MPSS: Multiprotocol Stateless Switching

András Zahemszky[1], Petri Jokela[1], Mikko Särelä[1], Sami Ruponen[2], **James Kempf[3]**, Pekka Nikander[1]

[1]Ericsson Research Finland

[2]VTT – Technical Research Centre of Finland

[3]Ericsson Research Silicon Valley

Overview

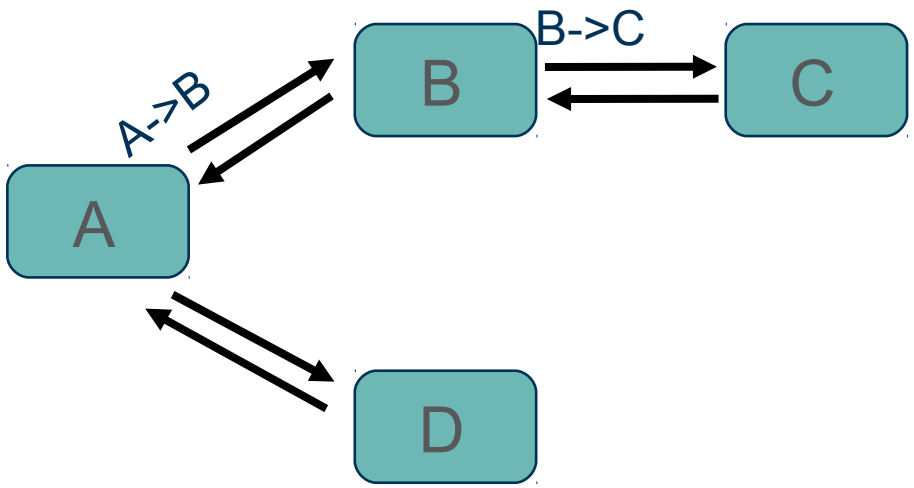
- Background
  - Forwarding with in-packet Bloom filters
  - (G)MPLS basics
- MPSS: Multiprotocol Stateless Switching
  - Basics
  - Use case: Multicast VPNs
- Implementation and evaluation efforts
- Summary and future work

Overview

- Background
  - Forwarding with in-packet Bloom filters
  - (G)MPLS basics
- MPSS: Multiprotocol Stateless Switching
  - Basics
  - Use case: Multicast VPNs
- Implementation and evaluation efforts
- Summary and future work

Forwarding with in-packet Bloom filters

- Bloom Filter
  - Compact representation of a set
- Link Ids
  - Unidirectional links are named
  - “Sparse” bitstrings
- Source Routing
  - Links to visit are encoded into a Bloom Filter
- Basic forwarding operation
  - Is outbound link *L* in Bloom Filter *B*?



A->B	0 1 0 0 0 1 0 0 1
B->C	1 0 0 0 0 1 1 0 0
zF: A->B->C	1 1 0 0 0 1 1 0 1

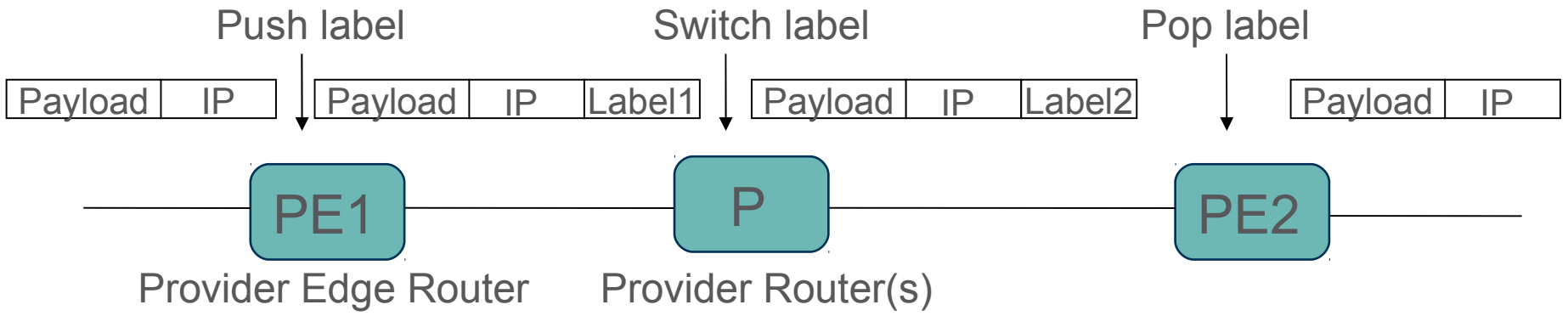
Forwarding with in-packet Bloom filters - 2

- Multicast-capable
- Fast forwarding in hardware
- Compact forwarding tables
- Probabilistic
  - Extra forwarding due to false positives
  - **No false negatives**
  - Practical limit: ~5%-10% extra traffic for 30-40 links with 256-bit Bloom Filters
- See:
  - P. Jokela, A. Zahemszky, C. Esteve, S. Arianfar, P. Nikander, *"LIPSIN: Line Speed Publish/Subscribe Inter-Networking"*, in SIGCOMM '09
  - C. Esteve, P. Jokela, P. Nikander, M. Särelä, J. Ylitalo, *"Self-routing Denial-of-Service Resistant Capabilities using In-packet Bloom filters", in EC2ND '09*

(G)MPLS

- Evolution: MPLS → MPLS-TE → GMPLS

- (G)MPLS is a rich set of protocols
  - Setting up Label Switched Paths (LSPs)
  - Forwarding on LSPs
  - Traffic Engineering (TE), resiliency (e.g. fast reroute)
  - VPNs
  - Unified control plane for many different forwarding technologies

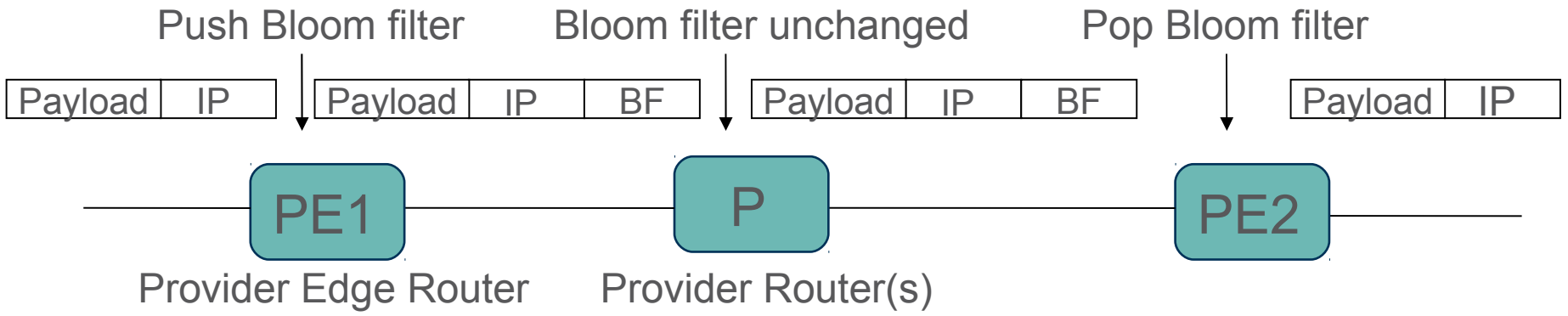


Overview

- Background
  - Forwarding with in-packet Bloom filters
  - (G)MPLS basics
- MPSS: Multiprotocol Stateless Switching
  - Basics
  - Use case: Multicast VPNs
- On-going implementation and evaluation
- Summary and future work

MPSS basic idea

- MPLS labels are replaced by in-packet Bloom filters
  - Slightly longer packet headers
  - Easier (multicast) tree setup than in MPLS
  - “Stateless” sparse mode multicast



MPSS mechanisms

- Task 1: Computing a tree/path
  - Distributed by utilizing underlying routing
  - Centralized by using PCE locally/remotely

As in MPLS

- Task 2: Computing a Bloom filter for the tree/path
  - Hop-by-hop
  - Centralized using the network map

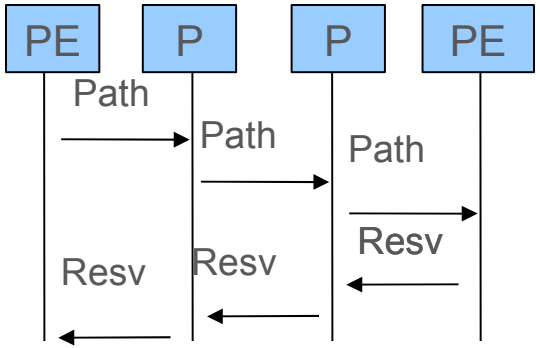
MPLS extensions

- Task 3 (optional): Bandwidth reservation
  - RSVP-TE style hop-by-hop
    - Routed by IP or by in-packet Bloom filters
  - Bandwidth broker

MPLS extensions

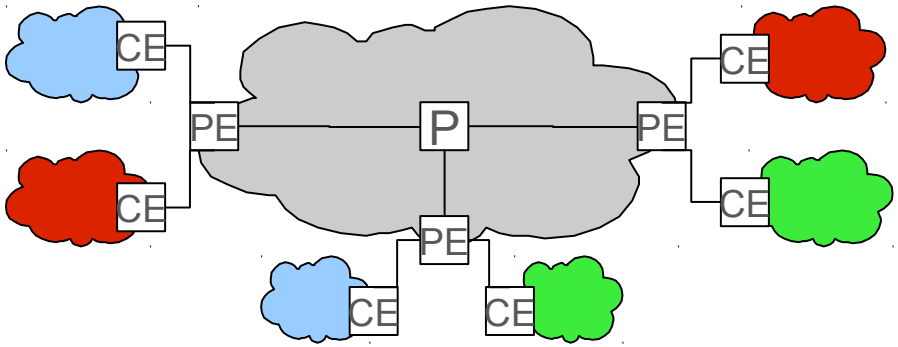
Two example scenarios for setting up trees

- Scenario 1: Explicit paths, no bandwidth reservation
  - Link Ids are advertised with OSPF-TE
  - Source computes a desired tree on the network map
  - Source determines the in-packet Bloom filter
- Scenario 2: Explicit paths, bw. reservation
  - Source computes a desired tree on the network map
  - Source signals the branches separately with *Path* messages
  - Returned *Resv* messages contain the hop-by-hop collected in-packet Bloom for each branch



MPSS in Multicast VPNs

- L3VPNs are currently being standardized in IETF
- Trade-off bandwidth usage and state
  - Difficult process, non-optimal setups
  - State maintenance
- MPSS avoids the fine-tuning process and the in-network state maintenance
  - No need to aggregate trees
  - Trees “exist” in the packet headers
  - Karpilovsky et. al. measurements from 2009:
    - ~86% of multicast flows in MVPN core has 2-3 PE receivers – fits well with small BFs
    - ~13% of multicast flows has in average 20 PE receivers – optimization for BFs might be needed



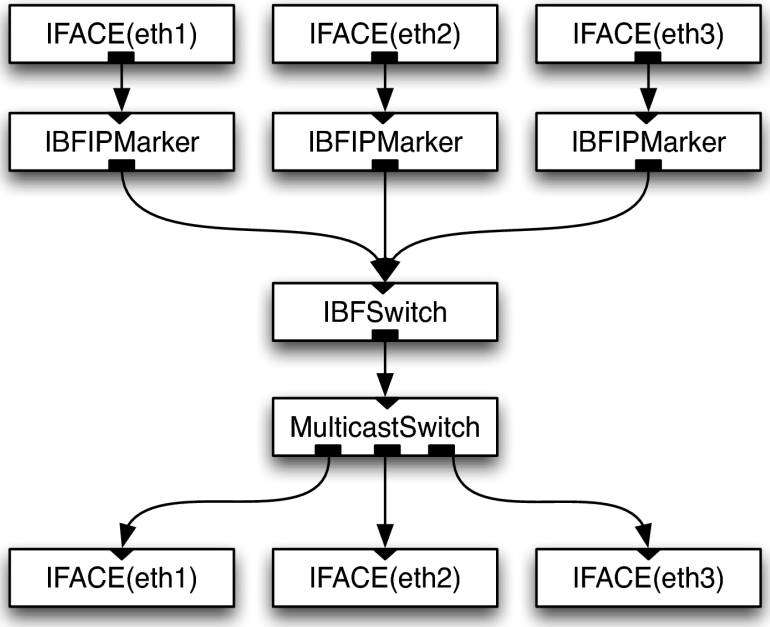
Overview

- Background
  - Forwarding with in-packet Bloom filters
  - (G)MPLS basics
- MPSS: Multiprotocol Stateless Switching
  - Basic idea
  - Setting up paths
  - Use case: Multicast VPNs
- Implementation and evaluation efforts
- Summary and future work

Implementation

- Forwarding plane implementation in Click modular router
  - Both Provider Edge and Provider Routers are implemented
- NetFPGA and FreeBSD implementations available
  - In PSIRP EU FP7 project

- Control plane implementation is on-going
  - OSPF extensions to Quagga for advertising Link IDs
  - RSVP-TE extensions in Dragon for zFilter calculation/bandwidth reservation

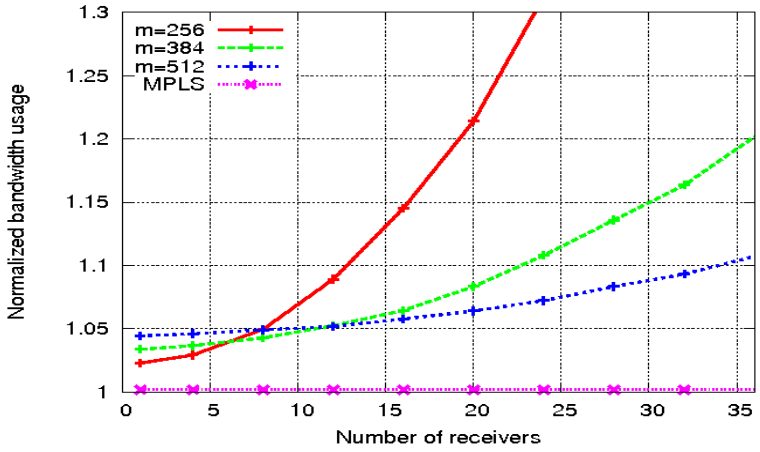


Preliminary evaluation

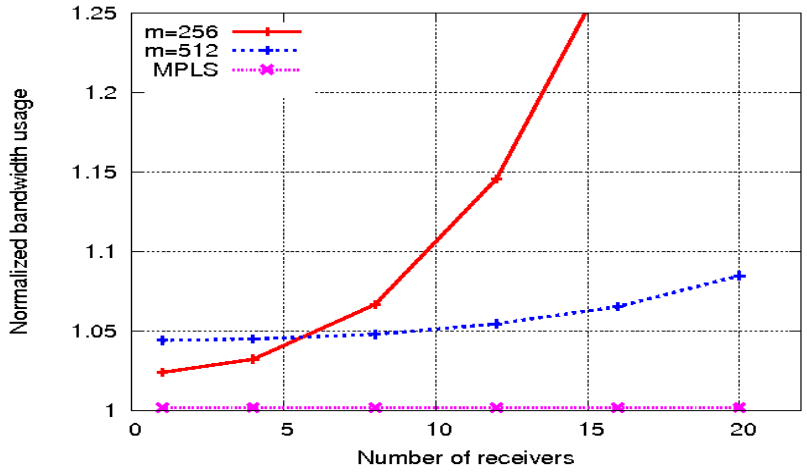
- Characterizing the bandwidth overhead
  - Different BF sizes, tree sizes (on-going)
  - Compare to MVPN tree allocation strategies (future)
  - Effect of moderate in-network state (future)
- On Rocketfuel networks & topologies from RFC 5439

Normalized bandwidth usage for different Bloom Filter sizes (m)

AS3257 – 161 nodes



Snowflake – 1055 nodes

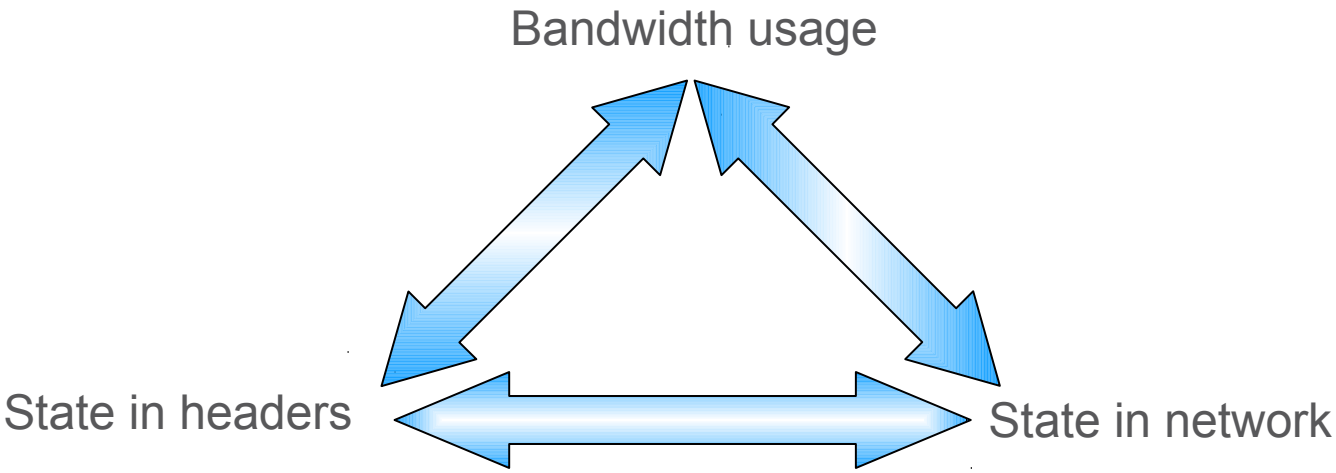


Overview

- Background
  - Forwarding with in-packet Bloom filters
  - (G)MPLS basics
- MPSS: Multiprotocol Stateless Switching
  - Basic idea
  - Setting up paths
  - Use case: Multicast VPNs
- Implementation and evaluation efforts
- Summary and future work

Summary and Future Work

- MPSS uses in-packet Bloom filters instead of MPLS labels
  - State inside the packet headers
  - Compact forwarding tables
  - Signaling not necessarily needed for path/tree setup
- Future work
  - Explore the multicast routing design space wrt. in-packet Bloom Filters



ERICSSON