5.4 Wegewahl (Routing) für Multicast-Netze

Definition Multicast

Unter **Multicast** versteht man die Übertragung eines Datenstroms von einem Sender an mehrere Empfänger.

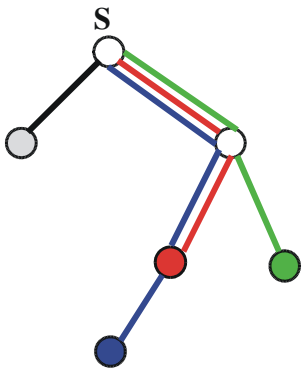
Warum ist Multicast wichtig für Multimedia?

- Multimedia-Anwendungen erfordern oft eine 1:n - Kommunikation. Beispiele:
  - Videokonferenz
  - Tele-Kooperation (CSCW) mit gemeinsamem Arbeitsbereich
  - near-Video-on-Demand
  - Verteil-Kommunikation (Broadcast)
- Digitale Video- und Audioströme haben sehr hohe Datenraten (1,5 MBit/s und mehr). Eine Übertragung über n einzelne Verbindungen würde das Netz überlasten.

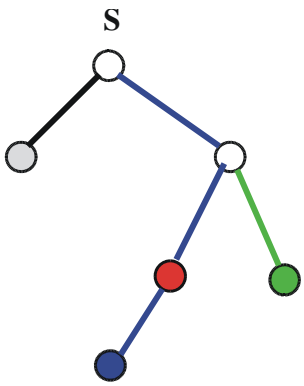
Motivation für Multicast

Mehr „Intelligenz“ im Netz verringert

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



n Ende-zu-Ende-Verbindungen



eine Multicast-Verbindung

Anforderungen an Multicast für Multimedia (1)

- 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"

Anforderungen an Multicast für Multimedia (2)

- Erfordert Formate und Protokolle für eine **Gruppenadressierung**
- Erfordert neue Algorithmen für die **Fehlerkorrektur** (zum Beispiel Forward Error Correction (FEC) oder Reliable Multicast)
- Erfordert Algorithmen für **dynamisches Hinzufügen und Löschen** von Teilnehmern in einer Session

Multicast in LANs

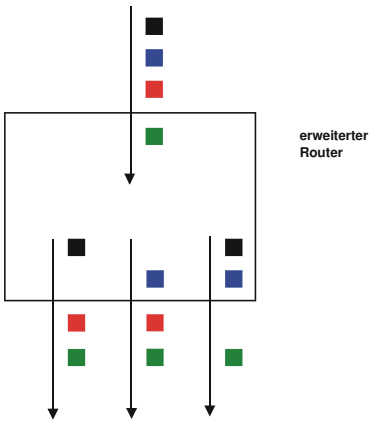
Ethernet, Token Ring, Wireless LAN

- Die Topologie hat Broadcast-Eigenschaft.
- Die Schicht-2-Adressen nach IEEE 802.2 erlauben die Verwendung von Gruppenadressen für Multicast. Dadurch lässt sich Multicast in einem LAN-Segment leicht und effizient realisieren.
- **Aber:** Ab Schicht 3 wurden in der Internet-Protokollarchitektur lange Zeit nur Peer-to-Peer-Adressen unterstützt! Und im weltweiten Verbund (insbesondere im Internet) muss Multicast auch WAN-Strecken überbrücken.

Multicast in der Netzwerkschicht

Prinzip: Duplizierung von Paketen so "tief unten" im Multicast-Baum wie möglich.

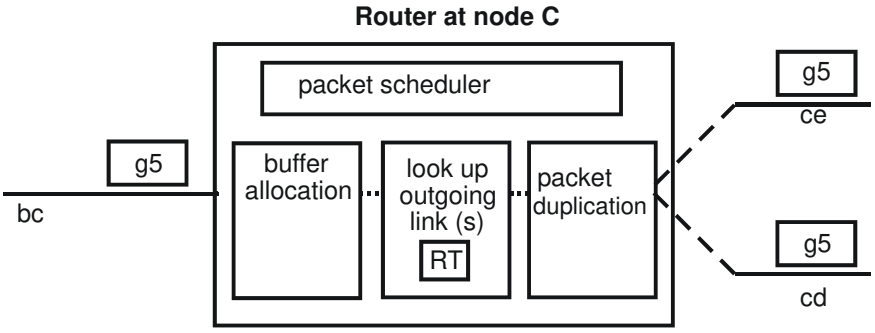
- Erfordert ein Multicast-Adressierungsschema in Schicht 3 und mehr "Intelligenz" in den Schicht 3-Vermittlungsstellen (Routern)
- verbindungslos oder verbindungsorientiert?



Router mit Multicast-Erweiterung

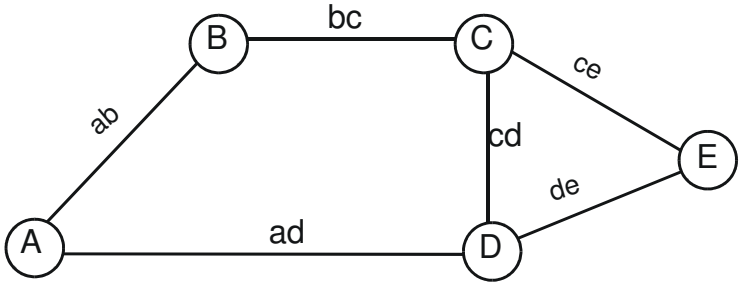
RT **Routing Table**

From C to	link	cost
g5	{ce, cd}	

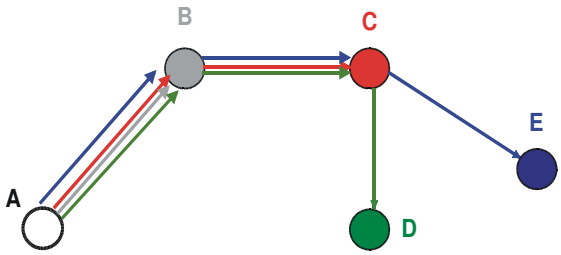


g5 = IP address of group 5

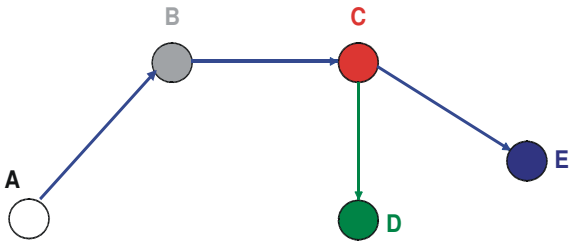
Beispiel-Topologie



Der Vorteil von Multicast in der Beispiel-Topologie



(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 Multicast-Gruppe aufgenommen werden möchte.

Prinzipien des Multicast-IP-Protokolls

- Übertragung von IP-Datenpaketen an eine **Gruppenadresse** (IP-Adresse vom Typ D)
- verbindungslos (Datagrammdienst)
- Best-Effort-Prinzip (keine Dienstgütegarantien):
  - keine Fehlerkontrolle
  - keine Flusskontrolle
- empfängerorientiert:
  - Der Sender sendet Multicast-Pakete an die Gruppe.
  - Der Sender kennt die Empfänger nicht, hat auch keine Kontrolle über diese.
  - Jeder Host im Internet kann einer Gruppe beitreten.
- Eine Beschränkung des Sendebereiches ist nur durch den Time-To-Live-Parameter möglich (TTL = hop counter im Header des IP-Pakets)

Multicast-Adressen in IP

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 existierenden Gruppe aufgenommen werden möchte. Die IP-Gruppenadresse wurde als IP-**Adresse der Klasse D** standardisiert.

Gruppenadressen werden dynamisch zugewiesen. Einen Mechanismus zur eindeutigen Vergabe einer Gruppenadresse gibt es in IP nicht! Um eindeutige Gruppenadressen müssen sich die höheren Schichten kümmern. Hierfür wurde beispielsweise das Software-Tool *sdr* (session directory) entwickelt.

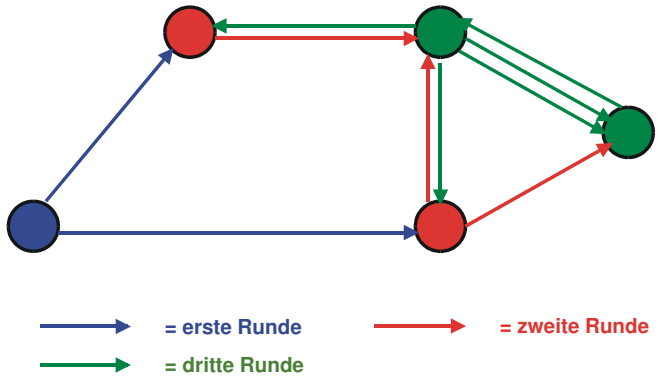
Routing-Algorithmen für Multicast

Flooding

Die einfachste Möglichkeit zum Erreichen aller Empfänger einer Gruppe wäre Flooding (Broadcasting).

Algorithmus Flooding

Wenn ein Paket eintrifft, wird eine Kopie auf jeder Ausgangsleitung weiter gesandt außer derjenigen, auf der das Paket ankam.



Reverse Path Broadcasting (RPB)

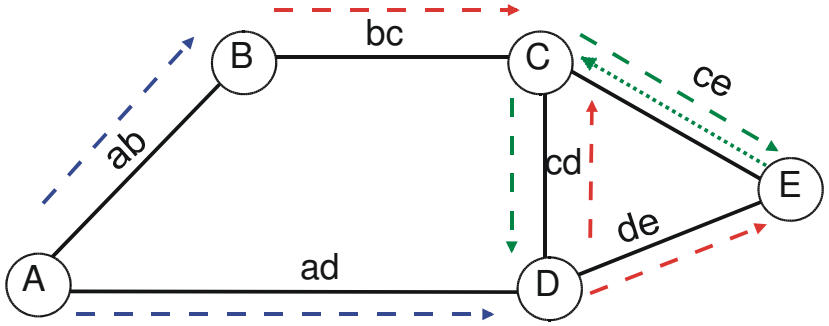
Effizienter als das Flooding ist der Reverse Path Broadcasting-Algorithmus (RPB). Er nutzt die Tatsache aus, dass jeder Knoten seinen kürzesten Pfad zum Sender aus der klassischen (point-to-point)-Routing-Tabelle kennt! Man bezeichnet diesen Pfad als **Reverse Path**.

Die erste Idee ist nun, dass ein Knoten nur diejenigen Pakete an seine Nachbarn weiter gibt, die auf dem kürzesten Pfad vom Sender angekommen sind.

Dieses Verfahren generiert wesentlich weniger Pakete als Flooding.

Beispiel für Reverse Path Broadcasting (unvollständiger Algorithmus)

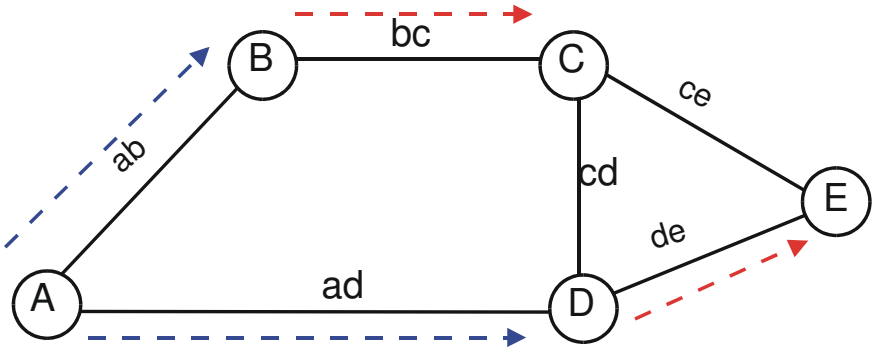
Für unsere Beispieltopologie arbeitet der (bisher noch unvollständige) RPB-Algorithmus wie folgt:



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

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. Ein Knoten leitet Pakete dann nur noch an diejenige “Söhne“ weiter, von denen er weiß, dass er auf ihrem kürzesten Pfad zum Sender liegt. Den Paketfluss für den vollen RPB-Algorithmus zeigt dann die unten stehende Abbildung.



Truncated Reverse Path Broadcasting (TRPB)

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 Blatt-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 zurück zu schneiden, dass Pakete nur noch dorthin weitergeleitet werden, wo sie wirklich gebraucht werden.

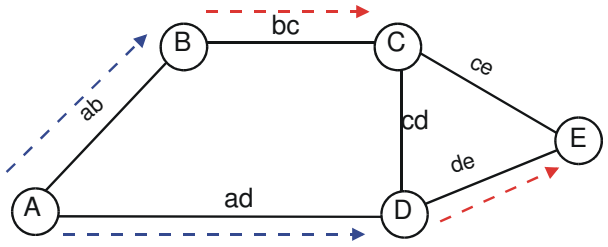
Dies geschieht durch die Generierung von “**prune messages**“. Diese wandern im Baum von den Blättern zur Wurzel hin und teilen den Knoten der jeweils höheren Ebene mit, dass es weiter unten im Baum keine Empfänger mehr gibt. So wird aus dem Broadcast-Baum ein Multicast-Baum. Das Verfahren wird als **Reverse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

Im Internet heißt das Protokoll zum RPM-Algorithmus **DVMRP** (Distance Vector Multicast Routing Protocol).

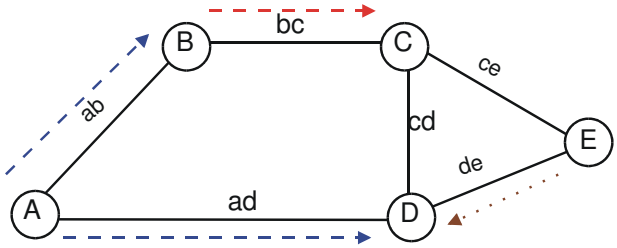
Algorithmus Pruning

- 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 vorher gehenden Router im Multicast-Baum.
- Router, die von allen untergeordneten Routern NMRs empfangen haben, senden ebenfalls einen NMR an den übergeordneten Router.
- NMRs enthalten eine Zeitschranke, nach der das Pruning wieder aufgehoben werden soll.
- NMRs können auch per Nachricht aufgehoben werden, wenn ein neues Gruppenmitglied unterhalb eines Links aktiv wird.

Beispiel für Reverse Path Multicasting (1)

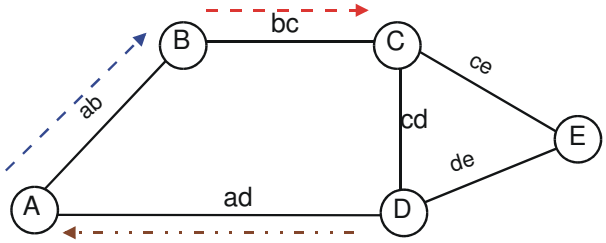


(a) Baum in der anfänglichen RPB-Phase



(b) E hat eine "prune message" versandt

Beispiel für Reverse Path Multicasting (2)



(c) D hat eine "prune message" versandt

Vor- und Nachteile von RPM

Vorteil

- Reduzierung des Datenverkehrs im Vergleich zu TRPB

Nachteile

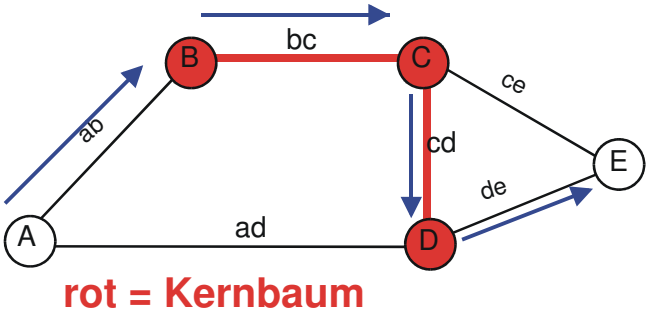
- Periodischer Versand der Daten an alle Router weiterhin nötig, damit sie „es sich anders überlegen“ können
- Statusinformation in jedem Knoten für jede Gruppe und für jeden Sender nötig
- Für jedes Paar (Sender, Gruppenadresse) muss ein eigener Routing-Tree aufgebaut werden.

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.

Das beste heute im Internet verfügbare Multicast-Routing-Protokoll heißt PIM-SM (Protocol-Independent Multicast – Sparse Mode). Es basiert auf der Idee des Kernbaums.

Beispiel für einen Kernbaum

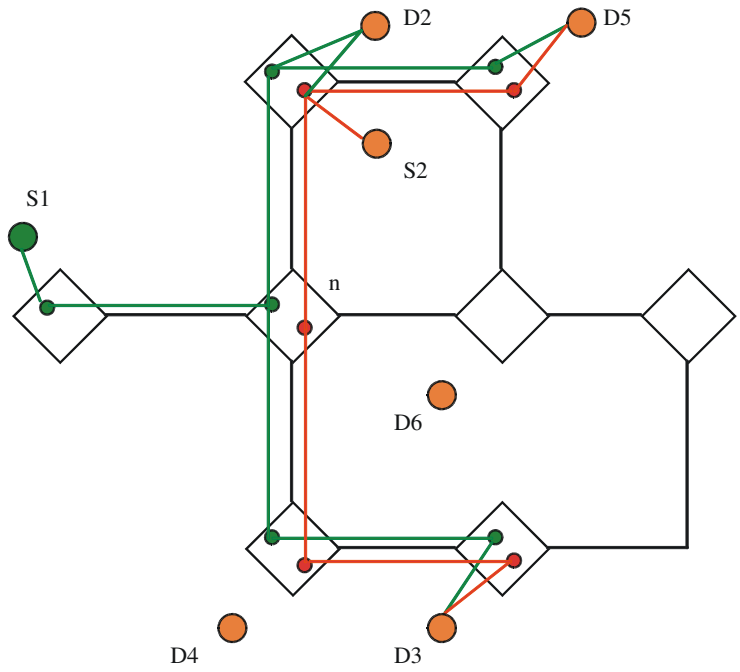


rot = Kernbaum

QoS-Based Routing

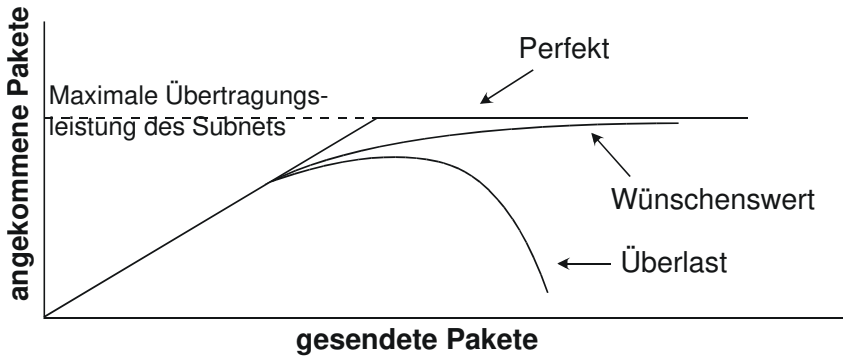
Multicast-Routing für IP ist nach wie vor ein aktuelles Forschungsthema. Noch weitgehend ungelöst ist das Problem eines Routings unter Einbeziehung von Dienstgüteanforderungen ("**QoS-based routing**").

Dynamic Join and Leave mit QoS-Garantie



5.5 Überlastkontrolle in der Vermittlungsschicht

Das Problem



Gründe für eine Überlast

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

Überlast: Tendenz zur Selbstverstärkung

Überlastung tendiert dazu, sich selbst zu verstärken.

Beispiel: Ein Knoten (Router) wirft wegen Überlastung Pakete weg

- Pakete müssen erneut gesendet werden (zusätzlicher Verbrauch an Bandbreite)
- Sender kann seine Puffer nicht freigeben (zusätzliches Binden von Ressourcen)

Besonders kritisch in Datagramm-Netzen!

Verfahren 1: Pufferreservierung (1)

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)

Verfahren 1: Pufferreservierung (2)

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, zum Beispiel bei der digitalen Sprachübertragung über paketvermittelte Netze.

Verfahren 2: Wegwerfen von Paketen (1)

Prinzip

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

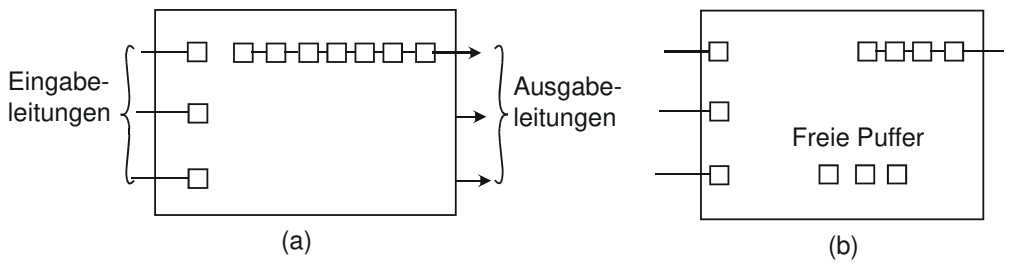
Datagramm-Dienst: Keine weiteren Vorkehrungen notwendig

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

Verfahren 2: Wegwerfen von Paketen (2)

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 kann trotz freier Puffer vorkommen.



Verfahren 2: Wegwerfen von Paketen (3)

Eigenschaften

- sehr einfach

aber

- wiederholt übertragene Pakete verschwenden Bandbreite

Ein Paket muss 1/(1-p) -mal gesendet werden, bevor es akzeptiert wird
(p = Wahrscheinlichkeit, dass das Paket verworfen wird)

Kleine, einfache Optimierung:

Zuerst diejenigen Pakete wegwerfen, die noch nicht weit gekommen sind
(Streckenzähler auswerten)

Verfahren 3: Isarithmische Überlastkontrolle

Prinzip

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

- Menge von "Permits" kreist im Netz
- Zum Senden wird ein "Permit" benötigt
  - Senden eines Pakets: „Permit" wird zerstört
  - Empfangen eines Pakets: neuer „Permit“ wird generiert

Probleme

- Teile des Netzes können überlastet werden, während andere Teile unterbelastet sind
- Gleichmäßige Verteilung der Permits über das Netz ist schwierig
- Zusätzliche Bandbreite wird für Permit-Transfer benötigt
- Ungeeignet zur Übertragung großer Datenmengen (z. B. Dateitransfer, multimedialer Datenstrom)
- Endgültiger Verlust von Permits durch Fehler im Netz schwer zu erkennen und zu beheben

Verfahren 4: Flusskontrolle missbrauchen

Prinzip

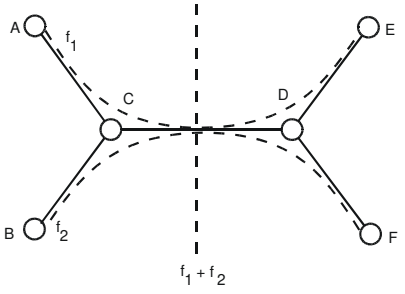
Flusskontrolle zur Überlastvermeidung missbrauchen:

- Die Flusskontrolle ist definiert zwischen Paaren von Endsystemen. Sie soll eigentlich nur das “Überschwemmen“ des Empfängers durch einen zu schnellen Sender verhindern.
- Das Netz darf bei Verfahren 4 aber nun in den inneren Knoten von sich aus die Fenstergröße des Sliding-Window-Protokolls einer Verbindung verändern. Dadurch kann der Fluss der Pakete auf der Verbindung verlangsamt werden.
- Implementiert zum Beispiel in Schicht 3 von SNA (IBM)

Im Internet ist eine sehr merkwürdige Variante implementiert. Nur TCP (nicht UDP) versucht, in den Endsystemen *auf Schicht 4* zu raten, wann das Netz verstopft ist. Dies geschieht auf der Basis von beobachteten Paketverlusten. Wenn der TCP-Sender Verstopfung vermutet, reduziert er freiwillig seine Senderate, um die Überlast im Inneren des Netzes zu mindern. Protokoll-Details dazu werden wir im TCP-Kapitel besprechen.

Nachteile der Überlastkontrolle durch Flusskontrolle

- Es ist im Sinne der Schichtenarchitektur unsauber, wenn die Überlastkontrolle in Schicht 4 gemacht wird. Denn Schicht 3 muss den Flusskontrollparameter im Paketkopf der Schicht 4 verändern, obwohl sie eigentlich das Protokoll der Schicht 4 nicht kennen sollte.
- Funktioniert nur bei verbindungsorientierter Kommunikation, nicht für Datagrammverkehr! Problem: Wie implementiert man ein „TCP-friendly flow control for UDP“?
- Oft führen mehrere Paketflüsse über einen gemeinsamen Link der Schicht 3. Wie kann die Flussreduzierung **fair** erfolgen?



Verfahren 5: "Choke"-Pakete

Prinzip

Netzmanagement-Pakete drosseln den Verkehr bei Überlast:

- Jede Ausgangsleitung eines Routers ist mit einer Variablen u ($0 \leq u \leq 1$) 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
- Wenn die Quelle ein Choke-Paket empfängt, reduziert sie den Datenverkehr 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.