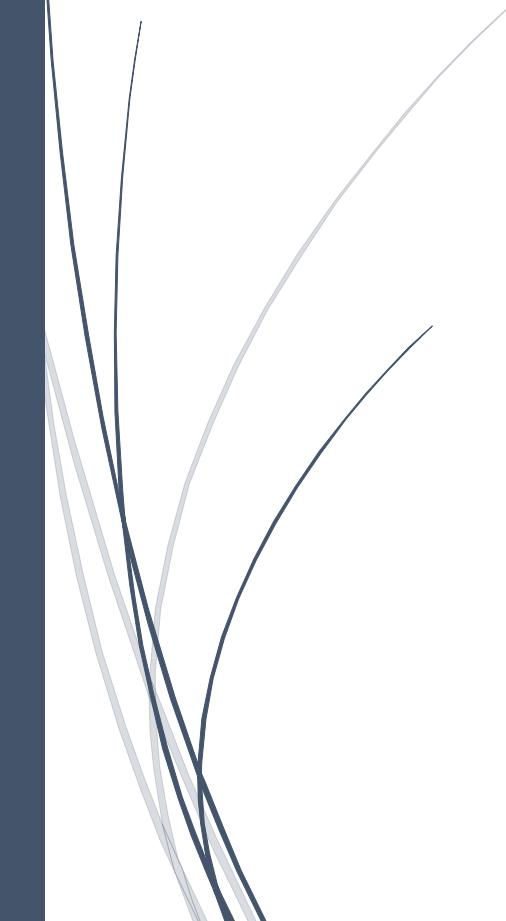




NETWORK DERS NOTLARI



Dr.Erdal ÖZDOGAN
2020

TEMEL KAVRAMLAR

IP ADRESİ NEDİR

Gerçek hayatı iletişim kurarken uymak zorunda olduğumuz kurallar olduğu gibi, bilgisayar ağlarında da cihazların kendi aralarında konuşurken uymak zorunda olduğu kurallar vardır. Bilgisayar ağları(network) dünyasındaki bu kuralları protokoller belirler. Bir iletişim söz konusu olduğunda aynı dili konuşmak, anlamak ve anlaşılmak için ne kadar önemliyse cihazların haberleşmesinde de aynı dili kullanmak o derece önemlidir. İnsanlar arasındaki iletişim esnasında; dil, konuşma hızı, cümle yapısı, konuşma sırasını bekleme, karşısındakiin sözünü kesmeme, anlaşılmadığı durumda tekrar etme, anlaşıldığına dair karşidan bir onay beklemeye gibi kuralları bilgisayar ağları dünyasına uyarlamak ve bu iletişimini alıcı ile gönderici açısından anlaşılır bir hale getirmekten sorumlu olan kurallar bütünü **protokol** olarak tanımlanabilir. Kısacası protokoller, farklı üreticilerin farklı ürünlerinin aynı platformda iletişimini yani standardizasyonu sağlar.

Bilgisayar ağları söz konusu olduğunda birçok üretici firmanın ve ürününün olması kaçınılmazdır. Standartlar olmadan önce, her üretici sadece kendi cihazları arasındaki iletişimini destekliyor, farklı üreticilerin ürünleri ile iletişim kurmakta zorlanıyor ya da kuramıyordu. Bu kısımda, bu protokollerden TCP/IP protokol kümesine ve bir takım temel kavamlara değineceğiz.

Bugünlerde pek posta ile haberleşme kullanılmasa da ağ cihazları arasındaki iletişimini anlaşılabilmesi için iyi bir örnek olacaktır. Uzak bir şehirdeki arkadaşınızla posta yoluyla haberleştiğinizi düşünün. Bu arkadaşınıza mektup yollamadan önce, onun konumunu belirleyen bir adres bilgisine ihtiyaç duyarsınız. Arkadaşınızın ismini ve adresini zarfin üzerine yazar, zarfin hedefe ulaşması için postaneye verirsiniz. Daha sonraki işlemlerin neler olduğu ilgimizi çekmez. Ancak mektubunuzun hedefine ulaşabilmesi için farklı illerdeki / ilçelerdeki postaneler arasında dolaşması gerekecek ve son postane de bir dağıtıci aracılığıyla mektubunuza alıcıya ulaştıracaktır. İşte insanlar arasındaki iletişimde adres kavramına karşılık, ağ cihazları arasındaki **bu mantıksal adres** kavramını **internet protokolü (IP)** karşılar. Yalnız adres bilgisindeki cadde, sokak gibi kavamlar yerine ağ cihazları arasındaki iletişimde sayıları kullanırız. Bu sayılar **binary (ikilik sayı)** ile gösterilir. Burada bir kısıtlama olarak sayıda 32 karakter kullanılır. Başka bir deyişle IP adresleri (Versiyon 4) 32 bitten oluşur. Örneğin bir IP adresi aşağıdaki gibi olabilir;

1100000010101000000010100000011

Tabi bu adresi kişinin okuyabilmesi ve aklında tutabilmesi zor olduğu için her biri 8 bit (1 byte) olan 4 kısma (oktet) ayırır ve aralarına birer nokta koyarız.

11000000.10101000.00000101.00000011

Günlük hayatı onluk sayı düzenini kullandığımız için bu gruplandırma kolaylık sağladığı söylenemez. Bu adresi daha okunaklı bir hale getirelim ve her bir oktetin onluk sayı sistemine dönüştürelim:

192. 168. 5. 3

Eskisine göre biraz daha göze hitap ettiği söylenebilir. Bu adres üzerinde biraz matematiksel işlem yapalım ve toplamda ne kadar adres oluşturabileceğimizi görelim. İkilik sayıda bu 32 karakterimiz ile toplamda 2^{32} (4 milyardan fazla) adres tanımlayabiliriz. 4 milyar adres büyük bir oran olmakla beraber daha sonra değinilecek birkaç sebepten dolayı bu adreslerin tümünü kullanamadığımızı göreceğiz. IP adres kavramı ilk oluştuğunda bu sayı yeterli görülmüyordu ancak bugün dünya üzerinde adreslenmesi gereken cihaz sayısı düşünüldüğünde bu sayının aslında yetersiz olduğu tespit edilmiştir. Yani günümüzde IP adresi kıtlığı yaşanmaktadır. Çünkü iletişimde her cihazın sadece kendisini gösteren tekil birer adresi olmak zorundadır. Yine daha sonra bahsedilecek olan birtakım çalışmalar ile bu adres sıkıntısı giderilmeye çalışılmış ve daha fazla adres tanımlayabilecek yeni sürüm IP adresi devreye girmiştir. Bu yeni sürüm IP adresi, IPv6 olarak bilinir ve 128 bitten oluşur. Yani bit sayısı 4 katına, tanımlanabilen adres sayısı da 2^{96} katına çıkacaktır.

IP ADRES TÜRLERİ

İnsanlar arasındaki iletişimde hitap ettiğiniz kesim her zaman aynı olmayabilir. Bazen bir topluluk karşısında konuşurken bazen de bir kişi ile ya da birkaç kişi ile konuşabilirsınız. Bilgisayar ağlarında da bu durum söz konusudur.

Eğer iletişimdeki hedef adres sadece bir cihazsa **tekli yayın (Unicast)**;

Ortamda bulunan tüm cihazlarsa **yayın (broadcast)**;

Ortamdaki belli bir grup cihazsa **çoklu yayın (multicast)** adını alır.

IP ADRES SINIFLARI

Günümüzde kullanılan IP adres sürücümüne (IPv4) dönelim ve biraz yapısını inceleyelim. IP adresleri 5 sınıfa ayrılır:

- a) A sınıfı adresler, IP adreslerinin % 50 sine denk gelir ve ilk 8 bitlik (oktet) değerinin ondalık karşılığı 1 ile 127 arasına denk gelir.

Örneğin; **95.120.130.240** adresi A sınıfı bir adresdir. Çünkü ilk oktet değeri 1 ile 127 arasındadır. **127.0.255.16** adresi yine A sınıfı bir adresdir.

A sınıfı adreslerin ilk oktetleri ikilik sayıda yazılırsa,

00000001 – 01111111 aralığında olması gereklidir.

(1 – 127)

- b) B sınıfı adreslerin ilk oktetleri 128 ile 191 arasındadır ve Toplam IP adres sayısının %25ini kapsar.

Örneğin; **130.34.0.200**

İkilik sayıda ilk oktet,

10000000 – 10111111 aralığına denk gelir.

- c) C sınıfı adreslerin ilk oktetleri 192 ile 223 arasındadır.

Örneğin; **192.168.5.3**

İkilik sayıda ilk oktet,

11000000 – 11011111 aralığına denk gelir.

- d) D sınıfı adresler çoklu yayın (multicast) adresleri olarak bilinir ve ilk oktetleri 224 – 239 arasındadır.

Örneğin; **224.0.0.9** bir multicast adresidir.

İkilik sayıda ilk oktet,

11100000 – 11101111 aralığına denk gelir.

e) E sınıfı adresler, özel amaçlı kullanılan adresler olup ilk oktetleri 240 – 255 arasındadır.

Örneğin; **249.0.0.4**

İkilik sayıda ilk oktet,

11110000 – 11111111 aralığına denk gelir.

IP adresleri hiyerarşik bir yapıya sahiptir. Örnek olması açısından telefon numaralarındaki yapıyı düşünelim. Ankara'daki tüm telefon numaraları 0312 ile başladığını, hatta belirli bir semtteki telefon numaralarının belirli karakterlerinin aynı olduğunu biliyoruz. 0312 212..... numarası ile başlayan kullanıcının XYZ semtinde olduğunu anlayabiliriz. Aynı yapıya benzer olarak da, belirli bir ağdaki cihazların IP adreslerinin bir kısmı bağlı bulunduğu ağı (**network kısmı**), diğer bir kısmı ise cihazın kendisini (**host kısmı**) gösterir. Yani *aynı ağa bulunan cihazların IP adreslerinin ağ kısmı tüm cihazlarda aynı iken, host kısmı farklıdır*. Bu yapı ağın büyülüğüne göre değişkenlik gösterebilir. Ancak varsayılan olarak bu adreslerin ağ ve host kısımları, IP adres sınıflarına göre bellidir.

A sınıfı adreslerde ilk oktet ağ kısmını gösterirken son üç oktet host kısmını gösterir. Yani host kısmı için 24 bit ayrılmıştır. O halde A sınıfı bir ağa varsayılan olarak 2^{24} adres oluşturulabilir. Ancak, host kısmının tümünün 0 olması durumu (ağın kendisi tanımlayan bir özel adres) ile host kısmının tümünün 1 olması durumu (ağdaki tüm cihazlara yapılan yayın adresi) cihazlara atanır. Geçerli birer adres olmayacağından, o halde A sınıfı bir ağa cihazlara atayabileceğimiz adres sayısı $2^{24} - 2$ olacaktır.

Örnek, 10.X.X.X IP aralığına sahip bir ağa aşağıdaki iki adres cihazlara atanamayacaktır.

Host kısmının tümünün 0 olması durumunda oluşacak adres, 10.0.0.0 (ağ adresi)

Host kısmının tümünün 1 olması durumunda oluşacak adres, 10.255.255.255 (broadcast adres)

10.1.3.5 gibi bir adres için;

10 – Ağ Kısımlı

1.3.5 – Host kısmını gösterir.

10.1.3.5 IP adresine sahip bir cihaz ile aynı anda bulunan başka bir cihazın adresi 10.2.4.6 olabilir. Çünkü ağ kısmını her ikisinde de aynıdır.

B sınıfı adreslerde ilk iki oktet ağ kısmını gösterirken son iki oktet host kısmını gösterir.

Örnek;

172.16.1.2

C sınıfı adreslerde ilk üç oktet ağ kısmını, son oktet host kısmını gösterir

Örnek;

192.168.1.5

192.168.1.5

C sınıfı yukarıdaki adres için altı çizili kısım ağ kısmını gösterirken altı çizili olmayan kısım host kısmını gösterir. Şimdi de bu ifadeyi bitler bazında yazalım.

11000000.10101000.00000001.01000101

Ancak cihazımız “altı çizili” ifadesini anlamayacağı için 32 bitlik bir değişken kullanıp, bu değişkenin durumuna göre (0 veya 1) IP adresinde karşılık gelen bitin ağ kısmı mı yoksa host kısmı mı olduğunu belirleyebiliriz. Yani;

IP ADRESİMİZ: 11000000.10101000.00000001.01000101

DEĞİŞKENİMİZ: 1111111.1111111.1111111.0000000 olsun.

Cihaz IP adresinin her bitini, değişkenimizdeki bit ile sırasıyla eşleştirecek, değişkenin 1 olduğu durumda IP adresinin aynı sıradaki bitinin ağ kısmına ait olduğunu anlayacaktır.

IP adres yapısındaki, ağ ve host kısımlarını bulmak için kullanılan bu değişkene ALT AĞ MASKESİ (Subnet Mask) denir.

Cihazımız bu IP adresinin host ve ağ kısmını öğrendiğine göre sıra bizim daha rahat okuyacağımız şekle yani onluk sisteme dönüştürmeye geliyor.

Örneğin;

IP ADRESİMİZ : 192.168.1.5

ALT AĞ MASKESİ : 255.255.255.0

Alt ağ maskesi / ifadesi ile de gösterebiliriz. Alt ağ maskesinde 1 olan değerlerin sayısını / işaretinden sonra yazarız. Örneğin;

IP Adresi 192.168.1.5 , Subnet Mask : 255.255.255.0 olan bir ifadeyi: 192.168.1.5 / 24 olarak gösterebiliriz. Buradaki 24 ifadesi subnet mask adresindeki 1 olan bitlerin sayısını gösterir.

Tablo: IP Adres Sınıfları, ilk oktetler, varsayılan alt ağ maskesi

SINIF	IP Adres aralığı	İlk oktet (decimal)	İlk oktet (binary)	Varsayılan Alt Ağ Maskesi	Örnek
A	1.0.0.0 - 127.255.255.255	1 – 127	00000001 01111111	255.0.0.0	100.15.14.13 /8
B	128.0.0.0 – 191.255.255.255	128 – 191	10000000 10111111	255.255.0.0	160.14.15.16 /16
C	192.0.0.0 – 223.255.255.255	192 – 223	11000000 11011111	255.255.255.0	192.168.1.5 / 24
D	224.0.0.0 239.255.255.255	224 – 239	11100000 11101111	—	224.0.0.9
E	240.0.0.0 – 255.255.255.255	240 – 255	11110000 11111111	—	240.3.4.5

REZERVE EDİLMİŞ IP ADRESLERİ

IP adreslerinin dağıtımından IANA (Internet Assigned Numbers Authority) sorumludur. IANA, Regional Internet Registry (RIR)'ler aracılığıyla bu IP adreslerini dağıtır. Örneğin, AfriNIC, Afrika kıtasındaki IP adreslerinin dağıtımından sorumlu olan bir RIR'dır.



IP adreslerinin internet üzerinde benzersiz olması gerektiğini belirtmiştık. Ancak IPv4 adreslerinin kısıtlı olmasından dolayı, ağdaki her cihaza benzersiz IP adresi ataması imkânsız hale gelmiştir. Bu yüzden bir takım çalışmalar başlamış ve sadece yerel ağa kullanmak için belli grup IP adresleri rezerve edilmiştir. Bu IP adresleri, RIR'ler tarafından dağıtılmayan özel IP adresleridir (Private IP Addresses).

Bu adresler;

- 10.0.0.0 ile 10.255.255.255 arasındaki A sınıfı IP adresleri
- 172.16.0.0 ile 172.31.255.255 arasındaki B sınıfı IP adresleri
- 192.168.0.0 ile 192.168.255.255 arasındaki C sınıfı IP adresleridir.

Ayrıca bu adresler dışında, 169.254.0.0 ile 169.254.255.255 arasındaki adresler, ortamda bir DHCP sunucu bulunmadığı durumlarda işletim sistemi tarafından cihaza atanan adres grupperidir.

Yine, cihazdaki TCP /IP protokol takımının doğru bir şekilde çalıştığını testi amaçlı 127.0.0.0 ile 127.255.255.255 arasındaki IP adresleri loopback test adresi olarak kullanılır ve RIR ler tarafından atanamazlar.

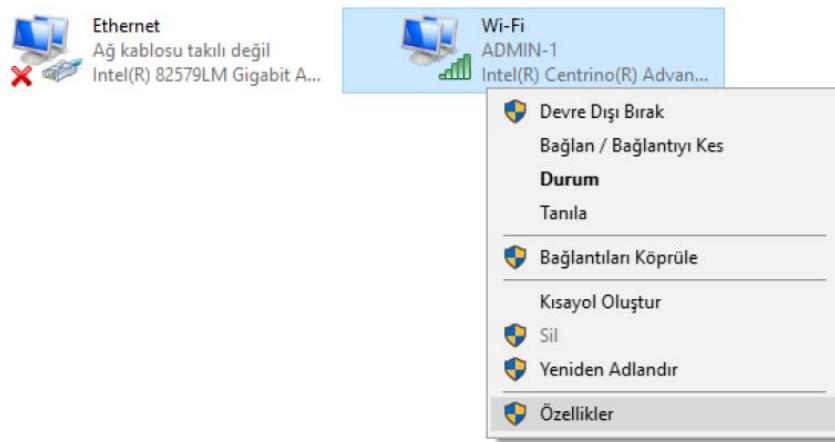
Ayrıca, 192.0.2.0 / 24 ağ aralığı TEST-NET adresleri olarak ayarlanmıştır. Bu IPler dokümantasyonda ve eğitimlerde kullanılır.

IP ADRES ATAMASI

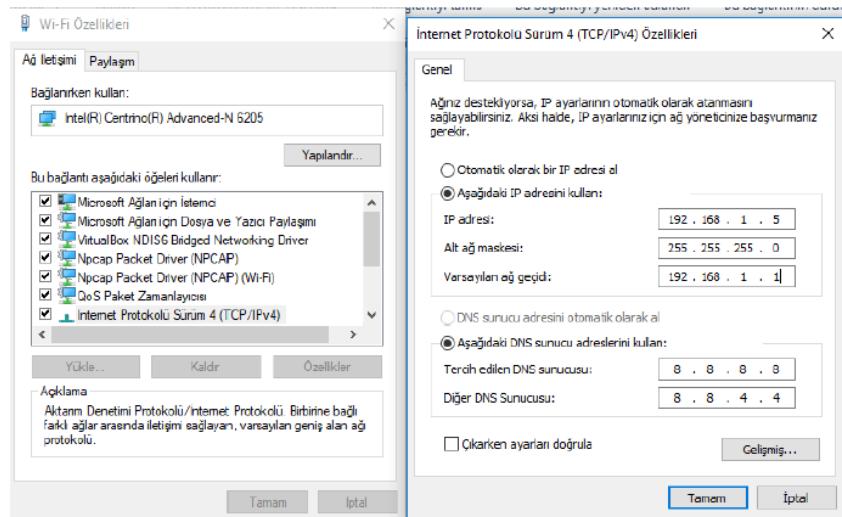
Cihazlara IP ataması statik ve dinamik olmak üzere iki şekilde yapılabilir. Statik IP ataması yapılrken, ağdaki benzersiz IP adresini, SubnetMask, Default Gateway (Varsayılan Ağ Geçidi) gibi bilgileri cihaza statik olarak atamak gereklidir.

Windows işletim sistemi yüklü bir bilgisayara IP adresi ve diğer gerekli parametreleri vermek için Şekil'de gösterildiği gibi statik atama işlemi gerçekleştirilebilir.

Denetim Masası\Tüm Denetim Masası Öğeleri\Ağ Bağlantıları yolu aracılığıyla IP ataması yapılması istenen Ethernet bağdaştırıcısı seçilir.



İlgili arayüze sağ tıklanır ve **Özellikler** seçilir. Gelen ağ bağıdaştırıcısı özellikleri ekranından **Internet Protokolü Sürüm 4 (TCP/IPv4)** seçilir ve **Özellikler** düğmesi tıklanır.



IPv4 sürüm 4 özellikler ekranından IP, Ağ Geçidi, Alt Ağ Maskesi ve DNS bilgileri ekranda gösterildiği gibi yazılır. Cihaza atanacak IPv4 adresi için kurumunuzda tasarlanan IP aralığı kullanılmalıdır. Yukarıdaki örnekte C Class private IP adres (192.168.1.5) kullanılmıştır.

Dinamik IP atamasında ise, IP adresi, Default Gateway, Subnet Mask gibi bilgilerin bir DHCP (**Dynamic Host Configuration Protocol**) sunucu tarafından dağıtılması gereklidir. Önceden sunucuda tanımlanan bu bilgiler, IP adresi talebinde bulunan cihazlara kiralanan. Yukarıdaki ekranda “**Otomatik olarak bir IP adresi al**” seçildiğinde gerekli bilgileri dinamik olarak alınır

MANTIKSAL VE FİZİKSEL ADRES

Bir ağda iletişimini gerçekleştirmek için host cihazların mantıksal ve fiziksel olmak üzere iki adresi ihtiyacı vardır. Mantıksal adres, cihazın bulunduğu yere ve konumuna göre değişen

adrestir. Bu amaçla kullanılan adres genellikle IP adreslerdir. Fiziksel adres ise, cihazın değişimeyen, üretici tarafından atanmış adresidir. Ethernet ağlarında bu adres için Media Access Control (**MAC**) adresi kullanılır. MAC adres 48-bitlik bir sayıdır ve NIC üreticisi tarafından atanmış tekil bir adresdir. Aşağıda örnek bir MAC adresi gösterilmiştir.

A0-88-B4 -68-DC-3C	
A0-88-B4	68-DC-3C
Üretici Kısımlı	Üreticinin atadığı değer

Bu 48 bitlik sayı onaltılık sayı sistemi şeklinde gösterilir. İlk 24-bit IANA tarafından tahsis edilen üreticinin kodu iken sonraki 24 bit üreticinin atadığı seri numarası olarak düşünülebilir.

Bilgisayarın fiziksel adresi (MAC) ve mantıksal adresini (IP) görmek için komut satırından **ipconfig /all** komutu kullanılır.

```
Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . : WAG160N
Description . . . . . : Intel(R) Centrino(R) Advanced-N 6205
Physical Address. . . . . : A0-88-B4-68-DC-3C
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
IPv6 Address. . . . . : 2001:db8:fb:1907::10(PREFERRED)
Link-local IPv6 Address . . . . . : fe80::902f:b52f:3e63:466d%18(PREFERRED)
IPv4 Address. . . . . : 192.168.1.184(PREFERRED)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : 30 Mayıs 2018 Çarşamba 21:03:19
Lease Expires . . . . . : 6 Haziran 2018 Çarşamba 20:41:56
Default Gateway . . . . . : 2001:db8:fb:1907::1
                           192.168.1.1
DHCP Server . . . . . : 192.168.1.1
DHCPv6 IAID . . . . . : 396396724
DHCPv6 Client DUID. . . . . : 00-01-00-01-21-7C-FE-62-2C-41-38-04-26-6D
DNS Servers . . . . . : 8.8.8.8
                           8.8.4.4
NetBIOS over Tcpip. . . . . : Enabled
```

OSI MODELİ, TCP/IP MODELİ VE ENCAPSULATION

Günlük hayatımızda bilgisayarlar ve bilgisayar ile yapılan işlemler arttıkça, bilgilerin bir bilgisayardan diğerine taşınması ve haberleşmesi ihtiyacı da artmaktadır. Belli bir işte ortak bir dosya üzerinde çalışma, doküman paylaşımı gibi birçok özellik artık kaçınılmaz hale gelmiştir. İlk önceleri bu tür çok kullanıcının üzerinde çalışması gereği ortak dosyalar, disketler aracılığıyla aktarılıyordu (**Sneakernet**). Daha sonraları bilgisayarların kendi aralarında haberleşmesi için çalışmalar başlatıldı. Birçok üretici kendi network donanımını geliştirmeye başladı. Ancak farklı üreticilerin donanımları kendi aralarında haberleşmesi ciddi sıkıntılar doğuruyordu. Bu sıkıntıyı aşmak ve bir standartlaşmak adına ISO, Open System Interconnection (OSI) adında bir model geliştirdi. Bu modele göre, Network çalışması 7 adımda inceleniyordu. Her adımın işlevi ve o adımda yapılması gerekenler kesin olarak belirleniyordu. Network'ün çalışma prensiplerinin

belirlendiği bu adımlara “Layer (Katman)” diyoruz. OSInin bu mimarisindeki katmanlar birbirinden bağımsız olarak çalışıyordu. Bu standardizasyon sayesinde farklı üreticilerin ürünleri OSI modeline göre tasarlandığı için birbirleri ile haberleşebiliyordu.

OSI modeli:

L7 : Application

L6 : Presentation

L5: Session

L4: Transport

L3: Network

L2: Data Link

L1: Physical

katmanlarından oluşmuştur. OSI katmanlarını özetle anlatacak olursak;

Application Layer (Uygulama Katmanı), kullanıcı ile etkileşimin olduğu, uygulamaların bulunduğu katmandır. Örneğin bir web sayfasına bağlanmak için Internet Explorer, Firefox, Chrome gibi bir web tarayıcı uygulaması çalıştırılmak gereklidir.

Presentation Layer (Sunum Katmanı), uygulama katmanından gelen verilerin şekillendirildiği (sıkıştırma, şifreleme vb.), yani kaşdaki uygulamanın ya da hizmetin anlayabileceği bir biçimde sunulmasının gerçekleştiği katmandır.

Session Layer, cihazlar arasında sanal oturumların kurulduğu, yönetildiği ve sonlandırıldığı katmandır.

Bu üç katmana, üst katmanlar (**upper layer**) denir.

Transport Layer, üst katmanlardan gelen verilerin alt katmanları ile iletişimini sağlayan katmandır. Diğer bakış açısından, alt katmanlardan gelen verinin, üst katmanlarda hangi uygulama ya da servis ile iletişime geçeceğini belirlendiği katmandır. Bu belirleme PORT numaraları ile sağlanır.

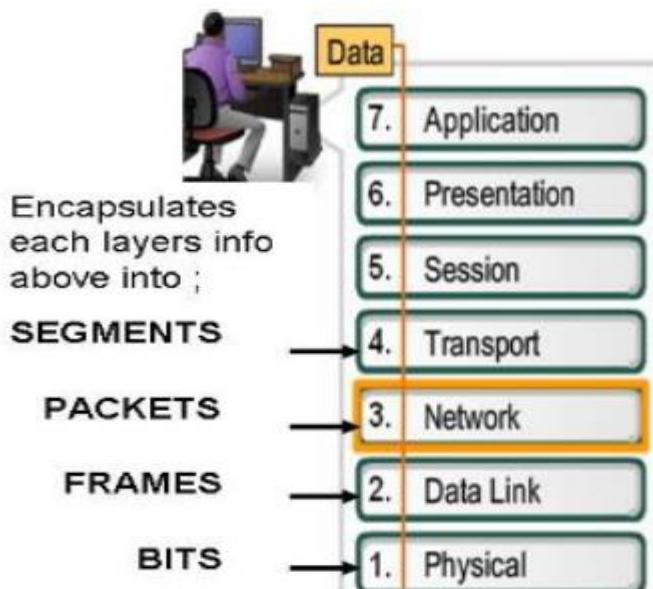
Network Layer, mantıksal adreslemenin yapıldığı katmandır. Burada veriyi gönderen ve alacak olan cihazların mantıksal adresleri bulunur. Mantıksal adres olarak burada Internet Protocol (IP) adresleri eklenecektir. IP adresi dışında IPX, AppleTalk gibi mantıksal adresler bulunmakla beraber büyük bir çoğunlukla IP kullanıldığı için bu eğitim boyunca IP adresleri örnek olarak gösterilecektir.

Data – Link Layer, fiziksel adreslemelerin yapıldığı yerdir. Local ağdaki, gönderici ve alıcı cihazların fiziksel adreslerinin belirlendiği alandır. Burada Fiziksel Adres olarak MAC adresleri kullanılır.

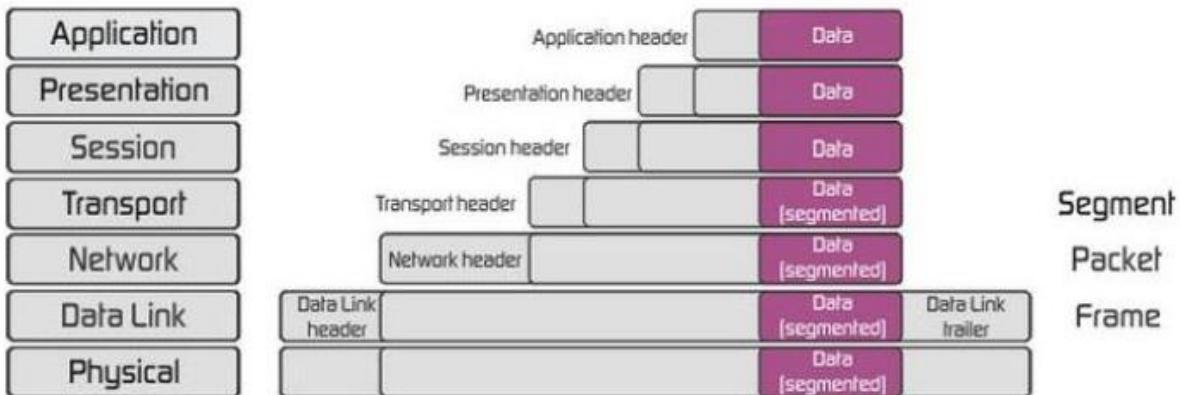
Physical Layer, verilerin taşınacak ortama (kablo, hava gibi) aktarılmak için bitlere dönüştürüldüğü katmandır.

İsimlendirmede kolaylık olsun diye tüm katmanlara aşağıdan yukarıya olmak üzere numaralar verilmiştir. Buna göre örneğin Physical Layer : **Layer 1 (L1)**, Application Layer : **Layer7(L7)** olarak adlandırılmıştır.

Bir ağ uygulamasını kullanan bir kullanıcı, başka bir bilgisayar veya cihaz ile iletişim kurar. Bu durumda kullanıcının göndereceği bilgiler (**data**) her katmanda, bulunduğu katmana özgü değişikliğe uğrar yani veriye katmanla ilgili bazı başlık bilgileri (**header**) eklenir ve bir alt katmana aktarılır. En son fiziksel ortama aktarılincaya kadar bu işlem devam eder. Buna **encapsulation** (kapsülleme) denir. Her katmanda, şekillenen bu veriye **Protocol Data Unit (PDU)** adı verilir. Buna göre katmanlar ve bu katmanlara ait PDU ların isimleri aşağıda verilmiştir.



Alicı tarafta ise bu işlemin tersi gerçekleşir. Yani fiziksel ortamdan alınan bitler, başlık bilgileri her katmanda çıkarılıp bir üst katmana aktarılır. Bu işleme de **de-encapsulation** denir.



OSI Modeline göre, uygulamalardan gelen veri, 7., 6. Ve 5. Katmanlarda büyük bir değişikliğe uğramadan Transport katmanına gelir. Bu katmanda uygulamanın türüne göre TCP ya da UDP başlık bilgisi eklenir ve SEGMENT oluşur.

Eklenen bu başlık bilgisinde özetle gönderen ve alıcının port numaraları bulunur. Bu sayede hedef ile kaynak arasında hangi uygulamaların veya servislerin haberleşeceği belirlenmiş olur. Segment olarak kapsüllenmiş veri bir alt katmana aktarılır.

Network katmanında, gönderen IP adresi, alıcı IP adresi, TTL(*) gibi bir takım bilgiler eklenir.

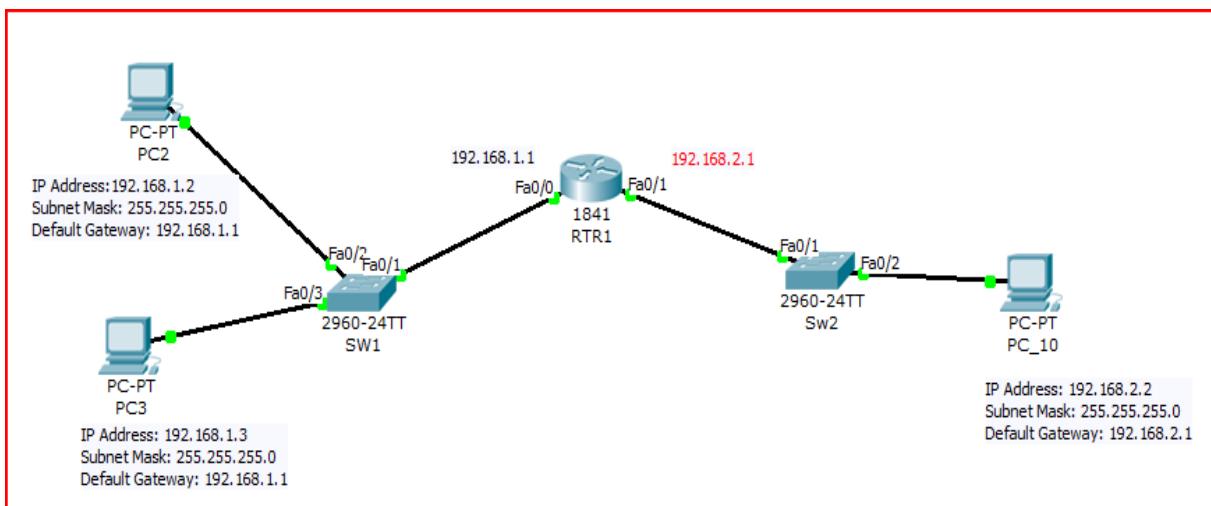
IP başlık bilgisi eklenip packet oluşur. Ethernet teknolojisinde bir packetin minimum boyutu 46 Byte; maximum boyut ise 1500 Byte olmalıdır.

İkinci katmanda (Data Link Layer) 3. Katmandan gelen packet yapısına bu kez Kaynak MAC adresi, Hedef MAC adresi, Type/Length başlık bilgisi eklenir. Ayrıca sonuna da, bit bazında hata kontrolü için FCS alanı eklenir.

Frame yapısına dahil olmamakla beraber, alıcı ve gönderici arasında senkronizasyonu sağlamak amacıyla PREAMBLE denen 8-BYTE uzunluğunda bir alan eklenir. Bu alanın boyutu frame yapısında hesaba katılmaz. O halde bir frame için minimum boyut, 64Byte; maximum boyut da 1518 Byte olmalıdır.

TEMEL İLETİŞİM

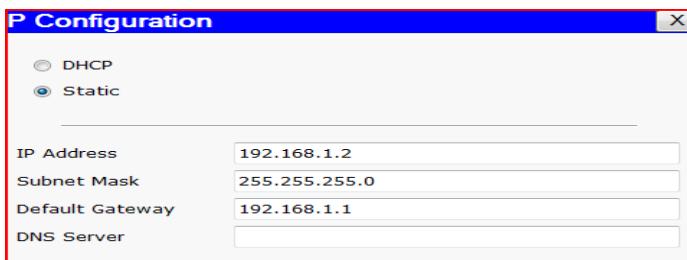
Şimdiye kadar öğrendiğimiz tüm bilgiler ışığında iki bilgisayar arasındaki iletişim adım adım açıklamaya çalışalım



Tüm cihazların yeni açıldığını varsayıyalım. Router, kendisine bağlı olan networkler hakkında bilgi sahibidir. Bu durumda Router için routing tablosu:

Routing Table for RTR1				
Type	Network	Port	Next Hop IP	Metric
C	192.168.1.0/24	FastEthernet0/0	---	0/0
C	192.168.2.0/24	FastEthernet0/1	---	0/0

PC ler,Kendi IP, Gateway ve Subnet Mask bilgilerini bilir.



PC lerin ARP tablosu, boştur.

ARP Table for 192.168.1.3		
IP Address	Hardware Address	Interface

Switchlerin de MAC tablosu boştur.

MAC Table for Switch2		
VLAN	Mac Address	Port

A. AYNI AĞDA İLETİŞİM

PC2, aynı ağa bulunan PC3 ile iletişime geçmeye, örneğin PING (**ICMP Echo Request**) atmaya çalışın.

```

Packet Tracer PC Command Line 1.0
PC>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

```

PC2, hedef IP (192.168.1.3) ile kendi subnet mask bilgisini (255.255.255.0) AND leyecektir.

$$192.168.1.3 \wedge 255.255.255.0 = \textbf{192.168.1.0}$$

Kendi IP'si ile Subnet Mask AND leyecektir.

$$192.168.1.2 \wedge 255.255.255.0 = \textbf{192.168.1.0}$$

PC2, Her iki değer eşit olduğuna göre, PC3'ün kendisi ile aynı ağıda olduğunu anlayacaktır.

Not: Aslında PC ler de tipki Router gibi routing tablosuna göre gelen pakete ne yapcağını belirleyecektir. PC routing tablosunu görmek için Windows İşletim sistemlerinde ROUTE PRINT komutunu kullanın.

Örneğin Benim PC için IPv4 routing tablosu :

IPv4 Yol Tablosu					
Etkin Yollar:					
Ag Hedefi	Ag Maskesi	Ag Geçidi	Arabirim	Ölçüt	
0.0.0.0	0.0.0.0	192.168.1.99	192.168.1.233	276	
127.0.0.0	255.0.0.0	On-link	127.0.0.1	306	
127.0.0.1	255.255.255.255	On-link	127.0.0.1	306	
127.255.255.255	255.255.255.255	On-link	127.0.0.1	306	
169.254.0.0	255.255.0.0	On-link	192.168.1.233	30	
169.254.255.255	255.255.255.255	On-link	192.168.1.233	276	
192.168.1.0	255.255.255.0	On-link	192.168.1.233	276	
192.168.1.233	255.255.255.255	On-link	192.168.1.233	276	
192.168.1.255	255.255.255.255	On-link	192.168.1.233	276	
224.0.0.0	240.0.0.0	On-link	127.0.0.1	306	
224.0.0.0	240.0.0.0	On-link	192.168.1.233	276	
255.255.255.255	255.255.255.255	On-link	127.0.0.1	306	
255.255.255.255	255.255.255.255	On-link	192.168.1.233	276	

- Satırda** hedefi bilinmeyen tüm paketlerin 192.168.1.99'a iletileceğini gösterir. 192.168.1.99 PC'nin Gateway adresidir. Bu gateway'e ulaşmak için ise 192.168.1.233 arabirimini (PC nin IP adresi, dolayısıyla Ethernet kartı) kullanır. Bu sayede aynı ağıda olmayan veya internetteki herhangi bir IP ye ulaşmak için gateway adresi ile iletişime geçilmesi gerektiğini,
- 3. ve 4. Satırlarda**, 127.0.0.0 (127 ile başlayan tüm IP lerin) PCnin kendisine (Onlink) yönlendirileceğini,
- Satırda**, APIPA adresi sayesinde kendi ağımdaki APIPA'dan IP almış cihazlara Ethernet Kartım ile iletişim kurabileceğini,
- Satırda**, APIPA Broadcast isteklerin Ethernet kartına yönlendirileceğini,
- Satırda**, Broadcast isteklerin yine Ethernet kartına yönlendirildiğini,
- 8. ve 9. Satırlarda**, PCnin bir multicast gruba dahil olduğunu ve Ethernet kartına yönlendirildiğini(224.0.0.0),
- 10.ve 11. Satırlarda**, genel broadcast mesajlarının Ethernet kartına yönlendirildiğini görebiliriz.

Her iki cihaz aynı ağıda olduğuna göre oluşturulacak çerçeveyi Switch aracılığıyla hedefe ulaşılabilir olduğu anlaşılmır. Bu durumda PC2, bir çerçeve oluşturamaya çalışacaktır.

Hedef IP : 192.168.1.3

Hedef MAC:???? (bilinmiyor)

PC2, Hedef MAC (**Destination MAC**) için 192.168.1.3'e karşılık gelen MAC adresini öğrenmek için kendi ARP tablosuna bakacaktır.

ARP tablosu boş olduğu için, **192.168.1.3** e karşılık gelen MAC adresi öğrenmek için ARP isteği (**ARP Request**) yayınılayacaktır. Bu ARP mesajı Broadcast bir mesajdır. Hedef MAC adresi **FFFF.FFFF.FFFF**, kaynak MAC ise, PC2'nin MAC adresidir.

PC2 MAC= 0000.0C7E.A791

Arp mesajı yayınlanıp Switch'e ulaşacaktır. Switch, gelen çerçevedeki kaynak MAC adresine **0000.0C7E.A791** bakıp bunu gelen port (**FastEthernet 0/2**) ile eşleştiricektir. Bu sayede MAC tablosunda aşağıdaki gibi bir kayıt oluşacaktır.

Switch gelen ARP çerçevesinin hedef MAC adresine bakıp **FFFF.FFFF.FFFF** (broadcast) buna gelen port haricindeki tüm aktif portlara gönderecektir. Dolayısıyla çerçeve hem PC3'e hem Router'a gidecektir. Router gelen ARP mesajını dikkate almayacak ve çöpe atacaktır. PC3 ise bu ARP mesajındaki hedef IP nin kendi IP si olduğu için cevap verecektir. Bu arada PC3' PC2 ile iletişime geçtiği için MAC- IP eşleşmesini bilecektir. Yani ARP tablosu aşağıdaki gibi olacaktır.

PC3, PC2 nin hem MAC hem IP sini bildiği için aşağıdaki gibi bir Unicast ARP cevabı yayınılayacaktır.

Ethernet II			
0	4	8	14
PREAMBLE: 101010...1011		DEST MAC: 0000.0C7E.A791	SRC MAC: 000D.BD96.831D
TYPE: 0x806	DATA (VARIABLE LENGTH)		FCS: 0x0

ARP			
0	8	16	31 Bits
HARDWARE TYPE: 0x1		PROTOCOL TYPE: 0x800	
HLEN: 0x6	PLEN: 0x4	OPCODE: 0x2	
SOURCE MAC: 000D.BD96.831D (48 bits)		SOURCE IP (32 bits) ==>	
192.168.1.3			
TARGET MAC: 0000.0C7E.A791 (48 bits)			
TARGET IP: 192.168.1.2 (32 bits)			

ARP mesajında PC2'nin ihtiyaç duyduğu PC3 MAC adresi bulunmaktadır.

Bu unicast ARP cevabı (**ARP Reply**) switch'e ulaşacaktır. Switch gelen çerçevedeki kaynak MAC (Source MAC) alanına bakacak ve geldiği port ile eşleştirip MAC tablosuna aktaracaktır. Bu durumda Switch MAC tablosu aşağıdaki gibi olacaktır.

MAC Table for SW1		
VLAN	Mac Address	Port
1	0000.0C7E.A791	FastEthernet0/2
1	000D.BD96.831D	FastEthernet0/3

Switch, port eşleştirme yaptıktan sonra Hedef MAC adresine bakacaktır. Hedef MAC **0000.0C7E.A791** adresinin hangi portta olduğu (**FastEthernet 0/2**) bilindiği için Switch gelen çerçeveyi sadece bu porta yönlendirecektir. Dolayısıyla sadece PC2 bu cevabı alacaktır. Gelen cevaptan yola çıkararak PC2, 192.168.1.3 IP sinin MAC adresini **000D.BD96.831D** öğrenecek ve ARP tablosuna yazacaktır. Bu durumda PC2'nin ARP tablosu aşağıdaki gibi olacaktır.

ARP Table for PC2		
IP Address	Hardware Address	Interface
192.168.1.3	000D.BD96.831D	FastEthernet

Artık PC2, PC3' PING paketi (**ICMP Echo Request**) göndermek için hem IP hem MAC bilgisine sahip olduğu için ICMP paketi oluşturabilecektir. Oluşan paket aşağıdaki gibidir.

Ethernet II					
0	4	8	14	19	Bytes
PREAMBLE: 101010...1011		DEST MAC: 000D.BD96.831D	SRC MAC: 0000.0C7E.A791		
TYPE: 0x800		DATA (VARIABLE LENGTH)		FCS:	0x0
IP					
0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 128		
			0x0	0x0	
TTL: 128	PRO: 0x1		CHKSUM		
		SRC IP: 192.168.1.2			
		DST IP: 192.168.1.3			
	OPT: 0x0		0x0		
		DATA (VARIABLE LENGTH)			
ICMP					
0	8	16	31	Bits	
TYPE: 0x8	CODE: 0x0	CHECKSUM			
ID: 0x2		SEQ NUMBER: 1			

ICMP paketi Switch' e ulaşacaktır. Switch öncelikle gelen çerçeveyi kaynak MAC adresine bakıp bunun ilgili porta işlenip işlenmediği kontrol edilecektir. Ardından hedef MAC adresine bakacak ve bu MAC adresin hangi portta olduğu bilgisine göre yönlendirecektir. Yukardaki hedef MAC bilgisi ve Switch MAC tablosundan yola çıkarak switch'in bunu FastEthernet 0/3 portuna yönlendireceğini görebiliriz.

Not: Çerçeveeler alındıktan sonra öncelikle FCS eşleştirmesi yapılır. Eşleşme hatalı ise Hedef MAC adrese bakılmaksızın çerçeve çöpe atılır. (Discard) Ayrıca switchlerin çalışma prensibine göre, FCS hesaplaması yapılp, bozuk çerçeveler anahtarlanmadan da imha edilebilir. Store and Forward switchler FCS hesaplarken, Fast Forward switchler FCS hesaplamaz

Bu sayede ICMP paketi PC3'e ulaşacaktır. PC3 gelen ICMP isteği hedef MAC adresine bakarak kendisine ait olduğunu öğrendikten sonra PING isteğine cevap (**ICMP Echo Reply**) verecektir.

Gönderilen 4 ICMP Echo Request paketine karşılık verilen ICMP Echo Reply cevaplarına göre PC2'de durum aşağıdaki gibi olacaktır.

```

PC>
Packet Tracer PC Command Line 1.0
PC>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 192.168.1.3: bytes=32 time=8ms TTL=128
Reply from 192.168.1.3: bytes=32 time=4ms TTL=128
Reply from 192.168.1.3: bytes=32 time=4ms TTL=128
Reply from 192.168.1.3: bytes=32 time=4ms TTL=128

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 8ms, Average = 5ms

```

Bazen ilk ICMP paketleri ARP istekleri sebebiyle zaman aşımına uğrayabılır, ancak diğer paketlere cevap gelecektir.

B. FARKLI AĞLARDA İLETİŞİM

PC1, PC100 ile ICMP Echo Request (PING) ile iletişim kurmak istesin.

```

PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

```

Bu durumda AND işlemi sonunda 192.168.2.2 cihazının PC2 ile aynı ağda olmadığı anlaşılacaktır. PC2 routing tablosunda **0.0.0.0** rotasının (default rota) gateway adresine yönlendirileceği görülür.

PC2, PC100'ün farklı ağda olduğunu anladıktan sonra oluşturacağı çerçeveyenin hedef IP ve hedef MAC yerine aşağıdaki gibi bir paket oluşturacaktır.

Hedef IP : 192.168.2.2 (PC100 IP)

Hedef MAC: Router Fastethernet 0/0 (Gateway) MAC Address

PC2, Hedef MAC adresi yerine gateway MAC adresini yazacaktır. Daha önce PC2 ile Gateway iletişim kurmadığı için PC2 ARP tablosunda gateway IP adresi için MAC kaydı yoktur. Sadece daha önce iletişim kurduğu PC3 'e ait IP ve MAC bilgileri vardır.

IP Address	Hardware Address	Interface
192.168.1.3	0000D.BD96.831D	FastEthernet

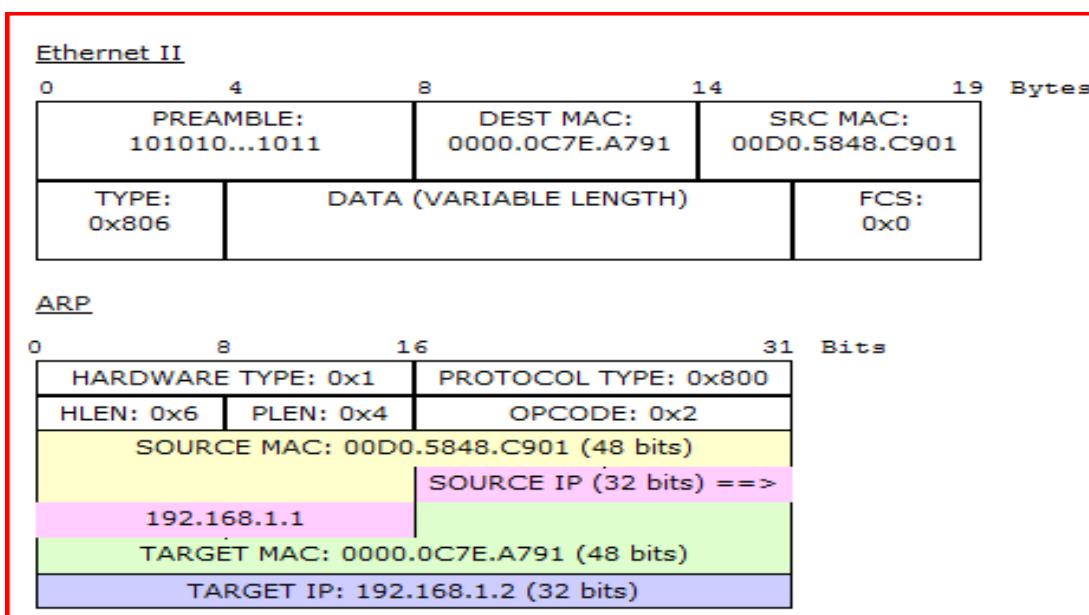
Bu yüzden PC2, gateway adresine (192.168.1.1) karşılık gelen MAC bulmak için ARP Request yayınılayacaktır.

Bu yayın Switchten geçecektir. Switch, gelen porttaki Kaynak MAC ile port eşleştirmesi olup olmadığına bakacak, buna göre kayıt yoksa MAC tablosuna ekleyecektir. Bu örnekte daha önceki iletişimden dolayı ilgili kayıt vardır. Switch hedef MAC adresine bakacaktır. Hedef MAC adres, ARP istekleri için FFFF.FFFF.FFFF dir. Dolayısıyla Switch bunu gelen port haricindeki tüm aktif portlara gönderecektir. PC3 de bu isteği alacak ancak ARP Reply yapmayacaktır. Gateway ise bu ARP mesajına cevap verecektir.

Gelen cevap switchten geçerken, Switch Kaynak MAC ile gelen portu (**FastEthernet 0/0**) eşleştirecek ve MAC tablosuna Router MAC ve ilgili portu işleyecektir. Bu durumda Switch PC2, PC3 ve gateway MAC adreslerini (**00D0.5848.C901**) hangi portta olduğunu bileyecktir. Switch MAC tablosu aşağıdaki gibidir.

VLAN	Mac Address	Port
1	0000.0C7E.A791	FastEthernet0/2
1	000D.BD96.831D	FastEthernet0/3
1	00D0.5848.C901	FastEthernet0/1

Gelen ARP cevabı Unicast bir mesaj olduğu, hedef MAC adresi PC2 nin MAC adresidir. ARP Reply paketi aşağıdaki gibidir.



Switch hedef MAC adresi (**0000.0C7E.A791**) bakacak ve bunu MAC tablosu aracılığıyla ilgili porta (**FastEthernet 0/2**) yönlendirecektir. Gelen ARP cevabında Gateway MAC adresi (**00D0.5848.C901**) bulunduğuundan PC2, Gateway MAC adresini öğrenecek ve ICMP Echo Request paketini oluşturabilecektir.

ICMP Echo Request Paketi aşağıdaki gibidir

Ethernet II

0	4	8	14	19 Bytes
PREAMBLE: 101010...1011		DEST MAC: 00D0.5848.C901	SRC MAC: 0000.0C7E.A791	
TYPE: 0x800	DATA (VARIABLE LENGTH)			FCS: 0x0

IP

0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 128		
ID: 0x9		0x0	0x0		
TTL: 128	PRO: 0x1	CHKSUM			
SRC IP: 192.168.1.2					
DST IP: 192.168.2.2					
OPT: 0x0		0x0			
DATA (VARIABLE LENGTH)					

ICMP

0	8	16	31 Bits
TYPE: 0x8	CODE: 0x0	CHECKSUM	
ID: 0x3		SEQ NUMBER: 5	

Burada Hedef IP, PC100 IP adresidir. Hedef MAC, Gateway MAC adresidir. Kaynak IP, PC2 IP adresi, Kaynak MAC, PC2 MAC adresidir.

Bu paket, Switch'e ulaşacaktır. Switch Kaynak MAC – Port eşleştirmesini kontrol ettikten sonra, hedef MAC adresine bakıp bunu Gateway adresine gönderecektir.

Router (gateway), gelen paketteki hedef MAC adresine bakıp framenin kendisine geldiğini anlayacak ve bu kez hedef IP adresin (192.168.2.2) bakacaktır. Bu IP adresini Subnet Mask ile AND leyecek ve 192.168.2.0 adresini bulacaktır.

Bulduğu bu adresi, Routing Tablosu ile karşılaşacaktır. Routing tablosunda bununla ilgili bir kayıt ya da varsayılan bir rota (**default rota**) yoksa paketi çöpe atacaktır (**Discard**). Router için routing tablosu aşağıdadır.

Routing Table for RTR1

Type	Network	Port	Next Hop IP	Metric
C	192.168.1.0/24	FastEthernet0/0	---	0/0
C	192.168.2.0/24	FastEthernet0/1	---	0/0

Routing tablosunda 192.168.2.0 ağı için FastEthernet 0/1 arayüzünden çıkış yapması gerektiğini anlayacak ve bu porta yönlendirecek ve IP başlık bilgisindeki TTL değerini bir azaltacaktır.

192.168.2.0 ağı, directly connected (C) bir ağıdır.

Hedef IP adresi 192.168.2.2 unicast bir adres olduğu ve FastEthernet 0/1 arayüzünün IP adresi ile (192.168.2.1) aynı ağa olduğu için direk olarak ulaşılabilirdir.

Bu yüzden router gelen çerçeveyi aşağıdaki gibi değiştirecektir.

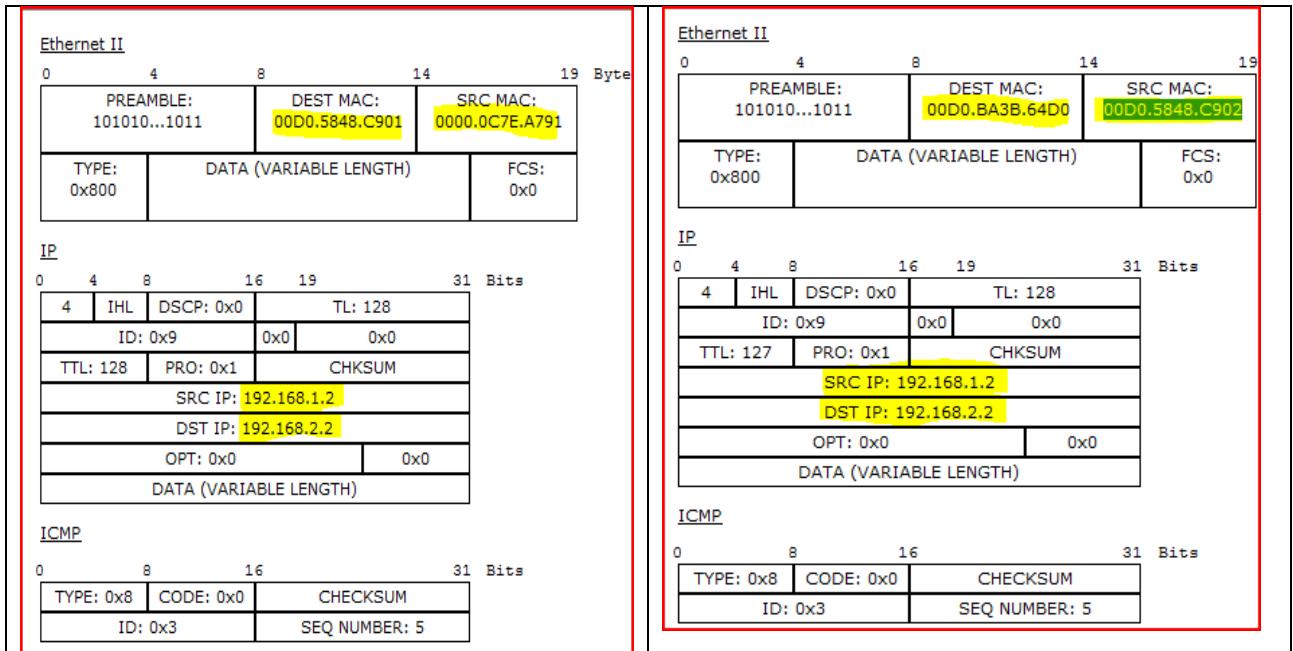
Hedef IP : 192.168.2.2 (Değişmez)

Hedef MAC : PC100 MAC adresi

Kaynak IP: 192.168.1.2 (PC2 IP adresi) – NAT yapılmamışsa değişmez.

Kaynak MAC : Router FastEthernet 0/1 MAC adresi : **00D0.5848.C902**

Bu durumda Router'a giren ICMP paketi ile çıkan ICMP paketi aşağıdaki gibi olacaktır.



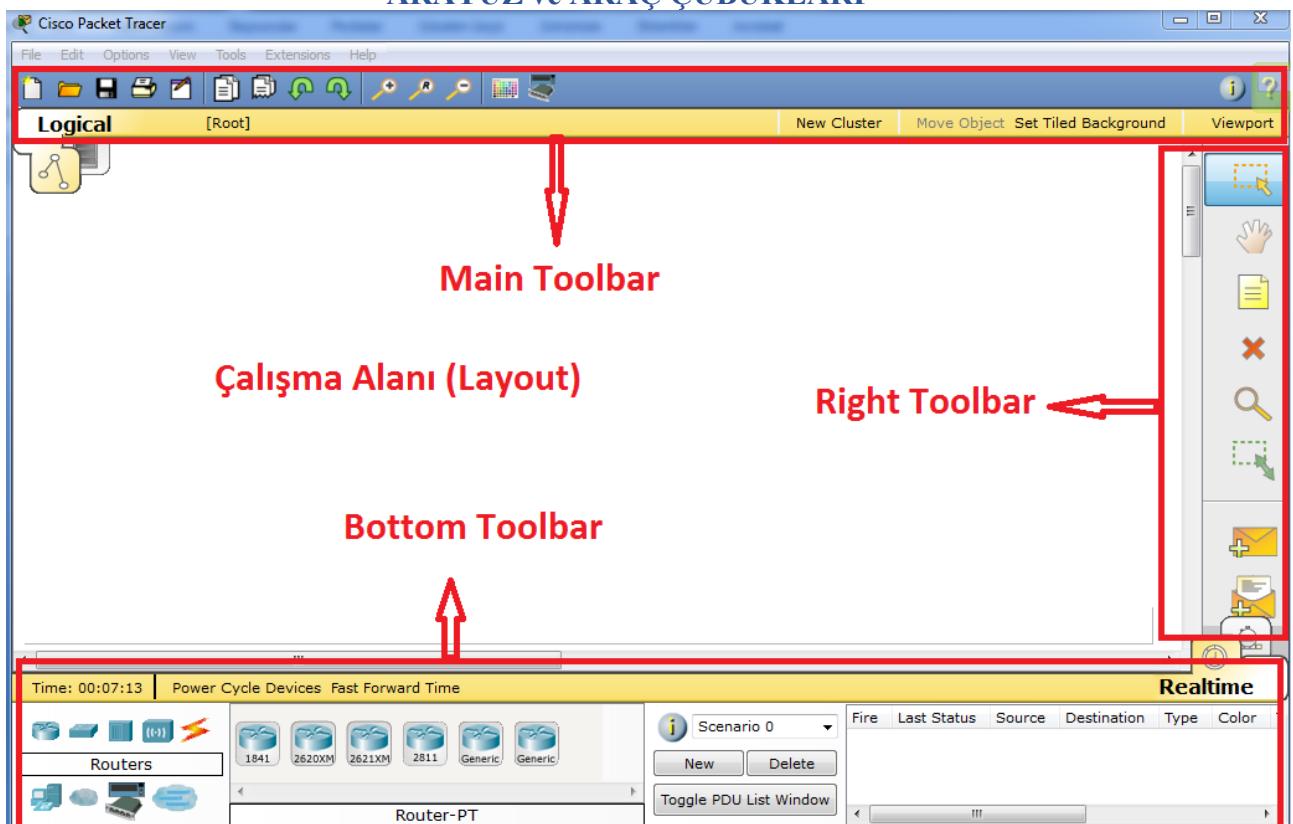
Burada HEDEF ve KAYNAK MAC adreslerin değiştiğiine ama IP adreslerin değişmediğine ; ayrıca TTL değerinin girerken 128, çıkarken 127 olduğuna (1 azaltıldığına) dikkat edin.

Hedef MAC adresi için 192.168.2.2'ye karşılık gelen MAC adresi yazılacaktır. Bunun için Router ARP tablosuna bakacaktır. Yukardaki örnekte ARP tablosunda ilgili kayıt vardır. ARP tablosunda kayıt olmasaydı, Router ARP Request yapıp öğrenecekti.

Son olarak Router'in oluşturduğu bu paket switche ulaşacaktır. Switch gelen port- Kaynak MAC eşleştirme yaptıktan sonra Hedef MAC adresine bakacak ve buna göre mesajı ilgili porta yönlendirecektir. Switch MAC tablosunda PC100' için MAC adresi olmasaydı, Switch gelen bu unicast çerçeveyi, gelen haricindeki aktif olan tüm portlara gönderecektir (flooding).

PACKET TRACER ARAYÜZÜ VE KULLANIMI

ARAYÜZ ve ARAÇ ÇUBUKLARI



Cisco Packet Tracer arayüzü Main Toolbar, Bottom Toolbar ve Right Toolbar olmak üzere 3 araç çubuğundan oluşmaktadır.

MAIN TOOLBAR

Burada kullanılan düğmelerin büyük çoğunluğu Windows ara yüzünden alışık olduğumuz düğmelerdir yaklaşık 30 yıldır aynı yerdeler .



New - Open - Save :

Sırasıyla **yeni** packet tracer uygulaması başlat, uygulama **aç** ve **kaydet** düğmeleridir.



Print : Mevcut fiziksel topolojiyi ya da seçilen cihazda yazılmış konfigürasyonları yazdırır.



Copy – Paste – Undo – Redo :

Yine birçok uygulamada kullanılan, kopyala, yapıştır, geri al (undo) ve yinele (redo) düğmeleridir.



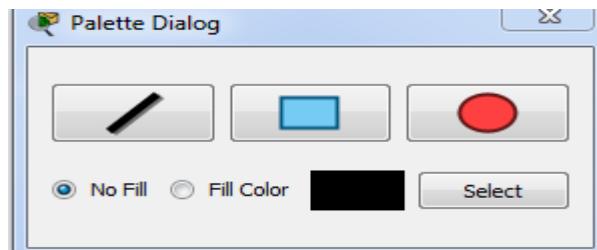
Zoom In - Zoom Reset - Zoom Out:

Topoloji ekranını **büyüt**, **orijinal boyut** ve **küçült** düğmeleridir.

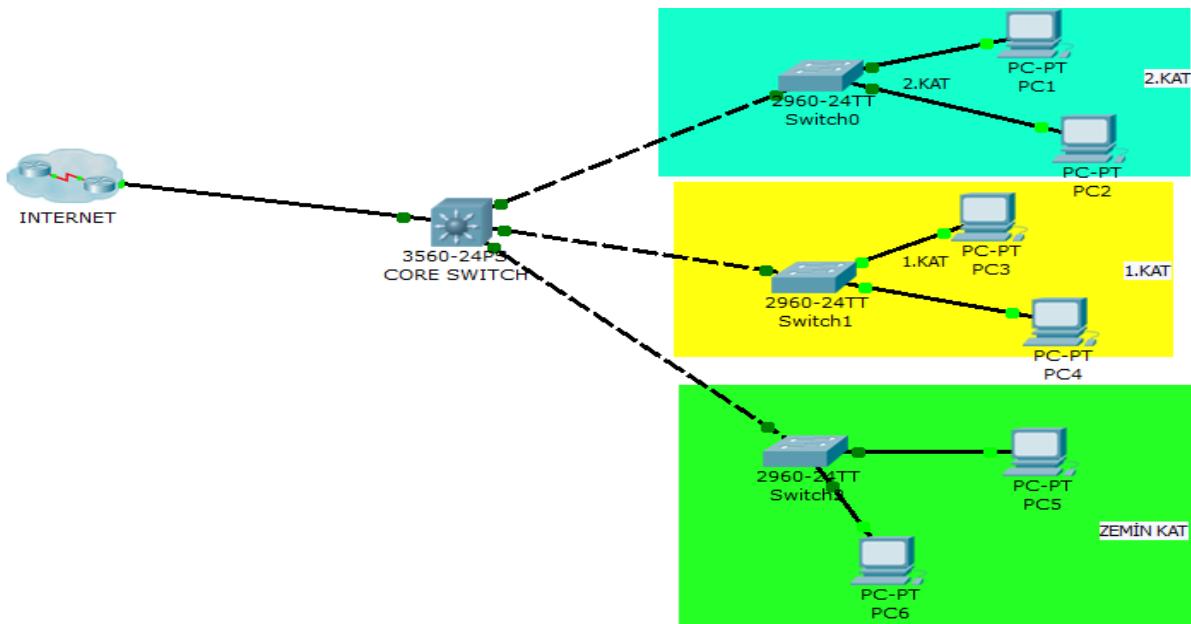


Drawing Palette:

Topoloji ekranına yardımcı olması açısından kare, daire gibi şekiller çizilmesini sağlayan **çizim aracıdır**.



Örnek olarak oluşturulan bir fiziksel topoloji üzerinde aşağıdaki gibi yardımcı çizimler yapılabilir.



RIGHT TOOLBAR



Select: Seçim aracıdır. Topolojide bulunan cihazları seçmek için kullanılır. Çoklu seçim için belirli bir dörtgen çizilip seçim yapılabilir.



Move Layout: Çalışma alanı üzerinde hareket etmeyi sağlayan araçtır. Büyük topolojilerde, cihazların tümü aynı ekrana sığmayabilir. Bu araç ile çalışma alanını gezebilirsiniz.



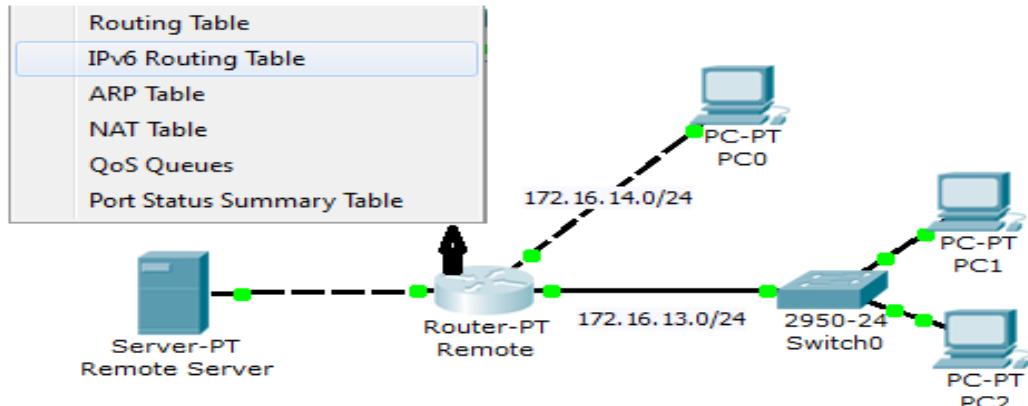
Place Note: Çalışma alanı üzerinde açıklayıcı notlar almanızı sağlayan araçtır.



Delete: Çalışma alanında bulunan bir veya daha fazla cihazın, bağlantının, şeklin ya da notların silinmesini sağlayan araçtır.



Inspect: Topolojide bulunan cihazların ARP tablosu, Routing Tablosu, Mac Tablosu gibi tabloların görüntülenmesini sağlar.



Resize Shape: Çalışma alanında bulunan şekillerin boyutlandırılmasını sağlayan araçtır.

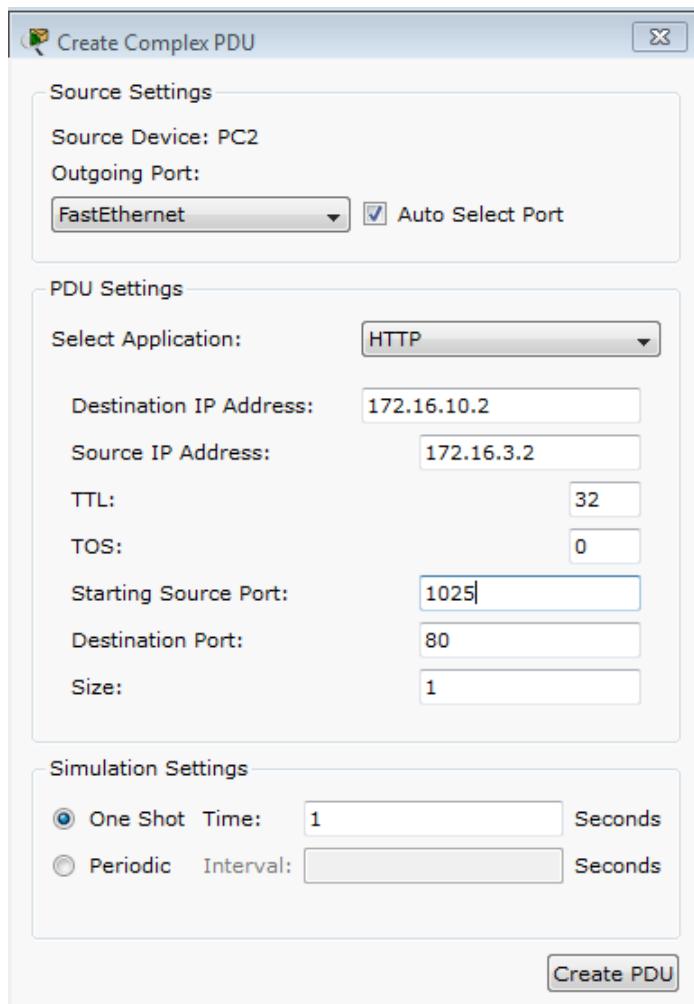


Add Simple PDU: Seçilen iki cihaz arasında bir ICMP paketi gönderilir. Cihazlar arasında iletişim olup olmadığını testi kısaca bu paket ile yapılabilir. İletişimin olabilmesi için cihazların IP adresinin bulunması gereklidir. İlk tıklanan cihaz kaynak, ikinci tıklanan ise hedefdir. Eğer iletişim başarılı ise(successful), Bottom Toolbar kısmında aşağıdaki gibi bir görüntü oluşur.

Fire	Last Status	Source	Destination	Type	Color	Time (sec)	Periodic
	Successful	PC2	PC1	ICMP		0.000	N

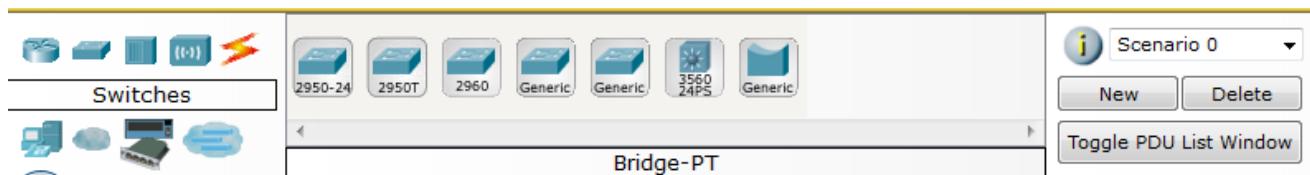


Add Complex PDU : Kaynak ile hedef arasında protokolü belirleyebileceğiniz, özelleştirilmiş paketler göndermenizi sağlar.



Yukarıdaki örnekte olduğu gibi hedef IP, kaynak IP, Hedef Port, Kaynak Port gibi parametreleri kullanarak farklı paketler oluşturabilirsiniz.

BOTTOM TOOLBAR



Çalışma alanına Router, Switch, PC gibi cihazların ve bunlar arasındaki kablolarının yapıldığı en sık kullanılan menüdür.



Cihazlar ve kablolar: Sırasıyla router, switch, hub, wireless cihazları ve kablolarını gösterir. Seçilen cihaza göre sağda modeller olacaktır.

ROUTER MODELLERİ



Cisco router modellerinden biri seçili, çalışma alanına tıklanarak cihazın topolojiye dahil olması sağlanabilir. Generic router seçildiğinde ise cihaz üzerinde takılacak olan modüller belirlenebilir.

ROUTER FİZİKSEL AYARLANMASI: Seçilen ve çalışma ekranına aktarılan bir router için modül ekleme gibi fiziksel ayarlamalar **Physical** sekmesinden yapılabilir. Aşağıda 1841 bir router için fiziksel arayüz bulunmaktadır.

Physical Device View

Zoom In Original Size Zoom Out

Açma / Kapama Düğmesi

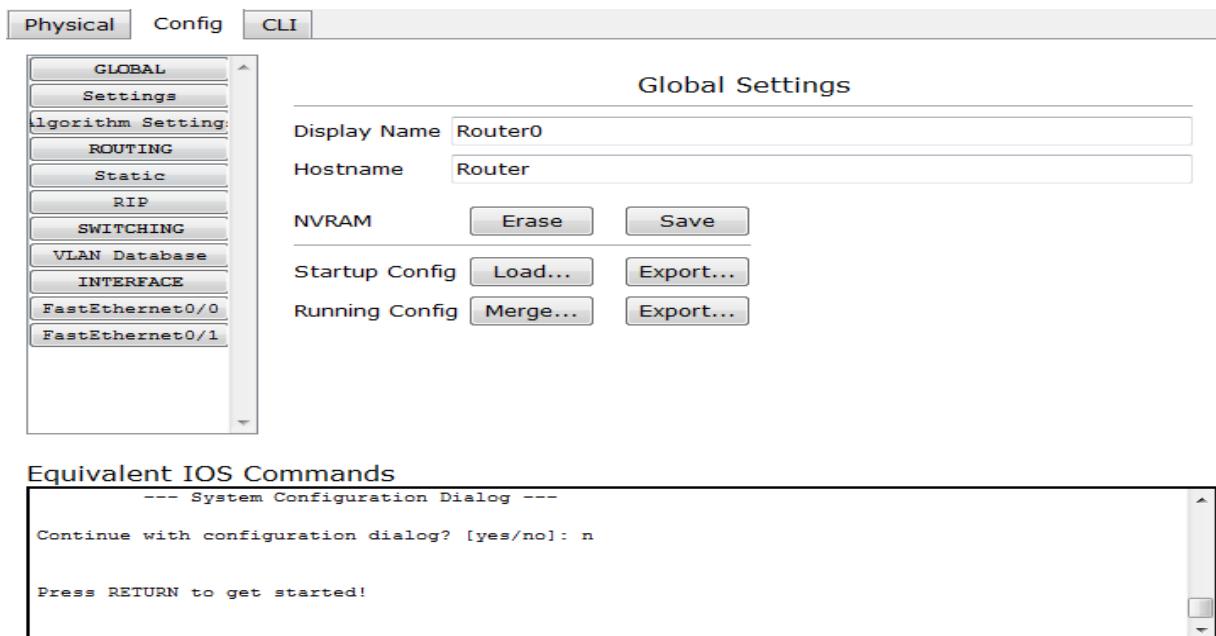
Customize Icon in Physical View Customize Icon in Logical View

Adding Modules: Drag the module to an available slot on the device.
Removing Modules: Drag the module from the device to the module list.

Fiziksel ekranda, routerda üzerinde modüller eklenip kaldırılabilir. MODULES ekranında uygun bulunan modüllerin isimleri, alta da bu modülün özellikleri ve görünümünü bulunmaktadır. Sürükle bırak yöntemi ile modül eklenebilir ve çıkarılabilir. Modül ekleme/ çıkarma işlemlerinden önce routerın **kapatılmış** olması gerekmektedir.

ROUTER KONFIGÜRASYONU (GÖRSEL ARAYÜZ):

Router üzerinde konfigürasyon komut arayüzünden yapılır. Ancak ilk başlayanlar için bu komutlar zorluk çıkarabilir. Config arayüzünden görsel olarak konfigürasyon yapılabilir.



Görsel ekranдан yapılan konfigürasyonların CLI arayüz komutları, **Equivalent IOS Commands** kısmından görülebilmektedir.

GLOBAL / SETTINGS Ayarları:

Display Name: Cihazın çalışma ekranında bulunan ismini değiştirebilirsiniz.

Hostname: Cihazın konfigürasyon ekranındaki ismidir. Burada yapılan değişiklik, cihazda CLI arayüzünde yazılmış aşağıdaki koda karşılık gelmektedir. Bu kod “**Equivalent IOS Commands**” alanında görülebilir.

Router(config)# hostname ERDAL

NVRAM bilgilerinin silinmesini ya da kaydedilmesini sağlayan düğmeler, Startup-Config ve Running-Config kodlarının kaydedilmesi ya da geri yüklenmesi bulunmaktadır.

ROUTING AYARLARI

Static Routing Ayarlaması

Static Routes

Network	10.0.0.0
Mask	255.0.0.0
Next Hop	192.168.1.1

Add

Network Address
10.0.0.0/8 via 192.168.1.1

Remove

Yukarda bir **static** yönlendirmenin nasıl yapıldığı gösterilmiştir. Bu örneğe göre, 10.0.0.0 /8 ağına 192.168.1.1 IP adresi üzerinden ulaşabileceğiniz konfigürasyonu yapılmıştır. Bu işlem yapılip Add düğmesi tıklandığında “**Equivalent IOS Commands**” ekranında aşağıdaki IOS – CLI komutları görülmektedir. Remove düğmesi ile de yazılan konfigürasyonun kaldırılması sağlanır.

Equivalent IOS Commands

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 192.168.1.1
Router(config)#

```

RIP KONFIGÜRASYONU

RIP Routing

Network	180.10.0.0
---------	------------

Add

Network Address
172.16.0.0
180.10.0.0

Remove

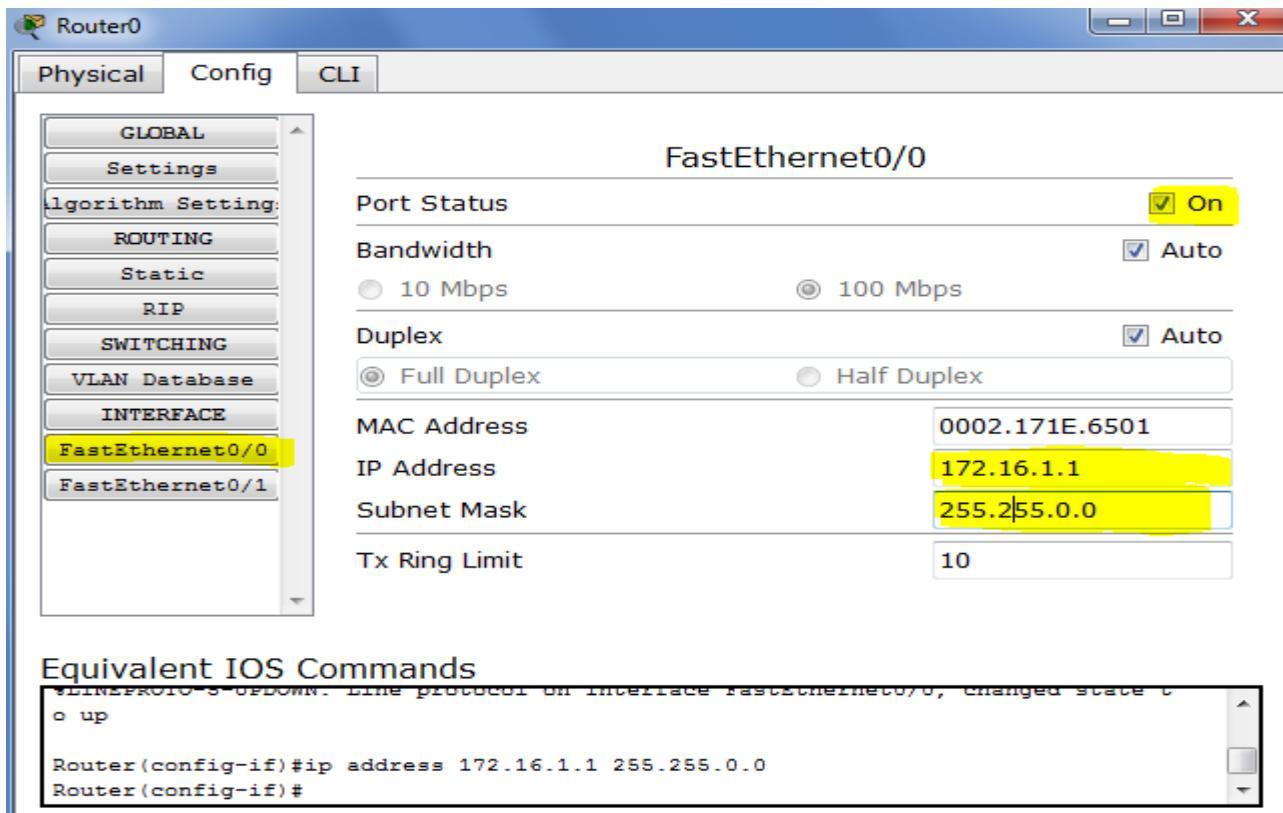
Equivalent IOS Commands

```
Router(config-router)#
Router(config-router)#exit
Router(config)#router rip
Router(config-router)#network 180.10.0.0

```

Tanıtımı yapılacak ağlar Network kısmına eklenip, ADD düğmesine basılır. Yine CLI komutları da görülebilir.

INTERFACE AYARLARI

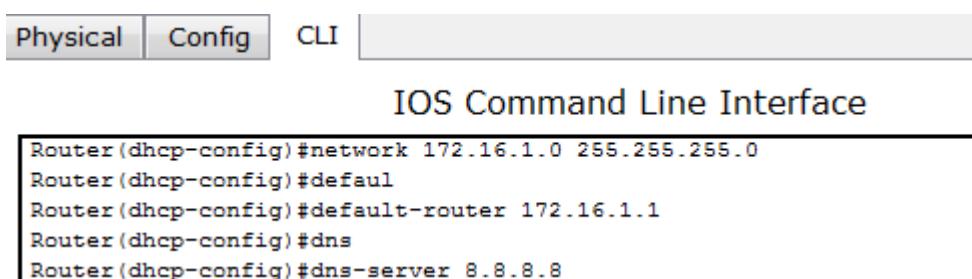


Router'ın herhangi bir arayüzünün açılıp IP Adresinin verilmesini yukarıdaki gibi konfigürasyon yapılabılır. Yine yapılan konfigürasyonun CLI çıktısı da gösterilmektedir.

CLI ARAYÜZÜ

Gerçek routerlar üzerinde konfigürasyonlar çoğunlukla komut arayüzinden (CLI) yapılmaktadır. Buradaki ekran gerçek router üzerinde çalışan birçok komutu desteklemektedir.

Aşağıdaki ekranda örnek bir DHCP konfigürasyonu komutları bulunmaktadır.



Komut ekranında konfigürasyon için, cisco cihazların üzerinde bulunan işletim sistemi (IOS) bilgisi gerektirmektedir.

SWITCH MODELLERİ



24 portlu Cisco 2950 ve Cisco 2960 gibi layer 2 cihazların olduğu gibi, Cisco 3560 Layer3 cihazlar ve portları ayarlanabilen generic switchler de vardır.

KABLOLAR



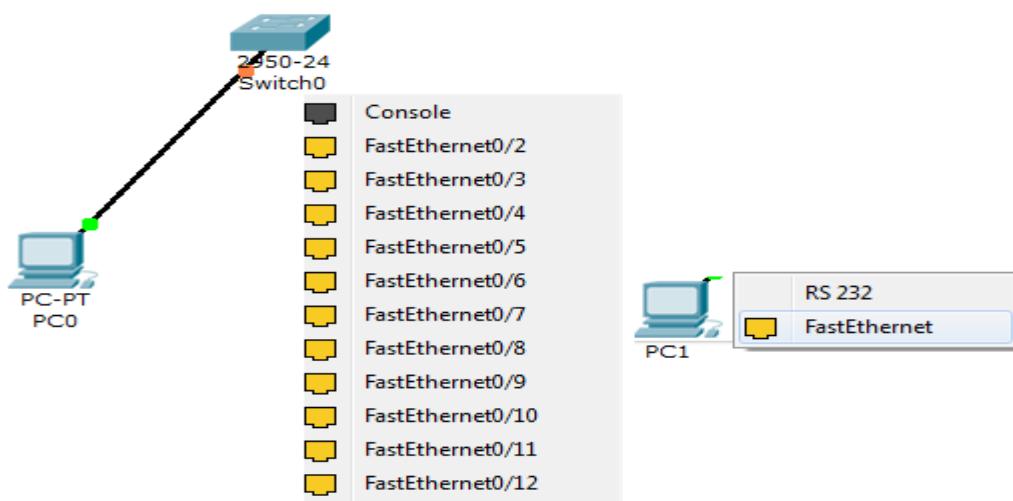
Sırasıyla, otomatik kablo seçimi, Console kablosu, düz kablo, çapraz kablo, fiber kablo, telefon kablosu, coaxial kablo, Seri (DCE) kablo ve Seri (DTE) kablolarıdır.

SON KULLANICI CİHAZLARI



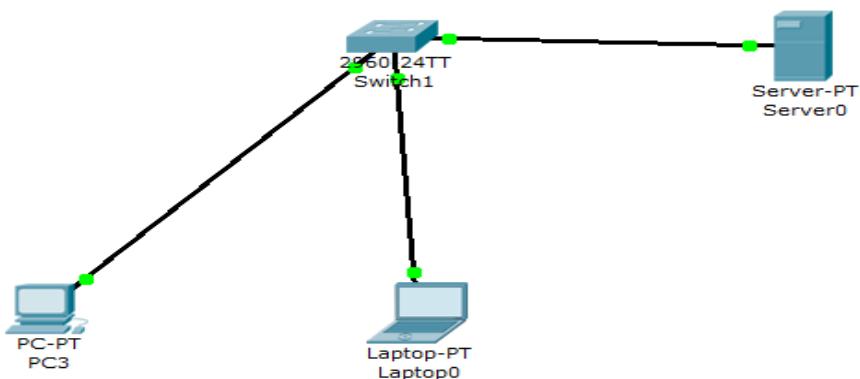
PC, Laptop, Server, IP Yazıcı, Yerel Yazıcı, IP Telefon gibi cihazların çalışma alanına eklenmesini sağlayan bölümdür.

CİHAZLAR ARASINDAKİ KABLO BAĞLANTILARI



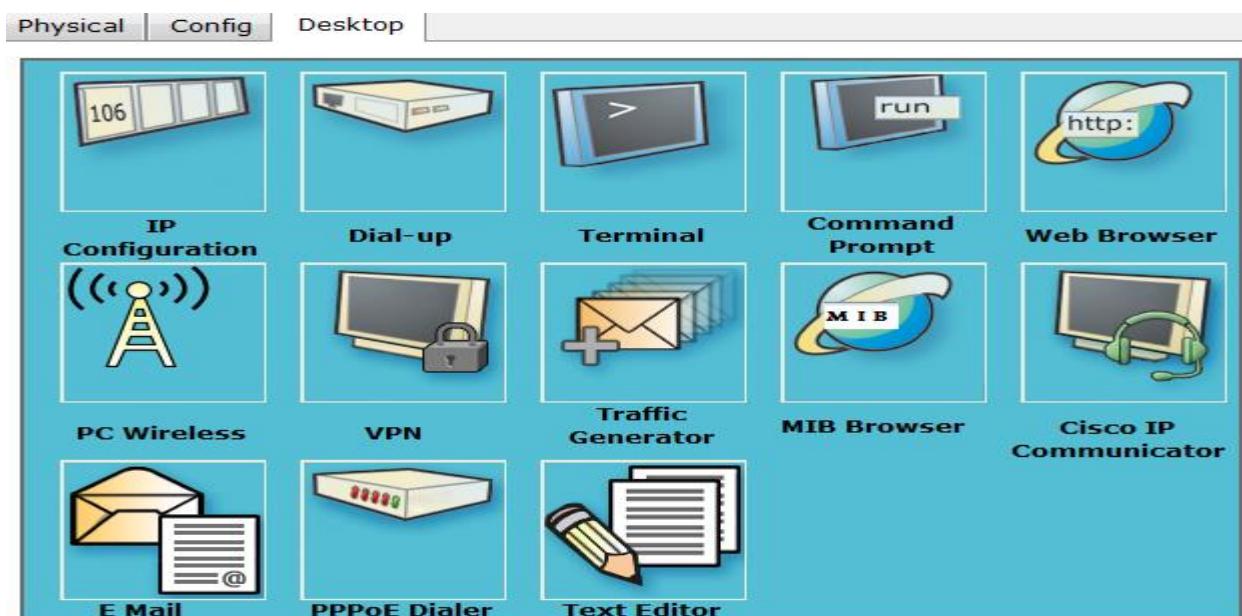
Yukarda olduğu gibi, düz kablolar kullanılarak PC ile SWITCH arasındaki bağlantı ve bu bağlantının hangi arayüze takılacağı belirlenebilir.

TEMEL PC KONFIGÜRASYONU

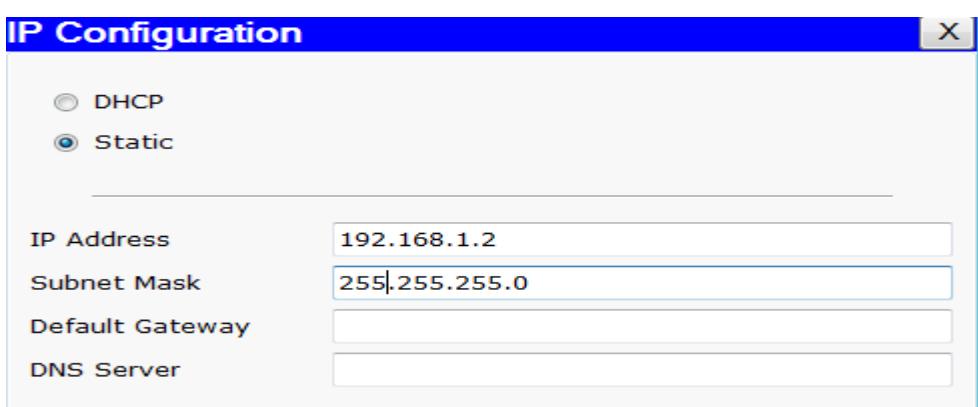


Yukarıdaki örnekte olduğu gibi PC lerin iletişime geçebilmesi için IP adreslerinin verilmesi gereklidir. PC üzerinde tıklanınca açılacak pencereden Desktop kısmından bu ayarlamalar yapılabilir.

PC 'lerde uygulamalarda yardımcı olacak, Komut Satırı (Command Prompt), Web Browser, Hyper Terminal yazılımı gibi bir çok araç bulunmaktadır.

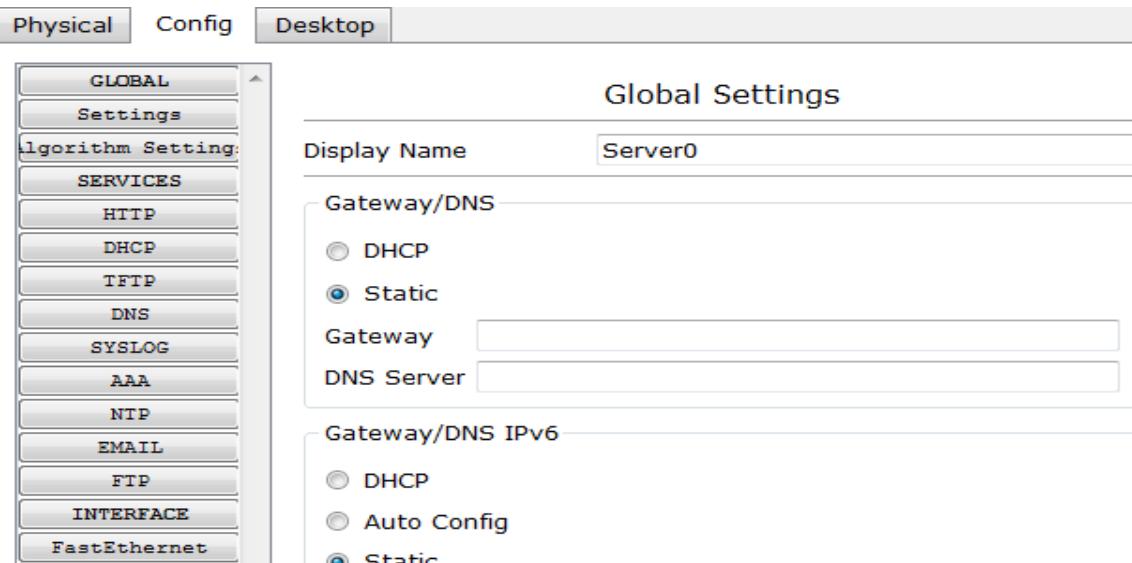


IP ayarlarının yapılabilesi için, **IP Configuration** düğmesi tıklanır.



Açılan pencereden bir statik IP ya da ortamda bulunan bir DHCP serverdan otomatik IP verilebilir.

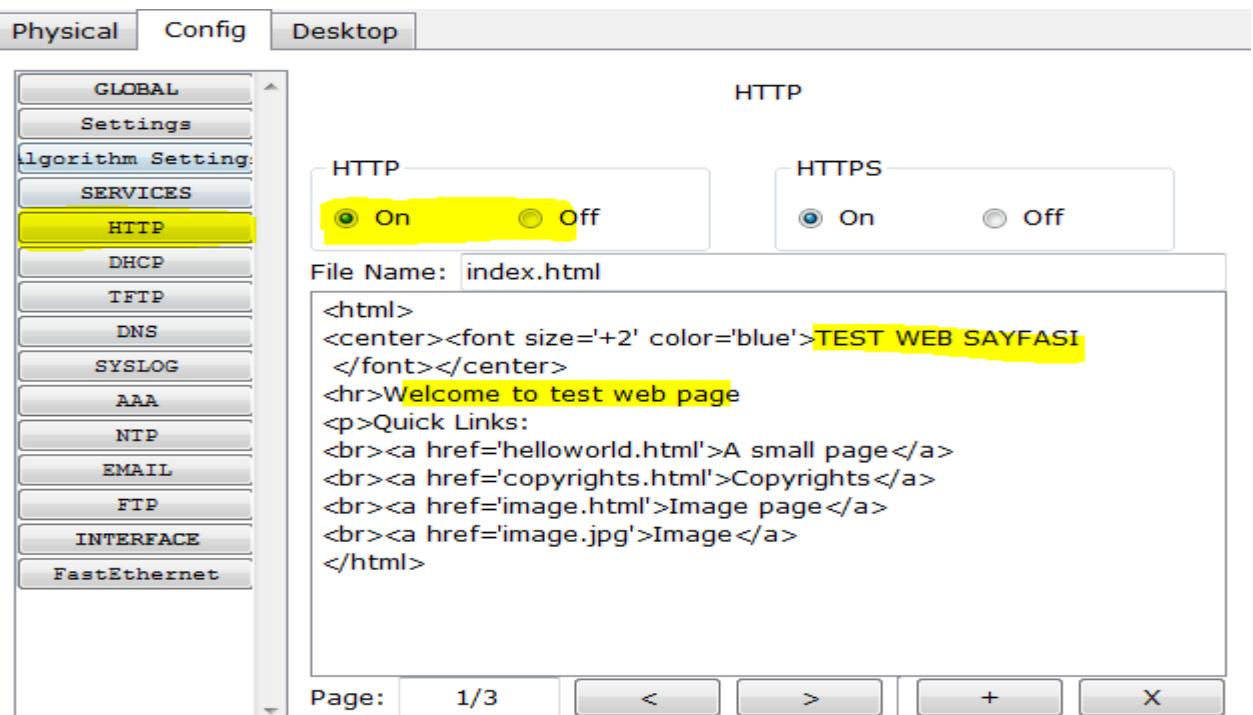
SERVER KONFIGÜRASYONU



Son kullanıcı cihazları arasında bulunan serverlar üzerinde DHCP, HTTP, FTP, DNS gibi sunucular tanımlanabilir. Aşağıdaki örnekte DNS ve HTTP server tanımlaması yapılmıştır.

HTTP SERVER KONFIGÜRASYONU

Server üzerinde Config sekmesinde http başlığı altında aşağıdaki gibi örnek bir web sayfası html kodları yazılabilir.



Burada tanımlanan web sayfasına alan adıyla erişilebilmesi için, DNS serverde de tanımlanması gereklidir. DNS ayarları da yine DNS tabından yapılabilir.

The screenshot shows a network configuration interface with tabs for Physical, Config, and Desktop. The Desktop tab is active. On the left, a sidebar lists services: GLOBAL, Settings, Algorithm Setting, SERVICES, HTTP, DHCP, TFTP, DNS (which is selected and highlighted in yellow), SYSLOG, AAA, NTP, EMAIL, FTP, INTERFACE, and FastEthernet. The main panel is titled 'DNS' and contains the following information:

- DNS Service: On (radio button selected)
- Resource Records:
 - Name: www.test.com
 - Type: A Record
 - Address: 192.168.1.4
- Table of records:

No.	Name	Type	Details
1	www.test.com	A Record	192.168.1.4

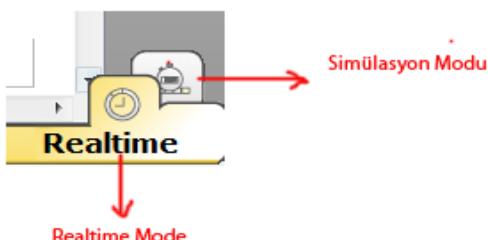
Örnekte, www.test.com adresinin IP adresi, 192.168.1.4 olarak DNS servere işlenmiştir. Bu örnekte, DNS server ve HTTP Server aynı server üzerinde tanımlı olduğuna dikkat edilmesi gerekmektedir.

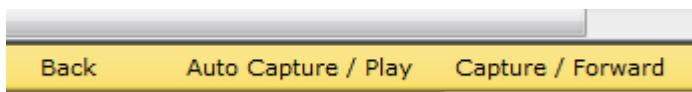
PC üzerinden DNS ayarlaması yapıldığı taktirde web sayfasına Web Browser aracılığıyla aşağıda görüldüğü gibi alan adı ile ulaşılması mümkündür.



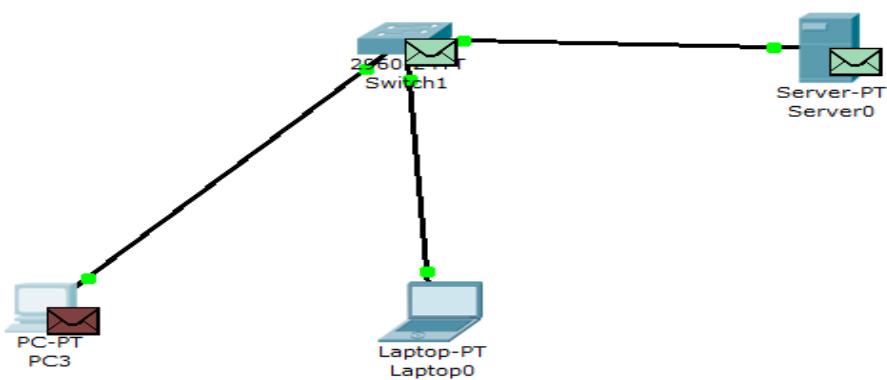
ADIM ADIM PAKET ANALİZİ

Packet Tracer üzerinde, **bottom toolbar** üzerinde yer alan **Realtime Mode** ve **Simulation Mode** dönüşümleri aracılığıyla adım adım gelen-giden paketlerin içeriğinin incelenmesi mümkündür.

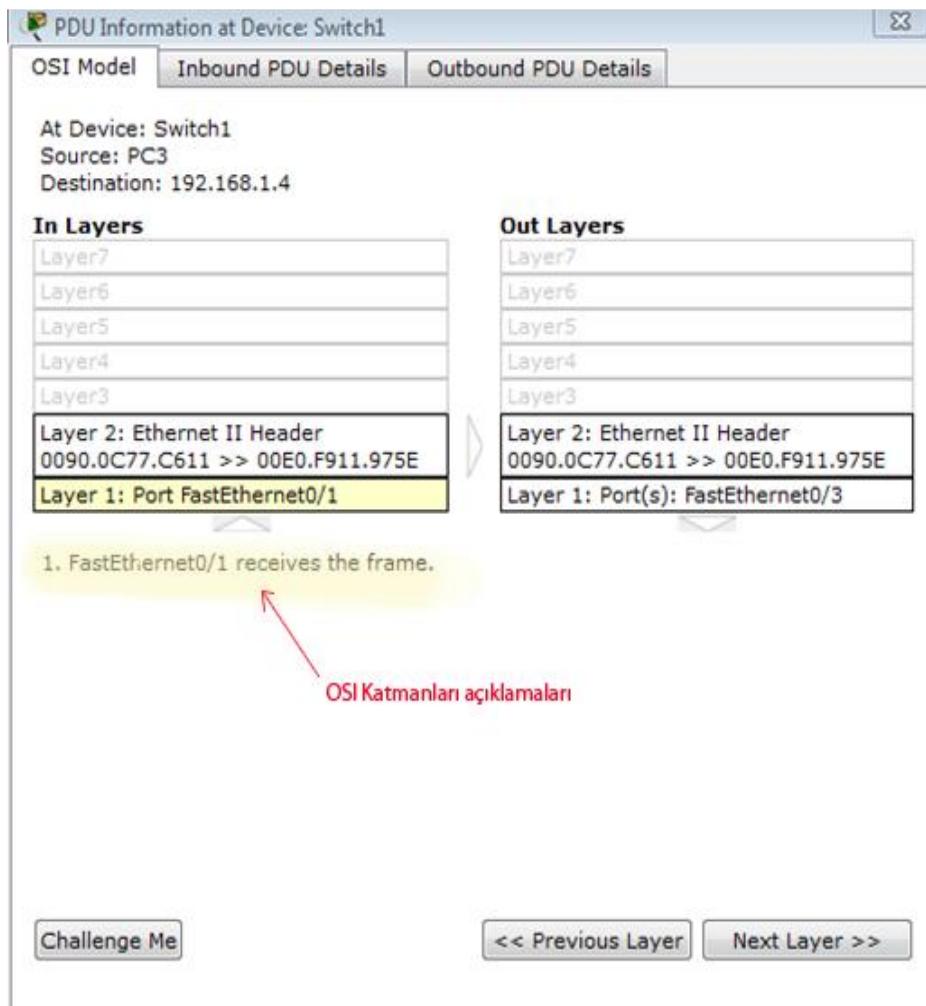




Paketlerin hareketleri adım adım incelenebilir. *Geri*, *ileri* ve *Auto* özelliği ile trafiklerin akışını incelemek mümkündür.



Yukarıdaki örnekte **Simulation** moda PC3'ten gönderilen http ve dns paketleri görülmektedir. Paketler çift tıklanarak incelenebilir.



Next Layer düğmesine tıklanarak adım adım bir çerçevenin ya da paketin OSI katmanı boyunca ne tür işlemlerden geçtiği görülebilir. Bir sonraki katman için Next Layer düğmesi tıklanabilir.

- 1. Sending a valid LACP/PAgP frame to the higher process.
- 2. The frame source MAC address was found in the MAC table of Switch.
- 3. This is a unicast frame. Switch looks in its MAC table for the destination MAC address.

Challenge Me **<< Previous Layer** **Next Layer >>**

Ayrıca INBOUND / OUTBOUND PDU DETAILS tıklanarak, frame yapısı, IP yapısı ve TCP/UDP yapısı da incelenebilir.

Frame Yapısı:

Ethernet II

0	4	8	14	19 Bytes
PREAMBLE: 101010...1011	DEST MAC: 00E0.F911.975E		SRC MAC: 0090.0C77.C611	
TYPE: 0x800	DATA (VARIABLE LENGTH)		FCS: 0x0	

Yukarıdaki yapıda PC ile Server arasındaki iletişimın çerçeve (frame) yapısı görülmektedir.

IP Packet Yapısı:

IP

0	4	8	16	19	31 Bits
	4	IHL	DSCP: 0x0	TL: 44	
			ID: 0x17	0x2	0x0
	TTL: 128		PRO: 0x6	CHKSUM	
			SRC IP: 192.168.1.2		
			DST IP: 192.168.1.4		
			OPT: 0x0	0x0	
			DATA (VARIABLE LENGTH)		

TCP Paket Yapısı:

TCP

0	16	31 Bits
SRC PORT: 1029	DEST PORT: 80	
SEQUENCE NUM: 0		
ACK NUM: 0		
OFF.	RES.	SYN
CHECKSUM: 0x0	WINDOW	URGENT POINTER
OPTION	PADDING	
DATA (VARIABLE)		

ALT AĞLARA BÖLMEK (SUBNETTING)

IPv4 ağlarda IP tasarımını yaparken adresleri verimli kullanmak, ağ yönetimini kolaylaştırmak amacıyla ağları altağlara bölmek (subnetting) gereklidir. Örneğin bir okuldaki öğrencileri, yöneticileri ve öğretmenleri ayrı ağlara almak hem yönetimsel açıdan hem de tasarım açısından büyük kolaylık ve güvenlik sağlayacaktır. Bu sayede örneğin öğrencilerin trafiklerini yöneticilerin ya da öğretmenlerin trafiklerinden ayıralabilir, 3.katman düzeyinde erişim kontrolleri ya da filtrelemeler yapabiliriz.

Örneğin bir okul ağında muhasebe bölümü, bilgisayar bölümü, elektrik bölümü olsun. Yine bu okulda çeşitli amaçlarla sunucuların da bulunduğu düşünelim. Bu okulda sunucuların yönetimini sadece bilgisayar bölümüne vermek isteyelim. Diğer bölümlerdeki kullanıcılar sunuculara sadece kısıtlı servisler ile (örneğin sadece web hizmeti) erişmesini isteyelim. Bu senaryoda tüm bölümler ve sunucular aynı IP aralığında ise bu bölümlerdeki cihazları birbirinden ayırt etmek, kimlerin sunuculara erişim yapacağını ya da yapamayacağını belirlemek güçleşir. Bu nedenle IP tasarımını yaparken her bölümün ve sunucuları ayrı ayrı IP ağlarına dahil etmek gerekecektir.

“Ayrı IP ağlarda olma” kavramı subnet mask ile belirlenir. Bu nedenle IP adreslerini ve Subnet Mask kavramını iyi anlamak gereklidir.

IP adres yapısındaki, ağ ve host kısımlarını bulmak için kullanılan bu değişkeni ALT AĞ MASKESİ (Subnet Mask) denir.

Cihazımız bu IP adresinin host ve ağ kısmını öğrendiğine göre sıra bizim daha rahat okuyacağımız şekle yani onluk sisteme dönüştürmeye geliyor.

IP ADRESİMİZ : **192.168.1.5**

ALT AĞ MASKESİ : **255.255.255.0**

Alt ağ maskesi / ifadesi ile de gösterebiliriz. Alt ağ maskesinde 1 olan değerlerin sayısını / işaretinden sonra yazarız. Örneğin;

IP Adresi 192.168.1.5 , Subnet Mask : 255.255.255.0 olan bir ifadeyi: 192.168.1.5 / 24 olarak gösterebiliriz. Buradaki 24 ifadesi subnet mask adresindeki 1 olan bitlerin sayısını gösterir.

IP adresinin network ve host kısımlarını belirleyebilmek için sürekli olarak maske ile birlikte kullanılması gereklidir. Cihazlara IP adresi verirken verilen IP adresinin sınıfına göre varsayılan olarak bir maske atanır.

Ancak bu değer değiştirilebilir. Aşağıdaki örnekte B sınıfı bir adres verildiğinden varsayılan olarak B sınıfının maskesi 255.255.0.0 kullanılmıştır.

Otomatik olarak bir IP adresi al
 Aşağıdaki IP adresini kullan:

IP adresi:	172 . 16 . 1 . 2
Alt ağ maskesi:	255 . 255 . 0 . 0
Varsayılan ağ geçidi:	. . .

Yani yukarıdaki IP adresini 172.16.1.2/16 olarak gösterebiliriz. (255.255.0.0 maskesi bit bazında yazıldığında 16 tane “1” içerdığından)

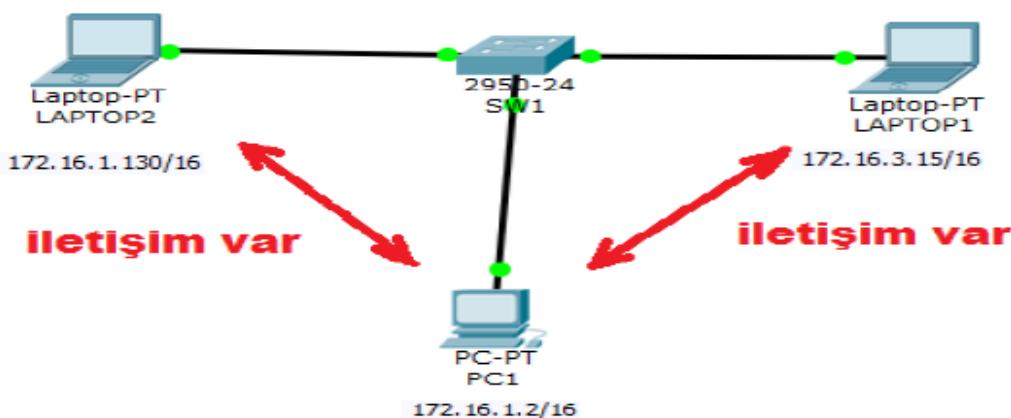
O halde herhangi bir IP’nin 172.16.1.2/16 ile aynı ağda olabilmesi için, o IP’nin ilk 16 bitinin 172.16.1.2 IP’sinin ilk 16 biti ile aynı olması gereklidir.

Aşağıdaki tabloda 172.16.1.2/16 IP adresi ile diğer IP’lerin aynı aralıkta olup olmadığı karşılaştırılmıştır. Maske 255.255.0.0 (yani /16) olmak üzere;

Onluk Olarak IP	Binary Olarak IP (Maske /16)	
172.16.1.2	10101100.00010000.00000001.00000010	
192.168.1.2	11000000.10101000.00000001.00000010	Aynı ağda değil
172.16.3.15	10101100.00010000.00000011.00001111	Aynı ağda
172.16.1.130	10101100.00010000.00000001.10000010	Aynı ağda

255.255.0.0 maskesine göre 172.16.1.2 ile 172.16.3.130 IP adresleri **aynı ağdadır ve birbirleri ile doğrudan iletişime gecebilirler.**

Aşağıdaki Packet Tracer görüntüsünde PC1 ile LAPTOP1 doğrudan iletişime (Ör. ping) gecebilirler.



172.16.1.2'den 172.16.3.15' gönderilen ping paketi başarılı olmuştur

```
PC>ping 172.16.3.15
```

```
Pinging 172.16.3.15 with 32 bytes of data:  
  
Reply from 172.16.3.15: bytes=32 time=1ms TTL=128  
Reply from 172.16.3.15: bytes=32 time=0ms TTL=128  
Reply from 172.16.3.15: bytes=32 time=2ms TTL=128  
Reply from 172.16.3.15: bytes=32 time=0ms TTL=128  
  
Ping statistics for 172.16.3.15:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 0ms, Maximum = 2ms, Average = 0ms
```

Yine 172.16.1.2'den 172.16.1.130'a gönderilen ping paketi başarılı olmuştur

```
PC>ping 172.16.1.130
```

```
Pinging 172.16.1.130 with 32 bytes of data:  
  
Reply from 172.16.1.130: bytes=32 time=1ms TTL=128  
Reply from 172.16.1.130: bytes=32 time=0ms TTL=128  
Reply from 172.16.1.130: bytes=32 time=0ms TTL=128  
Reply from 172.16.1.130: bytes=32 time=0ms TTL=128  
  
Ping statistics for 172.16.1.130:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Şimdi aynı IP adresi için bu kez maskeyi değiştirip tekrar inceleyelim.

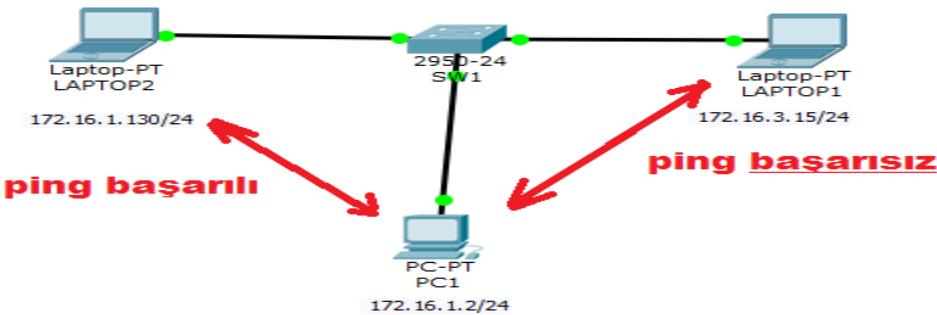
Aşağıdaki IP adresini kullan:

IP adresi:	172 . 16 . 1 . 2
Alt ağ maskesi:	255 . 255 . 255 . 0
Varsayılan ağ geçidi:

Bu durumda IP ve Maske: 172.16.1.2/24 olacaktır. Yani aynı aralıkta olmak için ilk 24 bit benzemelidir. Tekrar tabloda inceleyelim.

Onluk Olarak IP	Binary Olarak IP (Maske /24)	
172.16.1.2	10101100.00010000.00000001.00000010	
192.168.1.2	11000000.10101000.00000001.00000010	Aynı ağda değil
172.16.3.15	10101100.00010000.00000011.00001111	Aynı ağda değil
172.16.1.130	10101100.00010000.00000001.10000010	Aynı ağda

O halde 255.255.255.0 yani /24 maskesine göre 172.16.1.2 ile 172.16.1.130 aynı ağda iken; aynı maskeye göre 172.16.1.2 ile 172.16.3.15 aynı ağda değildir. Çünkü ilk 24 bitleri aynı değildir.



*** Aynı ağda olmayan cihazlar birbirleri ile doğrudan iletişime geçemeyizler 3.katman bir cihaz (Örneğin Router) aracılığıyla haberleşebilirler.

```
PC>ping 172.16.3.15
Pinging 172.16.3.15 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 172.16.3.15:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

172.16.1.2 IP adresi ile 172.16.3.15 IP adresinin ilk 24 biti (Çünkü maske 255.255.255.0) aynı olmadığı için aynı ağda değildirler. Bu nedenle doğrudan iletişimleri yoktur.

Ancak aşağıda görüldüğü gibi 172.16.1.2 ile 172.16.1.130 arasında iletişim vardır.

```
PC>ping 172.16.1.130
Pinging 172.16.1.130 with 32 bytes of data:
Reply from 172.16.1.130: bytes=32 time=1ms TTL=128
Reply from 172.16.1.130: bytes=32 time=2ms TTL=128
Reply from 172.16.1.130: bytes=32 time=0ms TTL=128
Reply from 172.16.1.130: bytes=32 time=0ms TTL=128

Ping statistics for 172.16.1.130:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms
```

172.16.1.2 ile 172.16.1.130 cihazlarının aynı ağıdadır. Çünkü ilk 24 biti aynıdır.

Aşağıdaki tablo dikkatli bir şekilde incelenirse 172.16.1.2 ile 172.16.1.130 IP'lerinin ilk 24 bitinin aynı olduğu ancak 25.bitten itibaren farklılığıın başladığı görülür.

Onluk Olarak IP	Binary Olarak IP (Maske /24)	
172.16.1.2	10101100.00010000.00000001.00000010	
172.16.1.130	10101100.10101000.00000001.10000010	İlk 24 bit aynı
172.16.1.65	10101100.10101000.00000001.01000001	İlk 25 bit aynı

Yukarıdaki tabloya göre 172.16.1.2 ile 172.16.1.130'un ilk 24 biti aynıdır. Oysa 172.16.1.2 ile 172.16.1.65'in ilk 25 biti aynıdır.

Eğer aynı ağda olmak için ilk 25 bite bakmış olsaydık 172.16.1.2 ile 172.16.1.130 farklı ağlarda olacak; 172.16.1.2 ile 172.16.1.65 ise aynı ağda olacaktır.

Yani 172.16.1.2/25 ile 172.16.1.130/25 aynı ağda değildir. O halde bu cihazlara maskeyi /25 şeklinde verirsek bu iki cihaz artık doğrudan haberleşemezler.

/25 demek, IP adresinin ilk 25 bitinin AĞ KISMI olduğunu geri kalan 7 bitin ise (Çünkü IP 32 bit) HOST KISMI olduğunu gösterir.

/25 'i bit olarak yazalım:

11111111.11111111.11111111.10000000

Ondalık sayıya çevirelim:

255.255.255.128

Şimdi de IP tasarımini bu maskeye göre yapalım:

PC1'in yapılandırması aşağıdaki gibi olacaktır.

<input type="radio"/> DHCP	<input checked="" type="radio"/> Static
IP Address	172.16.1.2
Subnet Mask	255.255.255.128
Default Gateway	

Laptop2'nin yapılandırması işe aşağıdaki gibi olacaktır.

<input type="radio"/> DHCP	<input checked="" type="radio"/> Static
IP Address	172.16.1.130
Subnet Mask	255.255.255.128
Default Gateway	

Bu iki cihaz doğrudan haberleşemezler.

```
PC>ping 172.16.1.130
Pinging 172.16.1.130 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 172.16.1.130:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

/25 maskesine göre 172.16.1.2 ile 172.16.1.65 aynı ağda iken, 172.16.1.2 ile 172.16.1.130 aynı ağda değildir. Şimdi de bu maskeye göre 172.16.1.2 ile hangi IP'lerin aynı ağda olduğunu hesaplayalım. Buna göre ilk 25 bit değiştirmemek şartıyla son 7 bit ne olursa olsun tüm bu IP'ler 172.16.1.2 ile aynı ağda olduğunu söyleyebiliriz.

Aşağıdaki tablo bu IP'leri daha net gösterecektir.

Binary Olarak IP (Maske /25)	Onluk IP	
10101100.00010000.00000001.00000000	172.16.1.0	Bu bloktaki IP'lerin ilk 25 biti kendi arasında aynıdır.
10101100.00010000.00000001.00000001	172.16.1.1	
10101100.00010000.00000001.00000010	172.16.1.2	
10101100.00010000.00000001.00000011	172.16.1.3	
10101100.00010000.00000001.00000100	172.16.1.4	
10101100.00010000.00000001.00000101	172.16.1.5	
.....	
10101100.00010000.00000001.01111110	172.16.1.126	Bu bloktaki IP'lerin ilk 25 biti kendi arasında aynıdır.
10101100.00010000.00000001.01111111	172.16.1.127	
10101100.00010000.00000001.10000000	172.16.1.128	
10101100.00010000.00000001.10000001	172.16.1.129	
10101100.00010000.00000001.10000010	172.16.1.130	Toplam 128 IP
10101100.00010000.00000001.10000011	172.16.1.131	
.....	
10101100.00010000.00000001.11111110	172.16.1.254	Toplam 128 IP
10101100.00010000.00000001.11111111	172.16.1.255	

/25 maskesine göre 172.16.1.0 ile 172.16.1.127 aralığındaki tüm IP'ler aynı ağdadır.

Aynı maskeye göre 172.16.1.128 ile 172.16.1.255 aralığındaki tüm IP'ler farklı bir ağdadır.

Şimdi yapıyı tekrar düşünelim:

/24 olduğunda 172.16.1.0 ile 172.16.1.255 arasındaki tüm IP'ler aynı ağdadır. Yani tek bir ağ vardır. Oysa /25 olduğunda bu kez iki farklı ağ olacaktır.

172.16.1.0 ----- 172.16.1.127 (ilk Alt Ağ)

172.16.1.128 ----- 172.16.1.255 (ikinci Alt Ağ)

Böylece 172.16.1.0/24 olan toplam 156 IP'lik ağımızı her birinde 128 IP bulunan iki farklı ağa bölmüş olduk. Bu işlem subnetting olarak adlandırılır.

Subnet Mask, bizlere IP'ye nasıl bakacağımızı gösteren bir değişkendir, bir maskedir. Subnet mask, IP adresini NETWORK ve HOST olmak üzere iki ayırr. Ancak burada kullanılabilir IP

adresi ile ilgili durumu unutmamak gererkir. Yani, IP'nin host bitlerinin tümünün "0" ya da tümünün "1" olduğu durumlardaki IP'lerin cihazlara verilemediğini unutmamak gereklidir.

Buna göre 172.16.1.0/25 ve 172.16.1.128/25 IP adresleri bir cihaza verilemez. Çünkü son 7 bit "0"dur. Bu adreslere Network Adresi denir ve ağı temsil eden bir adresdir. Yani 172.16.1.0 ile 172.16.1.127 arasındaki tüm IP'ler 172.16.1.0/25 ağındır bu IP'lerin ağ adresi 172.16.1.0/25 tir.

Yine 172.16.1.127/25 ile 172.16.1.255/25 IP adresleri de cihazlara verilemez. Çünkü son 7 bit "1" dir. Bu adreslere de Broadcast adresler denir. Bu ağdaki tüm cihazlara bir bilgi gönderileceğine bu adresler kullanılır.

Örneğin yukarıdaki Subnet'ler için;

172.16.1.127'ye gönderilen bir IP paketi tüm ilk Subnet bilgisayarlarına (172.16.1.1 ile 172.16.1.126 arasındaki tüm IP'ler) gider ve tüm bu bilgisayarlar bu paketi alıp gerekli işlemleri yapar.

Şimdi de 192.168.1.0/24 ağının subnetting işlemlerini yapalım.

Normalde 192.168.1.X IP adresleri (192.168.1.0 olarak ifade edilebilir) C sınıfı IP adresleri olduğundan varsayılan maskeleri 255.255.255.0 yani /24 tür.

192.168.1.0/24 ağının sınırları aşağıdaki tabloda belirtilmiştir.

192.168.1.0/24 ağındaki IP'ler		
IP Adresi (Decimal)	IP Adresi Binary	Açıklama
192.168.1.0	11000000.10101000.00000001.00000000	Kullanılabilir IP değil. (Son 8 bit "0")
192.168.1.1	11000000.10101000.00000001.00000001	Kullanılabilir ilk IP
192.168.1.2	11000000.10101000.00000001.00000010	Kullanılabilir ikinci IP
.....	
192.168.1.254	11000000.10101000.00000001.11111110	Kullanılabilir son IP
192.168.1.255	11000000.10101000.00000001.11111111	Kullanılabilir IP değil (Son 8 bit "1")

Bu tabloda 192.168.1.0/24 ağında toplam 256 IP adresi vardır. Ancak bunlardan ikisi kullanılabilir (cihazlara atanabilir) IP değildir.

Maske /25 olduğunda bu yukarıdaki ağ iki farklı ağa bölünmüştür. Yani 25. biti "0" olan IP'ler ve "1" olanlar.

/26 olduğunda ise bu kez yukarıdaki ağ (192.168.1.0/24) dört farklı parçaya ayrılır. Bu adresler aşağıdaki tabloda gösterilmiştir.

192.168.1.0/24 ağındaki IP'ler (4 subnet)			
IP Adresi (Decimal)	IP Adresi (Binary)	Açıklama	Subnet
192.168.1.0	11000000.10101000.00000001.00 000000	Kullanılamaz. Son 6 bit "0"	İlk Subnet (*Subnet Zero)
192.168.1.1	11000000.10101000.00000001.00 000001	Kullanılabilir ilk IP	
192.168.1.2	11000000.10101000.00000001.00 000010	Kullanılabilir ikinci IP	
.....	
192.168.1.62	11000000.10101000.00000001.00 111110	Kullanılabilir son IP	
192.168.1.63	11000000.10101000.00000001.00 111111	Kullanılamaz son 6 bit "1" Broadcast adresi	
192.168.1.64	11000000.10101000.00000001.01 000000	Kullanılamaz. Son 6 bit "0"	İkinci Subnet
192.168.1.65	11000000.10101000.00000001.01 000001	Kullanılabilir ilk IP	
.....	
192.168.1.126	11000000.10101000.00000001.01 111110	Kullanılabilir son IP	
192.168.1.127	11000000.10101000.00000001.01 111111	Kullanılamaz son 6 bit "1" Broadcast adresi	
192.168.1.128	11000000.10101000.00000001.10 000000	Kullanılamaz. Son 6 bit "0"	Üçüncü Subnet
192.168.1.129	11000000.10101000.00000001.10 000001	Kullanılabilir ilk IP	
.....	Kullanılabilir ikinci IP	
192.168.1.190	11000000.10101000.00000001.10 111110	
192.168.1.191	11000000.10101000.00000001.10 111111	Kullanılabilir son IP	
192.168.1.192	11000000.10101000.00000001.11 000000	Kullanılamaz. Son 6 bit "0"	Dördüncü Subnet
192.168.1.193	11000000.10101000.00000001.11 000001	Kullanılabilir ilk IP	
.....	
192.168.1.254	11000000.10101000.00000001.11 111110	Kullanılabilir son IP	
192.168.1.255	11000000.10101000.00000001.11 111111	Kullanılamaz son 6 bit "1" Broadcast adresi	

192.168.1.0/24 ağının maskesi /26 olarak ayarlanırsa network 4 ağa ayrılmış olur. Yani 24 olan maskeye 2 ekstra bit eklendiğinde 4 alt ağa ayrılmış olur. Çünkü iki bit 4 farklı değer alabilir.

24 + 2 olduğunda Network 4'e ayrılır.

24 + 3 olduğunda ise Network 8'e ayrılır. (3 bit 8 farklı değer alır)

O halde “n” alt ağa maskesine eklenen yeni bitler olmak üzere; 2^n alt ağ oluşur.

* Yukarıdaki örnekte 192.168.1.0 ağının 4 ağa ayrılmıştır. **Bu ağlardan ilk ağ 1.subnet denmez.** Yani sayma işlemi için sayma sayıları kümesi kullanılmaz ☺ İlk subnete 0.Subnet (**Subnet Zero**) denir.

SUBNETTING ÖRNEKLER

1.ÖRNEK:

192.168.2.0/24 ağını 8 eşit parçaya ayıralım.

Çözüm: 8 alt ağ oluşturmak için 3 bit yeterlidir. Yani host bitlerinin 3 tanesini network bitine kaydıracağız.

Normalde /24 iken 24-bit network; 8-bit host bitidir. 3-bit kaydırduğumuz için yeni durumda Network bitinin sayısı $24+3=27$ olacak, host bitlerinin sayısı ise 3 azalacak ($8-3$) 5 bit kalacaktır.

$/27 = \textcolor{red}{1111111.1111111.1111111.11}00000$ (27 tane “1”)

Yani 255.255.255.224'tür.

Şimdi oluşacak olan bu 8 alt ağın Ağ Adreslerini bulalım:

- Host için 5 bit varsa, her grupta oluşacak adres sayısı $2^5 = 32$ 'dir. O halde son oktette 32'nin katları network adresleri olacaktır.

/27'ye göre subnetting		Ağ Adresi
32'nin 0 katı; ($32*0=0$)	0	192.168.2. 0
32'nin 1 katı ($32*1=32$)	32	192.168.2. 32
32'nin 2 katı ($32*2=64$)	64	192.168.2. 64
32'nin 3 katı ($32*3=96$)	96	192.168.2. 96
32'nin 4 katı ($32*4=128$)	128	192.168.2. 128
32'nin 5 katı ($32*5=160$)	160	192.168.2. 160
32'nin 6 katı ($32*6=192$)	192	192.168.2. 192
32'nin 7 katı ($32*7=224$)	224	192.168.2. 224

Bu tablodaki adresler Ağ Adresleridir. Bu alt ağların broadcast adresleri, bir sonraki ağ adresinin bir eksigidir.

Örneğin 192.168.2.64 ağının broadcast adresi 192.168.1.95 'tir (96-1)

Bu sekiz ağın adres aralıklarını aşağıdaki tabloda görebilirsiniz.

Network Adresi	Kullanılabilir ilk IP	Kullanılabilir Son IP	Broadcast Adresi	Maske	Bu ağdaki kullanılabilir IP sayısı
192.168.2.0	192.168.2.1	192.168.2.30	192.168.2.31	255.255.255.224	30
192.168.2.32	192.168.2.33	192.168.2.62	192.168.2.63	255.255.255.224	30
192.168.2.64	192.168.2.65	192.168.2.94	192.168.2.95	255.255.255.224	30
192.168.2.96	192.168.2.97	192.168.2.126	192.168.2.127	255.255.255.224	30
192.168.2.128	192.168.2.129	192.168.2.158	192.168.2.159	255.255.255.224	30
192.168.2.160	192.168.2.161	192.168.2.190	192.168.2.191	255.255.255.224	30
192.168.2.192	192.168.2.193	192.168.2.222	192.168.2.223	255.255.255.224	30
192.168.2.224	192.168.2.225	192.168.2.254	192.168.2.255	255.255.255.224	30

2.ÖRNEK:

10.0.0.0/8 ağını 4 eşit parçaya ayıralım. Bu parçalardan ikincisinin aralığını bulalım.

Çözüm: 4'e ayırmak için 2 bit transferi yapmalıyız. Yani 24 host bitinden 2'sini network bitlerine ekleyeceğiz. O halde network biti sayısı 10 olur. (8+2)

Yani maske /10 =11111111.11000000.00000000.00000000 (=255.192.0.0)

Yine pratik yoldan bu dört alt ağı bulalım.

İlk olarak bölümlemenin hangi oktette olduğunu görmek gereklidir. Maskenin 1'lerden 0'lara geçtiği yer ikinci oktetidir. O halde ikinci oktet değişecektir.

İkinci oktette 2 tane network biti 6 tane host biti vardır. (**11000000**)

6-bit 64 farklı değer üretir. ($2^6=64$) O halde ikinci oktet 64'ün katları olarak gider.

10.0.0.0

10.64.0.0

10.128.0.0

10.192.0.0

Aralıklar ise aşağıdaki gibi olacaktır;

İlk Subnet : 10.0.0.0 – 10.63.255.255

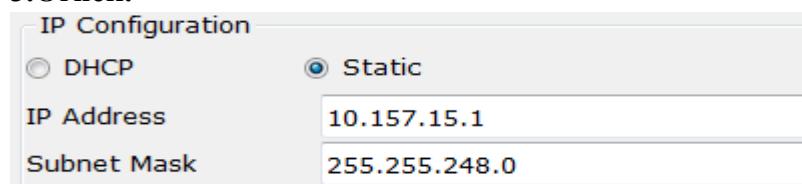
İkinci Subnet : 10.64.0.0 – 10.127.255.255

Üçüncü Subnet : 10.128.0.0 – 10.191.255.255

Dördüncü Subnet : 10.192.0.0 – 10.255.255.255

Burada ilk adresler Network Adresi, son adresler ise broadcast adreslerdir. Aradaki tüm IPler kullanılabilir IPlerdir. Örneğin 10.13.255.255 kullanılabilir bir IP adresidir ve ilk subnette yer alır.

3.Örnek:



Şekildeki IP ve maske yapılandırmasına göre, bu bilgisayarın dahil olduğu ağın **Ağ Adresi**, **Broadcast Adresi**, **kullanılabilir ilk IP adresi** ve **kullanılabilir son IP adresini** bulalım.

Çözüm:

Subnet maskesinin ilk iki okteti tamamen "1"lerden, son oktet ise tamamen "0"lardan oluşuyor. Oysa üçüncü oktet bit bazında yazıldığındá "1111000" şeklindedir. Yani maske bize IP'nin üçüncü oktetini ile ilgileneceğimizi söyler.

Üçüncü oktetteki host bitleri 3 tanedir. 3 bit 8 farklı değer üretir. O halde IP'nin üçüncü oktetinde 8'in katlarını bulmalıyız. Yani; X 8'in katları olmak üzere 10.157.X.0 network adresleridir.

10.157.**0**.0 → 10.157.7.255'e kadar gider

10.157.**8**.0 → 10.157.15.255'e kadar gider. (* Bizim PC bu aralıktadır)

10.157.**16**.0 → 10.157.23.255'e kadar gider.

10.157.**24**.0

.....

10.157.240.0

Yani 10.157.15.1 IP adresi 255.255.248.0 maskesine göre ;

Ağ Adresi : 10.157.8.0

Kullanılabilir İlk IP : 10.157.8.1

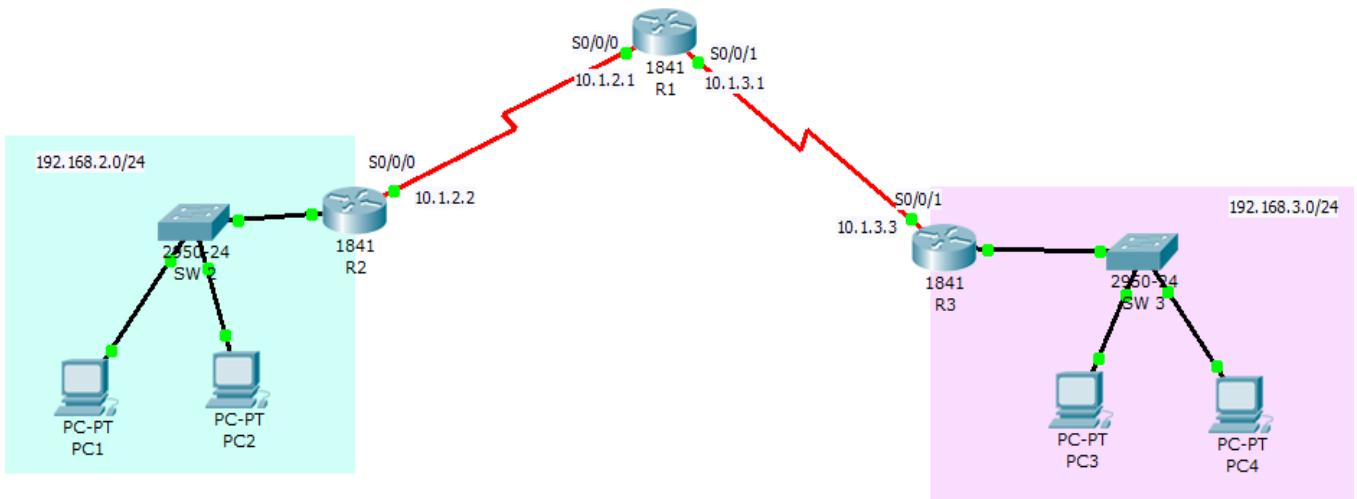
Kullanılabilir son IP : 10.157.15.254

Broadcast Adres : 10.157.15.255 dir.

ROUTING ve YÖNLENDİRME TABLOSU

Yönlendirme Tablolarının anlaşılabilmesi için bir yönlendiricinin çalışma prensibinin iyi bilinmesi gereklidir. Bir yönlendirici kendisine gelen bir paketi ilgili yere yönlendirmek için öncelikle hedef adresine bakar. Bu hedef adresinin hangi aralığa denk geldiği hesaplar ve paketin iletilmesi gereken hedef ağ adresi bulur. Bulunan bu ağ adresine ulaşmak için yönlendiricinin paketi hangi arayüzden çıkarması gerektiğini bilmesi gerekmektedir. İşte hedef ağ adreslerinin ve bu adreslere ulaşmak için hangi arayüzden çıkarılması gereği bilgisi yönlendiriciler üzerinde bir tablo tutulur. Bu tabloya yönlendirme tablosu denir. Her yönlendirici kendisine ait böyle bir tablo tutar.

Aşağıdaki örnektan yola çıkarak R1 yönlendiricisinin yönlendirme tablosunu inceleyelim.



Açıklamalar;

R2 yönlendiricisinin yerel ağında IP adresi 192.168.2.... ile başlayan cihazlar bulunsun. Bu durumda varsayılan olarak bu ağa 192.168.2.0 ağını diyebiliriz. Aynı şekilde R3 cihazına bağlı yerel ağa da 192.168.3.0 ağını diyebiliriz.

R1 yönlendiricisinin 192.168.2.0 ve 192.168.3.0 ağlarına nasıl ulaşacağını (rota) bilmesi gereklidir.

R1 için ;

192.168.2.0 ağına erişim R1 yönlendiricisinin S0/0/0 portundan çıkış ile sağlanır. Aynı şekilde 192.168.3.0 ağına erişim ise S0/0/1 portundan çıkış ile sağlanabilir. Yani R1 cihazına gelen herhangi bir paketin hedef IP adresi örneğin 192.168.2.... ile başlıyorsa (192.168.2.0/24) yönlendirici bunu S0/0/0 arayüzünden çıkarması gerekecektir.

Bu görevleri gerçekleştirebilmek için R1 yönlendiricisinin yönlendirme tablosu aşağıda gibi olmalıdır.

Type	Network	Port
C	10.1.2.0/24	Serial0/0/0
C	10.1.3.0/24	Serial0/0/1
S	192.168.2.0/24	Serial0/0/0
S	192.168.3.0/24	Serial0/0/1

Yönlendirme tablosu incelendiğinde hedef ağ adresleri Network başlığı altında, çıkış arayüzü ise Port başlığı altında görüntülenir. Yine bu tabloda bu rotaların nasıl öğrenildiğini bildiren Type alanında S,C,D, R.. gibi harfler gösterilmektedir.

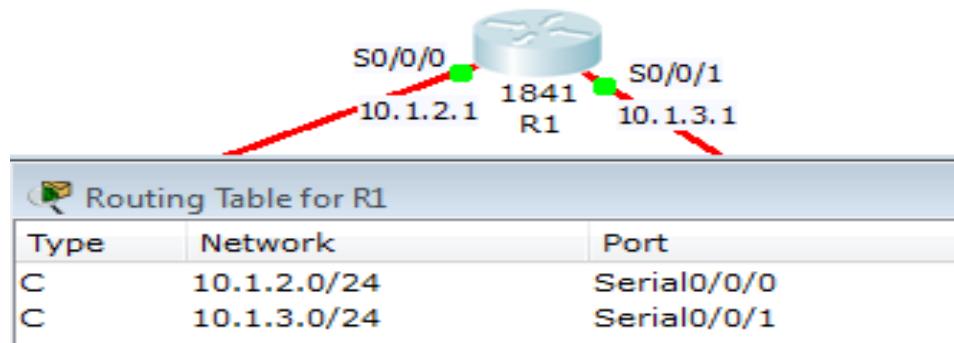
Yönlendirme Tablosu Nasıl Oluşturulur?

Herhangi bir yönlendiricinin tablosundaki bu rota satırları, statik, dinamik ve direk bağlı olmak üzere 3 farklı yöntemle oluşturulabilir.

1. Direk Bağlı Rotalar (C):

Yönlendiriciye direk olarak bağlı olan rotalar, “C” harfi ile gösterilir. Yönlendiricinin arayüzüne IP adresi verilip port açıldığında bu Ağ adresleri otomatik olarak tabloya eklenir. Yönlendiriciler bu ağ adreslerinden başka ağ adreslerini bilemezler. Yani bir yönlendirici varsayılan olarak sadece bu ağları tanır. Diğer yönlendiricilere bağlı olan ağ adreslerini bilemez. Paketlerin diğer ağlara iletilebilmesi için diğer rotaların statik olarak eklenmesi, ya da yönlendirme protokolleri aracılığıyla dinamik olarak öğrenilmesi gerekmektedir.

Yukarıdaki örnekten yola çıkarsak, başlangıçta R1 yönlendiricisinin yönlendirme tablosu aşağıdaki gibidir.



2. Statik Yönlendirme (S):

Bir yönlendirici, kendisine direkt bağlı olmayan rotaları bilemez. Bu sebeple bu rotaların yönlendiriciye öğretilmesi gereklidir. Rotaların bir yönetici tarafından eklenmesine statik yönlendirme denir ve tabloda “S” harfi ile gösterilir.

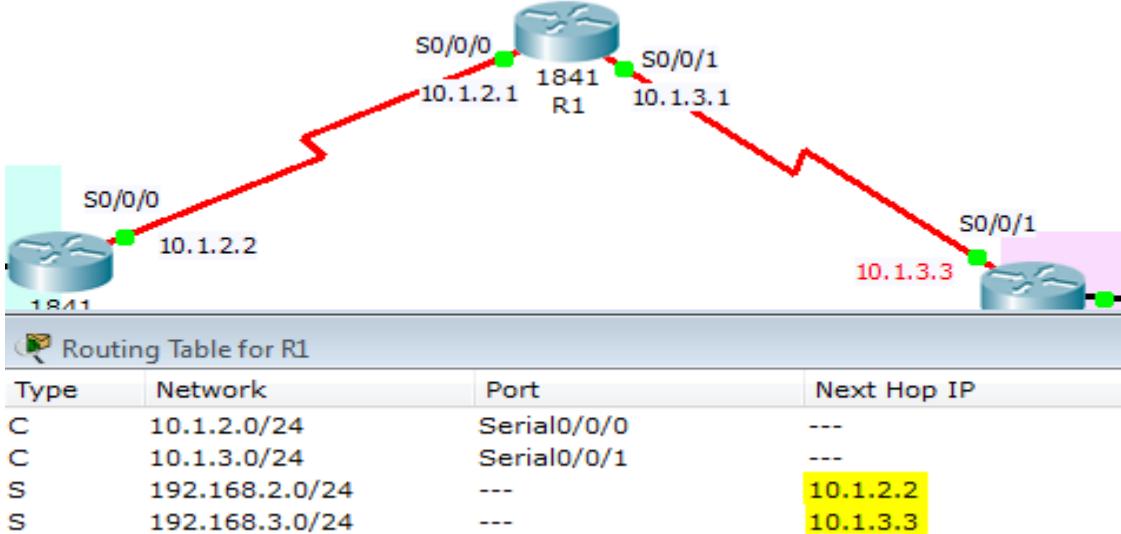
Yine yukarıdaki örnekten yola çıkarsak, R1 yönlendiricisine R2 ‘ye bağlı ağı (192.168.2.0/24) ve R3’e bağlı ağı (192.168.3.0/24) aşağıdaki komutlar ile öğretmek gereklidir. Bu komutun üreticiye göre değişkenlik göstereceği unutulmamalıdır.

```
R1(config)# ip route 192.168.2.0 255.255.255.0 s0/0/0
R1(config)# ip route 192.168.3.0 255.255.255.0 s0/0/1
```

Bu komutlar girildikten sonra R1’in yönlendirme tablosunda “S” ile işaretlenmiş statik rotalar görüntülenecektir.

Type	Network	Port
C	10.1.2.0/24	Serial0/0/0
C	10.1.3.0/24	Serial0/0/1
S	192.168.2.0/24	Serial0/0/0
S	192.168.3.0/24	Serial0/0/1

Böylece hedef ağlara ulaşmak için hangi çıkışların kullanılacağı yönlendiriciye öğretilebilir. Ancak yapılandırmanız göre hedef ağlara ulaşmak için çıkış arayüzü (port) yerine bir sonraki yönlendirici adresi (Next Hop) de gösterilebilir. Aşağıdaki yönlendirme tablosunda çıkış arayüzleri yerine bir sonraki yönlendirici adresinin yazıldığına dikkat ediniz.



Burada 192.168.2.0/24 ağına ulaşmak için S0/0/0 çıkış arayüzü yerine bir sonraki yönlendiricinin IP adresinin (10.1.2.2) yazıldığına dikkat ediniz.

3. Dinamik Yönlendirme:

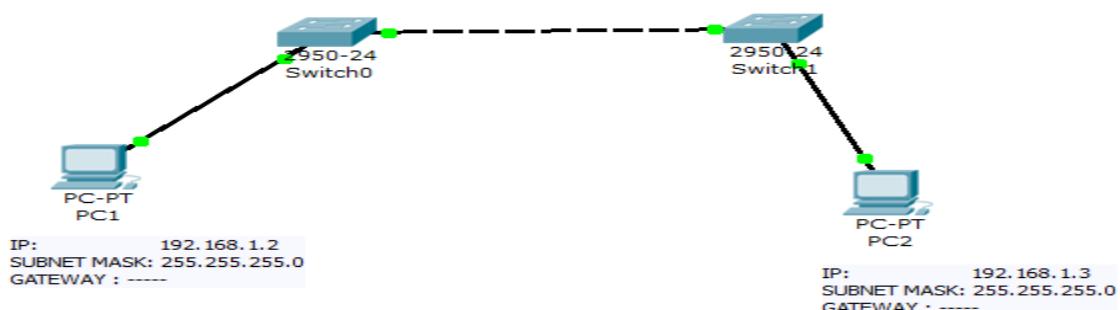
Dinamik yönlendirme, özellikle büyük ağlarda rota sayısının fazla olduğu ağlarda dinamik olarak rotaların öğrenilmesini amaçlar. Bunun için RIP, IGRP, EIGRP, OSPF gibi yönlendirme protokollerı kullanılır. Böyle bir durumda yönlendiriciler birbirlerine kendi ağ adresleri hakkında güncellemeler yaparlar. Böylece diğer yönlendiricilerin öğrenmeleri sağlanmış olur.

Yukarıdaki örnek topoloji için bu kez Dinamik Yönlendirme Protokollerinden RIP kullanılmış ve Yönlendirme Tablosu aşağıdaki gibi dinamik olarak oluşmuştur.

Type	Network	Port	Next Hop IP	Metric
C	10.1.2.0/24	Serial0/0/0	---	0/0
C	10.1.3.0/24	Serial0/0/1	---	0/0
R	192.168.2.0/24	Serial0/0/0	10.1.2.2	120/1
R	192.168.3.0/24	Serial0/0/1	10.1.3.3	120/1

Bu tür bir yönlendirmede hem çıkış arayüzünün hem de sonraki yönlendirici IP adresinin birlikte görüntüülüğüne dikkat ediniz.

ÖRNEK İLETİŞİMLER



Şekildeki PC1 ile PC2 fiziksel ve mantıksal olarak aynı ağdadır. Mantıksal olarak aynı ağa olmaları, aynı IP aralığında olmalarından kaynaklanır. (192.168.1.0/24) Bu durumda PCler arasında iletişim vardır. Aşağıda PC1 'den PC2'ye iletişimin başarılı olduğu gösterilmektedir.

```
Packet Tracer PC Command Line 1.0
PC>ipconfig

IP Address.....: 192.168.1.2
Subnet Mask....: 255.255.255.0
Default Gateway.: 0.0.0.0

PC>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 192.168.1.3: bytes=32 time=15ms TTL=128
Reply from 192.168.1.3: bytes=32 time=60ms TTL=128
Reply from 192.168.1.3: bytes=32 time=27ms TTL=128
Reply from 192.168.1.3: bytes=32 time=35ms TTL=128

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 15ms, Maximum = 60ms, Average = 34ms

PC>
```

Yukarıdaki örnekte görüldüğü gibi, PC1'in Default Gateway adresine ihtiyaç yoktur. Çünkü tek bir fiziksel ağ vardır ve ağın dışı diye bir kavram olmadığı için Default Gateway yazmak gereksizdir.



Bu yapıda ise PCler fiziksel olarak aynı ağa bulunmalarına rağmen, mantıksal olarak farklı ağlardadırlar. PC1, 192.168.2.0/24 ağında, PC2 ise 192.168.1.0/24 ağındadır. Bu durumda iletişim sağlanamazlar. Aşağıda bu durumda iletişim isteğin Request Timeout olarak geri döndürüldüğü görülmektedir.

```

PC>ipconfig

IP Address.....: 192.168.2.2
Subnet Mask....: 255.255.255.0
Default Gateway.: 0.0.0.0

PC>ping 192.168.1.3

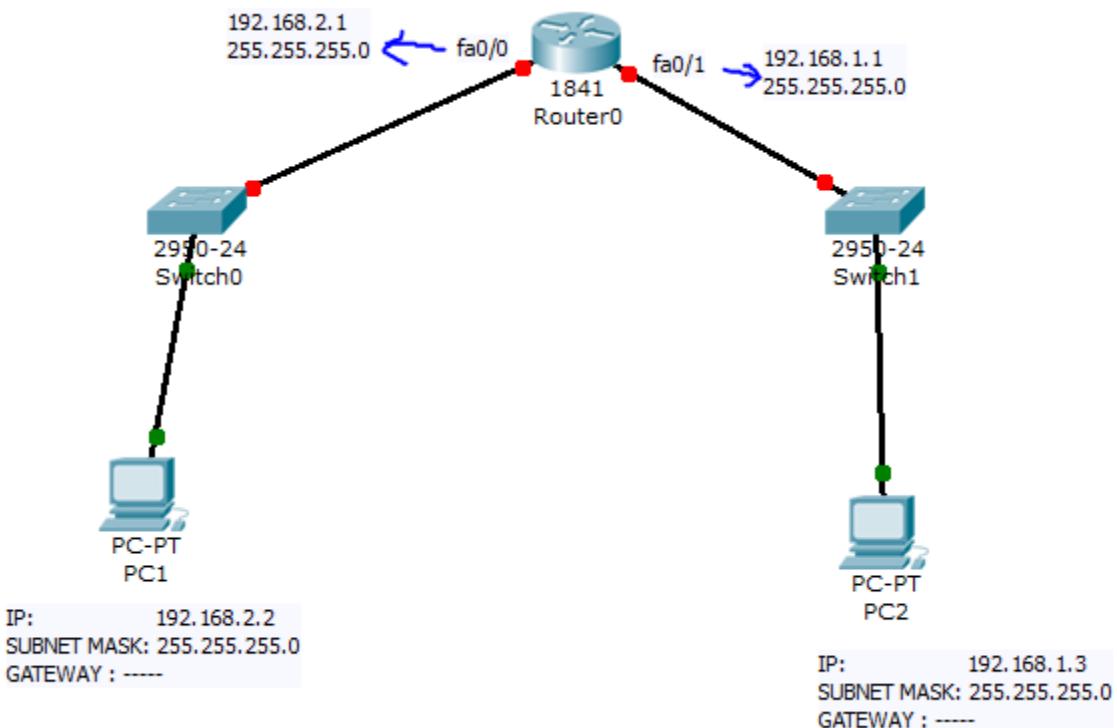
Pinging 192.168.1.3 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

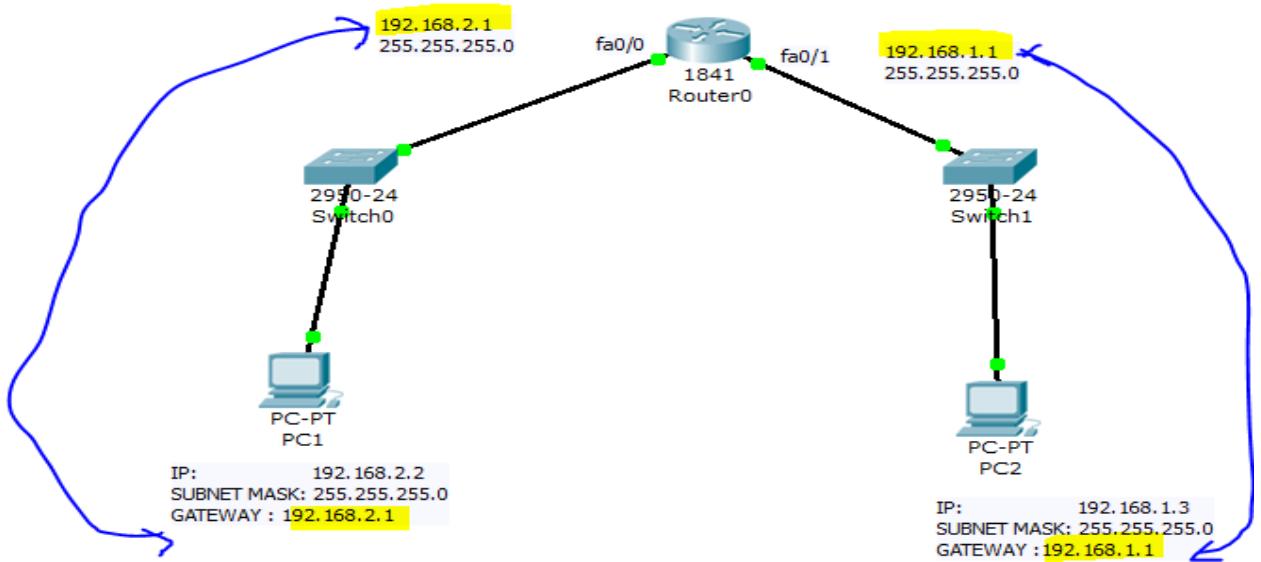
```

Bu durumda iletişimimin gerçekleşmesi için iki PC yi de aynı mantıksal ağa almak gereklidir. Ya da farklı mantıksal ağa olmaları gerekiyorsa, **3. Katman bir cihaz ile** bu iki **FARKLI** mantıksal ağı birleştirmek gerekecektir. Aşağıda bu iki ağın Router üzerinden bağlantısı görülmektedir.



Bu durumda PC1 ile Router'ın fa0/0 portu aynı switche bağlıdır. Aynı mantıksal ağa olmaları gereklidir. Bu sebeple Router fa0/0 portuna verilecek adres de yine 192.168.2.0/24 ağında bir adres olmalıdır. Bu ağdaki herhangi bir IP verilebilir ancak geleneksel olarak ilk IP adresi (**192.168.2.1/24**) ya da son IP adresi (**192.168.2.254/24**) verilir. Örnekte ilk adresler verilmiştir. Yine fa0/0 portu ile PC1 in aynı mantıksal ağa olduklarıını anlayabilmeleri için Subnet Mask bilgisi 255.255.255.0 verilebilir. Yine PC2 ile Router Fa0/1 portu da aynı ağa olmalıdır. Router fa0/1 portuna örnekte 192.168.1.1/24 (ilk) adres verilmiştir. Mantık olarak, router farklı ağları birleştirdiği için fa0/0 IP adresi ile fa0/1 IP adresi de farklı aralıkta olması **zorunludur**.

Bu durumda fa0/0 portu PC1 için (daha doğrusu Switch0'a bağlı olan tüm cihazlar için) fa0/1 portu da PC2 için default gateway olacaktır. O halde PC'lere default gateway bilgisi girilmesi gerekmektedir. Aksi takdirde PC1 ve PC2 ağlarının dışına çıkamayacaklardır.



Bu durumda PC1 PC2 ile iletişim kurabilecektir.

```
PC>ipconfig

IP Address.....: 192.168.2.2
Subnet Mask.....: 255.255.255.0
Default Gateway...: 192.168.2.1

PC>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 192.168.1.3: bytes=32 time=87ms TTL=127
Reply from 192.168.1.3: bytes=32 time=110ms TTL=127
Reply from 192.168.1.3: bytes=32 time=40ms TTL=127
Reply from 192.168.1.3: bytes=32 time=110ms TTL=127

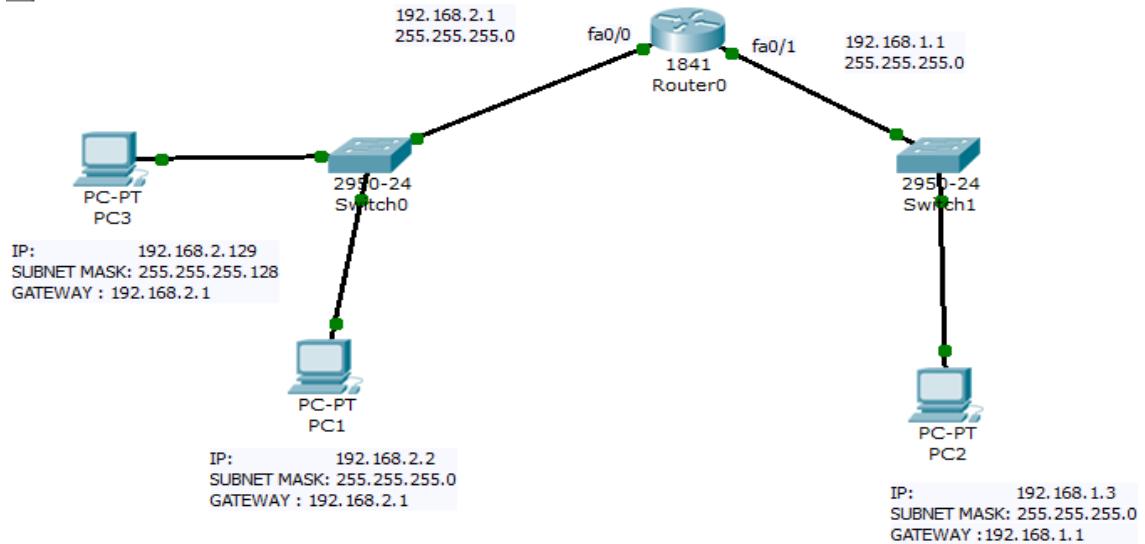
Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 40ms, Maximum = 110ms, Average = 86ms
```

Şimdi Switch0'a bir PC3 bağlayıp aşağıdaki yapılandırmayı yapalım

IP: 192.168.2.129

Subnet Mask: 255.255.255.128

Gateway : 192.168.2.1



Bu durumda PC1 ile PC3 arasında bir iletişim olacağını düşünelim.

PC1'den PC3'e ping başarısız olacaktır. Çünkü;

PC1'in iletişim kuracağı IP adresi 192.168.2.129

PC1 bu hedef IP adresi ile **KENDİ SUBNETMASK** bilgisini and leyecektir.

$192.168.2.129 \wedge 255.255.255.0 = 192.168.2.0$ (**hedef Network Adresi**)

Kendi Network Adresini bulmak için Kendi IP si ile Subnet Mask AND leyecektir.

$192.168.2.2 \wedge 255.255.255.0 = 192.168.2.0$ (**kaynak Network Adresi**)

Hedef Network Adresi ile Kaynak Network Adresi aynı olduğuna PC1; PC3 'ün kendisi ile aynı ağda olduğunu düşünecektir.

Oysa PC3 için AND işlemini yaptığımızda,

PC3 IP : 192.168.2.129 Subnet Mask = 255.255.255.128

PC3 ün iletişim kuracağı IP 192.168.2.2

PC3, Hedef IP (192.168.2.2) ile KENDİ Subnet Mask AND leyecektir.

$192.168.2.2 \wedge 255.255.255.128 = 192.168.2.0$ (Hedef Network Adresi)

Kendi IP adresi ile Subnet Mask AND leyecektir.

$192.168.2.129 \wedge 255.255.255.128 = 192.168.2.128$ (Kaynak Network Adresi)

PC3'e göre PC1 farklı ağdadır. BU sebeple PC3'ten 192.168.2.2 'ye giden her veri Default Gateway'e gönderilecektir.

ANAHTARLAMA

MOD ADI	GÖSTERİM	KOMUT ÖRNEĞİ
User mode	Switch>	-- / disable
Privileged Exec Mode	Switch#	enable
Global Configuration Mode	Switch(config)#	configure terminal
Interface Mode	Switch(config-if)#	interface fa0/1

Switch Modları Arasında Geçişler

Herhangi bir cisco Switch'e console ile bağladığınızda - console parolası belirlenmemişse - direkt olarak User Moduna girilir. User modundan privileged exec moduna geçmek için **enable** komutu kullanılır. Priv. Exec moddan user moda geçiş için **disable** komutu kullanılır. Aşağıda modlar arasındaki geçişlerin hangi komutlar ile yapıldığı gösterilmektedir.

User - Privileged Exec Mod geçişleri:

```
Switch> enable → Switch#
Switch# disable → Switch>
```

Privileged Exec Mod - Global Configuration Mod Geçişleri:

```
Switch#configure terminal → Switch(config)#
Switch(config)# exit → Switch#
```

Global Configuration Mod - Interface Moda Geçişleri:

```
Switch(config)#interface fa0/1 → Switch(config-if)#
Switch(config-if)#exit → Switch(config)#
*** Herhangi bir moddan direkt olarak Privileged Exec moda geçmek için end komutunu veya Ctrl +Z tuş kombinasyonunu kullanınız.
```

```
Switch(config-if)#end → Switch#
```

Temel Switch Yapılandırması

Switch lokal saatini yapılandırmak

Switch lokal saatini yapılandırmak için privileged exec moda aşağıdaki syntax yapısında komut kullanılır.

```
clock set SS:DD:SS AY GÜN YIL
```

```
Switch#clock set 09:22:14 JUNE 14 2011
```

show clock komutu ile sistem saatı görüntülenebilir.

```
Switch#show clock
*9:22:16.150 UTC Sal Haz 14 2011
```

Switch'e bir isim vermek

```
Switch(config)#hostname KAT3_SWITCH
```

```
KAT3_SWITCH(config) #
```

Enable Parolası Tanımlamak

```
Switch(config)#enable password KOLAYSIFRE
```

```
Switch(config)#enable secret ZORSIFRE
```

Parolaları Type-7 ile Criptolama

```
Switch(config)#service password-encryption
```

DUPLEX MOD ve HIZ YAPILANDIRMA

```
S1(config)#interface fastEthernet 0/1
```

```
S1(config-if)#duplex {auto | half | full}
```

```
S1(config-if)#speed {auto | 10 | 100 }
```

SWICTH YÖNETİM ARAYÜZÜNÜ (MANAGEMENT INTERFACE)

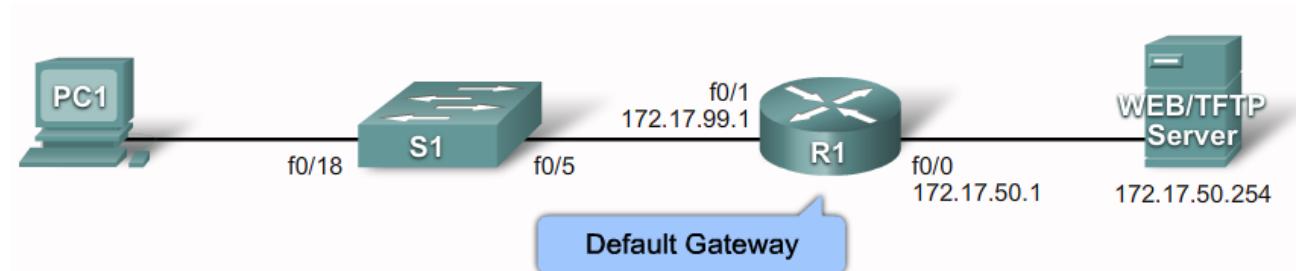
YAPILANDIRMAK

Switchlerdeki portlar varsayılan olarak Layer2 portlardır ve bu sebeple IP verilemezler. Bir switch'i telnet veya ssh ile erişilebilir yapmak için yönetimsel arayüzüne IP adresi verilir. Switch'lerde yönetim arayüzü default olarak Vlan1'dir. Aşağıdaki komutlar ile VLAN1 yapılandırılmalıdır.

```
Switch(config)#interface vlan 1
```

```
Switch(config-if)#ip address 172.17.99.100 255.255.255.0
```

```
Switch(config-if)#no shutdown
```



Böyle bir yapıda, S1'in ağın dışı ile iletişime geçebilmesi için Gateway bilgisini ihtiyacı vardır. Aşağıdaki yapılandırma, bir switch'e gateway bilgisini öğretir.

```
S1(config)#ip default-gateway 172.17.99.1
```

MAC ADRES TABLOSU

Switch'ler MAC adres tablosuna göre anahtarlama yaparlar. MAC adres tablosu, switchten geçen frame'lerin kaynak MAC adres bilgisi – port eşleşmesi ile dynamic öğrenilir ve RAM'da 300 sn tutulur.

```
Switch#show mac-address-table  
Mac Address Table
```

Vlan	Mac Address	Type	Ports
1	0001.c933.a6d2	DYNAMIC	Fa0/3
1	000b.be9d.2e87	DYNAMIC	Fa0/4
1	00e0.8fbc.c628	DYNAMIC	Fa0/1

Örneğin MAC adresi AAA.BBB.CCC olan ve Fa0/20 portuna bağlı olan bir MAC adresi aşağıdaki komut ile bu tabloya static olarak ekleyebiliriz.

```
Switch(config)#mac-address-table static AAA.BBB.CCC vlan 1 interface fa0/20
```

Mac Address Table			
Vlan	Mac Address	Type	Ports
1	0001.c933.a6d2	DYNAMIC	Fa0/3
1	000b.be9d.2e87	DYNAMIC	Fa0/4
1	00e0.8fbc.c628	DYNAMIC	Fa0/1
1	0aaa.0bbb.0ccc	STATIC	Fa0/20

SWITCH PASSWORD RECOVERY

Switch açılırken 15sn içinde Mode düğmesi, System LED'in Turuncu ve ardından Yeşil'e dönünceye kadar basılı tutulur.

```
flash_init
load_helper
```

Parolaları tutan dosyanın adı değiştirilir ve yeniden başlatılır.

```
rename flash:config.text flash:config.text.old
boot
```

Sistem açıldıktan sonra isimleri eski haline getirilip RAM'a aktarılır.

```
rename flash:config.text.old flash:config.text
copy flash:config.text system:running-config
```

Ardından enable secret komutu ile parola değiştirilir ve copy runn start ile kaydedilir.

CONSOLE PAROLASI TANIMLAMA

```
S1(config)#line console 0
S1(config-line)#password cisco
S1(config-line)#login
```

TELNET PAROLASI TANIMLAMA

```
S1(config-line)#line vty 0 15
S1(config-line)#password cisco
S1(config-line)#login
```

PORT SECURITY

Switch portlarında MAC adres tabanlı güvenlik sağlar. Belirlenen porttan, belirlenen sayıda ve/veya belirlenen MAC adresli cihazların iletişim kurmasını sağlar.

STATIC PORT SECURITY:

Static yöntemde, MAC adres bilgisi manual olarak yazılır. Yazılan bu adres MAC tablosunda saklanır ve running dosyasında tutulur. Static MAC adresi eklenmiş bir porta port-security uygulanmaz.

Switch portları default olarak dynamic modadır. Bu modun Access olarak değiştirilmesi gereklidir.

```
Switch(config)#interface fastEthernet 0/5  
Switch(config-if)#switchport mode access
```

Port security enable edilmelidir.

```
Switch(config-if)#switchport port-security
```

MAC adresi 0111.0222.0333 olarak tanımlayalım.

```
Switch(config-if)#switchport port-security mac-address 111.222.333
```

BU porttan maximum 1 adres öğrenilebilir olduğunu belirtelim.

```
Switch(config-if)#switchport port-security maximum 1
```

Kural ihlali durumunda (violation) yapılacak action işlemini portu kapat (**shutdown**) olarak yapılandırıralım.

```
Switch(config-if)#switchport port-security violation shutdown
```

Violation türleri, shutdown | restrict | protect olabilir.

DYNAMIC PORT SECURITY:

Bu yöntemde switch dynamic olarak öğrenir ve switch restart edildiğinde adres unutulur.

```
Switch(config)#interface fastEthernet 0/6  
Switch(config-if)#switchport mode access  
Switch(config-if)#switchport port-security  
Switch(config-if)#switchport port-security maximum 2  
Switch(config-if)#switchport port-security violation restrict
```

STICKY PORT SECURITY:

Bu yöntemde switch dynamic olarak öğrenir ve running-config dosyasına yazılır.

```
Switch(config)#interface fastEthernet 0/6  
Switch(config-if)#switchport mode access  
Switch(config-if)#switchport port-security  
Switch(config-if)#switchport port-security mac-address sticky  
Switch(config-if)#switchport port-security maximum 2  
Switch(config-if)#switchport port-security violation restrict
```

VIOLATION TÜRLERİ

Shutdown: Portu kapatır. Default mod budur. Bu portun tekrar açılabilmesi için **shutdown** ve ardından **no shutdown** komutlarının kullanılması gereklidir.

Protect : Kural ihlali durumunda, yabancı adresin iletişimini kesilir. Port kapanmaz

Restrict : Kural ihlali durumunda yabancı adresin iletişimini kesilir, port kapanmaz. Ancak log sunucusuna bir uyarı gönderilir ve kural ihlali sayacı artar.

Port-Security Doğrulama

```
Switch#show port-security interface fastEthernet 0/1
Port Security          : Enabled
Port Status             : Secure-shutdown
Violation Mode         : Shutdown
Aging Time              : 0 mins
Aging Type              : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses     : 0
Configured MAC Addresses : 0
Sticky MAC Addresses    : 0
Last Source Address:Vlan : 00D0.FF8D.7A9A:1
Security Violation Count : 1
```

```
Switch#show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
                  (Count)      (Count)      (Count)
-----
Fa0/1           1           0           1       Shutdown
Fa0/2           1           1           0       Shutdown
Fa0/3           1           0           0       Shutdown
```

```
Switch#show port-security address
      Secure Mac Address Table
-----
Vlan  Mac Address Type          Ports      Remaining Age
                  (mins)
-----
1     0060.4761.9632  DynamicConfigured FastEthernet0/2      -
-----
Total Addresses in System (excluding one mac per port) : 0
Max Addresses limit in System (excluding one mac per port) : 1024
```

*** Birden çok port üzerinde port-security uygulamak için tek tek arayüz yapılandırma yerine aşağıdaki gibi belli bir aralıkta bu yapılandırma uygulanabilir.

Örnek: Fa0/1 ve Fa0/5 portlarında port-security uygulayalım

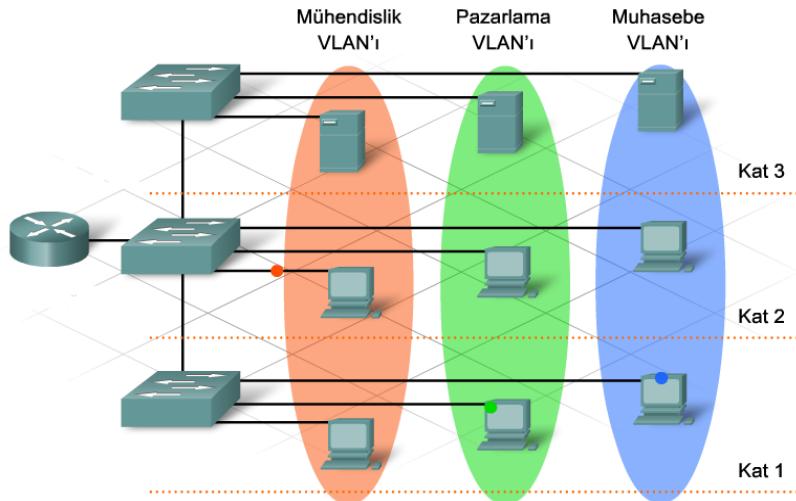
```
Switch(config)#interface range fastEthernet 0/1 , fa0/5
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport port-security
Switch(config-if-range)#switchport port-security mac-address sticky
Switch(config-if-range)#switchport port-security violation restrict
```

Örnek: Fa0/6 ile Fa0/24 aralığındaki tüm portlarda port-security uygulayalım

```
Switch(config)#interface range fastEthernet 0/1 - 24
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport port-security
Switch(config-if-range)#switchport port-security mac-address sticky
Switch(config-if-range)#switchport port-security violation shutdown
```

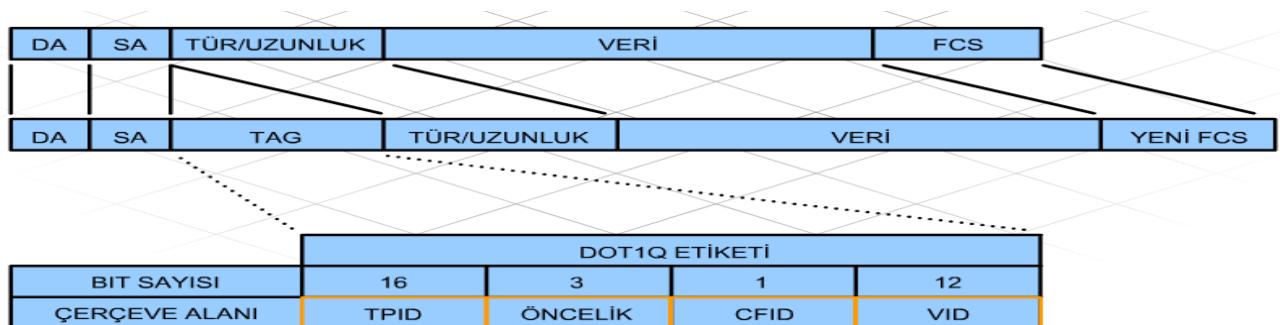
VLAN (Virtual LAN) YAPILANDIRMA

Bir VLAN, çoklu fiziksel LAN kesimlerini kapsayabilen mantıksal broadcast alanlarıdır. Mantıksal işleve, proje ekiplerine veya uygulamalara göre, kullanıcıların fiziksel konumundan bağımsız olarak gruplandırmasını sağlar. Broadcast mesajlar vlan içinde kalır. Farklı VLAN'lar mantıksal olarak farklı ağlar olduğundan L3 bir cihaz olmadan haberleşemezler.



Switch'te vlan oluşturulduktan sonra portlar oluşturulan bu VLAN'lere üye yapılır. Switch üzerinde default olarak VLAN1 bulunur ve her port bu vlan üyesidir. Bu port aynı zamanda yönetim VLAN'ıdır.

VLAN FRAME YAPISI



VLAN uygulamalarında Normal Ethernet çerçevesine 4 Byte bir bilgi daha eklenir(tagging). VID alanı, 12 bittir. Bu da 4096 VLAN tanımlanabilir anlamına gelir.

ADIM 1: VLAN OLUŞTURMA

```
Switch(config)#vlan vlan_numarası
Switch(config-vlan)#name vlan_adi
Switch(config-vlan)#exit
```

Örnek: Numaraları 5 ve 10 olan iki vlan oluşturup, sırasıyla MUHASEBE ve PAZARLAMA olarak isimlendirilelim. SW1 üzerinde ilk 10 portu MUHASEBE, sonraki 10 portu da PAZARLAMA vlan'a dahil edelim.

```
SW1(config)#vlan 5
SW1(config-vlan)#name MUHASEBE
SW1(config-vlan)#exit
```

```
SW1(config)#vlan 10
SW1(config-vlan)#name PAZARLAMA
SW1(config-vlan)#exit
```

ADIM 2: PORTLARI VLAN ÜYESİ YAPMA

```
SW1(config)#interface range fa0/1-10
SW1(config-if-range)#switchport access vlan 5

SW1(config)#interface range fa0/11-20
SW1(config-if-range)#switchport access vlan 10
```

Şimdi yapılandırmanın doğru olup olmadığını kontrol edelim.

```
SW1# show vlan

VLAN Name Status Ports
--- -----
1 default active Fa0/21, Fa0/22, Fa0/23, Fa0/24
                  Gig1/1, Gig1/2
5 MUHASEBE active Fa0/1, Fa0/2, Fa0/3, Fa0/4
                  Fa0/5, Fa0/6, Fa0/7, Fa0/8
                  Fa0/9, Fa0/10
10 PAZARLAMA active Fa0/11, Fa0/12, Fa0/13, Fa0/14
                  Fa0/15, Fa0/16, Fa0/17, Fa0/18
                  Fa0/19, Fa0/20
```

Bir port bir anda sadece bir VLAN üyesi olabilir. Fa0/1-10 arasındaki portları VLAN5 üyesi; Fa0/11-20 arasını da VLAN10 üyesi yaptık. Diğer portlar; Fa0/21-24 arasındaki portlar ve diğerleri default vlan'a (VLAN1) ait olacaktır.

ACCESS ve TRUNK PORT KAVRAMLARI

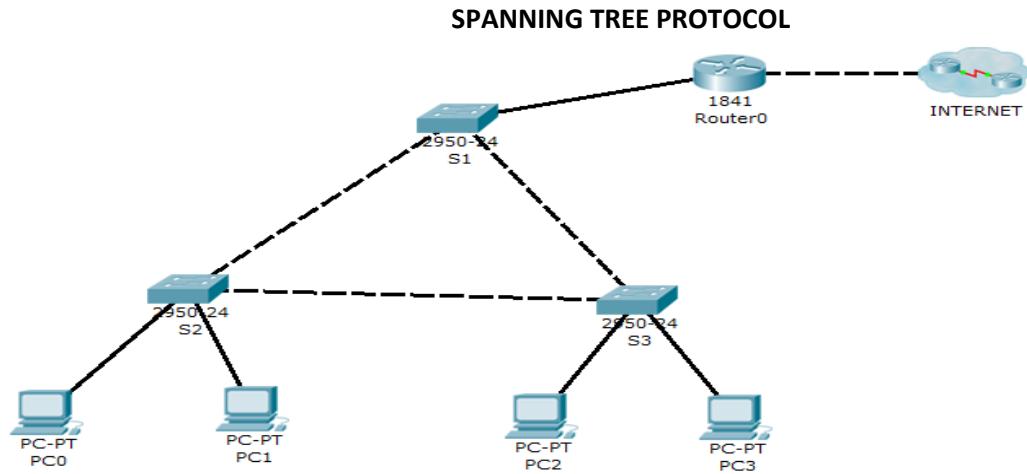
Herhangi bir VLAN üyesi olan porta **ACCESS** port denir. Farklı Vlan'a üye Access portlar arasında direk olarak bir iletişim sağlanamaz. Bazı portlardan tüm VLAN trafiğinin (ya da birden çok) geçmesi gereklidir. Bu tür portlara da **TRUNK** port denir. Örneğin Switch'ler arasındaki bağlantılar ya da Switch Router arasındaki bağlantılar Trunk bağlantı olabilir. Trunk portlar için **IEEE 802.1Q** adından bir Standard geliştirilmiştir.

SW1 üzerindeki fa0/1 – fa0/10 arasındaki portları access, Fa0/24 portunu da trunk olarak yapılandırıralım.

```
SW1(config)#interface range fastEthernet 0/1-10
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#exit
SW1(config)#interface fastEthernet 0/24
SW1(config-if)#switchport mode trunk
```

Bazı Switchlerde trunk türü belirtilmelidir. 2960 gibi modellerde bu gereksizdir çünkü sadece **dot1q** destekler.

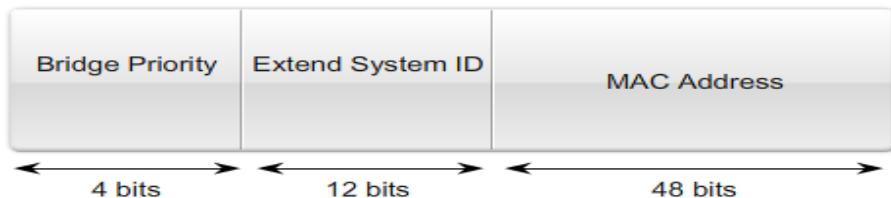
```
SW1(config-if)#switchport trunk encapsulation {dot1q | isl | negotiate}
```



Yedekli olarak Switchler arasında yukarıdaki gibi bağlantılar oluşturulduğunda döngüsel yapıdan dolayı **Broadcast Storm**, **Duplicate Unicast Frame** ve **MAC Address Table tutarsızlığı** gibi sorunlar oluşabilir. STP (Spanning Tree Protocol), STA (Spanning Tree Algorithm) ile böyle bir durumda döngü oluşturan portlardan birinin(*) bloklanmasını sağlar. Döngüsel durum ortadan kalktığında ise bu portun tekrar aktif olarak çalışması sağlanır.

STP, Switchlerden birini referans olarak tespit eder ve bu doğrultuda hangi Switch'in hangi portunun bloklanması gerektiğini belirler. Bu seçilen referans switche **ROOT BRIDGE** denir. Root Bridge olan switchin hiçbir portu blokmaz.

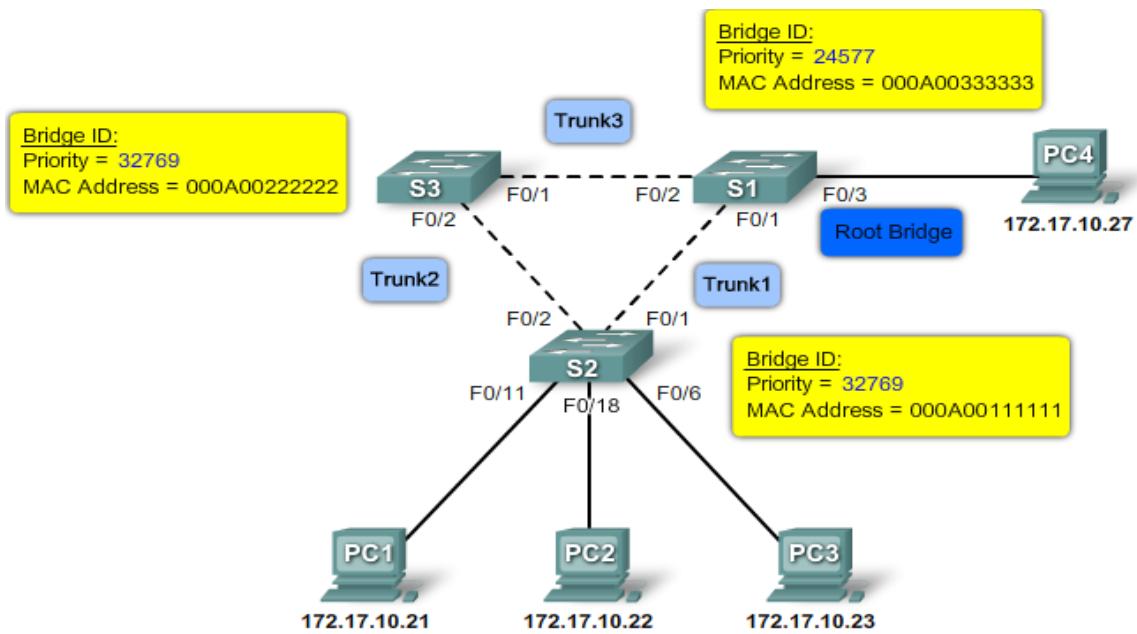
STP çalıştırın Her switch Bridge ID (BID) denen 64-bitlik bir değere sahiptir. Bu değer Priority ve MAC adres değerlerinden oluşmaktadır.



BID değerlerinin karşılaştırılması sonucu, **en düşük** BID değerine sahip olan switch ortamda **ROOT BRIDGE** olarak seçilir. Default olarak switchler 32768 Priority değerine sahiptir. Burada Extend System ID, VLAN numarası olarak düşünülebilir. Örneğin VLAN 1 için Priority değeri;

$32768 + 1 = 32769$ olacaktır. Varsayılan olarak tüm switchlerde bu değer eşit olacağından Root Bridge seçiminde MAC adres etkin olacaktır. Dolayısıyla varsayılan değerler göz önüne alındığında en düşük MAC adres sahip olan switch Root Bridge olacaktır.

Örneğin aşağıdaki örnekte, priority değerlerine bakılarak S1' in ROOT BRIDGE seçildiği görülür.



Örnekte priority değerleri eşit olsaydı bu kez MAC adrese bakılacaktı ve en düşük MAC adresli switch olan S2, root bridge seçilecekti. (**Soru, hangi portun MAC adresi???**)

PRIORITY DEĞERİNİ DEĞİŞTİRME

Priority değeri değiştirilerek istenilen switchin root bridge olması sağlanabilir.

S(config)# spanning-tree vlan 1 priority 4096

veya

S(config)# spanning-tree vlan 1 root primary

BPDU (Bridge Protocol Data Unit)

STP'ye dahil olan switchler her iki saniyede **BPDU** (Bridge Protocol Data Unit) denen hedef adresi **01:80:C2:00:00:00** olan multicast çerçeveler yayırlarlar. (**Soru kaynak MAC??**) Bu BPDU mesajları ile kimin Root Bridge olacağı belirlenir.

Field #	Bytes	Field
4	2	Protocol ID
	1	Version
	1	Message type
	1	Flags
8	8	Root ID
	4	Cost of path
	8	Bridge ID
	2	Port ID
12	2	Message age
	2	Max age
	2	Hello time
	2	Forward delay

Protocol ID : 2 Byte. Her zaman 0.

Version: 1 Byte, her zaman 0.

Message Type: BPDU mesaj türü.,

Flags : TC (Topology Change) ve TCA (Topology Change Ack.) içerir.

Root ID : 8 Byte. Root Bridge olan cihazın BID değeridir. Başlangıçta bu değer her switch için kendi BID değeridir.

Cost of Path: Root'a giden yolun maliyet değeridir.

Bridge ID : Switchin kendi BID değeri.

Port ID : STP ye dahil portun değeridir. Fa0/1=0x8001...

Message Age : Root'dan gelen configurasyon mesajından itibaren geçen süre.

Max Age : Geçerli yapılandırmanın ne zaman silinmesi gerektiğini belirten süredir. Message Age değeri, Max Age değerine ulaştığında, Root'a erişilebilirlik kaybolduğu farzedilir ve seçim yeniden başlar. 6 ile 40 saniye arası değişebilir. Default = 20 sn.

Hello Time :BPDU mesajları gönderme periyodu . 1- 10 sn arası değişir, default 2 sn.

Forward Delay : Topolji değişikliği olduktan sonra, bir sonraki duruma (**) geçmek için ne kadar süre bekleyeceğini belirten değerdir. Default 15 sn dir. (4 ile 30 sn arası yapılandırılabilir)

POR COST DEĞERLERİ

Link Speed	Cost (Revised IEEE Specification)
10 Gb/s	2
1 Gb/s	4
100 Mb/s	19
10 Mb/s	100

Port cost değerleri aşağıdaki gibi değiştirilebilir.

S(config)#interface fa 0/1

S(config – if)# spanning-tree cost 25 (1 ile 200 000 000 arasında değer verilebilir)

YAPILANDIRMA DOĞRULAMA

Switch#show spanning-tree

VLAN0001

Spanning tree enabled protocol ieee

Root ID Priority 32769

Address 0001.C702.3E60

Cost 19

Port 1(FastEthernet0/1)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

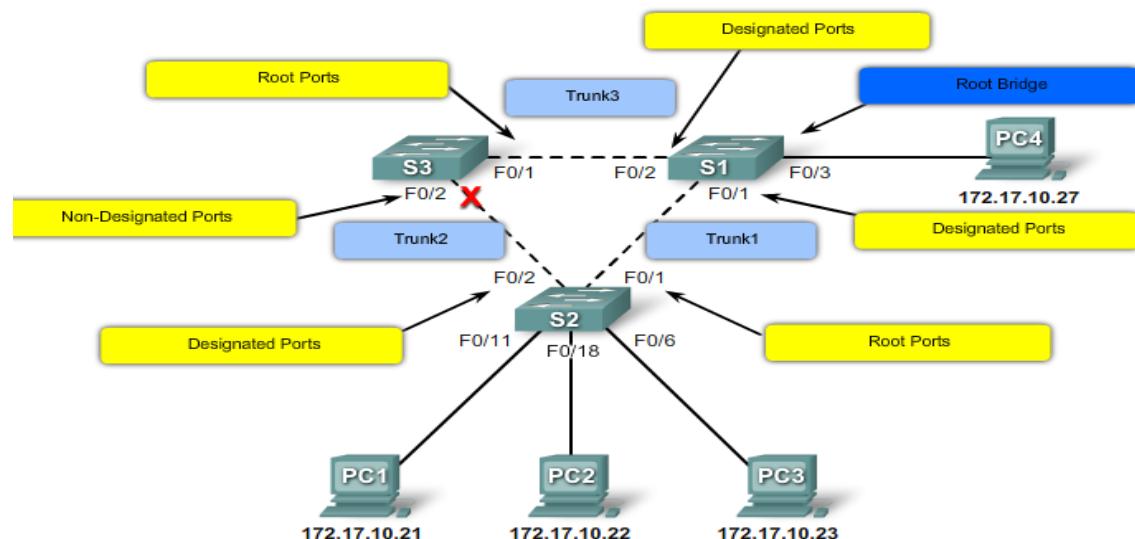
Bridge ID Priority 32769 (priority 32768 sys-id-ext 1)

Address 0007.EC92.2774

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

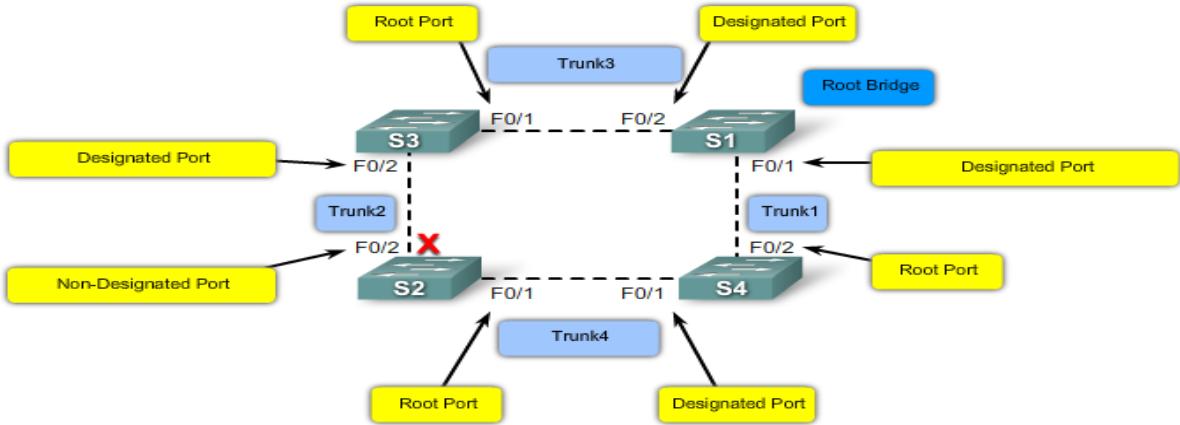
Aging Time 20

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/2	Altn	BLK	19	128.2	P2p
Fa0/1	Root	FWD	19	128.1	P2p

STP PORT ROLLERİ**PORT ROLES**

BPDU mesajları sonunda Root Bridge belirlenir ve diğer switchler bu roota ulaşmak için yolları bulur. Yıldızdaki örnekte S2 için bu switche giden iki yol vardır. Bu yollardan en düşük maliyete (path cost) sahip olan port **ROOT PORT** olarak belirlenir. Bloklanmış (**NON-DESIGNATED**) ve root portları dışındaki tüm portlar **DESIGNATED PORT** olarak adlandırılır. ROOT BRIDGE'in tüm portları **DESIGNATED PORT** olarak belirlenir.

** Yukarıdaki örnekte, S1 root olarak seçilir. S2 ve S3'ten en yüksek BID değerine sahip olan switch'in portu bloklanır.



Bu örnekte, BPDU değişimleri sonucunda BID değeri en düşük olan Switch (S1) root olarak belirlenir. Root'un göndereceği BPDU mesajları sonunda S3 ve S4, gelen BPDU daki ROOT ID değeri ile, kendi BID değerini karşılaşacaktır ve S1'in root olduğunu kabul edecek, path cost olara da 19 (Fa) yazacaklardır.

S4, Fa0/2 portunu root; Fa0/1 portunu designated olarak işaretleyecektir.

S3, Fa0/1 portunu root; Fa0/2 portunu da designated olarak işaretleyecektir.

S2 ise, roota ulaşmak için iki eşit maliyete sahip yolu vardır. Bu yollardan birini root port yapacak, diğerini ise bloklayacaktır. Bu durumda port priority değeri (default 128) en düşük olan port ROOT port seçilecek, diğer ise bloklanacaktır. Port priority değerleri eşit ise bu durumda Port ID değerine bakacaktır. Port ID değeri düşük olan ROOT PORT olacaktır.

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/2	Altn	BLK	19	128.2	P2p
Fa0/1	Root	FWD	19	128.1	P2p

Buradaki "128.1" için: 128= Priority, 1 ise port numarasıdır. Fa0/1 için 1, Fa0/4 için 4...vs.

Port Priority değeri değiştirilebilir. Default 128.

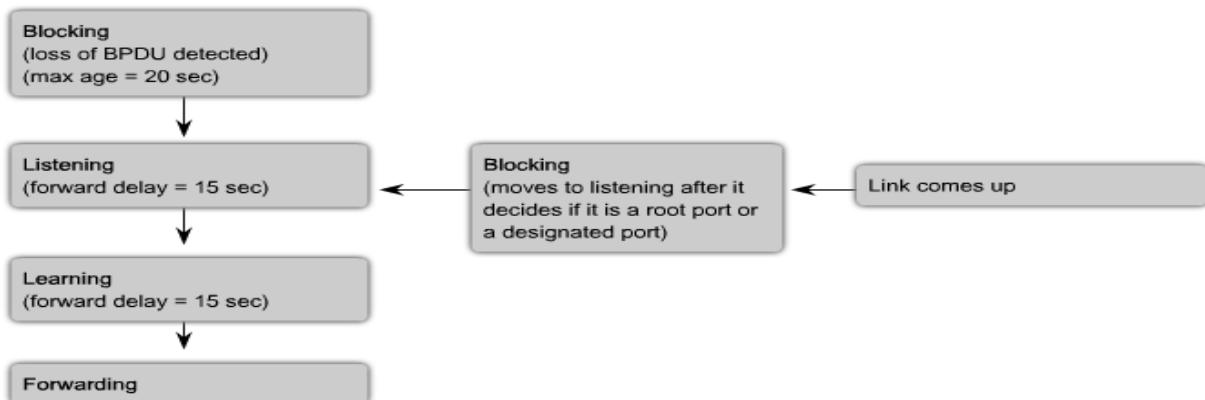
```
Switch(config)#interface fastEthernet 0/1
Switch(config-if)#spanning-tree vlan 1 port-priority ?
<0-240> port priority in increments of 16
```

```
Switch(config-if)#spanning-tree vlan 1 port-priority 16
```

PART DURUMLARI

STP de portlar; **Blocking, Listening, Learning, Forwarding ve Disable** olmak üzere 5 farklı durumda bulunabilir. Aslında Disable, STP içinde değil, Administratively Shutdown modudur.

Processes	Blocking	Listening	Learning	Forwarding	Disable
Receives and process BPDUs	YES	YES	YES	YES	NO
Forward data frames received on interface	NO	NO	NO	YES	NO
Forward data frames switched from another interface	NO	NO	NO	YES	NO
Learn MAC addresses	NO	NO	YES	YES	NO



BPDU GUARD

Portfast olan portlardan BPDU gelmesi durumunda portun error-disabled durumuna geçmesini sağlar. Global ya da interface modda yapılandırılabilir.

Global Modda yapılandırma (Tüm portlara etki eder)

```
ALS2 (config) #spanning-tree portfast bpduguard default
```

Interface Modda yapılandırma:

```
ALS2 (config-if) #spanning-tree bpduguard enable
```

1- BPDU FILTERING

Portlardan BPDU gelmesi durumuna karşı geliştirilen diğer bir yöntemdir. Yine Global ya da interface modda yazılabilir.

Global Modda yapılandırma :

```
ALS2 (config) #spanning-tree portfast bpdufilter default
```

Bu durumda, tüm arayüzlere etki eder. Ancak bu portlardan bir BPDU gelirse port portfast özelliğini yitirir.

Interface Modda yapılandırma:

```
ALS2 (config-if) #spanning-tree portfast bpdufilter enable
```

Bu durumda ise porttan BPDU gönderilmesi ya da alınması engellenir.

2- ROOT GUARD

Yanlış ya da saldırgan olan switchin Root Bridge olmasını önler. Porttan daha yüksek priority değerlikli bir BPDU gelirse bu port root-inconsistent moda düşer. Root olma talebi artık o porttan gelmezse port otomatik olarak normal moduna geçer.

```
ALS1(config)#int ra fa0/1-24
ALS1(config-if-range)#spanning-tree guard root
ALS1(config-if-range)#

ALS1(config)#
%LINK-5-CHANGED: Interface FastEthernet0/18, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/18, changed
state to up
%SPANTREE-2-ROOTGUARDBLOCK: Port 0/18 tried to become non-designated in VLAN
1.

Moved to root-inconsistent state

ALS#show spanning-tree inconsistentports
Name Interface Inconsistency
-----
VLAN0001 FastEthernet0/18 Root Inconsistent

Number of inconsistent ports (segments) in the system : 1
```

UNIDIRECTIONAL LINK DETECTION (UDLD)

Normalde bir switch bir portun fiziksel olarak kırık olduğunu anlayabilmesi için **Layer 1** keepalive mesajları (**link beat**) kullanır. Bazen port bu **keepalive** mesajlarını iletcektir. Ancak data mesajları her iki yönde geçemeyecek duruma gelebilir. Buna Unidirectional Link denir. UDLD yapılandırıldığında, periyodik **hello** mesajları ve bu mesajlara gelen onay mesajları sayesinde haberleşme sağlayarak **Layer 2** iletişim kontrolü sağlanır.

UDLD'nin **aggressive** ve **normal** olmak üzere iki modu vardır. Normal modda karşılık bir hello gelmeyince port undetermined moduna düşer. Ancak aggressive modda port error-disabled modda düşer.

Tüm Fiber Portlarda etkinleştirmek için:

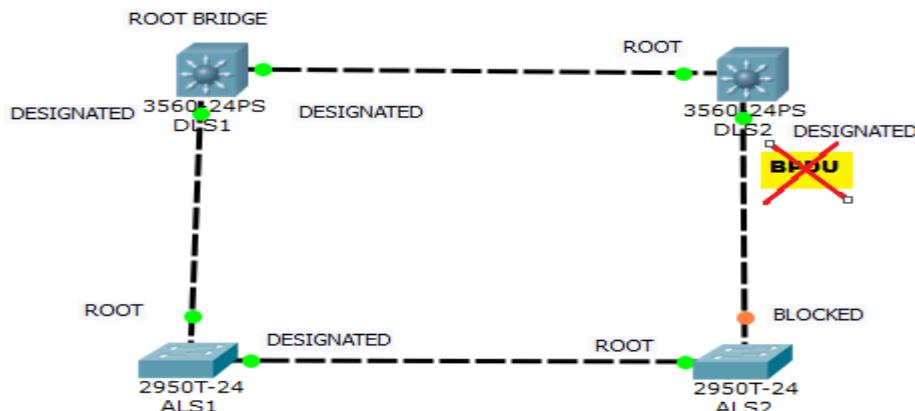
```
SW(config)# udld [enable | aggressive]
```

Belirli bir portta yapılandırmak için:

```
SW(config-if)# udld port {aggressive | disable}
```

3- LOOP GUARD

Bloklanmış bir port yanlışlıkla (Ör. Unidirectional Link hatası veya hatalı konfigürasyon nedeniyle)forwarding duruma geçerse topolojide döngü oluşabilir. Aşağıdaki örnekte olduğu gibi, ALS2 Switch'deki Blocked porttan Unidirectional Link hatası nedeniyle belli bir süre BPDU gelmezse, switch burada döngü olmadığını farzeder ve portu forwarding moduna alır. Bu durumda döngü oluşur. LoopGuard, bu tür hataların önüne geçmek için kullanılır.



Bu durumda bir porttan belli bir süre BPDU gelmezse bu port loop inconsistent moda düşürülür. Root ya da designated olması muhtemel portlarda kullanılır.

Global Modda:

```
SW(config)# spanning-tree loopguard default
```

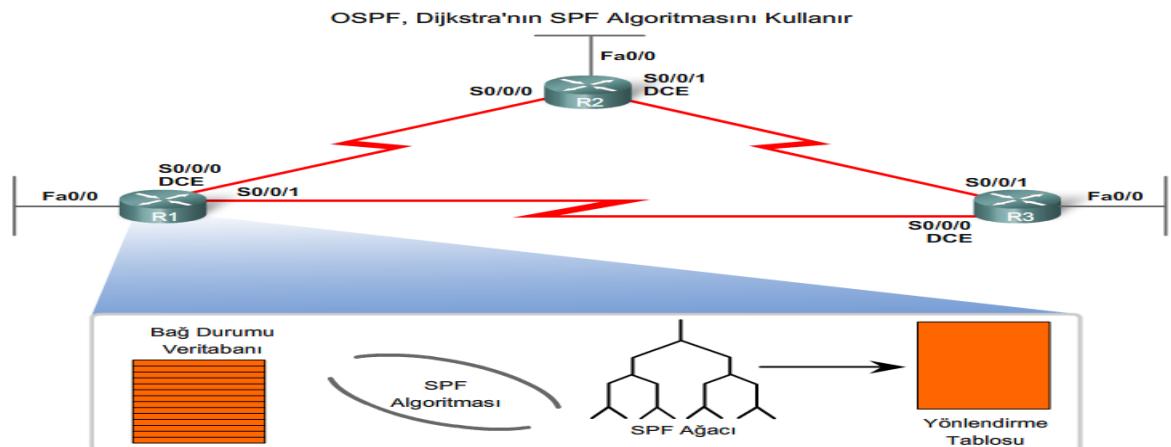
Interface Modda:

```
SW(config-if)# spanning-tree guard loop
```

OSPF (OPEN SHORTEST PATH FIRST)

- IETF tarafından 1987 yılında geliştirildi. 1998 yılında OSPFv2, 1999 yılında IPv6 destekli OSPFv3 geliştirildi.
- **Link-State, classless** bir routing protokoldür.
- Administrative Distance değeri **110** dur.
- Her 30 dk da bir tam güncelleme yapar.
- Network Convergence olduktan sonra, periyodik update yapmaz, bunun yerine değişiklige bağlı güncelleme yapar.(**Triggered Update**)
- **Adjacency**, OSPF routerlar arasında veritabanlarının senkronize olduğu gelişmiş bir komşuluk türüdür. (Full Adjacency)
- **Dijikstra Algoritmasını** (SPF Algoritması) kullanır
- Frame Yapısında MAC adres olarak : 01-00-5E-00-00-05 ve 01-00-5E-00-00-06 kullanır
- Hedef IP olarak: **224.0.0.5** ve **224.0.0.6** kullanır.
- IP paket başlığında bulunun **protocol** field alanı **89** OSPF kullanıldığını gösterir.
- **Authentication** ve **encryption** desteği vardır.
- Metric olarak bandwidth değerini kullanır. (**cost = 10^8 / bw (kbps)**)
- **VLSM** ve **CIDR** desteği vardır.

LINK – STATE Routing Protokollerin çalışma prensipleri



- 1- Her router kendine direk bağlı (directly connected) networkleri öğrenir.
- 2- Her router kendine direk bağlı komşu routelara HELLO paketi gönderir. Böylece komşuluklar keşfedilir.
- 3- Her router kendine direk bağlı networklerin durumlarını gösteren LSP (Link –State Packet) paketleri oluşturur.
- 4- Her router LSP lerini komşularına flood eder ve buna ilişkin bir database oluşturur. (LSDB)
- 5- SPF algoritması database bilgileri ile topolojiyi belirler ve en iyi yolu tespit eder.

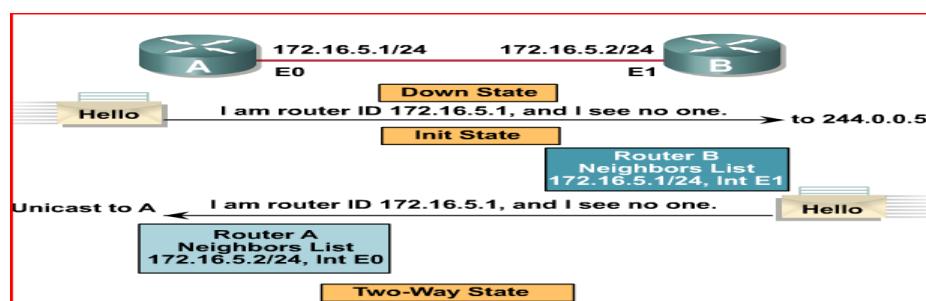
Adjacency (Komşuluk) Oluşumu:

1. Hello paketleri, iki komşu routerin komşuluk oluşturabilmesi için Router ID, Area ID, Authentication Setting, Timer Setting, Router Priority ve DR / BDR bilgilerini içerir.
2. Bu bilgilerin değişimi tamamlanınca komşuluk kurulur.
3. Hello paketleri ile komşuluk kurulduktan sonra, iki router Link-State Database (LSDB) lerini senkronize olması için LSA paketlerini kullanırlar. Bu durumda iki router FULL ADJACENCY konumuna gelir.

Yönlendirici, komşusuyla bitişiklik (adjacency) kurarken çeşitli durum değişiklikleri meydana gelir.

Init	Exstart	Loading
2-Way	Exchange	Full

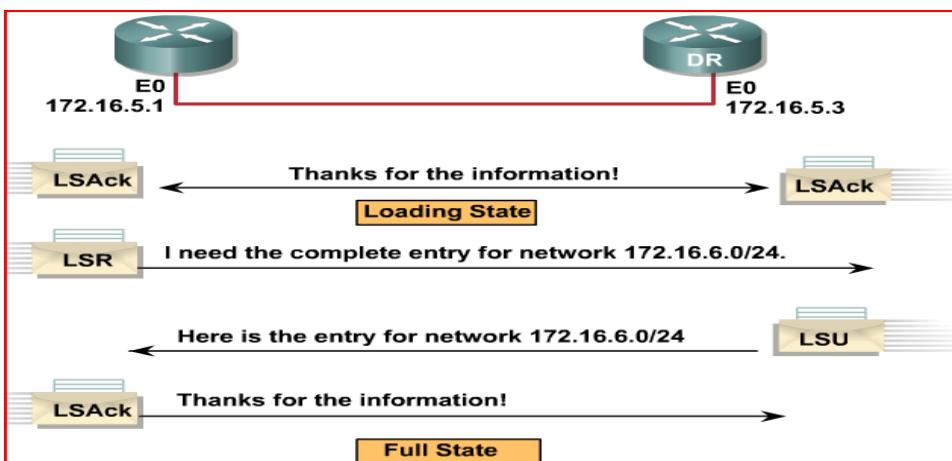
DOWN – INIT ve TWO-WAY DURUMLARI:



EXSTART ve EXCHANGE DURUMLARI :



LOADING ve FULL DURUMLARI :



ÖZETLE OSPF KOMŞULUK OLUŞUM BASAMAKLARI

Durum	Tanım
Init	Yönlendirici, komşusundan ilk merhaba paketini almıştır. Yönlendirici, komşusundan merhaba paketini alınca, gönderen yönlendirici kimliğini, kendi merhaba paketinde onay olarak listeler.
2-Way	İki yönlü iletişim, birbirlerinin merhaba paketlerini gören yönlendiriciler arasında kurulur. Bu duruma, merhaba paketini alan yönlendirici, merhaba paketinin komşu alanında kendi yönlendirici kimliğini görmesiyle ulaşılır. Yönlendirici, komşusuyla tam bitişiklik kurmaya bu durumda karar verir.
Exstart	Yönlendiriciler bir master-slave ilişkisi kurar ve bitişiklik düzeni için ilk sıra numarasını seçer. İki yönlendirici arasında, en yüksek yönlendirici kimliğine sahip olan master olur ve takası başlatır.
Exchange	OSPF yönlendiricileri, sadece bağı durumu tanıtımı (LSA) başlıklarını içeren takas veritabanı tanımlayıcısı (DBD) paketlerini takas eder. DBD, bağı durumu veritabanının tamamının içeriğini tanımlar. Her DBD paketi, sadece master'i tarafından artırlılabilecek bir sıra numarasına sahiptir.
Loading	DBD'ler tarafından sağlanan bilgiye bağlı olarak, yönlendiriciler daha fazla bilgi için bağı durumu istem paketleri gönderir. Komşu, istenen bağı durumu bilgilerini, bağı durumu güncelleme paketlerine ekleyerek sağlar.
Full	Yönlendirici ve ağ LSA'larının tamamı takas edilir ve yönlendirici veritabanları senkronize olur.

OSPF PACKET TÜRLERİ

Type	Packet Name	Description
1	Hello	Discovers neighbors and builds adjacencies between them
2	Database Description (DBD)	Checks for database synchronization between routers
3	Link-State Request (LSR)	Requests specific link-state records from router to router
4	Link-State Update (LSU)	Sends specifically requested link-state records
5	Link-State Acknowledgement (LSAck)	Acknowledges the other packet types

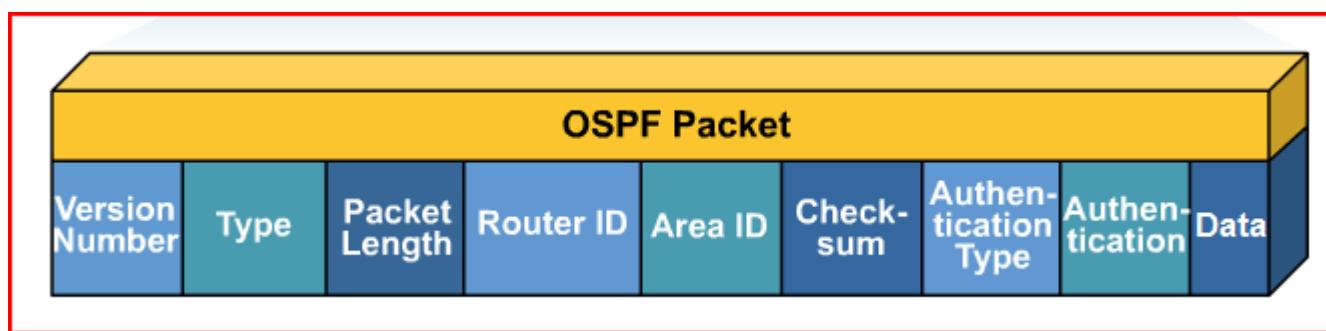
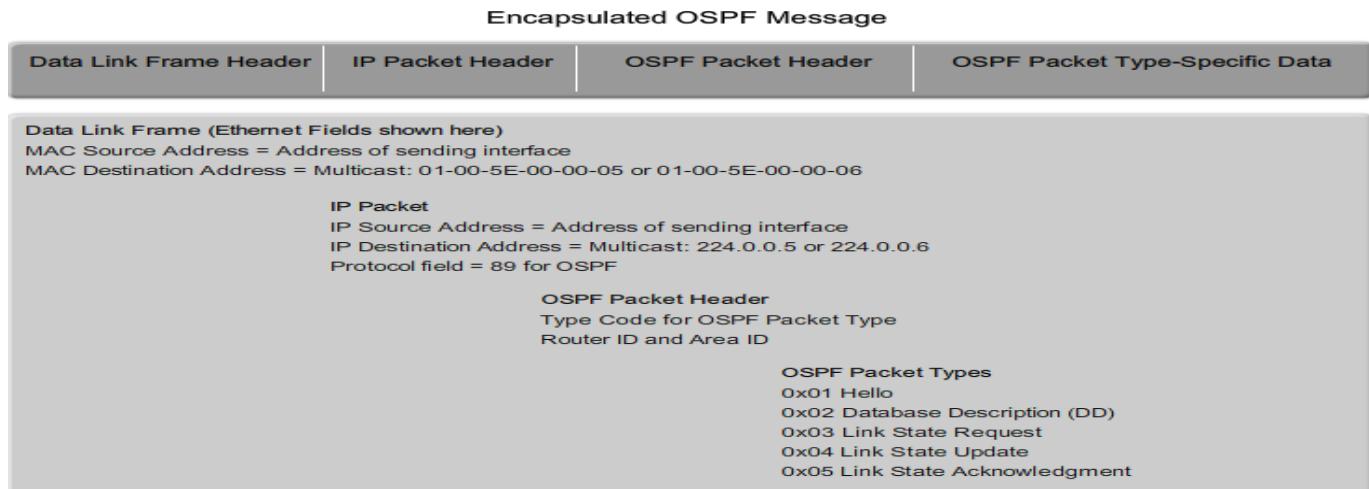
0x01: Hello - OSPF routerlar arasında bitişikliğin (adjacency) oluşturulması ve devam ettirilmesini sağlayan paketleridir.

0x02. DBD – Routerlar arasında database senkronizasyonunu sağlar. Bu paketler gönderen routerın özet link-state bilgilerini içerir. Alıcı router bu paketler ile kendi link-state database bilgilerini karşılaştırır.

0x03. LSR – Alıcı routerin, DBD hakkında detaylı bilgi istediği paket türleridir.

0x04. LSU – LSR paketlerine verilen cevaplarıdır. LSU'lar yedi tür LSA içerir.

0x05. LSAck – LSU ulaştığında alıcı router tarafından gönderilen bir tür onay paketleridir.



Version number: OSPF Versiyon numarası (2 veya 3)

Type: 5 OSPF paket türünden hangisinin olduğu

Packet length: Byte cinsinden paket büyüklüğü

Router ID: Paketi gönderen cihazın ID değeri

Area ID: Paketi gönderenin dahil olduğu area.

Checksum: Paket başlığı için hata kontrol alanı.

Authentication type: Clear- Text ya da MD5 kimlik doğrulama türünü belirler.

Authentication: Kimlik doğrulamada kullanılır.

Data

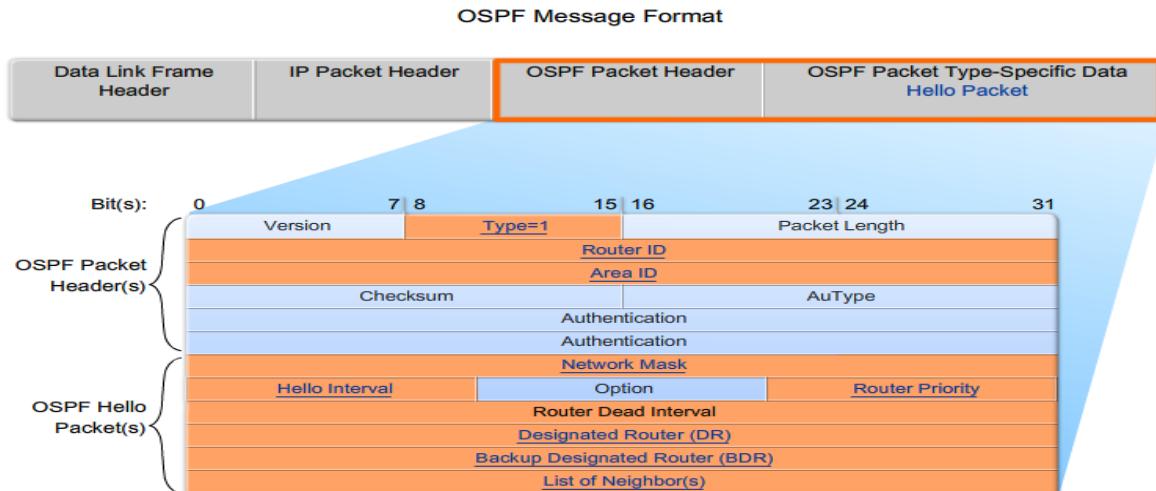
Hello packeti için: Bilinen komşuların listesini içerir.

DBD packeti için: LSDB özet bilgisini içerir. (Bilinen tüm routerların ID değerleri, son sequence numaraları)

LSR packeti için: LSU paketine ihtiyaç duyan router ID ve ihtiyaç duyduğu LSU tipi

LSU packeti için: Full SLA kayıtlarını içerir. Birde çok LSA kaydı bir OSPF paketinde bulunabilir.

LSAck packeti için:boştur.



Router ID: Kaynak Router ID değeri

Area ID: Kaynak Router'in dahil olduğu area.

Network Mask: Kaynak interface subnet mask

Hello Interval: Saniye cinsinden gönderici router'ın hello paketleri periyodu

Router Priority: DR / BDR seçimi için öncelik değeri

Designated Router (DR): DR routerin ID değeri

Backup Designated Router (BDR): BDR routerin ID değeri

List of Neighbors: Komşu OSPF routerların listesi

- Router, Link state bilgilerini yayınılmadan önce herhangi bir OSPF router olup olmadığını anlamak için her 10 (veya 30) saniyede hedef adresi **224.0.0.5** olan bir Hello paketi gönderir. Bu paket Router ID değerini içerir. Ortamda bir OSPF router varsa, bu pakete cevap verir. Gelen cevapta, ilk routerin ID değeri bulunur. Hello paketini gönderen ilk router gelen paketteki kendi ID adresini görünce komşuluk kurulmaya başlanır.
- Hello Interval, Dead Interval ve Network Type değerleri her iki routerda da eşitse routerlar "**adjacency**" durumuna geçerler.
- Hello Interval**, hangi sıklıkta hello paketlerinin gönderileceğidir. Default olarak bu değer point-to-point ağlarda 30 sn, NBMA (non-Broadcast multi-Access)ağlarda ise 10 sn dir.
- Dead Interval**, komşu routerin "down" olduğunu anlamak için geçen süreyi gösterir ve "hello interval" değerinin 4 katıdır. Bu süre içerisinde komşudan "hello paketi" gelmezse komşu routerin down olduğu varsayıılır. Neighbboor listesinden kaldırılır ve diğer routerlara bildirilir.
- Multiaccess networklerde ağda oluşan değişiklikleri diğer routerlara bildirmek için **Designated Router (DR)** ve **Backup Designated Router (BDR)** seçilir.
- Her OSPF router diğer routerlardan gelen LSA lar ile kendi Link State veritabanını oluşturur. Dijkstra algoritması kullanılarak SPF ağaçları oluşturulur ve networklere ulaşabilecek routing tablosunu oluşturur.

- The acronyms LSA and LSU are often used interchangeably.
- An LSU contains one or more LSAs.
- LSAs contain route information for destination networks.
- LSA specifics are discussed in CCNP.

LSA Type	Description
1	Router LSAs
2	Network LSAs
3 or 4	Summary LSAs
5	Autonomous System External LSAs
6	Multicast OSPF LSAs
7	Defined for Not-So-Stubby Areas
8	External Attributes LSA for Border Gateway Protocol (BGP)
9, 10, 11	Opaque LSAs

CONFIGURATION:

R(config)# router ospf <1-65535> // process ID

R(config – router)# network 192.168.1.0 0.0.0.255 area 0 //area ID

Process-ID değeri EIGRP den farklı olarak “adjacency” oluşturmak için aynı olmak zorunda değildir.

Area-ID link state bilgilerini paylaşacak olan routerlerin oluşturduğu alanı gösterir.

ROUTER – ID seçimi,

router – id komutuyla ID belirlenir.

R(config – router)# router-id 10.1.1.1

Eğer router ID belirlenmemişse, loopback adreslerine atanmış en büyük IP adresi router-id olur.

Loopback adresi verilmemişse, herhangi bir aktif interfacedeki en büyük IP adresi, o routerin Router-id değeri olur.

Bir yönlendiricinin ya da arayüz önceliğinin ID'sini değiştirdikten sonra, komşu bitişikliklerini sıfırlayın. Bunu yapmak için, clear ip ospf process komutunu kullanırsınız. Bu komut, yeni değerlerin kullanılacağından emin olmanızı sağlar.

R# clear ip ospf process

* Aynı ID değerine sahip iki router varsa IOS “...duplicate router-id..” hmasını verir.

Up – Down olarak sürekli değişen bir arayüz (*flapping link*) SPF algoritmasının her defasında çalışmasını gerektirir. Bu durumu önlemek için, router LSU aldıktan 5 sn sonra SPF hesaplar. (**SPF Schedule Delay**). Ayrıca SPF hesaplandıktan sonra ilk 10 saniye içinde herhangi bir SPF hesaplaması tekrar yapılmaz.

OSPF Metrik olarak **10^8 / bw** (kbps) değerini kullanır.

Cisco routerlarda seri arayuzlerin varsayılan bandwidth değeri, T1 değerine eşittir (1,544 Mbps) OSPF'in doğru bir biçimde cost hesabı yapabilmesi için doğru bw değeri girilmelidir.

R(config – if)# bandwidth 64 //kbps

Bu durumda cost, $10^8 / 64 = 1562$ olacaktır.

Veya bandwith değeri girmek yerine direk olarak o arayüzün cost değeri yazılabilir.

R(config – if)#ip ospf cost 1562

Multiaccess ağlarda DROthers olan routerlar LSA bilgilerini 224.0.0.6 adresini kullanarak DR ve BDR routerlara iletirler. Bu LSA bilgilerinin diğer routerlara iletilmesinden DR sorumludur. DR, bu yayını 224.0.0.5 multicast adresini kullanarak gerçekleştirir.

Herhangi bir router açıldıktan ve “network” komutu ile tanımlama yapıldıktan sonra DR seçimi başlar. DR seçildikten sonra aşağıdaki durumlar haricinde DR değişmez.

- DR devre dışı kalırsa
- DR'daki OSPF işlemi devre dışı kalırsa
- DR daki multiaccess interface devre dışı kalırsa

Bu durumda BDR, yeni DR olur yeni BDR seçilir.

Bir router'ı DR seçitmeye zorlamak için priority değeri verilebilir.

R(config – if)# ip ospf priority <0 – 255> komutu ile bu işlem gerçekleştirilir. **Default** olarak priority değeri her routerda **1** dir. Bir routerin priority değerini 0 yapmak o routeri *DROthers* olmaya zorlar.

Yüksek priority değerine sahip router DR seçilir. Priority belirtilmemişse (tüm routerlar için priority = 1 durumu) en yüksek router-id değerine sahip router DR seçilir.

DEFAULT ROUTE BİLGİSİNİ YAYINLAMAK İÇİN,

R(config – router)# default-information originate komutu kullanılır. Bu durumda routing tablosunda (sh ip route çıktısında)

*O*E2 0.0.0.0/0 [110 / 1] via 192.168.1.1, 00:05:34 serial 0/0/0*

gibi bir satır görünür. E2 değeri OSPF External Type 2 olduğunu gösterir.

COST REFERANS BANDWITH DEĞERİNİ DEĞİŞTİRMEK

Cost hesaplanırken 10^8 değeri referans alınır. Gigabit ethernetlerde cost hesabı bu sebepten dolayı yanlış çıkacaktır. Bunun için referans değeri değiştirilmelidir.

R(config – router)# auto – cost reference – bandwith 10000 //Mbps cinsinden değeri

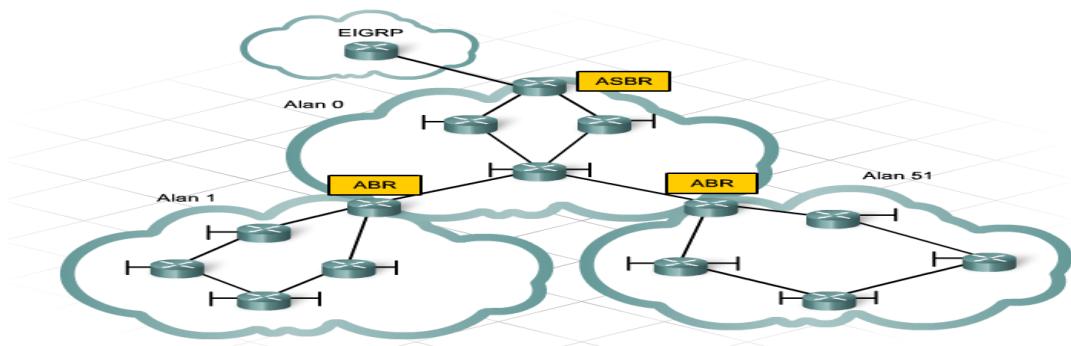
OSPF INTERVAL DEĞERLERİNİ DEĞİŞTİRMEK:

R(config – if)# ip ospf hello – interval 5 // saniye cinsinden

R(config – if)# ip ospf dead – interval 20 // saniye cinsinden

- Bir broadcast ortamında, yönlendirici Full durumuna ancak belirtilen yönlendiriciyle (DR) ve yedek belirtilen yönlendirici (BDR) ile ulaşacaktır. Diğer tüm komşular 2-way durumunda gözükmeli.

- Bir bağ arızalandığında, bağ hakkında bilgi sahibi yönlendirici bilgisi 224.0.0.6 çoklu yayın adresi aracılığıyla DR'ye gönderir. Değişikliğin 224.0.0.5 çoklu yayın adresiyle OSPF yönlendiricilerinin tamamına gönderilmesinden DR sorumludur. Bu işlem, ağ üzerinde gönderilen güncelleme sayısını azaltmanın yanı sıra, yönlendiricilerin tümünün aynı bilgisi, aynı anda tek bir kaynaktan alacağından emin olmanızı sağlar.
- BDR, hataların veri akışında kesintiye sebep olmasını engeller. DR gibi, BDR de 224.0.0.6 adresinden gelen bilgileri toplar ve DR'ye gönderilen tüm güncellemeleri alır. Eğer DR arızalanırsa, DR'nin yerini BDR alır ve yeni bir BDR seçilir. DR ya da BDR olarak seçilmemiş yönlendiriciler DROther ismini alır.
- OSPF ağlarının tamamı Alan 0 (area 0) ile başlar; buna omurga alan (backbone area) da denir. Ağ genişledikçe, Area 0 alanına yakın farklı alanlar oluşturulabilir. Bu diğer alanlar, 65.535'e kadar herhangi bir sayı alabilirler. Bir alanda en fazla 50 yönlendirici kullanılabilir. ***Diger bütün arealar area 0 ile baglantili olmak zorundadir.***



Bir alanı, omurga alanına bağlayan yönlendiriciye, Alan Sınır Yönlendiricisi (Area Border Router -**ABR**) denir. Bir alanı, EIGRP gibi farklı bir yönlendirme protokolüne bağlayan ya da OSPF alanına yönlendirilmiş statik rotaları yeniden dağıtan bir yönlendiriciye ise, Otonom Sistem Sınır Yönlendiricisi (Autonom System Border Router – **ASBR**) denir.

Bir OSPF ASBR yönlendiricisini bu ağları özetleyecek şekilde yapılandırmak için, yönlendirici yapılandırma modunda aşağıdaki komutu kullanılır:

```
R(Config – router)# area 0 range 192.168.0.0 255.255.252.0
```

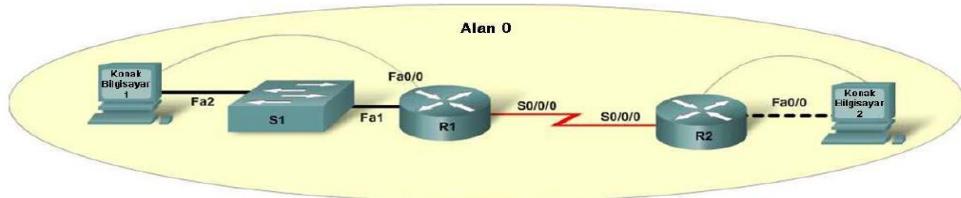
KİMLİK DOĞRULAMA (AUTHENTICATION)

OSPF kimlik doğrulaması (authentication) ayarlayabilirsiniz. Bir bölgede (area) kimlik doğrulaması ayarlandığında, yönlendiriciler sadece kimlik doğrulaması bilgileri eşleştiğinde bilgi paylaşımı gerçekleştirir.

Basit parola doğrulaması için, her yönlendiriciye, anahtar adı verilen bir parola ayarlarsınız. Bu yöntem, sadece temel bir güvenlik düzeyi sağlar; yönlendiriciler arasında kullanılan parola düz metin biçimindedir. Parolayı görmek, düz metin olduğu için aynı derecede kolaydır.

Kimlik doğrulamanın daha güvenli bir yöntemi ise, İleti Özeti 5'tir (MD5). Her yönlendiricide bir anahtara ve anahtar ID'sine gerek duyar. Yönlendirici, OSPF paketi adı verilen, anahtarı işleyen bir algoritma ve anahtar ID'sini kullanarak şifrelenmiş bir sayı oluşturur. Her OSPF paketi bu şifrelenmiş

sayısı içerir. Packet sniffer yazılımları bu anahtarı elde etmek için kullanılamaz; çünkü anahtar asla gönderilmez.



OSPF kimlik doğrulamasının yapılandırılması, iki adımlı bir süreçtir. İlk olarak bir alan için yönlendiricide

etkinleştirilir, daha sonra da alandaki arayüzlerde yapılandırılır.

a. İki yönlendiricide de, 0 alanında MD5 kimlik doğrulamayı etkinleştirin

R1(config)#router ospf 1

R1(config-router)#area 0 authentication message-digest

R2(config)#router ospf 1

R2(config-router)#area 0 authentication message-digest

b. R1'in S0/0/0 arayüzünde OSPF kimlik doğrulamayı etkinleştirin.

R1(config)#interface s0/0/0

R1(config-if)#ip ospf message-digest-key 10 md5 secretpassword

R2'nin S0/0/0 arayüzü üzerinde OSPF kimlik doğrulamayı etkinleştirin.

R2(config)#interface s0/0/0

R2(config-if)#ip ospf message-digest-key 10 md5 secretpassword

TROUBLESHOOTING

```
R1#show ip protocols
Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 10.1.1.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    172.16.1.16 0.0.0.15 area 0
```

```
R1#show ip ospf
<some output omitted>
Routing Process "ospf 1" with ID 10.1.1.1
Start time: 00:00:19.540, Time elapsed: 11:31:15.776
Supports only single TOS(TOS0) routes
Supports opaque LSA
Supports Link-local Signaling (LLS)
Supports area transit capability
Router is not originating router-LSAs with maximum metric
```

```
R1#show ip ospf interface serial0/0/0
Serial0/0/0 is up, line protocol is up
  Internet Address 192.168.10.1/30, Area 0
  Process ID 1, Router ID 10.1.1.1, Network Type POINT_TO_POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
oob-resync timeout 40
Hello due in 00:00:07
  Supports Link-local Signaling (LLS)
```

```
R1#show ip route
Codes: <output ommited>

Gateway of last resort is 192.168.10.2 to network 0.0.0.0

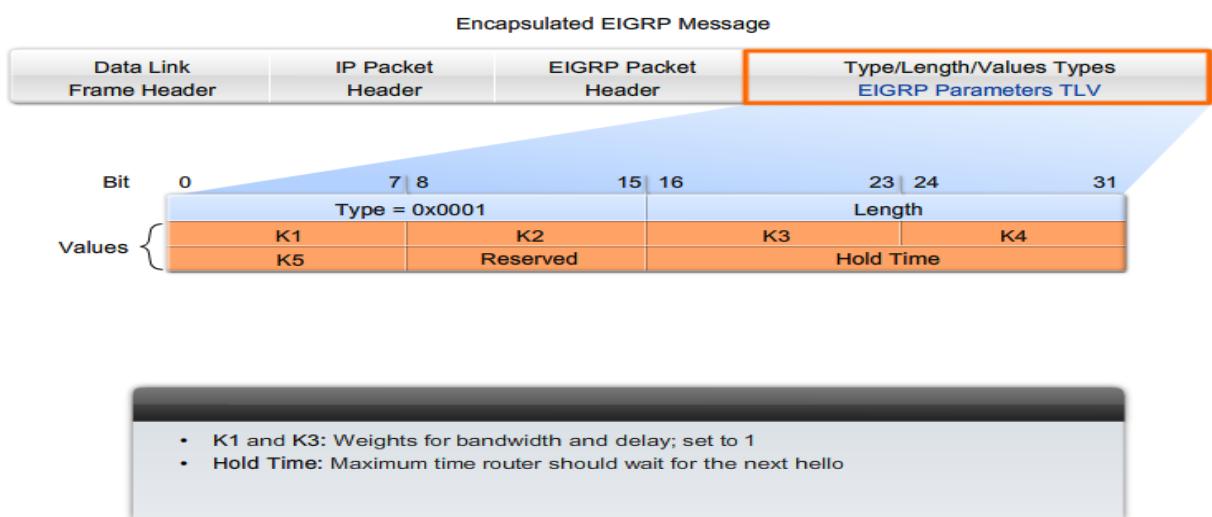
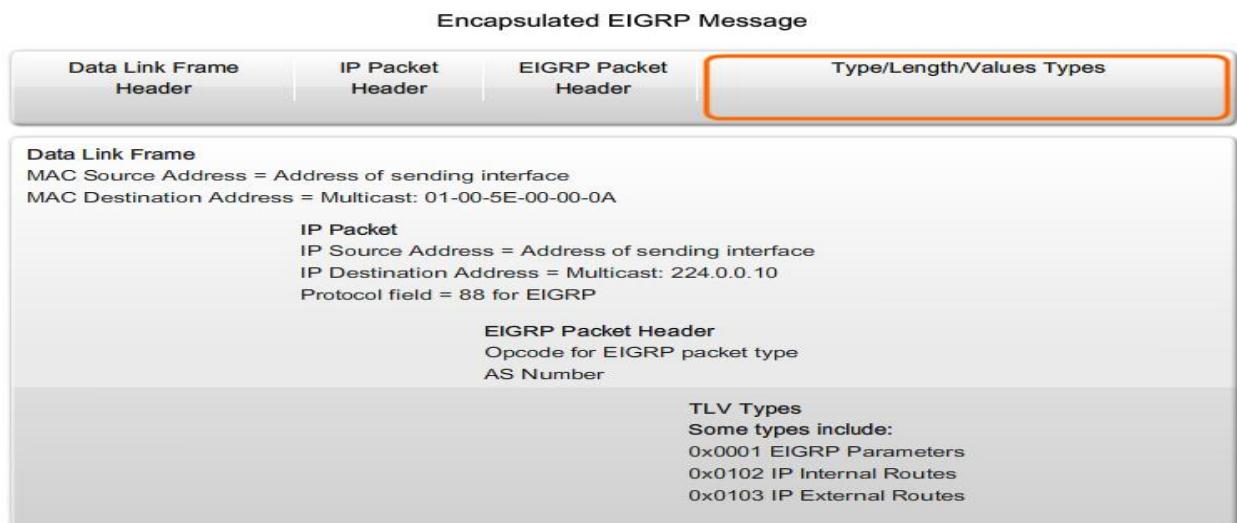
  192.168.10.0/30 is subnetted, 3 subnets
C        192.168.10.0 is directly connected, Serial0/0/0
C        192.168.10.4 is directly connected, Serial0/0/1
O        192.168.10.8 [110/1562] via 192.168.10.6, 00:01:34, Serial0/0/1
```

```
R1#show ip ospf neighbor
Neighbor ID      Pri  State      Dead Time    Address          Interface
10.2.2.2          0    FULL/-    00:00:32    192.168.10.2   Serial0/0/0
10.3.3.3          0    FULL/-    00:00:32    192.168.10.6   Serial0/0/1
R1#
```

ENHANCED INTERIOR GATEWAY ROUTING PROTOCOL (EIGRP)

- Metrik olarak varsayılan **bandwidth** ve **delay** değerlerini kullanır. Ancak istenirse, **reliability** ve **load** da eklenebilir
- **DUAL** (Diffusing Update Algorithm) algoritması kullanır.
- Administritive Distance Değeri **90** dır.
- IP paket başlığında protocol field alanı **88** dır.
- Maximum Hop count **255** dır.
- Kullandığı Multicast adres: **224.0.0.10 – 01-00-5E-00-00-0A**
- EIGRP için hybrid dense de aslında **Distance Vector** bir routing protokoldür.
- **VLSM** ve **CIDR** desteği vardır,
- **Cisco ürün spesifik** bir protokoldür. Sadece cisco cihazlarda çalışır.
- İlk komşuluk kurulduğunda full update yapar bunun dışında sınırlı güncelleme (**bounded update**) yapar. Yani oluşan değişiklikler hakkındaki kısmi güncellemeleri sadece bu bilgiye ihtiyaç duyan routelara iletir.
- Periyodik routing update yapmaz.
- **RTP** (Reliable Transport Protocol) protokolünü kullanır. RTP, tipki TCP gibi güvenli bir protokoldür.
- Protocol Dependet Module (**PDM**) sayesinde sadece IP değil, IPX, AppleTalk gibi çoklu L3 katman protokollerinin de iletişimini destekler.

- IP, IPX, AppleTalk protokollerinin herbiri için, Neighbor, Topology ve Routing tablolarını tutar.



Encapsulated EIGRP Message														
Data Link Frame Header	IP Packet Header	EIGRP Packet Header	Type/Length/Values Types IP Internal Routes TLV											
Bit	0	7 8	15 16	23 24	31									
Values	Type = 0x0102			Length										
	Next Hop													
	Delay													
	BandWidth													
	MTU			Hop Count										
	Reliability	Load	Reserved											
	Prefix Length	Destination												
	<ul style="list-style-type: none"> • Delay: Sum of delays in units of 10 microseconds from source to destination; 0xFFFFFFFF indicates unreachable route • Bandwidth: Lowest configured bandwidth of any interface along the route • Prefix Length: Specifies the number of network bits in the subnet mask • Destination: The destination address of the route 													
Data Link Frame Header	IP Packet Header	EIGRP Packet Header	Type/Length/Values Types IP External Routes TLV											
Bit	0	7 8	15 16	23 24	31									
Values	Type = 0x0103			Length										
	Next Hop													
	Originating Routers													
	Originating Autonomous System Number													
	Arbitrary Tag													
	External Protocol Metric													
	Reserved		Ext. Protocol ID	Flags										
	Delay													
	BandWidth													
	MTU			Hop Count										
	Reliability	Load	Reserved											
	Prefix Length	Destination												
Value Fields used to track external source of route														
Same value fields used in the IP Internal TLV														

Burada destination fileds alanı 24 bit olarak gösterilmiştir. 10.1.0.0 / 16 gibi bir örnekte destination alanına 10.1 yazılır. Ancak 24 bitten daha uzun bir hedef tanımlandığında (192.168.1.32 / 27) bu durumda extra 32 bitlik bir alan eklenir. (toplamda 56 bit) Geriye kalan bitler padding işlemine tabi tutulur (0 olarak işaretleme)

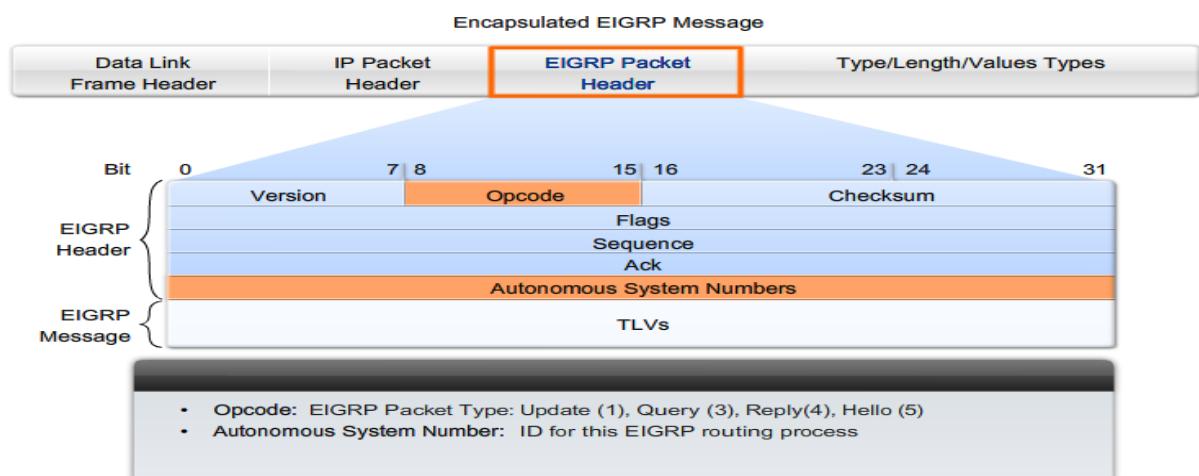
TLV

EIGRP paketlerindeki Type, Length, Value (TLV) üçlüsü, successor rota aramada ve komşunun halen ulaşılabilir olduğunu tespitinde kullanılan bir alandır.

RTP

Reliable Transport Protocol (RTP), EIGRP tarafından kullanılan paketlerin alınmasında ve gönderilmesinde güvenli iletişimini sağlayan L3 protokollerinden bağımsız bir protokoldür.

EIGRP PAKET TÜRLERİ



EIGRP paket başlığında bulunan OPCODE alanı, EIGRP paketinin ne tür bir paket olduğunu gösterir. Bu paket türleri :

- Update (1)
- Query (3)
- Reply (4)
- Hello (5) paketleridir.

Hello Packet, komşulukları belirlemede ve bu komşulukların bitişikliğini (adjacency) belirlemede kullanılır. Güvenli değildir, yani hello paketlerine cevap beklenmez. T1 gibi ağlarda her 5sn'de, NBMA ağlarda ise her 30 sn'de gönderilir.

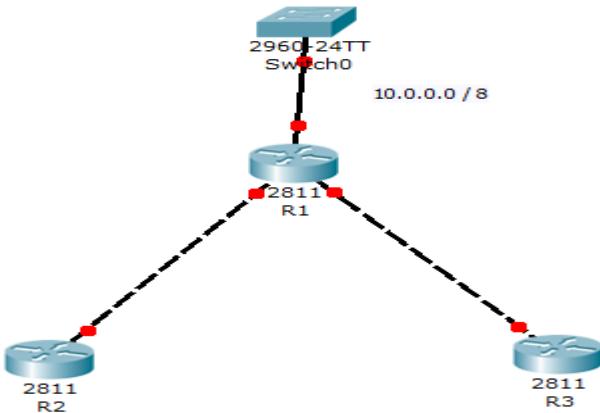
Update Packet, routing bilgilerini yayılmamada kullanılan paket türüdür. Update bilgisi eğer birden çok cihaza gidecekse multicast, tek cihaza gidecekse unicast olur.

Ack Packets, daima unicast bir pakettir, güvenli bir iletişim gereğinde onay olarak kullanılır. Örnek olarak komşudan down olan bir update bilgisi geldiğinde buna cevap olarak Ack gönderilir.

Query Packets, DUAL tarafından networklerin araştırılmasında kullanılır. Hem multicast hem de unicast olabilir.

Reply Packets, DUAL tarafından network araştırmasında kullanılır. Daima unicast bir pakettir.

Örnek :



10.0.0.0 networkü down olsun;

- R1, R2 ve R3'e 10.0.0.0 down olduğunda dair bir multicast update paketi yollar.
- R2 ve R3 buna ACK ile cevap verir.
- R1, 10.0.0.0 networkü için yeni bir rota olup olmadığını keşfetmek için, Query yapar.
- R2 ve R3 bu query'ye karşılık önce ACK, ve ardından (varsayımsa) Reply cevabı yapar.

EIGRP YAPILANDIRMA

R(config)# router eigrp 1 // 1 değeri ASN değeridir. 1 ile 65535 arasında verilebilir.

R(config – router)# network 192.168.1.0 0.0.0.255 //wildcard mask değeri kullanılır.

* * * ASN, IANA tarafından atanmış autonom sistem numarası değildir. Proccess – id gibi çalışır. Ancak bu değer tüm **routerlarda aynı olmalıdır**. *AS numarası 2007 yılına kadar 16 bit idi, bu tarihten sonra 32 bit oldu*

EIGRP birbirine bağlı olarak çalışan çoklu tablolar kullanır. Bunlar:

- Neighbor Table
- Topology Table
- Routing Table

Neighbor Table (Komşuluk Tablosu):

Doğrudan bağlı komşu routerların bilgilerini tutar. Neighbor Table her router tarafından RAM'de tutulan tablodur. Yeni bir komşu öğrenilince bu komşunun adresi ve arayüzü saklanır. Her PDM (protocol dependent module) için bir tablo tutulur. Update mesajlarındaki sequence numarası saklandığı için, sıralı olmayan bir update paketi dikkate alınmaz.

Router, komşuya bir **hello** paketi gönderdiğinde, buna bir cevap bekler. Belli bir süre (**hold-time**) cevap gelmezse komşunun down olduğu farz edilir ve **DUAL** yeniden topolojiyi hesaplar.

Komşuluk tablosunu görmek için;

R# show ip eigrp neighbors komutu kullanılır.

R2#show ip eigrp neighbors						
IP-EIGRP neighbors for process 1						
H	Address	Interface	Hold (sec)	Uptime (sec)	SRTT (ms)	RTO (ms)
1	192.168.10.10	Se0/0/1	10	00:01:41	20	200 0 7
0	172.16.3.1	Se0/0/0	10	00:09:49	25	200 0 28

Diagram illustrating the fields in the EIGRP neighbor table:

- Address of neighbors**: Points to the Address column.
- Interface connected to neighbor**: Points to the Interface column.
- Amount of time left before neighbor is considered "down"**: Points to the Hold (sec) column.
- Amount of time since adjacency was established**: Points to the Uptime (sec) column.

- H (handle)**: Cisco IOS tarafından komşu router takibi için kullanılan değerdir. Öğrenilen ilk router'dan itibaren 0,1,2... şeklinde gider.
- Address**: Komşu router Layer3 adresidir.
- Interface**: Komşu router'a bağlı olan local arayüz.
- Hold (hold time)**: Saniye cinsinden komşunun down olduğunu deklere etmesi için beklenen süredir. Bu süre içinde hello paketi beklenir ancak herhangi bir EIGRP paket türü alındığında da bu değer sıfırlanır.
- Uptime**: Komşu router'ın Komuşuluk tablosuna eklendiğinden beri geçen (saat, dk, sn cinsinden) süredir.
- SRTT (smoothed round-trip time)**: Komşu routera gönderilen bir EIGRP paketine alınan ack için geçen milisaniye cinsinden toplam ortalama süredir. Bu sayaç, retransmit interval değerini (RTO) belirler.
- RTO (retransmission timeout)**: EIGRP Update, Query ve Relpy paketleri gönderildikten sonra Ack bekler. Gönderilen bir paketin kopyası kuyruğa atılır. RTO süresi (milisaniye) sonunda ACK gelmezse, paket tekrar gönderilir.
- Q Cnt (queue count)**: Kuyrukta gönderilmeyi bekleyen paket sayısıdır. Bu değer 0'dan farklı ise congeston problemi olduğu anlamına gelir.
- Seq Num**: Komşudan en son alınan Update, Query ya da Reply paketinin sıra numarasıdır. Paket sıralamasında kullanılır.

Topology Table:

Her komşu tarafından her hedef network için rapor edilen (AD) yolu tutan tablodur. (Yolların metrik değerlerini de içeren Network haritası olarak düşünülebilir)

```
R2#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.1.1.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
<output omitted>
P 192.168.1.0/24, 1 successors, FD is 3014400
  via 192.168.10.10 (3014400/28160), Serial0/0/1
  via 172.16.3.1 (41026560/2172416), Serial0/0/0
P 192.168.10.8/30, 1 successors, FD is 3011840
  via Connected, Serial0/0/1
<output omitted>
```

Bir rota kaybolduğunda “aktif - A” duruma geçer ve DUAL hedefe yeni bir yol arar. Bulunan rota Routing tablosuna “pasif – P” olarak yerleştirilir. Topoloji tablosunda, A ,P ve bunların dışında değerler de bulunabilir.

- P (Passive):** Network ulaşılabilirdir. Routing tablosunda kayıt aktarılmıştır. Stable network için bu değer P olmalıdır. DUAL'in anlık olarak çalışmadığı anlamına gelir. Çünkü hedef network için rota vardır.
- A (Active):** Network ulaşılamaz durumdadır. DUAL çalışıyor. Routing tablosunda hedef network için rota yoktur.
- U (Update):** Network hakkında update yapılıyor. Aynı zamanda Update için ACK beklenliğinde de bu değer görünür.
- Q (Query):** Hedef network için Query yapılıyor. Aynı zamanda bu query'ye cevap bekleneme (ACK) süresince de görünen koddur.
- R (Reply status):** Router network hakkında Reply yapıyordur ya da Reply'e ACK bekleniyordur.
- S (Stuck-in-active status):** EIGRP convergence problemi vardır.

DUAL, topology ve neighbor tablolarını kullanarak en düşük maliyetli yolu hesaplar. Default olarak 4 eşit costa sahip rota için load balancing yapar. Bu değer 16'ya kadar çıkarılabilir.

Feasible Distance (FD), hedefe giden yollar arasındaki en iyi rotanın metrik değeridir.

Advertised Distance, komşu tarafında bildirilen en iyi metriktir.

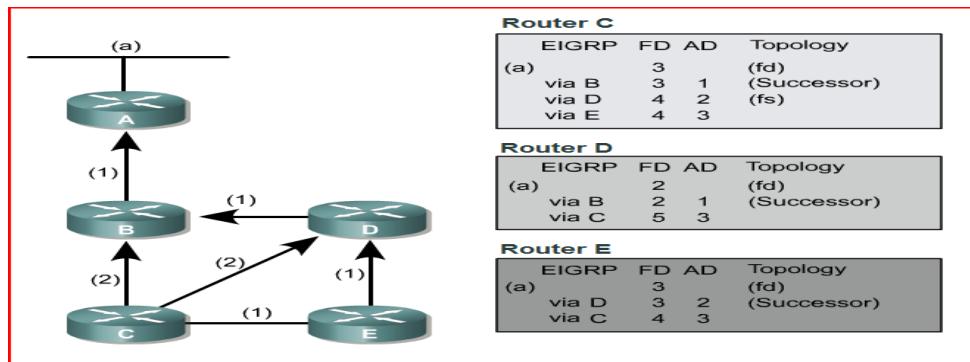
Successor Route: Routing tablosunda tutulan, “loop” oluşturmayan, hedefe en düşük metrik değere sahip yol olarak tanımlanır (Varis Rota)

Feasible Successor (FS): Topoloji tablosunda görünür ancak routing tablosunda görünmeyen hedefe ulaşılacak yedek rotayı temsil eder. EIGRP max 6 FS 'yi topoloji tablosunda tutar ve bunlardan en iyi metrik değerine sahip olan routing tablosuna aktarılır (successor).

sh ip EIGRP topology komutu ile bu yollar görüntülenebilir.

DUAL, loop oluşturmayan en iyi yolu (Successor) ve ikincil yolu (Feasible Successor) belirler. En iyi yol routing tablosuna aktarılır. En iyi rota down olduğunda yedek rota en iyi yol olarak hemen routing tablosuna aktarılır.

DUAL ÇALIŞMA PRENSİBİ ÖRNEĞİ



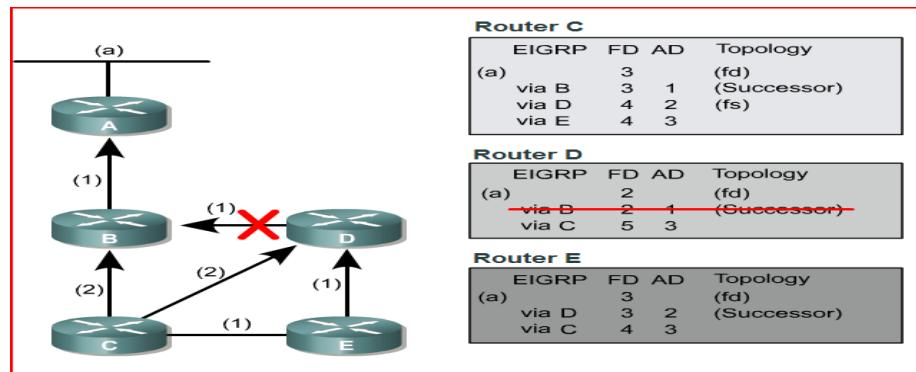
Şekil1'de tüm routerlar converged şekildedir. Her bir Router'in topoloji tablosu gösterilmiştir. Her router (a) ağına ulaşmak için en az bir yolu vardır.

Örneğin,

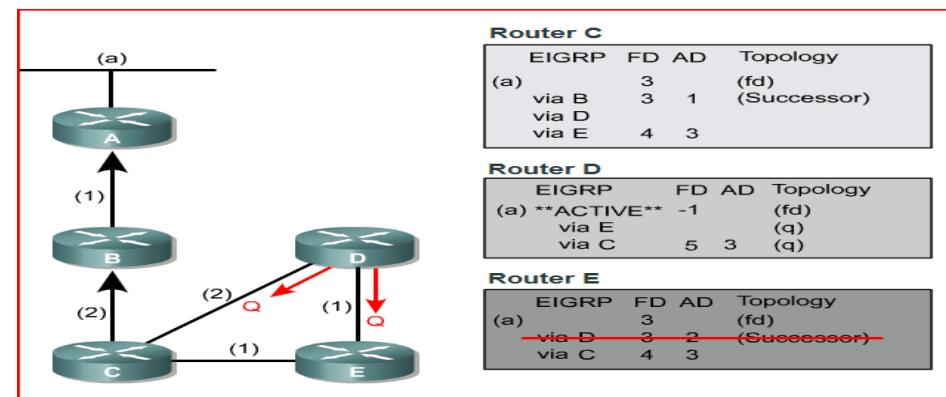
- Router C, Router B' den geçen yolu Successor rota olarak belirlemiştir. (FD =3)
- Router D, Router B den geçen yolu successor rota olarak belirlemiştir. (FD=2)
- Router E, RouterD üzerinden geçen yolu successor olarak belirlemiştir. (FD=3)

Bir rotanın Feasible Successor olması için, o rotanın AD değeri, en iyi rotanın değerinden (FD) daha düşük olmalıdır. Bu durumda, Router C aynı zamanda (a) networkü için AD(2) değeri FD (3) değerinden düşük olan ikinci bir yedek rotaya sahiptir. RouterD üzerinden geçen rota, **feasible successor (fs)** rota olarak belirlenmiştir.

Diğer rotalar için FS tanımlı değildir. Çünkü, diğer tüm rotalarda AD değeri FD değerinden yüksek ya da eşittir.



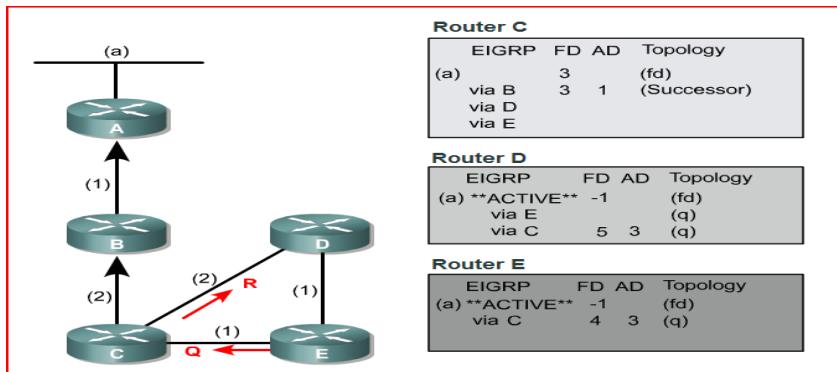
RouterD , (a) ağ ile olan iletişimini kaybettiğini farz edelim. RouterD için bir FS tanımlı olmadığı için, (a) networküne ulaşacak yeni bir yol bulmak için DUAL çalıştırır ve RouterB üzerinden geçen rotayı silecektir.



RouterD, ACTIVE moda düşecektir ve RouterE ile RouterC ‘ye (a) networku için başka bir yol olup olmadığını sorgular (Query yapar)

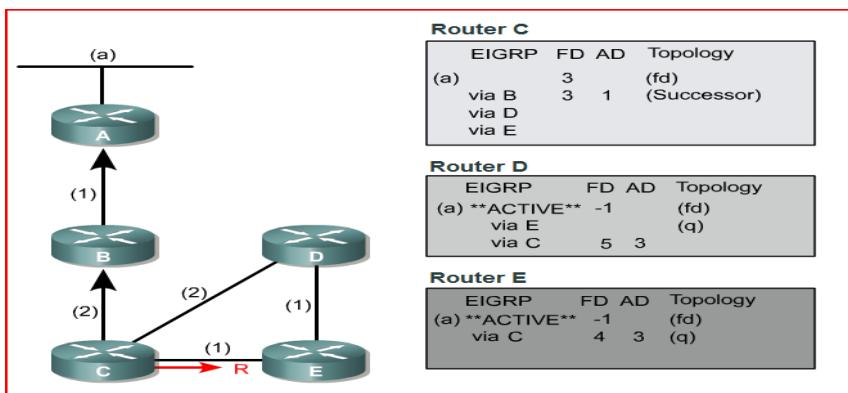
RouterE, bu soruyu (kendi successor rotasından) aldıktan sonra RouterD üzerinden geçen rotayı topoloji tablosundan silecek ve ACTIVE moda düşecektir. Çünkü kendisinin de FS rotası yoktur.

RouterC, RouterD’den gelen soruyu aldıktan sonra RouterD üzerinden geçen FS rotayı silecektir. Ancak ACTIVE moda düşmeyecektir çünkü (a) networkü için successor bir yolu vardır.

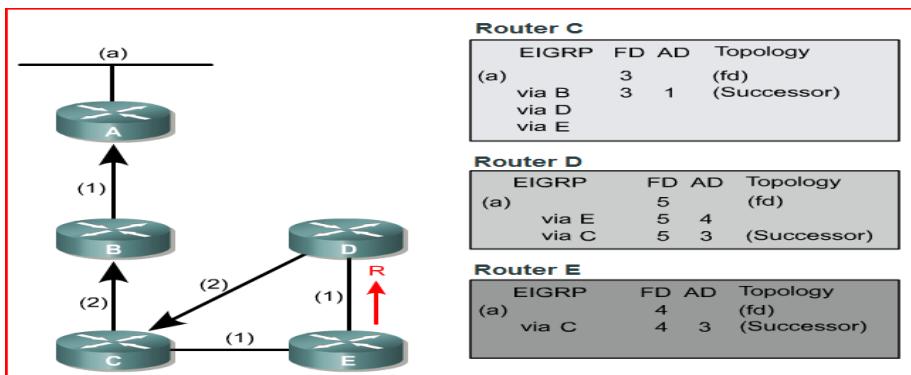


RouterC, RouterD'nin yaptığı soruya cevap (REPLY) verir. Bu cevapta, RouterB üzerinden bir yol olduğu bildirilir.

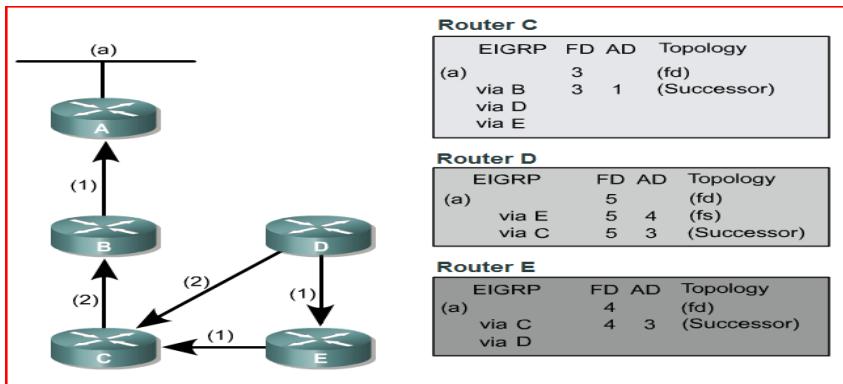
Bu arada RouterE de ACTIVE moda olduğu için RouterC'ye (a) networkü için Query yapar.



RouterD, RouterC tarafından yayınlanan yeni bilgiyi topoloji tablosuna ekler. Bu arada RouterC, RouterE'nin sorusuna cevap verir (REPLY)



RouterD, (a) networkü için RouterC üzerinden geçen rotayı Successor rota olarak belirler. RouterE de RouterC tarafından yayınlanan rotayı öğrenmiş, Successor olarak atamıştır. Ayrıca bu yeni bilgiyi RouterD'ye bildirir.

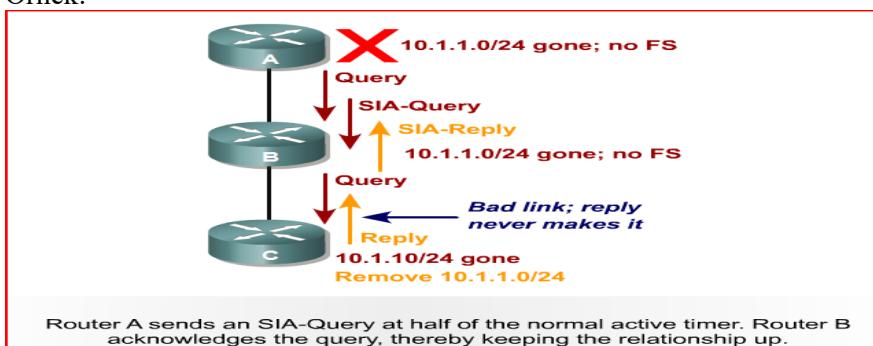


RouterD, RouterE'den gelen bilgiyi aldıktan sonra, (a) networkü için ikinci bir rotası olduğunu öğrenir. Çünkü RouterD tarafından gönderilen AD değeri (4), kendi FD değerinden (5) daha düşüktür. Bu sebepten dolayı (a) networkü için RouterE'den geçen rotayı FS olarak işaretler.

Router, ACTIVE duruma düşüğünde Query yapar ve bu Query'lere cevap bekler. Default olarak **180** saniye içinde bir Reply gelmezse ilgili rota **SIA (Stuck In Active)** olarak tanımlanır. Bu durumda komşu ile varolan bağlantının sorunlu olduğu varsayılar ve komşuluk ilişkisi sıfırlanır.

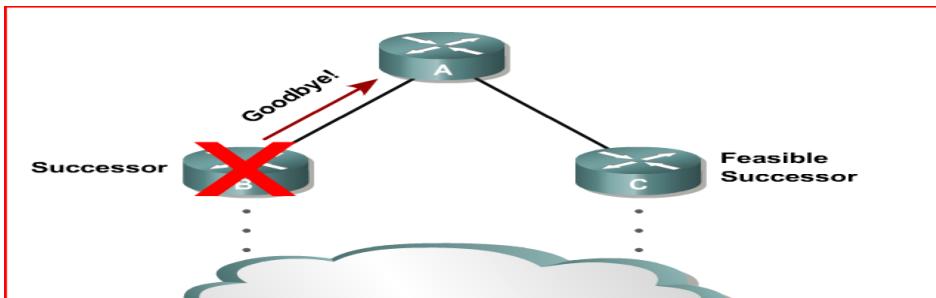
ACTIVE PROCESS ENHANCEMENT

Örnek:



Farzedelim ki RouterA üzerinde 10.1.1.0 / 24 ağı için rota kaybolsun ve ACTIVE duruma düşsun. Bu durumda Router A, Router B 'ye Query yapacaktır. Router B üzerinde bu ağ için bir rota yoksa bu durumu Router C'ye Query olarak gönderecektir. Router B ile Router C arasındaki hattın problemlerinden dolayı REPLY cevabı gelmediğini farzedelim. Bu durumda Router A, Query yaptıktan sonra 180 sn içinde bir REPLY alamadığı için RouterB ile olan komşuluğunu sıfırlayacaktır. Bu sebeple RouterB üzerinden öğrenilen diğer geçerli rotalar da silinecektir. Bu sorunu çözmek için, Router B, Router A 'dan Query aldıktan sonra 90 sn (default) sonra ACK gönderip, komşuluğunun sürdürülmesini sağlar.

EIGRP GRACEFUL SHUTDOWN



RouterA, bir çok rota için RouterB üzerinden geçen yolu Successor; RouterC üzerinden geçen yolu da Feasible Successor olarak belirlemiştir. RouterB üzerinde EIGRP iptal edilsin. Normal şartlar altında RouterA, hold-down süresi boyunca bekledikten sonra RouterB nin down olduğunu anlayacaktır. Bu arada geçen süre boyunca routing kayıtları olmaması için, RouterB tarafından RouterA'ya , tüm K değerleri 255 olarak set edilmiş Hello Paketi (**Graceful Shutdown**) gönderilecektir.

```
*Apr 26 13:48:42.523: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1:  
Neighbor 10.1.1.1 (Ethernet0/0) is down: Interface  
Goodbye received
```

RouterA, yukarıdaki gibi bir uyarı alacaktır.

STATIC NEIGHBORSHIP

Statik olarak EIGRP komşuluk kurulabilmesi için her iki routerda da aşağıdaki formatta komut yazılmalıdır. **neighbor Komşu_IP_Adresi Çıkış_Arayüzü**

Örnek:

R1#show running-config

```
router eigrp 9  
network 172.16.0.0  
no auto-summary  
neighbor 10.10.15.5 Serial0/0/0.5
```

R5#show running-config

```
router eigrp 9  
no auto-summary  
neighbor 10.10.15.1 Serial0/0.1
```

Static neighbor görülebilmesi için, [show ip eigrp topology detail](#) komutu kullanılmalıdır.

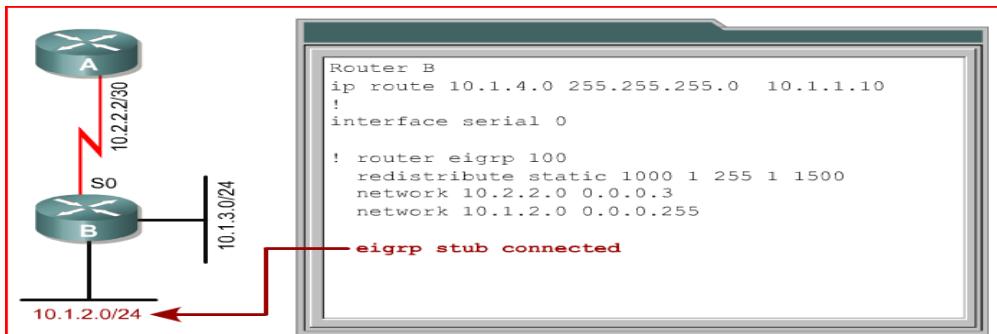
EIGRP STUB

Router(config – router)# eigrp stub [receive-only | connected | static | summary]

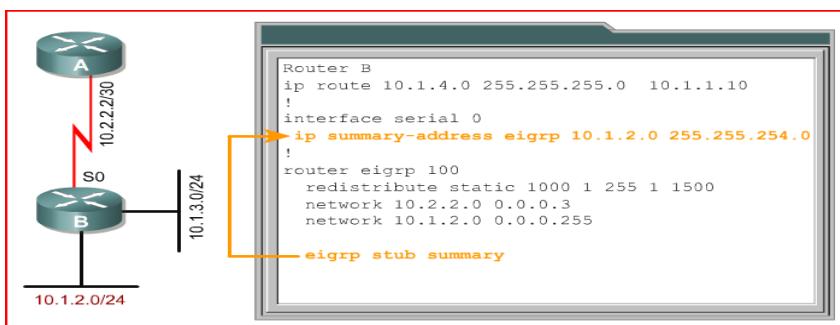
- **receive-only**: Stub ‘in herhangi bir rota yayılması engeller
- **connected**: Stub’ın connected rotaları yayılmasına izin verir.
- **static**: static rotaları göndermeye izin verir.
- **summary**: Özettemeye izin verir.

Default olarak **connected** ve **summary** olarak çalışır

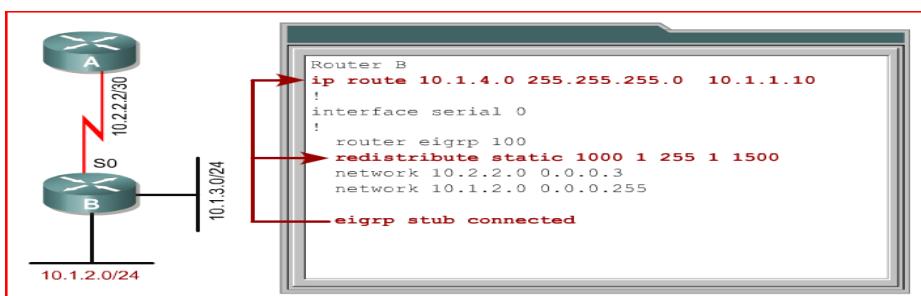
Örnek:



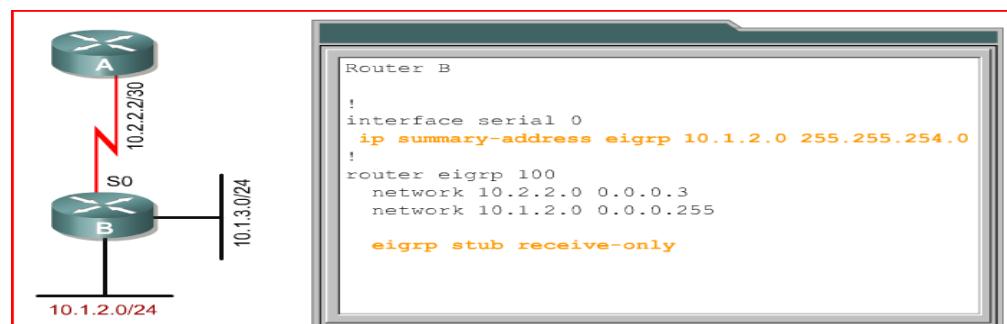
Router B üzerinde, **eigrp stub connected** komutu kullanıldığında, sadece 10.1.2.0/24 networkünün yayımı yapılacaktır. Her ne kadar 10.1.3.0 /24 networkü de connected olsa da, EIGRP tanımlaması içinde **network 10.1.3.0** ağının tanımlanmadığından bu ağın yayımı yapılmaz.



RouterB üzerinde **eigrp stub summary** komutu kullanıldığında RouterB üzerinden sadece 10.1.2.0 / 23 yayımı yapılacaktır.



Bu örnekte ise, **eigrp stub static** komutu kullanıldığında, Router B sadece 10.1.4.0 / 24 statik rotasını yaymayıacaktır.



Bu örnekte ise Router B de kullanılan **eigrp stub receive-only** komutu ile herhangi bir yayım yapmayacağındır.

EIGRP ile öğrenilen rotalar, routing tablosunda D ile gösterilir. Diğer protokollerden öğrenilen bilgiler ise D EX olarak işaretlenir.

EIGRP yönlendiricileri, komşularını keşfetmek ve bitişiklik (**adjacency**) gerçekleştirmek için hello paketleri kullanır.

T1 ve üzeri hızlardaki hatlarda hello paketi gönderme sıklığı 5 sn, (hold – time : 15)

T1 den düşük hızlardaki hatlarda ise bu süre 60 sn.dir (hold – time : 180 sn)

Hold Time süresi ise hello interval değerinin 3 katıdır.

Hold Time süresi, komşudan belli bir süre hello paketi gelmediği durumda, komşunun ulaşılmasının devam etmesi için geçmesi gereken süredir.

METRIC: Bandwidth, Delay, Load, Reliability değerlerinin kompozit bir ölçüğünü kullanır, ancak varsayılan olarak sadece BW ve DELAY hesaba katılır.

$$\text{METRIC} = (K1 * bw) + [(K2 * bw) / (256 - load)] + K3 * delay * [K5 / (reliability + K4)]$$

Varsayılan olarak K2, K4 ve K5 değerleri 0 dır. Bu K5 değerinin 0 olduğu durumda, turuncu işaretlenmiş ifadeler metrik hesabında kullanılmaz, bu durumda metrik, **Metric = (K1 * BW + K3 * DELAY)** olur.

Formülize edersek;

$$\text{metric} = \{[10^7 / \text{en düşük bw (kbps)}] + (\text{toplam gecikme}/10)\} * 256$$

Burda DELAY = toplam gecikme(micro sn)/10 * 256 dır.

BW, rotadaki en düşük bandwidth (kbps) olmak üzere, **10^7 /bw * 256**

R# show ip protocols Komutu ile K değerleri görülebilir.

K değerleri her routerda aynı olmalıdır. K değerleri **metrics weights tos K1 K2 K3 K4 K5** (K değerleri 0 – 255 arasındadır) komutuyla değiştirilir. Tos değeri IGRP den kalma bir değer olup kullanılmamaktadır ve her zaman 0 olarak ayarlanmalıdır. Bu durumda örnek olarak;

R(config – router) # metrics weights 0 1 2 2 1 1

K değerlerinin farklı olması durumunda komşuluk kurulmayacak ve aşağıdaki gibi bir log görüntülenecektir.

Feb 23 18:48:24.907: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.12.1 (Serial0/0/1) is down: K-value mismatch

Bandwidth, kbps biriminde bant genişliğini ifade eder. Varsayılan olarak bu değer 1544 Kbps dır.

Eğer bir seri interface'de bandwidth komutu ile bir bandwidth değeri belirlenmemişse, EIGRP bunu default olarak T1 hattı olarak kabul eder. Bu değer doğru bir şekilde yazılmazsa convergence olayı geç olusabilir ve best path seçimi hatalı yapılabilir.

PPP veya HDLC seri hatlarında bandwidth değerinin hattın hızı olarak ayarlanması gereklidir. Point – to – Point Frame Relay hatlarında ise bu değer CIR değeri olarak kullanılması mantıklıdır. Multipoint Frame Relay hatlarında ise bandwidth değeri toplam CIR olarak yazılması gereklidir. Eğer PVC'ler farklı CIR değerleri kullanıyorsa bu durumda en düşük CIR değeri ile PVC sayısı çarpımı (CIR min * PVS Count) yazılması gereklidir.

Delay, ölçüde kullanılan gecikmeyi temsil eder ve otomatik olarak hesaplanan bir değer değildir. Bu değeri değiştirmek fiziksel gecikmeye etki etmez.

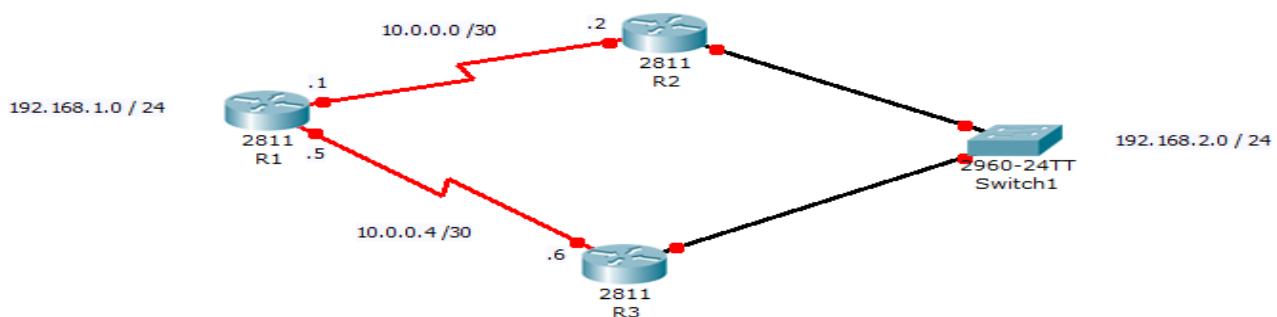
R(config – if) # delay 100 // mikrosaniye cinsinden değer komutuyla değiştirilir.

Reliability, otomatik olarak hesaplanır. 255 / 255 demek %100 güvenirliliği temsil eder.

Load, trafik miktarını belirler. Düşük değer tercih edilir. 1/255 değeri 255/255'e göre daha uygundur. (5dk de bir dynamic olarak işlenir)

R# Show Interface Serial 0/0/0 komutu ile MTU, BW, DLY, REL, LOAD değerleri öğrenilebilir.

METRİK HESAPLAMA ÖRNEĞİ:



Bw = rotadaki en düşük değerli bandwidth

Dly = rotadaki toplam gecikme süresi.

En düşük bw = $(10\ 000\ 000 / \text{bw kbps}) * 256$

+ Toplam gecikme = $(\text{toplam delay} / 10) * 256$

Metric = EIGRP METRIC

1.ADIM

R1 – R2 BW = 1544 kbps R1 – R3 BW = 1544 kbps

R1 – R3 dly = 20 000 R1 – R2 dly = 20 000

R2 – SWITCH dly = 100 R3 – SWITCH dly = 100

R1#sh ip route

```
.....  
D 192.168.2.0 /24      [90 / 2172416]  via  10.0.0.2  s0/0/0  
                           [90 / 2172416]  via  10.0.0.6  s0/0/1
```

*** burada en iyi rota eşit maliyetli iki rotadır.****

R1 – R2 – SW yolu üzerindeki toplam gecikme = $20\ 000 + 100 = 20100$ / $10 = 2010$ * $256 = 514.560$

R1 – R2 – SW yolundaki en düşük bw = 1544 ise, $10\ 000\ 000 / 1544 = 6.476$ * $256 = 1.657.856$

Metric = $1.657.856 + 514.560 = 2.172.416$

2.ADIM

R1 – R3 arasındaki bandwidth değerini değiştirelim,

R1(config – if)# bandwidth 64 (kbps)

R1# sh ip route

```
.....  
D 192.168.2.0 /24      [90 / 2172416]  via  10.0.0.6  s0/0/1  
****burda 2172416 FD , 10.0.0.6 ise successor dur.****
```

R1# show ip eigrp topology

```
P 192.168.2.0 / 24  1 successor, FD is 2172416  
                     via 10.0.0.6 (2172416 / 28160), s0/0/1  
                     via 10.0.0.2 (40514560/28160), s0/0/0
```

***** burdaki 28160 değeri, R3'ün 192.168.2.0 /24 networkü için rapor ettiği değerdir. (Reported Distance)

Yukardaki örnekte,

R# show ip eigrp topolgy 192.168.2.0 komutu kullanılrsa,

Rotadaki min bw, toplam gecikme gibi bilgileri verir.

ROTA ÖZETLEME

EIGRP, clasfull olarak otomatik özetleme yapar. Otomatik özetlemeyi Null 0 arayüzüne yapar. Null 0 arayüzüne yapılan yönlendirme satırı otomatik yapılır. Böylece otomatik özetlemeye uygun paketletler routera geldiğinde ve uygun child bir rota ile eşleşmezse null 0 arayüzüne forward edilir. (Discard) IP CLASSLESS komutunun kullanılmış olup olmadığı durumu değiştirmez, çünkü otomatik özetin kendisi de bir child rotadır. Otomatik özetlemeyi iptal etmek için;

R(config – router)# no auto-summary komutunu kullanmak gereklidir.

Manuel Summerization

R(config – if)# ip summary – address eigrp 1 192.168.0.0 255.255.252.0

Authentication başarısız olursa hello mesajları ignore edilir, komşuluk kurulmaz.

```
R(config)# key chain zinciradi  
R(config – keychain)# key numara
```

```
R(config - keychain - key)# key-string text
```

Burdaki zinciradi ifadesi her router için localdir, aynı olmak zorunda değildir ancak, text ifadesi parola olarak düşünülebilir ve tüm routertlarda aynı olmak zorundadır.

```
R(config - if )# ip authentication key - chain eigrp 1 zinciradi //buradaki 1 değeri ASN numarasıdır.
```

MD5 Authentication için;

```
R(config - if )# ip authentication mode eigrp 1 md5
R(config - if )# ip authentication key - chain eigrp 1 zinciradi
```

```
R(config - keychain - key)# accept - life time start - time 10:45:34 11 29 2009 //
SA - DK - SN AY GUN YIL
```

```
R(config - keychain - key)# send - life time start - time 10:45:34 DEC 29 2009
```

Örnek



R1 Config.

```
key chain R1chain
key 1
key-string firstkey
accept-lifetime 04:00:00 Jan 1 2006 infinite
send-lifetime 04:00:00 Jan 1 2006 04:01:00 Jan 1 2006
key 2
key-string secondkey
accept-lifetime 04:00:00 Jan 1 2006 infinite
send-lifetime 04:00:00 Jan 1 2006 infinite
```

R2 Config

```
key chain R2chain
key 1
key-string firstkey
accept-lifetime 04:00:00 Jan 1 2006 infinite
send-lifetime 04:00:00 Jan 1 2006 infinite
key 2
key-string secondkey
accept-lifetime 04:00:00 Jan 1 2006 infinite
send-lifetime 04:00:00 Jan 1 2006 infinite
```

```
R1#debug eigrp packets
EIGRP Packets debugging is on
(UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE,
ACK, STUB, SIAREQUEST, SIAREPLY)
*Jan 21 16:38:51.745: EIGRP: received packet with MD5
authentication, key id = 1
*Jan 21 16:38:51.745: EIGRP: Received HELLO on Serial0/0/1
nbr 192.168.1.102
*Jan 21 16:38:51.745: AS 100, Flags 0x0, Seq 0/0 idbQ
0/0 iidbQ un/rely 0/0 peerQ un/rely 0/0
```

** show key chain veya debug eigrp packet ile authentication check edilebilir.

HELLO PACKET INTERVAL ve HOLD TIME DEĞERLERİNİ DEĞİŞTİRME

R(config – if) # ip hello – interval eigrp 1 20 // 1 değeri ASN, 20 değeri saniye cinsinden interval değeridir.

R(config – if) # ip hold – time eigrp 1 60

STATIC DEFAULT ROUTE CONFIGURATION

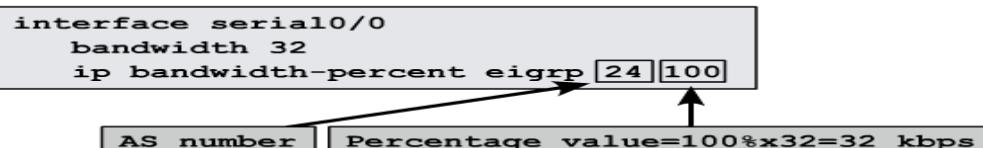
R(config)# ip route 0.0.0.0 0.0.0.0 serial 0/0/0

R(config – router) # redistribute static

EIGRP Bandwidth Utilization

Varsayılan olarak EIGRP bandwidth değerinin sadece %50 ‘sini kullanır. **ip bandwidth-percent eigrp as-number percent** komutu ile bu değer değiştirilebilir.

Router(config-if)#ip bandwidth-percent eigrp 100 25



EIGRP INTERFACE'LER

```
R1#show ip eigrp interfaces
IP-EIGRP interfaces for process 100
      Xmit Queue Mean Pacing Time Multicast Pending
Interface Peers Un/Reliable SRTT Un/Reliable Flow Timer Routes
Fa0/0     0       0/0      0      0/10      0      0
Se0/0/1   1       0/0      10     10/380    424     0
```

Interface: EIGRP konfigüre edilen arayüz.

- **Peers:** Directly Connected EIGRP komşu numarası .
- **Xmit Queue Un/Reliable:** Kuyrukta gönderilmeyi bekleyen Güvenilir / Güvensiz paket sayısı.
- **Mean SRTT:** Milisaniye cinsinden Mean SRTT zaman aralığı.
- **Pacing Time Un/Reliable:** Hız denetimi süresi
- **Multicast Flow Timer:** Gönderilen Multicast EIGRP paketi için max. saniye
- **Pending Routes:** Kuyruktaki paketteki gönderilmeyi bekleyen max. rota sayısıdır

```
R1#show ip eigrp traffic
IP-EIGRP Traffic Statistics for AS 100
Hellos sent/received: 429/192
Updates sent/received: 4/4
Querries sent/received: 1/0
Replies sent/received: 0/1
Acks sent/received: 4/3
Input queue high water mark 1, 0 drops
SIA-Queries sent/received: 0/0
SIA-Replies sent/received: 0/0
Hello Process ID: 113
PDM Process ID: 73
```

SHOW KOMUTLARI ÖRNEK ÇİKTILAR

R2#show ip protocols

```

Routing Protocol is "eigrp 1"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Default networks flagged in outgoing updates
Default networks accepted from incoming updates
EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
EIGRP maximum hopcount 100
EIGRP maximum metric variance 1
Redistributing: eigrp 1
EIGRP NSF-aware route hold timer is 240s
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
10.0.0.0
Passive Interface(s):
FastEthernet0/0
FastEthernet0/1
Routing Information Sources:
Gateway Distance Last Update
10.1.12.1 90 00:00:39
10.1.23.1 90 00:00:39
Distance: internal 90 external 170

```

R1#show ip eigrp interfaces detail fa0/1
 IP-EIGRP interfaces for process 9

Xmit Queue Interface	Mean Peers	Pacing Time Un/Reliable	Multicast SRTT	Pending Un/Reliable
Flow Timer	Routes			
Fa0/1	3	0/0	535	0/1
50	0			

Hello interval is 2 sec
Next xmit serial <none>
Un/reliable mcasts: 0/1 Un/reliable ucasts: 4/9
Mcast exceptions: 1 CR packets: 1 ACKs suppressed: 1
Retransmissions sent: 2 Out-of-sequence rcvd: 0
Authentication mode is not set
Use multicast

R1#sh ip eigrp topology

IP-EIGRP Topology Table for AS(28)/ID(192.168.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
 r - reply Status, s - sia Status

P 192.168.1.0/24, 1 successors, FD is 28160
 via Connected, FastEthernet0/0
 P 192.168.2.0/24, 2 successors, FD is 2172416
 via 172.16.1.2 (2172416/28160), Serial1/0
 via 172.16.2.2 (2172416/28160), Serial1/1

```

via 172.16.3.2 (3847680/28160), Serial1/2
P 172.16.1.0/24, 1 successors, FD is 2169856
    via Connected, Serial1/0
P 172.16.2.0/24, 1 successors, FD is 2169856
    via Connected, Serial1/1
P 172.16.3.0/24, 1 successors, FD is 3845120
    via Connected, Serial1/2

```

R1#sh ip route

```

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    172.16.0.0/24 is subnetted, 3 subnets
C  172.16.1.0 is directly connected, Serial1/0
C  172.16.2.0 is directly connected, Serial1/1
C  172.16.3.0 is directly connected, Serial1/2
C  192.168.1.0/24 is directly connected, FastEthernet0/0
D  192.168.2.0/24 [90/2172416] via 172.16.2.2, 00:00:23, Serial1/1
                           [90/2172416] via 172.16.1.2, 00:00:23, Serial1/0

```

Dikkat edilirse Topology Table'da yer alan 3 yoldan metric değerleri eşit olan (2172416) yolar Routing Table'da ve Load Balancing yapılıyor. Metric değeri 3847680 olan üçüncü bir yol ise Topology Table'da. Burada variance komutunu kullanarak üçüncü yol üzerinden de trafiğin akmasını sağlamaya çalışacağız. Basit bir hesap yaparak kullanacağımız variance değerini şu şekilde belirleyebiliriz;

$3847680 / 2172416 = 1.77$ (Yani 2 kullanabiliriz)

R1 Router'ı üzerindeki konfigürasyona variance komutunu ekleyelim.

```

R1(config)#router eigrp 28
R1(config-router)#variance 2
R1(config-router)#

```

Routing Table'ımız artık aşağıdaki şekilde olacaktır.

R1#sh ip route

```

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route

```

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

```
    172.16.0.0/24 is subnetted, 3 subnets
C  172.16.1.0 is directly connected, Serial1/0
C  172.16.2.0 is directly connected, Serial1/1
C  172.16.3.0 is directly connected, Serial1/2
C  192.168.1.0/24 is directly connected, FastEthernet0/0
D  192.168.2.0/24  [90/3847680] via 172.16.3.2, 00:01:01, Serial1/2
                           [90/2172416] via 172.16.2.2, 00:01:01, Serial1/1
                           [90/2172416] via 172.16.1.2, 00:01:01, Serial1/0
```

Burada default olarak 4 yola kadar Load Balancing yapar ve bu istenirse 16 yola kadar çıkartılabilir.

R1(config)#router eigrp 28

R1(config-router)#maximum-paths ?

<1-16> Number of paths

R1(config-router)#maximum-paths 6

R1(config-router)#maximum-paths 1 *yapılırsa load balancing iptal olur.*

KISA KISA

- EIGRP 224.0.0.10 multicast adresinden yayın yapar, eğer bir cevap gelmezse bu kez unicast mesaj gönderilir. 16. Unicast mesajına da cevap gelmezse komşunun öldüğü ilan edilir.
- Default olarak classfull auto summary yapar.
- EIGRP eşit maliyete sahip olan 4 (default=4, max=6) yol arasında load balancing yapar.

Pod1R1(config-router)# maximum-paths ?

<1-6> Number of paths

- EIGRP tipki IGRP gibi 255 hop count'a kadar destekler (default =100, max=255)

Pod1R1(config-router)# metric maximum-hops ?

<1-255> Hop count

- EIGRP yayınının yapılmasını ve alınmasını istemediğimiz bir arayüz için;

Router(config-router)# passive-interface serial 0/1 komutu kullanılır.

RIP de EIGRP den farklı olarak bu komut yayın yapılmasını engeller ama yayın alınmasını engelmez.

- EIGRP sadece MD5 authentication destekler
- EIGRP'de ASN numarası ve K değerleri her routerda aynı olmalıdır. Ancak hello-interval, hold-interval değerleri aynı olmak zorunda değildir.

FRAME – RELAY (ÇERÇEVE AKTARIM)

ITU – T tarafından standartlaştırılmış data-link katmanında çalışan yüksek performanslı bir WAN teknolojisidir. Frame – Relay bağlantılarında üç cihazlar, DTE cihazlardır. DCE cihazlar ise Frame – Relay switchlerdir.

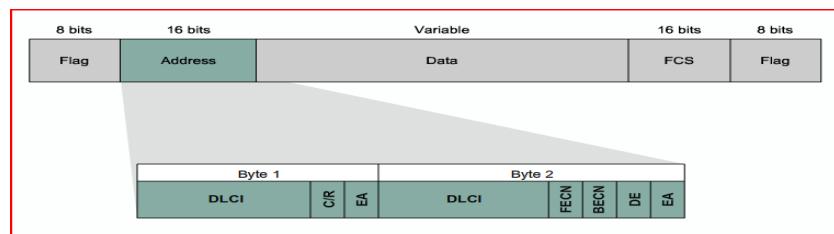
Frame – Relay bağlantıları sanal devreler (VC) kullanır. Bu sanal devre kalıcıdır (PVC) yani hat kullanılmadığı durumda bile up durumundadır. Her bir sanal devrenin OSI II. katmanda tanımlanan bir

DLCI (Data Link Connection Identifier) değeri vardır. DLCI numaraları 10 bit uzunluğundadır ve her Frame – Relay Switchte tekil olmak zorundadır.

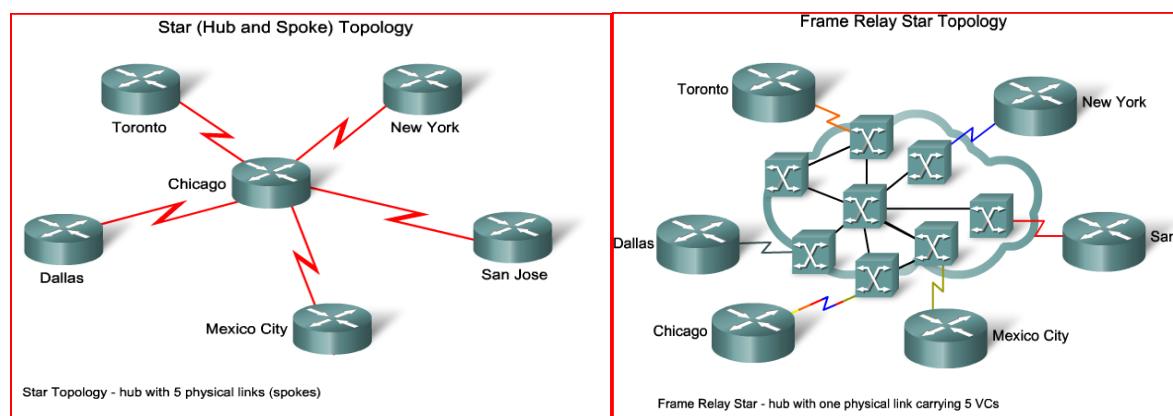
Virtual Circuit oturumunun kurulması için iki yol vardır.

SVC: dinamik olarak oluşturulur, PVC ise ön tanımlı bir yol kullanır.

FRAME – RELAY FRAME FORMATTI



FRAME – RELAY TOPOLOJİLERİ



Router, veri transferi için L2 – L3 eşleşmesini yapmak zorundadır. Bunun için Inverse – ARP yöntemini ya Static MAP yöntemini kullanır.

R# show frame – relay map komutu ile IP – DLCI eşleşmesi öğrenilebilir.

Serial 0/0 (up) : ip 10.1.1.2 dlci 102 (0 x 66, 0x1860), static, broadcast,

Cisco, status defined, active.

Burada, 10.1.1.2 adresi remote address, 102 ise, bu IP ye ulaşmak için kullanılan lokal DLCI numarasıdır.

- Inverse – ARP cisco cihazlarda default olarak açık gelir.

STATIC MAP

R(config – if) # no frame – relay inverse arp

R(config – if) #frame – relay map ip 10.1.1.2 102 <broadcast> <cisco / ietf>

KAVRAMLAR:

Access Rate / Port Speed: Port hızını tanımlar. FR Switch, clock rate belirlediğinden veri transferi bu sınırı aşamaz.

CIR (Committed Information Rate) : Hizmet sağlayıcı tarafından belirlenen maximum veri transfer hızını temsil eder. Her DLCI numarası için bir CIR belirlenir. Frame – Relay paylaşımı bir ortam sağladığından, diğer kullanıcılarla tesis edilen hat kullanılmadığında bu CIR değeri kullanılabilir (**CBIR**). Eğer transfer CIR hızından daha yüksek bir hızla transfer edilirse, bu paketler atılabilir olarak (Discard Eligible, **DE**) işaretlenir. Herhangi bir yoğunluk anında DE işaretli paketler öncelikli olarak atılır. Bazı servis sağlayıcılar, **Zero – CIR** uygular, yani tüm paketler varsayılan DE olarak set edilmiştir.

LMI (Local Management Interface): DTE cihaz ile DCE cihaz arasındaki bir sinyalleşme standartıdır. DTE ile DCE cihaz arasındaki haberleşmenin kurulmasından ve devam ettirilmesinden sorumludur. Cisco cihazlar 3 tür LMI standartını destekler. Bunlar, Cisco, ANSI, Q933a standartlarıdır.

R(config – if) #frame – relay lmi-type <cisco/ansi/q933a> komutu ile belirlenir.

16 ile 991 arasında bir değer alır. Burda bazı değerler rezerve edilmiştir. 0 – LMI, 1023 – Cisco için; 1 – 15 arasındaki değerler ise gelecekte kullanım için rezerve edilmiştir.

FECN (Forward Explicit Congestion Notification): Rotadaki bir tıkanıklık anında, hedef cihaza DCE cihaz tarafından gönderilen bir bilgidir (**indirect**). FECN, FR başlık bilgisinde bir alandır ve bu bit 1 olarak set edilir.

BECN (Backward Explicit Congestion Notification): DCE cihaz tarafından kaynak DTE cihaza gönderilen ağır tıkanıklığı bilgisidir (**direct**). (BECN bitinin set edilmesidir). BECN mesajını alan kaynak cihaz, hızını % 25 oranında düşürür ve DE olarak set edilmiş paketler atılır.

Committed Brust: Normal şartlarda networkün taşıyabileceği maximum bit.

Excess Information Rate : Garanti edilen hız ile max. hız arasındaki faktör.

CONFIGURATION

```
R(config)# interface serial 0/0/0
R(config – if )# ip address 210.0.0.1 255.255.255.0
R(config – if )#encapsulation frame – relay <ietf>
R(config – if )#no shutdown
R(config – if )# frame – relay interface – dlc 100 (100, o VC hatta kullanılan kaynak DLCI numarasıdır)
**R(config – if )# no frame – relay inverse – arp
**R(config – if )# frame – relay map ip 10.16.0.2 110 broadcast (10.16.0.2 = Hedef IP, 110 = Kaynak DLCI) **
```

** Bu satırlar sadece Inverse ARP istenmediği durumda kullanılır.

Frame – Relay, ATM ve X.25 bağlantıları NBMA bağlantılarıdır. Bu bağlantılar, VC üzerinden bir cihazdan sadece bir cihaza veri transferi yapabilirler. Multicast ve Broadcast trafiği desteklemezler.

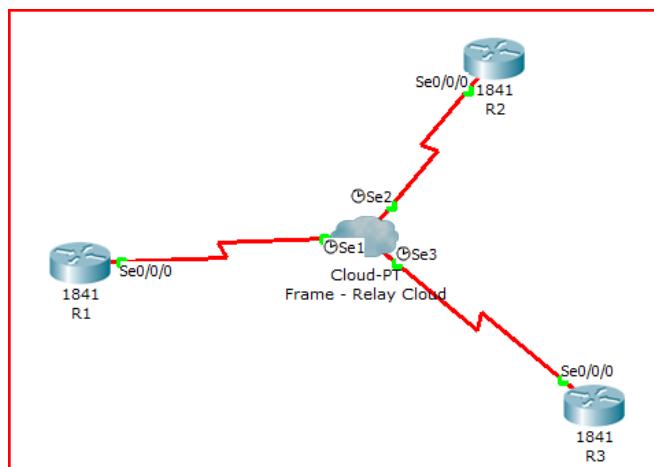
BROADCAST parametresi ile paketler manual olarak çoğaltılmış tüm hedeflere gönderilir.

RIP, EIGRP ve OSPF uygulamalarında bu parametre kullanılmalıdır. Böylece routing updateler yapılabilir.

FRAME – RELAY SUBINTERFACE KULLANIMI

FR'de bir fiziksel interface birden çok sanal devreyi destekleyebilir. Split – Horizontan dolayı, update gönderilen bir interfaceden başka update'ler alınmaz. Bu durumda diğer uçlardan routing update bilgisi alınamaz. Bu sorunu gidermek için bir interface sub-interface'lere ayrılr.

İki tür sub interface vardır. Point – to Point ve MultiPoint. MultiPoint subinterface'ler split horizontan dolayı sorun oluşturulmaması için, Distance Vector Routing Protocoller kullanıldığında split horizon özelliği devre dışı bırakılmalıdır.



R2, R1'e routing update yapar. Bu update bilgisi R1'e serial 0/0/0 arayüzünden ulaşacaktır. R1, gelen bu update bilgisini R3'e ulaştırmak isteyecektir. Ancak split-horizon özelliğinden dolayı, (farklı DLCI kullanılsa bile) R1, R2 ye ait bilgileri R3 ulaştırmayacaktır.

Bu sorunun önüne geçebilmek için, sub-interface kullanılabilir. Ayrıca IP networklerde split-horizon devre dışı bırakılabilir. Ancak, IPX ve Apple Talk ağlarda bu yapılamaz. Burada unutulmaması gereken bir nokta da, split- horizon özelliğinin devre dışı bırakılması, Routing Loop oluşmasına sebebiyet verebilir.

Frame – Relay konfigürasyonunda her PVC için bir subinterface oluşturulur. Bir subinterface, point-to-point ya da multipoint olabilir.

Point – To – Point, her subinterface, bir DLCI numarası ile karşı router arayüzüne bağlanır.

Multipoint, Bir subinterface, karşısında birden çok subinterface'e bağlıdır. Tüm arayüzler aynı subnette olmalıdır. Multipoint subinterface'de static route map kullanmak gerekmeyez. Bu durumda split-horizontan kaynaklanan sorun çözülemez.

ÖRNEK POINT-TO-POINT: Sub Interface yöntemi ile Split Horizon sorununu aşmak,

```
R(config – if )# encapsulation frame – relay  
R(config – if )# no ip address  
R(config – if )#no shutdown  
R(config – if )#exit  
R(config)# interface serial 0/0/0.10 point – to – point  
R(config – subif )# frame – relay interface – dlc1 110  
R(config – subif )#exit
```

```
R(config)# interface serial 0/0/0.20 point – to – point  
R(config – subif)# frame – relay interface – dlci 120
```

TROUBLESHOOTING

R # show frame – relay pvc 110

VC hatların durumları hakkında bilgi verir.

Active – Hattın sorunsuz çalıştığını

Deleted – Hattın silindiğini (bize yakın taraftaki switchte)

Inactive – Hattın bize yakın switchten sonrası ile ilgili bir problem olduğunu gösterir.

R # show frame – relay map

IP eşleşmelerini gösterir. Ayrıca, dinamik, statik ve broadcast bilgilerini de gösterir.

R # clear frame – relay inarp komutu ile InverseArp ile öğrenilenler silinir.

R # show interface serial 0/0/0

LMI Type, LMI DLCI, FR DTE/DCE, Encapsulation bilgilerini verir.

R # show frame – relay lmi paket istatistiklerini gösterir.

R # debug frame – relay lmi

out, gönderilen,

in, alınan paketleri gösterir.

dlci 100, status 0x2 = DLCI 100 aktif.

* INVERSE-ARP isteği olduğunda router map tablosunu aşağıdaki 2 tip LMI durumuna göre belirler.

ACTIVE: DTE – DTE uça bağlantı başarılı. (0x2)

INACTIVE: DTE – SWITCH bağlantısı başarılı ancak SW ötesinden sorun var.(0x0)

DELETED: DTE, DLCI konfigürasyonu yapılmış ancak FR Switch tarafından tanımlanamıyor.(0x4)

BİR CISCO ROUTER, FRAME – RELAY SWITCH GİBİ DAVRANABİLİR.

FRRSW(config)# frame – relay switching

FR switchte, DCE taraf olacağından Clock rate ile hız tanımlanmalıdır. Ayrıca varsayılan olarak routerların portları DTE olarak ayarlandığından, Switch olarak çalışacak router'ın ilgili interface'i DCE olarak belirlenmelidir.

```
FRSW(config)#int s0/0  
FRSW(config-if)#encapsulation frame-relay  
FRSW(config-if)#frame-relay intf-type dce  
  
FRSW(config-if)#encapsulation frame-relay ietf
```

FRSW(config-if)#frame-relay route 201 interface s0/1 202----- 201 Kaynak, 202 hedefdir. S0/1 çıkış arayüzüdür

```
FRSW(config-if)#int s0/1  
FRSW(config-if)#encapsulation frame-relay ietf  
FRSW(config-if)#frame-relay intf-type dce  
FRSW(config-if)#frame-relay route 202 interface s0/0 201
```

```
FRSW(config-if)#no sh
```

IPV6 ÖRNEK

Aşağıdaki komut ile IPv6 unicast routing enable ediyoruz, aksi taktirde ileride yapacağımız dinamik routing (RIP) çalışmaz.

```
RTR1 (config)# ipv6 unicast-routing
```

Maalesef Packet Tracer üzerinde bir Server koyup onu IPv6 DHCP server olarak yapılandırmak mümkün değil. (IPv4 için bu mümkün) O yüzden Router'ları DHCP server olarak yapılandırıyoruz. Aşağıdaki komular bu işi görüyor. Büyük harfler ile yazdığım kısımlar değişken, istedığını yazabilirsin.

DHCP yapılandırmasında 3333:3333:3333:3333::3 adresini DNS olacak şekilde dağıtmasını istedim. Router3' e bir server bağlayıp onu DNS server yapacağız ve IP adresi 3333:3333:3333:3333::3 olacak. Ayrıca domain olarak gazi.edu.tr seçtim. Değiştirebilirsın.

```
RTR1 (config)#ipv6 dhcp pool TEST
RTR1 (config-dhcp)#prefix-delegation pool DENEME
RTR1 (config-dhcp)# dns-server 3333:3333:3333:3333::3
RTR1 (config-dhcp)# domain-name gazi.edu.tr
RTR1 (config-dhcp)#exit
RTR1 (config)#ipv6 local pool PREFIX 1111:1111:1111:1111::/48 64
```

Böylece 1111:1111:1111:1111.... ile başlayıp geri kalanını MAC adresten türeten bir IPv6 adresini cihazlara dağıtabiliyor olacağız. Tabi bizim cihazımızın da FastEthernet 0/0 arayüzünün de IPv6 adresi bu belirttiğim seri ile başlaması gereklidir. Aşağıda Fa0/0 arayüzüne bu IP adresini veriyoruz.

```
RTR1 (config)#interface fastethernet 0/0
RTR1 (config-if)#ipv6 address 1111:1111:1111:1111::1/64
RTR1 (config-if)#no shutdown
```

Bu adres ağdaki diğer cihazlar için Gateway olacağından dolayı IPv6 adresinin kolay olması gerekiyor. O yüzden son kısmını :1 yaptım. İlk adresi verdim diyebiliriz. AutoConfig veya benzeri yaparsak, geri kalanını yine MAC adresten türetecek ve bu yüzden okunması zor olacak. ☺

```
RTR1 (config-if)#ipv6 dhcp pool TEST
```

komutu ile cihazımızın IPv6 DHCP server olması gerektiğini belirtiyoruz.

Şimdi de RTR1 ile RTR2 arasındaki serial bağlantıyı yapılandırıyoruz. Bu bağlantı Serial0/0/0 arayüzüne bağlı. 1. Router ile 2.Router arasında olduğu için akılda kolay yerleşin diye 1212:1212:1212:1212::1 adresini verdim.

```
RTR1 (config)#interface serial0/0/0
RTR1 (config-if)#ipv6 address 1212:1212:1212:1212::1/64
RTR1 (config-if)#clock rate 64000
RTR1 (config-if)#no shutdown
```

Şimdi RTR1 ile RTR3 arasını yapılandıracagız ve bu sebeple 1313... şeklinde IP vermem. Bu da Serial 0/0/1 arayüzüne bağlı. (Clock Rate farklı olabildiğine dikkat)

```
RTR1 (config)#interface serial0/0/1
RTR1 (config-if)#ipv6 address 1313:1313:1313:1313::1/64
RTR1 (config-if)#clock rate 128000
RTR1 (config-if)#no shutdown
```

Aynı mantıkla RTR2'yi yapılandırıyoruz

```
RTR2(config)#ipv6 dhcp pool POOL2
RTR2(config-dhcp)#prefix-delegation pool POOL2-PREFIX
RTR2(config-dhcp)# dns-server 3333:3333:3333:3333::3
RTR2(config-dhcp)# domain-name gazi.edu.tr
RTR2(config-dhcp)#exit
RTR2(config)#ipv6 local pool PREFIX 2222:2222:2222:2222::/48 64
RTR2(config)#interface fastethernet 0/0
RTR2(config-if)#ipv6 address 2222:2222:2222:2222::1/64
RTR2(config-if)#no shutdown
```

RTR2 ile RTR1 'in birbirine bağlı olan portları aynı IP aralığında olması gerekiyor. RTR1 için 1212.... şeklinde bir adres vermiştık. O yüzden buna da aynı aralıktan IPv6 adresini vermeliyiz. 1212:1212:1212:1212::2 şeklinde bir IP verelim. Burada clock-rate vermeye gerek yok.

```
RTR2(config)#interface serial0/0/0
RTR2(config-if)#ipv6 address 1212:1212:1212:1212::2/64
RTR2(config-if)#no shutdown
```

RTR2 ile RTR3 arasını yapılandırıralım. 2 ile 3 arasında olduğu için 2323:2323... şeklinde bir IP verelim.

```
RTR2(config)#interface se0/0/1
RTR2(config-if)#clock rate 2000000
RTR2(config-if)#no shutdown
RTR2(config-if)#ipv6 address 2323:2323:2323:2323::2/64
```

Router3 yapılandırması;

```
Router(config)#hostname RTR3
RTR3(config)#ipv6 unicast-routing
RTR3(config)#interface fastEthernet 0/0
RTR3(config-if)#no shutdown
RTR3(config-if)#ipv6 address 3333:3333:3333:3333::1/64
```

```
RTR3(config)#ipv6 dhcp pool POOL3
RTR3(config-dhcp)#prefix-delegation pool POOL3-PREFIX
RTR3(config-dhcp)#domain-name gazi.edu.tr
RTR3(config-dhcp)#dns-server 3333:3333:3333:3333::3
RTR3(config-dhcp)#exit
RTR3(config)#ipv6 local pool POOL3 3333:3333:3333:3333::/48 64
```

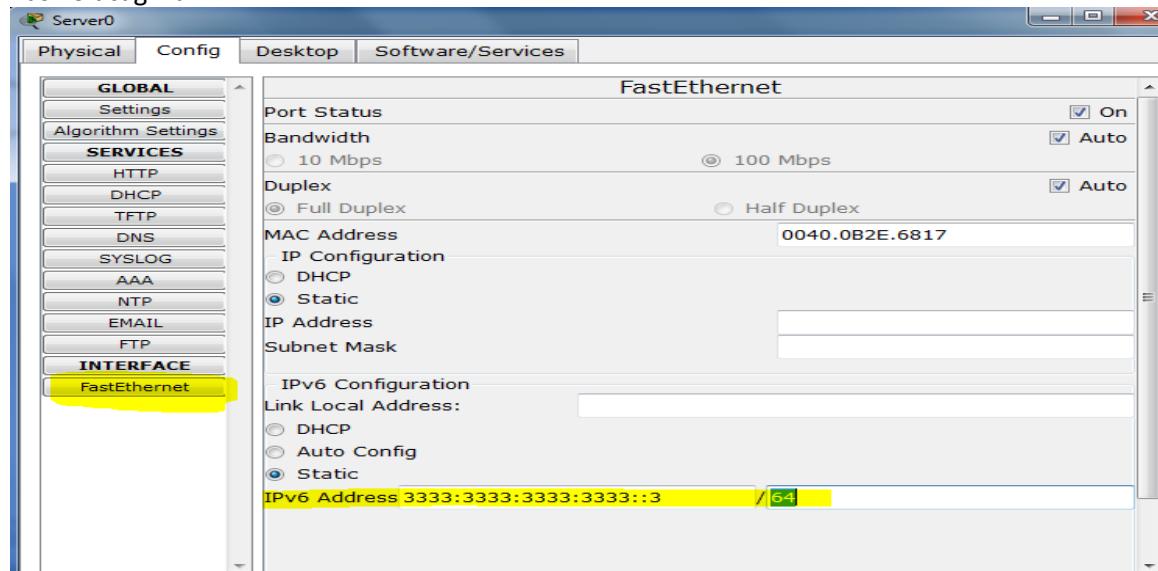
```
RTR3(config)#interf se0/0/0
RTR3(config-if)#ipv6 address 1313:1313:1313:1313::3/64
RTR3(config-if)#no shutdown
```

```
RTR3(config-if)#interf se0/0/1
RTR3(config-if)#no shutdown
RTR3(config-if)#ipv6 address 2323:2323:2323:2323::3/64
RTR3(config-if)#exit
RTR3(config)#interf fa0/0
RTR3(config-if)#ipv6 dhcp server POOL3
```

Artık bu adımdan sonra PC'ler DHCP'den otomatik IP alabilirler. PC'lerdeki yapılandırmada DHCP seçmelişin, Auto-Config değil.

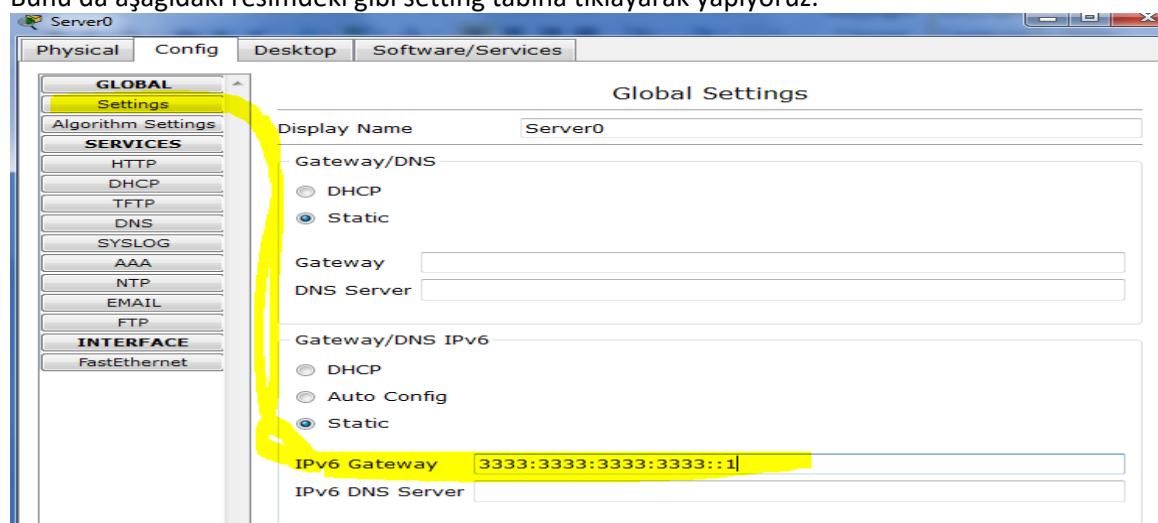
Şimdi sıra 2 tane Server bağlamaya geldi. Server'lerden biri en başta mantığını kurduğumuz ve IP adresini 3333:3333:3333:3333::3 olarak tasarladığımız DNS server, diğeri de herhangi bir yere koyacağımız http server.

Önce DNS olanı ekleyelim. RTR3'e bağlı olması gereklidir. Server'lara IP adresi elle verilir. Aşağıdaki resimde nasıl olacağı var.

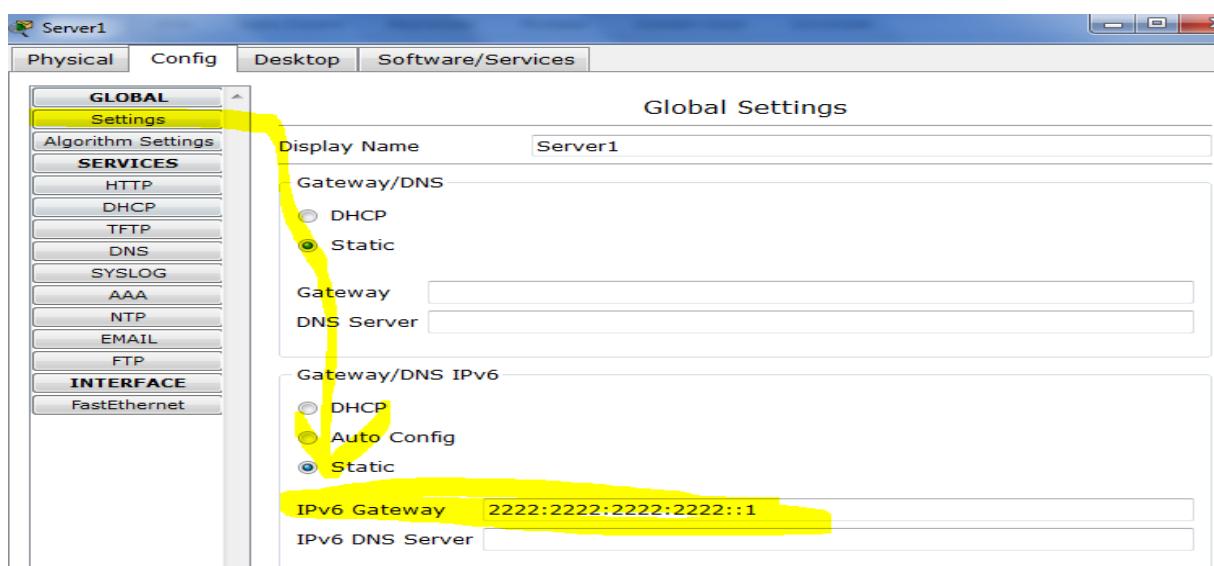
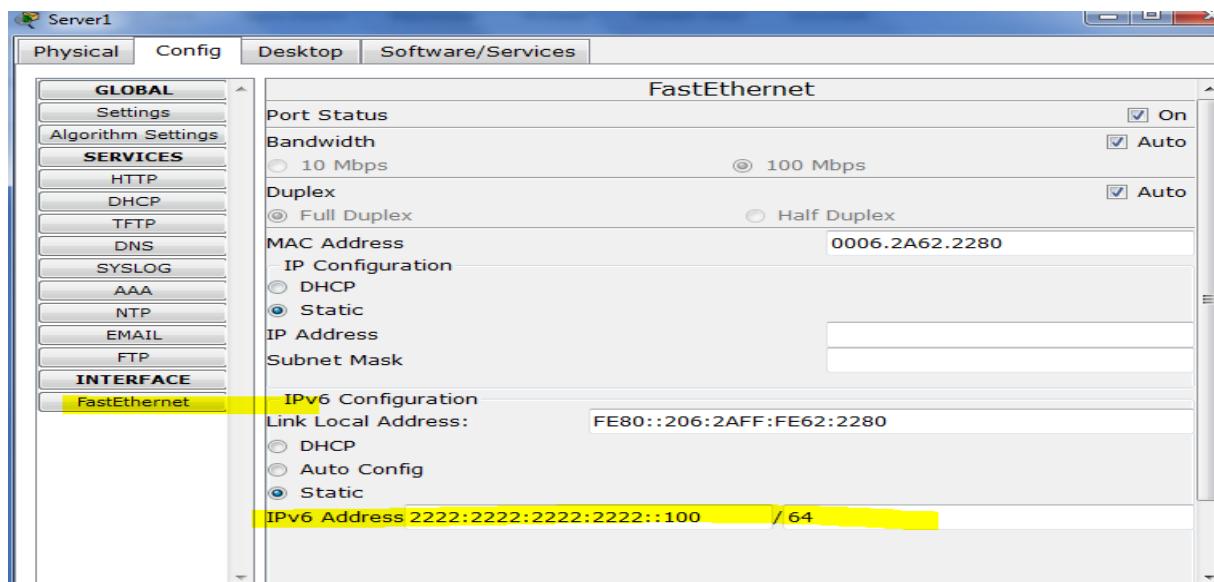


Ayrıca bu cihaza Gateway 'de tanımlamalıyız. FastEthernet 0/0'ın IP adresi bizim gateway yani: 3333:3333:3333:3333::1

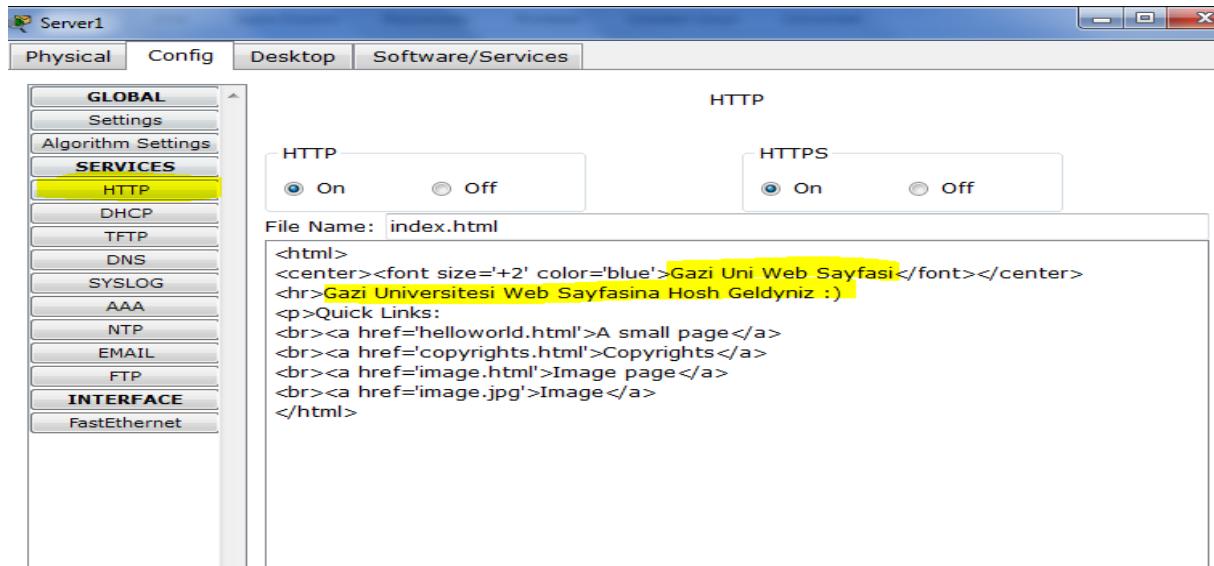
Bunu da aşağıdaki resimdeki gibi setting tabına tıklayarak yapıyoruz.



İkinci server'imizi de örneğin RTR2'ye bağlı switche bağlayalım. Bu yüzden IP si 2222... ile başlar. Ben 2222:2222:2222:2222::100 olsun diyorum. Yine server olduğu için static olarak elle veriyoruz bu IP yi. Aynı şekilde Setting kısmından gateway olarak da 2222:2222:2222:2222::1 veriyoruz.



Şimdi de http server olarak tasarladığımız bu cihazda web hizmeti açalım. Aslında default açık, yapmamız gereken html kodlarında ufak değişiklik yapmak. Aşağıdaki resimde nasıl olacağı var.



Artık bu Server'da http açık.

DİNAMİK YÖNLENDİRME

Routerlar arasında iletişim şimdilik yok. Karşılıklı olarak ping atabilirler ancak farklı routelara bağlı PC'ler birbirlerine ping atamazlar. Bu yüzden yönlendirme yapmak gereklidir. Bunun birkaç yolu var. Uzun yolu static yönlendirme yapmak. Ancak senin gönderdiğin uygulamada dinamik RIP yapılandırması yaptığı için ben de aynısını yapıyorum.

Her cihazda yapılandırma yapmak gerek CLI modda.

RTR1 YAPILANDIRMA

```
RTR1 (config)#interface fastEthernet 0/0
RTR1 (config-if)#ipv6 rip RTR1 enable
RTR1 (config-if)#int ser0/0/0
RTR1 (config-if)#ipv6 rip RTR1 enable
RTR1 (config-if)#int ser0/0/1
RTR1 (config-if)#ipv6 rip RTR1 enable
```

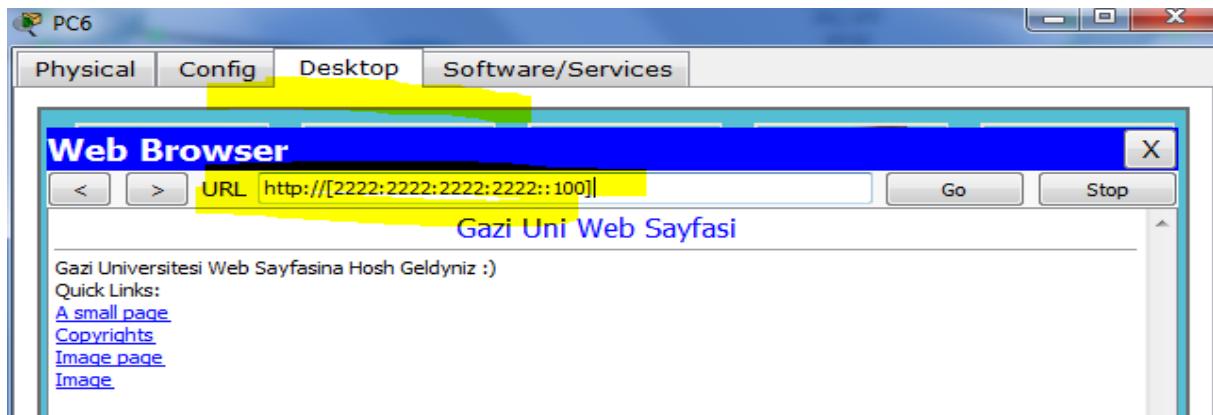
RTR2 YAPILANDIRMA

```
RTR2 (config)#interface fastEthernet 0/0
RTR2 (config-if)#ipv6 rip RTR2 enable
RTR2 (config-if)#interface s0/0/0
RTR2 (config-if)#ipv6 rip RTR2 enable
RTR2 (config-if)#interface s0/0/1
RTR2 (config-if)#ipv6 rip RTR2 enable
```

RTR3 YAPILANDIRMA

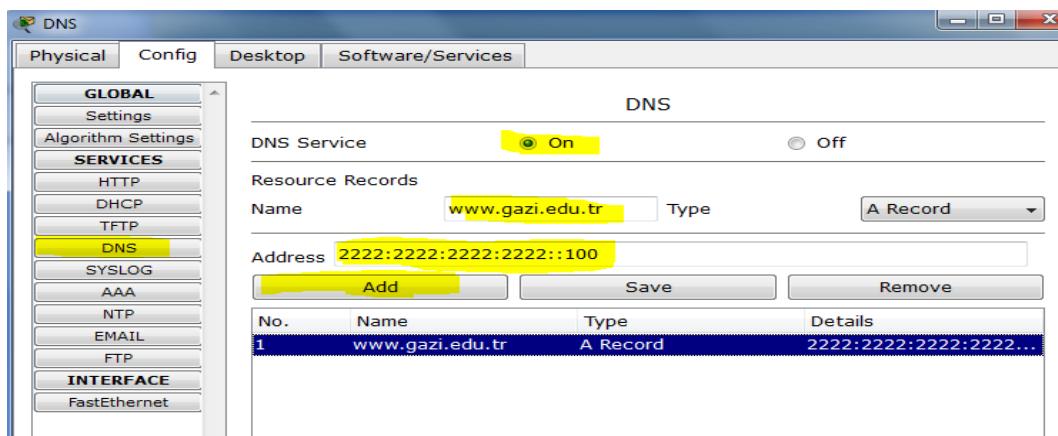
```
RTR3 (config)#interface fastEthernet 0/0
RTR3 (config-if)#ipv6 rip RTR2 enable
RTR3 (config-if)#interface s0/0/0
RTR3 (config-if)#ipv6 rip RTR2 enable
RTR3 (config-if)#interface s0/0/1
RTR3 (config-if)#ipv6 rip RTR2 enable
```

Bu durumda artık PC'ler birbirlerine ping atabilirler. Ayrıca herhangi bir PC'den http server'a IPv6 adresi üzerinden bağlanalım.



Burada dikkat edilmesi gereken nokta şu, IPv4 adreslerinde IP 'yi direkt tarayıcıya yazabiliyorduk. Ancak IPv6 da durum farklı. Adresi köşeli parantez içinde yazmak gerekiyor. Aksi durumda cihaz port bilgisi ile karıştırılabilir. Yukarıdaki örnek'te buna dikkat.

Tabi cihazlara IPv6 adresi üzerinden erişmek hoş değil, zaten akılda kalıcı da değil. Onun yerine bu servera bir isim verelim. Örneğin www.gazi.edu.tr olsun. Dolayısıyla Bizim DNS'te bunu kaydetmemiz gerek (Add). Aşağıdaki resimde DNS server'da bunun nasıl yapıldığı var.



Bundan sonra artık PC'ler www.gazi.edu.tr yazdıklarıında sayfa gelir.

