TraceFlow

Arun Viswanathan
Subi Krishnamurthy
Vishal Zinjuvadia
Rajeev Manur

--Force10 Networks

Agenda…

- Motivation
- Problem Statement
- Use-case scenario
- TraceFlow Protocol Operation
- Legacy (non-compliant devices) support
- Encapsulation
- Open Issues
- Q & A

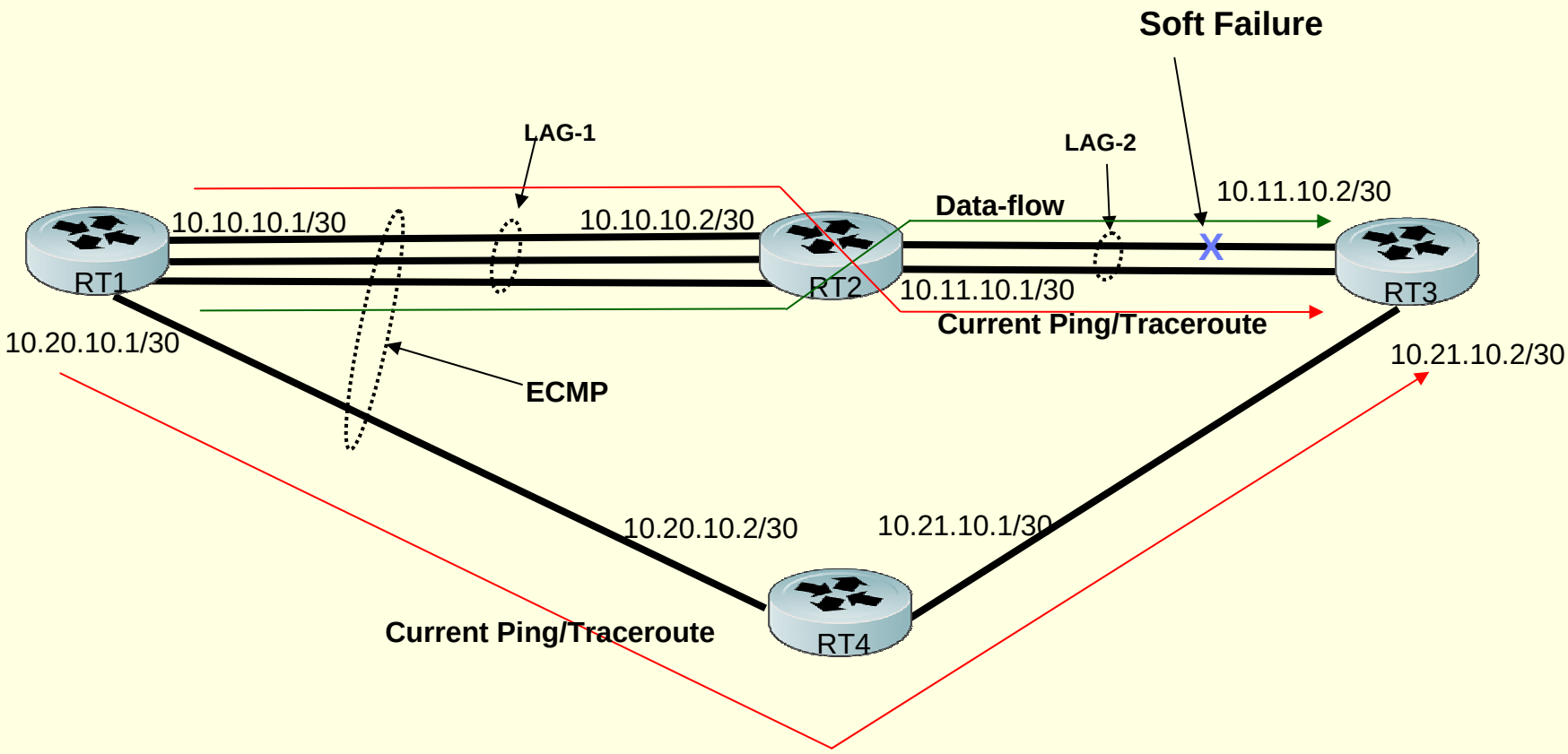
Motivation…

- Today's networks have evolved into more complex heterogeneous entities
  - LAG & ECMP are commonly used for redundancy and load-balancing
  - E2E paths are typically an intermix of L2/L3/MPLS hops
- OAM tools have not kept pace with the OAM requirements of evolving networks.
  - Destination address based Ping/Traceroute do not provide extra level of information needed in networks that use LAG/ECMP/flow based forwarding function like ACLs, Policy Based Routing
  - Need a mechanism to collect specific data along the flow path to help diagnose the network problems better.
- Many of our customers (SPs, Datacenters) would like to have a solution to this problem.

Problem Statement…

- Need to extend ping/traceroute to have,
  - Ability to trace 3/5 tuple user defined flow to exercise the same ECMP path or component link in a LAG as that taken by corresponding data packets. (An example on the following slide)
  - Ability to selectively collect relevant diagnostic data along the flow path.
- Increased security built into the protocol for tracing intra-area and inter-area E2E paths
- Extend this ping/trace functionality to L2/MPLS hops along the way

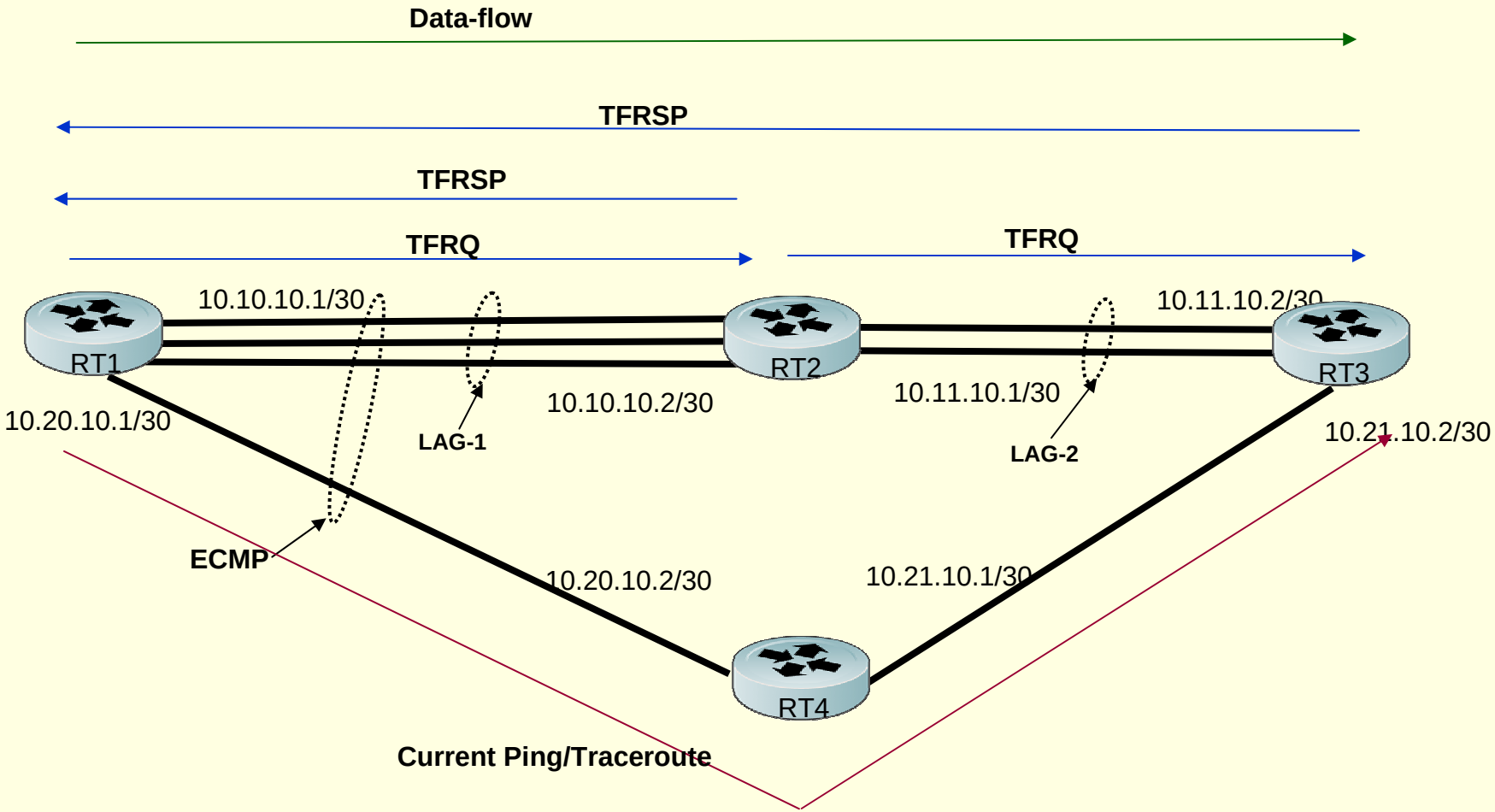
Use-case scenario…



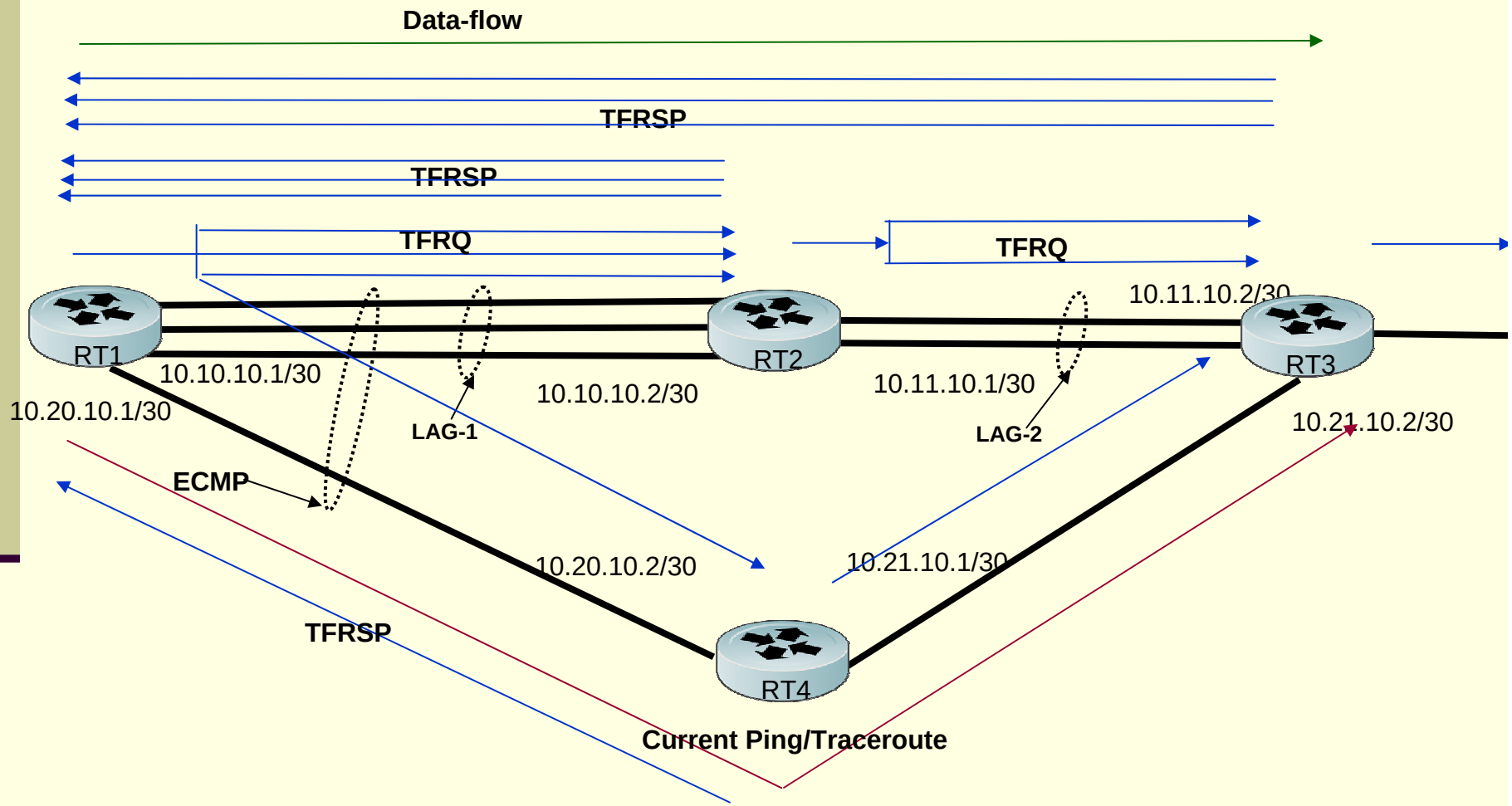
TraceFlow Protocol

- It is a protocol to address the requirements we have discussed so far
  - Design is extendable to future needs
- We have presented this in the OPSAWG Working Group at IETF
  - Internet Draft URL: http://tools.ietf.org/html/draft-zinjuvadia-traceflow-00 (work in progress)

Protocol Operation -- 1



Protocol Operation (Fan-out) – 2



Protocol Operation (Messages) -- 3

- Request Message – Composed of the following Type-Length-Value tuples
  - Flow Descriptor (first 256 bytes of flow packet)
  - Flow Information (MTU, fragmentation, etc.)
  - Originator address
  - Termination
  - Requested information
  - Authentication
- Response Message – Composed of the following Type-Length-Value tuples
  - Flow Descriptor (first 256 bytes of flow packet)
  - Encapsulated Packet Mask
  - Record-Route
  - Interface Info TLV(incoming/outgoing)
  - Result TLV

Legacy (non-compliant devices) support…

- Non-compliant nodes likely to forward TraceFlow messages along a different path compared to data packets for the same traffic flow
- Potential solution
  - Use Probe packets that look like data packets with TTL Scoping (like traceroute) to determine the path and the boundary of the legacy segment
  - Then use regular TraceFlow & hop over the legacy segment along the direction of the actual traffic flow
- Issues
  - Probe packets are indistinguishable from data
  - We run the risk of leaking spoofed probe packets into end-node receiver applications
  - Protocol complexity increases

Encapsulation…

- We are currently proposing UDP
  - Hop-by-Hop termination may be an issue
- Other options
  - New IP Protocol?
  - New ICMP message types?
    - Not sure if all hardware can look into ICMP subtypes for re-directing the packet to the CPU?

Open Issues…

- Encapsulation
- Legacy support – Do we need it?

Questions/Feedback?

Thank You