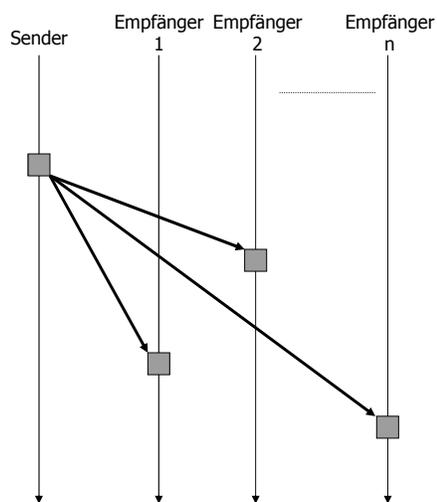


Verteilte Systeme

6. Multicast

Multicast



Senden einer Nachricht an mehrere Empfänger

Möglichst effiziente Verbreitung an alle Empfänger

- Nachrichteneffizienz, d.h. Gesamtzahl der im Netz verschickten Einzelnachrichten
- Direkte Nutzung multicast-fähiger Netze
- Multicastsemantik

Aufbau und Verwaltung von Empfängergruppen

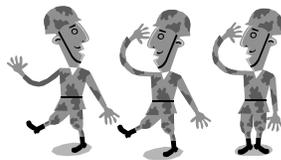
- Konsistente Sichten
- Dynamik: Hinzukommen und Weggehen von Mitgliedern

Broadcast = Alle Knoten sind in der Empfängergruppe

Einsatzgebiete für Multicast

Verbreitung von Information

- Beispiele
 - Internetradio
 - Video, Video on Demand
 - Fernsehen
- Meist in Verbindung mit QoS
- Abrechnungsproblematik
 - Empfänger zahlt



Fehlertolerante Dienste

- Aufträge an eine Replikatgruppe übermitteln
- Konsistenzproblematik
 - Gleichlauf der Replikatzustände

6.1 Verbreitung der Multicast-Nachrichten

Einfluß der Netzarchitektur

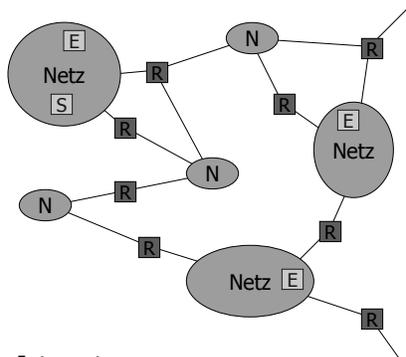
Einzelne Netze

Multicast-fähige Netze

- Einfache Umsetzung
- Effizienz häufig mit Unicast vergleichbar
- Beispiele
 - Ethernet
 - Ringe
 - ...

Nicht multicast-fähige Netze

- Überflutungsalgorithmen
 - Broadcast okay
 - Multicast
 - Welche Teile des Netzes sind betroffen, welche nicht?



Internetze

Innerhalb der einzelnen Netze

- siehe links

Zwischen den Netzen

- Überflutungsalgorithmen innerhalb der Router- und Gateway-Topologie
- Tunneln oder Auslassen nicht betroffener Netze

Verteilte Systeme, Wintersemester 2000/2001
Folie 6.5

Multicast-Semantiken

Trivial-Multicast

- Wieviele und welche Empfänger ist unbestimmt
- Reihenfolge aufeinanderfolgender Multicasts unbestimmt

Wie hilfreich ist dieser Multicast?

Zuverlässigkeitsgrad

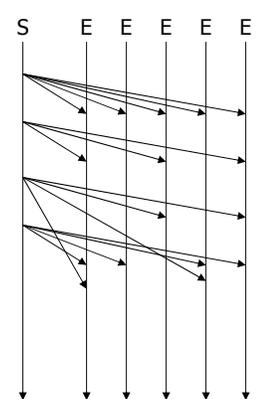
- Garantien bzgl. Der Empfängeranzahl

Ordnungsgrad

- Garantien über die Empfangsreihenfolge Multicasts eines Senders
- Multicasts verschiedener Sender

Quizfrage

- Wieviele Möglichkeiten gibt es, wenn n Sender jeweils k Multicasts an eine Gruppe mit m Mitgliedern verschicken?



Verteilte Systeme, Wintersemester 2000/2001
Folie 6.6

Zuverlässigkeitsgrad

Keiner

- Keine Garantie darüber, wieviele und welche Empfänger eine Nachricht erhalten

K-Zuverlässigkeit

- Mindestens k Mitglieder einer Gruppe erhalten die Multicast-Nachrichten
- Es müssen nicht immer dieselben k Empfänger sein

Atomar

- Es erhalten entweder alle Empfänger oder kein Empfänger die Multicast-Nachricht

Ordnungsgrad



Keine Ordnungsgarantien

FIFO-Ordnung

- Nachrichten eines Senders kommen bei allen Gruppenmitgliedern in FIFO-Ordnung an

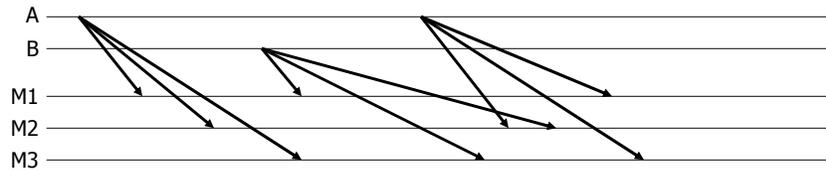
Kausale Ordnung

- Eine von einer Multicast-Nachricht m_1 kausal abhängige Nachricht m_2 ($m_1 <_k m_2$) erhalten alle Gruppenmitglieder erst nach dem Empfang von m_1
Meist auf Multicast-Nachrichten beschränkt

Totale Ordnung

- Alle Multicast-Nachrichten werden global geordnet und gemäß dieser Ordnung an die Gruppenmitglieder zugestellt
Umfaßt kausale Abhängigkeiten
„Willkürliches“ Kriterium für kausal unabhängige Nachrichten

FIFO-Ordnung

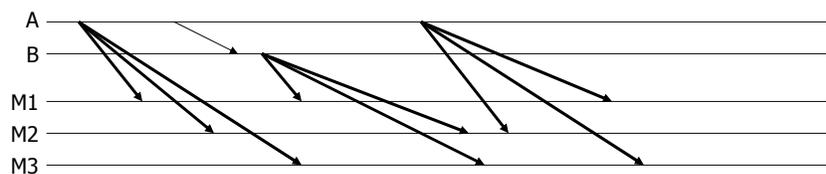


FIFO-Ordnung der Nachrichten eines Senders

Beliebige Überlappungen zwischen Nachrichten unterschiedlicher Sender

Realisierbarkeit?

Kausale Ordnung



Entgegen obigem Beispiel ist kausale Ordnung meist auf Multicast-Nachrichten beschränkt

– Warum?

Kausal unabhängige Nachrichten (verschiedener Sender) können weiterhin in beliebiger Reihenfolge eintreffen

Realisierbarkeit?

Totale Ordnung

Kausale Ordnung bleibt erhalten
 Zusätzlich „willkürliche“ Ordnung kausal unabhängiger Multicasts
 – Vgl. erweiterte Lamportzeit
 Realisierbarkeit?

Verteilte Systeme, Wintersemester 2000/2001 Folie 6.11

Receive und Deliver

Bei höheren Anforderungen an Zuverlässigkeits- und Ordnungsgrad kann empfangene Multicast-Nachricht u.U. noch nicht an Anwendung weitergegeben werden

Beispiele?

Verteilte Systeme, Wintersemester 2000/2001 Folie 6.12

Beispiel: Ethernet-Multicast

1 Multicast-Adresse (47)

Wahl einer gemeinsamen Multicast-Adresse für eine Empfängergruppe

- Höheres Protokoll

MAC speichert k abonnierte Gruppenadressen

- Betriebssystem verwaltet diese Einträge

Zu wenig Einträge im MAC:

- Alle Multicast-Nachrichten müssen vom Betriebssystem entgegen genommen werden

Rechner A

G17,22

G22

G85

MAC

G17
G22
G85

Rechner B

G17

G16

G85

G205

MAC

*
*
*

Verteilte Systeme, Wintersemester 2000/2001 Folie 6.13

Überflutungsalgorithmen

Virtueller Multicast oder Broadcast bei fehlender Netzwerkfunktionalität

Idee

- Ausgangsgrade > 1 nutzen

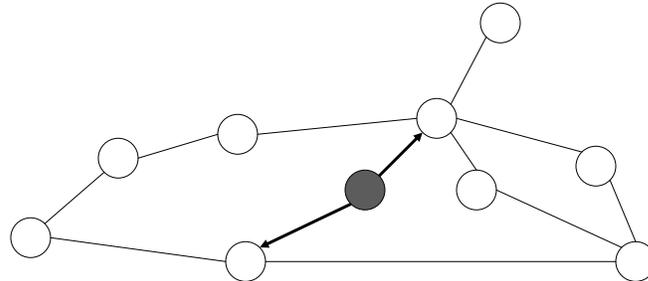
Sei $|M(G,m)|$ die Anzahl der benötigten Einzelnachrichten eines Multicasts m an Gruppe G im Graphen $V(v,e)$

$$|G| \text{ Unicasts } \quad |M(G,m)| = \sum_{g \in G} d(s,g)$$

Ziel $|M(G,m)| = O(e)$

Verteilte Systeme, Wintersemester 2000/2001 Folie 6.14

Ablauf (1): Start

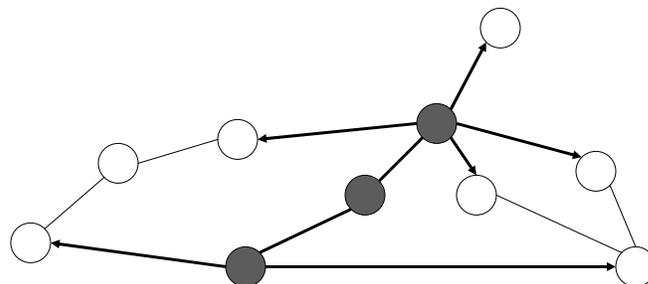


Versenden der Multicast-Nachricht $MC(G,m)$
Sender schickt Multicast über alle ausgehenden Kanten

Verteilte Systeme, Wintersemester 2000/2001

Folie 6.15

Ablauf (2): Ausbreitung



Zwischenknoten leiten Multicast ggf. an Empfänger weiter
Erneute Weitergabe der Nachricht über alle ausgehenden Kanten

Verteilte Systeme, Wintersemester 2000/2001

Folie 6.16

Ablauf (3): Reflektionen am Rand

Eine der beiden Nachrichten kommt zuerst an; andere ist ebenfalls Reflektion

Reflektionen des Multicasts am „Graphenrand“
 – Redundante Kanten

Verteilte Systeme, Wintersemester 2000/2001 Folie 6.17

Ablauf (4): Abfallprodukt „Spannbaum“

Verteilte Systeme, Wintersemester 2000/2001 Folie 6.18

Probleme

Überflutungsalgorithmen sind besonders gut für Broadcast-Nachrichten geeignet

- Gesamtnetz tatsächlich überfluten

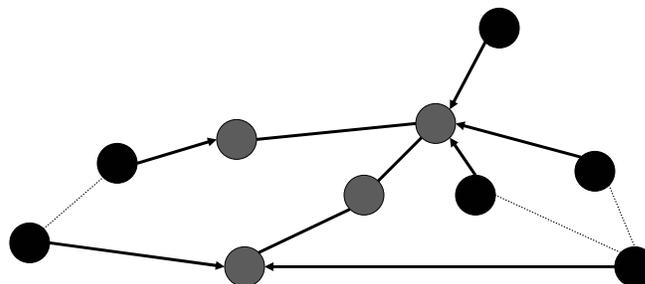
Allgemein: Wachsende Effizienzprobleme mit kleiner werdender Empfängergruppe

$$|G| \ll |V|$$

Lösungsansatz

- Knoten speichern Zusatzinformation, ob „darunterliegende“ Knoten Gruppenmitglieder beherbergen
- Bei hierarchischer Netzstruktur einfach
- Redundante Wege möglichst beibehalten

Echo-Algorithmus



Überflutungsalgorithmus mit Rückantwort

- Ausdehnungsphase analog zu Überflutung
- Kontraktionsphase

Echo-Nachrichten laufen entlang Spannbaum Richtung Sender

Echowelle Richtung Sender

Sinnvollerweise Anwendungsquittungen

- Blätter senden spezifische Rückantwort
- Innere Knoten können mehrere Rückantworten zusammenfassen und eine gemeinsame Antwort berechnen

Verteilte Systeme, Wintersemester 2000/2001 Folie 6.21

Amoeba-Multicast

Verteiltes Betriebssystem

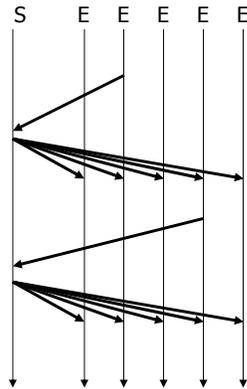
- Tanenbaum et al. 1981-, Universität Amsterdam
- RPC-basierte Kommunikation über Ports

Global zeitgeordneter Multicast

- Multicast nur innerhalb einer Multicast-Gruppe (Geschlossene Gruppe)

Verteilte Systeme, Wintersemester 2000/2001 Folie 6.22

Grundsätzlicher Ablauf



1. Multicast-Nachricht wird durch `SendToGroup()` an den Amoeba-Kern übergeben
2. Kern blockiert sendenden Prozeß
3. Kern leitet Nachricht mittels `RequestForMulticast()` an den aktuellen Sequencer weiter (RPC, Unicast)
4. Sequencer nummeriert Nachricht und sendet diese in der Nummerierungsreihenfolge als Multicast-Nachricht ab
5. Kern des sendenden Prozesses empfängt Multicast-Nachricht
6. Deblockade des Senders

Verteilte Systeme, Wintersemester 2000/2001

Folie 6.23

Bemerkungen

Total geordneter, k-zuverlässiger Multicast

Offene Fragen

- Nachrichtenverluste
- Sequencer-Abstürze
- Realisierung von k-Zuverlässigkeit
- Wann können empfangene Multicasts ausgeliefert werden

Effizienz

- Auf Broadcast-fähigen Medien schwer schlagbar

Verteilte Systeme, Wintersemester 2000/2001

Folie 6.24