CEF Polarization

Document ID: 116376

Contributed by Anvitha Prabhu, Shashank Singh, and Shridhar
Dhodapkar, Cisco TAC Engineers.

Jul 26, 2013

Contents

Introduction

This document describes how Cisco Express Forwarding (CEF) polarization can cause suboptimal use of redundant paths to a destination network. CEF polarization is the effect when a hash algorithm chooses a particular path and the redundant paths remain completely unused.

Prerequisites

Requirements

There are no specific requirements for this document.

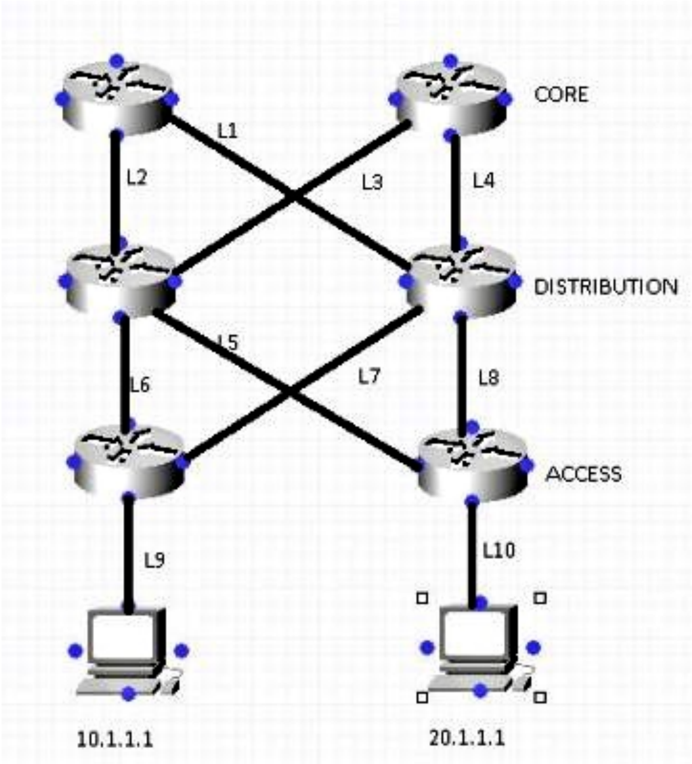
Components Used

The information in this document is based on a Cisco Catalyst 6500 switch that runs on a Supervisor Engine 720.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Background Information

CEF switches the packets based on the routing table that is populated by the routing protocols, such as Enhanced Interior Gateway Routing Protocol (EIGRP) and Open Shortest Path First (OSPF). CEF performs load−balancing once the routing table (RIB) is calculated. In a hierarchical network design, there can be many Layer 3 (L3) equal−cost redundant paths. Consider this topology where traffic flows from the access layer across the distribution and core and into the data center.



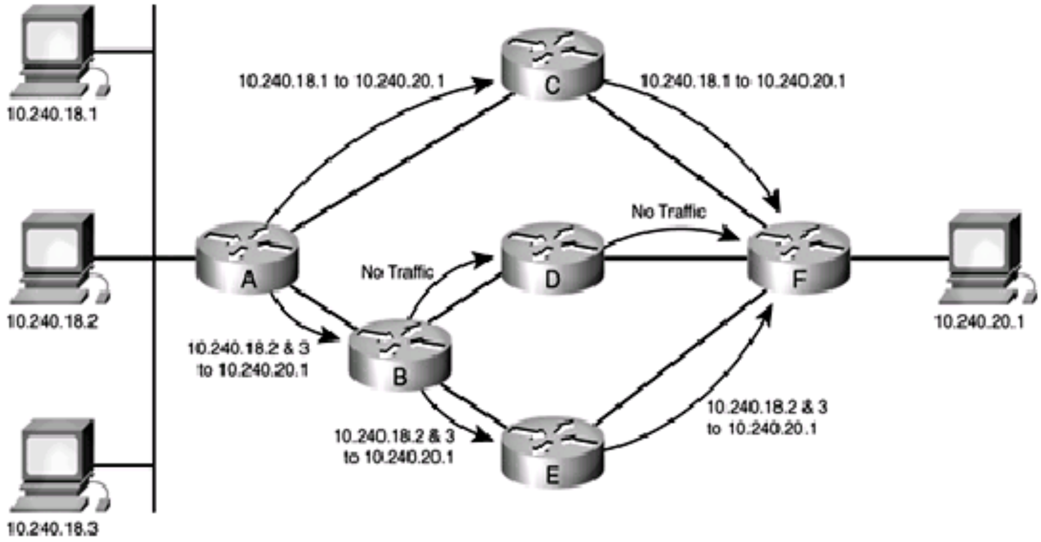
Assume that in order to reach the network 10.1.1.1 from Router 1 (R1) [Top Left], there are two equal−cost paths (L1 , L2). The decision about which of the two links is used is made by a hashing algorithm. By default, the Source IP (SIP) and destination IP (DIP) are used as the parameters in the hashing algorithm.

Here is a description of how the hashing algorithm works:

When there are only two paths, the switch/router performs an exclusive−OR (XOR) operation on the lower−order bits (one bit when either of two links need to be selected, two bits for 3−4 links, and so on) of the SIP and DIP. The XOR operation of the same SIP and DIP always results in the packet use of the same link.

The packet then passes onto the distribution layer, where the same hashing algorithm is used along with the same hash input, and picks a single link for all flows, which leaves the other link underutilized. This process is called CEF polarization (use of the same hash algorithm and same hash input which results in the use of a single Equal−Cost Multi−Path (ECMP) link for ALL flows).

This example illustrates this process in more detail:


10.240.18.1
10.240.18.2
10.240.18.3
10.240.18.1 to 10.240.20.1
10.240.18.1 to 10.240.20.1
No Traffic
No Traffic
10.240.18.2 & 3 to 10.240.20.1
10.240.18.2 & 3 to 10.240.20.1
10.240.18.2 & 3 to 10.240.20.1
A
B
C
D
E
F
10.240.20.1

1. Traffic sourced from 10.240.18.1 and destined to 10.240.20.1 enters the network at Router A and is CEF−switched. Because there are two equal−cost paths to the 10.240.20.0/24 network, the source and destination addresses in the packet go through the hash algorithm, and the result is a specific path used to reach the destination. In this case, the path the packets take is toward Router C. From there, the packets go to Router F, and on to their final destination.

2. Traffic sourced from 10.240.18.2 and destined to 10.240.20.1 enters the network at Router A and is CEF−switched as well. Because there are two equal−cost paths to the 10.240.20.0/24 network, the source and destination addresses in the packet go through the hash algorithm, and CEF chooses a path. In this case, the path the packets take is toward Router B.

3. Traffic sourced from 10.240.18.3 and destined to 10.240.20.1 enters the network at Router A and is also CEF−switched. Because there are two equal−cost paths to the 10.240.20.0/24 network, the source and destination addresses in the packet go through the hash algorithm, and CEF chooses a path. In this case, the path the packets take is toward Router B.

4. The packets sourced from 10.240.18.2 and 10.240.18.3 both arrive at Router B, which again has two equal−cost paths to reach 10.240.20.1. It again runs these sets of source and destination pairs through the hash algorithm, which produces the same results that the hash algorithm on Router A produced. This means that both streams of packets pass along one path − in this case, the link toward Router E. The link toward Router D receives no traffic.

5. After the traffic sourced from 10.240.18.2 and 10.240.18.3 is received on Router E, it is switched along the path to Router F, and then on to its final destination.

How to Avoid CEF Polarization

1. Alternate between ***default*** (SIP and DIP) and ***full*** (SIP + DIP + Layer4 ports) hashing inputs configuration at each layer of the network.

   The Catalyst 6500 provides a few choices for the hashing algorithm:

    - Default − Use the source and destination IP address, with unequal weights given to each link in order to prevent polarization.
    - Simple − Use the source and destination IP address, with equal weight given to each link.
    - Full − Use the source and destination IP address and Layer 4 port number, with unequal weights.
    - Full Simple − Use the source and destination IP address and Layer 4 port number, with equal weights given to each link.

```
6500(config)#mls ip cef load−sharing ?
  full    load balancing algorithm to include L4 ports
  simple  load balancing algorithm recommended for a single−stage CEF router

6500(config)#mls ip cef load−sharing full ?
  simple  load balancing algorithm recommended for a single−stage CEF router
  <cr>
```

Currently, no commands exist to check the load−sharing algorithm in use. The best way to find out which method is in use is to check the current configuration via the ***show running−config*** command. If no configuration is present starting with ***mls ip cef load−sharing***, the default source and destination unequal weight algorithm is in use.

Note: 1) The Catalyst 6500 does not support per packet load−sharing. 2) The ***full*** option does NOT include a universal ID in hash. If it is used at every layer of a multi−layer topology, polarization is possible. It is advisable to use the ***simple*** option with this command in order to achieve better load−sharing and to use fewer hardware adjacencies.

2. Alternate between an even and odd number of ECMP links at each layer of the network.
The CEF load−balancing does not depend on how the protocol routes are inserted in the routing table. Therefore, the OSPF routes exhibit the same behavior as EIGRP. In a hierarchical network where there are several routers that perform load−sharing in a row, they all use same algorithm to load−share.

The hash algorithm load−balances this way by default:

```
1: 1
2: 7−8
3: 1−1−1
4: 1−1−1−2
5: 1−1−1−1−1
6: 1−2−2−2−2−2
7: 1−1−1−1−1−1−1
8: 1−1−1−2−2−2−2−2
```

The number before the colon represents the number of equal−cost paths. The number after the colon represents the proportion of traffic which is forwarded per path.

This means that:

    - For two equal cost paths, load−sharing is 46.666%−53.333%, not 50%−50%.
    - For three equal cost paths, load−sharing is 33.33%−33.33%−33.33% (as expected).
    - For four equal cost paths, load−sharing is 20%−20%−20%−40% and not 25%−25%−25%−25%.

This illustrates that, when there is even number of ECMP links, the traffic is not load−balanced
One way to disable CEF polarization is ***anti−polarization weight***, which was introduced in Version 12.2(17d)SXB2.

In order to enable ***anti−polarization weight***, enter this command :

```
6500(config)# mls ip cef load-sharing full simple
```

Use this command if there are two equal cost paths and both need to be used equally. The addition of the keyword ***simple*** allows the hardware to use the same number of adjacencies as in the Cisco IOS® CEF adjacency. Without the ***simple*** keyword, the hardware installs additional adjacency entries in order to avoid platform polarization.

3. Cisco IOS introduced a concept called ***unique−ID/universal−ID*** which helps avoid CEF polarization. This algorithm, called the universal algorithm (the default in current Cisco IOS versions), adds a 32−bit router−specific value to the hash function (called the universal ID − this is a randomly generated value at the time of the switch boot up that can can be manually controlled). This seeds the hash function on each router with a unique ID, which ensures that the same source/destination pair hash into a different value on different routers along the path. This process provides a better network−wide load−sharing and circumvents the polarization issue. This unique −ID concept does not work for an even number of equal−cost paths due to a hardware limitation, but it works perfectly for an odd number of equal−cost paths. In order to overcome this problem, Cisco IOS adds one link to the hardware adjacency table when there is an even number of equal−cost paths in order to make the system believe that there is an odd number of equal−cost links.
   In order to configure a customized value for the universal ID, use:

   ```
   6500(config)ip cef load−sharing algorithm universal <id>
   ```