EBERHARD KARLS UNIVERSITÄT TÜBINGEN

FACULTY OF SCIENCE
Communication Networks

P4 Tofino Implementation Experiences with Advanced Stateless Multicast Source Routing

Michael Menth, Steffen Lindner, Toerless Eckert

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

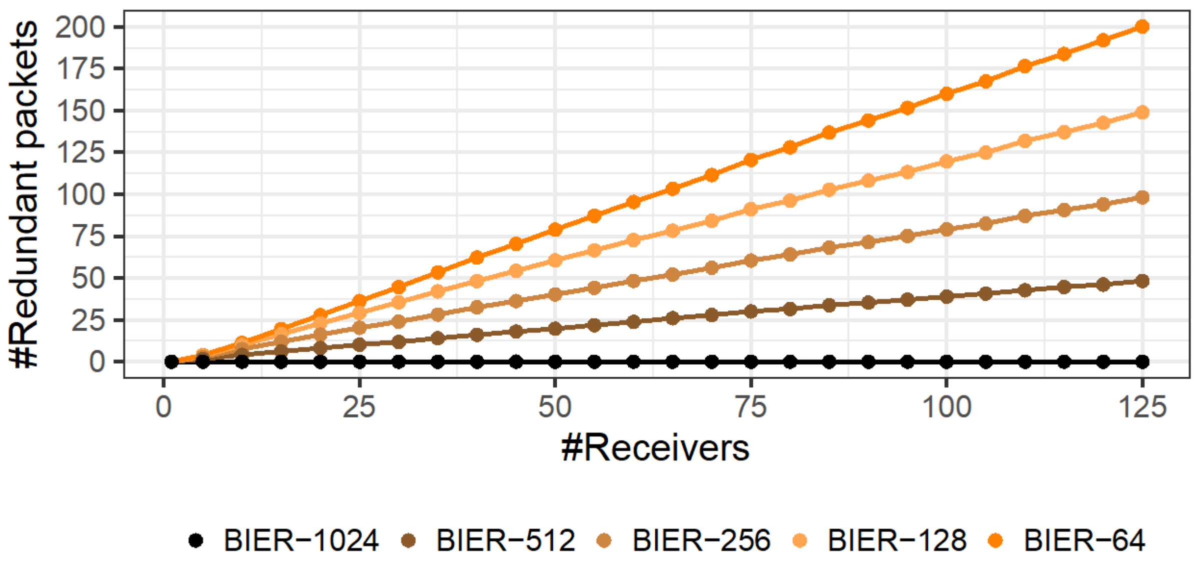
Agenda

- Motivation
- Concept of Advanced Stateless Multicast Source Routing
  - BitString-Encoded Explicit Trees (BEET)
  - Segment-Encoded Explicit Trees (SEET)
  - Combination thereof
- Conclusion

Motivation

- BIER bitstring typically 256 bits
  - Not large enough to cover large networks
- Partition BIER domain into sets ≤ 256 nodes
  - Use set identifier (SIs) to indicate bitstrings for that set
  - Helps to scale BIER to large domains
- But …
  - Large BIER domains require many sets to support all BFERs
  - One packet sent per set w/ a receiver
  - Leads to redundant packet copies if receivers are in different sets

- Experiment
  - BIER domain with 1024 nodes, average node degree: 4
  - BIER-X: bitstring length of x bit
  - BFERs randomly assigned to SIs
  - Random source sends BIER packet to *n* random receivers; averaged over 50 runs
  - #Redundant packets = (#Packets in BIER domain on all links) – (#Packets in IPMC domain on all links)

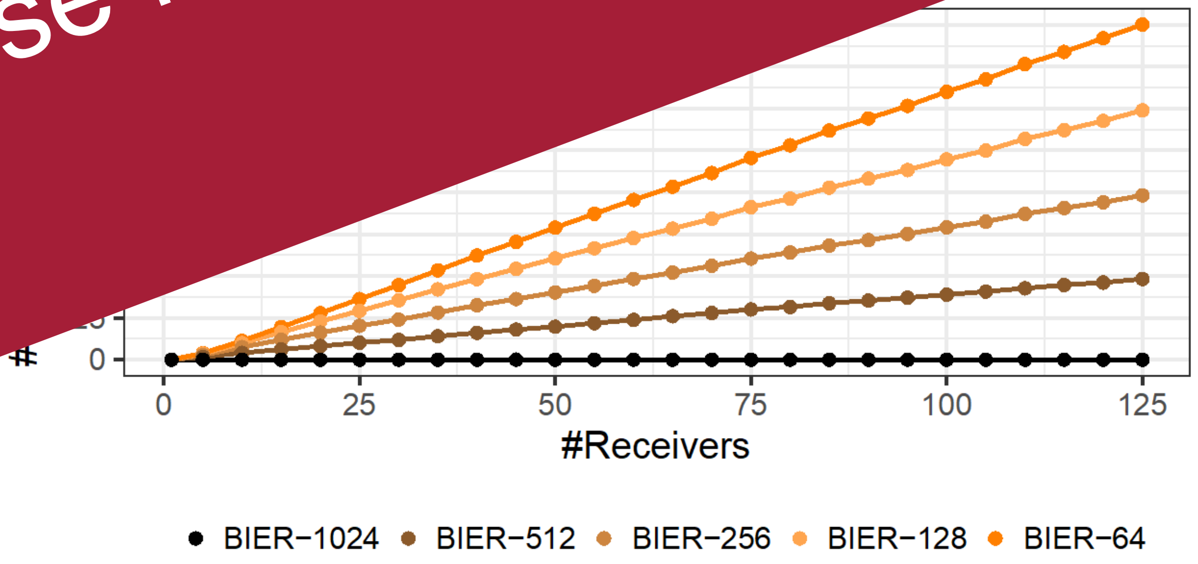


Motivation

- BIER bitstring (e.g., 256 bits) not large enough to cover large networks
- Set identifiers (SIs) used to support multiple bitstrings inside the same BIER domain
- SIs help to scale BIER to l… domains
  - Large BIER … to support all …
  - One packet ne… addressing a B…
  - Leads to redund… receivers are in di…

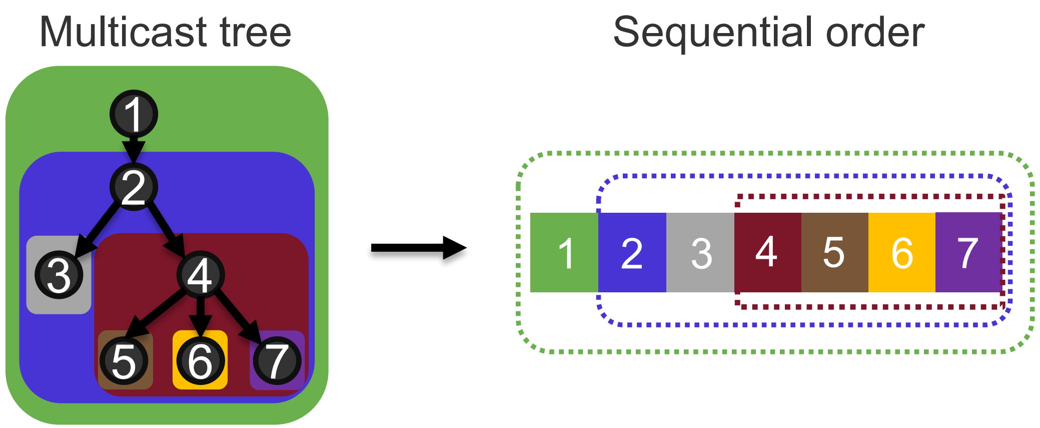
- Example: BIER do… and nodes randomly a…
  - A…
  - …n receivers;
  - …on all links)

Even worse for BIER-TE !

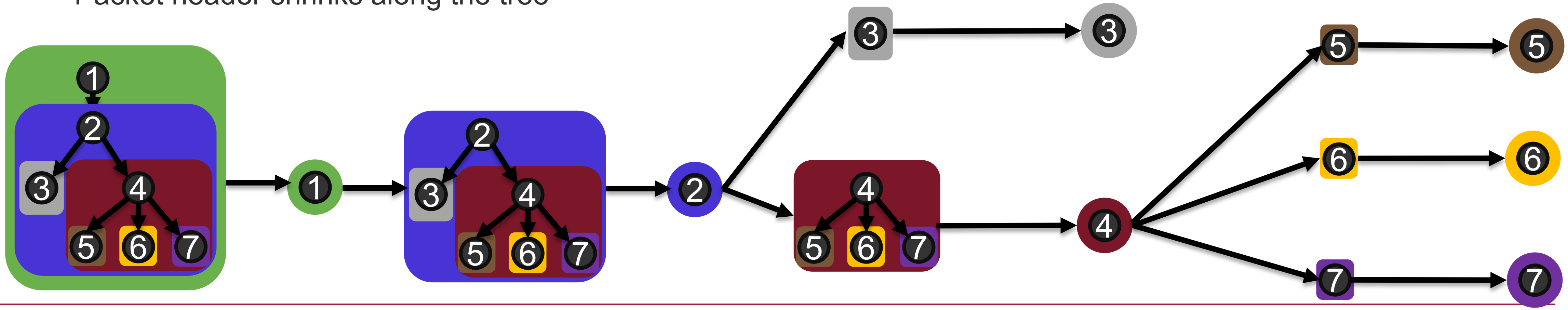


Concept of Advanced Stateless Multicast Source Routing (1)

- Idea
  - Convert distribution tree into a list and encode it in the header
    - Instead of using a flat bitstring
  - No need for sets or SIs
  - Send multiple packets only if header size does not suffice



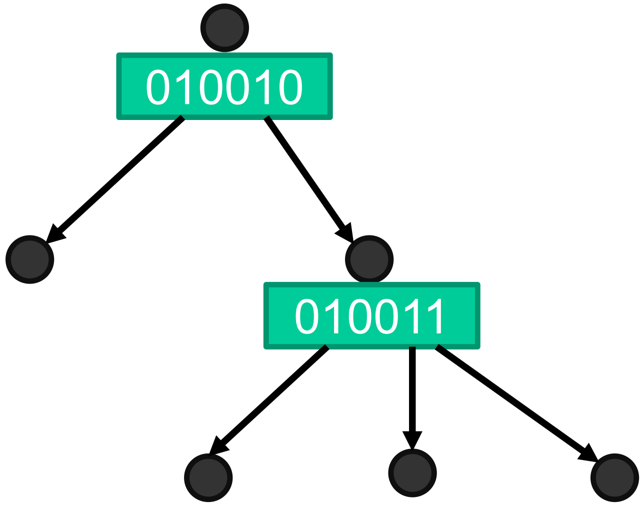
- Forwarding principle for replication nodes
  - Partition tree in header into subtrees
  - Send packet copies encapsulated in headers with single subtree to next hops
  - Packet header shrinks along the tree



Concept for Advanced Stateless Multicast Source Routing (2)

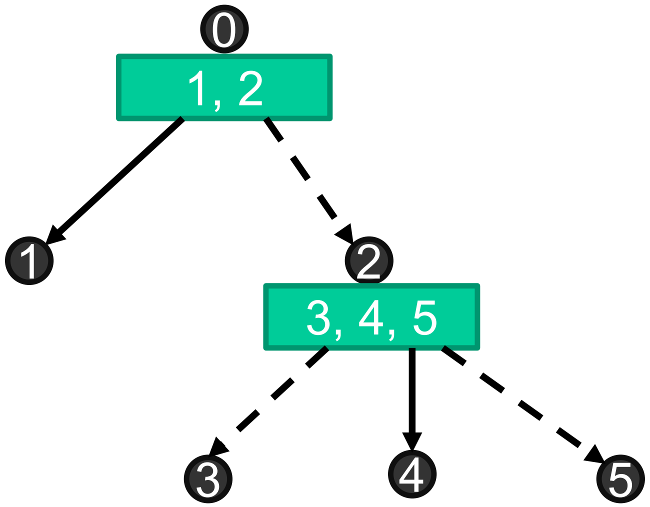
Bitstring-Encoded Explicit Trees (BEET)

- CGM2 / RBS
  - Use bitstring to address next hops
  - Already presented in BIER-WG
    - draft-eckert-bier-cgm2-rbs-01
    - draft-eckert-bier-rbs-00



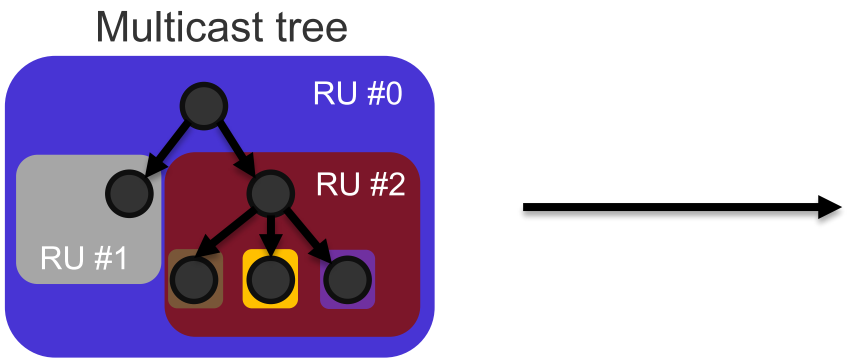
Segment-Encoded Explicit Trees (SEET)

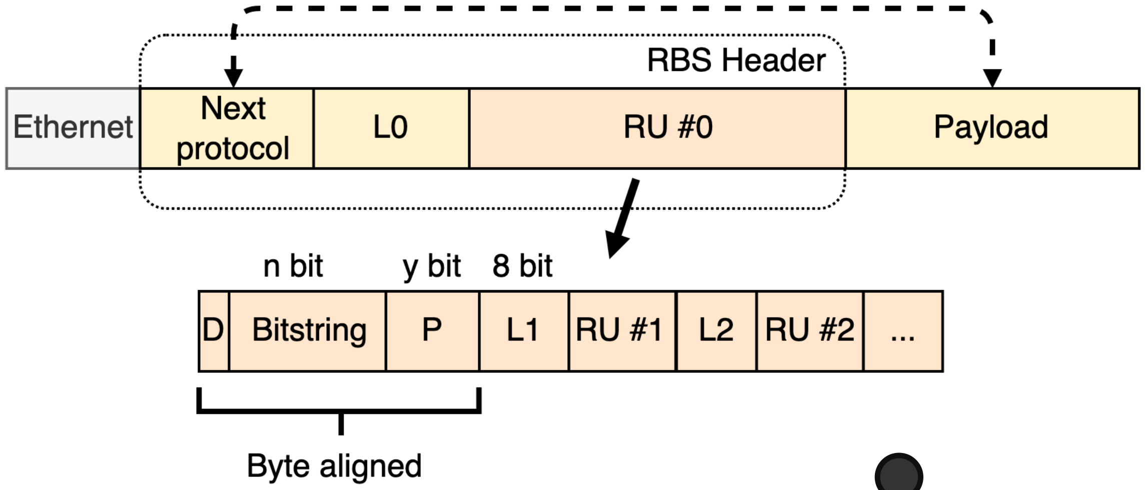
- Use domain-wide IDs to address next hops
  - New concept, included in
    - draft-eckert-pim-rts-forwarding



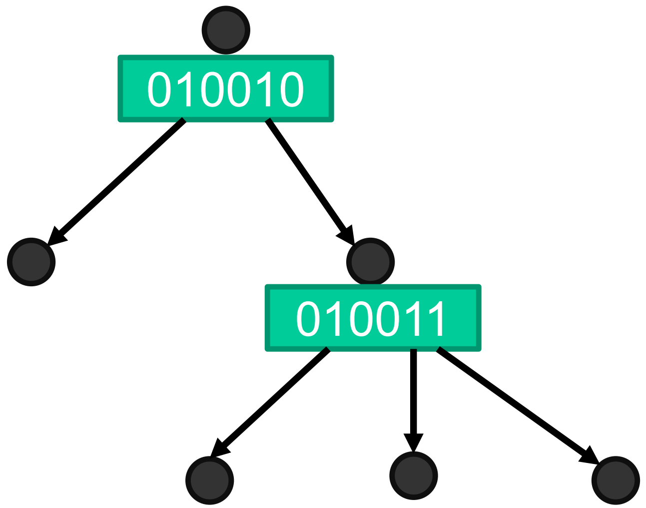
- Proof of concept
  - Prototype of “RBS Light“ and SEET on P4/Intel Tofino™ (100 Gb/s)
  - Whatever is implemented with P4 on Intel Tofino should be simple enough for other platforms

Recursive BitString Structure (RBS) (1)

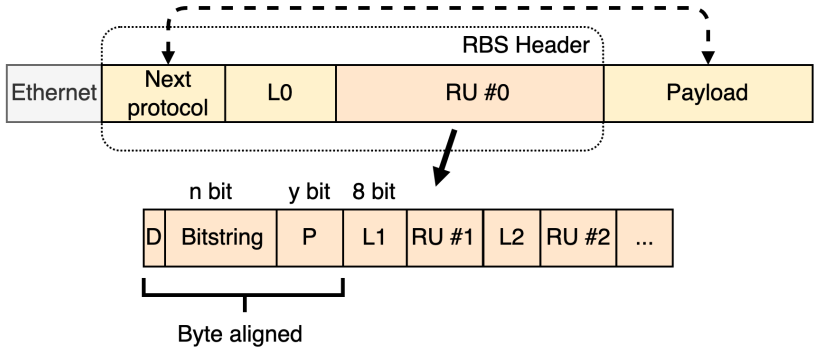


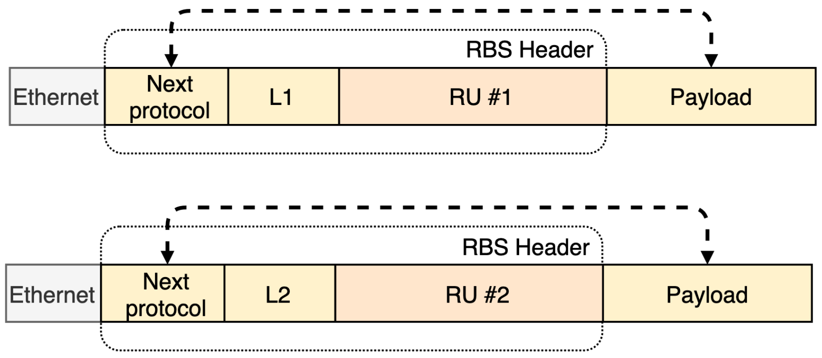


- Header structure
  - Next protocol: identifies protocol of payload
  - LX: length of the next recursive unit (RU #X) in byte
  - RU #X: recursive unit (RU) of length LX
    - Bitstring: local fixed-size BIER-like bitstring identifies neighbors
      - Different nodes can have different bitstring lengths
      - D: first bit in bitstring is “deliver bit”, D=1 → processing node receives a packet copy
    - B: broadcast bit → triggers local broadcast to a pre-defined group of neighbors
    - P: padding → bitstring + B + P need to be byte-aligned



Recursive BitString Structure (RBS) (2)





Forwarding operation

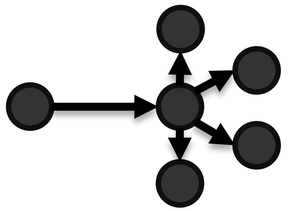
- For each activated bit in Bitstring
  - Copy LX and the next LX byte (RU #X) into a new packet
  - Forward packet copy to the neighbor that is identified by that bit

Optimization

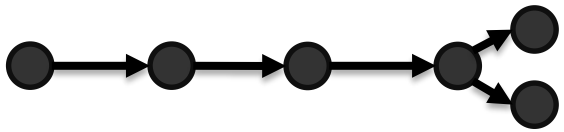
- Explicit leaf nodes
- Enough to send them a non-encapsulated packet copy
- Do not require an own RU ⇒ saves header space
- Not implemented

Recursive BitString Structure (RBS) (3)

- Advantage
  - Efficiently encodes packet replication to many neighbors

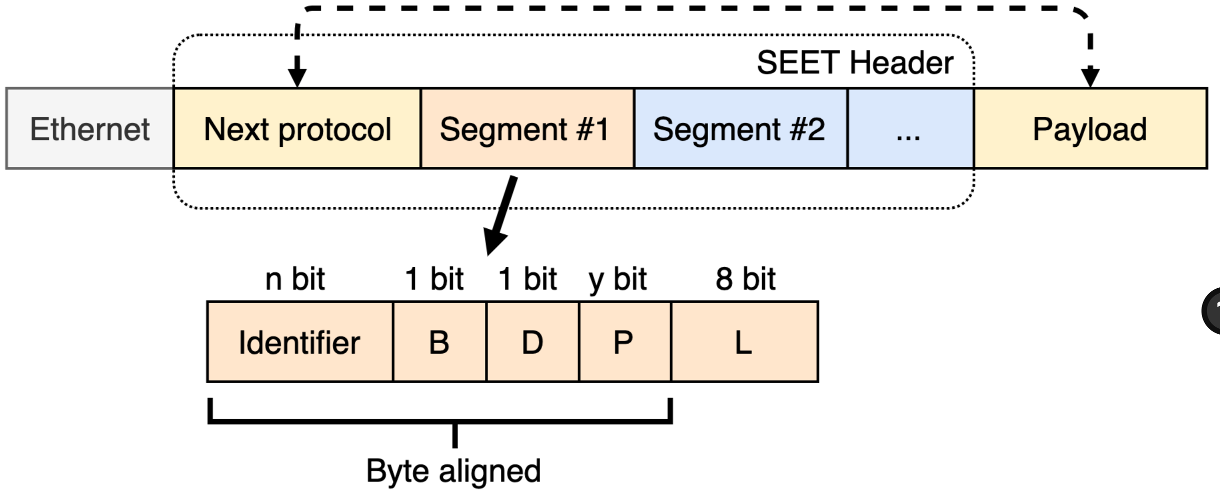


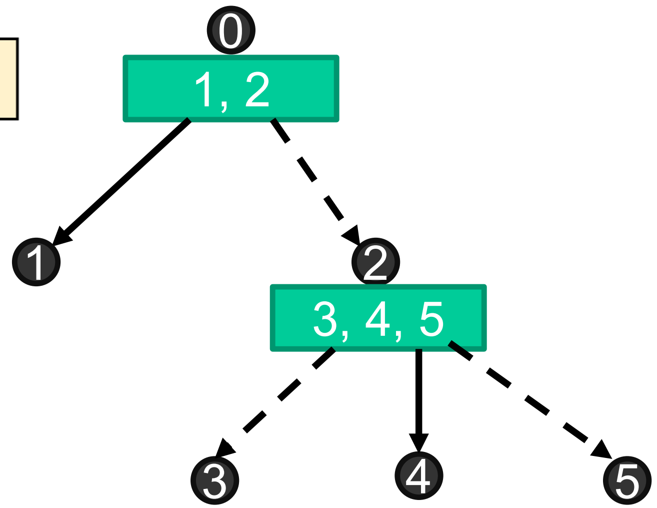
- Disadvantage
  - Less efficient if multicast tree has long “line” paths



Segment-Encoded Explicit Trees (SEET) (1)

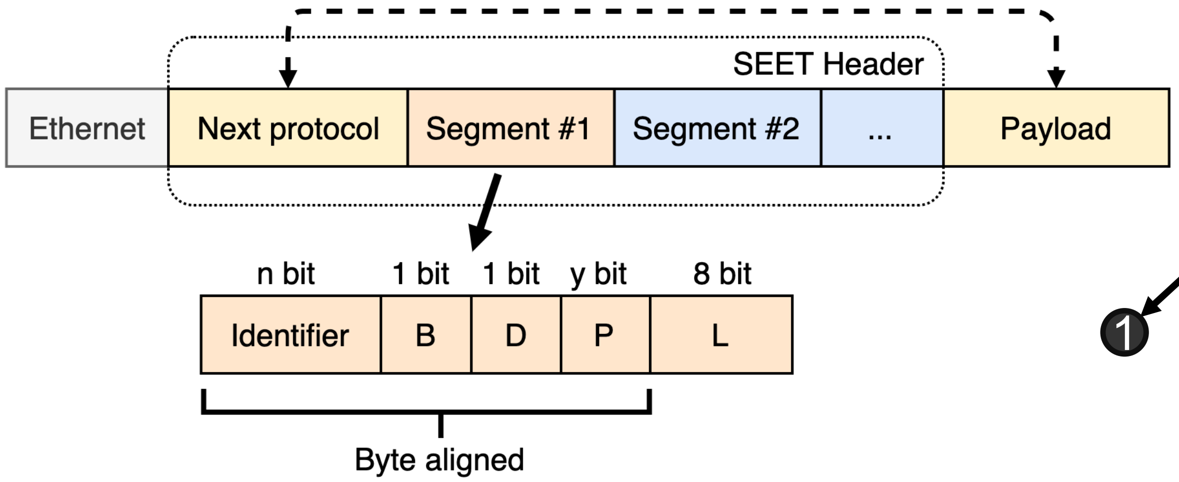
- Use segment IDs (SIDs) instead of “local bitstring” to address neighbors
- SIDs can have domain-wide meaning
  - Allows addressing of remote nodes several hops away
- Also implemented on P4/Intel Tofino™

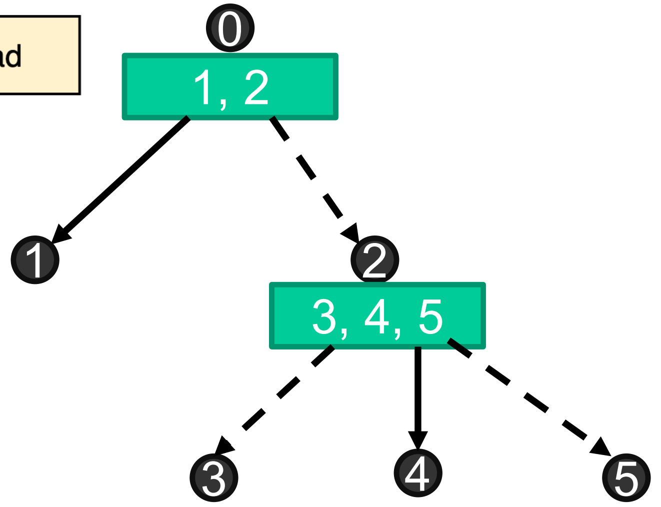




Segment-Encoded Explicit Trees (SEET) (1)

- Use segment IDs (SIDs) instead of "local bitstring" to address neighbors
  - SIDs can have domain-wide meaning
  - Allow addressing of remote nodes several hops away

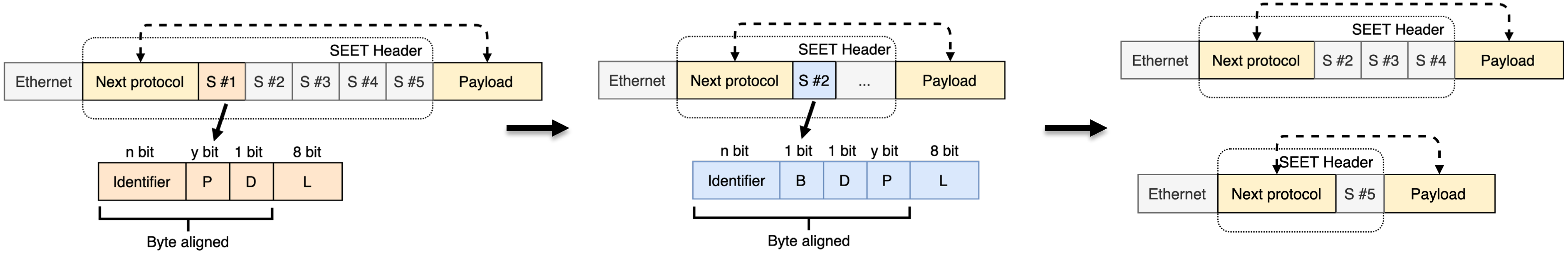




- Header structure
  - Next protocol: identifies protocol of payload
  - Multiple segments
    - First segment relates to next hop

- Segment structure
  - Identifier: SID
  - Instructions for the denoted node
    - B: broadcast bit → triggers local broadcast to a pre-defined group of neighbors
    - D: deliver bit → denoted node receives a packet copy
    - P: padding → Identifier + B + D + P needs to be byte-aligned
    - L: number of following header bytes for the denoted node

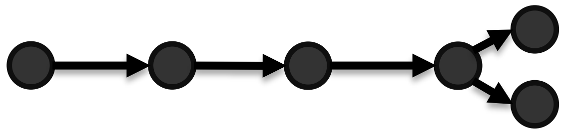
Segment-Encoded Explicit Trees (SEET) (2)



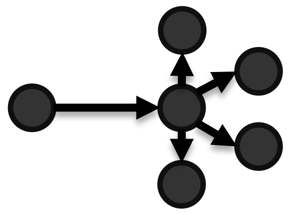
- Forwarding operation
  - Check SID in first segment identifies processing node
  - Check D-bit
    - Yes: deliver a copy to processing node
  - Remove first segment
  - Repeat until original packet empty
    - Copy next segment and the next L byte into a new packet
    - Forward new packet according to SID in first segment
    - Remove the segment and the next L byte from original packet

Segment-Encoded Explicit Trees (SEET) (3)

- Advantage
  - Efficiently encodes multicast trees with long “line” paths



- Disadvantage
  - Less efficient for replication to many neighbors

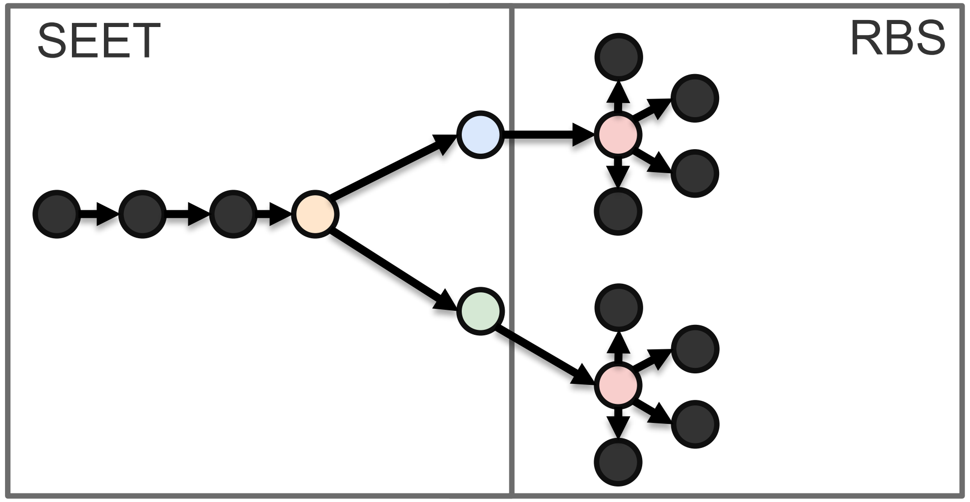


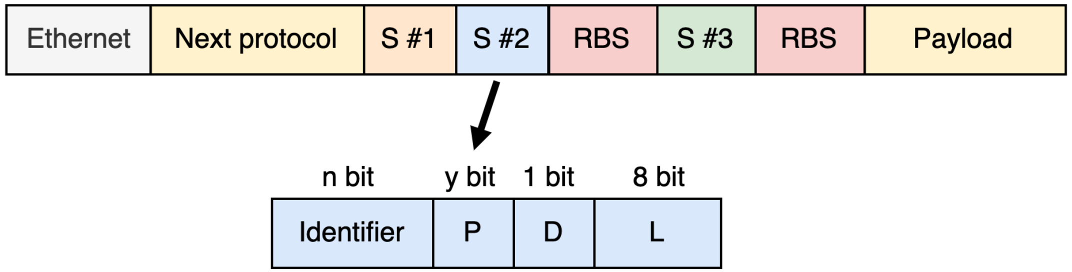
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Combination of SEET and BEET

- SEET efficient for sparse multicast trees with long lines
- BEET efficient for multicast trees with many leaves at some penultimate node

- Idea
  - Use SEET to reach penultimate nodes
  - Use encapsulated bitstring at penultimate hops to efficiently replicate to many leaf nodes

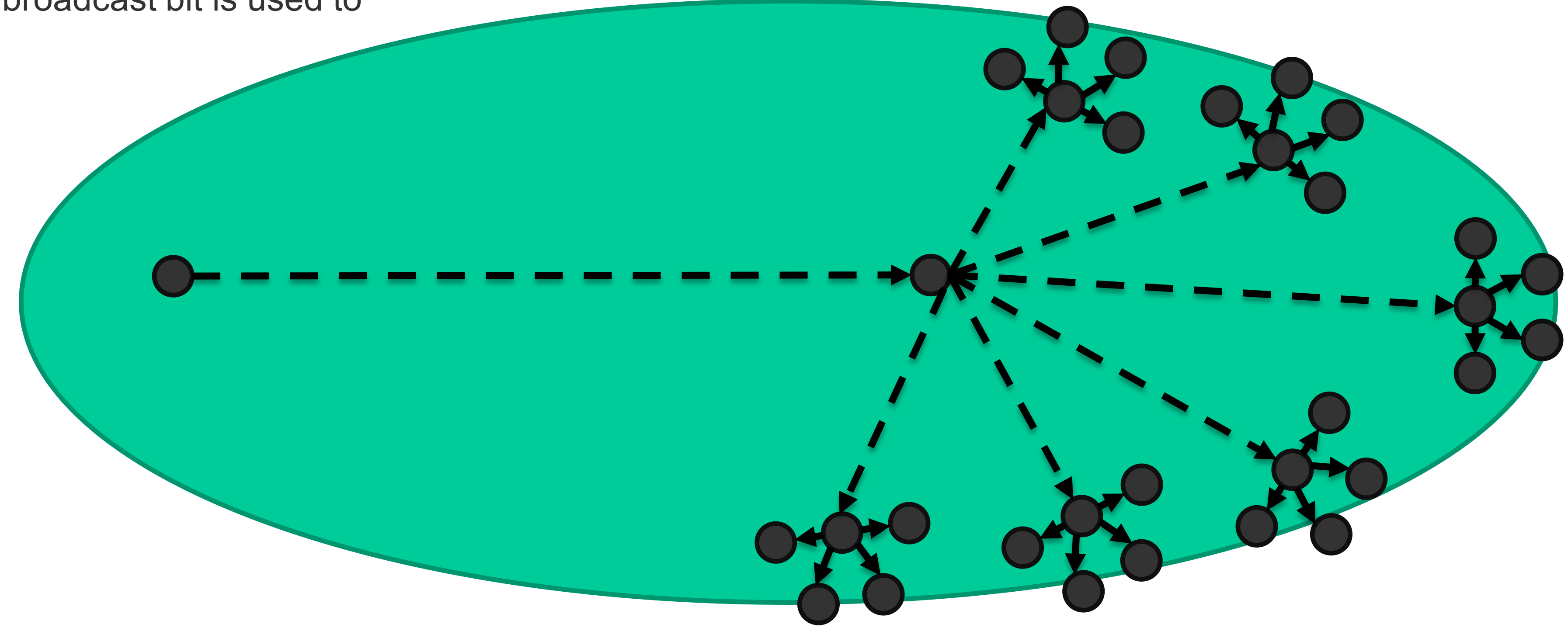




Encode in S #2 that RBS header follows

Special Case

- „Tunnel 8-bit bitstrings“ to penultimate hops over two hops
  - SID (2 bytes)
  - Length field (1 byte)
  - Bitstring (1 byte)
- Efficiency even larger if broadcast bit is used to reach more hops
- Header size 256 bits (32 bytes)
  - ⇒256/8/4-1=7 penultimate hops can be addressed
  - ⇒7*8=56 receivers can be addressed
- Header size 1024 bits (128 bytes)
  - ⇒1024/8/4-1=31 penultimate hops can be addressed
  - ⇒31*8=243 receivers can be addressed



Conclusion

- BEET / SEET allow encoding of generic multicast trees
  - Scales better than BIER(-TE) in large domains with sparse multicast trees
- Implemented on P4/Intel Tofino^{TM} with 100 Gb/s
- Advantages of SEET and BEET can be combined
  - Tunneling of bitstrings over SEET to address leaf nodes of the tree
  - Not yet implemented
- Technical report with documentation to come