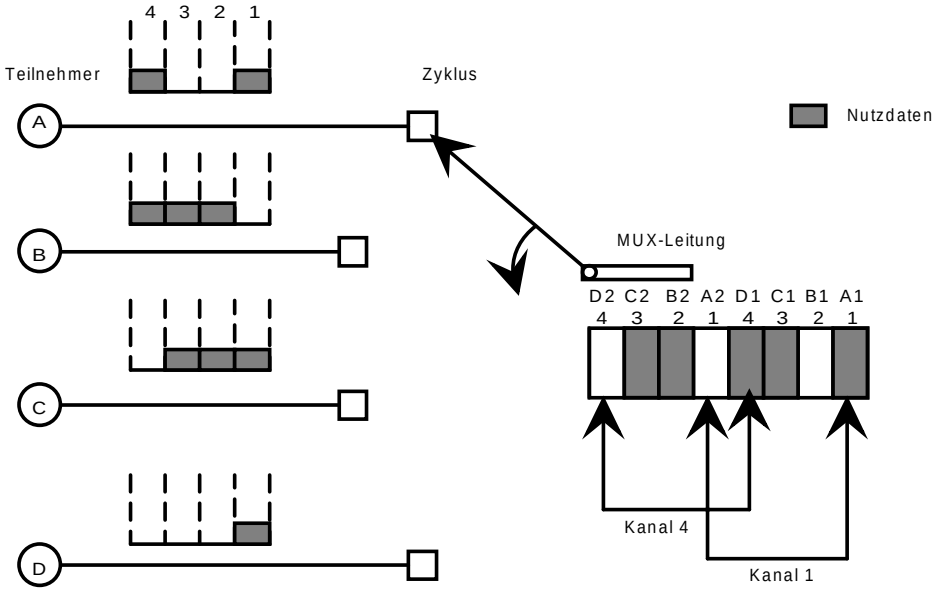
Multiplexing auf der Leitung und Vermittlungstechnik

Man kann konzeptionell unterscheiden:

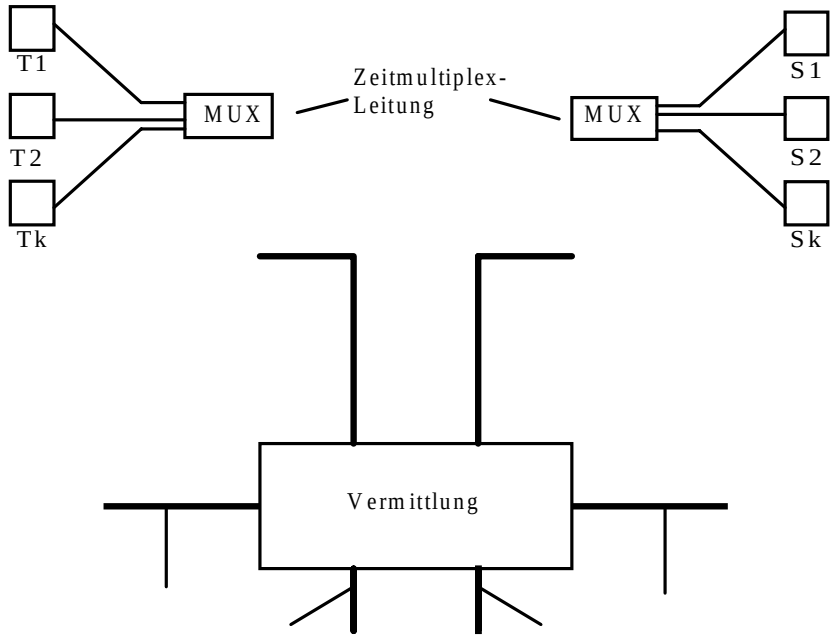
- Synchrones und asynchrones Multiplexen auf der Leitung
- Synchrone und asynchrone Vermittlungstechnik

Synchrones Zeitmultiplex-Verfahren

Synchronous Time-Division Multiplexing (STD)

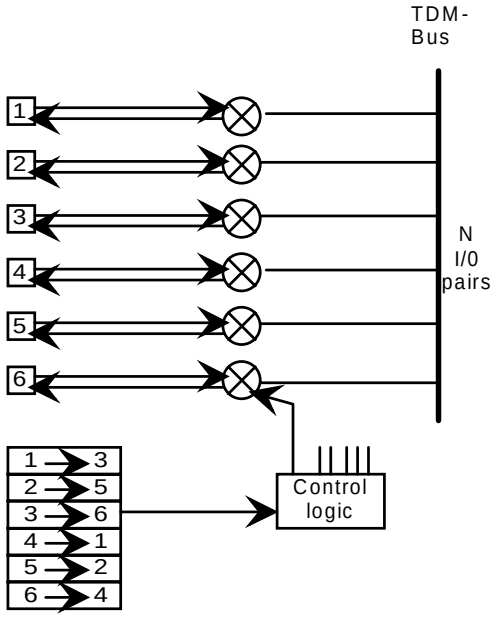


Übertragung und Vermittlung im Zeitmultiplexverfahren



Vermittlungsstelle mit internem TDM-Bus

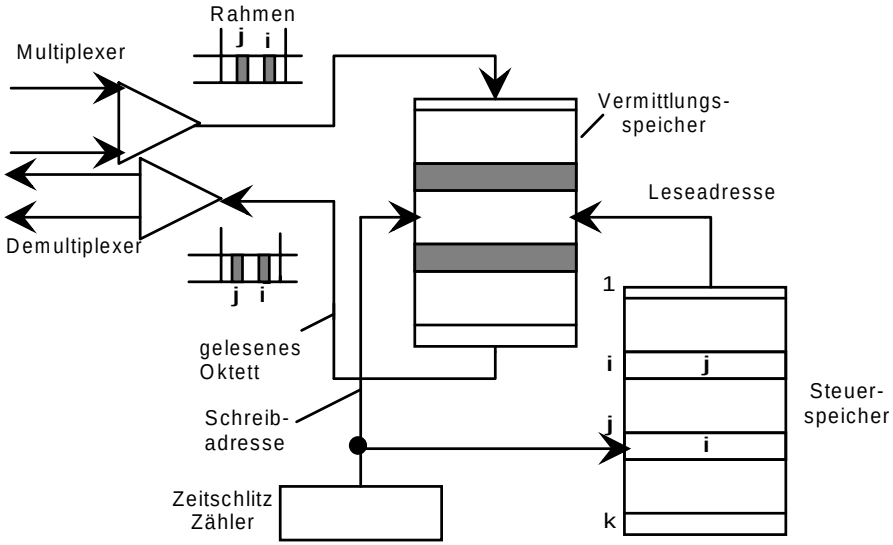
- Verwendung von STM auf einem schnellen Bus innerhalb des Vermittlungsrechners
- Jeweils eine Eingangs- und eine Ausgangsleitung werden für eine kurze Zeitperiode auf den Bus geschaltet
- Leitungspuffer dienen zum Geschwindigkeitsausgleich zwischen langsamen externen Leitungen und dem schnelleren TDM-Bus



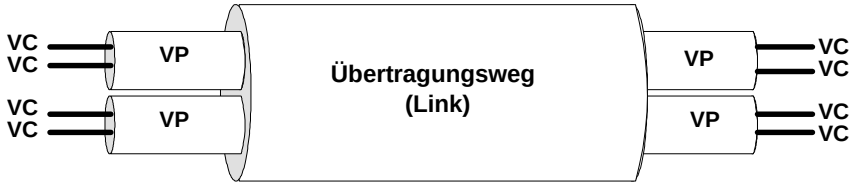
Nachteil:Der interne Bus muß so schnell sein wie die Summe der gleichzeitig aktiven Verbindungen!

Vermittlungsstelle mit internem Vermittlungsspeicher

"Time Slot Interchange"



Virtuelle Kanäle und virtuelle Pfade in ATM



Virtueller Kanal (Virtual Circuit, VC):

- virtuelle Verbindung zwischen ATM-Endgeräten über mehrere Übertragungsabschnitte hinweg

Virtueller Pfad (Virtual Path, VP):

- auf einer (Teil-)Strecke gebündelte VCs

ATM-Zellenformate

UNI

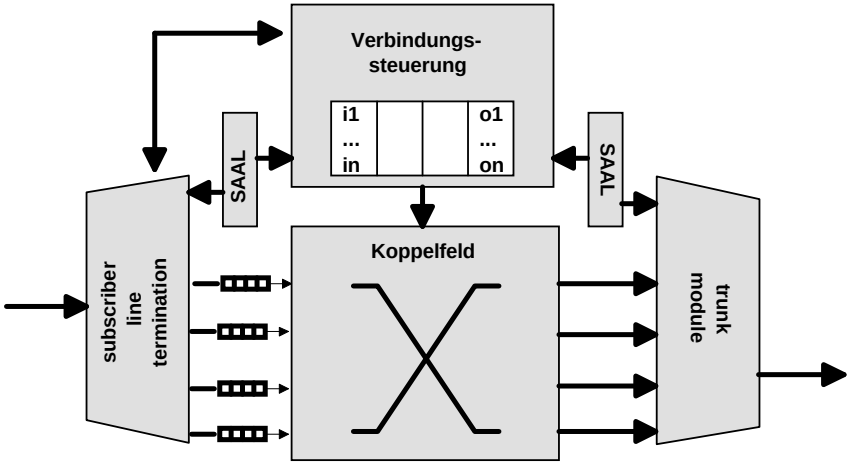
8 7 6 5 4 3 2 1		
GFC	VPI	
VPI	VCI	
VCI		
VCI	PT	CLP
HEC		
Cell Payload 48 Bytes		

NNI

8 7 6 5 4 3 2 1		
VPI		
VPI	VCI	
VCI		
VCI	PT	CLP
HEC		
Cell Payload 48 Bytes		

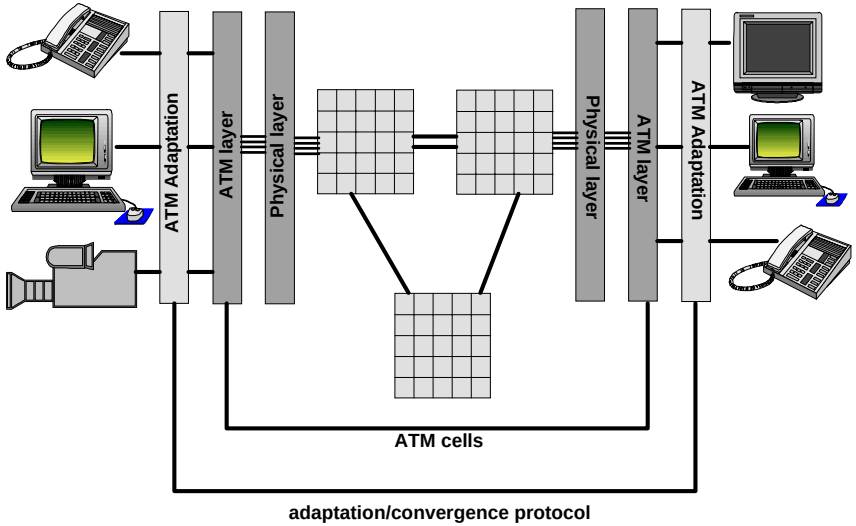
GFC: Generic Fow Control
VPI: Virtual Path Identifier
VCI: Virtual Circuit Identifier
PT: Payload Type
CLP: Cell Loss Priority
HEC: Header Error Check

ATM-Vermittlungsknoten



Schnelle Vermittlung durch "label swapping"

ATM Adaptation-Protokolle



ATM-Dienstklassen

	Klasse A	Klasse B	Klasse C	Klasse D
Synchronität	isochron		asynchron	
Bitrate	konstant	variabel		
Verbindungs-modus	verbindungsorientiert			verbindungslos
Anwendungen	Emulation synchroner Dienste (ISDN)	variabel bitratiges Video (MPEG, ...)	Verbindungs-orietnierte Da-ten kommu-nikation	verbindungs-lose Daten-kommunikaton

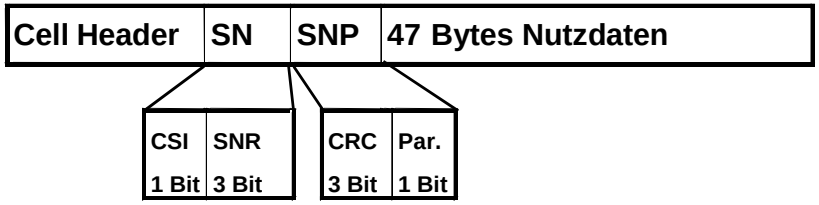
ATM-Adaptionsschichten

- AAL1: Constant Bit Rate (CBR) mit Synchronisation
- AAL2: Variable Bit Rate (VBR)
- AAL3/4: für Datenverkehr, überwiegend in öffentlichen Netzen
- AAL5: Standard-AAL für Datenverkehr

AALs beschreiben Ende-zu-Ende-Protokolle. Weitere AALs können definiert werden, ohne daß dies die ATM-Zellvermittlungsschicht betrifft.

AAL 1

- Transport von CBR-Verkehr
- Sicherung gegen Zellverluste durch Sequenznummern



SN= Sequenz Number Field **SNP= Sequence Number Protection**

CSI= Convergence Sublayer Indication Bit **SNR= Sequence Number**

CRC= Cyclic Redundancy Check (X^3+X+1) **Par.= Parity Bit für CRC Feld**

- Beispiel: ISDN Audio-Kanal:
  - jede Zelle enthält 47 "samples" aus dem PCM-Bitstrom des Sprachkanals. 6 ms Verzögerung durch das Paketieren
  - bei Sprachpausen können die Zellen unterdrückt und die entsprechende Bandbreite eingespart werden

AAL 3/4

Ursprünglicher von der ITU definierter AAL für Datenverkehr. Daher überwiegend in öffentlichen Netzen verwendet, (z.B. im ATM-Pilotnetz der Deutschen Telekom).

Anzahl Bits:	2	4	10		6	10
Cell Header	ST	SNR	MID	44 Oktetts Nutzdaten	LEN	CRC

St: Segment Type LEN: Nutzdatenlänge

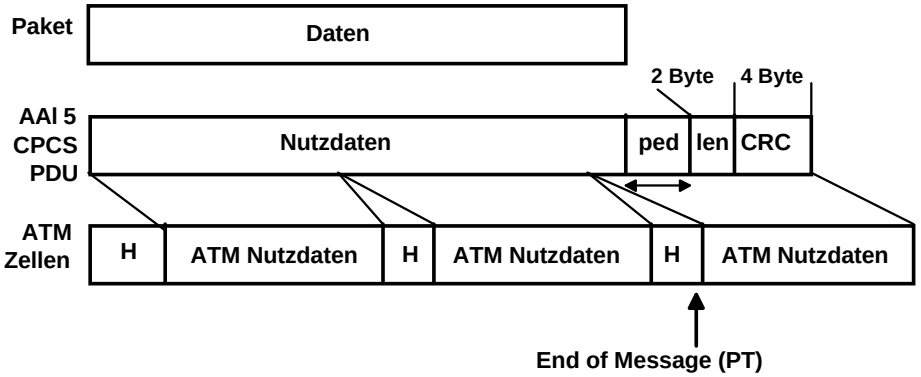
SNR: Sequenznummer CRC: Fehlerschutz

MID: Message ID

- AAL3/4 Overhead etwa 17% (pro Zelle 5+4 Oktetts)
- Fehlerschutz pro ATM-Zelle (bei AAL5 pro CPCS-Nachricht)
- Das MID-Feld erlaubt das „Interleaving“ mehrerer Nachrichten
- Dadurch können „Connection-Less-Server“ Nachrichten weiterleiten, ohne sie vorher zu reassemblieren.

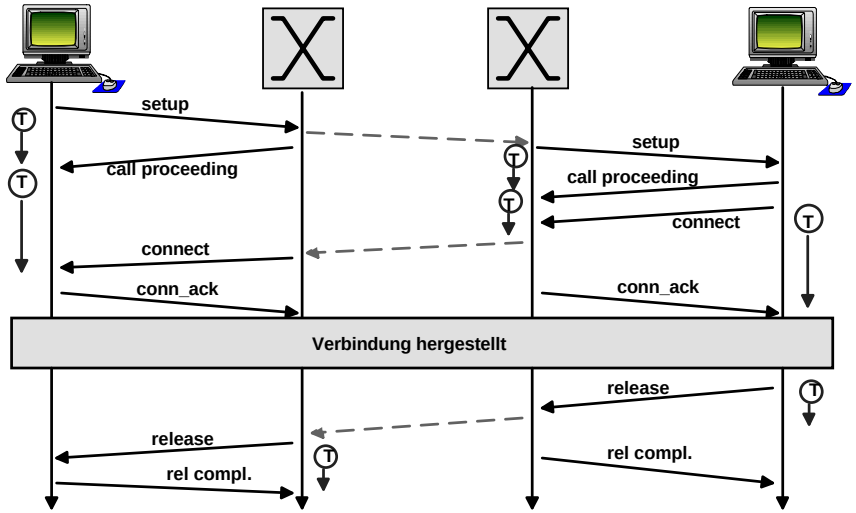
AAL 5

- Adaptionsschicht, ursprünglich für Datenverkehr gedacht, inzwischen zum Beispiel auch verwendet für VBR Video (MPEG II)
- Ursprünglich als SEAL (Simple Efficient Adaptation Layer) im ATM Forum entstanden, war zunächst kein ITU Standard)



Verbindungsverwaltung

Timer-basierte Verbindungsverwaltung



ATM-Verkehrsklassen

UBR: Unspecified Bitrate

- Für Datenanwendungen, nutzt verfügbare (Rest-) Bandbreite
- Keine „admission control“ und kein „Policing“
- Bei Überlast hohe Zellverluste

CBR: Constant Bitrate

- Für „Circuit Emulation Services“ mit festen PCR, CTD, CDV
- Minimale Zellverlustrate

VBR: Varable Bitrate

- Für variabel bitratig komprimierte Videoströme
- VBR und VBR-RT je nach Synchronitätsanforderung

ABR: Available Bitrate

- Zuverlässige Übertragung für Datenanwendungen
- z.B. TCP-Durchsatz schwankt stark bei häufigen Zellverlusten
- realisiert eine Flußregelung im ATM-Netz

ATM-Verkehrsvertrag

- Benutzer und Netz schließen beim Verbindungsaufbau einen *„Verkehrsvertrag“*
- Der Benutzer liefert nur den beim Verbindungsaufbau spezifizierten Verkehr (*Verkehrsbeschreibung, traffic description*)
- Der Benutzer spezifiziert die für diese Verkehrsbeschreibung gewünschten Dienstqualitäten (*QoS: Quality of Service*)
- Das Netz prüft, ob der spezifizierte Verkehr mit der gewünschten Qualität noch transportiert werden kann (*admission control*)
- Das Netz kontrolliert während der Verbindung die Einhaltung der Verkehrsbeschreibung am Netzzugang (*UPC: usage parameter control* oder *source policing*)
- Nicht konforme Zellen werden:
  - Am Netzeingang mit CLP=1 gekennzeichnet
  - Zellen mit CLP=1 werden bei Überlast am Netzeingang oder im Inneren des Netzes verworfen

Verkehrsparameter

Verkehrsbeschreibung

- **PCR**: Peak Cell Rate (cells/s)
- **SCR**: Sustainable Cell Rate (cells/s)
- **MBS**: Maximum Burst Size (cells), auch spezifiziert als **BT**: Burst Tolerance = (MBS-1)/(1/SCR-1/PCR)
- **MCR**: Minimum Cell Rate (nur für ABR)

Dienstqualitäten (QoS-Parameter)

- CLR: Cell Loss Ratio (Anzahl der verlorenen Zellen/Anzahl der gesendeten Zellen)
- CTD: Cell Transfer Delay (vom Netzzugang bis zur Ablieferung beim Empfänger)
- CDV: Cell Delay Variation (CTD variance) (Delay Jitter)

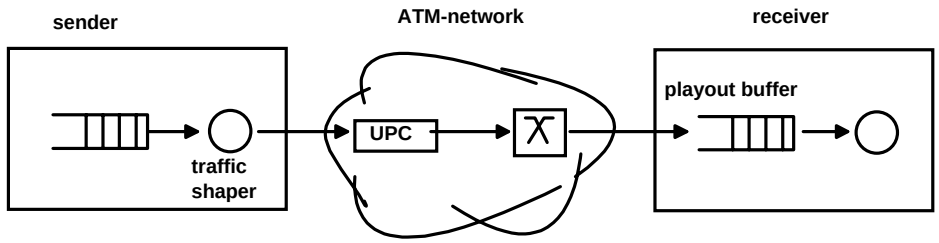
Beim Verbindungsaufbau können

- bestimmte Verkehrsparameter
- bestimmte Verkehrsklassen

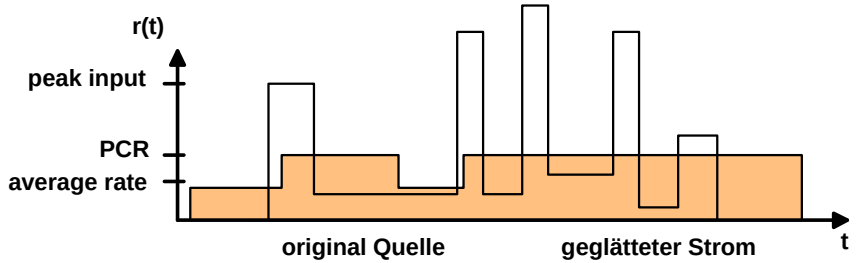
spezifiziert werden.

Traffic Shaping

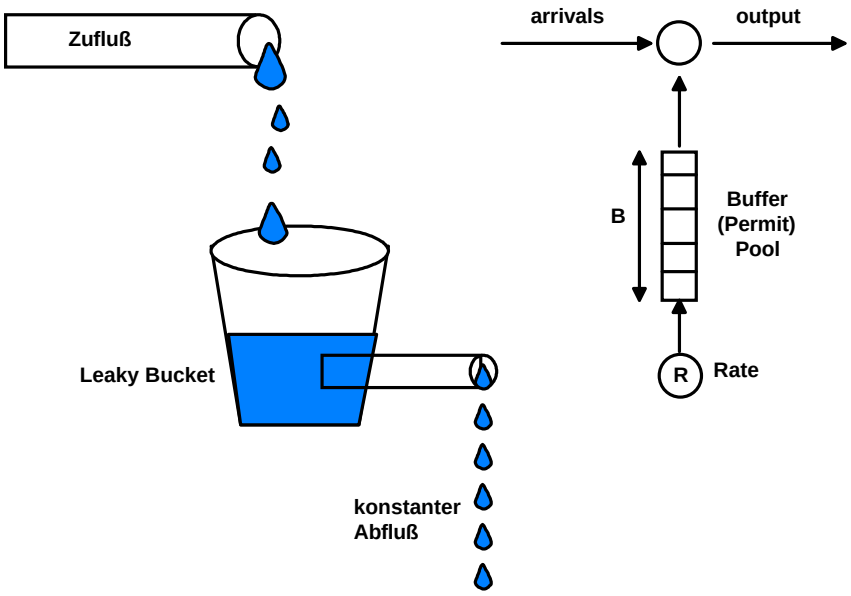
Das Endgerät **formt** den Verkehr, um den Verkehrsvertrag einzuhalten, Trade-Off: Zellverluste vs. Zellverzögerung



„traffic shaper“ glättet stark schwankenden Verkehr



Verkehrsformung durch den Algorithmus "Leaky Bucket"



Bandbreitenaufteilung durch asynchrones Multiplexing