June 2021

INBAND MULTICAST FAULT DETECTION TO REDUCE SERVICE COST

Mankamana Mishra

Anuj Budhiraja

Nitin Kumar

Sridhar Santhanam

INBAND MULTICAST FAULT DETECTION TO REDUCE SERVICE COST

AUTHORS:
Mankamana Mishra
Anuj Budhiraja
Nitin Kumar
Sridhar Santhanam

ABSTRACT

Techniques herein define a simple, but very useful, extension to hop-by-hop signaling that can be utilized to determine a failed node in a network, which may help to reduce fault detection time. In one instance, techniques described herein may involve multicast Label Distribution Protocol (mLDP)-based signaling, however, other replication technologies that involve underlay signaling may be utilized in accordance with techniques described herein.

DETAILED DESCRIPTION

Multicast technologies are deployed throughout many networks. However, multicast distribution trees (MDTs) can be difficult to debug when network failures occur. Figure 1, below, illustrates how current network failures are often handled.

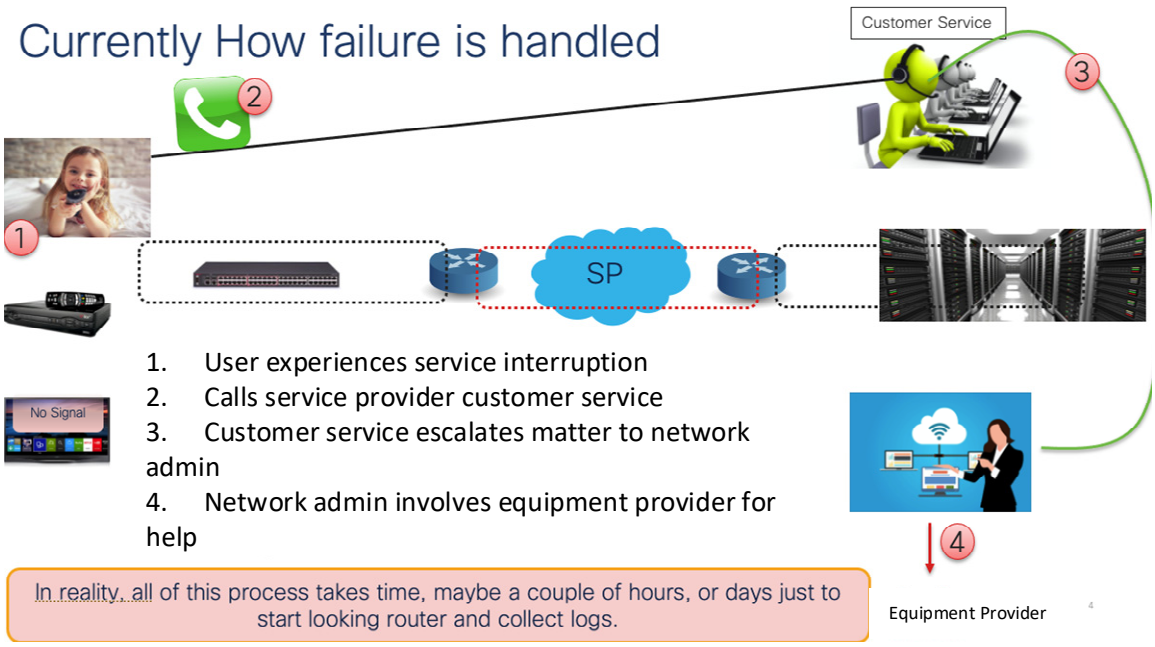

Figure 1: Example Network Failure Handling Flow

As shown in Figure 1, a network failure typically involves:

1. An end user experiencing a service disruption;
2. After some time (potentially after many failures), the user decides to call customer service of a service provider;
3. The customer service identifies a potential network-related issue and reaches out to a local operations team; and
4. The local operations team contacts an equipment provider, which starts trying to understand the potential issue (e.g., asking for logs from different network locations).

In some instances, the above process can take many days before an actual network node is identified as having an issue. For example, in some instances, for a very large network, more than one week of debugging may be involved such that initial debugging may start at a local data center network and, once operation there is confirmed, the debugging may continue to different network segments. In some instances, different parts of a network may also involve debugging from different equipment providers/vendors, which can also lead to increased debugging time.

Accordingly, it can be inherently difficult to debug network failures involving multicast traffic, as opposed to unicast traffic, because there can be many fork points in a network at which issues can be potentially introduced. As business landscapes and competitors change among network equipment provides/vendors, it becomes increasingly important to improve quality of service as a competitive differentiator. Further, there is also often a need for network operators to be able to detect failures within their networks as soon as possible in order to identify the actual node(s) having issues so that quick recovery/resolution is possible. Further, network failures can impact end user experience (e.g., for streaming video, etc.) and may be costly in terms of service cost (e.g., help desk time, network operator troubleshooting costs, equipment provider/vendor troubleshooting costs) and/or lost revenue, which further motivates network operators and equipment providers/vendors to identify/resolve network issues as quickly as possible.

When considering various solutions that may be utilized to address the challenges noted above, two types of deployment scenarios may considered that vary in scale, which

can impact potential in-band versus out-of-band fault detection solutions. For example, a first deployment scenario may be considered in which the number of multicast flows may be much higher than the number of nodes in a given network deployment (e.g., > 350,000+ flows) and a second scenario may be considered in which the number of multicast flows may be much lower than the number of nodes in a given network deployment (e.g., flows expected to be on the order of a few thousand).

When considering network deployments under the second scenario in which the number of nodes may be much higher than the number of flows (e.g., potentially on the order of millions of nodes), it can become very difficult to utilize an out-of-band telemetry-based solution that involves exporting data to a centralized location that would detect traffic loss and take some corrective action. Although such a centralized solution may work well for lower scale deployments (e.g., fewer network nodes), it can still be challenging to handle large scale data for out-of-band solutions. Further, given the massive scale of many Internet Protocol (IP) networks, it may be difficult to maintain an out-of-band solution. Rather, different types of solutions may be needed to handle different issues.

This proposal provides a distributed scale solution that can help to identify problematic network nodes, potential identifying a problem on the order of a number of minutes, which can save many weeks of cost and/or lost revenue for network operators, as well as for equipment providers/vendors.

Broadly, techniques of this proposal may involve:

1. Detecting the traffic loss at an Egress node using available hardware capabilities;
2. Once traffic loss has been detected, an in-band signaling mechanism is utilized to notify an upstream node about traffic loss for given flow; and
3. The upstream node checks for traffic loss. If there is traffic loss, the upstream node notifies its own upstream node or notifies a network administrator as this node to be responsible for traffic loss.

Consider Figure 2, below, which illustrates various example flow characteristics.

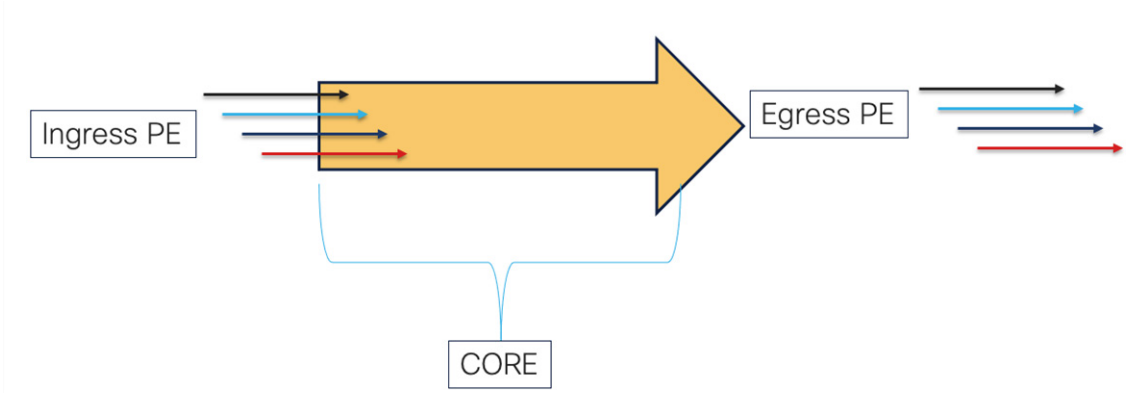

Figure 2: Example Flow Characteristics

Referring to Figure 2, many multicast deployments use the term 'Data multicast distribution tree (MDT)', which could potentially refer to an aggregated representation of multiple customer flows for a given Virtual Routing and Forwarding (VRF) element. In case of any traffic loss, there would not be per-flow traffic loss considering the core network is not even aware of individual flows. So any traffic loss detection would be for all flows associated with a given Data MDT.

Various example details are discussed herein with reference to an example multicast over Virtual Private Network (mVPN) profile, labeled 'profile-X', however it is to be understood that techniques of this proposal may be utilized in conjunction with any profile and multicast technology, which can be enhanced using appropriate type-length-value (TLV) objects and associated protocol extensions (e.g., mLDP, Protocol Independent Multicast (PIM), etc.).

Consider example mVPN profile-X flow Data MDT signaling, as shown in Figure 3, below.

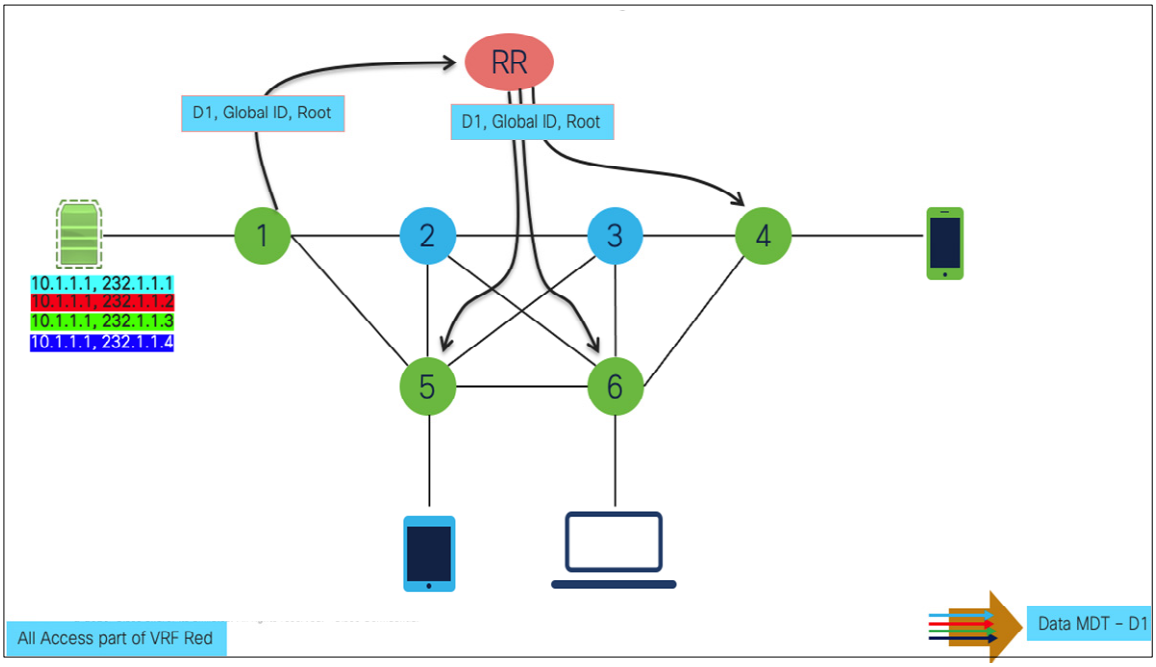

Figure 3: Example mVPN profile-X flow Data MDT Signaling

During operation, once a flow meets the criteria of being moved to the Data MDT (either a configured threshold is met or an immediate switch is configured), the Ingress router notifies each of the Egress Provider Edge (PE) routers participating in given VRF about the flow to Data MDT mapping.

For an implementation in which the core is a Multiprotocol Label Switching (mPLS) core, the Ingress router may send a Global ID and root to use for the mLDP Forwarding Equivalence Class (FEC). For example once receiver behind router-4 (R4) sends multicast join for all of the groups behind router-1 and if Data MDT conditions are met. Router-4 may obtain the Data MDT information as, shown in Table 1, below.

TABLE 1: Data MDT Information Obtained by Router-4 For Example Flows

```
RP/0/0/CPU0:R4#show pim vrf red mdt cache
Sun Nov  1 12:31:57.857 PST

Core Source      Cust (Source, Group)              Core Data        Expires
1.1.1.1          (10.1.1.1, 232.1.1.1)             [global-id 6]    never


1.1.1.1          (10.1.1.1, 232.1.1.2)             [global-id 6]    never


1.1.1.1          (10.1.1.1, 232.1.1.3)             [global-id 6]    never


1.1.1.1          (10.1.1.1, 232.1.1.4)             [global-id 6]    never


RP/0/0/CPU0:R4#
```

Even though there are 4 different multicast flows in the present example, the information would be sent as one label mapping to an upstream node considering all of the flows belong to the same Data MDT. Thus, router-4 may store/maintain the Data MDT, as shown in Table 2, below.

TABLE 2: Data MDT Maintained by Router-4 For Example Flows

```
RP/0/0/CPU0:R4#show mpls mldp database
Sun Nov  1 12:36:40.353 PST
LSM-ID: 0x00006  Type: P2MP  Uptime: 00:05:15
  FEC Root           : 1.1.1.1
  Opaque decoded     : [global-id 6]
  Features           : MBB
  Upstream neighbor(s) :
  Is CSI accepting     : N
    3.3.3.3:0 [Active] [MBB] Uptime: 00:05:15     >>>>>>>>>> Next Hop
      Local Label (D) : 24016
  Downstream  client(s):
    PIM MDT             Uptime: 00:05:15
      Egress intf     : Lmdtred
      Table ID        : IPv4: 0xe0000011 IPv6: 0xe0800011
      RPF ID          : 5
      RD              : 1:1
```

From the data base, as shown in Table 2, it is clear that a join should be sent to next hop 3.3.3.3, as shown in Table 3, below.

TABLE 3: Downstream Neighbor of Router-4

```
RP/0/0/CPU0:R4#show mpls mldp neighbors
Sun Nov  1 12:38:41.778 PST
mLDP neighbor database
 MLDP peer ID       : 3.3.3.3:0, uptime 00:13:06 Up,
  Capabilities      : GR, Typed Wildcard FEC, P2MP, MP2MP, MBB
  Target Adj        : No
  Upstream count    : 2
  Branch count      : 0
  LDP GR            : Enabled
                    : Instance: 1
  Label map timer   : never
  Policy filter in  :
  Path count        : 1
  Path(s)           : 34.34.34.1         GigabitEthernet0/2/0/3 LDP
  Adj list          : 34.34.34.1         GigabitEthernet0/2/0/3
  Peer addr list    : 3.3.3.3
                    : 23.23.23.2
                    : 34.34.34.1
                    : 35.35.35.1
                    : 36.36.36.1
```

Router-3 (R3) may receive an mLDP join for the Data MDT, which would be aggregated for all of the flows, as shown in TABLE 4, below.

TABLE 4: Data MDT Maintained by Router-3 For Example Flows

```
RP/0/0/CPU0:R3#show mpls mldp database
Sun Nov  1 12:40:21.097 PST
mLDP database
LSM-ID: 0x00004  Type: P2MP  Uptime: 00:08:56
  FEC Root           : 1.1.1.1
  Opaque decoded     : [global-id 6]
  Features           : MBB
  Upstream neighbor(s) :
  Is CSI accepting    : N
    2.2.2.2:0 [Active] [MBB] Uptime: 00:08:56
      Local Label (D) : 24012
  Downstream  client(s):
    LDP 4.4.4.4:0      Uptime: 00:08:56
      Next Hop         : 34.34.34.2
      Interface        : GigabitEthernet0/2/0/3
      Remote label (D) : 24016
RP/0/0/CPU0:R3#
```

A network administrator can configure certain flows to be in a monitoring category at the Egress node to enable failure detection at the Egress node. For example, as shown below in Figure 4, traffic is expected to be received on a virtual tunnel from a source and Egress router-4 can send individual flows out on appropriate interfaces. Once a network administrator marks particular flow for monitoring, the associated Data MDT flow can be monitored. Based on nature of Data
MDT, either loss occurs for the whole data MDT (impacting each flow) or not at all.

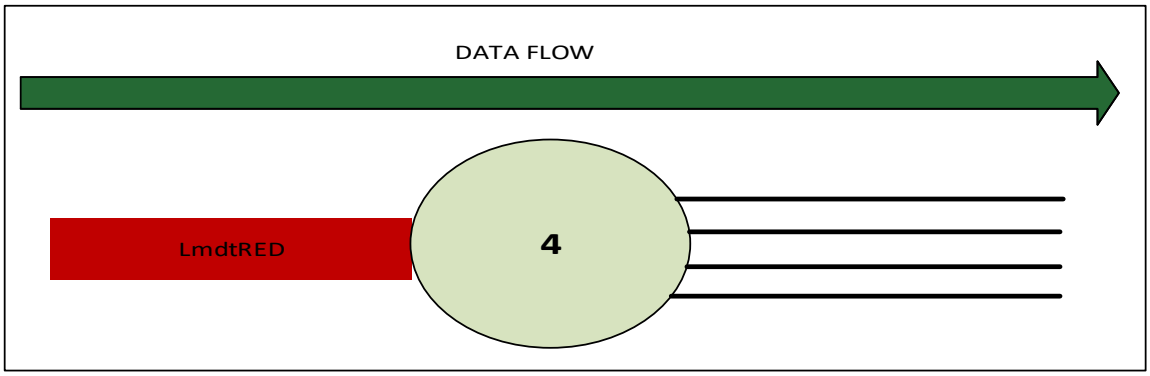

Figure 4: Detecting a Failure at the Egress Router

Once the Data MDT is monitored, flows can also be monitored for Ingress statistics, depending on configuration. For example, as shown below via Figure 5, the network administrator can mark flows to monitor at the Ingress router and, once monitored, if there is traffic loss detected, the next hop can be signaled using mLDP signaling.

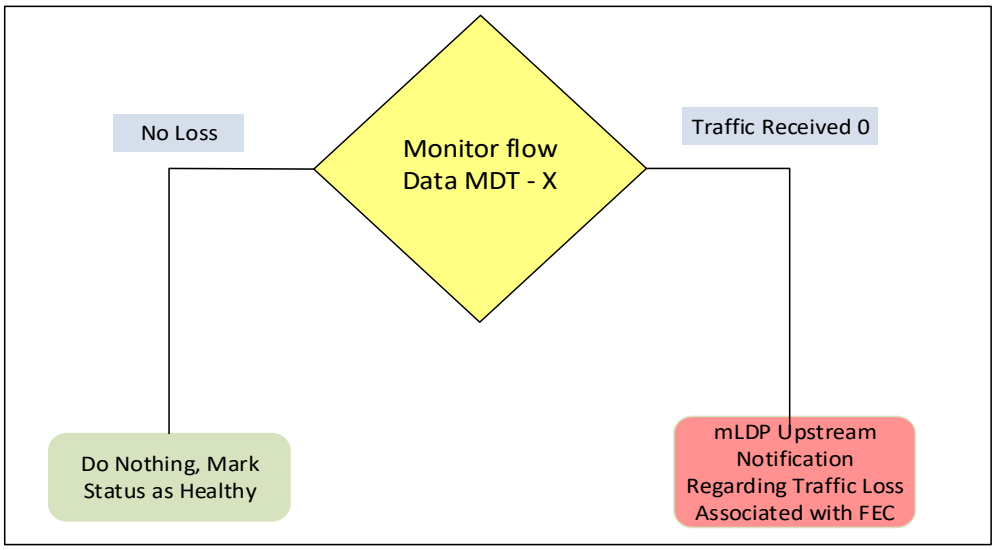

Figure 5: Example Ingress Router Monitoring

Consider an example in which traffic loss is detected at Egress router-4, as shown below in Figure 6.

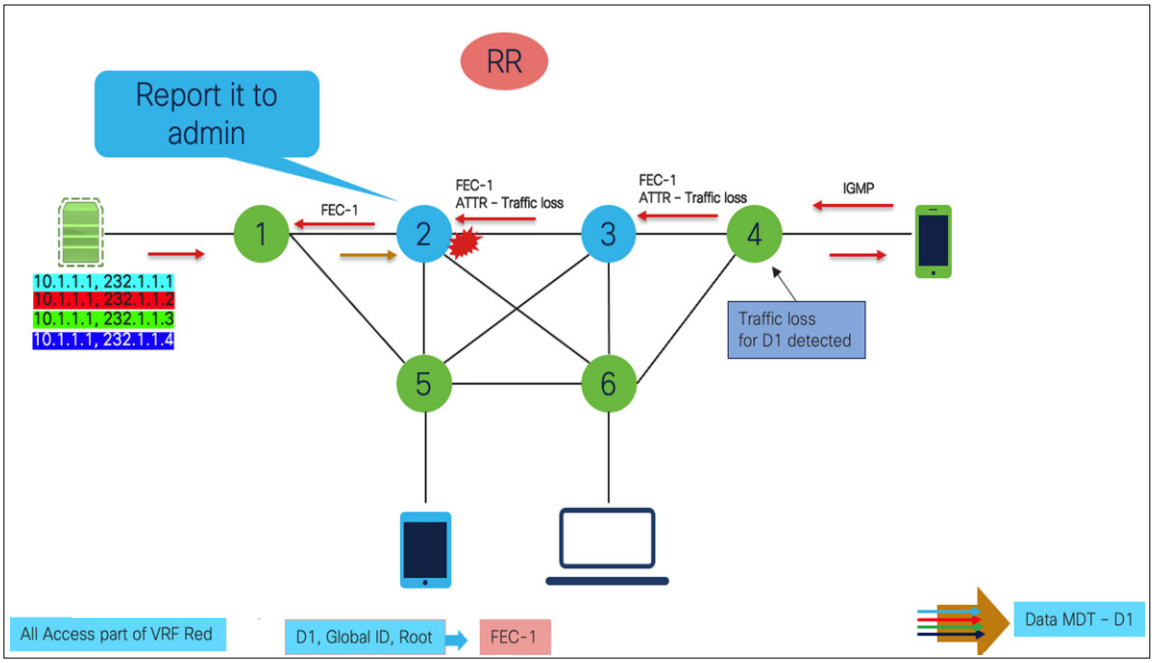

Figure 6: Traffic Loss Example

The intention of this proposal is to cover cases in which traffic is dropped completely. In one example, the Egress node can monitor its counters per "X" seconds, where value of X can be configured based on user need and platform capability. Once Egress detects a complete traffic loss it would start signaling the upstream node (router-3) regarding the loss, as discussed in further detail, below. In another example, overlay signaling can be extended to report a traffic rate range for a given flow and the Egress node can keep monitoring for some %-deviation before determining that a traffic loss has occurred.

If there is a complete traffic drop there are likely two reasons:

1. The source stopped sending traffic, or
2. There is an actual traffic drop in network.

In either case, it would be still good for service provider to know of such traffic losses in advance, before an end-user reaches out to the service provider.

Once traffic loss is detected for the example as shown in Figure 6, mLDP can be utilized to notify the next hop about the loss, as shown below in Table 5.

TABLE 5: Example Loss Notification

```
RP/0/0/CPU0:R4#show mpls mldp database
Thu Nov  5 11:30:15.301 PST
LSM-ID: 0x00006  Type: P2MP  Uptime: 3d22h
  FEC Root           : 1.1.1.1
  Opaque decoded     : [global-id 6]
  Features           : MBB
  Traffic Loss Status: True  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Traffic loss is sent to next hop.
  Upstream neighbor(s) :
  Is CSI accepting    : N
    3.3.3.3:0 [Active] [MBB] Uptime: 3d22h
      Local Label (D) : 24016
  Downstream  client(s):
    PIM MDT            Uptime: 3d22h
      Egress intf     : Lmdtred
      Table ID        : IPv4: 0xe0000011 IPv6: 0xe0800011
      RPF ID          : 5
      RD              : 1:1
```

As shown in Table 6, the following FEC mapping is received on router-3 at which an attribute is set that indicates that there is traffic loss.

TABLE 6: Router-3 FEC Mapping

```
RP/0/0/CPU0:R3#show mpls mldp database
Thu Nov  5 11:34:12.377 PST
LSM-ID: 0x00004  Type: P2MP  Uptime: 3d23h
  FEC Root           : 1.1.1.1
  Opaque decoded     : [global-id 6]
  Features           : MBB
  Traffic Loss Status: False >>>>>>>>>>>>>>>>> Default is expected that there is no traffic loss.
  Upstream neighbor(s) :
  Is CSI accepting    : N
    2.2.2.2:0 [Active] [MBB] Uptime: 3d23h
      Local Label (D) : 24012
  Downstream  client(s):
    LDP 4.4.4.4:0      Uptime: 3d23h
      Next Hop         : 34.34.34.2
      Interface        : GigabitEthernet0/2/0/3
      Remote label (D) : 24016
      Traffic Loss Status: True >>>>>>>>>>>>>>>>>>>>>>>> Notification from downstream about traffic loss.
RP/0/0/CPU0:R3#
```

Since router-3 received a signal from downstream node, an internal check is triggered for router-3 in which router-3 also determines that it is not obtaining traffic from the upstream node. Following the determination, an upstream notification is triggered for router-3, as shown in Table 7, below.

TABLE 7: Upstream Notification Triggered for Router-3

```
RP/0/0/CPU0:R3#show mpls mldp database
Thu Nov  5 11:34:12.377 PST
LSM-ID: 0x00004  Type: P2MP  Uptime: 3d23h
  FEC Root           : 1.1.1.1
  Opaque decoded     : [global-id 6]
  Features           : MBB
  Traffic Loss Status: True >>>>>>>>>>>>>>>> Traffic loss at this node too. Notify upstream.
  Upstream neighbor(s) :
  Is CSI accepting    : N
    2.2.2.2:0 [Active] [MBB] Uptime: 3d23h
      Local Label (D) : 24012
  Downstream  client(s):
    LDP 4.4.4.4:0      Uptime: 3d23h
      Next Hop         : 34.34.34.2
      Interface        : GigabitEthernet0/2/0/3
      Remote label (D) : 24016
      Traffic Loss Status: True
RP/0/0/CPU0:R3#
```

Thereafter, router-3 can send an in-band notification to an upstream node, as shown in Table 8, below.

TABLE 8: Upstream Notification Provided by Router-3

```
RP/0/0/CPU0:R2#show mpls mldp database
Thu Nov  5 11:39:33.159 PST
LSM-ID: 0x00004  Type: P2MP  Uptime: 3d23h
  FEC Root           : 1.1.1.1
  Opaque decoded     : [global-id 6]
  Features           : MBB
  Traffic Loss Status: False.  >>>>>>>>>>>>>>>>>>>> default is false.
  Upstream neighbor(s) :
  Is CSI accepting    : N
    1.1.1.1:0 [Active] [MBB] Uptime: 3d23h
      Local Label (D) : 24012
  Downstream client(s):
    LDP 3.3.3.3:0       Uptime: 3d23h
      Next Hop          : 23.23.23.2
      Interface         : GigabitEthernet0/2/0/1
      Remote label (D) : 24012
      Traffic Loss Status: True >>>>>>>>>>>>>>>>>>>>>>>> Notification from downstream about traffic loss.
RP/0/0/CPU0:R2#
```

Upon obtaining the notification, router-2 (R2) can verify its incoming rates and determine that it is receiving traffic; thus, router-2 would notify the network administrator that it is the node introducing a traffic loss. An example notification is shown below in Table 9.

TABLE 9: Example Notification from the Node at which Loss is Occurring

```
RP/0/0/CPU0:Nov  5 11:39:40.942 PST: mpls_ldp[1228]: %ROUTING-MLDP-TRAFFIC-LOSS  : MLDP downstream traffic loss. LSM-
ID: 0x00004, Local Label: 24012, Downstream client:
LDP 3.3.3.3:0 , NH:23.23.23.2, Remote Label:24012, Outgoing interface:GigabitEthernet0/2/0/1
```

The amount of information that could be provided to the network administrator, as shown in Table 9, would be sufficient to start debugging the node. For example, automated scripts can be provided that can examine traces and commands related to the above information that may provide enough context to identify the possible root cause of the issue.

As noted above, traffic is being monitored on a per-Default MDT or Data MDT and is based on detecting a 100% traffic loss. Although there may be considerations concerning false positives and potentially unnecessary signaling if some flows are removed from the

Default MDT, if data rate is reduced (e.g., variable bitrate), or when a source stops sending data, these cases may be addressed through different configurations that can be managed for a given implementation/deployment.

Broadly, techniques of this proposal can operationally be realized as follows:

1. Traffic for a multicast flow starts flowing.
2. Initially, it flows in a Default MDT and then finally settles in a Data MDT.
3. Now, the service provider/network operator decides which all Data MDTs or Default MDT it wants to monitor.
4. The Egress node starts polling counters every "X" seconds/minutes (depending on hardware capability and user tolerance).
5. If the Egress detects traffic rate as 0 for those "X" seconds, it starts the procedures as discussed above with reference to Figure 6.

Based on the above, consider various issues that can be addressed through network configuration. For example, regarding bursty flows, there are many ways to 'break' a deployment. Consider the case where a network administrator configures 10 seconds as the timer to poll and detect traffic loss and the source sends traffic for 2 seconds and stops for the next 11 seconds. In this example, a loss would be detected. Theoretically, it may be true that additional signaling may be triggered for bursty traffic utilizing the techniques herein, but such a bursty traffic example is a non-practical use-case. If there is some source behaving like this for traffic such as IP television (IPTV), such bursty traffic is a bigger problem that should be addressed rather than detecting traffic loss.

Regarding potential false positives for instances in which a source has stopped sending traffic for a long period of time, the only case where false positives could potentially occur is with bursty flows, which is not a valid case (as explained above). If the source really stops sending traffic, it is only a one-time signaling end to end. The network administrator would receive a loss report and there is no additional signaling would occur since the traffic itself has stopped. In reality, with the IPTV use case, this may be a scenario that may occur every now and then. So, this can be considered as noise.

Regarding flow removal from Default MDT to Data MDT, since monitoring is per MDT not inside the MDT, it does not matter if flows are being moved around. By nature,

there is no way to have traffic loss for only one flow in an MDT if multiple flows are aggregated.

When considering data rates on different nodes, depending on how they are sampled, there may be some deviations; however, techniques of this proposal involve comparing data rates from zero to non-zero, so sampling may not be a concern. Consider an example, as shown below in Figure 7.

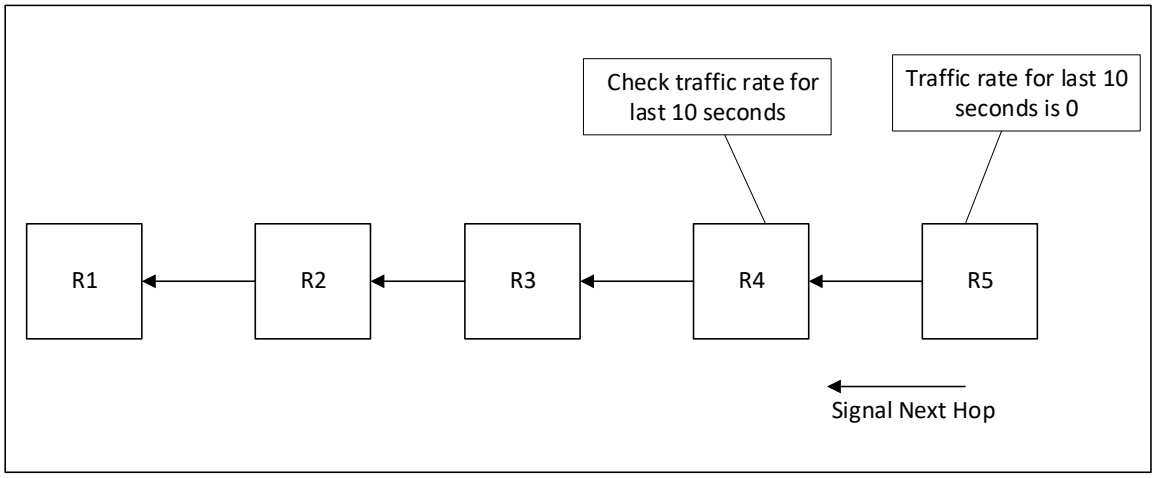

Figure 7: Data Rate Consideration Example

Consider for the above figure an example in which router-1 (R1) is sending traffic downstream that is being received by router-5 (R5) in which the following occurs:

1. Router-5 detects traffic loss for 10 seconds at time t0.
2. Router-5 signals the next hop for this flow stating that there is traffic 0 for the last 10 seconds.
3. By the time the signaling reaches router-4 (R4) an amount of time has already elapsed.
4. Thus, sampling at router-4 will start only at t0+delta. It also takes a sample for the last 10 seconds. So, there are only two possible cases:
    a. Router-4 is receiving traffic, which means the issue is between router-4 and router-5, or
    b. Router-4 also has zero traffic, so it needs to signal next hop.

As illustrated through the above example, the intention is to detect a complete traffic loss; thus, there won't be any deviation unless the source is bursty in nature (sending traffic for every "X" seconds and pausing for the next "Y" seconds) which, as discussed above, may not be a realistic use-case.

Further, the innovation of the proposal is discussed with reference to mLDP but it is not limited to just one protocol. The techniques described herein remain the same irrespective of different tree-building in-band protocols; the only changes may involve respective TLVs to be defined in other respective protocols. The primary novelty of the techniques of this proposal involve the application and set of procedures to achieve fault detection in a network environment. Thus, a mechanism is provided through which rest of a network could poll data on-demand, only if there is real loss detected in the last-hop router.

In summary, techniques herein define a simple, but very useful, extension to hop-by-hop signaling that can be utilized to determine a failed node in a network, which may help to reduce fault detection time as well as reduce service provider/network operator cost and equipment provider/vendor cost in debugging network failures.