The Hybrid Shared Tree Architecture

Matthias Wählisch[1,2]   Thomas C. Schmidt[1]
{t.schmidt, waehlisch}@ieee.org

[1]HAW Hamburg & [2]link-lab

69th IETF Metting, 2007

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

link-lab

Agenda

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

link-lab

Motivation

There are . . .

- ▶ discrepancies between intra- and inter-domain deployment
  - ▶ Use of hybrid overlay multicast approaches
    (draft-irtf-sam-hybrid-overlay-framework-01.txt)
- ▶ DHT-based routing schemes
  - ▶ Typical: *hash*(*groupaddress*) defines rendezvous point +
    routing like PIM-SM
  - ▶ SCRIBE distribution tree build on RP: triangular routing
- ▶ problems with efficient multicast mobility
  - ▶ Multicast mobility PS: draft-irtf-mobopts-mmcastv6-ps-01.txt
 - ▶ Mobility agnostic routing with Bi-directional PIM

The Hybrid Shared Tree Architecture

- Complement to draft-irtf-sam-hybrid-overlay-framework
- Introduce Inter–domain Multicast Gateways (IMGs)
  - Similar to Peers
  - Provide gateway functions
  - Reside between overlay and intra–domain underlay
  - Interconnect local multicast with distributed overlay peering
- Network layer multicast unchanged in end system domains
- Overlay network based on well established DHT, equipped with a new overlay routing scheme
  - Distribution tree independent of source location
  - Homogenously efficient forwarding, no RPs
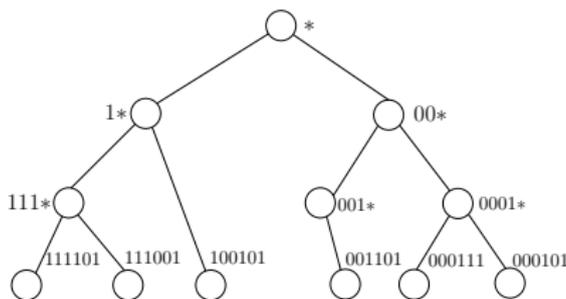- Use Pastry due to its proximity-awareness and prefix table

Architectural Overview
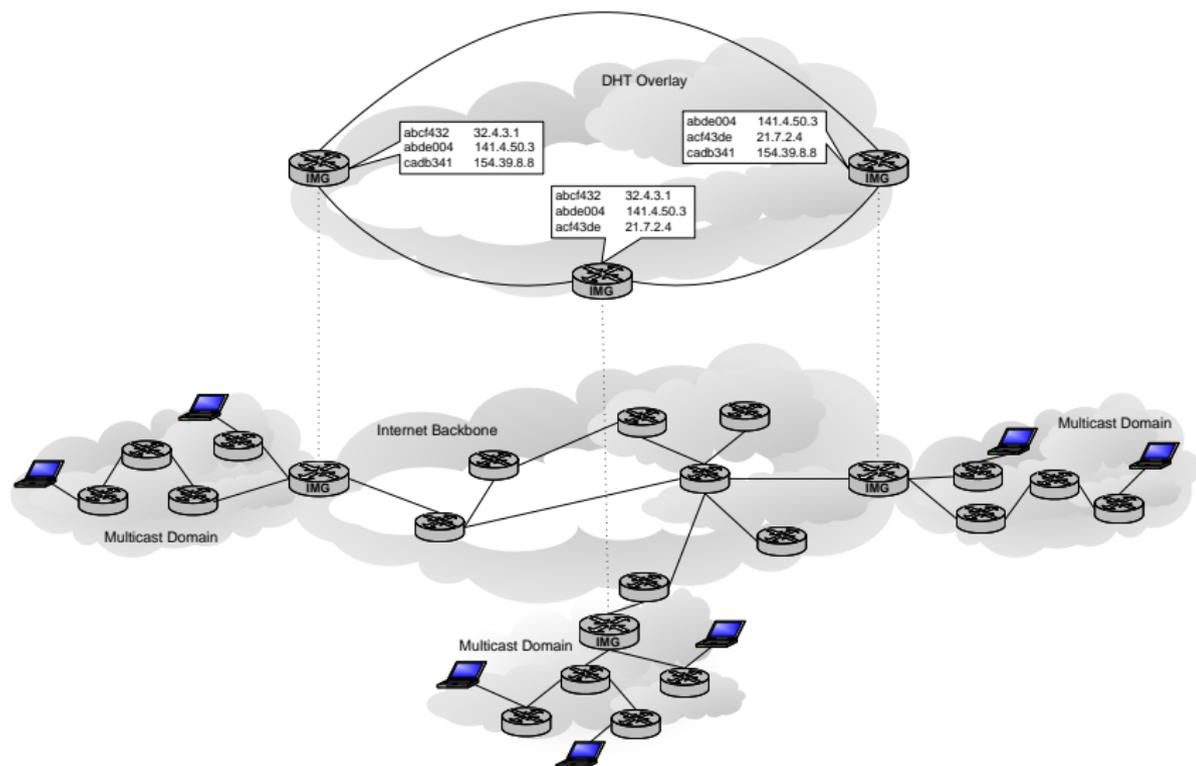
Constructing the Overlay Distribution Tree

- ▶ Every IMG has an overlay address: *hash(IMG ID)*
- ▶ IMGs learn about all group memberships
  - ▶ Membership updates are communicated incrementally
- ▶ Each IMG constructs a groupwise common prefix tree
  - ▶ IMGs of multicast receiver domains represent the leaves
  - ▶ Inner vertices correspond to longest common prefixes
  - ▶ Vertices on path to root share prefix with node
  - ▶ Tree will be used as bi-directional shared tree

Constructing the Overlay Distribution Tree

- ▶ Every IMG has an overlay address: *hash*(*IMG ID*)
- ▶ IMGs learn about all group memberships
  - ▶ Membership updates are communicated incrementally
- ▶ Each IMG constructs a groupwise common prefix tree
  - ▶ IMGs of multicast receiver domains represent the leaves
  - ▶ Inner vertices correspond to longest common prefixes
  - ▶ Vertices on path to root share prefix with node
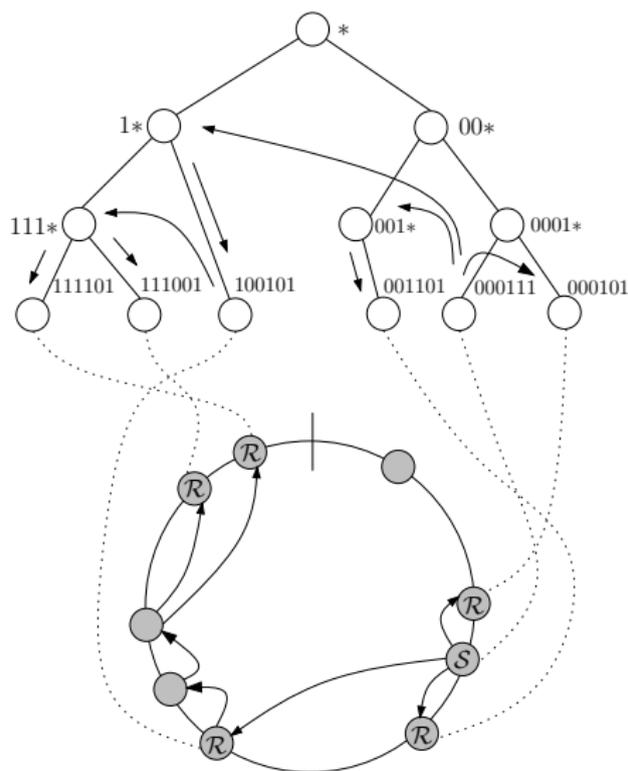 - ▶ Tree will be used as bi-directional shared tree
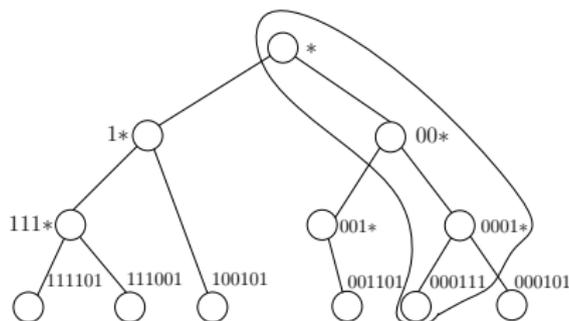
Routing within the Overlay Distribution Tree

- Prefix tree is a routing overlay to the DHT
- Source IMG determines its position on the tree
  - Longest common prefix
- Multicast traffic distributed to prefix neighbors
  - Only downward flow
- Underlay routing correspondence extracted from Pastry routing table

Discussion

- Unmodified layer2/3 multicast in end system domains
- HST inherits from Pastry
  - Proximity selection benefits
  - Number of overlay routing hops: $\leq \log_{2^b}(n)$
- Replication load on forwarders limited by size of prefix alphabet 2^b
  - Strictly predictable per packet processing costs
  - With g number of receiver domains: $\leq \log_2(g)(2^b - 1)$
  - $\Rightarrow$ Number of neighbor states: $\leq \log_2(g)(2^b - 1)$
- No dedicated overlay nodes
  - Advoids bottlenecks and single points of failure
- In combination with Bidir-PIM: mobility-agnostic routing
  - Prefix tree will be built only receiver-based
  - HST decouples group and state management from forwarding plane

Outlook

📄 M. Wählisch, T. C. Schmidt:
*Between Underlay and Overlay: On Deployable, Efficient,
Mobility-agnostic Group Communication Services*.
In: Internet Research, 2007, to appear.

▶ Protocol optimization of prefix-controlled forwarding
▶ Further analysis of the Hybrid Shared Tree approach
  ▶ Large scale experiments based on PlanetLab platform
▶ Work of interest for SAM RG?

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

≫ link-lab

Appendix

The Hybrid Shared Tree Architecture
Routing within the Distribution Tree (2)

- ▶ Intermediate vertices need to know tree position
  - ▶ Overlay packets carry destination prefix
- ▶ Check if associated with destination prefix
  - ▶ Yes? Forward to next prefix neighbour(s) (routing in prefix tree + DHT)
  - ▶ No? Just forward to destination prefix (routing in DHT)