Statik routing'i oluşturup onu çalışır hale getirmeniz, ağ topluluğunu iyi bildiğiniz anlamına geldiğinden, network'ümüzdeki statik routingi açıklayıp uygulayarak başlayacağım. Hadi başlayalım!

## Statik Routing

Her router'ın, routing tablosuna manuel olarak route'lar girdiğinizde, statik routing olur. Statik route'a lehte ve aleyhte olanlar vardır, fakat tüm routing prosesleri için bu böyledir.

Statik routing, aşağıdaki faydaları sağlamaktadır:

- Router CPU'suna ek yük getirmez. Yani, dinamik routing kullandığınızdan daha ucuz bir router almanız mümkündür.
- Router'lar arasında bant genişliği kullanımı yoktur. Yani, WAN linklerinde, para tasarrufu sağlayabilirsiniz.
- Yöneticiler, sadece belirli network'lere routing erişimine izin vermeyi seçebildiklerinden, güvenlik sağlar.

Statik routing aşağıdaki dezavantajlara sahiptir:

- Yönetici, ağ topluluğunu ve doğru yapılandırmak için her router'ın nasıl bağlandığını iyi bilmelidir.
- Ağ topluluğuna bir network eklenirse yönetici elle, her router'da ona bir route eklemelidir.
- Büyük network'lerde uygulanabilir değildir, çünkü onun devamını sağlamak, tam-zamanlı bir iştir.

Aşağıda, bir routing tablosuna statik route eklemek için kullandığınız komut dizisi vardır:

```
ip route [destination_network] [mask] [next-hop_address or
  exitinterface] [administrative_distance] [permanent]
```

Dizgide ki her komut aşağıda açıklanmaktadır:

**`ip route`:** Statik route oluşturmak için kullanılan komut.

***`destination_network`:*** Routing tablosuna yerleştirilen network.

***`mask`:*** Ağda kullanılan subnet maskı.

***`next-hop_address`:*** Paketi alıp onu uzak network'e gönderecek next-hop router'ın adresidir. Bu, direk bağlı network'lerdeki bir router interface'idir. Route eklemeden önce router interface'ini pingleyebilmelisiniz. Şayet yanlış next-hop adresi girerseniz yada bu router'a interface'i down ise, statik route, router konfigürasyonunda görünecektir, fakat routing tablosunda görünmeyecektir.

***`exitinterface`:*** İsterseniz, next-hop adresinin yerine kullanılır ve direkt bağlı bir route gibi görünür.

***`administrative_distance`:*** Varsayılan olarak, statik route'un administrative distance'ı 1'dir (next-hop adresi yerine bir çıkış interface'i kullanırsanız, 0'dır) Komutun sonuna bir administrative weight ekleyerek varsayılan değeri değiştirebilirsiniz. Dinamik routing bölümüne geçince, bu konu hakkında daha çok bahsedeceğim.

**`permanent`:** Şayet interface kapalıysa yada router, next-hop router ile haberleşemiyorsa, route, routing tablosundan otomatik olarak atılacaktır. *Permanent* seçeneği, ne olursa olsun, routing tablosundaki kaydı korur.

Statik route konfigürasyonuna girmeden önce, örnek bir statik route'a ve ondan ne anladığımıza bir bakalım.

```
Router(config)#ip route 172.16.3.0 255.255.255.0 192.168.2.4
```

- `ip route` komutu, bunun bir statik route olduğunu söyler.
- 172.16.3.0, paketleri göndermek istediğimiz uzak network'tür.
- 255.255.255.0, uzak network'ün maskıdır.
- 192.168.2.4, paketleri göndereceğimiz next hop ya da router'dır.

Ancak, statik route'ın şöyle olması düşünülür:

```
Router(config)#ip route 172.16.3.0 255.255.255.0 192.168.2.4 150
```

Sondaki 150, 1 olan varsayılan administrative distance(AD)'ı, 150 olarak değiştirir. Endişelenmeyin, Dinamik routing'e geçtiğimizde, AD'den daha fazla bahsedeceğim. Şimdi, AD'nin bir route'ın güvenilirliği olduğunu hatırlayın (0, en iyisi, 255 en kötüsüdür).

Bir örnek daha yapalım, sonra konfigürasyona geçeceğim:

```
Router(config)#ip route 172.16.3.0 255.255.255.0 s0/0/0
```

Next-hop adresi kullanmak yerine, bir çıkış interface'i kullanabiliriz. Bu, route'u, direk bağlı network gibi gösterir. İşlevsel olarak, next hop ve çıkış interface'i tamamen aynıdır. Statik route'ın nasıl çalıştığını anlamanıza yardımcı olması için, daha önce Şekil 6.9'da gösterilen ağ topluluğundaki konfigürasyonu göstereceğim.

## Corp

Her routing tablosu, direk bağlı network'leri otomatik olarak görür. Ağ topluluğundaki tüm network'e route olabilmesi için, routing tablosunun, diğer network'lerin nerede olduklarını ve onlara nasıl erişileceğini tanımlayan bilgiye sahip olması gerekir.

Corp router, beş network'e bağlıdır. Corp router'ın tüm network'lere route'a sahip olması için, aşağıdaki network'ler, routing tablosunda yapılandırılmalıdır:

- 10.1.6.0
- 10.1.7.0
- 10.1.8.0
- 10.1.9.0
- 10.1.10.0
- 10.1.11.0
- 10.1.12.0

Aşağıdaki router çıktısı, Corp router'ındaki statik route'ları ve konfigürasyon sonrası routing tablosunu gösterir. Corp router'ının, uzak ağları bulması için, uzak network'ü, uzak maskı ve paketlerin nereye gönderileceğini açıklayan bir kaydı routing tablosuna girdim. Administrative distance'ı yükseltmek için her satırın sonuna "150" ekleyeceğim. (dinamik routinge geçtiğimde, bunu neden kullandığımı göreceksiniz.)

```
Corp(config)#ip route 10.1.6.0 255.255.255.0 10.1.2.2 150
Corp(config)#ip route 10.1.6.0 255.255.255.0 10.1.3.2 151
Corp(config)#ip route 10.1.7.0 255.255.255.0 10.1.3.2 150
Corp(config)#ip route 10.1.7.0 255.255.255.0 10.1.2.2 151
Corp(config)#ip route 10.1.8.0 255.255.255.0 10.1.4.2 150
Corp(config)#ip route 10.1.9.0 255.255.255.0 10.1.4.2 150
Corp(config)#ip route 10.1.10.0 255.255.255.0 10.1.5.2 150
```

```
Corp(config)#ip route 10.1.11.0 255.255.255.0 10.1.5.2 150
Corp(config)#ip route 10.1.12.0 255.255.255.0 10.1.5.2 150
Corp(config)#do show run | begin ip route
ip route 10.1.6.0 255.255.255.0 10.1.2.2 150
ip route 10.1.6.0 255.255.255.0 10.1.3.2 151
ip route 10.1.7.0 255.255.255.0 10.1.3.2 150
ip route 10.1.7.0 255.255.255.0 10.1.2.2 151
ip route 10.1.8.0 255.255.255.0 10.1.4.2 150
ip route 10.1.9.0 255.255.255.0 10.1.4.2 150
ip route 10.1.10.0 255.255.255.0 10.1.5.2 150
ip route 10.1.11.0 255.255.255.0 10.1.5.2 150
ip route 10.1.12.0 255.255.255.0 10.1.5.2 150
```

10.1.6.0 ve 10.1.7.0 network'leri için, her network'e iki yolu ekledim, fakat linklerden birinin AD'sini 151 yaptım. Diğer linkin arızalanması durumunda bu yedek bir route olacaktır. Şayet ikisine de aynı AD'yi verseydim, bir routing kısır döngüsü oluşacaktı. (Statik routing, aynı hedefe çok sayıda linki kullanamaz.) Router yapılandırıldıktan sonra, statik route'ları görmek için `show ip route` yazabilirsiniz:

```
Corp(config)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
S       10.1.11.0 [150/0] via 10.1.5.2
S       10.1.10.0 [150/0] via 10.1.5.2
S       10.1.9.0 [150/0] via 10.1.4.2
S       10.1.8.0 [150/0] via 10.1.4.2
S       10.1.12.0 [150/0] via 10.1.5.2
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
C       10.1.1.0 is directly connected, FastEthernet0/1
S       10.1.7.0 [150/0] via 10.1.3.2
S       10.1.6.0 [150/0] via 10.1.2.2
C       10.1.5.0 is directly connected, Serial0/2/0
C       10.1.4.0 is directly connected, Serial0/1/0
```

Corp router, route etmek için yapılandırıldı ve tüm network'lere giden route'ları bilmektedir. R1'de, her uzak network'e iki route yapılandırdım, fakat routing tablosu sadece düşük AD olanını gösterecektir. Diğer link sadece, kullanılan düşük AD'li link arızalandığında routing tablosunda görünecektir.

Şayet route'lar, routing tablosunda görünmezse, router'ın, yapılandırdığınız next-hop router ile haberleşemediğini anlamanızı istiyorum. Next-hop cihazıyla bağlantı kurulamasa dahi, routing tablosundaki route'u saklamaya devam etmek için permanent parametresini kullanabilirsiniz.

Yukarıdaki routing tablosundaki S'in anlamı, network'ün bir statik kayıt olduğudur. [1/0], uzak network'e, administrative distance ve metriği belirtir. Burada, netxt-hop interface'inin 0 olması, onun direk bağlı olduğunu işaret eder.

NOT *Statik konfigürasyonun sonundaki 150/151 için kaygılanmayın. Bu modülde, en kısa sürede bundan bahsedeceğim. Bu noktada, bundan endişelenmemeniz gerektiğinden emin olabilirsiniz.*

Tamam, iyi gidiyoruz. Corp router şimdi, tüm uzak ağlarla haberleşmesi için gerekli bilgilere sahiptir. Fakat unutmayın ki, R1,

R2, R3 ve 871W router'ları, aynı bilgilerle yapılandırılmadıysa, paketler, atılacaktır. Bunu, statik route'lar yapılandırarak düzeltebiliriz.

## R1

R1 router'ı, 10.1.2.0, 10.1.3.0, 10.1.6.0 ve 10.1.7.0 ağlarına direk bağlıdır, bu nedenle, R1 router'ında, aşağıdaki statik route'ları oluşturduk:

- 10.1.1.0
- 10.1.4.0
- 10.1.5.0
- 10.1.8.0
- 10.1.9.0
- 10.1.10.0
- 10.1.11.0
- 10.1.12.0

R1 router'ı için konfigürasyon şöyle olacaktır; direk bağlı olduğumuz network'ler için statik route oluşturmadığımızı ve Corp ve R1 router'ları arasında iki linkimiz olduğundan, next hop olarak, 10.1.2.1 ya da 10.1.3.1'i kullanabileceğimizi hatırlayın. Next hop'ları değiştireceğim, böylece, tüm veri, tek linkten gitmeyecek. Statik routing ile load-balance yapmadığımdan, hangi linki kullanacağım önemli değildir. RIP, EIGRP ve OSPF gibi dinamik routing kullandığımızda, load-balance yapabileceğiz, fakat şimdi, linkler, her network'e bir yedek route sağlayacaktır. Çıktıyı kontrol edelim:

```
R1(config)#ip route 10.1.1.0 255.255.255.0 10.1.2.1 150
R1(config)#ip route 10.1.1.0 255.255.255.0 10.1.3.1 151
R1(config)#ip route 10.1.4.0 255.255.255.0 10.1.2.1 150
R1(config)#ip route 10.1.4.0 255.255.255.0 10.1.3.1 151
R1(config)#ip route 10.1.5.0 255.255.255.0 10.1.2.1 150
R1(config)#ip route 10.1.5.0 255.255.255.0 10.1.3.1 151
R1(config)#ip route 10.1.8.0 255.255.255.0 10.1.3.1 150
R1(config)#ip route 10.1.8.0 255.255.255.0 10.1.2.1 151
R1(config)#ip route 10.1.9.0 255.255.255.0 10.1.3.1 150
R1(config)#ip route 10.1.9.0 255.255.255.0 10.1.2.1 151
R1(config)#ip route 10.1.10.0 255.255.255.0 10.1.3.1 150
R1(config)#ip route 10.1.10.0 255.255.255.0 10.1.2.1 151
R1(config)#ip route 10.1.11.0 255.255.255.0 10.1.3.1 150
R1(config)#ip route 10.1.11.0 255.255.255.0 10.1.2.1 151
R1(config)#ip route 10.1.12.0 255.255.255.0 10.1.3.1 150
R1(config)#ip route 10.1.12.0 255.255.255.0 10.1.2.1 151
R1(config)#do show run | begin ip route
ip route 10.1.1.0 255.255.255.0 10.1.2.1 150
ip route 10.1.1.0 255.255.255.0 10.1.3.1 151
ip route 10.1.4.0 255.255.255.0 10.1.2.1 150
ip route 10.1.4.0 255.255.255.0 10.1.3.1 151
ip route 10.1.5.0 255.255.255.0 10.1.2.1 150
ip route 10.1.5.0 255.255.255.0 10.1.3.1 151
```

```
ip route 10.1.8.0 255.255.255.0 10.1.3.1 150
ip route 10.1.8.0 255.255.255.0 10.1.2.1 151
ip route 10.1.9.0 255.255.255.0 10.1.3.1 150
ip route 10.1.9.0 255.255.255.0 10.1.2.1 151
ip route 10.1.10.0 255.255.255.0 10.1.3.1 150
ip route 10.1.10.0 255.255.255.0 10.1.2.1 151
ip route 10.1.11.0 255.255.255.0 10.1.3.1 150
ip route 10.1.11.0 255.255.255.0 10.1.2.1 151
ip route 10.1.12.0 255.255.255.0 10.1.3.1 150
ip route 10.1.12.0 255.255.255.0 10.1.2.1 151
```

Her network'e iki yol yapılandırdığımdan, bu oldukça uzun bir konfigürasyon oldu. Routing tablosuna bakarak, R1 router'ının, tüm network'lere nasıl ulaşacağını bildiğini görebilirsiniz:

```
R1(config)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
S       10.1.11.0 [150/0] via 10.1.3.1
S       10.1.10.0 [150/0] via 10.1.3.1
S       10.1.9.0 [150/0] via 10.1.3.1
S       10.1.8.0 [150/0] via 10.1.3.1
S       10.1.12.0 [150/0] via 10.1.3.1
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
S       10.1.1.0 [150/0] via 10.1.2.1
C       10.1.7.0 is directly connected, FastEthernet0/1
C       10.1.6.0 is directly connected, FastEthernet0/0
S       10.1.5.0 [150/0] via 10.1.2.1
S       10.1.4.0 [150/0] via 10.1.2.1
```

NOT *Yüksek administrative distance 'a sahip route'un, düşük olanın kullanım dışı kalması dışında routing tablosunda görünmeyeceğini hatırlayın.*

R1 router, artık komple routing tablosuna sahiptir. Ağ topluluğundaki diğer router'lar, routing tablolarında bütün ağlarla ilgili bilgiye sahip olur olmaz, R1, tüm uzak ağlarla haberleşebilecektir.

## R2

R2 router'ı, 10.1.4.0, 10.1.8.0 ve 10.1.9.0 ağlarına direk bağlıdır, bu nedenle, şu route'lar eklenmelidir:

- 10.1.1.0
- 10.1.2.0
- 10.1.3.0
- 10.1.5.0
- 10.1.6.0
- 10.1.7.0
- 10.1.10.0
- 10.1.11.0
- 10.1.12.0

R2 router konfigürasyonu aşağıdaki gibidir:

```
R2(config)#ip route 10.1.1.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.2.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.3.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.5.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.6.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.7.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.10.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.11.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.12.0 255.255.255.0 10.1.4.1 150
R2(config)#do show run | begin ip route
ip route 10.1.1.0 255.255.255.0 10.1.4.1 150
ip route 10.1.2.0 255.255.255.0 10.1.4.1 150
ip route 10.1.3.0 255.255.255.0 10.1.4.1 150
ip route 10.1.5.0 255.255.255.0 10.1.4.1 150
ip route 10.1.6.0 255.255.255.0 10.1.4.1 150
ip route 10.1.7.0 255.255.255.0 10.1.4.1 150
ip route 10.1.10.0 255.255.255.0 10.1.4.1 150
ip route 10.1.11.0 255.255.255.0 10.1.4.1 150
ip route 10.1.12.0 255.255.255.0 10.1.4.1 150
```

Aşağıdaki çıktı, R2 router'ındaki routing tablosunu göstermektedir:

```
R2(config)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
S       10.1.11.0 [150/0] via 10.1.4.1
S       10.1.10.0 [150/0] via 10.1.4.1
C       10.1.9.0 is directly connected, FastEthernet0/0
C       10.1.8.0 is directly connected, Dot11Radio0/3/0
S       10.1.12.0 [150/0] via 10.1.4.1
S       10.1.3.0 [150/0] via 10.1.4.1
S       10.1.2.0 [150/0] via 10.1.4.1
S       10.1.1.0 [150/0] via 10.1.4.1
S       10.1.7.0 [150/0] via 10.1.4.1
S       10.1.6.0 [150/0] via 10.1.4.1
S       10.1.5.0 [150/0] via 10.1.4.1
C       10.1.4.0 is directly connected, Serial0/2/0
```

R2, şimdi ağ topluluğundaki 12 network'ün hepsini göstermektedir, bu nedenle artık o da tüm network ve router'larla (şimdiye kadar yapılandırılmış olanlarla) haberleşebilecektir.

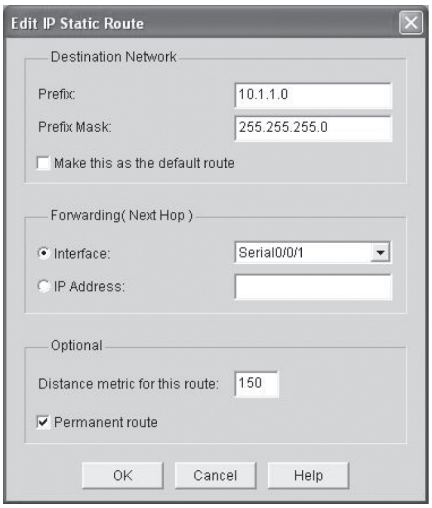
## R3

R3 router'ı, 10.1.5.0, 10.1.10.0 ve 10.1.11.0 ağlarına direk bağlıdır, bu nedenle, şu routeları eklememiz gerekir:

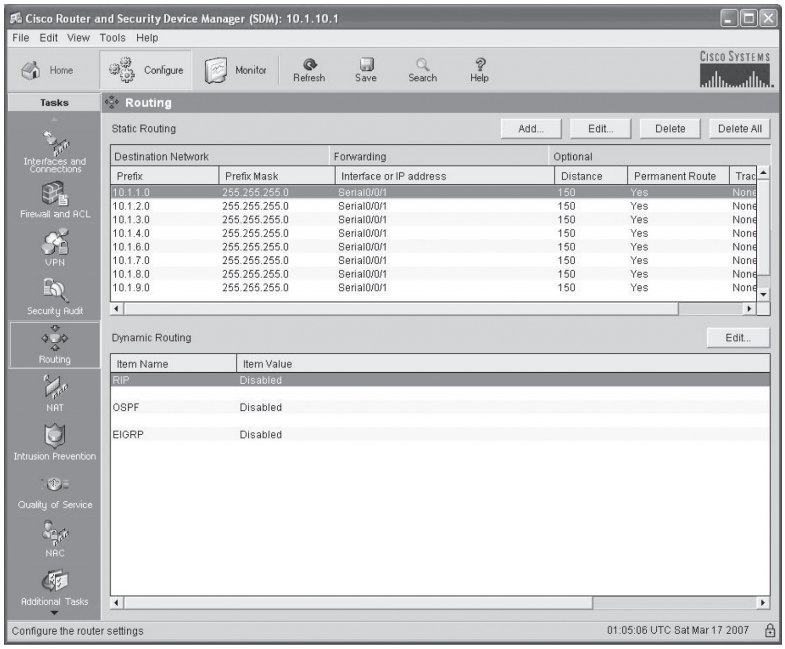
- 10.1.1.0
- 10.1.2.0

- 10.1.3.0
- 10.1.4.0
- 10.1.6.0
- 10.1.7.0
- 10.1.8.0
- 10.1.9.0
- 10.1.12.0

Önceden yaptığım gibi, R3 router'ına statik routing yapılandırmak için SDM'i kullanacağım. Konfigürasyon oldukça basittir. Next-hop adresi ya da çıkış interface'ini kullanabilirim. Mümkün olduğu kadar kısa yazmayı sevdiğimden, sadece bir mouse tıklaması olduğundan, çıkış interface'ini kullanacağım.




Tüm route'ları yapılandırdıktan sonra, onları routing ekranında görebiliriz.




Bu ekrandan, statik route'ları düzenlemek kolaydır.

Şimdi, SDM'den router'a yüklenen konfigürasyona ve routing tablosuna bir bakalım:

```
R3#show run | begin ip route
ip route 10.1.1.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.2.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.3.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.4.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.6.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.7.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.8.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.9.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.12.0 255.255.255.0 FastEthernet0/1 150 permanent
R3#show ip route
     10.0.0.0/24 is subnetted, 12 subnets
C       10.1.11.0 is directly connected, FastEthernet0/1
C       10.1.10.0 is directly connected, FastEthernet0/0
S       10.1.9.0 is directly connected, Serial0/0/1
S       10.1.8.0 is directly connected, Serial0/0/1
S       10.1.12.0 is directly connected, FastEthernet0/1
S       10.1.3.0 is directly connected, Serial0/0/1
S       10.1.2.0 is directly connected, Serial0/0/1
S       10.1.1.0 is directly connected, Serial0/0/1
S       10.1.7.0 is directly connected, Serial0/0/1
S       10.1.6.0 is directly connected, Serial0/0/1
C       10.1.5.0 is directly connected, Serial0/0/1
S       10.1.4.0 is directly connected, Serial0/0/1
R3#
```

`Show ip route` komut çıktısına bakarak, statik route'ların direkt bağlı olarak listelendiğini görebilirsiniz. Garip? Değil, çünkü next-hop adres yerine, çıkış interface'i kullandığım ve işlevsel olarak fark yoktur. Gerçekte, `permanent` komutu kullanmamıza gerek yoktur. Çünkü tek yapacağı, bu linke giden link arızalansa bile, route'un, routing tablosunda kalmasını sağlamaktır. `Permanent` komutunu sırf SDM ile yapması kolay olduğundan (tek fare tıklaması) yapılandırdım. Neredeyse tamamlamak üzereyiz, sadece 871W kaldı.

## 871W

Şimdi, bu router için 871W'i stub olarak yapılandıracağımdan dolayı default routing kullanacağım. Stub, bu tasarımdaki kablosuz ağların, diğer ağlara ulaşmak için sadece bir yola sahip olduklarını belirtir. Şimdiki bölümde, konfigürasyonu size gösterip network'ün doğruluğunu kontrol ettikten sonra default routing'ten detaylı bahsedeceğim. İşte konfigürasyon:

```
871W(config)#ip route 0.0.0.0 0.0.0.0 10.1.11.1
871W(config)#ip classless
871W(config)#do show ip route
     10.0.0.0/24 is subnetted, 2 subnets
C       10.1.11.0 is directly connected, Vlan1
C       10.1.12.0 is directly connected, Dot11Radio0
S*   0.0.0.0/0 [1/0] via 10.1.11.1
871W(config)#
```

Bu, çok daha kolay gözüküyor, değil mi? Evet öyledir, fakat bunu tüm router'larda yapamazsınız, sadece stub network'lerde yapabilirsiniz. R1 ve R2 router'larında da, default routing kullanabilirdim ve kolayca yapabileceğim halde, 150'yi bu default route'a eklemedim. Sonradan dinamik route kullandığımızda route'u kaldırmak oldukça basit olduğundan, bunu yapmadım.

Böylece tamamladık. Tüm router'lar doğru routing tablosuna sahiptir. Bu nedenle, tüm router ve host'lar problemsiz haberleşebilmelidir. Şayet ağ topluluğuna bir tane daha network ya da router eklersek, her router'ın routing tablosunu elle güncellemek zorundasınız. Şayet küçük bir network'ünüz varsa, bu bir problem sayılmaz. Fakat büyük bir ağ topluluğuyla uğraşıyorsanız, çok fazla zaman alacaktır.

## Konfigürasyonunuzun Doğruluğunu Kontrol Etmek

Henüz tamamlayamadık. Tüm router'ların routing tablosu yapılandırılınca, doğruluklarının kontrol edilmesi gerekir. Bunu yapmanın en iyi yolu, `show ip route` komutunu kullanmanın yanında, Ping programını kullanmaktır. 1242 AP'den 871W router'ını pingleyerek başlayacağım.

İşte çıktısı:

```
871W#ping 10.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4
ms

Router871W'dan HostA, B, C ve D'ye ping atmak, iyi bir IP
bağlanırlığı testi olacaktır. Router çıktısı şöyledir:
871W#ping 10.1.6.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.6.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
4/6/12 ms
871W#ping 10.1.7.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.7.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4
ms
871W#ping 10.1.9.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.9.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4
ms
871W#ping 10.1.10.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.10.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5)
```

Ayrıca, paketin HostA'ya gidene kadarki uğradığı noktaları görmek için, 871W router'dan trace kullanabiliriz:

```
871W#trace 10.1.6.2
Type escape sequence to abort.
Tracing the route to 10.1.6.2
  1 10.1.11.1 0 msec 0 msec 0 msec
  2 10.1.5.1 4 msec 0 msec 4 msec
  3 10.1.2.2 0 msec 0 msec 4 msec
  4 10.1.6.2 4 msec 4 msec *
```

Uçtan uca ve her host'la, problemsiz haberleşebildiğimiz için statik route konfigürasyonumuz başarılıdır!

## Default Routing

Default routing'i, routing tablosunda olmayan uzak bir hedef network için paketleri, next-hop router'a göndermek için kullanırız. Default routing'i sadece, network'ten dışarı tek çıkış yolu olan stub network'lerde kullanmalısınız.

Önceki bölümde kullanılan ağlar arası haberleşme örneğinde, stub bir network'te olduğu düşünülecek router'lar, R1, R2 ve 871W'dir. R3 router'ına bir default route koymayı denerseniz, diğer router'lara yönlendirilmiş birden fazla interface olduğundan, paketler, doğru network'lere gönderilmeyecektir. Default routing ile kolayca kısır döngüler oluşturabilirsiniz. Bu yüzden çok dikkatli olun!

Bir default route oluşturmak için statik route'un network adres ve mask lokasyonlarında wildcard maskları kullanırsınız (871W konfigürasyonunda gösterdiğim gibi). Aslında, default route'u, network ve mask bilgisinin yerine wildcard'lar kullanan bir statik route olarak düşünebilirsiniz.

Bir default route kullanarak, sadece bir statik route kaydı oluşturabilirsiniz. Bu, tüm route'ları yazmaktan daha kolaydır!

```
871W(config)#ip route 0.0.0.0 0.0.0.0 10.1.11.1
871W(config)#ip classless
871W(config)#do show ip route
  Gateway of last resort is 10.1.11.1 to network 0.0.0.0
 10.0.0.0/24 is subnetted, 2 subnets
C       10.1.11.0 is directly connected, Vlan1
C       10.1.12.0 is directly connected, Dot11Radio0
S*   0.0.0.0/0 [1/0] via 10.1.11.1
871W(config)#
```

Routing tablosuna baktığınızda, sadece iki direkt bağlı network ve bu kaydın bir default route adayı olduğunu belirten bir S* göreceksiniz. Default route komutunu başka bir şekilde tamamlayabilirdim:

```
871W(config)#ip route 0.0.0.0 0.0.0.0 vlan1
```

Burada bize söylenen, "routing tablosunda bir network için kayıt yoksa onu VLAN1'e gönder" dir. (Onu FastEthernet0/0'dan gönderecektir.) Next-hop router'ın IP adresini ya da çıkış interface'ini seçebilirsiniz, her ikisinde de, aynı şekilde çalışacaktır. Bu çıkış interface konfigürasyonunu, R3 statik router yapılandırmasında kullandığımı hatırlayın.

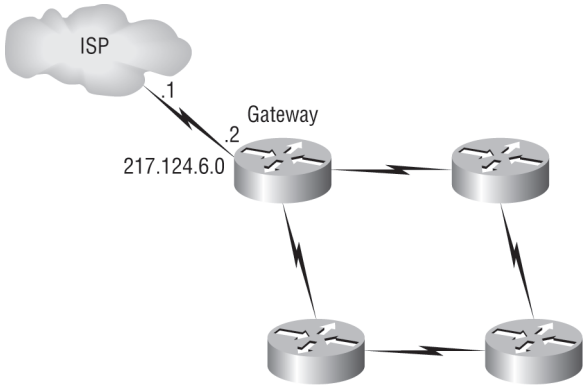
Ayrıca, routing tablosunda gateway of last resort'un ayarlandığına dikkat edin. Böyle olsa da, default route'lar kullanırken, bilmeniz gereken bir komut daha vardır: ***ip classless*** komutu.

Tüm Cisco router'ları, classful router'lardır. Yani, onlar router'ın her interface'inde varsayılan subnet maskı olmasını bekler. Router, routing tablosunda olmayan bir hedef subnet için paket alırsa varsayılan olarak bu paket atılacaktır. Şayet default routing kullanırsanız, routing tablosunda uzak subnet'lerin olmaması mümkün olacağından, ***ip classless*** komutu kullanılmalıdır.

Router'larımda, 12.4 IOS versiyonu olduğundan, `ip classless` komutu varsayılanda aktiftir. Şayet, default routing kullanıyorsanız ve bu komut konfigürasyonunuzda yoksa router'larınızdaki, network'leri subnet'lediyseniz, onu eklemeniz gerekecektir. Komut şöyle kullanılmaktadır:

```
871W(config)#ip classless
```

Onun bir global configuration mod komutu olduğuna dikkat edin. `İp classless` komutunun ilginç tarafı, onsuz, default routing'in bazen çalışması, bazen çalışmamasıdır. Güvenilir olması için, default routing kullandığınızda, daima `ip classless` komutunu çalıştırın.




***Şekil 6.10:*** *Bir gateway of last resort yapılandırmak.*

Bir gateway of last resort yapılandırmak için kullanabileceğiniz diğer bir komut, ***ip default-network*** komutudur. Şekil 6.10, yapılandırılmış bir gateway of last resort ibaresi olması gereken bir ağı göstermektedir.

ISP için gateway router'a bir gateway of last resort eklemek için kullanılan üç komut şöyledir (hepside aynı çözümü sağlar):

```
Gateway(config)#ip route 0.0.0.0 0.0.0.0 217.124.6.1

Gateway(config)#ip route 0.0.0.0 0.0.0.0 s0/0

Gateway(config)#ip default-network 217.124.6.0
```

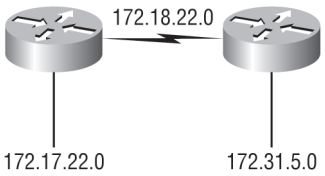
Daha önce söylediğim gibi, bu komutların üçü de, gateway of last resort oluşturacaktır, fakat aralarında bazı farklılıklar vardır. İlk olarak, çıkış interface çözümü, 0 AD'ye sahip olduğundan, diğer ikisinden daha çok düşünülür. Ayrıca, `ip default-network` komutu, bir IGP (RIP gibi) yapılandırdığınızda, default network'ü yayınlar. Bundan dolayı, ağ topluluğunuzdaki diğer router'lar, bu route'ı varsayılan olarak otomatik alır.

Şayet bir default route yanlış yapılandırılırsa, ne olur? Bir `show ip route` komut çıktısına bakalım, Şekil 6.11'deki network ile mukayese edelim ve bir problem bulup bulamayacağınızı görelim:

```
Router#sh ip route
[output cut]
```

```
Gateway of last resort is 172.19.22.2 to network 0.0.0.0

C       172.17.22.0 is directly connected, FastEthernet0/0
C       172.18.22.0  is directly connected, Serial0/0
S*      0.0.0.0/0 [1/0] via 172.19.22.2
```




***Şekil 6.11:*** *Yanlış yapılandırılmış default route.*

Bir şey bulabildiniz mi? Şekle ve routing tablosundaki direk bağlı route'lara bakarak, WAN linkinin, 172.18.22.0 network'ünde olduğunu ve default route'un tüm paketleri 172.19.22.0 network'üne gönderdiğini görebilirsiniz. Bu oldukça kötü, o, asla çalışmayacaktır. Problem, yanlış yapılandırılmış statik (default) route'dur.

Dinamik routing'e geçmeden önce son bir şey daha var. Şayet aşağıdaki gibi bir routing tablosu çıktısı varsa, router, 10.1.6.100 den 10.1.8.5 host'una hedeflenmiş bir paket alırsa ne olur?

```
Corp#sh ip route
[output cut]
Gateway of last resort is 10.1.5.5 to network 0.0.0.0

R     10.1.3.0 [120/1] via 10.1.2.2, 00:00:00, Serial 0/0
C     10.1.2.0  is directly connected, Serial0/0
C     10.1.5.0  is directly connected, Serial0/1
C     10.1.6.0  is directly connected, Fastethernet0/0
R*    0.0.0.0/0 [120/0] via 10.1.5.5, 00:00:00 Serial 0/1
```

Bu, şimdiye kadar yukarda gösterdiklerimden farklıdır. Çünkü default route, onun RIP-enjekte route olduğunu belirten `R*` olarak listelenmiştir. Bundan dolayı uzak bir router'da hem `ip default-route` komutunu kullanan hem de RIP yapılandıran birisi, RIP'in bu route'ı, ağ topluluğuna bir default route gibi yaymasına neden olur. Hedef adresi, 10.1.8.5 olduğundan ve 10.1.8.0 network'üne route olmadığından router, default route'u kullanacaktır ve paketi, serial0/1'e gönderecektir.

# Dinamik Routing

Dinamik routing, protokollerin, network'leri bulması ve router'lardaki routing tablolarının güncellenmesi için kullanılmaktadır. Doğru, bu, statik ve default routing kullanmaktan daha kolaydır, fakat bu size, network linklerindeki router CPU'su ve bant genişliği açısından pahalıya malolacaktır. Bir routing protokolü, komşu router'lar arasında routing bilgisini ilettiği zaman, bir router tarafından kullanılan kurallar bütününü tanımlamaktadır.

Bu modülde bahsedeceğim roting protokolleri, Routing Information Protocol (RIP) versiyon 1 ve 2 ile biraz da Interior Gateway Routing Protocol'dür (IGRP).

Ağ topluluklarında, iki tip routing protokolü kullanılmaktadır: Interior gateway protokolleri (IGP) ve exterior gateway protokolleri (EGP). IGP'ler, aynı autonomous system'de (AS)bulunan router'ların, routing bilgilerinin değiş tokuş edilmesi için kullanılmaktadır. Bir AS, ortak yönetilen bir domain'deki ağların topluluğudur. Bunun basit olarak anlamı şudur: aynı routing tablosunu

paylaşan tüm router'lar, aynı AS'dedirler. EGP'ler, AS'leri arasında iletişim için kullanılmaktadır. EGP'ye örnek, Border Gateway Protoco'dür l (BGP). (BGP, bu kitabın kapsamında değildir.)

Routing protokolleri, dinamik routing için çok gerekli olduğundan, ilerde bilmeniz gereken temel bilgileri vereceğim. Bu modülün ilerleyen bölümlerinde konfigürasyona odaklanacağım.

## Routing Protokol Temelleri

RIP'e detaylı olarak geçmeden önce, bilmeniz gereken önemli noktalar vardır. Özellikle, administrative distance, üç farklı routing protokolünü ve routing döngülerini anlamanız gerekmektedir. Bunların her birine aşağıdaki bölümlerde detaylı bakacağız.

### Administrative Distance

Administrative Distance(AD), komşu bir router'dan bir router'a alınan routing bilgisinin güvenirliğini derecelendirmek için kullanılır. Bir administrative distance, 0 ve 255 arası bir sayıdır. 0, en güvenilendir ve 255, bu route yoluyla trafiğin geçirilmeyeceği anlamına gelmektedir.

Şayet bir router, aynı uzak ağı listeleyen iki güncelleme alırsa, router'ın kontrol ettiği ilk şey, AD'dir. Advertised (yayınlanan) route'lardan birisi, diğerinde daha düşük AD'ye sahipse, en düşük AD'li route, routing tablosuna konacaktır.

Şayet, aynı ağ için iki advertised route, aynı AD'ye sahipse, uzak ağa en iyi yolu bulmak için, routing protokol metrikleri (hop count ya da hattın bant genişliği) kullanılacaktır. En düşük metriğe sahip advertised route, routing tablosuna konacaktır. Şayet iki advertised route, hem aynı AD'ye hem de aynı metriklere sahipse, o zaman routing protokolü, uzak network'e load-balance yapacaktır. (paketler her iki linkten de gönderilecektir).

Tablo 6.2, uzak bir ağa ulaşmak için hangi route'u kullanacağına karar vermek için bir Cisco router'ın kullandığı varsayılan administrative distance'ları göstermektedir.

**Tablo 6.2:** Varsayılan Administrative Distance'lar

| Route Kaynağı | Varsayılan Ad |
|---|---|
| Bağlı interface | 0 |
| Statik route | 1 |
| EIGRP | 90 |
| IGRP | 100 |
| OSPF | 110 |
| RIP | 120 |
| External EIGRP | 170 |
| Bilinmeyen | 255 (bu route asla kullanılmayacaktır.) |

Şayet bir network direk bağlıysa router, daima network'e bağlı interface'i kullanacaktır. Şayet bir statik route yapılandırırsanız, router bu route'un diğer öğrenilen route'lardan öncelikli olacağına inanacaktır. Statik route'ın administrative distance'ını değiştirebilirsiniz. Fakat varsayılan olarak AD'si, 1'dir. Statik route konfigürasyonlarımızda, her router'ın AD'sini 150 veya 151 olarak ayarladık. Bu bizim, routing protokollerini, statik route'ları kaldırmak zorunda kalmadan yapılandırmamıza izin verir. Onlar, routing protokollerinde bazı arızaların olması durumunda yedek route olarak kullanılacaktır.

Örnek olarak, aynı network'ü belirten bir statik route, RIP tarafından yayınlanan route ve IGRP tarafından yayınlanan route varsa, varsayılan olarak, router, (statik route'un AD'sini değiştirmediğiniz müddetçe) daima statik route'u kullanacaktır.

### Routing Protokolleri

Üç routing protokol sınıfı vardır:

**Distance vector:** Distance vector protokolleri, uzaklığı muhakeme ederek, uzak bir network'e en iyi yolu bulur. Bir paketin bir router'a uğradığı her sefer, hop olarak belirtilir. Network'e, en az sayıda hop ile giden route, en iyi route olarak kabul edilir. Vektör, uzak network'e yönü işaret eder. RIP ve IGRP, distance-vector protokolüdür. Bunlar, routing tablolarının tamamını, direk bağlı komşularına gönderirler.

**Link State:** shortest-path-first protokolleri olarak da bilinen link state protokollerinde, router'ların hepsi, üç ayrı tablo oluştururlar. Bu tablolardan birisi, direkt bağlı komşularının kayıtlarını tutar, diğeri, tüm ağ topluluğunun topolojisini oluşturur ve sonuncusu da, routing tablosu olarak kullanılır. Link-state router'lar, distance-vector routing tablosundan daha fazla bilgiye sahiptirler. OSPF, tamamıyla link-state olan bir IP routing protokolüdür. Link-state protokolleri, ağdaki diğer tüm router'lara linklerinin durumunu içeren güncellemeleri gönderirler.

**Hybrid:** Hybrid protokolleri, distance vector ve link state'in özelliklerini kullanmaktadır (örneğin EIGRP).

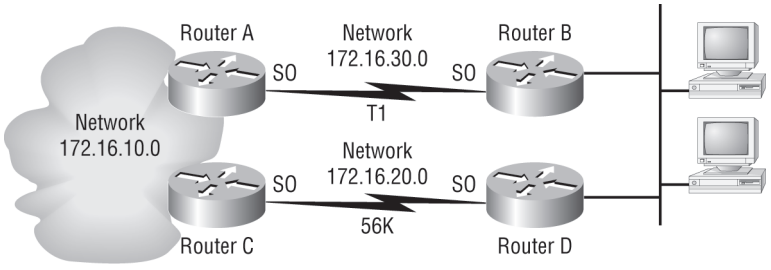
Her iş için, routing protokollerini yapılandırmanın belirlenmiş bir yolu yoktur. Bu, her bir durum için ayrı yapmanız gereken bir şeydir. Farklı routing protokollerinin nasıl çalıştığını bilirseniz, her işin kendi ihtiyaçlarına uygun iyi ve güvenilir kararlar verebilirsiniz.

## Distance-Vector Routing Protokolleri

Distance-vector routing algoritması, routing tablosunun içindekilerin tamamını komşu router'lara gönderir. Böylece, router'ların routing tablosunu tamamlamak için kendi routing tablolarıyla, alınan routing tablo kayıtları birleştirilir. Bu, kulaktan duyma routing olarak tanımlanır. Çünkü komşu router'dan güncelleme alan bir router, kendisi için olduğunu anlamaksızın uzak network'lerle ilgili bilgiye inanır.

Bir network'ün, aynı uzak ağa çok sayıda linke sahip olması mümkündür. Böyle olursa, ilk olarak, alınan her güncellemenin administrative distance'ı kontrol edilir. Şayet AD'ler aynı ise protokol, bu uzak ağa gitmek için kullanacağı en iyi yolu belirlemek için, diğer metrikleri kullanmak zorunda kalacaktır.

RIP, bir ağa giden en iyi yolu bulmak için sadece hop sayısını kullanır. Şayet RIP, aynı uzak ağa giden, aynı hop sayısında birden fazla link bulursa, otomatik olarak bir round-robin yük dengelemesi çalıştırır. RIP, altı eşit cost'a sahip linke kadar yük dengelemesi çalıştırabilir (varsayılan dört'tür).




***Şekil 6.12:*** *Pinhole tıkanıklığı.*

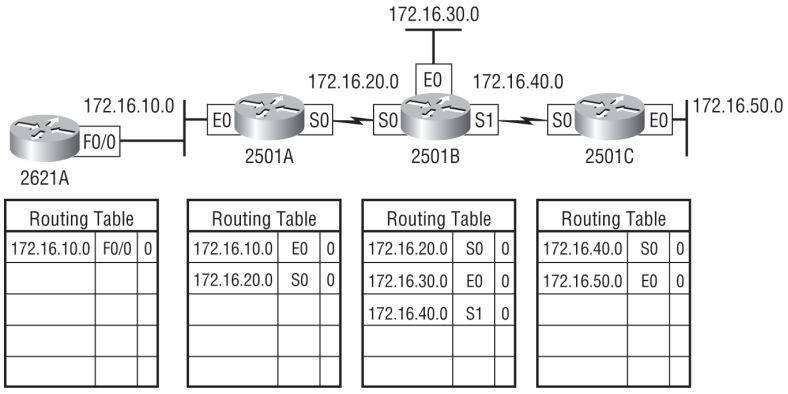
Bununla beraber, uzak network'e giden iki link, farklı bant genişliğine, fakat aynı hop sayısına sahipse, bu tip routing metrikleri problem oluşturur. Şekil 6.12, 172.16.10.0 uzak ağına, iki link göstermektedir.

172.16.30.0 network'ü, 1.544Mbps bant genişliğine sahip T1 linki ve 172.16.20.0, 56K bir link olduğundan router'ın 56K yerine T1 linkini seçmesini istersiniz, değil mi? Fakat hop sayısı, RIP

routing ile kullanılan tek metrik olduğundan, iki linkte, eşit cost değerli gibi görünür. Bu küçük problem, pinhole tıkanıklığı olarak tanımlanmaktadır.

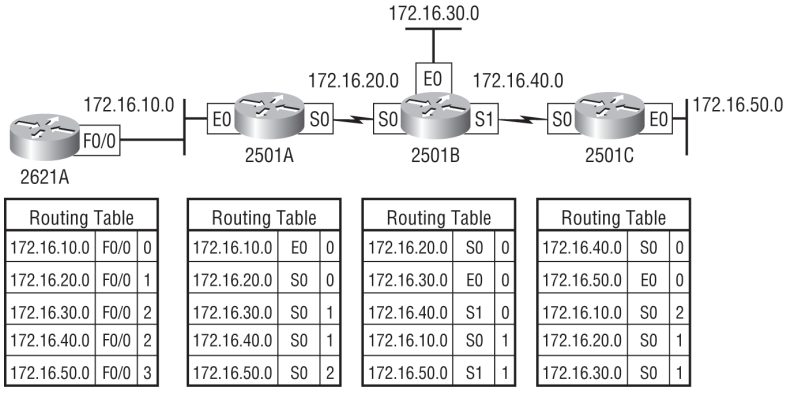
Çalışmaya başladığında bir distance-vector protokolünün ne yaptığını anlamak önemlidir. Şekil 6.13'te dört router, routing tablolarındaki direkt bağlı network'lerle çalışmaya başlar. Her router'da, distance-vector routing protokolü çalışmaya başladıktan sonra routing tabloları, komşu router'lardan toplanan route bilgileriyle güncellenir.

Şekil 6.13'te gösterildiği gibi her router, sadece direkt bağlı network'lere sahiptir. Her router, kendi routing tablosunun tamamını, aktif interface'lerine gönderir. Her router'ın routing tablosu, network numarasını, çıkış interface'ini ve network için hop sayısını içerir.




***Şekil 6.13:*** *Distance vector routing kullanan ağ topluluğu.*

Şekil 6.14'de, ağ topluluğundaki tüm network'ler hakkında bilgi içerdiği için routing tabloları tamamlanmıştır. Onlar, converged kabul edilirler. Router'lar, converge olurlarken, verilerin aktarılmaması mümkündür. Bu sebeple, hızlı convergence zamanı önemli bir artıdır. Aslında, RIP'le yaşanan problemlerden birisi, düşük convergence zamanı olmasıdır.



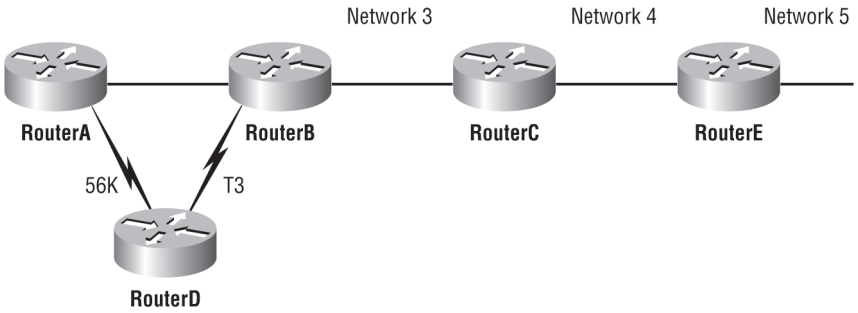

***Şekil 6.14:*** *Converged routing tabloları.*

Her router'daki routing tablosu, uzak network'ün adresi, router'ın network'e ulaşması için paketleri gönderdiği interface ve network için hop sayısı ya da metrik hakkındaki bilgileri tutar.

## Routing Kısır Döngüleri

Distance-vector routing protokolleri, aktif interface'lerinden periyodik routing güncellemelerini broadcast ederek, ağ topluluğundaki değişiklikleri takip eder. Bu broadcast, routing tablosunun tamamını içerir. Bu, oldukça iyi çalışır, fakat CPU prosesi ve linkin bant genişliği bakımından pahalıdır. Şayet network'te bir kesinti olursa, gerçekten sıkıntı verici problemler çıkabilir. Artı, distance-vector protokollerinin düşük convergence süresi, tutarsız routing tablolarına ve routing kısır döngülerine sebep olabilir.

Her router eşzamanlı, hatta yakın zamanda güncellenmezse routing kısır döngüleri olur. İşte bir örnek: Şekil 6.15'deki, Network 5'e bağlı interface'in arızalandığını düşünelim. Tüm router'lar Network 5 hakkındaki bilgiyi RouterE'den öğrenir. RouterA, tablosunda, Network 5'e RouterB üzerinden bir yola sahiptir.




***Şekil 6.15:*** *Routing kısır döngü örneği.*

Network 5 arızalandığında, RouterE, RouterC'ye söyler. Bu, RouterC'nin, RouterE üzerinden Network 5'e routing yapmasını durdurmasına sebep olur. Fakat RouterA, B ve D, henüz Network 5'in arızalandığını bilmezler. Bu nedenle, güncelleme bilgisi göndermeye devam ederler. RouterC sonunda güncellemesini gönderecektir ve RouterB, Network5'e routing yapmayı durduracaktır. Fakat RouterA ve D, hala güncel değildir. Onlara göre, Network 5, RouterB üzerinden 3 metriğiyle hala erişilebilir görünmektedir.

RouterA, kendi düzenli 30-saniye "Hello, ben hala buradayım" mesajını gönderir. Bunlar, hakkında bilgi sahibi oldukları linklerdir ve Network5'e ulaşabilmeyi de içermektedir. Şimdi, RouterB ve D, Network 5'e, RouterA'dan erişilebileceği haberini alırlar ve böylece, RouterB ve D, Network 5'in erişilebilir bilgisini gönderirler. Network 5 için hedeflenen her paket, RouterA'ya, RouterB'ye ve sonra tekrar RouterA'ya gidecektir. Bunun adı, routing kısır döngüsüdür. Bunu nasıl durdursunuz?

## Maximum Hop Count

Şimdi açıklanan routing kısır döngü problemi, counting to infinity (sonsuza saymak) olarak tanımlanır ve ona, dedikodu (broadcast) ile ağ topluluğu üzerinde iletilen ve yayılan yanlış bilgi sebep olur.

Bazı araya girme yöntemleri olmaksızın hop sayısı, bir paketin bir router'a uğradığı her sefer artarak devam eder.

Bu problemi çözmenin bir yolu, maximum hop count tanımlamaktır. RIP, 15'e kadar hop sayısına izin verir. Böylece, 16 hop, erişilemez kabul edilir. Diğer bir deyişle, 15 hop'lu bir döngüden sonra, Network5, arızalı kabul edilir. Böylece, maximum hop count, bir routing tablosu kaydının, geçersiz ya da şüpheli olması için ne kadar gideceğini kontrol eder.

## Split Horizon

Routing kısır döngü problemine diğer bir çözüm, split horizon olarak belirtilir. Bu, yanlış routing bilgisini ve (routing bilgisinin, alındığı yönden gönderilemeyeceği kuralını mecbur tutarak), distance vector network'te routing ek yükünü düşürür.

Diğer bir deyişle, routing protokolü bir network route'nun öğrenildiği interface'i tespit eder. Bu belirlenince route, aynı interface'den gönderilmeyecektir. Bu, RouterA'nın RouterB'den aldığı güncel bilgiyi RouterB'ye tekrar göndermesini engelleyecektir.

## Route Poisoning

Tutarsız güncellemelerin sebep olduğu problemlerden kaçınmanın ve network kısır döngülerini durdurmanın diğer yolu, route poisoning'tir. Örnek olarak, Network 5 gittiğinde, RouterE, Network

5'i 16.cı hop ya da erişilemez (bazen *sonsuz* olarak belirtilir) olarak yayınlayarak, route poisoning başlatır.

Network 5'e giden route'ın zehirlenmesi RouterC'nin, Network 5'e giden route hakkında yanlış güncellemelerden etkilenmesini sağlar. RouterC, RouterE'den bir route poisoning aldığında, *poison reverse* olarak belirtilen bir güncellemeyi tekrar RouterE'ye gönderir. Bu, segment'teki tüm router'ların zehirli route bilgisi almalarını kesinleştirir.

### Holddown

Bir holddown, düzenli güncelleme bilgilerinin, up ve down olan (flapping denir) bir route'da tekrarlanmasını engeller. Genellikle, bağlantısı kaybolup tekrar gelen seri linklerde olur. Bunu tespit etmek için bir yol yoksa network asla converge olmayacak ve bu gidip gelen interface, tüm network'ün down olmasına sebep olacaktır.

Holddown'lar, sonraki en iyi route ile değiştirmeden önce, down olan route'un tekrar up olması ya da network'ün dengelenmesi için bir süre geçmesine izin vererek, route'ların çok hızlı değişmesini engellerler. Bu, router'lara, belirli bir zaman dilimi için, yakınlarda kaldırılan route'ların etkilenebileceği değişiklikleri kısıtlamasını da söyler. Bu, çalışmayan route'ların, diğer router tablolarına vaktinden önce konmasını engeller.

## Routing Information Protocol (RIP)

Routing Information Protocol (RIP), gerçek bir distance-vector routing protokolüdür. RIP, routing tablosunun tamamını her 30 saniyede tüm aktif interface'lerine gönderir. RIP, uzak bir ağa en iyi yolu belirlemek için sadece hop sayısını kullanır. Fakat varsayılan olarak, maksimum 15 kabul edilebilir hop sayısına sahiptir. Yani, 16 erişilemez kabul edilmektedir. RIP, küçük network'lerde iyi çalışır, fakat düşük WAN linklerine sahip, geniş ağlarda ve çok sayıda router'ın kurulduğu network'lerde yetersizdir.

RIP versiyon1, sadece classfull routing kullanırlar. Yani, network'teki tüm cihazlar, aynı subnet maskını kullanmak zorundadır. Bundan dolayı, RIP versiyon1, beraberinde subnet mask içeren güncelleme göndermez. RIP versiyon2, prefix routing sağlar ve route güncellemeleriyle subnet masklarını gönderir. Bu, classless routing olarak belirtilir.

Aşağıdaki bölümlerde, RIP timer'ları ve sonra da RIP yapılandırmasını tartışacağız.

### RIP Timer'ları

RIP, performansını düzenlemek için, dört farklı timer kullanır:

**Route update timer:** Router'ın tüm komşularına, routing tablosunun tam kopyasını gönderdiği periyodik routing güncellemeleri arasındaki zaman aralığını ayarlar (genelde 30 saniyedir).

**Route invalid timer:** Bir router'ın, bir route'u geçersiz kabul etmesinden önce geçmesi gereken zamanın uzunluğunu belirler (180 saniye). Bu sonuca, bu periyotta belirli bir route hakkında herhangi bir güncelleme almayarak ulaşır. Bu olduğunda router tüm komşularına, route'un geçersiz olduğunu bilmelerini sağlayan güncellemeleri gönderecektir.

**Holddown timer:** Routing bilgilerinin kesildiği sıradaki zaman miktarını ayarlar. Route'un erişilemez olduğunu belirten bir güncelleme paketi alındığında, router'lar, holddown durumuna geçer. Bu, daha iyi metriğe sahip bir update paketi alınana ya da holddown timer süresi dolana kadar devam eder. Varsayılanda 180 saniyedir.

**Route flush timer:** Bir route'un geçersiz olması ile onun routing tablosundan çıkartılması arasındaki zamanı ayarlar (240 saniye). Tablodan çıkartılmadan önce, router, bu router'ın ölmesinin yakın olduğunu komşularına duyurur. Route invalid timer değerinin, route flush timer dan daha küçük olması gerekir. Bu router'a, lokal routing tablosu güncellenmeden önce, geçersiz route hakkında komşularının bilgilendirilmesi için yeterli zamanı sağlar.

## RIP Routing Konfigürasyonu

RIP routing konfigürasyonu için, `router rip` komutu ile protokolü açın ve RIP routing protokolüne, hangi network'lerin yayınlanacağını söyleyin. Hepsi bu kadar. Şimdi, bizim 5-router'lu ağ topluluğumuzu (Şekil 6.9), RIP routing ile yapılandıralım.

### Corp

RIP, 120 administartive distance'a sahiptir. Statik route'lar, varsayılanda 1 AD'ye sahiptir ve biz yapılandırılmış statik route'lara sahip olduğumuzdan, routing tabloları, RIP bilgileri ile dağıtılmayacaktır. Bununla beraber, her statik route'un sonuna 150/151'i eklediğimden, RIP'i kullanabiliriz.

RIP routing protokolünü, router rip ve network komutlarını kullanarak ekleyebilirsiniz. Network komutu, routing protokolüne, hangi classful network'ün yayınlanacağını söyler.

Corp router konfigürasyonuna bakın ve bunun ne kadar kolay olduğunu anlayın:

```
Corp#config t
Corp(config)#router rip
Corp(config-router)#network 10.0.0.0
```

Bu kadar. İki veya üç komut ve tamamladınız. Statik route kullanmaktan daha kolay olduğu kesin, değil mi? Yine de, fazladan router CPU prosesi ve bant genişliği kullandığınız aklınızda olsun.

Subnet'leri yazmadığıma dikkat edin. Sadece classful network adresleri vardır.(subnet ve host bit'lerinin hiçbiri kullanılmaz). Subnet'leri bulmak ve routing tablolarını yerleştirmek, routing protokolünün görevidir. RIP çalışan bir router'ımız olmadığından, routing tablosunda henüz bir RIP güncellemesi göremiyoruz.

NOT

*Network adresleri yapılandırıldığında, RIP'in, classful adres kullandığını hatırlayın. Bundan dolayı, network'teki tüm cihazlardaki subnet mask aynı olmalıdır (bu classful routing olarak tanımlanır). Bunu açıklamak için 172.16.10.0, 172.16.20.0 ve 172.16.30.0 subnet'leri ile 172.16.0.0/24 ClassB network adresi kullandığınızı farz edelim. Sadece 172.16.0.0 classful network adresini yazın, RIP'in subnet'leri bulmasını ve onları routing tablosuna yerleştirmesini sağlayabilirsiniz.*

### R1

R1 router'ımızı yapılandıralım:

```
R1#config t
R1(config)#router rip
R1(config-router)#network 10.0.0.0
R1(config-router)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
S       10.1.11.0 [150/0] via 10.1.3.1
S       10.1.10.0 [150/0] via 10.1.3.1
S       10.1.9.0 [150/0] via 10.1.3.1
S       10.1.8.0 [150/0] via 10.1.3.1
S       10.1.12.0 [150/0] via 10.1.3.1
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
R       10.1.1.0 [120/1] via 10.1.3.1, 00:00:04, Serial0/0/1
                 [120/1] via 10.1.2.1, 00:00:04, Serial0/0/0
C       10.1.7.0 is directly connected, FastEthernet0/1
C       10.1.6.0 is directly connected, FastEthernet0/0
R       10.1.5.0 [120/1] via 10.1.3.1, 00:00:04, Serial0/0/1
```

```
                [120/1] via 10.1.2.1, 00:00:04, Serial0/0/0
R       10.1.4.0 [120/1] via 10.1.3.1, 00:00:09, Serial0/0/1
                [120/1] via 10.1.2.1, 00:00:09, Serial0/0/0
R1(config-router)#
```

Bu oldukça basittir. Bu routing tablosundan biraz bahsedelim. Routing tablomuzu değiş tokuş ettiğimiz bir tane RIP komşumuz olduğundan, Corp router'ından gelen RIP ağlarını görebiliriz. (Diğer tüm route'lar, hala statik görünmektedir.) RIP, Corp router'ına her iki bağlantıyı da bulacak ve onlar arasında yük-paylaşımı yapacaktır.

## R2

R2 router'ımızı RIP ile yapılandıralım:

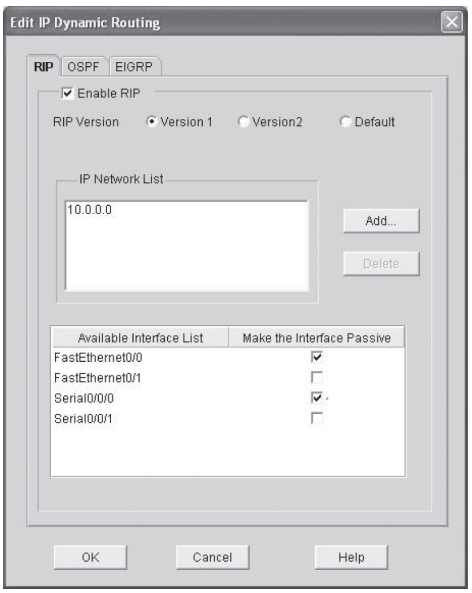
```
R2#config t
R2(config)#router rip
R2(config-router)#network 10.0.0.0
R2(config-router)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
S       10.1.11.0 [150/0] via 10.1.4.1
S       10.1.10.0 [150/0] via 10.1.4.1
C       10.1.9.0 is directly connected, FastEthernet0/0
C       10.1.8.0 is directly connected, Dot11Radio0/3/0
S       10.1.12.0 [150/0] via 10.1.4.1
R       10.1.3.0 [120/1] via 10.1.4.1, 00:00:03, Serial0/2/0
R       10.1.2.0 [120/1] via 10.1.4.1, 00:00:03, Serial0/2/0
R       10.1.1.0 [120/1] via 10.1.4.1, 00:00:03, Serial0/2/0
R       10.1.7.0 [120/2] via 10.1.4.1, 00:00:03, Serial0/2/0
R       10.1.6.0 [120/2] via 10.1.4.1, 00:00:03, Serial0/2/0
R       10.1.5.0 [120/1] via 10.1.4.1, 00:00:03, Serial0/2/0
```

RIP komşularını ekledikçe, routing tablosu R'lerle büyümektedir. Routing tablosundaki route'lardan bazılarının hala statik olduğunu görebiliyoruz. Tamamlamak için iki router kaldı.

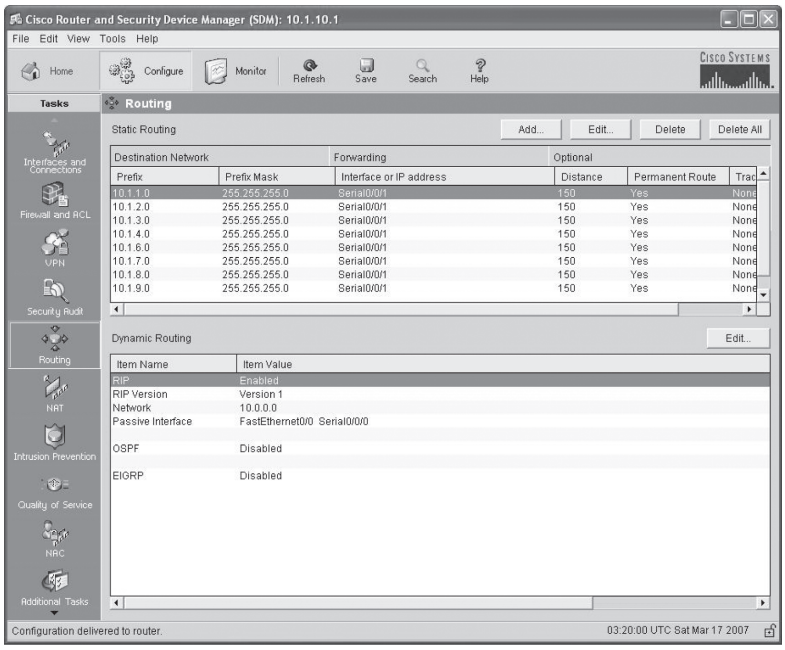
## R3

R3 router'ımızı, RIP ile yapılandıralım, alışıldığı gibi R3'te SDM kullanacağız.

Routing ekranından, Dynamic Routing'in sağındaki Edit butonuna tıkladım. Şimdi, RIP'i ve network adreslerini yapılandırabiliyorum. RIP'in broadcast edilmesini istemediğim interface'lere tıkladım. RIP'in broadcast edileceği interface'lere işaret konulmadı.



Bunlar, pasif interface olarak tanımlanır ve kısa bir süre sonra bahsedeceğim. Router'ların olmayacağı bir interface'den RIP broadcast'i göndermenin anlamı yoktur.




SDM ekranından, R3 konfigürasyonunu tamamladığımızı görebiliriz.

## 871W

Son router'ın RIP konfigürasyonu şöyledir:

```
871W#config t
871W(config)#no ip route 0.0.0.0 0.0.0.0 10.1.11.1
871W(config)#router rip
871W(config-router)#network 10.0.0.0
871W(config-router)#do sh ip route
     10.0.0.0/24 is subnetted, 12 subnets
C       10.1.11.0 is directly connected, Vlan1
R       10.1.10.0 [120/1] via 10.1.11.1, 00:00:23, Vlan1
R       10.1.9.0 [120/3] via 10.1.11.1, 00:00:23, Vlan1
R       10.1.8.0 [120/3] via 10.1.11.1, 00:00:23, Vlan1
C       10.1.12.0 is directly connected, Dot11Radio0
R       10.1.3.0 [120/2] via 10.1.11.1, 00:00:23, Vlan1
R       10.1.2.0 [120/2] via 10.1.11.1, 00:00:23, Vlan1
R       10.1.1.0 [120/2] via 10.1.11.1, 00:00:23, Vlan1
R       10.1.7.0 [120/3] via 10.1.11.1, 00:00:24, Vlan1
R       10.1.6.0 [120/3] via 10.1.11.1, 00:00:24, Vlan1
R       10.1.5.0 [120/1] via 10.1.11.1, 00:00:24, Vlan1
R       10.1.4.0 [120/2] via 10.1.11.1, 00:00:24, Vlan1
871W#
```

Son olarak, routing tablosunda görünen tüm route'lar, RIP-enjekte route'lardır.

Administrative distance'ları ve RIP routing'i eklemeden veya onları bizim yaptığımız gibi 120'den yüksek yapmadan önce, statik route'ları çıkartmamız gerektiğini hatırlamak önemlidir.