



# BIER Ping

IETF 92

draft-kumarzheng-bier-ping-00

Nagendra Kumar

Carlos Pignataro

Nobo Akiya

Mach Chen

Vero Zheng

[Greg Mirsky](#)

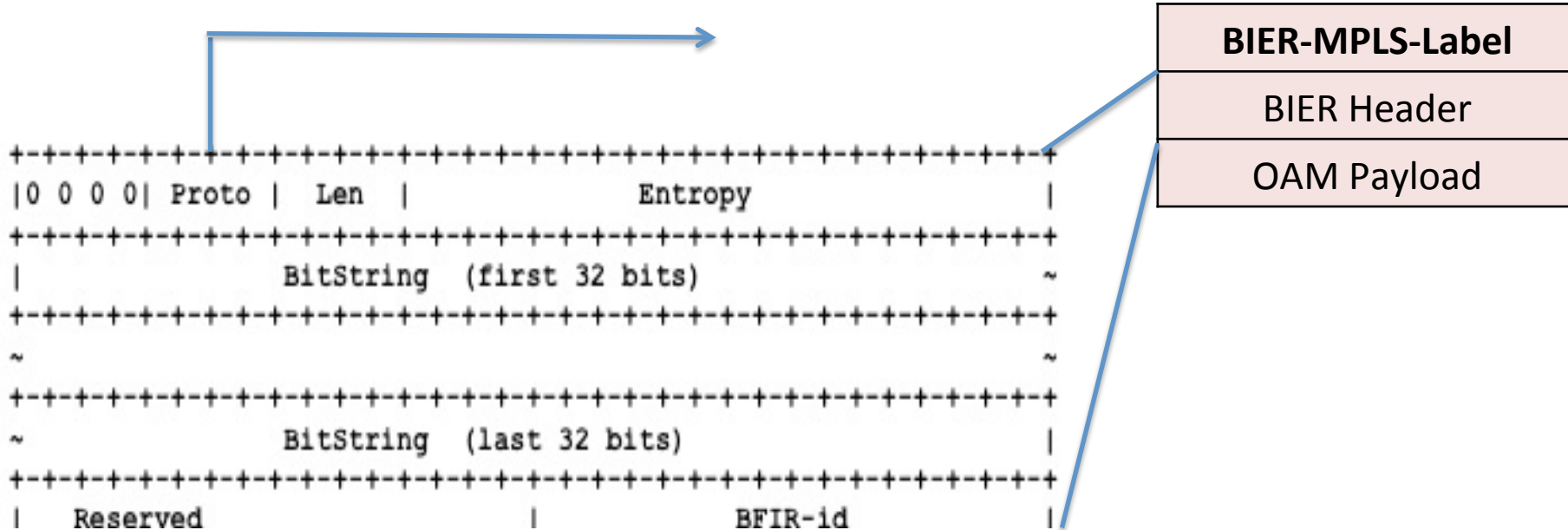
# OAM Requirement

- Layer Independent and transport agnostic
  - Work on the BIER layer itself and avoid any dependency on other layers.
  - Work on all BIER supported transport.
- Avoid leaking OAM packet outside the BIER domain.
- One tool for OAM purpose.
  - Continuity Check
  - Fault Detection and Isolation
  - Performance Measurement
- Capability to perform ECMP path discovery and path validation.
- OAM payload should be flexible to accommodate the OAM functionality in different BIER use cases.

# Why not existing tools

- Historical Multicast OAM tools are hard to extend for BIER.
  - Mtrace, Ping
- LSP Ping is good, but specific to MPLS transport.
- Creating transport agnostic BIER OAM by leveraging the characteristic and benefits of LSP Ping is more reasonable.

# BIER OAM packet format



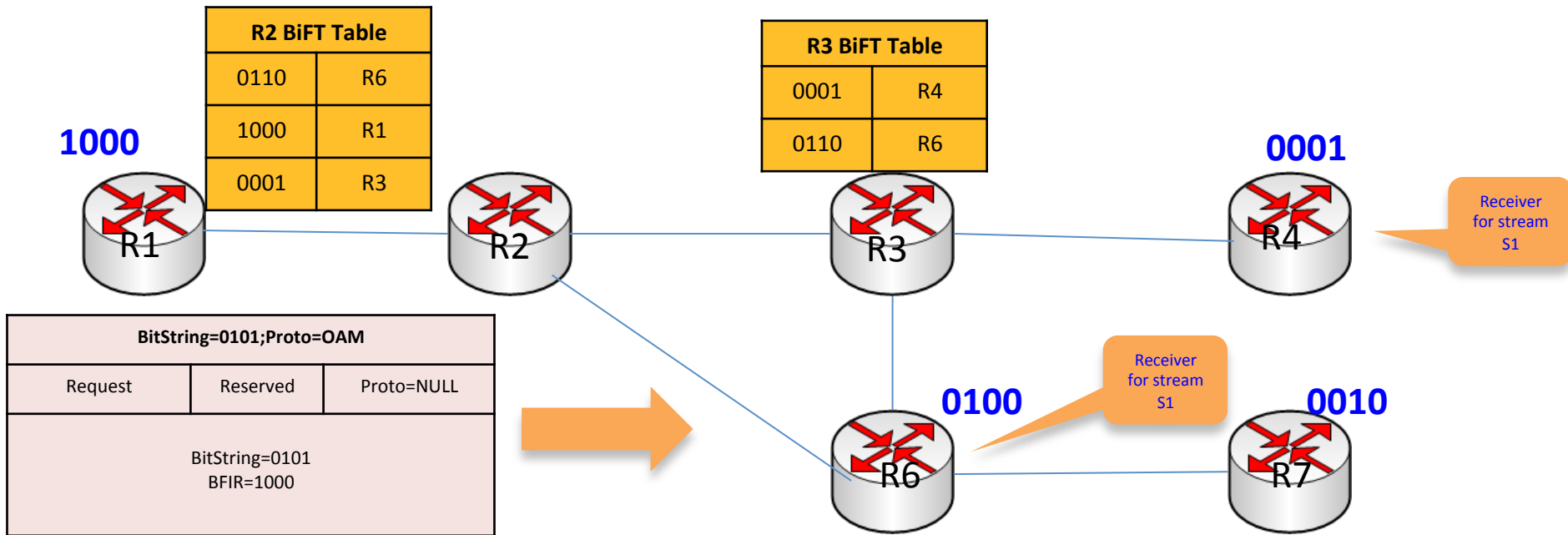
## Interpreting OAM packet:

- Should be BFER.
- BIER-MPLS label TTL expired.
- Presence of RA label in label stack

## Various TLV for different purpose

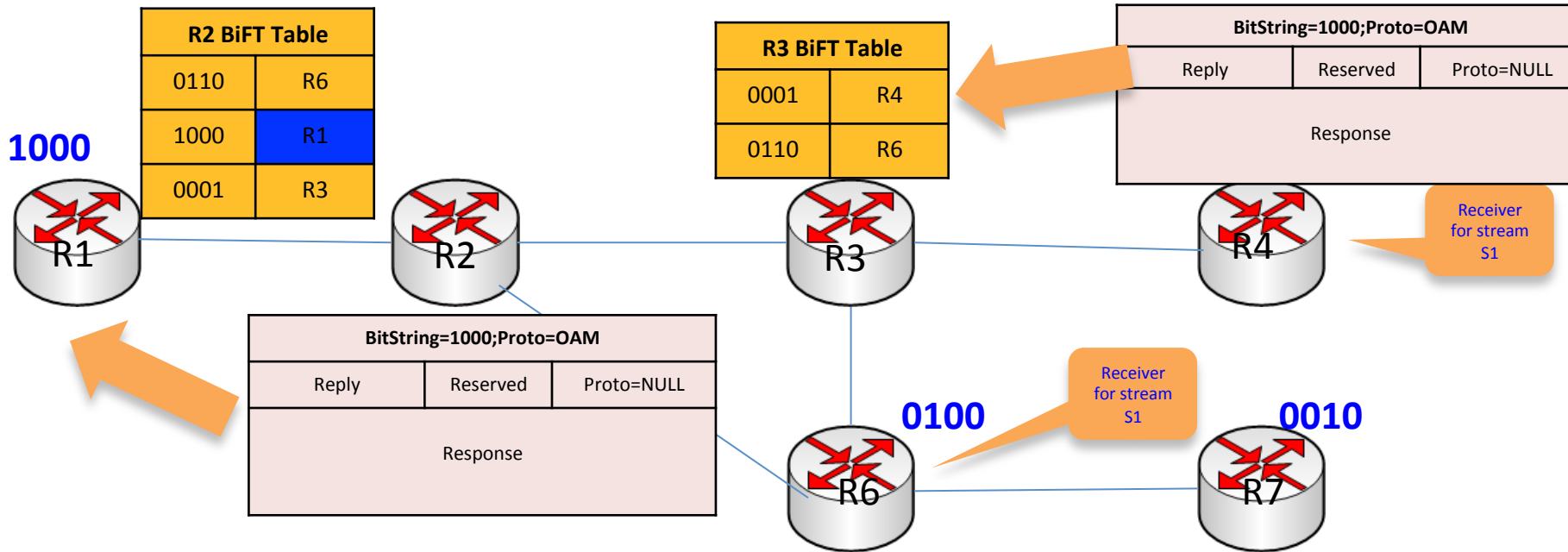
- ECMP Discovery
- Downstream/Upstream details
- Received BitString details
- etc

# Connectivity Verification – Ping (Initiator behavior)



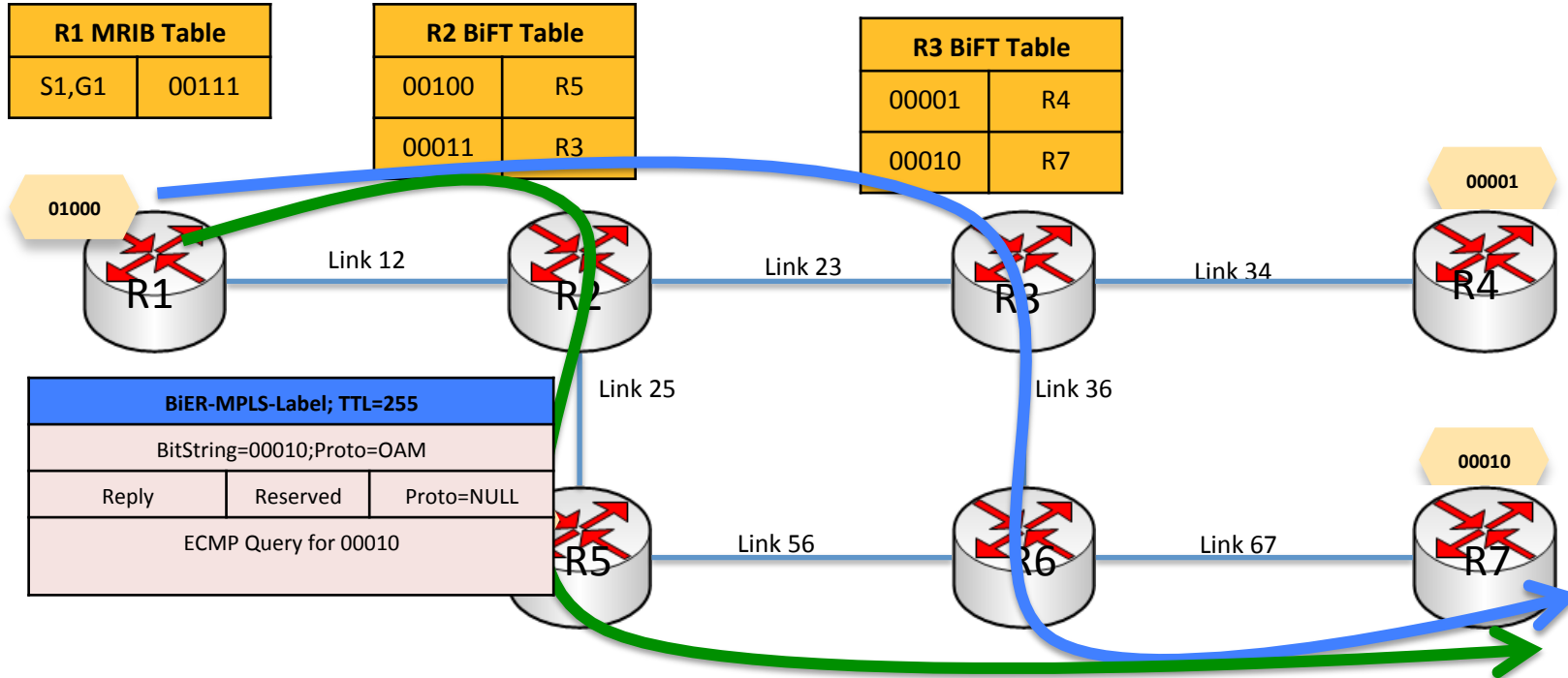
- R1 will generate OAM packet as below:
  - Content carrying all BitString to be validated.
  - Include BFIR details.
  - Set Proto=NULL in OAM packet.
  - Set the Message type as TBD1 (Request)
  - Include BiER Header, set O bit and set Proto=OAM.
- Each BFR will follow BIFT table and send to downstream BFRs.

# Connectivity Verification – Ping (Responder behavior)



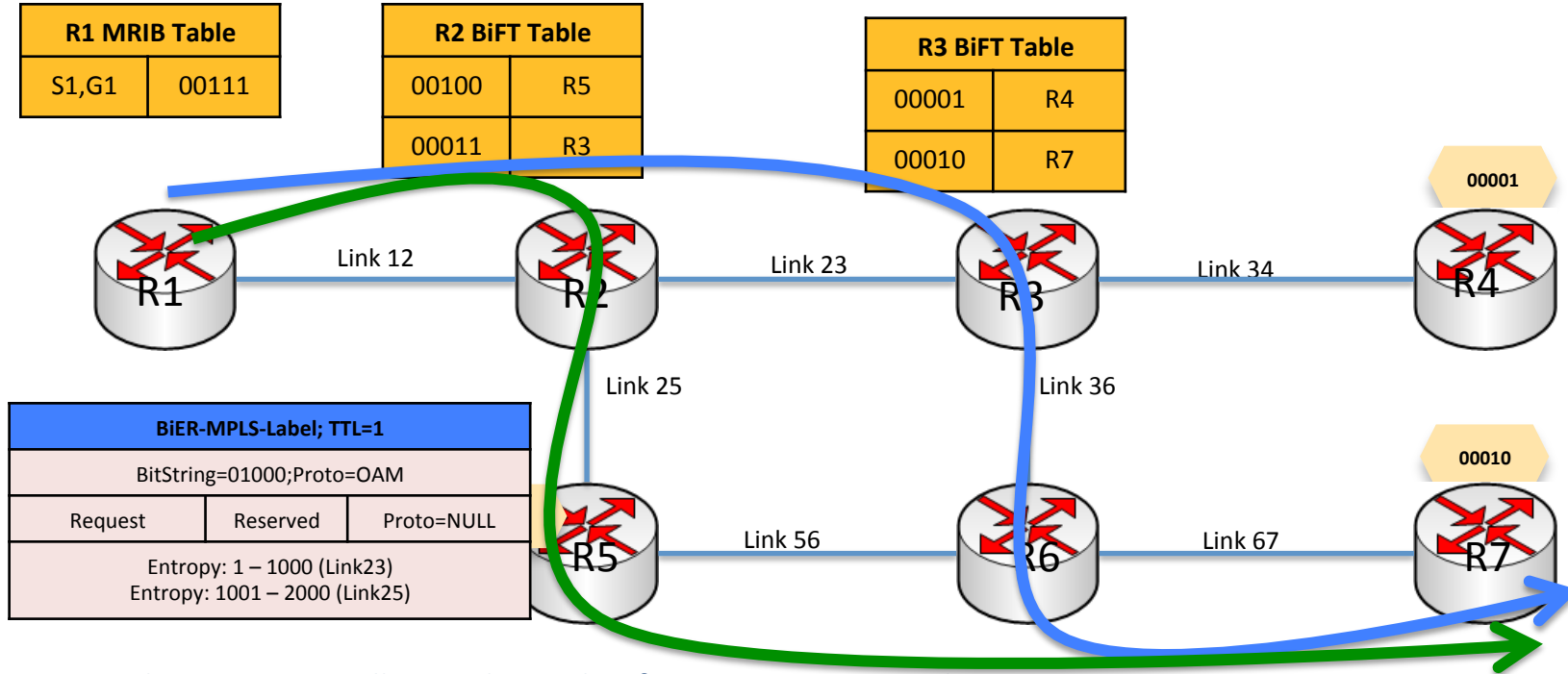
- Transit BFR (Ex; R3) on receiving the packet will simply forward based on BiFT table.
- BFER will use Proto field to punt for OAM processing.

# Path Trace and ECMP Discovery – Initiator behavior



- In the above topology R1 have 2 possible ECMP paths between R1 and R7 as below:
  - PATH1 – R1-R2-R3-R6-R7
  - PATH2 – R1-R2-R5-R6-R7
- R1 will generate OAM packet as below:
  - Content carrying Bit ID for which ECMP trace to be performed. (In this case, 00010)
  - Include BFIR details. (Use I bit)
  - Set Proto=NULL in OAM packet.
  - Set the Message type as TBD1 (Request)
  - Include BiER Header (for specific BFER), set O bit and set Proto=OAM. Start from TTL=1 and increment for each reply.

# Path Trace and ECMP Discovery – Responder behavior



- Each transit BFR will punt the packet for OAM processing due to TTL expiry.
- OAM module will reply back with Entropy value range for each downstream link. In our case, R2 will reply as below:
  - Entropy = (1-1000) Downstream Interface: Link23
  - Entropy = (1001 – 2000) Downstream Interface: Link25
- R1 will continue the query to build the entropy table and then uses the same to validate each ECMP path.



## Next Steps?

- OAM requirement as an informational draft??
- Good Discussion in the list.
- Comments will be included in next revision.