P4-Based Implementation of BIER and BIER-FRR for Efficient Multicast

Daniel Merling, Steffen Lindner, Michael Menth

http://kn.inf.uni-tuebingen.de
The Need for BIER

- IP multicast is widely used to address multiple receivers
- Hosts can subscribe and unsubscribe leveraging IGMP
- Connected routers propagate this information using multicast routing protocols, e.g., PIM

**Problem**
- Intermediate routers need to store (S, G) state
- All routers need to recalculate their states upon group changes

**Solution:** Bit Index Explicit Replication (BIER)
Domain concept
Domain concept

- Ingress nodes (BFIR)
  - Add BIER header
    - Contains all destinations of the packet
Domain concept

- **Ingress nodes (BFIR)**
  - Add BIER header
    - Contains all destinations of the packet

- **Core nodes (BFR)**
  - Forward and replicate packets on paths from the routing underlay (e.g., IGP)
  - Distribution on tree structure
Domain concept

- **Ingress nodes (BFIR)**
  - Add BIER header
    - Contains all destinations of the packet

- **Core nodes (BFR)**
  - Forward and replicate packets on paths from the routing underlay (e.g., IGP)
  - Distribution on tree structure

- **Egress nodes (BFER)**
  - Remove BIER header

P4-Based Implementation of BIER and BIER-FRR for Efficient Multicast
BIER Forwarding

► BIER header

- Contains bit string where each BFER is assigned to a position
  - If BFER should receive a packet copy, its bit is activated in the packet header

- Individual BIER header for each IPMC group

- Before a BFRs forwards a packet to a NH, it clears bits of BFERs that are reached via other NHs from the packet header to avoid duplicates

<table>
<thead>
<tr>
<th>BFER</th>
<th>Next-Hop</th>
<th>Forwarding Bitmask (F-BM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>100110</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>100110</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>011000</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>011000</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>100110</td>
</tr>
</tbody>
</table>
**BIER Forwarding**

- **BIER header**
  - Contains bit string where each BFER is assigned to a position
    - If BFER should receive a packet copy, its bit is activated in the packet header
  - Individual BIER header for each IPMC group
  - Before a BFRs forwards a packet to a NH, it clears bits of BFERs that are reached via other NHs from the packet header to avoid duplicates

**BIFT of BFR 1**

<table>
<thead>
<tr>
<th>BFER</th>
<th>Next-Hop</th>
<th>Forwarding Bitmask (F-BM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>100110</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>100110</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>011000</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>011000</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>100110</td>
</tr>
</tbody>
</table>

111110 & 100110

![Diagram of BIER Forwarding](image)
**BIER Forwarding**

### BIER header

- Contains bit string where each BFER is assigned to a position
  - If BFER should receive a packet copy, its bit is activated in the packet header
- Individual BIER header for each IPMC group
- Before a BFRs forwards a packet to a NH, it clears bits of BFERs that are reached via other NHs from the packet header to avoid duplicates

#### BIFT of BFR 1

<table>
<thead>
<tr>
<th>BFER</th>
<th>Next-Hop</th>
<th>Forwarding Bitmask (F-BM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>100110</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>100110</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>011000</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>011000</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>100110</td>
</tr>
</tbody>
</table>

![Diagram showing BIER forwarding with BIFT entries]
BIER - Properties

► BIER header
  ▪ Contains bit string where each BFER is assigned to a position
    - If BFER should receive a packet copy, its bit is activated in the packet header
  ▪ Individual BIER header for each IPMC group
  ▪ Before a BFRs forwards a packet to a NH, it clears bits of BFERs that are reached via other NHs from the packet header to avoid duplicates

► Iterative forwarding mechanism
  ▪ In each iteration one packet copy is forwarded to a NH

► Scalability
  ▪ No dynamic state within core devices
  ▪ Signaling only to BFIRs
Motivation for BIER implementation
- Solves issues of traditional multicast (state, signaling, …)
- Significant support from several global players (Cisco, Juniper, Nokia, Google, …)
- Yet, there is no working BIER implementation!
  - Evaluation, operability, …

Implemented with P4 on Intel Tofino ASIC
- Runs at 100 Gb/s
- Published in IEEE Access
Why P4?

- Efficient implementation with legacy devices not possible
- P4 offers required flexibility
  - Define new header
  - Define processing pipeline
  - Iterative forwarding procedure
- Easier management with separated control plane
In each iteration, one next-hop is served

- Clone packet & recirculate

Network Switch

Path of original BIER packet
Path of cloned BIER packet

Recirculation port

Regular port

Path of original BIER packet
Path of cloned BIER packet

1st pipeline iteration
2nd pipeline iteration
3rd pipeline iteration
**Problem:** Recirculation requires capacity

- 100 Gbit/s multicast traffic with 5 next-hops results in 400 Gbit/s recirculation traffic
- Solution: Add dedicated recirculation ports to increase recirculation capacity
Fast Reroute (FRR) deviates traffic around a local failure
- Link failure
- Node failure

No native FRR support in P4
- Tofino generates a special packet when ports are up/down
- We store this information in registers to apply FRR
Steffen Lindner
University of Tuebingen
Faculty of Science
Department of Computer Science
Chair of Communication Networks