

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. (200Mbps x 8= 1600 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: ... xxxxxxx0 Addr2: ... xxxxxxx0	... xxxxxxx0: Use link 0
Addr1: ... xxxxxxx0 Addr2: ... xxxxxxx1	... xxxxxxx1: Use link 1
Addr1: ... xxxxxxx1 Addr2: ... xxxxxxx0	... xxxxxxx1: Use link 1
Addr1: ... xxxxxxx1 Addr2: ... xxxxxxx1	... xxxxxxx0: Use link 0

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

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

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

Genel olarak bu fiziksel portların aynı VLAN'de olması gerekir. 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) cisonun 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 switch'e EtherChannel yapılandırma isteği gönderir. **Auto** (default) modda ise, local switch karşı switch'ten 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 switch'leri dinler ve onlardan istek aldığı anda Etherchannel'a katılır. **desirable**

modunda ise port aktif olarak uzaktaki potansiyel katılımcıya Etherchannel kurulum isteği gönderir. Varsayılan olarak PAgP, desirable ve auto modlarıyla birlikte **silent** alt modunda çalışır. Bu alt mod karşı taraftaki porttan herhangi bir PAgP paketi alınmasa dahi Etherchannel kurulabilmesi anlamına gelmektedir. Bu özellik, Etherchannel kurmak istediğimiz uzak cihaz bir sunucuysa ve sunucunun PAgP kurulum sürecinde herhangi bir PAgP paketi göndermesi mümkün olmadığından yine de Etherchannel'in kurulabilmesini sağlar. Bu alt modu 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 gerekir. 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.

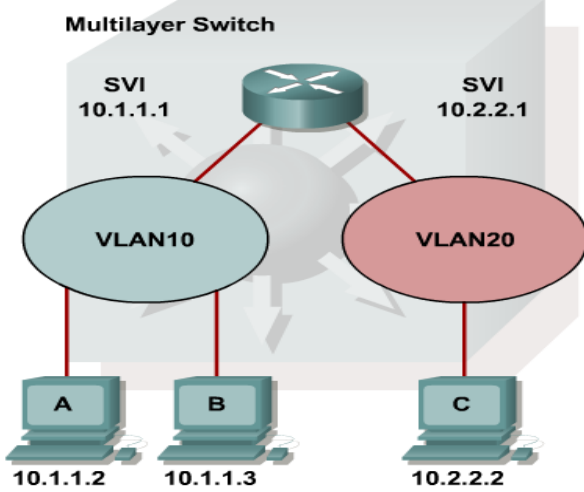
Portların modu **ON** olarak seçilmişse, LACP ve PAgP paketleri gönderilmez.

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
```

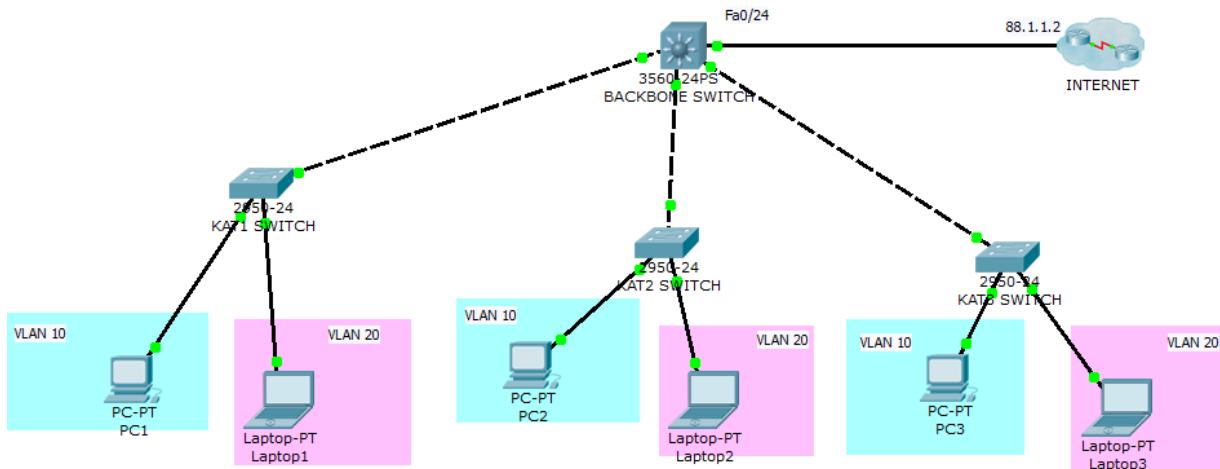
MULTI-LAYER SWITCH



Layer3 ya da daha üst katmanlarda çalışan switchlerdir. Dolayısıyla routing kabiliyetleri olan switchlerdir. Daha önceki konularda farklı VLAN'leri haberleştirebilmek 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 gerekir.

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 switchte bağlı olan portları Dynamic modda olduğu için, otomatik olarak trunk olacaktırlar. Bu sebeple DTP protokollü 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.

```
KAT3_SWITCH#sh vlan
```

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 Backbone 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

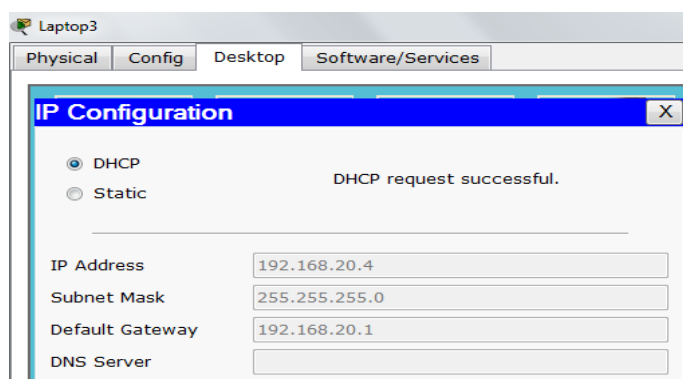
BACKBONE(config-if)#interface vlan 20
%LINK-5-CHANGED: Interface Vlan20, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to up
BACKBONE(config-if)#ip address 192.168.20.1 255.255.255.0
```

PC'lerin ve Laptopların otomatik IP alabilmeleri için BackBone switchi DHCP Server olarak yapılandırılalım.

```
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
```

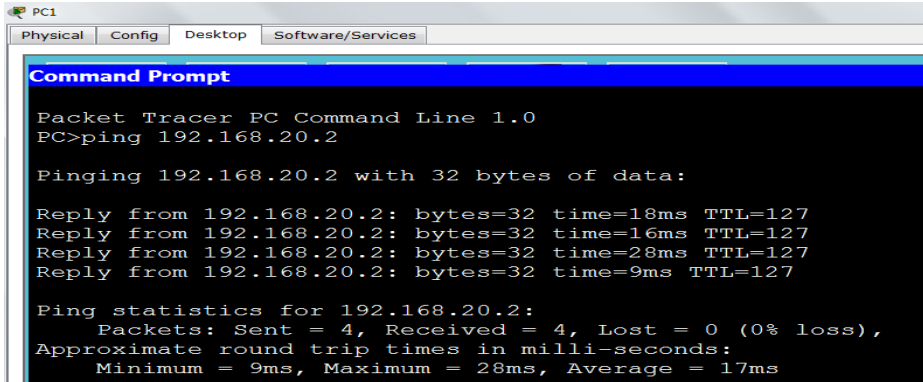
Test için Laptop3'ün IP alıp almadığı kontrol edilebilir.



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 haberleşebilecektir. Test için PC1'den Laptoplara ping atalım



```
PC1
Physical Config Desktop Software/Services
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.2

Pinging 192.168.20.2 with 32 bytes of data:

Reply from 192.168.20.2: bytes=32 time=18ms TTL=127
Reply from 192.168.20.2: bytes=32 time=16ms TTL=127
Reply from 192.168.20.2: bytes=32 time=28ms TTL=127
Reply from 192.168.20.2: bytes=32 time=9ms TTL=127

Ping statistics for 192.168.20.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 9ms, Maximum = 28ms, Average = 17ms
```

L3 Switch için internete erişim 88.1.1.2 üzerinden gerçekleştirilmektedir. O halde switchin internete 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 gerekir. 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.

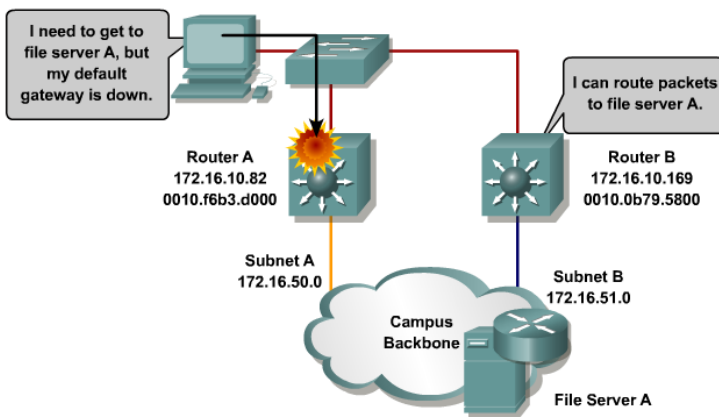
```
PC1
Physical Config Desktop Software/Services
Command Prompt
PC>
PC>ping 88.1.1.2

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
```

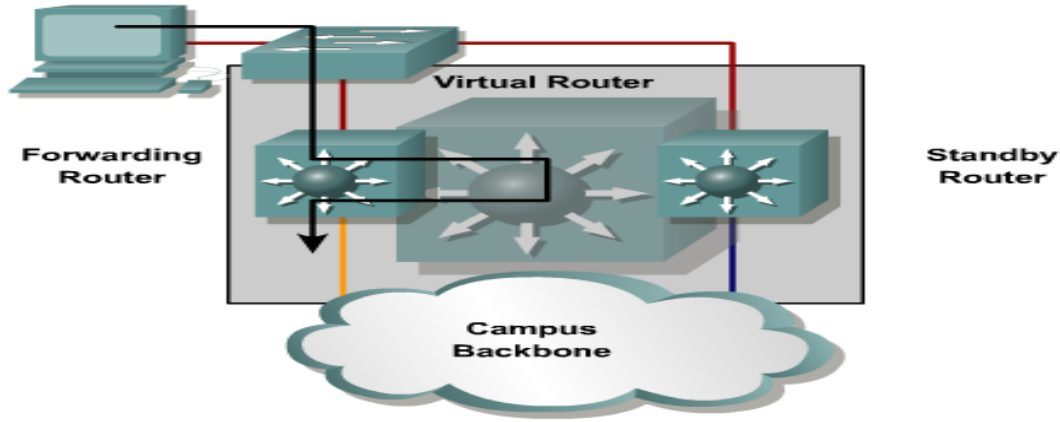
HIGH AVAILABILITY IN A CAMPUS ENVIRONMENT



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 davranması 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.

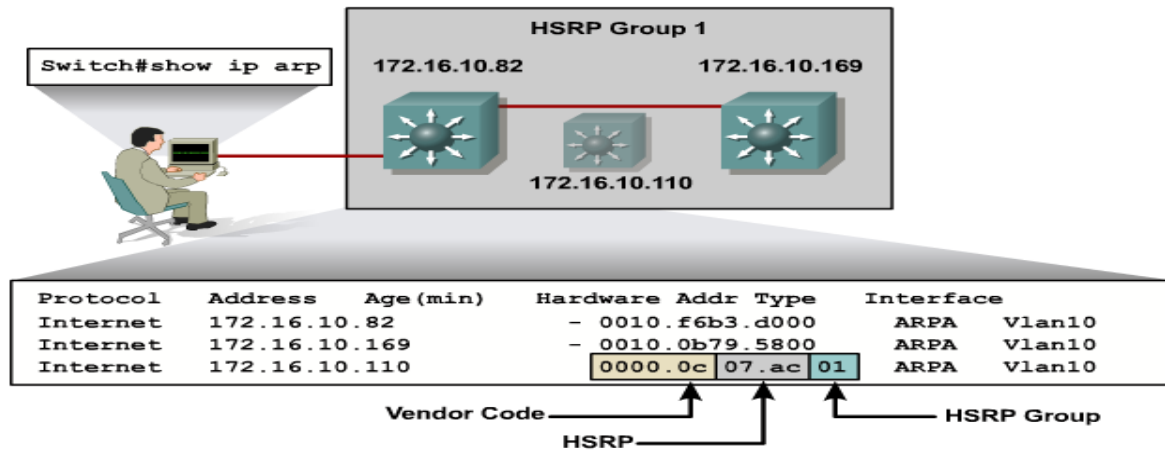
HSRP (HOT STANDBY ROUTER PROTOCOL)

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 routerdır.

Standby Router, Aktif router'ın yedeğidir.

Burada Group MAC adresi, **0000.0C07.ACXX** formundadır. **XX** burada grup numarasıdır. Aşağıdaki örnekte HSRP Group 1 için IP ve MAC yapılandırması görülmektedir.

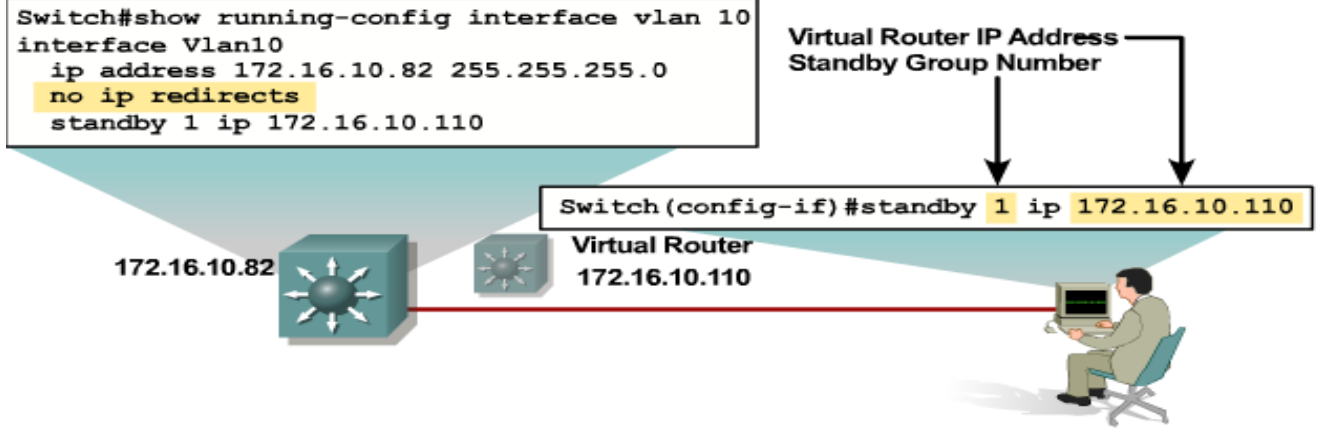


YAPILANDIRMA ve DOĞRULAMA

```
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 Virtual 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 routerlarda 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
```

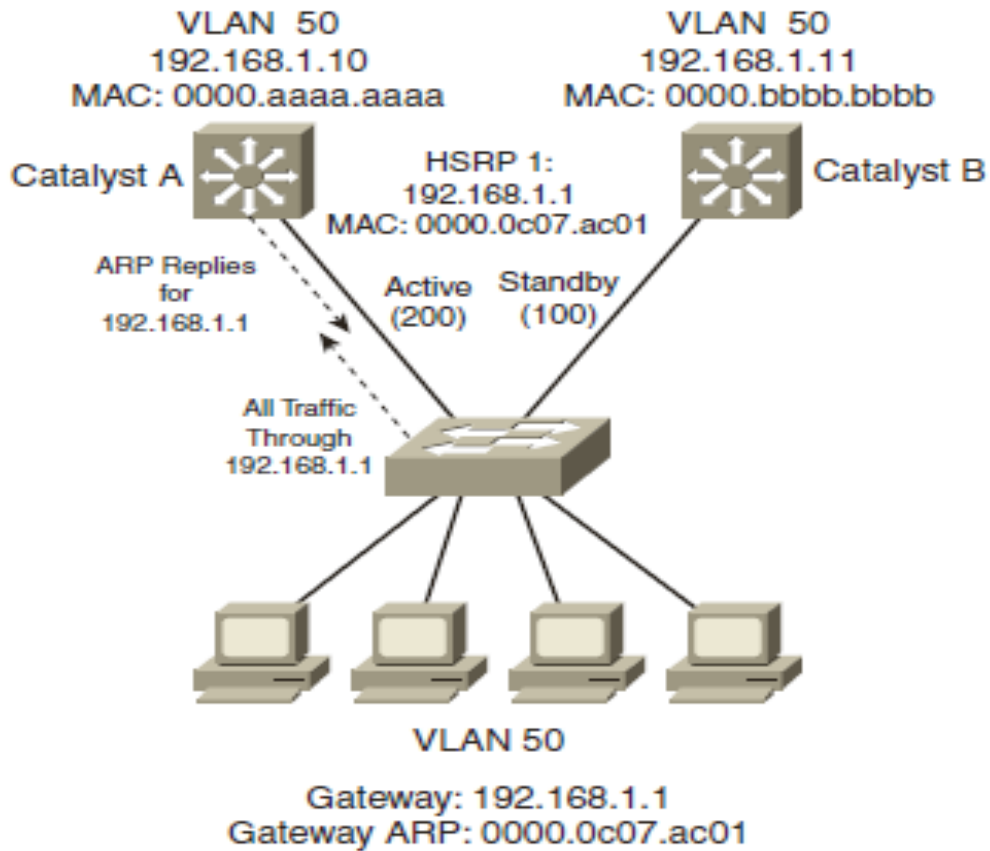
```
SW(config-if)# standby group authentication md5 key-chain chain-name
```

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 arayüzünü 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
```