MEB: an Efficient and Accurate Multicast using Bloom Filter with Customized Hash Function

Zihao Chen, Jiawei Huang, Qile Wang, Jingling Liu, Zhaoyi Li, Shengwen Zhou, Zhidong He
Central South University, China
Introduction

Multicast Traffic: A Crucial Component in DCNs

Real-Time Applications
Stock quotes, trading information

Distributed Applications
Distributed Learning, big data processing
Introduction

IP Multicast: IGMP Table Guides Switch Forwarding

Controller

Join/Leave Update

IGMP

rule1
rule2
rule3

S
R
R
Introduction

Source-Routed Multicast: Storing Routing Information in Packets

Controller

Encode multicast

Parsing label

labels payload

S R R
Solution 1: Storing port bitmaps in packet headers

Example: Yeti[nsdi22], elmo[sigcomm19]

A multicast group encoded as a list of (Switch, Ports) pairs

<table>
<thead>
<tr>
<th>Switch 2: [Bitmap]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch 3: [Bitmap]</td>
</tr>
<tr>
<td>Switch 4: [010101]</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

Significant overhead in packet headers!
Motivation

- Solution 2: Use **Bloom filter** instead of port bitmaps
- Example: LIPSIN [sigcomm09]

Bloom filter supports membership queries

**Insertion**: Set the hashed $M$ positions to 1

![Diagram showing Bloom filter with hash mapping and payload](image)
Motivation

- Solution 2: Use Bloom filters instead of port bitmaps
- Example: LIPSIN[ sigcomm09 ]

Query: All M hashes are 1, then it exists

Issue of false positives!
Motivation

Experiment observation

Under Fat-Tree networks of different scales

Label overhead increases

False positive rate increases
Design an efficient multicast forwarding system for networks

- Low burden on switch
  Adapt to the growth of the network scale

- Low Label Overhead
  Improve the efficiency of data transmission

- Low False Positive Rate
  Decrease redundant on traffic
MEB overview

- The packet header carries a bloom filter encoding all positive ports.
- At the sender end, custom the hash function for positive ports that brings false positive port.

![Diagram of MEB overview]
1. Encoding the packet header

Encode the ports to be forwarded using the default hash functions into the Bloom filter
1. Encoding the packet header

MEB design

Find all the false positive ports
MEB design

1. Encoding the packet header

Customize hash functions for true ports

True ports: \(p_1, p_2, p_3\)
False ports: \(p_4, p_5\)

Value: 0 1 1 0 1 0 1 0 0 1
1. Encoding the packet header

Add the MEB label to the packet header

| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | P3,010,h4 | payload |

True ports: $p_1$, $p_2$, $p_3$

False ports: $p_4$, $p_5$

$\{h_1, h_4, h_3\}$
2. Choosing an appropriate label size

A marginal benefit between the size of the bloom filter and the false positive rate.
We choose a position with a drop rate lower than 1% as the final label size.
3. Parsing labels and forwarding data packets

Use a custom hash function to hash the port
If three hash results are all 1, then forward
Evaluation

Under Fat Tree networks of different scales
40% of the hosts act as receivers

Label size ↓
Switch overheard zero

False positive rate ↓
Forwarding efficiency ↑
Evaluation

Under different network type
40% of the hosts act as receivers

Label size

Switch overheard zero

False positive rate

Forwarding efficiency
Summary

- Designed for various network topologies
- Switches do not have any storage overhead
- Significantly reduces label size and the False Positive Rate

MEB
Multicast with Customized Hash Function
Thank you!

Q&A

MEB: an Efficient and Accurate Multicast using Bloom Filter with Customized Hash Function

Zihao Chen, Jiawei Huang, Qile Wang, Jingling Liu, Zhaoyi Li, Shengwen Zhou, Zhidong He

Central South University, China

Contact: chenzihaohappy@csu.edu.cn