January 02, 2019

PERFORMANCE DELAY MEASUREMENT AND FAULT DETECTION OF SR / SRV6 TE POLICIES IN SOFTWARE DEFINED NETWORKS

Clarence Filsfils

Rakesh Gandhi

Tarek Saad

Patrick Khordoc

Sagar Soni

PERFORMANCE DELAY MEASUREMENT AND FAULT DETECTION OF SR / SRV6 TE POLICIES IN SOFTWARE DEFINED NETWORKS

AUTHORS:

Clarence Filsfils

Rakesh Gandhi

Tarek Saad

Patrick Khordoc

Sagar Soni

ABSTRACT

[001] Mechanisms are provided for Performance Measurement (PM) Delay Measurement (DM) as well as Fault Detection (FD) for Traffic Engineering (TE) Segment Routing (SR) policies (applicable to both SR-MPLS and SRv6 data-planes) in Software Defined Networks (SDN). For an SR policy, end-to-end atomic paths are "dynamically" computed using adjacency SIDs (or prefix SIDs in some cases) of all its ECMP paths by the path computation process running locally or on a controller. Key of the atomic path based on IP address of each hop is used for Telemetry, histogram and Client notification. The PM probe messages (or FD messages) are injected for the SR policy where the packets are sent with each atomic path SID stacks. These packets are sent on all atomic paths of the SR policy in order to "deterministically" detect performance issues and faults where the atomic paths do not need to be installed in forwarding table. Additionally, these packets may be crafted by appending VNF function SIDs to steer them to a specific VNF. Controller then may update the segment-list of the SR policy in forwarding to prune degraded atomic paths.

DETAILED DESCRIPTION

[002] Segment-routing (SR) is a new technology that greatly simplifies network operations and makes networks SDN-friendly. SR is applicable to both MPLS (SR-MPLS) and IPv6 (SRv6) data-planes. Built-in Performance Measurement (PM) and Fault Detection (FD) are one of the

essential requirements for the success of this new technology. SR policies are used to steer traffic through a specific, user-defined path using SID list for Traffic Engineering (TE).

[003] Presented herein are mechanisms for Performance Measurement (PM) Delay Measurement (DM) as well as Fault Detection (FD) for Traffic Engineering (TE) Segment Routing (SR) policies (applicable to both SR-MPLS and SRv6 data-planes) in Software Defined Networks (SDN). For an SR policy, end-to-end atomic paths are "dynamically" computed using adjacency SIDs (or prefix SIDs in some cases) of all its ECMP paths by the path computation process running locally or on a controller. The PM probe messages (or FD messages) are injected for the SR policy where the packets are sent with each atomic path SID stacks. These packets are sent on all atomic paths of the SR policy in order to "deterministically" detect performance issues and faults where the atomic paths do not need to be installed in forwarding table. Additionally, these packets may be crafted by appending VNF function SIDs to steer them to a specific VNF. Controller then may update the segment-list of the SR policy in forwarding to prune degraded atomic paths.

[004] The techniques presented herein measure “end-to-end” performance delay values of all atomic paths of one or more candidate-paths of an SR Policy (irrespective of candidate-path installed in the forwarding table or not) and compute delay metrics.

[005] The solution is applicable to both SR-MPLS and SRv6 data planes.

[006] In an SR network, there is a requirement to measure end-to-end performance delay of customer traffic as well as detect fault in data plane on SR policies (as shown in FIG. 1, from PE node 2 to PE node 5) for the following use-cases:

- Performance delay measurement for end-to-end SR policy.
- Router-based and centralized controller-based PM and FD use-cases for end-to-end SR policy.

[007] Performance delay measurement then can be used to provide Service Level Agreements (SLAs).

[008] Segment Identifier (SID) lists for an SR policy are computed with a minimal SID stack depth that take advantage of all its Equal Cost Multi-Path (ECMP) paths in the network. This

eliminates the need to install SID stacks for all possible end-to-end forwarding paths (called atomic paths) in forwarding table that has scale impact due to large number of ECMP paths.

[009] As shown in FIG. 1, SR-MPLS Policy from ingress PE node 2 to egress PE node 5 has six atomic paths due to presence of ECMP paths between ingress and transit nodes, between any two transit nodes and between transit and egress nodes. ECMP paths have the same cost (metric) values. Metric used may be link delay metric or link IGP metric.

[010] End-to end atomic paths for this SR Policy using prefix SIDs can be represented as:

<16009,16004,16005>, outgoing-interface1

<16003,16004,16005>, outgoing-interface2 (top link between nodes 3 and 4)

<16003,16004,16005>, outgoing-interface2 (bottom link between nodes 3 and 4)

<16007,16008,16005>, outgoing-interface3

<16006,16008,16005>, outgoing-interface4 (top link between nodes 6 and 8)

<16006,16008,16005>, outgoing-interface4 (bottom link between nodes 6 and 8)

[011] Delay experienced by the traffic on atomic path <16009,16004,16005> might be different than the delay experienced on atomic path <16006,16008,16005>.

[012] This SR Policy requires to create 6 PM sessions on ingress PE node 2 to measure delays on all 6 of its atomic paths.

[013] In addition to the atomic paths, PM measurements and fault detection information may need to be collected for packets steered over certain VNFs on the same physical network path. This is usually realized by appending VNF SIDs - in addition to the network-path adjacency/node/SIDs (e.g. for services like firewall, network address translation (NAT), etc.).

[014] The following conventions are used for the examples presented herein.

Segment Routing Global Block: 16000 to 23999

Node k has Prefix SID 1.1.1.k/32

Node k has Prefix SID label 16000+k

Link Address of n^{th} adjacency between XY: 99.X.Y.nX

MPLS label n^{th} adjacency between XY: 24nXY

A SID list is represented as <S1, S2, ... Sn> where S1 is the first SID

Atomic Paths

[015] Recall that on Ingress PE node 2, for an SR policy **segment-lists (SID list)** LIST1 *<s4, s5>* and LIST2 *<s8, s5>* are installed in forwarding table to carry traffic.

[016] Atomic paths are defined as actual end-to-end forwarding paths of the SR policy, i.e. paths taken by the data traffic flows in the network.

[017] Following steps are defined for end-to-end performance delay measurement and fault detection using atomic-paths of this SR policy as shown in FIG. 2, where atomic paths are not installed in the forwarding table.

[018] FIG. 3 shows a SID list and Atomic Paths (SID stack) of an SR Policy.

Solution using Atomic Paths

[019] 1. SRTE Path computation is enhanced to expand all segment-list (SID list) of the SR policy into end-to-end atomic paths (e.g. MPLS label stack) using adjacency SIDs of each hop of the path. Path computation contains an entire Shortest Path Tree (SPT) for the SR policy covering all ECMP paths.

[020] 2. For example, SID list <s4, s5> may be expanded as 3 atomic paths: *<A29, A94, A45>, <A23, $A_{(1)}$34, A45> <A23, $A_{(2)}$34, A45>*. Here, A_i34 is the i^{th} adjacency SID between node 3 and node 4.

[021] 3. If there is only one path between two nodes (i.e. a single adjacency), prefix SID of the node may be used instead of adjacency SID.

[022] 4. These atomic path SID stacks are not installed in forwarding.

Example of Atomic Path Expansion

SR-MPLS Policy - End-to-end Atomic Path Expansion

[023] Reference is now to FIG4. SRTE expands the segment-list(s) of the candidate-path(s) of the SR Policy into end-to-end atomic paths with adjacency SID for each hop or prefix SID for each node (if there is only one ECMP path between two nodes) as following:

[024] LIST1 *<16004, 16005>*

Atomic path 1: Label stack <24129, 16004, 16005> (outgoing-interface1, next-hop-address1)

Atomic path 2: Label stack <24123, 24134, 16005> (outgoing-interface2, next-hop-address2)

Atomic path 3: Label stack <24123, 24234, 16005> (outgoing-interface2, next-hop-address2)

[025] LIST2 *<16008, 16005>*

Atomic path 4: Label stack <24127, 16008, 16005> (outgoing-interface3, next-hop-address3)

Atomic path 5: Label stack <24126, 24168, 16005> (outgoing-interface4, next-hop-address4)

Atomic path 6: Label stack <24126, 24268, 16005> (outgoing-interface4, next-hop-address4)

[026] Atomic paths for SR Policy are dynamically computed and updated when there is a topology change.

[027] Performance Measurement (PM) measures delay values on these expanded end-to-end atomic paths of the SR Policy. Atomic paths allow deterministic performance measurement for SR Policies.

[028] If there is only one ECMP path between two nodes, then prefix SID of the next node is used instead of adjacency SID(s) when expanding the segment-list.

SR-MPLS Policy - Keys for End-to-end Atomic Paths

[029] SRTE builds Keys for the end-to-end atomic paths of the SR Policy using IP address of each hop.

[030] The Keys allow to uniquely identify the atomic paths of a Candidate-path.

[031] The key is used for Telemetry, Histogram and Client notifications.

LIST1 *<16004, 16005>*

Atomic path 1: Key <99.2.9.12, 1.1.1.4, 1.1.1.5>

Label stack <24129, 16004, 16005>

Atomic path 2: Key <99.2.3.12, 99.3.4.13, 1.1.1.5>

Label stack <24123, 24134, 16005>

Atomic path 3: Key <99.2.3.12, 99.3.4.23, 1.1.1.5>

Label stack <24123, 24234, 16005>

LIST2 *<16008, 16005>*

Atomic path 4: Key <99.2.7.12, 1.1.1.8, 1.1.1.5>

Label stack <24127, 16008, 16005>

Atomic path 5: Key <99.2.6.12, 99.6.8.16, 1.1.1.5>

Label stack <24126, 24168, 16005>

Atomic path 6: Key <99.2.6.12, 99.6.8.26, 1.1.1.5>

Label stack <24126, 24268, 16005>

[032] Strictly speaking: For atomic path 1, 1.1.1.4 and 16004 are not required in the label stack. For atomic path 4, 1.1.1.8 and 16008 are not required in the label stack.

Probe Packets on Ingress Node

[033] Probe query packets contain the label stacks of the atomic paths of the SR Policy.

[034] Probe query packets are timestamped (T1) just before they go out on wire.

Atomic Paths for Two-way Measurement and Detection

[035] For two-way delay measurement, SRTE computes the atomic path SID stack (e.g. MPLS label stack) for the forward direction as well as co-routed reverse direction path. The reverse atomic path SID stack is built using the adjacency SIDs of the opposite end of the links.

[036] Control-plane processes (PM/FD) can then uses the bidirectional SID stacks (forward and reverse combined) for sending probe packets from the control-plane.

[037] As packet contains the SID stack for the entire path in the forward and reverse direction, egress node of the SR policy is agnostic to these packets with the reverse SID stack.

Atomic Paths using L2-Adjacency SIDs

[038] *draft-ietf-isis-l2bundles* defines L2-adjacency SIDs for bundle members. SRTE can compute end-to-end atomic paths (label stacks) over L2 bundle members using the L2-adjacency SIDs from the topology database. Such end-to-end atomic path label stacks over bundled member interface can be used for sending performance delay measurement probe messages.

Usage of Delay Measurement

[039] 1. When upper-bound is crossed:

i. SRTE may de-activate the candidate-path in forwarding table, or

ii. SRTE may de-activate the segment-list in forwarding table, or

[040] 2. SRTE may advertise the delay metric of the SR Policy via BGP-LS.

[041] 3. SRTE may verify the delay metric of an in-active candidate-path or its segment-lists before activating them in forwarding table.

SR-MPLS Policy with Controller for Atomic Paths

[042] Reference is now made to FIG. 5 that shows a controller based atomic paths computation embodiment. A centralized controller may be employed for computing end-to-end atomic path label stacks using adjacency SIDs of an SR Policy. The SRTE on a router provides the segment-list for an SR Policy to the controller and the controller returns the atomic path label stacks using adjacency SIDs of the SR Policy.

[043] The controller may optionally provide atomic path label stacks using L2-adjacency SIDs when using bundled interface. The controller may download in the SRTE on the router the new atomic path label stacks when there is a change in network topology.

SR-MPLS Policy Controller - EDT Handling

[044] FIG. 6 illustrates handling of Event Driven Telemetry (EDT) on a controller. A controller may get notified of the delay metrics per atomic path (SID stack) for an SR policy via EDT. Due to the large scale of atomic paths (SID stacks), it may be more suitable to send the end-to-end atomic path PM data to the controller. Controller may re-compute the segment-list of the SR policy pruning the degraded atomic paths. Controller may send the updated segment-list of the SR policy to the router that will be installed in forwarding table to carry data traffic. Updated segment-list in forwarding table avoids the degraded atomic paths for the data traffic.

Performance Measurement and Fault Detection over VNFs

[045] In addition to the atomic path, PM measurements can be collected for traffic steered over certain VNFs on the same physical network path. This is usually realized by appending VNF SIDs - in addition to the network-path adjacency/node/prefix SIDs (e.g. for services like firewall, NAT, etc.) for PM query packets, or fault detection packets.

SR Policy - Scale Challenge

[046] Reference is now made to FIG. 7.

[047] For a multi-hop SR Policy, there can be ECMP paths between ingress and transit nodes, between any two transit nodes, and between transit and egress nodes. This can result in a very large number of end-to-end atomic paths (e.g., 3x3x3=27 as shown in the Topology for 3 ECMP paths between two nodes) for the SR Policy.

[048] This "explosion" of end-to-end atomic paths can create a scale problem as a large number of PM sessions are needed to be created for delay measurement for the SR Policy.

Support for SRv6

[049] SR Policy can be created for SR-MPLS or SRv6 data-planes. Although, in this document PM probe packets are shown with SR-MPLS labels, the solution applies equally to SRv6 data-plane.

[050] Reference is now made to FIG. 8. FIG. 8 illustrates a block diagram of a network element (node) 500 configured to perform the operations described in connection with FIGs. 1-5. The network node 500 includes one or more control processors 510, memory 520, a bus 530 and a network processor unit 540. The control processor 510 may be a microprocessor or microcontroller. The network processor unit 540 may include one or more Application Specific Integrated Circuits (ASICs), linecards, etc., and facilitates network communications between the node 500 and other network nodes as well as a controller.

[051] There are a plurality of network ports 542 at which the node 500 receives packets and from which the node 500 sends packets into the network. The processor 510 executes instructions associated with software stored in memory 520. Specifically, the memory 520 stores instructions for control logic 550 that, when executed by the processor 510, causes the processor 510 to perform the operations described herein or to control the network processor unit 540 to perform the operations described herein. The memory 520 also stores configuration information 560 received from a network controller to configure the network node according to desired network functions. It should be noted that in some embodiments, the control logic 550 may be implemented in the form of firmware implemented by one or more ASICs as part of the network processor unit 540.

[052] The memory 520 may include read only memory (ROM) of any type now known or hereinafter developed, random access memory (RAM) of any type now known or hereinafter developed, magnetic disk storage media devices, tamper-proof storage, optical storage media devices, flash memory devices, electrical, optical, or other physical/tangible memory storage devices. In general, the memory 520 may comprise one or more tangible (non-transitory) computer readable storage media (e.g., a memory device) encoded with software comprising computer executable instructions and when the software is executed (by the processor 510) it is operable to perform the network node operations described herein.

[053] Reference is now made to FIG. 9. FIG. 9 illustrates a block diagram of a computing/control entity 600 that may perform the functions of the controller as described herein. The computing/control entity 600 includes one or more processors 610, memory 620, a bus 630 and a network interface unit 640, such as one or more network interface cards that enable network connectivity. The memory 620 stores instructions for control and management logic 650, that when executed by the processor 610, cause the processor to perform the software defined network controller operations described herein.

[054] The memory 610 may include ROM of any type now known or hereinafter developed, RAM of any type now known or hereinafter developed, magnetic disk storage media devices, tamper-proof storage, optical storage media devices, flash memory devices, electrical, optical, or other physical/tangible memory storage devices. In general, the memory 620 may comprise one or more tangible (non-transitory) computer readable storage media (e.g., a memory device) encoded with software comprising computer executable instructions and when the software is executed (by the processor 610) it is operable to perform the network controller operations described herein.

[055] The embodiments presented herein define mechanisms for Performance Measurement (PM) Delay Measurement (DM) as well as Fault Detection (FD) for Traffic Engineering (TE) Segment Routing (SR) policies (applicable to both SR-MPLS and SRv6 data-planes) in Software Defined Networks (SDN). For an SR policy, end-to-end atomic paths are "dynamically" computed using adjacency SIDs (or prefix SIDs in some cases) of all its ECMP paths by the path computation process running locally or on a controller. Key of the atomic path based on IP address of each hop

is used for Telemetry, histogram and Client notification. The PM probe messages are injected for the SR policy where the packets are sent with atomic path SID stacks. These packets are sent on all atomic paths of the SR policy in order to "deterministically" detect performance issues and faults where the atomic paths do not need to be installed in forwarding table. Additionally, these packets may be crafted by appending VNF function SIDs to steer them to a specific VNF. Controller then may update the segment-list of the SR policy in forwarding to prune degraded atomic paths.

[056] The above description is intended by way of example only. Although the techniques are illustrated and described herein as embodied in one or more specific examples, it is nevertheless not intended to be limited to the details shown, since various modifications and structural changes may be made within the scope and range of equivalents of the claims.