Operations, Maintenance and Administration (OAM) Tutorial

83rd IETF, Paris, France
March, 2012

Tissa Senevirathne and Sam K Aldrin

What is OAM

- Means different things to different people and organizations.
- Worst, some times it means different things to different people within the same organization
- IETF standardized the meaning of OAM within the IETF
  - June 2011, RFC 6291

IETF definition of OAM (RFC 6291)

- **O**perations: Operational activities to keep network up and running. *E.g. Monitoring, finding faults*
- **A**dministration: Involves keeping track of network resources. *E.g. Bookkeeping, (available ports, BW)*
- **M**aintenance: Involves repair and upgrades. *E.g. Software upgrades, configurations, corrective and preventive measures.*

Scope of the Tutorial

- Today's presentation mainly focus on IETF defined Operations aspects of OAM.
- Summary of applicable Administrative and Maintenance IETF standards are presented

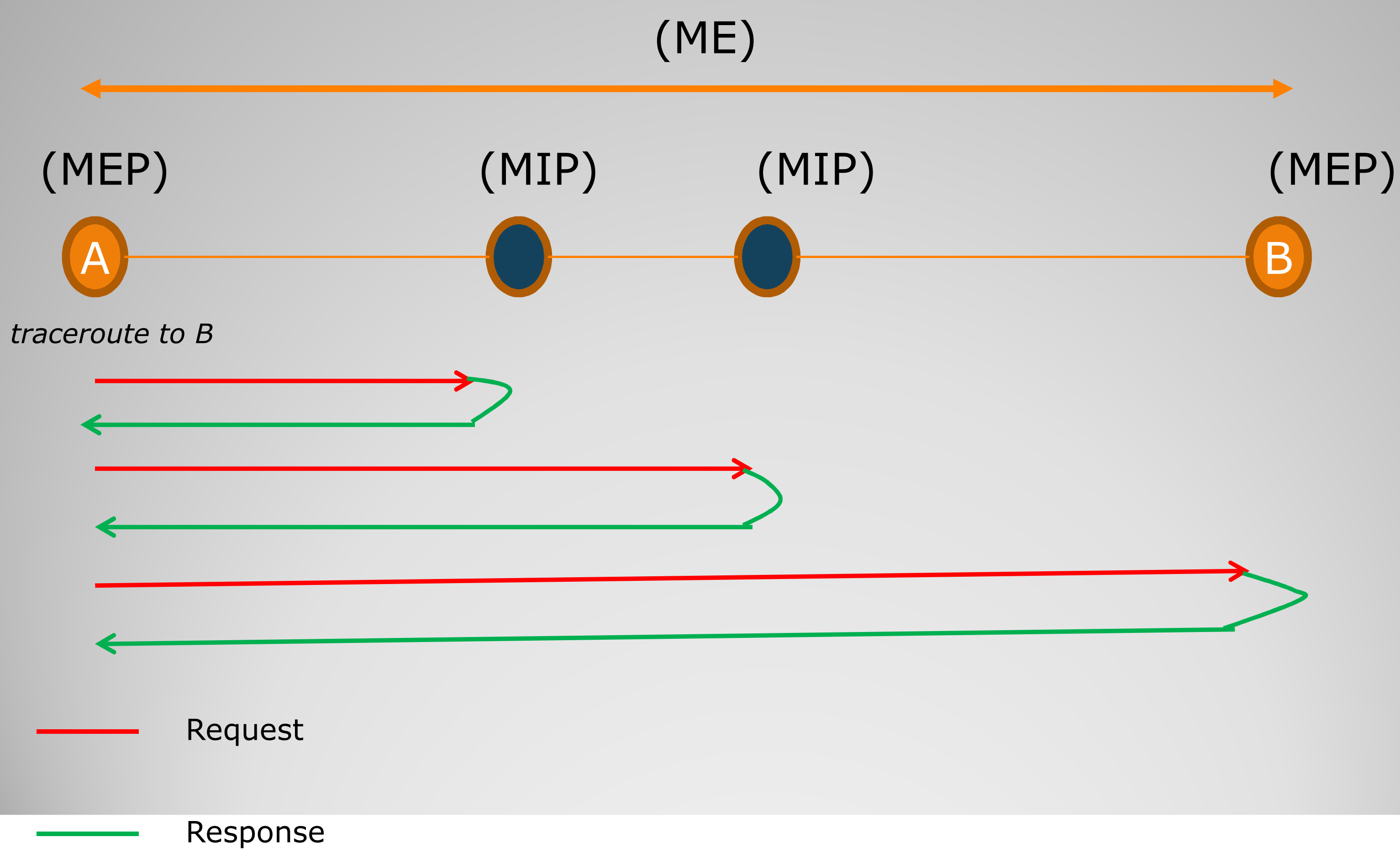
Important Terminologies

- Before we dive deeper, it is important to understand some of the terminologies and their meanings
- What are they ?
  - Various organizations (IEEE, ITUT, IETF) all have their version
  - We will discuss here selected set of definitions from RFC 5860, RFC 6371 and draft-ietf-opsawg-oam-overview-05
- Good understanding of these Terminologies will help us to appreciate modern OAM protocols better.

Important Terminologies

- Maintenance Point (MP)
  - Is a functional entity that is defined within a node that either initiate or react to a OAM message
- Maintenance Entity (ME)
  - Point to Point relationship between two MP
  - In MPLS this is LSP,
  - In BFD this is session
- Maintenance Point can be either MEP or MIP
  - Maintenance End Point (MEP)
    - Can either initiate or react to OAM Messages
    - MEP are the two end points of the ME
  - Maintenance Intermediate Point (MIP)
    - Is an intermediate MP between two MEP
    - It can only respond to OAM messages

Relationship of MP



Important Terminologies (contd..)

- Continuity Check
  - Ability of endpoint to monitor liveliness of a path (BFD)
- Connectivity Verification
  - Ability of an endpoint to verify it is connected to a specific endpoint. (BFD,Ping)
- Route Tracing
  - This is also known as path tracing, allows to identify the path taken from one MEP to another MEP (traceroute)
- Fault Verification
  - Exercised on demand to validate the reported fault. (Ping)
- Fault Isolation
  - Localizing and isolating the failure domain/point (traceroute)
- Performance
  - Includes Packet Loss Measurements and Packet Delay Measurements
  - E.g. IP Performance Metrics (IPPM) (RFC 2330)

Summary of OAM tools and Functions

	Continuity Check	Connectivity Verification	Path Discovery	Defect Indications	Performance Monitoring
ICMP		Echo (Ping)	Traceroute		
BFD	BFD control	BFD Echo			
LSP Ping		Ping	Traceroute		
IPPM					-Delay - Packet loss
MPLS-TP OAM	CC	CV	Traceroute	-Alarm Reporting - Client failure Ind - Remote Defect	

Ref: draft-ietf-opsawg-oam-overview-05

Ping

- Ping refers to tools that allows to detect liveliness of a remote host
- Most commonly known Ping is based on ICMP Echo Request and Response
- Security policies and firewalls sometimes prevent forwarding of ICMP messages.
  - This may reduce usefulness of ICMP Echo Request in some deployments.
- UDP/TCP version of the Ping has surfaced to circumvent barriers introduced by security policies and Firewalls on ICMP Echo Requests
  - "echoping" in Linux is one such example
  - RFC 4379 use UDP port 3503 for LSP Ping
  - NOTE: Linux default traceroute is based on udp.
- Different implementations of Ping has different options
  - Example: MS windows use –i for TTL and –n for number of packets
  - Linux use –t for TTL and –c for number of packets
  - Please read the manual pages of your implementation

Ping sample output from Linux

ping -c 2 -s 2000 -p ff00fffe 10.35.75.3

```
PATTERN: 0xff00fffe
PING 10.35.75.3 (10.35.75.3) 2000(2028) bytes of data.
2008 bytes from 10.35.75.3: icmp_seq=0 ttl=255 time=1.17 ms
2008 bytes from 10.35.75.3: icmp_seq=1 ttl=255 time=1.19 ms

--- 10.35.75.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.172/1.184/1.196/0.012 ms, pipe 2
```

- c = Count, -s = Size, -p = pattern

Min/avg/max/mdv = round trip delay (Minimu/average/maximum/meandeviati on)

There are whole lot of different options in ping please read manual pages.

Ping6

- Ping for IPv6 is based on ICMPv6 defined in RFC 4443
- IPv6 Next Header==56 indicate it is ICMPv6

```
ping6 ipv6.google.com

PING ipv6.google.com(2001:4860:b002::68) 56 data bytes
64 bytes from 2001:4860:b002::68: icmp_seq=0 ttl=59 time=58.4 ms
64 bytes from 2001:4860:b002::68: icmp_seq=1 ttl=59 time=56.4 ms
64 bytes from 2001:4860:b002::68: icmp_seq=2 ttl=59 time=62.1 ms
64 bytes from 2001:4860:b002::68: icmp_seq=3 ttl=59 time=56.8 ms
64 bytes from 2001:4860:b002::68: icmp_seq=4 ttl=59 time=56.5 ms
64 bytes from 2001:4860:b002::68: icmp_seq=5 ttl=59 time=59.5 ms
--- ipv6.google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5002ms
rtt min/avg/max/mdev = 56.443/58.329/62.150/2.045 ms, pipe 2
```

Ping – traceroute simulation

- Ping an IP address with increasing the TTL count at each step.
- In the example below TTL increased by 1 at each iteration..

```
ping -c 1 -t 2 -n www.yahoo.com
PING any-fp3-real.wa1.b.yahoo.com (98.139.127.62) 56(84) bytes of data.
From 10.35.78.17 icmp_seq=0 Time to live exceeded

--- any-fp3-real.wa1.b.yahoo.com ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
, pipe 2

ping -c 1 -t 3 -n www.yahoo.com
PING any-fp3-real.wa1.b.yahoo.com (98.139.127.62) 56(84) bytes of data.
From 10.34.159.13 icmp_seq=0 Time to live exceeded

--- any-fp3-real.wa1.b.yahoo.com ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
, pipe 2
```

10.35.78.17 is second hop router
10.34.159.13 is third hop router and

So on..

traceroute

- Design to trace the path taken from a node A to a node B.
- Probe packets are generated with monotonically increasing TTL value
  - Forcing ICMP TTL expiry message from each intermediate node.
  - In Linux Echo request packet is UDP (default destination port is UDP:33434)
  - In some other platforms it can be ICMP Echo request.

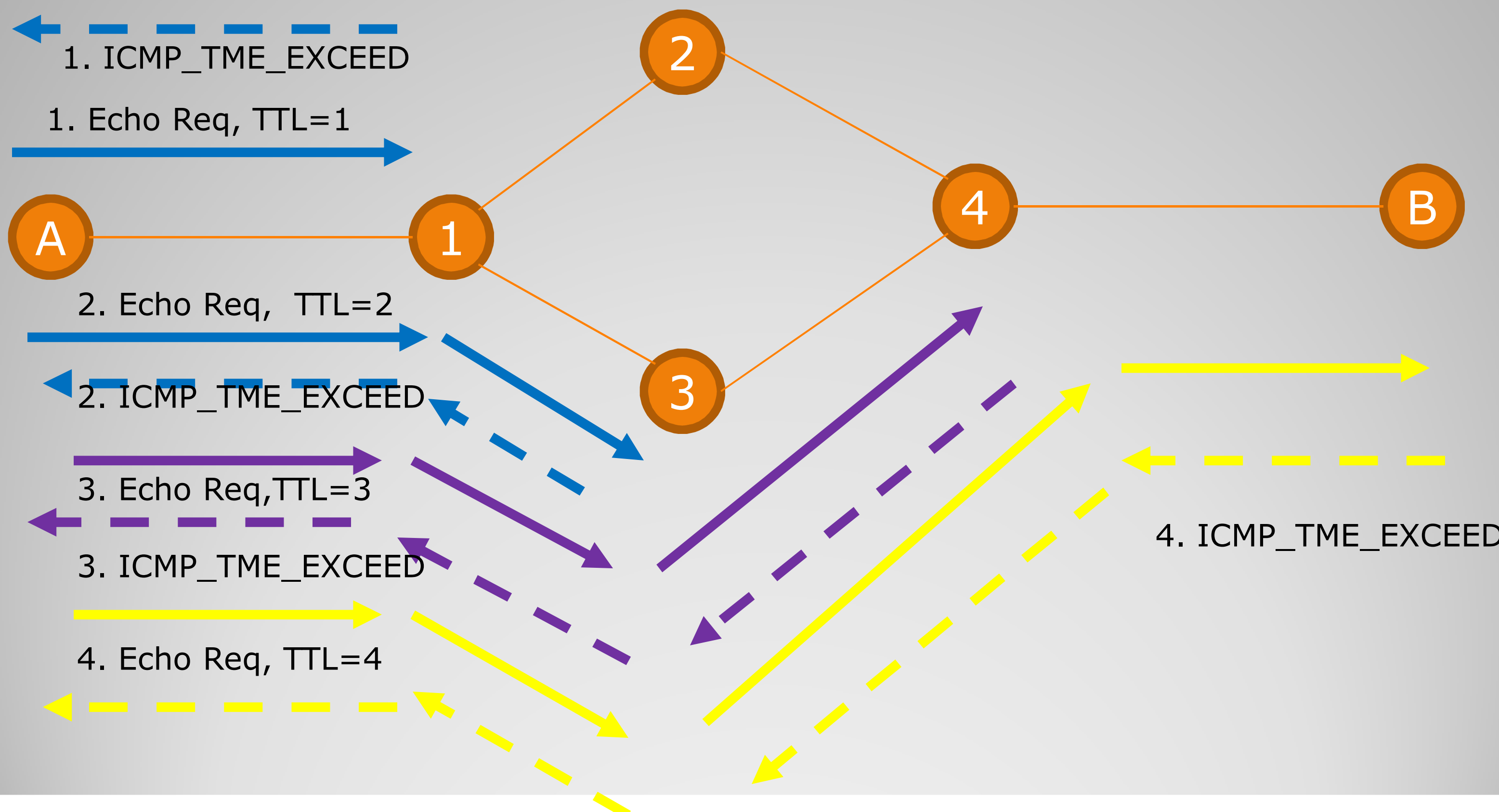
traceroute sample output linux

```
traceroute -n 10.35.78.17
traceroute to 10.35.78.17 (10.35.78.17), 30 hops max, 46 byte packets
1  10.35.75.3  0.292 ms  0.366 ms  0.213 ms     TTL=1
2  10.35.78.17  0.642 ms  0.429 ms  0.369 ms    TTL=2

traceroute -n -I 10.35.78.17
traceroute to 10.35.78.17 (10.35.78.17), 30 hops max, 46 byte packets
1  10.35.75.3  0.271 ms  0.219 ms  0.213 ms     TTL=1
2  10.35.78.17  0.442 ms  0.265 ms  0.351 ms    TTL=2
```

-I represent ICMP based traceroure
Default use UDP based ping
3 packets are sent at each TTL level
Round trip delay is printed out.

traceroute



1. ICMP_TME_EXCEED
1. Echo Req, TTL=1
2
4
B
A
1
2. Echo Req, TTL=2
3
2. ICMP_TME_EXCEED
3. Echo Req,TTL=3
3. ICMP_TME_EXCEED
4. Echo Req, TTL=4
4. ICMP_TME_EXCEED

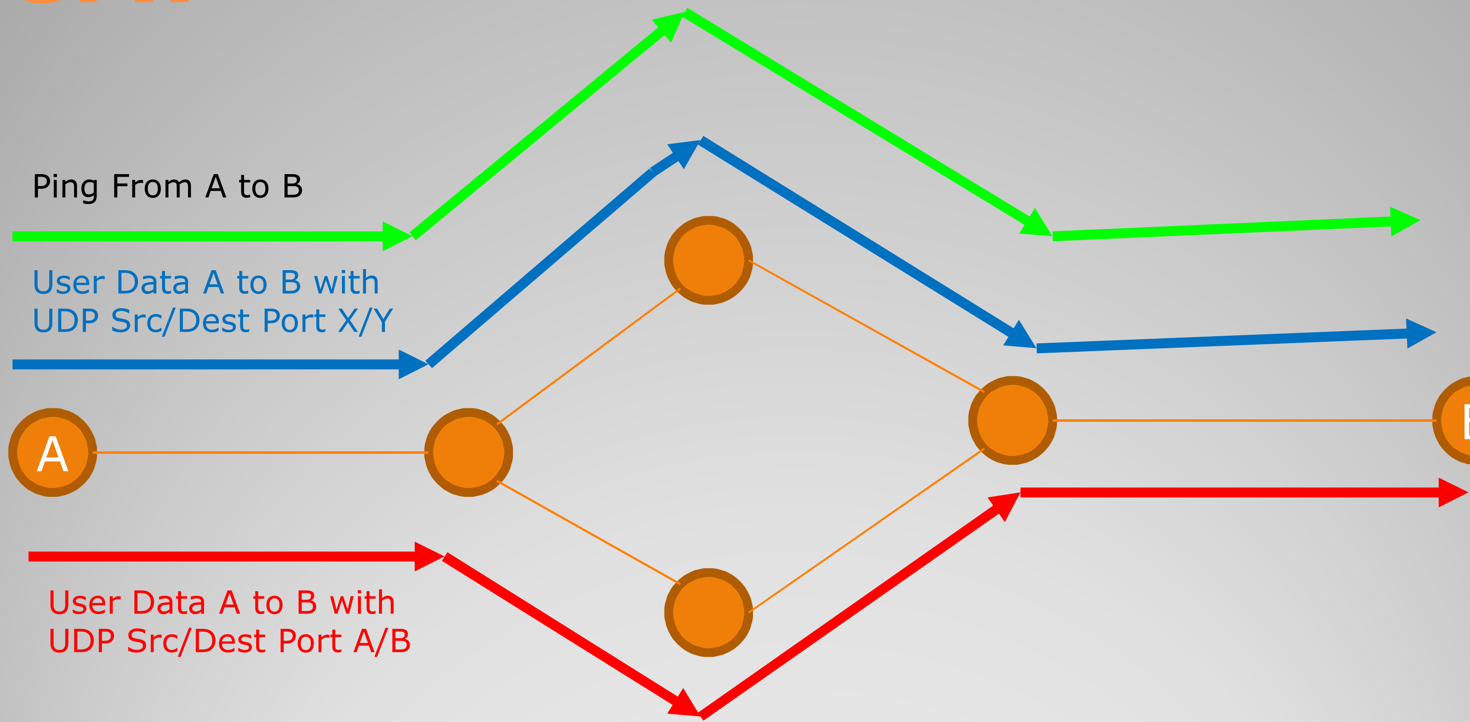
Challenges

- Over the years networking has evolved with that comes OAM challenges
  - ECMP (Equal Cost Multi Path)
  - Multicast
  - Tunneling (MPLS, PW, VPN, TRILL)
  - Firewalls
- ICMP and more traditional OAM are designed for unicast traffic with single path to the destination.

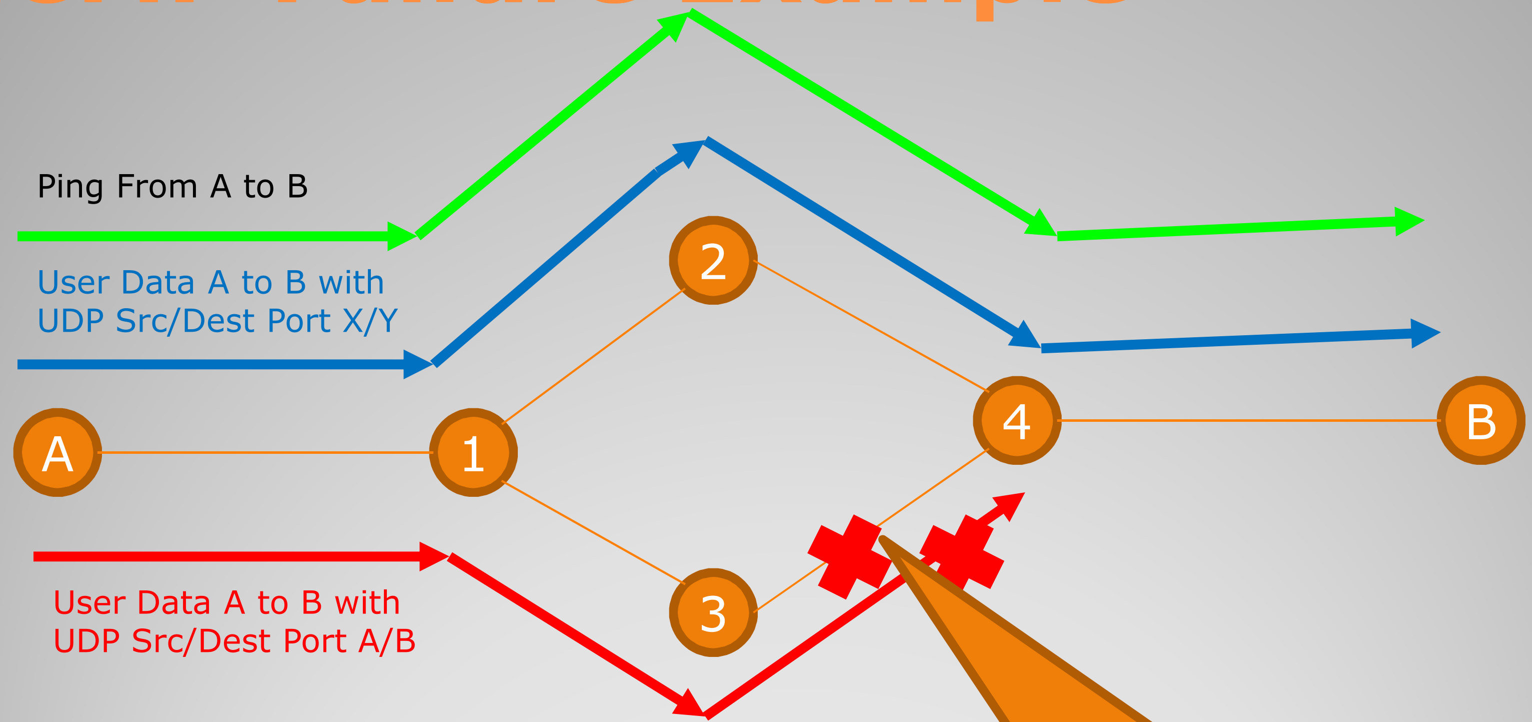
Equal Cost Multipath

- Equal Cost Multi Path (ECMP) allows
  - Protection against failures
  - Increased overall end-end BW
  - ECMP is becoming increasingly popular
- Devices typically use fields in the MAC or IP header to select the forwarding path among multiple equal cost paths
- Connectivity and Continuity verification messages MUST follow the same path as user data.
  - How can we accomplish this ?
  - There is no standard way of doing this in IP world
  - MPLS RFC 4379 has payload discovery approach

ECMP

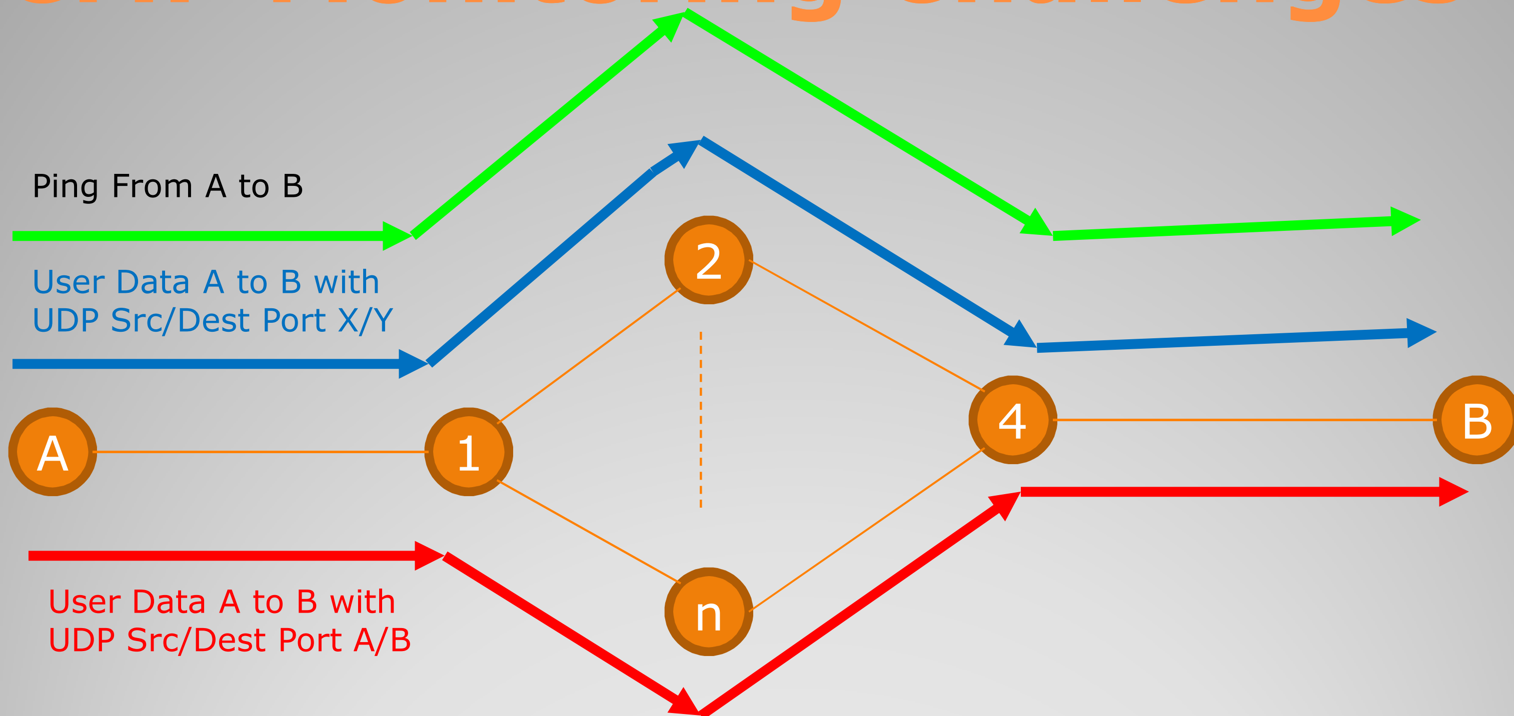

Ping From A to B
User Data A to B with
UDP Src/Dest Port X/Y
A
B
User Data A to B with
UDP Src/Dest Port A/B

ECMP Failure Example



- Can not utilize end-end connectivity tools to quickly detect the failure
- May need to wait until control protocol time-out
- If it is an oversubscribed link that causing intermittent traffic drops, protocols would not timeout

ECMP Monitoring Challenges



Challenges:

- Ingress Node (A) may not even know how many ECMP from intermediate node (1)
- Monitoring probes SHOULD take the same path as the normal data
- Different vendors utilize different hash algorithms in selection ECMP paths

ECMP challenges

- Conclusion
  - No standard method to exercise end-end continuity and connectivity verifications that covers all of the ECMP in IP networks

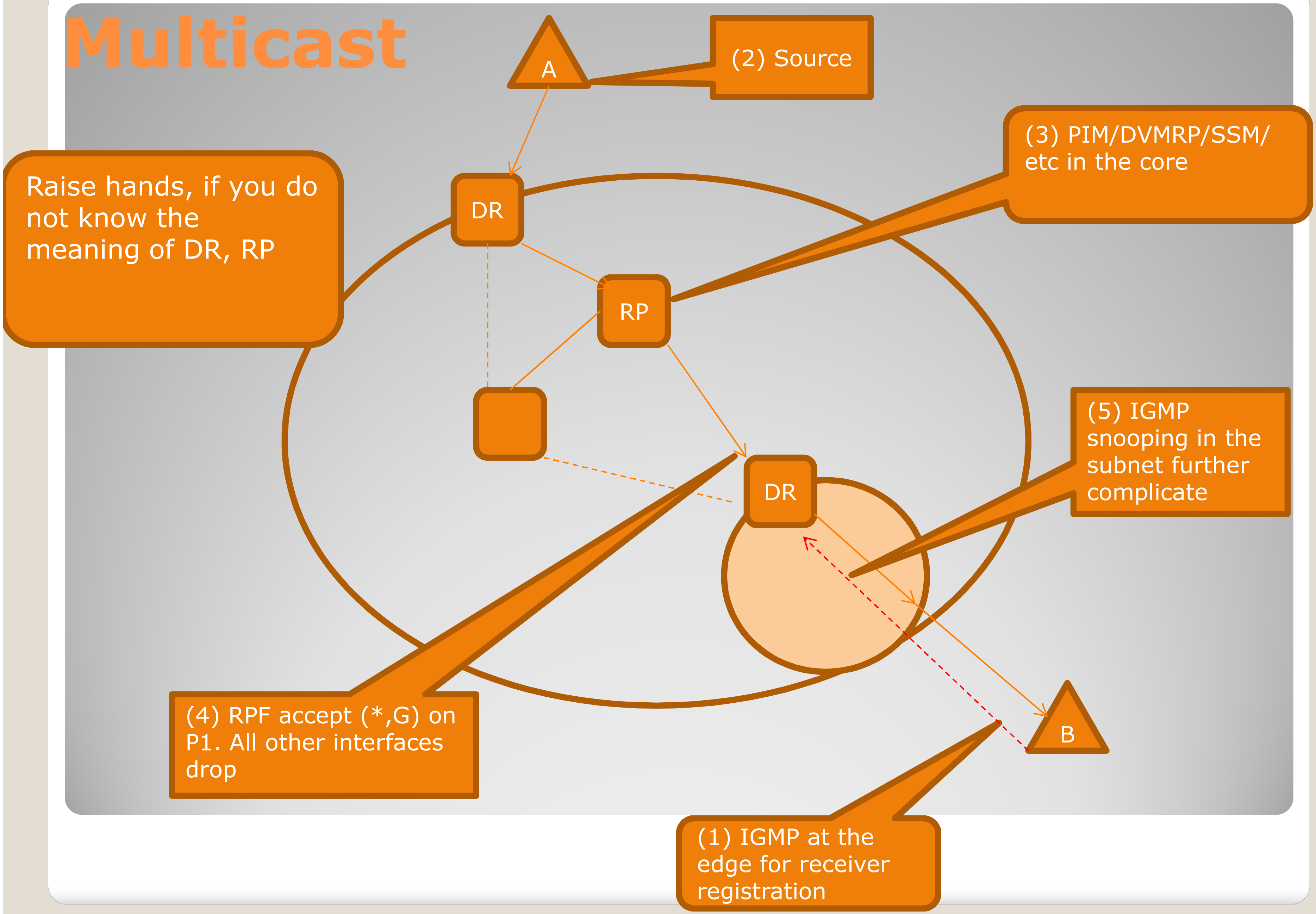
Multicast

- Why multicast is an OAM challenge ?
  - Most of us mostly deal with unicast problems
  - Multiple protocol interactions (IGMP at the edge, PIM at the core)
  - Multicast data flow is uni-directional
  - Multicast forwarding build a Shortest Path Tree for data forwarding
    - Users need to be familiar with concepts such as Reverse Path Forwarding (RPF)
      - RPF is needed to avoid transient loops
    - Users need to be familiar with other multicast architecture elements such as Designated Router (DR), Rendezvous Point (RP), Bootstrap Routers (BSR).
  - Multicast traffic continues to grow with the ever increasing demand for multi-media applications

Multicast

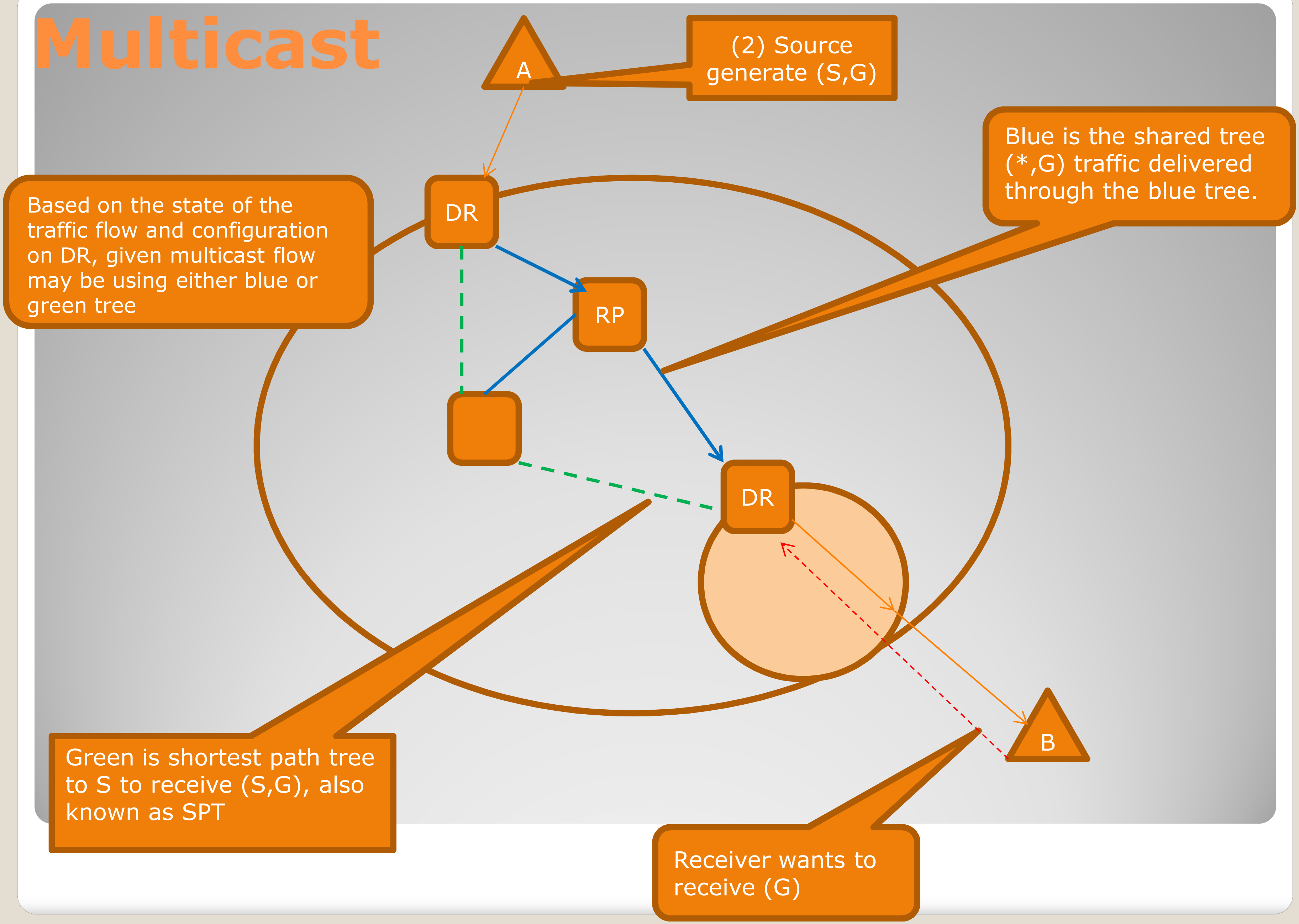
- Limited set of standard OAM protocol suite for multicast
  - Only old fashion troubleshooting available. i.e. moving from one device to other, looking for information
  - OR vendor specific tools and show commands

Multicast



A
(2) Source
(3) PIM/DVMRP/SSM/ etc in the core
Raise hands, if you do not know the meaning of DR, RP
DR
RP
DR
(5) IGMP snooping in the subnet further complicate
(4) RPF accept (*,G) on P1. All other interfaces drop
B
(1) IGMP at the edge for receiver registration

Multicast



A
(2) Source generate (S,G)
Blue is the shared tree (*,G) traffic delivered through the blue tree.
Based on the state of the traffic flow and configuration on DR, given multicast flow may be using either blue or green tree
DR
RP
DR
B
Green is shortest path tree to S to receive (S,G), also known as SPT
Receiver wants to receive (G)

Multicast troubleshooting

- Troubleshooting at the edge
  - How do I know problematic group G has joined as a receiver ?
    - SNMP or Vendor specific Show command on DR
  - Which Router is my Designated Router (DR) for the subnet ?
    - SNMP or vendor specific command like “Show ip pim interface x/y” tells who the current DR for the subnet.
  - Do I have IGMP snooping between my receivers and the DR
    - If so more troubleshooting needed

Multicast troubleshooting

- Troubleshooting at the core
  - Which Router is RP(s) for a group G?
    - SNMP or vendor specific command such as "Show ip pim rp" on all the router
  - Does RP knows about the source after transmitting ?
    - SNMP or vendor specific commands such as "Show ip pim route" or "show ip mroute (mcast rib)" should show S,G has been created through register
  - If the route is (*,G), troubleshoot whether traffic is coming on the shared tree (RPT)
    - SNMP orvendor specific command such as "Show ip mroute details" show the stat for G
  - If the route is source route (S,G), debug whether traffic is coming on the SPT
    - Same as above
  - Do I have RPF issues ?
    - Use mtrace command (more details later)

Multicast troubleshooting

- Data plane verification
  - Ping multicast-group address G from the source.
  - Every device in the network that is receiving G reply to the Ping. (Overwhelming number of replies)
  - Intermediate Routers such as DR and RP do not respond to G (They do not have an interface active for G)

Multicast troubleshooting

- Data plane verification **Ping**

ping 225.1.1.1
PING 225.1.1.1 (225.1.1.1) 56(84) bytes of data.
64 bytes from 10.35.75.11: icmp_seq=0 ttl=64 time=0.027 ms
64 bytes from 10.35.75.3: icmp_seq=0 ttl=255 time=0.593 ms (DUP!)
64 bytes from 10.35.75.2: icmp_seq=0 ttl=255 time=0.600 ms (DUP!)
64 bytes from 10.35.75.2: icmp_seq=0 ttl=255 time=0.646 ms (DUP!)
64 bytes from 10.35.75.11: icmp_seq=1 ttl=64 time=0.015 ms

Too many responses

ping 225.1.1.1 | grep 10.35.75.3
64 bytes from 10.35.75.3: icmp_seq=0 ttl=255 time=0.506 ms (DUP!)
64 bytes from 10.35.75.3: icmp_seq=1 ttl=255 time=0.472 ms (DUP!)
64 bytes from 10.35.75.3: icmp_seq=2 ttl=255 time=0.458 ms (DUP!)
64 bytes from 10.35.75.3: icmp_seq=3 ttl=255 time=0.449 ms (DUP!)
64 bytes from 10.35.75.3: icmp_seq=4 ttl=255 time=0.488 ms (DUP!)

Use ***grep*** "ip address" to filter out specific address

- Need to issue the Ping from the source it self to validate the multicast path
- Ping "G" from other host may indicate incorrect results.

Multicast troubleshooting

- *mtrace*
- Provide reverse path forwarding (RPF) validation.
- *mtrace* is a control plane tool. Hence can miss certain faults related to miss-alignment of control and data plane.

Multicast troubleshooting

- mtrace [source] [destination] [group]

```
Router# mtrace 171.69.215.41 171.69.215.67 239.254.254.254
Type escape sequence to abort.
Mtrace from 171.69.215.41 to 171.69.215.67 via group 239.254.254.254
 From source (?) to destination (?) Querying full reverse path...

 0 171.69.215.67
-1 171.69.215.67 PIM thresh^ 0 0 ms
-2 171.69.215.74 PIM thresh^ 0 2 ms
-3 171.69.215.57 PIM thresh^ 0 894 ms
-4 171.69.215.41 PIM thresh^ 0 893 ms
-5 171.69.215.12 PIM thresh^ 0 894 ms
-6 171.69.215.98 PIM thresh^ 0 893 ms
```

Multicast troubleshooting

- mtrace [source] [destination] [group]

```
dc3rtg-d4# mtrace 1.3.0.3 225.1.1.1
Mtrace from 1.3.0.3 to 1.4.0.4 via group 225.1.1.1
Querying full reverse path...
  0  ? (1.4.0.4)
 -1  ? (1.4.0.4) PIM RPF Interface
 -2  ? (1.4.0.2) PIM
 -3  ? (1.2.0.5) PIM
 -4  ? (1.1.0.1) PIM
Round trip time 26 ms; total ttl of 4 required.
```

Mtrace output from a different implementation

Multicast: (*,G) vs. (S,G)

```
dc3rtg-d4# show ip mroute detail IP Multicast Routing Table for VRF "default"

Total number of routes: 3
Total number of (*,G) routes: 1
Total number of (S,G) routes: 1
Total number of (*,G-prefix) routes: 1

(*, 225.1.1.1/32), uptime: 02:28:56, igmp ip pim
  Stats: 1/100 [Packets/Bytes], 13.333  bps
  Attached oif(s) : Yes
  Incoming interface: Ethernet2/4, RPF nbr: 1.4.0.2
  Outgoing interface list: (count: 2)
    loopback2, uptime: 00:08:57, igmp
    loopback1, uptime: 02:28:56, igmp

(1.3.0.3/32, 225.1.1.1/32), uptime: 00:00:52, ip mrib pim
  Stats: 22/2200 [Packets/Bytes], 338.462 bps
  Attached oif(s) : Yes
  Incoming interface: Ethernet2/5, RPF nbr: 1.5.0.2
  Outgoing interface list: (count: 2)
    loopback2, uptime: 00:00:52, mrib
    loopback1, uptime: 00:00:52, mrib

(*, 232.0.0.0/8), uptime: 02:33:27, pim ip
  Stats: 0/0 [Packets/Bytes], 0.000   bps
  Attached oif(s) : No
  Incoming interface: Null, RPF nbr: 0.0.0.0
  Outgoing interface list: (count: 0)
```

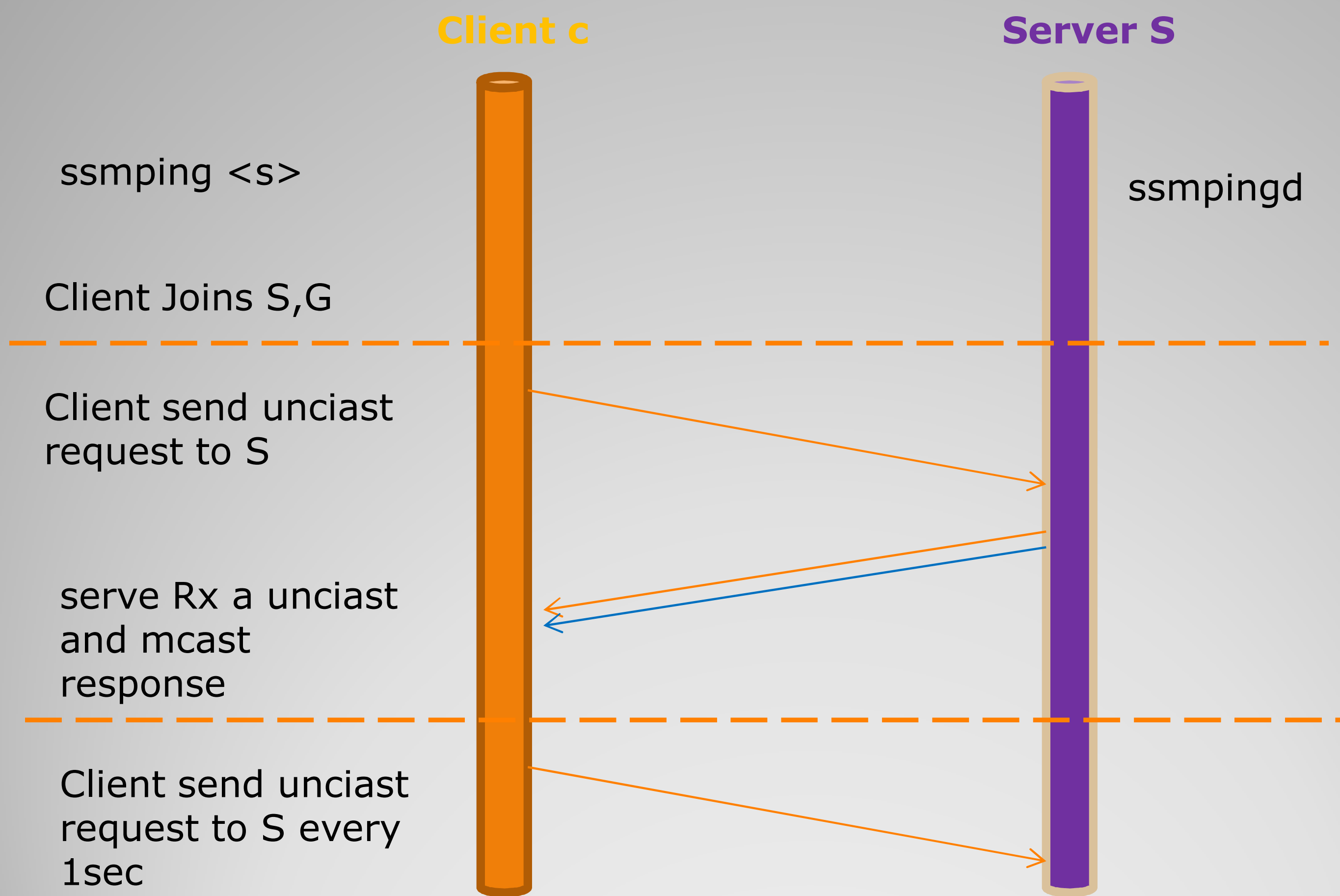
(*,G) route. Notice, that there is only one packet. ***Why ?***

(S,G) route. Notice, that RPF interface is different than (*,G). ***Why ?***

ssmping and asmping

- Standardized as RFC 6450
- Ssmpingd server is required run on the multicast source.
- Receivers ssmping the server via a unicast message
- Server sends one unicast packet to the receiver and one multicast packet to the group G. (with a specific UDP port).
  - Receiver is expected to receive both unicast and multicast packets.
- Challenge is operators are required to have admin access to the multicast server
  - Operational challenge in hosting services

Ssmping – packet flow



ssmping

```
#ssping 192.168.0.12
ssmping joined (S,G) = (192.168.0.12,232.43.211.234)
pinging S from 192.168.0.11
  unicast from 192.168.0.12, seq=1 dist=-1 time=9.001 ms
multicast from 192.168.0.12, seq=1 dist=-1 time=10.001 ms
  unicast from 192.168.0.12, seq=2 dist=-1 time=4.000 ms
multicast from 192.168.0.12, seq=2 dist=-1 time=5.000 ms
  unicast from 192.168.0.12, seq=3 dist=-1 time=11.001 ms
multicast from 192.168.0.12, seq=3 dist=-1 time=13.001 ms
  unicast from 192.168.0.12, seq=4 dist=-1 time=19.002 ms
multicast from 192.168.0.12, seq=4 dist=-1 time=21.002 ms
  unicast from 192.168.0.12, seq=5 dist=-1 time=12.001 ms
multicast from 192.168.0.12, seq=5 dist=-1 time=14.001 ms
  unicast from 192.168.0.12, seq=6 dist=-1 time=6.000 ms
multicast from 192.168.0.12, seq=6 dist=-1 time=7.001 ms

--- 192.168.0.12 statistics ---
6 packets transmitted, time 6000 ms
unicast:
   6 packets received, 0% packet loss
   rtt min/avg/max/std-dev = 4.000/10.167/19.002/4.812 ms
multicast:
   6 packets received, 0% packet loss since first mc packet (seq 1) recvd
   rtt min/avg/max/std-dev = 5.000/11.667/21.002/5.219 ms
```

mcfirst

- Executed on a specific Receiver
- Joins specific group G
- Listens for multicast packets received on G
- By default Exit at receiving the first packet or one can specify the packet count
- Format of mcfirst

mcfirst [-46vr] [-I interface] [-c count] [-t time] [source] group port

Multicast troubleshooting

- Conclusion
  - Troubleshooting and monitoring can be very complicated.
  - This is an interesting area of work for anyone who is interested in.
  - Related drafts/RFC:
    - draft-tissa-mcastoam-00 – light weight multicast Ping with extensibility for role discovery. Can be executed from routers, do not have to log-in to servers.
    - RFC 6450 - Multicast Ping Protocol

Summary of IETF Management standards

Covers (A) and (M) of OAM

Fault Management

SNMP – notfications
IPFIX
PSAMP
SYSLOG

Configuration Management

SNMP – set
CAPWAP
NETCONF
AUTOCONF
ACAP
DHCP

Accounting Management

SNMP –get
IPFIX
PSAMP
RADIUS accounting
DIAMETER accounting

Ref: draft-ietf-opsawg-management-stds-05

Summary of IETF Management standards

Performance Management

SNMP – get IPFIX

PSAMP

Security Management

RADIUS – Authentication and Authorization
DIAMETER – Authentication and Autherization

OAM Tutorial – BFD Overview

Bidirectional Forward Detection (BFD)

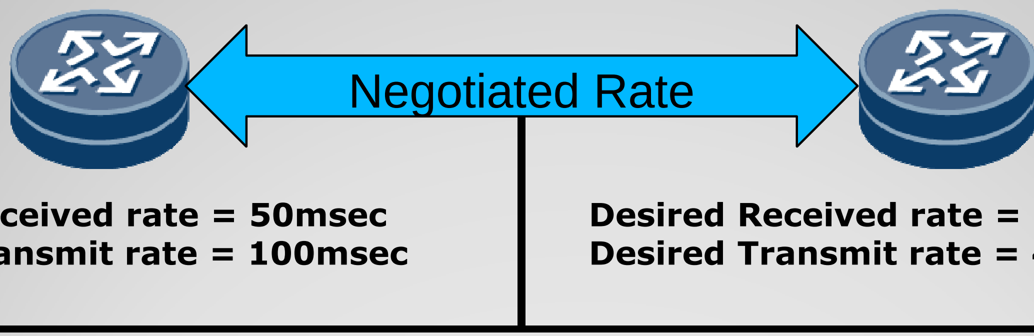
- Simple fixed-field, hello protocol
- Packets are periodically transmitted over respective directions of the path
- If a node stops receiving BFD packets, some component of the bidirectional path is assumed to have failed.
- Several modes of operation

BFD protocol Overview

- Typical hello protocol
- Neighbors continuously negotiate transmit and receive rates in micro seconds
- Dynamic rate adaption
- Neighbor is declared down when hello packets don't show up
- Uses UDP/IP or Non IP packets as BFD packets
- Ability to support single-hop and multi-hop

BFD Timer negotiation

- Neighbors continuously negotiate transmit and receive rates
- Designated UDP ports 3784 and 3785 are assigned to BFD
- Ability to support single-hop and multi-hop



Desired Received rate = 50msec
Desired Transmit rate = 100msec

Desired Received rate = 60msec
Desired Transmit rate = 40msec

Negotiated Transmit rate = 100msec

Negotiated Transmit rate = 50msec

OAM Tutorial – MPLS OAM

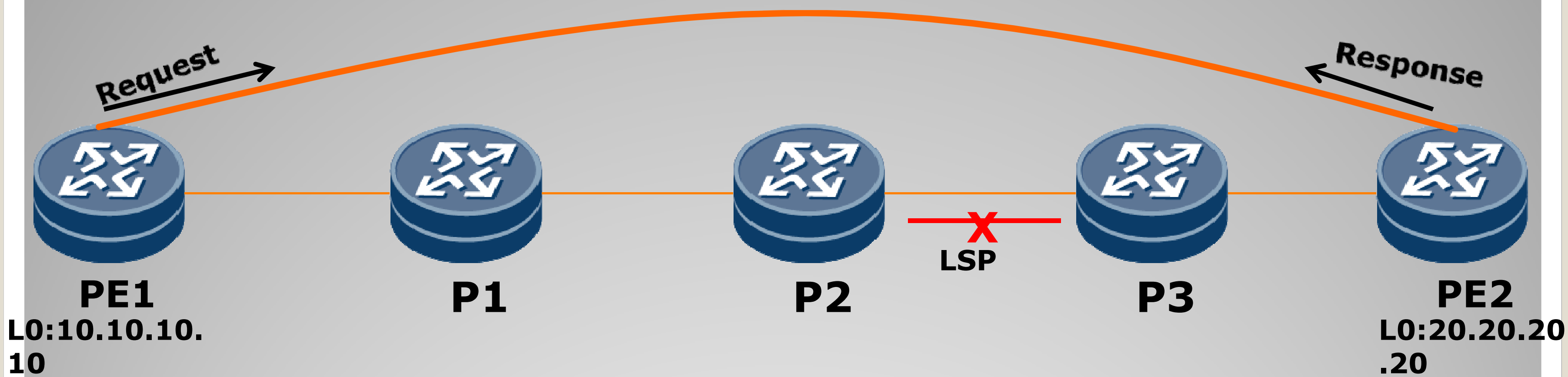
Problems in MPLS Networks

- Control Plane is working, Data Plane is broken
- IGP working but MPLS control protocol is broken
- Proactive monitoring of End-to-End MPLS LSP's
- Identifying the End-to-End packet path
- Unlabelled interface
- MTU issues
- Performance degradation and unable to provide QoS

Primitive Debugging Methods

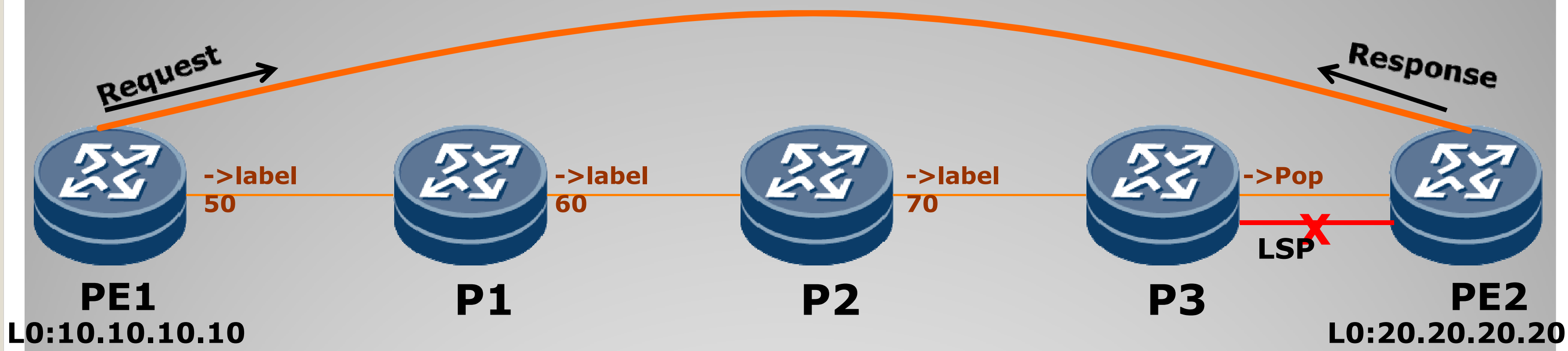
- ICMP provides connectivity verification
- VRF aware ping could test VPN path connectivity
- UDP ping could test the UDP transport
- Route table and Label table provides label entries programmed
- Interface status verification
- MPLS control plane protocols provides control plane information

ICMP ping



- ICMP ping emulates the data but can only verify IP layer
- It cannot verify if MPLS path is broken but IP is working
- It cannot verify ECMP
- It cannot validate control plane to data plane
- It cannot verify various MPLS control plane protocols
- It cannot verify for unlabelled interface, black-holes, control plane to data plane mismatch, etc.

VRF aware ping



- VRF aware could emulate VPN traffic
- Could test VPN connectivity
- Cannot detect LSP breakage
- If IP connectivity is working and MPLS is broken, it cannot detect
- Can detect if there is no label path, but not in all cases
- Cannot detect ECMP failures, CP to DP mismatch, etc.

LSP ping

Requirements

- Detect LSP failures
- Detect label mismatch
- Detect CP to DP mismatch
- Pin point the failure
- Detect MTU failures

Applications

- Verify all MPLS FEC types
- Verify PE, P, MPLS TP devices
- Ability to verify MPLS VPN, TE, LDP, TP, P2MP, etc., LSP's.

Solution

- LSP ping to detect connectivity checks
- LSP ping based traceroute for path verification
- LSP ping based topology tree verification

Standards

- RFC4379 and all other extensions

LSP Ping – What is it?

Function

- LSP ping is modeled like ICMP ping but based on UDP
- It checks the connectivity between two end points of an LSP

Format

- Emulated packet with label encapsulation of a data frame for the FEC
- The IP destination of the packet is local host address

Behavior

- Upon breakage of MPLS LSP, the packet is to be locally processed
- The response packet contains a code indicating the failure/error/reason along with other data
- The destination IP address could be manipulated to simulate ECMP scenario in order to verify LB paths
- OAM packets are treated the same as data packets on transit routers
- TTL field is used to test intermediate hops

LSP Ping – What can it verify?

Sub-Type	Length	Value field
1	5	LDP IPv4 Prefix
2	17	LDP IPv6 Prefix
3	20	RSVP IPV4 Prefix
4	56	RSVP IPv6 Prefix
5		Not Assigned
6	13	VPN IPv4 Prefix
7	25	VPN IPv6 Prefix
8	14	L2 VPN endpoint
9	10	FEC 128 PW (Deprecated)
10	14	FEC 128 PW
11	16+	FEC 129 PW
12	5	BGP Labeled IPv4 Prefix
13	17	BGP Labeled IPv6 Prefix
14	5	Generic IPv4 Prefix
15	1	Generic IPv6 Prefix
16	4	Nil FEC

LSP Ping – Constructs

LSP ping packet is encapsulated to simulate data packet in order to test a LSP

- Two types – Echo Request and Echo Response
- The FEC to be verified
- The Label stack for the FEC/LSP
- A UDP/IP packet with LSP ping payload to be send on the LSP
- The interface information on which the packet has to be forwarded
- Forwarding and interface information for the FEC for verification purposes

LSP Ping – Response Codes

Value	Meaning
-------	-----------
0	No return code
1	Malformed echo request received
2	One or more TLV's not understood
3	Replying router is egress for the FEC
4	No mapping for the FEC
5	DSMAP mismatch
6	Unknown upstream index
7	Reserved
8	Label switched at stack depth <RSC>
9	Label switched but no MPLS forwarding at stack depth <RSC>
10	Mapping for this FEC is not the given label at stack depth <RSC>
11	No label entry at stack depth <RSC>
12	Protocol not associated with interface at FEC stack depth <RSC>
13	Premature termination of ping due to label stack shrinking to a single label

LSP Ping – Echo Request

Echo Request is sent by the router to test LSP of a given FEC

MPLS encapsulation

- MPLS encapsulated IP/UDP packet
- Label stack is same as data packet for the FEC.
- TTL value for the label is 255 (set to right value to test a particular hop).
- FEC TLV contains the details of the FEC to be verified

IP Encapsulation

- IP/UDP Packet
- Source address: Valid source address to which response has to be sent
- Destination address: Local host address
- Destination Port: 3503
- RA option : Enable
- TTL : 1

LSP Ping – Echo Reply

Echo Reply is sent by the router to responding to the Echo Request

Reply Modes

- IP reply
- No Reply
- IP reply with RA option
- Control Channel

Packet Format

- IP source address : Replying router IP address
- Destination address : Source address from which echo request was received
- Source port : 3503/other chosen port
- Destination Port : Port number in the echo request
- TTL : 255

Downstream Mapping

0	7	15	23	31
Downstream Router ID				
MTU		Addr Type	DS Index	
Downstream Interface Address				
Hash Key	Depth Limit	MultiPath Length		
IP Address or Next Label				
More IP addresses or next labels				
Downstream Label			Protocol	
....				
Downstream Label			Protocol	

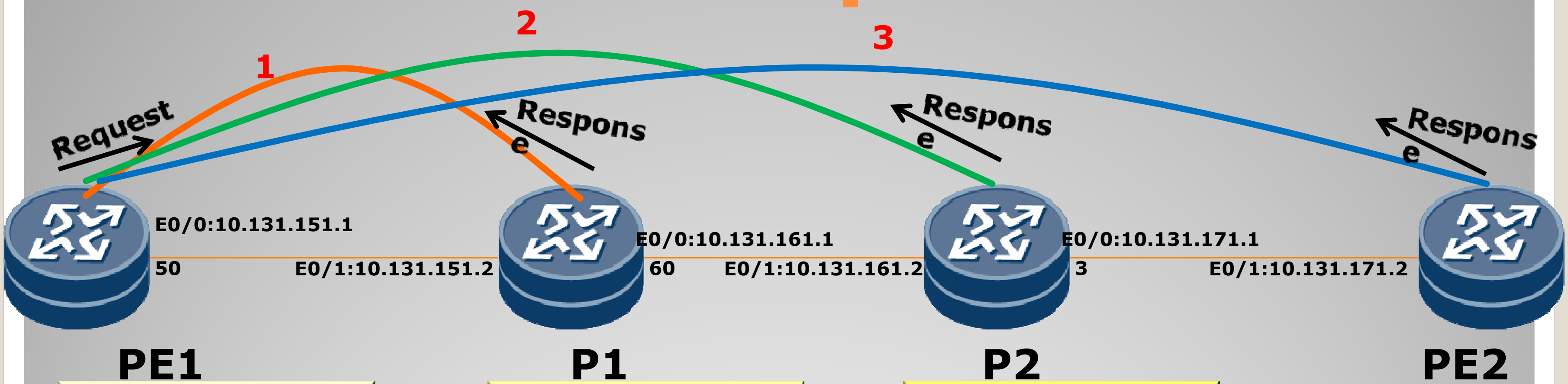
DSMAP TLV

- •Downstream interface address is IP address of outgoing interface for the LSP
- •Downstream label is the outgoing label for the LSP
- •Protocol associated with the label
- •DDMAP is enhanced version of the DSMAP TLV (Deprecated)

0	7	15	23	31
MTU		Addr Type	DS Flags	
Downstream Address (4 or 16 octets)				
Downstream Interface Address (4 or 16 octets)				
R Code	Return SC	SubTLV Length		
IP Address or Next Label				
List of SubTLV's 1. Multipath 2. Label Stack 3. FEC Stack change				

DDMAP TLV

Downstream Mapping TLV - Example



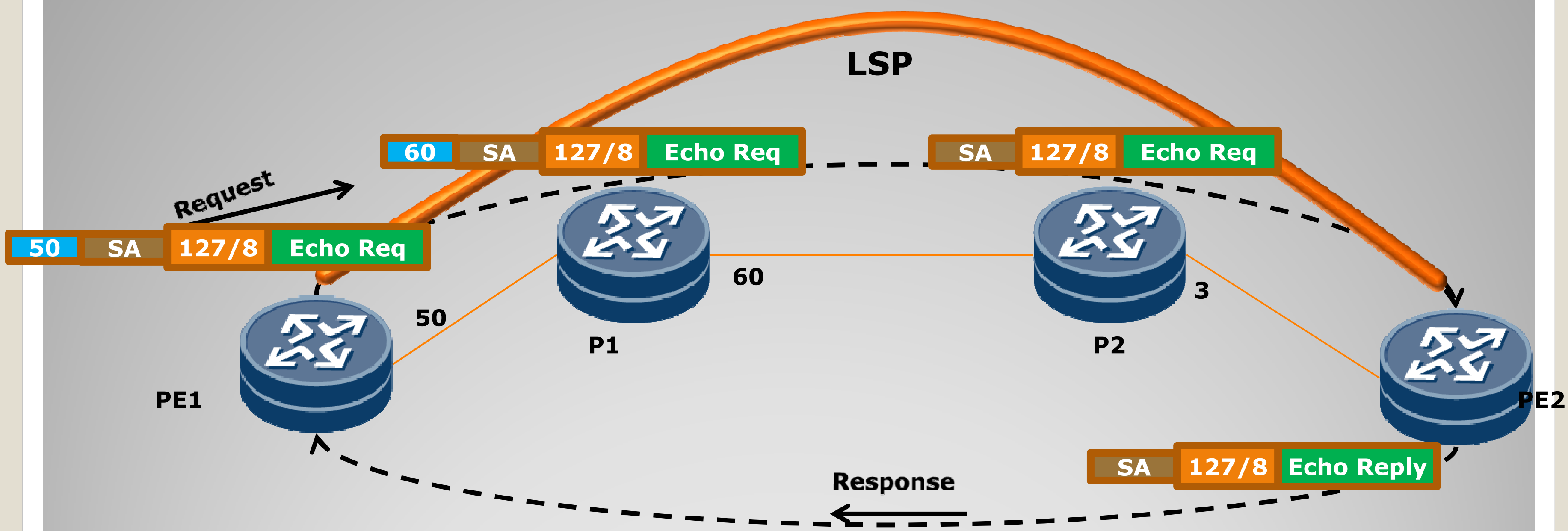
PE1 Downstream for PE2
MTU: MRU of E0/0
Address type: 1
DS Intf Addr: 10.131.151.1
DS Label : 50

P1 Downstream for PE2
MTU: MRU of E0/0
Address type: 1
DS Intf Addr: 10.131.161.1
DS Label : 60

P2 Downstream for PE2
MTU: MRU of E0/0
Address type: 1
DS Intf Addr: 10.131.171.1
DS Label : 3

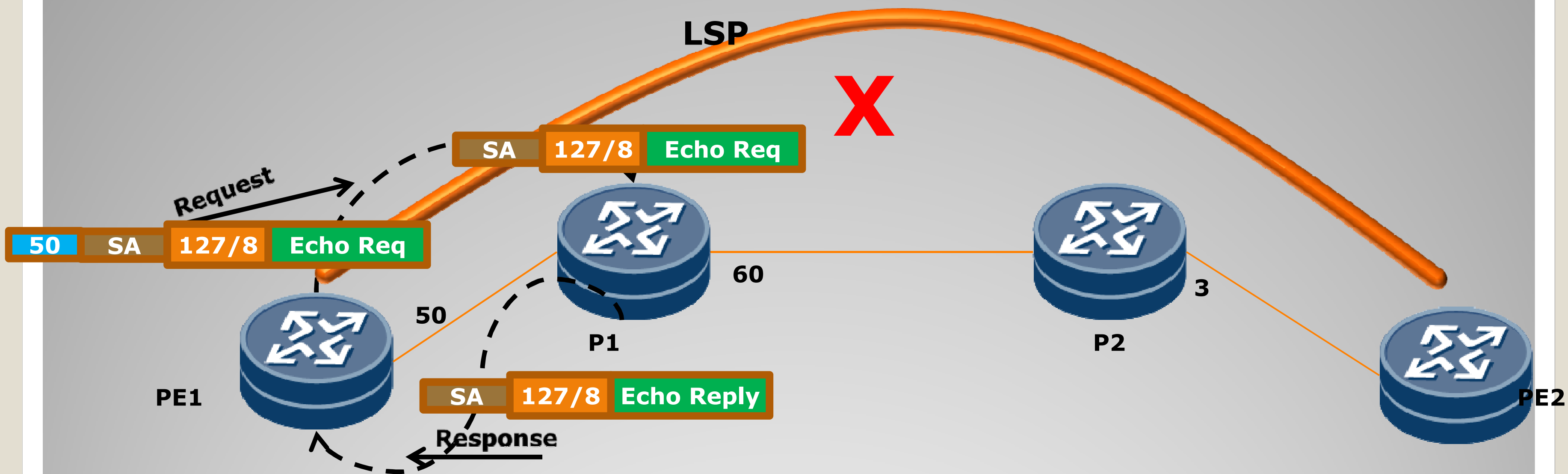
Note: No DSMAP TLV is sent by Egress router

Theory of Operation



- Packet is encodes with the same label stack as data packet
- The destination header of the packet is set as local host address
- The packet is forwarded on Egress interface identified for the FEC
- The packet get labeled switched on transit routers
- No special treatment of OAM packets on transit routers
- The Echo reply is sent as IP as default

LSP ping as diagnostic tool



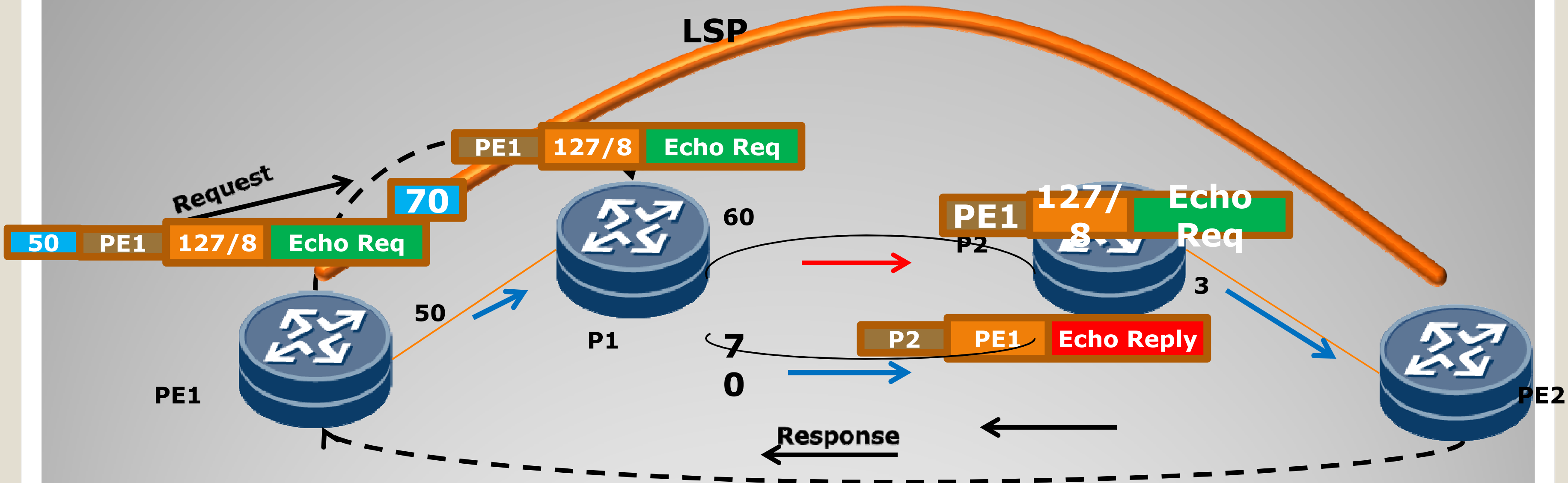
LSP could be broken due to various reasons

- No MPLS interface
- No LDP adjacency
- Label mismatch
- Control Plane and Data Plane mismatch

LSP ping Echo Request cannot get label forwarded due to LSP breakage

- Echo request gets locally processed due to local address
- Reply sent by the processing router with appropriate error code

LSP ping for Control Plane Data Plane Mismatch



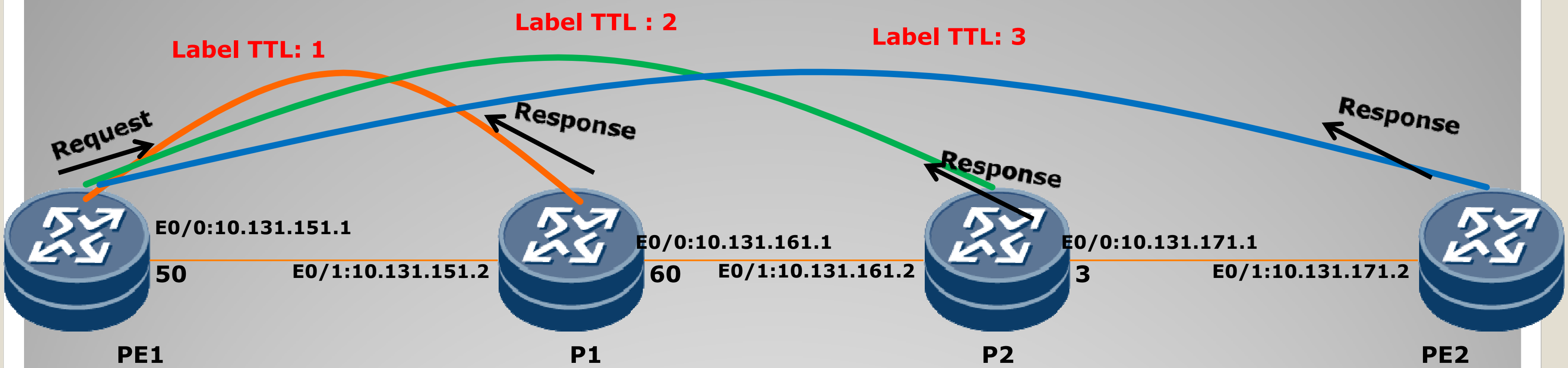
LSP control plane and data plane mismatch

- Control plane advertises label 60 to PE2 FEC
- Data Plane takes different path with label 70
- Though packets reach PE2, they traverse different path

LSP ping with DSMAP or Trace validation

- When LSP ping with DSMAP is set hop by hop, it can identify the fault
- DSMAP mismatch error will be return upon this error

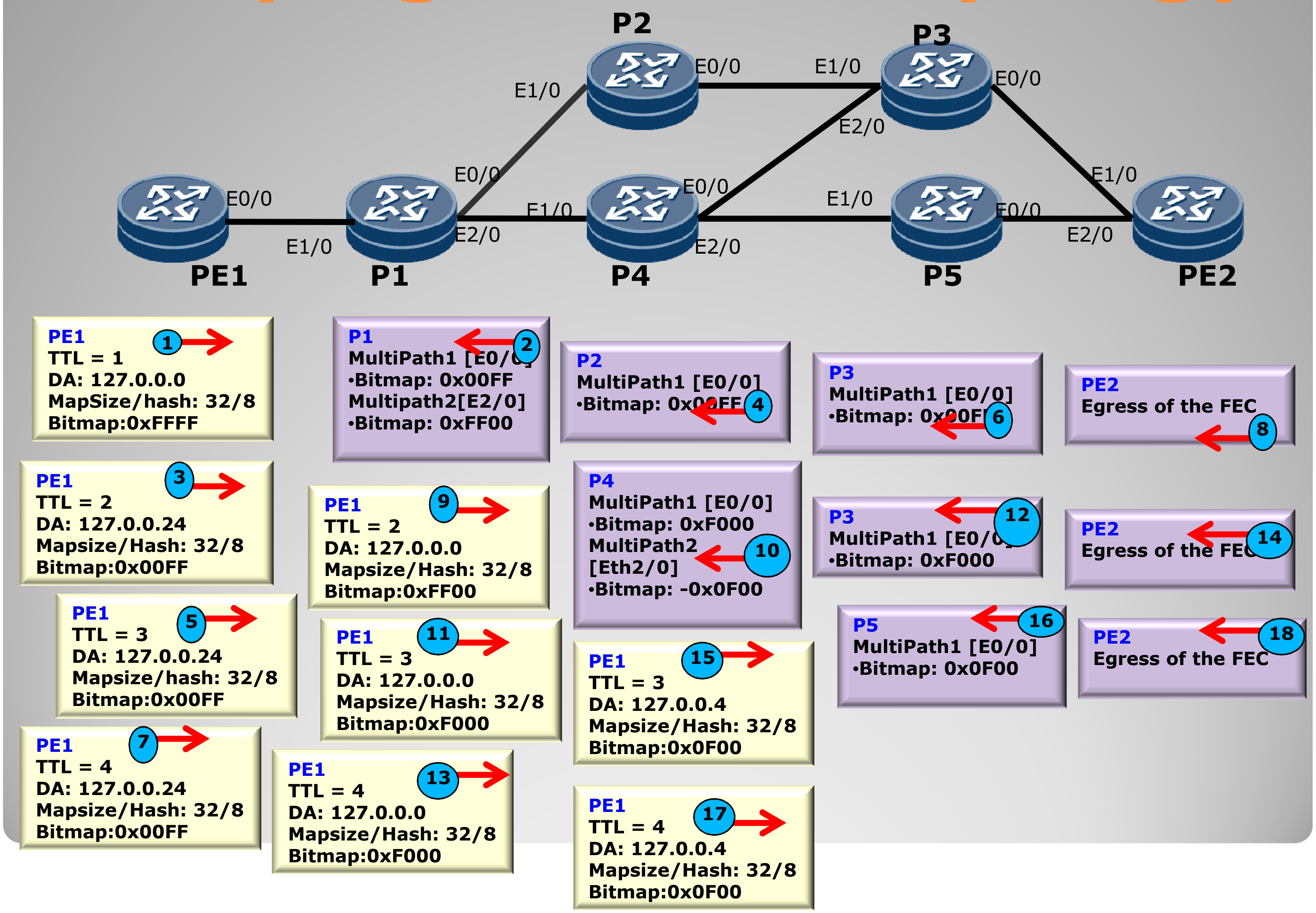
Trace with LSP Ping



PE1 Downstream for PE2	P1 Downstream for PE2	P2 Downstream for PE2
MTU: MRU of E0/0	MTU: MRU of E0/0	MTU: MRU of E0/0
Address type: 1	Address type: 1	Address type: 1
DS Intf Addr: 10.131.151.1	DS Intf Addr: 10.131.161.1	DS Intf Addr: 10.131.171.1
DS Label : 50	DS Label : 60	DS Label : 3

- LSP Ping with TTL is used to validate every hop of the LSP
- Downstream TLV is used to validate and request downstream info
- If the responding router is Egress of the FEC, a return code of 3 is returned.
- No DSMAP TLV is sent in the response by Egress router for the FEC

LSP ping in ECMP topology



FEC types support

LSP ping supports various FEC types

FEC Type	LSP Ping	LSP Trace	ECMP Trace
LDP IPv4 and IPv6	Yes	Yes	Yes
RSVP TE v4 and v6	Yes	Yes	N/A
PW v4 and v6	Yes	MSPW(Yes)	Entropy Label
VPN v4 and v6	Yes	Yes	N/A
BGP v4 and v6	Yes	Yes	N/A
P2MP TE and mLDP	Yes	Yes	N/A
MPLS-TP	Yes	Yes	N/A

LSP ping for Pseudowire FEC

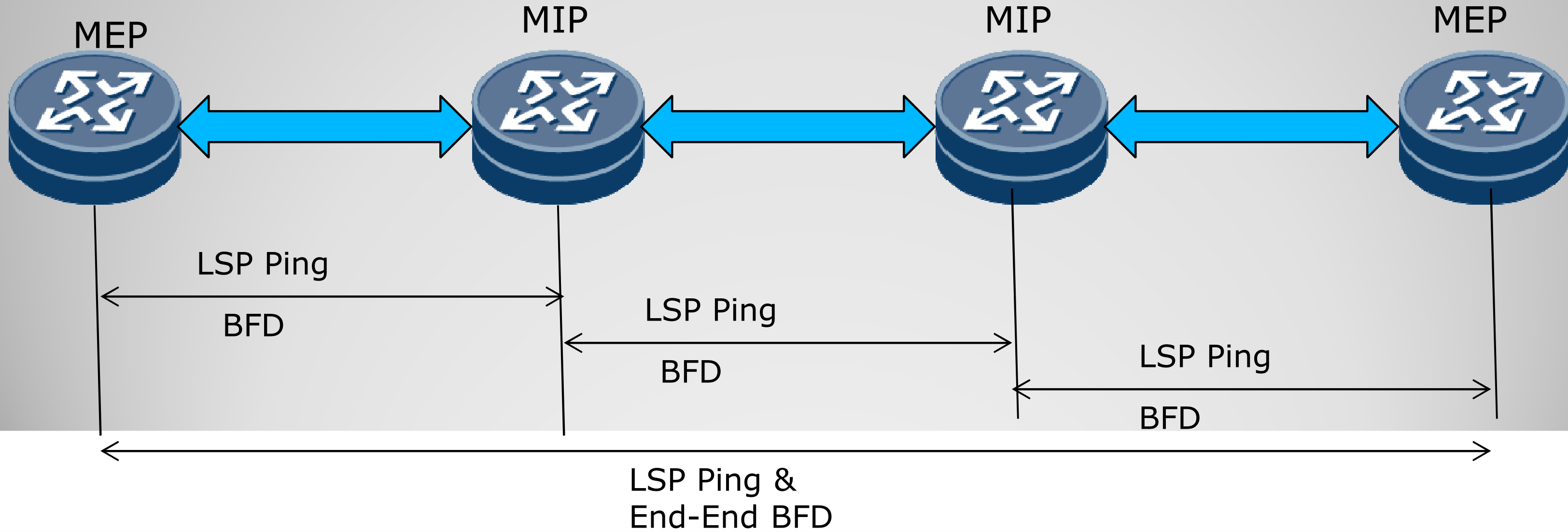
Requirement	**Provide end-to-end fault detection and diagnostic features for emulated Pseudowire service** •P2P PWE3 •MS-PW end-to-end Ping and Trace
Solution	**VCCV provides control channel to allow control packets over Pseudowires** •VCCV capabilities are signalled using control protocols •Ability to support Control Word encapsulation •Router Alert labeled packets are to be
Applications	**Layer 2 transport over MPLS** •EoMPLS •FRoMPLS •ATMoMPLS
Solution	**RFC5085**

BFD for MPLS

- Ability to verify LSP
- BFD to verify TE tunnels, TP tunnels, PW LSP's etc
- VCCV to be used to verify PW LSP's
- BFD could be used to complement or replace use of RSVP hellos for
- MPLS FRR Link/Node protection
- Health check for (PSC) FA-LSP
- BFD to carry AIS, RDI errors to end points of TP tunnels
- BFD the primary mechanism to make fast switchover and meet transport requirements
- BFD to play complimentary role to provide OAM within MPLS

LSP ping & BFD for MPLS-TP

- LSP ping got enhanced to support TP LSP's
- As TP LSP's are mostly statically provisioned, LSP ping plays crucial role in diagnosing faults.
- Ability to perform MEP-MEP, MIP-MEP and MIP-MIP OAM functions
- LSP ping also got enhanced to support performance measurement functions
- BFD is used to fast detect failures and to meet 50msec requirement
- GAL label(13) to identify OAM and BFD packets



Summary

- LSP ping started off with as a tool to detect failures in MPLS networks
- It got enhanced to perform diagnostic functions as well as performance and liveness detection
- BFD and LSP Ping to complement each other to provide OAM within MPLS networks
- Supports all new enhancements to MPLS networks
- Provides support for IPv4 and IPv6
- Capable of performing functions over P2P and P2MP topologies

OAM Best Practices for new Protocol Designers

- OAM MUST NOT be considered as an afterthought or not so “cool”
  - OAM is an essential element of any protocol
  - MUST BE built in to the design
- OAM Frames SHOULD follow identical path/forwarding logic as the regular packets
  - Protocol MUST have ways to identify OAM frames from Data frames, without altering the packet forwarding behavior
- SHOULD NOT Leak outside the OAM domain
  - E.g. MPLS OAM frames should not leak in to customer networks
  - SHOULD NOT trigger invalid packet count etc.
- Do not assume steady state network topology as same as the topology when experiencing network fault

OAM Best Practices for new Protocol Designers

- Need to cover not only Ping and Traceroute but also other aspects such as Performance, Fault Indications/notification etc.
- Special care MUST be taken to prevent attackers exploiting OAM tools
- Wherever possible, re-use existing OAM implementations,
  - We do not have to re-invent the wheel over and over again
- SNMP is not exactly an "O" tool, SNMP can not replace "O" aspects of OAM.
  - Also should not be mistaken with the intent of "O" tools as to replace SNMP
- Where possible include extensibility
  - This allow to accommodate forward compatibility without needing to redo.

Acknowledgments

- Special appreciation to IETF EDU team members for their valuable comments and feedback and for giving this opportunity
- Very Very Special thanks to Radia Perlman for proposing the idea of an OAM tutorial and encouraging us to put together the presentation that we just completed.
- AND Most of all Very Very special thanks to the audience for attending, great questions and above all being part of the 83rd IETF

"Profitez pleinement de votre séjour en France"

"Enjoy Your Stay in France"

Questions