By Elvin Arias.

Understanding EIGRP UCMP Traffic Share Ratios

Disclaimer

In this paper, we will explain how to manipulate the **traffic share** ratios, which means this is not an introductory explanation of the protocol. Knowledge about general IP routing, CEF, and EIGRP is highly recommended.

Overview

For many years EIGRP was the only IGP with support for unequal-cost multipath (UCMP). Thanks to the Feasibility Condition, primary and secondary paths are guaranteed to be loop-free. Even though nowadays OSPF, IS-IS and even static routing support unequal-cost multipath, EIGRP still remains as the first IGP in Cisco IOS to support this feature.

This document is divided into three sections the first is a brief explanation about CEF, the second is regarding EIGRP traffic share, and the final section is how to manipulate the traffic share ratios.

Cisco Express Forwarding Summary

Cisco Express Forwarding (CEF) is the switching method of Cisco routers, it controls how traffic is switched. CEF has different databases where the information is stored for traffic forwarding purposes:

1. Forwarding Information Base (FIB). - This database is populated from the Routing Information Base (RIB) or routing table. Once the routing information exist in the routing table, information about prefixes is downloaded to the FIB.

2. Adjacency Information Base (AIB).- This database uses proactive mechanisms in order to resolve adjacencies in order to pre-calculate different elements link:
    a. Next hop.
    b. Exit interface.
    c. Encapsulation

   NOTE: CEF performs load-sharing and not load-balancing.

When performing verifications in the CLI, we can use the following order:

1. Verify the RIB.

```
R1#show ip route 2.2.2.2
Routing entry for 2.2.2.2/32
  Known via "eigrp 1", distance 90, metric 10880, type internal
  Redistributing via eigrp 1
  Last update from 12.0.0.2 on GigabitEthernet1.12, 00:13:15 ago
```

```
  Routing Descriptor Blocks:
  * 12.0.0.2, from 12.0.0.2, 00:13:15 ago, via GigabitEthernet1.12
      Route metric is 10880, traffic share count is 1
      Total delay is 11 microseconds, minimum bandwidth is 1000000
Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

NOTE: A key field here is the “traffic share count is <VALUE>”, which denotes the amount of weight the interface has for switching outgoing traffic for this prefix.

2. Verify the FIB.

```
R1#show ip cef 2.2.2.2/32 internal
2.2.2.2/32, epoch 2, RIB[I], refcnt 6, per-destination sharing
  sources: RIB
  feature space:
    IPRM: 0x00028000
    Broker: linked, distributed at 4th priority
  ifnums:
    GigabitEthernet1.12(12): 12.0.0.2
  path list 7FD930491BC0, 3 locks, per-destination, flags 0x49
[shble, rif, hwcn]
    path 7FD937666B88, share 1/1, type attached nexthop, for IPv4
      nexthop 12.0.0.2 GigabitEthernet1.12, IP adj out of
GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
  output chain:
    IP adj out of GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
```

NOTE: When multiple paths for the same destination prefix exist in the routing table (equally or unequally), CEF utilizes the so-called **hash buckets**. The hash buckets are used as an internal mechanism of traffic distribution among interfaces, a maximum of 16 (0-15) hash buckets exist in any given router.

3. Verify AIB.

```
R1#show adjacency gigabitEthernet 1.12 detail
Protocol Interface                 Address
IP       GigabitEthernet1.12       12.0.0.2(12)
                                   3 packets, 174 bytes
                                   epoch 0
                                   sourced in sev-epoch 6
                                   Encap length 18
```

By Elvin Arias.

```
                                   000C2982079E000C297D2D768100000C
                                   0800
                                   L2 destination address byte offset
0
                                   L2 destination address byte length
6
                                   Link-type after encap: dot1Q
                                   ARP
```

A key part of the adjacency information is the encapsulation described in the output. The first 6 bytes of the hexadecimal string is the destination MAC address of the neighbor, the next six bytes is the source MAC address, and the rest is the encapsulation EtherType. See below:

Dst. MAC: 000C2982079E
Src. MAC: 000C297D2D76
Dot1q: 8100000C
IPv4: 0800

EIGRP and Traffic Share
By default EIGRP can only install equal-cost multipaths (ECMP) into the routing information base (RIB), depending of the IOS version, a maximum number of ECMP can be installed (in newer IOS versions the maximum is 32 ECMP). CEF allows the installation of unequal-cost paths into the RIB/CEF; EIGRP UCMP is deactivated, since the variance multiplier is set to 1 by default.

When performing ECMP, the **traffic share count** result is always set to 1 among all the paths in the routing table, this means that the weight of each path is the same, which in turn means that all the path are load-shared equally. Obviously, when the result of this number has discrepancies among the paths, it means that the paths are not load-shared equally.

Refer below,

```
R1#show ip route 2.2.2.2
Routing entry for 2.2.2.2/32
  Known via "eigrp 1", distance 90, metric 10880, type internal
  Redistributing via eigrp 1
  Last update from 21.0.0.2 on GigabitEthernet1.21, 00:00:04 ago
  Routing Descriptor Blocks:
    21.0.0.2, from 21.0.0.2, 00:00:04 ago, via GigabitEthernet1.21
      Route metric is 10880, traffic share count is 1
      Total delay is 11 microseconds, minimum bandwidth is 1000000
Kbit
      Reliability 255/255, minimum MTU 1500 bytes
```

```
      Loading 1/255, Hops 1
  * 12.0.0.2, from 12.0.0.2, 00:00:04 ago, via GigabitEthernet1.12
      Route metric is 10880, traffic share count is 1
      Total delay is 11 microseconds, minimum bandwidth is 1000000
Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

In this case, metrics are the same, which means that the paths are equally load-shared.

NOTE: As a general principle, paths installed in the RIB with the same metric always result in a traffic share count of 1.

EIGRP: Traffic-share balanced vs. traffic share min across-interfaces
By default, the traffic share manipulation of EIGRP is set to balanced. This can be check with the following command:

```
R1#show running-config all | include traffic-share
   traffic-share balanced
```

NOTE: The show running-config all command shows all the current configuration plus it's default parameters.

The traffic-share balanced allows EIGRP to signal and allow CEF to share the traffic inversely proportional to the metric of the route. This means that by default CEF will be allowed to use metrics that are not necessarily the current best and assign a **traffic share count** that correspond to the metrics EIGRP has installed.

To demonstrate this concept, the following topology is going to be used throughout the rest of this paper.

R1 (G1.12: 12.0.0.1/24)-----(G1.12: 12.0.0.2/24) R2 -----LO2: 2.2.2.2/32
(G1.21: 21.0.0.1/24)-----(G1.21: 12.0.0.2/24)

As seen, R1 as two links connected to R2. R2 is advertising the prefix 2.2.2.2/32.

Default configurations,

R1

```
!
interface GigabitEthernet1
 no ip address
```

```
 negotiation auto
!
interface GigabitEthernet1.12
 description BEST PATH
 encapsulation dot1Q 12
 ip address 12.0.0.1 255.255.255.0
!
interface GigabitEthernet1.21
 description WORST PATH
 encapsulation dot1Q 21
 ip address 21.0.0.1 255.255.255.0
 delay 10
!
router eigrp X
 !
 address-family ipv4 unicast autonomous-system 1
  !
  topology base
  exit-af-topology
  network 12.0.0.0
  network 21.0.0.0
 exit-address-family
!
end
```

R2

```
!
interface Loopback2
 ip address 2.2.2.2 255.255.255.255
!
interface GigabitEthernet1
 no ip address
 negotiation auto
!
interface GigabitEthernet1.12
 description TO R1
 encapsulation dot1Q 12
 ip address 12.0.0.2 255.255.255.0
!
interface GigabitEthernet1.21
 description TO R1
 encapsulation dot1Q 21
 ip address 21.0.0.2 255.255.255.0
!
router eigrp X
```

```
 !
 address-family ipv4 unicast autonomous-system 1
  !
  topology base
  exit-af-topology
  network 2.0.0.0
  network 12.0.0.0
  network 21.0.0.0
 exit-address-family
!
end
```

Verifying the RIB shows us that only one path is currently installed.

```
R1#show ip route 2.2.2.2
Routing entry for 2.2.2.2/32
  Known via "eigrp 1", distance 90, metric 10880, type internal
  Redistributing via eigrp 1
  Last update from 12.0.0.2 on GigabitEthernet1.12, 00:10:48 ago
  Routing Descriptor Blocks:
  * 12.0.0.2, from 12.0.0.2, 00:10:48 ago, via GigabitEthernet1.12
      Route metric is 10880, traffic share count is 1
      Total delay is 11 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

If we verify the EIGRP topology table, we will see the second path only of it's a feasible successor.

```
R1#show ip eigrp topology 2.2.2.2/32
EIGRP-IPv4 VR(X) Topology Entry for AS(1)/ID(12.0.0.1) for 2.2.2.2/32
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 1392640, RIB
is 10880
  Descriptor Blocks:
  12.0.0.2 (GigabitEthernet1.12), from 12.0.0.2, Send flag is 0x0
      Composite metric is (1392640/163840), route is Internal
      Vector metric:
        Minimum bandwidth is 1000000 Kbit
        Total delay is 11250000 picoseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
        Originating router is 12.0.0.2
  21.0.0.2 (GigabitEthernet1.21), from 21.0.0.2, Send flag is 0x0
      Composite metric is (7290880/163840), route is Internal
```

```
      Vector metric:
        Minimum bandwidth is 1000000 Kbit
        Total delay is 101250000 picoseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
        Originating router is 12.0.0.2
```

As seen, the path through 21.0.0.2 is worse due to the delay is higher. If we would want to proceed and install this path in the RIB as an UCMP, we will need to use the variance. The variance calculation is as follows:

Worst Path / Best Path = Variance. Which in this case would be: 7290880/1392640 = 5.23. The variance is effectively 6.

```
R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router eigrp X
R1(config-router)#address-family ipv4 unicast autonomous-system 1
R1(config-router-af)#topology base
R1(config-router-af-topology)#variance 6
```

NOTE: The variance value does not have any impact in the traffic share count result. A variance of 6 or a variance of 128 (maximum) would lead to the same traffic share ratios. Instead, metric values are the ones who influence the resultant traffic share count.

```
R1#show ip route 2.2.2.2
Routing entry for 2.2.2.2/32
  Known via "eigrp 1", distance 90, metric 10880, type internal
  Redistributing via eigrp 1
  Last update from 21.0.0.2 on GigabitEthernet1.21, 00:00:03 ago
  Routing Descriptor Blocks:
    21.0.0.2, from 21.0.0.2, 00:00:03 ago, via GigabitEthernet1.21
      Route metric is 56960, traffic share count is 23
      Total delay is 101 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
  * 12.0.0.2, from 12.0.0.2, 00:00:03 ago, via GigabitEthernet1.12
      Route metric is 10880, traffic share count is 120
      Total delay is 11 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

As seen, now the path is installed as UCMP, and the ratios (weight) are 120 for the primary and 23 for the secondary. If we check the FIB we will see how this traffic is being balanced.

```
R1#show ip cef 2.2.2.2/32 internal
2.2.2.2/32, epoch 2, RIB[I], refcnt 6, per-destination sharing
  sources: RIB
  feature space:
    IPRM: 0x00028000
    Broker: linked, distributed at 4th priority
  ifnums:
    GigabitEthernet1.12(12): 12.0.0.2
    GigabitEthernet1.21(14): 21.0.0.2
  path list 7FD930491940, 3 locks, per-destination, flags 0x49 [shble, rif,
hwcn]
    path 7FD93ECAD0B0, share 23/23, type attached nexthop, for IPv4
      nexthop 21.0.0.2 GigabitEthernet1.21, IP adj out of GigabitEthernet1.21,
addr 21.0.0.2 7FD93E97DF50
    path 7FD93ECAD008, share 120/120, type attached nexthop, for IPv4
      nexthop 12.0.0.2 GigabitEthernet1.12, IP adj out of GigabitEthernet1.12,
addr 12.0.0.2 7FD93E97E310
  output chain:
    loadinfo 7FD936F9DF48, per-session, 2 choices, flags 0003, 5 locks
      flags [Per-session, for-rx-IPv4]
      16 hash buckets
        < 0 > IP adj out of GigabitEthernet1.21, addr 21.0.0.2 7FD93E97DF50
        < 1 > IP adj out of GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
        < 2 > IP adj out of GigabitEthernet1.21, addr 21.0.0.2 7FD93E97DF50
        < 3 > IP adj out of GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
        < 4 > IP adj out of GigabitEthernet1.21, addr 21.0.0.2 7FD93E97DF50
        < 5 > IP adj out of GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
        < 6 > IP adj out of GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
        < 7 > IP adj out of GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
        < 8 > IP adj out of GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
        < 9 > IP adj out of GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
        <10 > IP adj out of GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
        <11 > IP adj out of GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
        <12 > IP adj out of GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
        <13 > IP adj out of GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
        <14 > IP adj out of GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
        <15 > IP adj out of GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
      Subblocks:
        None
```

NOTE: As mentioned earlier, these are the 16 hash buckets (0-15).

The next question would be: What impact does the **traffic-share min across-interfaces** has in this scenario? This command, which is not enabled by default and it’s use is in order to shared traffic among minimum metric paths. This means that the path through 21.0.0.2 is not going to be used as long as the path with the best metric is used, but nonetheless it's going to be installed in the RIB. Let's check,

Performing the change.

```
R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router eigrp X
R1(config-router)#address-family ipv4 unicast autonomous-system 1
R1(config-router-af)#topology base
R1(config-router-af-topology)#traffic-share min across-interfaces
```

Verifying the RIB.

```
R1#show ip route 2.2.2.2
Routing entry for 2.2.2.2/32
  Known via "eigrp 1", distance 90, metric 10880, type internal
  Redistributing via eigrp 1
  Last update from 21.0.0.2 on GigabitEthernet1.21, 00:00:21 ago
  Routing Descriptor Blocks:
    21.0.0.2, from 21.0.0.2, 00:00:21 ago, via GigabitEthernet1.21
      Route metric is 56960, traffic share count is 0
      Total delay is 101 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
  * 12.0.0.2, from 12.0.0.2, 00:00:21 ago, via GigabitEthernet1.12
      Route metric is 10880, traffic share count is 1
      Total delay is 11 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

As seen, the worse path is installed in the RIB, but the **traffic share count is 0**, which means that CEF will not utilize this path.

```
R1#show ip cef 2.2.2.2/32 internal
2.2.2.2/32, epoch 2, RIB[I], refcnt 6, per-destination sharing
  sources: RIB
  feature space:
    IPRM: 0x00028000
    Broker: linked, distributed at 4th priority
  ifnums:
    GigabitEthernet1.12(12): 12.0.0.2
  path list 7FD930491940, 3 locks, per-destination, flags 0x49 [shble, rif,
hwcn]
    path 7FD93ECAD008, share 1/1, type attached nexthop, for IPv4
```

```
      nexthop 12.0.0.2 GigabitEthernet1.12, IP adj out of GigabitEthernet1.12,
addr 12.0.0.2 7FD93E97E310
  output chain:
    IP adj out of GigabitEthernet1.12, addr 12.0.0.2 7FD93E97E310
```

As seen, the worse path is not seen in the FIB.

EIGRP Traffic Share Default Results

As seen in the earlier examples, traffic share values are somehow changed when performing UCMP in EIGRP. How the router came up with the values 120 and 23 for the ratios of each path respectively? Understanding how does the router perform the traffic share count computations is key in order to understand how the ratios work.

As mentioned earlier, CEF has the concept of hash buckets, these are 0 through 15, which makes in total 16 hash buckets. The maximum value that the traffic share count can set is 240, now, the next question would be: What is the relation between the hash buckets and the traffic share ratios?

Starting from the left on the most significant bit to the right to the least significant bits we have the following mathematics:

128 64 32 16 8 4 2 1
  1   1   1   1

A maximum of 16 hash buckets which lead us to a the sum of all bits from 128 through 16, which is 240. :-)

Now, taking example 1 as a reference, where traffic share count was as follows,

```
R1#show ip route 2.2.2.2
Routing entry for 2.2.2.2/32
  Known via "eigrp 1", distance 90, metric 10880, type internal
  Redistributing via eigrp 1
  Last update from 21.0.0.2 on GigabitEthernet1.21, 00:00:03 ago
  Routing Descriptor Blocks:
    21.0.0.2, from 21.0.0.2, 00:00:03 ago, via GigabitEthernet1.21
      Route metric is 56960, traffic share count is 23
      Total delay is 101 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
  * 12.0.0.2, from 12.0.0.2, 00:00:03 ago, via GigabitEthernet1.12
      Route metric is 10880, traffic share count is 120
      Total delay is 11 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

We will do the following assumptions:

1. Since the traffic share count (weight) value is always greater for the best metric, the best metric is always 240 (240 is the maximum integer value for the traffic share count).
2. We will only need to compute the worse metric in order to result in the worse metric ratio.

Performing the calculations

a) Take the route metrics:
    i) Best metric = 10880.
    ii) Worst metric = 56960.
b) The best metric is always set to 240.
c) The worst metric is calculated as follows:
    i) (Best metric/Worst metric) * 240.
    ii) (10880/56960) * 240 = 46 (round to the nearest integer).
d) Now we are left with two results:
    i) 240 of the best metric.
    ii) 46 of the worse metric.
e) We compute the greatest common denominator (GCD) of these two:
    i) G.C.D of 240, 46 = 2.
f) Now we divide the best and worse metrics respectively by the GCD result of 2.
    i) 240/2 = 120.
    ii) 46/2 = 23.

As seen, the results of 120 and 23 are exactly the traffic share count ratios computed by the router as seen in the show ip route output (and earlier).

```
R1#show ip route 2.2.2.2

--- output truncated ---

    21.0.0.2, from 21.0.0.2, 00:00:03 ago, via GigabitEthernet1.21
      Route metric is 56960, traffic share count is 23
  * 12.0.0.2, from 12.0.0.2, 00:00:03 ago, via GigabitEthernet1.12
      Route metric is 10880, traffic share count is 120
```

This simple computation is key in order to understand how these ratios work and how the router internally perform the calculations. Before, we could have think that this computation was kind of “esoteric” sometimes. :-)

NOTE: As seen in this example, we used the scaled metric, but we would have used the metric without scaling it down, it would have lead to the same results (maths :-)).

By Elvin Arias.

EIGRP Traffic Share Manipulations

Now that we now the principles of CEF, hash buckets, traffic-share [balanced vs. min across-interfaces], what is the traffic share, and how the computation of the ratios is performed, we will follow with the traffic share manipulations in EIGRP.

Before continuing, to simplify the scenario the same topology still remain.

R1 (G1.12: 12.0.0.1)-----(G1.12: 12.0.0.2/24) R2 -----LO2: 2.2.2.2/32
(G1.21: 21.0.0.1)-----(G1.21: 12.0.0.2/24)

Default configurations.

R1

```
!
interface GigabitEthernet1
 no ip address
 negotiation auto
!
interface GigabitEthernet1.12
 encapsulation dot1Q 12
 ip address 12.0.0.1 255.255.255.0
!
interface GigabitEthernet1.21
 encapsulation dot1Q 21
 ip address 21.0.0.1 255.255.255.0
!
router eigrp 1
 !
  network 12.0.0.0
  network 21.0.0.0
!
end
```

R2

```
!
interface Loopback2
 ip address 2.2.2.2 255.255.255.255
!
interface GigabitEthernet1
 no ip address
 negotiation auto
!
```

By Elvin Arias.

```
interface GigabitEthernet1.12
 description TO R1
 encapsulation dot1Q 12
 ip address 12.0.0.2 255.255.255.0
!
interface GigabitEthernet1.21
 description TO R1
 encapsulation dot1Q 21
 ip address 21.0.0.2 255.255.255.0
!
router eigrp 1
 !
  network 2.0.0.0
  network 12.0.0.0
  network 21.0.0.0
!
end
```

There are many ways to manipulate the traffic share count when performing UCMP in EIGRP, here are a few:

1) Changing the DLY of the outgoing interface.
2) Changing the minBW
3) Using offset-lists in order to manipulate the metrics.
4) Using the EIGRP formula for the vector metrics computation.

For this example, I will use method number 4, which is a better and more consistent method among all. Recalling that EIGRP metric formula (simplified) is:

EIGRP Metric Formula
[(10^7/minBW) + (DLY in Tens of Microeconds/10)] * 256

EIGRP Metric Formula for the Ratios
[(10^7/minBW) + (DLY in Tens of Microeconds/10)] * 256 = FD * RATIO

NOTE: If only DLY would have been enabled, then only the DLY part would have been taken into account for this computation.

Let’s say we want a ratio of 5:1.

Let’s verify the topology table and see what are the parameters for these paths,

```
R1#show ip eigrp topology 2.2.2.2/32
EIGRP-IPv4 VR(X) Topology Entry for AS(1)/ID(12.0.0.1) for 2.2.2.2/32
  State is Passive, Query origin flag is 1, 2 Successor(s), FD is 130816
  Descriptor Blocks:
  12.0.0.2 (GigabitEthernet1.12), from 12.0.0.2, Send flag is 0x0
      Composite metric is (130816/128256), route is Internal
      Vector metric:
        Minimum bandwidth is 1000000 Kbit
        Total delay is 5010 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
        Originating router is 12.0.0.2
  21.0.0.2 (GigabitEthernet1.21), from 21.0.0.2, Send flag is 0x0
      Composite metric is (130816/128256), route is Internal
      Vector metric:
        Minimum bandwidth is 1000000 Kbit
        Total delay is 5010 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
        Originating router is 12.0.0.2
```

Let’s verify the routing table and see what are the parameters for these paths,

```
R1#show ip route 2.2.2.2
Routing entry for 2.2.2.2/32
  Known via "eigrp 1", distance 90, metric 130816, type internal
  Redistributing via eigrp 1
  Last update from 21.0.0.2 on GigabitEthernet1.21, 00:00:35 ago
  Routing Descriptor Blocks:
    21.0.0.2, from 21.0.0.2, 00:00:35 ago, via GigabitEthernet1.21
      Route metric is 130816, traffic share count is 1
      Total delay is 5010 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
  * 12.0.0.2, from 12.0.0.2, 00:00:35 ago, via GigabitEthernet1.12
      Route metric is 130816, traffic share count is 1
      Total delay is 5010 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

As seen, traffic is being load-shared equally between the two exits.

Performing the Ratio Manipulation

By Elvin Arias.

[(10^7/minBW) + (DLY/10)] * 256 = FD * Ratio

NOTE: The minBW variable is always the minBW of the feasible successor.
NOTE: Not all ratios are mathematically possible.

[(10^7/1000000) + (DLY/10)] * 256 = 130816 * 5/1

[10 + (DLY/10] * 256 = 654080

(100 + DLY)/10 = 654080/256

(100 + DLY)/10 = 2555

(100 + DLY)/10 = 2555

100 + DLY = 25550

DLY = 25550-100

DLY = 25450

Now, we will need to proceed and subtract the advertised DLY of the path from R2, which in this case is 5000 tens of microseconds,

```
R2#show interfaces loopback 2 | include DLY
  MTU 1514 bytes, BW 8000000 Kbit/sec, DLY 5000 usec,
```

NOTE: In case it would have been a chain of routers, the same would have been applied, the sum of the outgoing delays minus the delay calculated in the formula.

DLY = 25450 - 5000

DLY = 20450

And since the DLY is in calculated in tens of microseconds, we will need to divide the DLY by 10 before setting it into the configuration.

DLY = 20450/10 = 2045.

This is the DLY value we will need to assign to the secondary interface of g1.21

Changing the DLY in GigabitEthernet1.21.

By Elvin Arias.

```
R1(config)#interface gigabitethernet1.21
R1(config-subif)#delay 2045
```

Verifying the RIB,

```
R1(config-subif)#do sh ip route 2.2.2.2
Routing entry for 2.2.2.2/32
  Known via "eigrp 1", distance 90, metric 130816, type internal
  Redistributing via eigrp 1
  Last update from 21.0.0.2 on GigabitEthernet1.21, 00:00:01 ago
  Routing Descriptor Blocks:
    21.0.0.2, from 21.0.0.2, 00:00:01 ago, via GigabitEthernet1.21
      Route metric is 654080, traffic share count is 1
      Total delay is 25450 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
  * 12.0.0.2, from 12.0.0.2, 00:00:01 ago, via GigabitEthernet1.12
      Route metric is 130816, traffic share count is 5
      Total delay is 5010 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

How much bigger is the metric and how it relates to the ratios has mentioned before?

654080/130816 = 5. Metric is 5 times bigger. :-)

Really neat.

Hope that helps you with your studies.