

2020

CCNA EĞİTİMİ DERS NOTLARI

DR.ERDAL ÖZDOĞAN

TEMEL İLETİŞİM

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 gerektiği 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 standardizasyon sağlamak 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

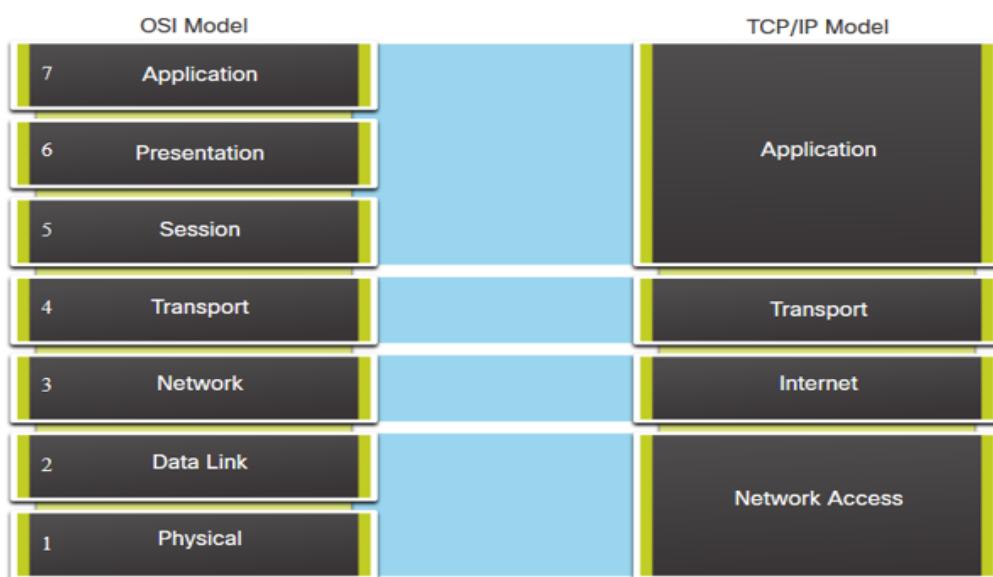
L4: Transport

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 kaçıdaki 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.

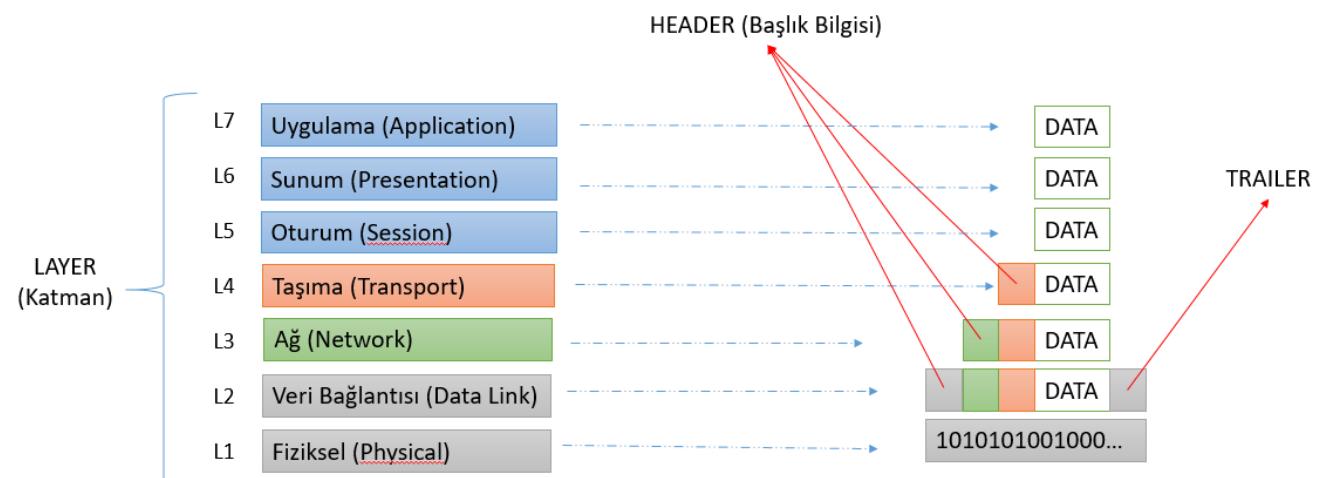
Transport Layer, üst katmanlardan gelen verilerin alt katmanları ile iletişimini sağlayan katmandır. Diğer bakış açısından, alt katmanlardan gelen verinin, üst katmanlarda hangi uygulama ya da servis ile iletişime geçeceğini belirlendiği katmandır. Bu belirleme PORT numaraları ile sağlanır.

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

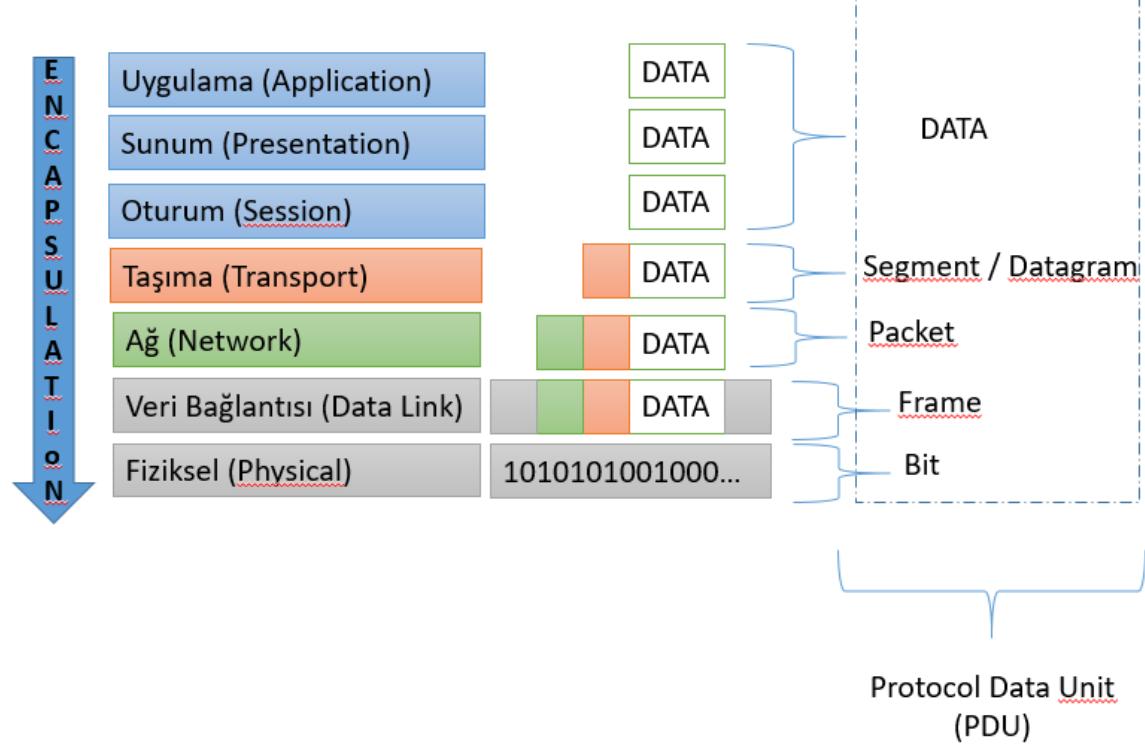
Data – Link Layer, fiziksel adreslemelerin yapıldığı yerdir. Local ağdaki, gönderici ve alıcı cihazların fiziksel adreslerinin belirlendiği alandır. Burada Fiziksel Adres olarak MAC adresleri kullanılır.

Physical Layer, verilerin taşınacak ortama (bakır kablo, fiberoptik kablo, hava gibi) aktarılmak için bitlere dönüştürüldüğü, elektriksel kodlamanın yapıldığı katmandır.

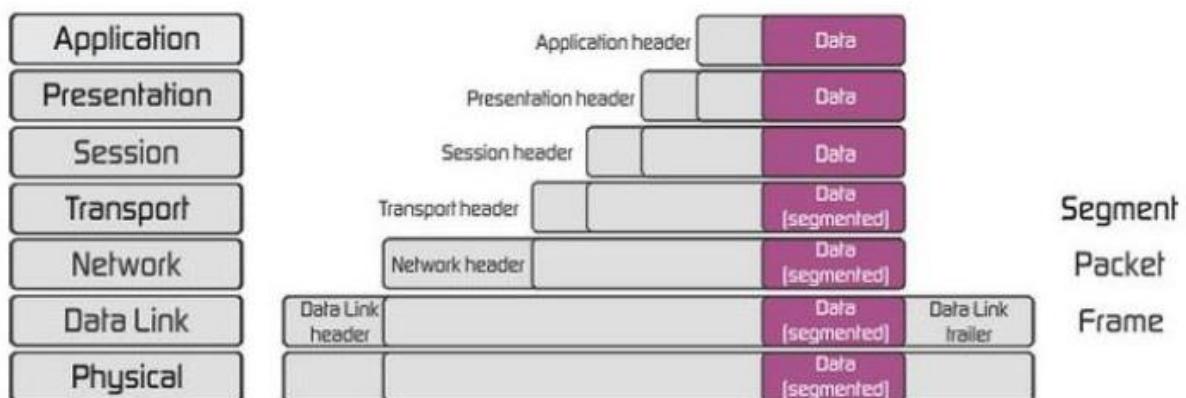
İsimlendirmede kolaylık olsun diye tüm katmanlara aşağıdan yukarıya olmak üzere numaralar verilmiştir. Buna göre örneğin Physical Layer : **Layer 1 (L1)**, Application Layer : **Layer7(L7)** olarak adlandırılmıştır. Aradaki katmanlar da sırasıyla L2, L3 ... olarak adlandırılırlar.



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 aktarılınca 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 **decapsulation** 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 olur.

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.

Version	IHL	Type of Service	Total Length					
Identification		Flags	Fragment Offset					
Time to Live	Protocol		Header Checksum					
Source Address								
Destination Address								
Options			Padding					

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 dâhil 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.

FİZİKSEL ve MANTIKSAL ADRESLEME

OSI Modelinde 2. ve 3. katmanda verilere eklenen adres bilgileri, sırasıyla Fiziksel Adres ve Mantıksal Adrestir. Veri iletişimini sırasında her iki adresin de kullanılması gereklidir. Fiziksel adres genellikle değişmeyen adres iken, mantıksal adres cihazın bulunduğu konuma göre değişkenlik gösterebilen adreslerdir. Benzetme yapılacak olursa, gönderilen bir posta üzerinde yer alan iki adres bilgisinden, Alıcının Adı Soyadı bilgisi Fiziksel adresi; alıcının ev/iş adresi ise mantıksal adresi temsil eder.



Fiziksel Adres Ethernet ağlarında, cihazlarının ağ arayüz kartlarına (NIC) atanan MAC adres olarak ifade edilen 48-bit uzunluğunda bir adresidir. Mantıksal adres ise yine günümüzde IP adresleridir (v4 ve v6).

```
C:\>ipconfig /all  
Windows IP Configuration  
  
Ethernet adapter Ethernet 2:  
  
    Media State . . . . . : Media disconnected  
    Connection-specific DNS Suffix . :  
    Description . . . . . : Intel(R) Ethernet Connection I219-V  
    Physical Address. . . . . : 1C-39-47-A3-ED-EF  
    DHCP Enabled. . . . . : Yes  
    Autoconfiguration Enabled . . . . : Yes
```

Windows işletim sistemlerinde MAC adresi (Fiziksel Adres) görmek için ipconfig/all komutu kullanılabilir. Örnek bir MAC adresi aşağıdaki gibidir.

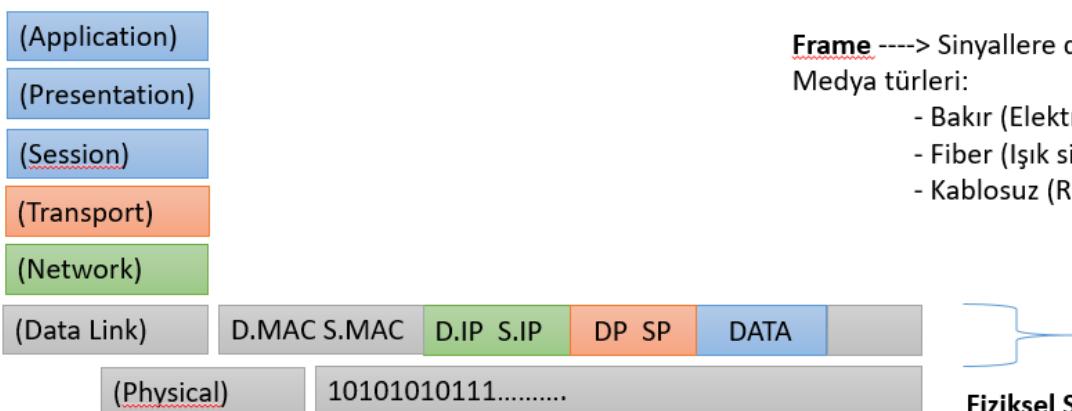
- FB:19:07:01:AA:21 veya 1C-39-47-A3-ED-EF gibi

Daha önce söz edildiği gibi, MAC adres bilgisi 48-bir uzunluğunda bir adresidir ve genellikle onaltılik sayılar ile temsil edilir.

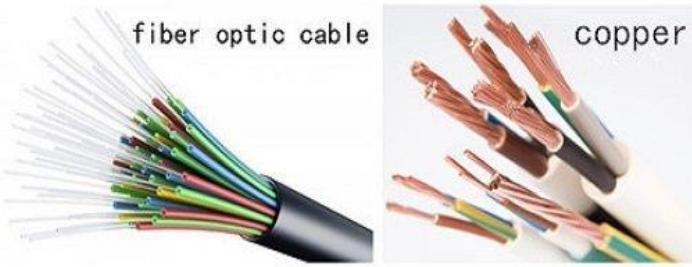
1C-39-47-A3-ED-EF

Bu adres, iki kısımdan oluşur. İlk 24-bit olan kısmı, OUI olarak ifade edilen üretici kodu, sonraki 24-bit ise üreticinin atadığı numaradır

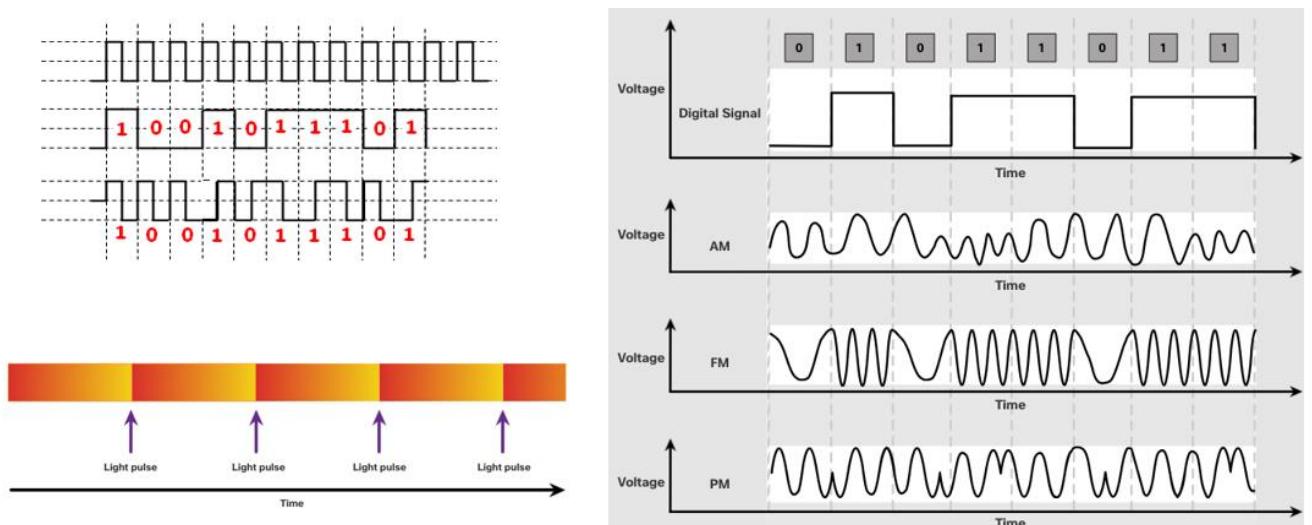
FİZİKSEL KATMAN



Data Link katmanına kadar üretilen bitler, veriyi ve bu verinin doğru adrese, doğru uygulamaya doğru şekilde transferi için eklenen bilgiler (header ve trailer) bir sıfırlar dizisini oluşturur. Daha sonra bu bitler, fiziksel ortamdan (medya) transfer edilmesi gereklidir. Fiziksel ortam, bakır, fiber ya da kablosuz ortam olabilir.

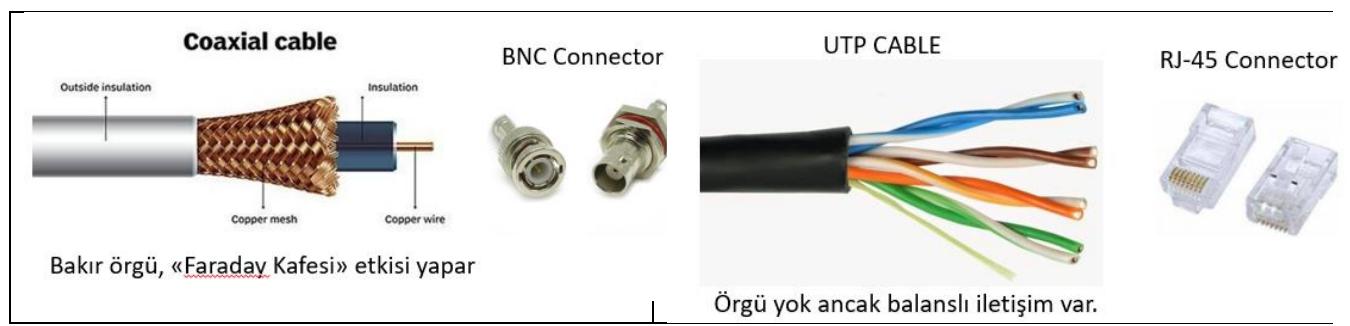


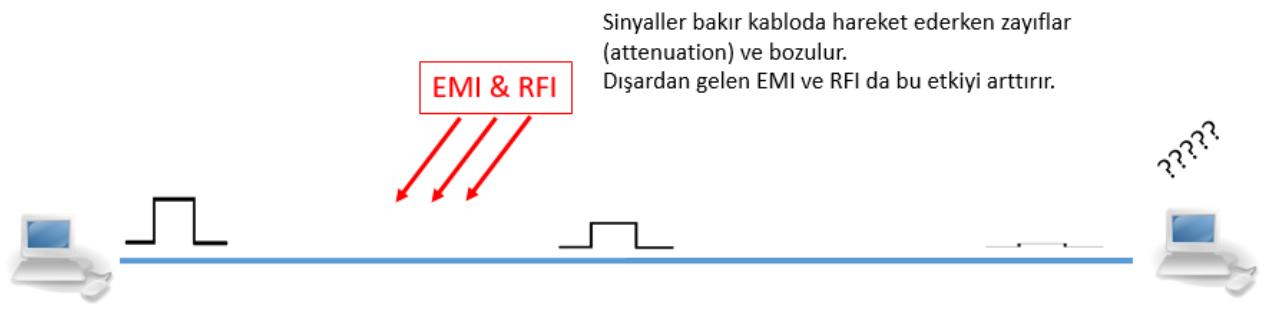
Verilerin, daha açık bir ifade ile bitlerin medya üzerinden transfer edilebilmesi için, 1 ve 0 ifadelerinin, kullanılan medya türüne göre gösterimi gereklidir. Örneğin, bakır medyadan veri transfer edilecekse, gerilimin olması “1” gerilimin olmaması “0” olarak göstermek bu yöntemlerden (ilkel ve kullanılmayan) biridir ve **kodlama** olarak ifade edilir. Elektrik üzerinden verilerin transferinde daha gelişmiş kodlama teknikleri kullanılmalıdır. Manchester, NRZI gibi kodlama teknikleri, yine eski olmasına karşın, anlaşılabilirlik amacıyla aşağıda bu örneklerden bazıları gösterilmiştir.



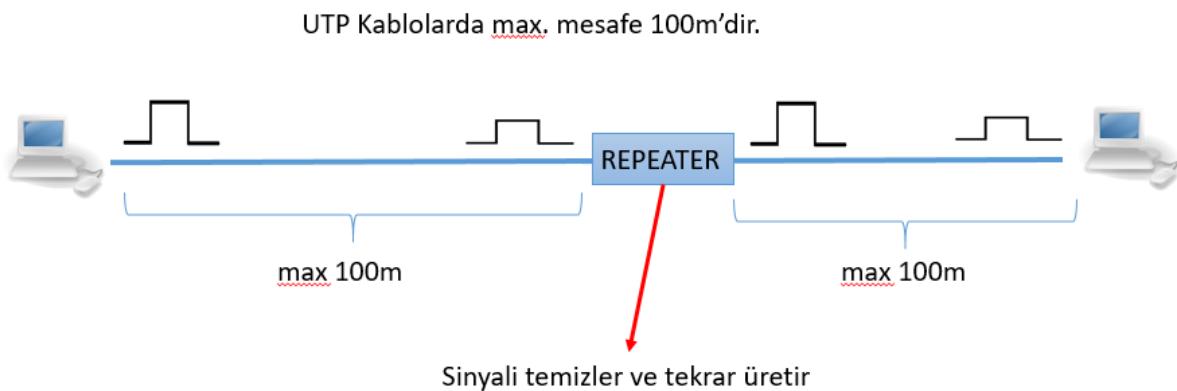
Günümüzde daha çok 4B/5B, 8B/10B gibi kodlama teknikleri daha yaygındır, ancak bu kursun kapsamı dışında olduğu için detaylarına girilmeyecektir. Özette ifade etmek gerekirse, bakır kablolarlarda elektriksel sinyaller, fiber ortamlarda ışık sinyalleri (laser veya LED), kablosuz ortamlarda ise Radyo Frekansları (RF) kullanılmaktadır.

Bakır Kablolar

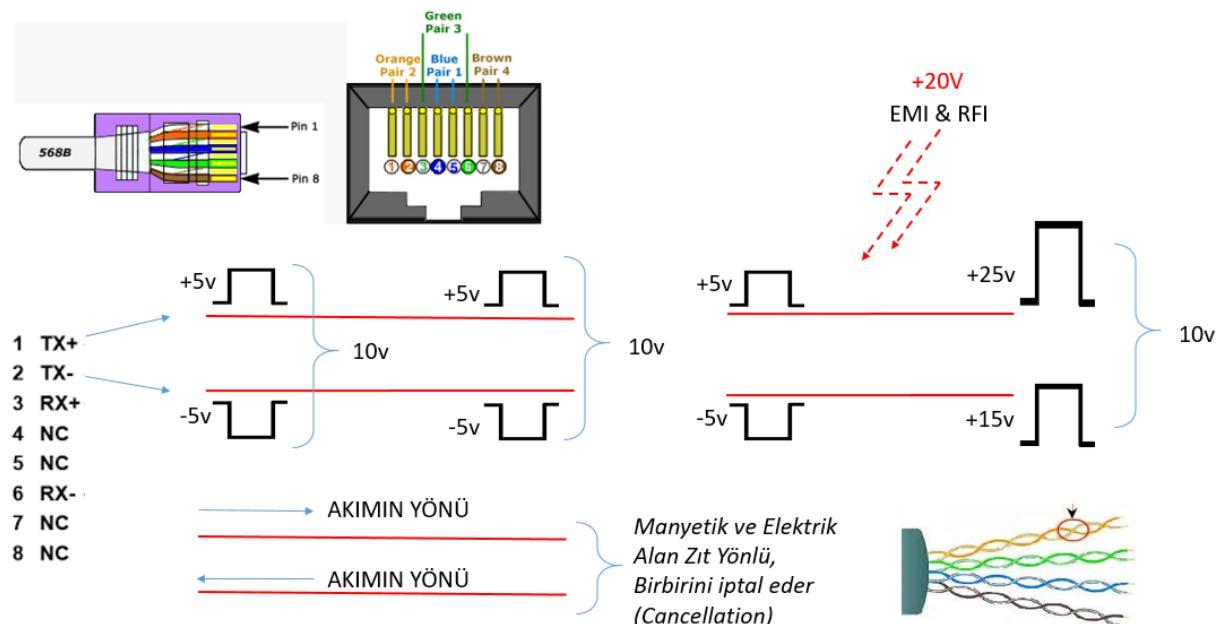


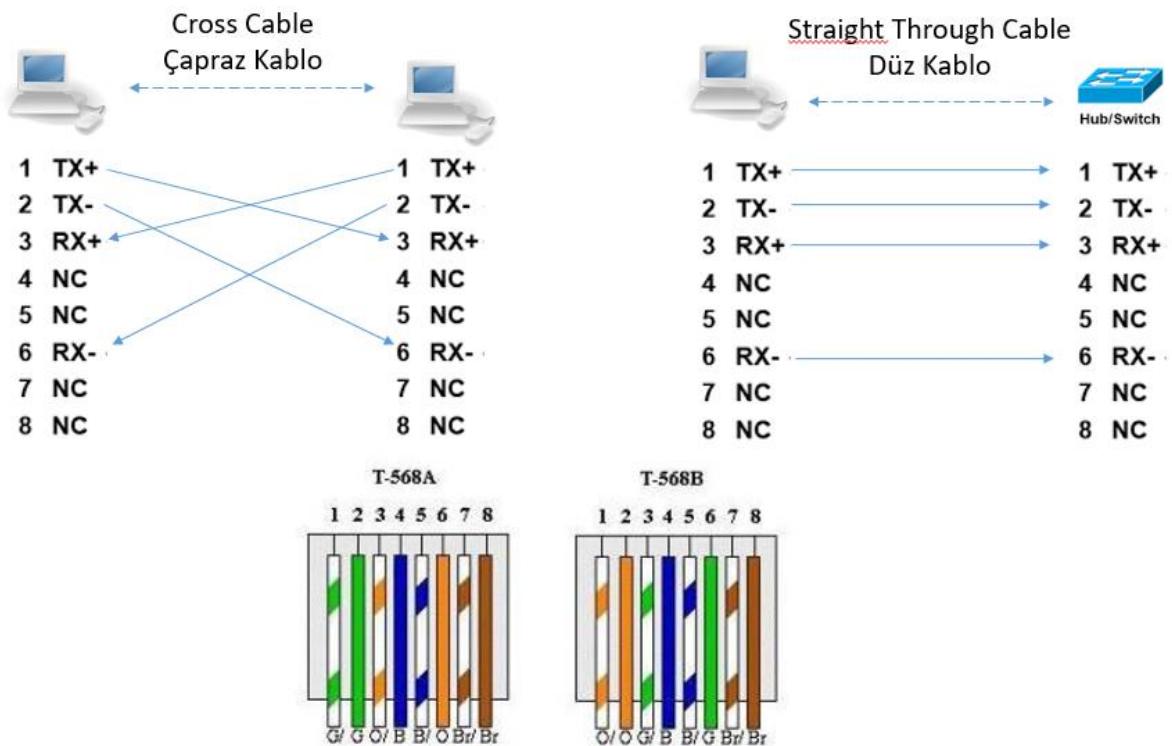


UTP Kablolarda max. mesafe 100m'dir.



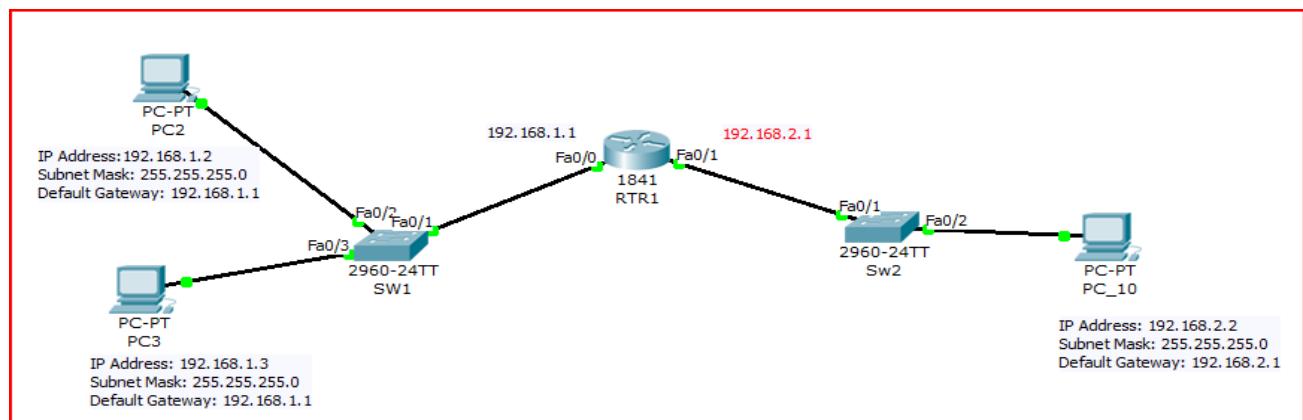
Sinyali temizler ve tekrar üretir





Host-Host İletişim Modeline Giriş

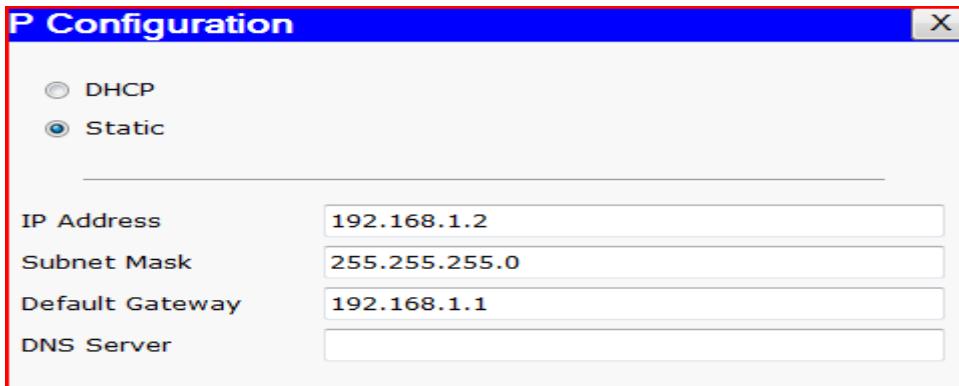
Şimdiye kadar öğrendiğimiz tüm bilgiler ışığında iki bilgisayar arasındaki iletişimini adım adım açıklamaya çalışalım



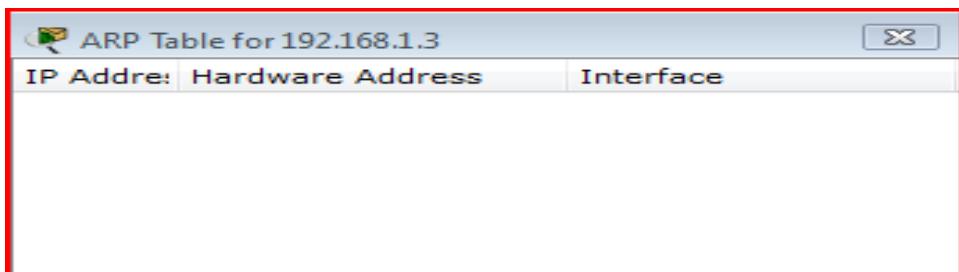
Tüm cihazların yeni açıldığını varsayıyalım. Router, kendisine bağlı olan networkler hakkında bilgi sahibidir. Bu durumda Router için routing tablosu:

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.



Switchlerin de MAC tablosu boştur.



2.1 AYNI AĞDA İLETİŞİM

PC2, aynı ağıda bulunan PC3 ile iletişimde 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.

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

Kendi IP'si ile Subnet Mask AND leyecektir.

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

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

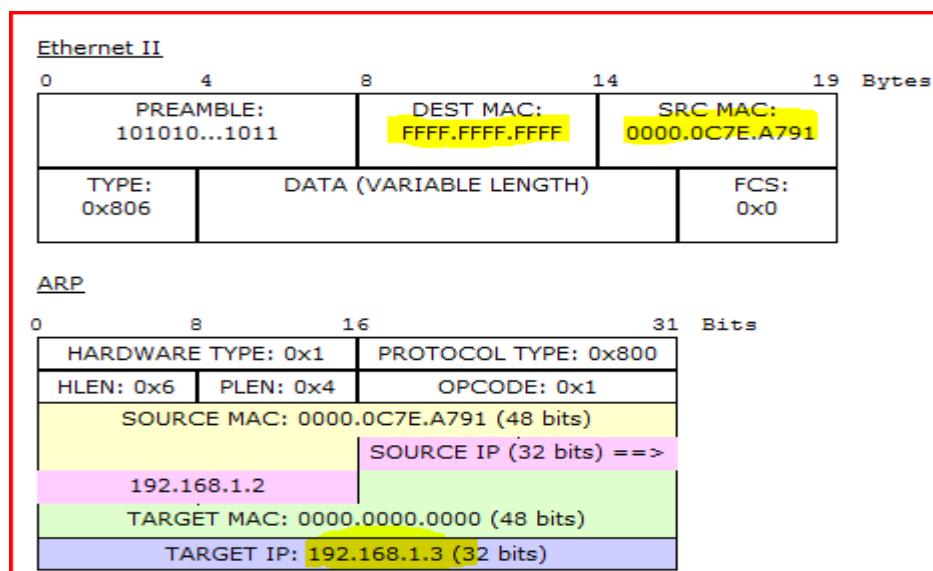
Her iki cihaz aynı ağıda olduğuna göre oluşturulacak çerçeveyi Switch aracılığıyla hedefe ulaşılabilir

olduğu anlaşılır. 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 eşleştirecektir. Bu sayede MAC tablosunda aşağıdaki gibi bir kayıt oluşacaktır.

VLAN	Mac Address	Port
1	0000.0C7E.A791	FastEthernet0/2

Switch gelen ARP çerçevesinin hedef MAC adresine bakıp **FFFF.FFFF.FFFF** (broadcast) bunu 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.

IP Address	Hardware Address	Interface
192.168.1.2	0000.0C7E.A791	FastEthernet

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

Not: Çerçeveler alındıktan sonra öncelikle FCS eşleştirmesi yapılır. Eşleşme hatalı ise Hedef MAC adresе 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

Ethernet II	
0	4
PREAMBLE: 101010...1011	8
DEST MAC: 0000.0C7E.A791	14
SRC MAC: 000D.BD96.831D	19
TYPE: 0x806	DATA (VARIABLE LENGTH)
	FCS: 0x0

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

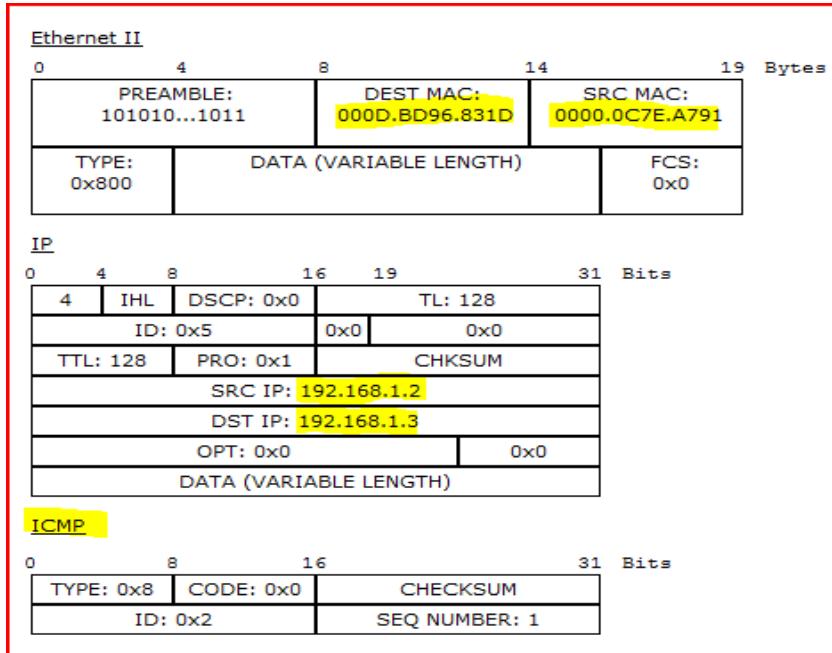
VLAN	Mac Address	Port
1	0000.0C7E.A791	FastEthernet0/2
1	000D.BD96.831D	FastEthernet0/3

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 çıkarak 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.



ICMP paketi Switch'e ulaşacaktır. Switch öncelikle gelen çerçeveyin 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 çıkararak switch'in bunu FastEthernet 0/3 portuna yönlendireceğini görebiliriz.

Bu sayede ICMP paketi PC3'e ulaşacaktır. PC3 gelen ICMP isteğin 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 paketi ARP istekleri sebebiyle cevapsız kalabilir, ancak diğer paketlere cevap gelecektir.

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:					
Ağ Hedefi	Ağ Maskesi	Ağ 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	

1. 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 ağda olmayan veya internetteki herhangi bir IP ye ulaşmak için gateway adresi ile iletişime geçilmesi gerektiğini,

2. 3. ve 4. Satırlarda, 127.0.0.0 (127 ile başlayan tüm IP lerin) PCnin kendisine (Onlink) yönlendirileceğini,

5.Satırda, APIPA adresi sayesinde kendi ağimdaki APIPA'dan IP almış cihazlara Ethernet Kartım ile iletişim kurabileceğimi,

6.Satırda, APIPA Broadcast isteklerin Ethernet kartına yönlendirileceğini,

7.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.

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 anladıkten sonra oluşturacağı çerçeveyen 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	000D.BD96.831D	FastEthernet

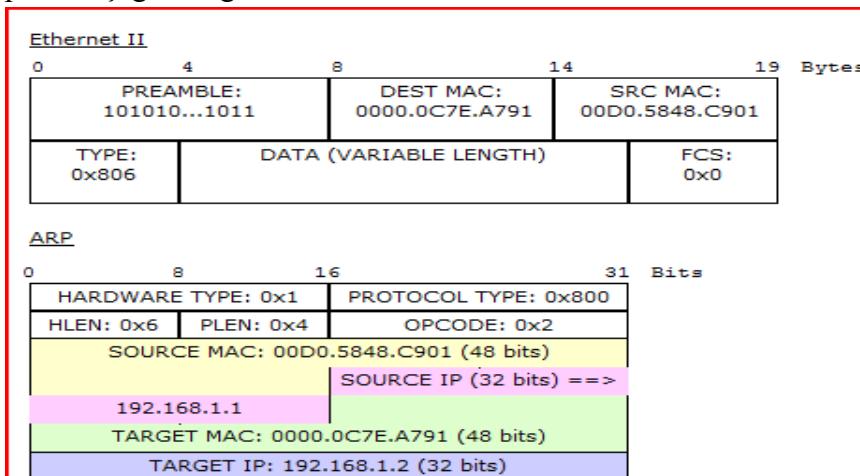
Bu yüzden PC2, gateway adresine (192.168.1.1) karşılık gelen MAC bulmak için ARP Request yayınılayacaktır.

Bu yayın Switchten geçecektir. Switch, gelen porttaki Kaynak MAC ile port eşleştirme 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 bilecektir. Switch MAC tablosu aşağıdaki gibidir.

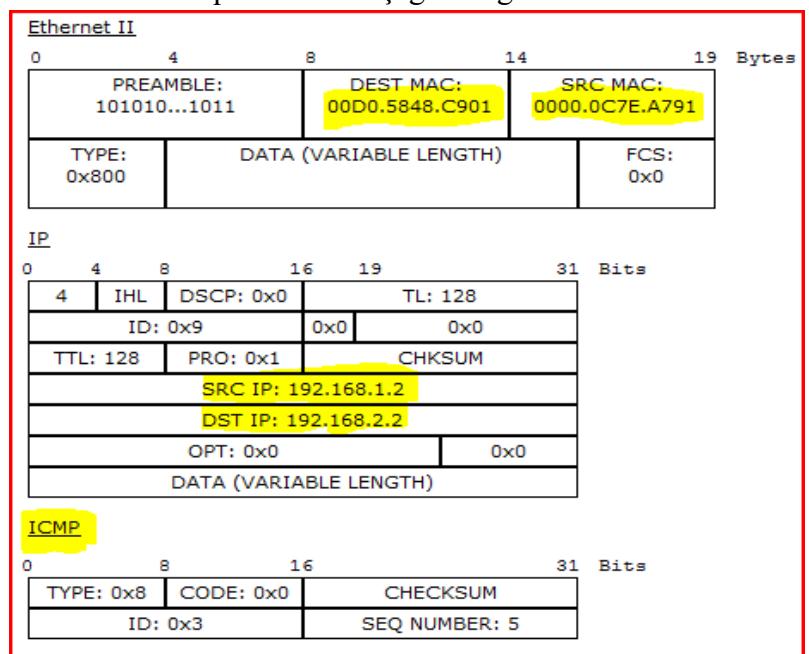
VLAN	Mac Address	Port
1	0000.0C7E.A791	FastEthernet0/2
1	000D.BD96.831D	FastEthernet0/3
1	00D0.5848.C901	FastEthernet0/1

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



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 eşleştirmeini kontrol ettikten sonra, hedef MAC adrese 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 ağının FastEthernet 0/1 arayüzünden çıkış yapması gerektiğini anlayacak ve bu porta yönlendirecek ve IP başlık bilgisindeki TTL değerini bir azaltacaktır. 192.168.2.0 ağının, directly connected (C) bir ağdır.

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

Bu yüzden router gelen çerçeveyi aşağıdaki gibi değiştirecektir.

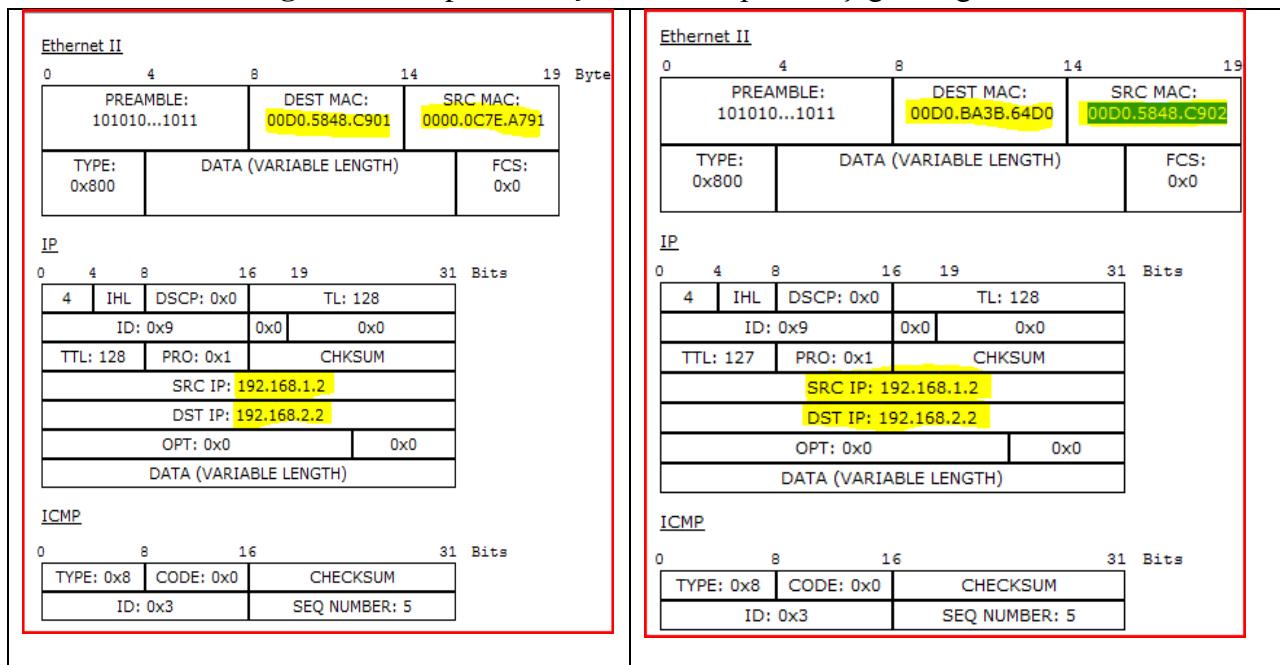
Hedef IP : 192.168.2.2 (Değişmez)

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 değiştiğine 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'ın 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)

• Bölüm 3: Cisco IOS Yazılımını Çalıştırma

TEMEL ROUTER YAPILANDIRMASI

Router>enable

Ayrıcalıklı Moda Geçmemizi sağlar. (Privileged Mode)

Router#configuration terminal

Konfigürasyon moduna geçmemizi sağlar.

Router(config)#hostname

Router'a isim tanımlamamızı sağlar.

Router(Config)#enable password

Ayrıcalıklı moda geçiş için şifre tanımlamamızı sağlar. Tanımladığımız şifre çalışan konfigürasyonda açık bir şekilde görülür.

Router(Config)#enable secret

Ayrıcalıklı moda geçiş için şifre tanımlamamızı sağlar. Tanımladığımız şifre çalışan konfigürasyonda MD5 algoritması ile özetlenmiş halde gözükmür.

```
Router(Config)#banner [motd | login] *
```

Router'a giriş yapıldığında kullanıcılarla mesaj gösterir. Motd (Günün mesajı) veya Login (Giriş Mesajı) seçenekleri mevcuttur. İkisinden biri seçilmelidir. Eğer ayrı ayrı iki mesaj da yapılandırılırsa önce Motd mesajı görüntülenir. Mesaj herhangi bir karakterle başlatılıp yine aynı karakterle bitirilmelidir.

```
Router(Config)#no ip-domain lookup
```

Router komut satırında yazdıklarımızı tanımlayamazsa (Kullanıcı Modunda) bunun host ismi olduğunu düşünür ve ip adresini öğrenmek için DNS sorgulaması yapar. Bu komut ile bu durumlarda DNS sorgulama yapması engellenir.

```
Router(Config)#line vty 0 15  
Router(Config-line)#login  
Router(Config-line)#password .....  
Router(Config-line)#logging synchronous
```

0 15 arası 16 tane sanal telnet bağlantıları için yapılandırma moduna giriş yapar. Login komutu ile telnet bağlantıları etkinleştirilir. Password komutu ile telnet bağlantılarına şifre atar. Logging Synchronous komut yazımı sırasında Router tarafından oluşturulan mesajların yazıyı bölmeyi engeller.

```
Router(Config)#line con 0  
Router(Config-line)#login  
Router(Config-line)#password .....  
Router(Config-line)#logging synchronous
```

Konsol bağlantısı için yapılandırma moduna giriş yapar. Login komutu ile bağlantı etkinleştirilir. Password komutu ile şifre atar. Logging Synchronous komut yazımı sırasında Router tarafından oluşturulan mesajların yazıyı bölmeyi engeller.

```
Router(Config)#service password-encryption
```

Router üzerindeki tüm atanmış parolaları şifreler. Burada kullanılan algoritma MD5 değildir ve MD5 algoritmasından daha zayıftır.

```
Router(Config)#interface fa0/0  
Router(Config-if)#ip address 172.16.1.1 255.255.0.0  
Router(Config-if)#no shutdown
```

Fast Ethernet 0/0 arayüzü yapılandırma moduna giriş yapar. Arayüze 172.16.1.1 IP adresini atar ve arayüzü açar.

```
Router(Config)#interface s0/0/0  
Router(Config-if)#ip address 192.168.1.1 255.255.255.252  
Router(Config-if)#no shutdown  
Router(Config-if)#clock rate 64000
```

Serial0/0/0 arayüzü yapılandırma moduna giriş yapar. Arayüze 192.168.1.1 IP adresini atar, arayüzü açar ve saat sinayı olarak saniyede 64000 bit belirtir.

Router#copy run start veya #write

Çalışan konfigürasyonu başlangıç konfigürasyonuna tıklar.

Router #sh run

Çalışan konfigürasyonu gösterir.

Router #sh ip int brief

Arayüzlerin durumunu özet olarak gösterir.

Router #sh ip route

Yönlendirme tablosunu gösterir.

YEDEKLEME

```
#copy run tftp
#copy start tftp
#copy flash tftp
```

İlk komutta çalışan konfigürasyon TFTP sunucusuna, ikinci komutta başlangıç konfigürasyonu TFTP sunucuna, son komutta ise IOS TFTP sunucuna kopyalanacaktır. Bu komutlar girildikten sonra bizden TFTP sunucunun IP adresi ve dosyanın hangi isimle kaydedileceği bilgileri istenecektir. Eğer TFTP sunucundan Router'a yedekleri geri yüklemek istersek komutlarda kaynak ve hedefin yerleri değiştirilmelidir.

Cisco Switch Yapılandırması

Bir IOS çalıştırılan cisco Switch'te aşağıdaki modlar bulunur.

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

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 direk 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 Kriptolama

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 }**

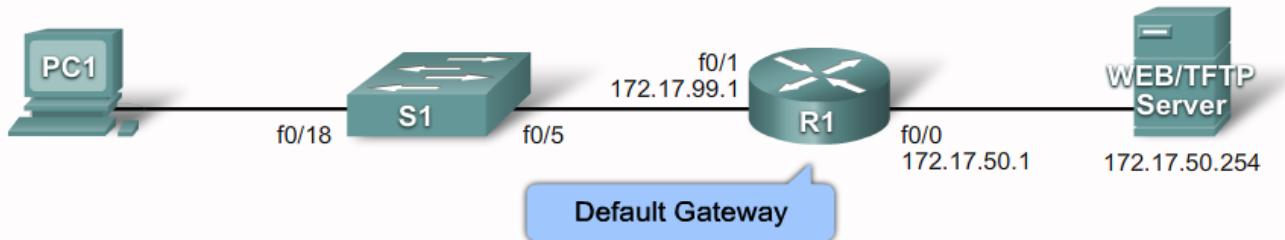
SWICTH YÖNETİM ARAYÜZÜNÜ (MANAGEMENT INTERFACE) YAPILANDIRMAK

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 iletişimde 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 dönüşüceye 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 adres 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
```

TCP / IP İnternet Katmanı, IPv4 Adresleme ve Alt Ağlara Giriş

IP ADRESİ NEDİR

Gerçek hayatta 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, karşısının sözünü kesmeme, anlaşılmadığı durumda tekrar etme, anlaşıldığına dair karşısından 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 kuramıyor. Bu kısımda, bu protokollerden TCP/IP protokol kümesine ve bir takım temel kavramlara değineceğiz.

Bugünlerde pek posta ile haberleşme kullanılmasa da ağ cihazları arasındaki iletişimini anlaşılabilmek 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 dağıtıcı aracılığıyla mektubunuza alıcıya ulaştıracaktır. İşte insanlar arasındaki iletişimde adres kavramına karşılık, ağ cihazları arasındaki bu **mantıksal adres** kavramını **internet protokolü (IP)**oları. Yalnız adres bilgisindeki cadde, sokak gibi kavramlar 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;

110000001010100000001010000011

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

11000000.10101000.00000101.00000011

Günlük hayatı onluk sayı düzenini kullandığımız için bu gruplandırma kolaylık sağladığı söylememez. 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 degeinilecek birkaç sebepten dolayı bu adreslerin tümünü kullanmadığımızı göreceğiz. IP adres kavramı ilk oluştuğunda bu sayı yeterli görüldüyordu ancak bugün dünya üzerinde adreslenmesi gereken cihaz sayısı düşünüldüğünde bu sayının aslında yetersiz olduğu tespit edilmiştir. Yani günümüzde IP adresi kılığı yaşınamaktadır. 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 konuşabilirisiniz. Bilgisayar ağlarında da bu durum söz konusudur.

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

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

Ortamda belli bir grup cihazsa **çoklu yayın (multicast)** adımı alır.

IP ADRES SINIFLARI

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

- 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 ağda bulunan cihazların IP adreslerinin ağ kısmı tüm cihazlarda aynı iken, host kısmı farklıdır*. Bu yapı ağın büyüğünü 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 ağda 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 atanın geçerli birer adres olmayacağıdır. O halde A sınıfı bir ağda cihazlara atayabileceğimiz adres sayısı $2^{24} - 2$ olacaktır.

Örnek, 10.X.X.X IP aralığına sahip bir ağda 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 Kısmı

1.3.5 – Host kısmını gösterir.

10.1.3.5 IP adresine sahip bir cihaz ile aynı ağda bulunan başka bir cihazın adresi 10.2.4.6 olabilir. Çünkü ağ kısmı 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 kısmın ağ kısmını gösterirken altı çizili olmayan kısmın 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: 1111111.1111111.1111111.00000000 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 şekele 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 belirtmiştik. 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 ağda kullanmak için belli grup IP adresleri rezerve edilmiştir. Bu IP adresleri, RIR'ler tarafından dağıtılmayan özel IP adresleridir (Private IP Addresses, RFC 1918).

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 atanınan adres gruplarıdır.

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

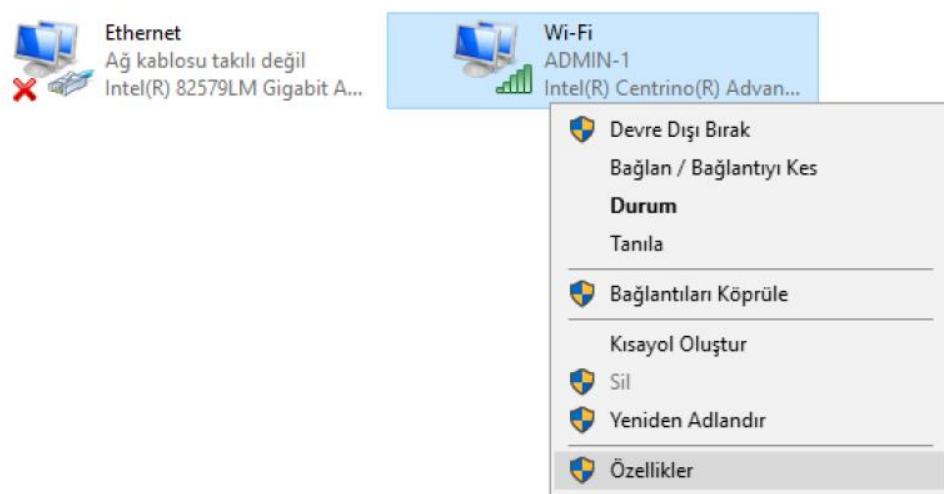
Ayrıca, 192.0.2.0 / 24 ağ aralığı TEST-NET adresleri olarak ayarlanmıştır. Bu IPler dokümantasyonda ve eğitimlerde kullanılır.

IP ADRES ATAMASI

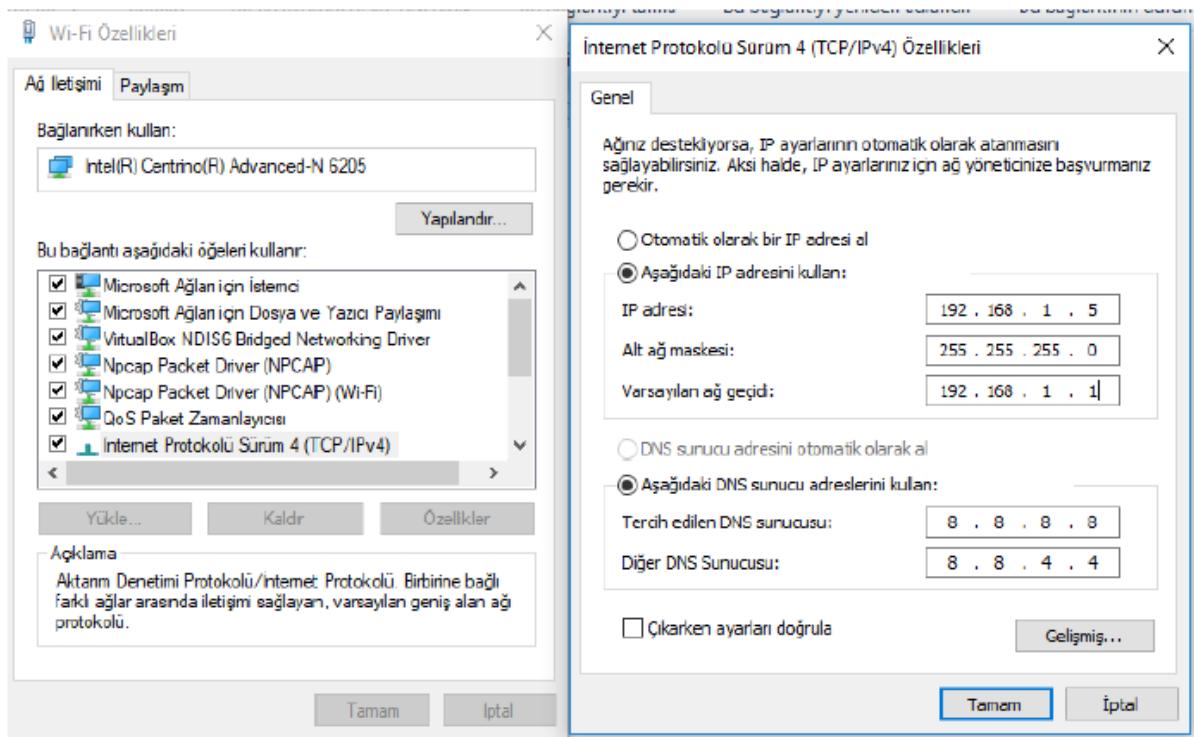
Cihazlara IP ataması statik ve dinamik olmak üzere iki şekilde yapılabilir. Statik IP ataması yapılırken, ağdaki benzersiz IP adresini, SubnetMask, Default Gateway (Varsayılan Ağ Geçidi) gibi bilgileri cihaza statik olarak atamak gereklidir.

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

Denetim Masası\Tüm Denetim Masası Öğeleri\Ağ Bağlantıları yolu aracılığıyla IP ataması yapılması istenen Ethernet bağıdaştırıcısı 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 kiralanır. Yukarıdaki ekranda “**Otomatik olarak bir IP adresi al**” seçildiğinde gerekli bilgileri dinamik olarak alınır

ALT AĞLARA BÖLMEK (SUBNETTING)

IPv4 ağlarda IP tasarımini yaparken adresleri verimli kullanmak, ağ yönetiminin kolaylaşdırılmak 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 ayırbilir, 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 bölümü 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 şekilde 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” içerdiginden)

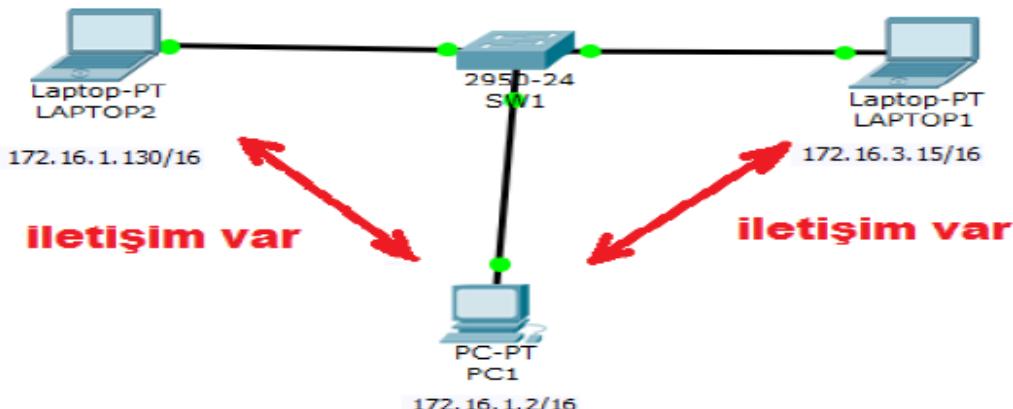
O halde herhangi bir IP'nin 172.16.1.2/16 ile aynı ağıda olabilmesi için, o IP'nin ilk 16 bitinin 172.16.1.2 IP'sinin ilk 16 biti ile aynı olması gereklidir.

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 ağıda değil
172.16.3.15	10101100.00010000.00000011.00001111	Aynı ağıda
172.16.1.130	10101100.00010000.00000001.10000010	Aynı ağıda

255.255.0.0 maskesine göre 172.16.1.2 ile 172.16.3.130 IP adresleri *aynı ağıdadır ve birbirileri ile doğrudan iletişime geçebilirler.*

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



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.

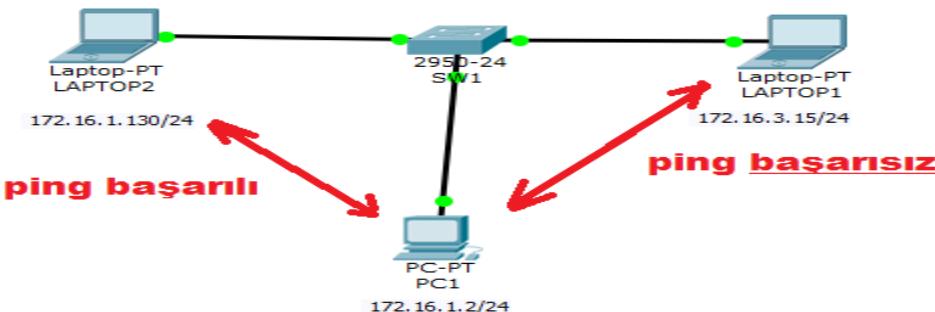
Aşağıdaki IP adresini kullan:

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 geçemezler 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 ağıda 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 farklılığın 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 ağıda 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 ağıda olacaktı.

Yani 172.16.1.2/25 ile 172.16.1.130/25 aynı ağıda 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:

1111111.1111111.1111111.10000000

Ondalık sayıya çevirelim:

255.255.255.128

Şimdi de IP tasarımını 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 değişimmemek 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. Toplam 128 IP
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	
10101100.00010000.00000001.01111111	172.16.1.127	
10101100.00010000.00000001.10000000	172.16.1.128	Bu bloktaki IP'lerin ilk 25 biti kendi arasında aynıdır. Toplam 128 IP
10101100.00010000.00000001.10000001	172.16.1.129	
10101100.00010000.00000001.10000010	172.16.1.130	
10101100.00010000.00000001.10000011	172.16.1.131	
.....	
10101100.00010000.00000001.11111110	172.16.1.254	
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 ayırır. Ancak burada kullanılabilir IP adresi ile ilgili durumu unutmamak gereklidir. 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"dır. Bu adreslere Network Adresi denir ve ağ temsil eden bir adresidir. Yani 172.16.1.0 ile 172.16.1.127 arasındaki tüm IP'ler 172.16.1.0/25 ağındır bu IP'lerim 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 gönderilecekse 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 ağı için 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	11000000.10101000.00000001.00000000	Kullanılamaz. Son 6 bit “0”	İlk Subnet (*Subnet Zero)
192.168.1.1	11000000.10101000.00000001.00000001	Kullanılabilir ilk IP	
192.168.1.2	11000000.10101000.00000001.00000010	Kullanılabilir ikinci IP	
.....	
192.168.1.62	11000000.10101000.00000001.00111110	Kullanılabilir son IP	
192.168.1.63	11000000.10101000.00000001.00111111	Kullanılamaz son 6 bit “1” Broadcast adresi	
192.168.1.64	11000000.10101000.00000001.01000000	Kullanılamaz. Son 6 bit “0”	İkinci Subnet
192.168.1.65	11000000.10101000.00000001.01000001	Kullanılabilir ilk IP	
.....	
192.168.1.126	11000000.10101000.00000001.01111110	Kullanılabilir son IP	
192.168.1.127	11000000.10101000.00000001.01111111	Kullanılamaz son 6 bit “1” Broadcast adresi	Üçüncü Subnet
192.168.1.128	11000000.10101000.00000001.10000000	Kullanılamaz. Son 6 bit “0”	
192.168.1.129	11000000.10101000.00000001.10000001	Kullanılabilir ilk IP	
.....	
192.168.1.190	11000000.10101000.00000001.10111110	Kullanılabilir son IP	
192.168.1.191	11000000.10101000.00000001.10111111	Broadcast Adresi	Dördüncü Subnet
192.168.1.192	11000000.10101000.00000001.11000000	Kullanılamaz. Son 6 bit “0”	
192.168.1.193	11000000.10101000.00000001.11000001	Kullanılabilir ilk IP	
.....	
192.168.1.254	11000000.10101000.00000001.11111110	Kullanılabilir son IP	
192.168.1.255	11000000.10101000.00000001.11111111	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 ağa 1.subnet denmez.** Yani sayma işlemi için sayılar kümesi kullanılmaz ☺ İlk subneta 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.

$/27 = \textcolor{red}{11111111.11111111.11111111.111}00000$ (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 eksiğidir.

Örneğin 192.168.2.64 ağının broadcast adresi 192.168.2.95 '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 oktettedir. 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:

IP Configuration	
<input type="radio"/> DHCP	<input checked="" type="radio"/> Static
IP Address	10.157.15.1
Subnet Mask	255.255.248.0

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

Çözüm:

Subnet maskesinin ilk iki oktet tamamen “1”lerden, son oktet ise tamamen “0”lardan oluşuyor. Oysa üçüncü oktet bit bazında yazıldığında “11111000” 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 araliktadır)

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

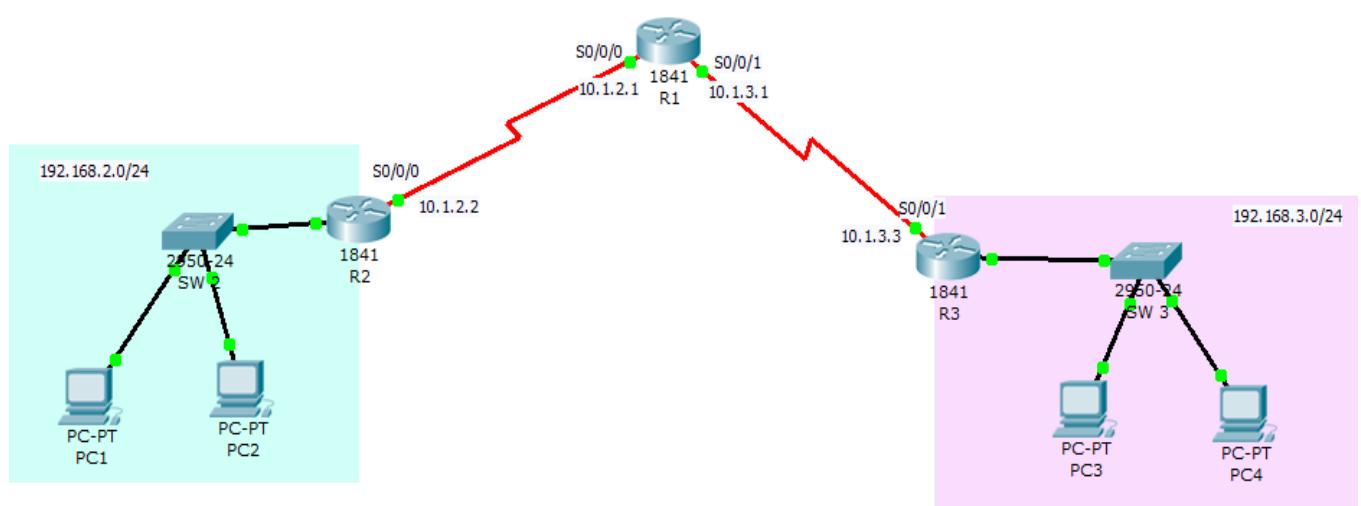
Kullanılabilir son IP : 10.157.15.254

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 gerektiği bilgisi yönlendiriciler üzerinde bir tabloda tutulur. Bu tabloya yönlendirme tablosu denir. Her yönlendirici kendisine ait böyle bir tablo tutar.

Aşağıdaki örnekten 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.

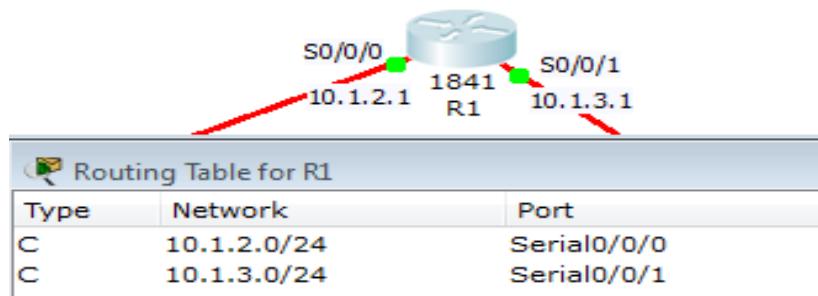
Yönlendirme Tablosu Nasıl Oluşturulur?

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

Direk Bağlı Rotalar (C):

Yönlendiriciye direkt 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 protokollerü aracılığıyla dinamik olarak öğrenilmesi gerekmektedir.

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



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

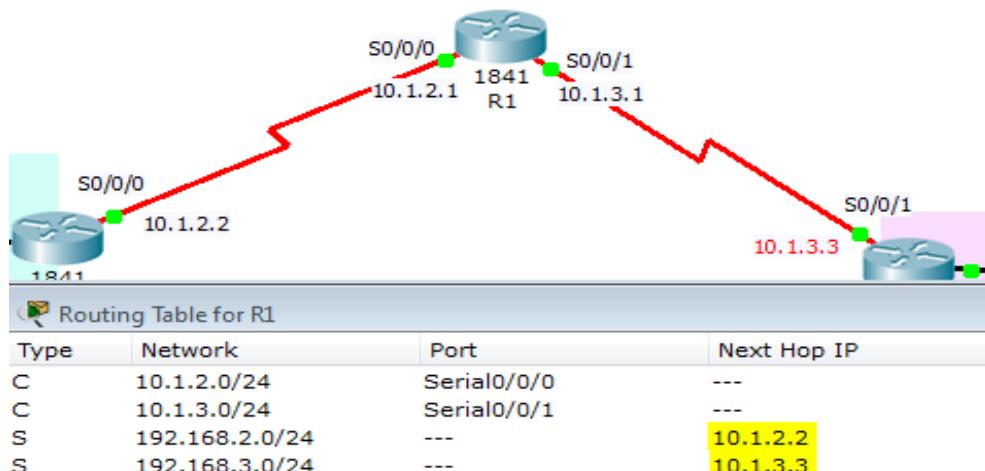
```
R1(config)# ip route 192.168.2.0 255.255.255.0 S0/0/0
```

```
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.

Dinamik Yönlendirme:

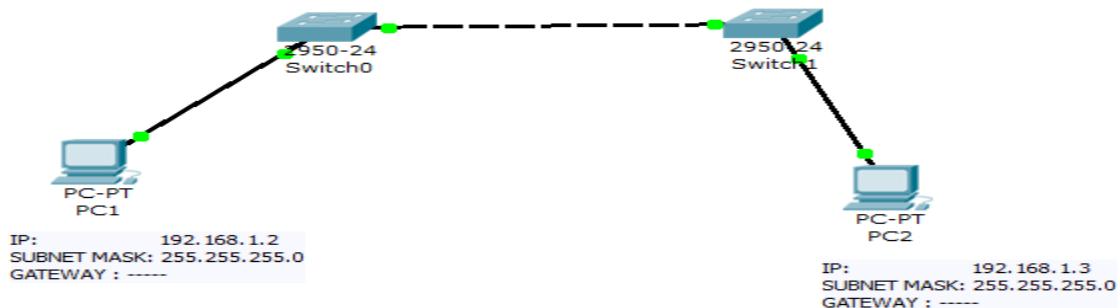
Dinamik yönlendirme, özellikle büyük ağlarda rota sayısının fazla olduğu ağlarda dinamik olarak rotaların öğrenilmesini amaçlar. Bunun için RIP, IGRP, EIGRP, OSPF gibi yönlendirme protokolleri kullanılır. Böyle bir durumda yönlendiriciler birbirlerine kendi ağ adresleri hakkında güncellermeler yaparlar.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

Yukarıdaki örnek topoloji için bu kez Dinamik Yönleendirme 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 görüntülendiğine dikkat ediniz.

ÖRNEK İLETİŞİMLER



Şekildeki PC1 ile PC2 fizikselleşmiş ve mantıksal olarak aynı ağdadır. Mantıksal olarak aynı ağa da olmaları, aynı IP aralığında olmalarından kaynaklanır. (192.168.1.0/24) Bu durumda PCler arasında iletişim vardır. Aşağıda PC1 'den PC2'ye iletişimini 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:

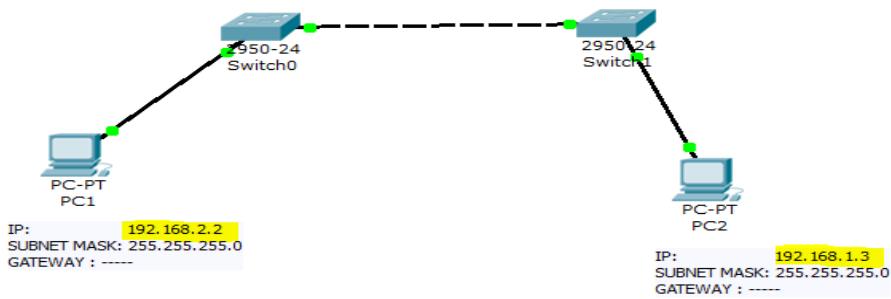
Reply from 192.168.1.3: bytes=32 time=15ms TTL=128
Reply from 192.168.1.3: bytes=32 time=60ms TTL=128
Reply from 192.168.1.3: bytes=32 time=27ms TTL=128
Reply from 192.168.1.3: bytes=32 time=35ms TTL=128

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 anda bulunmalarına rağmen, mantıksal olarak farklı ağlardadırlar. PC1, 192.168.2.0/24 ağında, PC2 ise 192.168.1.0/24 ağındadır. Bu durumda iletişim sağlayamazlar. Aşağıda bu durumdaki iletişim isteğinin 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

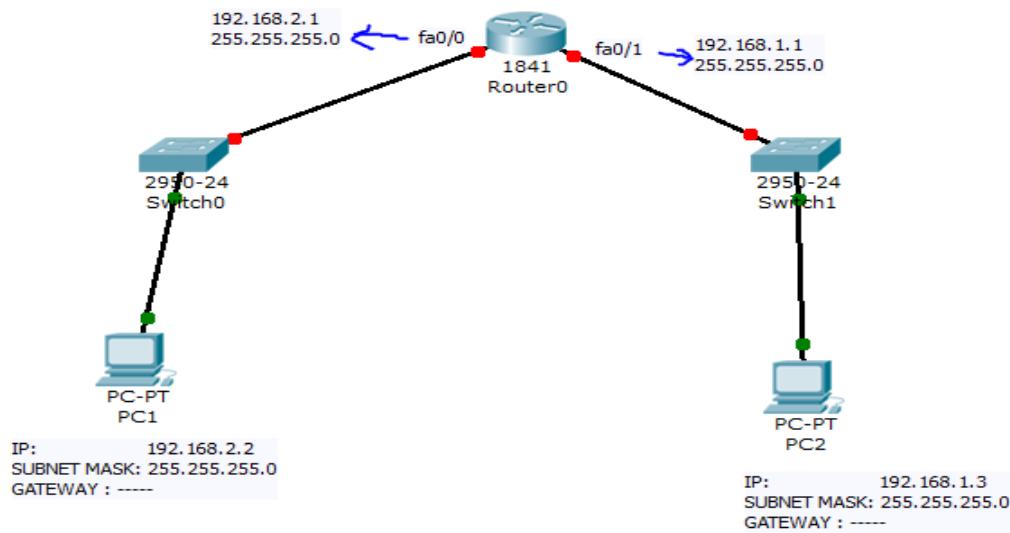
PC>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

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

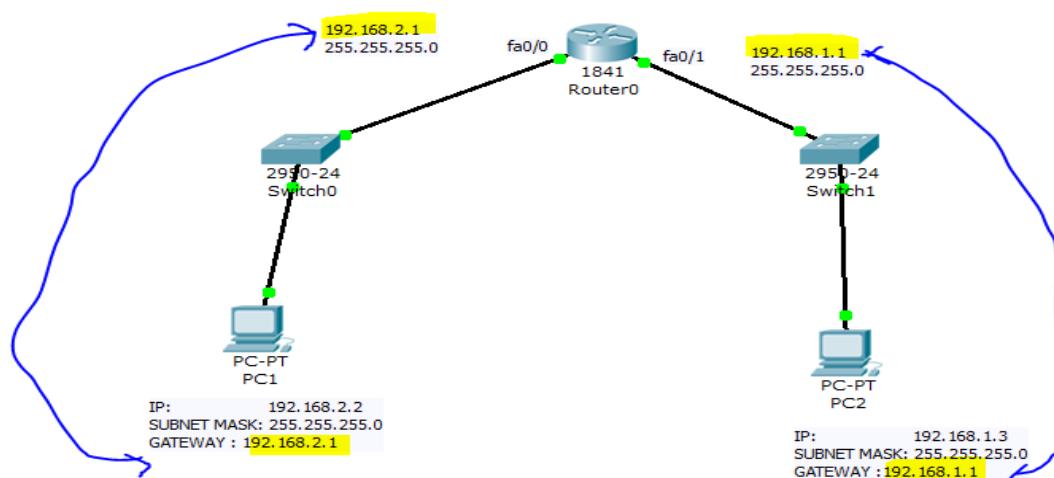
Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

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 ağda 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 ağda olduğunu anlayabilmeleri için Subnet Mask bilgisi 255.255.255.0 verilebilir. Yine PC2 ile Router Fa0/1 portu da aynı ağda 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 aralıktır. **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

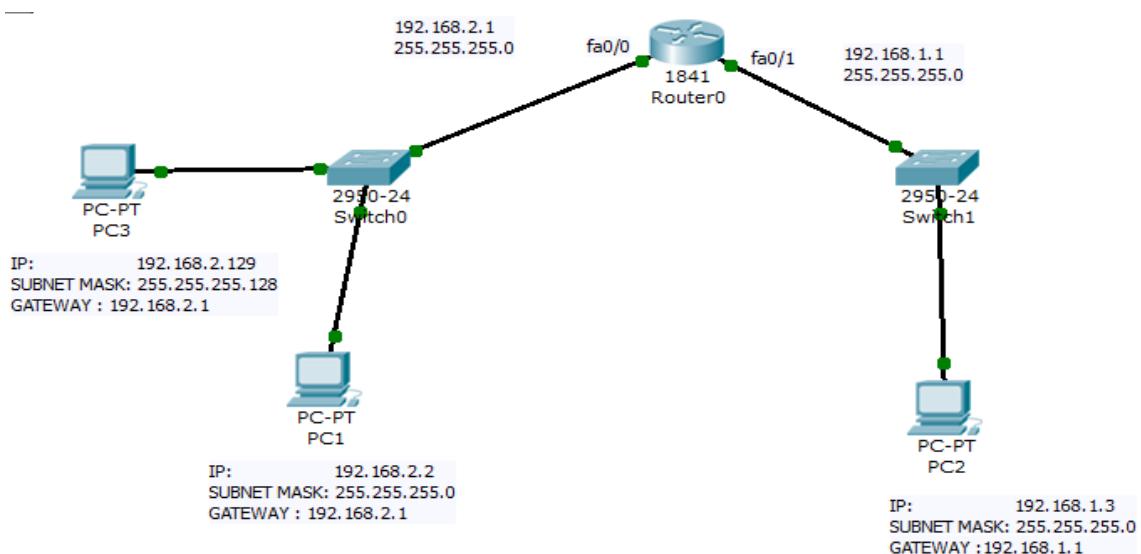
```

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 oalcaktır. Çünkü;

PC1'in iletişim kuracağı IP adresi 192.168.2.129

PC1 bu hedef IP adresi ile **KENDİ SUBNETMASK** bilgisini andleyecektir.

$192.168.2.129 \wedge 255.255.255.0 = 192.168.2.0$ (**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.

IPv6 ‘ya GİRİŞ

İnternette hâlihazırda kullanılan uyarlama olan IPv4 protokolü, yaklaşık 30 yıl önce geliştirilmiştir. IPv4’ün en önemli yetersizliği 32-bit olarak tanımlanan adres kapasitesidir. IETF tarafından geliştirilen IPv6’da, 32 yerine 128 bit’lik IP adresleri tahsis edilerek IPv4’ün adres kapasitesi geliştirilmiştir. Internetin başarısındaki hızlı yükseliş, IP adreslerinin tüketimini hızlandırmıştır. IPv4 adreslemenin başlangıçta etkin bir şekilde organize edilmemesi ve adaletli dağıtılmaması sebebiyle birçok ülke IPv4 adres ihtiyacını karşılayamamaktadır. Bu bölümde, yeni nesil IPv6’ya neden ihtiyaç duyuluğu, IPv6’nın avantajları, farklılıklar, IPv6 adres yapısı ve türleri, IPv4 ‘den IPv6’ya geçiş entegrasyonu konuları ele alınmıştır.

IPv4 YETERSİZLİKLERİ

IPv4 adresleri yaklaşık 25 yıldır kullanımda olmasına rağmen, hem adres kapasitesi hem de güvenlik yönleriyle günümüzde yetersiz gelmeye başlamıştır. 32-bit adres ile her ne kadar 4 milyardan fazla adres tanımlanabilse de, bu adreslerden yaklaşık 3,7 milyarı kullanılabılır adrestir. IPv4 adres yetersizliğine

çözüm olarak bazı teknikler geliştirilmiştir. RFC 1918 ile tanımlı olan bazı IP grupları iç ağda kullanılmak üzere rezerve edilmiştir.

A sınıfı 10.0.0.0 / 8

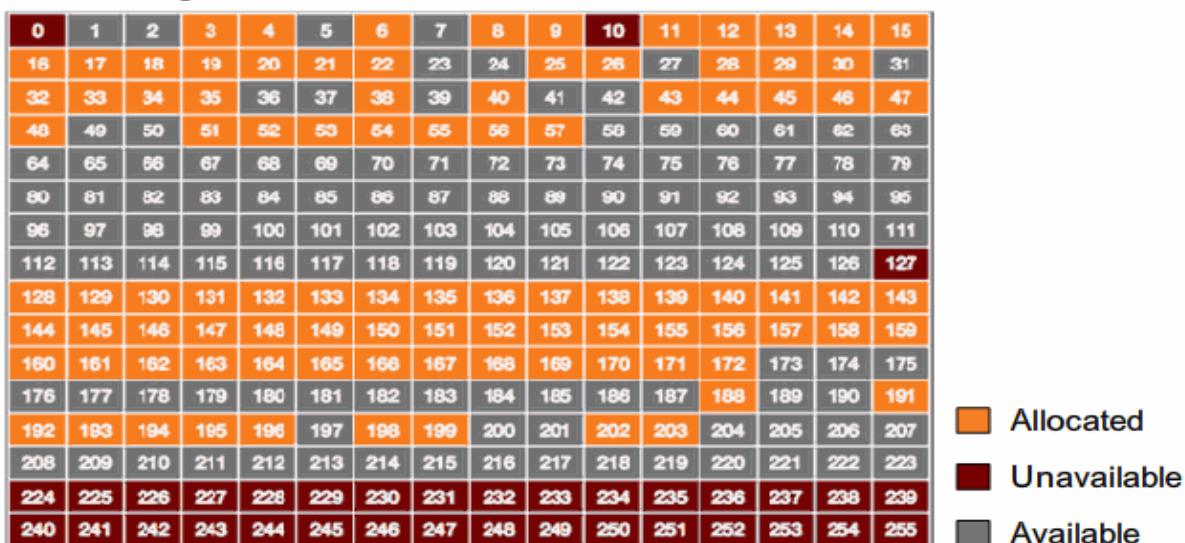
B sınıfı 172.16.0.0 / 12

C Sınıfı 192.168.0.0 / 16 adresleri rezerve edilmiş adreslerdir.

Rezerve edilmiş bu adresler, ISP'ler tarafından dağıtılmayacak ve sadece iç ağlarda kullanılacaktır. Ancak, iletişimde her cihazın benzersiz IP adresi alması gerektiğinden, bu adreslerin ağ dışına çıkışken tekil olan global bir adrese dönüştürülmesi gerekmektedir. Network Address Translation (NAT) denen bu yöntemle bu özel adresler, global adreslere dönüştürülür. Ancak NAT, birebir iletişim için bir sorun oluşturur ve IPv4 ağları için bir dar boğaz teşkil eder.

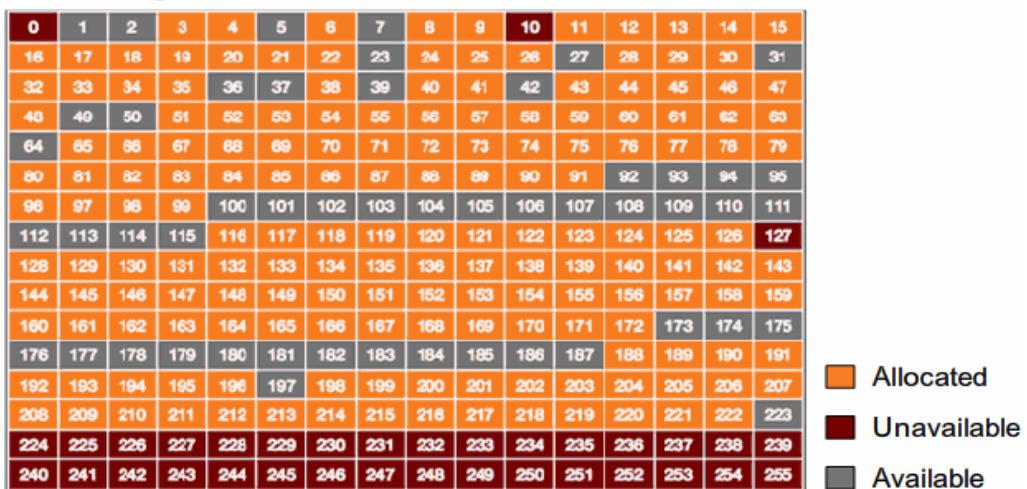
Aşağıdaki şekillerde IPv4 adreslerinin ne oranda tüketildiğini gösterilmektedir.

Blocks Assigned - 1993



1993 yılında IPv4 dağılımı.

Blocks Assigned - 2007



2007 yılında IPv4 dağılımı

Ocak 2007 yılında yapılan araştırmalarda, IPv4 adreslerin 2,4 milyar kadarı dağıtılmış; 1,3 milyar adres ise kullanılabilir olarak tespit edilmiştir.

Kasım 2005 yılında yapılan araştırmalarda 973 milyon kullanıcı interneti kullanmaktadır. Logaritmik olarak artan insan popülasyonu, dolayısıyla interneti kullanacak insan sayısının artışı başta olmaz üzere bir çok etken sonucunda IPv4 adreslerinin istatistiksel olarak 2011 yılında tükenmesi beklenmektedir.

Gelişen teknolojilerle beraber, IP tabanlı mobil telefonlar, PDA'lar gün geçtikçe daha çok kullanılmaktadır. Mobil kullanıcı sayılarındaki artış yine IPv4 kıtlığını tetikleyen diğer bir etmendir.

Ayrıca, otomotiv ve havacılık sektörlerinde, takip amaçlı olarak kullanılan IP adresleri de IPv4 havuzunun hızla tükenmesini sağlamaktadır.

Yeni teknolojiler gelişikçe, uzaktan müdahale amaçlı ev aletlerine de IP adresi atanması gündeme gelmiştir. Tüm bu etkiler sonucu IPv4 adreslerin yetersiz kalacağı ve dolayısıyla yeni bir yöntem gerekliliği doğmuştur.

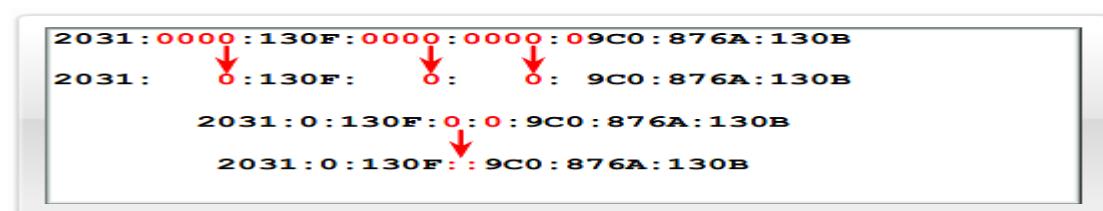
IPV6'YA GEÇİŞ

IPv4 adreslerindeki eksiklikler sonucu yeni nesil IP kavramı gündeme gelmiştir. IETF, 1992 yılında yeni IP için istekte bulunur. IPv6, 1995 yılında IETF tarafından resmen duyurulmuştur.

Cihaz imalatçıları ve yazılım yayınlarının anahtar rolü, omurga İşletmecilerin Engelleyici Hareketi, Mobilite ve haraketli kullanımın IPv6'yı uzun vadede kaçınılmaz yapması, IPv6'nın büyümeyi desteklemesi ve zorlaması, gibi hususlar IPv6'ya geçiş hızlandıracaktır.

IPV6 ADRES YAPISI

IPv6 adreslerini gösteren sekiz 16-bit onaltılık alanlar dizisi iki nokta işaretleri ile ayrılır. IPv4'ün tersine, IPv6'da adres dizgi biçimini sabit değildir.

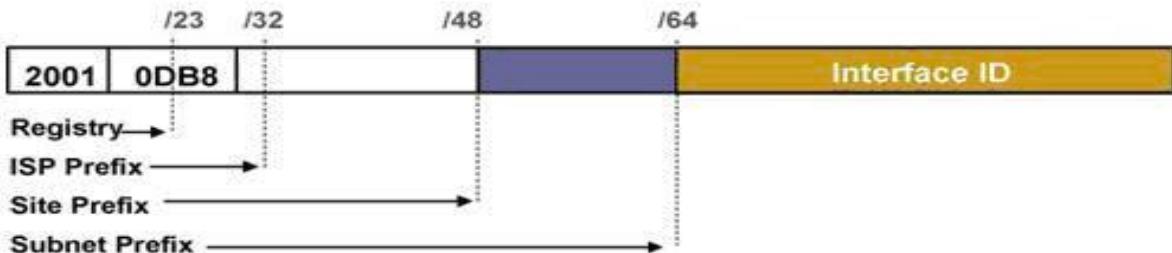


IPv6 adres dizgisi gösteriminde aşağıdaki kurallar izlenir:

- Alandaki onde yer alan 0'lar isteğe bağlıdır: 09C0 ile 9C0 ve 0000 ile 0 eşdeğerdir.
- 0'ların bir ya da daha fazla grubu ihmal edilerek "::" ile değiştirilebilir. Bir adresde sadece bir "::" işaretine izin verilir.
- Açıkça belirtilmemiş bir adres, sadece 0 içerdiginden "::" olarak yazılır.

"::" gösterimini kullanmak, birçok adresin boyutunu büyük ölçüde azaltır. Örneğin, FF01:0:0:0:0:0:1 gösterimi FF01::1 haline gelir. Bu biçim, IPv4'ün 32-bit noktalı ondalık gösteriminden farklıdır. IPv6 adresinin başlıca türü unicast yayın olarak adlandırılır.

IPv6 adreslerinin subnet maskları "/" li olarak gösterilmektedir. IPv4'ten farklı olarak sabit subnetmasklar kullanacaktır. Bu adreslerin ilk 64 biti her zaman network bitleri, son 64 biti ise her zaman, artık interface id olarak adlandırılacak host bitleridir.



IPv6 Adreslerinin hiyerarşik bir yapısı vardır. Buna göre,

İlk 23 bit : Registry Prefix , her RIR için farklı bir değeri gösterir.

Örnek: RIPE.NET

Sonraki 9-bit: ISP önekidir.

Örnek: Türk Telekom IP bloğunu

Sonraki 16-bit : Site Prefix, kurumsal ağı gösterir.

Örnek: Ankara Üni.IP blogu

Sonraki 16-bit: Kurumsal ağda, alt ağ oluşturmak için kullanılabilir.

Örnek: Elmadağ MYO IP blogu

Son 64 –bit : Son 64-bit, host cihazı temsil eder.

IPV6 ADRES TÜRLERİ

IPv4 adreslerde olduğu gibi, IPv6 adreslerde de yayın türleri bulunmaktadır. Yeni versiyon IP adreslerinde Unicast, Multicast ve Anycast yayınlar bulunmaktadır. IPv4'ten bildiğimiz Broadcast adresler IPv6 mimarisinde bulunmamaktadır.

UNICAST YAYINLAR

IPv6 paketlerinin belirli bir adressteki tek bir cihaza gönderilmesidir. Global Unicast ve Link Local Unicast adresler olmak üzere ikiye ayrılır.

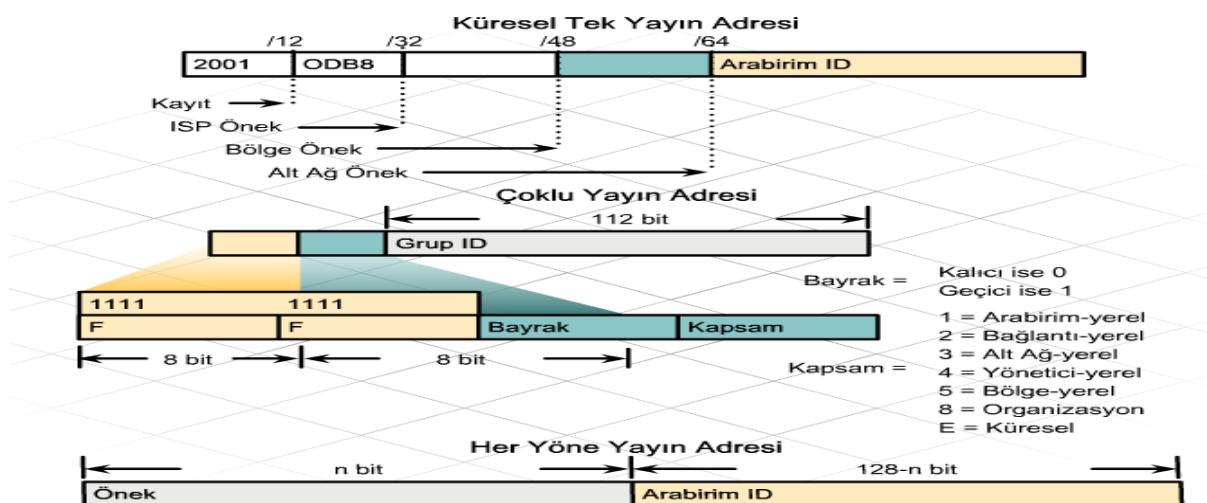
Global Unicast adresler IPv4 adreslerinden de bilgiğimiz **Global Unicast** adreslere eşdeğerdir.

48 bitlik Global Routing Prefix ve 16 bitlik Subnet ID'den oluşur. Global Unicast adreslerdeki bu Subnet ID ve Global Routinf Prefix'ler Route summarization için bir çözüm olarak düşünülmüştür. Internet erişimi sırasında bu adresler kullanılarak paketler gönderilir. Bu adresler 2000::/3 'den başlayıp

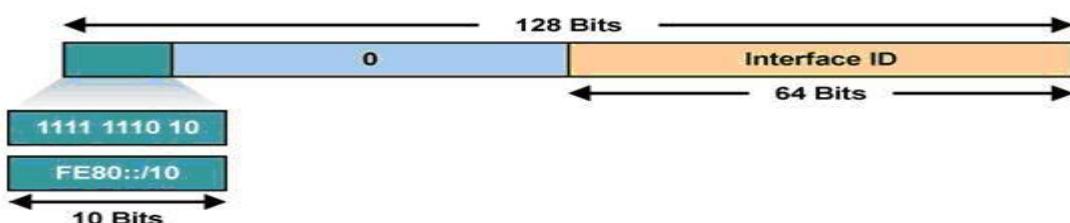
EEEE::/3 ‘e kadar gidebilir. Sadece bu aralıkta yer alan FF00::/8 prefıxi multicast adresler için rezerve edildiği için, Global Unicast adres olarak kullanılamaz.

IANA, bu IPv6 adres aralığını ARIN, RIPE, APNIC, LACNIC, ve AfriNIC olmak üzere beş Regional Internet Registry’lere tahsis edilmiştir.

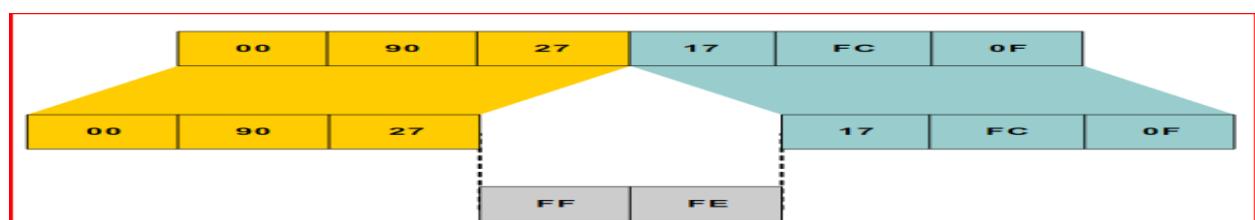
Global Unicast adresler, 48-bit global yönlendirme önekine (prefix) ve lokal ağlarda alt ağ (subnet) oluşturabilmek için 16-bit subnet ID değerine sahiptir. Bu sayede 65 535 alt ağa oluşturulabilir.



Link Local adresler ise, adından da anlaşılacağı gibi local olarak kullanılacak adreslerdir. FE80::/10 aralığı Link Local adresler için rezerve edilmiştir ve network cihazları 64 bitlik interface id’lerini otomatik olarak üretip, bu prefix altında çalışabilirler.



Link-Local Adresler: FE80::/10 aralığındadır. Yani ilk oktetleri FE8 ile FEB arasındadır. Ayrıca 64-bitlik Interface ID’ler MAC adresinden otomatik olarak hesaplanır (**EUI-64**)

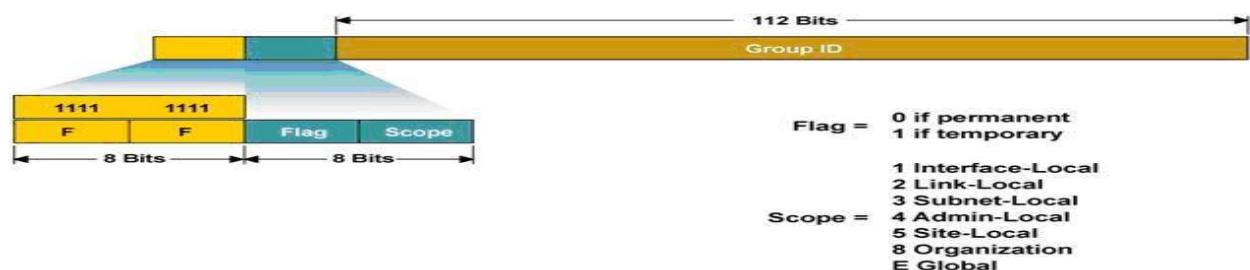


Şekil 3: EUI-64 Yöntemi

EUI-64 yöntemine göre, cihazın fiziksel adresinin (MAC Adresi) ilk 24 biti ile son 24 biti arasına 16-bitlik FFFE çifti yazılır. Böylece IPv6 yapısında 64-bit olarak bulunan Interface ID otomatik olarak hesaplanır. Link-Local adresler, otomatik adres ataması, komşuluk keşfi ve yönlendirici keşfi gibi amaçlarla sadece lokal ağlarda kullanılmak üzere tasarlanmıştır.

MULTICAST YAYINLAR

Paketlerin belli bir gruptaki üyelere gönderilmesidir. IPv4 adreslerinden de bildiğimiz Multicast adresler IPv6 içerisinde de aynı yer almaya devam edecek. Hatta artık broadcast adresler olmayacağı için multicast adreslerin daha da önem kazanmıştır. IPv4'ten bildiğimiz gibi multicast adresler aslında multicast grupları gösterir. Multicast bir adrese gönderilen paketler aslında o multicast grubuna üye olan bütün cihazlara gönderiliyor demektir.



IPv4 içerisinde 224.0.0.0'dan 239.255.255.255'e kadar olan multicast adresler için IPv6'da **FF00::/8** aralığı kullanılmaktadır. Burada ki "00" olan ikinci oktet multicast adresin geçerlilik süresini (flag olarak belirtilmektedir) ve kapsama alanını (scope) gösterir. Ipv6 Multicast adresler için TTL (Time-to-Live) süresi yoktur, bunun yerine flag alanındaki değerler kullanılır.

Geçerlilik süresi noktasında Multicast adresler için belirleyebildiğimiz iki tanım var: bunlardan birincisi sürekli (FF0X) diğeri ise geçicidir. (FF1X) Scope alanında ise Interface Local, Link Local, Subnet Local gibi farklı seçeneklerimiz var. Ipv6 adresleri içerisinde broadcast adresler olmadığı için, multicast paketler her yerde kullanılacak diyebiliriz. Buradan bütün cihazların birer multicast adresi olacağı anlamı çıkabilir ki doğrudur. Örneğin bütün cihazlar FF02::1 multicast grubuna dahildir. Bu ve benzeri multicast adresleri aşağıdaki tabloda gösterilmiştir.

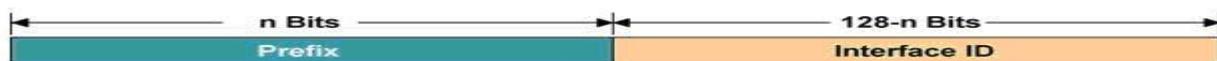
	Meaning	Scope
FF02::1	All nodes	 Link-local
FF02::2	All routers	 Link-local
FF02::9	All RIP routers	 Link-local
FF02::1:FFXX:XXXX	Solicited-node	 Link-local
FF05::101	All NTP servers	 Site-local

Routing protokollerden bildiğimiz multicast adresler, IPv4 multicast adreslerine benzer şekilde kullanılmaya devam edecektir. Örneğin FF02::9 bütün RIPv2 Routerlarının, FF02::5 ise bütün OSPF Routerlarının dahil olduğu multicast grup ip adresidir.

ANYCAST YAYINLAR

Paketlerin verimlilik açısından en yakın arabirimde teslim edildiği adreslerdir. Bu nedenle, her anycast yayın, en yakın arayüz türünde adres olarak düşünülebilir.

Anycast adresler birden fazla interface' e atanabilen ve bu adrese gönderilen isteklere, ilgili interfacelerden en yakın olanının cevap verdiği bir yapının ürünüdür. Burada en yakın kavramı Routing protokollere göre değişkenlik gösterir. Multicast adreslerin aksine anycast adresler için tanımlanmış herhangi bir aralık yoktur. Yani unicast olarak kullanılabilen her adres anycast olabilir. Burada bir adresin anycast olup olmadığı, konfigürasyon sırasında belirtilmelidir.



REZERVE EDİLMİŞ ADRESLER

IETF, IPv6 adreslerinin büyük bir kısmını çeşitli sebeplerle ve gelecekte kullanım amaçlı olarak rezerve etmiştir. Rezerve edilmiş bu adres aralığı, toplam IPv6 adreslerinin 1/256' ini kapsamaktadır.

PRIVATE ADRESLER

IPv6 adresleri içerisinde belli bir adres aralığı, IPv4'te olduğu gibi private adresler olarak ayırmıştır. Bu adresler sadece lokal ağlarda kullanılmak üzere ayrıldığı için, lokal ağların dışına yönlendirilmez. Private adreslerin ilk okteti onaltılık olarak, "FE" ile başlayıp bir sonraki değer 8 ile F arasında değişen değerler alır. Kapsama alanına göre bu adresler Site-Local adresler ve Link-Local Adresler olmak üzere iki tipe ayrılır. Site-Local adresler, IPv4 adreslerindeki RFC 1918 ile tanımlanan Private Adresler gibi

düşünülebilir. Ancak bu adres grubunun kullanımı, 2003 yılında yayımlanan RFC 3879 ile iptal edilmiştir. Site – Local adresler, “**FEC - FEF**” ile başlayan adres aralığındadır.

LOOPBACK ADDRESS

Tıpkı IPv4 adreslerde olduğu gibi, loopback testi için kullanılır. Ancak IPv4 adreslerde olduğu gibi büyük bir adres aralığı değil, sadece bir adres bu test için ayrılmıştır. Loopback adres, 0:0:0:0:0:0:1, veya kısa gösterim olarak "::1" ile ifade edilir.

TANIMSIZ ADRES

IPv4 adres mantığında olduğu gibi tüm bitleri "0" olan adres tanımsızdır. Bir ağ cihazının IP adresinin olmamasını temsil eder. Bu adres 0:0:0:0:0:0:kısa gösterim ile "::", tanımsız adres olarak gösterilir

IPV6 ADRES ATAMA TÜRLERİ

IPv6 adreslerin statik ya da dinamik olarak atanması mümkündür. Manual statik atama ve EUI-64 yöntemi ile statik atama olarak iki farklı statik atama yapılabilir. Bunun dışında Stateless Autoconfiguration yöntemi ve DHCPv6 ile dinamik atama da yapılabilir.

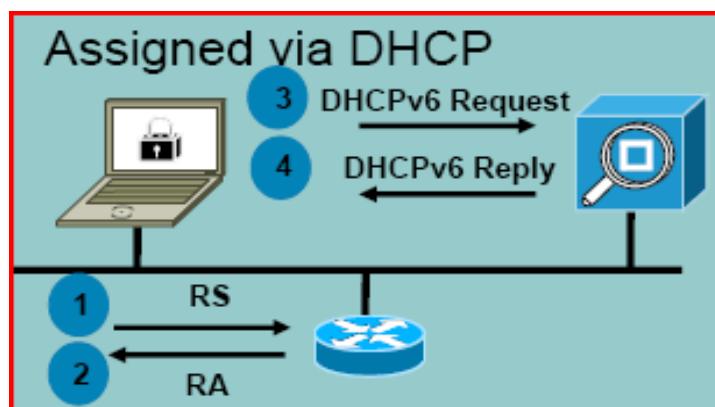
Manual Statik atamada, Network Prefix ve Interface ID değerlerinin her ikisi de statik olarak girilir. Örnek olarak 2001:DB8:2222:7272::72/64 gibi.

EUI-64 yönteminde ise IPv6 adresinin host kısmı OSI modeli 2.Katman (Data Link Layer) fiziksel adres bilgisinden (MAC Adres) otomatik olarak üretilir.

STATELESS AUTOCONFIGURATION

Autoconfiguration adres atamasında uçtaki cihazın PC olmadığı varsayılar. Cihaz ağa bağlılığı anda IP adresi alması sağlanır. Bu sayede yönetimsel yük de hafiflemiş olur.

DHCPV6 (STATEFUL)



DHCPv6 adres atamasında, ortamda bulunan bir DHCPv6 sunucunun, IP talep eden istemcilere IPv6 adreslerini ataması yöntemidir.

IPV4 VE IPV6 KARŞILAŞTIRMA

IPv6, IPv4'e göre güçlü iyileştirmeler sunar. Bu iyileştirmeler şunları içerir:

Taşınabilirlik ve güvenlik

Daha basit başlık

Adres biçimlendirmesi

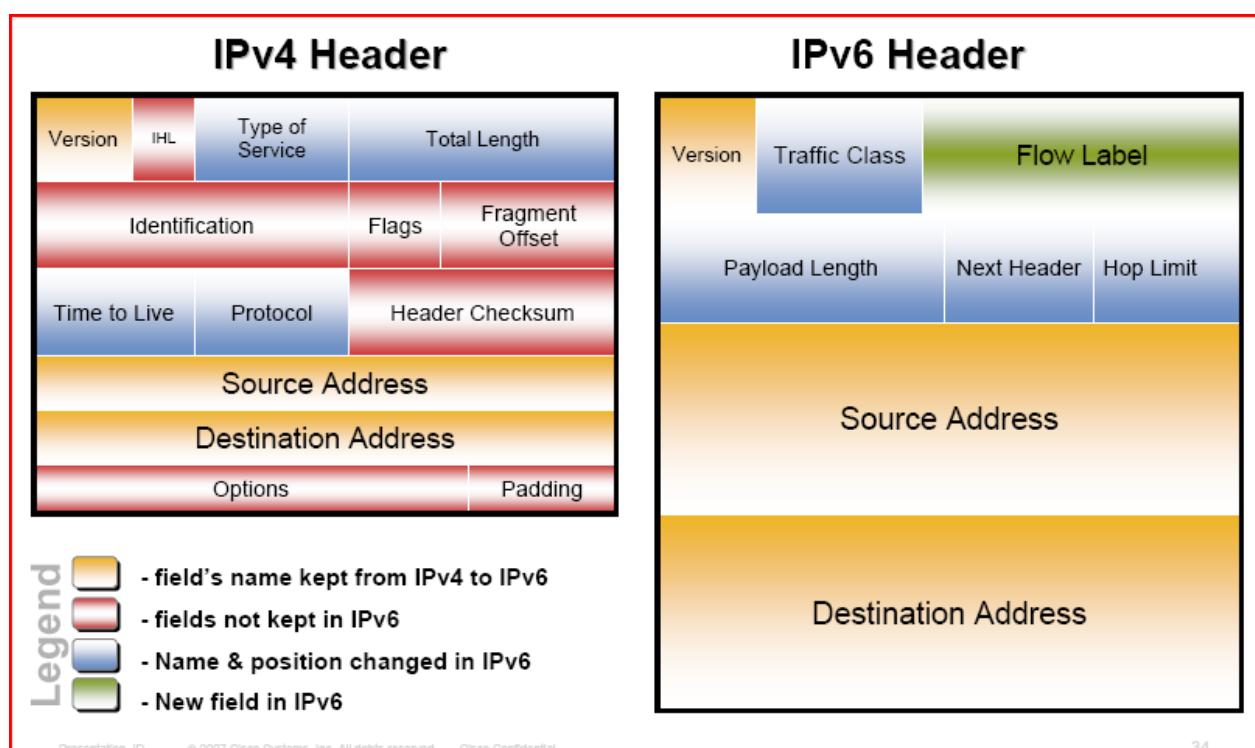
Taşınabilirlik ve Güvenlik

Taşınabilirlik, kişilerin taşınabilir ağ cihazları ile ağ içinde hareket edebilmesini sağlar. Mobil IP, IPv4 ve IPv6 için kullanılabilir bir IETF standartıdır. Bu standart, mobil cihazların, kurulmuş olan ağ bağlantılarını kesmeden hareket edebilmesini sağlar. IPv4 bu tür bir taşınabilirliği desteklemez. Taşınabilirlik, IPv6 özelliğidir.

Yine IPSec IP ağ güvenliği için bir IETF standartıdır. IPv4 ve IPv6'nın her ikisi için de kullanılabilir. IP ağ güvenliği işlevi, aslında her iki ortamda da aynıdır. IPSec IPv6 ortamında biraz daha bütünlüğündür ve her IPv6 düğümünde etkinleştirilebilir.

Daha Basit Başlık

IPv6 için kullanılan başlık, yönlendirme tablolarındaki girdi sayısını azaltarak yönlendirme verimliliğini artırır.



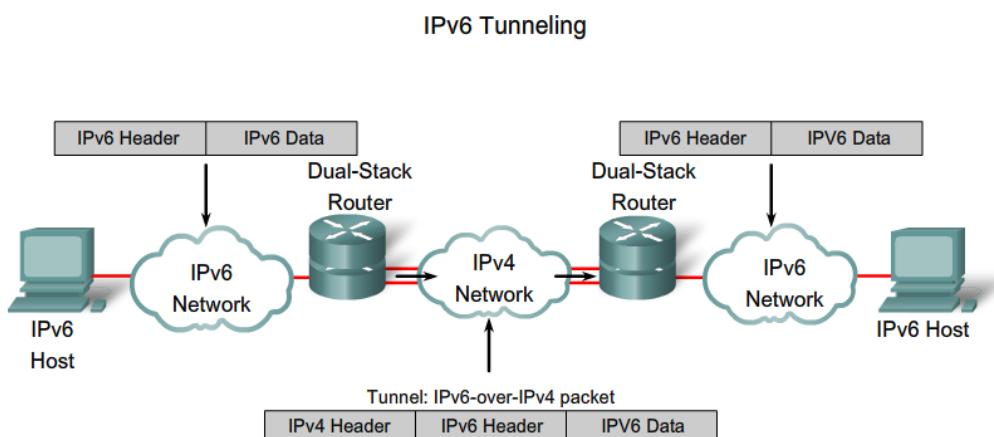
IPv6 ile herhangi bir yayın ilişkilendirilmemiştir. IPv4 ile oluşturulan yayınlar ağ içinde yüksek trafik oluşturur. Bu trafik, yayın fırtınası (broadcast strom) olarak adlandırılan ve tüm ağın çalışmasını durdurun bir olay meydana getirir. IPv6 broadcast yayınları yerine, multicast ve anycast yayınları kullanır.

IPV4 TEN IPV6 'YA GEÇİŞ

Bir IPv6 yapısını var olan bir IPv4 ağına entegre etmenin birkaç yolu vardır. IPv4'den IPv6'a dönüşümün tümü bir defada yapılmak zorunda değildir. En yaygın üç dönüşüm yöntemi şunlardır:

Dual Stack, Tünelleme ve Proxy NAT yöntemidir (NAT – PT)

Dual Stack dönüşüm yönteminde, hem IPv4 hem de IPv6 yapılandırmaları bir ağ cihazı üzerinde uygulanır. Her iki protokol yığınları aynı cihaz üzerinde çalışır. Bu yöntem, IPv4 ve IPv6'nın bir arada var olmasını sağlar.



Tünelleme, IPv6 büyümeye uyumluluk olarak daha önemli hale gelir. Tünelleme, bir protokol paketinin başka bir protokol içinde kapsülleňmesidir. Örneğin, bir IPv6 paketi bir IPv4 protokolü içinde kapsüllenebilir. Çeşitli IPv4 üzerinde IPv6 tünelleme yöntemleri vardır. Bazı yöntemler el ile yapılandırma gerektirir, bazıları ise otomatiktir.

IPv6 ve IPv4 arasında Ağ Adresi Çevirisi-Protokol Çevirisi (NAT-PT - Network Address Translation-Protocol Translation) dönüşüm yöntemi ile farklı versiyonlarda IP protokolü kullanan konak bilgisayarlar arasında direkt olarak iletişim kurulmasına imkan verir.

IPv6 NDP Protokolü

ICMP ND = Neighbor Discovery. ARP gibi çalışır.

Herhangi bir arayüze IPv6 adresi verilip enable (no shut) yapıldığında, Rtr kaynağı :: (adres yok) ve hedefi FF02:16 adresine multicast bir yayın yapar . FF02::16 tüm multicast grupları temsil eden özel bir multicast adrestir.

1. Ethernet portundan NS (Neighbor Solicitation) yayını yapar. Bu yayında kendi Link Local adresini (FE80 ile başlar) gönderir. ARP Request mekanizmasına benzetilebilir.
2. Eğer ortamda böyle bir adres yoksa (MAC adresten türetildiği için olmaması gereklidir) kendisine gelen DAD mesajında (Duplicate Address Detection) kullanılan LinkLocal adresin Unique olduğu belirtilir.
3. Ethernet portundan NA (Neighbor Advertisement) yayını yapıp kendini tanır.

Bir porta IPv6 adresi (Global Unique adres Ör: 2001:33AA::2) verilip enable yapıldığında RTR aşağıdaki multicast gruplara (FF02 ile başlayan adresler Multicast adresleridir) otomatik olarak eklenir.

- **All Local Devices:** FF02::1 adresi tüm lokal aygıtları temsil eder ve broadcast işlevini görür.
- **All Local Routers:** FF02::2 Tüm lokal routerlar.
- **Link Local Adres Multicast Group:** Link local adresin son 32-bit'i için de bir multicast grup oluşturulur. Örneğin Link local adres FE80::C202:2FF:FE8:0 olsun. Bu durumda RTR'nin dahil olacağı bir diğer multicast grup ta; FF02::1:FFE8:0 olacaktır. Bu adres, link local adres için multicast bir grubu temsil eder. Link Local adreslerin prefix değeri yoktur. (Subnet yoktur olarak düşünülebilir, çünkü LL adresler local dışından erişilemezdir)
- **Global Unicast Adres Multicast Group:**

Eğer bir cihaz, iletişim kurduğu IPv4 adresli bir cihazın MAC adresini öğrenmek istiyorsa BROADCAST yapar. Ancak IPv6 ortamında BROADCAST yoktur. O halde IP ADRESİ BİLİNEN bir cihazın MAC adresi IPv6 ortamında nasıl bulunur? Mantık basit! Eğer IPv6 adresini biliyorsam (Örneğin 2001:33AA::2) onun dahil olduğu Global Unicast Adres için multicast grubu da bilirim. Adresin son 24-bitinden üretilmiş olan Global Unicast Address Multicast grubu da, FF02::1:FF002 olacaktır. Böylece IPv4 teki gibi herkese giden bir BROADCAST yerine sadece ilgili cihaza giden bir MULTICAST yayınlanacaktır.

Temel IPv6 Router Yapılandırması

Her Router'da ipv6 paketleri için routing işlemini etkinleştirilmek gereklidir.

Router(config)# ipv6 unicast-routing

Bir interface'de IPv6 etkinleştirilmek :

Router(config-if)# ipv6 enable

Bir interface'e IPv6 Adres atama:

ipv6 address <address>/<prefix> [link-local] [eui-64]

Router(config-if)#ipv6 address 2001:db8:1:1::1/64

Interface'e unnumbered olarak yapılandırmak için:

```
Router(config-if)#ipv6 unnumbered Fa0/0
```

IPv6 Static Routing

```
ipv6 route <ipv6prefix>/<prefix-length> [ <next-hop-IPv6-addr> | <interface-type-#> ] [AD#]
```

```
show ipv6 route [connected | local | static | rip | bgp | isis | ospf]
```

```
show ipv6 route <ipv6prefix>/<prefix-length>
```

```
Ör: ipv6 route 2001:db8:1:1::/64 3001::1
```

Static Default Route

```
RTR(config)# ipv6 route ::/0 3001::1
```

IPv6 RIP

RIPng etkinleştirme:

```
RTR(config)# ipv6 router rip ERDAL
```

Arayüzde (interface) RIP etkinleştirme:

```
RTR(config-if)# ipv6 rip ERDAL enable
```

Default Rotanın yaylanması

```
RTR(config)#ipv6 route ::/0 2001::1
```

```
RTR(config-if)# ipv6 rip ERDAL default-information originate
```

AD değerini değiştirme

```
RTR(config-rtr)# distance 50
```

Distribute List Kullanma

```
RTR(config-rtr)# distribute-list prefix-list PREFIX_LIST_ADI [in | out] FastEthernet 0/0
```

Offset-List Kullanma

```
RTR(config-rtr)# metric-offset 30
```

Poison-Reverse , Split-Horizon Devreye alma

```
RTR(config-rtr)# poison-reverse
```

```
RTR(config-rtr)# split-horizon
```

Birden Çok RIPng çalışlığında gruplandırılacak. Her grup için farklı UDP portu kullanılabilir. Aşağıdaki örnekte 521 portu kullanılmıştır.

```
RTR(config-rtr)# port 521 multicast-group FF02::9
```

RIP süreleri değiştirme

```
RTR(config-rtr)# timers <update> <expire> <holddown> <garbage-collect>
```

RIP Redistribute

```
RTR(config-rtr)# redistribute [ connected | isis | ospf | static | bgp | rip <TAG> ] [metric <metric>] [level-1 | level-1-2 | level-2] [route-map <NAME>]
```

DEBUG ve SHOW KOMUTLARI

```
RTR# show ipv6 route
```

```
RTR# show ipv6 rip [database] [next-hops]
```

```
RTR# show ipv6 protocols
```

```
RTR# debug ipv6 rip <interface>
```

```
RTR# debug ipv6 routing  
RTR# clear ipv6 rip ERDAL
```

IPv6 EIGRP

```
interface FastEthernet 0/0  
ipv6 enable  
ipv6 eigrp 10  
ipv6 bandwidth-percent eigrp <as-number> <percent>  
ipv6 summary-address eigrp <as-number> <ipv6-address> [admin-distance]  
ipv6 authentication mode eigrp <as-number> md5  
ipv6 authentication key-chain eigrp <as-number> <key-chain>  
!  
ipv6 router eigrp 10  
router-id 10.1.1.1  
stub [receive-only | connected | static | summary | redistributed]  
log-neighbor-changes  
log-neighbor-warnings [seconds]  
metric weights tos k1 k2 k3 k4 k5  
!  
show ipv6 eigrp interfaces  
show ipv6 eigrp neighbors detail  
show ipv6 eigrp topology  
show ipv6 eigrp traffic  
  
clear ipv6 eigrp [as-number] [neighbor [ipv6-address | interface-type interface-number]]  
  
debug eigrp fsm  
debug eigrp neighbor [siatimer] [static]  
debug eigrp packet  
debug eigrp transmit [ack] [build] [detail] [link] [packetize] [peerdown] [sia] [startup] [strange]  
debug ipv6 eigrp [as-number] [neighbor ipv6-address | notification | summary]
```

Yukarıdaki topoloji için EIGRP yapılandırması:

```
R1(config)#ipv6 unicast-routing      //ipv6 routing enable edildi  
R1(config)#ipv6 router eigrp 100  
R1(config-rtr)#no shutdown          // EIGRP'nin açılması gereklidir.  
R1(config-rtr)#router-id 1.1.1.1 //IPv4 formatında router-id yazılır  
//tanıtımı yapılacak arayüzde enable edilir  
R1(config)#interface Ethernet 0  
R1(config-if)#ipv6 eigrp 100  
  
R2(config)#ipv6 unicast-routing  
R2(config)#ipv6 router eigrp 100  
R2(config-rtr)#no shutdown  
R2(config-rtr)#router-id 2.2.2.2  
R2(config)#interface Ethernet 0  
R2(config-if)#ipv6 eigrp 100
```

```
R2(config)#interface Ethernet 1
```

```
R2(config-if)#ipv6 eigrp 100
```

IPv6 OSPF (OSPFv3)

Örnek:

```
interface Ethernet 0
```

```
  ipv6 address 2001:100:1::1/64
```

```
  ipv6 enable
```

```
  ipv6 ospf 100 area 0
```

```
interface Ethernet 1
```

```
  ipv6 address 2001:200:2::1/64
```

```
  ipv6 enable
```

```
  ipv6 ospf 100 area 1
```

```
  ipv6 router ospf 100
```

```
    router-id 10.1.1.1
```

```
    area 1 range 2001:200:FFFF:1::1/64
```

```
RTR(config)# ipv6 router ospf 1
```

```
RTR (config-rtr)# router-ID 1.1.1.1
```

```
RTR (config-rtr)# area <v4areaID> range <ipv6addr/length>
```

```
RTR (config)# interface ethernet 0
```

```
RTR (config-if)# ipv6 ospf 1 area 5
```

```
RTR (config-rtr)# redistribute [bgp | isis | rip | static]
```

```
R1(config)#ipv6 unicast-routing //ipv6 routing enable edildi
```

```
R1(config)#ipv6 router ospf 1
```

```
R1(config-rtr)#router-id 1.1.1.1 //IPv4 formatında router-id yazılır
```

```
//tanıtımı yapılacak arayüzde enable edilir
```

```
R1(config)#interface Ethernet 0
```

```
R1(config-if)#ipv6 ospf 1 area 0 //arayüz hangi areada ise yazılır.
```

```
R2(config)#ipv6 unicast-routing
```

```
R2(config)#ipv6 router ospf 1
```

```
R2(config-rtr)#router-id 2.2.2.2
```

```
R2(config)#interface Ethernet 0
```

```
R2(config-if)#ipv6 ospf 1 area 0
```

```
R2(config)#interface Ethernet 1
```

```
R2(config-if)#ipv6 ospf 1 area 10 //Eth1 portu area10'a dahil edildi
```

```
RTR(config)# ipv6 router ospf 1
```

```
RTR (config-rtr)# router-ID 1.1.1.1
```

```
RTR (config-rtr)# area <v4areaID> range <ipv6addr/length>
```

```
RTR (config)# interface ethernet 0
```

```
RTR (config-if)# ipv6 ospf <process-ID> area <v4areaID>
```

```
RTR (config-rtr)# redistribute [bgp | isis | rip | static]
```

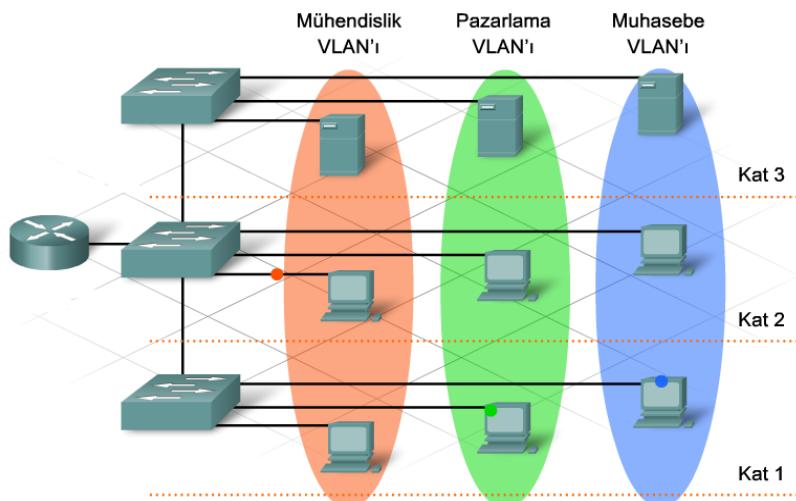
```

show ipv6 ospf <processID>
show ipv6 ospf database
show ipv6 ospf <processID> database link
show ipv6 ospf <processID> database prefix
show ipv6 ospf route ospf

```

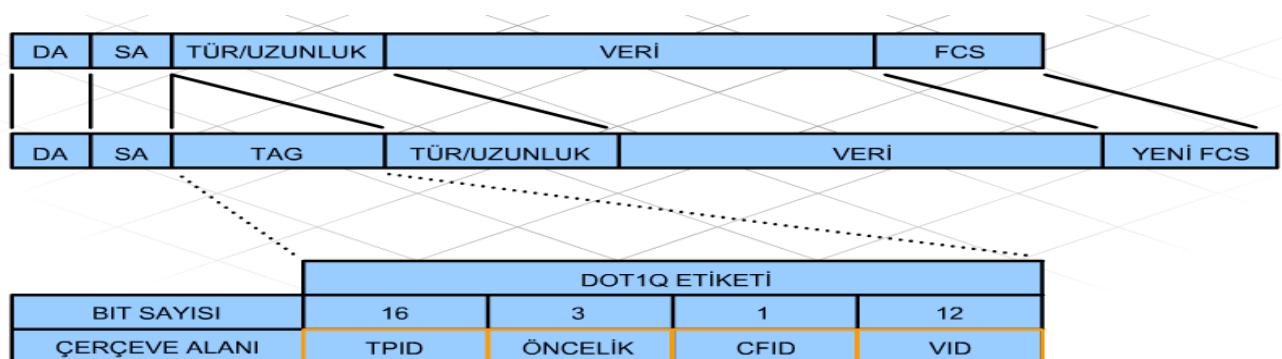
VLAN (Virtual LAN) YAPILANDIRMA

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_adı
Switch(config-vlan)#exit

```

Örnek: Numaraları 5 ve 10 olan iki vlan oluşturup, sırasıyla MUHASEBE ve PAZARLAMA olarak isimlendirelim. 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 yapılandırılmış.

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

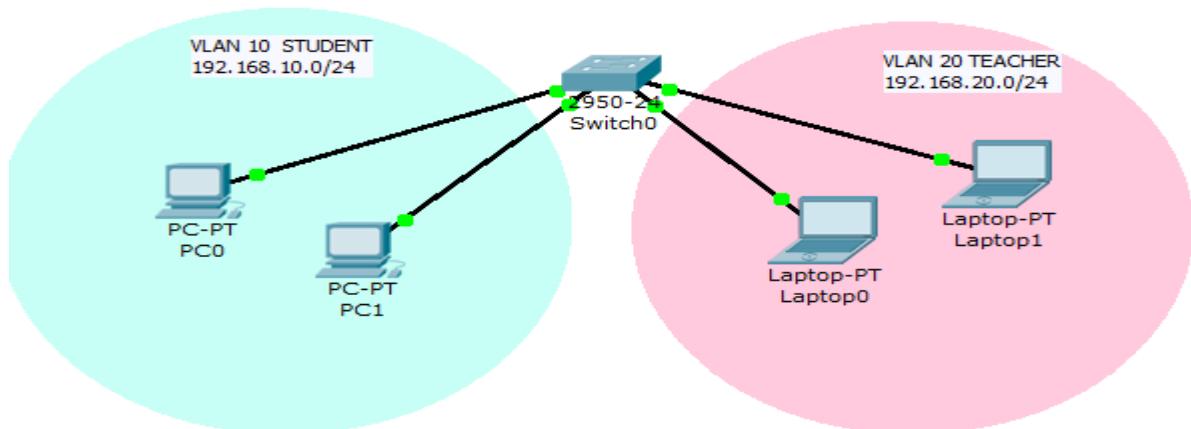
```
SW1(config-if-range)#exit  
SW1(config)#interface fastEthernet 0/24  
SW1(config-if)#switchport mode trunk
```

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

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

INTER-VLAN ROUTING

Switchlerde her VLAN farklı bir subnet içinde bulunması gerekmektedir. Bu sebeple birbirleri ile iletişime geçebilmeleri için Layer3 bir cihaza ihtiyaçları vardır. Aşağıdaki örnekte 10 ve 20 numaralı iki VLAN oluşturulmuştur.



Switch VLAN yapılandırması

```
SW(config)#vlan 10  
SW(config-vlan)#name STUDENT  
SW(config-vlan)#vlan 20  
SW(config-vlan)#name TEACHER  
SW(config-vlan)#exit
```

İlk 10 port VLAN 10 üyesi, sonraki 10 port ise VLAN 20 üyesi yapalım.

```
SW(config)#interface range fa0/1-10  
SW(config-if-range)#switchport access vlan 10  
SW(config)#interface range fa0/11-20  
SW(config-if-range)#switchport access vlan 20
```

Interface yapılandırılmışının doğruluğunu **show vlan** komutu ile doğrulayabiliriz.

```
SW#show vlan
```

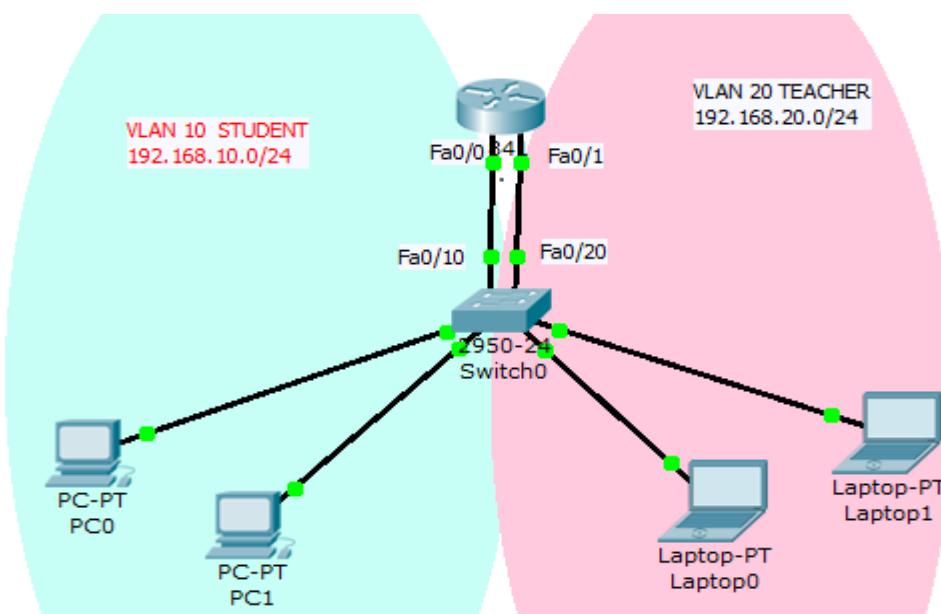
VLAN	Name	Status	Ports
1	default	active	Fa0/21, Fa0/22, Fa0/23, Fa0/24
10	STUDENT	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10
20	TEACHER	active	Fa0/11, Fa0/12, Fa0/13, Fa0/14 Fa0/15, Fa0/16, Fa0/17, Fa0/18 Fa0/19, Fa0/20

Bu durumda Fa0/1 – Fa0/10 portlarına bağlı PC ‘ler VLAN 10 STUDENT üyesidirler. Fa0/11-Fa0/20 arasındaki portlara bağlı olan PC’ler ise VLAN 20 TEACHER üyesidirler. Interface range komutu ile belirlenmeyen diğer portlar ise VLAN1 (Default VLAN) üyesidirler. Bu üç VLAN üyesi cihazlar birbirleri ile Layer-3 bir cihaz olmadan haberleşemezler.

VLAN 10 ile VLAN 20 arasındaki iletişimini sağlayabilmek için Router ekleyip, Router portlarından birini (**Router Fa0/0**) VLAN10 üyesi portlardan birine, diğer Router portunu da (**Router Fa0/1**) VLAN 20 üyesi portlardan birine bağlarız. Buna göre;

Router Fa0/0 ----Switch Fa0/10 (VLAN 10 için)

Router Fa0/1 ---- Switch Fa0/20 (VLAN 20 için) olacak şekilde yapılandırıralım.



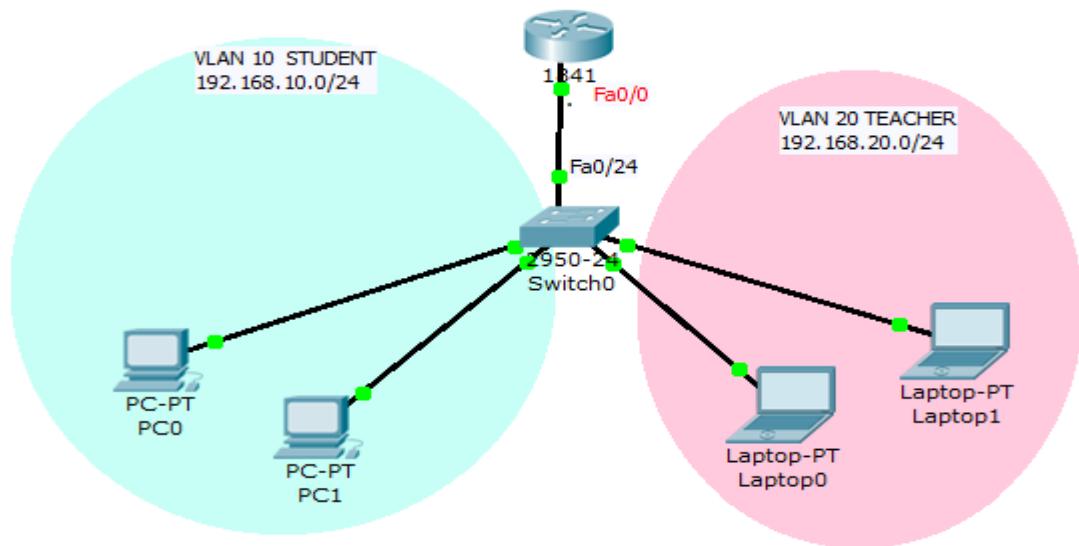
Bu durumda Router Fa0/0 portu VLAN 10 üyesi, Fa0/1 portu ise VLAN 20 üyesi olacaktır. O halde Fa0/0 portunun IP adresi 192.168.10.0/24 aralığında; Fa0/1 portunun IP adresi ise 192.168.20.0/24 aralığında olması gerekecektir. Buna göre Router yapılandırması:

```
Router(config)#interface fa0/0
Router(config-if)#ip address 192.168.10.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#interface fa0/1
Router(config-if)#ip address 192.168.20.1 255.255.255.0
Router(config-if)#no shutdown
```

Böyle bir durumda Router kendisine directly connected olarak 192.168.10.0 ve 192.168.20.0 ağlarını görecek ve her iki VLAN arasında iletişim gerçekleşecektir. Bu yöntemle **Traditional Inter-VLAN Routing** denir. Bu yöntemde her vlan için fiziksel olarak bir Router portu kullanılır. Yukarıdaki örnekte 10 ve 20 numaralı iki vlan için iki Router portu (Fa0/0 ve Fa0/1) kullanılmıştır. Vlan sayısı arttıkça maliyet de artacaktır.

ROUTER ON STICK INTERVLAN ROUTING

Geleneksel Inter-Vlan routing yöntemindeki dezavantajdan dolayı bu yöntemde Router üzerinde sadece bir port kullanılacaktır. Buna göre Switchten Routera sadece bir bağlantı yapılacak (**Router Fa0/0---Switch Fa0/24**) ve her iki vlan'ın de buradan geçmesi sağlanacaktır. Yine yukarıdaki önekten yola çıkarak topolojiyi aşağıdaki gibi yapılandırıralım.



Bu yapıda Switch Fa0/24 portu üzerinden hem VLAN 10 hem de VLAN 20 geçecektir. O halde bu portun VLANlardan bağımsız olarak herhangi bir VLAN üyesi olmaması gereklidir. Bu tür birden çok VLAN'ın üzerinden geçebileceğii portlara TRUNK port denir. **Trunk** portlarından varsayılanda tüm Vlanlar geçer. Yukarıdaki önekte sadece VLAN 10 ve VLAN 20 değil, diğer portların dahil olduğu VLAN 1 de geçecektir.

Switch fa0/24 portunu trunk olarak yapılandırıralım.

```

SW(config)#interface fa0/24
SW(config-if)#switchport mode trunk
    
```

Not: 2950 ve 2960 switchler, trunk kapsülleme türü olarak sadece 802.1q (dot1q) standardını destekler.

Ancak burada hemen akla şu soru gelmektedir. Bir Router portuna farklı ağlardan iki adres verilemediğine göre Router Fa0/0 portuna hangi IP adresi verilir? Vlan 10'a ait bir adres mi, yoksa Vlan 20'ye ait bir adres mi?

Bu durumda Router Fa0/0 fiziksel portu için mantıksal olarak Vlan sayısı kadar alt-arayüzler (**subinterface**) oluşturacağız. Oluşturacağımız her alt-arayüz, fiziksel arayüzler gibi davranış olacaktır. Fiziksel arayüz açık ve IP adresinin olmaması gereklidir.

Buna göre Router yapılandırması:

```
Router(config)#interface fa0/0  
Router(config-if)#no shutdown  
Router(config-if)#no ip address
```

Router üzerinde VLAN 10 için subinterface oluşturralım.

```
Router(config)#interface fa0/0.10
```

Burada **10** ile bir subinterface oluşturulmuştur. 10 yerine 1 ile 4294967295 arasında herhangi bir değer verilebilir. Ancak hangi VLAN ile ilişkili olduğunu ilk bakışta görebilmek için VLAN numarasının kullanılması daha mantıklıdır.

Not: Fa0/0.0 subinterface kullanılmaması demektir.

Subinterface oluşturulduktan sonra bunun hangi vlan ile ilişkili olduğunu aşağıdaki gibi bir yapılandırma ile belirtmek gereklidir. Ardından bağlı olunan subnet aralığından bir IP adresi verilir. (Bu sıralama önemlidir)

```
Router(config-subif)#encapsulation dot1q 10 //burda 10 vlan numarasıdır  
Router(config-subif)#ip address 192.168.10.1 255.255.255.0
```

Şimdi de Router üzerinde VLAN 20 için subinterface oluşturup IP adresini verelim.

```
Router(config)#interface fa0/0.20  
Router(config-subif)#encapsulation dot1q 20 //burda 20 vlan numarasıdır  
Router(config-subif)#ip address 192.168.20.1 255.255.255.0
```

Yapilandırmayı doğrulamak için Router üzerinde show ip interface brief komutu kullanılabilir.

Router#show ip interface brief	Interface	IP-Address	OK?	Method	Status	Protocol
	FastEthernet0/0	unassigned	YES	manual	up	up
	FastEthernet0/0.10	192.168.10.1	YES	manual	up	up
	FastEthernet0/0.20	unassigned	YES	unset	up	up
	FastEthernet0/1	192.168.20.1	YES	manual	up	down
	Vlan1	unassigned	YES	unset	administratively down	down
	Router#					

Göründüğü gibi Fa0/1.0 ve Fa0/0.20 diğer fiziksel interfaceler ile beraber görüntülenecektir.

Router routing tablosunda 192.168.10.0/24 ve 192.168.20.0/24 ağları directly connected olarak görünüğinden VLAN'lar arası iletişim gerçekleşecektir. Burda unutulmaması gereken, her vlan için bir subinterface oluşturulmasıdır. Yani yukarıdaki örneğe göre VLAN1 için yapılandırma yapılmadığına göre VLAN1 ile iletişim sözkonusu değildir.

DTP (Dynamic Trunking Protocol)

Cisco tarafından geliştirilen, birbirlerine bağlı iki switch arasındaki portların dinamik olarak yapılandırılmasını (trunk / Access) sağlayan Layer 2 bir protokoldür. Her 30 saniyede bir gönderilen Hello paketleri ile haberleşir. Timeout süresi 300 saniyedir.

Switchlerde portların yönetimsel durumları Dynamic auto, Dynamic Desirable, Access ya da Trunk olabilir. Cisco switchlerde portlar default olara Dynamic Auto moddadır. Aşağıdaki belirtilen komut ile portun modu değiştirilebilir.

```
Switch(config)#interface fa0/24
Switch(config-if)#switchport mode ?
    access  Set trunking mode to ACCESS unconditionally
    dynamic Set trunking mode to dynamically negotiate access or trunk mode
    trunk   Set trunking mode to TRUNK unconditionally
```

İletişimin sağlıklı çalışabilmesi için birbirlerine bağlı iki switchin ilgili portlarının aynı işlevsel modda (operational mode) olması gereklidir.



```
Switch#show interface switchport
.....
Name: Fa0/24
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
...
```

DTP protokolü ile portların bu durumları dynamic olarak yapılandırılır. Yukarıdaki örnekte, birbirlerine Fa0/24 portları ile bağlı iki switchin de Yönetimsel Modu (**Administrative Mode**) default olarak **Dynamic Auto** 'dur. Bu durumda her iki port da **Access** olarak çalışacaktır. Aşağıdaki örnekte gösterildiği gibi switchlerden birinin Fa0/24 portu **Trunk** olarak yapılandırılırsa, bu durumda diğer port **Dynamic Auto** modda olacak ve kendisini **Trunk** işlevsel modunda geçirecektir.

```
Switch0(config-if)#interface fa0/24
Switch0(config-if)#switchport mode trunk
Switch0(config-if)#end
```

```

Switch0#show interface switchport fa0/24
Name: Fa0/24
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)

```

Komutun yazıldığı switchteki Yönetimsel ve işlevsel modlar Trunk olarak görünecektir. Karşı switchte bu komut yazılmadığı için Yönetimsek Mod Dynamic Auto olacak, andak DTP protokolü sayesinde portun işlevsel modu trunk olacaktır.

```

Switch1#show interface switchport
.....
Name: Fa0/24
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
.....

```

Yukarıdaki örneğten de anlaşılacağı gibi portların işlevsel modları DTP protokolü sayesinde yapılan anlaşma (**negotiation**) ile belirlenir.

Aşağıdaki tabloda portların durumlarına göre işlevsel modları belirtilmiştir.

	Access	Trunk	Dynamic Auto	Dynamic Desirable
Access	Her iki uç access	HATALI YAPILANDIRMA	Her iki uç access	Her iki uç trunk
Trunk	HATALI	Her iki uç trunk	Her iki uç trunk	Her iki uç trunk
Dynamic Auto	Her iki uç Access	Her iki uç trunk	Her iki uç Access	Her iki uç trunk
Dynamic Desirable	Her iki uç trunk	Her iki uç trunk	Her iki uç trunk	Her iki uç trunk

Güvenlik açısından önerilen, DTP protokolünün kapatılıp, portların işlevsel modlarının komut olarak yazılmıştır. DTP protokolünü her iki switchte ilgili portta kapatmak için aşağıdaki gibi bir yapılandırma gereklidir.

```

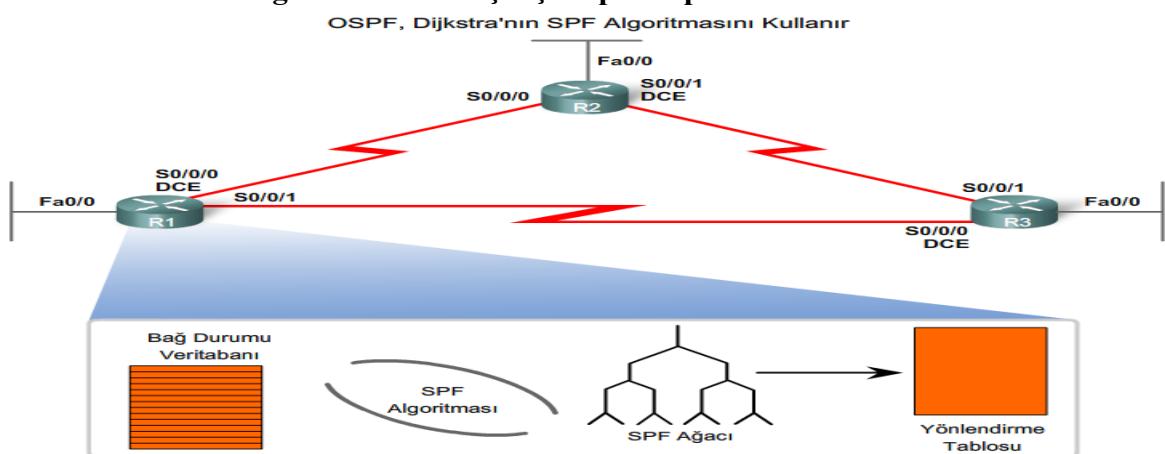
Switch0(config-if)#interface fastEthernet 0/24
Switch0(config-if)#switchport mode trunk
Switch0(config-if)#switchport nonegotiate
...
Switch1(config-if)#interface fastEthernet 0/24
Switch1(config-if)#switchport mode trunk
Switch1(config-if)#switchport nonegotiate
Switch1(config-if)#end
Switch1#show interface switchport
...
Name: Fa0/24
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Off

```

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 değişikiliğe 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 leri 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 routerın 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

2-Way

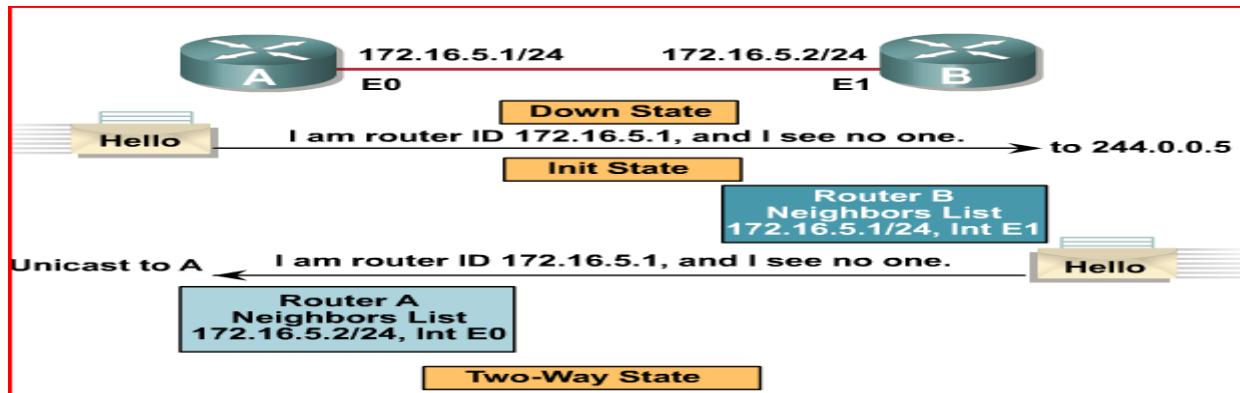
Exstart

Exchange

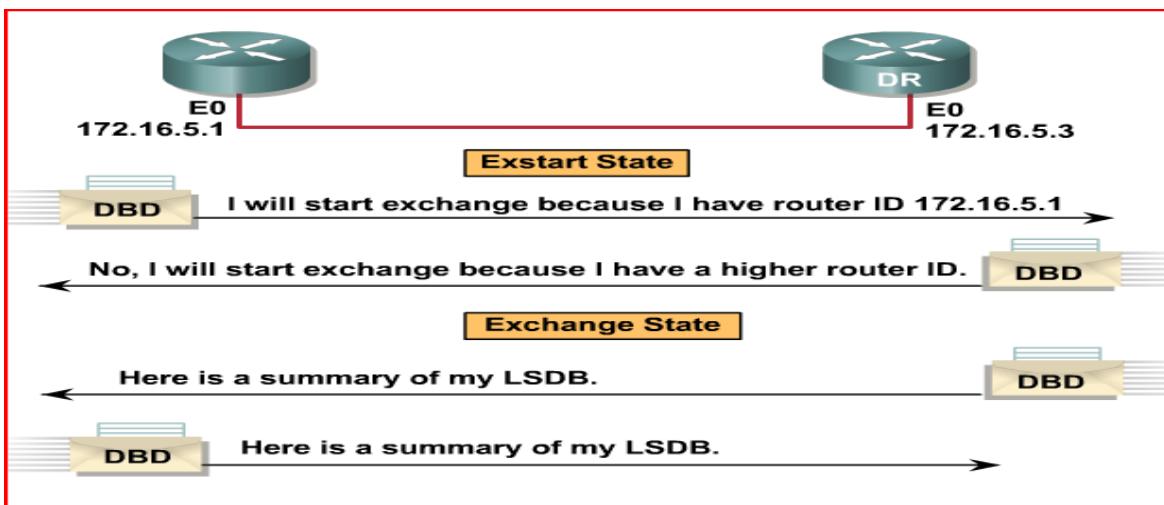
Loading

Full

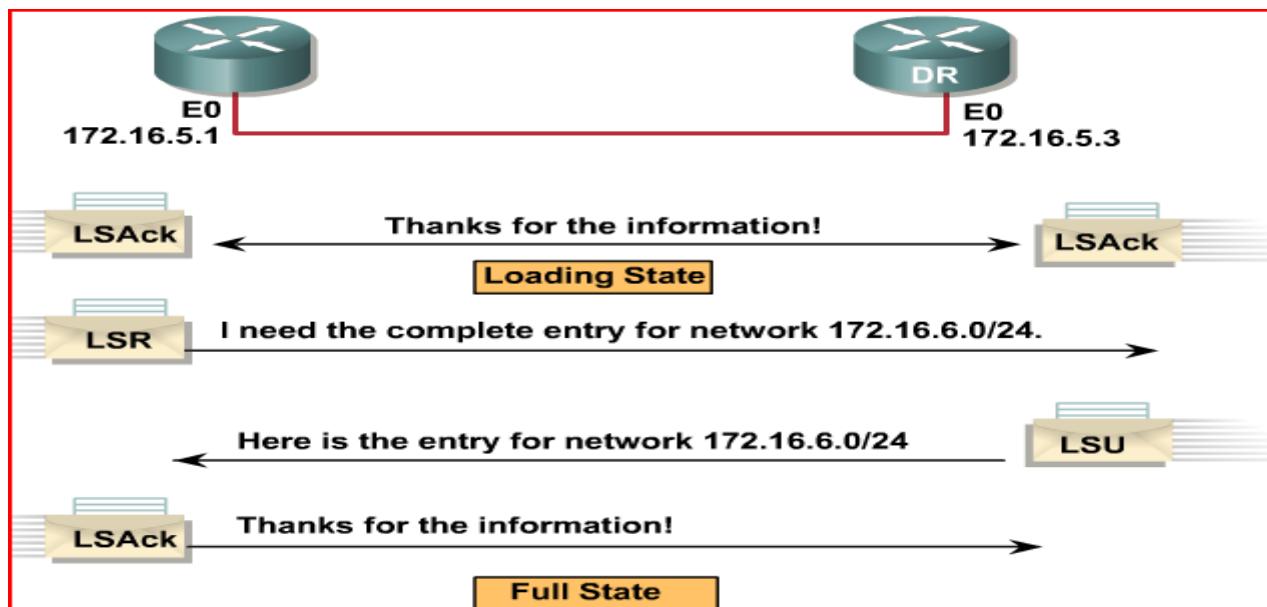
DOWN – INIT ve TWO-WAY DURUMLARI:



EXSTART ve EXCHANGE DURUMLARI :



LOADING ve FULL DURUMLARI :



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

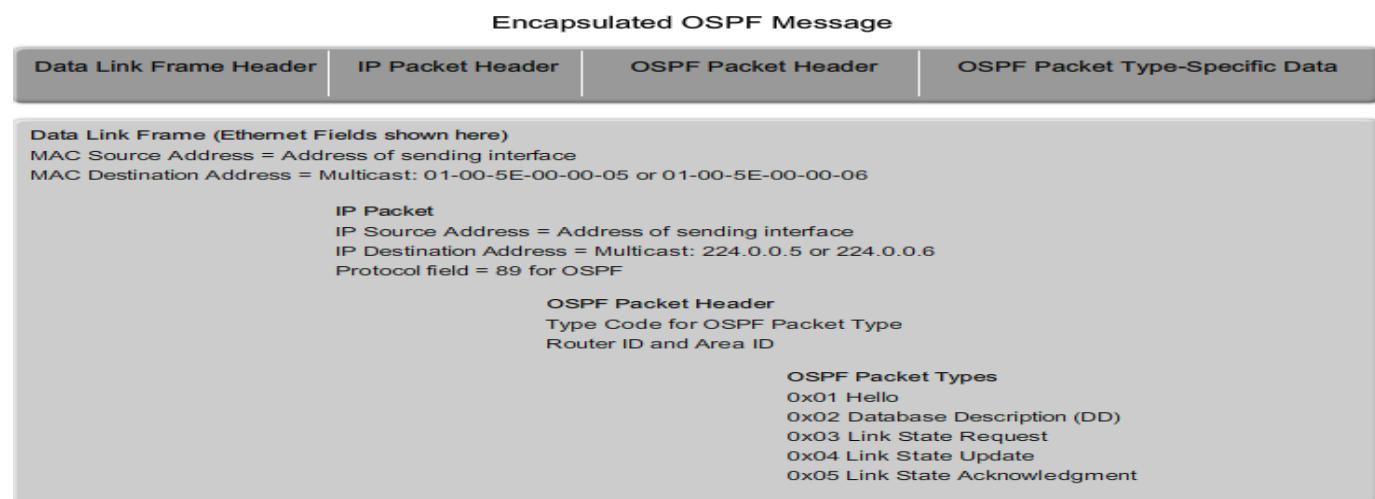
0x01: Hello - OSPF routerlar arasında bitişikliğin (adjacency) oluşturulması ve devam ettirilmesini sağlayan paketleridir.

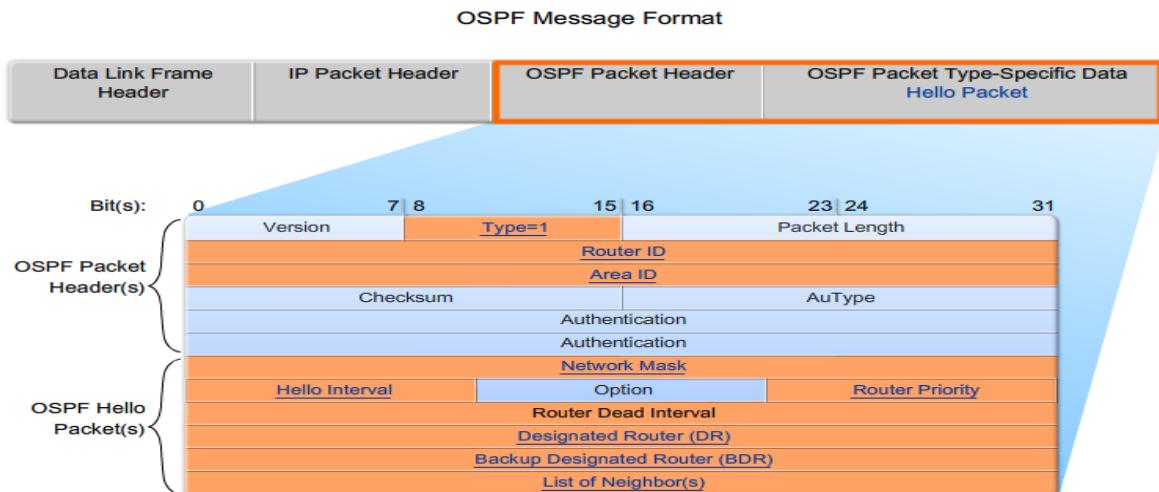
0x02. DBD – Routerlar arasında database senkronizasyonunu sağlar. Bu paketler gönderen routerın özet link-state bilgilerini içerir. Alıcı router bu paketler ile kendi link-state database bilgilerini karşılaştırır.

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.





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: Komsu OSPF routerlarının listesi

- Router, Link state bilgilerini yayınlamadan 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**, komuşu 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 routelara bildirilir.
 - Multiaccess networklerde ağıda oluşan değişiklikleri diğer routelara bildirmek için **Designated Router (DR)** ve **Backup Designated Router (BDR)** seçilir.
 - Her OSPF router diğer routelardan gelen LSA lar ile kendi Link State veritabanını oluşturur. Dijikstra algoritması kullanılarak SPF ağacı 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 routerların 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 arayüzlerin varsayılan bandwith değeri, T1 değerine eşittir (1,544 Mbps) OSPF'in doğru bir biçimde cost hesabı yapabilmesi için doğru bw değeri girilmelidir.

R(config – if)# bandwith 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.

DR SECİMİ,

Coklu erişimli ağlarda (BMA, NBMA) komşuluk sayısı artacaktır. Bu durumdan kurtulmak için bu tür ağlarda, DR ve BDR seçilir, LSA bilgileri DR'a ve BDR'a gider, DR ve BDR bu bilgileri DROther'lara iletir.

Point –to – Point networklerde DR BDR seçimi yapılmaz.

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 routerın 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)

```
0*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 – bandwidth 10000 //Mbps cinsinden değeri

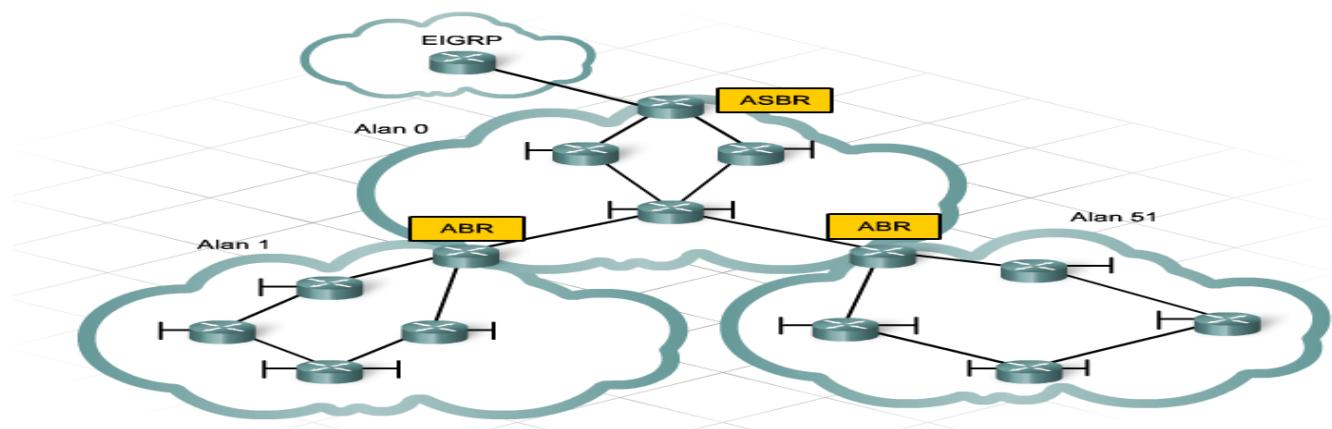
OSPF INTERVAL DEĞERLERİNİ DEĞİŞTİRMEK:

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 gözüktür.
- Bir bağ arızalandığında, bağ hakkında bilgi sahibi yönlendirici bilgiyi 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 bilgiyi, 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) 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 bağlantılı olmak zorundadır.



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:

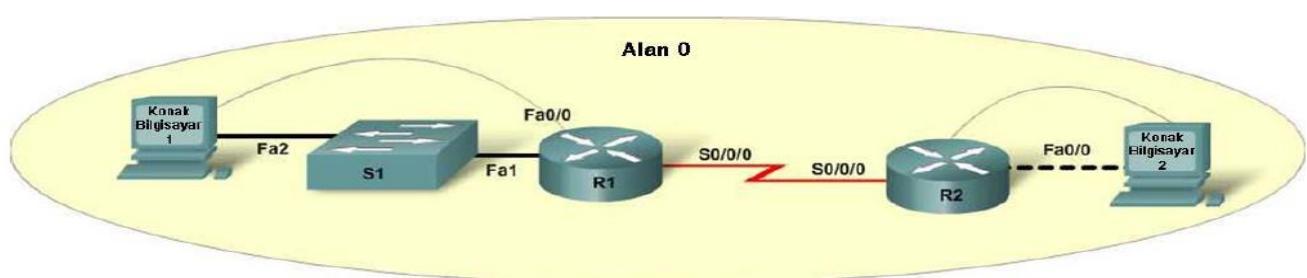
```
R(Config - router)# area 0 range 192.168.0.0 255.255.252.0
```

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 sayıyı 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 KOMUTLARI

```
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 omitted>

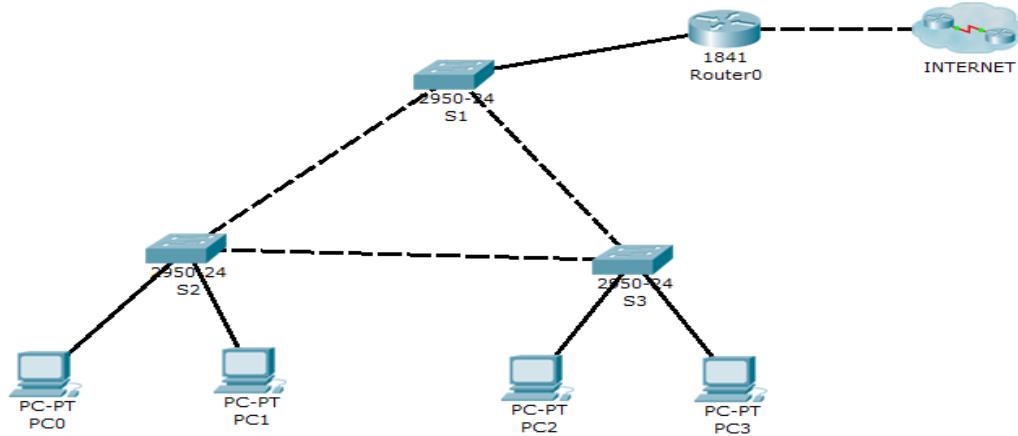
Gateway of last resort is 192.168.10.2 to network 0.0.0.0

  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
```

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

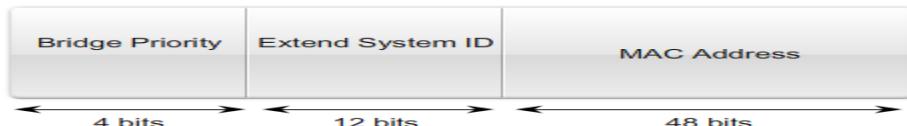
Yedekli Anahtarlamalı Topolojiler Oluşturma

Yedekli olarak Switchler arasında bağlantılar oluşturulduğunda döngüsel yapıdan dolayı **Broadcast Storm**, **Duplicate Unicast Frame** ve **MAC Adres Tablosu 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 bloklanmaz.

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

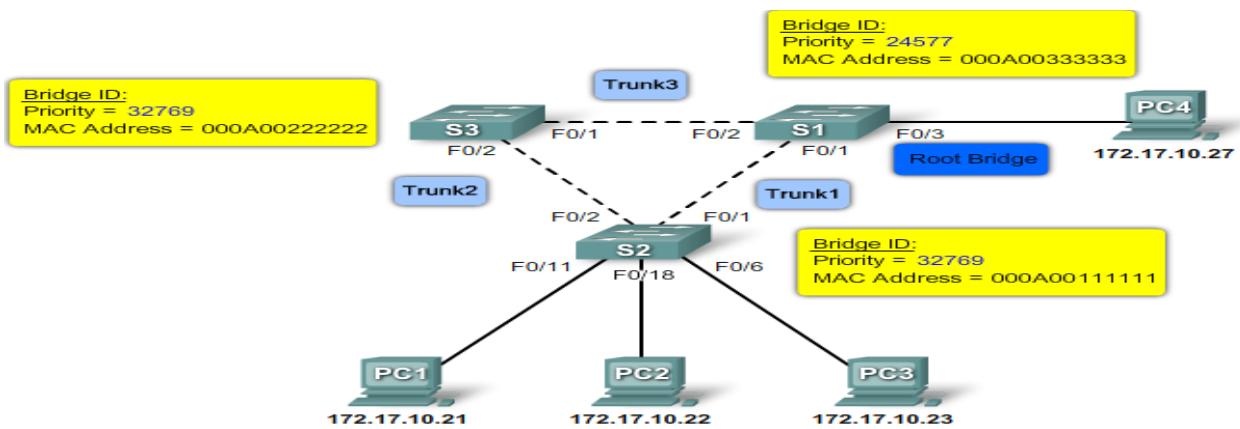


BID değerlerinin karşılaştırılması sonucu, **en düşük** BID değerine sahip olan switch ortamda **ROOT BRIDGE** olarak seçilir. Default olarak switchler 32768 Priority değerine sahiptir. Burada Extend System ID, VLAN numarası olarak düşünülebilir.

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 adrese 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. (**Peki, hangi interface MAC adresi???**)

SWITCH PORT DURUMLARI

STP çalışan her switche aşağıdaki beş durum gerçekleşir.

Blocking: STP'ye dahil olan ancak frame iletmeyen bloklanmış porttur. Port BPDU alır ve gönderir. Default olarak port 20 saniye bu durumda kalır. (MAX AGE)

Listening: Gelen BPDU değerine göre switchin forwarding duruma geçebileceği belirlenmiştir. BPDU alır ve gönderir. Karşı switch'e kendisinin aktif topolojide yer alacağını belirtir. Port bu durumda 15 saniye geçirir. (FORWARD DELAY)

Learning: Bu modda port MAC tablosunu oluşturabilmek için öğrenme moduna geçer. BPDU alır ve gönderir. 15 saniye bu sürede kalır. (FORWARD DELAY)

Forwarding: Portfonksiyonel olarak çalışır. BPDU alır ve gönderir.

Disabled: Aslında STP durumu değildir. Portun kapalı olması anlamına gelir.

SWITCH PRIORITY DEĞERİNİ DEĞİŞTİRME

Priority değeri değiştirilerek istenilen switchin root bridge olması sağlanabilir.

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

veya

```
SW(config)# spanning-tree vlan 1 root primary
```

Ayrıca aşağıdaki gibi bir yazım da mümkündür.

```
SW(config)# spanning-tree vlan 5,100-200 priority 4096
```

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 yayarlar. (**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.

MessageType: BPDU mesaj türü, (TCN veya Conf.)

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

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

CostofPath: **Root'a giden yolun** maliyet değeridir.

BridgeID : Switchin kendi BID değeri.

PortID : STP ye dahil portun değeridir.

MessageAge : Root'dan gelen konfigurasyon mesajından itibaren geçen süre.

MaxAge : 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.

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

ForwardDelay : 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)

İki tür BPDU mesajı vardır. Configuration BPDU ve Topology Change Notification (TCN). Root Bridge seçildikten sonra Configuration BPDU yalnızca Root tarafından gönderilir.

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)
```

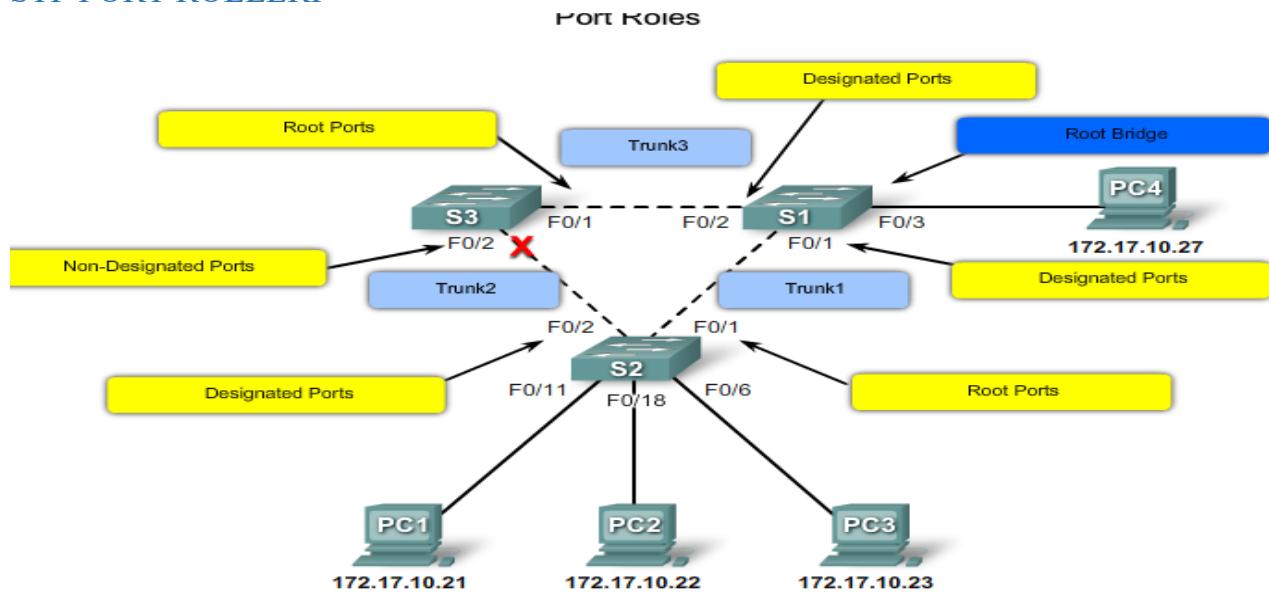
```
Address 0007.EC92.2774
```

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

```
Aging Time 20
```

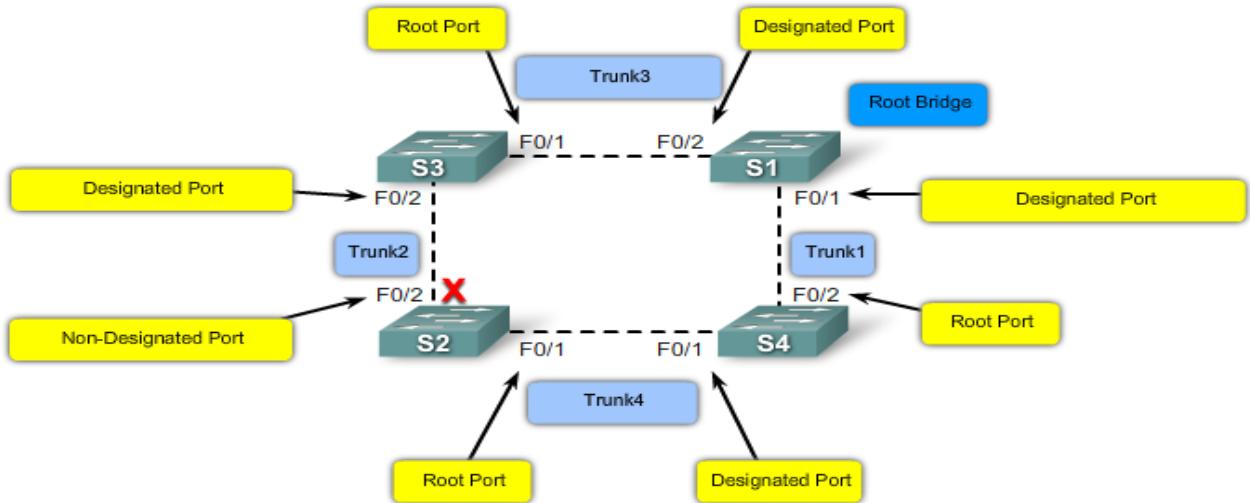
Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/2	Altn	BLK	19	128.2	P2p
Fa0/1	Root	FWD	19	128.1	P2p

STP PORT ROLLERİ



BPU mesajları sonunda Root Bridge belirlenir ve diğer switchler bu roota ulaşmak yolları bulur. Yulardaki 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. Bu portlar, RootBridge'den gelen BPDU'ları diğer (*downwards*) switchlere gönderir. ROOT BRIDGE'in tüm portları **DESIGNATED PORT** olarak belirlenir.

** Yukardaki 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 karşılaştıracak ve S1'in root olduğunu kabul edecek, path cost olarak 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 (root path cost *) 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, diğeri ise bloklanacaktır. Port priority değerleri eşit ise bu durumda Port ID değerine bakacaktır. Port ID değeri düşük olan ROOT PORT olacaktır.

* **Root Path Cost:** Root bridge giden yolun toplam maliyetidir. Bu değer ile hangi portun root olacağı belirlenir. Her switchte ayrıca **path cost** denen ve linkin türüne göre değişen bir değer vardır (Table 7-3). BPDU içinde path cost değeri değil, root path cost değeri taşınır.

Root Path Cost değeri aşağıdaki adımlarla belirlenir:

- 1- Root bridge BPDU mesajında root path cost değerini 0 olarak belirler.
- 2- BPDU alan ilk switch bu değere gelen portun cost değerini (Örnek Fa için 19) ekler ve yayınlar.
- 3- İkinci switchten gelen BPDU içinde path cost değerine (19) geldiği portun cost değerini (Örnek Fa için 19) ekler. Böylece root path cost değeri 38 olur. Bu switchte başka yönden gelen

BPDU içinde de root path cost değerleri bulunacaktır. Bu değerler karşılaştırılarak düşük costlu port root port seçilir.

Table 7-3 STP Path Cost

Link Bandwidth	Old STP Cost	New STP Cost
4 Mbps	250	250
10 Mbps	100	100
16 Mbps	63	62
45 Mbps	22	39
100 Mbps	10	19
155 Mbps	6	14
622 Mbps	2	6
1 Gbps	1	4
10 Gbps	0	2

Portun cost değerleri aşağıdaki gibi değiştirilebilir.

```
S(config)#interface fa 0/1
S(config - if)# spanning-tree cost 25 (1 ile 200 000 000 arasında olabilir)

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

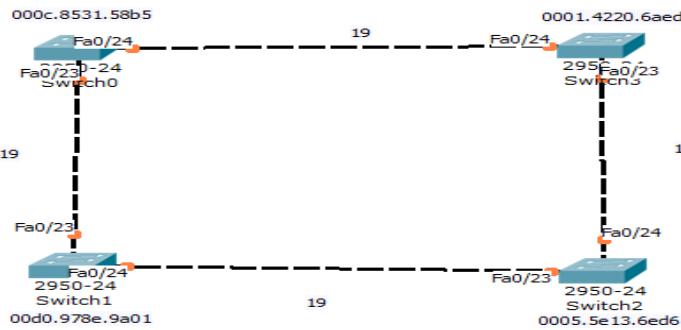
Buradaki “128.1” için: 128= Priority, 1 ise port numarasıdır. Fa0/1 için 1, Fa0/4 için 4...vs. Port Priority değeri değiştirilebilir. Default 128.

```
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
```

BLOKLANACAK PORTUN BELİRLENMESİ

- 1- Önce Root Bridge belirlenir
 - 2- Tüm switchler için roota giden yolu toplam maliyetlerine göre root port belirlenir.
 - 3- Her segment için Designated Port belirlenir.
- a- DP belirlenirken düşük root path cost değerine sahip cihazın portu DP olur.

Örnek:



- 1- Her switch default priority bulunduğuundan düşük MAC adrese sahip switch root olacaktır.

SWITCH 3

- 2- Her switch toplam cost değerinde göre en düşük cost değerine sahip portu **root port** olarak belirler.

SWITCH0 :

$$\text{Fa0/24} = 19 \quad (\text{Fa0/24 root port})$$

$$\text{Fa0/23} = 57$$

SWITCH2:

$$\text{Fa0/24} = 19 * (\text{Fa0/24 root port})$$

$$\text{Fa0/23} = 57$$

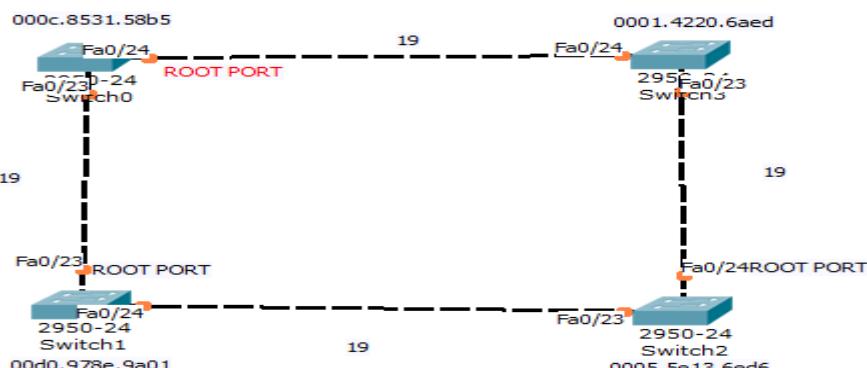
SWITCH1:

$$\text{Fa0/23} = 38$$

$$\text{Fa0/24} = 38$$

Switch1 için root path cost değerleri eşit çıktığinden, Sender BridgeID değerine bakılacaktır.

- a. Fa0/23 Neighbor =Sw0 ; Sw0 ID= 000cxxxx
 - b. Fa0/24 Neighbor = Sw2; Sw2 ID=0005xxxx (daha düşük) O halde **Fa0/24 Root Port**
- Bu portları şekilde işaretleyelim.



- 3- Her segment için Designated Port seçimi (düşük path cost)

S1- S2 arasında; S2 cost = 19; S1 cost=38 o halde **S2 Fa0/23 DESIGNATED**

S1-S0 arasında ; S1 cost=38; S0 cost=19 o halde **S0 Fa0/23 DESIGNATED**

S0-S3 arasında; S0 cost=19; S3 cost=0 o halde **S3 Fa0/24 DESIGNATED**

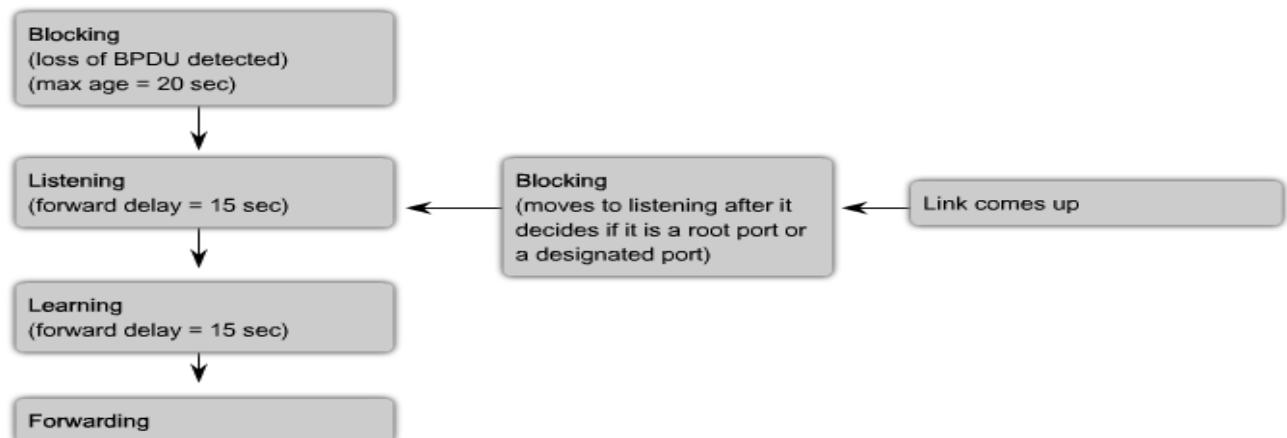
S2-S3 arasında; S2 cost=19; S3 cost=0 o halde **S3 Fa0/23 DESIGNATED**

DESIGNATED ya da ROOT olmayan port bloklanır. **S1 Fa0/23 BLOCKED PORT**

POR DURUMLARI (PORT STATES)

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



STP TIMER DEĞİŞTİRME

```
Switch(config)# spanning-tree [vlan vlan-id] hello-time saniye (1-10)
Switch(config)# spanning-tree [vlan vlan-id] forward-time saniye
Switch(config)# spanning-tree [vlan vlan-id] max-age saniye
```

Hello Time, default olarak 2 sn olup, BPDU periyodunu belirler. **1 ile 10** saniye arasında değişebilir.

Forward-Time, default olarak 15 sn dir. 4-30 sn değerleri alabilir.

Max-Age, default **20** sn, **6-40** sn değerlerini alabilir.

Ayrıca aşağıdaki gibi bir yapılandırma ile STP timer süreleri otomatik olarak da yapılandırılabilir.

```
Switch(config)# spanning-tree vlan 100 root primary diameter 3 hello-time 1
```

Bu durumda STP timer;

```

Switch# show spanning-tree vlan 100
VLAN0100
    Spanning tree enabled protocol ieee
        Root ID      Priority    100
                    Address     000c.8554.9a80
                    This bridge is the root
        Hello Time 1 sec Max Age 7 sec Forward Delay 5 sec

```

LINK CONVERGENCE

Link down olduğunda STP nin daha hızlı converge olması için bir takım özellikler vardır bunlar:

PortFast: Access portların hızlı bir şekilde açılmasını sağlar.

Normalde bir uç cihaz switch'e bağlandığında Blocking modundan Forwarding moduna geçmesi için 30 sn bekler. (15 sn, Listening-Learning + 15 sn Learning – Forwarding)

Ayrıca EtherChannel PAgP uygulamasında extra 20 sn daha bekler. (Toplam 50 sn)

PortFast uygulanacak portlar, loop oluşturmayan portlar olması gereklidir. Herhangi bir porta PortFast uygulamak için;

```
Sw(config-if)#spanning-tree portfast
```

Tüm portlarda portfast uygulamak için;

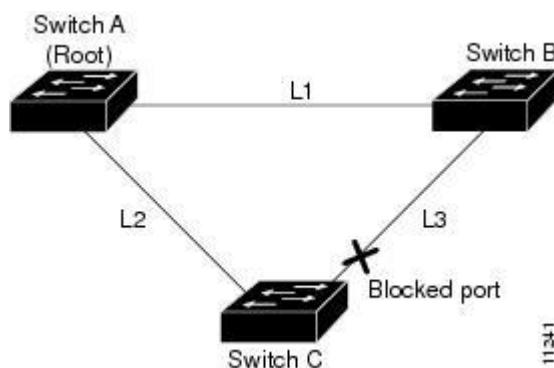
```
Sw(config)#spanning-tree portfast default
```

Yapilandırma doğrulamak için;

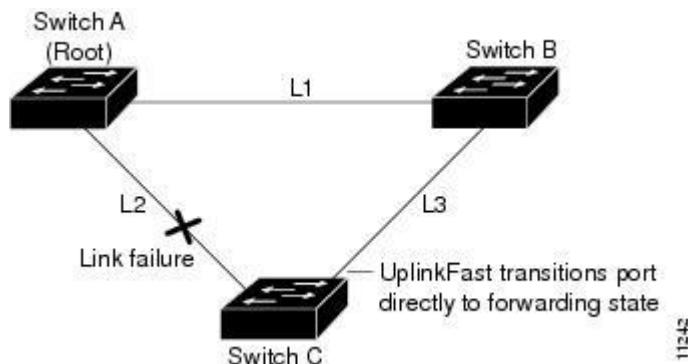
```
Sw# show spanning-tree interface fastethernet 0/1 portfast
```

UplinkFast:

Access katmanından Distribution katmanına çift uplink bağlantısı esnasında hızlı failover sağlar. Normalde UpLink portlarından biri blocking moddadır. Aktif olan port down olduğunda gecikme olmadan diğer portun Forwarding duruma geçmesi sağlanır. Yapılandırıldıgında tüm VLAN'ler için geçerli olur. Root Bridge üzerinde kullanılamaz.



Yukarıdaki şekilde herhangi bir link-failure olmadığını farzediyoruz. SwitchA, Root Bridge'dir. SwitchB ile SwitchC arasında L3 linkinde SwitchC portu bloklanmış moddadır.



Eğer SwitchC, L2 hattı üzerinde bir link-failure algılsarsa (direct link failure), UplinkFast, L3 üzerindeki bloklanmış portu açıp Listening ve Learning durumlarını beklemeden forwarding duruma geçirir (Yaklaşık 5sn).

UplinkFast, uygulanan switch üzerinde bir takım değişiklikler yapar. UplinkFast yapılandırılan bir switch'in root olmaması için Priority değeri otomatik olarak **49152** olacak şekilde artar. Böylece root olmayıp, diğer switchler için root bridge'e ulaşmak için transit switch olması önlenmiş olur.

Ayrıca tüm switch portlarının **cost** değeri **3000** artırılır. Böylece switchin Root olmaması amaçlanır.

Yapilandırma için ;

```
SW(config)# spanning-tree uplinkfast [max-update-rate pkts-per-second]
```

Uplink down olduğunda bu porttan öğrenilen MAC adres bilgileri yeni porta transfer edilir ve local switch bu bilgiyi kendisine bağlı switchlere de öğretecek şekilde (**Dest. MAC 0100.0ccd.ccd**) multicast yayın yapar. MAC tablosundaki kayıtlar Source MAC olarak kullanılır.

Max-update-rate parameteresi default olarak **150 pps** dir. Saniyede gönderilecek multicast paket sayısını ifade eder. Bu değer 0 ile 65 535 arasında yapılandırılabilir.

Yapilandırmayı doğrulamak için;

```
SW# show spanning-tree uplinkfast
UplinkFast is enabled
Station update rate set to 150 packets/sec.
UplinkFast statistics
Number of transitions via uplinkFast (all VLANs) : 2
Number of proxy multicast addresses transmitted (all VLANs) : 52
```

Name Interface List

VLAN0001 Gi0/1(fwd)

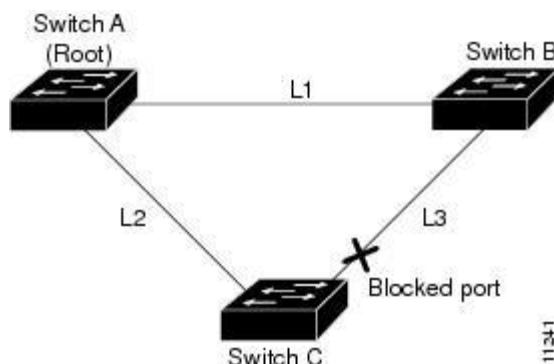
VLAN0010 Gi0/1(fwd)

BackboneFast:

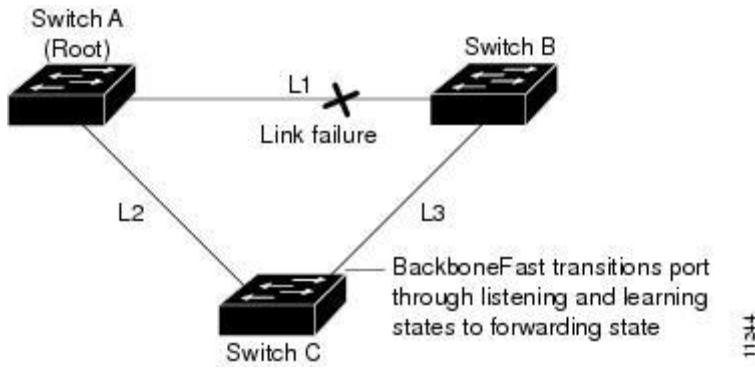
Core katmanında STP topoloji değişikliğinden sonra hızlı converge sağlar. Core katmanda *indirect link failure* olduğunda (*direk bağlı olmayan bir linkin down olması*) alternatif bir yol olup olmadığını kontrol eder. Eğer blocked ya da designated porttan daha düşük değerlikli BPDU alınırsa, root portun down olduğu anlaşılır. Normalde switch Max-Age süresi boyunca bekler. Ancak Backbone fast durumunda bu durum aşağıdaki gibi işler.

- Daha düşük değerlikli BPDU Block Porttan gelirse, root port ve diğer blocked portların alternatif port olur.
- Düşük değerlikli BPDU root porttan gelirse, tüm bloklanmış portlar root bridge için alternatif port olacaktır.
- Eğer düşük değerlikli BPDU root porttan geliyorsa ve herhangi bir port bloklanmamışsa switch root bridge giden yolu down olduğu farzedilir. Bu durumda, switch kendisini root bridge ilan edecektir.

Root Link Query (RLQ) protocol sayesinde alternatif yol bulunur.



Yukarıdaki topolojide linkler çalışmaktadır. SwitchA root bridge'dır. SwitchC ile SwitchB arasındaki L3 hattında SwitchC'ye ait port blokolanmıştır.



L1 fail olursa (indirect link failure), Switch C bunu algılayacaktır. Çünkü SwitchB'nin artık root'a ulaşacak portu değişmiştir. Buna göre BPDU paketi Switch C'ye bloklanmış porttan gelecektir. BackboneFast sayesinde SwitchC, Maximum-Age süresini (Def=20 sn) beklemeden Listening moda geçecektir. Bu süreç yaklaşık 30 sn dir.

Tüm switchlerde aşağıdaki gibi bir yapılandırma gereklidir

```
SW(config)# spanning-tree backbonefast
```

Yapılandırma doğrulama için;

```
SW# show spanning-tree backbonefast
BackboneFast is enabled
```

BPDU Filtering

BPDU filtering, ağa döngü olmasını önleyen diğer bir yöntemdir. Kullanım yerine göre amaç farklılık göstermekle beraber global config modunda ve interface modunda yazılabilir.

Global Config modunda yazıldığında;

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

Bu durumda portfast arayüze BPDU gelirse, PortFast durumunu yitirir.

Interface modunda yazıldığındá ise;

```
SW(config-if)# spanning-tree bpdufilter enable
```

Bu durumda ise BPDU alınmasını veya gönderilmesini önler.

STP GÜVENLİĞİ ve LOOP ÖNLEME OPTİMİZASYONU

Sahte Root Bridge isteklerini ve STP ataklarına karşı korunmak için **BPDU Guard** ve **Root Guard** olmak üzere iki tür güvenlik yapısından söz edilebilir.

BPDU Guard

Belirlenen portlardan herhangi bir BPDU mesajı alındığında bu portları err-disable durumuna düşürür. Dolayısıyla PortFast portlara yani Access portlama uygulamak gereklidir.

Tüm portlarda yapılandırılmak istenirse:

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

Belirli bir portta (veya portlarda) yapılandırılmak istendiğinde:

```
SW(config-if)# spanning-tree bpduguard enable  
SW# show spanning-tree summary totals
```

Root Guard

Hatalı switch'in Root Switch olmasını önleyen bir güvenlik önlemidir. Herhangi bir switch root olmaya çalıştığında bu yapılandırmanın yapıldığı port root-inconsistent moda düşer ve iletişim kesilir. BPDU göndermeyi bıraktığında ise otomatik olarak tekrar devreye girer.

Yapılardırma:

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

Doğrulama

```
SW#show spanning-tree inconsistentports
```

Unidirectional Link Detection (UDLD)

UDLD protokolü ile fiber ve twisted pair kabloların tek taraflı çalışmaları tespit edilebilmektedir. Unidirectional link fiber ve utp kabloların içindeki veri yollamak ve almak için transmit ve receive tellerinden herhangi birinin kopması durumudur. Bu durumda transmit edilen tellerden birisinden veri akışı varken diğerinden veri akışı olmayacağıdır. Bu da cihazları up down yapar. Cihazların karşılıklı interfacelerinin up down olması tehlikeli bir durumdur. İki taraf birden down olunca sorun olduğu hemen anlaşılır. Spanning tree ile yedek hatta geçilebilir. Fakat up down durumunda cihazlardan biri karşı tarafı ayakta gördüğü için yedek hatta geçmemeyi de düşünmez. Bu durum network'te loop oluşturabilir. Kısacası, fiber veya twisted pair linkler arasında portlardan herhangi biri veri akışını çift taraflı yapamayınca, portun biri up iken diğerini down olunca veya fiber tellerinden herhangi biri tam olarak bağlı olmayınca, unidirectional link hatası meydana gelir.

Bir switch, Layer-1 fiziksel bağlantı sorunlarından kaynaklanan, elektriksel keepalive sinyallerini algılayabilir. Bu sinyaller Ethernet teknolojisinde **link beat** olarak adlandırılır.

Ancak bazı durumlarda elektriksel sinyali iletebilecek kadar temas olmasına karşın, çift yönlü iletişimini sağlanamayabilir. Buna ***Unidirectional Link*** (Tek-yönlü bağlantı) denir. UDLD, arayüzlerden gönderilen periyodik hello mesajları ile çift yönlü iletişimimin devam edip etmediğini kontrol eder. Aynı zamanda diğer uç tarafından onayın gerçekleşip gerçekleşmediğini de kontrol eder.

Çalışma prensibi porttan belli sıklıkta paket (hello paketi) gönderip gönderilen bu paketlere cevap alma mantığına dayanır. Belli bir süre (hold time) gönderilen pakete cevap gelmiyorsa cihaz o portunu UDLD sorunu dolayısıyla shut eder. Protokolün çalışabilmesi için cihazların karşılıklı portlarının UDLD özelliğini desteklemesi gereklidir. UDLD paketlerinde kullanılan hedef mac adresi 0100.0CCC.CCCC'dir.

UDLD yapısında ***Normal*** ve ***Aggressive*** olmak üzere iki mod vardır. Önerilen aggressive mod kullanımıdır. Eğer hello mesaja cevap gelmezse, link Normal modda **Undetermined** durumuna düşer. Aggressive modda ise **error-disabled** moda düşer.

Aralarında temel olarak iki fark vardır. Birincisi twisted pair unidirectional link hataları sadece aggressive modda anlaşılabılır. Normal modda sadece fiber için hata tespiti mümkündür. İkinci fark ise, normal modda UDLD paketleri holdtime sonunda gelmezse port direk shut edilirken, aggressive modda ise cihazlar arasında UDLD komşugunu tekrar kurabilmek için direk olarak shut edip beklemek yerine 8 defa oturumu devam ettirme girişimi olur.

Fiber-Optic arayüzler için yapılandırma:

SW(config)# udld [enable | aggressive]

Bu komut global config modunda yapılmasına karşın sadece fiber portlara etki eder.

Belirli bir fiber-optic port için yapılandırma yapılmak istenirse:

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

Fiber olmayan portlar için yapılandırma arayüzde uygulanır:

SW(config-if)# udld [enable | aggressive]

Bütün arayüzlerde tekrar enable etmek için aşağıdaki komutu kullanın.

SW# udld reset

Yapılurma doğrulama:

SW# show udld interface

SW# sh udld UDLD portların durumunu gösterir.

SW# reset udldUDLD dolayısıyla kapatılmış portları resetler. Hello paketleri tekrardan gönderilmeye başlar.

Loop Guard

Normal şartlarda komşu switchlerden gelen BPDU paketlerine göre bazı portlar bloklanır. Ancak bazen Unidirectional link, yazılım veya yapılandırma hatası sebebiyle (komşu switch BPDU gönderemez veya gönderilen BPDU alınamazsa) bloklanması gereken port forwarding durumuna geçip döngüye sebebiyet verebilir. Çünkü STP artık bir loop olmadığını düşünenecektir.

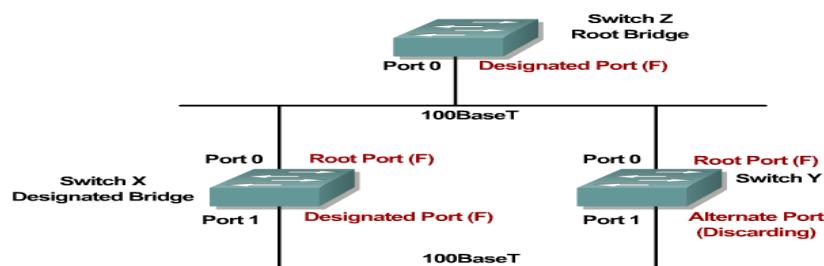
LoopGuard yapılandırıldığında; belli bir süre BPDU alınamazsa port “*loop inconsistent*” moda düşer ve bloklanmış olarak kalır. Root ya da designated port olma ihtimali olan her portta LoopGuard yapılandırılmalıdır.

SW (config)# spanning-tree loopguard default

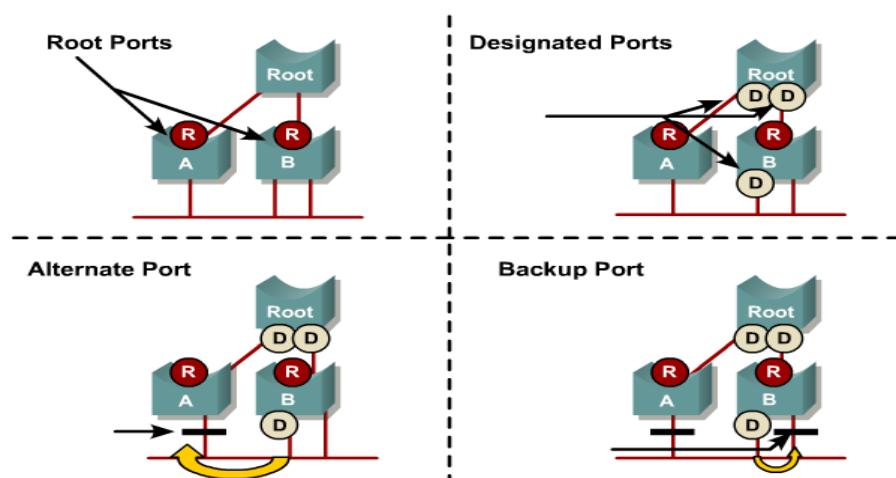
Sadece belirli bir arayüzde yapılandırmak için:

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

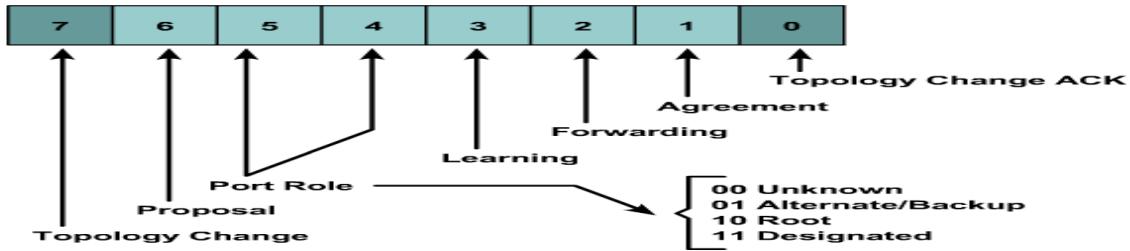
RAPID STP (RSTP) IEEE 802.1w



Rapid STP, STP'deki 30-50 sn arasında süren convergence süresini kısaltan, hızlı bir loop önleme protokolüdür. IEEE 802.1d (STP) ile uyumludur. RSTP'de STP'de bulunmayan *Alternate* ve *Backup* portları bulunmaktadır. Ayrıca spanning-tree sürecine dahil olmayan portlara da *Edge Port* denir.



RSTP BPDU formatı 802.1d ile tamamen aynıdır ancak Version alanı 2 dir. Ayrıca Flag için 8-bit kullanır.

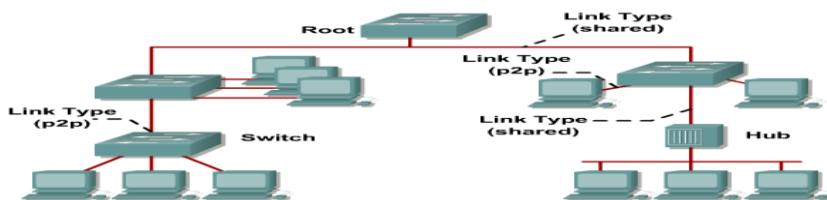


RSTP port durumları da switch çalışma mantığı ile uyumlu olan *discarding*, *learning* ve *forwarding* olan üç durumdan oluşur.

Operational Port State	STP Port State	RSTP Port State
Enabled	Blocking	Discarding
Enabled	Listening	Discarding
Enabled	Learning	Learning
Enabled	Forwarding	Forwarding
Disabled	Disabled	Discarding

RSTP LINK TÜRLERİ

RSTP'de linklerin full-duplex olması gereklidir. Linkler eğer paylaşımı bir ortama bağlanıyorsa (birden çok switch ya da hub içeren link) **Shared Link**; Bir switch'e ya da PC ye bağlanıyorsa **Point-to-Point (P2P) Link** adını alır.



RSTP YAPILANDIRMA ve DOĞRULAMA

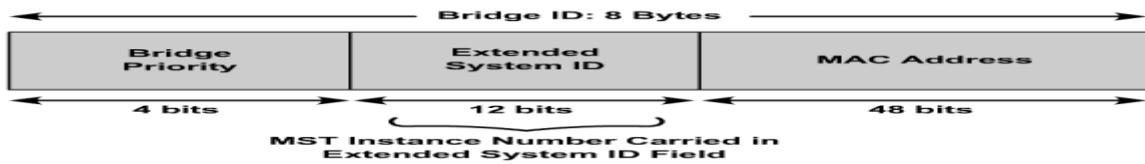
```
SW(config)#spanning-tree mode rapid-pvst
```

```
SW#show spanning-tree vlan 10
```

MULTIPLE SPANNING-TREE PROTOCOL (MSTP)

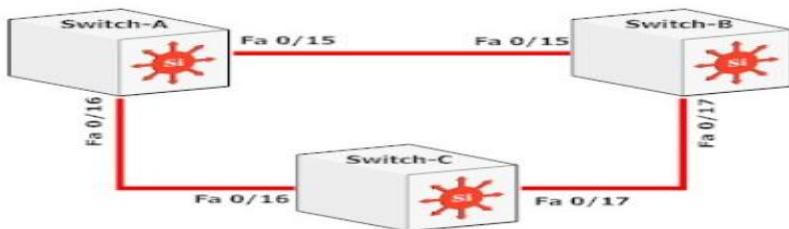
MSTP 'nin amacı birden fazla çalışan STP örneklerinin sayısını azaltarak kaynak tüketiminde verimlilik sağlamaktır. PVST+ yapısında teorik olarak her vlan için bir STP örneği çalışır. Bu durumda her STP örneği için BPDU'lar ve bunlara bağlı olarak Root seçim süreçleri ve Rootlar bulunacaktır.

MSTP, spanning-tree örneklerine dahil olan VLAN'ları grupperarak trunk üzerinden birden çok spanning-tree oluşturmayı sağlar.



MSTP Örnek (Instance) numarası 12-bitlik Extended System ID alanında taşınır.

MSTP YAPILANDIRMA



Yukardaki örnekte vlan 10-20 için SWA; Vlan 21-30 için SWB root olsun.

```
SWITCH-A(config)#vlan 10-30
SWITCH-A(config)#spanning-tree mode mst
SWITCH-A(config)#spanning-tree mst configuration
SWITCH-A(config-mst)#name TEST
SWITCH-A(config-mst)#revision 10
SWITCH-A(config-mst)#instance 1 vlan 10-20
SWITCH-A(config-mst)#instance 2 vlan 21-30
SWITCH-A(config)#spanning-tree mst 1 root primary
SWITCH-A(config)#spanning-tree mst 2 root secondary
```

```
SWITCH-B(config)#spanning-tree mode mst
SWITCH-B(config)#spanning-tree mst configuration
SWITCH-B(config-mst)#name TEST
SWITCH-B(config-mst)#revision 10
SWITCH-B(config-mst)#instance 1 vlan 10-20
SWITCH-B(config-mst)#instance 2 vlan 21-30
SWITCH-B(config)#spanning-tree mst 2 root primary
SWITCH-B(config)#spanning-tree mst 1 root secondary
```

```
SWITCH-C(config)#spanning-tree mode mst
SWITCH-C(config)#spanning-tree mst configuration
```

```

SWITCH-C(config-mst)#name TEST
SWITCH-C(config-mst)#revision 10
SWITCH-C(config-mst)#instance 1 vlan 10-20
SWITCH-C(config-mst)#instance 2 vlan 21-30

```

Revizyon numarası MSTP yapılandırmasının sürüm numarası olarak algılanabilir. Her değişiklikte bu değeri 1 artırmak gereklidir ve her cihazda aynı olması gereklidir. Instance numaraları MSTP örnek sayısıdır (IST) . Default olarak tüm VLAN'lar 0. instance ile eşleştirilmiştir.

```

SWITCH-C# show spanning-tree mst 1
MST1 vlans mapped: 10-20
Bridge address 001a.a181.1880 priority 32769 (32768 sysid 1)
Root address 000c.85d7.d780 priority 24577 (24576 sysid 1)
port Fa0/2 cost 200000 rem hops 19

```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Altn	BLK	200000	128.3	P2p Pre-STD-Rx
Fa0/2	Root	FWD	200000	128.4	P 2p Pre-STD-Rx

```

SWITCH-C# show spanning-tree mst 2
MST2 vlans mapped: 21-30
Bridge address 001a.a181.1880 priority 32770 (32768 sysid 2)
Root address 000c.85de.1100 priority 24578 (24576 sysid 2)
port Fa0/1 cost 200000 rem hops 19

```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Root	FWD	200000	128.3	P2p Pre-STD-Rx
Fa0/2	Altn	BLK	200000	128.4	P2p Pre-STD-Rx

ETHER CHANNEL

Switchlerde aynı özelliklere (speed ve duplex) ve türe (FastEthernet) sahip fiziksel portların döngü oluşturmadan gruplandırılarak bir mantıksal port gibi davranmasıdır. En az 2 en çok 8 portu gruplandırmak mümkündür. Bu sayede örneğin FastEthernet portunun kapasitesi full duplex yapıda 1600Mbps'ye çıkarmak mümkündür. ($200\text{Mbps} \times 8 = 1600\text{ Mbps}$)

Kaynaktan hedefe giden trafik, seçilecek load balancing algoritmasının sonucuna göre bu mantıksal porttaki bir fiziksel porttan gider. Bu algoritma, kaynak IP, Hedef IP, Kaynak MAC, Hedef MAC, TCP/UDP port numaralarına göre yapılabilir. Bu algoritmanın hash sonucunda mantıksal gruptaki bir link belirlenir ve trafik o linkten akar. Örneğin algoritma olarak sadece Kaynak MAC seçilirse, belli MAC adresine sahip bir NIC'e ait trafik her zaman aynı linkten geçecektir. Kaynak ve Hedef MAC gibi bir algoritma seçilirse mantıksal XOR işleminin sonucu kullanılacak linki belirler. Aşağıdaki örnekte 2 link yapısındaki bir ether channel mimarisinde adrese göre hangi linkin seçileceği görülmektedir. Bunun için tek bit kullanılmaktadır.

Table 6-2 Frame Distribution on a Two-Link EtherChannel

Binary Address	Two-Link EtherChannel XOR and Link Number
Addr1: ... xxxxxxxx0	... xxxxxxxx0: Use link 0
Addr2: ... xxxxxxxx0	
Addr1: ... xxxxxxxx0	... xxxxxxxx1: Use link 1
Addr2: ... xxxxxxxx1	
Addr1: ... xxxxxxxx1	... xxxxxxxx1: Use link 1
Addr2: ... xxxxxxxx0	
Addr1: ... xxxxxxxx1	... xxxxxxxx0: Use link 0
Addr2: ... xxxxxxxx1	

4 portlu yapıda ise son iki bit kullanılır. Buna göre;

00	00 0.Link	01	01 1.Link	10	10 2.Link	11	11 3.Link
00	01 1.Link	01	00 0.Link	10	11 3.Link	11	10 2.Link
00	10 2.Link	01	11 3.Link	10	00 0.Link	11	01 1.Link
00	11 3.Link	01	10 2.Link	10	01 1.Link	11	00 0.Link
11		11		11		11	

Fiziksel portlar arasında failover yapısı vardır. Herhangi bir fiziksel link down olduğunda, diğer fiziksel portlardan veri trafigi akabilmektedir.

Genel olarak bu fiziksel portların aynı VLAN'de olması gereklidir. Turnk moda için bu yapı kullanıldığında ise aynı Native Vlan tanımlaması yapılmalıdır.

LOAD BALANCE ALGORİTMALARI

Load balancing için algoritma aşağıdaki komutla belirlenir.

```
Switch(config)# port-channel load-balance method
```

Table 6-3 Types of EtherChannel Load-Balancing Methods

method Value	Hash Input	Hash Operation	Switch Model
src-ip	Source IP address	bits	All models
dst-ip	Destination IP address	bits	All models
src-dst-ip	Source and destination IP address	XOR	All models
src-mac	Source MAC address	bits	All models
dst-mac	Destination MAC address	bits	All models
src-dst-mac	Source and destination MAC	XOR	All models
src-port	Source port number	bits	6500, 4500
dst-port	Destination port number	bits	6500, 4500
src-dst-port	Source and destination port	XOR	6500, 4500

```
Sw#show etherchannel port-channel
```

```
Sw#show etherchannel load-balance
```

Komutları ile yapılandırmayı doğrulayabilirsiniz.

ETHERCHANNEL NEGOTIATION PROTOCOLS

İki switch arasında otomatik link yapılandırması için iki tür Negotiation Protokol kullanılır.

PAgP (Port Aggregation Protocol) Cisco'un geliştirdiği bir protokoldür. **LACP** (Link Aggregation Control Protocol) ise standarttır.

Table 6-4 EtherChannel Negotiation Protocols

Negotiation Mode	Negotiation Packets Sent?		Characteristics
PAgP	LACP		
On	On	No	All ports channeling
Auto	Passive	Yes	Waits to channel until asked
Desirable	Active	Yes	Actively asks to form a channel

Port Aggregation Protocol (PAgP) (Default)

Cisco'ya özgü bu protokolde otomatik etherchannel yapılandırması için PAgP paketleri ether-channel portlarından gönderilir. Gruptaki bir portta bir değişiklik olduğundan bu değişiklik diğer bütün portlara yansıtılır.

Portlar, **Auto** ya da **Desirable(Active)** modda olabilir. **Desirable** modda, local switch uzaktaki switche EtherChannel yapılandırma isteği gönderilir. **Auto** (default) modda ise, local switch karşı switchten etherchannel yapılandırma isteği gelmesini beklemektedir.

Etherchannel'a katılacak her interface, aynı grup numarası(1-64) altında toplanmalıdır. Mod olarak **on** seçildiğinde interface PAgP'ye ya da LACP'ye gerek duyulmadan koşulsuz olarak Etherchannel'a dahil olur. **Auto** modunda port pasif olarak uzak switchleri dinler ve onlardan istek alındığında Etherchannel'a katılır. **desirable** modunda ise port aktif olarak uzaktaki potansiyel katılımcıya Etherchannel kurulum isteği gönderilir. Varsayılan olarak PAgP, desirable ve auto modlarıyla birlikte **silent** alt modunda çalışır. Bu alt mod karşı taraftaki porttan herhangi bir PAgP paketi alınmása dahi Etherchannel kurulabilmesi anlamına gelmektedir. Bu özellik, Etherchannel kurmak istediğimiz uzak cihaz bir sunucuya ve sunucunun PAgP kurulum sürecinde herhangi bir PAgP paketi göndermesi mümkün olmadığından yine de Etherchannel'in kurulabilmesini sağlar. Bu alt mod istersek non-silent olarak değiştirebilir ve karşı tarafta PAgP sürecine katılabilecek bir cihaz olması gerektiğini porta öğretmiş oluruz. Yapılandırma;

PAgP

```
Switch(config)# port-channel load-balance src-dst-port
Switch(config)# interface range fa 0/21 - 24
Switch(config-if)# channel-protocol pagp
Switch(config-if)# channel-group 1 mode desirable non-silent
```

Link Aggregation Control Protocol (LACP)

IEEE 802.3ad standardı olarak tanımlanmıştır. PAgP deki gibi, paketler ile yine uzak switch ile negotiation sağlanması gereklidir. Burada düşük **system priority** (2-Byte Priority ve 6-Byte MAC) değerine sahip cihaz belli bir zamanda hangi portların aktif olarak etherchannel'a dahil olacağı kararını verir. Portlar, **port-priority** (2-Byte Port Priority ve 2-Byte Port No) değerine göre seçilip aktif olur. Aktif olan port iletim yaparken diğer portlar **stand-by** modda kalır. Aktif port down olduğunda devreye girerler.

PAgP'deki gibi yine burda da portlar aktif ya da pasif olabilir.

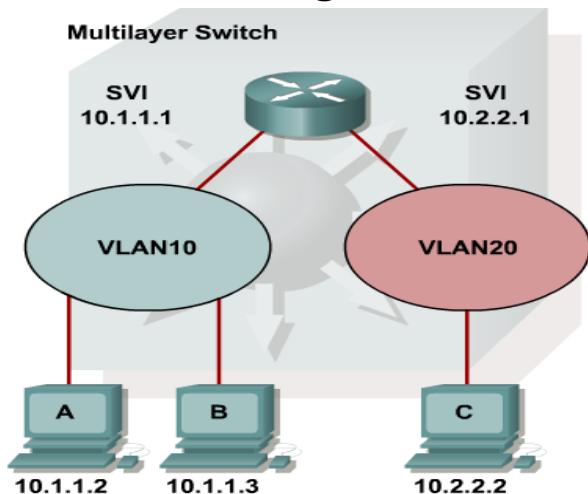
Portların modu **ON** olarak seçilmişse, LACP ve PAGP paketleri gönderilmez.

EtherChannel Configuration

LACP

```
Switch(config)# lacp system-priority 32768 //1-65535
Switch(config)# interface range fa0/21-24
Switch(config-if)# channel-protocol lacp
Switch(config-if)# channel-group 1 mode active
Switch(config-if)# lacp port-priority 1 //1-64
```

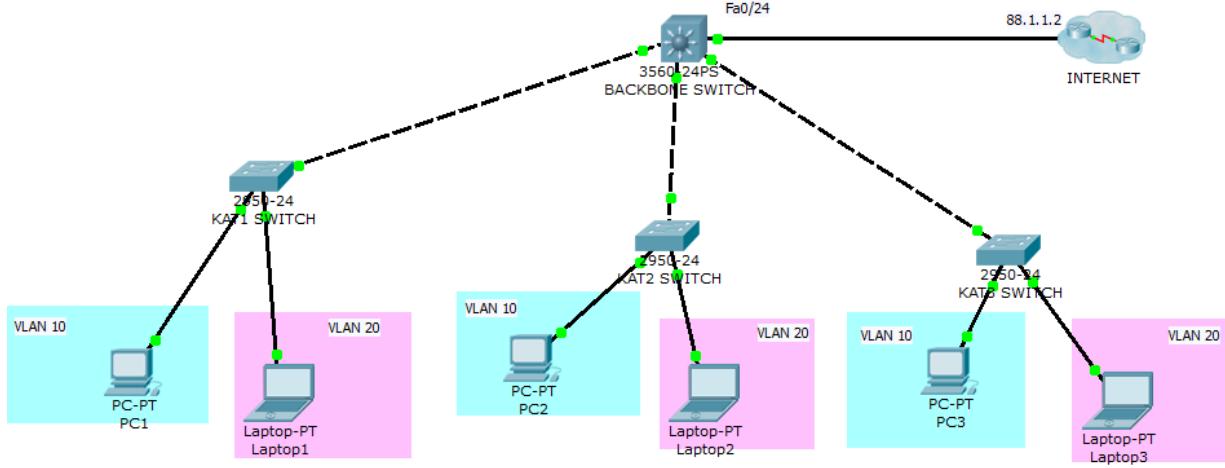
3.Katman Artıklığını Keşfetme



Layer3 ya da daha üst katmanlarda çalışan switchlerdir. Dolayısıyla routing kabiliyetleri olan switchlerdir. Daha önceki konularda farklı VLAN'leri haberleşirebilmek için bir router ya da en az Layer3 bir cihazın olması gerekliliği bahsedilmiştir. Bu örnekte Layer3 switch üzerinde VLAN'lar oluşturup yine bu switch üzerinde bu ağların haberleştirilmesi yapılacaktır. Dolayısıyla VLAN'lerin gateway adresleri için switch üzerinde sanal interface'ler (SVI= Switch Virtual Interface) oluşturulması gereklidir.

Örnek: Aşağıdaki örnekte Backbone Switch için Layer3 switch (Cisco 3560), Katlarda kullanılan Access Switchler için Cisco 2950 switchler kullanılmıştır. PC'ler 1. Porta, Laptoplar ise 11. Porta bağlanmıştır.

VLAN 10 IP Aralığı 192.168.10.0/24; VLAN 20 IP Aralığı 192.168.20.0/24



Burada VLAN10 ve VLAN20'nin BackBone switch üzerinden haberleşmesini sağlayacağız. Öncelikle switchler arasındaki bağlantının trunk olmasını sağlayıp BackBone switch'i VTP server yapalım ve VLAN bilgilerini diğer kat switchlerine dağıtalım. Layer3 switchte, access switchlere bağlı olan 21-22 ve 23 portlarını trunk yapalım.

```
BACKBONE(config)#interface range fastEthernet 0/21-23
BACKBONE(config-if-range)#switchport trunk encapsulation dot1q /* 
BACKBONE(config-if-range)#switchport mode trunk
```

* Cisco Layer3 switchte dot1q ve ISL olmak üzere iki tür trunk desteği vardır. Bu sebeple trunk türünü belirlemek gerekir. Access Layer switchlerin Backbone switch'e bağlı olan portları Dynamic modda olduğu için, otomatik olarak trunk oalçaklardır. Bu sebeple DTP protoklü çalıştığı sürece bu switchlerde portları trunk yapmaya gerek yoktur.

```
BACKBONE(config)#vtp mode server
Device mode already VTP SERVER.
BACKBONE(config)#vtp domain erdal.com
Changing VTP domain name from NULL to erdal.com
BACKBONE(config)#vtp password cisco
Setting device VLAN database password to cisco
BACKBONE(config)#vlan 10
BACKBONE(config-vlan)#name MUHASEBE
BACKBONE(config-vlan)#vlan 20
BACKBONE(config-vlan)#name BILGI_ISLEM
```

Şimdi de Access Layer switchlerde VTP yapılandırması ile oluşturulan bu VLAN'lerin alınmasını sağlayalım.

```

KAT1_SWITCH(config)#vtp mode client
Setting device to VTP CLIENT mode.
KAT1_SWITCH(config)#vtp password cisco
Setting device VLAN database password to cisco

KAT2_SWITCH(config)#vtp mode client
Setting device to VTP CLIENT mode.
KAT2_SWITCH(config)#vtp password cisco
Setting device VLAN database password to cisco

KAT3_SWITCH(config)#vtp mode client
Setting device to VTP CLIENT mode.
KAT3_SWITCH(config)#vtp password cisco
Setting device VLAN database password to cisco

```

Not: Access switchler ile Backbone Switch arasındaki bağlantılar trunk olduğundan vtp domain adı bu switchlerce zaten bilinmektedir. Bu sebeple **vtp domain erdal.com** yazmaya gerek yoktur.

VTP yapılandırması sonunda Access Switchlere VLAN bilgileri gelmiştir. Kontrol için Kat3 switchte aşağıdaki komut kullanılmıştır.

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

Access SWitchlerde 1 ile 10 arasındaki portları VLAN10, 11 ile 20 arasındaki portları ise VLAN20 üyesi yapalım.

```

KAT1_SWITCH(config)#interface range fastEthernet 0/1-10
KAT1_SWITCH(config-if-range)#switchport access vlan 10
KAT1_SWITCH(config-if-range)#interface range fastEthernet 0/11-20
KAT1_SWITCH(config-if-range)#switchport access vlan 20

KAT2_SWITCH(config)#interface range fastEthernet 0/1-10

```

```
KAT2_SWITCH(config-if-range)#switchport access vlan 10
KAT2_SWITCH(config-if-range)#interface range fastEthernet 0/11-20
KAT2_SWITCH(config-if-range)#switchport access vlan 20
KAT3_SWITCH(config)#interface range fastEthernet 0/1-10
KAT3_SWITCH(config-if-range)#switchport access vlan 10
KAT3_SWITCH(config-if-range)#interface range fastEthernet 0/11-20
KAT3_SWITCH(config-if-range)#switchport access vlan 20
```

Şimdi Bacbone Switch'te VLAN10 ve VLAN20 için Gateway olacak şekilde iki SVI oluşturalım.

```
BACKBONE(config)#interface vlan 10
%LINK-5-CHANGED: Interface Vlan10, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state to up
BACKBONE(config-if)#ip address 192.168.10.1 255.255.255.0

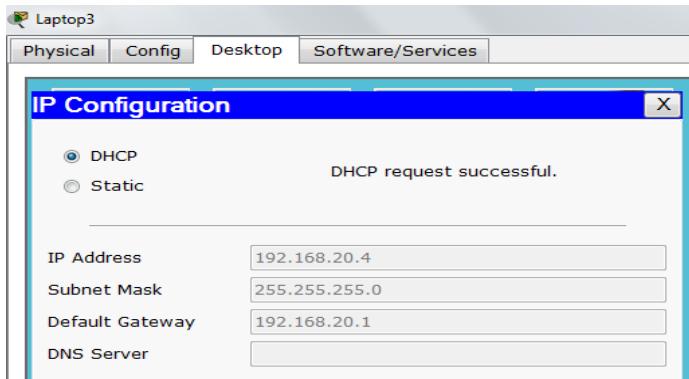
BACKBONE(config-if)#interface vlan 20
%LINK-5-CHANGED: Interface Vlan20, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to up
BACKBONE(config-if)#ip address 192.168.20.1 255.255.255.0
```

PC'lerin ve Laptopların otomaik IP alabilmeleri için BackBone switchi DHCP Server olarak yapılandırılmıştır.

```
BACKBONE(config)#ip dhcp pool VLAN10
BACKBONE(dhcp-config)#network 192.168.10.0 255.255.255.0
BACKBONE(dhcp-config)#default-router 192.168.10.1

BACKBONE(dhcp-config)#ip dhcp pool VLAN20
BACKBONE(dhcp-config)#network 192.168.20.0 255.255.255.0
BACKBONE(dhcp-config)#default-router 192.168.20.1
```

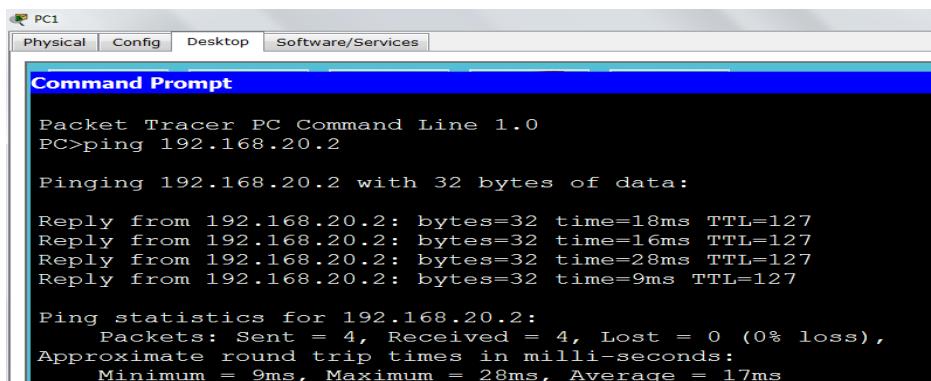
Test için Laptop3'ün IP alıp olmadığı kontrol edilebilir.



Backbone switchin var olan VLAN lar arasında yönlendirme yapabilmesi için Routing işleminin enable edilmesi gerekir.

BACKBONE(config)#ip routing

Artık VLAN'ler L3 switch üzerinden haberleşebilecekledir. Test için PC1'den Laptoplara ping atalım



L3 Switch için interne erişim 88.1.1.2 üzerinden gerçekleşmektedir. O halde switchin interne bağlı portuna (Fa0/24) 88.1.1.1/24 IP adresini verelim.

BACKBONE(config)#interface fastEthernet 0/24

BACKBONE(config-if)#ip add?

% Unrecognized command

Fa0/24 portu default olarak bir Switch Portu olduğundan yukarıda görüldüğü gibi IP adresi verilememektedir. O halde bu portu Switchport olmaktan çıkarıp IP adresi vermek gereklidir. IP adresini verip 88.1.1.1 adresi ile erişimi olup olmadığını test edelim.

BACKBONE(config-if)#no switchport

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24, changed state to up

```
BACKBONE(config-if)#ip address 88.1.1.1 255.255.255.0
```

```
BACKBONE(config-if)#do ping 88.1.1.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 88.1.1.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/2/4 ms

```
BACKBONE(config-if)#

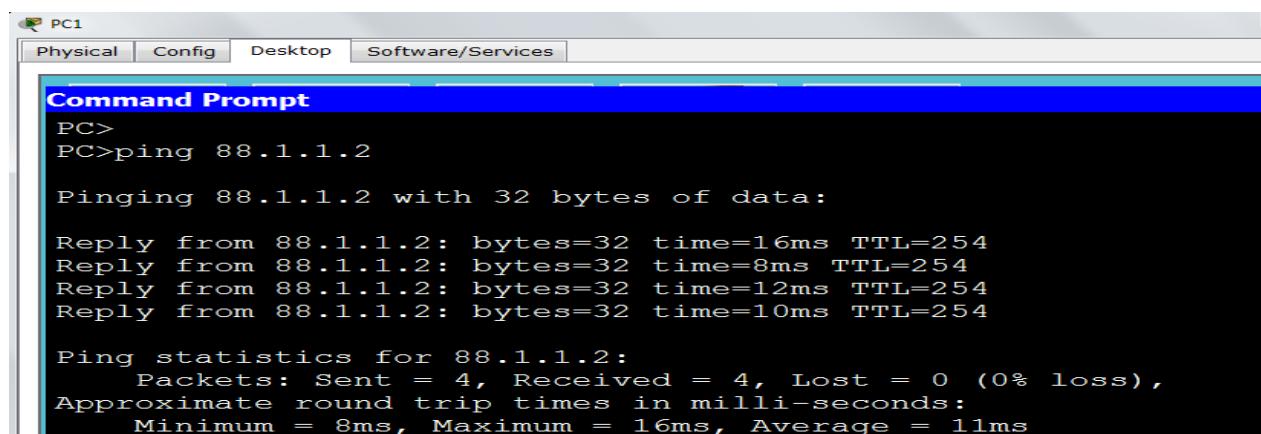
```

Son olarak bu switchte default rota yazalım.

```
BACKBONE(config)#ip route 0.0.0.0 0.0.0.0 88.1.1.2
```

Internet router'da dahili ağa doğru yazılmış bir rota varsa ya da BACKBONE_SWITCH 'te

NAT yapılandırması yapıldıysa iç ağdan Internet router'a ping başarılı olacaktır.



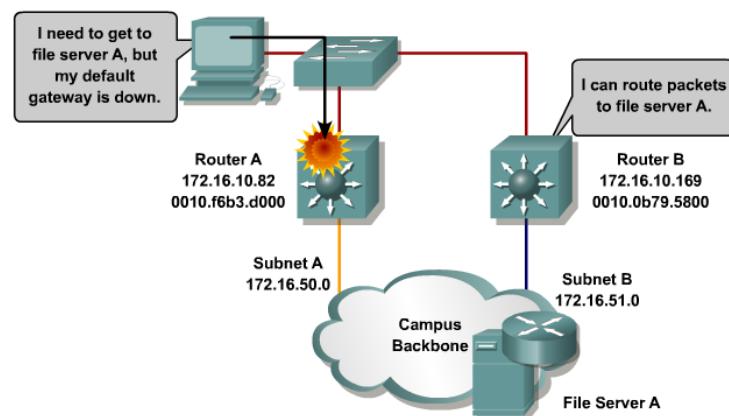
```
PC1
Physical Config Desktop Software/Services
Command Prompt
PC>
PC>ping 88.1.1.2

Pinging 88.1.1.2 with 32 bytes of data:

Reply from 88.1.1.2: bytes=32 time=16ms TTL=254
Reply from 88.1.1.2: bytes=32 time=8ms TTL=254
Reply from 88.1.1.2: bytes=32 time=12ms TTL=254
Reply from 88.1.1.2: bytes=32 time=10ms TTL=254

Ping statistics for 88.1.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 8ms, Maximum = 16ms, Average = 11ms
```

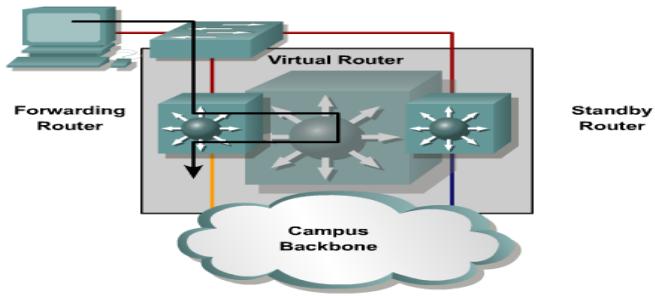
IMPLEMENTING HIGH AVAILABILITY IN A CAMPUS ENVIRONMENT



Bir ağ üzerinde tanımlı iki gateway bulunsun. PC üzerinde çoğunlukla ikinci bir gateway tanımlanmaz. Örneğin RouterA, SubnetA için gateway olsun ve RouterB SubnetB için gateway olsun. RouterA, down olduğunda RouterB'nin dinamik olarak SubnetA için gateway olarak yönlendirme işlemini üstlenir. Ancak PC'ler çoğunlukla bu dinamik yapıdan habersiz kalırlar. Çünkü uç cihazda (Örneğin PC) bir Gateway adresi tanımlanır ve bu adres dinamik olarak değişmez.

Cisco IOS, Proxy-ARP yöntemi ile cihazların default-gateway'in bilemedikleri durumlarda da uzak ağ ile iletişime geçmelerini sağlar. **Proxy-ARP** default olarak açıktır.

ROUTER REDUNDANCY



Birden fazla router'ın sanal olarak bir Router olarak davranışını mantığına dayanır. Sanal router'ın bir IP adresi ve MAC adresi vardır. Bu IP adresi ağdaki PC'lere Gateway adresi olarak verilmelidir. Sanal Router böylece lokal ağda bulunan cihazlar için Gateway görevini üstlenir. Tek bir sanal Router olarak görünen fiziksel routerlar arasında bu işlemin yürütülmesi için bir protokol çalışır. Bu protokol, gelen bir isteğin fiziksel olarak hangi Router tarafından işleme alınacağını belirler. Aktif olarak çalışan Router (Forwarding Router) bu yönlendirme işlemini yürütürken, gruptaki diğer routerlar (Standby /BackUp Router) yedek olarak beklerler.

HSRP (HOT STANDBY ROUTER PROTOCOL)

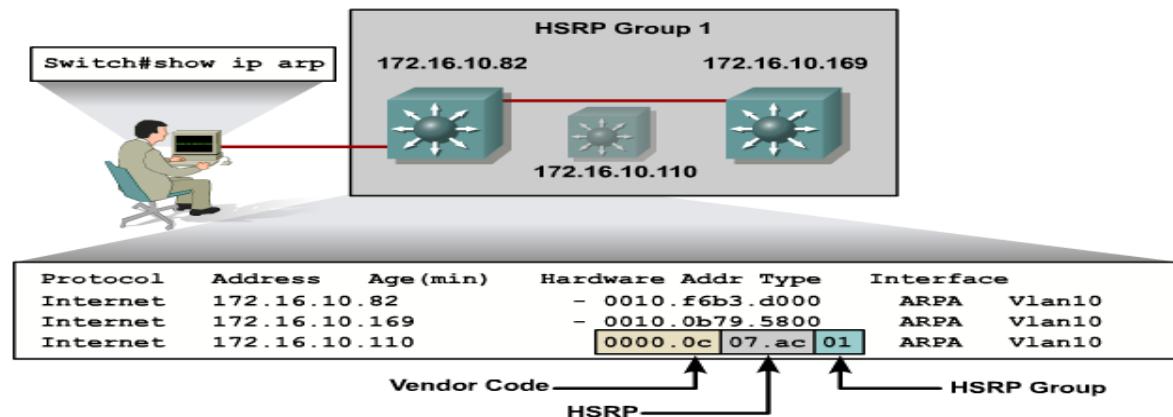
Cisco'ya özgü bir protokoldür. Bu yapıda bir numara ile tanımlanan grup içerisinde Active Router, StandBy Router, Virtual Router ve Other Router görevleri bulunmaktadır. Bu yapıdaki aktif ve standby routerlar **224.0.0.2** multicast adresler üzerinden **UDP** protokolü ve **1985** portunu kullanarak hello mesajları ile haberleşirler. Hello paketlerindeki bilgiler ile kimin hangi rolü üstleneceği belirlenir.

Active Router, Virtual Router adına aktif olarak yönlendirme işlemini yapan fiziksel routerıdır.

Standby Router, Aktif router'ın yedegidir.

Burada Group MAC adresi, **0000.0C07.ACXX** formundadır. XX burada grup numarasıdır.

Aşağıdaki örnekte HSRP Group 1 için IP ve MAC yapılandırması görülmektedir.

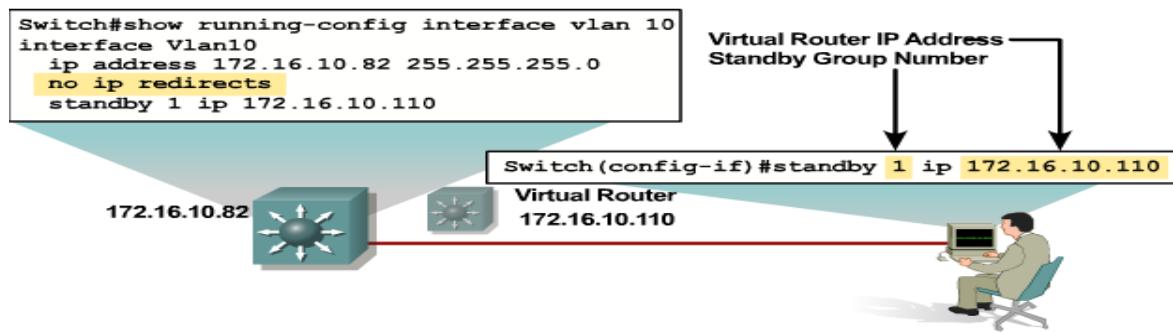


YAPILANDIRMA ve DOĞRULAMA

```
SW(config-if)# standby 1 ip 10.1.1.1
```

Burada 1 grup numarasını, 10.1.1.1 Virtual Router IP adresini gösterir. Bu adres host cihazlar için gateway adresidir.

Örnek:



```
SW#show standby [interface [group]] [active | init | listen | standby] [brief]
SW#show standby delay [type-number]
```

HSRP ROUTER SEÇİMİ

HSRP 'de seçim her router üzerinde yapılandırılan priority değerine göre yapılır (0-255). Varsayılan olarak bu değer her cihazda 100'dür. Yüksek priority değerine sahip cihaz active olarak seçilir. Priority değerleri eşit ise en yüksek IP adresine sahip cihaz Active olur. Priority değerini değiştirmek için interface modunda aşağıdaki komut kullanılır.

```
SW(config-if)# standby 1 priority 255
```

HSRP grubunda bulunan routerlar aşağıdaki altı durumun birinde bulunabilirler.

- Initial : HSRP çalışmıyor veya port kapatılmıştır.
- Learn : Grupdaki Active cihazı ve Virtaul IP yi öğrenir.
- Listen : Grupta Active ya da Standby olduğu durumda routerlar bu modda bekler.
- Speak : Active / Standby'dan hello gelmediği durumda bu moda geçilir ve seçime katılır.
- Standby : Standby Rtr bu modda kalır ve mesaj gönderip dinler.
- Active : Active Rtr bu moddadır.

Active ya da Standby olmayan Router, Listen modda kalır.

HSRP TIMER

Burada Hello Timer, Active Timer ve Standby Timer olmak üzere 3 tip zamanlayıcı vardır.

Active Router down olduğunda, HSRP routerları periyodik olarak gönderilen (Def=3 sn) **hello** mesajlarını alamayacaklardır. Active süresi (Def=10 sn) sonunda Standby Router Active Router görevini üstlenecektir. Diğer HSRP grubundaki routerlar yeni Standby seçim sürecine girerler. Standby Timer, Standby Routera ilişkin bir zamanlayıcıdır ve Standby Rtr'den gelen her hello sonrasında sıfırlanır.

Timer değiştirmek için tüm routelarda aşağıdaki yapılandırma uygulanmalıdır.

```
SW(config-if)# standby 1 timers [msec] hello [msec] holdtime
```

Hold süresi hello süresinin en az 3 katı olacak şekilde yapılandırılmalıdır. Msec parametresi kullanılırsa süreler milisaniye cinsinden girilebilir. Aşağıda sürelerin aralıkları bulunmaktadır.
Hold süresi (1-255 sn / 50-3000 ms), Hello Süresi (1-254 sn / 15 – 999 ms)

Örnek;

```
SW(config-if)# standby 1 timers msec 100 msec 300
```

Active cihaz down olmadan yeni eklenen cihaz priority yüksek olsa bile active olmaz. Down olan Active sonradan tekrar UP olursa Active olamaz. Bu durumun önüne geçmek için (sonradan UP olan cihazın tekrar active olabilmesi için) aşağıdaki yapılandırma gereklidir.

```
SW(config-if)# standby 1 preempt [delay [minimum seconds] [reload seconds]]
```

HSRP Plain-Text Authentication

```
SW(config-if)# standby 1 authentication ERDAL
```

HSRP MD5 Authentication

```
SW(config-if)# standby 1 authentication md5 key-chain [0 | 7] ERDAL
```

```
SW(config)# key chain chain-name
```

```
SW(config-keychain)# key key-number
```

```
SW(config-keychain-key)# key-string [0 | 7] string
```

```
SW(config)# interface type mod/num
```

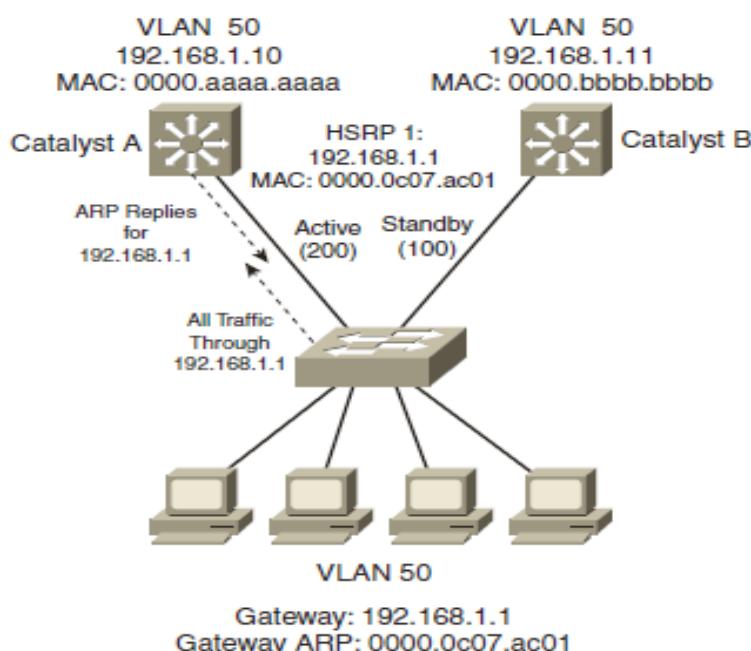
```
SW(config-if)# standby group authentication md5 key-chain chain-name
```

HSRP TRACKING

HSRP'ye dahil olan portların UP olup olmadıkları hello mesajları ile anlaşılabilir. Ancak bu interface up iken, çıkış interfacelerinin (örneğin internete çıkış arayüzü S0/0/0 olsun) durumları da track edilmelidir. Aşağıdaki komut, Router üzerinde yapılandırılan HSRP için S0/0/0 arayüzü track eder ve erişilemez durum söz konusu ise **priority** değerini **100** düşürür. (**Default** değeri **10**)

```
RTR(config-if)# standby 1 track s1/0/0 100
```

ÖRNEK HSRP YAPILANDIRMASI



```
CatalystA(config)# interface vlan 50
```

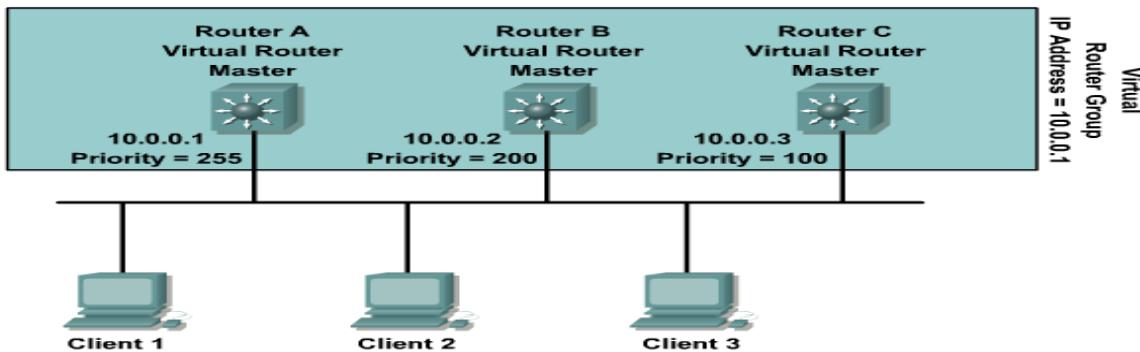
```

CatalystA(config-if)# ip address 192.168.1.10 255.255.255.0
CatalystA(config-if)# standby 1 priority 200
CatalystA(config-if)# standby 1 preempt
CatalystA(config-if)# standby 1 ip 192.168.1.1

CatalystB(config)# interface vlan 50
CatalystB(config-if)# ip address 192.168.1.11 255.255.255.0
CatalystB(config-if)# standby 1 priority 100
CatalystB(config-if)# standby 1 preempt
CatalystB(config-if)# standby 1 ip 192.168.1.1

```

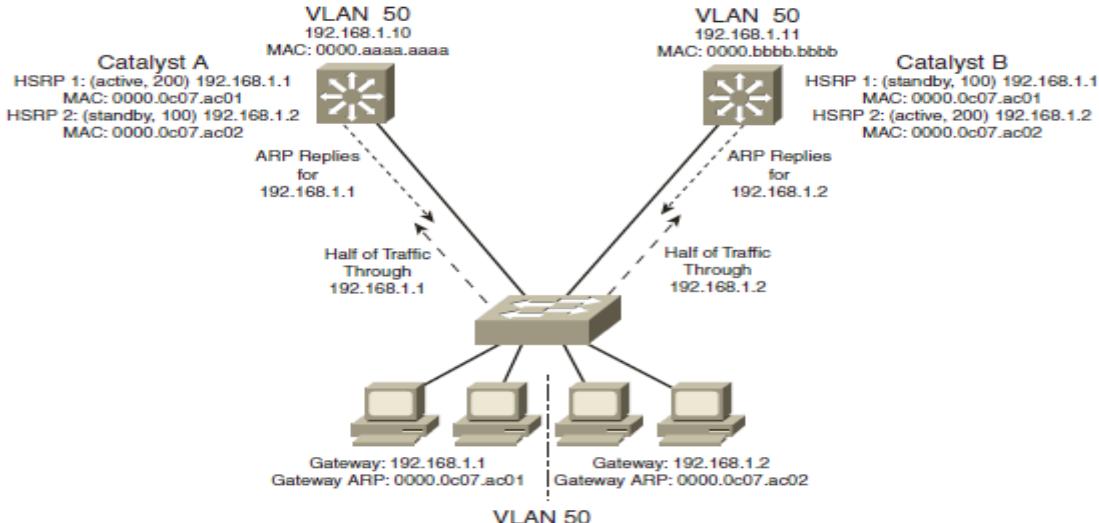
VIRTUAL ROUTER REDUNDANCY PROTOCOL (VRRP)RFC 2338



VRRP yapısında olarak aktif çalışan bir Router (**master**) ve bir veya daha fazla yedek (**backup Router**) bulunmaktadır. Burada HSRP'den farklı olarak routerlardan birinin adresi VRGP grubunun da adresi olabilir. VRRP Cisco 4500 ve 6500 serilerinde çalışır. VRRP Master olan Router yönlendirmeyi yapar. Gruptaki diğer bütün routerlar backup rolündedirler. **224.0.0.18** multicast adresi **UDP 112** portu ile haberleşirler.

Grubun mac-adresi **0000.5e00.01xx**tir. Burda XX VRRP grubunun numarasından türetilmiştir. VRRP mesajları 1-sn periyotlarla yayınlanır. Preempt özelliği default olarak açıktır. Interface track edemez.

YAPILANDIRMA



```
CatalystA(config)# interface vlan 50
CatalystA(config-if)# ip address 192.168.1.10 255.255.255.0
CatalystA(config-if)# vrrp 1 priority 200
CatalystA(config-if)# vrrp 1 ip 192.168.1.1
CatalystA(config-if)# vrrp 2 priority 100
CatalystA(config-if)# no vrrp 2 preempt
CatalystA(config-if)# vrrp 2 ip 192.168.1.2

CatalystB(config)# interface vlan 50
CatalystB(config-if)# ip address 192.168.1.11 255.255.255.0
CatalystB(config-if)# vrrp 1 priority 100
CatalystB(config-if)# no vrrp 1 preempt
CatalystB(config-if)# vrrp 1 ip 192.168.1.1
CatalystB(config-if)# vrrp 2 priority 200
CatalystB(config-if)# vrrp 2 ip 192.168.1.2
```

ACCESS CONTROL LIST (ACL)

Router üzerinden geçen trafik **Source IP**, **Destination IP**, **Protocol**, **Source Port**, **Destination Port** alanlarından biri ya da birkaç tanesi kullanılarak filtrelenebilir.

ACL'ler bu alanları kontrol ederek network güvenliğini sağlar. Bu sayede virus yayan PC'lerin çıkışını yasaklanabilir, sunuculara sınırlı porttan erişimin verilmesi sağlanabilir, belli portlara erişimin yasaklanması sağlanır.

ACL'ler **STANDART** ve **EXTENDED** olarak grüplendirilirler.

Standart ACL'ler Sadece kaynak ip adreslerine bakarak filtreleme yaparlar.

Extended ACL'ler Kaynak IP, Hedef IP, Protokol, ve Port numaralarına göre filtreleme yapabilir.

STANDART ACL

1.ADIM : ACL'nin yazılması

```
R(config)# access-list ACLNUMARASI permit/deny      kaynak_ip      wildcardmask
```

şeklinde tanımlanır.

Wildcardmask değerindeki 1' bitlerinin bulunduğu değere karşılık gelen IP bitleri kontrol edilmez, 0 değerine karşılık gelen IP bitleri ise kontrol edilir.

192.168.1.0 0.0.0.255 bu durumda 192 168 1 değerlerine karşılık gelen wildcard bitleri 0 olduğundan kontrol edilir ve 192 168 1 ile tam eşleşme gerekecektir.

192.168.1.5 0.0.0.0 ifadesine ise tüm bitlerin eşleşmesi gerekecektir. (HOST)

Örnek:

```
R(config)# access-list 1 permit 192.168.2.0 0.0.0.255
```

Standart ACL'de ACL numarası 1 ile 99 arasında olmalıdır.

- ACL' tanımlamada tek bir IP belirtiliyorsa HOST ifadesi kullanılabilir.

Örnek:

```
R(config)# access-list 2 permit 192.168.1.1 0.0.0.0
```

yerine,

```
R(config)# access-list 2 permit host 192.168.1.1 kullanılabilir.
```

- ACL' tanımlamada bütün IPler belirtiliyorsa ANY ifadesi kullanılabilir.

```
R(config)# access-list 3 permit 0.0.0.0 255.255.255.255
```

yerine

```
R(config)# access-list 3 permit any kullanılabilir.
```

- ACL running config dosyasında tutulur.
- EXTENDED ACL 100 ile 199 arasında olmalıdır.

EXTENDED ACL:

```
access - list <100 – 199> permit / deny ip/tcp/udp.. kaynak_IP wildcard hedef_IP  
widcart eq/neq/.. port_no
```

Örnek:

```
R(config)# access-list 101 deny tcp 192.168.2.0 0.0.0.255 192.168.1.0  
0.0.0.255 eq 80
```

```
R(config)# access - list 102 permit udp host 192.168.1.5 192.168.2.0 0.0.0.255  
eq 53
```

2.ADIM : ACL'nin IN ya da OUT yönünde uygulanması

Bir interface'ye giren paketler için IN (INBOUND), çıkan paketler için ise OUT (OUTBOND) yönüne uygulanmalıdır.

```
R(config)# interface fastethernet 0/0
R(config - if)# ip access-group 1 in
```

Burda 1 numara ile tanımlanan standart ACL FastEthernet 0/0 arayüzüne giriş yönünde uygulanmıştır.

KURALLAR:

- 1 – ACL dizisi sıra ile kontrol edilir, bir eşleşme bulunduğu anda alt satırlar incelenmez . Bu yüzden özelden genele doğru bir yapılandırma kullanılmalıdır.
- 2 – ACL satırlarının hiçbir ile eşleşmeyen paketler drop edilir. Yani ACL satırlarının en sonunda **implicit deny** olarak tanımlanan **deny any** olduğu varsayıılır.
- 3 – Genellikle Standart ACL'ler hedefe, extended ACL'ler ise kaynağı en yakın yerde uygulanır.

Yazılmış ACL'leri görmek için;

R# show access-list komutu kullanılır.

NAMED ACL (İSİMLENDİRİLMİŞ ACL) ve ARAYÜZE UYGULANMASI

```
R(config)# ip access-list standart DENEME
R(config - std - acl)# permit 192.168.1.0 0.0.0.255
R(config - std - acl)# deny 192.168.2.0 0.0.0.255
R(config - std - acl)# permit 0.0.0.0 255.255.255.255 // bu satır permit any anlamına gelir

R(config)# interface fa0/0
R(config - if)# ip access-group DENEME out
```

Örnek: Telnet Yasaklama

```
R(config)# access - list 101 deny tcp any any eq telnet
R(config)# access - list 101 permit ip any any
R(config)# interface fa0/0
R(config - if )# ip access-group 101 in
```

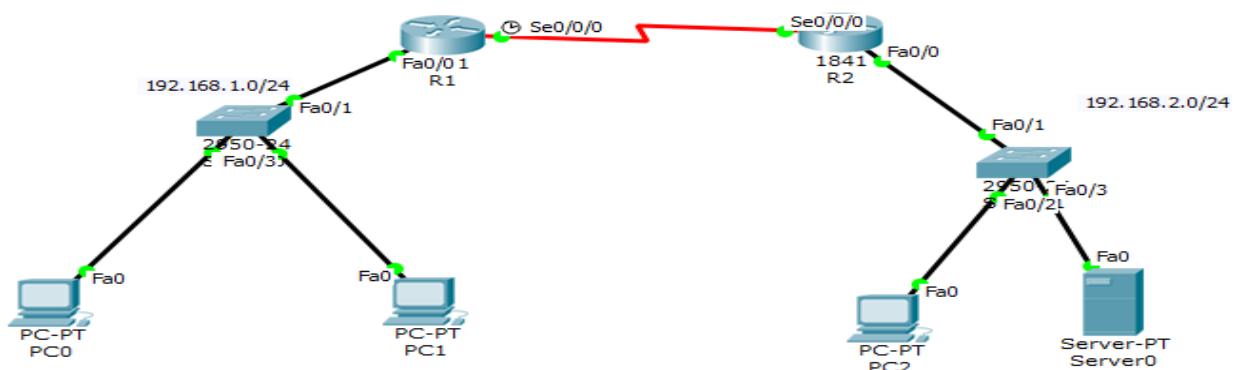
Bu sayede fa0/0 arayüzünden gelen tüm telnet istekleri yasaklanır.

ROUTER'a TELNET ERIŞİMİNİ YASAKLAMA

```
R(config)# access - list 5 permit 192.168.1.0 0.0.0.255
R(config)# line vty 0 4
R(config - line)# ip access-class 5 in
R(config - line)# password erdal
R(config - line)#login
```

Gelişmiş ACL Örnekleri -1

REFLEXIVE ACL ÖRNEK: 192.168.1.0/24 ağından çıkan web ve dns trafikleri reflect parametresi ile inceleneciktir. S0/0/0 arayüzüne gelen trafikler reflect edilen trafiklerin cevabı niteliğinde ise trafikler için bir hole oluşturulacak, trafiğin içeri geçmesi sağlanacaktır.



1. ADIM: Internal bir ACL oluşturma

```
R1(config)#ip access-list extended INTERNAL_ACL  
R1(config-ext-nacl)#permit tcp any any eq 80 reflect web_trafik  
R1(config-ext-nacl)#permit udp any any eq 53 reflect dns_trafik timeout 10
```

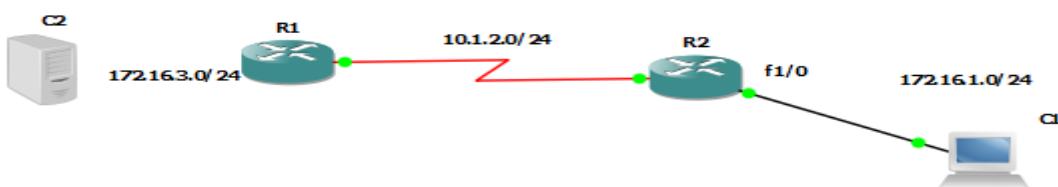
2. ADIM: External ACL tanımla

```
R1(config)#ip access-list extended EXTERNAL_ACL  
R1(config-ext-nacl)#evaluate web_trafik  
R1(config-ext-nacl)#evaluate dns_trafik  
R1(config-ext-nacl)#deny ip any any
```

3. ADIM: ACL'leri Arayzlere Uygulama

```
R1(config)#interface serial 0/0/0  
R1(config-if)# ip access-group INTERNAL_ACL out  
R1(config-if)# ip access-group EXTERNAL_ACL in
```

2. ÖRNEK:



172.16.1.0/24 ağından 172.16.3.0/24 ağındaki sunucuya giden tcp trafikleri için reflexive acl yazalım.

```
R2(config)#ip access-list extended INTERNAL
R2(config-ext-nacl)#permit tcp any any reflect TCP_TRAFIK
R2(config-ext-nacl)#exit
R2(config)#ip access-list extended EXTERNAL
R2(config-ext-nacl)#evaluate TCP_TRAFIK
R2(config-ext-nacl)#exit
R2(config)#interface fastEthernet 1/0
R2(config-if)#ip access-group INTERNAL in
R2(config-if)#int fa1/0
R2(config-if)#ip access-group EXTERNAL out
R2(config-if)#end
R2#
```

Şimdi C1 makinasından 172.16.3.1 sunucusuna erişelim. Ardından R2'de oluşan dinamik ACE satırlarına bakalım.

```
R2#sh access-lists
Extended IP access list EXTERNAL
  10 evaluate TCP_TRAFIK
Extended IP access list INTERNAL
  10 permit tcp any any reflect TCP_TRAFIK (753 matches)
Reflexive IP access list TCP_TRAFIK
  permit tcp host 172.16.3.1 eq www host 172.16.1.2 eq 7740 (5 matches) (time left 7)
  permit tcp host 172.16.3.1 eq www host 172.16.1.2 eq 7739 (27 matches) (time left 3)
```

Yine C1 'den Sunucuya ping atalım.

```
C:\Users\Erdal ÖZDOĞAN>ping 172.16.3.1
172.16.3.1 yoklanıyor 32 bayt veri ile:
172.16.1.1 cevabı: Hedef ağ ulaşılamaz.
172.16.1.1 cevabı: Hedef ağ ulaşılamaz.
İstek zaman aşımına uğradı.
172.16.1.1 cevabı: Hedef ağ ulaşılamaz.

172.16.3.1 için Ping istatistiği:
  Paket: Giden = 4, Gelen = 3, Kaybolan = 1 (<25 kayıp),
C:\Users\Erdal ÖZDOĞAN>
```

R2'nin Fa0/1 arayüzündeki ACL'nin implicit deny satırı nedeni ile paketler düşmüştür. Şimdi de INTERNAL trafiklere ICMP'yi de ekleyelim

```
R2(config)#ip access-list extended INTERNAL
R2(config-ext-nacl)#15 permit icmp any any reflect ICMP_TRAFIK
```

Tekrar C1'den ping atalım.

```
C:\Users\Erdal ÖZDOĞAN>ping 172.16.3.1
172.16.3.1 yoklanıyor 32 bayt veri ile:
İstek zaman aşımına uğradı.
İstek zaman aşımına uğradı.
İstek zaman aşımına uğradı.
İstek zaman aşımına uğradı.

172.16.3.1 için Ping istatistiği:
  Paket: Giden = 4, Gelen = 0, Kaybolan = 4 (<100 kayıp),
C:\Users\Erdal ÖZDOĞAN>
```

Bu kez trafik geçiş yapacak ancak harici trafikte (EXTERNAL) evaluate yapılmadığı için deny ip any any satırına takılacaktır.

R2#sh access-lists

Extended IP access list EXTERNAL

10 evaluate TCP_TRAFIK

20 deny ip any any (12 matches)

Reflexive IP access list ICMP_TRAFIK

permit icmp host 172.16.3.1 host 172.16.1.2 (7 matches) (time left 5)

Extended IP access list INTERNAL

10 permit tcp any any reflect TCP_TRAFIK (72 matches)

15 permit icmp any any reflect ICMP_TRAFIK (9 matches)

Reflexive IP access list TCP_TRAFIK

permit tcp host 10.1.5.25 eq 389 host 172.16.1.2 eq 9031 (4 matches) (time left 5)

permit tcp host 10.2.1.19 eq 389 host 172.16.1.2 eq 9029 (4 matches) (time left 4)

permit tcp host 10.1.1.25 eq 389 host 172.16.1.2 eq 9027 (4 matches) (time left 4)

permit tcp host 10.1.5.26 eq 389 host 172.16.1.2 eq 9025 (4 matches) (time left 4)

permit tcp host 10.135.1.2 eq 389 host 172.16.1.2 eq 9023 (4 matches) (time left 4)

permit tcp host 10.1.1.26 eq 389 host 172.16.1.2 eq 9021 (4 matches) (time left 4)

permit tcp host 10.4.5.28 eq 389 host 172.16.1.2 eq 9017 (7 matches) (time left 2)

permit tcp host 10.1.5.24 eq 389 host 172.16.1.2 eq 9015 (7 matches) (time left 1)

External trafiklerde ICMP_TRAFIK'leri evaluate edelim

R2(config)#ip access-list extended EXTERNAL

R2(config-ext-nacl)#15 evaluate ICMP_TRAFIK

Bu adımdan sonra artık ping paketlerine cevap gelecektir.

```
C:\Users\Erdal ÖZDOĞAN>ping 172.16.3.1
172.16.3.1 yoklanıyor 32 bayt veri ile:
172.16.3.1 cevabı: bayt=32 süre=99ms TTL=254
172.16.3.1 cevabı: bayt=32 süre=70ms TTL=254
172.16.3.1 cevabı: bayt=32 süre=35ms TTL=254
172.16.3.1 cevabı: bayt=32 süre=24ms TTL=254

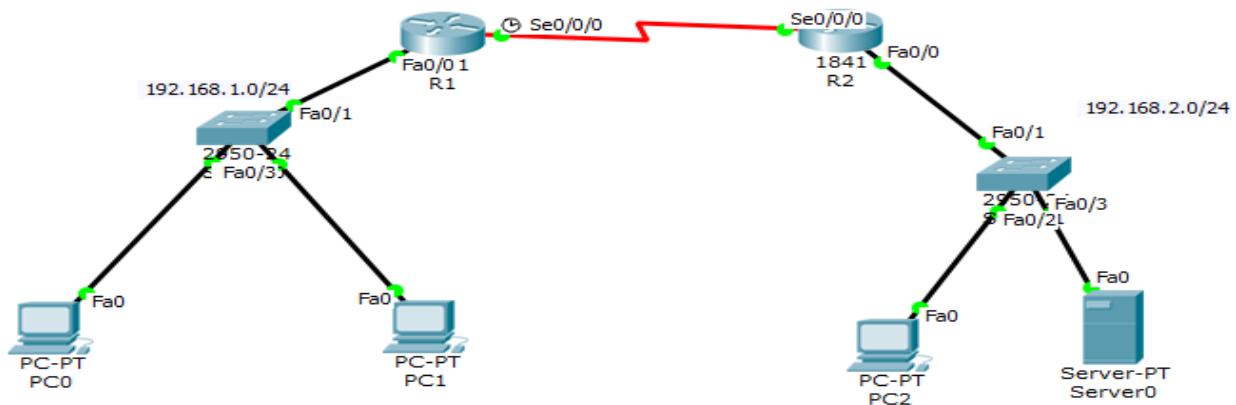
172.16.3.1 için Ping istatistiği:
  Paket: Giden = 4, Gelen = 4, Kaybolan = 0 (<0 kayıp),
Mili saniye türünden yaklaşık tur süreleri:
  En Az = 24ms, En Çok = 99ms, Ortalama = 57ms
```

Ayrıca aşağıdaki komut ile dinamik ACE satırlarının time-out süreleri ayarlanabilir.

R2(config)#ip reflexive-list timeout 300

DYNAMIC ACL ÖRNEK: 192.168.1.0/24 ağından 192.168.2.0/24 ağına erişimin gerçekleşebilmesi için kullanıcıların öncelikle kimlik doğrulamasından geçmesi gerekecektir. Kimlik doğrulama R1'deki local veri tabanı aracılığıyla gerçekleştirilecektir.

Bu uygulamada, 192.168.2.0'a erişmek isteyen kullanıcı öncelikle R2'ye telnet ile erişim yapacak, telnet erişiminde kendisine kullanıcı adı ve parola sorulacaktır. Kullanıcı adı ve parola doğru bilindiği takdirde telnet oturumu sonlandırılacak ve 192.168.2.0 ağının belirlenen süre (time-out=15 dk) boyunca trafik akışlarına izin verilecektir.



Öncelikle R2'de username ve password oluşturulmalıdır. Ardından R2'ye telnet erişimine izin verilmelidir.

```

R2(config)# username student password cisco
R2(config)# access-list 101 permit tcp any host 10.1.2.2 eq telnet
R2(config)# access-list 101 dynamic STUNDET_PASS timeout 15 permit ip 192.168.1.0
0.0.0.255 192.168.2.0 0.0.0.255

R2(config)#interface serial0/0/0
R2(config-if)# ip access-group 101 in

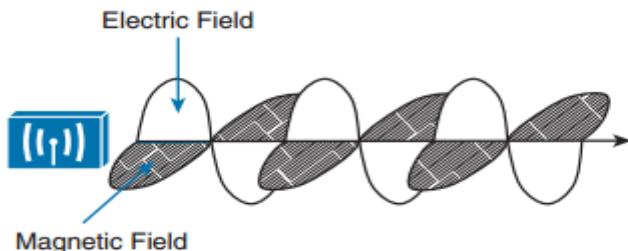
R2(config)#line vty 0 4
R2(config-line)#login local
R2(config-line)#autocommand access-enable host timeout 5
    
```

* son satırındaki timeout süresi idle-timeout süresidir.

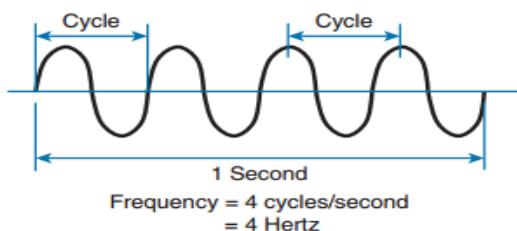
Kablosuz Ağlar

Temel Kavramlar

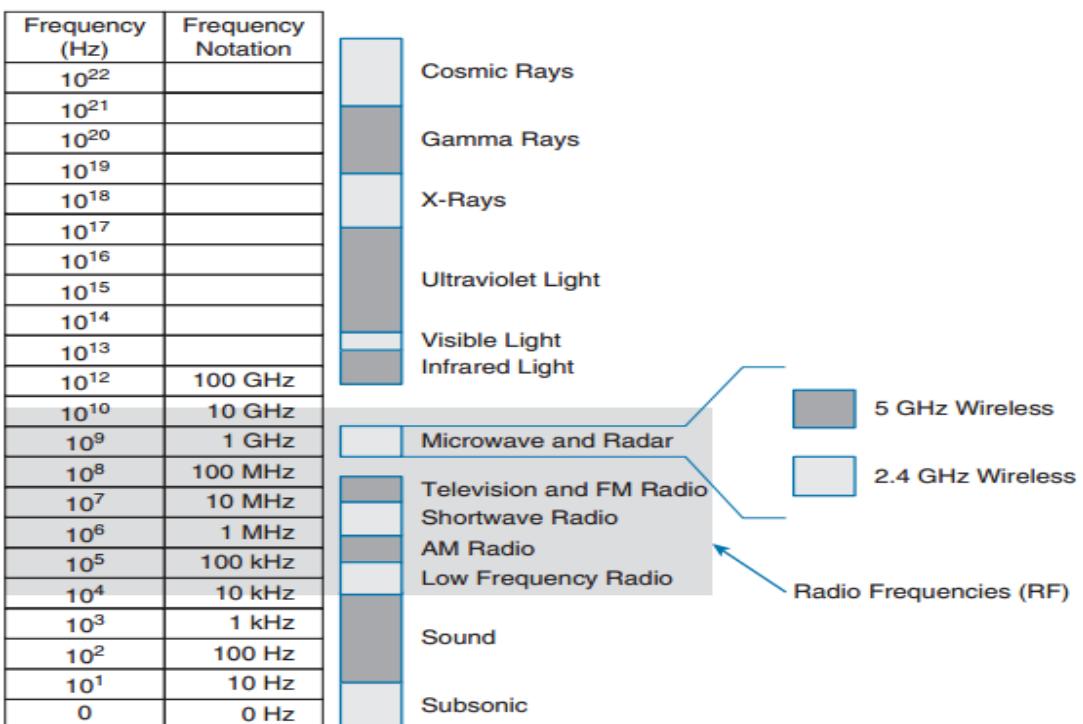
Dalga Yapısı: RF Sinyaller doğada dalga şeklinde yayılırlar. Fizik yasaları gereği bir dalga elektrik alan ve manyetik alan olmak üzere birbirlerine dik iki bileşenden oluşur.



Frekans: Saniyedeki sinüsoidal döngü sayısıdır. Dalga boyu ile ters orantılıdır. Hz, kHz, MHz, GHz..



Wireless LAN ağlarında genellikle ISM bandı diye bilinen 2,4 GHz ve 5GHz frkenası kullanılır.



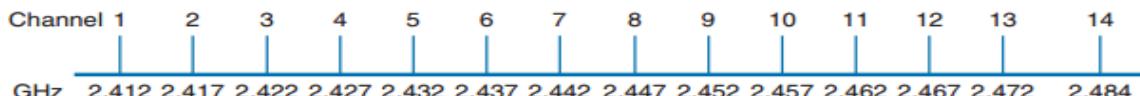
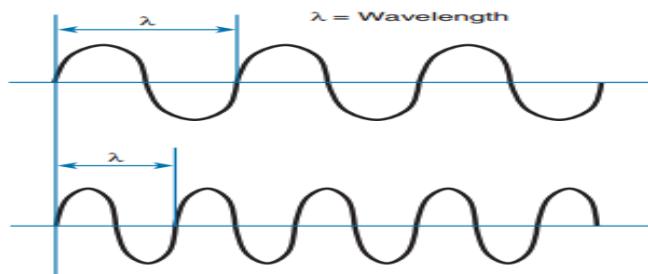


Figure 1-7 An Example of Channel Spacing in the 2.4-GHz Band.

DalgaBoyutu: İki dalga tepesi arasındaki mesafedir.



Dalga boyu ve frekans değeri birbiri ile ilişkilidir. $\text{DalgaBoyutu} = \text{C/Frekans}$ ($v=c/\lambda$)

Genlik: Farklı yönde iki dalga tepesi arasındaki mesafedir. Sinyalin gücünü temsil eder.



Figure 1-12 Signal Amplitude.

Sinyal gücünün birimi W olarak ölçülür. Örneğin AM radyolarının sinyal gücü 50.000 ; FM radyo vericilerinin sinyal gücü 16.000 W; Wireless LAN sinyal gücü ise 0,1 W=100 miliWatt (mW) olabilir.

Desibel (dB): İki sinyal gücünün karşılaştırılmasının oranıdır. Örneğin bir vericinin 1 mW olan sinyal gücü 2mW değerine çıkarılırsa, kazancı dB cinsinden hesaplanması şu şekildedir:

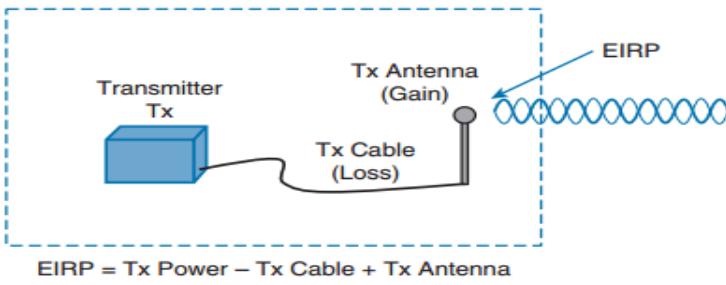
$$N=10 \log (2/1) = 3 \text{ dB}$$

Sinyal gücü yarısına indirilirse o zaman dB cinsinden kazanç:

$$N=10 \log (1/2) = -3 \text{ dB}$$

* Wireless sinyallerde sinyal gücü mW olarak kullanılır. Bu nedenle dB değeri **dBm** (desibel miliWatt) olarak kullanılabilir.

EIRP (effective isotropic radiated power): Sinyalin gücü sadece üretilen sinyal gücüne bağlı değildir. Şekilde de görüldüğü gibi, bir sinyalin gücü, üretim gücü, kullanılan kablodaki kayıp ve kullanılan antenden kaynaklanan kazancın toplamıdır. Buna **EIRP** denir.



İsotropik anten kullanımında (teorik) dBi kullanılır.

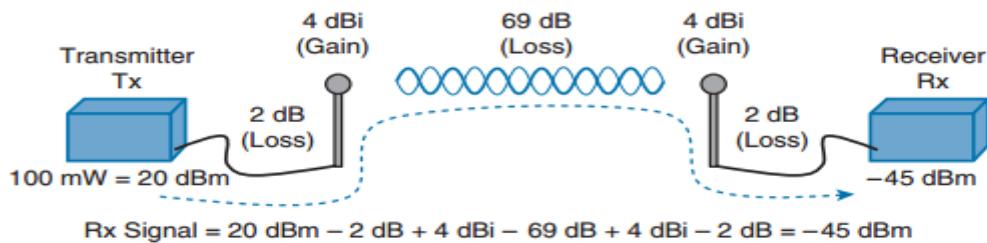
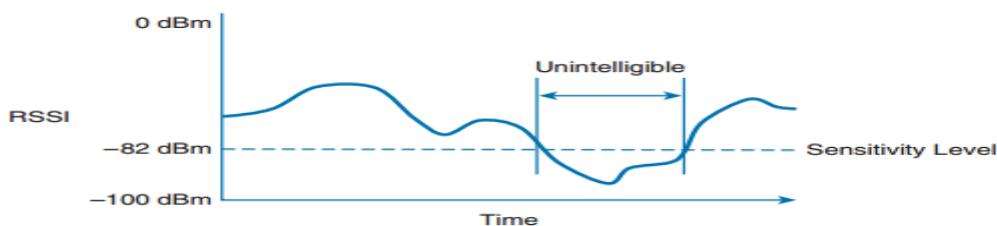


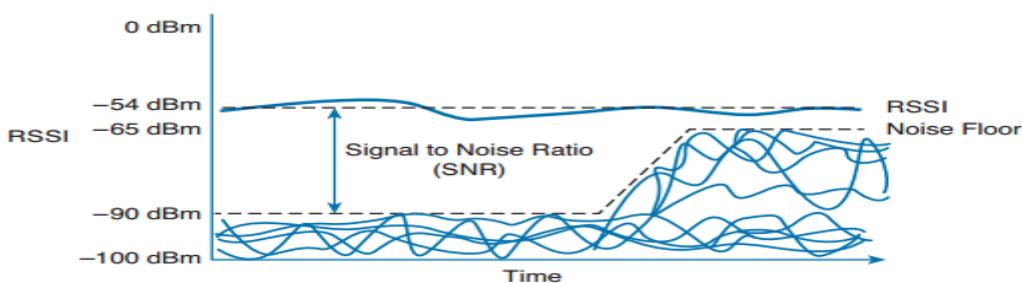
Figure 1-21 An Example of Calculating Received Signal Strength.

Alıcı taraf belli bir değerin altındaki sinyal gücünü algılayamaz. Buna **sensitivity level** denir. Anlık olarak algılanan sinyal gücüne Received Signal Strength Identifier (**RSSI**) denir.



*Not: dB değerinin sıfırın yakın olması tercih edilir.

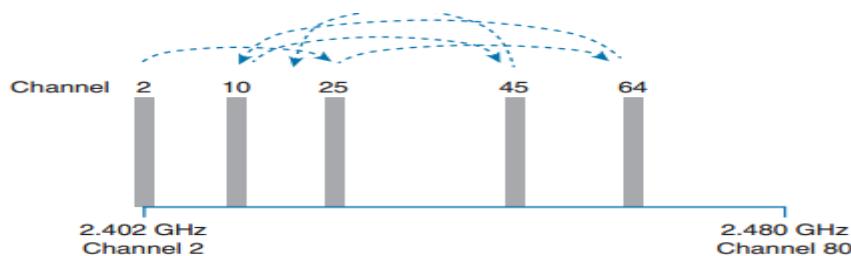
Noise (Gürültü): Yayın yapılan frekansta bulunan genellikle dış ortam kaynaklı sinyallerdir. Anlaşılabilirlik için sinyalin gücü, gürültüden daha yüksek olmalıdır. Sinyalin gücü ile gürültünün gücü arasındaki dB cinsinden farka **SNR** (Signal to Noise Ratio) denir. Yüksek SNR tercih edilir.



Modulasyon: RF sinyali üzerinden data taşınabilmesi gereklidir. RF sinyalinin karakteristiği gereği, Frekans, Genlik ve Faz kavramları değiştirilebilir. Belli bir mantık çerçevesinde yapılacak bu değişikliklere **modülasyon** denir.

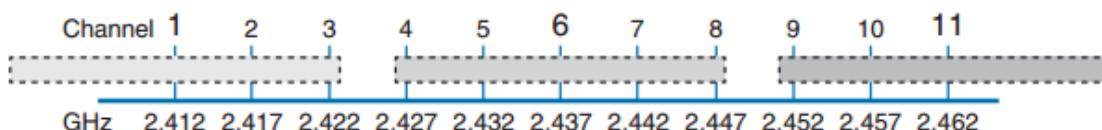
Yüksek bant genişliği yakalamak adına dar bir frekans aralığı yerine, belli bir frekans aralığı kullanılır. Buna **spread spectrum** denir. Bu kapsamda, **DSSS** (Direct-sequence spread spectrum), **FHSS** (Frequency-hopping spread spectrum) ve **OFDM** (Orthogonal frequency-division multiplexing) teknolojileri kullanılır.

FHSS: Kablosuz bant, her biri 1 MHz genişliğinde 79 kanala ayrılmıştır. Belli bir sıra dahilinde bu kanallar arasında atlamalar gerçekleştirilir. Alıcı-Verici taraflar bir kanaldan bilgi aldıktan-gönderdikten sonra, diğer bir kanala sırayla zıplama yaparlar. Aşağıdaki örnekte 2,25,64,10,45 kanalları arasında zıplamalar gösterilmiştir.



Bu yöntemler dış ortamda bulunan ve değişimle gürültüden kısmi olarak kurtulma sağlanabilir. Ancak kanalın 1 MHz genişliği düşük bir bant genişliği (1-2 Mb/s) sağlar. Yine ortamda birden çok gönderici varsa girişim oluşabilir.

DSSS: Her biri 22 MHz genişliğinde 11 Kanal kullanılır. Bu sebeple daha yüksek bant genişliği (11 Mb/s) sağlar.



Kanallar birbirine çakışmıştır. Bu sebeple bir ortamda ihtiyaç halinde çakışmayan kanallar kullanılmalıdır.

DSSS yapısı dış faktörlerden etkilenir. Bu nedenle CCK ve Barker Code yöntemleri ile data bozulması önlenmeye çalışılır. Burada temel düşünce, bir biti temsil etmek üzere birden çok bit dizisi (symbol) kullanılmıştır. Örneğin barker kod yapısında, 0 için data bit (10110111000) ; ve 1 için data bit (01001000111) şeklidir.

OFDM: DSSS yapısındaki kodlamadan dolayı bant genişliği en fazla 11 Mb/s olabilir. OFDM yapısında her biri 20 MHz genişliğinde kanallar kullanılır. Veriler çoklu kanallardan paralel olarak gönderilir. Her kanal, 64 alt-kanala ayrılmıştır (312,5 kHz). Ancak bu kanallardan 48 tanesi veri iletişimini için kullanılır.

KABLOSUZ İLETİŞİM DÜZENLEYİCİ KURUMLAR

1. Federal Communications Commissions (FCC)
2. European Telecommunications Standards Institute (ETSI)
3. ITU Radiocommunication (ITU-R)
4. Institute of Electrical and Electronic Engineers (IEEE)

IEEE 802.11 STANDARTLARI

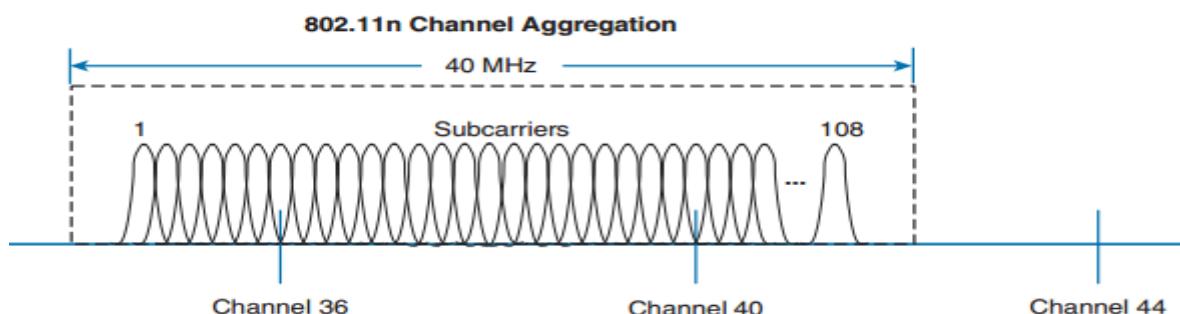
802.11: 1997 yılında IEEE tarafından yayınlandı. 2,4 GHz frekansında FHSS ve DSSS teknolojileri kullanılmıştır. Teorik bant genişlikleri FHSS için 1,2 Mb/s; DSSS için ise 2 Mb/s ‘dir.

802.11b: 802.11 standardının bant genişliğini artırmak amacıyla 1999 yılında yayınlandı. 2,4 GHz bandında 11 Mb/s bant genişliği sağlamaktadır.

802.11g: 2003 yılında OFDM teknolojisi kullanılarak, 2,4 GHz bandında 54 Mb/s bant genişliği sağlamak amacıyla geliştirildi.

802.11a: 802.11b ve 802.11g teknolojileri 2,4 GHz bandında sadece 3 tane çakışmayan kanal desteği sunmaktadır. 802.11a standardında 5GHz bandında, OFDM teknolojisi ile 54Mb/s bant genişliği sunar. Daha az girişim oluşurken daha fazla kanal kullanım imkânı vardır. 20 MHz genişliğindeki kanalların çakışma oranı çok daha azdır.

802.11n: 5GHz ve 2,4 GHz bandına çalışıp teorik 600 Mb/s bant genişliği sunar. Çoklu alıcı/verici (MIMO) antenleri kullanır. İki kanalı birleştirerek (2x20MHz) 40 MHz’lık kanallar kullanabilir.



802.11n yapısında bant genişliğinin artmasının tek sebebi çoklu anten kullanımı ve kanal birleştirme değildir. **Uzaysal çoğullama** (Spatial Multiplexing) olarak adlandırılan bir yöntem ile birbirinden yeterli uzaklıkta iki anten kullanılırsa, alıcıya bu antenlerden gelen bilgiler farklı zamanda (cisimlerden sekme sebebiyle vs.) ulaşacaktır. Yine iki verici gönderimi arasında 800 ns lik bir gecikme kullanılarak alıcının aynı frekansta ancak farklı zamanlarda sinyal alması sağlanır. Bunun yanında diğer 802.11 standartlarında olduğu gibi her frame için bir onay beklemez. Bunun yerine bir blok frame gönderiminden sonra onay beklenir. Bu sebeple gecikme daha aza indirgenir. Tüm bu etkilerin toplamı daha yüksek bant genişliği olarak yansır.

802.11ac: 5GHz bandında gigabit hızları destekleyen yeni standarttır.

RF SİNYALLERİ

Interference: Bir sinyal aynı frekanstaki ya da aynı kanaldaki başka bir frekans ile çakışması interference (girişim) olarak adlandırılır. Örneğin aynı kanalda iki farklı gönderici sinyal yayarsa **co-channel interference** oluşur.

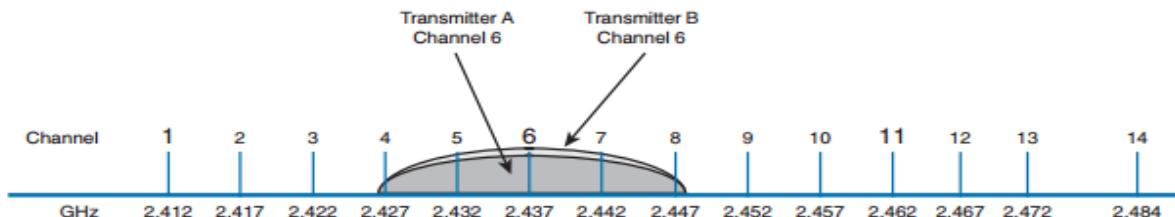


Figure 3-1 Co-Channel Interference.

Yine yakın (örtüsen) kanalların kullanımı da girişim oluşturabilir. Örneğin, 5. ve 7. kanalların aynı ortamda kullanımı da **neighboring channel interference** oluşturur.

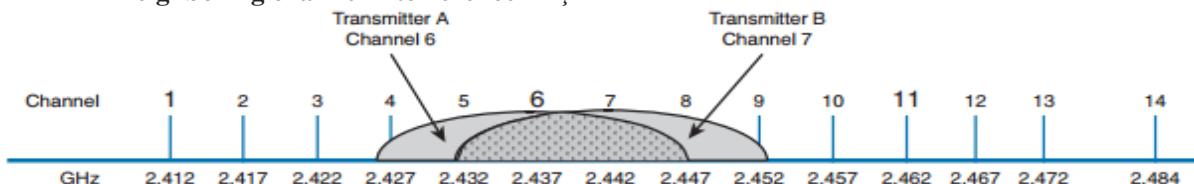


Figure 3-3 Adjacent Channel Interference.

Girişime sebep olan diğer bir kaynak da 802.11 olmayan micro dalga fırın gibi 2,4 GHz bandında yayın yapan cihazlardır.

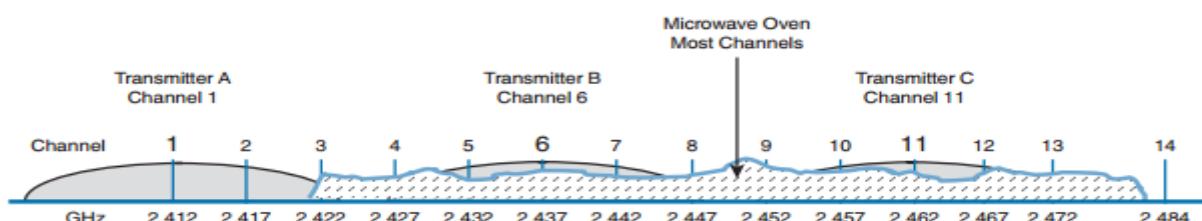


Figure 3-4 Non-802.11 Interference From a Microwave Oven.

Free Space Path Loss (FSPL): Sinyal antenden uzaklaştıkça herhangi bir fiziksel engel olmasa da genliğinde düşme olması, yani sinyalin gücünün azalmasıdır. Sinyal küresel olarak yayıldıktan sonra yüzeyde birim alana düşen enerji miktarı doğal olarak logaritmik düşüş gösterecektir. Mesafeye göre sinyal kaybının dB cinsinden değeri aşağıdaki gibi hesaplanır.

$$\text{FSPL} = 20\log_{10}(d) + 20\log_{10}(f) + 32.44 \quad (\text{f: Mhz cinsinden frekans; d: km cinsinden uzaklık})$$

* sinyal gücü kaybı mesafe ve frekans ile doğru orantılıdır. Yani yüksek frekansın (Ör: 5 GHz) kaybı düşük frekansa göre daha fazladır.

Sinyal gücü kaybına, sinyalin fiziksel dünyadaki nesneler ile olan etkileşimi de (yansıma, soğurma, saçılma ve kırılma gibi) neden olabilir.

Reflection (Yansıma): Radyo sinyallerinin bir yüzeyden yansımasıdır. Bina içinde genellikle metal nesneler buna sebep olurken, bina dışında su yüzeyleri buna neden olur.

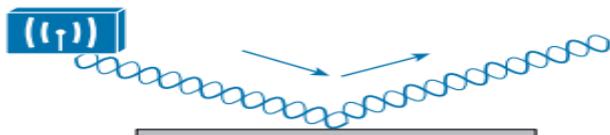
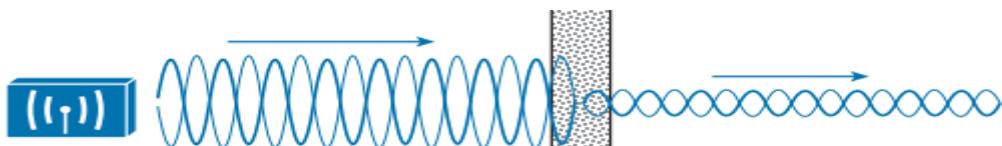


Figure 3-8 Reflection of an RF Signal.

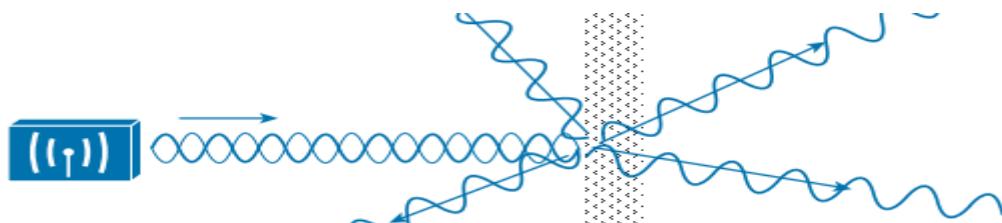
Yanışyan sinyaller farklı zamanlarda alıcıya ulaşır. Alıcı MIMO teknolojisi kullanmıyorsa bu durum sinyal bozulmasına neden olabilir.

Absorption (Soğurma): RF sinyaller bir maddeden geçtiğinde, sinyal sönmelenebilir (**attenuation**). Yani sinyal gücünde kayıp yaşanır. Farklı maddeler farklı oranlarda soğurma gerçekleştirir.

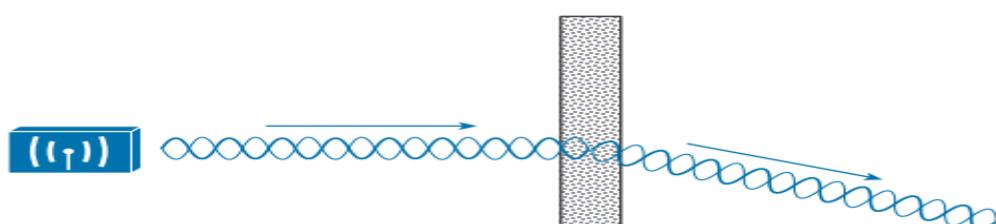


Örneğin alçı; -4 dB kayıp oluştururken duvar -12 dB kayba neden olur. Yine bina dışında ağaçlar, yağmur, sis gibi etkiler de sinyalin soğulmasına neden olur.

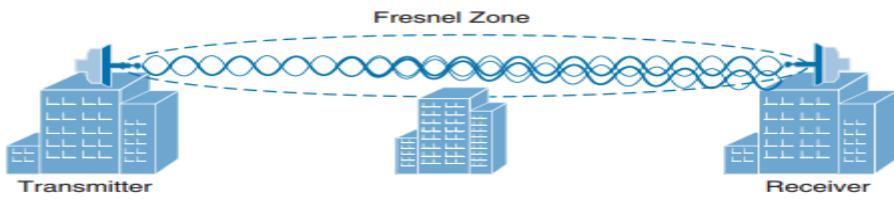
Scattering (Saçılma): RF sinyaller pürüzlü yüzeylerden geçerken farklı doğrultularda saçılabilir.



Refraction (Kırılma): RF sinyaller farklı yoğunluktaki bir ortam ile karşılaşlığında kırmızılaşır.



Fresnel Zone : Özellikle uzak mesafelerdeki kablosuz iletişimde alıcı-vericilerin birbirlerinin görüş alanında olmaları ve aralarında sinyali bozacak bina, ağaç gibi nesnelerin olmaması gereklidir.



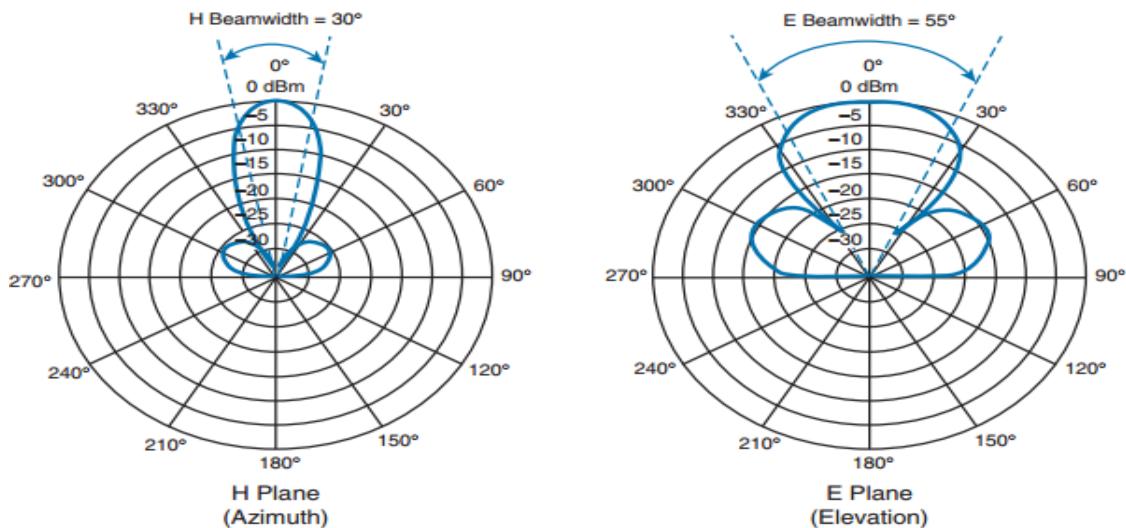
ANTENLER

Antenler pasif cihazlardır. Antenin sağladığı kazanç (gain), RF sinaylinin enerjisini ne oranda istenilen doğrultuya gönderebildiği ile ilgilidir. Antenin RF sinaylini doğrultmasının XYZ eksinindeki görüntüsüne patern denir. Teorik olarak üretilmesi imkansız olan isotropik antenler, RF sinyallerini küresel olarak yayarlar. Aşağıdaki şekilde patern türleri gösterilmektedir.

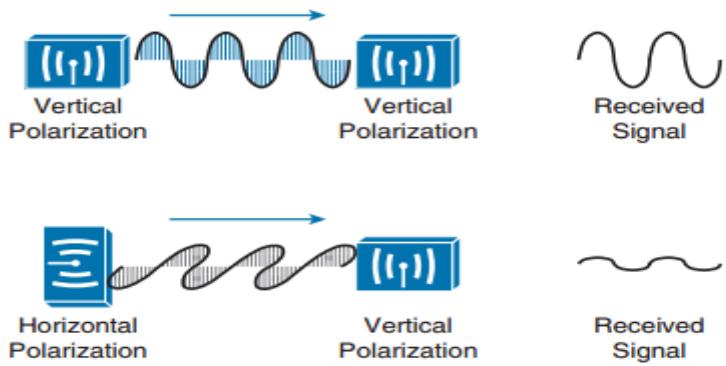


Anten paternleri üreticilerin dokümanlarında belirtilir.

BeamWidth: Anten paterninin yatay (H) ve dikey eksende (E) gösterimidir.



Polarization (Polarizasyon): RF sinyalinin elektrik alan bileşeninin hangi doğrultuda (yatay=horizontal veya dikey=vertical) yayıldığını gösterir. Çoğu cisco antenler dikey doğrultuda yayın yaparlar.



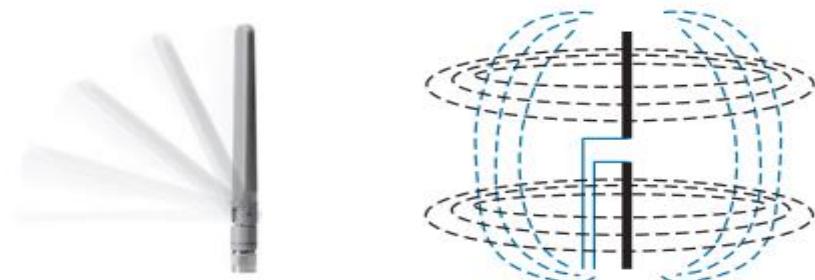
Anten polarizasyonu tek başına çok kritik bir öneme sahip değildir. Ancak, alıcı ile verici aynı polarizasyon türünü kullanmıyorsa sinyal gücünü çok zayıf olarak algılayabilir.

Not: *Antenin paternine göre olmasi gereken dışında döndürülmesi polarizyonun yanlış olmasına neden olabilir.*

Anten Türleri

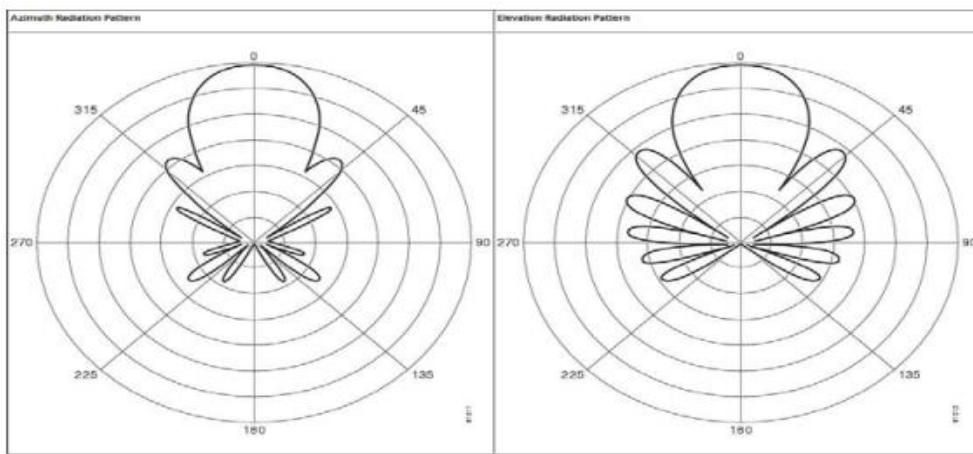
Wireless antenler farklı şekillerde, türlerde ve paternlerde olabilir. Ancak genel olarak **omnidirectional** ve **bidirectional** olmak üzere iki tür anten vardır.

OmniDirectional Anten: XY ekseninde sinyal yayan antenlerdir. Yaygın olarak kullanılan Dipol antenler bu tür antenlerdir.



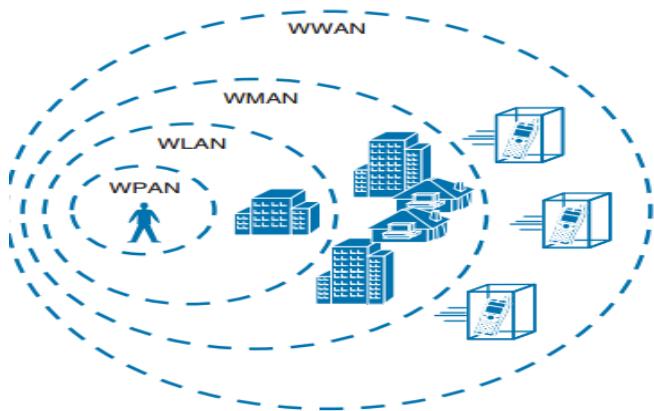
Sinyali belli bir doğrultuya yönlendirmediklerinden kazançları azdır (2 – 5 dBi).

Bidirectional Anten: RF sinyalini belirli bir doğrultuya yönlendirdiklerinden daha fazla kazanç sağlar. 2,4 GHz frekans bandında 6-8 dBi; 5 GHz bandında ise 7-10 dBi kazanç sağlar. Yine sıkılıkla kullanılan **Yagi** ve **dish** antenler bu tür antenlerdir.



WIRELESS AĞ TÜRLERİ

Kablosuz ağlar coğrafi kapsama alanlarına göre dört tipe ayrılır.



WPAN: 2,4 GHz bandında, 7-10 m mesafelerde çalışan çoğunlukla düşük güçteki, daha çok kişisel amaçlı kullanılan BlueTooth ve ZigBee gibi IEEE 802.15 ağlardır.

WLAN: IEEE 802.11 standardında 100m'ye kadar olan mesafelerde 2,4 GHz ve 5 GHz bantlarında çalışan yerel ağlardır.

WMAN: Coğrafi olarak bir şehrin bölgesi gibi nisbeten daha büyük ağlarda çalışan, WiMAX gibi (802.16) ağlardır. Genellikle lisanslı frekanslar kullanılır.

WWAN: Ülkesel bazda çalışabilen, çoğunlukla mobil telefon kullanıcıları destekleyen geniş çaplı ağlardır. Yine lisanslı frekanslar kullanılır.

WIRELESS LAN TOPOLOJİLERİ

BSS (Basic Service Set): Bir AP aracılığıyla istasyonların birbirlerine bağlanabildikleri, LAN'lardır.

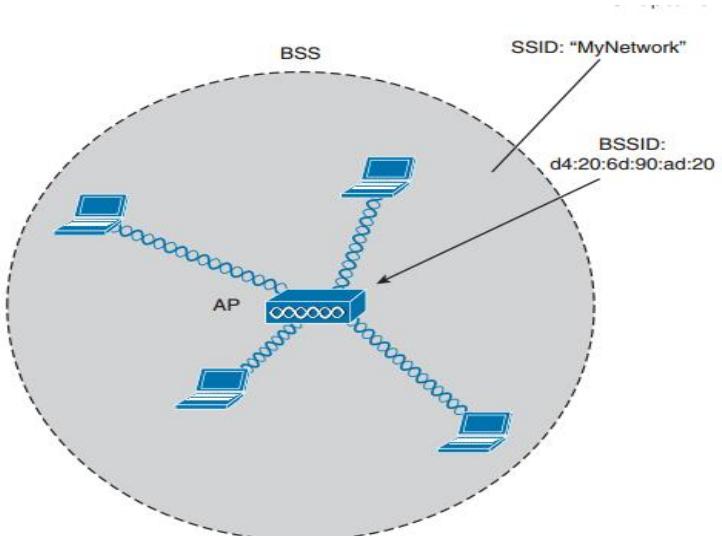
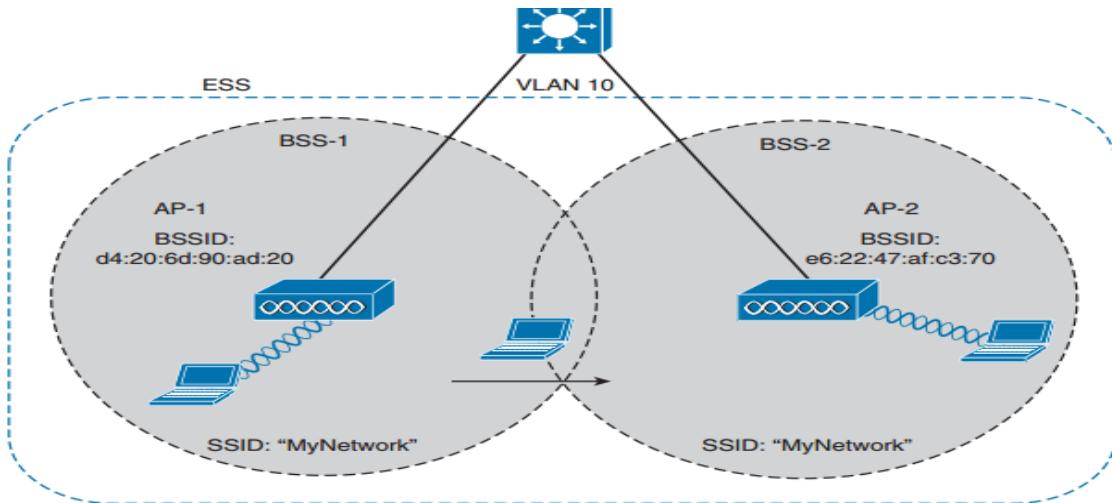


Figure 5-5 An 802.11 Basic Service Set.

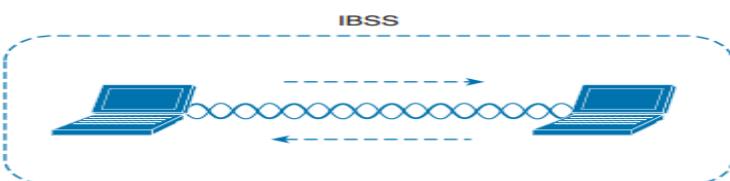
Bir BSA hücresinde AP'ye bağlanan istemcilere STA denir. STA'lar arasında doğrudan bir iletişim yoktur. Tüm iletişimler AP üzerinden gerçekleşir. AP bu yapıda Ethernet HUB gibi çalışır. Aslında AP, kablolu bir altyapıya bağlıdır ve 802.3 çerçeveleri 802.11 çerçevelere dönüştürür.

ESS (Extended Service Set): Birden fazla AP'nin bağlandığı ağlardır. Kullanıcıların mobilitesini sağlamak ve kapsama alanını artırmak amacıyla genişletilmiştir.



İstemciler yakın AP ile ilişkilendirilir. Bir AP'den geçip diğer AP'ye bağlanmasına **roaming** denir.

Independent Basic Service Set (IBSS): AP olmadan iki veya daha fazla istemcinin birbirlerine bağlandığı geçici (ad-hoc) ağlardır.



Repeater : Kapsama alanını artırmak amacıyla kablosuz köprü kullanılmıştır.

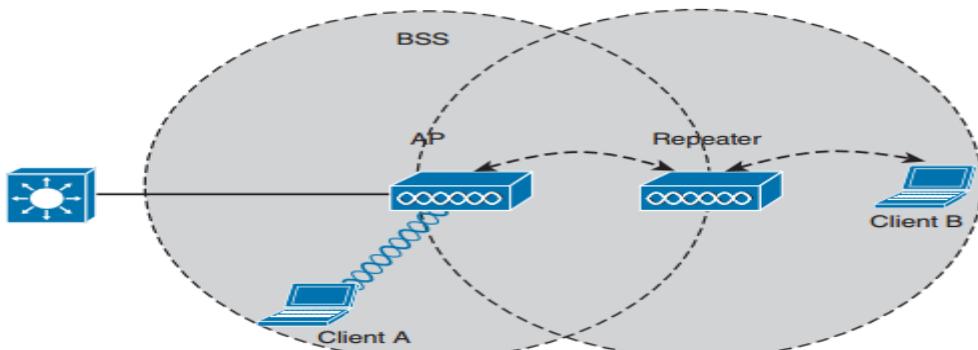


Figure 5-11 Extending the Range of an AP with a Wireless Repeater.

Outdoor Bridge: Birbirlerine uzak mesafede bulunan iki farklı kablolu ağın, kablosuz olarak birbirlerine bağlandıkları ağlardır. Farklı lokasyonlardaki iki farklı kampüs ağını bağlamak gibi.



Figure 5-13 A Point-to-Point Outdoor Bridge.

802.11 Frame Yapısı

802.11 frame yapısını anlayabilmek için öncelikle 802.3 ethernet frame yapısını ve Ethernet iletişiminin nasıl olduğunu bilmek gereklidir.

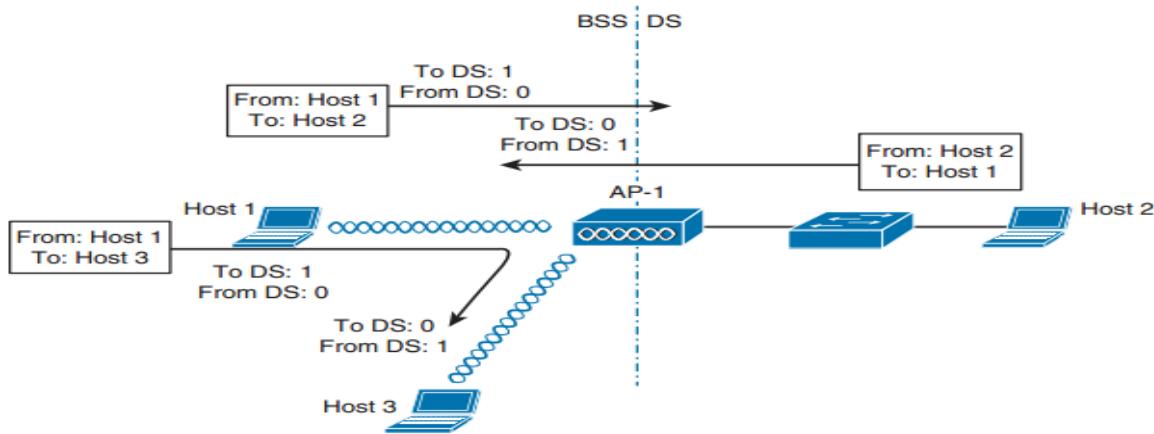
Preamble	Destination Address	Source Address	Type	Data	FCS
8 Bytes	6 Bytes	6 Bytes	2 Bytes	46–1500 Bytes	4 Bytes

Figure 6-1 IEEE 802.3 Ethernet Frame Format.

802.3 Ethernet yapısında cihazlar birbirleri ile iletişime geçerken Hedef ve Kaynak MAC adreslerini kullanmaktadır. Ancak 802.11 yapısında AP üzerinden haberleşmektedirler. Bu nedenle istemciler AP'nin MAC adres bilgisine ihtiyaç duyarlar (**BSSID=BSS Identifier**).

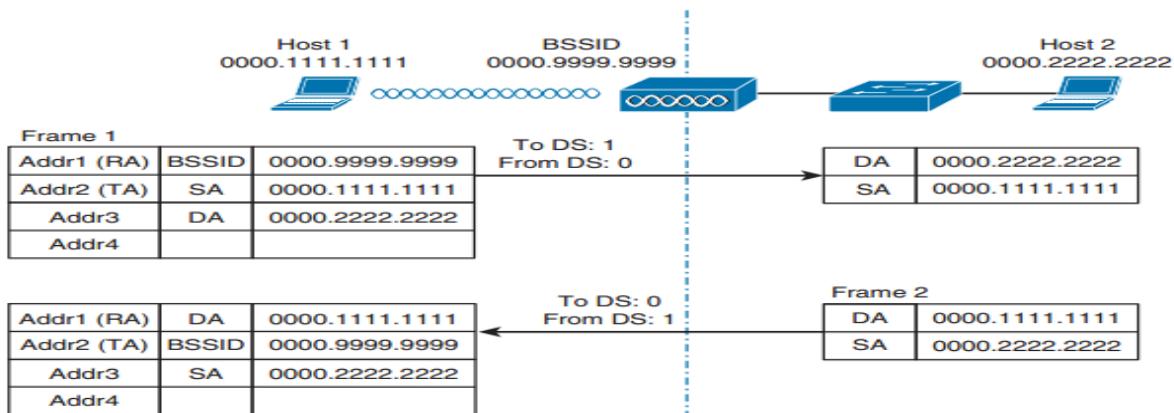
Frame Control	Duration /ID	Address1	Address2	Address3	Sequence Control	Address4	Data	FCS
2 Bytes	2 Bytes	6 Bytes	6 Bytes	6 Bytes	2 Bytes	6 Bytes	0–2312 Bytes	4 Bytes
<hr/>								
Protocol Version	Type	Subtype	To DS	From DS	More Frag	Retry	Pwr Mgmt	More Data
Bits: 2	2	4	1	1	1	1	1	1

802.11 frame boyutu maximum 2346 byte olabilir. Frame yapısındaki ilk iki byte Frame Control Field olarak adlandırılır. AP'ler genellikle bir dağıtım sistemine bağlıdır (DS). Yani çerçeveler DS katmanından istemcilere gidebildiği gibi, istemcilerden de DS katmanına gidebilir. Frame yapısındaki ToDS ve FromDS bitleri bu amaçla kullanılır.



802.11 Frame yapısında dört adres alanı bulunur. İlk adres (adres 1) alıcı istemcinin MAC adresi (RA) iken ikinci adres (adres 2) göndericinin MAC adresini (TA) içerir. Adres 3 ve Adres 4 alanları ToDS ve FromDS bitlerinin durumlarına göre değişkenlik gösterir.

Aşağıdaki örnek senaryoda bu adres alanlarının nasıl değiştiği gösterilmektedir.



Kablosuz Ortam Erişimi: Kablosuz erişim ortamı tipki hub gibi paylaşaklı bir ortamdır, aynı anda gönderim çakışmaya (collision) neden olur. Bu nedenle istemciler veri göndermeden önce hattı kontrol etmelidirler (carrier sense).

Physical Carrier Sense: Kablosuz bir istemci veri göndermiyor olsa da, diğer istemcilerin çerçeveleri kendisine ulaşır. Bu çerçevenin hedef MAC adresi istemcinin adresi ise bunu decapsulation işlemeye alır. Aksi durumda paylaşaklı ortamın kullanımında olduğunu (busy) anlar. Bu yönteme Clear Channel Assessment (CCA) denir.

Virtual Carrier Sense : Veri gönderecek istemci, göndereceği verinin boyutuna göre kablosuz kanalı bir süreliğine rezerve eder. Bu süre frame yapısında **Duration** alanına mikro saniye cinsinden yazılır. Böylece ortamdaki diğer istemciler bu çerçeveyi aldıklarından ne kadar süre

beklemeleri gerektiğini kestirebilirler. Bu değerin (NAV=Network Allocation Vector) bitiminde veri göndermeye başlarlar. Ancak yine de bu yapıda collision oluşabilir.

Bu her iki yöntem de fiziksel katmanda bu kontrolü gerçekleştirir. Kablolu ortamlarda collision algılanması CSMA/CD ile mümkündür. Kablosuz ortamlarda ise collisiondan kaçınmak amacıyla CSMA/CA mekanizması kullanılır (Layer 2).

Kablosuz bir istemciler veri göndereceği zaman rastgele bir bekleme süresince bekler (**backoff timer**). Bu sayaç geriye doğru sayar ve 0 olduğunda ortam uygunsa veri gönderimi başlar. Dolayısı ile daha düşük timer değerine sahip istemci daha erken veri gönderebilir.

Backoff Timer süresi 0 olmadan önce wireless kanal başka bir iletişimde kullanılmaya başlanırsa, Backoff Timer durdururlur (pause) ve NAV değeri bu süreye eklenir. Bir istemci veri gönderimi yapmadan önce bu toplam sürelerin (**contention window**) sıfırlanması gereklidir.

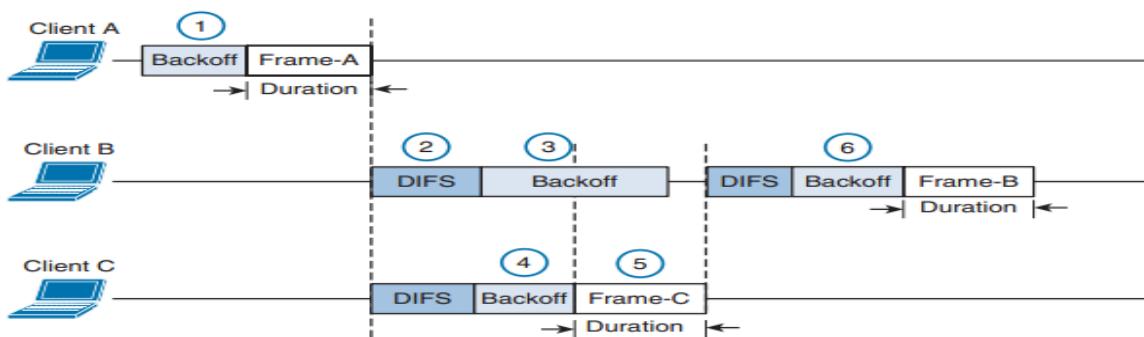
Çok erişimli bir ortamda gönderilen 802.3 Ethernet çerçeveler arasında belirli bir bekleme süresi (**Inter Frame Space=IFS**) vardır. Kablosuz 802.11 çerçeveler için birkaç farklı IFS vardır.

RIFS: 802.11n ortamında data frameler arasındaki bekleme süresidir.

SIFS : Data Frameler arasında veya Data- Ack Frameler arasındaki bekleme süresidir.

DIFS : Çoğu standart için varsayılan bekleme süresidir.

EIFS : Collision oluştuktan sonra beklenmesi gereken süredir.



802.11 FRAME TÜRLERİ

Management Frame: BSS'lerin tanıtılmamasında, istemcilerin BSS'e dahil olmalarını ya da ayrılmalarını yöneten çerçevelerdir. 14 farklı türde management frame vardır. Bunlardan bazıları:

1. Beacon Frame: AP tarafından BSS'i tanıtmak amacıyla 100 ms periyotlarla gönderilir. İçerisinde; kabul edilen data hızları, **ssid** adı (opsiyonel) ve üreticinin belirlediği bilgiler gibi bilgiler taşınır. Böylece istemciler BSS hakkında bilgi sahibi olurlar. (**Passive Scan**)

```

■ Frame 78: 71 bytes on wire (568 bits), 71 bytes captured (568 bits)
■ IEEE 802.11 Beacon frame, Flags: .....
    Type/Subtype: Beacon frame (0x08) ←
■ Frame Control: 0x0080 (Normal)
    Duration: 0
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Source address: AskeyCom_68:4d:1b (00:24:d2:68:4d:1b)
    BSS Id: AskeyCom_68:4d:1b (00:24:d2:68:4d:1b)
    Fragment number: 0
    Sequence number: 2649
■ IEEE 802.11 wireless LAN management frame
■ Fixed parameters (12 bytes)
■ Tagged parameters (35 bytes)
    ■ Tag: SSID parameter set: 4NG3B
    ■ Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 6, 9, 12, 18, [Mbit/sec]
    ■ Tag: DS Parameter set : Current Channel: 1
    ■ Tag: Traffic Indication Map (TIM): DTIM 0 of 0 bitmap
    ■ Tag: ERP Information
    ■ Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]

```

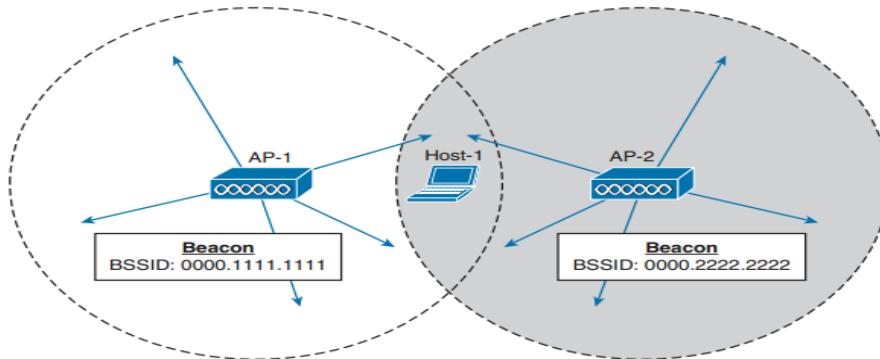


Figure 6-9 Using a Passive Scan to Discover BSSs.

Tipik bir beacon frame yapısı aşağıdaki gibidir.

```

■ IEEE 802.11 Beacon frame, Flags: .....
■ IEEE 802.11 wireless LAN management frame
■ Fixed parameters (12 bytes)
    Timestamp: 0x000000005ca7c8e9
    Beacon Interval: 0.104448 [seconds]
■ Capabilities Information: 0x0411
■ Tagged parameters (222 bytes)
    ■ Tag: SSID parameter set: TestSSID
    ■ Tag: Supported Rates 1, 2, 5.5(B), 6, 9, 11, 12, 18, [Mbit/sec]
    ■ Tag: DS Parameter set : Current Channel: 1
    ■ Tag: Traffic Indication Map (TIM): DTIM 0 of 0 bitmap
    ■ Tag: Country Information: Country Code US, Environment Any
    ■ Tag: QBSS Load Element 802.11e CCA Version
    ■ Tag: ERP Information: no Non-ERP STAs, do not use protection, long preambles
    ■ Tag: HT Capabilities (802.11n D1.10)
    ■ Tag: RSN Information
    ■ Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]
    ■ Tag: Reserved tag Number
    ■ Tag: Cisco CCX1 CKIP - device name
    ■ Tag: Cisco Unknown 4 Beacon Frame Len 6
    ■ Tag: Vendor Specific: Microsoft: WME
    ■ Tag: Vendor Specific: Aironet: Aironet Unknown
    ■ Tag: Vendor Specific: Aironet: Aironet CCX version = 5
    ■ Tag: Vendor Specific: Aironet: Aironet Unknown
    ■ Tag: Vendor Specific: Aironet: Aironet Unknown

```

SSID adı, desteklenen hızlar gibi bilgiler gösterilmektedir.

```
■ IEEE 802.11 wireless LAN management frame
  ■ Fixed parameters (12 bytes)
    Timestamp: 0x00000e5e9822d8e8
    Beacon Interval: 0.104448 [seconds]
    ■ Capabilities Information: 0x0431
  ■ Tagged parameters (137 bytes)
    ■ Tag: SSID parameter set:
      Tag Number: SSID parameter set (0)
      Tag length: 1
      SSID:
```

SSID Broadcast devre dışı bırakıldığında ilgili kısım boş gelecektir.

```
■ Tag: RSN Information
  Tag Number: RSN Information (48)
  Tag length: 20
  RSN Version: 1
  ■ Group Cipher Suite: 00-0f-ac (ieee8021) AES (CCM)
    Pairwise Cipher Suite Count: 1
  ■ Pairwise Cipher Suite List 00-0f-ac (ieee8021) AES (CCM)
    Auth Key Management (AKM) Suite Count: 1
  ■ Auth Key Management (AKM) List 00-0f-ac (ieee8021) WPA
  ■ RSN Capabilities: 0x0028
```

RSN (Robust Security Network) information kısmında güvenlik özellikleri gösterilmektedir. Bu örnekte IEEE 802.11i desteğinin olduğu görülmektedir.

```
■ Tag: HT Capabilities (802.11n D1.10)
  Tag Number: HT Capabilities (802.11n D1.10) (45)
  Tag length: 26
  ■ HT Capabilities Info: 0x186e
  ■ A-MPDU Parameters: 0x001b
  ■ RX Supported Modulation and Coding scheme set: MCS Set
  ■ HT Extended Capabilities: 0x0000
  ■ Transmit Beam Forming (TxBF) Capabilities: 0x0000
  ■ Antenna Selection (ASEL) Capabilities: 0x00
```

High Throughput (HT) kısmında hangi standardın desteklendiği görülmektedir. TxBF ve ASEL değerlerinin 0 olması multiple anten kullanılmadığı anlamına gelir.

2. Probe Frame: İstemciler tarafından BSS hakkında bilgi almak amacıyla gönderilen sorgulardır. AP, bu mesaja cevap verir. (**Active Scan**)

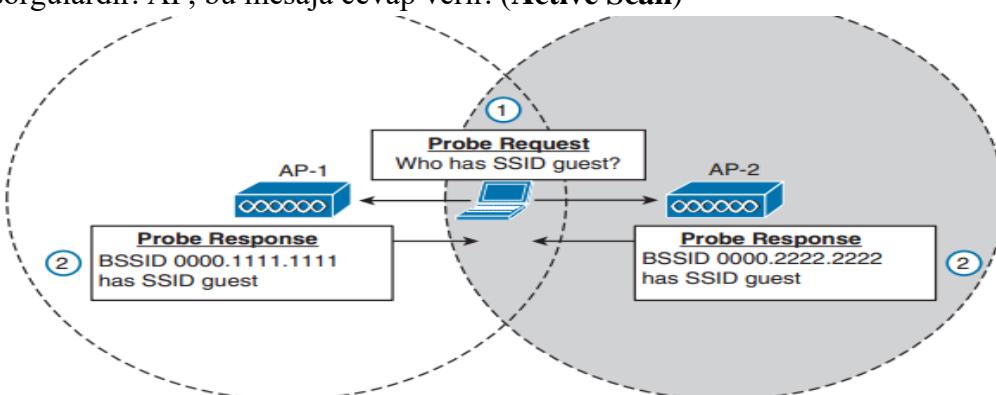


Figure 6-10 Using an Active Scan to Discover BSSs.

```

Frame 4199: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
IEEE 802.11 Probe Request, Flags: .....
  Type/Subtype: Probe Request (0x04)
  Frame Control: 0x0040 (Normal)
  Duration: 0
  Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
  Source address: IntelCor_3d:d6:e0 (24:77:03:3d:d6:e0)
  BSS Id: Broadcast (ff:ff:ff:ff:ff:ff)
  Fragment number: 0
  Sequence number: 3275
IEEE 802.11 wireless LAN management frame
  Tagged parameters (54 bytes)
    Tag: SSID parameter set: blizzard
    Tag: Supported Rates 1, 2, 5.5, 11, 6, 9, 12, 18, [Mbit/sec]
    Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]
    Tag: HT Capabilities (802.11n D1.10)
      [Frame is ignored: False]
      [Protocols in frame: wlan]
IEEE 802.11 Probe Response, Flags: ....R...
  Type/Subtype: Probe Response (0x05)
  Frame Control: 0x0850 (Normal)
  Duration: 0
  Destination address: IntelCor_70:f7:96 (00:1f:3c:70:f7:96)
  Source address: AskeyCom_68:4d:1b (00:24:d2:68:4d:1b)
  BSS Id: AskeyCom_68:4d:1b (00:24:d2:68:4d:1b)
  Fragment number: 0
  Sequence number: 914
IEEE 802.11 wireless LAN management frame
  Fixed parameters (12 bytes)
  Tagged parameters (29 bytes)
    Tag: SSID parameter set: 4NG3B
    Tag: Supported Rates 1(B), 2(B), 5.5(B), 6, 9, 11(B), 12, 18, [Mbit/sec]
    Tag: DS Parameter set : Current Channel: 1
    Tag: ERP Information
    Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]

```

3. Authentication / De-Authentication Frame: İstemicinin BSS'e dahil olması için öncelikle AP'ye authentication request çerçevesi gönderir. AP de authentication Response ile cevap verir.

```

IEEE 802.11 Authentication, Flags: .....
  Type/Subtype: Authentication (0x0b)
  Frame Control: 0x0080 (Normal)
  Version: 0
  Type: Management frame (0)
  Subtype: 11
  Flags: 0x0
    .... .00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x00)
    .... .0.. = More Fragments: This is the last fragment
    .... 0... = Retry: Frame is not being retransmitted
    ...0 .... = PWR MGT: STA will stay up
    ..0. .... = More Data: No data buffered
    .0... .... = Protected flag: Data is not protected
    0.... .... = Order flag: Not strictly ordered
  Duration: 0
  Destination address: Private_e8:8a:e4 (f0:a2:25:e8:8a:e4)
  Source address: AskeyCom_6a:25:a0 (00:21:63:6a:25:a0)
  BSS Id: AskeyCom_6a:25:a0 (00:21:63:6a:25:a0)
  Fragment number: 0
  Sequence number: 3787
IEEE 802.11 wireless LAN management frame
  Fixed parameters (6 bytes)
    Authentication Algorithm: Open System (0)
    Authentication SEQ: 0x0002
    Status code: Successful (0x0000)

```

DisAuthentication (deauthentication) tek yönlüdür ve AP bu mesajı kabul etmek zorundadır.

```

IEEE 802.11 Deauthentication, Flags: ...P.M.T
  Type/Subtype: Deauthentication (0x0c)
Frame Control: 0x15C3 (Normal)
  Version: 3
  Type: Management frame (0)
  Subtype: 12
Flags: 0x15
  .... ..01 = DS status: Frame from STA to DS via an AP (To DS: 1 From DS: 0) (0x01)
  .... .1.. = More Fragments: More fragments follow
  .... 0... = Retry: Frame is not being retransmitted
  ...1 .... = PWR MGT: STA will go to sleep
  ..0. .... = More Data: No data buffered
  .0... .... = Protected flag: Data is not protected
  0... .... = Order flag: Not strictly ordered
Duration: 61213
Destination address: 5a:50:e3:83:52:d7 (5a:50:e3:83:52:d7)
Source address: dd:29:38:a9:5d:d6 (dd:29:38:a9:5d:d6)
BSS Id: bf:73:4b:a8:6c:1c (bf:73:4b:a8:6c:1c)
Fragment number: 9
Sequence number: 2380
Data (56 bytes)
  Data: cb7cef007b8d9eb3bc6d59542bd2e230133a57d432f3c2ad...
  [Length: 56]

```

4. Association , disAssociation ve reAssociation Frame: Kimlik doğrulamanın ardından istemci ağa dahil olmak için Association Request çerçevesi gönderilir.

```

IEEE 802.11 Association Request, Flags: .pm....T
  Type/Subtype: Association Request (0x00)
Frame Control: 0x6101 (Normal)
  Version: 1
  Type: Management frame (0)
  Subtype: 0
Flags: 0x61
  .... ..01 = DS status: Frame from STA to DS via an AP (To DS: 1 From DS: 0) (0x01)
  .... .0.. = More Fragments: This is the last fragment
  .... 0... = Retry: Frame is not being retransmitted
  ...0 .... = PWR MGT: STA will stay up
  ..1. .... = More Data: Data is buffered for STA at AP
  .1.. .... = Protected flag: Data is protected
  0... .... = Order flag: Not strictly ordered
Duration: 18079
Destination address: 26:ff:a1:96:9d:c3 (26:ff:a1:96:9d:c3)
Source address: f5:9a:d8:d2:fa:95 (f5:9a:d8:d2:fa:95)
BSS Id: e8:b0:db:c8:bb:3f (e8:b0:db:c8:bb:3f)
Fragment number: 3
Sequence number: 726

```

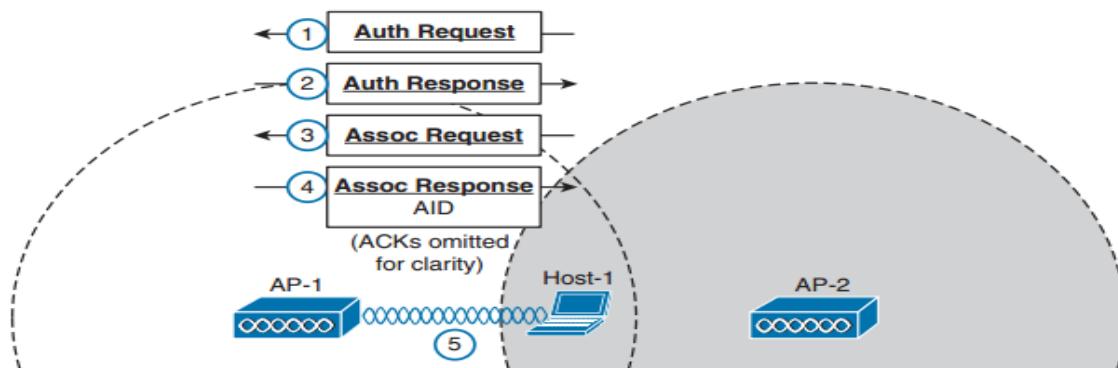
Parametrelerin uygunluğu sağlanırsa Association Response ile istemcinin BSS'e dahil olması sağlanır.

```

IEEE 802.11 Association Response, Flags: op...MF.
Type/Subtype: Association Response (0x01)
└ Frame Control: 0xC612 (Normal)
  Version: 2
  Type: Management frame (0)
  Subtype: 1
  └ Flags: 0xC6
    .... ..10 = DS status: Frame from DS to a STA via AP (To DS: 0 From DS: 1) (0x0
    .... .1.. = More Fragments: More fragments follow
    .... 0... = Retry: Frame is not being retransmitted
    ...0 .... = PWR MGT: STA will stay up
    ..0. .... = More Data: No data buffered
    .1.. .... = Protected flag: Data is protected
    1.... .... = Order flag: Strictly ordered
Duration: 62616
Destination address: 1f:c0:9d:76:35:b5 (1f:c0:9d:76:35:b5)
Source address: e8:55:3a:77:58:da (e8:55:3a:77:58:da)
BSS Id: b2:d5:3e:69:49:40 (b2:d5:3e:69:49:40)
Fragment number: 1
Sequence number: 2160
└ WEP parameters
Data (60 bytes)
Data: 21971998e7a9d44be83c8322b3136115b3fac714f3839862...
[Length: 60]

```

Bu response içinde istemci için unique bir AID (1-2007) değeri bulunur. İstemci ayrılmak isterse disAssociation Frame ile ayrılabilir. Ayrıca, aynı SSID içinde kalmak şartıyla bir BSS'ten diğerine geçmek için (**roaming**) yeni AP'ye reAssociate Request çerçevesi gönderir.



Not: İstemci associate olduğunda AP bu istemciye ait AID değerini 5 dk boyunca tutar. 5 dk içinde istemci ile haberleşme sağlanamazsa (örneğin istemci kapsama alanı dışına çıktığında) bu kayıt silinir.

Control Frame: Verilerin wireless kanallar üzerinden doğru bir şekilde iletimini kontrol etmek amacıyla kullanılan, sadece header kısmı olan (**no data payload**) çerçevelerdir. 9 farklı kontrol çerçevesi vardır.

1. ACK: Unicast çerçeve alındığına (hata yoksa) kaynak STA'ya gönderilen kısa çerçevelerdir. Eğer belli bir sürede ACK gönderilmezse kaynak STA veriyi tekrar gönderir.

```
IEEE 802.11 Acknowledgement, Flags: opmp.MFT
  Type/Subtype: Acknowledgement (0x1d)
  □ Frame Control: 0xF7D7 (Normal)
    Version: 3
    Type: Control frame (1)
    Subtype: 13
  □ Flags: 0xF7
    .... .11 = DS status: Frame part of WDS from one AP to another AP (To DS: 1 From DS: 1) (0x03)
    .... .1.. = More Fragments: More fragments follow
    .... 0... = Retry: Frame is not being retransmitted
    ...1 .... = PWR MGT: STA will go to sleep
    ..1. .... = More Data: Data is buffered for STA at AP
    .1.. .... = Protected flag: Data is protected
    1... .... = Order flag: Strictly ordered
Duration: 12021
```

2. Block ACK: Belli bir block çerçeve alındığına dair gönderilen kısa onay çerçevesidir.

3. PS-Poll: İstemci buffer'ında bulunan çerçevenin bir sonrakini istemek amacıyla AP'ye gönderilir.

4. RTS / CTS: Kanalı rezerve etmek için kullanılır.

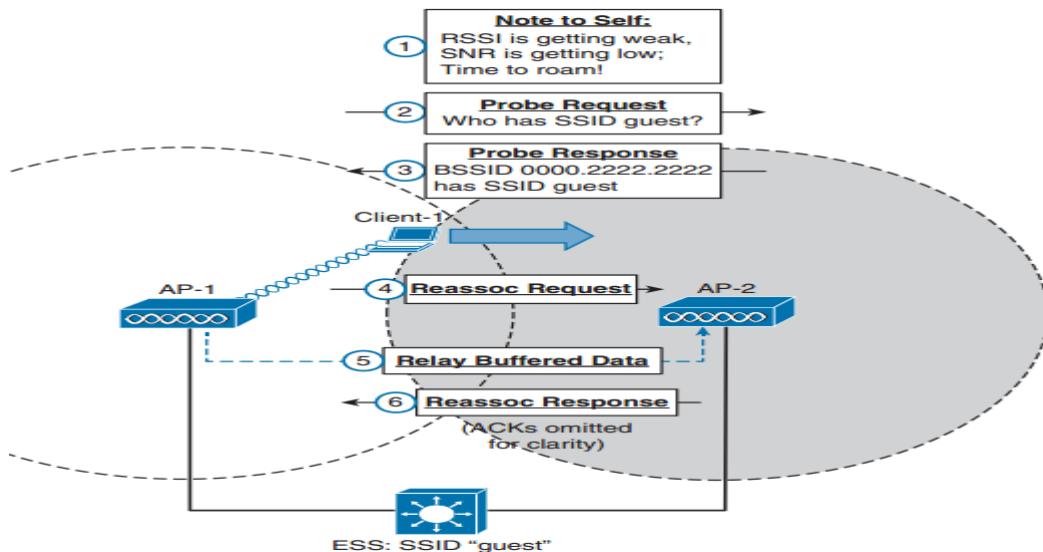
```
IEEE 802.11 Clear-to-send, Flags: o.m.R.F.
  Type/Subtype: Clear-to-send (0x1c)
  □ Frame Control: 0xAAC6 (Normal)
    Version: 2
    Type: Control frame (1)
    Subtype: 12
  □ Flags: 0xAA
    .... .10 = DS status: Frame from DS to a STA via AP(To DS: 0 From DS: 1) (0x02)
    .... 0.. = More Fragments: This is the last fragment
    .... 1... = Retry: Frame is being retransmitted
    ...0 .... = PWR MGT: STA will stay up
    ..1. .... = More Data: Data is buffered for STA at AP
    .0.. .... = Protected flag: Data is not protected
    1... .... = Order flag: Strictly ordered
```

Data Frame: İstemciler arasında veri göndermek amacıyla kullanılır.

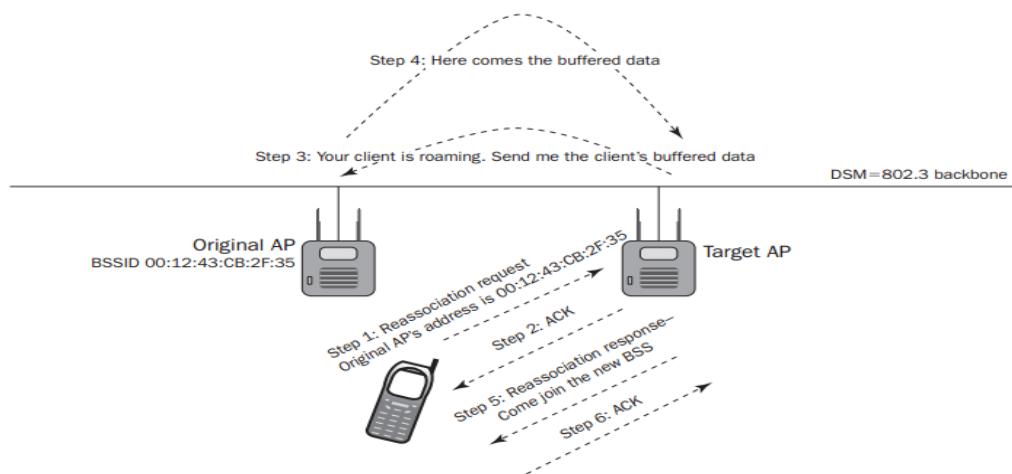
```
IEEE 802.11 Data, Flags: .p....F.
  Type/Subtype: Data (0x20)
  □ Frame Control: 0x4208 (Normal)
    Version: 0
    Type: Data frame (2)
    Subtype: 0
  □ Flags: 0x42
    .... .10 = DS status: Frame from DS to a STA via AP(To DS: 0 From DS: 1) (0x02)
    .... 0.. = More Fragments: This is the last fragment
    .... 0... = Retry: Frame is not being retransmitted
    ...0 .... = PWR MGT: STA will stay up
    ..0. .... = More Data: No data buffered
    .1.. .... = Protected flag: Data is protected
    0... .... = Order flag: Not strictly ordered
Duration: 52
Destination address: AskeyCom_25:7b:7a (4c:ed:de:25:7b:7a)
BSS Id: AskeyCom_6a:25:a0 (00:21:63:6a:25:a0)
Source address: WestellT_cc:bb:a7 (00:18:3a:cc:bb:a7)
```

Roaming

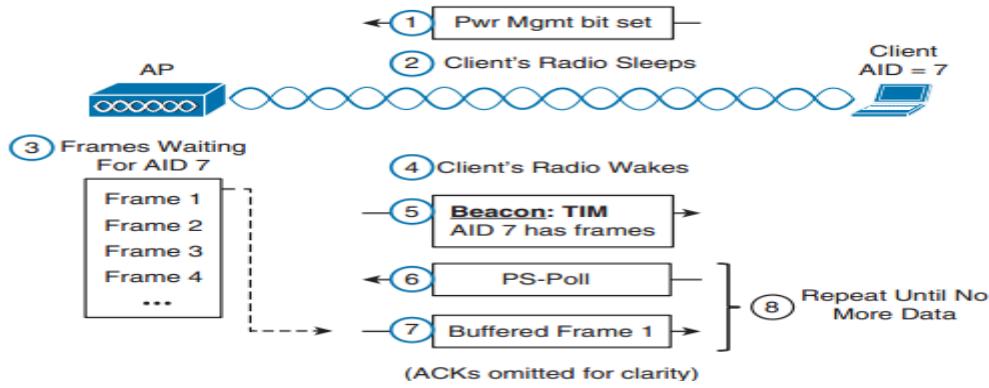
Kablosuz istemcinin bir BSS'ten diğer bir BSS'e kesinti olmaksızın geçiş yapmasıdır. Algılanan sinyal gücü (RSSI) değişimine ve SNR değerine bağlı olarak bu durum gerçekleşir.



5.Adımda handoff süreci içinde AP1 ve AP2 kablolulu ortam üzerinden haberleşirler. Bu süre içinde istemci için gelen paketler AP-1 bufferda tutulur ve bufferdaki bu bilgiler AP-2'ye transfer edilir. Böylece kullanıcı veri kaybı yaşamaz.



Legacy Power Save Mode: Kablosuz ortamlarda mobil cihazların batarya tüketimini azaltmak için bir takım mekanizmalar çalışır. Yani radyo vericileri her zaman açık değildir. Gerek olmadığı durumlarda kısa süreliğine uyku moduna veya düşük güç tüketim moduna geçerler (**Power Save Mode**) PSM moduna geçen istemciler AP'ye bilgi vermek amacıyla başlık bilgisi içindeki **Pwr Mgmt** bitini set ederler. Bundan sonra AP bu istemciler için gelen çerçeveleri bufferda tutar. Ancak istemciler periyodik olarak uyku modundan çıkış kendi için gelen çerçeveleri almaları gereklidir.

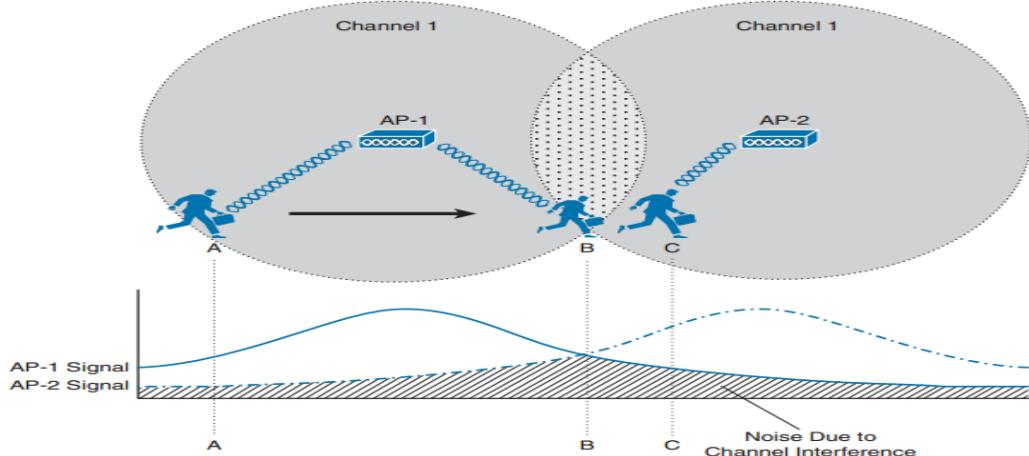


Broadcast ve multicast frame'ler bufferda tutulmayabilir. İstemciler uykuda bunları kaçırır. Ancak AP'lerin **DTIM** desteği varsa belirli periyotlarla bu çerçeveleri de gönderebilir.

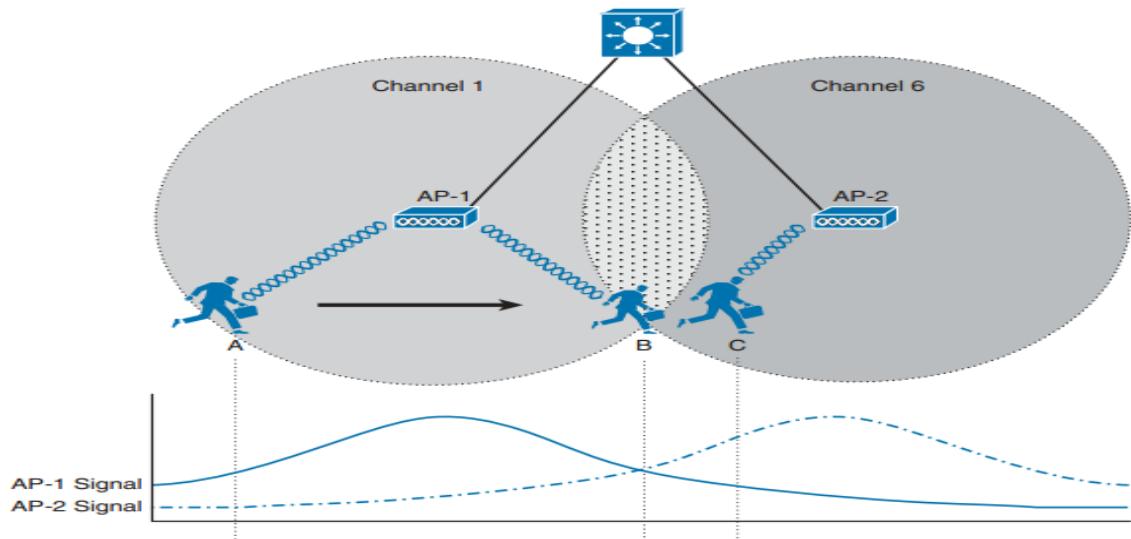
AP1200'ler için bu ayar Network Interface Menüsü altında görülebilir.

Beacon Period:	<input type="text" value="100"/> (20-4000 Kusec)	Data Beacon Rate (DTIM):	<input type="text" value="2"/> (1-100)
Max. Data Retries:	<input type="text" value="64"/> (1-128)	RTS Max. Retries:	<input type="text" value="64"/> (1-128)
Fragmentation Threshold:	<input type="text" value="2346"/> (256-2346)	RTS Threshold:	<input type="text" value="2347"/> (0-2347)

İstemcilerin mobilitesini sağlamak amacıyla ESS yapısı oluşturulmalıdır. Ancak BSS'ler komşu BSS'ler ile çakışma yaşamayacak şekilde yapılandırılmalıdır. Aşağıdaki örnekte aynı kanalları kullanan iki BSS arasında roaming yapan bir kullanıcı B noktasında iken sinyal karmaşası yaşayacaktır.

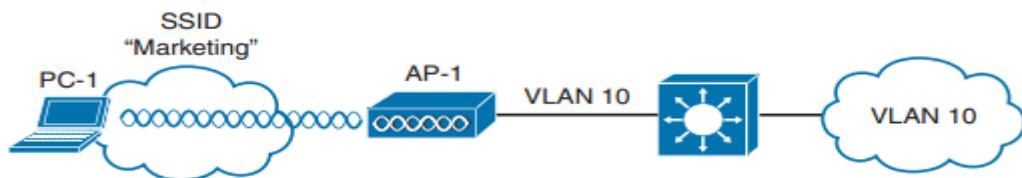


Olması gereken yapı overlap olmayan kanalların kullanımıdır.



Autonomous AP

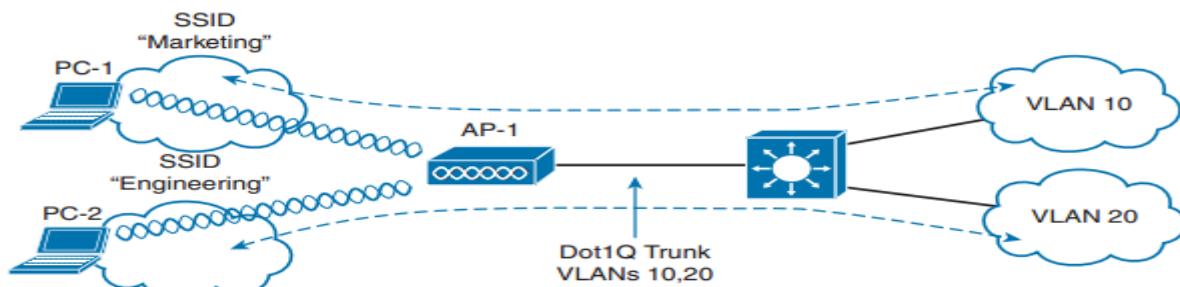
Single SSID: Genellikle küçük işletmelerde, evlerde kullanılan yapıdaki AP rolüdür. Kablosuz istemcilerin kablolu ağa bağlanmalarını sağlar. Örneğin bir şirketteki satış departmanı (VLAN 10) için hizmet vermek üzere bir AP ayarlaması aşağıdaki gibidir.



AP, switch'in VLAN10'a dahil olan portuna bağlanır. AP yapılandırmasında SSID bu VLAN ile ilişkilendirilir. Böylece kablosuz satış istemcileri VLAN10 kablolu ağına erişebileceklerdir.

Not: VLAN desteği olmayan cihazlarda (örneğin doğrudan ADSL Modem Routera bağlanan bir AP) **native** vlan kullanılmalıdır.

Multiple SSID: Daha çok kurumsal yapılarda ya da kampüs ortamlarında kullanılan, mobil kullanıcıları ilgili ağlara dahil eden yapıdır. Örneğin bir kurumda Satış ve Mühendislik bölümleri için kablosuz istemcilere destek vermek için iki SSID tanımlanır ve bunlar ilgili VLAN'ler ile ilişkilendirilir.



Local mode — Varsayılan mod. BSS ler için AP görevi görür. Transfer yoksa gürültü ve interference ölçer, sahte AP tespit eder IDS için çalışır.

Monitor mode — Transfer yok. Bir sensör olarak çalışır. IDS olaylarını izler.

FlexMode — CAPWAP down olduğunda SSID ile VLAN arasında local olarak trafiği yönlendirir

Sniffer mode —Belli trafikleri dinler ve trafiği analyzere gönderir.

Rogue detector mode —An LAP dedicates its radios to listening for other neighbor APs and clients to determine whether they are rogue devices. Device MAC addresses heard on the LAP's wired interface are compared with those heard over the air. Rogue devices are those that appear on both networks.

Bridge —An LAP becomes a dedicated bridge (point to point or point to multipoint) between two networks. Two LAPs in bridge mode can be used to link two locations separated by a distance. Multiple LAPs in bridge mode can form an indoor or out-door mesh network.

OEAP —The LAP operates as an OfficeExtend AP, intended for teleworkers in remote home offices. The LAP connects to the local broadband service and builds a CAPWAP tunnel to the central WLC. User data can be encrypted over the CAPWAP data tunnel using DTLS.

SE-Connect —The LAP dedicates its radios to spectrum analysis on all wireless channels. You can remotely connect a PC running software such as the Cisco Spectrum Expert to the LAP to collect and analyze the spectrum analysis data to discover sources of interference.

WIRELESS SECURITY

Kablolu veya Kablosuz güvenlik üç temel kavrama dayanır: Authentication, Privacy ve Integrity

AUTHENTICATION

Kablosuz ağa bağlanacak istemcinin kimliğinin doğrulanması sürecidir. Yetkili kullanıcıların, misafirlerin ya da bilinmeyen istemcilerin ağa erişim politikaları farklı olmalıdır. Bu nedenle bir istemci kablosuz ağa dahil olmadan önce kimlik doğrulamasından geçmesi gereklidir.

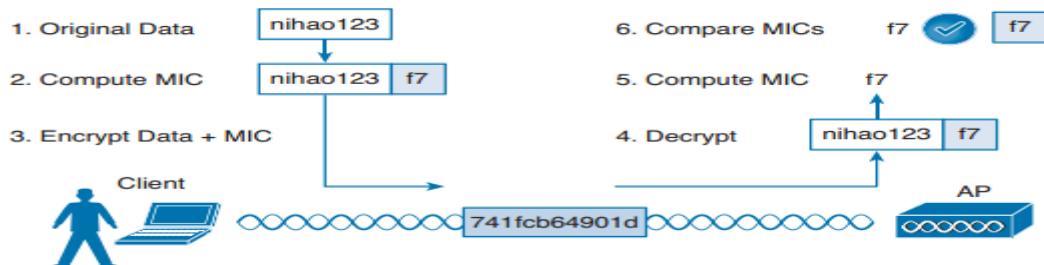
MESSAGE PRIVACY

Mesajın ilgilisi dışındakiler tarafından okunamamasını yani gizliliğini ifade eder. Bunu sağlamak amacıyla kriptolama kullanılır. Bu amaçla AP associate olmuş her istemci için bir kriptolama anahtarı kullanır.

Yine 802.11 Frame yapısındaki WEP alanı kriptolamadın olup olmayacağı belirtir. Bit değeri 1 ise KEY ile kriptolama yapılır, bit 0 ise kriptolama yapılmaz.

MESSAGE INTEGRITY

Mesajın değiştirilip değiştirilmediğini belirtir. Bu amaçla **message integrity check** (MIC) kullanılır.



Wireless LAN'larda genel itibarı ile 3 tip kimlik doğrulama sistemi vardır. **Open Authentication** (kimlik doğrulanmanın olmaması), **Pre-Shared Key authentication** (taraflar arasında önceden paylaşılan bir anahtar) ve **Server-Based Authentication** (kullanıcılar veri tabanında kayıtlı ve kimlik doğrulama sunucusu üzerinden gerçekleştir).

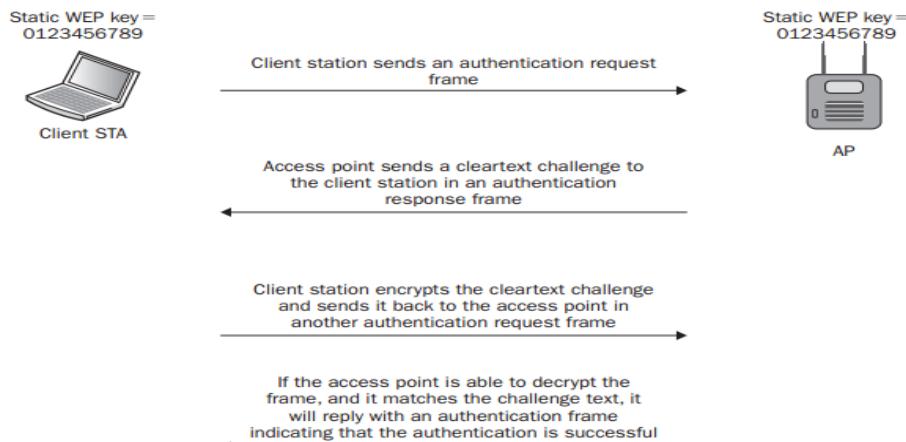
Frame Control	Duration /ID	Address1	Address2	Address3	Sequence Control	Address4	Data	FCS
2 Bytes	2 Bytes	6 Bytes	6 Bytes	6 Bytes	2 Bytes	6 Bytes	0-2312 Bytes	4 Bytes
Protocol Version	Type	Subtype	To DS	From DS	More Frag	Retry	Pwr Mgmt	More Data
Bits: 2	2	4	1	1	1	1	1	1

Orijinal 802.11 frame yapısında authentication için bir-bit ayrılmıştır. Bit 0 ise, Open Authentication; bit 1 ise WEP Authentication.

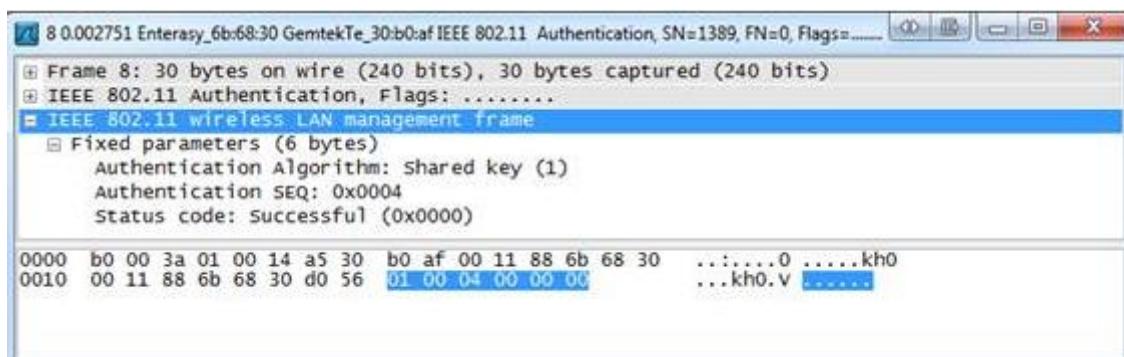
Open Authentication: Bu yapıda herhangi bir kimlik denetimi yoktur. Ancak yine de authentication request ve response çerçevesi alış verisi vardır.

WEP:

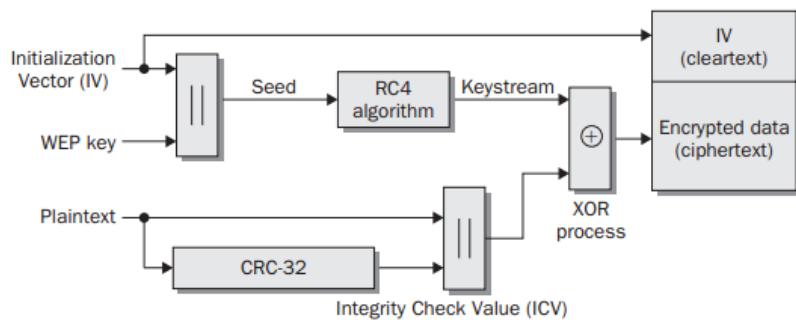
Wireless Equivalent Privacy (WEP) RC4 algoritmasını kullanarak her çerçeveyi kriptolar. Yine aynı algoritma ile kripto çözme işlemi gerçekleşir. Bu algoritmada, key üretmek için WEP Key olarak adlandırılan bir anahtar kullanılır. Gönderici ve alıcı aynı anahtara sahipse kriptolama ve kripto çözme işlemi sorunsuz gerçekleşir. Bu key authentication için de kullanılır.



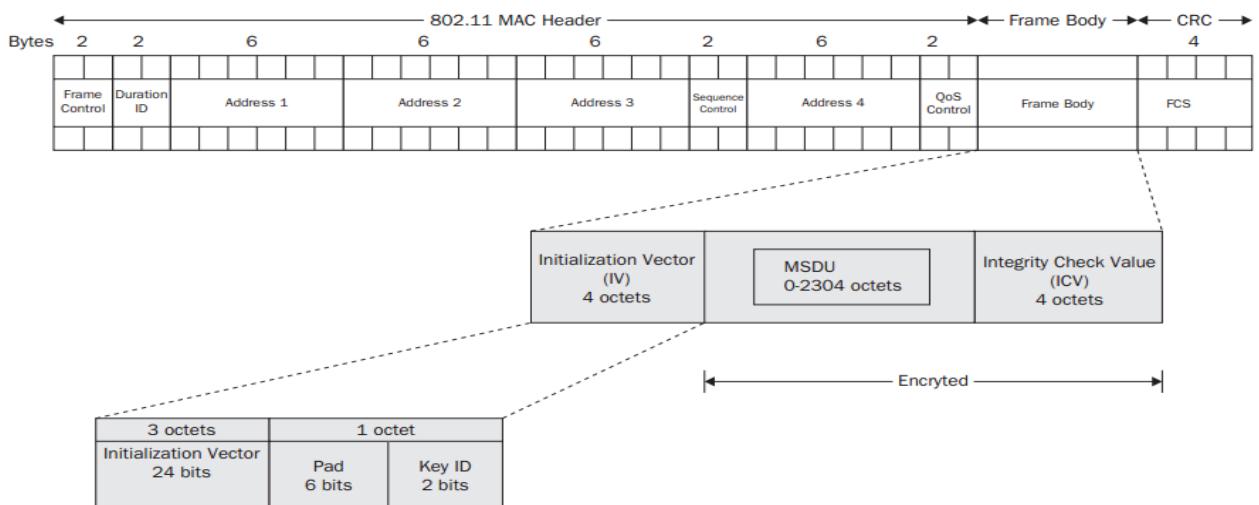
Burada kullanılan WEP-KEY 40-bit ya da 104-bit uzunluğundadır (10 hex / 5 ASCII veya 26 hex / 13 ASCII). Ayrıca algoritma tarafından 24-bitlik bir Initial vector (IV) eklenip kriptolamada kullanılan anahtar boyutu 64 veya 128-bit uzunluğunda olur.



Static WEP encryption key and initialization vector		Static WEP Keys	
64-bit WEP	24-bit IV	40-bit static key	WEP Key Size 40 128
128-bit WEP	24-bit IV	104-bit static key	Already Set? <input type="checkbox"/> WEP Key 1: <input checked="" type="radio"/> 01234567890123456789abcdef <input type="checkbox"/> WEP Key 2: <input type="checkbox"/> WEP Key 3: <input type="checkbox"/> WEP Key 4: Transmit Key Key Entry Method: <input checked="" type="radio"/> Hexadecimal (0-9, A-F) <input type="radio"/> ASCII Text OK Cancel Help



Ancak 2001 yılında yapılan araştırmalarda WEP authentication ve kriptolamanın zayıflıkları ortaya çıktı ve çok kısa zaman aralığında WEP KEY tespit edildi. Buradaki zayıflık 24-bit uzunluğundaki IV key'in yeteri kadar uzun olmamasından ($16,777,216$) ve belli paket sayısından sonra aynı KEYlerin tekrar edilmesi sonucu anahtarın tespit edilmesine dayanır. Araştırmalarda, 5000 paket içinde %50 ihtimalle aynı iki IV Key kullanımı ortaya çıkmıştır.



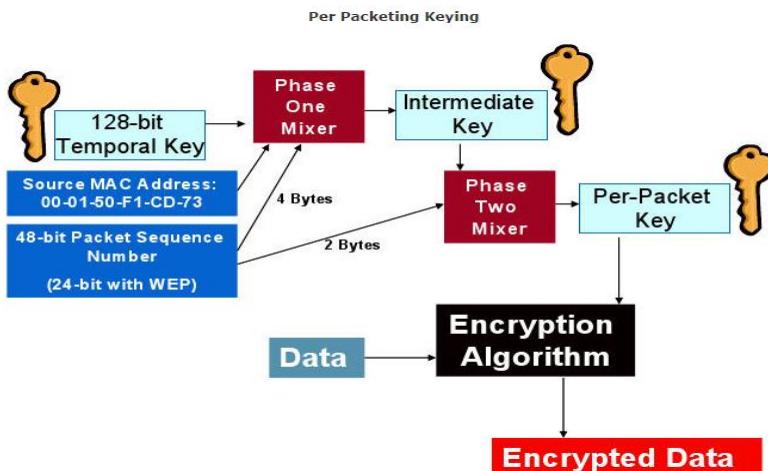
TKIP (Temporal Key Integrity Protocol): WEP zayıflıklarını kapatmak amacıyla Wi-Fi Alliance tarafından geliştirilmiştir. Yine WEP gibi RC4 algoritması kullanır. Mesajlara aşağıdaki özellikler eklenmiştir:

Time Stamp: Tekrar saldırılalarını önlemek için zaman damgası eklenir.

Sender MAC Address: Göndericinin kaynak MAC adresi eklenir.

TKIP Sequence Counter: Gönderici MAC adresinden gelen çerçeveyin sıra numarasını belirtir.

Per Packet Keying: Her frame için 128-bit WEP anahtarı üretilir.



Long IV: 48-bitlik IV key kullanılır.

Broadcast Key Rotation: İstemciye dinamik olarak kriptolu bir anahtar atanır.

CCMP (Counter/CBC-MAC Protocol): WEP ve TKIP'den daha güvenlidir ve kriptolama algoritması olarak RC4 değil, AES kullanır.

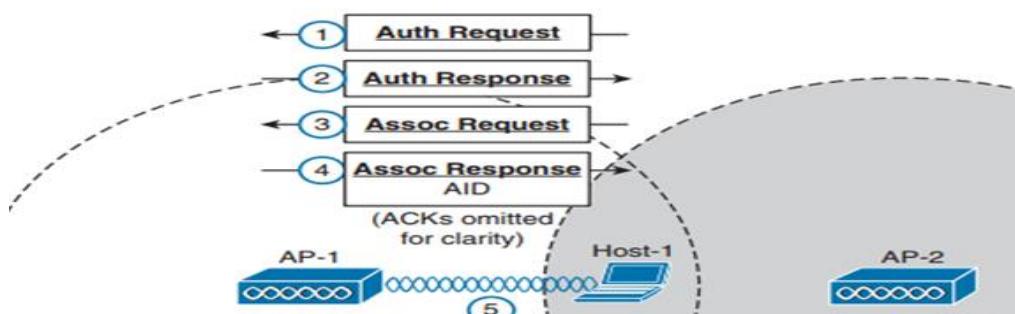
128-bit temporal key, 48-bit Packet Number, 104-bit Random Number kullanır. Message Integrity kullanır. Bu değerler Data ile AES fonksiyonu aracılığıyla kriptolanır.

WEP, WPA ve WPA2 integrity, privacy ve authentication WEP, TKIP veya CCMP yöntemlerini kullanır.

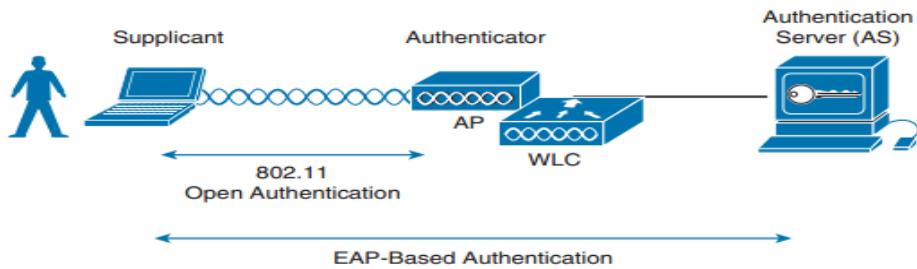
WEP authentication için WEP yöntemi; WPA için TKIP yöntemi, WPA2 için CCMP yöntemi kullanılır.

SUNUCU TABANLI KİMLİK DOĞRULAMA

802.1x/EAP Authentication: Extensible Authentication Protocol (EAP), kullanıcı kimlik doğrulama için kullanılan bir protokoldür. EAP protokolü kablosuz veya kablolu port bazlı kimlik doğrulamayı da (802.1x) sağlar. Kullanıcı EAP kimlik doğrulamasından geçmeden ağa bağlanamaz. Yani kablosuz bir istemci associate olsa bile EAP ile kimlik doğrulamadan geçmeden ağ kaynaklarını kullanamaz. Burda WEP Authentication yapısından farklı bir durum söz konusudur.



Normalde Association işleminden önce authentication gerçekleşir. EAP yapısında ise kimlik doğrulama sunucuya iletir. Bu nedenle kullanıcının Associate olmasına izin vermek (Frame yapısındaki WEP bitini 0 yapmak) dolayısı ile Open Authentication kullanmak gereklidir. EAP süreci bundan sonra başlar.



Suplicant, kimlik doğrulamasından geçip ağ kaynaklarına erişmek isteyen istemcidir. **Authentication Server** kullanıcıların kimlik doğrulamalarını kendindeki veri tabanına göre sağlayan sunucudur. **Authenticator** ise trafiğin geçip geçmeyeceğine karar veren AP/WLC cihazdır.

Kablolu ya da kablosuz istemcinin 802.1x yapısına bağlanabilecek şekilde yapılandırılması gereklidir.

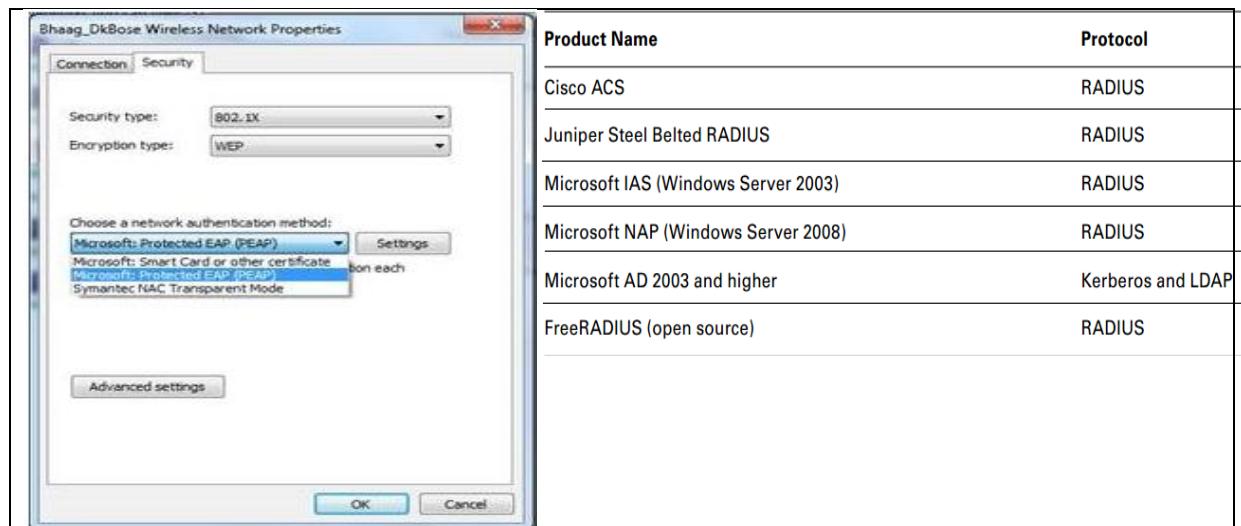
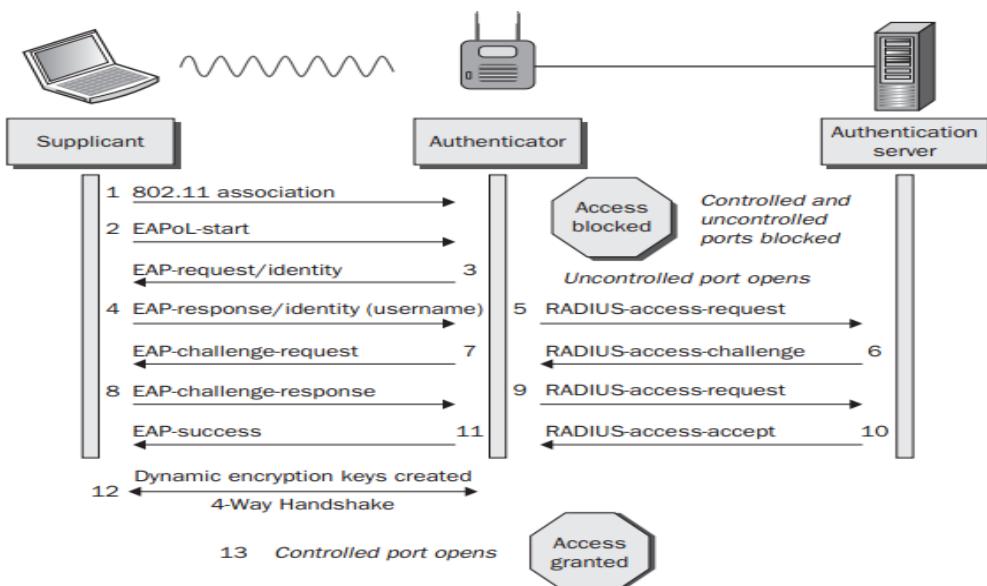


FIGURE 4.24 Generic EAP exchange



LEAP(Lightweight EAP)

Cisco tarafından geliştirilmiş bir protokol olan LEAP'te AP'ler kullanıcıların kimlik denetimlerini bir RADIUS(Remote Authentication In User Server/Service) üzerinden gerçekleştirirler. Kimlik denetimi için kullanıcı adı ve parola kullanılır.

LEAP ayrıca WEP'i kullanarak veri güvenliğini de sağlar. Her kablosuz ağ kullanıcısı için dinamik olarak RADIUS sunucu tarafından bir WPA anahtarı üretilir. Böylece kullanıcı bazlı veri güvenliği sağlanmış olur. Kriptolama olarak MS-CHAPv2 kullanır.

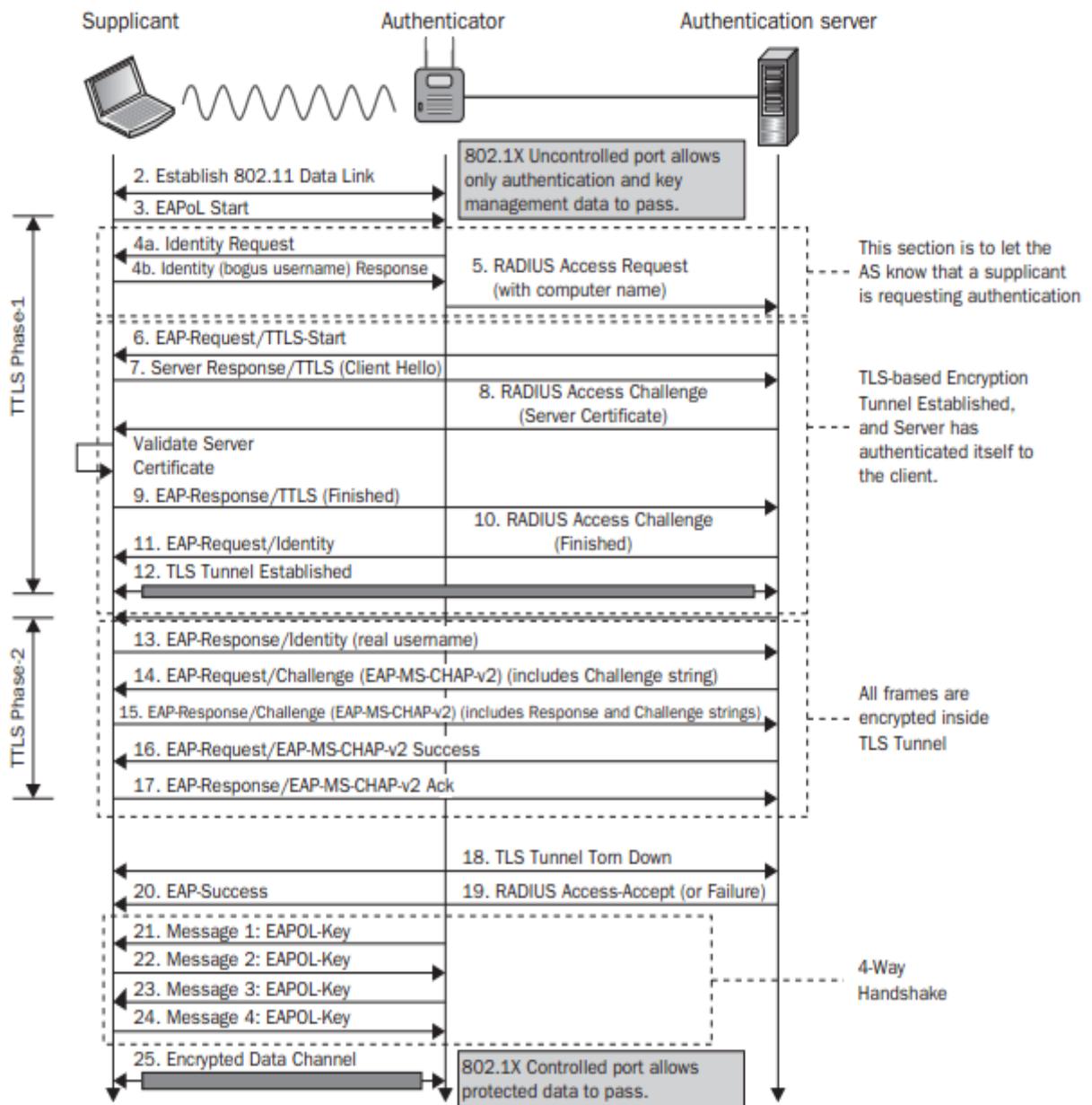
Zayıflıkları: Kullanıcı adı clear-text olarak gönderilir ve zayıf hashing algorithm (MSCHAP) kullanır. Kimlik doğrulamada sadece username&password kullanılır. Ayrıca sunucunun kimliğinin doğrulanması aşaması yoktur.

EAP-TLS

Daha güçlü EAP potokollerinde tünelleme mekanizması kullanılır. Tünellemede iki tür kimlik bilgisi gönderimi vardır. İlk, tünel dışından (**outer identity**) clear-text olarak gönderilen ve gerçek kimliği yansıtmayan (çoğu zaman anonymous kullanılır) kimliktir. Oysa gerçek kimlik tünelden geçer (**inner identity**) ve gizlidir. Burada oluşturulan tünelin amacı kimlik bilgilerinin güvenli transferi içindir ve birkaç mili saniye sürede gerçekleşir ve kapanır.

EAP-TLS metodu, güvenli kimlik denetimi için TLS(Transport Layer Security) protokolünü kullanır. TLS'in temeli güvenli web oturumları sağlamak için kullanılan SSL(Security Socket Layer) protokolüne dayanır. EAP-TLS kimlik denetimi için dijital sertifikaları kullanır. Bu yüzden AP'nin ve kullanıcının sertifika otoritesi tarafından üretilmiş bir sertifikaya ihtiyacı vardır.

EAP-TLS; sunucu, kullanıcıyı her yeniden kimlik denetiminden geçmeye zorladığında otomatik olarak WEP anahtarını üretecek veri güvenliğini de sağlar. Kimlik denetiminden geçen her kullanıcı için TLS oturum anahtarı, WEP anahtarını üretmek için kullanılır ve veri güvenliği sağlanır.

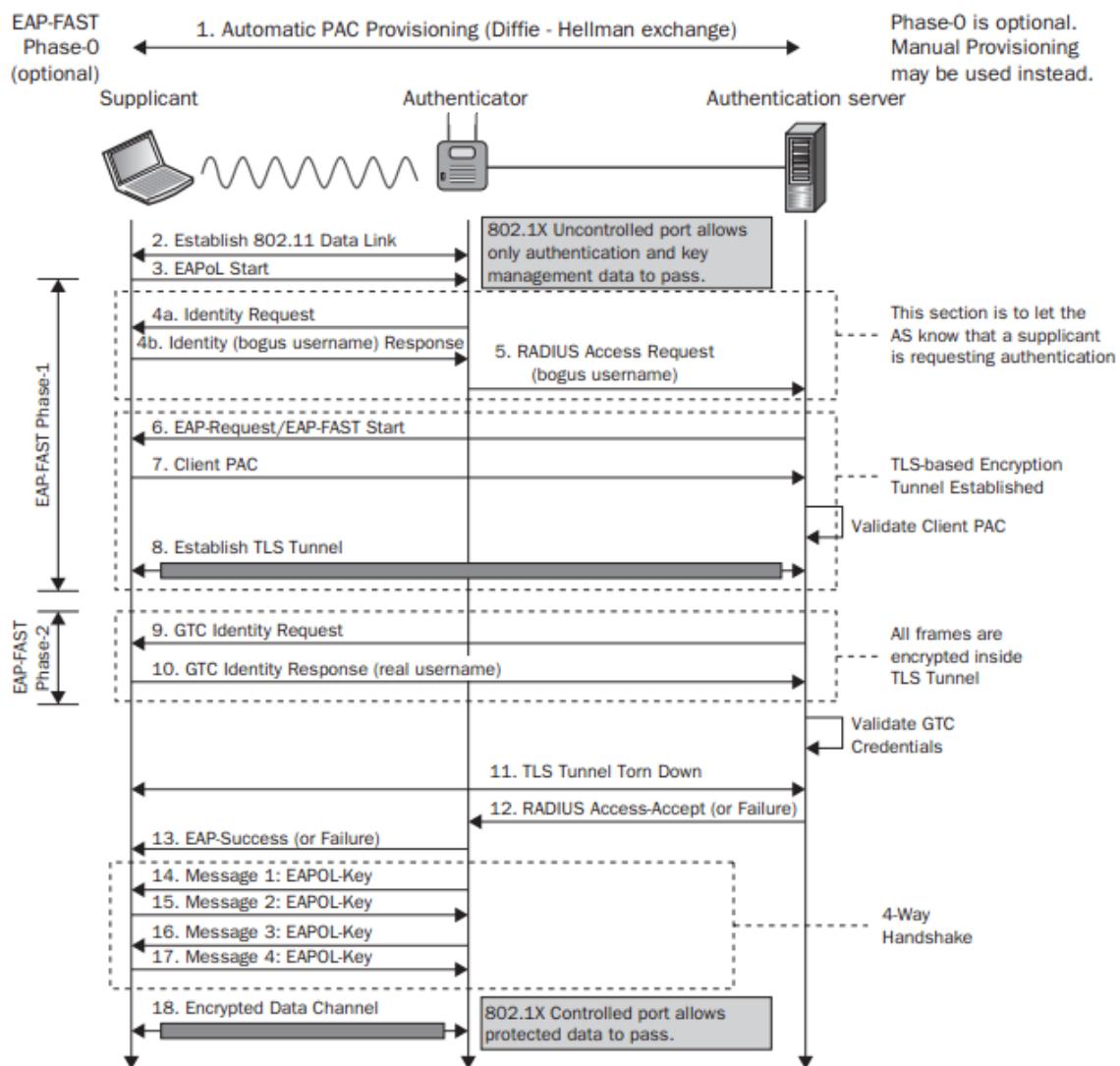


PEAP(Protected EAP)

PEAP'te de EAP-TLS'te olduğu gibi kimlik denetimi için TLS oturumu temel alınır. Fakat PEAP'te dijital seritifikaya sadece kimlik denetimi sunucusunda gerek duyulur. Kullanıcılar kimlik denetiminden geçmek için MSCHAPv2'yi(Microsoft Challenge Handshake Authentication Protocol version 2) kullanırlar.

EAP-FAST(EAP Flexible Authentication via Secure Tunneling)

EAP-FAST Cisco tarafından geliştirilmiş bir protokoldür. Yönetimsel karışıklıkları azalttığı için esnek bir protokoldür. Kullanıcıların dijital sertifikalar kullanmasına ve güçlü parola kurallarına gerek yoktur.



EAP-FAST ile kimlik denetim sunucusu ve kullanıcı arasında güvenli bir tünel oluşturulur. Tüneli oluşturmak için **PAC**(Protected Access Credential) adında bir referansa ihtiyaç duyular. PAC bir PAC sunucusu vasıtasıyla veya EAP-FAST fazlarında dinamik olarak oluşturulabilir. Tünel bir kez kurulduğunda, kullanıcılar kullanıcı adı ve şifreleriyle kimlik denetiminden geçerler.

Ayrıca PEAP'te WEP sayesinde veri güvenliği de garanti altına alınabilir.

	EAP-MD5	EAP-LEAP	EAP-TLS	EAP-TTLS	PEAPv0 (EAP-MSCHAPv2)	PEAPv0 (EAP-TLS)	PEAPv1 (EAP-GTC)	EAP-FAST
Security Solution	RFC-2284	Cisco proprietary	RFC-2716	IETF draft	IETF draft	IETF draft	IETF draft	IETF draft
Digital Certificates—Client	No	No	Yes	Optional	No	Yes	Optional	No
Digital Certificates—Server	No	No	Yes	Yes	Yes	Yes	Yes	No
Client Password Authentication	Yes	Yes	N/A	Yes	Yes	No	Yes	Yes
PACs—Client	No	No	No	No	No	No	No	Yes
PACs—Server	No	No	No	No	No	No	No	Yes
Credential Security	Weak	Weak (depends on password strength)	Strong	Strong	Strong	Strong	Strong	Strong (if Phase 0 is secure)
Encryption Key Management	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mutual Authentication	No	Debatable	Yes	Yes	Yes	Yes	Yes	Yes
Tunneled Authentication	No	No	Optional	Yes	Yes	Yes	Yes	Yes
Wi-Fi Alliance supported	No	No	Yes	Yes	Yes	No	Yes	Yes
Man-in-the-Middle Protection	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Dictionary Attack Resistance	No	No	Yes	Yes	Yes	N/A	Yes	Yes
Token support	No	No	Yes	Yes	No	Yes	Yes	Yes

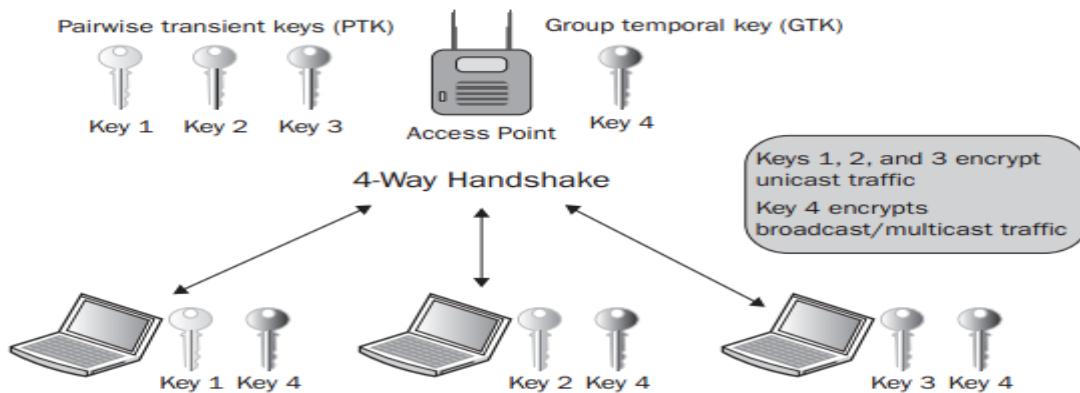
WPA /WPA2 (Wi-Fi Protected Access) : Wi-Fi alliance tarafından geliştirilmiştir. WPA TKIP kullanırken, WPA2 CCMP veya TKIP kullanır.

WPA ya da WPA2 kullanımında pre-shared key (**Personal Mode**) kullanılabildiği gibi sunucu tabanlı (**Enterprise Mode**) EAP protokollerini de kullanılabılır.

802.11 Standard	Wi-Fi Alliance Certification	Authentication Method	Encryption Method	Cipher	Key Generation
802.11 legacy		Open System or Shared Key	WEP	RC4	Static
802.11- 2007	WPA-Personal	WPA Passphrase (also known as WPA PSK and WPA Pre-Shared Key)	TKIP	RC4	Dynamic
	WPA-Enterprise	802.1X/EAP	TKIP	RC4	Dynamic
	WPA2-Personal	WPA2 Passphrase (also known as WPA2 PSK and WPA2 Pre-Shared Key)	CCMP (mandatory)	AES (mandatory)	Dynamic
802.11-2007	WPA2-Enterprise	802.1X/EAP	TKIP (optional) CCMP (mandatory) TKIP (optional)	RC4 (optional) AES (mandatory) RC4 (optional)	Dynamic

4-WAY HANDSHAKE

BSS yapısında, AP ile Client arasında authentication gerçekleşikten sonra unicast, multicast ve broadcast tüm data paketleri Master Key'den (**PMK**) üretilen anahtarlarla kriptolanarak gönderilir. AP ile client arasında paylaşılan dinamik anahtarlar ile kriptolama/dekriptolama sağlanır. Bu anahtarlarla **pairwise transient key** (PTK) denir. Unicast haberleşmeler istemciye özgü olduğu için her istemci için bir key oluşur ancak multicast ve broadcast yayınları birden çok kullanıcıyı ilgilendirebileceğinden bu anahtarı birden fazla kişi kullanması gerekecektir. Bu tür anahtarlarla da **Group Temporal Key** (GTK) denir.



PMK, istemcinin her authentication işleminde yenilenir ve güvenli kanalardan Authenticator'a gönderilir.

4-WAY HANDSHAKE'de AMAÇ;

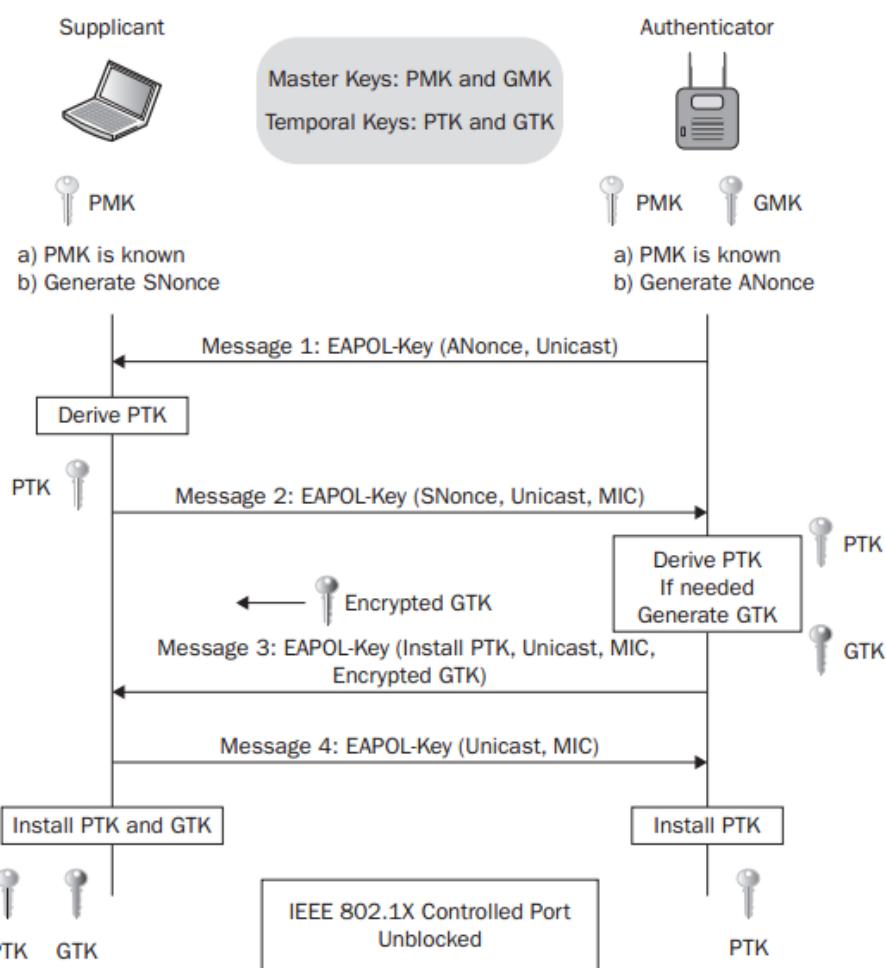
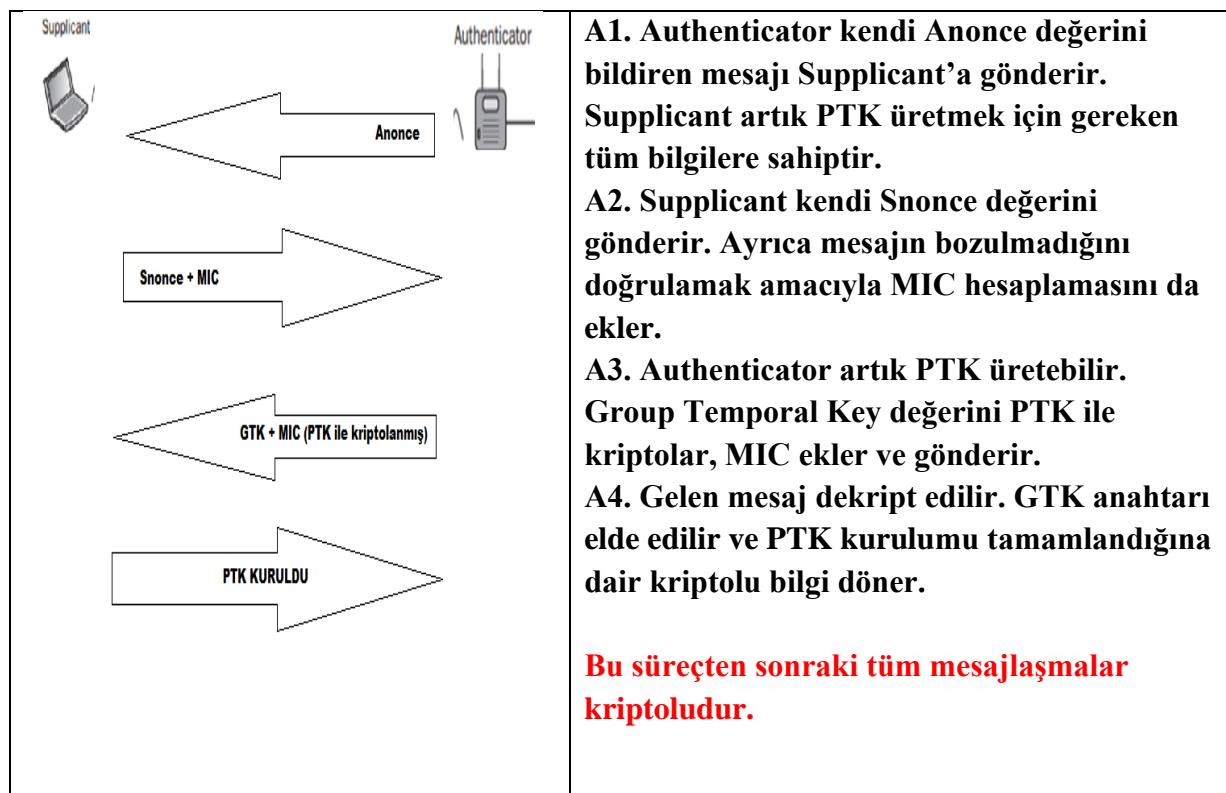
- PMK'nın geçerliliğini doğrulamak
- Yeni PTK üretmek
- PTK'yi Authenticator ve Supplicant'a yüklemek
- Uygun kriptolamayı sağlamaktır.

Data帧leri kriptolamak için kullanılan PTK'lar aşağıdaki bilgilerden türetilir.

- PMK
- Anonce (Authenticator tarafından üretilen random değer)
- Snounce (Supplicant tarafından üretilen random değer)
- Authenticator MAC adres
- Supplicant MAC adres

PTK= PRF (PMK+Anonce+Snounce+AA, SPA)...Haliyle her istemci için farklıdır.

ADIMLAR:



Sistem İzlemeye Giriş

Loglar, network güvenliğinin sağlanması açısından son derece önem arzeden unsurlardır. Interface durumlarında değişiklikler, Access-listlerden geçen trafikler, hatalı login girişimleri gibi bir çok konuda uyarılar içerirler.

Network güvenliği sağlanırken loglar; **Console**, **Terminal Line**, **Buffered Logging**, **SNMP Traps** ya da **Syslog** yöntemlerinden bir veya bir kaçına göndermek gerekebilir. Cisco cihazlarda loglar 8 kategoriye ayrırlırlar. En düşük numaralı kategori en kritik öneme sahiptir.

LEVEL	KEYWORD	AÇIKLAMA	TANIMLAMA	Örnek
0	emergencies	Sistem kullanılamıyor.	LOG_EMERG	IOS yüklenemiyor
1	alerts	Acil önlem gereklili.	LOG_ALERT	Sıcaklık çok yüksek
2	critical	Kritik durum oluştu	LOG_CRIT	Hafiza ayrılamıyor.
3	errors	Hata oluştu.	LOG_ERR	Geçersiz hafiza boyutu
4	warnings	Uyarı durumu oluştu	LOG_WARNING	Crypto işlemi başarısız
5	notification	Normal ancak özel bir durum	LOG_NOTICE	Interface up/ down oldu.
6	informational	Sadece bilgilendirme mesajı	LOG_INFO	Paket ACL tarafından deny edildi.
7	debugging	Debug mesajlar	LOG_DEBUG	Paket türü geçersiz

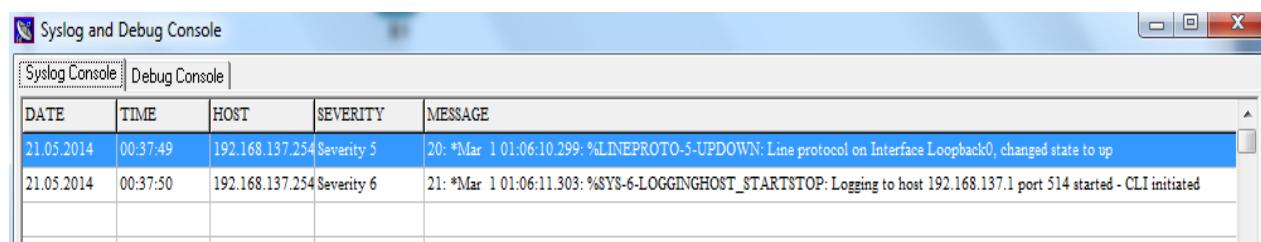
Loglar, Zaman Damgası, log mesaj adı ve seviyesi, mesaj yazısı olmak üzere üç kısımdan oluşuyor.

Örnek:

Mar 1 05:53:47.414: %LINK-5-CHANGED: Interface FastEthernet0/0, changed state to administratively down

Zaman Damgası	Log Adı ve Seviyesi	Mesaj Yazısı
Mar 1 05:53:47.414	LINK-5-CHANGED	Interface FastEthernet0/0, changed state to administratively down

Cihazlar tarafından üretilen logların bir sunucuya gönderilmesi ve burada incelenmesi network güvenliği açısından önemlidir. Network anomalisinin belirlenmesine ve buna dayalı olarak zero-day ataklara karşı tespit sürecinde bu loglar büyük bir önem teşkil eder. Aşağıdaki resimde bir syslog sunucusuna gelen loglar görüntülenmektedir.



Logların tutulduğu sunucuya **Syslog Server**; bu sunucuya log gönderen router ya da switch gibi ağ cihazları da **syslog client** olur.

LOGGING SERVER YAPILANDIRMASI

Syslog Server olarak yapılandırılacak bilgisayara Sylog yazılımı kurulur. Syslog clientlerde ise aşağıdaki yapılandırma uygulanır.

```
RTR1(config)#interface loop 0
RTR1(config-if)#ip address 1.1.1.1 255.255.255.255
RTR1(config-if)#exit
RTR1(config)#logging host 192.168.137.1
RTR1(config)#logging trap 5
RTR1(config)#loggin source-interface loopback 0
RTR1(config)#logging on
```

Konsol ekranına, log sunucuya ya da RAM'de farklı seviyelerde loglar göndermek için;

```
RTR1(config)#logging console 7 //konsol ekranına tüm loglar düşer
RTR1(config)#logging trap 4 // sunucuya 1-4 arası loglar gider
RTR1(config)#logging buffered 5 //Buffer'a 1-5 arası loglar kaydedilir.
```

RAM'da default olarak 4000 log tutulur. Bunu değiştirmek için;

```
RTR1(config)#logging buffered 8000
RTR1#show logging
```

2.3.4 NETWORK GÜVENLİĞİ İÇİN SNMP KULLANIMI

IP ağlarında router, switch, server, firewall gibi cihazların izlenmesinde ve yapılandırmasında kullanılan bir Uygulama katmanı TCP/IP yiğini protokolüdür. Özellikle ağ güvenliğinin izlenmesinde kullanılan SNMP'nin üç versiyonu vardır. SNMP; SNMP manager (network management system **NMS**), **agent** ve Management Information Base (**MIB**) olmak üzere üç bileşenden oluşur. Yönetimi yapılacak olan router, switch gibi cihazlara agent yazılımı çalıştırılır. Bu agentten gelen bilgiler, NMS tarafından işlenir. SNMP manager, agentten gönderilen MIB kaydına göre bilgi okuyabilir ya da cihaza bilgi yazabilir. Örneğin, bir cihazdaki aktif portların durumlarını sorgulayabilir ya da cihazdaki bir portu açıp kapatabilir.

SNMP Manager **GET** mesajı ile cihazdan veri okurken, **SET** mesajı ile cihaza bilgi yazar (değişiklik yapar). Agent kurulu node'lardan gelen asenkron bilgilere SNMPv1'de **trap** denirken, daha sonraki versiyonlarda buna **information** denir.

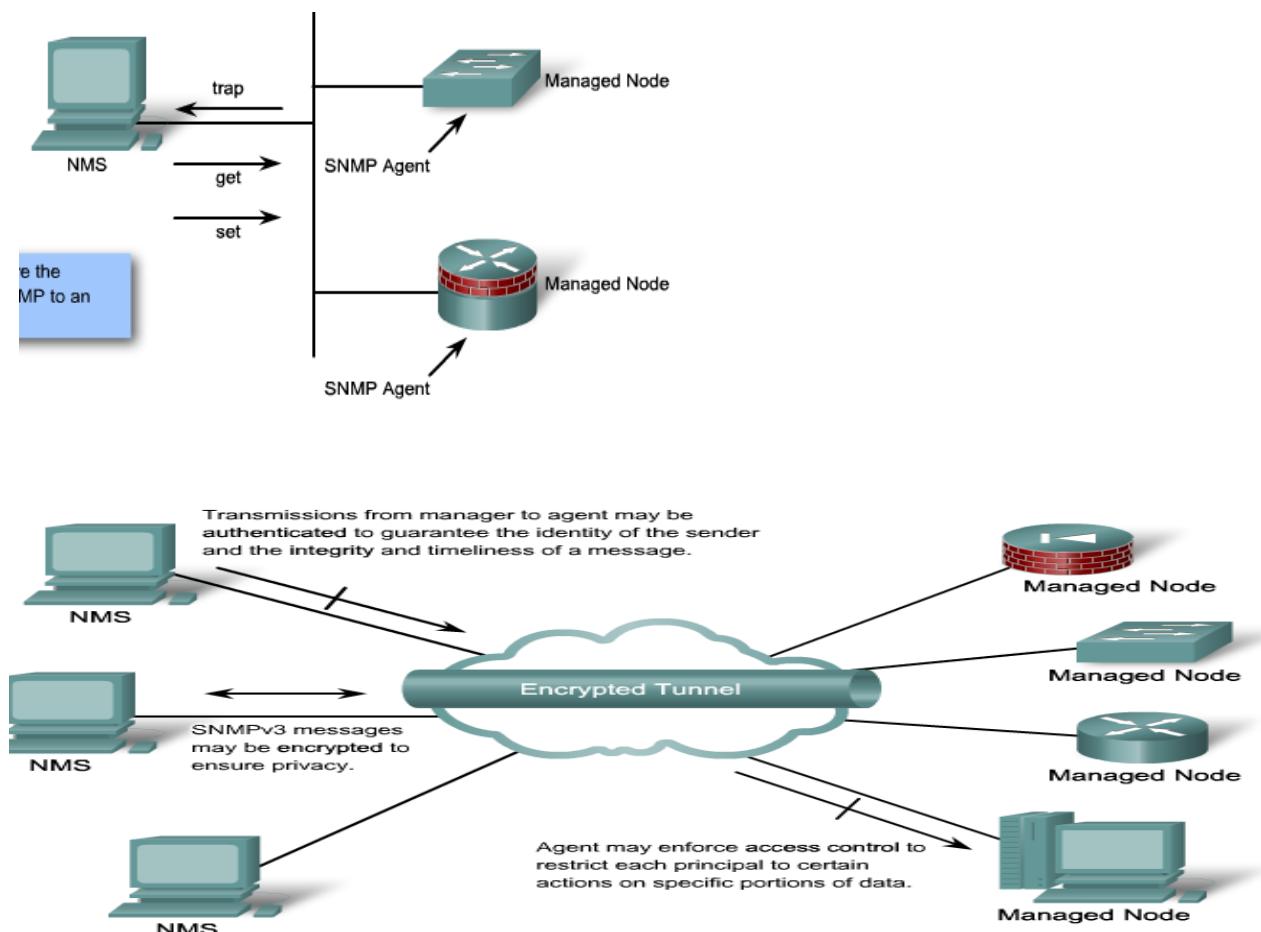
SNMPv1 ve SNMPv2'de Manager ile agent arasındaki kimlik denetimi **community string** denen bir anahtar ile sağlanır. Sadece okumayı sağlayan string (**ro**) ve hem okuma hem yazmayı sağlayan string (**rw**) olmak üzere iki farklı community string bulunur. Örneğin sadece okumayı sağlayan string kullanılarak agent'tan gönderilen MIB kaydına göre bilgi okunabilir. Birçok SNMP uyumlu cihazda community string değeri default olarak "**public**" tir. Bu durum

büyük bir güvenlik açığı oluşturur. Bu sebeple community string değerinin değiştirilmesi son derece önemlidir. Ancak, SNMPv1 ve SNMPv2'de community string değeri ağda kriptolanmadan gönderildiği için, packet sniffer'lar ile tespit edilmesi mümkündür. Her iki SNMP sürümünde de bu durum halen büyük bir güvenlik açığıdır. Bu sebeple SNMPv3 kullanmak gereklidir.

[SNMPv3](#)

Ağ güvenliğinde **authentication (kimlik denetimi)**, **encryption (kriptolama, şifreleme)** ve **integrity (mesaj bütünlüğü)** olmak üzere başlıca üç kavram söz konusudur.

SNMPv3'te bu desteklerin yanında bir de **Access Control** özelliği bulunmaktadır. Yani full yetki vermek yerine, verilerin belli bir kısmı üzerinde belli değişiklikler yapma yetkisi verilebilir.



Sadece SNMPv3 auth ve priv özelliğini destekler.

[SNMP YAPILANDIRMA](#)

Aşağıdaki örnekte okuma için “*sadeceOku*”, hem okuma hem yazma için “*okuYaz*” community string olarak tanımlanmıştır.

```
R1(config)#snmp-server community sadeceOku ro
```

```
R1(config)#snmp-server community okuYaz rw
```

Sadece belirli kullanıcılara (IPlere) bu yetki verilerek bir ACL ile yapılandırılmak daha güvenli bir yöntemdir. Örneğin sadece 192.168.1.0/24 ağından gelen SNMP sorgulara cevap verilmesini sağlayalım.

```
R1(config)#access-list 77 permit 192.168.1.0 0.0.0.255
```

```
R1(config)#snmp-server community sadeceOku ro 77
```

```
R1(config)#snmp-server community okuYaz rw 77
```

NETWORK TIME PROTOCOL (NTP)

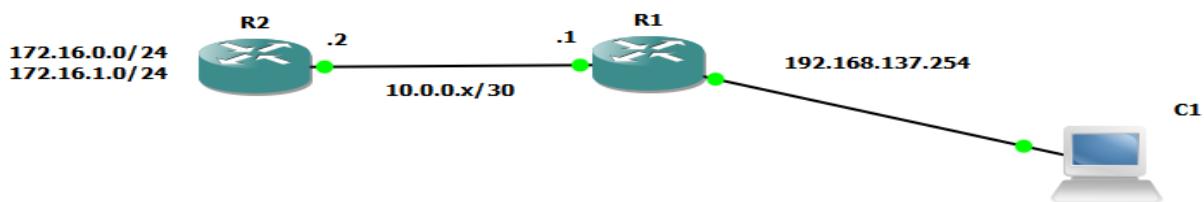
Network güvenliğinde logların analiz edilmesi, saldırıların ya da zero-day atakların önceden algılanmasında işe yarar bir yöntemdir. Ayrıca geçmişte gerçekleşen olaylarla ilgili raporlara ihtiyaç duyulduğunda da loglar incelenebilir. Bu sebeple oluşturulan logların zaman damgasının da doğru olması, doğru tespitlerin yapılmasını sağlayacaktır. Cihazlarda zaman (tarih ve saat) manuel olarak yapılandırılabilir. Manuel olarak zamanı yapılandırmak için **clock set** komutu kullanılır.

```
R1#clock set 00:48:00 21 MAY 2014
```

```
*May 21 00:48:00.000: %SYS-6-CLOCKUPDATE: System clock has been updated from  
01:15:50 UTC Fri Mar 1 2002 to 00:48:00 UTC Wed May 21 2014, configured from  
console by console.
```

Farklı cihazlardan gelen logların birlikte analizinin yapılması gereği durumlar söz konusu olduğunda, tarih ve saatlerin tüm cihazlarda yapılandırılması gerekliliği ortaya çıkacaktır. Ancak senkronizasyon olması açısından, zamanın elle yapılandırılması değil, bir zaman sunucusundan (NTP Server) alınması daha doğru bir yöntem olacaktır. NTP, UDP 123 portunu kullanan bir protokoldür. Cisco cihazlar bu protokol ile internetten public bir NTP sunucusundan zamanı öğrenebildiği gibi, NTP Server olarak da çalışıp, cihazlara zaman bilgisini private olarak da sunabilir.

Çoğu NTP sunucusu kimlik denetimi istemediğinden Internetten herhangi bir public sunucudan zamanı öğrenmek riskli bir yöntemdir ve saldırılara açıktır. Sahte NTP mesajları ile sistemin zamanı yanlış yapılandırılabilir. Bunun sonucunda zamana duyarlı sertifikaların geçersiz olması sağlanabilir. Bu sebeple güvenilir bir sunucu kullanmak ya da private yöntemi kullanmak gerekebilir.



Yukarıdaki topolojide R1 cihazını NTP Server yapalım.

```
R1(config)#ntp server
```

R2'nin zaman bilgisini R1'den almasını sağlayalım.

```
R2(config)#ntp server 10.0.0.1
```

Cihazların NTP durumlarını görmek için **show ntp associations detail** komutu kullanılır.
NTP Server cihazdaki durum aşağıdaki gibidir.

```
R1#show ntp associations detail
127.127.7.1 configured, our_master, sane, valid, stratum 0
ref ID .LOCL., time D727D4B8.95816145 (01:43:20.584 UTC Thu May 22 2014)
our mode active, peer mode passive, our poll intvl 64, peer poll intvl 64
root delay 0.00 msec, root disp 0.00, reach 377, sync dist 0.015
delay 0.00 msec, offset 0.0000 msec, dispersion 0.02
precision 2**18, version 3
org time D727D4B8.95816145 (01:43:20.584 UTC Thu May 22 2014)
rcv time D727D4B8.95816145 (01:43:20.584 UTC Thu May 22 2014)
xmt time D727D4B8.9581454F (01:43:20.584 UTC Thu May 22 2014)
filtdelay = 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
filtoffset = 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
filterror = 0.02 0.99 1.97 2.94 3.92 4.90 5.87 6.85
Reference clock status: Running normally
Timecode:
```

R2 cihazında durum ise

```
R2#show ntp associations detail
10.0.0.1 configured, our_master, sane, valid, stratum 1
ref ID .LOCL., time D727D4B8.95816145 (01:43:20.584 UTC Thu May 22 2014)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 0.00 msec, root disp 0.03, reach 377, sync dist 41.183
delay 60.46 msec, offset -11.8857 msec, dispersion 10.93
precision 2**18, version 3
org time D727D4D2.4EDB423F (01:43:46.308 UTC Thu May 22 2014)
rcv time D727D4D2.571BBF6A (01:43:46.340 UTC Thu May 22 2014)
xmt time D727D4D2.3E7EDF1D (01:43:46.244 UTC Thu May 22 2014)
filtdelay = 96.13 60.46 68.18 64.18 72.14 108.15 72.08 136.11
filtoffset = 15.83 -11.89 -7.04 -20.75 10.27 13.62 -16.07 -12.14
filterror = 0.02 0.99 1.97 2.94 3.92 4.90 5.87 6.85
```

Bu durumda R1 ve R2 cihazları NTP peer olarak adlandırılır. NTP'nin daha güvenli çalışması için NTPv3 kullanılması önerilir. Ayrıca NTP peer'lerin kimlik denetimi ile güvenliğini sağlamak gereklidir.

```
R1(config)#ntp authenticate  
R1(config)#ntp authentication-key 1 md5 ntpPa55  
R1(config)#ntp trusted-key 1  
  
R2(config)#ntp authenticate  
R2(config)#ntp authentication-key 1 md5 ntpPa55  
R2(config)#ntp trusted-key 1
```

VİRÜSLER

Başka bir yazılıma ya da çalıştırılabilir dosyaya kendini ekleyerek zararlı kodların çalışmasını sağlayan yazılımlardır. Belge açma, eposta eki çalışma gibi bir etkileşim olmadan çalışmazlar.

Virüslerin karakteristik özelliği kendini kopyalaması (çoğalma) ve bulaşmasıdır. Zararlı kod kısmı ise her virüste farklılık gösterebilir. Kimlik bilgilerinin/verilerin çalınması, değiştirilmesi ya da silinmesi, donanımın işlevselliliğinin zarar görmesi gibi etkiler gösterebilirler.

Eskiden virüsler yayılmak için genellikle floppy diskler kullanırdı. Günümüzde ise, USB Disk, CD/DVD ya da e-mail ile yayılabilirler. En sık karşılaşılan yöntem de e-mail ile yayılma yöntemidir.

İlk Virüs: Creeper Virus (ARPAnet) Deneysel amaçla ARPAnet'te kendi kendini kopyalama amacıyla yazılmıştır. TENEX OS işletim sistemlerinde etkili oldu. Bulaştığı bilgisayarda "*I'm the creeper, catch me if you can!*" diye bir mesaj görüntüleniyordu. Daha sonraları tespit edilip silinmesi amacıyla Reaper diye bir program yazıldı.

Belli Başlı Virüsler:

1970'ler: Creeper Virus (ARPAnet)

1974: Rabbit Virus

1975: ANIMAL Virus

1982: Elk Cloner Virus

1983: Bilgisayar Virüsü kavramı doğdu.

1986: Brain Boot Sector Virüsü

1989: GhostBall Virus

1991: Michelangelo Virüsü

1998: Chernobil Virüsü (CIH)

1999: Melissa Virüsü

2000: I Love You

...

2011: Morto Virüsü

WORMS (SOLUCANLAR)

Kullanıcı etkileşimine ihtiyaç duymadan, çoğunlukla ağdaki, işletim sistemlerindeki açıklardan yararlanarak kendi kendini çoğaltma yeteneğine sahiptirler. Ağın performansını düşürürler. Veri çalma, kimlik hırsızlığı gibi çeşitli amaçlarla kullanılabilirler.

Örneğin Ocak 2003'te MS SQL Server'larda buffer overflow hatasını tetikleyen SQL Slammer solucanı DoS saldırıları ile global ölçekte internet trafiğini yavaştırmıştır. 30 dk gibi bir süre içinde 250 binden fazla host bundan etkilenmiştir.

1988: Morris Worm

1999: Melissa Worm

2000: Love Bug Worm

2001: Code Red Worm

2003: SQL Slammer

...

2008: Conficker

Solucanlar genel itibarı ile beş fazda incelenir.

1. **Probe** : Keşif sürecidir. Zaafiyet bulunan sistemin bulunmasına yönelik bir aşama. Örneğin ICMP ile L3 düzeyinde hedef belirlenir ve L7 taramalarla uygulama zaafiyeti olup olmadığı tespit edilir.
2. **Penetrate**: Sızma sürecidir. Zararlı kodların hafızaya ya da bir uygulamaya bulaşması, eklenmesidir.
3. **Persist**: Devamlılık sürecidir. Hafızaya saldırı başarı ile sağlanırsa, hedef sistem üzerinde devamlılığını sağlamak sürecidir. Sistem yeniden başlatıldığında da kodların çalışmasını sağlamak gibi... Bu amaçla sistem dosyaları, registry kayıtları gibi değişiklikler yapılır.
4. **Propogate**: Yayılma sürecidir. Öncelikle yakın cihazların taranarak diğer cihazlara bulaşma denemesidir. Örneğin, IRC, FTP, RDP gibi servislerle bulaşmaya çalışması gibi.
5. **Paralyze**: Hizmet durdurma, aksatma sürecidir. Sisteme asıl zarar bu aşamada verilir. Yazılan kodun amacına uygun olarak, dosyalar silinebilir, bilgiler çalınabilir, hizmet engellenebilir (DoS)

Worm and Virus – Exploit Comparison (~10 Yrs)							
	Code Red 2001	Slammer 2003	MyDoom 2004	Zotob 2005	MS RPC DNS 0day 2007	Koobface 2008	Morto 2011
Probe	Scans for IIS	N/A	N/A	Scans for MS directory services	Scans for endpoint Mapper query	N/A	Scans for Windows systems listening on RDP port (3389)
Penetrate	Causes buffer overflow in IIS	Causes buffer overflow in SQL and MSDE	Arrives as email attachment	Causes buffer overflow in UPnP service	Causes buffer overflow in RPC service	Arrives as Facebook message	Attempts login as Administrator to RDP
Persist	Executes script to download code	N/A	Creates executables and edits the registry	Creates executables and edits the registry, downloads code	Executes payload to download code	Directs user to download malicious code	Creates new files
Propagate	Picks new addresses and spreads to new victims	Picks new addresses and spreads to new victims	Opens address book and emails copies of itself to new victims	Starts FTP and TFTP services, looks for addresses and spreads to new victims	Looks for addresses and spreads to new victims	Sends messages to Facebook friends, builds botnet	Scans local network for other victims
Paralyze	Spawns many threads which slow the system	Generates many packets which slows the network	Worm spreads	Deletes registry keys and files, and terminates processes	Worm spreads	Steals confidential information, injects advertising, blocks access to sites	Worm spreads

TROJAN HORSE (TRUVA ATI)

Tespiti zor olmasına rağmen, sistem ve network kaynaklarını kullandıklarından sistemde ve ağıda oluşan anormal yavaşlıklardan anlaşılabilirler.

1998: NetBus

1998: Back Orifice

1999: Sub7

2003: ProRat

2004: Vundo

2005: SpySheriff

2005: Zlob

2007: Storm

2007: Zeus

2008: Torping

2008: Koobface (Java Based, MultiPlatform, MS, MACOS, Linux)

2011: ZeroAccess

Uzak erişim sağlamak, dosya göndermek, port açmak (Ör: 21), Proxy Server yapmak, DoS saldırısı yapmak gibi birçok farklı amaçlarla kullanılabilirler.

Sosyal platformların ve internet tabanlı iletişimler sayesinde en küçük bir tehdidin, dünya çapında yayılması çok kısa sürelerde gerçekleşebildiği göz önüne alınırsa, güvenlik önlemlerinin ne derece acil olarak uygulanması gerektiği ortaya çıkacaktır. Bu sebeple güncel

ya da sürekli güncellenen bir antivirüs yazılımı kullanmak gereklidir. Antivirüs kullanımı Security Policy olarak da uygulanması kritik bir öneme sahiptir.

Worm'lar virüslere göre daha fazla network etkileşimi yaparlar.

ATAK METODOLOJİLERİ

Genel itibarı ile ataklar, **Reconnaissance Attacks**, **Access Attacks** ve **Denial of Service Attacks** olmak üzere üç kategoride incelenebilir.

Reconnaissance Attacks:

Sistemlerdeki ya da servislerdeki zayıflıkların tespitine yönelik keşif ataklarıdır. Port Scanner ve Packet Sniffer bu aşamada sıkılıkla kullanılan araçlardır.

Kullanılan Programlar: Packet Sniffers, Ping Sweeps,

Access Attacks:

Web sayfalarına, veritabanına ya da bunun gibi hassas bilgilerin bulunduğu bir hizmete/uygulamaya erişim için yapılan saldırılardır. Örneğin parolayı tahmin etmek için sözlük kullanarak yapılan Brute-Force saldırılar gibi.

Programlar: L0phtCrack, LC5 (Brute-Force); Trojan Horse, Key Logger

Access ataklar, logların incelenmesi, bandwidth ve process yönetimi ile anlaşılabilir.

Denial of Service Attacks:

Hedefe aşırı derecede istek göndermektir. Hedef cihaz, çoğu zaman sahte ya da eksik olan bu isteklere cevap vermek ile uğraşırken esas görevini yerine getiremez.

Ping of Death, Smurf Attack (Broadcast adrese gönderilen ping), TCP SYN Flood (Half-Open TCP Connection)

Mitigating Network Attacks

Reconnaissance Attack Mitigation Techniques include:

- Implement authentication to ensure proper access.
- Use encryption to render packet sniffer attacks useless.
- Use anti-sniffer tools to detect packet sniffer attacks.
- Implement a switched infrastructure.
- Use a firewall and IPS.

- Güçlü parolalar kullanmak. Gerekirse one-Time Password kullanılabilir.
- Encryption (Kriptolama) kullanmak

- Anti-Sniffer kullanarak sniffer kullanımını tespit etmek.
- Efektif olarak Switch kullanımı.
- IPS/IDS kullanımı

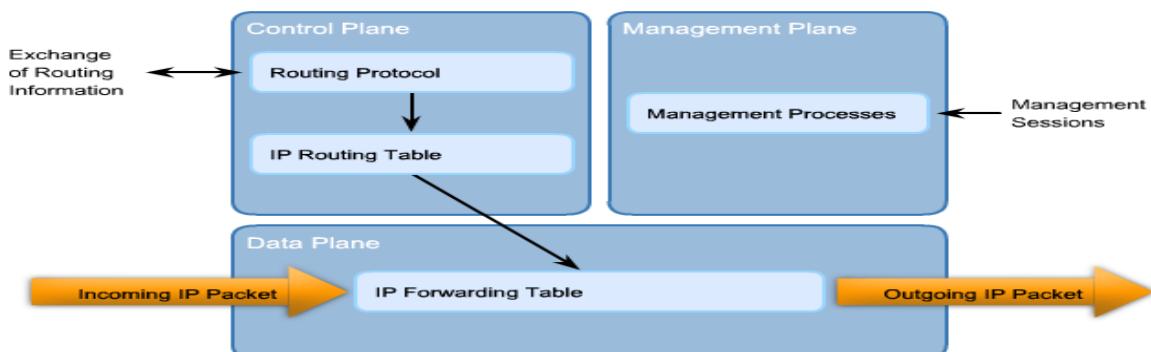
Bunların yanında Access atakları önlemek amacıyla belli bir sayıdan sonra başarısız login işleminde hesabı kilitlemek de faydalıdır. Network dizayn edilirken *minimum given* ilkesi ile tasarlanmalıdır. Yine uygulamaların ve işletim sistemlerinin güncel yamalarının uygulanması da güvenlik açısından özellikle de solucanları önlemeye yönelik son derece önemlidir.

Network Ataklarına karşı en iyi 10 önlem:

- 1- Güncel yamaları uygulama
- 2- Gereksiz hizmet ya da portların kapatılması
- 3- Güçlü parola kullanımı ve periyodik değişimi
- 4- Sistemlere fiziksel erişimin kontrol altına alınması
- 5- Web sayfalarında gereksiz girişlerden kaçınma. (SQL Injection vs.)
- 6- Periyodik yedekleme ve yedeklerin sağılıklı olduğunun testi
- 7- Çalışan personele eğitim
- 8- Parola ve kritik bilgilerin kriptolanması
- 9- Antivirüs, İçerik filtreleme, Firewall, IPS, VPN çözümlerini kullanmak
- 10- Yazılı bir güvenlik ilkesinin kullanılması.

Cisco Network Foundation Protection (NFP)

Network altyapısını korumaya yönelik olarak geliştirilen bir frameworktür. Bu yapıda routerlar ve switchler işlevselliklerine göre üç mantıksal kısma ayrırlırlar.



Control Plane : Verilerin doğru bir şekilde yönlendirilmesinden sorumludur. Genellikle cihazların üretikleri trafikleri denetler(ARP, OSPF vs.)

Secure the Control Plane using:

- AutoSecure
- Routing protocol authentication
- Control Plane Policing (CoPP)

Burada Cisco Auto Secure, Routing Protocol kimlik denetimi ve Control Plane Policing (CoPP) kullanılır. CoPP, gereksiz trafiklerin route işlemcisine erişimini önlemek için dizayn edilmiştir.

Management Plane: Ağ unsurlarının yönetiminden sorumludur. (AAA, Telnet, SNMP vs.)

Secure the Management plane by:

- Enabling login and password policy
- Presenting legal notification
- Ensuring the confidentiality of data using SSH and HTTPS
- Enabling role-based access control
- Authorizing actions
- Enabling management access reporting

Data Plane: Verilerin iletiminden sorumludur. Kullanıcıların ürettiği trafiklerin uç sistemlere iletildmesinden sorumludur.

Secure the Data plane using:

- ACLs
- Antispoofing
- Layer 2 security including port security, DHCP snooping, dynamic ARP inspection (DAI)

Cihaz Hardening Uygulanması

Giden network trafiginin güvenliğini sağlamak ve gelen trafigi incelemek network güvenliği açısından kritik bir öneme sahiptir. Ağ güvenliğini sağlamak takımlı ilk önemli adım, dış ağa bağlanan sınır router'ın (Edge Router) güvenliğini sağlamaktır. Ağ güvenliğinde cihazları güvenlik yönünden güçlendirmek kritik bir görevdir. Routerin fiziksel güvenliğinin yanında CLI ya da CCP ile erişiminin de kontrol altınması gereklidir.

Geçmişte bazı sebeplerden dolayı bazı hizmetler veya özellikler varsayılan olarak açık gelir. Ancak günümüzde bunların açık olmasının bir gerekliliği yoktur. **auto secure** komutu ile bu özellikler kapatılabilir.

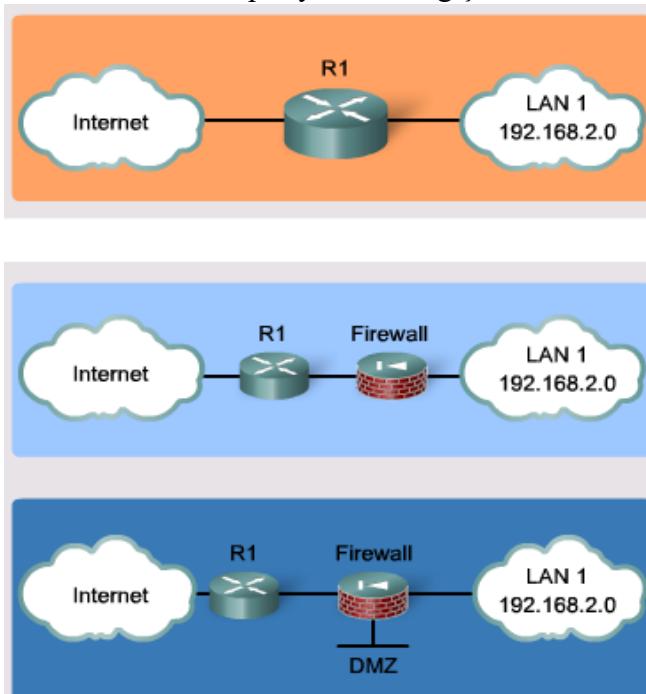
SECURING EDGE ROUTER

Genellikle sınır routerlar yandaki şekilde gösterildiği gibi 3 farklı şekilde dizayn edilmiştir.

Bu cihazlara erişim inband ya da outband gibi birkaç yolla olur. Telnet, SSH, WEB, SNMP gibi IP kullanılan yöntemler inband yöntemlerdir. Console gibi cihaza fiziksel erişim gerektiren yöntemler de outband yöntemlerdir.

Her iki türdeki erişimlerde de parolalar kullanılmalıdır. Router ya da diğer ağ cihazlarına erişim için Cain&Abel gibi araçlar kullanılarak Brute-Force attack yapılabilir. Bu sebeple aşağıda belirtilen önlemler göz önünde tutularak parola güvenliği sağlanmalıdır.

- 10 ya da daha fazla karakterden oluşan parola kullanımı
- Küçük / Büyük harf, sayı ve işaretler kullanılarak komplex parolalar kullanımı
- Sözlük atakları ile tespit edilebilecek kelimeler, kişisel ya da biyografik bilgilerden oluşan parolaların kullanılmaması
- Parolaların bilinçli bir şekilde benzetim ile değiştirilmesi. Örneğin “Security” yerine “SecurIty” gibi bir parola kullanımı.
- Parolaların belli periyotlarla değiştirilmesi. Önerilen süre, parolanın kırılabilirlik sınırının



yarısı dolmadan değiştirilmesi. Örneğin 7 karakterden oluşan bir parolanın kırılması 6 ay sürerse, 3 ay dolmadan parola değiştirilmelidir.

- Parolaların kağıda, masaya ya da ulaşılabilir herhangi bir yere yazılmaması gereklidir.

2.1.1 ROUTER ERİŞİM GÜVENLİĞİ

Console Parolası: Cihaza konsol erişiminde varsayılan olarak parola yoktur. Ancak güvenlik açısından parola kullanılması son derecede önemlidir. Bunun için;

```
Router(config)#  
Router(config)#line console 0  
Router(config-line)#password ConP@55
```

```
Router(config-line)#login
```

Enable Secret Parolası: Cihaza erişimde global konfigürasyon moduna geçmek için privileged EXEC modunda aşağıdaki gibi bir yapılandırma yapılır:

```
Router(config)#enable secret EnaP@55
```

Virtual Terminal Lines (VTY) : Varsayılan olarak cisco cihazlar eş zamanlı 5 vty destekler. Bu erişimleri parola ile korumak için:

```
Router(config)#line vty 0 4
```

```
Router(config-line)#password VtyP@55
```

```
Router (config-line)#login
```

Cihazlara erişimi daha güvenli hale getirmek için yukarıda belirtilen parolaların yanında aşağıdaki güvenlik önlemlerini de sağlamak gereklidir.

Minimum Parola Uzunluğu : Cisco IOS 12.3(1) ve sonraki versiyonlarda gelen bir özellik ile parola uzunlukları 16'ya kadar kullanma zorunluluğu getirilebilir.

```
Router(config)#security passwords min-length 10
```

```
Router(config)#enable secret FB
```

```
% Password too short - must be at least 10 characters. Password not configured.
```

```
Router(config)#enable secret FenerB@hce1907
```

```
Router(config)#
```

TimeOut Süresi Yapılandırma: Varsayılan olarak 10 dakika boyunca herhangi bir etkileşim olmadığında timeout gerçekleşecektir. Güvenlik açısından bu sürenin de belirlenmesi gereklidir. İdeal süre kurumsal yapıya göre değişmekte beraber 2-3 dakikadır. Aşağıdaki örnekte 2 dakika 30 saniye olarak bu süre belirlenmiştir.

```
Router(config)#line console 0
```

```
Router(config-line)#exec-timeout 2 30
```

```
Router(config-line)#
```

Parola Kriptolama: Cisco yapılandırma dosyasında **enable secret** haricindeki çoğu parolalar açık kriptosuz bir şekilde görünür. Bu sebeple kriptolanması gereklidir. Ancak burada kullanılan kripto Type-7 kriptolamadır ve kolaylıkla kırılabilir.

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

```
Router#show runn
```

```
...
```

```
line con 0
```

```
exec-timeout 2 30
```

```
password 7 0802434039395042 //kriptolu görünüm
```

```
login
```

```
...
```

Cisco cihazlarda kimlik denetiminde kullanılmak amacıyla kullanıcı adı ve parolanın lokal veri tabanında saklanarak kullanılabilir. Yine bu yapıda da parolalar md5 ya da Type-7 olarak saklanabilir.

Aşağıdaki örnekte service password-encryption kullanılmışsa **Type-7** kriptolama yapılır.

```
Router(config)#username admin password type7Ornek  
Router(config)#username erdal password benimparolam
```

Aşağıdaki örnekte ise **md5** kriptolama yapılır.

```
Router(config)#username admin secret 1234567890  
Router(config)#username erdal secret benimparolam
```

2.1.3 VTY GÜVENLİĞİNİ ARTIRMA

Vty erişimi kurum dışından cihazlara erişmek için sıkılıkla kullanılan bir yöntemdir. Bu sebeple brute-force gibi saldırılara karşı açıktır. VTY güvenliğini artırmak için, cihaza bağlanmaması gereken IP adreslerinin bloklanması, bağlanabilecek IP adreslerinin belirlenmesi, yetkisiz erişim teşebbüsünden sonra bloklama gibi bir takım extra önlemler almak gereklidir.

VTY nin maximum oturum sayısı kadar oturum açma talebinde bulunan bir saldırgan, başarısız olsa da bu süre içerisinde cihaza bağlanmak isteyen kişinin erişim yapmasını önlemiş olur. Bu durumu çözmek için, kullanıcılarından gelen vty talepleri belli bir süre içerisinde belli sayıda hatalı login işlemi başlatırsa, cihaz kendini bloklayacaktır. Bu durumda ACL ile belirlenen kullanıcılar haricindekilerin bir süreliğine (quiet-time) cihaza erişimi bloklanacaktır. Böylece cihaza yetkisiz erişim teşebbüsü için yapılan DoS saldırıları etkisiz kalacaktır.

```
Router(config)#login block-for 15 attempts 5 within 30
```

Yukarıdaki örnekte, 30 saniye içinde 5 geçersiz login erişimi olduğunda 15 saniye boyunca cihaz bloklanacaktır.

Örneğin IP adresi 192.168.1.0/24 olanları ACL ile belirtip, bunların dışındakilerin bu süre içerisinde erişimini durduralım.

```
R1(config)#access-list 5 permit 192.168.1.0 0.0.0.255  
R1(config)#login quiet-mode access-class 5
```

Girişimler arasındaki bekleme süresini 5 saniye olarak belirleyelim.

```
R1(config)#login delay 5
```

Hatalı ya da başarılı login işlemlerinin log kayıtlarının tutulması istenirse aşağıdaki komutlar kullanılabilir.

```
R1(config)#login on-failure log //Başarısız login denemeleri loglanır  
R1(config)#login on-success log //Başarılı loginler loglanır
```

**** login on-failure log every 5** komutu ile her 5 başarısız login işleminde log görüntülenecektir. Default değer 1 dir. Yani her başarısız login için log görüntülenir.

Ayrıca aşağıdaki gibi bir yapılandırma ile de bu değer değiştirilebilir.

```
R1(config)#security authentication failure rate 5 log
```

vty login işlemlerine yönelik yaptığımız yapılandırmayı görmek için;

```
R1#show login
    A login delay of 2 seconds is applied.
    Quiet-Mode access list 5 is applied.
    All failed login is logged.

    Router enabled to watch for login Attacks.
    If more than 5 login failures occur in 30 seconds or less,
    logins will be disabled for 60 seconds.

    Router presently in Normal-Mode.
    Current Watch Window
        Time remaining: 7 seconds.
        Login failures for current window: 0.
    Total login failures: 13.
```

Başarısız login denemelerini görmek için :

```
R1#show login failures
Total failed logins: 13
Detailed information about last 50 failures

Username      SourceIPAddr   1Port Count TimeStamp
admin         192.168.137.1  23     6      00:14:02 UTC Fri Mar 1 2002
asd          192.168.137.1  23     5      00:15:51 UTC Fri Mar 1 2002
asdasd       192.168.137.1  23     1      00:14:17 UTC Fri Mar 1 2002
ad           192.168.137.1  23     1      00:15:47 UTC Fri Mar 1 2002
R1#
```

Yukarıda belirtilen bu güvenlik önlemlerinin yanında erişim teşebbüsünde bulunanlar için aşağıdaki gibi bir uyarı yazısı görüntülemek de son derece önemlidir.

```
R1(config)#banner login # $(hostname) cihazina yetkisiz erisim
yasaktir #
```

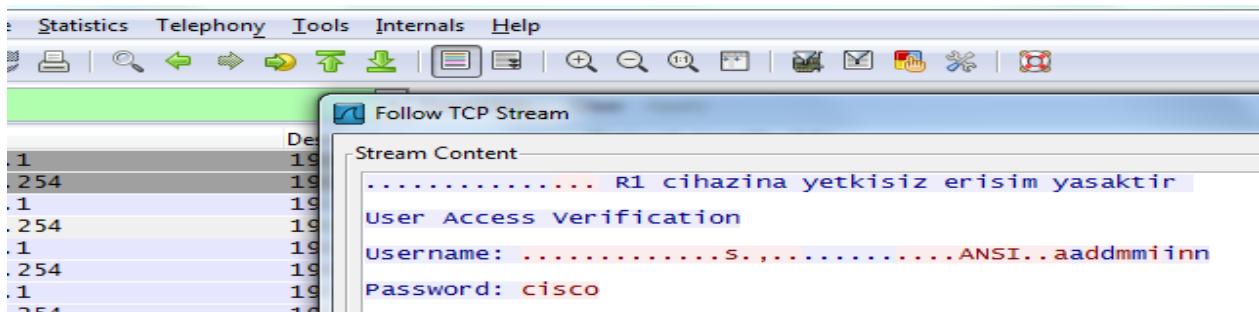
*** \$(hostname) ile cihaz adı; \$(domain) ile domain adı da bu uyarı içerişine eklenebilir.
Erişim teşebbüsünde aşağıdaki gibi bir uyarı ekranı oluşacaktır.

```
R1 cihazina yetkisiz erisim yasaktir
User Access Verification
Username:
```

Ancak cihaz adı ya da domain adının login ekranında görüntülenmesi sakıncalı olabilir.

SSH YAPILANDIRMA

Yukarıdaki yapılandırmada telnet ile cihaza erişim yapılmıştı. Telnet ile yapılan bu erişim Wireshark gibi bir program ile takip edildiğinde aşağıdaki resimde de gösterildiği gibi, kullanıcı adı ve parolanın ele geçirilmesi mümkündür.



Telnet iletişiminde veriler kriptolanmadan plain-text olarak gönderilir. Bu sebeple güvenli olması açısından telnet yerine **ssh** kullanılması önerilir.

SSH yapılandırmak için öncelikle cihaza benzersiz bir Hostname vermek ve cihazın domain adını belirtmek gereklidir.

Router(config)#hostname RTR1

RTR1(config)#ip domain name erdal.com

Ardından bir RSA key oluşturulur. Aşağıdaki örnekte 1024-bitlik bir key oluşturulmuştur.

RTR1(config)#crypto key generate rsa modulus 1024

Oluşturulan bu anahtarı görüntülemek için aşağıdaki komutu kullanın.

```
RTR1#show crypto key mypubkey rsa
% Key pair was generated at: 02:26:14 UTC Mar 1 2002
Key name: RTR1.erdal.com
Storage Device: not specified
Usage: General Purpose Key
Key is not exportable.
Key Data:
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00BBFA68
B432D4CD 979324E8 67266574 DC116F97 BE03A16F 8C72A1F2 8438EEC0 B9F6A356
65AC3C01 887CE6D5 B62FDCEA DC69BE5A 75D1CCD8 23CDD4F2 3CEF8614 2280661A
145D4B21 0E48E186 AFBEB07D CF87D76C 30678538 3E58B61D CE6379E8 C9463801
C0BD1390 2E6E8324 89743ABF 468FB33F D3FA3B14 3F5F4522 3EEE21BB 55020301 0001
% Key pair was generated at: 02:26:18 UTC Mar 1 2002
Key name: RTR1.erdal.com.server
Temporary key
Usage: Encryption Key
Key is not exportable.
Key Data:
307C300D 06092A86 4886F70D 01010105 00036B00 30680261 009C391B A3352E9C
428C7DBC 8CA68B21 17805436 EC535193 6C646330 1F44A1ED 6723BF62 76C6CB64
48FD8B31 FB7E3BC9 E0392D13 6A48908F 15660A73 08736E20 0BD98168 DD969108
46CA4904 0CAFEB6D 3D31A1F5 20AA306C 218501E7 DAE4EB79 3D020301 0001
```

Yukarıda belirtilen komutu kullanmadan önce daha önceden RSA anahtarı varsa silinmesi için aşağıdaki komutu kullanın.

RTR1(config)#crypto key zeroize rsa

% All RSA keys will be removed.

% All router certs issued using these keys will also be removed.

Do you really want to remove these keys? [yes/no]: yes

RTR1(config)#!

***Mar 1 02:32:54.939: %SSH-5-DISABLED: SSH 1.99 has been disabled**

```
RTR1(config)#
```

Şimdi de vty için ssh protokolünü devreye alalım.

```
RTR1(config)#line vty 0 4
```

```
RTR1(config-line)#transport input ssh
```

SSH'ı daha güvenli hale getirmek için aşağıda belirtilen komutlar kullanılır.

```
RTR1(config)#ip ssh version 2 //Version 2 SSHı kullanmak için
```

```
RTR1(config)#ip ssh time-out 30 //Time-Out süresi 60 sn.(Def=120)
```

```
RTR1(config)#ip ssh authentication-retries 3 //Tekrar deneme sayısı 3
```

Bir router'dan ssh bağlantısı için; **ssh -l Admin 192.168.1.1 komutu kullanılır.

Konu ile ilgili diğer komutlar.

```
RTR1#show ip ssh
```

```
SSH Enabled - version 2.0
```

```
Authentication timeout: 60 secs; Authentication retries: 3
```

HTTP Server Yapılandırma:

```
RTR1(config)#ip http server
```

```
RTR1(config)#ip http secure-server
```

```
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
```

```
*Mar 1 03:07:22.119: %PKI-4-NOAUTOSAVE: Configuration was modified. Issue "write memory" to save new certificate
```

```
RTR1(config)#ip http authentication local
```

```
RTR1(config)#ip http access-class 5
```

2.2.1 Privilege Levels (Yetki Düzeyleri)

Cisco IOS'larda varsayılan olarak 16 düzey bulunmaktadır. Level0, Level1 ve Level15 önceden tanımlıdır. Ancak aradaki leveller tanımlı değildir.

Level1, user moda karşılık gelir ve **Router>** modu olarak düşünülebilir. Level15 ise enable modunda karşılık gelir ve **Router#** olarak düşünülebilir.

Örnek: Level5 tanımlayalım ve routing tablosunu görme ile ping atma yetkisi verelim.

```
RTR1(config)#privilege exec level 5 show ip route
RTR1(config)#privilege configure level 5 router
RTR1(config)#privilege exec level 5 configure terminal
RTR1(config)#

```

Level5'e geçmek için

```
RTR1#enable 5
RTR1#show run?
% Unrecognized command
RTR1#show ip route
...
C 192.168.137.0/24 is directly connected, FastEthernet0/0
...
RTR1(config)#ip route ?
```

```
% Unrecognized command
```

Level 5'e parola atayalım.

```
RTR1(config)#enable secret level 5 ciscoPass5
```

Oluşturulan bu levellere kullanıcı da atanabilir. Örneğin Level5 haklarına sahip bir kullanıcı oluşturalım.

```
RTR1(config)#username HalfAdmin privilege 5 secret AdPass5
```

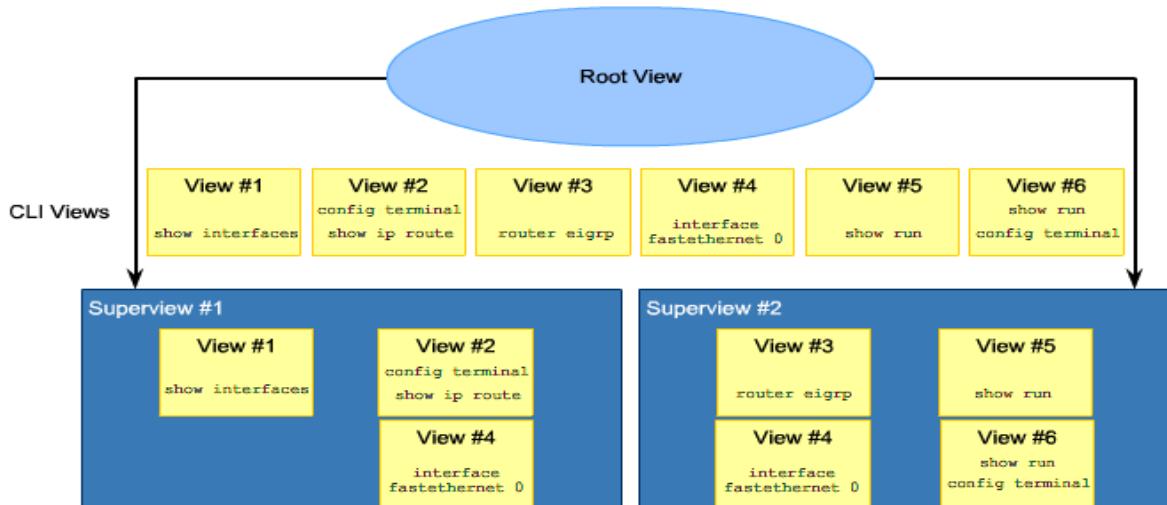
Kullanıcı adı HalfAdmin ve Parolası AdPass5 olan bir Level5 kullanıcısı oluşturuldu. Bu kullanıcı giriş yaptığımda Level5 haklarına sahip olacaktır.

Not: Bir leveldeki kullanıcılar, alt leverllerdeki tüm haklara sahiptir. Örneğin level 5, 1-2-3-4 leverllerin tüm haklarına sahiptir.

Hangi düzeyde olduğumuzu görmek için **show privilege** komutu kullanılır.

2.2.2 ROLE BASED CLI

Cisco IOS 12.3(11)T ve sonrası güncellemelerde var olan Role-Based CLI ile Privilege Level yapısına göre daha esnek bir yapıya geçilmiştir. Bu yapıda Root View, Cli View ve Super View olmak üzere üç kategori bulunmaktadır. Root View, Level 15 yapıya denk gelir. Cli View ise ihtiyaca göre şekillendirilir. Super View birden fazla Cli View'den oluşur.



CLI VIEW YAPILANDIRMA:

```
RTR1(config)#aaa new-model  
RTR1#enable view root  
Password: *****  
RTR1#conf t  
RTR1(config)#parser view test  
*Mar 1 04:40:34.586: %PARSER-6-VIEW_CREATED: view 'test' successfully created.  
RTR1(config-view)#secret v1Pass //test View için parola tanımlandı.
```

```

RTR1(config-view)#commands exec include all show
RTR1(config-view)#exit
RTR1(config)#parser view test2
*Mar 1 04:43:29.982: %PARSER-6-VIEW_CREATED: view 'test2' successfully created.
RTR1(config-view)#secret v2Pass //test2 View için parola tanımlandı.
RTR1(config-view)#commands exec ?
  exclude          Exclude the command from the view
  include           Add command to the view
  include-exclusive Include in this view but exclude from others
RTR1(config-view)#commands exec include ping
RTR1(config-view)#commands exec include telnet
RTR1(config-view)#commands exec include traceroute
RTR1(config-view)#exit
RTR1(config)#

```

Şimdi bu oluşturduğumuz, viewlerden test2'ye geçiş yapalım.

```

RTR1#enable view test2
Password:****
RTR1#conf t
  ^
% Invalid input detected at '^' marker.

RTR1#?
Exec commands:
  credential  load the credential info from file system
  enable      Turn on privileged commands
  exit        Exit from the EXEC
  ping        Send echo messages
  show        Show running system information
  telnet      Open a telnet connection
  traceroute  Trace route to destination

```

Göründüğü gibi **configure terminal** komutu test2 için çalışmadı. test2 sadece **ping**, **telnet** ve **traceroute** komutlarını kullanabilir. Şimdi de DESTEK adlı bir süper view oluşturup bu iki view'i bu grup altında bağlayalım.

```

RTR1(config)#parser view DESTEK superview
*Mar 1 04:52:58.234: %PARSER-6-SUPER_VIEW_CREATED: super view 'DESTEK'
successfully created.
RTR1(config-view)#secret destekPass
RTR1(config-view)#view test
*Mar 1 04:53:36.958: %PARSER-6-SUPER_VIEW_EDIT_ADD: view test added to superview
DESTEK.

```

```

RTR1(config-view)#view test2
*Mar 1 04:53:39.666: %PARSER-6-SUPER_VIEW_EDIT_ADD: view test2 added to superview
DESTEK.
RTR1(config-view)#end
RTR1#enable view DESTEK
Password:*****
RTR1#
*Mar 1 05:01:11.434: %PARSER-6-VIEW_SWITCH: successfully set to view 'DESTEK'.
RTR1#show parser view all
Views/SuperViews Present in System:
 test
 test2
 DESTEK *
-----(*) represent superview-----

```

IOS ve YAPILANDIRMA DOSYASI GÜVENLİĞİ

Cisco cihazlara erişen yetkisiz kişiler (saldırgan ya da bilinçsiz kullanıcı), cihazda bir çok şey yapabilir. Örneğin verileri yanlış yönlendirebilir, iletişimini dinleyebilir. Ayrıca cihazın işletim sistemini ya da startup-config dosyasını silebilir. Bu durumda cihazın tekrar işlevsel hale getirilmesi uzun zaman alacaktır. Cisco IOS Resilient Configuration özelliği dosya işlemleri üzerinde güvenlik sağlayarak kopyalama, değiştirme ve silmeye izin vermez. Cisco IOS imajı ve running-config yapılandırma yedeği flashta görünmez. Bunun için **secure boot-image** ve **secure boot-config** komutları kullanılır.

```

RTR1(config)#secure boot-image
%IOS_RESILIANCE-5-IMAGE_RESIL_ACTIVE: Successfully secured running image
RTR1(config)#secure boot-config
%IOS_RESILIANCE-5-CONFIG_RESIL_ACTIVE: Successfully secured config archive
[flash:.runcfg-19930301-000035.ar]
RTR1(config)#secure boot-config restore //Geri yükleme
Router# show secure bootset
IOS resilience router id FHK085031MD

IOS image resilience version 12.3 activated at 05:00:59 UTC Fri Feb 10 2006
Secure archive flash:c1841-advsecurityk9-mz.123-14.T1.bin type is image (elf) []
file size is 17533860 bytes, run size is 17699528 bytes
Runnable image, entry point 0x8000F000, run from ram

IOS configuration resilience version 12.3 activated at 05:01:02 UTC Fri Feb 10 2006
Secure archive flash:.runcfg-20060210-050102.ar type is config
configuration archive size 4014 bytes

```