EIGRP – 2

Named Mode EIGRP Configuration, konfigürasyonu için EIGRP prosesi başlatılırken bir proses numarası vermek yerine bir isim tanımı oluşturuluyor. IEGRP protokolü üzerine yapılacak bütün konfigürasyonlar (port arayüzü altında yapılması gerekenler dahi) burada tanımlanıyor. Bu sayede EIGRP protokolüne yönelik bütün konfigürasyonlar bir bütün halde bulunuyor. Named Mode EIGRP konfigürasyonunun sağladığı avantajlara bakıldığında;

- Router üzerindeki bütün EIGRP konfigürasyonu bir bütün halde bulunduruluyor/görüntülenebiliyor.
- EIGRP protokolünde halihazırda olan özellikleri desteklediği gibi yeni nesil özellikleri de destekleniyor (Örnek olarak Metrik hesabıyla ilgili yeni özellikleri kullanmak için Named Mode tercih edilmesi gerekiyor).
- EIGRP protokolünün farklı L3 adresleri desteklediğinden de bahsedilmişti. Farklı L3 adres yapıları (bu adresleri birbirinden izole çalıştırmak için VRF (Virtual Router Forwarding) vs. kullanılıyorsa) kullanılması durumunda EIGRP protokolünün yönetimi konusunda network yöneticisine kolaylık sağlayacaktır.

Named Mode yapısı gereği Address Family, Interface ve Topology olmak üzere üç parçadan oluşuyor. Bu parçalarda hangi kapsamda tanımlar yapıldığına bakıldığında;

- **Address Family**, ilgili router için EIGRP çalışacak portlarının seçilmesi, K sabitinin değerinin belirlenmesi, loglama ayarları, stub ayarlamaları gibi bulunduğu AS üzerindeki genel EIGRP tanımlarının (özetle Classic Mode kullanılırken EIGRP prosesi altında yapılan tanımlar) yapıldığı kısımdır.
- **Interface**, router portları altında uygulanması gereken EIGRP tanımlarının (Hello Interval, Split-Horizon, Authentication, Summary route gibi) yapıldığı kısımdır. Burada router üzerindeki her port için ayrı ayrı section tanımı bulunmaktadır.
- **Topolojy**, Administrative Distace, Route Redistribution gibi daha çok EIGRP toploloji tablosuna yönelik tanımların yapıldığı kısımdır.

Named Mode EIGRP konfigürasyonu için;

- İlk olarak “**router eigrp <Proccess Name>**” komutuyla bir EIGRP prosesinin başlatılması gerekiyor (Eski routerlar Named Mode desteklemeyebiliyor).
- Oluşturulan proses tanımı altında “**address-family {IPv4 | IPv6} {unicast | vrf <VRF NAME>} autonomous-system <AS Number>**” komutuyla hangi AS’e dair hangi ip adres türüyle ve hangi yayınla network adreslerinin anons edileceği belirtiliyor.

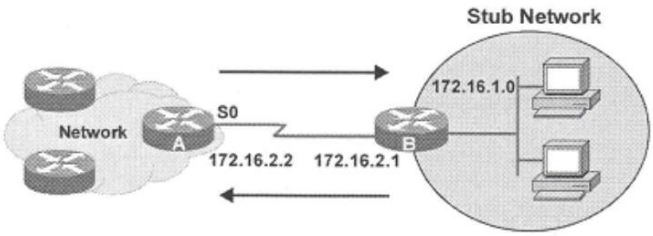
- Address-Family tanımı altında “**network <Ip Address> <Wildcard Mask>**” komutuyla anons edilecek network adresleri ve dolayısıyla EIGRP protokolünün çalışacağı router portları belirtilmelidir. Classic Mode yönteminde olduğu gibi burada da network adresleri 4 farklı şekilde tanımlanabiliyor.
- İsteğe bağlı olarak 32 bitlik Router-ID değerini manuel olarak belirlemek için “**eigrp router-id <Router ID>**” komutu kullanılmaktadır. Bu değer manuel olarak tanımlanmadığı takdirde;
  - Router Id değeri manuel olarak tanımlanmadığı durumlarda routerda tanımlanan en büyük LoopBack ip adresi (birden fazla LoopBack adres tanımlanmış olabilir) Router Id olarak seçilir.
  - Routerda LoopBack interface tanımı yapılmamışsa **routerun aktif arayüzlerinden en büyük ip adresine sahip arayüzün ip adresi Router Id olarak seçilir (ip adresi atanmış ama Shutdown durumdaysa değerlendirilmiyor).**
  - Son durumda routerda aktif arayüz de bulunmuyorsa EIGRP protokolünün başlatılmasına izin verilmiyor.

```
address-family ipv4 unicast autonomous-system 100
 !
 topology base
 exit-af-topology
 network 10.0.0.0 0.0.0.255
 network 20.0.0.0 0.0.0.255
exit-address-family
```

Passive Interface

EIGRP protokolünde “network “komutuyla networkleri öğretmenin iki asli amacı vardı. Bu amaçlardan ilki, ilgili networkün EIGRP protokolüyle anons edilmesini sağlamaktı. İkinci asli amacı ise bu networke bağlı port üzerinde EIGRP protokolünün çalışacağını belirleyerek Anons paketleri göndermeye başlamasıydı. Bu anonslar sayesinde aynı networke dahil ve EIGRP çalışan routerlar ile komşuluk kuruluyordu. Ne yazık ki “network” komutuyla öğretilen her networke router bulunmayabiliyor. Anons edilen network son kullanıcıya hizmet veriyor olabiliyor. Networkten paketlerin gönderileceği tek bir gateway çıkışına sahip networklere **Stub network** deniliyor.

Stub networkler dahi olsa “network” komutuyla anons edilen network adreslerinin çalıştığı portlarından Hello paketleri gönderilmeye devam ediliyor. Bu durum, bir saldırganın bu routera komşuluk kurup EIGRP anons paketleri göndererek routerun oluşturduğu topoloji tablosunu manipüle etme riskini doğuruyor. Her ne kadar kimlik denetimi dahi açılmış olsa da bu portlardan anons paketleri gönderilmesi risk oluşturacaktır.



Stub network gibi EIGRP anons paketlerinin gönderilmesi gerekmeyen portlarda anons paketlerini engellenebiliyor. Anons paketlerinin engellendiği portlara ise **Passive Interface** deniliyor. Passive Interface konfigürasyonu iki farklı şekilde yapılabiliyor;

- İlk seçenek (Classic mode EIGRP konfigürasyonu için) olarak eğer ki Passive Interfac eyapılacak port sayısı az ise her bir port için ilgili EIGRP prosesi altında "**passive-interface <Interface ID>**" komutuyla Anons paketlerinin durdurulacağı portun belirtilmesi yeterlidir.
    - Named Mode konfigürasyonu için "**address-family {IPv4 | IPv6} {unicast | vrf <VRF NAME>} autonomous-system <AS Number>**" komutuyla ilgili ip adres ailesinin altında "**af-interface <Interface ID>**" komutuyla ilgili port altına girildikten sonra "**passive-interface**" komutuyla Passive Interface tanımı uygulanıyor.

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# router eigrp 100
R1(config-router)# passive-interface g10/2

R2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)# router eigrp EIGRP-NAMED
R2(config-router)# address-family ipv4 unicast autonomous-system 100
R2(config-router-af)# af-interface g10/2
R2(config-router-af-interface)# passive-interface
```

- İkinci seçenek olarak Passive Interface olacak portların sayısı fazlaysa ilgili EIGRP prosesi altında "**passive-interface default**" komutuyla bütün portlardan anons paketleri engellendikten sonra anons paketlerinin yapılacağı portlar "**no passive-interface <Interface ID>**" komutuyla tek tek belirtilebilir.
    - Named Mode konfigürasyonu için "**address-family {IPv4 | IPv6} {unicast | vrf <VRF NAME>} autonomous-system <AS Number>**" komutuyla ilgili ip adres ailesinin altında "**af-interface default**" ve "**passive-interface**" komutuyla bütün portlardaki EIGRP anonsu kaldırıldıktan sonra "**af-interface <Interface ID>**" ve "**no passive-interface**" komutlarının anons yapılacak portlarda tanımlanması yeterlidir (Burada kullanılan "**af-interface**" komutu Address Family Interface'in kısaltmasıdır).

```
R1(config)# router eigrp 100
R1(config-router)# passive-interface default
04:22:52.031: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.12.1.2 (GigabitEther-
net0/1) is down: interface passive
R1(config-router)# no passive-interface g10/1
*May 10 04:22:56.179: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.12.1.2 (Giga-
bitEthernet0/1) is up: new adjacency

R2(config)# router eigrp EIGRP-NAMED
R2(config-router)# address-family ipv4 unicast autonomous-system 100
R2(config-router-af)# af-interface default
R2(config-router-af-interface)# passive-interface
04:28:30.366: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.12.1.1
(GigabitEthernet0/1) is down: interface passiveex
R2(config-router-af-interface)# exit-af-interface
R2(config-router-af)# af-interface g10/1
R2(config-router-af-interface)# no passive-interface
R2(config-router-af-interface)# exit-af-interface
*May 10 04:28:40.219: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.12.1.1
(GigabitEthernet0/1) is up: new adjacency
```

NOT: Tüm portlardaki EIGRP anonsunun kapatılıp tek tek açılması, aynı AS içerisindeki bütün routerları etkileyeceğinden önerilmemektedir. Benzer şekilde uzak bağlantı (SSH, HTTPS gibi) ile bağlanılıyorsa ve bağlantı için kullanılan ip adresi EIGRP protokolü üzerinden öğrenilmişse bu durumda routerun tüm portlarındaki anonslar durdurulduğunda routera olan uzak bağlantı kesileceği için yeniden devreye alınamayabilir.

EIGRP Authentication

Passive Interface kısmında da açıklandığı üzere bir saldırgan EIGRP protokolü çalışan routerlardan birisine komşuluk kurup anons paketleri göndererek routerun dahil olduğu AS içerisindeki routerların Topoloji tablolarını manipüle edebilir. Bunun önüne geçmek için routerlar arasında kimlik denetimi devreye alınabiliyor.

EIGRP protokolündeki kontrol mekanizması devreye alınırken routerlar arasında kullanılmak üzere belirli bir Key değeri belirleniyor. Routerlar anons paketlerini şifrelemeden göndermeye devam ediyorlar. **Anons paketleri içerisine ek olarak anons edilecek network bilgisi ile öncesinde belirlenen Key değerini bir Hash algoritmasına (MD5, SHA-256 gibi çeşitli Hash algoritmaları kullanılabiliyor) tabi tuttuktan sonra ortaya çıkan değeri de ekliyor**. Anons paketi hedef routera ulaştığında komşu router kendisinde bulunan Key değeri ile anons paketi içerisindeki network bilgisini ayni Hash algoritmasına tabi tutuyor. Elde ettiği değer ile anons paketi içerisinde gönderilen Hash çıktısını karşılaştırıyor. Eğer ki Hash çıktıları aynı ise bu rota bilgisini topoloji tablosuna ekleyerek değerlendirmeye alıyor. EIGRP Authentication konfigürasyonu için;

- (Classic Mode) İlk olarak Global konfigürasyon modu altında "**key chain <Key Chain Name>**" komutuyla bir anahtar zinciri oluşturuluyor. Burada "**key <Key ID>**" komutuyla bir anahtar numarası belirtiliyor. Bu anahtar numarası altında ise "**key-string <Key>**" komutuyla Key bilgisi tanımlanıyor.
    - Bu şekilde bir anahtar zincirinde birden fazla key id tanımı oluşturularak kimlik denetimi sürecinde birden fazla key ile kimlik doğrulama süreci gerçekleştirilmesi sağlanabiliyor.
    - **Ek olarak anahtar zincirinde tanımlanan her bir Key için Deal Line süresi belirlenebiliyor. Bu sayede EIGRP protokolünde kullanılan anahtarların belirli zamanlarda değiştirilmesi sağlanabiliyor. Herhangi bir süre belirtilmediği takdirde sürekli geçerli olarak tanımlanıyor.**
- Anahtar zinciri tanımı yapıldıktan sonra komşuluk kuracak portun altında "**ip authentication mode eigrp <AS Number> <Hash Algorithm Name>**" komutuyla hesaplama yapılacak Hash algoritması belirlenmelidir.
- Son olarak "**ip authentication key-chain <Key Chain Name>**" komutuyla kimlik denetiminde kullanılacak anahtar zinciri tanımlanmalıdır.
    - Aşağıdaki görselden de anlaşılacağı üzere Classic Mode EIGRP konfigürasyonunda kullanılan komutlar cihaz geneline dağılıyor.

```
R1(config)# key chain EIGRPKEY
R1(config-keychain)# key 2
R1(config-keychain-key)# key-string CISCO
R1(config)# interface g10/1
R1(config-if)# ip authentication mode eigrp 100 md5
R1(config-if)# ip authentication key-chain eigrp 100 EIGRPKEY
```

- (Named Mode) İlk olarak Global konfigürasyon modu altında "**key chain <Key Chain Name>**" komutuyla bir anahtar zinciri oluşturuluyor. Burada "**key <Key ID>**" komutuyla bir anahtar numarası belirtiliyor. Bu anahtar numarası altında ise "**key-string <Key>**" komutuyla Key bilgisi tanımlanıyor.
- Anahtar zinciri tanımı yapıldıktan sonra EIGRP prosesi altına giriş yapılıp "**address-family {IPv4 | IPv6} {unicast | vrf <VRF NAME>} autonomous-system <AS Number>**" komutuyla ilgili ip adres ailesinin altında ister "**af-interface default**" komutuyla bütün portlarda, ister "**af-interface <Interface ID>**" komutuyla belirli portların altında "**authentication mode <Hash Algorithm Name>**" ve "**authentication key-chain <Key Chain Name>**" komutlarıyla kullanılacak Hash algoritması ve anahtar zinciri belirtilmelidir.

NOT: OSPF protokolünde Authentication Mode olarak Hash algoritması seçilmediği takdirde Key değerini Plain-Text olarak gönderse de kimlik denetimi mekanizması devreye alınıyor. EIGRP protokolünde ise Authentication Mode olarka herhangi bir Hash algoritması belirtilmediği takdirde kimlik denetimi devreye alınmıyor (sadece Key-Chain tanımın belirtilmesi yeterli olmuyor).

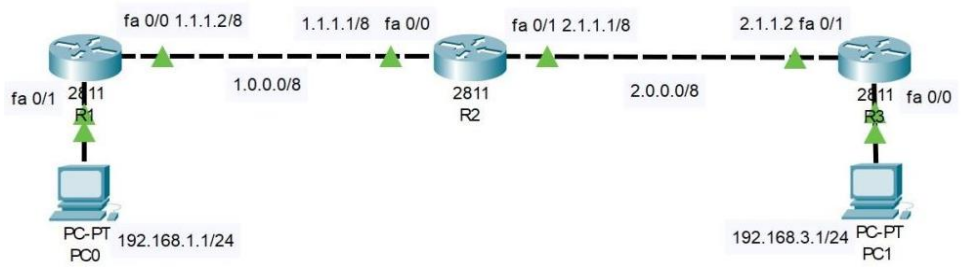
```
Authentication mode is not set
```

```
R2(config)# key chain EIGRPKEY
R2(config-keychain)# key 2
R2(config-keychain-key)# key-string CISCO
R2(config-keychain-key)# router eigrp EIGRP-NAMED
R2(config-router)# address-family ipv4 unicast autonomous-system 100
R2(config-router-af)# af-interface default
R2(config-router-af-interface)# authentication mode md5
R2(config-router-af-interface)# authentication key-chain EIGRPKEY
```

Split Horizon

EIGRP, RIP gibi Distance Vector algoritmaları Neighbor Perspective mekanizmasıyla çalışır. Dolayısıyla komşu routerda öğrendiği rota bilgisini yine komşusuna öğretme ihtimali vardır. Bu durumu aşağıdaki örnek topoloji üzerinden açıklamak gerekirse;

- R2 routeru R3 routeruna üzerinde bulunan 1.1.1.1/8 ve 2.1.1.1/8 networklerini öğretecektir. R3 2.1.1.2/8 networkü kendisine doğrudan bağlı olduğu için sadece 1.1.1.1/8 networkünü yönlendirme tablosuna ekleyecektir.
- EIGRP protokolünde networkte bir değişiklik meydana gelmediği sürece rota bilgilerinin paylaşılmadığından bahsedilmişti. R2 routerunun 1.1.1.1/8 netwoküne erişimi kesildiğinde yedek rota (Feasible Successor) seçilmemişse 1.1.1.1/8 networküne erişebilen komşu routeru olup olmadığını sorgulamak isteyecektir. Bu süreçte R3 henüz güncelleme almadığı takdirde R2'den öğrendiği 1.1.1.1/8 netowrkünü R2'ye anons etmeye devam edecektir.
- Son durumda R2, R3'den gelen anonslar doğrultusunda 1.1.1.1/8 networküne R3 üzerinden gidebileceğini düşünüp 1.1.1.1/8 networküne gidecek paketleri R3'e göndermek üzere yönlendirme tablosuna kaydı ekler.
- R3 routerunun yönlendirme tablosunda da 1.1.1.1/8 networküne gidecek paketleri R2'ye göndermek üzere kayıt bulunacağından paketler R2 ila R3 arasında Loop'a girecektir.
    - Bu durum RIP protokolünde de gerçekleşebilmektedir. RIP protokolünde routerlar her 30 saniyede bir bütün yönlendirme tablosunu komşu routeruna anons eder. Bunun gibi bir durumda komşu router komşusu üzerinden öğrendiği networkün erişiminin kesildiğine dair anons paketini geç alırsa Loop oluşturabilir.

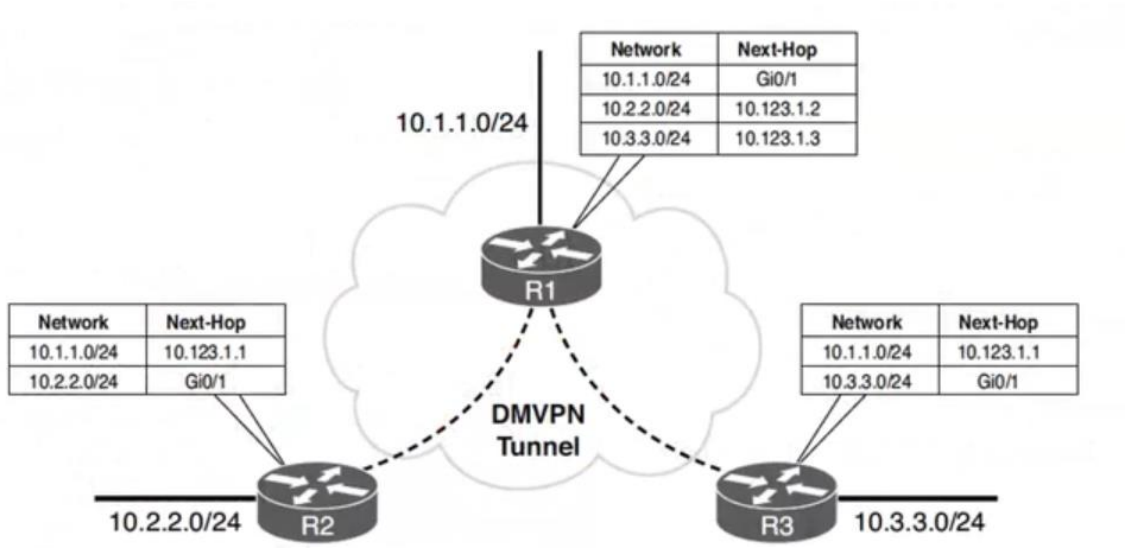


Split Horizon, routerların bir portundan öğrendiği network bilgisinin, o port üzerinden tekrar anons edilmemesini sağlayan özelliktir. Bu sayede komşusundan öğrendiği rota bilgisini yine komşusuna öğretmeyeceği için (R2'nin 1.1.1.1/8 networküne erişimi kesildiğinde R3 benim 1.1.1.1/8 networküne erişimim var diye dönüş sağlamayacaktır) bir noktaya kadar Loop oluşumunun önüne geçilir. Split Horizon özelliği varsayılanda RIP protokolünde de EIGRP protokolünde de devrede gelmektedir (Farklı durumlar için farklı routing Loop koruma mekanizmaları da bulunuyor. Varsayılanda bu koruma mekanizmalarının tamamı açık gelmektedir).

Split Horizon gibi routing Loop önleme mekanizmaları varsayılanda açık gelse de çeşitli teknolojiler ile kullanıldığında sorunlara neden olabiliyor. Bu nedenle kapatılması gerekebiliyor ama bu durum da Loop riski doğuruyor (Bu nedenle OSPF kullan!!! – Loop riski yok, dolayısıyla Loop önleme mekanizmaları da yok). Split Horizon özelliğini kapatmak için ilgili portun arayüzüne giriş yapıldıktan sonra "**no split-horizon**" komutunun kullanılması yeterlidir.

Split Horizon özelliğinin kapatılması gereken bir duruma örnek vermek gerekirse, çok şubesi olan bir kurumun merkezi ile şubeleri arasında DMVPN bağlantısı olan bir topoloji düşünelim. Bu durumda merkez ve şubeler arasında IP telefon kullanılıyorsa ve network bilgileri EIGRP protokolü üzerinden öğreniliyor ise DMVPN tek port üzerinde sanal bağlantılar kuracaktır. Dolayısıyla şubeler haberleşme sürecinde tek bir portu kullanacaktır. Split Horizon özelliğinin açık olması durumunda şubelerden gelen network bilgileri geldiği fiziksel port üzerinden tekrar anons edilmek istendiğinde engellenecektir. Yani şubeler arasında telefon trafiğinin haberleşmesi sağlanamayacaktır. Bu durumda Split Horizon özelliği kapatılarak şubeler arasındaki ip telefon için kullanılan network bilgilerinin yine aynı fiziksel port üzerinde anons edilmesi sağlanmalıdır.

NOT: Tek bir port üzerinde Sub-Interface tanımları oluşturularak, Sub-Interface'ler arasında Split Horizon özelliğinden etkilenmeden EIGRP anons bilgileri gönderilebiliyor.

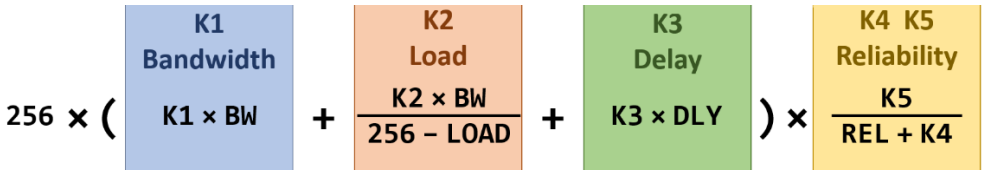


Path Metric Calculations

EIGRP protokolünün metrik hesabında ilgili arayüzün Bandwitdth, Load, Delay ve Realibility özellikleri için sayısal değerler belirleniyor (BW, LOAD, DLY ve REL) ve bu değerler matematiksel bir formüle tabi tutuluyor (Buna **Composite Metric** de deniliyor). Bu formül sonucunda oluşan sayısal ifade bağlantının metrik değerini oluşturuyor (Formül gereği hesaplama sonucunda Metric değerleri çok yüksek olabiliyor)

EIGRP protokolünde varsayılanda Metrik değeri sadece Bandwidth ve Delay özellikleri kullanılarak hesaplanıyor (Nedeni varsayılanda K1 ve K3 değeri 1 olarak gelirken, K2, K4 ve K5'in değerleri 0'dır – yani formülde etkisizdir) ama isteğe bağlı olarak bu değişkenlere 0'dan farklı değerler verilerek diğer özelliklerin de Metrik hesabına dahil olması sağlanabiliyor.

Metrik hesabında varsayılanda BW ve Delay parametrelerinin kullanılmasının nedeni portların çalıştığı hızlara göre BW ve Delay süreleri otomatik olarak belirlenebiliyor ve diğer parametrelere kıyasla daha stabil kalıyor olmasıdır. Bu sayede Metric hesabının daha doğru yapılabilmesi sağlanıyor. Diğer parametreler ile kıyaslandığında, daha sık değişkenlik gösteren olan parametreler (Load, Reliability) de Metric hesabına dahil edilseydi bu parametreler her değişiklik gösterdiğinde ilgili rotanın Metric değerinin güncellenmesi ve akabinde en uygun rotaya karar verebilmek için rotaların yeniden hesaplanması gerekecekti.



$$\text{Metric} = 256 * \left[\left(1 * \frac{10^7}{\text{Min. Bandwidth}} + \frac{\frac{10^7}{0 * \text{Min. Bandwidth}}}{256 - \text{Load}} + \frac{1 * \text{Total Delay}}{10}\right) * \frac{0}{0 + \text{Reliability}}\right]$$

Equals

$$\text{Metric} = 256 * \left(\frac{10^7}{\text{Min. Bandwidth}} + \frac{\text{Total Delay}}{10}\right)$$

|→ Reliability, portta Drop edilen/Bozuk gelen paket sayısıyla ilişkili router hesap yaparak bir güvenilirlik puanı hesaplıyor. Bu puan en fazla 255 olabiliyor.

|→ Load, portun kullanım/Utilization oranına yönelik hesaplama yapılıyor (1/255 hattın boş olduğunu, 255/255 hattın dolu olduğunu gösteriyor (Kullanım miktarı/255).

|→ Formülde Bandwidth değeri 10Gb referans alınarak hesaplanıyor. Bu sayede Bandwidth değer büyük olan arayüzlerde Metrik değerlerinin daha düşük çıkması sağlanıyor.

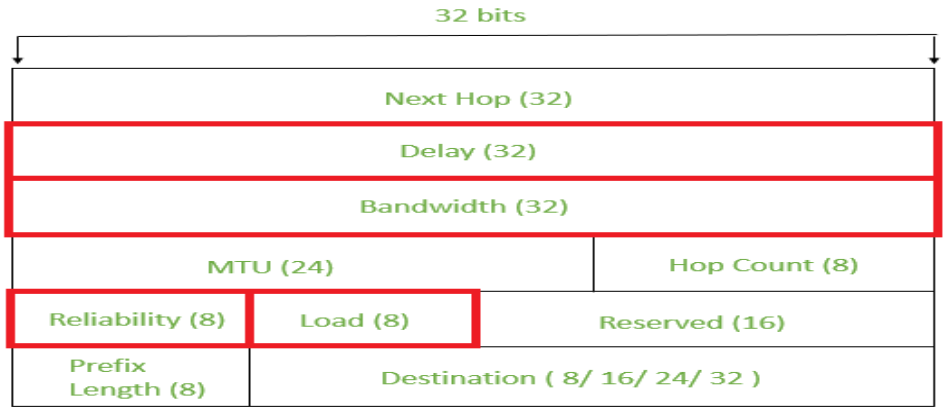
|→ Farklı bir durum ise Metrik hesabında router sadece portun çalıştığı bant genişliğini baz alarak hesap yapar. Kimi zaman ise port 1Gbit çalışıyor görünse de ISP hattından 150Mbit hizmet alınmış olabiliyor. Bu durumda metrik hesabında bant genişliği 1Gbit baz alınarak hesaplanıyor. Dolayısıyla metrik hesabının sağlıklı olmuyor. Bu gibi durumlarda metrik hesabının daha sağlıklı yapılabilmesi için port altında çalışacağı bant genişliği manuel olarak revize edilmelidir.
|→ Benzer şekilde Delay süresi de network yöneticisi tarafından manuel olarak revize edilebilmektedir. Load ve Reliability parametreleri ise sadece router tarafından dinamik olarak hesaplanmaktadır. Network yöneticisi manuel olarak müdahale edememektedir (Reliability ve Load parametreleri Metrik hesabına dahil edilmesi istendiği durumda network admini tarafından sadece K2, k4, k5 değerleri belirlenebiliyor).
|→ Benzer durum Delay parametresini hesaplarken de yaşanıyor. Hesaplamada birim olarak Mikro saniye kullanılıyor. 1Gbit üzerinde çıkıldığında Delay hesabının referans aldığı bant genişliği düşük olduğu için gerçekleştirilen hesaplama sonucu da aynı çıkmaya başlıyor (Totalde 10Gbit üzerine çıkıldığında BW ve Delay parametreleri aynı sonucu veriyor. Rota seçimi sağlıklı yapılmıyor).
|→ Formülasyona K6 değeri getirilerek Jitter (Delay değişimi), enerji değeri gibi çeşitli özellikler de Metrik hesabına eklenebiliyor.

Interface Type	Link Speed (Kbps)	Delay	Metric
Serial	64	20,000 µs	40,512,000
T1	1544	20,000 µs	2,170,031
Ethernet	10,000	1000 µs	281,600
Fast Ethernet	100,000	100 µs	28,160
GigabitEthernet	1,000,000	10 µs	2816
TenGigabitEthernet	10,000,000	10 µs	512

|→ 10 Gbit referans değeri günümüz bant genişlikleriyle karşılaştırıldığında yeterli olmayabiliyor. **10Gbit ve üzeri Bandwidth değere sahip bağlantılarda aynı sonuçları verebiliyor (10Gbit ve üzeri bant genişlikleri için sonucu 0 olarak hesaplıyor).** Bunun için **Wide Metrics** özelliği kullanılıyor ve Bandwidth değeri 10Gbit*65535 değeri baz alınarak hesaplanıyor (BW->65 Tbit, Delay -> piko saniye (10^-12)'ye kadar). Bu sayede çok daha yüksek bant genişlikleri kullanılabiliyor.

$$\text{Wide Metric} = 65{,}535 * \left[\left(\frac{K_1 * 10^7}{\text{Min. Bandwidth}} + \frac{\frac{K_2 * 10^7}{\text{Min. Bandwidth}}}{256 - \text{Load}} + \frac{K_3 * \text{Latency}}{10^{\wedge}6} + K_6 * \text{Extended}\right) * \frac{K_5}{K_4 + \text{Reliability}}\right]$$

NOT: EIGRP protokolünde Wide Metric referanslar değerleri baz alınarak metrik değerinin hesaplanması isteniyorsa Named Mode EIGRP konfigürasyonunun kullanılması gerekiyor. Classic Mode EIGRP konfigürasyonunda bu referansların kullanılarak metrik hesabının yapılabilmesi için değerlerin tek tek manuel olarak değiştirilmesi gerekiyor (**K6 değişkeni sadece Named Mode EIGRP de mevcut**).

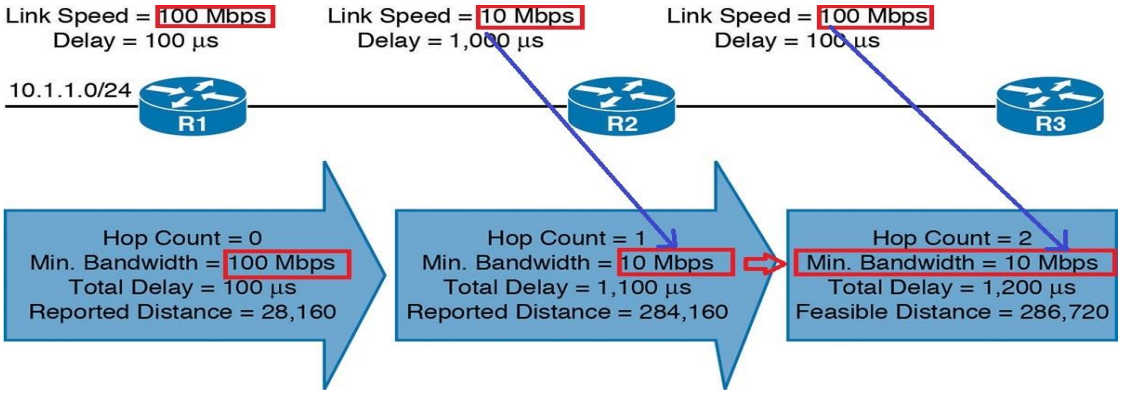


Metrik hesabında Load, Reliability gibi özelliklerin de göz önünde bulundurulması isteniyorsa bu konfigürasyon için K değerlerinde yapılan değişikliklerin AS içerisindeki bütün routerlarda uygulanması gerekiyor. Nedeni, EIGRP anonsu/Update paketi içerisinde bu değerler komşu routera gönderilerek komşu routerun bu bağlantı için kendi Metrik değerini hesaplaması sağlanıyor. Gönderilen değerler komşu routerla uyuşmazsa komşuluk koparılıyor/kurulamıyor.

SORU: Aynı AS içerisinde hem Named Mode EIGRP hem de Classic Mode EIGRP konfigürasyonları kullanılabiliyor mu? Nedeni Classic Mode kullanıldığında metrik hesabında referans değerleri farklı, Named Mode kullanıldığında metrik hesabında referans değerleri farklı oluyor. Dolayısıyla yukarıdaki açıklama üzerine farklı mod kullanılarak EIGRP konfigürasyonu yapılan iki router arasında komşuluk kurulur mu?

CEVAP: Evet çünkü, her ne kadar hesaplanan metrik değerleri farklılık gösterse de K değerleri aynı olduğu sürece komşuluk kurup rota bilgilerini paylaşmaya devam eder.

EIGRP protokolünde varsayılanda Bandwidth ve Delay değerleri kullanılarak Metrik hesabı yapılıyordu. Bantwidth değeri hedef network ve router arasındaki en düşük bant genişliğine sahip bağlantının bant genişliği baz alınarak (Formülde de **Min Bandwidth** olarak geçiyor) hesaplanmaktadır. Bu değer hedef networkten kaynak routera gelene kadar her router arasındaki Bandwidth değeri ile karşılaştırılıyor. Karşılaştırılan Bandwidth değerinden daha düşük Bandwidth değerine sahip bağlantı olduğu anlaşılırsa bu değer güncellenerek iletilmeye devam ediyor. Örneğin;



|→ 10.1.1.0/24 networkünde Bandwidth değeri 100Mbit iken R1-R2 arasında Bandwidth değeri 10Mbit olduğu için R2-R3 arasında da Bandwidth değeri 10Mbit olarak gönderiliyor. R2-R3 arası Bandwidth değeri 10Mbitten daha yüksek olmadığı için Bandwidth değeri güncellenmeden iletilmeye devam ediyor.
|→ Farklı bir bakış açısıyla R1 ve R2 üzerinden erişilecek her rota tanımı için rota boyunca 10Mbit değerinden daha düşük Bandwidth değeriyle karşılaşılmadığı sürece Metrik değerleri 10Mbit olarak hesaplanacaktır.

|→ Delay sürelerinin kimülatif toplanarak iletildiğine dikkat edilmeli.

Portlarda varsayılanda gelen Delay süresi ilgili port altında "**delay <Micro Second>**" komutuyla güncellenebiliyor.

Portlarda çalışan BW değerini değiştirmek için ilgili port altında "**bandwidth <Kbit>**" komutuyla güncellenebiliyor. Burada yapılan değişiklik sadece EIGRP/OSPF gibi protokolleirn metrik değerini hesaplamak için kullandığı değerdir. Portun çalışma hızını etkilememektedir.

Metrik hesabındaki K değerleri ayarlanmak istendiğinde ilgili port altında "**metric weights <ToS Value(QoS)> <K1> <K2> <K3> <K4> <K5>**" komutuyla ayarlanabiliyor. Classic Mode EIGRP konfigürasyonunda 5 tane K değeri vardır. Komutta tanımlanan ilk değer kullanılmamaktadır.

|→ **Burada K değerlerinde yapılan değişikliklerin AS içerisindeki bütün routerlarda aynı uygulanması gerektiği unutulmamalıdır.**

```
R1(config)#router eigrp 1
R1(config-router)#metric weights 0 0 0 1 0 0
R1(config-router)#metric weights QoS K1 K2 K3 K4 K5
```

```
Router(config)#interface fa0/0
Router(config-if)#bandwidth 500
Router(config-if)#delay 50
```

K1 = Bandwidth K2 = Load **K3 = Delay** K4 & K5 = Reliability

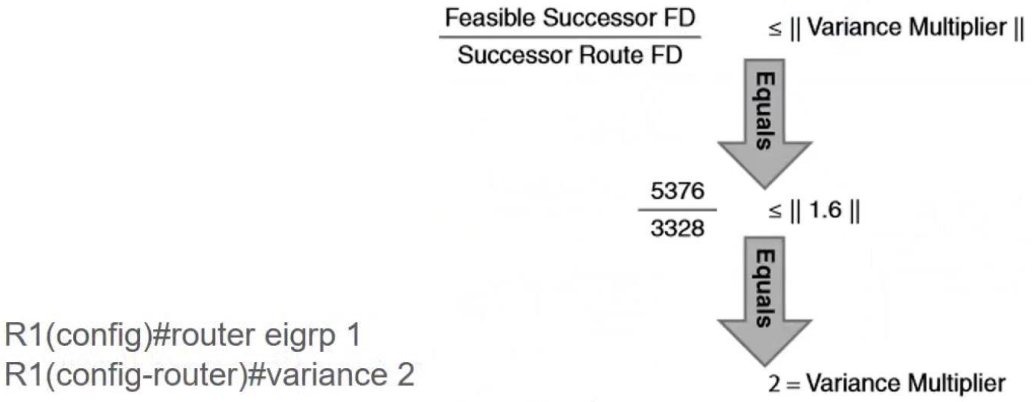
```
Router#show interfaces fastEthernet 0/0
.....
MTU 1500 bytes, BW 500 Kbit, DLY 500 usec,
reliability 255/255, txload 1/255, rxload 1/255
```

Load Balancing

EIGRP protokolünde metrikler aynı olması durumunda rotalar arasında yük paylaşımı yapılabiliyor. Varsayılanda 4 rota arasında yük dengeleme yapılabildiği gibi isteğe bağlı olarak EIGRP prosesi altında "**maximum-paths <Path Count>**" komutuyla arttırılabiliyor.

Diğer dinamik yönlendirme protokollerinden farklı olarak eşit metrik değerlerine sahip olmayan rotalar arasında dahi yük paylaşımı yapılmasına imkan veriyor. Bunun için Variance adı verilen bir komut kullanılıyor. Bu komut kullanıldığında Successor Route seçiminin yanında Feasible Successor (yedek rota) de seçilmişse EIGRP prosesi altında "**variance <Variance Rate>**" komutu kullanmak yeterli oluyor (**Feasible Successor seçilmemiş bir networke alternatif rotalar dahi bulunda Load-Balance yapmaz**).

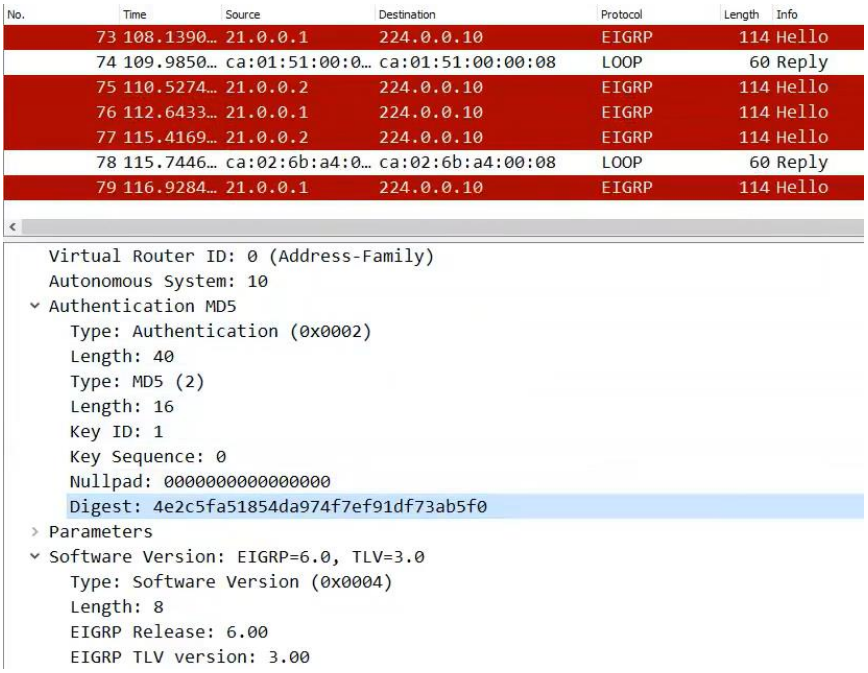
Yük dengeleme sürecinde kullanılacak rota sayısını ise Variant Rate adı verilen değere göre belirliyor. Bu değer Successor Route ve Feasible Successor arasındaki metrik değerlerinin oranlanması sonucunda hesaplanıyor. Örnek olarak "**variance 2**" komutu kullanıldığında Feasible Successor'un metrik değeri Successor Route'un metrik değerinin iki katını geçmeyen her Feasible Successor seçilen rotayı yük dengeleme sürecine dahil ediyor (bu rotaların metrik değerlerini eşit kabul edilerek yük dengeleme yapılıyor).



Notlar

- **EIGRP protokolündeki kimlik denetimi süreci** aslında paket içeriğinin şifrelenmesini sağlamıyor. Paket içerisindeki veriyi bir Hash algoritmasına tabi tutarak bir Hash çıktısı elde ediyor. Elde edilen bu Hash değeri komşu routera gönderilen paket içerisine yerleştiriliyor. Bu süreçte paket içerisindeki bilgiler Plain-Text olarak gönderiliyor. Paket komşu routera ulaştığında, router kendisinde bulunan Key-Chain tanımıyla birlikte paket içeriğini Hash algoritmasına tabi tutarak paket içerisinde gönderilen Hash değeriyle eşleşip eşleşmediğini kontrol ediyor. Eğer ki Hash çıktısı eşleşiyorsa bu iki anlama geliyor;

- Paket içeriğinin kendisine gelene kadarki süreçte içeriğinin değiştirilmediğini (değiştirilmiş olsaydı kendisi paket içeriğiyle Key-Chain değerini Hash algoritmasına tabi tuttuğunda farklı Hash çıktılarıyla karşılaşılacaktı)
- Paketi gönderen router Hash bilgisini doğru hesapladığına göre paketi gönderen routerun doğru router olduğu anlaşılıyor.



- EIGRP ve OSPF arasında rota aktarımı yapabilmek için;
  - Öncelikle rota aktarımı yapılacak “**router ospf <Proccess ID>**” komutuyla OSPF prosesinine giriş yapılır ve “**redistribute eigrp <EIGRP AS Number> {subnets | metric | metric-type| …}**” komutu kullanılarak rota bilgilerinin ilgili EIGRP AS’i ile paylaşılması sağlanır.
    - Örnek komut olarak “redistribute eigrp 15” verilebilir.
    - Aynı şekilde EIGRP protokolünden OSPF protokolüne rota aktarılabilmesi için “**router eigrp <AS Number>**” komutuyla ilgili EIGRP AS içerisine girilir ve burada da “ **redistribute ospf <OSPF Proccess ID> {metric|match|route-map} <Bandwidth> <Delay Metric> <EIGRP Reliability Metric> <EIGRP Effective Bandwidth Metric> <MTU of the Path>**” komutuyla rota bilgilerinin aktarılabilmesi sağlanır.
    - Örnek komut olarak “**redistribute ospf 10 metric 1000 100 255 1 1500**” verilebilir.

```
R7(config)#router ospf 5
R7(config-router)#redistribute eigrp ?
  <1-65535>  AS number

R7(config-router)#redistribute eigrp 15 ?
  metric       Metric for redistributed routes
  metric-type  OSPF/IS-IS exterior metric type for redistributed routes
  nssa-only    Limit redistributed routes to NSSA areas
  route-map    Route map reference
  subnets      Consider subnets for redistribution into OSPF
  tag          Set tag for routes redistributed into OSPF
  <cr>

R7(config-router)#redistribute eigrp 15 metric ?
  <0-16777214>  OSPF default metric

R7(config-router)#redistribute eigrp 15 metric 1 sub
R7(config-router)#redistribute eigrp 15 metric 1 subnets ?
  metric-type  OSPF/IS-IS exterior metric type for redistributed routes
  nssa-only    Limit redistributed routes to NSSA areas
  route-map    Route map reference
  tag          Set tag for routes redistributed into OSPF
  <cr>

R7(config-router)#redistribute eigrp 15 metric 1 subnets
R7(config-router)#
```

```
R7(config)#router eigrp 15
R7(config-router)#redistribute ospf ?
  <1-65535>  Process ID

R7(config-router)#redistribute ospf 5 ?
  match      Redistribution of OSPF routes
  metric     Metric for redistributed routes
  route-map  Route map reference
  <cr>

R7(config-router)#redistribute ospf 5 metric ?
  <1-4294967295>  Bandwidth metric in Kbits per second

R7(config-router)#redistribute ospf 5 metric 1000 ?
  <0-4294967295>  EIGRP delay metric, in 10 microsecond units

R7(config-router)#redistribute ospf 5 metric 1000 100 ?
  <0-255>  EIGRP reliability metric where 255 is 100% reliable

R7(config-router)#redistribute ospf 5 metric 1000 100 255 ?
  <1-255>  EIGRP Effective bandwidth metric (Loading) where 255 is 100% loaded

R7(config-router)#redistribute ospf 5 metric 1000 100 255 1 ?
  <1-65535>  EIGRP MTU of the path

R7(config-router)#redistribute ospf 5 metric 1000 100 255 1 1500 ?
  match      Redistribution of OSPF routes
  route-map  Route map reference
  <cr>

R7(config-router)#redistribute ospf 5 metric 1000 100 255 1 1500
R7(config-router)#
```

Kontrol Komutları

- Sh key chain
- Sh ip eigrp interface detail