

# Next Generation Internet

## Kapitel 7: Multicast-Transport

INSTITUT FÜR TELEMATIK



# Überblick Kapitel 7

## I. Einführung

1. Einführung

## II. Internet-Architektur

2. Internet-Architektur
3. NAT & IPv6
4. Dienstgüte

## III. Multicast

5. Grundlagen
6. Multicast Routing
7. Multicast Transport

## IV. Flexible Dienste und Selbstorganisation

8. Neuere Transportprotokolle
9. Aktive Netze
10. Peer-to-Peer

- 7.1 Multicast-Transport: Eigenschaften
- 7.2 Transportprotokolle für Multicast:  
Anforderungen und Mechanismen
- 7.3 Multicast-Transportprotokolle:  
Ausgewählte Beispiele

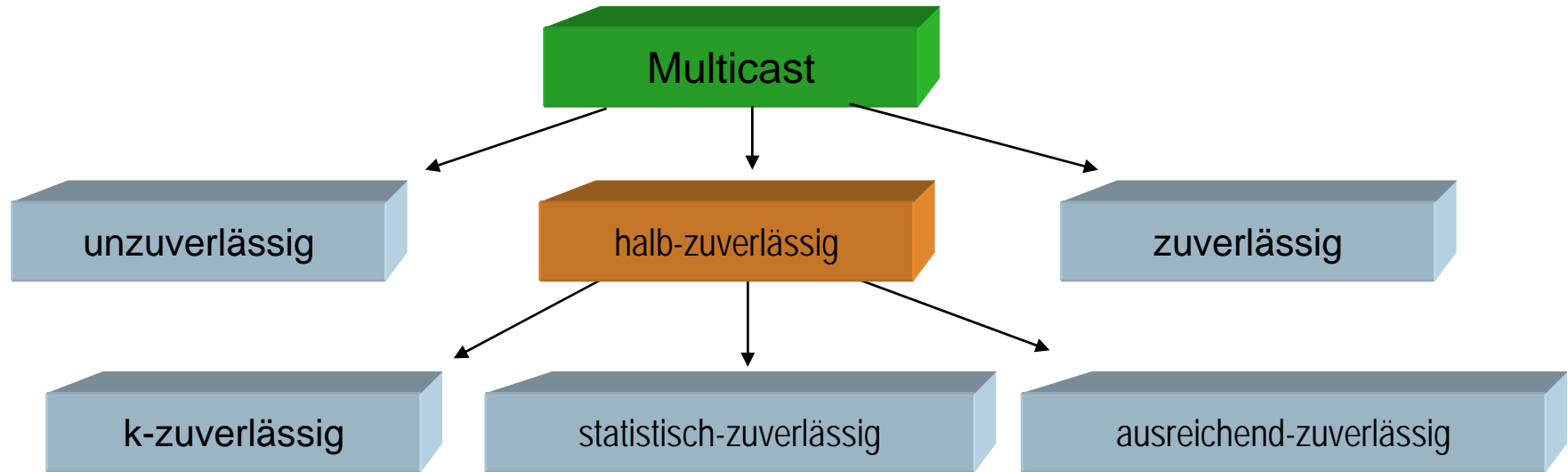
# 7.1 Gruppen-Transportdienste

- Traditionelle Transportdienste für Punkt-zu-Punkt-Kommunikation: **unzuverlässig** (UDP), **zuverlässig** (TCP)
- Erweiterung auf Gruppenkommunikation
  - Bei unzuverlässigen Diensten
    - Nutzung von UDP möglich
    - Verwendung von Multicast-Adressen als Zieladressen
    - Beitritt zu Gruppen und Verlassen von Gruppen
  - Bei zuverlässigen Diensten zusätzlich erforderlich:
    - Verbindungsauf- und -abbau
    - Fehlerkontrolle, Quittungsbehandlung, Sendewiederholungen, Staukontrolle
    - Sicherheit (z.B. Denial-of-Service-Schutz)
  - Ein oder mehrere Sender:
    - **1:n** → meist ohne menschliche Interaktion, z.B. SW-/Datenverteilung,
    - **m:n** (Multipeer dann meist m:m) → oft interaktiv, z.B. verteilte Spiele/Simulationen, Tele-/Videokonferenzen

# Problem

- Traditionelle Mechanismen nicht ohne Weiteres anwendbar/übertragbar
  - Nur für Punkt-zu-Punkt-Kommunikation definiert
  - Skalierbarkeitsprobleme bzgl. Anzahl der Empfänger
    - Kontrolldaten
      - Menge der übertragenen Nachrichten (z.B. Anzahl Quittungen)
      - Anzahl der zu verarbeitenden Nachrichten
    - Zustandshaltung
- Es zeichnet sich keine einheitliche Lösung ab
  - Gründe hierfür sind u.a.:
    - Aufgabenbereiche für Transport variieren stark
    - Erfahrungen mit Gruppenkommunikation sind noch gering
  - Daher werden **modulare Lösungen** entwickelt

# Zuverlässigkeitsklassen



## ■ k-zuverlässig

- k Gruppenmitglieder müssen Daten korrekt erhalten haben. Absolutes Maß.

## ■ Statistisch zuverlässig

- Prozentsatz von Gruppenmitgliedern muss Daten korrekt erhalten haben. Relatives Maß.

## ■ Ausreichend zuverlässig

- Wird in der Praxis oft verwendet
- Keine Aussage, wieviele Empfänger die Daten korrekt erhalten haben, möglich



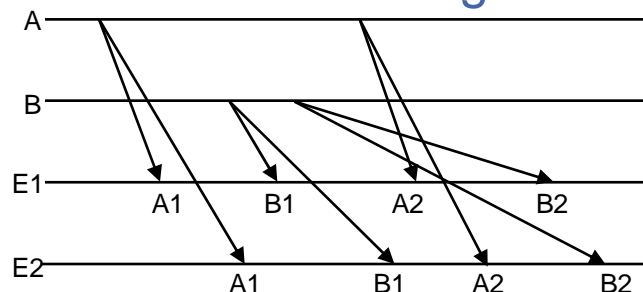
# Zuverlässiger Gruppendienst

- Zuverlässigkeit: Alle Empfänger erhalten Daten
  - fehlerfrei
  - ohne Duplikate
  - in der korrekten Reihenfolge (Sendereihenfolge)
- Atomar zuverlässig
  - Korrektheit, keine Duplikate, Auslieferung an alle oder keinen
- Globale Ordnungserhaltung
  - Ordnungserhaltend
    - atomar zuverlässig
    - Daten eines Senders kommen reihenfolgetreu an
  - Total Ordnungserhaltend (nur Multipeer)
    - ordnungserhaltend
    - alle Daten aller Sender erreichen alle Empfänger in der gleichen Reihenfolge

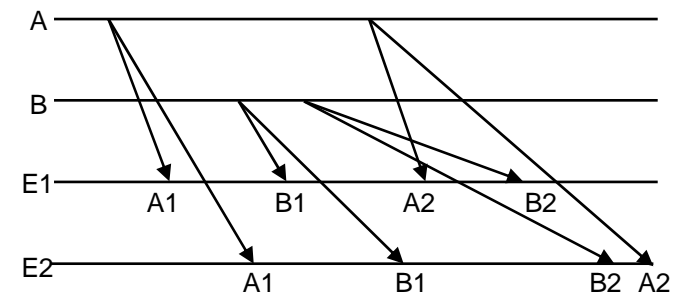
# Ordnungserhaltung

- **Multicast:** Entspricht Unicast-Kommunikation
- **Multipeer:** Berücksichtigung unterschiedlicher Sender
- **Ordnungskriterien**
  - **Globale Ordnung:**
    - Daten werden in genau der Reihenfolge ausgeliefert, in der sie gesendet wurden
  - **Totale Ordnung:**
    - Daten werden bei allen Empfängern in der gleichen Reihenfolge ausgeliefert
  - **Source-Ordered:**
    - Daten werden nur bezogen auf eine Quelle in der richtigen Reihenfolge ausgeliefert


## Totale Ordnung



## Source-Ordered



# 7.2 Transportprotokolle für Multicast: Allgemeines

- Anforderungen aus Sicht der IETF  [RFC3048]
  - **Staukontrolle**
    - Keine Überlastung durch **Zusatzverkehr** (**Redundanz**) oder Verkehr zur Behebung von Fehlersituationen (sog. **Reparatur-Verkehr** – „**Repair Packets**“)
    - Erreichen einer guten Link-Auslastung
    - Keine Ausgrenzung konkurrierender Datenströme („Verhungern“)
  - **Skalierbarkeit**
    - Einsetzbar unter verschiedenen Bedingungen
      - Verschiedene Netztopologien
      - Verschiedene Link-Geschwindigkeiten
      - Verschiedene Empfängergruppen
    - Gutes Verständnis erforderlich, wann Protokoll einsetzbar bzw. nicht geeignet
  - **Sicherheit**
    - Authentisierung des Senders
    - Bekämpfung von Denial-of-Service-Attacken





# Multicast-Transport-Unterstützung innerhalb des Netzes

## ■ Keine Unterstützung im Netz

- Router leiten Daten nur weiter
- Mechanismen zur Multicast-Transport-Unterstützung komplett in Sender und Empfängern (Ende-zu-Ende-Prinzip)
- Geschichtete Datenströme (Layered Streams)
  - Daten werden in unterschiedliche Datenströme aufgeteilt, die jeweils einer Gruppe zugeordnet werden
  - Empfänger treten den Gruppen einzeln bei, beispielsweise je nach individuell gewünschter Dienstqualität

## ■ Server-basiert

- Dedizierte Systeme im Netz unterstützen Sender und Empfänger
- Systeme sind nicht notwendigerweise im Datenpfad

## ■ Router-basiert

- Router unterstützen Sender und Empfänger
- Router müssen Ressourcen hierfür bereitstellen

# Fehlerbehandlung

- Zuverlässigkeit erfordert Fehlerbehandlung
- **Multicast-Fehlerkontrolle** ist grundsätzlich aufwändiger als bei Punkt-zu-Punkt-Kommunikation
  - **Fehlererkennung** wird in der Regel vom Empfänger durchgeführt
  - **Fehlerbehebung** erfolgt oftmals durch Übertragungswiederholung
  - Quittungen werden zum Sender geschickt
- Skalierbarkeit
  - Mechanismen müssen so ausgelegt werden, dass sie bei **großen** und bei **geographisch weit verstreuten** Gruppen **einsetzbar** sind

# Quittungsverarbeitung (1)

## ■ Sender muss Quittungen vieler Empfänger behandeln

### ■ Problem der **Quittungs-Implosion**

- Hohe Netzbelastung
- Hoher Pufferbedarf beim Sender
- Hoher Bearbeitungsaufwand beim Sender

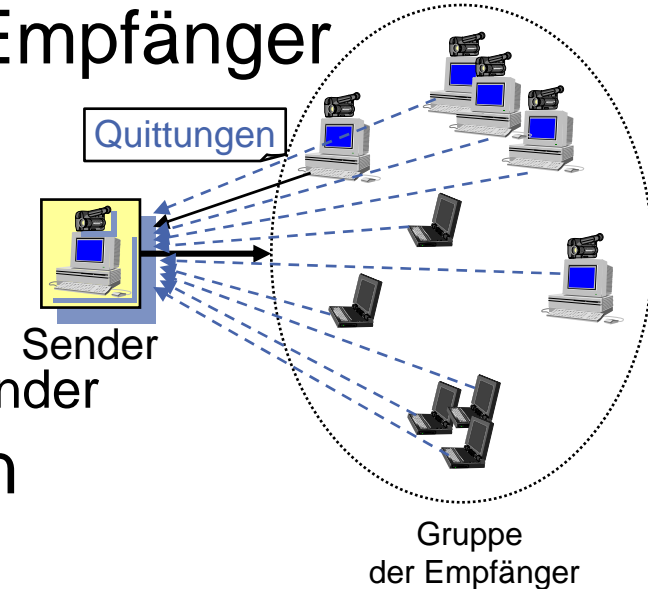
## ■ Es ist zu unterscheiden zwischen

### ■ Multicast **ohne Empfängerliste**

- Quittungen werden der gesamten Gruppe zugeordnet
- Reaktion des Senders auf Quittung kann sofort oder verzögert erfolgen

### ■ Multicast **mit Empfängerliste**

- Quittungen können einzelnen Empfängern zugeordnet werden
- Sender besitzt detaillierte Information über jedes einzelne Gruppenmitglied



# Quittungsverarbeitung (2)

■ Ziel: Vermeiden der Quittungs-Implosion

■ Verfahren

■ Lokale Bearbeitung von Quittungen für Teilgruppen in Servern bzw. Routern

■ Vorteil

■ Verteilung der Bearbeitungslast auf mehrere Kommunikationssysteme, parallele Abarbeitung möglich.

■ Ausgezeichnete Empfänger bzw. spezielle Zwischensysteme bearbeiten Quittungen bzw. Kontrolldateneinheiten

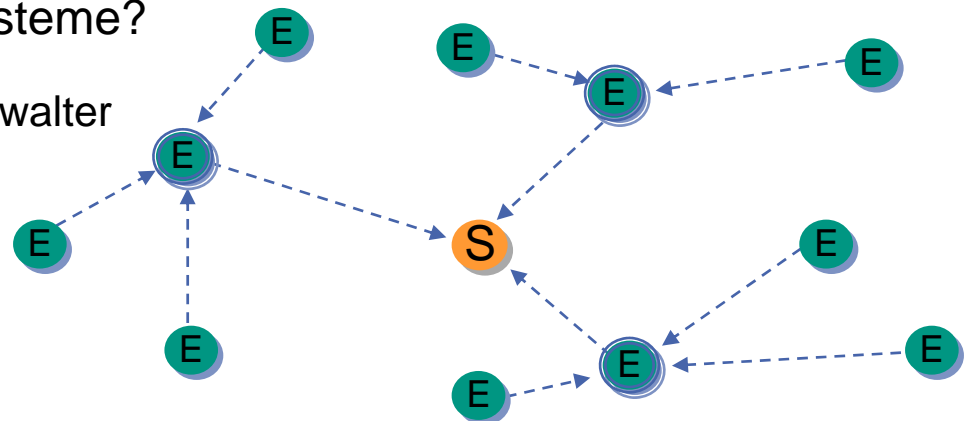
■ Bestimmung dieser Systeme?

⊙ Empfänger/Verwalter

⊙ Empfänger

⊙ Sender

--- Quittung

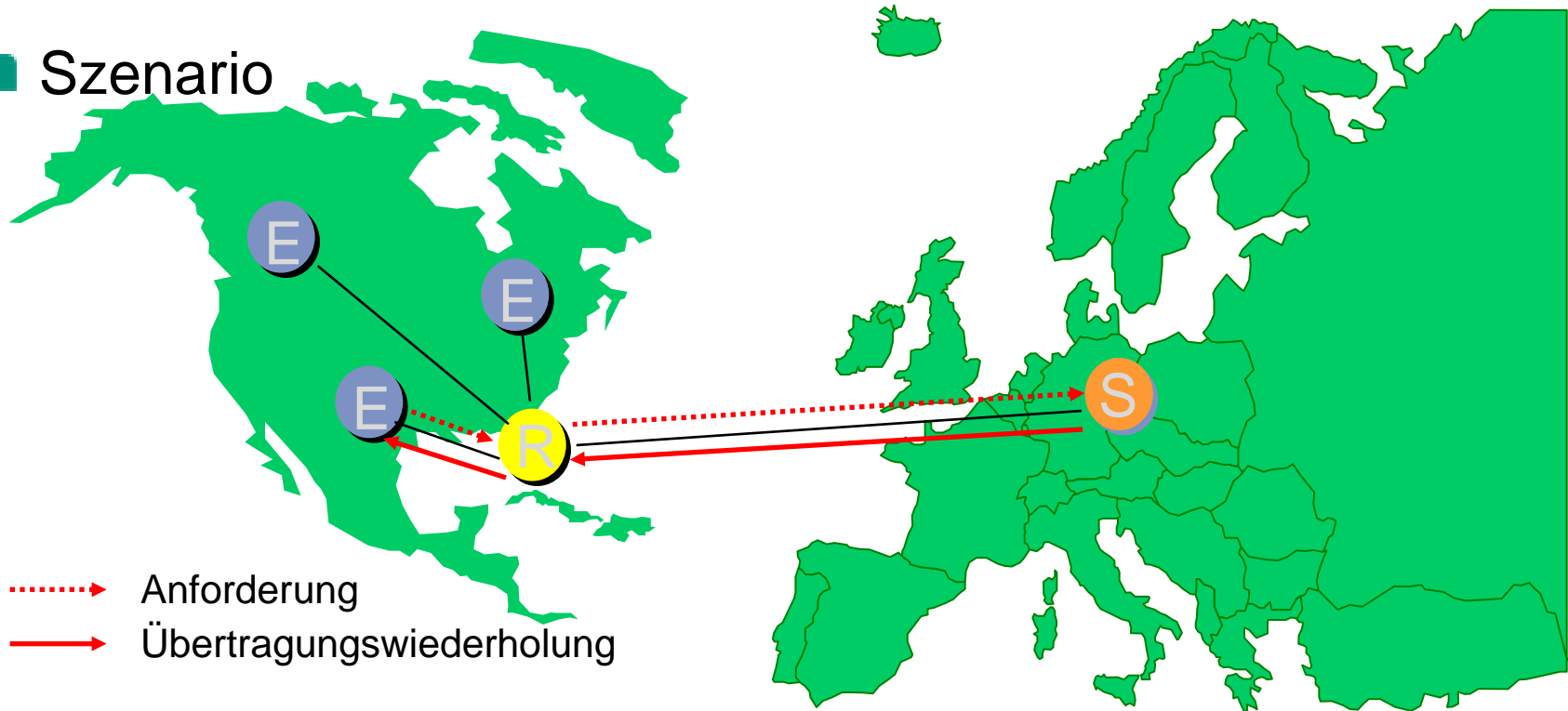


# Übertragungswiederholung

- Grundsätzlich dieselben Mechanismen einsetzbar wie für Unicast
  - Go-Back-N
  - Selektive Übertragungswiederholung
- Möglichkeiten zur Wiederholung der Daten
  - Per Multicast
    - Erfordert nur eine Wiederholung, auch für mehrere Empfänger
    - Belastet alle Empfänger und involvierte netzinterne Zwischensysteme
    - Verbesserung durch **Subcasting** (s. NGI-Kapitel: Reliable Multicast Transport Protocol)
  - Per Unicast
    - Erfordert eine dedizierte Wiederholung pro Empfänger, d.h., die Empfänger müssen bekannt sein
    - Ineffizient, wenn Daten von mehreren Empfängern angefordert werden

# Beispiel: Geographisch verteilte Gruppen

## ■ Szenario



## ■ Basisverfahren

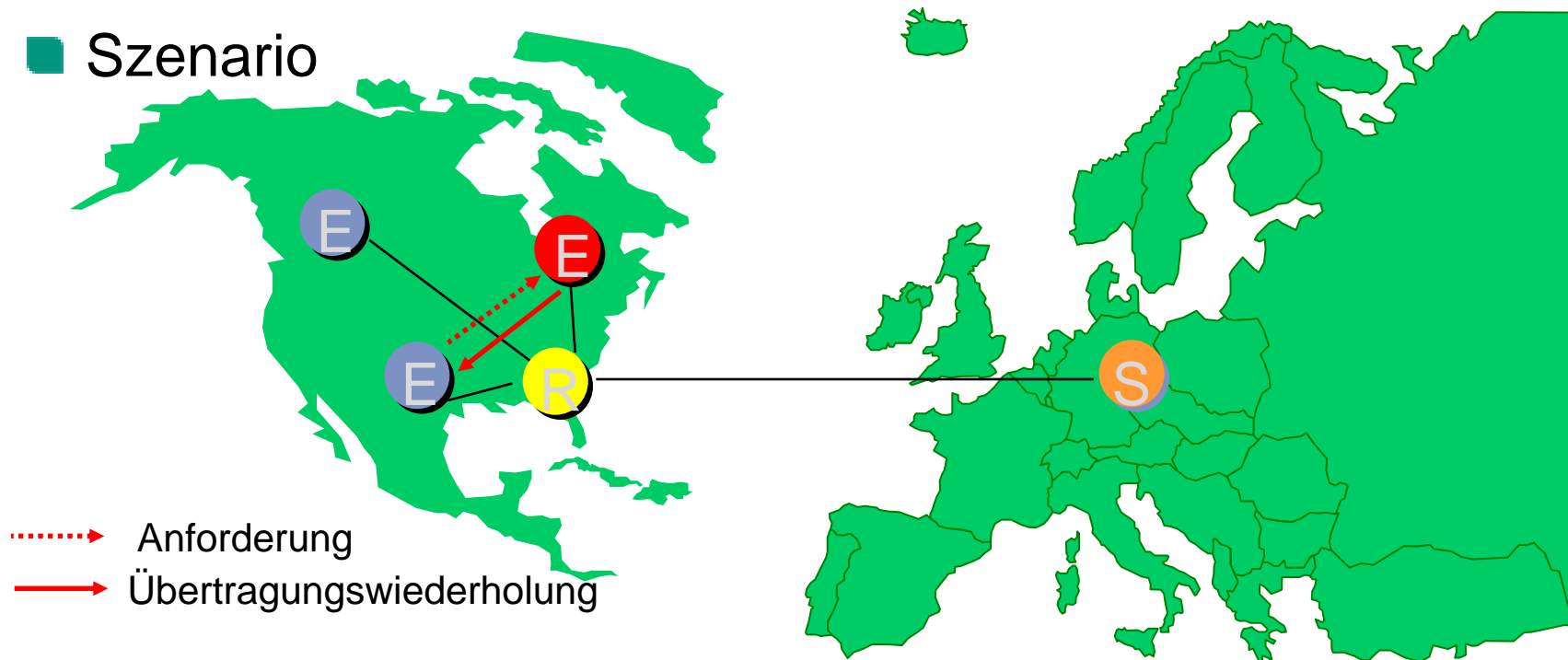
- Empfänger fordern fehlende Daten stets direkt beim Sender an

## ■ Problem

- Eventuell hohe Belastung beim Sender und lange Laufzeiten, da keine Berücksichtigung der Gruppenstruktur und der Netztopologie

# Optimierte Fehlerbehebung durch dedizierte Systeme

## ■ Szenario



## ■ Optimierung

- Übertragungswiederholungen durch dedizierte Systeme, hier „lokale“ Empfänger

## ■ Vorteil

- Minimierung der (globalen) Netzbelastung und der durchschnittlichen Übertragungsverzögerung durch lokale Übertragungswiederholungen



## 7.2.1 Verfahren zur Multicast-Fehlerbehebung

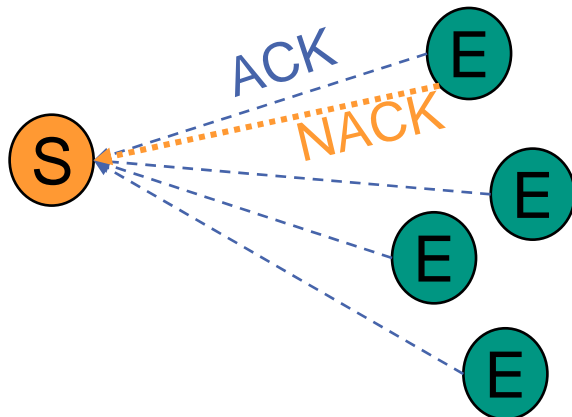
- Die folgenden Varianten der Fehlerbehebung können im Kontext von Multicast-Transportprotokollen unterschieden werden
  - Sender-gesteuert
  - Empfänger-gesteuert
  - Ring-basiert
  - Baum-basiert
  - Zeitgeber-gesteuert
  - Vorwärtsfehlerkorrektur



# Sender- vs. Empfänger-gesteuert

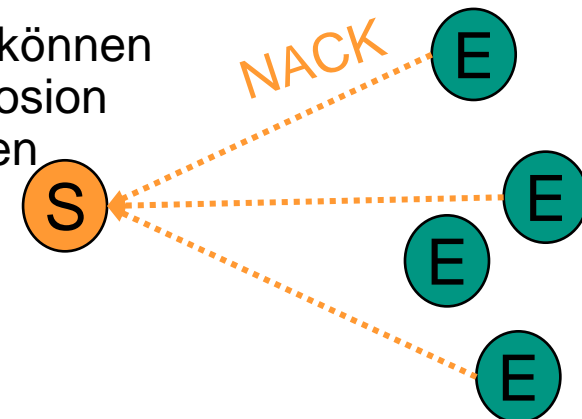
## Sender-gesteuert

- Empfänger bestätigen korrekten Empfang mit **ACKs**
- Sender stößt Übertragungswiederholung an
- Bekannte Gruppe erforderlich
- Problem der Quittungs-Implosion inhärent
- Geringer Durchsatz
- Vollzuverlässiger Dienst realisierbar mit endlichem Speicher
- Mischform mit NACKs möglich



## Empfänger-gesteuert

- Empfängerseitige Fehlerkontrolle
- Empfänger fordern im Fehlerfall mit **NACKs** Übertragungswiederholungen an
- Gruppenmitglieder müssen nicht bekannt sein
- Sender muss keine Information über Zustand der Empfänger verwalten
- Höherer Durchsatz als bei Sender-gesteuerten Verfahren erreichbar
- Freigabe der Sende-Puffers aus NACKs nicht ableitbar
- NACKs allein können Quittungsimplosion nicht vermeiden





# Empfänger-gesteuert mit NACK-Vermeidung

## ■ Ziel

- Der Sender (bzw. ein entsprechender anderer Knoten) soll **nur ein NACK** für eine fehlende Dateneinheit erhalten
- Die Empfänger, die eine Sendewiederholung veranlasst haben, sollen nur eine Kopie der Daten erhalten

## ■ Beispiele:

### ■ Zeitgeber

- Empfänger senden NACK im Fehlerfall erst nach Ablauf eines Zeitgebers
  - Zeitgeber-Dimensionierung schwierig
- Trifft NACK für die gleiche Dateneinheit ein, wird der Zeitgeber erneut gestartet
- NACKs und Übertragungswiederholungen belasten die gesamte Gruppe

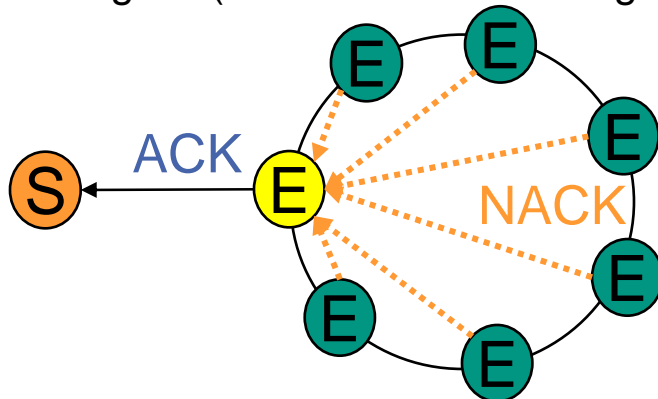
### ■ Router-Unterstützung

- Router unterdrücken doppelte NACKs

# Ring- vs. Baum-basiert

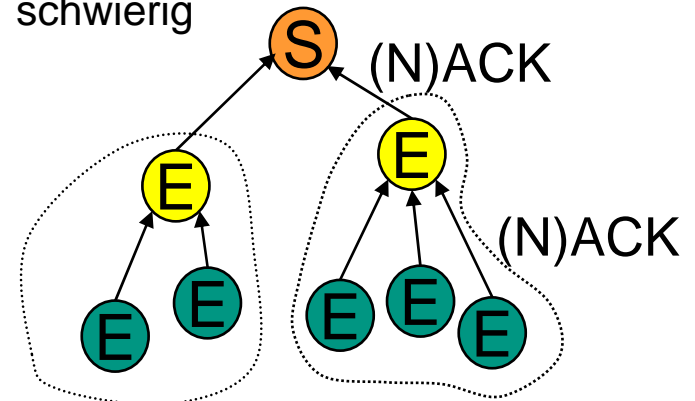
## ■ Ring-basiert

- Gruppe bildet einen virtuellen Ring, in dem ein **Token** zirkuliert
- Empfänger senden NACKs an den aktuellen **Token-Halter**
- Token-Halter sendet ACK an Sender
- Vermeidung der Quittungsimplosion
- Totale Ordnungserhaltung realisierbar
- Sender kann Speicher nach endlicher Zeit freigeben
- Durchsatz hängt von Gruppengröße ab, skaliert daher nicht gut
- Quittungs-Implosion beim Token-Halter möglich (→ Hierarchische Ringe)

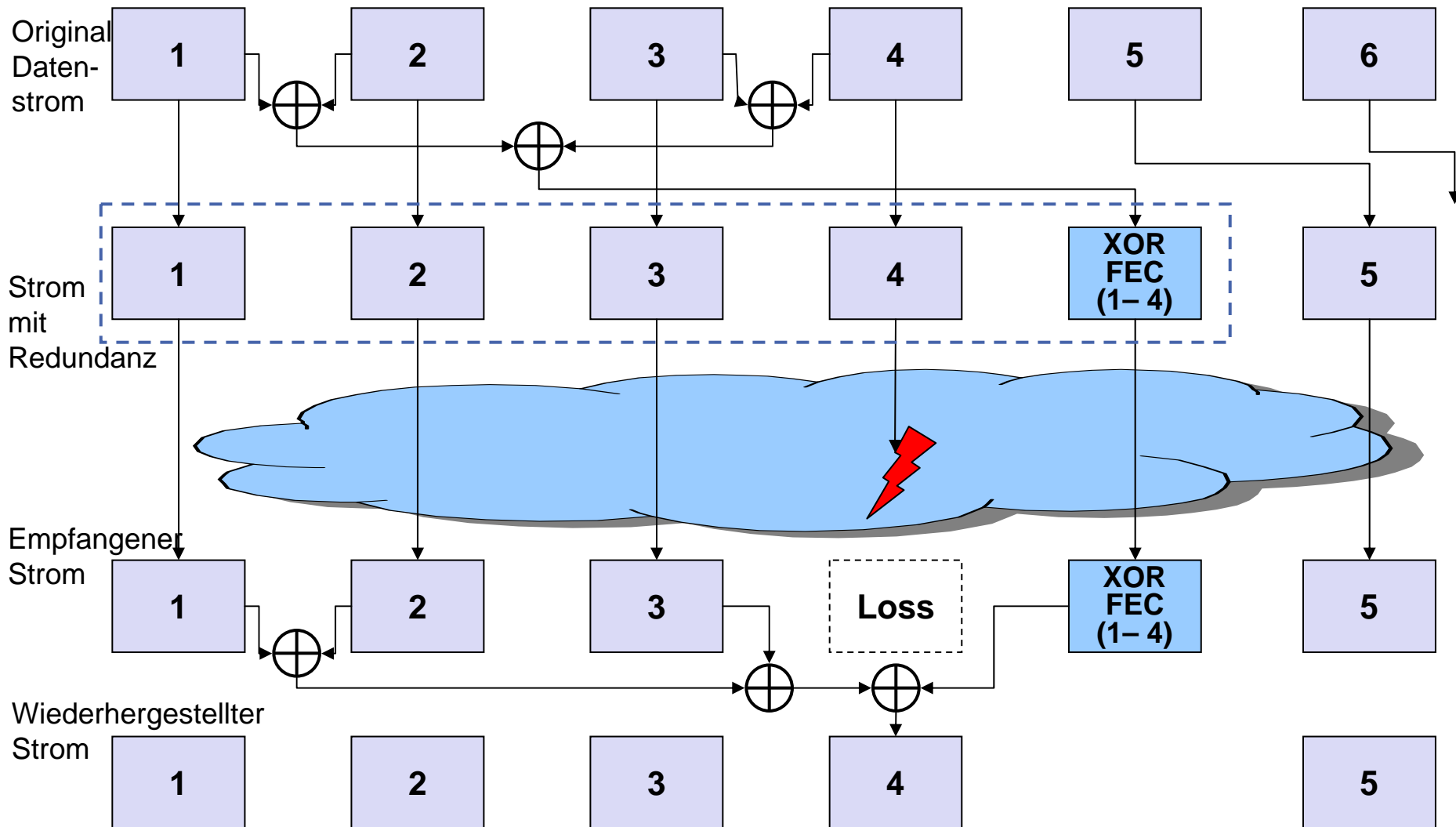


## ■ Baum-basiert

- Hierarchie von lokalen Teilgruppen mit ausgezeichnetem **Group Leader**
- Quittungen werden an jeweils zuständigen Group Leader gesendet
- **Aggregation von Quittungen** in den Group Leaders
- Quittungs-Implosion wird vermieden
- Mit aggregierten ACKs ist „komplett“ zuverlässiger Dienst möglich (mit endlichem Speicher)
- Erhöhte Ende-zu-Ende-Verzögerung
- Aufbau und Verwaltung des Baums schwierig



# Vorwärtsfehlerkorrektur





## 7.2.2 Verfahren zur Multicast-Staukontrolle

### ■ Sender-kontrolliert

- mit einer Gruppe
  - Empfänger liefern Feedback-Information an Sender
  - Sender stellt Senderate auf Empfänger mit langsamsten Pfad ein
- mit mehreren Gruppen
  - initiale Gruppe wird dynamisch in Teilgruppen gegliedert
  - angepasste Senderate für verschiedene Teilgruppen

### ■ Empfänger-kontrolliert

- mit einer Gruppe
  - Empfänger verlassen die Gruppe, falls Verlustrate zu hoch
- mit geschichteter Organisation (Layered Multicast)
  - Sender verteilt Daten auf mehrere Multicastgruppen
  - Empfänger treten den jeweiligen Multicastgruppen bei bzw. verlassen diese in Abhängigkeit der aktuellen Netzlast
  - Empfänger „hinter“ einem Engpass im Netz müssen koordiniert den Multicastgruppen beitreten bzw. diese verlassen

# Router-basierte Multicast-Staukontrolle

## ■ Router-basiert

- zusätzliche Mechanismen zur Staukontrolle in Routern

Beispielsweise:

- Bedingte Beitritte zu einer Gruppe
  - Oberhalb einer angegebenen Verlustrate verweigert der Router den Beitritt
- Filtern von Daten (Verwerfen von Paketen)
  - Router filtert Daten, die eine derzeit vernünftige Datenrate übersteigen

## ■ Grundsätzlich erfordern Router-basierte Verfahren zusätzliche **Zustandshaltung in den Routern**

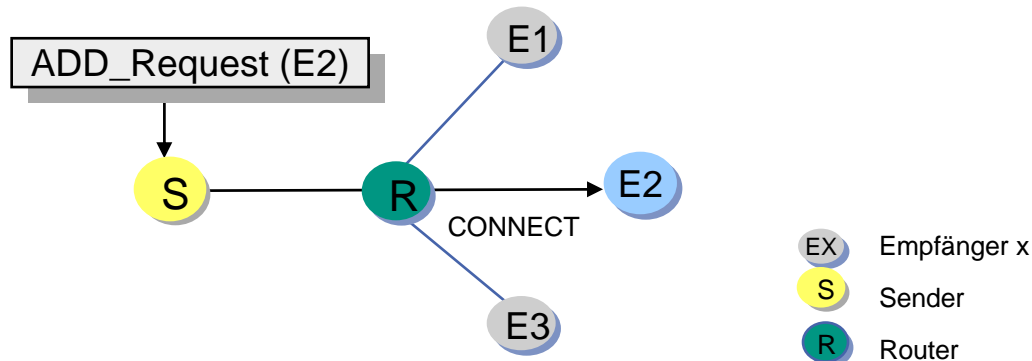
- Werden derzeit deshalb nicht im Backbone zu erwarten sein
- Typische Anwendung für aktive/programmierbare Netze

## 7.2.3 Multicast-Verbindungsaufbau

### ■ Sender-gesteuerter Beitritt zur Multicast-Verbindung

- Sender erhält die Aufforderung zur Aufnahme eines weiteren Empfängers (ADD\_Request)
- Sender übermittelt eine Verbindungsaufbaunachricht

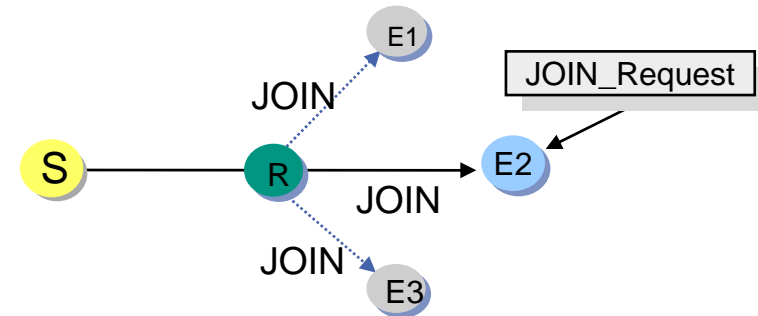
- Geeignet z.B. für Konferenzsysteme mit einer zentralen Verwaltung der Kommunikationsteilnehmer



### ■ Empfänger-gesteuerter Beitritt zur Multicast-Verbindung

- Beitrittswillige Empfänger senden JOIN\_Request
- Sender übermittelt unter Nutzung der Multicast-Adresse den Beitrittswunsch (JOIN) an alle Gruppenmitglieder
- Nur der Empfänger reagiert auf den Beitrittswunsch

- Geeignet z.B. für „Video on Demand“ oder „Internet Radio“



## 7.3 Multicast-Transportprotokolle: Ausgewählte Beispiele

- Es gab bereits vor einiger Zeit eine Reihe von experimentellen Transportprotokollen  
Beispiele sind
  - MTP (Multicast Transport Protocol)
  - RMP (Reliable Multicast Protocol)
  - RMTP (Reliable Multicast Transport Protocol)
  - PGM (Pragmatic General Multicast)
- Zuverlässigkeit steht beim Entwurf der meisten Protokolle im Mittelpunkt
- Im Folgenden werden einige Beispiele präsentiert. Konkret etabliert hat sich heute noch kein Protokoll.



## 7.3.1 Multicast Transport Protocol (MTP)

- Eigenschaften von MTP
  - Halbzuverlässiger Multipeer-Transportdienst mit globaler Ordnungserhaltung
  - Empfänger-gesteuert
- Anforderungen
  - Unicast- und Multicast-fähige Vermittlungsschicht
- Grundlegendes Schema
  - Token regelt Sendeberechtigung
  - Multicastgruppe wird bei MTP als **Web** bezeichnet und für die Mitglieder werden drei Rollen unterschieden
    - **Master** – einmal pro Gruppe, kontrolliert das Verhalten des Webs → dient zur Ordnungserhaltung
    - **Producer** – Sender von Nutzdaten, erhält Kontrolldateneinheiten
    - **Consumer** – Agiert nur als Empfänger

# Datentransfer

## ■ Drei grundlegende Parameter

### ■ Heartbeat

- Für Raten- und Fehlerkontrolle

### ■ Window

### ■ Retention

- Speicherdauer für potentielle Übertragungswiederholung  
→ halb-zuverlässig

## ■ Ablauf

- Producer muss im Heartbeat-Intervall mindestens eine Dateneinheit senden

- **EMPTY**, falls sonst nichts zu senden

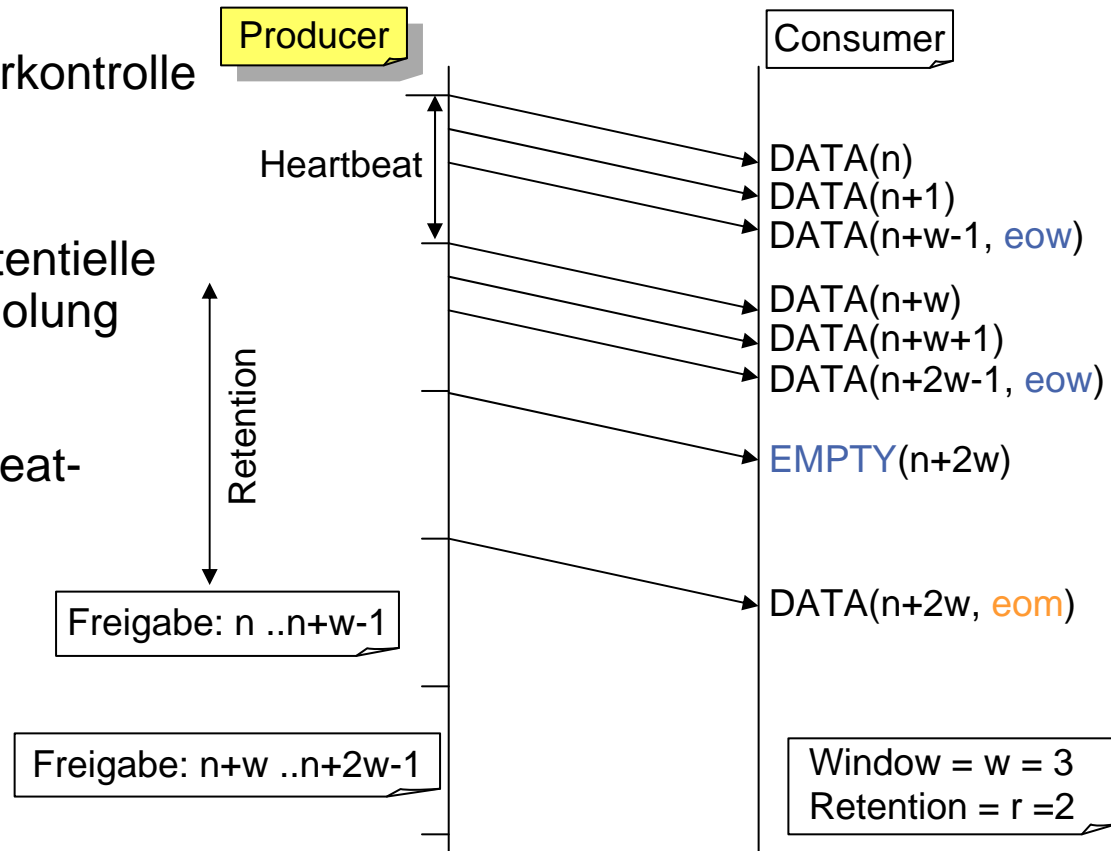
### ■ Tokenfreigabe

- Durch **eom** angezeigt

- Fenster ausgeschöpft

- Durch **eow** signalisiert

## ■ Schema



# Fehlerkontrolle

## Ablauf

- Empfänger erkennt Verlust anhand der Sequenznummer, die höher ist als erwartet
- Negative, selektive Quittungen
- Wiederholungen unterliegen der Ratenkontrolle
  - Sind Daten nicht mehr vorhanden, antwortet der Producer mit NACK-DENY

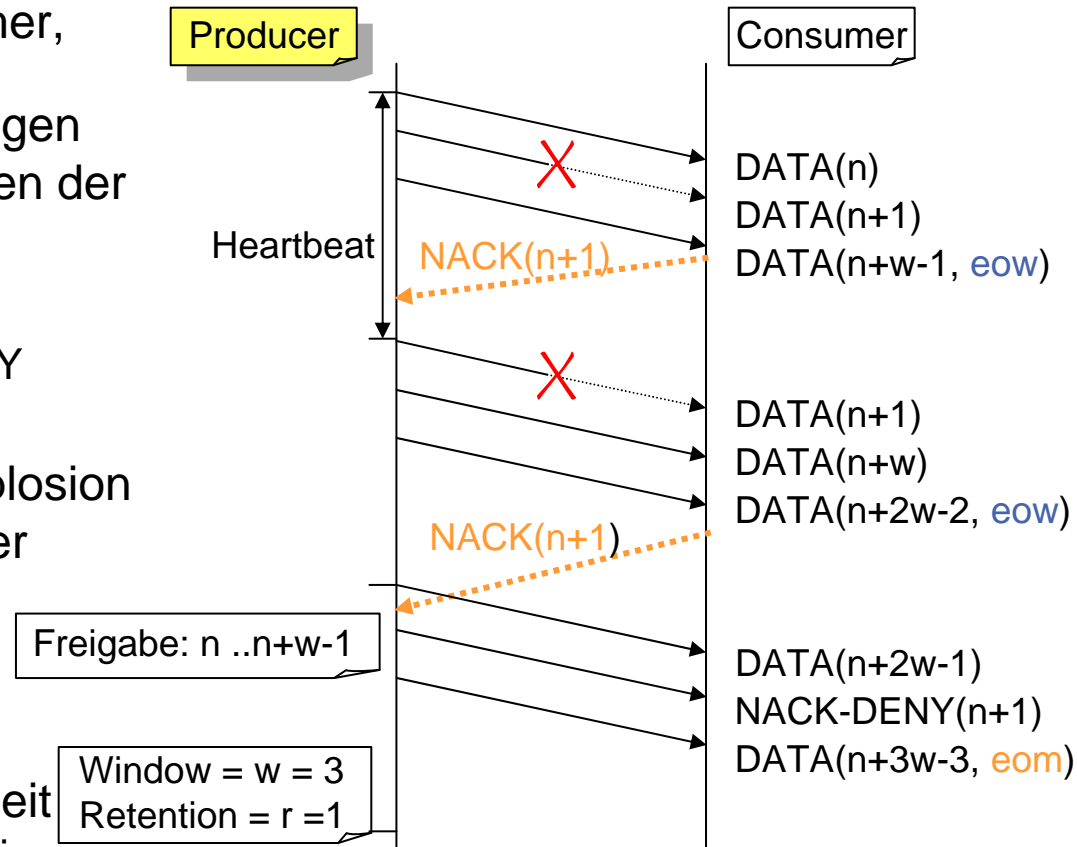
## Bewertung

- Problem der Quittungs-Implosion
- Gefahr, dass alle Consumer quasi gleichzeitig reagieren

## Partitionierung des Netzes

- Consumer erhält während Heartbeat keine Dateneinheit
  - Abbruch der Kommunikation

## Schema





## 7.3.2 Reliable Multicast Protocol (RMP)

### ■ Eigenschaften

- Ring-basiertes Protokoll
- Zuverlässige und **ordnungserhaltende** m:n-Kommunikation

### ■ Anforderungen

- Basiert auf UDP und IP

### ■ Grundlegendes Schema

- Modifikation des Token-Passing-Protokolls
- Token-Ring
  - Grundlegende Einheit zur Gruppenkommunikation
  - Basis zum Ordnen der Dateneinheiten innerhalb einer Gruppe
  - Alle Mitglieder einer Gruppe sind im gleichen Token-Ring angeschlossen
  - Gleichzeitige Mitgliedschaft in mehreren Token-Ringen ist möglich

# 7.3.3 Reliable Multicast Transport Protocol (RMTP)

## ■ Eigenschaften

### ■ Halb-zuverlässiger Multicast

- Zielanwendung von RMTP sind Verteildienste (z.B. Software-Versionen oder Aktienpreise)

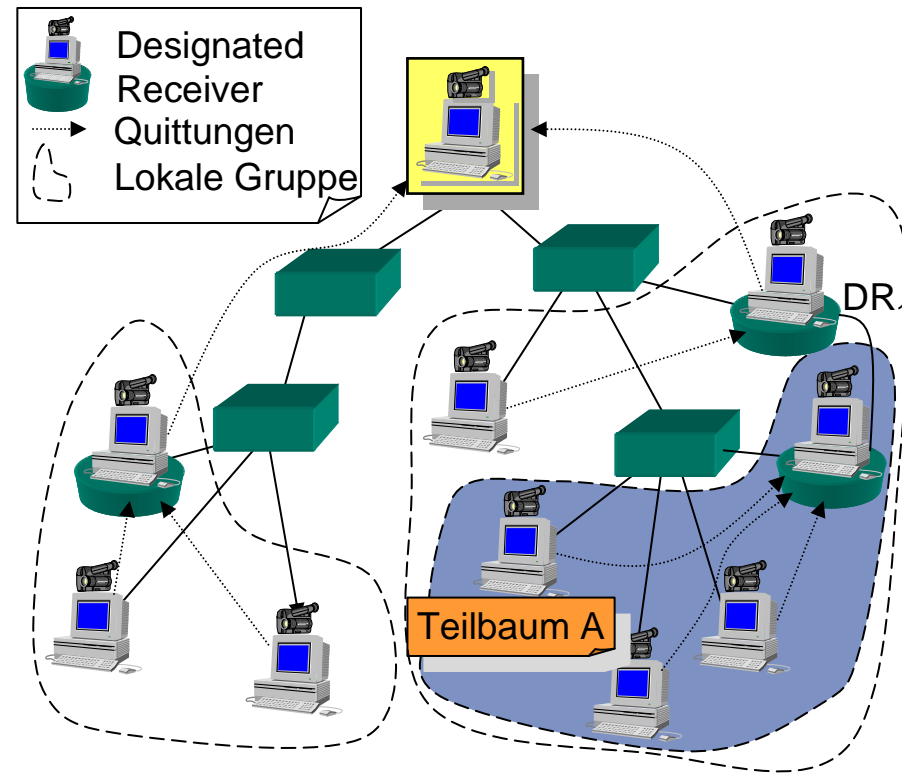
## ■ Grundlegendes Schema

- Aufbau eines **Baums**, in dem sich die Empfänger befinden
- Lokale Übertragungswiederholungen, um Skalierbarkeit zu erhöhen
  - Auswahl sogenannter **Designated Receiver**
    - Hierfür werden ausgesuchte Empfänger herangezogen
    - Aspekte der Sicherheit in Bezug auf Designated Receiver werden nicht angesprochen
    - Verarbeiten Quittungen für den „darunter“ liegenden Teilbaum
- Zustandshaltung ist unabhängig von der Gruppengröße
- Mechanismen zur Fluss-, Raten- und Staukontrolle werden bereitgestellt

# Fehlerkontrolle

- Empfängerorientiert
  - Periodische positive, selektive Quittungen
    - Anzahl der Quittungen steigt linear mit Anzahl der Empfänger
    - Quittungen werden an Designated Receiver gesendet
    - DR wiederholt Daten (per Multicast oder Unicast)
  - Designated Receiver kann Dateneinheit anfordern, falls er sie nicht besitzt
  - Sendewiederholungen sind auf entsprechenden Teilbaum limitiert
- Subcasting
  - Weiterleiten der Daten nur auf Teilbaum, der logisch unterhalb des Routers liegt
  - In Multicast-IP nicht verfügbar, deshalb wird Tunneling verwendet

- Designated Receiver
  - Dynamische Auswahl erforderlich, wegen Ausfall oder Netzpartitionierung



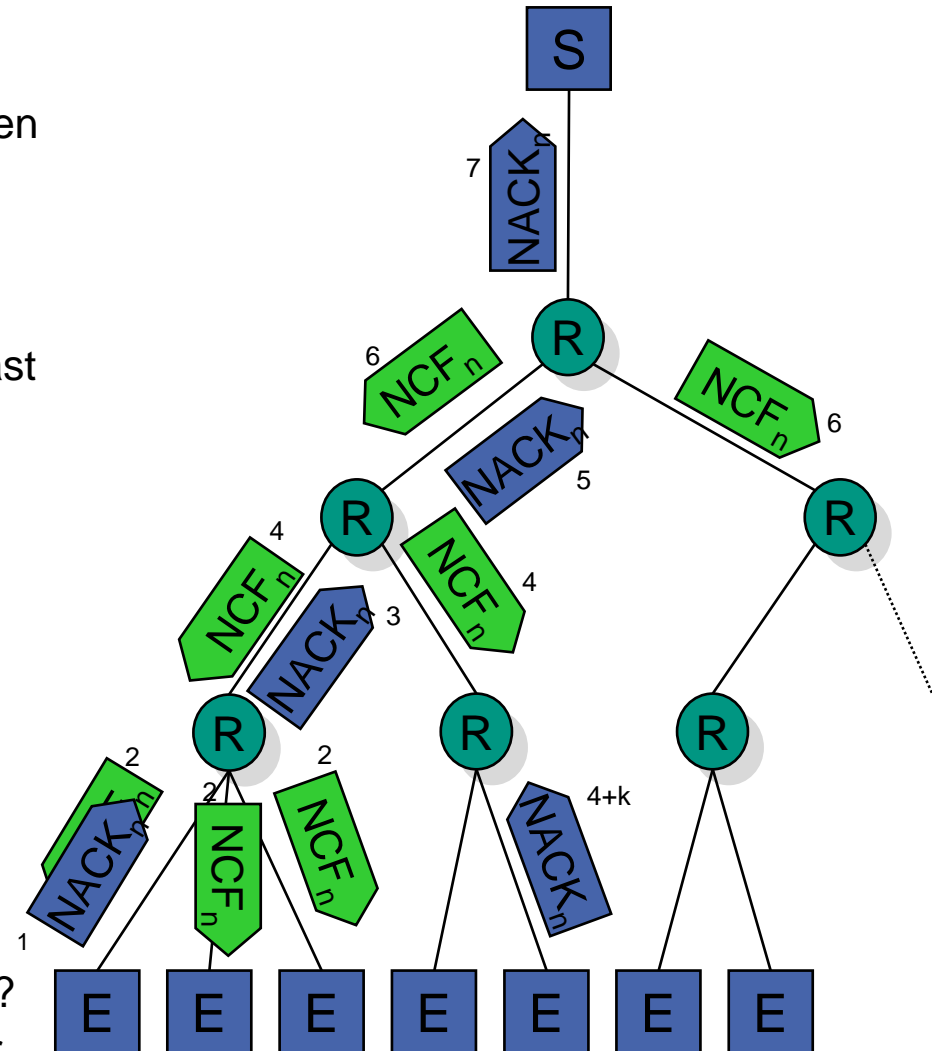
## 7.3.4 Pragmatic General Multicast (PGM)

- Eigenschaften
  - Halbzuverlässiges Multipeer-Protokoll
  - Ordnungserhaltung liegt beim Empfänger: Source-Ordered oder nicht geordnet
- Anforderungen
  - Unzuverlässiger Multicastdienst, z.B. Multicast-IP
- Grundlegendes Schema
  - Empfänger-basierte Fehlerkontrolle mit NACKs
  - Router sind aktiv an der Bereitstellung des Multicastdienstes beteiligt (NACK-Vermeidung)
    - Aufhebung der strikten Protokollebenen-Trennung
  - Gruppenmitglieder müssen nicht bekannt sein
- Bemerkung
  - An der Entwicklung war ein namhafter Routerhersteller beteiligt, trotzdem konnte sich das Protokoll bis jetzt nicht durchsetzen.

# Fehlerkontrolle

## ■ Negative Quittungen

- NACKs per Unicast an übergeordneten Router
  - Verteilbaum muss bekannt sein
  - Periodische Source Path State Messages (SPM) vom Sender aus
- NACK Bestätigung (NCF) per Multicast auf Interface, über das NACK empfangen wurde
  - Vorteile beim Endsystem? Beim Router?
- Unterdrückung gleicher NACKs im Teilbaum soll Quittungs-Implosion vermeiden
  - Statushaltung in den Routern pro empfangenem NACK
    - Welche NACKs schon empfangen?
    - Auf welchen Interfaces gleiche NACKs empfangen?
- Wiederholungen werden vom Sender per Multicast an die Gruppe gesendet





# Flusskontrolle

## ■ Sender-gesteuert

### ■ Flusskontrollfenster



- Durch verschiedene Kontrollschemata zu beeinflussen
  - Keines fest implementiert
- Definiert durch die Menge an Daten, die sich der Sender für mögliche Übertragungswiederholungen merken muss
- Gegeben durch die Zeit, die eine Dateneinheit gespeichert bleibt

### ■ Bandbreitenregulierung durch Ratenkontrolle

- Maximale Übertragungsrate definiert

## 7.3.5 IETF Reliable Multicast Transport (RMT) Working Group

### ■ Ziele

- Zunächst ausschließlich 1:n-Multicast
- Zwei Protokolldefinitionen für unterschiedliche Anforderungen
  - NACK Oriented Reliable Multicast (NORM)
  - Asynchronous Layered Coding (ALC)
- Definition von Building-Blocks für RMT
  - Für mehrere Protokolle
  - Forward Error Correction Building Block  [RFC5052]
    - Basic Schemes & verschiedene Verfahren (z.B. Reed-Solomon)
  - NACK Building Block  [RFC3941]
  - Layered Coding Transport (LCT) Building Block
    - Basis für ratenadaptive Transportprotokolle
  - Building Blocks für Ratenkontrolle und TCP-freundliche Staukontrolle
- Generelle Diskussionen (z.B. Security für RMT)

# NACK Oriented Reliable Multicast (NORM)



[RFC3940, RFC3941]

- Ende-zu-Ende-Transport
  - Zwischenknoten benötigen nur Standard-IP-Multicast Support
- TCP-kompatible Staukontrolle
- Nutzt FEC und NACK Building Blocks
- Selektive, negative Quittungen
- NACK-Unterdrückung
  - Skalierbarkeit: für zehntausende Teilnehmer
  - Empfänger: Zustand pro Sender
  - Sender: Zustand pro Empfänger, der Staukontrollinformationen liefert
- Soll auch in dynamischen, fehlerbehafteten Umgebungen gut funktionieren
  - Mit hohen Verlustraten und Delays
  - z.B. in mobilen/drahtlosen Netzen



# Asynchronous Layered Coding (ALC)



[RFC3450]

## ■ Nutzt

- Layered Coding Transport Building Block
- Multi-Rate Congestion Control Building Block
- FEC Building Block

## ■ Skalierbarkeit:

- Millionen von Empfängern
- Hunderte von Gigabytes pro Session
- Minimales Protokoll
  - „Bootstrapping“ nicht definiert
  - Feedback an Sender nicht vorgesehen

## ■ Ansätze basierend auf ALC (Drafts):

- FLUTE: File Delivery over Unidirectional Transport
  - Beispiel-Protokoll für die Nutzung von ALC
  - Erlaubt Signalisierung von Dateieigenschaften (Meta-Daten) und Multiplexing von Dateien
- FCAST: Scalable Object Delivery
  - Nutzung von ALC für beliebige Objekte (beliebige Meta-Daten)



# Layered Coding Transport (LCT)



[RFC3451]

## ■ Multi-Raten Zustellung

- Mehrere Kanäle mit unterschiedlichen Raten pro Multicast-Session (mit einem Sender)
- Empfänger abonnieren Kanäle entsprechend der zur Verfügung stehenden Bandbreite
- Dadurch empfangergesteuerte Staukontrolle möglich
  - Ohne Feedback, d.h. massiv skalierbar
  - Mechanismus fest eingebaut, erlaubt aber beliebige Verfahren

## ■ Coding

- Wenn Daten durch Kodierung auf unterschiedliche Kanäle aufgeteilt werden können
  - Erlaubt dem Empfänger, die Qualität zu steuern
  - Ursprünglich für Audio-/Video-Streams verwendet
  - Für Bulk-Daten: Unterschiedliche Redundanz mit FEC

## 7.4 Übungen

- 7.1 Welche Gründe gibt es für die Einführung halb-zuverlässiger Multicastdienste?
- 7.2 Erläutern Sie die Unterschiede einer Ordnungserhaltung bei Unicast und Multicast.
- 7.3 Weshalb wird ein einziges Multicastprotokoll nicht ausreichend sein?
- 7.4 Vergleichen Sie den Sender-gesteuerten mit dem Empfänger-gesteuerten Beitritt.
- 7.5 Erläutern Sie die Vor- und Nachteile von Empfängerlisten.
- 7.6 Bei welchen Typen von Gruppen bieten sich Empfängerlisten an?
- 7.7 Stellen Sie die Verfahren zur Multicast-Fehlerkorrektur gegenüber.
- 7.8 Wie unterscheiden sich Unicast und Multicast hinsichtlich der Sicherheitsmechanismen?
- 7.9 Wie unterscheiden sich die Token-Mechanismen von MTP und RMP?
- 7.10 Kategorisieren Sie die vorgestellten Protokolle hinsichtlich des realisierten Multicast-Dienstes.
- 7.11 Wie garantiert MTP globale Ordnungserhaltung?
- 7.12 Vergleichen Sie den Aufwand für Übertragungswiederholungen bei MTP und RMP.
- 7.13 Diskutieren Sie die Vor- und Nachteile von PGM.
- 7.14 Wie werden unterschiedliche Übertragungsraten bei ALC realisiert?

## 7.5 Literaturhinweise

- [Atwo04] J. Atwood: A Classification of Reliable Multicast Protocols, IEEE Network Magazine, May/June 2004
- [DiDC97] C. Diot, W. Dabbous, J. Crowcroft; Multipoint communications: A survey of protocols, functions and mechanisms; IEEE Journal on Selected Areas in Communications, Vol. 15, No. 3, April 1997
- [Paul98] S. Paul; Multicasting on the Internet and its Applications; Kluwer Academic Publishers, 1998
- [ToKP97] D. Townsley, J. Kurose, S. Pingali; A comparison of sender-initiated and receiver-initiated reliable multicast protocols; IEEE Journal on Selected Areas in Communications, Vol. 15, No. 3, April 1997
- [WiZi01] R. Wittmann, M. Zitterbart; Multicast Communication: Protocols and Applications; Morgan Kaufmann Publishers, 2001
- [RFC 2887] M. Handley, S. Floyd, B. Whetten, R. Kermode, L. Vicisano und M. Luby. The Reliable Multicast Design Space for Bulk Data Transfer. RFC 2887 (Informational), August 2000. URL: <http://www.ietf.org/rfc/rfc2887.txt>

# Literaturhinweise (2)

- [RFC 3048] B. Whetten, L. Vicisano, R. Kermode, M. Handley, S. Floyd und M. Luby. Reliable Multicast Transport Building Blocks for Oneto-Many Bulk-Data Transfer. RFC 3048 (Informational), Januar 2001. URL: <http://www.ietf.org/rfc/rfc3048.txt>
- [RFC 3451] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, M. Handley und J. Crowcroft. Layered Coding Transport (LCT) Building Block. RFC 3451 (Experimental), Dezember 2002. URL: <http://www.ietf.org/rfc/rfc3451.txt>
- [RFC 3738] M. Luby und V. Goyal. Wave and Equation Based Rate Control (WEBRC) Building Block. RFC 3738 (Experimental), April 2004. <http://www.ietf.org/rfc/rfc3738.txt>
- [RFC 3940] B. Adamson, C. Bormann, M. Handley und J. Macker. Negative acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Protocol. RFC 3940 (Experimental), November 2004. URL: <http://www.ietf.org/rfc/rfc3940.txt>
- [RFC 3941] B. Adamson, C. Bormann, M. Handley und J. Macker. Negative- Acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Building Blocks. RFC 3941 (Experimental), November 2004. URL: <http://www.ietf.org/rfc/rfc3941.txt>



# Literaturhinweise (3)

- [RFC 4654] J. Widmer und M. Handley. TCP-friendly Multicast Congestion Control (TFMCC): Protocol Specification. RFC 4654 (Experimental), August 2006. URL: <http://www.ietf.org/rfc/rfc4654.txt>
- [RFC 5052] M. Watson, M. Luby und L. Vicisano. Forward Error Correction (FEC) Building Block. RFC 5052 (Experimental), August 2007. URL: <http://www.ietf.org/rfc/rfc5052.txt>