

Source Routed Multicast

Ein Multicast-Routing-Header für IP Version 6

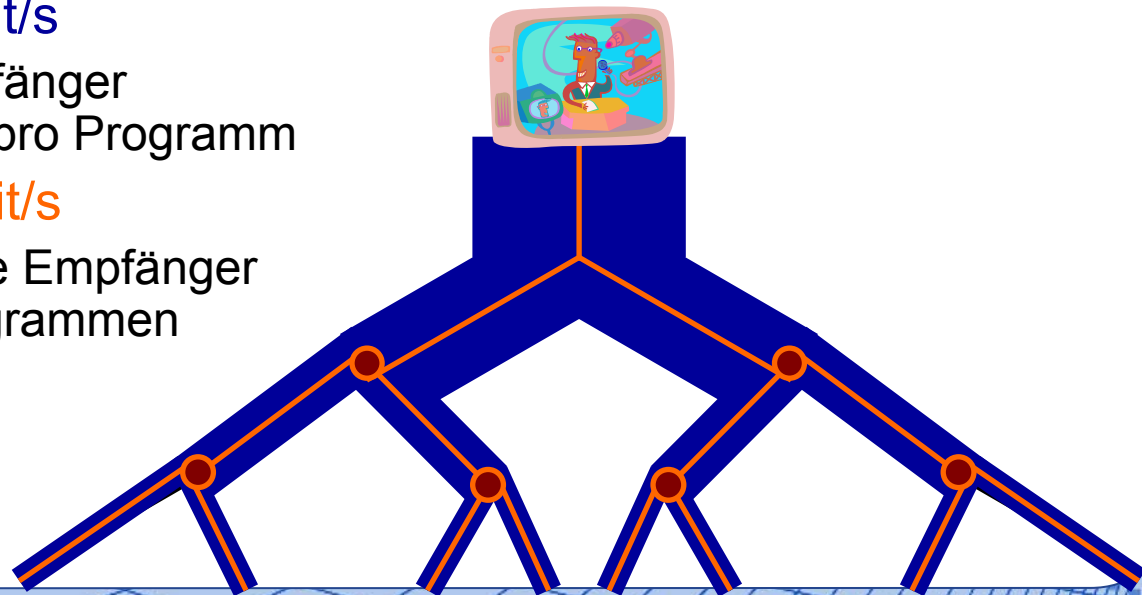


Mark Doll

Institut für Telematik, Prof. Dr. M. Zitterbart
Universität Karlsruhe (TH)



- Warum Multicast?
- Multicast spart Bandbreite!
 - konstanter Bandbreitenbedarf unabhängig von Empfängerzahl
 - ermöglicht beliebige Gruppengrößen
- Anwendungsfall 1: sehr große Gruppen
 - Beispiel IPTV: Quelle benötigt
 - Unicast: **40 Gbit/s**
 - ▶ 10 000 Empfänger
bei 4 Mbit/s pro Programm
 - Multicast: **1 Gbit/s**
 - ▶ beliebig viele Empfänger
bei 250 Programmen



Anwendungsfall 2: schmalbandige Anbindung

- Unicast

- höchste Verkehrskonzentration bei Quelle
- verfügbare Bandbreite limitiert Empfängeranzahl

- Multicast

- beliebig viele Empfänger
gleichzeitig bessere Qualität

Beispiel Videokonferenz

- DSL-16000 (1024 kbit/s Upstream)
- Unicast: 3 Teilnehmer (=2 Streams) à 512 kbit/s
- Multicast: beliebig viele Teilnehmer à **1024 kbit/s**

Source Routed Multicast

Zielanwendung **schmalbandige Anbindung**

Warum? Vermeidet Per-Flow-Zustand!

IP-Multicast

- Warum reicht nicht IP-Multicast?
 - skaliert hervorragend (konstant) bzgl. Gruppengröße
- benötigt **Zustand per-flow!**
 - je aktivem Quelle-Gruppe-Tupel ein Zustand in jedem Router des Verteilbaums
 - **skaliert schlecht** (linear) bzgl. **gleichzeitig aktiver** Gruppen
- ideal für wenige sehr große Gruppen
- ungeeignet für sehr viele (kleine) Gruppen

Source Routed Multicast
beliebig viele gleichzeitig aktive Gruppen

Alternative Lösungen

- alle verlagern Per-Flow-Zustand aus den Routern
- in die Anwendungsschicht
 - Application Layer Multicast (NARADA, NICE,...)
 - Overlay Multicast (OMNI, TOMA,...)
- in das IP-Paket
 - Explicit Multicast (XCast, Somecast,...)
 - Multicast for Small Conferences
 - **Source Routed Multicast**
- Warum noch ein Verfahren?

- Application Layer Multicast
 - Teilnehmerdynamik verursacht Instabilitäten
 - längere Pfade, größere & schwankende Ende-zu-Ende-Verzögerung → Dienstgüte problematisch
- Overlay Multicast
 - erfordert dedizierte Systeme im Netzinneren
- Explicit Multicast
 - N Routinglookups je Paket in *jedem* Router auf Pfad
 - ▶ bei N Empfängern im noch folgenden Teilbaum
 - hoher Weiterleitungsaufwand, dafür optimale Dienstgüte
- Multicast for Small Conferences (MSC)
 - hoher Weiterleitungsaufwand, immerhin nur MSC-Router
 - ▶ Unicast überbrückt Nicht-Verzweigungsrouter
 - Tradeoff Dienstgüte (Verzögerung) vs. Duplikatanzahl
 - ▶ lokale Policy des MSC-Routers
 - nicht eindeutige Weiterleitung = unnötige Komplexität

Source Routed Multicast

- Unicast zwischen Knoten im Verteilbaum
- abschließend nativer IP-Multicast
- IP-Paket trägt
 - *vollständigen* Verteilbaum (Source Routing)
 - ▶ Adressen
 - ▶ Vater-Kind-Beziehung (per Zeigerinkrement)
 - Zeiger auf aktuelle Position im Verteilbaum
 - verbleibende Anzahl Duplikate (Segments Left)
- dupliziertes IP-Paket erhält
 - neue **Zieladresse** gemäß Verteilbaum (eindeutig)
 - aktualisierte Werte für **Positionszeiger** & **Segments Left**
- keine weiteren Modifikationen am IP-Paket!

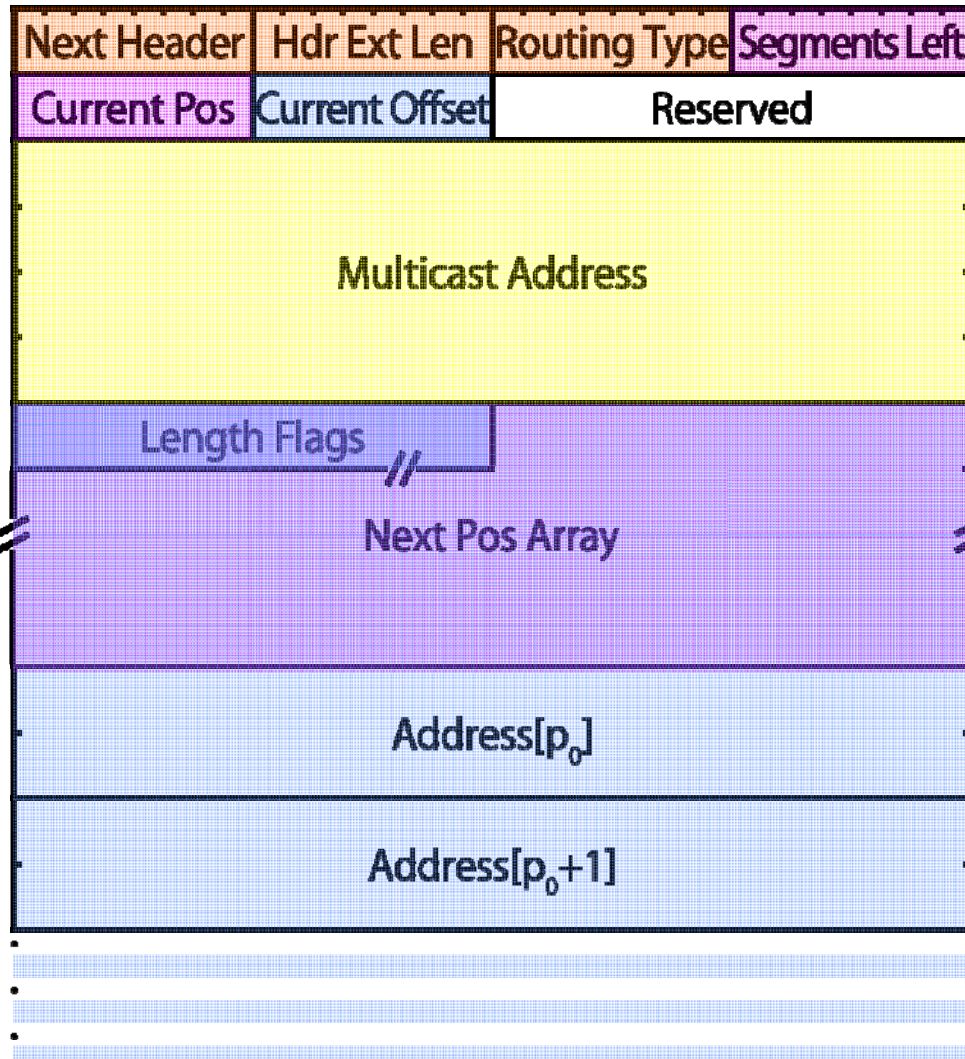
Source Routed Multicast
einfache und **eindeutige** Paketduplizierung

Source Routed Multicast

- definiert
 - IPv6 Multicast Routing Header
 - ▶ nicht IPv4, da IPv4-Kopf auf 60 Oktette begrenzt
 - Weiterleitungsalgorithmus
- überlässt der Quelle
 - Auffinden der Empfänger
 - Aufbau des Verteilbaums
 - ▶ Ermittlung geeigneter Verzweigungsrouter
 - ▶ Tradeoff zwischen Overhead und Ende-zu-Ende-Verzögerung/Dienstgüte

Source Routed Multicast
verlagert **Komplexität in die Quelle**

IPv6 Multicast Routing Header Platzeffiziente Kodierung



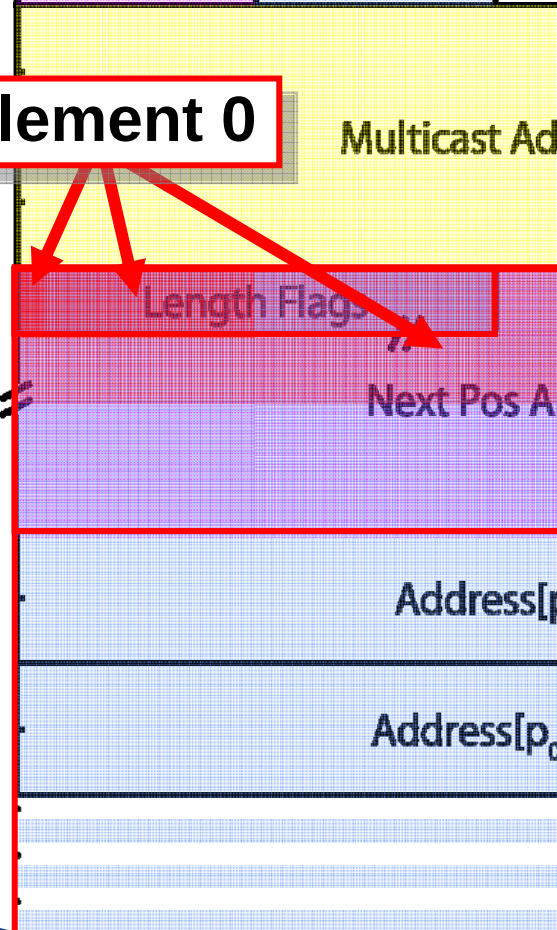
- Standardfelder
 - *Next Header*
 - *Hdr Ext Len*
 - *Routing Type – Multicast*
- entgültige Zieladresse
 - *Multicast Address*
- Baumstruktur/Kindlisten
 - *Next Pos[p]* (1 Oktett)
 - *Current Pos – Start*
 - *Segments Left – Länge*
- Knotenadressen
 - *Address[p]* (8 Oktette)
 - *Length Flag[p]* (1 Bit)
 - ▶ 64/128 bit-Adressen
 - *Current Offset*
 - ▶ Anzahl gesetzter LengthFlag[0..CurrPos]
 - ▶ Optimierung

IPv6 Multicast Routing Header Platzeffiziente Kodierung

Next Header	Hdr Ext Len	Routing Type	Segments Left
Current Pos	Current Offset		

• Standardfelder

Element 0

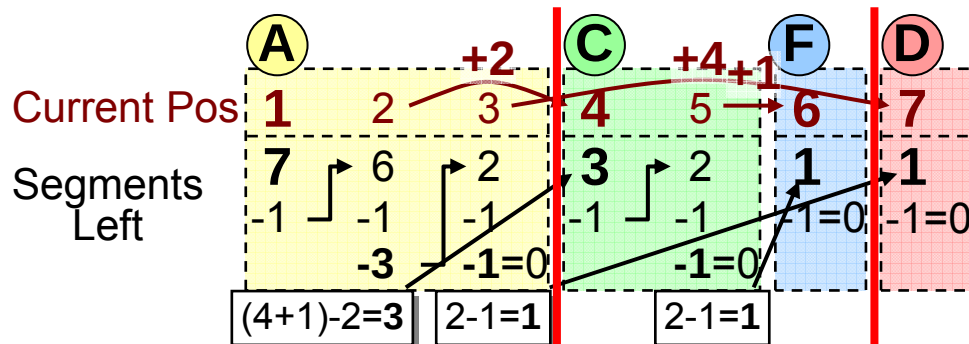
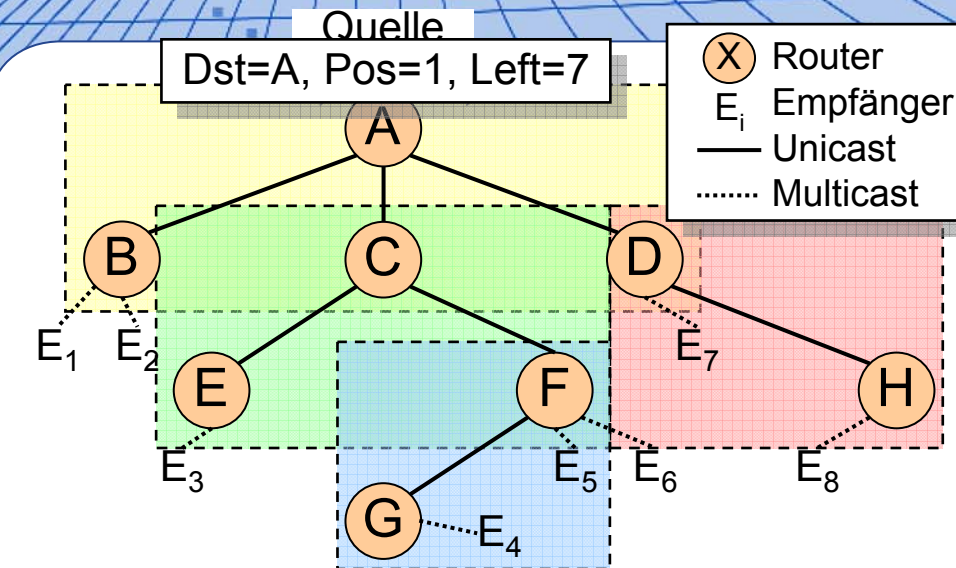


Überlappende Arrays!

- Vorteil: bekannter Startpunkt
 - immer direkt hinter Multicast Address
 - Startpunkt muss nicht vermerkt werden
 - keine Zeigerarithmetik $\text{Current Pos} + \text{Start}$
- Nachteil: erste $\lceil N/7 \rceil$ Elemente nicht brauchbar
 - bei Verteilbaum mit N Knoten
 - erstes gültiges Element $p_0 = \lceil N/7 \rceil$
 - Grund: Größenverhältnis 1:8
 - ▶ Length Flag : Next Pos = 1:8
 - ▶ Next Pos : Address = 1:8
- trotzdem nur max. 6 ungenutzte Oktette!
 - genau: $6 - N \bmod 7$ Oktette
 - letzte Elemente Next Pos[i] vor Address[p₀]

→ **effizient beim Zugriff**
effizient beim Platzbedarf

Weiterleitungsalgorithmus I: Next Pos Array Relative Zeiger & sortierte Kinderliste



10

Next Pos	0	+2	+4	0	+1	0	0
Address	B	C	D	E	F	G	H

A Beginn Teilbaum 1. Kind begrenzt Kindknotenliste

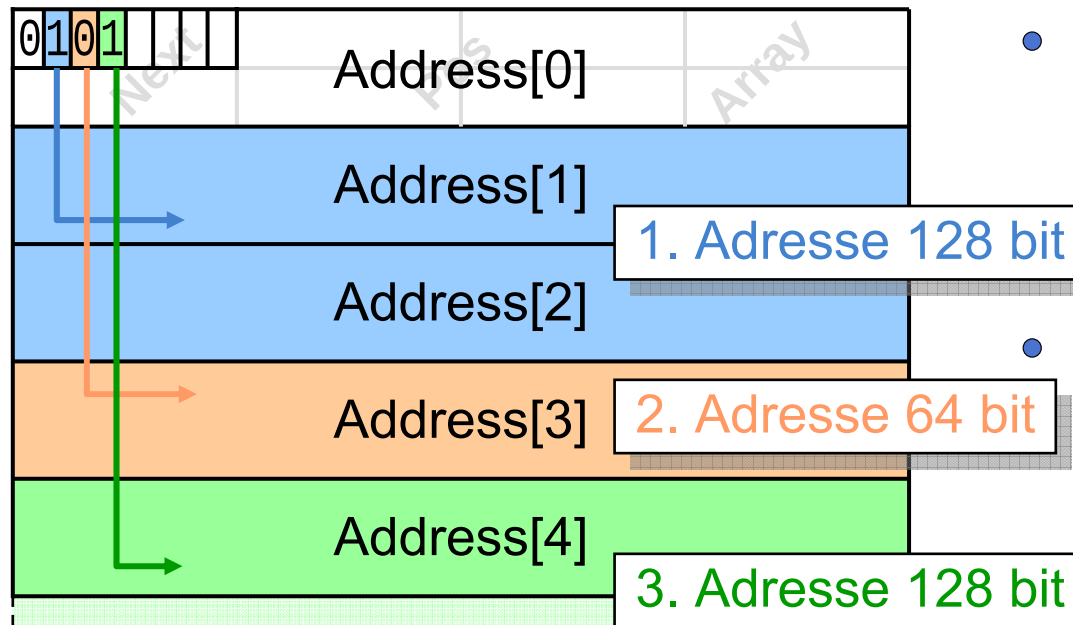
F alle Kinder Blattknoten: Knotenliste begrenzt durch Segments Left=0

- Next Pos Array
 - zuerst Blattknoten
 - danach *vollständige* Teilbäume
 - *sortiert* nach Startindex
 - ein Element = ein Kind/Duplikat
- Current Pos = Startindex
- sende Duplikat an Kind
 - Zieladresse $\leftarrow$ p-te Adresse (vereinfacht: Address[p])
 - Current Pos $\leftarrow$ p + Next Pos[p]
 - setze Segments Left
 - ▶ 0, falls Next Pos[p] = 0
 - ▶ sonst Differenz zu Current Pos für folgendes Kind
 - ▶ für letzten Teilbaum Segments Left - 1
- Segments Left = 0
 - Current Pos $\neq$ 0
 - ▶ Zieladresse $\leftarrow$ Multicast Address
 - ▶ Current Pos $\leftarrow$ 0
 - Current Pos = 0
 - ▶ überspringe Routing Header
 - ▶ IP-Multicast

- Problem: Overhead
 - zusätzlich Adressen aller Verzweigungsrouter im IP-Paket
 - ▶ Worst Case: ein Verzweigungsrouter je Blattrouter
 - ▶ bis **doppelt so viele Adressen** verglichen mit XCast/MSD
 - Idee: verwende Subnet Router Anycast Address
 - *jeder* Router muss über sie erreichbar sein [RFC4291]
 - Bildungsregel: setze HostID = Null
 - ▶ z. B. Subnetz 2001:DB8:1234::/64 → Adresse 2001:DB8:1234::
 - /64 ist typische Subnetzgröße
 - 8 Oktette genügen zur Speicherung
 - restliche 8 Oktette werden mit Null aufgefüllt
- **halbiert Platzbedarf** je Adresse

Weiterleitungsalgorithmus II: 128 bit- oder 64 bit-Adressen

- Address Array
 - speichert Adressen aller Knoten im Verteilbaum
 - Arrayelemente konstant 64 bit lang
 - Address[p] speichert erste 64 bit (MSBs) einer Knotenadresse
 - Unterscheidung 64 bit oder 128 bit mittels
- Length Flag



- Length Flag[p] = 1
Address[p+1] enthält
folgende 64 bit (LSBs)

- Length Flag[p] = 0
fülle mit Nullen
auf 128 bit auf

- Problem
 - Zahl gesetzter Length Flag vor Current Pos
 - bestimmt Index einer Adresse im Address Array
- Lösung: Current Offset
 - Current Offset = gesetzte Length Flag[0..Current Pos-1]
 - ▶ Adresse in Address[Current Pos+Current Offset]
 - erspart Zählen der gesetzten Length Flag – fast...
- Berechnung Current Offset für Duplikat an Kindknoten
 - erfordert Zählen der gesetzten Length Flags
 - in übersprungenen Teilbäumen der Geschwisterknoten
 - **linearer Aufwand** mit Größe des übersprungenen Teilbaums

Varianten mit konstantem Aufwand

- „4+4“
 - Elemente des Next Pos Arrays zweiteilen
 - ▶ 4 bit Zeigeroffset zum Kind (statt bisher 8 bit)
 - ▶ 4 bit Anzahl gesetzter Length Flag im übersprungenen Teilbaum
 - Problem: reduziert **max. Teilbaumgröße** auf **15 Knoten**
 - ▶ ab 32 Knoten nicht mehr alle möglichen Bäume darstellbar
- „POPCOUNT“
 - POPCOUNT zählt gesetzte Bits in einem Wort
 - Problem: **Befehl nicht immer verfügbar**
 - Current Offset überflüssig
 - ▶ $\text{CurrentOffset} = \text{POPCOUNT}(\text{LengthFlag}[0..\text{CurrentPos}-1])$

Maximale Größe des Verteilbaums

- theoretisch
 - 110 128 bit- oder 221 64 bit-Adressen
 - begrenzt durch Ext Hdr Len (2048 Oktette)
- praktisch
 - IP-Paket mit 1280 Oktetten und 50 % Overhead
 - max. 31 128 bit- oder 62 64 bit-Adressen
- typisch
 - 2/3 64 bit-Adressen
 - ▶ alle Empfängernetze /64er
 - ▶ 1/3 /64-Transitnetze im Internet (Stichprobe mit Traceroute)
 - ▶ ein Verzweigungsrouter je Blattknoten (Worst Case)
- 50 % Overhead: ca. 54 Knoten = mind. 27 Blattknoten

Source Routed Multicast
Klassenraumgröße als typische Obergrenze

- inkrementeller Einsatz
 - Legacy-Router antworten Quelle mit
 - ▶ ICMP Parameter Problem Code 0
 - ▶ mit Verweis auf Feld Routing Type im Routing Header
 - ▶ Quelle kann anderen Verzweigungsrouter wählen
- administrative Nutzungskontrolle/-beschränkung
 - Segments Left begrenzt Baumgröße
 - ▶ Segments Left gibt zu jedem Zeitpunkt Anzahl noch zu erzeugender Unicast-Duplikate an
 - Duplikatanzahl durch abschließenden IP-Multicast unbekannt
 - ▶ grundsätzliches Problem von IP-Multicast
 - ▶ ggf. Einschränkung auf Link oder Interface Local Scope

Source Routed Multicast für IP Version 6

- Multicast Routing für kleine Gruppen
 - bis zu einige Dutzend Empfänger
 - für schmalbandige Anbindung
 - optimale Ende-zu-Ende-Verzögerung & **Dienstgüte** möglich
- einfacher Weiterleitungsalgorithmus für Router
 - **konstanter Aufwand** je dupliziertem IP-Paket
 - Nutzungskontrolle/-beschränkung über Segments Left
- verlagert **Komplexität** aus den Routern **in die Quelle**
 - Aufbau und Wartung des Verteilbaums
 - Auffinden von Empfängern

„Keep it simple, stupid!“

Zusatzfolien

18



- erster Empfänger
 - als Blattknoten im Routing Header eintragen, fertig
 - anschließend alle passierten Router bestimmen
 - ▶ mittels Traceroute (Testpakete mit kleinem Hop Limit)
 - ▶ Voraussetzung zum Hinzufügen weiterer Empfänger
- Hinzufügen eines weiteren Empfängers
 - Traceroute zum neuen Empfänger mit Intervallhalbierung des Hop Limits
 - findet schnell letzten gemeinsamen Router (logarithmischer Aufwand)
 - gefundenen Router als Verzweigungsknoten und
 - Empfänger als Blattknoten in Routing Header eintragen, fertig
 - anschließend die übrigen passierten Router ermitteln (Traceroute)
- Entfernen von Empfängern
 - Empfänger austragen und
 - falls Empfänger vorletztes Kind des Vaterknotens, dessen anderes Kind an dessen Vater hängen und zusätzlich Vaterknoten austragen, fertig

- Änderungen im Unicastrouting
 - periodisches Neubestimmen aller passierten Router
 - ▶ Traceroutes zu allen Empfängern
- Ausfall von Knoten
 - Erkennung durch ICMP Destination Unreachable
 - hänge Kinder des Knotens an dessen Vater, fertig
 - anschließend Optimierung
 - ▶ Router zwischen dessen Vater und Kindern neu bestimmen und besseren Verzweigungsrouter wählen
 - Ausfall Knoten mit Subnet Router Anycast Adresse
 - ▶ anderer Router im Subnetz übernimmt automatisch
 - ▶ Pfade zu den Kindern aber nicht mehr optimal
 - ▶ Erkennung und erneute Optimierung im Rahmen der Überprüfung auf Routingänderungen