Technical Disclosure Commons

Defensive Publications Series

November 20, 2018

MULTICAST ROUTING DIAGNOSTICS SYSTEM USING MACHINE LEARNING

Luc De Ghein

Santosh Patil

Follow this and additional works at: https://www.tdcommons.org/dpubs series

Recommended Citation

Ghein, Luc De and Patil, Santosh, "MULTICAST ROUTING DIAGNOSTICS SYSTEM USING MACHINE LEARNING", Technical Disclosure Commons, (November 20, 2018) https://www.tdcommons.org/dpubs_series/1688



This work is licensed under a Creative Commons Attribution 4.0 License.

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

MULTICAST ROUTING DIAGNOSTICS SYSTEM USING MACHINE LEARNING

AUTHORS: Luc De Ghein Santosh Patil

ABSTRACT

Techniques are described herein for using Machine Learning to detect anomalies that cannot be detected by programming or scripts. These anomalies must be discovered by prior issues that have been troubleshooted by classic network tools (e.g., debugs, show commands, packet capturing tools, etc.).

DETAILED DESCRIPTION

Multicast traffic relies on the forwarding state on routers to be forwarded correctly. Even when the forwarding state is present and correct on routers, the multicast traffic can be dropped or some degradation can occur. These issues are difficult to detect. The best way to track the correct forwarding of all multicast traffic is to track the state and the forwarding statistics at the same time on all nodes in the network. There are currently no mechanisms that achieve this.

Multicast is mostly used to deliver in-time video streams to the end-user. Existing solutions use scripting or programming to detect known or previously discovered issues that might occur with multicast routing in networks. As such, intermittent multicast traffic degradation and faulty topology states are often discovered very late. Accordingly, techniques described herein use Machine Learning to learn multicast state, topology, and forwarding statistics on routers.

The following examples are issues that can be detected using the techniques presented herein:

- incorrect state (incoming interface, outgoing interfaces, the multicast state itself, multicast entry and interface flags)
 - tracking/learning the total number of multicast forwarding entries
 - tracking/learning the type of entries (shared tree, source tree, BiDIR)
 - tracking/learning the sources
 - tracking/learning the used rendezvous points

- forwarding rate of traffic per multicast forwarding entry; this can be the rate of forwarding, stale forwarding statistics, change of type of forwarding counters (for example Reverse Path Forwarding (RPF) failures)
 - tracking/learning the timers and their change (often too low or high)
- tracking/learning the complete tree of the multicast trees, if the Machine Learning gathers the information on all routers (directly or through a collector)
- any multicast state on routers, switches, servers, or other devices can be tracked by Machine Learning, including multicast state (control plane and hardware forwarding entries), multicast application state, and multicast client state

Figure 1 below illustrates an example output of the multicast state learning (entries, timers, flags, etc.). Anomalies are shown in blue.

```
P#show ip mroute
                      IP Multicast Routing Table
                      Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
L - Local, P - Pruned, R - RP-bit set, F - Register flag,
                             T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
                             X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
                             U - URD, I - Received Source Specific Host Report,
                             Z - Multicast Tunnel, z - MDT-data group sender,
                             Y - Joined MDT-data group, y - Sending to MDT-data group,
                             G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
                             N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
                             Q - Received BGP S-A Route, q - Sent BGP S-A Route,
                             V - RD & Vector, v - Vector, p - PIM Joins on route, shared tree (*,G) entry
                             x - VxLAN group, c - PFP-SA cache created entry
                      Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
                      Timers: Uptime/Expires
                       Interface state: Interface, Next-Hop or VCD, State/Mode
                      (*, 239.0.0.150), 5d21h/00:02:56, RP 10.128.0.20, flags: S +
                                                                                                   multicast entry flags
one incoming interface -
                       Incoming interface: Null, RPF nbr 0.0.0.0
                       Outgoing interface list:
                          GigabitEthernet3, Forward/Sparse, 5d21h/00:02:56
 list of outgoing
                          GigabitEthernet2, Forward/Sparse, 5d21h/00:02:49
 interfaces
                                                                                                 source tree (S.G) entry
                          GigabitEthernet4, Forward/Sparse, 5d21h/00:02:36
                      (10.128.0.18, 239.0.0.150), 5d21h/00:03:27, flags: TJ
                                                                                                       timers
RPF is NULL
                        Incoming interface: Null, RPF nbr 0.0.0.0
                        Outgoing interface list:
                          GigabitEthernet3, Forward/Sparse, 5d21h/00:03:17
                          GigabitEthernet4, Forward/Sparse, 5d21h/00:02:51
```

Figure 1

Figure 2 below illustrates an example output of multicast forwarding rates. Anomalies are shown in blue.

```
P#show ip mroute count
              Use "show ip mfib count" to get better response time for a large number of
              mroutes.
              IP Multicast Statistics
              11 routes using 14922 bytes of memory
              4 groups, 1.75 average sources per group
              Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per
                                                                                                       successful
                                                                                                       forwarding
              Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)
                                                                                                       counters
shared tree
              Group: 239.0.0.150, Source count: 3, Packets forwarded: 112073, Packets
(*,G) entry
              received: 112078
                RP-tree: Forwarding: 8/0/69/0, Other: 8/0/0
                Source: 10.128.0.17/32, Forwarding: 25798/0/87/0, Other: 25801/3/0 Source: 10.128.0.18/32, Forwarding: 60470/0/95/0, Other: 60471/1/0 Source: 10.100.1.3/32, Forwarding: 25797/0/87/0, Other: 25798/1562/0
 source tree
 entries
              Group: 239.0.0.100, Source count: 3, Packets forwarded: 112232, Packets
              received: 112236
                                                                                                    increase of RPF
                 RP-tree: Forwarding: 7/0/70/0, Other: 7/0/0
                                                                                                    failures
                 Source: 10.128.0.17/32, Forwarding: 51620/0/93/0, Other: 51621/1/0
                 Source: 10.128.0.18/32, Forwarding: 34809/0/85/0, Other: 34811/2534
                 Source: 10.100.1.3/32, Forwarding: 25796/0/87/0, Other: 25797/1/0
                                                                                             increase of other drops
                                                    forwarding counter has stopped
```

Figure 3 below illustrates usage of Machine Learning for multicast on routers.

Figure 2

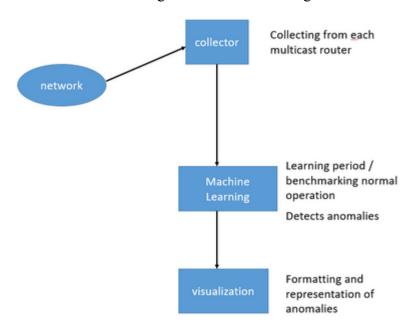


Figure 3

Techniques described herein provide real time feedback on multicast state and forwarding traffic rates. Issues may be detected that today are only possible to detect through superficial testing or complex and lengthy troubleshooting techniques. Faster warnings may be provided for issues that are compared to the learned benchmark information in the network. No complex input to tracking devices is required because

Machine Learning is used. Moreover, no rules need to be written to define the normal operation or need to be maintained for specific devices / networks for all the multicast traffic state and forwarding rates on all the routers (no scripting or programming is required). Machine Learning learns the multicast flows and their rates and benchmarks it.

In summary, techniques are described herein for using Machine Learning to detect anomalies that cannot be detected by programming or scripts. These anomalies must be discovered by prior issues that have been troubleshooted by classic network tools (e.g., debugs, show commands, packet capturing tools, etc.).

4 5743