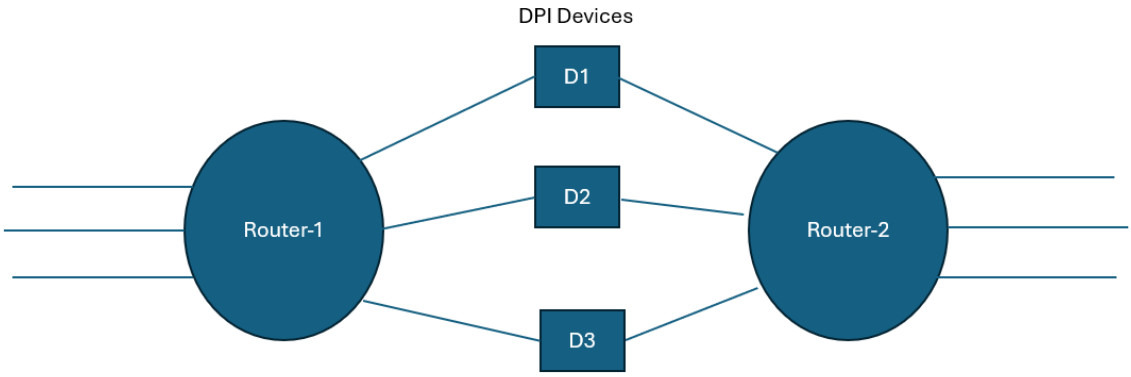
Symmetric Load Balancing

Introduction

This document describes the support for symmetrical load-balancing over an 802.3ad link aggregation group on Trinity-based platforms. This feature has been supported on multiple services routers with certain configuration requirements. The goal of this document is to spell out all these differences and provide the user with everything needed to achieve symmetric load-balancing functionality.

Business Case

Symmetric load-balancing is useful when two routers are connected transparently through deep packet inspection (DPI) devices over a LAG bundle. DPI devices keep track of flows and require information of a given flow in both forward and reverse directions. Without symmetrical load balancing on an 802.3ad LAG, the DPIs could misunderstand the flow, leading to traffic disruptions. This feature ensures that bi-directional traffic for a given flow picks the same link on both routers, hence passing the same DPI device.



Requirements

In order to achieve symmetrical load-balancing, the following needs to be done:

Compute Symmetric Hash

It is imperative that both routers compute the same hash value from the flow in the forward and reverse directions. In legacy platforms, this is achieved by turning on some CLI knobs. The calculated hash value is independent of the direction of the flow and hence is always symmetric in nature. For this reason, no specific configuration is needed to compute symmetric hash value on different platforms. However it should be noted that the fields used to configure the hash should have identical include/exclude settings on both ends of the LAG.

Configure Link Indexes

In order to let both routers pick the same link using the same hash value, the links within the LAG must be configured with the same link index on both routers. This can be achieved with the existing MX CLI ‘set interface <child-ifd> gigether-options 802.3ad link-index <0-15>’.

Enable Symmetric Load Balancing

We need to introduce a new CLI knob to configure symmetric load-balancing under the ‘set forwarding-options enhanced-hash-key’ hierarchy. This knob needs to be disabled by default; otherwise, some customers will experience traffic polarization over LAGs that are deployed across cascaded routers. This solution will prevent the user from achieving symmetric load-balancing; hence we need a new CLI knob to enable symmetric load-balancing. This needs to be configured on both ends of the LAG.

Achieving Symmetry for Bridged and Routed Traffic

In some customer deployments, the LAG bundle on which symmetry is desired is traversed by L2 bridged traffic in the upstream direction and by IPv4 routed traffic in the downstream direction. In such cases, the computed hash would be different in each direction because the Ethernet MAC addresses are also taken into account for bridged packets.

New CLI

The new CLI knob to enable symmetric load-balancing is an extension to the ‘set forwarding-options enhanced-hash-key’ hierarchy. The knob is agnostic to the protocol family and will enable symmetric load-balancing for all AE bundles on the system.

```
Router-1# set forwarding-options enhanced-hash-key symmetric
```

This knob needs to be enabled on both routers at either end of the AE bundle.

The CLI knob to exclude source and destination MAC addresses from enhanced-hash-key computation has the following syntax.

```
Router-1# set forwarding-options enhanced-hash-key family multiservice
no-mac-addresses
```

This knob is disabled by default.

Qualifications

1.1. When symmetric load-balancing is enabled, the user will experience traffic polarization when LAGs are configured on cascaded routers. For example, in figure 2 below, if a certain flow picked link-1 of the AE bundle between R1-R2, it will also pick link-1 of the AE bundle between R2-R3. This is unlike having a random link selection algorithm,

where a flow could pick link-1 in the AE between R1-R2, and link-2 in the AE between R2-R3.

1.2. Symmetric load-balancing is not applicable to per-prefix load-balancing where the hash is computed based on the route prefix.

1.3. Symmetric load-balancing is not applicable to MPLS/VPLS traffic as the labels are not the same in both directions.

Further Technical Details

Currently in Trinity platforms, even though the calculated hash is the same on both routers, the selector tables used for picking a child are shuffled differently. The shuffling is done by the PFE software and it is based on a random number generator with a seed derived from a key containing IFLs. While this guarantees the same seed on different slots of one chassis, this is not true across chassis because the IFL indices would be different. In order to ensure having identical distribution tables on both routers, we need to fix this software seed to the same value. This seed can be fixed to zero when the CLI knob is enabled for symmetric load-balancing. Alternately with the help of another RE CLI, this seed could be user-configurable.

In addition the hardware seed used in PFE should be set to the same value on both routers. This is already set to zero, but it is not configuration driven, meaning it is not changed dynamically.

Another requirement is that the ‘rotate-hash’ parameter for next-hops of type RNH_UNILIST must be turned off if the next-hop has a RNH_AGGREGATE child. Otherwise, both routers would not be using the same hash to pick a member link. When symmetric is enabled, we turn off the rotate-hash feature for all load-balancing next-hops. This is achieved by a new knob in the master record which will skip_rotate_hash even if a JNH_UNILIST next-hop has the rotate-hash enabled.

When the symmetric knob is enabled, we need to reprogram all the AE bundles in the system. This is achieved by traversing all selectors and re-computing the seed based on the current symmetric setting.