EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

MATHEMATISCH-
NATURWISSENSCHAFTLICHE FAKULTÄT
**Kommunikationsnetze**

# Efficiency of BIER Multicast in Large Networks


Daniel Merling, Thomas Stüber, Michael Menth

***http://kn.inf.uni-tuebingen.de***

# Why BIER?

- Problems with IP multicast (IPMC)
  - State in core nodes (for MC groups)
  - Signalling overhead per MC group
  - Severe signalling load in case of link/node failures due to reconvergence

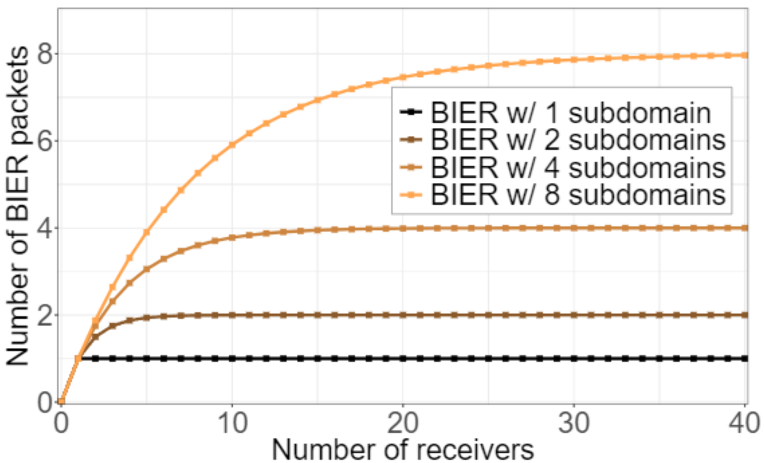
- IETF‘s answer: Bit Index Explicit Replication (BIER)
  - Define BIER domain
  - Ingress nodes (BFIRs) add BIER header w/ bitstring
  - Bitstring encodes potential receivers (BFERs)
    - Each bit position corresponds to one BFER
    - Packet delivered if bit is set to 1
  - Core nodes (BFRs) forward and replicate packets according to BIER header and “routing underlay”

# Set Identifiers (SIs)

- What if more BFERs than bits in bitstring?
  - Use multiple bitstrings and number them w/ set identifiers (SIs)

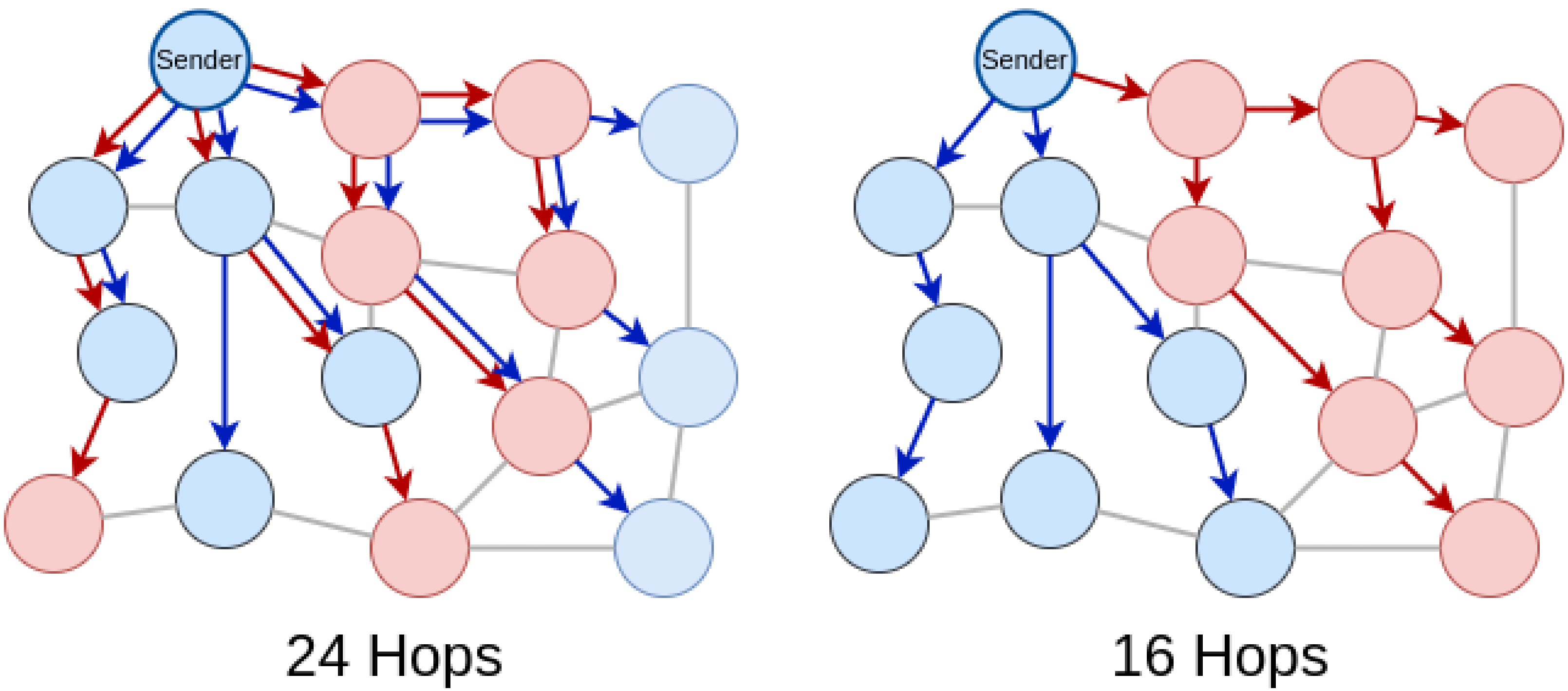
- What if receivers belong to different SIs?
  - Send one packet per SI w/ receivers
    - Multiple BIER pkts sent per IPMC pkt
    - Overhead compared to IPMC

- How many packets are required from the sender?
  - IPMC: only a single packet
  - Unicast: for every receiver
  - BIER: Number quickly approaches number of SIs with increasing number of receivers



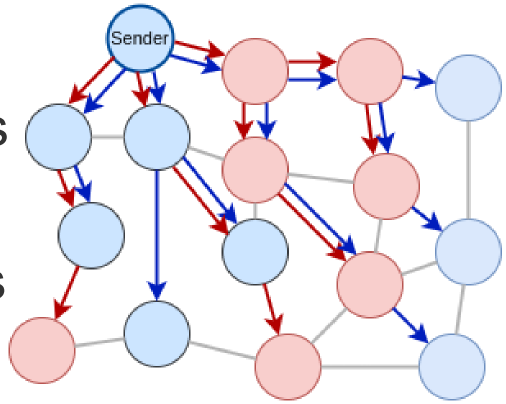
# Sets

- Links may carry multiple copies of MC packets
- (BFER) set: BFERs assigned to same SI
- Partitioning of BFERs in a BIER domain into sets matters



# Optimization of BFER Sets

- Traffic model
  - Every node sends one MC pkt to all other nodes
- Performance metric
  - Overall number of per-hop packet transmissions




24 Hops

- Methods to create sets
  - Random assignment of nodes to sets
  - Integer Linear Program (ILP) to obtain best theoretic solution
  - Fast heuristic algorithm that covers large networks

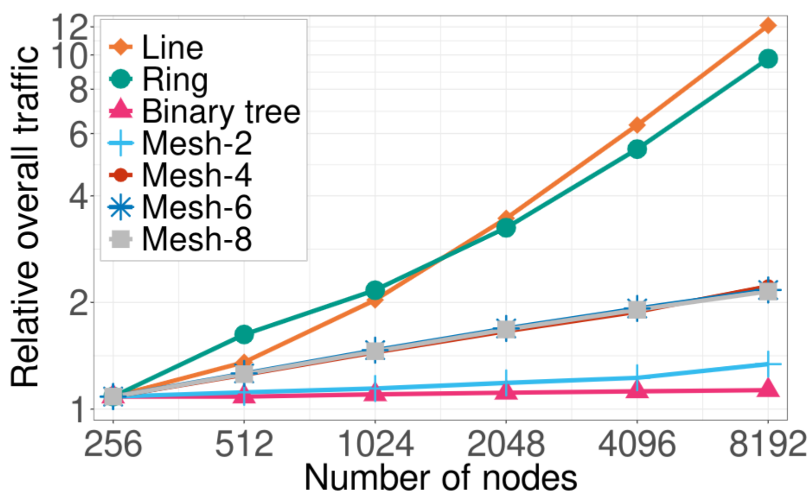
- Comparison on small problem instances
  - Heuristic is close to optimum
  - Random assignment worse for many sets

| Topology | $n$ | $s$ | Heuristic (%) | Random (%) |
|---|---|---|---|---|
| Mesh-2 | 64 | 2 | 100.3 | 132.6 |
| | | 4 | 100.7 | 162.2 |
| | 128 | 2 | 100.5 | 133.7 |
| | | 4 | 101.5 | 179.8 |
| Mesh-4 | 64 | 2 | 100.3 | 115.2 |
| Mesh-6 | 64 | 2 | 100.4 | 110.6 |
| Mesh-8 | 64 | 2 | 100.3 | 107.1 |

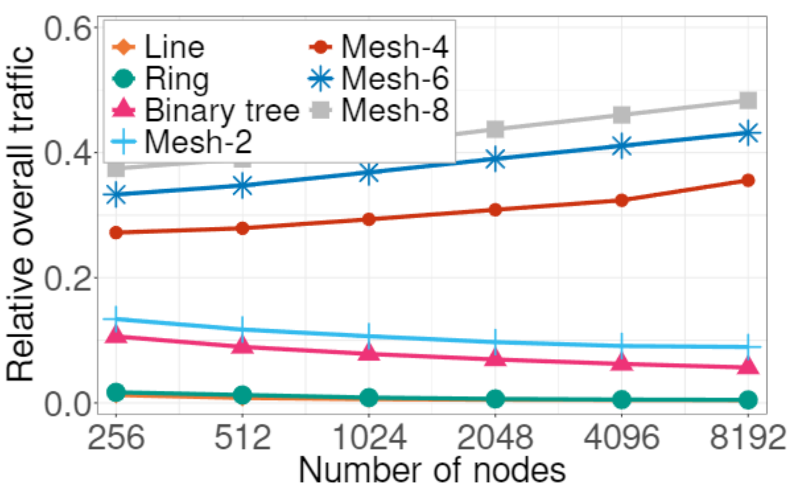
# Comparison wrt Overall Traffic

- Experiment
  - Bitstring size: 256
  - Networks w/ different topologies and numbers of nodes
  - Optimal sets (w/ heuristic algorithm)
- Results
  - IPMC more efficient than BIER, but BIER more efficient than unicast
  - Efficiency depends on topology
  - Similar results for small receiver sets (not shown)

**BIER vs. IPMC**




**BIER vs. Unicast**

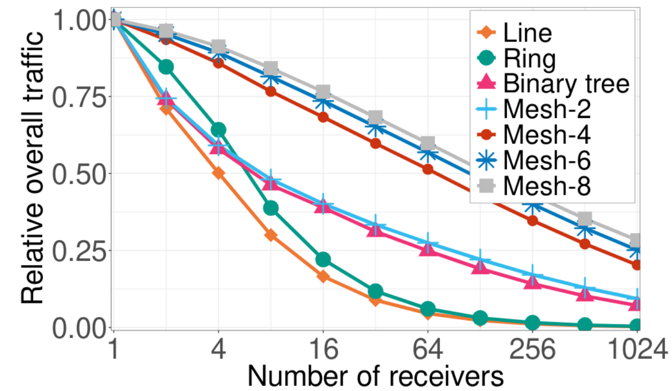


EBERHARD KARLS UNIVERSITÄT TÜBINGEN

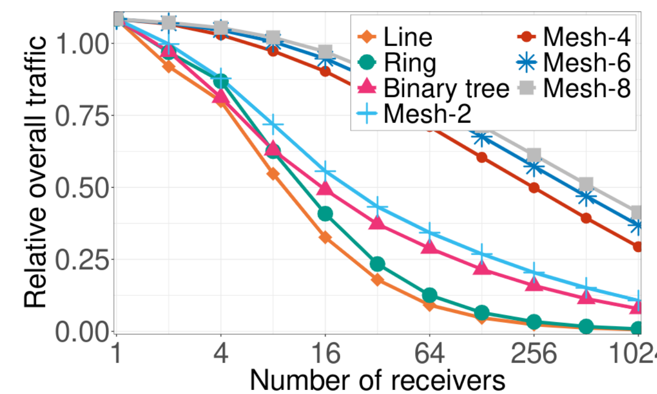
# What about smaller MC groups?

- ▶ Experiment
  - BIER-256
  - Network size 1024 nodes
- ▶ Results depend on topology
  - Effectiveness of IPMC and BIER increases w/ MC size
  - Good in sparse topologies
  - Effectiveness of BIER suffers especially for small MC groups

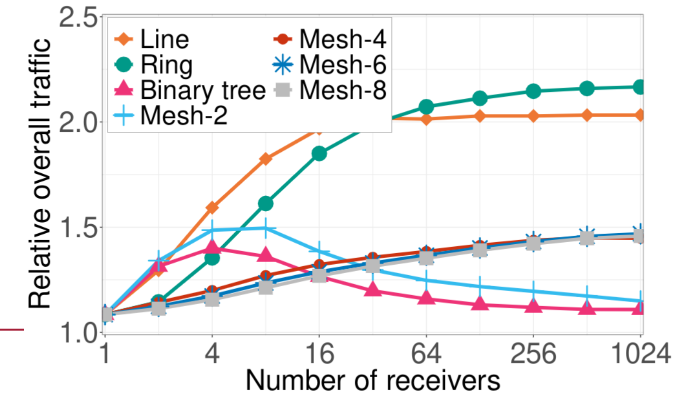
**IPMC vs. unicast**




**BIER vs. unicast**



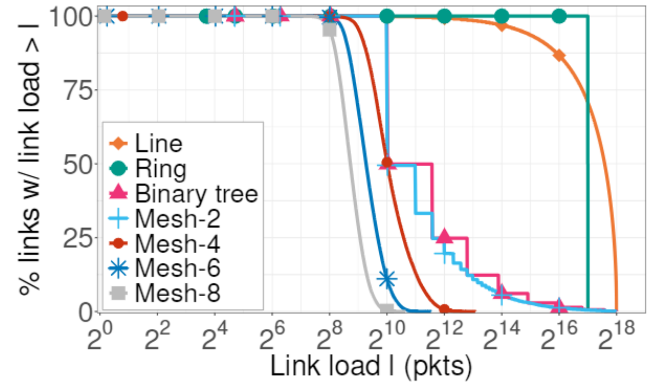

**BIER vs. IPMC**



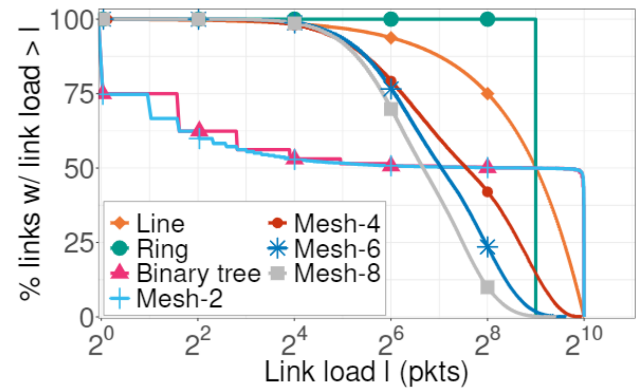
# Comparison: Load on Central Links

- Experiment
  - BIER-256
  - Network size 1024 nodes
  - Every node sends pkt to all other nodes
  - Metric: overall #pkts on links
- Complementary cumulative distribution function (CCDF) of link loads
  - Indicates percentage of links w/ load > l (pkts)
  - Most loaded links easy to see
- Results depend on topology
  - Unicast: up to $2^{14} – 2^{18}$ pkts
  - IPMC: up to $2^{8} – 2^{10}$ pkts
  - BIER: up to $2^{9} – 2^{11}$ pkts
- IPMC and BIER effectively reduce load on most loaded links

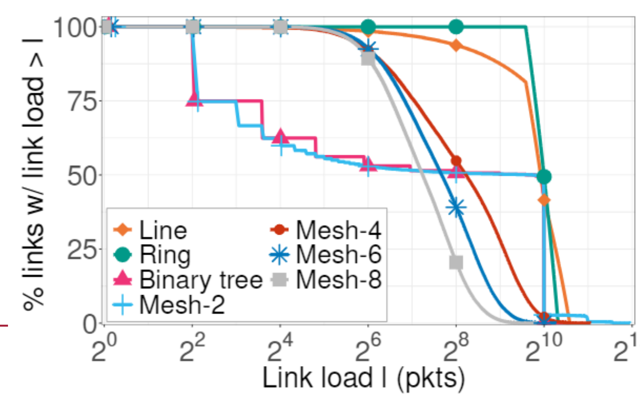
**Unicast**




**IPMC**

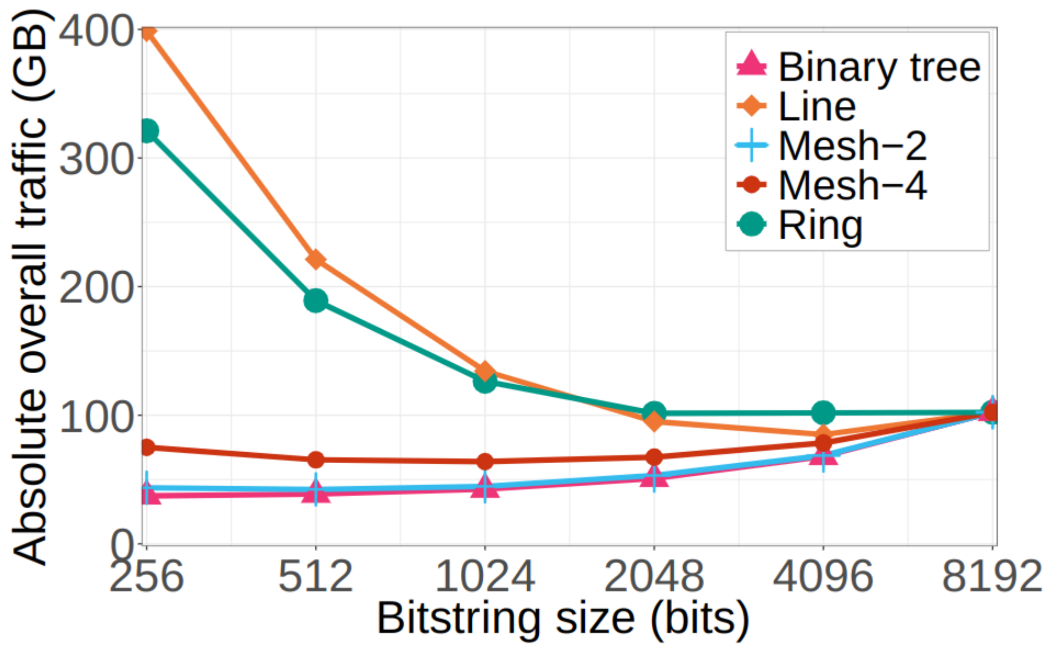



**BIER**



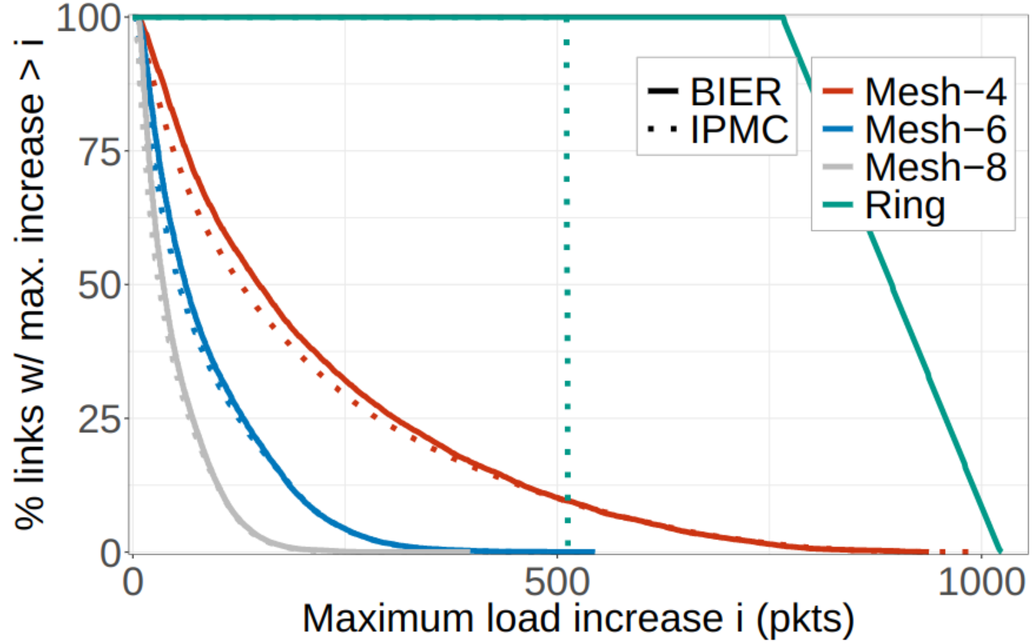
# What is the best bitstring size?

- Tradeoff
  - Large bitstring: few pkts but large BIER header
  - Small bitstring: many pkts but small BIER header
- Experiment
  - Network size 8192 nodes
  - Pkt size: 500 byte payload + IPMC header + BIER header
  - Every node sends pkt to all other nodes
  - Metric: overall traffic (GB)
- Results
  - There is an optimum bitstring size
  - Line and ring networks benefit from large bitstrings
  - But BIER-256 bits is good enough for all other network topologies



# What about forwarding in failure cases?

- Problem
  - Sets optimized for routing w/o failures
  - What happens to link loads in failure cases?
- Experiment
  - Network size 1024 nodes
  - BIER-256
  - Every node sends pkt to all other nodes
  - Check routing for all single-link failures
  - Metric: maximum load increase on any link
  - Consider only resilient topologies
- Results
  - Load increase on links almost identical to IPMC
  - Ring is an exception



# Open Issues / Next Steps

- Input appreciated for more realistic evalutions
  - What topologies are realistic?
  - What are typical MC group sizes?

- What about small MC groups in large networks?
  - Multiple BIER pkts needed for receivers in different sets

- New kids on the block
  - Explicit tree structures in pkt headers combining BIER and SR ideas
    - https://datatracker.ietf.org/doc/draft-eckert-bier-rbs/
    - https://datatracker.ietf.org/doc/draft-eckert-pim-rts-forwarding/
  - See talk on „P4 Tofino Implementation Experiences with Advanced Stateless Multicast Source Routing“ in BIER WG

# Conclusion

- BIER requires sets for scaling to large networks
  - Send one pkt per set w/ receivers
  - Developed fast heuristic to find sets
- Performance comparison: BIER vs. IPMC and Unicast
  - Packets from sender
  - Overall network load
  - Load on most loaded links
  - Load increase in failure cases
- Bitstring w/ 256 bits is good enough for most network topologies
  - Exceptions: line and ring, small MC groups
- Reference
  - D. Merling, T. Stüber, and M. Menth: Efficiency of BIER Multicast in Large Networks, IEEE Transactions on Network and Service Management (TNSM), vol. N/A, pp. N/A, 2023 (Early Access)
  - https://atlas.cs.uni-tuebingen.de/~menth/papers/Menth21-Sub-5.pdf