Scaling Bloom filter-based multicast via filter switching

Christos Tsilopoulos and George Xylomenos

Mobile Multimedia Laboratory, Athens University of Economics and Business

DEPARTMENT OF INFORMATICS

European Union
European Social Fund

OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION, LIFELONG LEARNING AND RELIGIOUS AFFAIRS
MANAGING AUTHORITY

Co- financed by Greece and the European Union

NSRF
2007-2013
EUROPEAN SOCIAL FUND

Presentation at a glance

- LIPSIN [JOK2009]
  - Packet forwarding with in-packet Bloom filters
  - Stateless multicast forwarding
  - Scales w.r.t. number of multicast groups
  - Poor scalability w.r.t. group/network size

Presentation at a glance

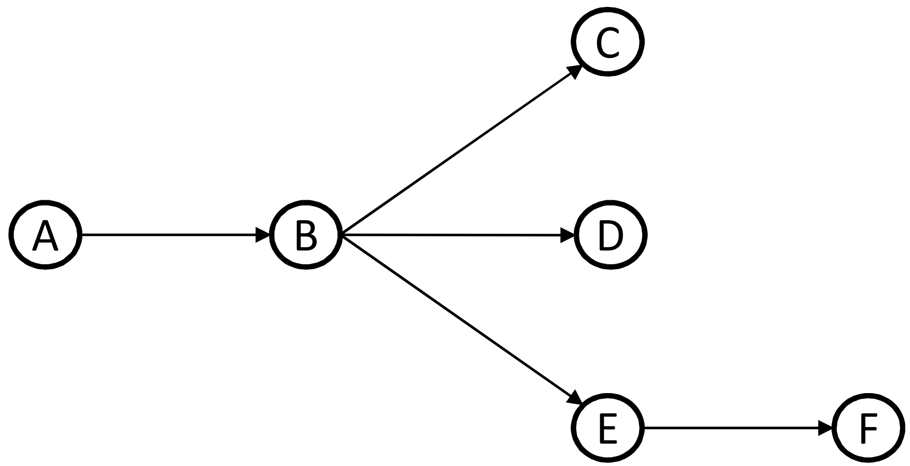
- LIPSIN [JOK2009]
  - Packet forwarding with in-packet Bloom filters
  - Stateless multicast forwarding
  - Scales w.r.t. number of multicast groups
  - Poor scalability w.r.t. group/network size
- Bloom filter switching
  - Use of relay points
  - Scale w.r.t. group/network size
  - Sacrifice fully stateless operation
  - Measure the trade-offs

Presentation Outline

- Problem statement
  - Bloom filter-based packet forwarding
  - Scalability issues w.r.t. to group/network size
- Bloom filter switching
- Evaluation
  - Compare state requirements with other multicast schemes
- Conclusion and future work

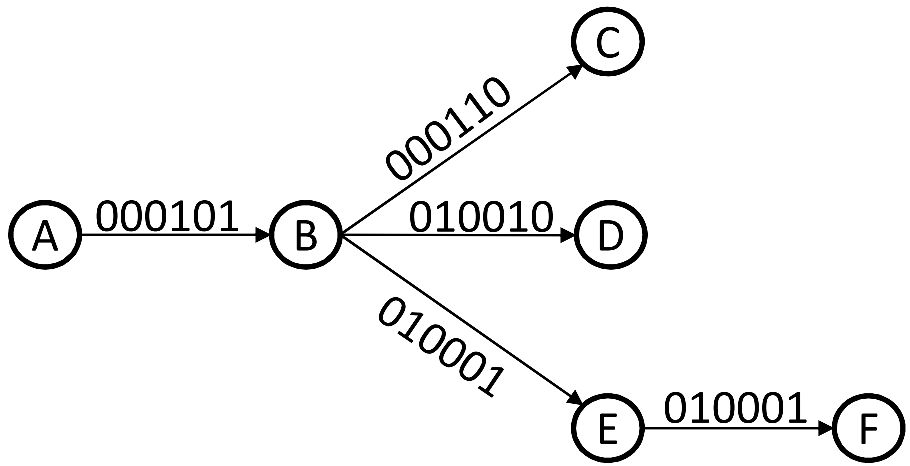
In-packet Bloom filters (iBF)

- Source-routing scheme
- Path links encoded to Bloom filters



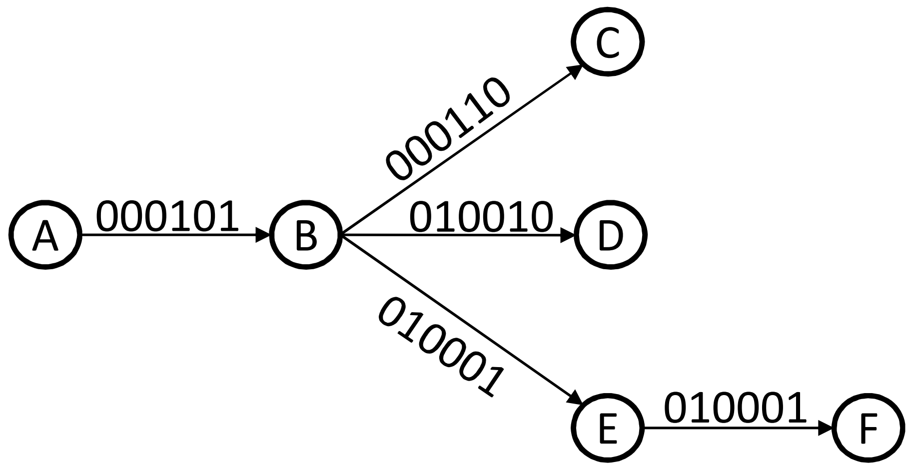
In-packet Bloom filters (iBF)

- Source-routing scheme
- Path links encoded to Bloom filters
  - Link IDs (LID)
    - m bits, k bits set to 1 ($k << m$)
    - Unidirectional

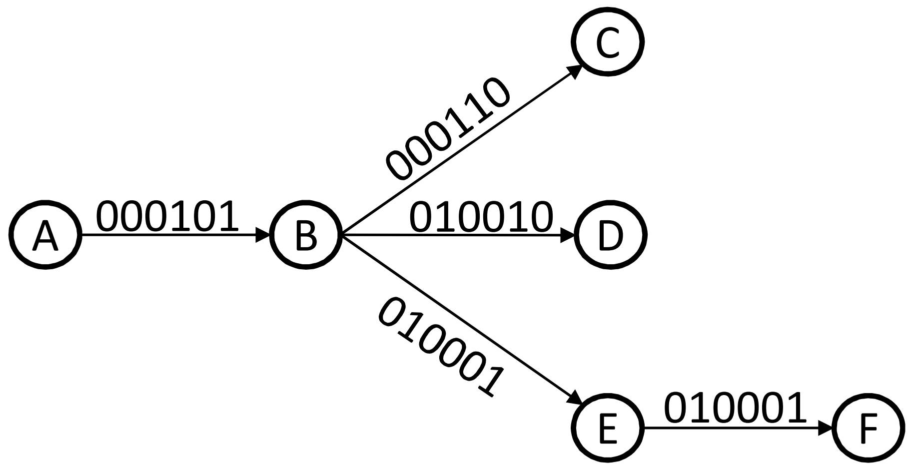


In-packet Bloom filters (iBF)

- Source-routing scheme
- Path links encoded to Bloom filters
  - Link IDs (LID)
    - *m* bits, *k* bits set to 1 (*k* << *m*)
    - Unidirectional
  - OR delivery path LIDs



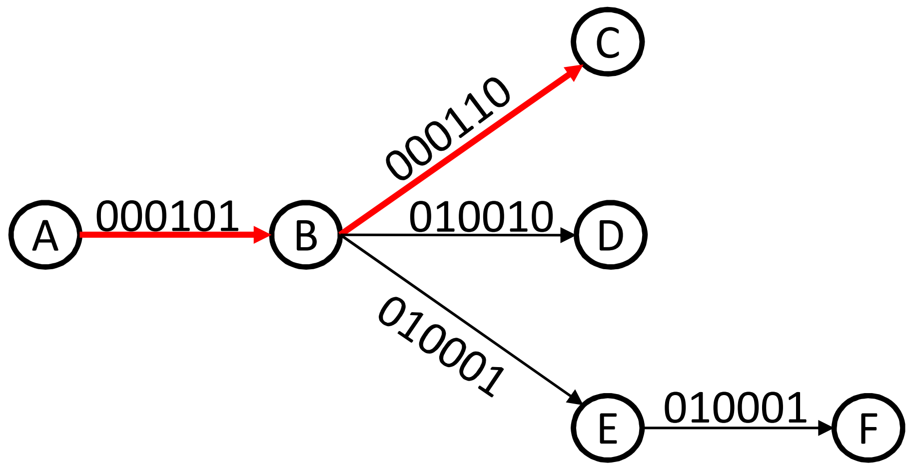
In-packet Bloom filters (iBF)



- Source-routing scheme
- Path links encoded to Bloom filters
  - Link IDs (LID)
    - *m* bits, *k* bits set to 1 (*k* << *m*)
    - Unidirectional
  - OR delivery path LIDs
- Place Bloom filter in packet header

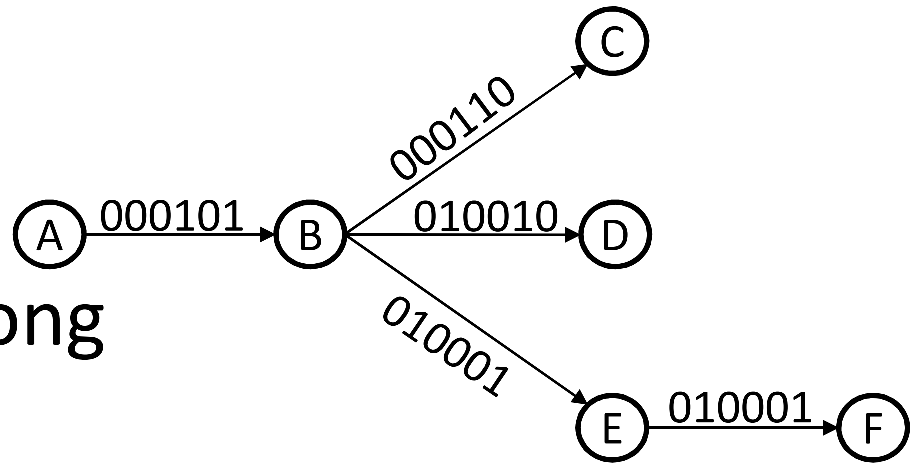
In-packet Bloom filters (iBF)

- Source-routing scheme
- Path links encoded to Bloom filters
  - Link IDs (LID)
    - *m* bits, *k* bits set to 1 (*k* << *m*)
    - Unidirectional
  - OR delivery path LIDs
- Place Bloom filter in packet header
- E.g., $iBF_{AC} = LID_{AB} \mid LID_{BC} = 000111$



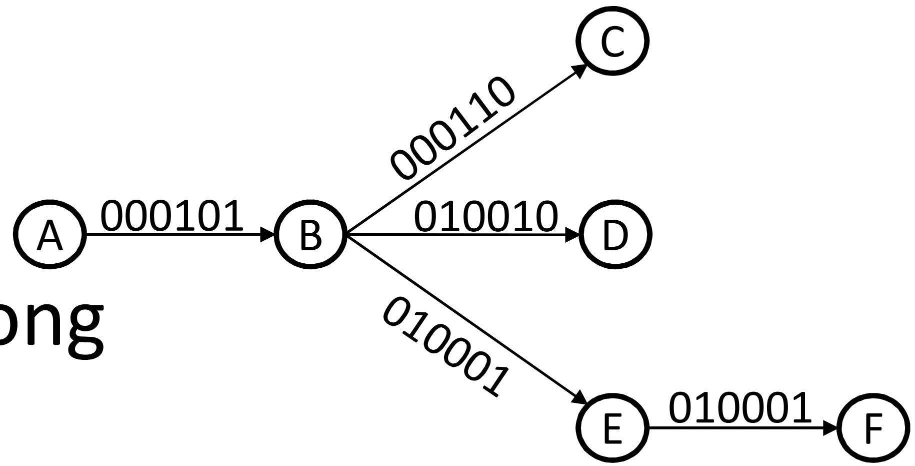
Data Plane

- Routers extract iBF
- Check which of their outgoing LIDs belong to the iBF



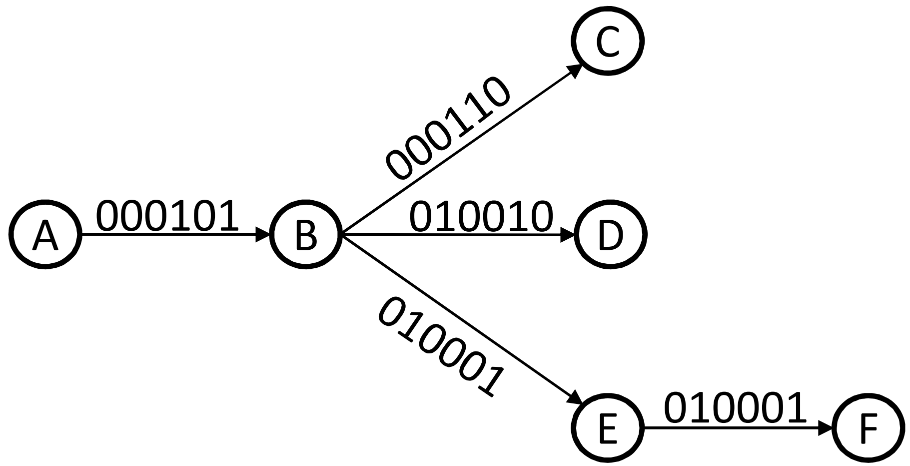
Data Plane

- Routers extract iBF
- Check which of their outgoing LIDs belong to the iBF
- Bitwise AND operation
  - *iBF & LID_i == LID_i*



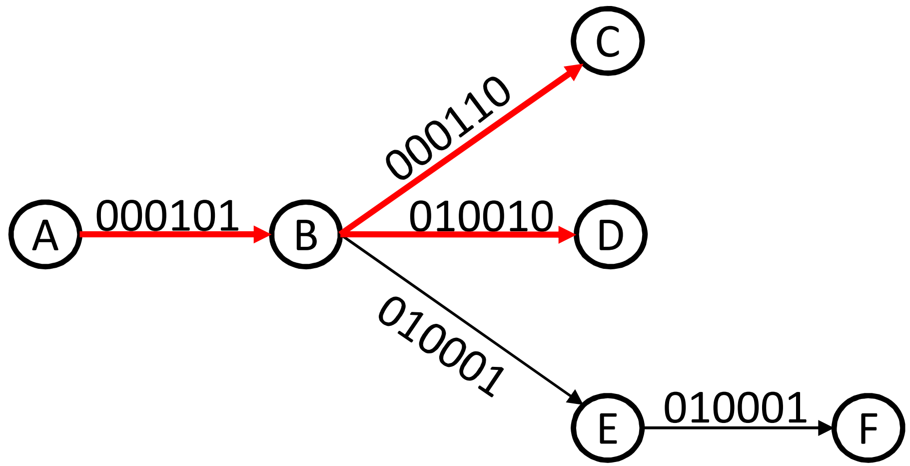
Single-Source Multicast

- Add multicast tree links



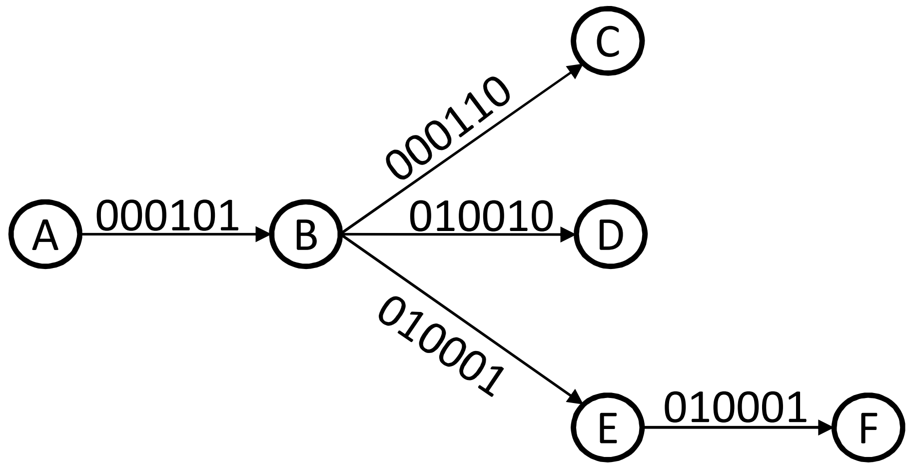
Single-Source Multicast

- Add multicast tree links
  - $iBF_{A\text{-}\{C,D\}} = 010111$



Single-Source Multicast

- Add multicast tree links
  - $iBF_{A\text{-}\{C,D\}} = 010111$
- Forwarding logic remains the same

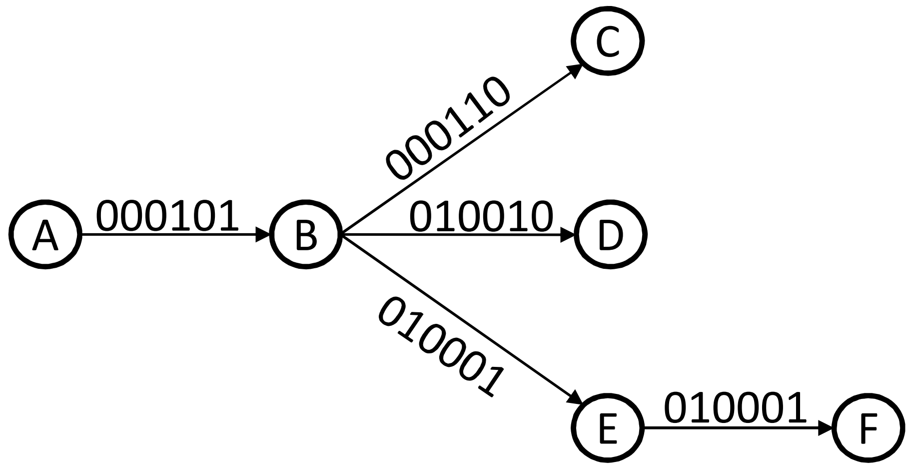


Single-Source Multicast

- Add multicast tree links
  - $iBF_{A\text{-}\{C,D\}}$ = 010111
- Forwarding logic remains the same

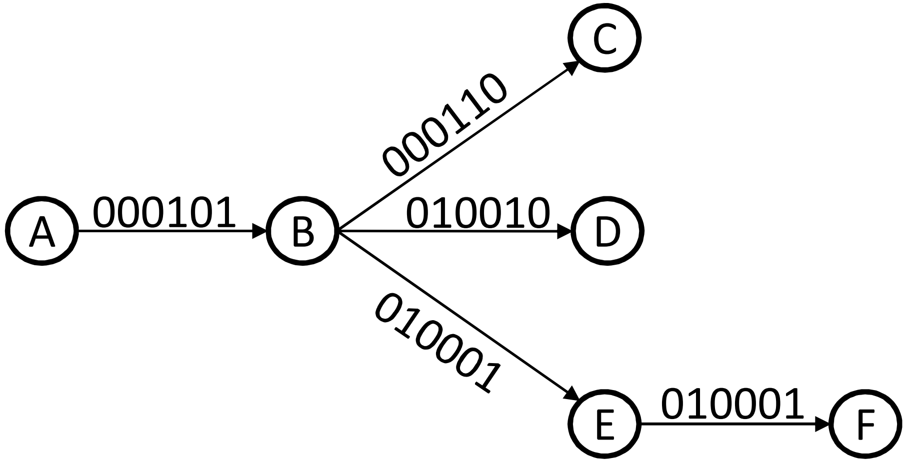
✓ No multicast state at routers

✓ Fixed size header
  - Line-speed operation [JOK2009]



Single-Source Multicast

- Add multicast tree links
  - $iBF_{A\text{-}\{C,D\}}$ = 010111
- Forwarding logic remains the same



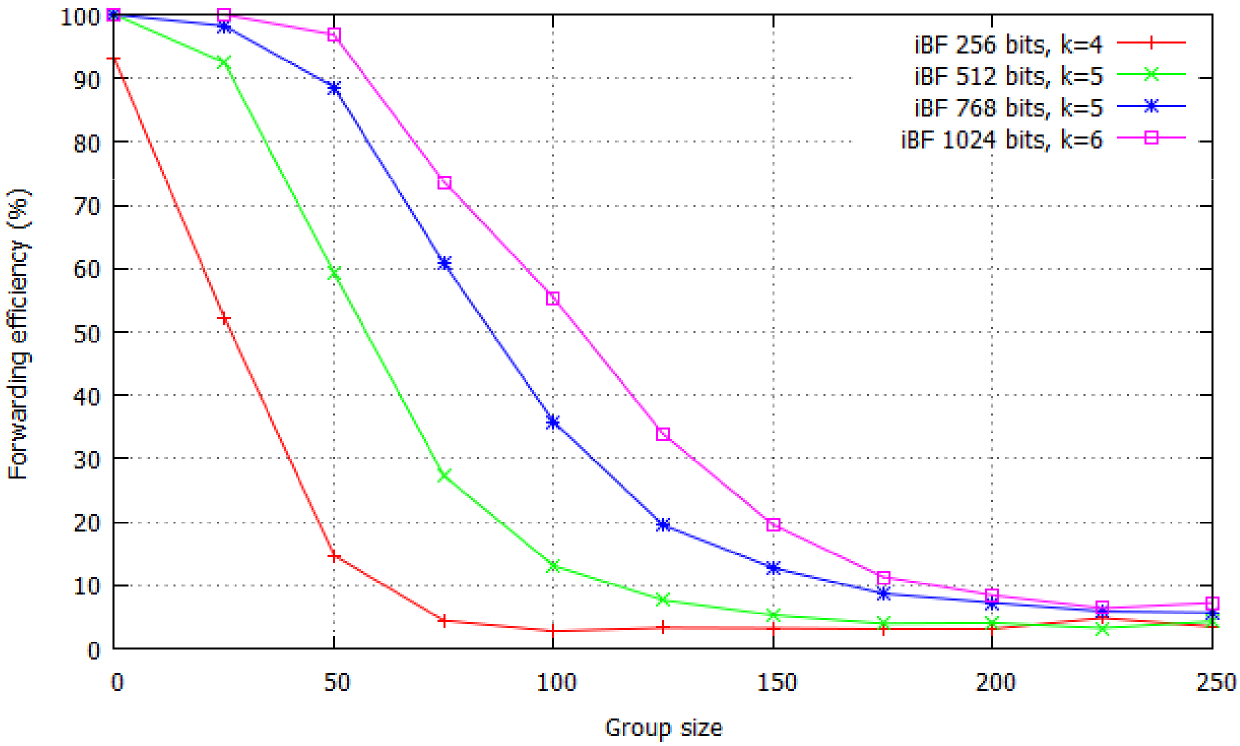
✓ No multicast state at routers

✓ Fixed size header
  - Line-speed operation [JOK2009]

✘ False positives in Bloom filters
  - $fpp = (1 - e^{-kn/m})^k$

False Forwarding Decisions

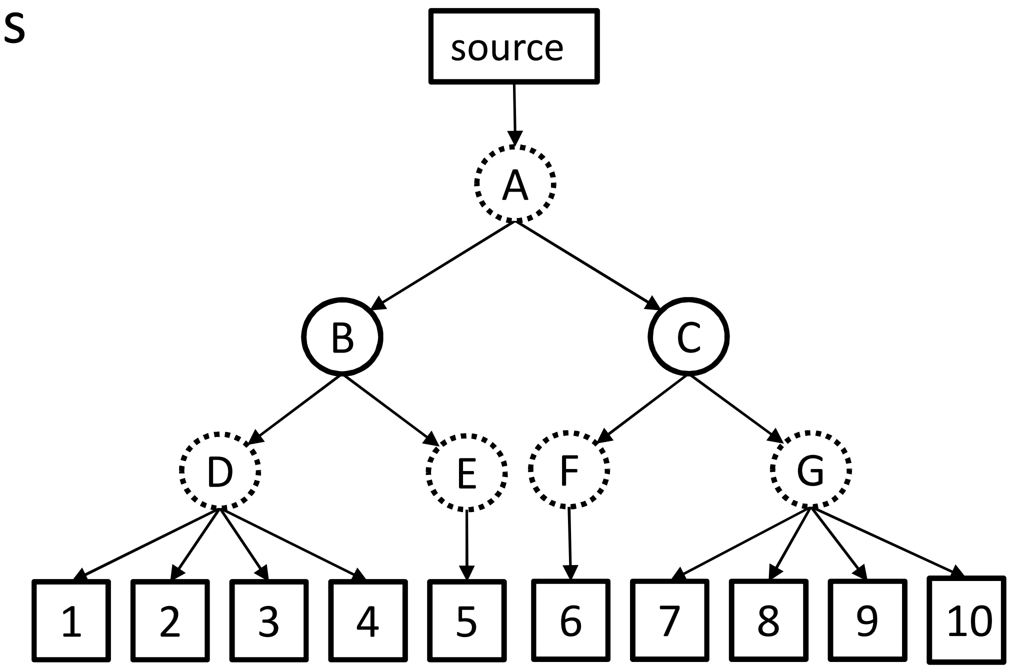
- False positives probability increases as more LIDs are added
- Poor scalability w.r.t. group/network size



Forwarding efficiency = (# tree links) / (total packets transmitted)

Bloom filter switching

- Idea: sacrifice fully stateless operation
- Select a *few* nodes that act as relays
  - Install multicast state
- Relay points switch iBFs



Relay Node Selection

- Define a false positive probability threshold (fpp_{thres})

Relay Node Selection

- Define a false positive probability threshold (fpp_{thres})
- Compute max number of links per subtree

$$n_{max} = -ln(1 - fpp_{thres}{}^{-1/k})(\frac{m}{n})$$

Relay Node Selection

- Define a false positive probability threshold (fpp_{thres})
- Compute max number of links per subtree

$$n_{max} = -ln(1 - fpp_{thres}{}^{-1/k})(\frac{m}{n})$$

- Bottom-up post-order tree traversal

Relay Node Selection

- Define a false positive probability threshold (fpp_{thres})
- Compute max number of links per subtree

$$n_{max} = -ln(1 - fpp_{thres}{}^{-1/k})(\frac{m}{n})$$

- Bottom-up post-order tree traversal
- At node *i,* if $subtree_i$ has more than n_{max} links
  - *i* becomes relay point
  - Compute iBF for subtree rooted at *i*
  - Install iBF at *i*'s multicast forwarding table
  - Consider *i* as leaf and continue

Algorithm

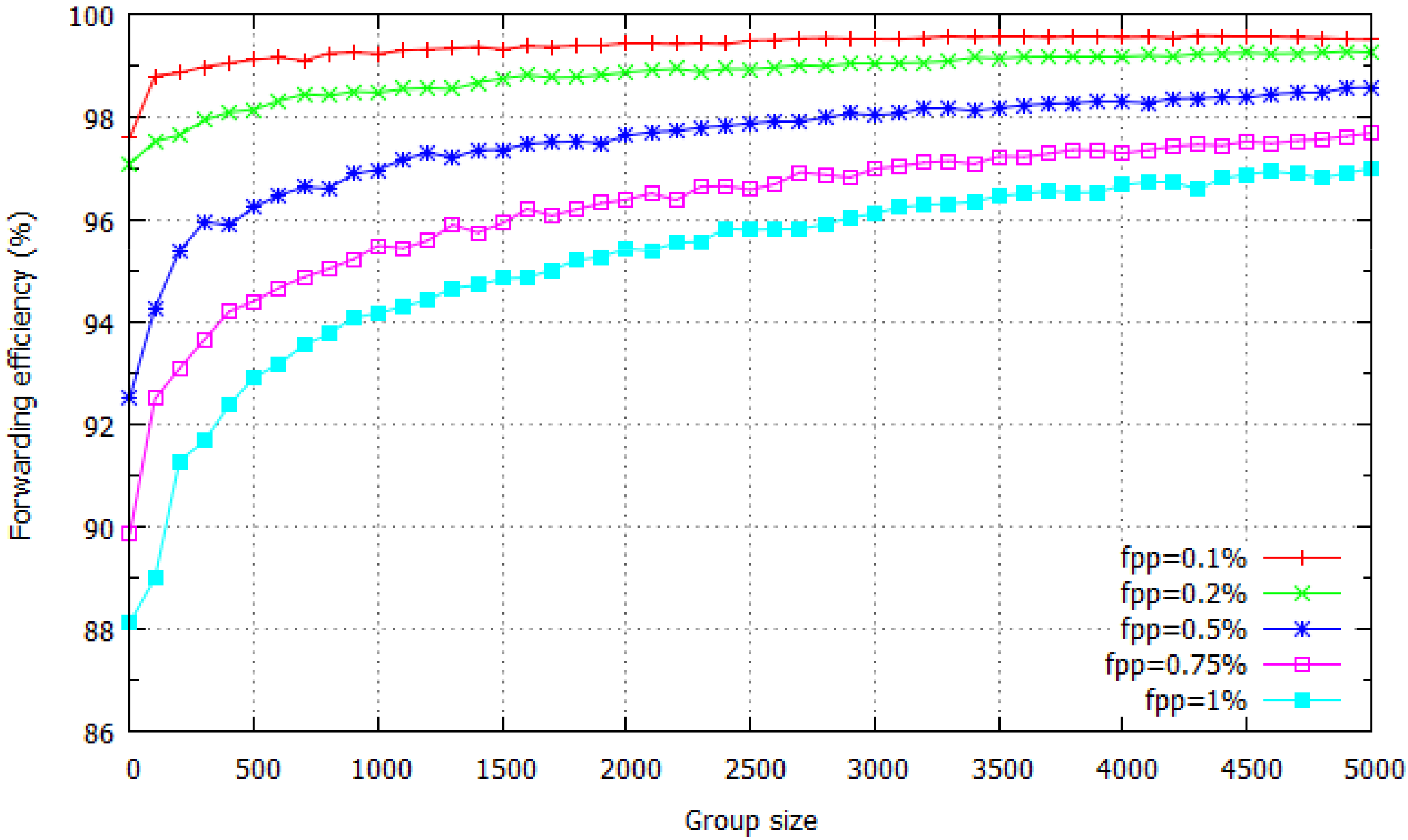
```
Method: sub_tree_traverse
Input: t: multicast tree;
           i: current root node;
           n: maximum number of nodes;
n_i := 0;
for (j in C_i) {
     n_i := n_i + 1 + sub_tree_traverse (t, j, n);
}
if ( n_i ≥ n_max){
     iBF_i := compute_iBF(t, i);
     installState(iBF_i, i);
     removeSubtree(t, C_i);
     n_i = 0;
}
return n_i;
end method
```

Evaluation

- Focus on forwarding efficiency and state requirements
  - Compare state requirements against other multicast schemes
- Input
  - Synthetic scale free graphs
  - Barabási–Albert algorithm [BAR1999]
  - 500 to <u>5000</u> nodes

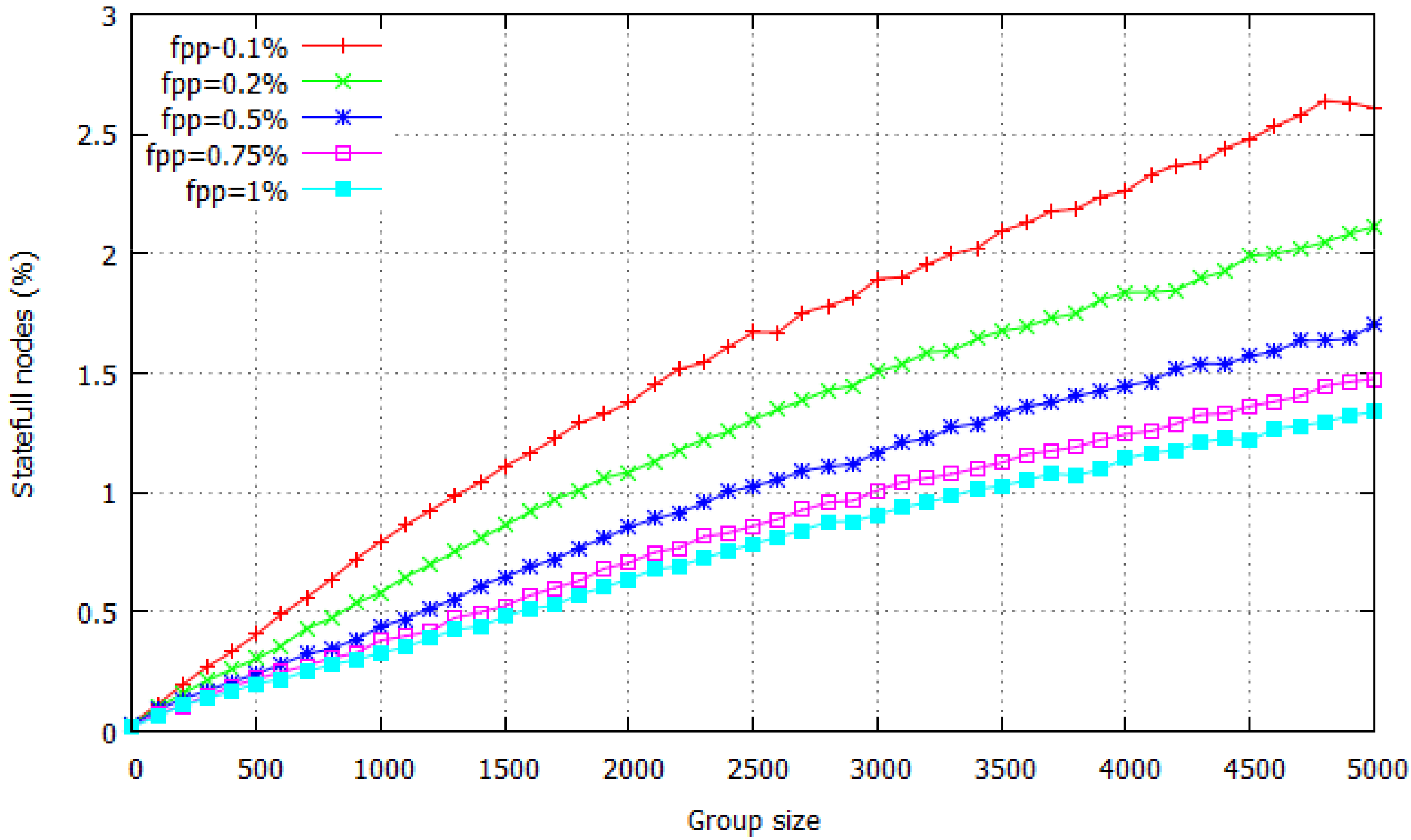
Forwarding Efficiency

- 256-bit iBF, k=4



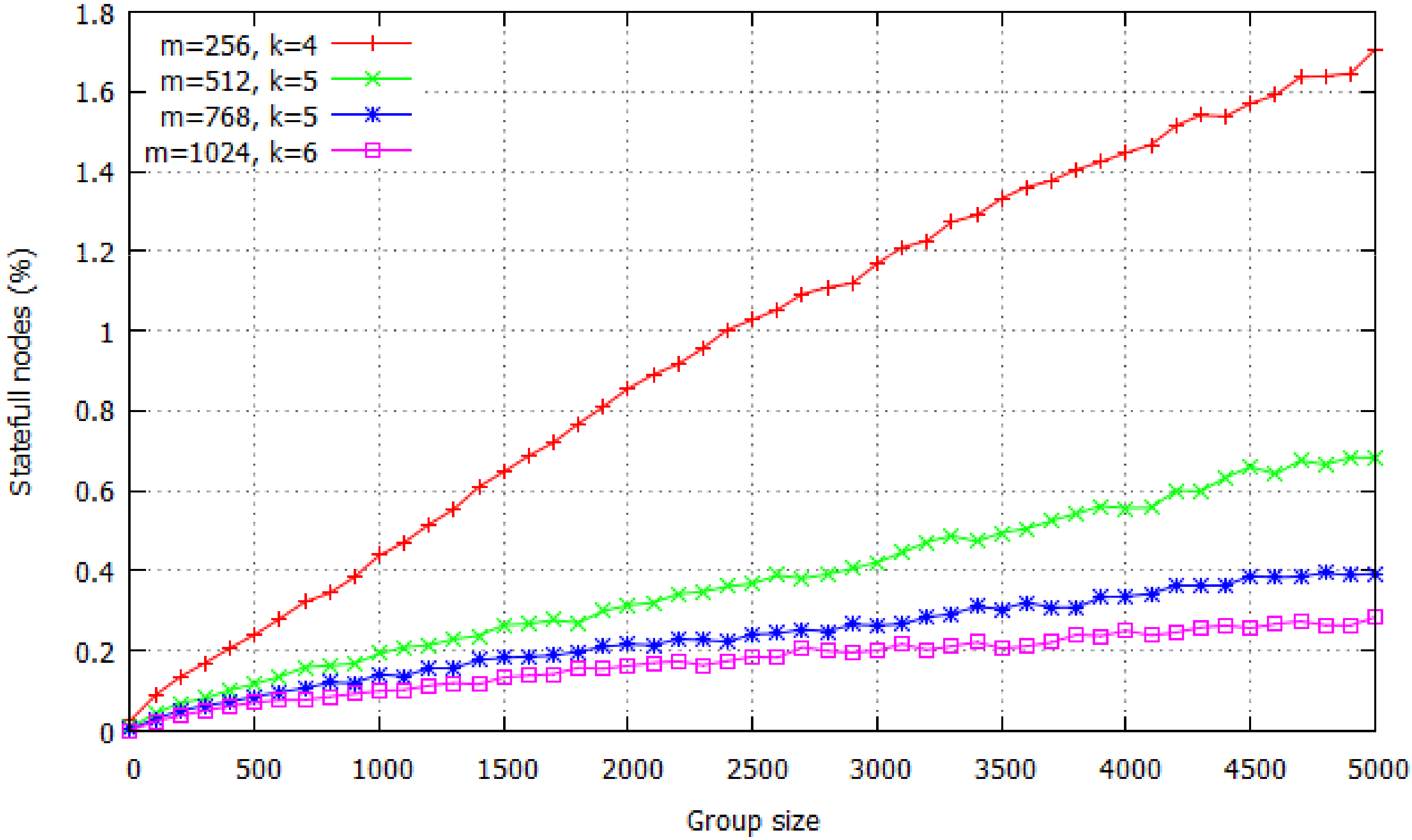
State Requirements (Relay Points)

- 256-bit iBF, k=4



State Requirements - 2

- fpp = 0.5%

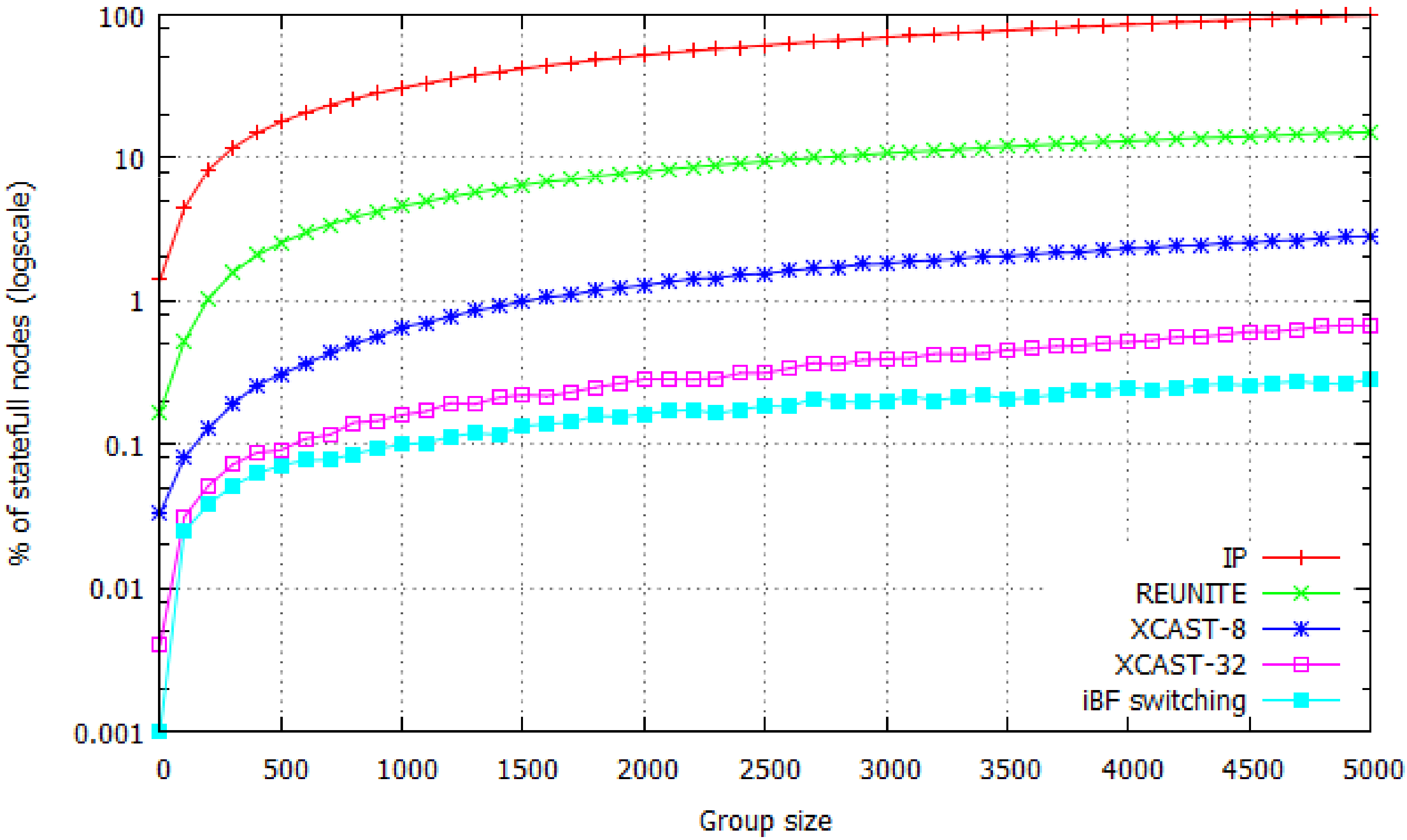


Comparison

- Amount of stateful nodes compared to
  - Hop-by-hop (IP, overlay multicast)
    - Forwarding state per multicast tree at all nodes
  - REUNITE [STO2000]
    - Forwarding state per multicast tree at branching points only
  - Semi-stateful Xcast [NIA2008]
    - List of receiver addresses in packet header
    - Maximum number of receiver addresses in header
    - Use of relay points

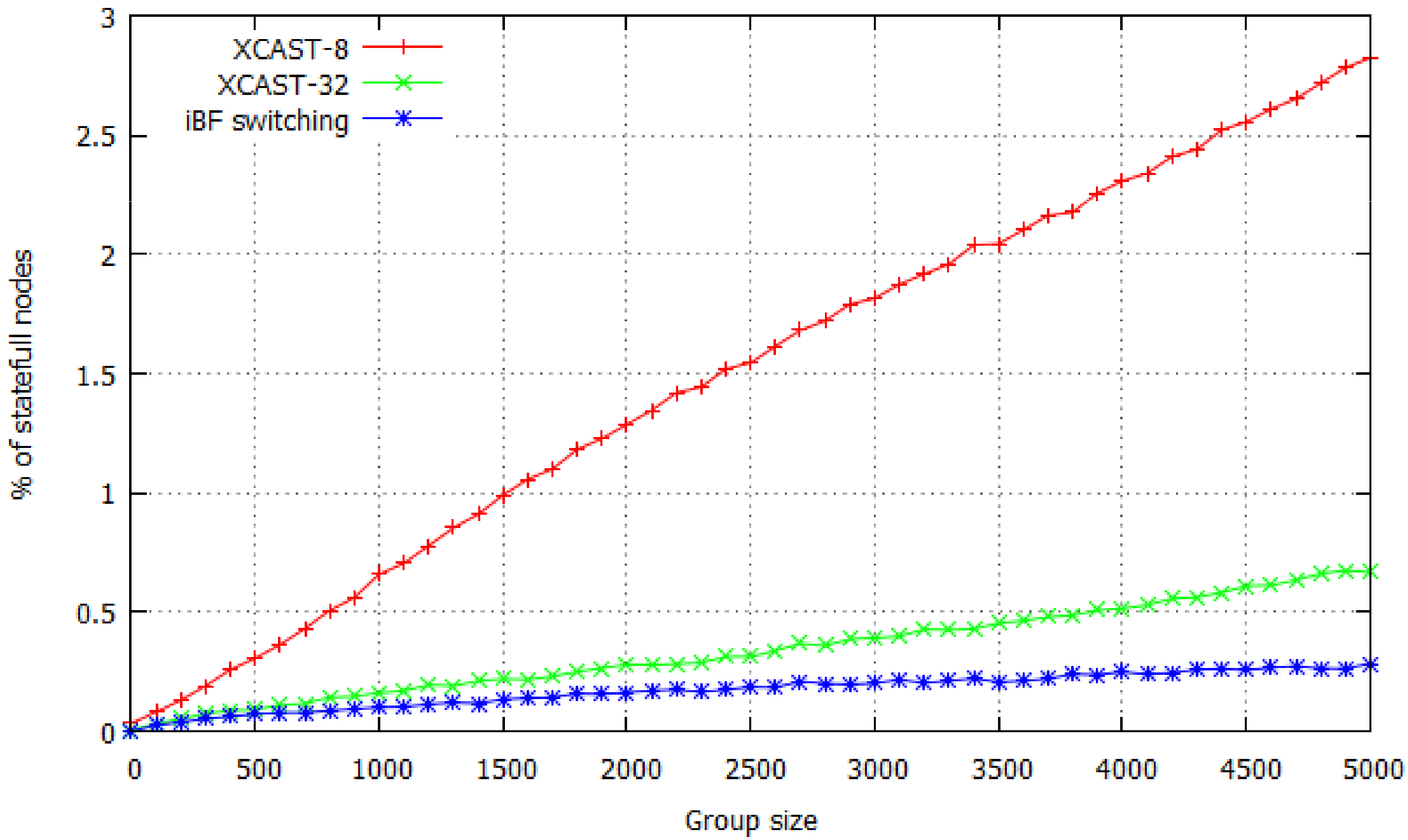
Comparison - 2

- 1024-bit headers for iBF and Xcast
- Y-axis in log scale



Comparison - 3

- 1024-bit headers for iBF and Xcast



Conclusions

- iBF switching handles scalability w.r.t. group/network size
- Trade-off: place multicast forwarding state at some routers
- Still, far less state requirements than other multicast schemes
- Requires centralized routing module
  - Suitable for multicast applications with low dynamicity, e.g. orchestrated software updates
  - Not suitable with dynamic user behavior, e.g. IPTV channel switching

Future Work

- Distribution of multicast state
  - *Central* nodes tend to concentrate state
  - Relay node selection algorithm
- Distribution of group sizes
  - Zipf distribution
  - Small groups require no state
- Fast Join/Leave operation
  - Dynamic user behavior
  - Distributed operation

Thank you

Email: tsilochr@aueb.gr

References

- [JOK2009] P. Jokela, A. Zahemszky, S. Arianfar, P. Nikander, and C. Esteve, “LIPSIN: Line speed publish/subscribe inter-networking,” in Proc.. ACM SIGCOMM, Barcelona, Spain, Aug. 2009.
- [BAR1999] A. Barabási and R. Albert, "Emergence of Scaling in Random Networks," Science, vol. 286, pp. 509–512, 1999.
- [STO2000] I. Stoica, T.S.E. Ng and H.zhang, "REUNITE: a recursive unicast approach to multicast," in Proc. IEEE INFOCOM, March 2000.
- [NIA2008] De-Nian Yang and W. Liao, "Protocol design for scalable and adaptive multicast for group communications," in Proc. IEEE ICNP, October 2008.