Technical Disclosure Commons

Defensive Publications Series

June 2020

MULTIPART, MULTIPATH ENTROPY QUERY BASED ON EGRESS LINK COMBINATION

Nagendra Kumar Nainar

Carlos M. Pignataro

Sagar Soni

MULTIPART, MULTIPATH ENTROPY QUERY BASED ON EGRESS LINK COMBINATION

AUTHORS:
Nagendra Kumar Nainar
Carlos M. Pignataro
Sagar Soni

ABSTRACT

Dynamic entropy ranges are runtime operational state pre-use cached results of entropy calculations based on an actual state of links. These results may include proactive entropy calculations based on the up- and down-states of links. A non-binary state for links that describes their load can also be included. This information enables network optimization and traffic engineering by introducing the capability of querying additional details in the multipart-multipath entropy query. Querying can be accomplished using Label Switched Path (LSP) Ping or any other Operations, Administration, and Maintenance (OAM) extensions, such as Yet Another Next Generation (YANG). Upon receiving such a query, a transit node will reply back with entropy ranges for different combinations of link failure. The response is cached by the initiator and used accordingly based on failure detection.

DETAILED DESCRIPTION

Equal Cost Multi-Path (ECMP) is commonly utilized in networks to provide efficient load balancing and network resiliency. In ECMP environments, path selection is performed by a transit node based on local hashing mechanisms that considers various key values from the packet header, such as IP header info, IPv6 flow labels, entropy labels, and the like, as well as local variables, such as incoming interface IDs and loopback addresses.

The entropy along ECMP paths can be discovered by performing a LSP ping multi-path tree trace to query the entropy along the queried paths by setting some header fields (such as destination addresses, label stacks, or UDP ports) as static while using other header field (such as source addresses in the 127.0.0.0/8 range) to query the range of entropy for each available ECMP paths. The obtained ranges are used to query subsequent nodes until the operation is completed. The collected entropy information is primarily used for OAM

purposes, such as end-to-end path validation using distributed or centralized OAM. Service Assurance for Intent-based Network (SAIN) is another use case that leverages such capability for service assurance.

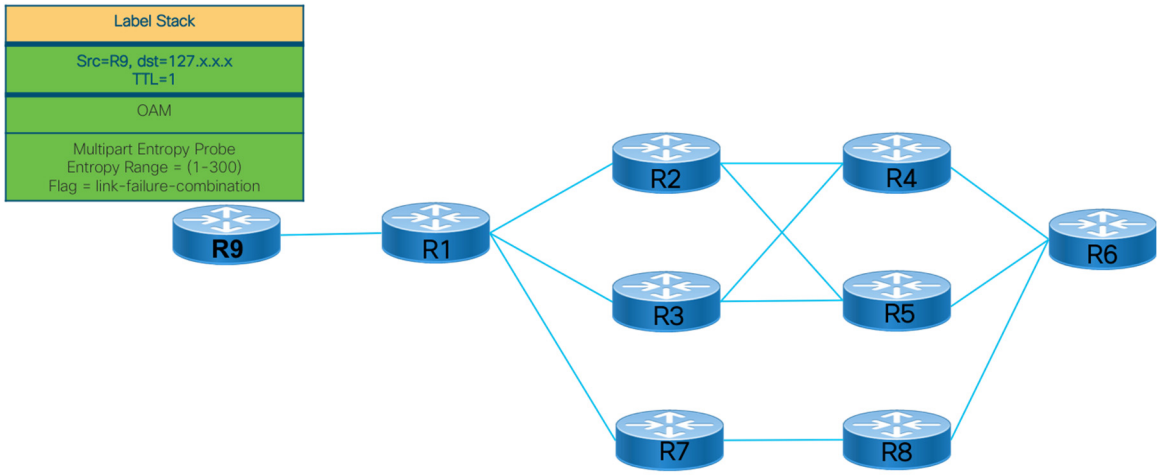

Figure 1

As shown in Figure 1, node R9 performs discovery on all ECMP paths towards node R6 using an existing mechanism (such as LSP Ping) to query the entropy along the path. As an example, node R1 may reply back with the following information:

Entropy (1 - 100) --> Link 1 (towards R2)

Entropy (101 - 200) --> Link 2 (towards R3)

Entropy (201 - 300) --> Link 3 (towards R7)

The reply is based on the current entropy-bucket mapping on node R1. Subsequent entropy queries would be performed based on these ranges, For example, when node R9 queries node R2, node R9 sends the range as (1-100). In most of the network, it is not very common to see rapid changes in the core network. In particular, once the core network is designed, addition, or decommission of any link is uncommon. However, it may be common to see link flapping and links coming back up. For example, if one of the egress link on R1 goes down, the entropy-bucket mapping changes, causing the probes with specific entropy ranges (based on the previous query) to not validate the correct paths. Conventional solutions may query the network again for entropy ranges, which introduces delays, imposes control plane loads, and requires the control plane to intervene on multiple transit nodes.

In contrast, the embodiments presented herein utilize a simple extension that enables an initiator (i.e., a head end node or a centralized server) to query not only the entropy range based on the current entry in the routing information base (RIB), but also based on additional link failure combinations. The response from the transit node is cached and used when the link is down (as identified by the interior gateway protocol (IGP) event). In particular, additional details are queried in the multipart-multipath entropy query (e.g., using LSP ping or any other OAM extensions such as YANG). Upon receiving such a query, a transit node will reply back with entropy ranges for different combinations of link failure. The response is cached by the initiator and used accordingly based on the failure detection.

Any initiator will query the transit nodes with ECMP paths to send the entropy range for each ECMP egress path and also include additional flags to signal the intent to collect the entropy range for different link failures combinations. As depicted in Figure 1, when node R9 generates the query, it includes the entropy range as (1-300) for the destination based on the current RIB entry. In addition, node R9 may also include a flag or a type, length, and value (TLV) to query the transit node with an entropy range for different link failure combinations. In one example, node R9 may simply query all possible combinations. In another example, node R9 can query based on a behavior learning process that identifies particular links that appear to be flapping frequently.

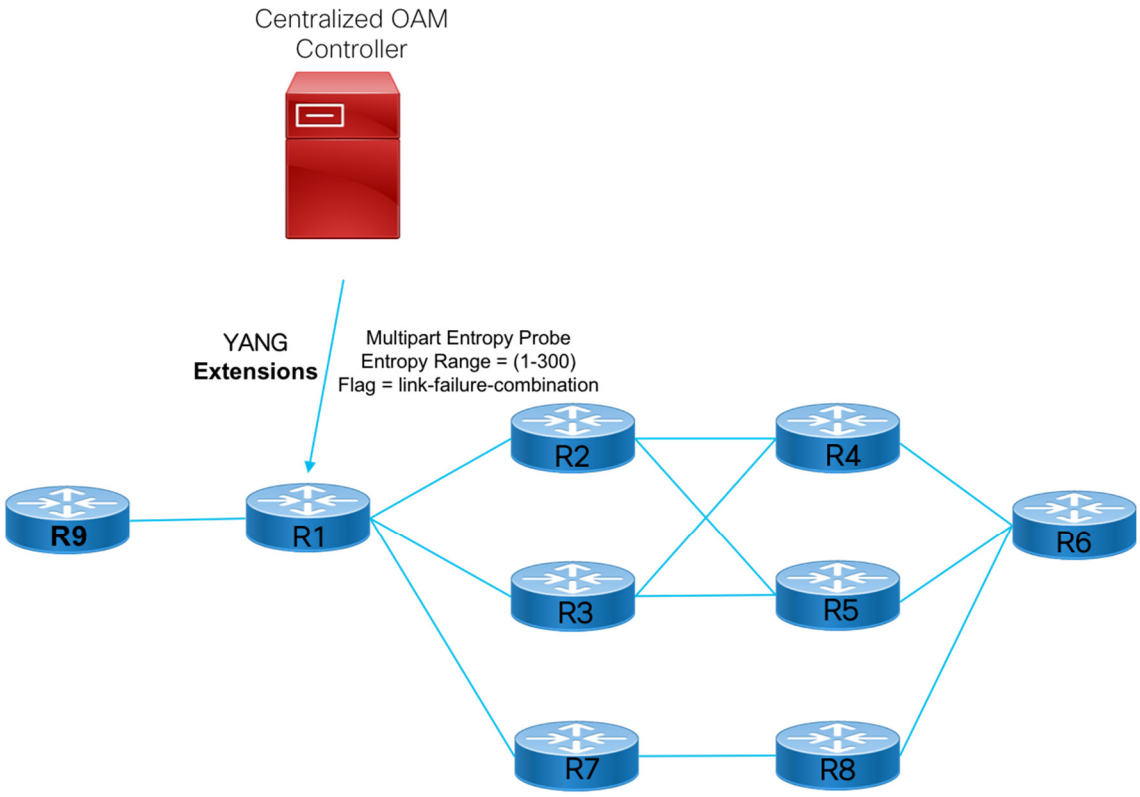

Figure 2

As depicted in Figure 2, a query can be made by a centralized OAM controller/server. The centralized OAM controller/server may include OAM extensions or utilize YANG-formatted requests to query the entropy information.

Any transit node will respond back with the entropy range for different combinations of egress links.

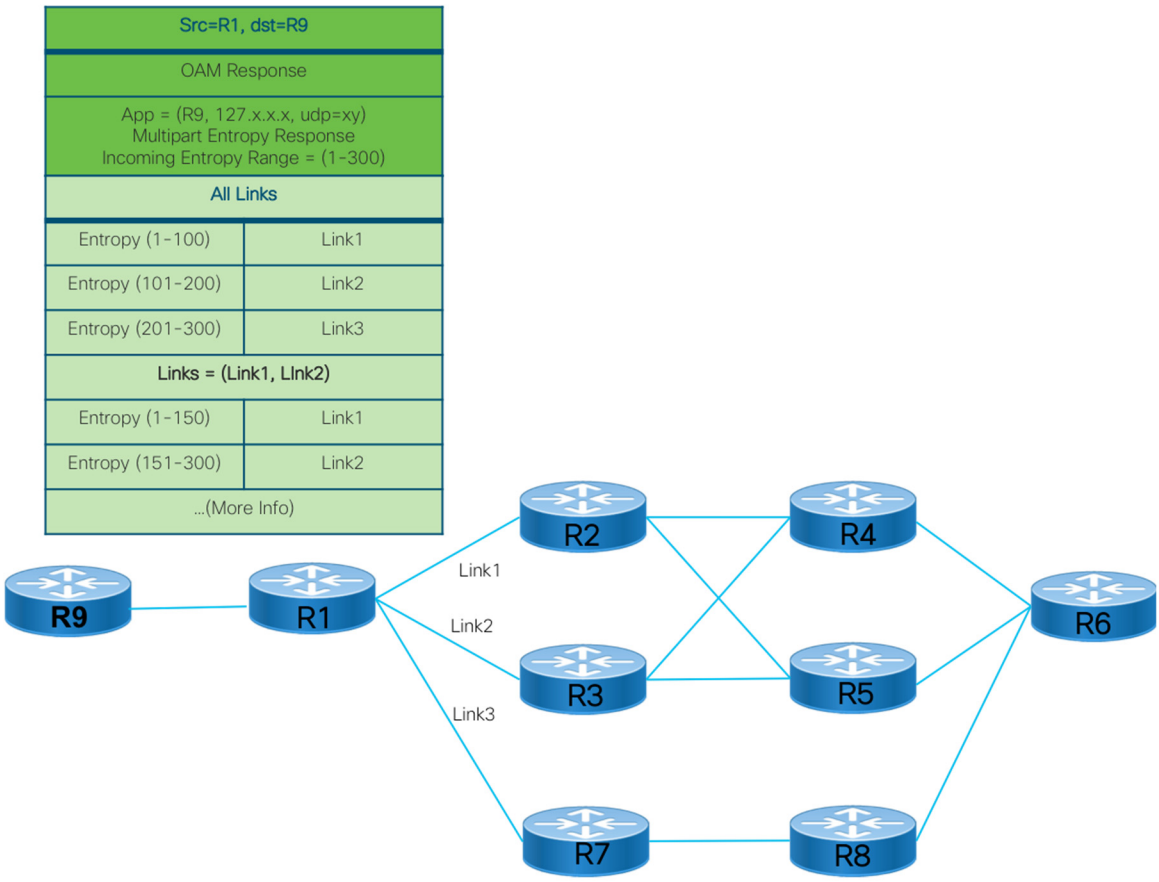

Figure 3

As depicted in Figure 3, node R1 will reply back with the entropy range for each link combination: link 1 (R1-R2), link 2 (R1-R3), and link 3 (R1-R7). Subsequent queries can be made based on the resulting entropy ranges. The initiator will cache the entropy ranges for different link combinations and use the relevant entropy ranges whenever there is a link failure, thus avoiding the need to query all the transit nodes. For example, if link 1 fails on node R1, then node R9 (or any other initiator) will receive IGP failure event about link1. This event can be used as a trigger to utilize the entropy range that is cached for a link 1 failure.

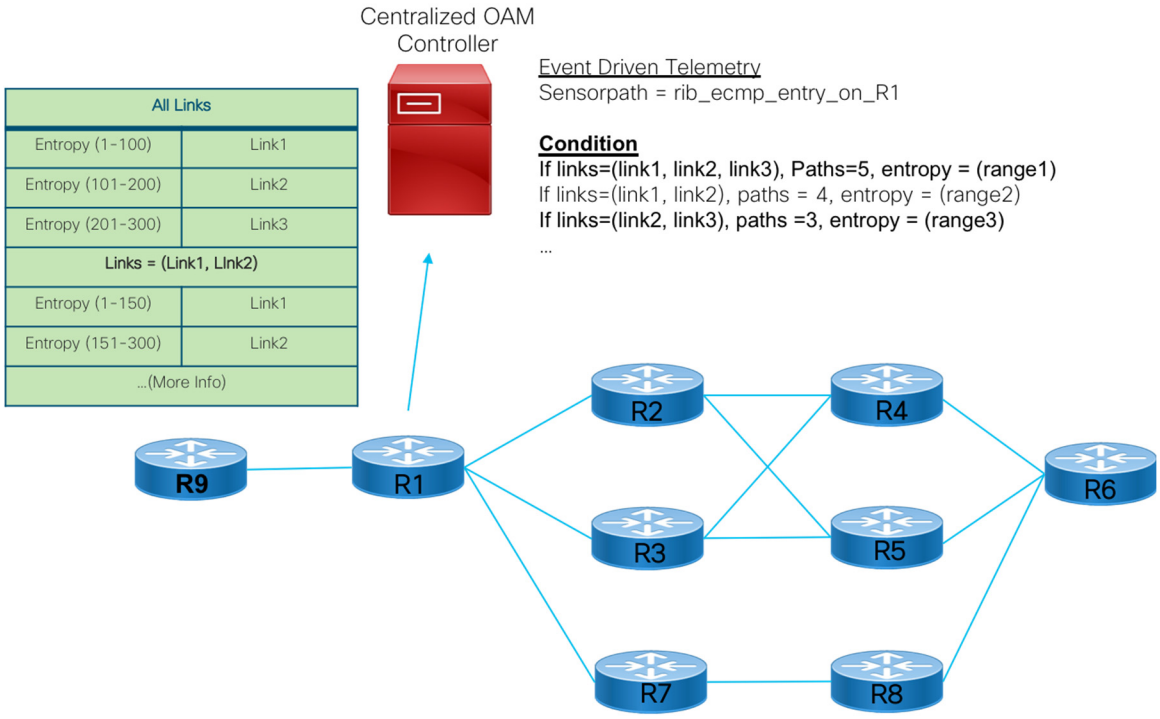

Figure 4

Figure 4 depicts an example using a centralized OAM (cOAM) controller. Upon receiving the multipart entropy response, the controller subscribes to the relevant sensor paths on node R1 and cache the following information for OAM (or other) purposes:

Under normal condition, Paths = 5, and entropy = (range1)

If link 1 fails, Paths = 3, and entropy = (range2)

If link 2 fails, Paths = 3, and entropy = (range3)

If link 3 fails, Paths = 4, and entropy = (range4)

If there is any link failure on R1 or beyond, there will be a change in the sensor path telemetry data (e.g., an egress link change on R1) that will be used as a trigger by the controller/server to use the relevant entropy range and path count. Thus, presented embodiments avoid needing to query the network on every occurrence of a change caused by link failure (as opposed, e.g., to a design change).

In summary, presented herein is a solution that introduces the concept of "dynamic entropy ranges", which are runtime operational state pre-use cached results of entropy calculations based on an actual state of links. In the simplest case, this is a proactive

entropy calculation based on links up and down states. This may be expanded by introducing a non-binary state for links and including their load. This may be used to provide network optimization and traffic engineering. The capability is provided for querying additional details in the multipart-multipath entropy query (using LSP Ping or any other OAM extension, such as YANG). Any transit node, upon receiving such a query, will reply back with entropy range for different combinations of link failure. The response is cached by the initiator and used accordingly based on the failure detection.