Wir erfinden IP Multicasting

Felix von Leitner
Convergence
felix@convergence.de

Dezember 1999

Zusammenfassung

Multicast ist neben Unicast und Broadcast eine fundamentale Methode, wie man Daten versenden kann. Multicast ist *enabling technology* für die Konvergenz der Medien.

Agenda

1. Was ist Multicast? Wozu brauchen wir es?
2. Wie würde *ich* das spezifizieren?
3. Layer 2 Multicast (d.h. im Ethernet)
4. Welche Anwendungen sind denkbar?
5. Multicast im WAN
6. Dense Mode vs. Sparse Mode
7. Beschreibung der Routing-Protokolle

Was ist Multicast?

Multicast nennt man das Versenden des selben Datenstroms an mehrere Empfänger (genannt **Multicast-Gruppe**).

- RFC 966 (19851201): “Host groups: A multicast extensions to the Internet Protocol“
- RFC 1112 (19890801): „Host extensions for IP multicast“. (STD0005)

Wozu braucht man das?

- Ausstrahlungen
  1. Radio/Fernsehen
  2. „Push“-Dienste: Newsticker, Mailinglisten
- Replizierte Datenbestände
  1. Usenet
  2. Cache, Datenbank
- Virtuelles Ethernet
- Shared Whiteboard

Aber das alles gibt es doch schon?!

Von Multicast fordert man zusätzlich:

1. **Kein unnötigen Pakete**: Pakete sollen nur entlang solcher Routen verschickt werden, hinter denen mindestens ein Empfänger sitzt.

2. **Keine doppelten Pakete**: Jedes Paket soll jedes Kabel höchstens einmal durchwandern.

3. **Skalierbarkeit**: Soll mit zehn und zehn Millionen Empfängern gut funktionieren

Der Sender soll die Daten nur einmal abschicken müssen.

Komplexität beschränken

- Die Stellen, an denen man Know-How braucht, sollen minimiert werden, so daß für ein funktionierendes Gesamtsystem am besten nur an einer Stelle ein Profi sein muß, der versteht, was er tut.
- Am besten: Selbstorganisierendes System
- Früher: Nur wenige Server, viele Luser.
- Dann: Immer mehr Server.
- Zukunft: Jedem Luser seinen Server.

  Zusammenbruch?

Was tun?

1. Man macht es sehr einfach, einen Server zu benutzen?
2. Man verlangt einen „Führerschein“ von Server-Betreibern?

   Beobachtung: Invariante: $|\text{Router}| = O(\log(\text{Server}+\text{Clients}))$
3. Man verlagert die Komplexität in die Router!

Versandarten auf IP-Ebene

Name	Empfänger
Unicast	Einer
Multicast	Variabel
Broadcast	Alle
Anycast	Einer von n

Anycast ist im Moment nur für Router definiert, an einer Spezifikation für Hosts wird aber gearbeitet.

Anfängliche Überlegungen

Router entscheiden anhand der IP-Nummer. Also: Jede Multicast-Gruppe hat eine andere IP-Nummer.

Eine Transmission kann mehrere Komponenten haben, z.B. Audio und Video. Also sind Port-Nummern weiter sinnvoll.

Fehlermeldungen wie „port unreachable“ verhindern Skalierbarkeit.

Path MTU Discovery geht nicht. Also: Fragmentierung im Router.

Auch sonstige Rückmeldungen gehen nicht. Also: kein TCP, keine Fehlerkorrektur.

Umsetzung im Ethernet

Ethernet-Pakete haben eine Zieladresse.

Die Netzwerkkarten-Hardware filtert nach der Zieladresse.

Nur Pakete an die eigene oder die Broadcast-Adresse gelangen auf den Bus.

Wie kann man da multicasten?

1. Man aktiviert *promiscuous mode*
2. Multicasts gehen an die Broadcast-Adresse
3. Hardware-Filter mit mehr als einer Zieladresse

1. Ansatz: Multicast immer als Broadcast?

Gegenbeispiel: ein Gigabit-Ethernet voller Multicast-Traffic.

2. Ansatz: Promiscuous Mode?

Promiscuous Mode wird für Fehler-Diagnose benutzt. Er schaltet an der Ethernet-Karte das Hardware-Filtern aus, d.h. die Netzwerkkarte liefert *alle* Pakete an das Betriebssystem aus.

Promiscuous Mode sorgt auf nicht geswitchten Netzen mit viel Last auf schwachen Maschinen für 100% CPU-Last. Auf Servern wird der Platten-Durchsatz gesenkt, weil der PCI-Bus mit unnötigen Ethernet-Paketen belastet wird.

Daher Gegenbeispiel: Gigabit-Ethernet.

Das ist also auch keine gute Lösung, aber als Fallback-Lösung denkbar.

3. Ansatz: Hardware-Support?

Man könnte auch der Netzwerkkarte die Möglichkeit geben, nach mehr als einer Zieladresse zu filtern.

Nur: nach wie vielen? Rechner müssen an n Multicast-Gruppen teilnehmen können!

- Nach einer Liste von z.B. 16 MAC-Adressen filtern.
- Eine Hashtabelle mit z.B. 512 Einträgen. Wenn der Hash der Zieladresse eine 1 in dieser Bitmap hat, wird das Paket durchgelassen.

Beide Lösung sind unvollkommen. Das Betriebssystem muß auf jeden Fall nachfiltern, eventuell sogar in den *promiscuous mode* schalten.

Multicast-Router im Ethernet

Router müssen *alle* Multicast-Pakete erhalten!

Es muß also einfach in Hardware möglich sein, alle Multicast-Pakete herauszufiltern. Weder Liste noch Hashtabelle taugen dafür.

Man könnte z.B. den MAC-Adressen für Multicast einen gemeinsamen Prefix geben.

Zusätzlich möchte man noch *Domänen* definieren können, damit lokale Ausstrahlungen nicht nach außen geleitet werden. Der Router sollte die lokalen Ausstrahlungen am besten in Hardware erkennen können.

Woher kommt die MAC-Adresse?

Multicast-Gruppen haben IP-Nummern aus dem Bereich **224.0.0.0/4** (28 signifikante Bits). Multicast MAC-Adressen beginnen mit dem Prefix **01 00 5e** (23 signifikante Bits).

Der Bereich **224.0.0.0/24** ist für das lokale Netz reserviert und wird von Routern nicht geforwarded. **224.0.0.1** beinhaltet alle Hosts, **224.0.0.2** beinhaltet alle Router.

Man gewinnt die MAC-Adresse algorithmisch aus der IP-Nummer. Das Mapping ist aber nicht eindeutig.

Das Betriebssystem muß also auch mit nicht erwünschten Paketen rechnen und nachfiltern.

Netzwerk-Equipment und Multicast

Hubs schauen sich die Pakete nicht an und sind daher neutral.

Alte Switches leiten Multicast-Pakete an alle angeschlossenen Stränge weiter. Neuere Switches implementieren *IGMP Snooping* und können damit optimieren.

IGMP ist das Protokoll, mit dem Hosts dem Router sagen können, an welchen Multicast-Groups sie teilnehmen möchten. Im LAN ist das noch nicht relevant, aber in WANs oder eben für moderne Switches.

Internet Group Management Protocol (IGMP)

IGMP wurde im Jahre 1988 im RFC 1112 definiert. Es ist ICMP nachempfunden und sitzt direkt auf IP auf.

Version (bits 0-3)	Type (4-7)	Code (8-15)	Checksum (16-31)
IP			

IGMP kennt nur zwei Arten von Paketen:

1. Host Membership Query

2. Host Membership Report

Router fragen periodisch die *all-hosts* Gruppe **224.0.0.1** nach den gewünschten Gruppen, Hosts können auch ungefragt Reports an die *all-routers* Gruppe **224.0.0.2** schicken, wenn ein Prozess sich irgendwo einschreibt.

In neueren IGMP-Versionen einigen sich zwei Router im Ethernet, so daß nur einer zuständig ist pro Gruppe.

Was für Daten kann ich übertragen?

Multicast ist unidirektional!

- keine Rückmeldungen
- kein TCP!
  - keine Fehlerkorrektur
  - keine *congestion control*
  - Pakete können doppelt ankommen
  - Pakete können gar nicht ankommen
  - Pakete können in der falschen Reihenfolge ankommen

Reliable Multicast ist ein Forschungsgebiet. Im Moment gibt es noch keine tragfähigen, skalierbaren Konzepte.

Was kann man damit machen?

IPv6 demonstriert eindrucksvoll die Schönheit von Multicast im LAN. Die lokalen Router finden sich gegenseitig über eine fest definierte Multicast-Group mit lokalem Scope.

Hosts finden den lokalen Router, indem sie die *all-routers* Multicast-Gruppe pingen.

Uhrenabgleich läuft über eine Multicast-Gruppe.

Weitergehende Ideen: Cache-Abgleich, Updates von NIS-Tabellen o.ä.

Multicast und Router

Wem teile ich mit, daß ich an einer Multicast-Gruppe teilnehmen möchte?

Mögliche Modelle:

1. Man subscribed sich beim Sender.

2. Man subscribed sich beim lokalen Router.

Subscription beim Sender

1. Der Sender muß eindeutig und bekannt sein!
2. Der Sender hat ein Skalierungsproblem!
3. Der Sender muß alle Empfänger in die Pakete reinschreiben
4. Zugangsbeschränkung und Billing sind relativ einfach möglich

Subscription beim lokalen Router

1. Es kann mehrere Sender geben
2. Jeder Dialup-User kann Radio ausstrahlen
3. der Sender weiß nicht, wer alles zuhört!
4. Zugangsbeschränkung ist nur über Verschlüsselung möglich

Wie funktioniert das im WAN?

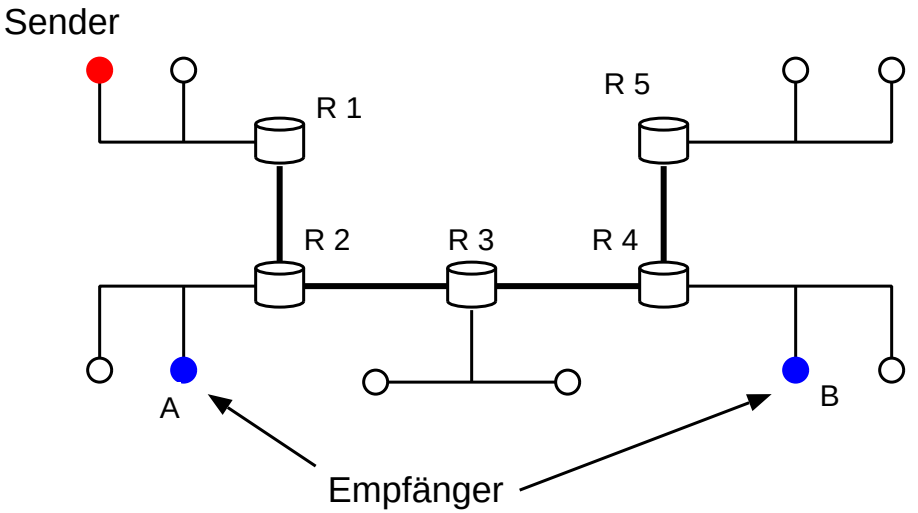
Grundsätzlich funktioniert das Routing bei Multicast in umgekehrter Richtung, d.h. vom Empfänger zum Sender. Bei Multicast-Gruppen gibt es aber keinen eindeutigen Empfänger. Daher verlassen sich aktuelle Routing-Protokolle darauf, daß sie für jede Gruppe wissen oder herausfinden können, wo der Strom momentan herkommt.

Ein Host signalisiert dem Router per IGMP, wenn er einer Multicast-Gruppe beitreten oder sie verlassen will.

Multicast-Routing ist im Moment noch ein aktives Forschungsgebiet.

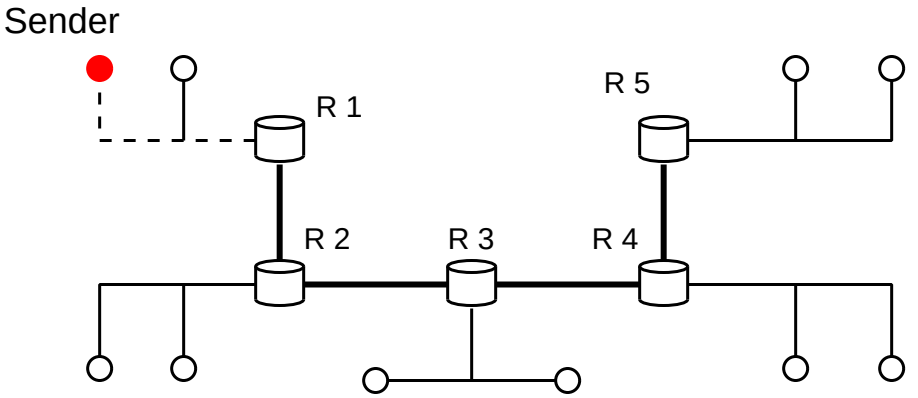
Es gibt keine Fehlermeldungen bei Multicast-Transmissionen!

Welche Probleme löst Multicast Routing?



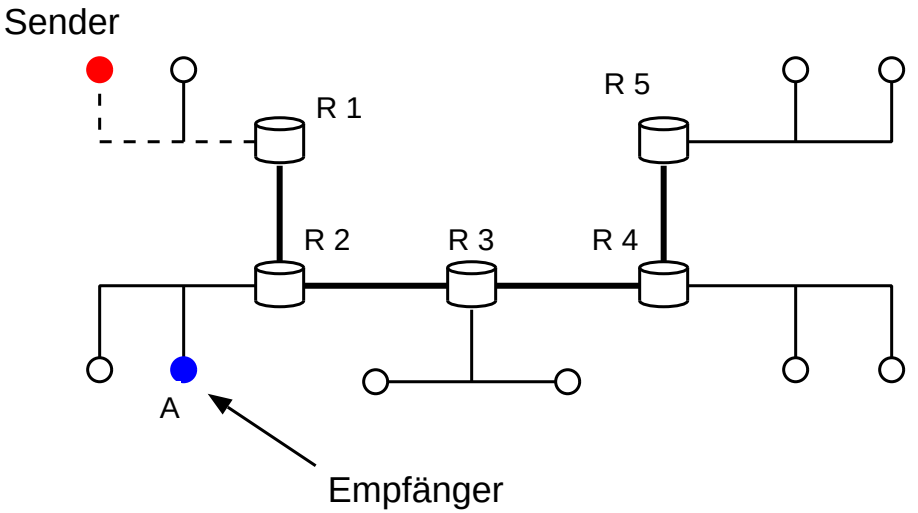
- Wie kommen die Daten zu allen Empfängern?
- Wie verschwendet man möglichst wenig Bandbreite dabei?

Stadien eines Multicasts



Solange niemand der Gruppe beitritt, kommen die Pakete bis zum lokalen Router, R1.

Jemand tritt bei



Woher weiß R2, wie es weitergeht?

Strategien für Multicast Routing

- Jeder Router kennt für jede Multicast-Gruppe einen Router, der den Traffic führt?
- Designierte Core-Router kennen alles, andere Router folgen der *default*-Route?
- Erst Fluten, dann Einschränken
- Die Router unterhalten einen *Spanning Tree*

Grundsätzliche Ansätze für Multicast-Routing

Man teilt Algorithmen für das Multicast-Routing in zwei Ansätze auf:

1. *Dense Mode*, d.h. viele Empfänger
2. *Sparse Mode*, d.h. einige wenige Netze enthalten Empfänger

Die Unterscheidung bezieht sich nicht auf die Anzahl der Empfänger pro Netz, sondern auf den Quotienten

$$\frac{\text{Netze mit Empfängern}}{\text{Netze insgesamt}}$$

Dense Mode Routing

Dense Mode Protokolle gehen gewöhnlich davon aus, daß viel Bandbreite verfügbar ist und eignen sich für LANs und Campusnetze. Sie fluten periodisch das Netzwerk, um einen optimalen *spanning tree* zu bestimmen.

1. Distance Vector Multicast Routing Protocol (DVMRP)
2. Multicast Open Shortest Path First (MOSPF)
3. Protocol-Independent Multicast – Dense Mode (PIM-DM)

Diese Protokolle bauen einen Baum auf, wenn der Sender Daten verschickt.

Distance Vector Multicast Routing Protocol

DVMRP war das erste Multicast-Routingprotokoll (RFC 1075). Es ist im freien `mrouted` implementiert und wird noch heute im MBone benutzt.

1. Ein Multicast-Paket kommt an.
2. Akzeptiere Pakete nur von dem Interface, auf dem die Unicast-Route zum Sender liegt. Das verhindert Zyklen und hält Pfade kurz.
3. Leite das Paket an alle anderen benachbarten Router weiter.
4. Vermerke für diese Multicast-Gruppe die Liste der benachbarten Router.

Das Vorgehen bisher nennt man *Reverse Path Forwarding*.

Per IGMP stellt der Router die benachbarten Interfaces fest, hinter denen keine Interessenten sitzen.

Zu alte Einträge in der Routing-Tabelle werden gelöscht und führen zu erneutem Fluten.

Die Metrik für Routen ist die Anzahl der Hops.

Wenn ein Router feststellt, daß er keine Interessenten mehr hat, meldet er das per IGMP dem Quell-Router für diese Gruppe.

Zusätzliche Ersparnis: pro Interface speichert DVMRP, ob man dahinter als upstream erkannt wird.

DVMRP verschwendet Bandbreite und die Router müssen viel Zustand halten.

Multicast Open Shortest Path First

MOSPF (RFC 1584) setzt das Unicast Routingprotokoll OSPF voraus. Bei OSPF führt jeder Router eine Baumdarstellung des ganzen Netzwerks.

Aus diesem Baum werden mit dem Dijstra-Algorithmus die optimalen Routen extrahiert, wenn das erste Paket eines Senders zu einer Multicast-Gruppe empfangen wird.

MOSPF sammelt periodisch per IGMP Mitgliedschaften.

Der komplette Baum wird zwischen den Routern ausgetauscht.

MOSPF verschwendet Bandbreite, braucht sehr viel State pro Router, und braucht viel CPU-Zeit bei größeren Netzen.

Protocol Independent Multicast – Dense Mode

- Wird von der IETF entwickelt.
- Unabhängig vom Unicast-Routingprotokoll
- Benutzt wie DVMRP Reverse Path Forwarding

PIM-DM ähnelt DVMRP ohne die Upstream-Ersparnis. Die Ingenieuere wollten lieber ein einfaches Protokoll, das unabhängig vom Unicast-Routing ist.

PIM-DM erzeugt mehr überflüssigen Traffic als DVMRP.

Sparse Mode Routing

Sparse Mode Protokolle gehen von dünnen Leitungen und wenigen einzelnen Empfängern aus, d.h. von dem Internet. Fluten kommt nicht in Frage.

1. Core Based Trees (CBT)

2. Protocol-Independent Multicast – Sparse Mode (PIM-SM)

Diese Protokolle bauen einen Baum auf, wenn sich jemand einschreibt, nicht wenn jemand sendet.

Core Based Trees

- Zentraler „Core“-Router!
- gemeinsamer *spanning tree*
- Verteilung immer über diesen Baum
- Verteilung unabhängig vom Sender

Router schreiben sich ein, indem sie einen Request zum Core-Router schicken. Die Anfrage kann schon vorher von einem Router abgefangen und beantwortet werden.

Manche CBT-Versionen erlauben mehrere Core-Router.

Der Traffic akkumuliert sich um die Core-Router. Load-Probleme. Wer bezahlt den Core-Router?

Protocol Independent Multicast – Sparse Mode

PIM-SM (RFC 2117) spricht von *Rendezvous Points* statt von Core Routern.

Router können bei PIM-SM einen gemeinsamen Baum haben oder auch einen Baum der kürzesten Pfade.

Initial wird ein gemeinsamer Baum aufgebaut. Bei viel Traffic kann der Empfänger eine PIM JOIN Nachricht an dem Sender schicken, und so den kürzesten Pfad mitgeteilt bekommen.

Probleme

Die Anzahl der Hops ist kein gutes Maß für Routing-Protokolle (Tunnel!).

Die verschiedenen Protokolle arbeiten nicht zusammen.

Die PIM Designer schlagen Border Router vor. Border Router sprechen untereinander Sparse Mode Protokolle. Dahinter wird Multicast im Dense Mode geroutet.

Wie benutze ich es?

```
fd = socket(AF_INET,SOCK_DRAM,0);
setsockopt(fd,SOL_SOCKET,SO_REUSEADDR,
  &eins,sizeof(int));
setsockopt(fd,IPPROTO_IP,
  IP_MULTICAST_TTL,&ttl,sizeof(char));
setsockopt(fd,IPPROTO_IP,
  IP_MULTICAST_LOOP,&loop,sizeof(char));
struct ip_mreq blub;
blub.imr_multiaddr.s_addr = inet_addr("224.1.2.3");
blub.imr_interface.s_addr = INADDR_ANY;
setsockopt(socket,IPPROTO_IP,
  IP_ADD_MEMBERSHIP,&blub,sizeof(blub));
```

Zusammenfassung

Es gibt viel zu tun!

Multicast-Routing kann momentan noch nicht als gelöst bezeichnet werden.

IGMPv3 ist spezifiziert, benutzt wird größtenteils noch IGMPv1.

Eine Teilnahme am MBone ist im kommerziellen Internet schwierig, da die ISPs sie nicht haben. Hausnummer: 7 mbps Traffic nur für MBone-Verwaltung.

Danke für die Aufmerksamkeit!

Fragen?