İÇİNDEKİLER

INTERNETWORKING TEMELLERI

Network Nedir, Ne İşe Yarar?

Birden fazla bilgisayarın çeşitli sebeplerden dolayı birbirlerine bağlandığı yapıya network (ağ) denir.

Bir çok bilgisayarın aynı yapı içerişinde bulup birbirleriyle haberleşebiliyor olması çok ciddi yararlar sağlar. Bilgi paylaşımı, yazılım ve donanım paylaşımı, merkezi yönetim ve destek kolaylığı gibi konular göz önüne alındığında birden fazla bilgisayarın bulunduğu ortamlarda artık bir network kurulması zorunlu hale gelmiştir diyebiliriz.

Networklerin kurulmasıyla birlikte disketle data taşıma devri bitmiş, tek tuşla istenilen bilgiye ulaşma kolaylığı meydana gelmiştir. Bir veya birkaç yazıcı ile bir işletmenin bütün print ihtiyaçları da yine network sayesinde karşılanabilmektedir.

Yönetim ve destek hizmetleri kolaylaşmış, network yöneticisi tek bir bilgisayardan çok daha hızlı bir şekilde bütün networkü izleyebilir ve sorunları çözebilir hale gelmiştir.

Bilgisayarlar networklerde çeşitli görevler üstlenebilirler. Genel olarak bir bilgisayar bir networkte client (istemci) yada server (sunucu) rollerinden birini üstlenir.

Network ortamında paylaşılan yazılım ve donanımlara sahip bir bilgisayara server yada Ana Bilgisayar denir. Burada Server sahip olduğu kaynakları istemci bilgisayarların kullanıma açarken bazen de tüm verinin toplandığı ana merkez konumundadır.

Network ortamında kaynak yada veri isteyen bilgisayarlara ise Client adı verilir. Client sadece kendisinden donanımsal olarak büyük olan Server lardan değil gerektiğinde diğer client' lardan da kaynak yada veri talebinde bulunabilir.

Bilgisayarlar birbirleriyle protokoller sayesinde iletişim kurarlar. Protokoller, izlenmesi gereken iletisim kuralları koyarlar. İnsanlar için diller ne ise, bilgisayarlar için de protokoller o demektir. Haberlesme protokollerine örnek olarak **TCP/IP, NetBEUI, IPX/SPX** gibi protokolleri gösterebiliriz.

NETWORK TİPLERİ

Networkler Peer To Peer ve Client/Server mimarisi başlıkları altında incelenebilirler. Peer To Peer networklerde ana bir bilgisayar yoktur.

Ağ servisleri ağdaki PC'lerin network alt yapısından talep edebileceği isteklere bağlı olarak geliştirilmiştir. Bu olanakları düzenleyen özel bir PC vardır. Buna server denir. Sadece bu olanaklardan faydalanan PC'lerede client denir.

- Server sadece servis sağlar
- Client sadece servis ister
- Peer her iki işi de bir arada yapar

Single Server Ağlar:

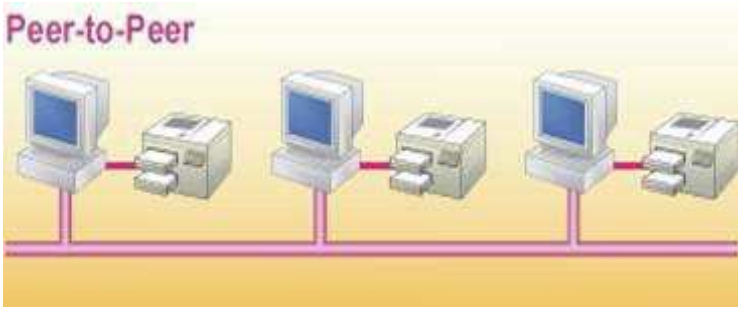
Keskin biçimde belirlenmiş rolleri yerine getirirler. Bir servis sağlayıcı (server) ve servis isteyen (client)'lardan meydana gelir. 10-50 kullanıcılı networklerdir. Network ve data paylaşım güvenliği çok kolaydır. Çünkü tüm ağ yönetimi tek bir PC tarafından yapılmaktadır. Genelde 10-50 kullanıcılı networklerde Distributed Component Object Model çalışma mantığı vardır. Yani client'lar kendi işlemlerini kendileri yaparlar, server'ı ise sadece print ve dosya depolama merkezi olarak kullanırlar.

Client / Server mimarisinde iş, adından da anlaşılacağı gibi hem donanımsal hem de yazılımsal olarak diğer bilgisayarlardan üstün, atanmış bir ana bilgisayar vardır.

Burada Server olarak atanmış bilgisayarın yetersiz kaldığı durumlarda networke başka Serverlarda dahil edilebilir. Örneğin gelişmiş bir networkte Mail Server'ın, DHCP ve DNS gibi Serverların farklı bilgisayarlarda bulunması performansı olumlu yönde etkileyebileceği için önerilebilir.

Peer-to-Peer Ağlar:

Tüm birimler servis istediğinde bulunma hakkına sahiptir. Yani networkteki herhangi bir bilgisayar farklı zaman dilimlerinde hem client hem de server olabilir. Bütün birimler yönetim mekanizması yönünden birbirine benzer. Yönetim ve data paylaşımı merkezi değildir . Kullanıcıların PC bilgisinin iyi olması gerekir. Çünkü her bilgisayar sadece kullanıcısı tarafından yönetilebilir. Genelde IPX/SPX veya NetBEUI protokolleri kullanılmaktadır. 2-10 kullanıcılı netwrokler için tavsiye edilir



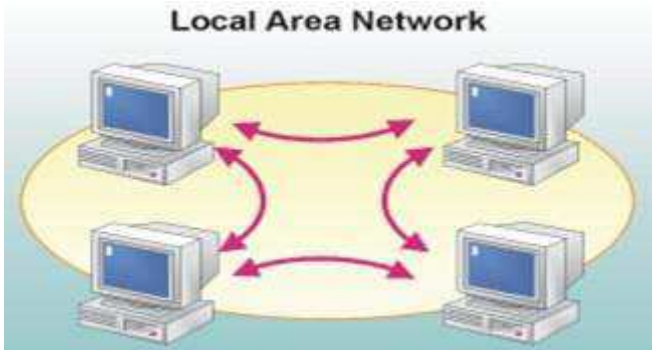
server temelli (domain)

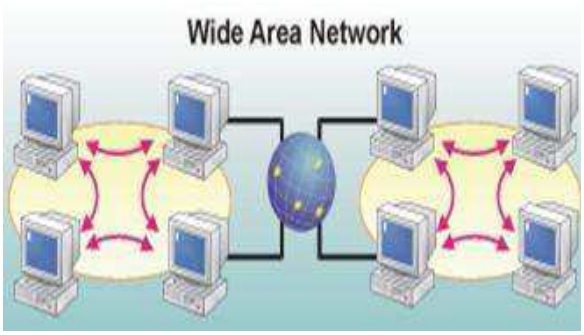
workgroup(eşler arası ağ)

İPUCU: Windows ortamında Peer-to-Peer (eşler arası ağlar) workgroup olarak, server temelli olan ağlar ise domain olarak bilinir.

LAN, WAN ve MAN

Ağlar büyüklüklerine göre LAN (Local Area Network), WAN (Wide Area Network) ve MAN (Metropolitan Area Network) olmak üzere üçe ayrılırlar.

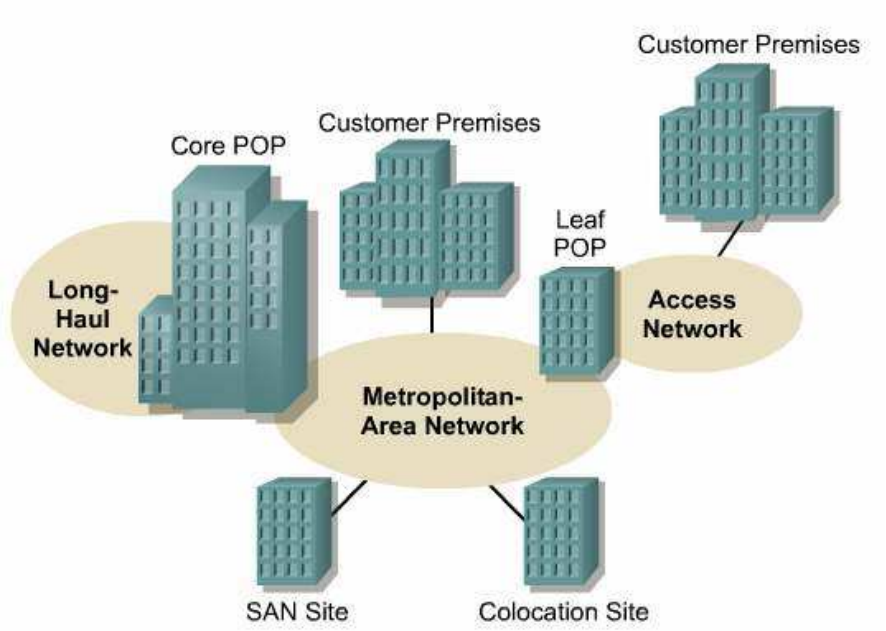




Birbirine yakın yerlerde konumlandırılmış ve kablolar ile fiziksel olarak birbirlerine bağlanmış yapıdaki networkler LAN olarak adlandırılırlar. Örneğin bir binada bulunan bütün bilgisayarların birbirlerine bağlanmasıyla oluşan yapı bir Local Area Network'tür.

İki yada daha fazla LAN' ın birbirlerine telefon hatları, kiralık hatlar yada benzer yollardan birbirlerine bağlanmasıyla oluşan yapı ise Wide Area Network olarak adlandırılır. Burada bilgisayarların fiziksel olarak birbirlerine yakın olmalarına gerek olmadığı gibi çok uzakta olabilirler.

MAN ise, kavram olarak açıklaması zor olmakla birlikte, örneğin bir şehir yada bir bölgenin iki ayrı LAN ile birleşmesi gibi düşünülebilir.



Örnek vermek gerekirse birden fazla subesi bulunan bir bankanın kullanabileceği bir yapı diyebiliriz.

NETWORK TEKNOLOJİLERİ

Bahsedeceğimiz teknolojiler şunlar:

_ Ethernet

_ Token Ring

_ ATM (Asynchronous Transfer Mode – Asenkron Veri Aktarımı Modu)

_ FDDI (Fiber Distributed Data Interface – Fiber Üzerinde Veri Aktarım Standardı)

_ Frame Relay

Bütün bu teknolojilere taşıyıcı protokoller diyebiliriz. Çünkü bir de, bilgisayarların birbirleriyle konuşmalarının dil kurallarını koyan iletişim protokolleri vardır. TCP/IP, NetBEUI gibi protokolleri iletişim protokolü diye adlandırabiliriz. Bir insanın bir şehirden bir şehre gitmesi için nasıl, otobüs, otomobil, uçak, gemi vb. gibi teknolojiler varsa, network'lerdeki veri paketlerinin de bir bilgisayardan bir bilgisayara gitmesi için Ethernet vb. gibi teknolojiler vardır. Bu bağlamda TCP/IP, NetBEUI protokollerinin kuralları ile oluşturulan veri paketleri, taşıyıcı protokollerinin veri paketlerinin içlerine bindirilerek network üzerinde yolculuklarına çıkarlar. Bu olaya encapsulation denir.

ETHERNET

Ethernet yada diğer ağ teknolojilerinin anlattığı şey, verinin bir yerden bir başka yere nasıl ve hangi kurallarla gittiğini belirlemektir. Bu kurallara uygun cihazlar ve bu cihazları kontrol edebilen işletim sistemleri, network'ü oluşturan diğer öğelerdir.

Bir ağ teknolojisini diğer bir ağ teknolojisinden ayıran temel özellik; bu teknolojinin veriyi kablolar üzerinde nasıl ve hangi yolla aktardığıdır. Örneğin, ATM, veriyi bir yerden bir başka yere gönderirken, verinin bozulmaması için her ATM cihazında önlemler alan kurallara sahiptir. Bu sebeple iki şehir arasında yer alacak bir WAN'ın kurulması için ATM uygun bir metottur. Ama bir ofis için bir ATM network'ü kurmak inanılmaz maliyetler getirir. Bunun yerine Ethernet'in sağladığı teknolojilere dayanan cihazlarla ofis içi ağı kurmak daha mantıklı olacaktır.

Çoğumuzun, ofis yada okulda gördüğü network'lerde Ethernet teknolojisine dayalı olan cihazlar kullanılır. Bu bölümün basında anlattığımız network kartları ve kabloların hemen hemen hepsi Ethernet'in bir parçasıdır. Örneğin STP yada UTP gibi kavramlar yine yıllar içerişinde Ethernet teknolojilerinden doğmuş standartlardır. Bir ATM yada FDDI teknolojileri için ise daha farklı kablolama ve daha farklı cihazların bilgisayarlara bağlanması gerekmektedir. Örneğin bir telefon santrali yada bankalarda para çekmek için kullandığınız makinelerde ATM yada Frame Relay gibi teknolojilerin birer parçası olan ağ kartları yada kablo standartları kullanılmaktadır.

TOKEN RİNG

Token ring teknolojisi, bir ring topolojisi içinde uygulanmaktadır. Aslında, fiziksel olarak kullanılan topoloji bir star topoloji olmasına rağmen, merkezdeki cihazın yapısı nedeniyle, veriler sanki kapalı bir zincir üzerinde dolaşıyormuş gibi hareket eder. Bir veri her bilgisayara teker teker gönderilir. Ve veri merkezi birimden gereksiz yere birden fazla kez geçmiş olur. Token ring şeklindeki bir yapının kullanılması sebebi altında, verilerin çok az şekilde kayba uğraması, yatar. Oldukça önemli verilerin taşınması için düşünülmüş bir teknolojidir. Örneğin sağlık uygulamaları yada eski Internet alt yapısı için düşünülmüş olsa da yavaşlığı nedeniyle çok az uygulama alanı bulmuştur.

Bu network'te veriler bir paket halinde her bilgisayara sıra ile merkezi birim tarafından gönderilirler. Merkezde bir hub yada switch gibi bir cihaz olmasına rağmen bu yapı bir star topoloji gibi çalışmaz. Zira star topolojilerde, bir bilgisayardan bir başka bilgisayara gidecek olan veri en kısa biçimde merkezdeki cihaz üzerinden geçip aktarılır.

Token ring teknolojisinde ise, veriler merkezi birim tarafından, ona bağlı bilgisayarlara teker teker gönderilir. Bu sırada her makine paketin kendisine ait olup olmadığını kontrol eder. Bu sebeple merkezde yer alan hub yada switch benzeri cihaza, MSAU (MultiStation Access Unit) adı verilmektedir.

Zira verilerin her bilgisayara teker teker sıra ile gönderiliyor olması; network'de sanki bir veri paketinin bir zincir etrafından dolaşıyormuşçasına döndüğü izlenimini verir. Dolasan bu veri paketine Token ismi ve bu paketin tüm network'de dolaşarak veriyi dağıtma isine de Token Passing ismi verilmektedir. Token ring teknolojisi 4 ile 16 Mbps'lik bir veri aktarım hızına ulaşabilmektedir.

ATM

ATM (Asyncronous Transfer Mode – Asenkron Veri Aktarımı Modu), 1988 yılında her tür veriyi, telefon hatları, ses, TV sinyali, network üzerinde gidip gelen veriler gibi, her tür veri sinyalini taşıması için oluşturulmuş bir standarttır. ATM, paket anahtarlama yapan bir teknolojidir. Paket anahtarlama (packet switching) birden fazla parçası olan bir network'de

verinin parçalara bölünüp, teker teker ayrı yollardan gönderilmesi ve ulaştığı yerde tekrar birleştirilmesi esasına dayanır. Kabaca bir örnek vermek gerekirse, 1 MB'lık bir verinin bir kısmı A yolundan diğer bir kısmı da B yolundan gidebilir. Bu teknolojiye packet switching adını veriyoruz ATM, packet switching prensibini kullanan ve verileri eşit parçalara bölerek gönderen bir teknolojidir. 1988 yılında CCITT tarafından bulunmuş ve son derece modern prensiplerle çalışan bir teknoloji olarak kabul edilebilir. ATM aradan geçen 14 yıla rağmen halen WAN'larda kullanılan temel veri aktarım teknolojilerinden biridir. ATM teknolojisi 155 Mbps ile 622 Mbps arasında bir veri aktarım hızına ulaşmaktadır.

FDDI

1986 yılından ANSI X3T9.5 komitesi tarafından tanıtılmış bir teknolojidir. FDDI, 100 Mbps'nin üzerindeki hızlarda veri aktarmak için, fiber optik kabloların kullanıldığı bir yapıyı oluşturmaktadır. FDDI, 1986 yılında ilk bulunduğunda yüksek kapasiteli bilgisayarlar için, o günlerde var olan 10 Mpbs'lik Ethernet ve 4 Mbps'lik Token Ring teknolojilerine bir alternatif olarak sunulmuştur.

FDDI, prensip olarak iki kapalı zincir üzerinde ters yönde hareket eden veri trafiğine göre yapılandırılmıştır. Bu kapalı hat yada zincir tabir edilen yapılardan biri bos olarak hazırda tutulur. Veri taşıyan zincirde bir problem olduğunda ikinci zincir devreye girer ve veriyi ters yönde taşımaya baslar. FDDI'da da Token ring teknolojisinde olduğu gibi Token isimli veri paketleri kullanılır. Paket yapıları birbirinden farklı olsa da veri bir zincir etrafında dolaştırılarak taşınır ve tıpkı Token ring'deki gibi her bilgisayardan bir kez geçer. FDDI, son derece yüksek bir güvenilirliğe ve veri aktarım hızına sahiptir. Günümüze kadar güncellenen standartlar sayesinde, veri transfer hızı, 155 ile 622 Mbps arasında tanımlanabilir hale geldi. Bu sebeple veri taşımak için ATM ile birlikte son derece büyük bir öneme sahiptir. ATM kadar esnek bir yapıya sahip değildir. Zira ATM telefon hatları yada TV sinyalleri gibi verileri de taşıyabilmektedir.

FRAME RELAY

Frame Relay, verileri packet switching prensibi ile taşıyan ve değişken veri paketlerinin kullanıldığı bir veri aktarım teknolojisidir. Veriler bir network'de bir noktada bir başka noktaya hedeflenerek gönderilirler. Ama bu verilerin network üzerinde gittiği yol, ulaştığı nokta tarafından bilinmez. Frame Relay, bu sayede, network'deki trafiğin kolayca gözlemlenebilmesini mümkün kılar.

Frame Relay bu sayede onu kullanan abone yada mensuplarından, kullanıldığı ölçüde para ödenmesini sağlar. Örneğin Türk Telekom'dan Türkiye içinde yer alan Frame Relay ağında bir hat kiralarsanız, gönderdiğiniz veri kadar para ödersiniz.

Frame Relay'in diğer bir esnekliği ise, veri paketlerinin istenildiğinde hep aynı yoldan gönderilebiliyor olmasıdır. Bu sayede, network'deki trafik istenildiği gibi düzenlenebilir.

ETHERNET TEKNOLOJİLERİ

Bilgisayarların bir networke bağlanıp veri alış verişinde bulunabilmesini sağlayan elektronik devredir. Farklı yerlerde Ethernet kartı, network kartı, ağ kartı yada NIC şeklinde isimlendirmeleri yapılmıştır. NIC, İngilizce Network Interface Card' ın kısaltmasıdır.

Bir network kartı, bilgisayarınızın genişleme yuvalarından (slot) birine takılır. Bu bir dizüstü bilgisayarda PCMCIA yuvası ya da bir masaüstü bilgisayarda bir PCI yuvası olabilir. USB ya da benzeri arabirimlere de takılan pek çok network adaptörü vardır.

Bir bilgisayar üzerindeki, Windows'a benzer bir isletim sistemi, aktarılması gereken bilgileri, üzerinde tasıdıgı herhangi bir network adaptörüne aktardıgında, network adaptörünün ilk yaptıgı is bu verileri, veri paketlerine bölmektir. Veriler parçalara bölünür. Daha sonra bu verilerin basına ag üzerinde hangi noktaya ulasacagına ve nasıl tasınacagına dair veriler eklenir.

Paketlerin bası ve sonuna "bu paketin basıdır" ve "bu veri paketinin sonudur" gibi etiketler eklenir. Bu islem her bölünmüs veri paketi için yapılır ve sonuç olarak, bu veri paketleri sadece 1'ler ve 0'lardan olusmaktadır. Veri paketleri, birbiri ardına yazılmıs, milyonlarca 1 ve 0 içinde hayali olarak yaratılmıs bölümlerdir.

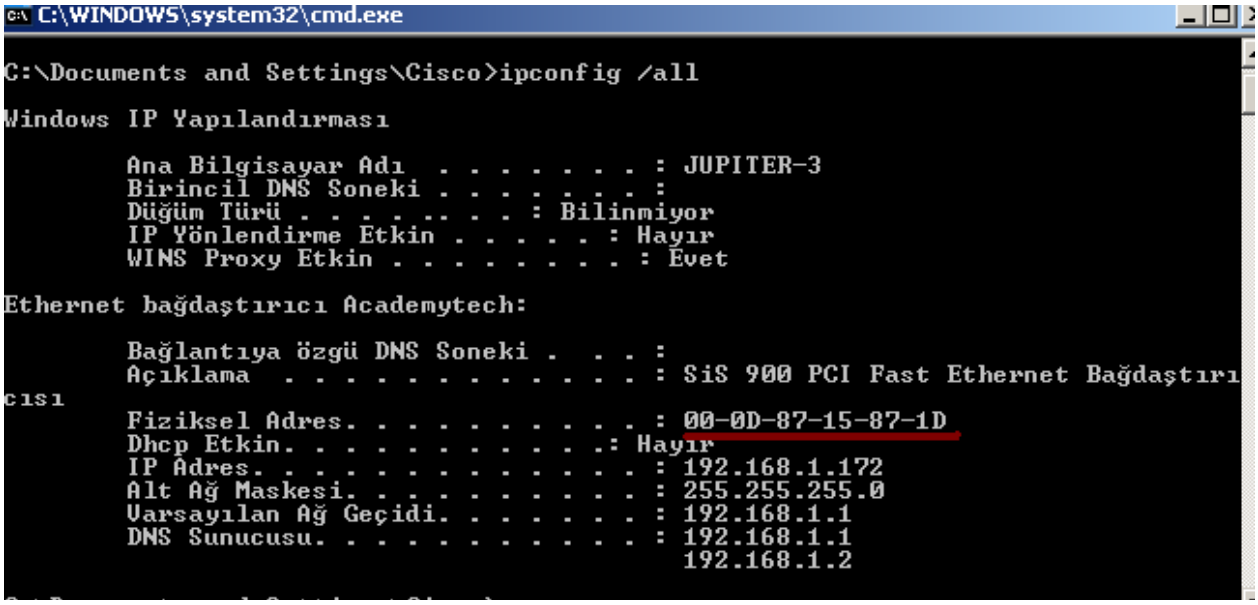
Her paketin başında **header** isimli ve bir önceki paragrafta belirttiğimize yakın bilgiler taşıyan bir kısım yer alır ve her header içinde de o veri paketinin nereye gideceği ve nereden geldiğine dair veriler yer alır. Bir NIC'ye yani network adaptörüne, onun almaması gereken milyonlarca veri paketi ulaşır. Network adaptörleri, bu paketlerin başındaki header kısmına bakarak, o paketin kendisi için gönderilmiş olup olmadığını anlarlar. Eğer paketin başında yer alan header kısmında, kendi adresi varsa, o paketi alır ve bilgisayarda değerlendirmek üzere işleme sokar.

Her Ethernet kartının üretimden itibaren kendine ait farklı bir tanımlama numarası vardır ve bu sayede diğer bütün kartlardan ayırt edilebilir. Bu tanımlama numarasına MAC adresi (Media Access Control) yada Fiziksel Adres denir ve 6 oktet, 48 bitten oluşur. Bu 6 oktetin ilk 3 okteti Internet Assigned Numbers Authority (IANA) tarafından belirlenir. Bir firma Ethernet Kartı üretmeye karar verirse ilk başvuracağı yer IANA' dır. IANA firmaya o firmanın ID' si gibi düşünebilecek 3 oktetli bir sayı verir son okteti de firmaya bırakır. Bu şekilde bir standart sağlanırken aynı MAC adresine sahip Ethernet kartlarının üretilmesi de engellenmiş olur.

Bir network adaptörü su işleri yapar:

1. Ona gönderilen işlenmiş ve gönderilmek için hazır veriyi, veri paketlerine çevirir.
2. işletim sisteminin istediği yere, bu veri paketlerinin gitmesi için veri paketlerine header'lar ekler.
3. Veri paketlerini alır ve gönderir. Ve bunların doğruluğunu kontrol eder.
4. Bir network adaptörü, ona gelen veri paketi, bozuksa aynı paketin karşı bilgisayarın network adaptörü tarafından tekrar gönderilmesi için bir komut içeren veri paketi hazırlayıp gönderir.
5. Network adaptörleri buna benzer pek çok is yaparlar. Bu komutlar ve fonksiyonların hepsi, birer protokolle, nasıl yapılacağı belirlenmiştir. Bu gibi, network'deki işlemlerin en alt seviyesinde yer alan işlemlerle, çoğu zaman işletim sistemleri ilgilenmezler. Bu gibi işlemler doğrudan network adaptörü tarafından yürütülür.

Bir bilgisayarın MAC adresi komut satırında **"ipconfig /all"** yazılara öğrenilebilir. (Windows 9x ortamında ipconfig.exe yerine **Winipcfg.exe** kullanılır.)



Bir sistem yöneticisi kendi bilgisayarından diğer bilgisayarların MAC adreslerini de öğrenmek isteyebilir. Sözgelimi DHCP ile ip konfigürasyonunu dağıtmak ve bazı bilgisayarların her seferinde aynı ip'yi almasını sağlamak için MAC adreslerini kullanarak DHCP' nin Reservations özelliğini kullanmak isteyen bir yönetiçinin her bilgisayarı tek tek dolaşarak ipconfig /all komutunu kullanması ve MAC adreslerini not alması çok uzun ve yorucu olur.

Bu durumda sistem yöneticilerinin yardımına ARP (Address Resolution Protocol) protokolü yetişir. Bir bilgisayara en az bir kere ulaşmış olmak kaydıyla, komut satırında **"arp –a"** yazılarak o bilgisayarın MAC adresi öğrenilebilir.

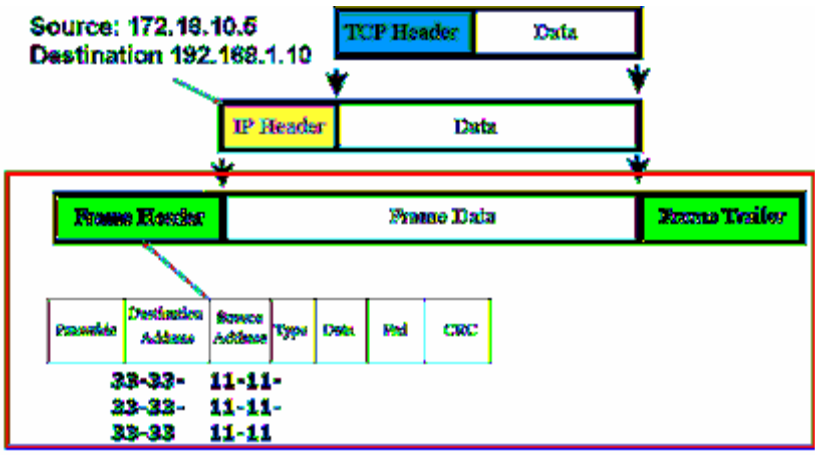
Komut satırından alınmış şekilde, önce 192.168.1.4 bilgisayarına bir kez ulaşmak için ping atılmış. Daha sonra kullanılan "arp- a" komutu ile oturum boyunca ulaşılan tüm bilgisayarların bellekteki fiziksel adres tablosuna erişilmiştir.

(ARP Protokolü ileride detaylı olarak anlatılacaktır.)

Ethernet teknolojiler IEEE 802.3 standardı ile tanılanmıştır ve ethernetin datayı framle' ler halinde taşıdığı söylenebilir. Genel olarak ethernet frame' leri aşağıdaki gibidir.

ETHERNET					
8	6	6	2	46-1500	4
Preamble	Destination Address MAC	Source Address MAC	Type	Data	FCS

Destination ve Source adresler hedef ve kaynak cihazların fiziksel adreslerini ifade eder. FCS değeri ise datanın sağlıklı iletilip iletilmediğinin kontrol edilmesini sağlayan bir değerdir.



OSI Referans modeli içerişinde detaylı anlaşılabilecek datanın iletimi sırasin data katmanlar ilerler. Ve her katmanda data üzerine o katmanın çalışma mantığı içerişinde gereken bilgi etiketlenir. Şekilde şimdilik önemli olan kısmın sırasıyla TCP Header ve IP Header eklenir. LAN içerişinde data 2. katmanda, yani fiziksel adresler yardımıyla haberleşecektir. Frame Header ile bu bilgiler dataya eklenir.

IP Header ve Frame Header arasındaki en önemli fark frame' lerin TTL (Time To Live) değerine sahip olmamasıdır. Dolayısı ile ikinci katmanda oluşabilecek bir döngü döngüyü yasayan cihazlar kapatılmadığı sürece devam edecektir.

CSMA/CD

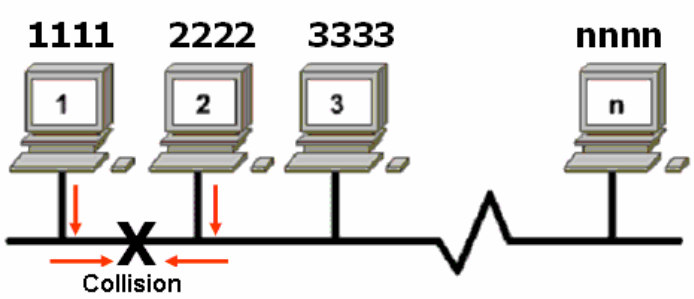
Ethernet networkleri belli bir anda kabloyu hangi bilgisayarın kullanacağını CSMA (Carrier Sense, Multiple Access/Collision Detection) tekniğiyle belirler. Bu teknikte paket gönderilmeden önce kablo kontrol edilir. Diğer bir iletişimin oluşturduğu trafik yoksa iletişime izin verilir.

İki bilgisayarın birden kabloyu kullanmaya çalışması collision olarak adlandırılır. Her ikisinin de trafiği kaybolur. Ve hattın boş olduğu anı yakalamak için yeniden beklemeye ve hat dinlenmeye başlanır.

CS (Carrier Sense): Bir network kartı tek bir kablo üzerinde gidip gelen veriler varsa o hatta veri göndereceği zaman kablo üzerinde veri taşınıp taşınmadığını dinler ve eğer kabloda bir veri trafiği yoksa veri paketlerini yol çıkarır. Bu durum, tek bir kabloya birden çok adaptörün bağlı olduğu eski tip network teknolojileri için geliştirilmiştir. Tıpkı trafiğe çıkmak isteyen bir arabanın yolun bos olup olmadığını kontrol ettikten sonra, yola çıkması gibi, network adaptörleri de CS teknolojisi sayesinde, kablo bos olduğu takdirde veri gönderirler.

MA (Multiple Access): Bir network kartına aynı anda birden çok veri paketi gönderilmişse, o takdirde o adaptör bunlardan hiçbirini almaz. Network'ün tümüne, "bana gönderdiğiniz veri paketleri aynı anda yola çıktığından dolayı tekrar gönderilmelidir" anlamına gelen bir mesajı broadcast eder. Bu mesajın alınmasından itibaren, karsı network adaptörleri aynı veri paketini tekrar gönderir.

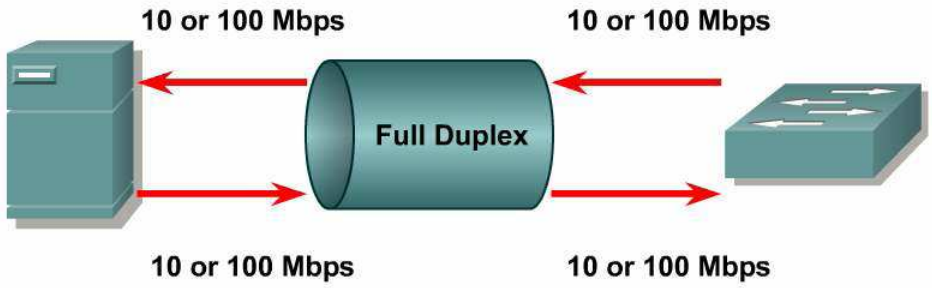
CD (Collision Detection): Eğer karşılıklı iki network adaptörü, aynı anda iki veri paketini birbirine gönderirlerse, veri paketleri çarpışacaktır. Bu durumu tespit eden adaptörler, rasgele bir zaman dilimi kadar bir süre (milisaniyeler mertebesinde) tekrar aynı paketi network'e çıkarırlar. Bu sayede, çarpışan paket sinyallerinin, kablolar üzerinde yitirilip gitmesi engellenmiş olur. Bu çarpışmayı, tek şeritli ve sadece tek yön çalışan bir yolda, karşılıklı iki arabanın aynı anda yol almak istemesinden doğan kazaya benzetebiliriz.



CSMA/CD networklerinde, beklemelerin çoğalmaması için bus olarak tanımlanan kablonun iki ucunun sonlandırılması gerekir.

FULL DUPLEX

Full Duplex ethernet aynı anda hem data iletimini hem de data alınmasını sağlar.



IEEE 802.3x ile tanımlanmış Full Duplex çalışma içerişinde, her iki cihaz da Full Duplex modada olduğu zaman sonuç alınabilecektir. Örneğin, host to switch, switch to switch yada switch to router bağlantılarında her iki tarafta Full Duplex modda çalışabiliyor olmalıdır.

Half Duplex çalışma içerişinde collisionlardan dolayı bant genişliğini yüzde 50 yada altmışı kullanılabilirken Full Duplex çalışma da bant genişliğinin tamamı kullanılabilir. Sözgelimi 10 Mbps olan bir hattın Half Duplex ile 5-6 Mbps' i kullanılabilir, oysa Full Duplex çalışma da ideal olarak aynı bant genişliğinde 20 Mbps'e ulaşıldığı varsayılır.

Full Duplex içerişinde CSMA/ CD ' den bahsedilemez. Çünkü aynı anda hem iletim hem alım olabileceği için Carrier Sense olmayacaktır.

KABLO STANDARTLARI

Bilgisayarlar ethernet kartlarına takılan kablo aracılığıyla birbirine bağlanırlar. Networkün yapısına göre farklı özelliklerde kullanabilecek bir çok çeşit kablo standardı vardır. Ana başlıkları söyle sıralayabiliriz;

-Koaksiyel (Coaxial)

-Twisted-Pair

-UTP (Unshielded Twisted-Pair / Koruyucusuz Dolanmış-Çift)

-STP (Shielded Twisted-Pair / Koruyuculu Dolanmış-Çift)

-Fiber-Optik

KOAKSİYEL KABLOLAR

Koaksiyel kablolar yaygın olarak kullanılan ağ kablolarıdır. Çok tercih edilmesi ve çok sık kullanılmasının baslıca nedenleri uygun fiyatı, hafifliği, esnekliği ve kolay kullanılmasıdır. Bir koaksiyel kablo bir iletken metal telin önce plastik bir koruyucu ile, ardından bir metal örgü ve dış bir kaplamadan oluşur. Bu koruma katları sayesinde iletilen verinin dış etkenlerden etkilenmesi minimuma indirgenmeye çalışılmıştır.

Koaksiyel kablonun içinde kullanılan tek genellikle bakırdır.

Koaksiyel kablonun iki tipi vardır:

-Thin (thinnet) İnce

-Thick (thicknet) Kalın

TİP	EMPADANS	KULLANIM
RG-8	50 Ohm	10BASE5
RG-58	50 Ohm	10BASE2
RG-75	75 Ohm	Kablo TV

İNCE (10Base2) KABLOLAR

10Base2 olarak da adlandırılan bu kablo RG-58/V tipinde bir koaksiyel kablodur. İnce Ethernet kablolamada bilgisayar bağlantıları T şeklindeki konnektörlerle yapılır. Bunlara BNC konnektör denir. 10Base 2 olarak isimlendirmede 10: veri iletim hızının saniyede 10 Megabit olduğunu, Base: basebant bir sistem olduğunu yani herhangi bir anda kabloda sadece bir sinyalin iletilebileceğini, 2: maksimum kablo boyunun yaklaşık 200 metre olduğunu gösterir

Thinnet (ince) koaksiyel kablo 0.25 inç genişliğindedir. Yaygın olarak kullanılır. Verileri sağlıklı olarak 185 metre uzağa iletebilirler. Thinnet koaksiyel kablolar RG-58 standardı olarak değişik biçimde üretilmektedir.

Bu Tür Ethernet Kablolamada Şu Kurallar Uygulanır

1. Ethernet kartı harici değil dahili transceiver kullanacak şekilde ayarlanmalıdır.

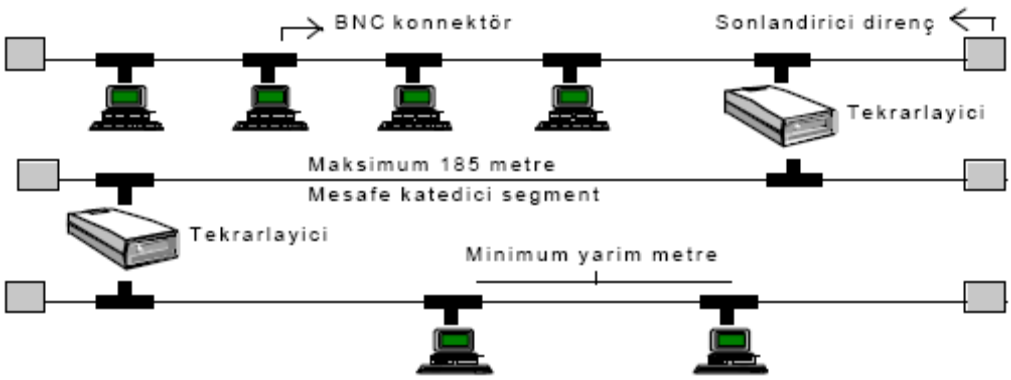
2. Farklı Ethernet segmentlerini bağlamak için en fazla 4 tekrarlayıcı kullanılabilir. Dolayısıyla birbirlerine tekrarlayıcıyla bağlanan en fazla 5 segmentli bir ağ oluşturulabilir. Bu 5 segmentin sadece 3'üne bilgisayar bağlanabilir, diğer 2 segment mesafe katetmek içindir.

3. Bir segmentin boyu en fazla 185 metredir ve toplam ağ trunk'u (sinyalin iletilebileceği maxsimum kablo boyu) en fazla 925 metredir.

4. Makinelerın bağlantıları arasında en az yarım metre mesafe bırakılmalıdır.

5. Bir trunk segmentteki en fazla node (bilgisayar yada tekrarlayıcı gibi aygıtlar) sayısı 30 dur.

6. Trunk segmentlerin her iki tarıfı 50 ohm dirençlerle sonlandırılmalıdır.



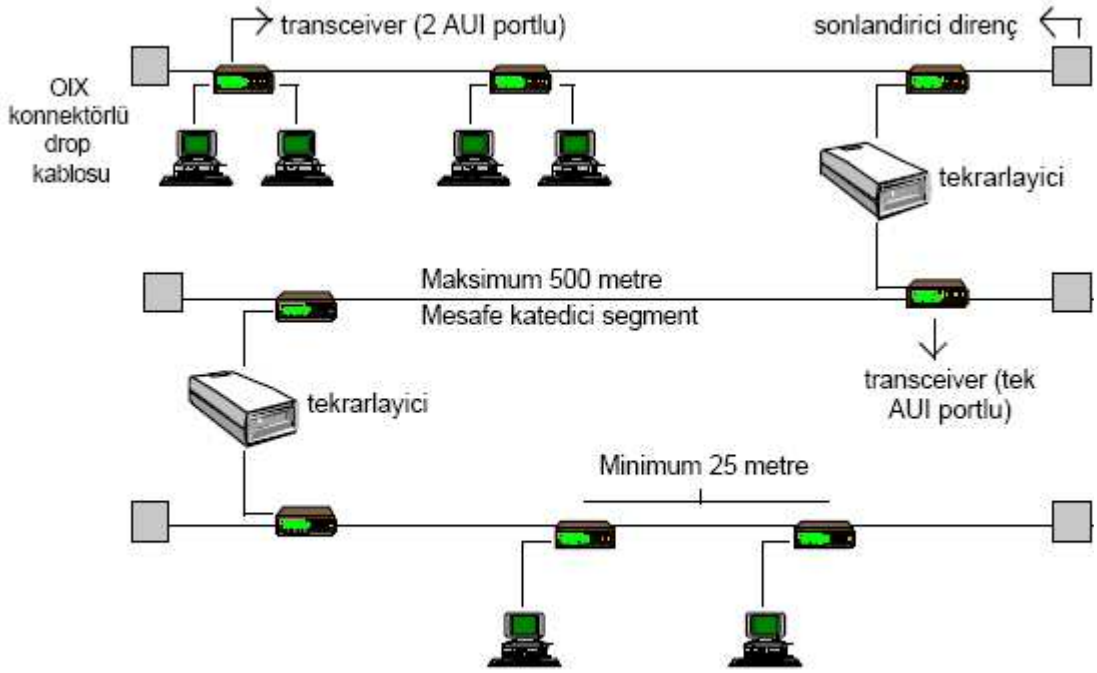
Resim 3. 5- Ince Ethernet Agi

KALIN (10Base5) ETHERNET KABLOSU

Genellikle Ethernet backbone (diğer ağlar arasında bağlantı görevi gören omurga) olarak kullanılan bu kablo da yine koaksiyel bir kablodur. RG-8 yada RG-11 tipinde koaksiyel kablolar olabilir. 10Base5 olarakta bilinir yani saniyede 10 Megabitlik veri iletim hızlı, baseband ve maksimum 500 metre uzunluğu destekler. Bilgisayarlar ince ethernette olduğu gibi doğrudan bu kabloya bağlanmaz. Kablo üzerinde transceiver denilen bir aygıt takılır; bir ucu bu aygıta bağlanan bir ara kablonun diğer ucu da bilgisayarın Ethernet kartına takılarak bağlantı gerçekleştirilir. Bu ara kablo drop kablosu olarak isimlendirilir. Bu kablonun iki ucunda DIX konnektörler yer alır ve tarnsceiver ile Ethernet kartının AUI portuna bağlanır.

Kalın Ethernet Kablosunun Kuralları

1. Ethernet kartı harici tarsceiver kullanılacak şekilde ayarlanır.
2. Drop kablosunun maksimum uzunluğu 50 metredir.
3. Farklı Ethernet segmentlerini bağlamak için en fazla 4 tekrarlayıcı kullanılabilir. Dolayısıyla birbirlerine tekrarlayıcıyla bağlanan en fazla 5 trunk segmentli bir ağ oluşturulabilir. Bu 5 segmentin sadece 3'üne bilgisayar bağlanabilir, diğer 2 segment mesafe katetmek içindir.
4. Bir trunk segment en fazla 500 metre olabilir, dolayısıyla bir ağ trunk'u 2,5 km yi geçemez.
5. Kabloya takılan transceiverler arasında en az 2,5 metre mesafe olmalıdır.
6. Bir trunk segmentte yer alabilecek en fazla node sayısı 100 dür.
7. Trunk segmentlerin uçları 50 ohm'luk dirençlerle sonlandırılmalıdır.



Resim 3. 6- Kalın Ethernet Agı

Mesafe	Koaksiyel kablo
185 m	Thinnet
500 m	Thicknet

10Base2

10Base5

Bir thinnet koaksiyel kabloyu thicknet kabloya bağlamak için ise transceiver denilen ara birim kullanılır. Transceiver'ın network adaptörüne bağlanması için AUI yada DIX adı verilen çıkış kullanılır. AUI (Attachment Unit Interface) anlamındadır. DIX (Diğital Intel Xerox) anlamına gelir.

AUI

Koaksiyel kabloların network adaptörüne bağlanması için, ayrıca iki kablonun birbirine eklenmesi BNC Konektörleri kullanılır.

BNC kablo konnektörü kablonun ucunda yer alır. T konnektör ise koaksiyel kabloyu network adaptörüne bağlamak için kullanılır. Barrel konnektör ise iki koaksiyel kablonun birbirine bağlanmasını sağlar. Sonlandırıcılar ise kablonun sonunda yer alırlar.

BNC T Konnektörü

Bus yerleşim biçiminde kurulan network'lerde kullanılan koaksiyel kablonun iki ucunda sonlandırıcı kullanılır. Bu sonlandırıcılar kablonun sonuna gelen sinyali yok ederler.

TWİSTED-PAİR KABLOLAR

LAN'larda ve sınırlı veri iletiminde kullanılan bir diğer kablolama türü de twisted-pair kablolardır. Twisted-Pair (Dolanmış-çift) kablo iki telden oluşan bir kablodur. Twistedpair kablolar iki türdür:

-UTP (Unshielded Twisted-Pair)

-STP (Shielded Twisted-Pair)

UTP Kablo

STP Kablo

10BaseT network'lerde ve diğer LAN ortamlarında yaygın olarak UTP kablolar kullanılır. Kablo üzerinde düşük voltajda DC iletişim yapılır. Genelde ismini hep duyduğumuz UTP (Unshielded Twisted Pair) kablolama çeşiti kullanılır. UTP maximum kablolama uzunluğu 100 metredir. RJ45 Connector kablonun iki ucundada kullanılır. Çevre etkenlerine karşı çok duyarlıdır. UTP'den daha sağlam olan Shielded Twisted Pair de UTP gibi maximum 100 metre uzunluğunda kullanılabilir.

UTP kablo iki izoleli bakır kablodan oluşur. UTP kablolar ayrıca telefon sistemlerinde de kullanılır. 10BaseT kablolar RJ-45 sonlandırıcıları ile sonlandırılırlar.

UTP bir kablonun içinde 8 ince kablo yer alır. Normal bir bağlantıda bu şekiz ince tel her iki uçta aynı olacak şekilde bağlanmalıdır. Crossover denilen özel bağlantıda ise kablo içindeki ince tellerin sırası değiştirilebilir. Fakat bu tür bağlantılarda genellikle yapılan, hub üzerindeki bir anahtar ayarıyla crossover özelliğini sağlamaktır. Bu şekilde yapılan bağlantıya en fazla 4 hub birbirine bağlanabilir.

Kablonun kategorisi üretim kalitesiyle ilgilidir. Yapılan çeşitli testler ile kablonun belirtilen hızlarda elektrik sinyalini ne kadar sağlıklı ve az kayıpla iletebildiği, manyetik alan etkisine karşı sinyali ne kadar koruyabildiği ölçülür. Testler ile ortaya konan değerler kategorinin kriteridir. Bu kriterleri tutturabilen kablo bu kategoriyi almaya hak kazanır.

Category 1: UTP'nin telefon kablosu standardıdır. Yalnızca ses taşımak için kullanılır. Internet halen bu basit kablolar üzerinde hayatını sürdürüyor.

Category 2: Bu kategori 4 Mbps'lik (Megabyte/saniye) bir hızda veri iletebilir. Kablonun merkezinde iki çift iletken bakır tel vardır.

Category 3: 10 Mbps'lik hızda veri taşıyabilir. Dört adet bakır telden oluşur.

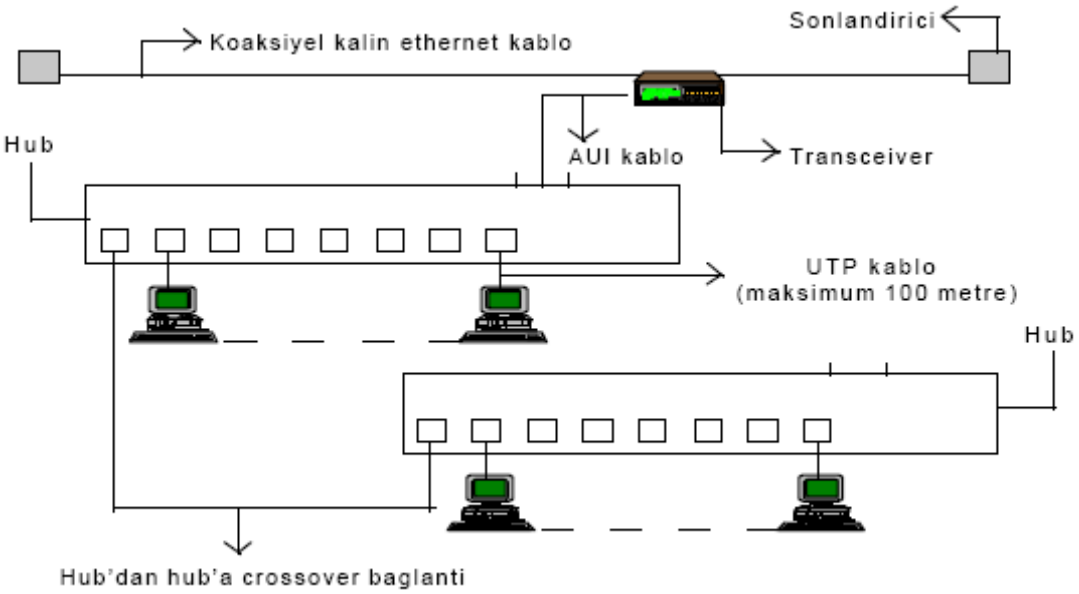
Category 4: 16 Mbps'lik hızda veri taşımaya yönelik yapılmıştır. Dört iletken telden oluşur.

Category 5: En yeni standarttır. 100 Mbps'lik bir hıza ulaşabilir. Oldukça esnek ve ucuzdur.

CAT5 ile 100 Mbit hızında veri aktarımı yapılabilir. Bir sonraki standart CAT5e (Enhanced CAT5, gelişmiş CAT5) standardıdır. Bu CAT5 ile aynı yapıda olup, daha üst seviye değerlere erişebilen bir kablodur. CAT5e ile Gigabit hızına ulaşılabilir. Gigabit ethernet'te CAT5 kullanılabilmekle bereber CAT5e tavsiye edilir.

CAT6'da da aynı durum söz konusu CAT5e'den de daha yüksek değerlere erişebilir. CAT6 şu anda 568A standardına eklenmiş yani resmen kullanıma sunulmuştur. 1000Mhz hızı için, yani Gigabit ethernet için en uygun kablodur.

UTP/STP Catagöri	Speed
Cat 2	**4 Mbps**
Cat 3	**10 Mbps**
Cat 4	**16 Mbps**
Cat 5	**100 Mbps**
Cat 6	**1000 Mbps**



Resim 3. 7–7 UTP Kablolama

Hub'ın biri AUI portundan koaksiyel kabloyla bağlı ve iki hub arasında crossover bağlantı var.

FİBER-OPTİK KABLOLAR

Fiber-optik kablolar verileri ışık olarak ileten yüksek teknoloji iletim ortamlarıdır. Fiberoptik kablolar hızlı ve yüksek kapasiteli veri iletimi için uygundur. Özellikler 100 Mbps hızında veri iletimi için kullanılır. Verilerin güvenliği açısından daha iyidir. Çünkü ışık olarak temsil edilen veriler başka bir ortama alınamazlar.

Fiber-optik kablo üzerinden veri aktarımı; ince fiber cam lifi (ışık iletkeni) üzerinden ışık dalgası şeklinde gerçekleştirilir. Aktarılacak her bir ışık işareti için ayrı bir ince fiber cam kullanılır. Bu çerçevede en basit hali ile bir Fiber-optik kablo 3 temel kısımdan oluşmaktadır:

Fiber Optic Kablo ve Sonlandırıcılar

Işığın geçtiği tabaka olan Asıl Işık İletkeni, ışığı yansıma ve kırılmalara karsı koruyan ve yine bir cam tabaka olan Cam Örtü ve tüm cam kısmı koruyan Koruyucu Kılıf olarak adlandırılabilecek kısımlardır. Uygulamada bunlara ilave olarak Fiber-optik kabloya; kablonun bina içi/bina dışı kullanım yeri ve şartlarına bağlı olarak çelik zırh yada jel tabakası gibi başka koruyucu ve esneklik kazandırıcı kısımlarda ilave edilebilmektedir.

Multimode Fiber SX port : 220 metre

Multimode Fiber LX port : 550 metre

Singlemode Fiber Lx port : 5000 metre

ETHERNET KABLOLAMA SİSTEMİ

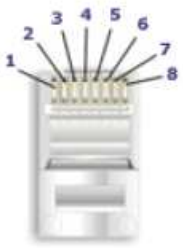
Ethernet network'lerinde dört çeşit kablolama sistemi kullanılır:

1. . Thick coaxial(kalın)
2. . Thin coaxial(ince)
3. . Unshielded Twisted Pair (UTP)
4. . Fiber-optic

UTP KABLO YAPIMI

UTP kablo yapmak hem kolay hem de eğlenceli bir iştir. Bunun için kablo ve RJ-45 ile bir de Jack pensesine ihtiyacımız var.

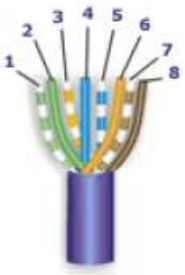
Kablo yapılırken dikkat edilecek unsurlardan biri de kablonun Düz, Cross yada Roll Over mi olacağı ve bu çeşitlerin renk sıraları.



1-Yeşil-Beyaz

2-Yeşil

3-Turuncu-Beyaz



4-Mavi

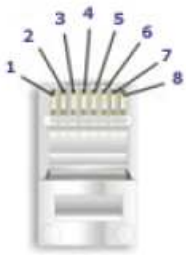
5-Mavi-Beyaz

6-Turuncu

7-Kahverengi-Bayaz

8-Kahverengi

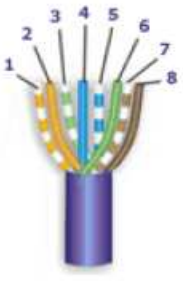
586-A ya göre bağlama



1-Turuncu-Beyaz

2-Turuncu

3-Yeşil-Beyaz



4-Mavi

5-Mavi-Beyaz

6-Yeşil

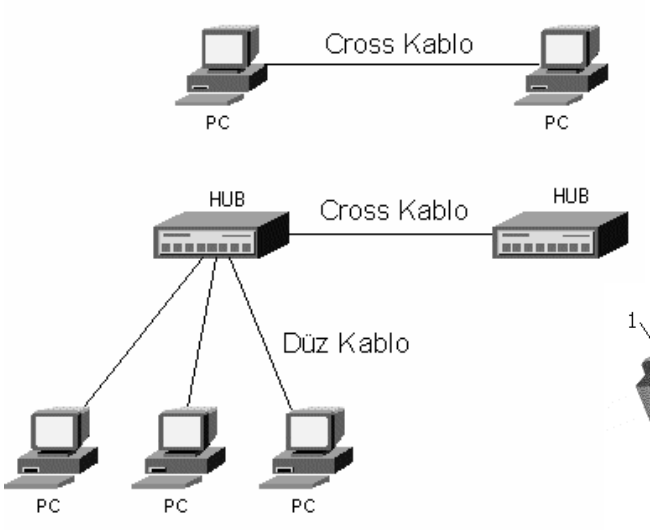
7-Kahverengi-Bayaz

8-Kahverengi

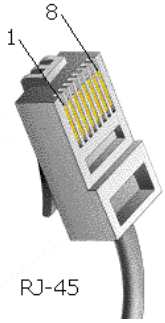
586-B ye göre bağlama

UTP KABLO NASIL YAPILIR

Kablo yaparken, yani bir kablonun iki ucuna jak takarken, kabloyu nerede kullanacağınıza bağlı olarak iki tipten bahsedilebilir. Düz kablo, cross(çapraz) kablo.

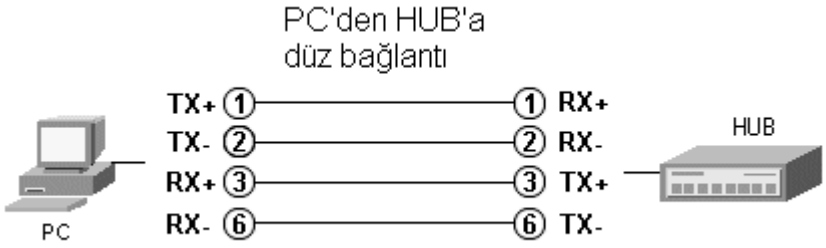


Gördüğünüz gibi aynı cihazlar arasında(PC-PC veya Hub-Hub) cross kablo kullanıyoruz. PC'den hub'a gidecek kablo ise düz kablo oluyor.



UTP kablonun ucuna taktığımız RJ-45 jak üzerindeki pinler jakın pinleri size bakacak şekilde tutulduğunda soldan sağa 1'den 8'e kadar sıralı kabul edilir.

DÜZ KABLO



Yanda düz bağlantıyı görüyoruz.

Dikkat ederseniz bilgisayarın ağ kartında 1. pin TX+ iken hub tarafında 1. pin RX+.

Dolayısı ile kabloyu yaparken kablonun iki ucundaki jaklarda, birebir bağlantı yaparsak, yani 1. pin karşıda da 1'e gidecek, 2. pin 2'ye... şeklinde yaparsak düz kablo yapmış oluruz. Böylece PC'nin gönderim yapan uçları(TX) hub'ın alım yapan uçlarına(RX) denk gelmiş olur. PC'nin direkt olarak hub'a bağlanmadığı ortamlarda, bilgisayar ile duvar prizi arasındaki kablolar, duvar prizlerinden patch panellere giden kablolar ve patch panelden hub'a giren kablolar hep düz kablodur. Kısacası, daima düz bağlantı yaparız ancak bazı özel durumlarda çapraz kablo gerekebilir.

ÇAPRAZ KABLO

İki Pc'yi, arada hub olmadan tek bir kablo ile bağlayabilirsiniz.

Ama her iki tarafta da 1 ve 2. pinler TX, 3 ve 6. pinler RX olduğuna göre, çapraz bağlamalısınız ki, TX ve RX'ler karşı karşıya gelsin.

İki hub arasında çapraz kabloda böyle oluyor. Farkı mı...

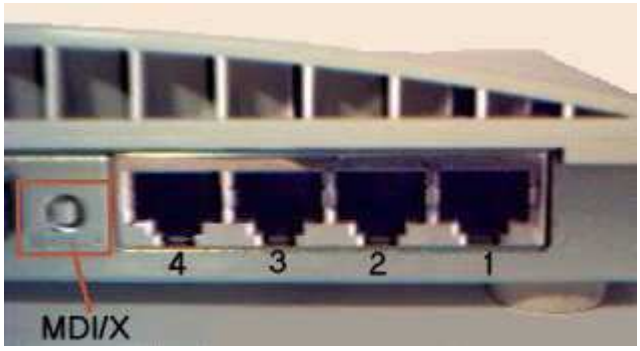
Kanalların ismi farklı olsa da sonuçta aynı çapraz kablo hem PC-PC hem de hub-hub bağlantısı için kullanılabilir...

Yani sizin yapacağınız çapraz kablo aynı.

HUB'LARIN BİRBİRİNE BAĞLANMASI

Hub'lar ile ilgili sık sık problem yaratan bir "kolaylıktan" bahsetmek gerekiyor.

Bugün 16 port bir hub alırsınız, bu bana uzun bir süre gider dersiniz, ama networkünüz o kadar hızlı büyür ki kısa zamanda bir hub daha alırsınız. Bu hubları da birbirine bağlamanız gerekir. Yani hub'ların birbirine bağlanması çok sık karşılaşılan bir durumdur. Eee, bizde ne yaparız, hub'ın üzerinde bilgisayar taktığımız portlardan ama bu sefer çapraz kablo ile iki hub'ı bağlarız.

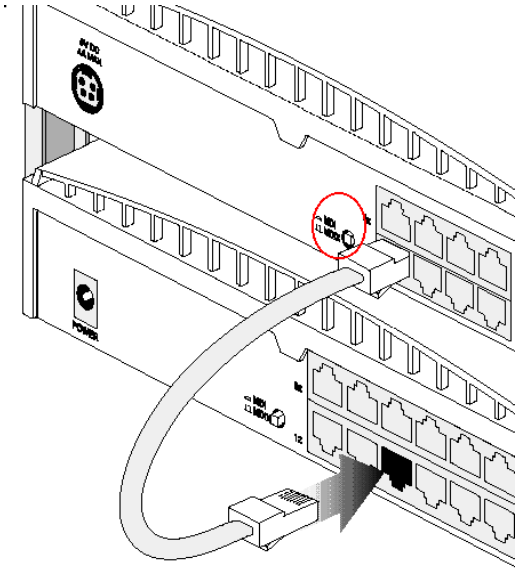


Hub üreticileri vatandaş çapraz kablo ile uğraşmasın diye şöyle bir güzellik yapmışlar, hubların bir çoğunda portlardan en büyük numaraya sahip olanın yanında crossover, uplink, out, MDI/X gibi ibareler bulunur. Bu şu anlama gelir:

"Eğer bu hub ile başka bir hub'ı bağlayacaksan, **düz** kablo kullanabilirsin. Düz kablonun bir ucunu bu porta tak ve portun yanında bir düğme varsa ona bas, kablonun diğer ucunu ise, diğer hub'ın normal bir portuna tak."

4. numaralı portun yanındaki düğmeye dikkat.

İki hub'ı düz kablo ile bağlarken, kablonun bir ucu 1. hub'un uplink portuna, diğer ucu ise diğer hub'ın normal bir portuna takılır.



Üçüncü bir hub daha bağlanırken bu sefer 2. hub'ın uplink portu kullanılacaktır.

Bazen bu uplink portu normal portlardan ayrıdır ve basmanız gereken bir düğme yoktur.

Eğer iki hub'da da BNC çıkışı varsa koaksiyel kablo ile de hub'ları bağlayabilirsiniz. Tabii ki iki uçta sonlandırıcı olması gerekiyor.

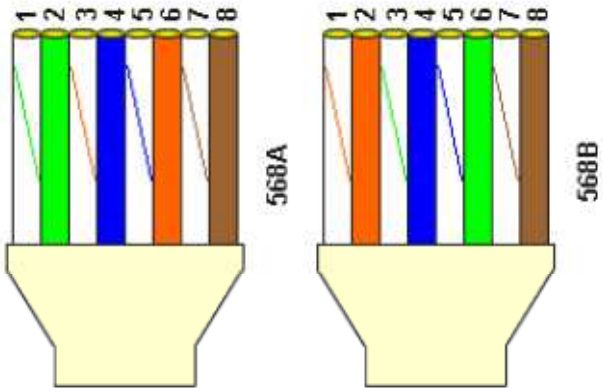
KABLO BAĞLANTI STANDARTLARI

Kablo uçlarını yaparken uymanız gereken, daha doğruyu uyarsanız sizin ve sizden sonra ağa müdahale edecek kişinin işini kolaylaştıracak standartlar vardır. Bu standarda uygun yaptığınız kablo veri kanallarının aynı tel çiftini kullanması kuralına uygun olacaktır.

EIA/TIA isimli kuruluş "EIA/TIA -568-A 'Commercial Building Wiring Standard' " isimli kablolama ile ilgili standartları belirlemiştir. Tüm dünyada üreticiler ve teknisyenler bu standartları takip ederler.

"EIA/TIA -568-A" standardı içinde kablo uçlarını yaparken kullanabileceğiniz elektriksel olarak birbirinin tamamen aynısı iki şema önerilmiştir. T568A şeması ve T568B şeması.

Her iki şemada da 1-2 ve 3-6'nın aynı çifte ait tellere denk geldiğine dikkat ediniz.

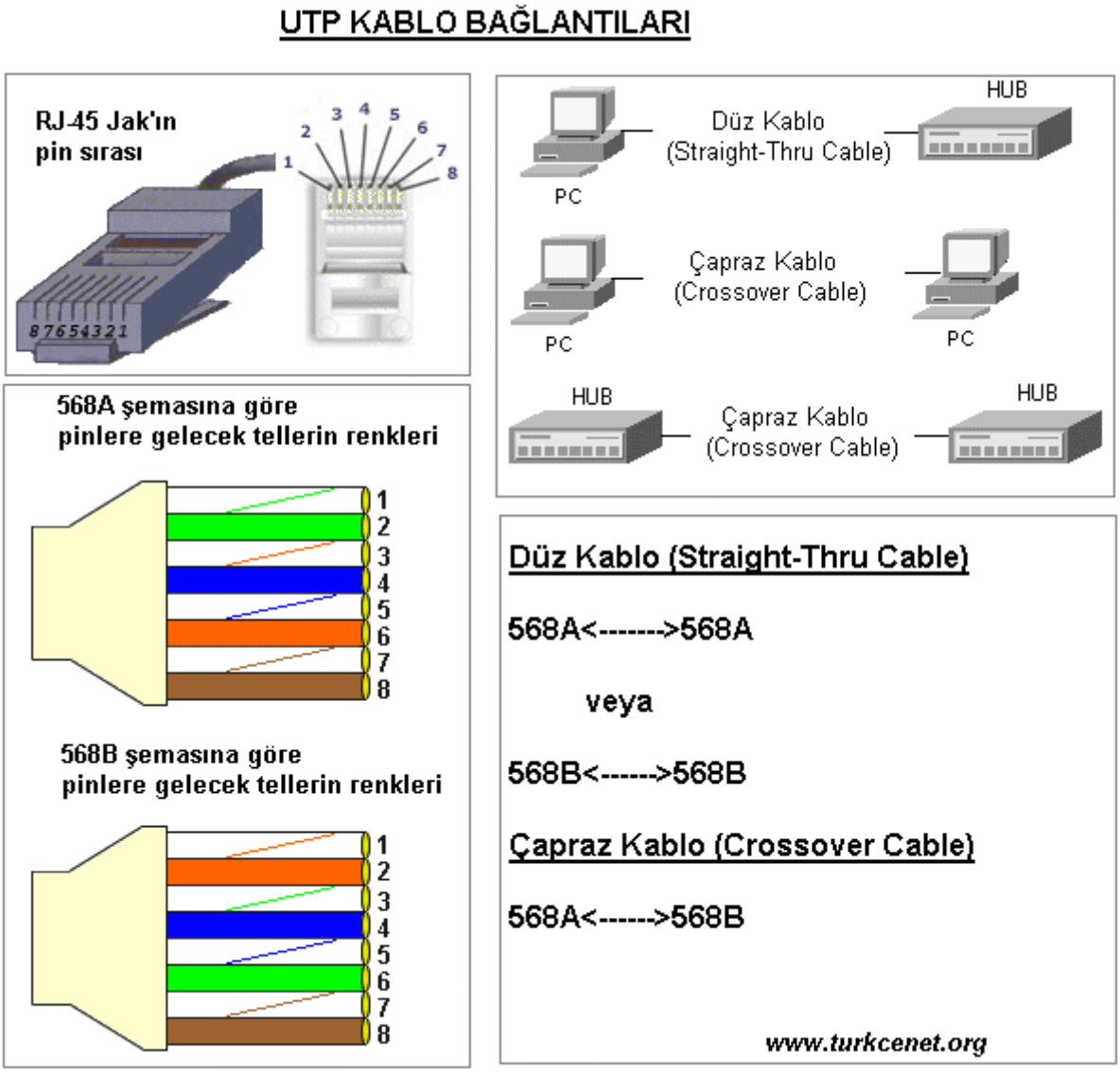


DÜZ KABLO

Düz kablo yapmak için iki uçta aynı şemada olmalı, yani 568A<->568A veya 568B<->568B şeklinde. Dolayısı ile iki seçeneğiniz var.

ÇAPRAZ KABLO

Eğer çapraz kablo yapmak istiyorsanız bir ucu 568A diğerini 568B şemasına göre yapmalısınız.

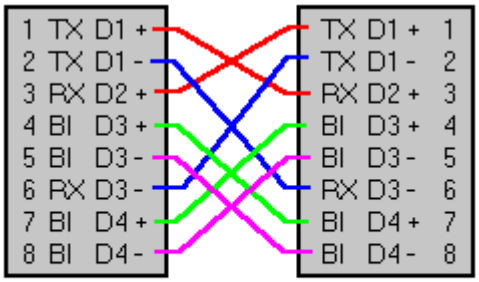


Düz kablo yaparken iki şemadan birini seçip renk kodlarını ezberlerseniz ve her yaptığınız kabloda bunu kullanırsanız, bir kablonun ucu bozulduğunda gidip diğer ucunu kontrol etmenize gerek kalmaz**.**

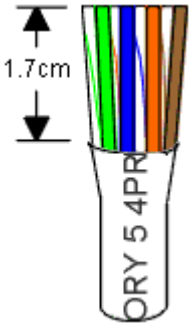
Peki hangisini seçeyim derseniz, bir çok kaynakta 568A<->568A şemasının dünyada en yaygın kullanılan şema olduğu söyleniyor...

GİGABİT ETHERNET

Yukarıdaki kablo bağlantıları 10BaseT ve 100BaseTX için yani 10Mbit ve 100Mbit ethernet için geçerlidir. 1000BaseT yani UTP kablo üzerinden gigabit ethernet kullanacaksanız düz bağlantıda bir farklılık yok. 568A<-->568A bağlantısını kullanabilirsiniz. Çapraz kabloda ise durum değişik, gigabit çapraz için alttaki şemayı kullanmanız gerekiyor.



Gigabit Ethernet için çapraz(cross)
UTP bağlantısı



Gördüğünüz gibi teller düzgün sırada ve uçları da dümdüz. Bu noktada açıkta olan tellerin boyu 1.7cm den daha uzun olmamalı. Aksi halde teller arasında sinyal bozulması olabilir özellikle 100Mbit için kullanılacaksa sakat..

NETWORK CİHAZLARI

Bir networkü sade bilgisayar ekleyerek genişletemeyiz. Bu bize kablolamanın zorlaşması, sinyal zayıflaması gibi sebeplerden sorun yaratır. Bu sebeple bir networkü genişletmek, güvenliğini saklamak ve aynı zaman da hiyerarşi kazandırmak için bazı cihazlar kullanmalıyız. Bu cihazlar genel olarak şunlardır:

1. Microtransceiver
2. Transceiver
3. Hub
4. Switch
5. Repeater
6. Bridge
7. Router
8. Firewal
9. Gateway

MİCROTRANSCEİVER

Ethernet ağlarında kullanılan kablo tiplerini birini diğerine çevirmeye yarar. Üzerinde 2 port yer alır ve çeşitli tipleri vardır. Örneğin BNC'yi AUI ya yada UTP'yi AUI ya çevirmek gibi.

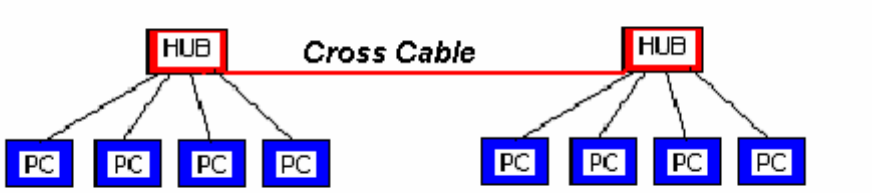
TRANSCEİVER

Transceiverlar kalın Ethernet kullanılan ağlarda koaksiyel kabloya bağlantı yapmak için kullanılır. Transceiverın kabloya takılan kısmında iğne gibi bir parça kablonun iletken kısmıyla temas sağlar. Ağrızalarda ilk kontrol edilmesi gereken aygıttır.

HUB

En basit network cihazıdır. Kendisine bağlı olan bilgisayarlara paylaşılan bir yol sunar. Yani Hub' a bağlı tüm cihazlar aynı yolu kullanırlar ve bu da aynı anda haberleşmek isteyen network cihazlarının, bir tek yol olduğu için hattın boşalmasını beklemelerine sebep olur. 8 – 12 – 16 – 24 portlu olarak üretilirler.

Eğer bir network kartı kendisine ait olmayan bir paketi aldığında, kendi adresinin paketin ulaştırılması gereken yerdeki MAC adresiyle aynı olmadığını fark ederse, paketi yok eder. Bu işleme **drop etme** de denilmektedir.



SWİTCH

Kendisine bağlı cihazlara adından da anlaşılacağı gibi anahtarlamalı bir yol sunar. Hub ile kıyaslandığından en önemli farkı budur. Switch'ler, bir topolojinin merkezinde yer alırlar ve onlara gönderilen verileri, bağlı olan bilgisayarlardan birine gönderirler. Switch'lerin hub'larla farkı ise, switch'lerin veri paketlerini, ona bağlı olan iki bilgisayar arasında doğrudan iletebilmesidir. Hub'lar, swicth'lerin aksine A bilgisayarından B bilgisayarına gidecek olan veri paketini, tüm bilgisayarlara gönderirken, switch'ler A bilgisayarından B bilgisayarına doğrudan paketi taşır. Bu sayede Hub'a göre daha yüksek bir performans sağlanacaktır. 8 -12 – 16 – 24 – 36 – 48 port lu olarak yada saseli üretilebilirler. Saseli switchlerde boş yuvalar vardır ve gerektiğinde port eklenebilmektedir.

Fakat bir switch ilk kez çalıştırıldığında, hangi port'unda hangi MAC adresini taşıyan bilgisayarın bulunduğunu bilemez. Switch, bir süre hub gibi çalışarak bir tür öğrenme sürecine girer. Bir bilgisayardan gelen paketi diğer tüm bilgisayarlara (sadece ilk anda) gönderir. Sonra bilgisayarların Network Kartlarının MAC adreslerini içerişindeki çiplerde tutmaya başlarlar. Switch'in ilk anda yaptığı bu isleme **flood** adı verilmektedir.

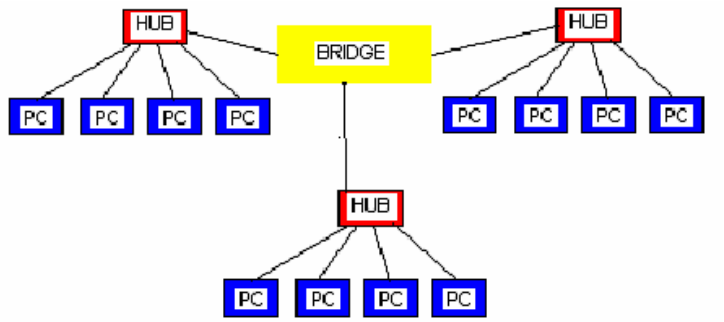
REPEATER

Repater bir ethernet segmentinden aldığı tüm paketleri yineler ve diğer segmente yollar. Repeater gelen elektrik sinyallerini alır ve binary koda yani 1 ve 0'lara çevirir. Sonra da diğer segmente yollar. Bu yönüyle repater'in basit bir yükseltici olmadığını anlıyoruz. Çünkü yükselticiler gelen sinyalin ne olduğuna bakmadan sadece gücünü yükseltir. Yolda bozulmuş bir sinyal yükselticiden geçince bozulma daha da artar.

Repeater ise gelen sinyali önce 1 ve 0'a çevirdiği için yol boyunca zayıflamış sinyal tekrar temiz 1 ve 0 haline dönüşmüş olarak diğer segmente aktarılır. OSI Katmanlarından 1. katmanda çalışır.

BRİDGE

İki TCP/IP ağını birbirine bağlayan bir donanımdır. Fazla karmaşık aygıtlar olmayan bridge'ler gelen frame'leri (veri paketleri) alır ve yönlendirirler. Bridge'ler fiziksel bağlantının yanı sıra network trafiğini kontrol eden aygıtlardır. Bridge bir çeşit yönlendirme yapar diyebiliriz fakat OSI Katmanlarından 2. katman yani Data-Link Katmanında çalışmasıyla Router' dan ayrılır.

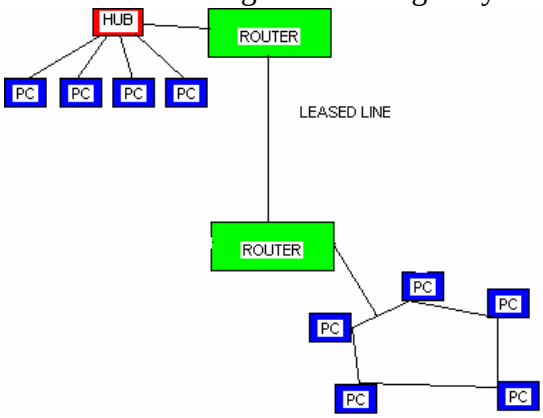


FİREWALL

Türkçe güvenlik duvarı anlamına gelen firewall, özel ağlar ile Internet arasında, her iki yönde de istenmeyen trafiği önleyecek yazılımsal yada donanımsal sistemdir. Firewall' ların verimli bir şekilde kullanılabilmesi için internet ve özel ağ arasında ki tüm trafiğin firewall üzerinden geçmesi ve gerekli izinlerin / yetkilerin kısaca erişim listelerinin uygun bir stratejiyle hazırlanmış olması gerekir. Donanımsal firewall' lara verilebilecek en güzel örnek CSecurity derslerinde detaylı anlatılan Pix Firewall' dır.

ROUTER

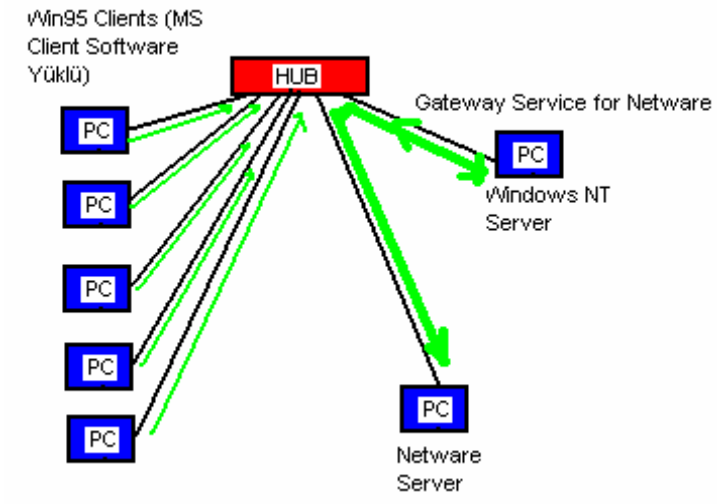
Router bir yönlendirme cihazıdır ve LAN-LAN yada LAN-WAN gibi bağlantılarda kullanılır. Router' ları basit bir yönlendirici olarak tanımlamak yetersiz olabilir. Çünkü Router' lar bir işletim sistemine sahiptirler (IOS – Internetworking Operating Systtem) dolayısıyla programlanabilirler ve gerekli konfigürasyonlar yapıldığında bir uzak networke erişmek için mevcut birden fazla yol arasında kullanabilecekleri en iyi yolun seçimini yapabilirler (Best Path Determination).



Üzerinde LAN ve WAN bağlantıları için ayrı portla bulunur ve saseli olarak ta üretilebilirler. Gereksinime göre bu yuvalara LAN yada WAN portları eklenebilir. İlerleyen bölümlerde Router' ların konfigürasyonları detaylı olarak anlatılacaktır. OSI Katmanlarının 3. katmanında çalışır.

GATEWAY

Gateway'ler, router'ların yaptığı ve farklı teknolojiler arasında gidip gelen veri paketlerinin dönüştürülmesi işlemini gerçekleştirirler. Router'ların ana görevi farklı segmentteki network'leri ayırmak ve yönetmektir. Ama "Router" başlığı altında da değindiğimiz gibi, bazı router'lar bu temel görev tanımını asarak, ATM ve Ethernet arasında verileri dönüştürme işlemini de gerçekleştirebilir.



Fakat esas olarak bu görev gateway denilen cihazlara verilmiştir. Gateway, genellikle adanmış bir aygıt veya adanmış bir bilgisayar üzerinde çalışan bir grup servistir. Gateway'ler örneğin FDDI network'ünden gelen paketleri alır, gidecekleri bilgisayarın adres bilgisini koruyarak, Ethernet network'ünde yol alabilecek şekilde

yeniden oluşturur ve bunu bir Ethernet network'üne gönderebilir. Bu işlem çok basit gibi gözükse de, iki network'de kullanılan paket yapıları, adresler ve adres yolları çok iyi bilinmek zorundadır. Gateway'ler bu işlem için özelleştirilmiş cihazlardır. Bu sebeple de bu fonksiyon için router'lara göre bir sınıf üstün sayılırlar. Zira router'ların asıl görevleri arasında farklı aktarım teknolojilerini birleştirmek hedeflenmez. Eğer aktarım teknolojisi açısından mimari bir farklılık varsa ve farklı fiziksel protokoller kullanılıyorsa, Gateway'ler kullanılmalıdır.

NETWORK TOPOLOJİLERİ

Topoloji dediğimiz de bir ağın fiziksel yada mantıksal yapısını anlamalıyız. Networkü oluşturan cihazların fiziksel yerleri, kabloların bağlanış şekilleri, iletişimde kullanılan protokoller gibi birçok unsur network topolojilerini belirler.

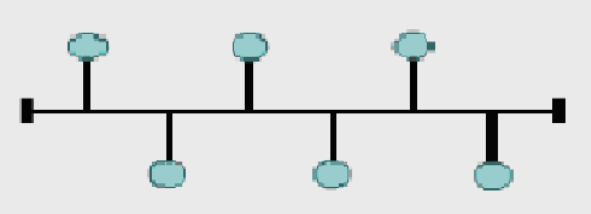
Bir topoloji hem fiziksel hem de kavramsal bir yapıdır.

Fiziksel yapı bir network'de cihazların nasıl birbiri ile bağlanacağını ve ne tür araçlar kullanılacağını belirler.

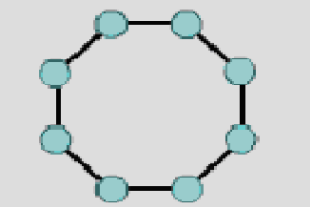
Kavramsal olarak bir topoloji network'teki veri trafiğinin planlanmasını sağlar.

FİZİKSEL TOPOLOJİLER:

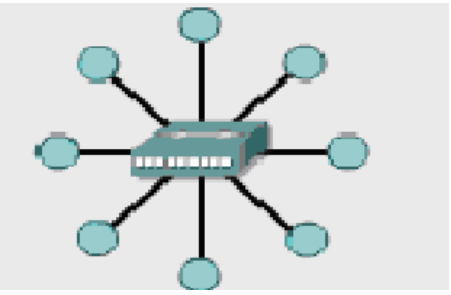
Bus Topoloji: Bu ağ topolojisine kuyruk adı da verilir. Bütün terminallerin tek bir doğrusal bir kablo ile birbirlerine bağlanmışlardır. Bu ağı taşıyan ana kabloya omurga **(backbone)** adı verilir. Burada hatta gönderilen sinyal bütün terminallere gider. Sinyal hedefe ulaşana yada bir sonlandırıcıya gelene kadar hatta dolaşır. Çok az miktarda kablo kullanılıyor olması avantaj gibi görünse de ana kabloda meydana gelebilecek bir kopma bütün networkün çökmesine sebep olabilecektir. Bir bus topolojisindeki ağda aynı anda sadece bir bilgisayar veri gönderebilir. Ayrıca sorun giderme zorluğu ve hatta eklenen her yeni bilgisayarın networkün yükünü artırması da dezavantajlar arasında sayılabilir. Maksimum kapasitesi 10 ile 12 bilgisayar olup 2 bilgisayar arası maksimum mesafe eş eksenli kablo kullanıldığında 185 metre kalın eş eksenli kablo kullanıldığında ise 500 metredir.



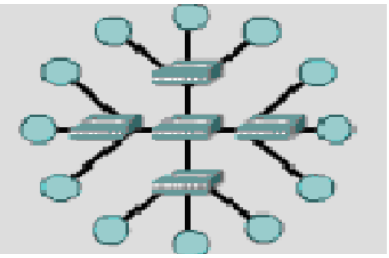
Ring Topoloji: Bu topolojide adından da anlaşılacağı gibi dairesel bir yapı söz konusudur. Hatta gönderilen sinyaller hedefe ulaşana kadar tüm terminallere uğrar. Tüm terminaller eşit haklara sahiptir. Genelde UTP korumasız çift dolanmış yada STP korumalı Çift dolanmış kablo kullanılarak oluşturulur. Bilgisayarlarla bağlantı cihazının maksimum mesafesi 100 metredir.



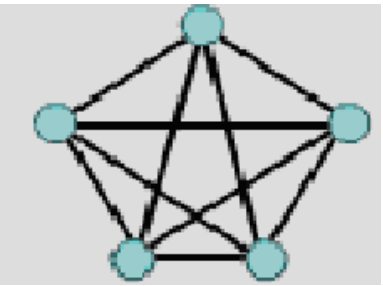
Star Topoloji: Star Topolojide her bilgisayar switch, hub yada başka bir server dediğimiz network cihazlarına direk bağlıdır. Hatta gönderilen sinyal önce switch yada hub'a gelir ve buradan hedefe gönderilir. Böyle bir yapının en büyük avantajı yeni bilgisayarlar ekleyerek büyümek çok kolaydır, yönetilmesi ve sorun giderilmesi kolaydır. Fakat diğer topolojilere göre çok daha fazla kablo kullanılmak zorunda kalınması ve switch yada hub'ın devre dışı kalmasıyla tüm networkün çökecek olması gibi dezavantajları vardır.



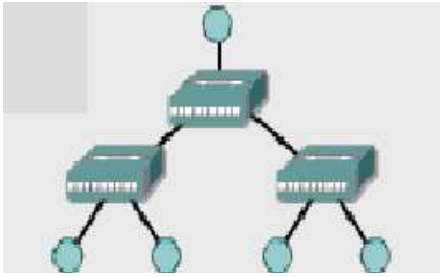
Extended Star Topoloji: Adından anlaşılacağı gibi Star Topolojinin geliştirilmesiyle ortaya çıkmıştır. Birden fazla yıldız topolojinin bir araya gelmesiyle oluşmuş bir yapıdır diyebiliriz.

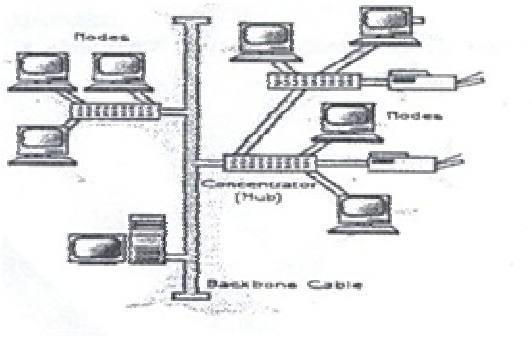


Mesh Topoloji: Networkte bulunan bütün bilgisayarla diğer bütün bilgisayarlara direk bağlıdırlar. Uçtan uca bütün bilgisayarlar birbirine direk bağlı olduğu için hedefe kısa zamanda ulaşılır, iki bilgisayar arasında ki bağlantının kopması durumda alternatif bir sürü yol olacaktır ama maliyetinin çok yüksek olması da unutulmamalıdır.



Hiyerarsik Topoloji: Üzerinde Bus topoloji ve Yıldız topolojiden özellikler taşır.





Ağaç Ağ Bağlantısı (Tree Network)

Ağaç topolojilerine dağınık ve yıldız topolojilerinin karakteriştiklerini birleştirir. Dağınık ağ omurga üzerinde yıldız topolojide bilgisayarlardan oluşur. Varolan bir ağın genişletilmesinde sıklıkla kullanılır. Ağaç topolojilerde 5-4-3 kuralı vardır. Nedir bu kural? Ethernet protokolü 5-4-3 kuralıdır. Bir sinyal gönderildiğinde belli bir süre içinde ağın parçalarına ulaşır. Her bir switch/hub veya bir repeater sinyalin ulaşma süresine nispeten çok küçük bir zaman dilimi daha ekler. Ağdaki iki terminal (file serverler,worstaionlar ve diğer çevre birimleri) arasında maksimum 5 segment ve 4 repeatrs/switches/hub ve eğer kablo kullanılmamışsa sadece 3 trunk segment olabilir.

Eğer ağ uca komple fiber optik kablo ile tesis edilmiş ise veya omurga fiber optik kablo ve UTP kablolama ile karma tesis edilmiş ise kural 7-6-5 olarak revize edilir.

MANTIKSAL TOPOLOJİLER:

Broadcast Topoloji: Bir bilgisayar hatta gönderdiği bir sinyali diğer bütün bilgisayarların alacağı bir yayın şeklinde yapar. Yayın hedefe ulaştığı ana kadar bütün terminalleri tek tek dolaşır.

Token Passing Topoloji: Burada taşıyıcı görevinde olan bir token her bir terminale uğrayarak ağ ortamında dolaşır. Uğradığı terminal ekleyeceği bir data varsa onu token' a ekleyerek, ekleyecek bir datası yoksa direk bir sonraki terminale aktarır. Bu şekilde çalıştığında bir repeater görevi de üstlenmiştir.

TCP / IP KATMANLARI

TCP/IP ile olarak DARPA (Defense Advanced Research Projects Ağency) ve Bekeley Software Distribution tarafından geliştirilen UNIX' de kullanılan bir protokoller gurubudur. Günümüzde internetin temel protokolü olarak yerini almış TCP/IP'nin açılımı Transmission Control Protocol / Internet Protocol' dür.

TCP /IP modeli OSI katmanlarından çok daha önce standartlaştığı için OSI içinde referans olmuş 4 katmanlı bir yapıdır.

1. Uygulama Katmanı
2. Nakil Katmanı
3. Internet Katmanı
4. Ağa Giriş Katmanı

Uygulama Katmanı OSI modelindeki Uygulama, Oturum ve sunum katmanlarına karşılık gelmekte ve o katmanların işlevlerini yerine getirmektedir. Bu katmanda TFTP, FTP, SMTP, SNMP gibi protokoller çalışmaktadır.

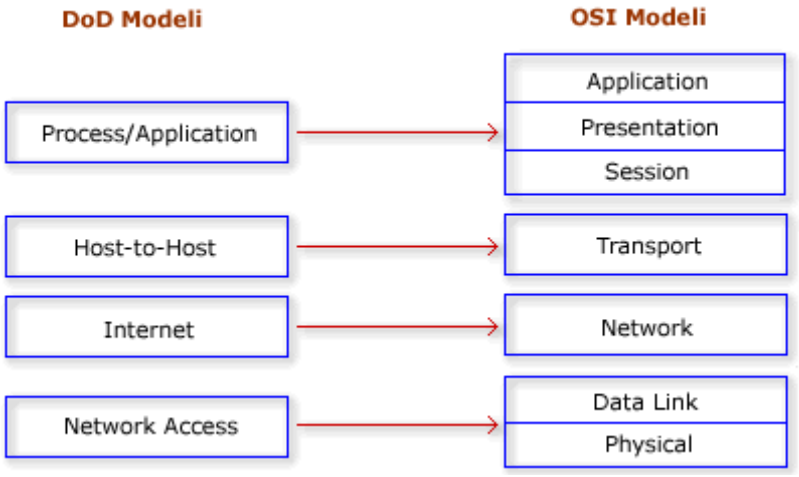
Nakil Katmanı OSI modelindeki Nakil katmanıyla bire bir esleştirilebilir. Bu katmanda iki farklı sınıfa ayrılabilecek iki protokol kullanılır. TCP ve UDP

1. Bağlantı Odaklı: TCP
2. Bağlantısız: UDP

Internet Katmanı OSI modelindeki Network katmanına denktir ve adresleme, en iyi yol seçimi gibi işlevleri yerine getirir.

Bu katman da IP (Internet Protocol), ICMP (Internet Control Messağe Protocol), BOOTP (Bootsrap Protocol), DHCP (Dynamic Host Configuration Protocol), ARP (Adres Resolution Protocol) ve RARP (Reverse Address Resolution Protocol) gibi protokoller çalışmaktadır.

Ağa Giriş Katmanı ise OSI modelinde ki Data-Link ve Fiziksel Katmana denk gelmektedir.



OSI REFERANS MODELİ

Kullanıcıların farklı talepleri ve dolayısıyla network üzerinde kullanılmak zorunda kalınan karmaşık uygulamalar, ağ kurulumlarında bir hiyerarşinin dogmasını kaçınılmaz yapmıştır. Bilgisayar ağları büyüdükçe bu ağları yönetmek ve sorun gidermek, standart bir yapı olmadığı da düşünülürse çok daha zorlaşmaya başladı.

Uluslararası Standartlar Organizasyonu (ISO) bir çok ağ yapısını inceleyerek 1984 yılında OSI referans modelini geliştirdi. Artık donanım ve yazılım firmaları bu standarda uygun ürünler üretmeye başladılar.

OSI modelinde 7 katmanlı bir yapı kullanılmış ve bu model; karmaşıklığı azaltmış, insanların belli katmanlarda uzmanlaşması için referans olmuş, katmanların işlevlerinin öğrenilmesi ve öğretilmesi kolaylaşmış, farklı donanım ve yazılım ürünlerinin birbirleriyle uyumlu çalışmasını sağlamış ve bir katmanda yapılan değişiklikler diğer katmanları etkilemediği için işbirliği, görev paylaşımı, problem çözümünü gibi konularda kolaylıklar getirmiştir.

Bahse konu OSI katmanlarını şu şekilde sıralayabiliriz.

7. Uygulama Katmanı (Application Layer)
6. sunum Katmanı (Presentation Layer)
5. Oturum Katmanı (Session Layer)

4. Nakil Katmanı (Transport Layer)

3. Ağ Katmanı (Network Layer)

2. Data Link Katmanı (Data Link Layer)

1. Fiziksel Katman (Physical Layer)

Burada Uygulama, Oturum ve sunum katmanları üst katmanlar olarak adlandırılırlar ve işlevlerini yazılımlar sağlamaktadır. (Bu katmanlar TCP/IP modelinde Uygulama Katmanı adı altında tek bir katman olarak yapıya dahil edilmiştir.) Nakil, Ağ, Data Link ve Fiziksel katmanlar ise alt katmanlar olarak adlandırılırlar ve işlevlerini bilgisayarların ve ağda kullanılan diğer cihazların donanımları ve bu donanımlar üzerindeki yazılımlar sağlar.

Application	BOOTP, DNS, FTP, SMTP, SNMP, Telnet
Presentation	SMB
Session	Named Pipes, NetBIOS, DLC
Ttansport	SPX, TCP, UDP, NetBEUI
Network	ARP, IP, IPX, NWLink, RIP
Data Link	MAC, CSMA/CD, 802.3, 802,5, 802.12**LLC, PPP
Phyical	802,2. PPP. SLIP. Ethernet. Token Ring

7. Uygulama Katmanı (Application Layer)

Kullanıcıya en yakın olan katmandır ve diğer katmanlara herhangi bir servis sağlamaz. Burada kullanılan bazı uygulamalara şu örnekleri verebiliriz; Uygulama katmanı programların ağı kullanabilmesi için araçlar sunar. Microsoft API'leri uygulamakatmanında çalışır. Bu API'leri kullanarak program yazan bir programcı, örneğin bir ağ sürücüsüneerişmek gerektiğinde API içindeki hazır aracı alıp kendi programında kullanır. Alt katmanlardagerçekleşen onlarca farklı işlemin hiçbirisiyle uğraşmak zorunda kalmaz.

FTP, TFTP, Telnet, SMTP, SNMP, HTTP

6. Sunum Katmanı (Presentation Layer)

Gönderilecek datanın, datayı alacak bilgisayar tarafından da anlaşılabilecek ortak bir formata dönüştürüldüğü katmandır. Bu katmanda data transferinin güvenli olması için şifreleme de mümkündür.

Dos ve Windows 9x metin tipli veriyi 8 bit ASCII olarak kaydederken (örneğin A harfini 01000001 olarak), NT tabanlı işletim sistemleri 16 bit Unicode'u kullanır (A harfi için 0000000001000001). Ancak kullanıcı tabii ki sadece A harfiyle ilgilenir. sunum katmanı bu gibi farklılıkları ortadan kaldırır.

Sunum katmanı günümüzde çoğunlukla ağ ile ilgili değil, programlarla ilgili hale gelmiştir. Örneğin eğer siz iki tarafta da gif formatını açabilen bir resim gösterici kullanıyorsanız, bir makinanın diğeri üzerindeki bir GIF dosyayı açması esnasında sunum katmanına bir iş düşmez, daha doğrusu sunum katmanı olarak kastedilen şey, aynı dosyayı okuyabilen programları kullanmaktır.

Data formatlarına şu örnekler verilebilir;

MPEG GIF JPEG ASCII

5. Oturum Katmanı (Session Layer)

Oturum katmanı bir bilgisayar birden fazla bilgisayarla aynı anda iletişim içinde olduğunda, gerektiğinde doğru bilgisayarla konuşabilmesini sağlar. Örneğin A bilgisayarı B üzerideki yazıcıya yazdırıken, C bilgisayarı B üzerindeki diske erişiyorsa, B hem A ile olan, hem de C ile olan iletişimini aynı anda sürdürmek zorundadır.

Bu katmanda çalışan NetBIOS ve Sockets gibi protokoller farklı bilgisayarlarla aynı anda olan bağlantıları yönetme imkanı sağlarlar.

SQL, Netbios Adları, NFS

4. Nakil Katmanı (Transport Layer)

Bu katman nakil edilecek datanın bozulmadan güvenli bir şekilde hedefe ulaştırılmasını sağlar. Üst katmanlardan gelen her türlü bilgi nakil katmanı tarafından diğer katmanlara ve hedefe ulaştırılır. Gönderilen datanın bozulmadan ve güvenli bir şekilde hedefe ulaşıp ulaşmadığını uygun protokollerle kontrol edebilir. Bu katmanda çalışan protokollere verilebilecek bazı örnekler şunlardır; TCP, UDP

Bu katmanın en önemli iki fonksiyonun Güvenlilik ve Akış kontroldür. Güvenlilik bilgisayarlar arasıdan gerçekleştirilen data transferinde datanın sağlıklı bir şekilde hedefe gönderilip gönderilmediğini yöneten, gönderilemediği durumlarda tekrar gönderilmesini sağlayan fonksiyondur.

İletişim halindeki bilgisayarlarda datayı gönderen bilgisayar alıcının kapasitesinden üzerinde datalar gönderebilirler. Böyle bir durumda datayı alan bilgisayar alamadığı paketleri yok edecektir ki önlemek için Nakil Katmanı Ara Bellekleme, tıkanıklıktan Kaçınma ve Pencereleme metotlarını kullanarak akış kontrolünü sağlar.

Ara bellekleme de datanın akış hızına müdahale etmeden, kapasitenin üzerindeki datanın ara belleğe alınması, tıkanıklıktan kaçınma metodun da ICMP Source Quench mesajı ile gönderen bilgisayarın gönderimini yavaşlatması, Pencereleme metoduyla paketlerin gruplar halinde gönderilmesi sağlanır.

3. Ağ Katmanı (Network Layer)

Bu katman bir paketin yerel ağ içerişinde yada diğer ağlar arasında ki hareketini sağlayan katmandır. Bu hareketin sağlanabilmesi için hiyerarşik bir adresleme yapısı gerekmektedir. Gelişen teknolojiyle birlikte mevcut ağlarında büyüme eğiliminde olması adresleme yapısının hiyerarşik olmasını gerektirmektedir. Ayrıca hiyerarşik sistem dataların hedef bilgisayara en etkili ve en kısa yoldan ulaşmasını da sağlar.

Bu katmanın bir özelliği olan Adresleme sayesinde bu sağlanabilmiştir. Adresleme Dinamik yada statik olarak yapılabilir. Sabit adresleme el ile yapılan adreslemedir. Dinamik adresleme de ise otomatik olarak ip dağıtacak örneğin DHCP gibi bir protokole ihtiyaç vardır.

Ayrıca bu katmanda harekete geçen bir datanın hedefine ulaşabilmesi için en iyi yol seçimi de yapılır. Bu isleme Routing bu işlemi yerine getiren cihaza ise Router diyoruz. Router en basit tarif ile en iyi yol seçimini yapar ve broadcast geçirmediği için ağ performansını olumsuz etkilemez. Bu katmanda kullanılan protokollere de şu örnekler verilebilir;

IP, ARP, RARP, BOOTP, ICMP

2. Data Link Katmanı (Data Link Layer)

Fiziksel adreslemenin ve network ortamında datanın nasıl taşınacağının tanımlandığı katmandır. Burada fiziksel adreslemeden kastettiğimiz şey MAC (Media Access Control) adresidir. Bu katman Hakemlik, Adresleme, Hata Saptama, Kapsüllenmiş Datayı Tanımlama fonksiyonlarına sahiptir.

Veribağlantısı katmanının büyük bir bölümü ağ kartı içinde gerçekleşir. Veri bağlantısı katmanı ağüzerindeki diğer bilgisayarları tanımlama, kablonun o anda kimin tarfından kullanıldığının tespiti ve fiziksel katmandan gelen verinin hatalara karşı kontrolü görevini yerine getirir. Veri bağlantısı katmanı iki alt bölüme ayrılır: Media Access Control(MAC) ve Logical Link Control(LLC).

MAC alt katmanı veriyi hata kontrol kodu(CRC), alıcı ve gönderenin MAC adresleri ile beraber paketler ve fiziksel katmana aktarır. Alıcı tarafta da bu işlemleri tersine yapıp veriyi veri bağlantısı içindeki ikinci alt katman olan LLC'ye aktarmak görevi yine MAC alt katmanına aittir.

LLC alt katmanı bir üst katman olan ağ katmanı(3. katman) için geçiş görevi görür. Protokole özel mantıksal portlar oluşturur (Service Access Points, SAPs). Böylece kaynak makinada ve hedef makinada aynı protokoller iletişime geçebilir(örneğin TCP/IP<-->TCP/IP). LLC ayrıca veri paketlerinden bozuk gidenlerin(veya karşı taraf için alınanların) tekrar gönderilmesinden sorumludur. Flow Control yani alıcının işleyebileğinden fazla veri paketi gönderilerek boğulmasının engellenmesi de LLC'nin görevidir.

Ethernet hakemlik için CSMA/CD (Carrier Sense Multiple Access with Collision Detect) adı verilen bir algoritmayı kullanır. Bu algoritma şu adımlardan oluşur;

1. Hatta boş olup olmadığını dinler

2. Boşsa data gönderir

3. Doluysa bekler ve dinlemeye devam eder

4. Data transferinde çarpışma olursa durur ve tekrar dinlemeye başlar.

Adresleme için, MAC adresi, Unicast adresi, broadcast adresi ve multicast adresi örnek olarak verilebilir.

Bu katman kullanılan protokollere şu örnekler verilebilir;

HDLC, PPP, ATM, Frame Relay

1. Fiziksel Katman (Physical Layer)

1. katman veya fiziksel katman verinin kablo üzerinde alacağı fiziksel yapıyı tanımlar. Diğer katmanlar 1 ve sıfır değerleriyle çalışırken, 1. katman 1 ve sıfırların nasıl elektrik, ışık veya radyo sinyallerine çevrileceğini ve aktarılacağını tanımlar. Gönderen tarafta 1. katman bir ve sıfırları elektrik sinyallerine çevirip kabloya yerleştirirken, alıcı tarafta 1. katman kablodan okuduğu bu sinyalleri tekrar bir ve sıfır haline getirir.

Fiziksel katman veri bitlerinin karşı tarafa, kullanılan medya(kablo, fiber optik, radyo sinyalleri) üzerinden nasıl gönderileceğini tanımlar. İki tarafta aynı kurallar üzerinde anlaşmamışsa veri iletimi mümkün değildir. Örneğin bir taraf sayısal 1 manasına gelen elektrik sinyalini +5 volt ve 2 milisaniye süren bir elektrik sinyali olarak yolluyor, ama alıcı +7 volt ve 5 milisaniyelik bir sinyali kabloda gördüğünde bunu 1 olarak anlıyorsa veri iletimi gerçekleşmez.

Fiziksel katman bu tip çözülmesi gereken problemleri tanımlamıştır. Üreticiler(örneğin ağ kartı üreticileri) bu problemleri göz önüne alarak aynı değerleri kullanan ağ kartları üretirler. Böylece farklı üreticilerin ağ kartları birbirleriyle sorunsuz çalışır.

Kablolar, hub, repeater cihazla bu katmanda yer alırlar.Bu katman da herhangi bir protokol tanımlanmamıştır.

DATA ENCAPSULATİON (VERİ PAKETLEME)

Veri paketleri her katmandan geçtikçe hem basına hem de sonuna gerekli eklemeler yapılır yada içeriği değiştirilebilir. Bu noktada katmanların her biri (ister OSI modeli içinde olsun; isterse TCP/IP içinde), verileri her seferinde bir parça değiştirir ve içeriğini bozmadan nereye gideceği, ne is için kullanılacağı yada hangi katmanda değerlendirilmesi gerektiği gibi

bilgiler eklenir. TCP/IP protokolünün katmanlarından çıkan bir veri paketinin header kısmında, temelde port numarası, ulaşacağı IP adresi yazılıdır. Veri paketleri, OSI katmanlarında hareket ettikçe, değişikliğe uğrarlar.

Ethernet teknolojisiyle taşınacak IP paketleri artık Ethernet teknolojisi ile gönderilecek şekle dönüştürülür. Bu dönüştürme işlemi için, IP paketleri Ethernet paketlerinin (Ethernet Frame'lerinin) içine eklenirler. Bu işleme ise (kapsülleme manasına gelen) **encapsulation** adı verilmiştir.

ARP işlemi ve ARP protokolünün getirdiği veriler ise Ethernet Frame'lerine eklenmektedir. IP paketinin header'inde destination (varış) ve originator (gönderen) IP adresleri yazar, Ethernet Frame'inin header'ından ise destination (varış) ve source (gönderen) MAC adresleri yazmaktadır

Data Encapsulation 5 adımdan oluşur.

1. Uygulama, sunum ve Oturum Katmanları kullanıcının girdiği veriyi 4. katman yani Nakil katmanına kadar getirir.

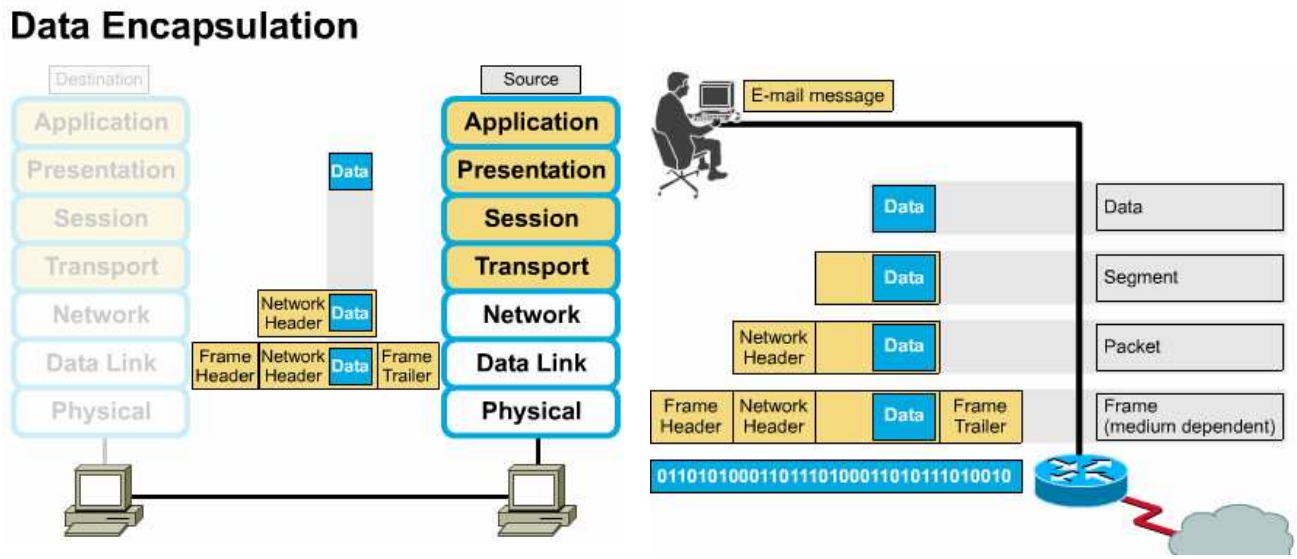
2. Nakil katmanı kendisine gelen bilgiyi segment adı verilen bölümlere ayırır ve datanın hangi protokolle gönderileceği (TCP - UDP) bilgisini de ekleyerek network katmanına gönderir.

3. Bu katmana gelen segment burada paketlere ayrılır ve IP header denen, hedef ve kaynak ipler gibi bilgileri bulunduğu başlığı ekleyerek bir alt katman olan data link katmanına gönderir.

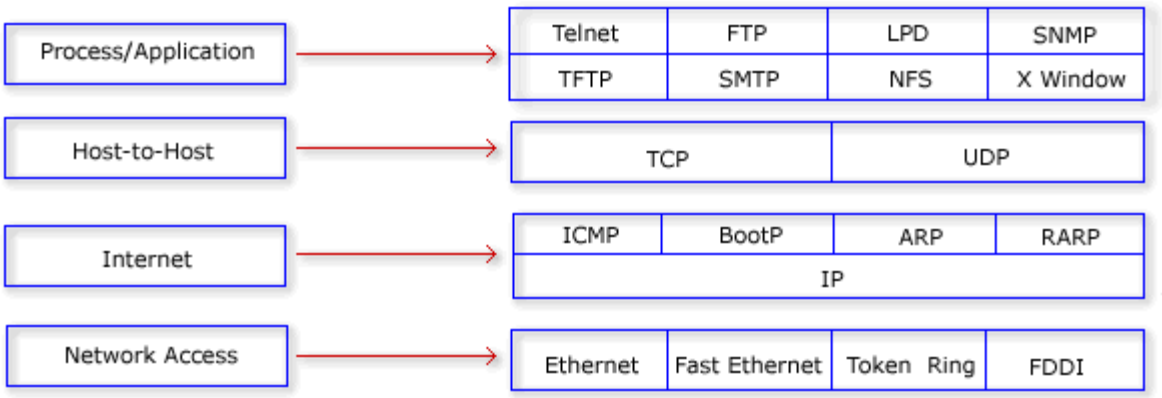
4. Burada data artık framelere çevrilir ve mac adresleri de eklenmiştir.

5. Frame yapı bu katmanda bitlere ayrılır ve iletilir.

Aşağıdaki iki şekil konun daha iyi anlaşılmasını sağlayacaktır.



TCP / IP PROTOKOLLERİ



PROCESS/ APPLİCATİON(UYGULAMA) KATMANI PROTOKOLLERİ

1- TELNET

Telnet bir terminal emülasyon protokolüdür. Bu protokol, kullanıcıların telmet istemci programlarını kullanarak Telnet sunuculara bağlanmalarını sağlar. Böylece telnet sunucuları uzaktan yönetilebilir.

2- FTP (FİLE TRANSFER PROTOCOL)

TCP tabanlı dosya transfer protokolüdür. FTP bağlantı kurulurken FTP sunucunun 21 numaralı portu kullanılır.

3- LPD (LİNE PRİNTER DEAMON)

Bu protokol yazıcı paylaşımını gerçekleştirmek için kullanılır.

4- SNMP (SİMPLE NETWORK MANAĞEMENT PROTOCOL)

SNMP protokolü ağlar üzerindeki birimleri denetlemek amacıyla geliştirilmiştir. Bir network cihazı üzerindeki sıcaklıktan o cihaza bağlı kullanıcılar, internet bağlantı hızından sistem çalışma süresine kadar bir çok bilgi SNMP protokolünde tanımlanmış bir yapı içerişinde tutulur.

5- TFTP (TRİVİAL FİLE TRANSFER PROTOCOL)

UDP tabanlı Cisco IOS tarafından desteklenen bir protokoldür. Router ve switchlerde dosya transferi için kullanılır, daha az hafıza ve işlemci gücü gerektirir. UDP tabanlı olduğu için hızlı bir iletişim söz konusudur fakat hata telafisi yoktur.

6- SMTP (SİMPLE MAİL TRANSFER PROTOCOL)

Mail göndermek için sunucu ile istemci arasındaki iletişim şeklini belirleyen protokoldür. Sadece mail yollamak için kullanılan bu protokolde, basitçe, istemci bilgisayar SMTP sunucusuna bağlanarak gerekli kimlik bilgilerini gönderir, sunucunun onay vermesi halinde gerekli maili sunucuya iletir ve bağlantıyı sonlandırır.

7- NFS (NETWORK FİLE SYSTEM)

Bu protokol farklı tipte iki dosya sisteminin bir arada çalışmasını sağlar.

8- X WİNDOW

Grakfiksel kullanıcı arayüzü tabanlı istemci sunucu uygulamaları geliştirmek için tanımlanmış bir protokoldür.

9- DNS (DOMAİN NAME SERVİCE)

Bu protokol internet isimlerinin (örneğin www.geocities.com gibi) IP adreslerine dönüştürülmesini sağlar.

10- DHCP (DYNAMİC HOST CONFİGURATİON PROTOCOL)

BOOTP protokolünün gelişmiş hali olan bu protokol ile tam dinamik ip konfigürasyon dağıtımı yapılabilir. sunucu – istemci ortamında çalışırlar ve istemcilerde ip adreslerini otomatik olarak alacaklarına dair bir konfigürasyon yapılmalıdır. DHCP ile belirlenen ip adresleri, subnet masklar, DNS server adresleri, varsayılan ağ geçidi gibi adresler dağıtılabilir, ip adresleri MAC adreslerine reserve edilebilir veya bazı ip adresleri tamamen kullanıma kapatılabilir. DHCP' den alınan ip adresleri DHCP server tarafından istemciye belirli sürelerle kiralanır ve istemci belirli aralıklara ile DHCP serverdan kira süresini yenilemesini ister. Yenilenme kira süresi dolana kadar yapılamazsa DHCP server tarafından istemciye yeni bir ip adresi verilir.

HOST- TO -HOS (NAKİL) KATMANI PROTOKOLLERİ

1- TCP (TRANSMİSSİON CONTROL PROTOCOL)

TCP, IP 'nin bir üst katmanında çalışan iki aktarım katmanı protokolünden birisidir.

TCP, güvenilir ve sanal devre üzerinden çalışan bir protokoldür. Aynı ağ üzerinde yada farklı ağlar üzerinde iki hostun birbirleriyle güvenilir bir şekilde haberleşmesini sağlar.

TCP 'nin baslıca özellikleri şunlardır:

1. Bir bağlantının (connection) kurulması ve sonlandırılması
2. Güvenilir (Reliable) paket dağıtımının sağlanması
3. Akış kontrolü (flow control) olanağı ile hortlarda veri tasmasının (overflow) önlenmesi
4. Bozulmuş yada ikilenmiş verinin düzeltilmesi (error recovery)
5. Alıcı host içerişinden birçok uygulama arasında demultiplexing yapılması

TCP, internet ortamında şu işlevleri sağlar:

1. Temel Veri Aktarımı (Basic Data Transfer)
2. Güvenilirlik (Reliability)
3. Uçtan uca akış Kontrolü (End to end flow control)
4. Çoğullama (Multiplexing)
5. Bağlantılar (connections)
6. Tam çift yönlü işlem (full duplex process)

TCP bağlantısının kurulması üç aşama (Three Way Handshake) sonucunda gerçekleşir:

1.Aşamada: Kaynak host bağlanmak istediği hosta bir istek paketi gönderir. Bu paketin TCP başlığında SYN = 1 ve ACK = 0 ‘dır. Gönderdiği paket içindeki segmente ait sıra numarası X ‘tir.

2.Aşamada: Bu paketi alan hedefe TCP başlığında SYN = 1, ACK = 1 bitlerini kurarak kendi paketini sıra numarasına SEQ Numarası=Y ve onay numarası, ACK Numarası = (X + 1) ‘i gönderir.

3.Aşamada: İsteğine karşılık bulan istemci son aşamada hedefe onay paketi gönderir ve bağlantı kurulmuş olur.

Sonra kaynak, hedefe göndermek istediği veri paketlerini gönderir.

TCP ve UDP üst protokollerle bağlantıda portları kullanırlar. 65535 adet port vardır ve IANA (Internet Assigned Numbers Authority) ilk 1024 portu Well-known portlar olarak ilan etmiştir. Bu portlardan bazılar şunlardır: **FTP: 21 Telnet: 23 SMTP: 25 DNS: 53**

Bir bilgisayar bir IP adresi ve bir port belirlediğinde buna soket **(socket)** ismi verilmektedir. Yani “X IP adresindeki bilgisayara, Y port’undan bilgi gönderildiğinde, bu bilgi su işlem için ele alınacaktır.” seklinde bir önerme ortaya çıkar.

TCP’nin adındaki kontrol lafından da anlayabileceğimiz gibi bu protokol iki bilgisayar arasındaki bilgilerin **doğru gidip-gelmesini** kontrol eder, eğer gelmemişse bunu karsıdan tekrar istemektedir, eğer geldi ise bunu **“Alınmıştır**” seklinde onaylar.

2- UDP (USER DATAĞRAM PROTOCOL)

UDP, TCP/IP protokol grubunun iki aktarım katmanı protokolünden birisidir. UDP, onay (acknowledge) gönderip alacak mekanizmalara sahip değildir. Bu yüzden veri iletiminde başarıyı garantileyemez. Yani güvenilir bir aktarım servisi sağlamaz. Uygulamalar güvenli ve sıralı paket dağıtımı gerektiriyorsa UDP yerine TCP protokolü tercih edilmelidir. Bazı UDP port numaraları şunlardır; **Who Is: 43 DNS: 53 NTP: 123 SNMP: 161**

TCP ile network üzerinden birbirini bularak haberleşen uygulamalara **connection oriented** (bağlantı yönelimli) denir. UDP kullanarak çalışanlara da **connectionless** (bağlantısız) denir.

İNTERNET KATMANI PROTOKOLLERİ

1- IP (INTERNET PROTOCOL)

Bağlantısız bir protokoldür. Bu protokol datanın hedefe ulaşması için gidebileceği en iyi yolu seçer ve gelen paketleri IP başlıklarını okuyarak networkteki bilgisayarların yerlerini belirler. IP başlıklarında gönderilecek datanın yasam süresi, datanın gönderilmesini sağlayacak protokol, kaynak ve hedef ip adresleri, kullanılan ip versiyonu gibi bilgiler bulunur.

2- ICMP (INTERNET CONTROL MESSAĞE PROTOCOL)

Internet protokolünün control ve yönetimine yardımcı olan bir protokoldür. Bu protokol sayesinde network üzerindeki sorunlar kolaylıkla tespit edilebilmektedir. **RFC 792** standardı ile belirlenmiştir ve iki bilgisayar arasındaki iletişimde, hedef bilgisayarda, varsayılan ağ geçidinde veya routerlarda oluşan hatalar ICMP mesajı olarak kaynak bilgisayara bildirilir.

Farklı durumlara göre farklı hata mesajları vardır. Bunlardan bazıları şunlardır:

Hedefe Ulaşılamıyor: Kaynak bilgisayara datanın gönderilmesiyle ilgili bir problem olduğu bilgisi döner.

Zaman Aşımı: Gönderilen datanın hedefe ulaşması için gereken zamanın dolduğunu ve bu sebeple paketin yok edildiğini belirten mesajdır.

Source Quench: Kaynak bilgisayara yönlendirmeyi yapan cihazdan daha hızlı data gönderdiğini ve yavaşlaması gerektiğini belirtir.

Tekrar Yönlendirme: Bu mesajı gönderen yönlendirici hedef için daha iyi bir yola sahip yönlendiricinin var olduğunu belirtir.

Yankı: Ping komutu tarafından bağlantıyı onaylamak için kullanılır.

Parameter Problem: Parametrenin yanlış olduğunu belirtmek için kullanılır.

Address Mask Request / Reply: Doğru subnet Maskın öğrenilmesi için kullanılır.

Bölüm Sonunda ICMP detaylı incelenecektir.

3- BOOTP (BOOTSRAP PROTOCOL)

UDP tabanlıdır ve RARP protokolü gibi sunucu - istemci ortamında çalışır. IP adresi isteyen bilgisayarlar bu isteklerini bir broadcast ile bildirirler. BOOTP sunucu ise ip adresini, kendi ip adresini ve varsayılan ağ geçidi adresini bir broadcast ile networke gönderiri. İstemciler MAC adreslerine bakarlar ve kendi MAC adreslerini gördüklerinde bu bilgileri alırlar.

4- HTTP (HYPERTEXT TRANSFER PROTOCOL)

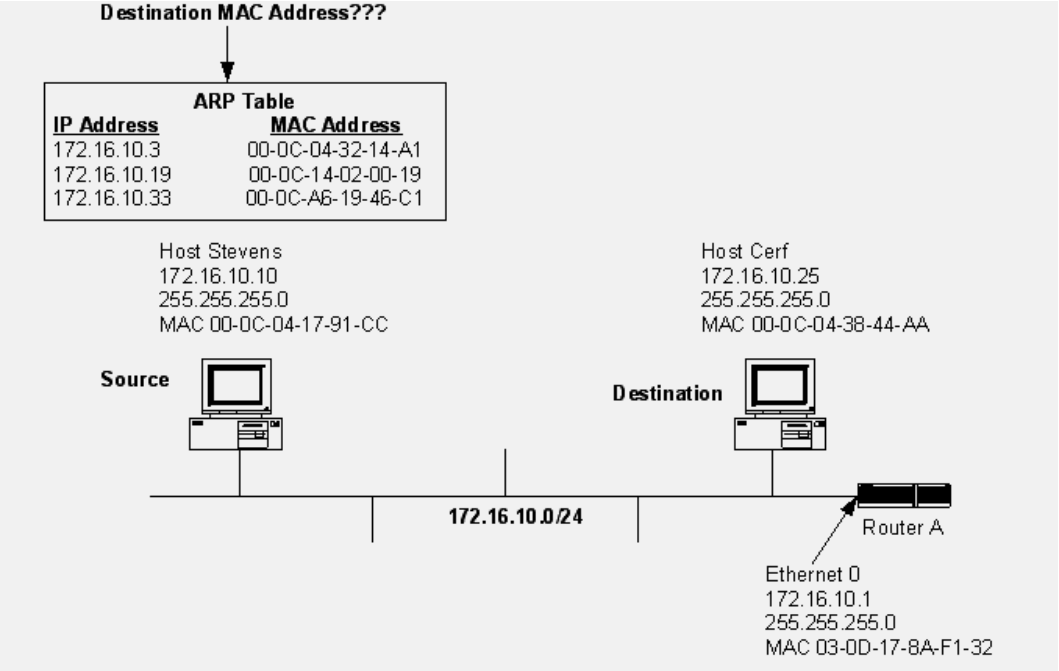
HTTP (HyperText Transfer Protocol – HiperMetin Aktarım Protokolü): HTTP, Internet'te bağlandığınız Web sayfalarının kodlarını aktarmak için kullandığımız protokoldür. Örneğin www.sistem.com.tr yazdığımızda, ilk olarak bu protokol alt seviyedeki protokollere bu adresin nereden, nasıl isteneceğini ve nasıl aktarılacağını söylemektedir.

5- RARP (REVERSE ADDRESS RESOLUTİON PROTOCOL)

Sabit diski olmayan aptal terminaller tarafından otomatik olarak ip adresi almak için kullanılan protokoldür. RARP istemci kendisiyle aynı segmentte bulunan RARP sunucudan ARP paket formatını kullanarak broadcast yapar ve ip adresi ister. RARP sunucu da uygun bir ip adresini istemciye gönderir.

6- ARP (ADDRESS RESOLUTİON PROTOCOL)

ARP protokolü ip adresi bilinen bir bilgisayarın MAC adresini öğrenmede kullanılır.



İki bilgisayar iletişim kuracağı zaman kaynak bilgisayar hedef bilgisayara MAC adresini sorar ve bu işlem **ARP Request** denen bir broadcast olan mesajla gerçekleşir. İlgili ip adresine sahip olan bilgisayar içinde MAC adresinin bulunduğu cevap paketini istemciye gönderir. Bu cevap mesajına **ARP Reply** denir. ARP protokolü Internet Katmanında çalışır. Kaynak bilgisayar ip adresi ve edindiği MAC adresini eşleştirerek ön belleğinde saklar. "ARP-a" komut satırı komutu ile ön bellekte bulunan MAC adresleri görüntülenebilir.

ARP aslında bir IP protokolü değildir ve dolayısıyla ARP datagramları IP başlığına sahip değildir. Varsayalım ki bilgisayarınız 128.6.4.194 IP adresine sahip ve siz de 128.6.4.7 ile iletişime geçmek istiyorsunuz. Sizin sisteminizin ilk kontrol edeceği nokta 128.6.4.7 ile aynı ağ üzerinde olup olmadığınızdır. Aynı ağ üzerinde yer alıyorsanız, bu Ethernet üzerinden direk olarak haberleşebileceksiniz anlamına gelir. Ardından 128.6.4.7 adresinin ARP tablosunda olup olmadığı ve Ethernet adresini bilip bilmediği kontrol edilir. Eğer tabloda bu adresler varsa Ethernet başlığına eklenir ve paket yollanır. Fakat tabloda adres yoksa paketi yollamak için bir yol yoktur. Dolayısıyla burada ARP devreye girer. Bir ARP istek paketi ağ üzerine yollanır ve bu paket içinde "128.6.4.7" adresinin Ethernet adresi nedir sorgusu vardır. Ağ üzerindeki tüm sistemler ARP isteğini dinlerler bu isteği cevaplandırması gereken istasyona bu istek ulaştığında cevap ağ üzerine yollanır. 128.6.4.7 isteği görür ve bir ARP cevabı ile "128.6.4.7 nin Ethernet adresi 8:0:20:1:56:34" bilgisini istek yapan istasyona yollar. Bu bilgi, alıcı noktada ARP tablosuna işlenir ve daha sonra benzer sorgulama yapılmaksızın iletişim mümkün kılınır.

Ağ üzerindeki bazı istasyonlar sürekli ağı dinleyerek ARP sorgularını alıp kendi tablolarını da güncelleyebilirler.

IP ADRESLEME

Bilgisayarlar veya diğer cihazlar Networkslere fiziksel olarak bağlanmanın yanında mantıksal olarak ta dahil olmalıdır. Bunun için aynı networkte ki iki cihazın aynı ip networkünde olması gerekir, yani Network adresleri aynı olmalıdır. Bilgisayarlara ip adresleri Statik veya Dinamik olarak verilebilir. Dinamik olarak ip adresi ataması için en güzel örnek DHCP Server' dir. Dynamic Host Configuration Protokolun kısaltması olan DHCP konfigürasyonu Router üzerinde de yapılabilmektedir. İlerleyen bölümlerde bir routerin nasıl DHCP server olarak konfigure edilebileceği anlatılmıştır.

IP protokolü, TCP/IP network'lerinde bilgisayarların birbiriyle iletişimi sürdürebilmek için kullandığı bir protokoldür. Pek çok kaynakta IP protokolü, TCP/IP protokolünün postacısı

seklinde tanımlanmıştır. IP protokolü network üzerinde paketlerin hangi host'a gideceğini belirtir. Bu, IP adreslemenin network katmanlarında (bu durumu açıklamak için ister OSI katmanını kullanalım; ister TCP/IP modelini) MAC adresleme ve ARP protokolüne göre daha sonra yer alan bir yapıya sahiptir.

IP adresleme de IP'nin bir görevi de bir veri paketinin sonsuza dek network üzerinde kalmamasını sağlamaktır. IP paketlerinin içinde TTL (Time To Live – Kullanım Süresi) seklinde tanımlanmış bir süre de yer almaktadır. Bu süre asıldığında paket artık network üzerindeki cihazlar tarafından ele alınmaz ve bu süreyi astıgı andan itibaren bir network kartına yada switch, router gibi bir cihaza ulaşırsa yok edilir..

IP adresleme ile yakalanabilen diğer bir avantaj da network'leri birbirine bölerken bir network'ten bir başka network'e belli tipte paketlerin geçmemesini sağlamaktır. Her paket hangi yazılıma ulaşacağı hangi port'ta ele alınacağı yada ne kadar süre ile hangi network'e ait olduğu gibi pek çok bilgi tek bir pakette taşınabildiği için bu paketlere çeşitli kısıtlamalar da getirmek olasıdır. IP filtreleme (IP filtering) denen bu olayın çok daha gelişmiş bir sekli, firewall adı verilen yazılım (ve donanım) grubu tarafından yapılmaktadır. Firewall'lar bir bilgisayara nasıl ulaşılacağından hangi paketin bilgisayara gireceğine kadar her tür işlem için ayarlanabilirler ve bir IP paketinin farklı yollar kullanarak, bir bilgisayardan bir başka bilgisayar ulaşması sağlanabilir. Böylece network üzerinde olan trafiğin kontrol edilmesi yada bazı özel network'lerin bazı paketlerden arındırılması sağlanabilir.

Burada Alt Ağ Maskesi ifadesi dikkatinizi çekmiştir. Başka bir deyişle genellikle Türkçe'mizde de kullanılan subnet Mask. subnet Mask bizim için önemli, çünkü daha ileride değineceğimiz bir networku alt networklere ayırabilmemiz için subnet Mask ile oynamamız gerekecek. Çünkü subnet Mask ile ip adresi binary durumda and işlemine sokulduğunda network adresini verir. Ip adresler 4 oktetten ve her oktette 8 Bitten oluşur.

	1st octet	2nd octet	3rd octet	4th octet
172.0.0.0	Network	Host	Host	Host
Subnet Mask: 255.0.0.0 or /8	255	0	0	0
192.4.0.0	Network	Network	Host	Host
Subnet Mask: 255.255.0.0 or /16	255	255	0	0
192.168.1.0	Network	Network	Network	Host
Subnet Mask: 255.255.255.0 or /24	255	255	255	0

	1st octet	2nd octet	3rd octet	4th octet
172.0.0.0	Network	Host	Host	Host
Subnet Mask	11111111	00000000	00000000	00000000
192.4.0.0	Network	Network	Host	Host
Subnet Mask	11111111	11111111	00000000	00000000
192.168.1.0	Network	Network	Network	Host
Subnet Mask	11111111	11111111	11111111	00000000

İlk şekilde subnet masklar ikinci şekilde o subnet maskların Binary gösterimi mevcut. İlk resimde ki /8, /16, /24 gibi ifadeler görüyorsunuz. Bunlar subnet Maski ifade eder, daha doğrusu subnet Maskin Binary gösterimi içindeki toplam 1 sayısıdır.

Bir networkteki ilk ip adresi o networkun network adresini ve son ip adresi de Broadcast adresidir. Bu adresler network cihazlarına atanamaz.

IP HESAPLARI VE SUBNETTING

TCP/IP protokolünde tüm bilgisayarlar 32 bitlik "özgün" bir IP numarasına sahip olacak şekilde adreslenirler.

IP adresleri sınıflara ayrılmıştır, bu sınıflar şunlardır;

Class A : 0.0.0.0 - 127.255.255.255 arasındaki ip adresleri.

Class B: 128.0.0.0 - 191.255.255.255 arasındaki ip adresleri.

Class C: 192.0.0.0 - 223. 255.255.255 arasındaki ip adresleri.

Class D: 224.0.0.0 - 239. 255.255.255 arasındaki ip adresleri.

Class E: 240.0.0.0 – 255. 255.255.255 arasındaki ip adresleri.

Her ip sınıfının subnet maskıda belirlenmiştir buna göre;

A sınıfı için subnet mask: 255.0.0.0,

B sınıfı için subnet mask: 255.255.0.0,

C, D, E sınıfları için subnet mask: 255.255.255.0 'dır.

NOT: Bir ip adresi yada protokol sınıfından bağımsız olarak bir subnet mask ile çalışıyor veya çalışabiliyorsa "classless" aksi durumda "classfull" denir.

Bilgisayarımızdan komut sistemini açıp **"ipconfig /all"** komutunu verdiğimizde kullandığımız bilgisayarın ip konfigürasyonunu görebiliriz.

Peki ip adreslerinin özel olması gerektiğine göre bütün dünyada bu ip adresinin aynısı kullanan bir başka bilgisayar yok mu ? Gerçekten de böyle olsaydı mevcut ip adreslerimiz çoktan bitmiş olurdu. Belki de bunu önlemek için bazı ip aralıkları iç networkte kullanılmak üzere ayrılmıştır ve herhangi bir şekilde dış networkte (internette) kullanılamaz.

Bu ip aralıkları şunlardır:

10.0.0.0 – 10.255.255.255

172.16.0.0 -172.31.255.255

192.168.0.0 – 192.168.255.255

Internet ortamında bu ip adresleri kesinlikle kullanılmaz, iç network kullanıcıları internete çıkarken, ISP tarafından sağlanan static veya dynamic bir ip adresine dönüşürler. İşte bu ip adresi tüm dünyada tek olacaktır.

Burada aklımıza söyle bir soru gelebilir; Neden özel olarak ayrılmış ip sınıflardan kullanmalıyım, sözgelimi benim 212.212.212.212 gibi bir ip adresi kullanmama engel olan şey nedir?

Eğer firmanız internete hiçbir şekilde çıkmıyorsa istediğiniz ip adresini kullanabilirsiniz fakat çıkıyorsa bu ip adresi belki de sizin o an ziyaret etmek istediğiniz bir sitenin ip adresi olabilir ve siz browser'ınız a sitenin adını yazdığınız da bir sonuç alamazsınız. Zira ip adresi sizinle aynı networkte.

Yerel networkler de ip adresi manuel olarak static konfigüre edilebileceği gibi örneğin DHCP gibi bir yazılımla dinamik olarak da dağıtılabilir.

Ip adreslerinin dağıtılması sırasında subnet maskların standar verilmesi ciddi sorunlara sebep olacaktı. Örneğin bir ISP firması sözgelimi 150 adet ip adresi almak istiyorsunuz. Bu durum standart subnet mask kullanılarak size verilebilecek minimum ip sayısı 255'dir. Daha vahim bir senaryo ise siz sözgelimi 500 tane ip adresi isteseniz ortaya çıkar çünkü o zaman size verilebilecek minimum ip sayısı 255*255 = 65025 ' dir.

Bunun önüne geçebilmek için yapılabilecek tek şey ise subnet masklar ile oynamaktan geçer. Bu sayede networkler sub-networklere bölünebilir ve ip israfın biraz olsun azalabilir.

Subnet mask IP adresinin mask kısmını oluşturur. Böylece TCP/IP, Network adresi ile TCP/IP adresini birbirinden ayırır. Bu sayede Network ID ve Host ID birbirinden ayırt edilir. Örneğin: 255.255.0.0 TCP/IP host'u iletişime başladığında; subnet mask host'un yerel mi yoksa uzak (remote) olduğunu belirtir.

Örnek:

Elinizde adresi 192.168.1.0 olan C Class bir network var ve bunu 4 subnetworke bölmek istiyorsunuz. Son oktete ikili sistemde (00000000) sıfır vardır. Onluk sisteme dönüşürken 2^8 =256 yapar.

Bu durumda 256/4 = 64 'er tane ip adresiniz olacak.

Subnet Maskın son oktetini 256-64 yaparsanız bunu gerçekleştirmiş olursunuz. Bu durumda subnet mask=255.255.255.192 olacaktır ve elimizde subnet maskı 255.255.255.192 ve network adresleri sırasıyla;

192.168.1.0

192.168.1.64

192.168.1.128

192.168.1.192

Olan 4 adet networkümüz, her networkte 64'er tane ip adresimiz olacak.

Bir networkün ilk ip adresi network adresini, son ip adresi broadcast adresini gösterdiği için kullanılamaz dolayısıyla bir networkte "useable" olarak adlandırılan, yani kullanılabilecek ip sayısı toplam ip sayısından 2 eksiktir. Useable Ip sayısı = toplam ip sayısı – 2

Network adresleri örneğin /24 şeklinde gösterilebilirler. /24 ip adresinin binary yazılımında soldan sağa 24 tane 1 olduğu anlamına gelir. Bu şekilde yazılımına CIDR denir. (Classless)

örneğin;

255.255.255.0 binary olarak

11111111.11111111.11111111.00000000'e eşikttir ve 24 tane 1 den dolayı /24 olarak gösterilebilir.

Yükarıdaki örneğimizdeki subnet mask ise binary olarak;

11111111.11111111.11111111.11000000 'a eşit olacak dolayısıyla /26 olarak gösterilebilecektir.

Örnekler:

Subnet Mask	Binary Yazılım	CIDR ifade
255.255.128.0	11111111.11111111.100000000.00000000	/17

255.255.255.128 11111111.11111111.111111111.01000000 /25

255.255.255.252 11111111.11111111.111111111.11111100 /30

Elimizde bir ip adresi ve onun subnet maskı varsa ikisinin binary yazılısını AND'leyerek network adresini bulabiliriz.

örneğin elimizde subnet maskı 255.255.255.128 olan 192.168.1.141 gibi bir ip var.

192.168.1.141 = 11000000.10101000.00000001.10001101

255.255.255.128 = 11111111.11111111.11111111.10000000

AND (çarpıyoruz) =11000000.10101000.00000001.10000000

Network Adresi = 192. 168. 1. 128

Network Adresi (ID):

Bir grup bilgisayarı temsil eden ve o grupta bulunan bütün bilgisayarlarda aynı olan adrese network (ağ) adresi denir. Kesinlikle IP adresi olarak bir cihaza atanamaz.

IP numarasının hangi network'te bulunduğunu gösteren kısmına **Network ID** ve geri kalan bilgisayarların adreslendirilmesi için kullanılan kısma ise **Host ID** olarak isimlendirilir. Bu sistemi sokak numarası, kapı numarası örneğine benzetebiliriz. Nasıl bir sokaktaki tüm evlerin kapı numaraları ayrı, sokak numaraları aynı olmak zorunda ise, bir IP network'ündeki tüm host'ların da, IP adreslerinin host ID bölümleri ayrı, Network ID bölümleri aynı olmak zorundadır.

Subnet Mask içindeki 1'ler IP numaramızın Network ID'sini kalan sıfırlar da Host ID'sini belirlemede kullanılmış olur.

	1st octet	2nd octet	3rd octet	4th octet
172.0.0.0	Network	Host	Host	Host
Subnet Mask: 255.0.0.0 or /8	255	0	0	0

Broadcast Adresi

Network adresinin en üst sınırıdır. Herhangi bir ağda bütün adresleri temsil etmek için kullanılan adreslere Broadcast adres denir. Bir network'teki tüm host'lara gönderilmek istenen paketler bu adrese yönlendirilirler. Broadcast adresinin üç bölümünün bütün bitleri ağ adresinin tersine 1'dir. Bu adreslerde kesinlikle IP adresi olarak bir cihaza atanamaz.

Network (ağ) Adresi (ID) Nasıl Bulunur?

Network adresi, ip adresi ile ip adresinin subnet maskı (ağ maskesi) AND (ve) işlemine tabi tutularak bulunur. IP adresi ve subnet mask ikilik sayı sistemine çevirilerek binary alt alta olarak yazılır, her iki tarafda **'1'** bit değerine sahip bölümler aynen (yani **1** olara) aktarılır, diğerleri ise **0** (sıfır) olarak yazılır.

Örnek

Ip Adresi: 132.15.78.202 = **1**0000**1**00. 0000**1111**. 01001110. 11001010

Subnet Mask: 255.255.0.0 = **1**1111**1**11. 1111**1111**. 00000000. 00000000 .

Network ID: 132.15.0.0 **1**0000**1**00. 0000**1111.** 00000000. 00000000

Burada **1**0000**1**00 ikilik sistemdeki sayının onluk sistemdeki karşılığı 132, 0000**1111** sayının onluk sistemdeki karşılığı 15 dir.

Broadcost adresi : **1**0000**1**00. 0000**1111. 11111111. 11111111** sayılarının onluk sistemde karşılığı: 132.15.255.255 dir.

128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0

Onluk sisteme çevirirken bitlerin karşılığı

NOT: Subnet masklarda bir kez sıfır rakamı kullanıldıktan sonra ardından 1 rakamının bir daha kullanılamaz. Yani bir Subnet Mask sadece birbiri ardına gelen 1'ler ve sonra yine birbiri ardına gelen 0'lardan oluşabilir.

NOT: Bir bilgisayar başka bir bilgisayarla iletişime geçeceği zaman, karşı bilgisayarın **IP** adresi ile kendi **SUNET MASK**'ını **AND** işleminden geçirerek network adresini bulur. Kendi network adresi ile iletişime geçeceği bilgisayarın network adresi aynı ise o zaman bağlantı kurulabilir.

Subnettig Örnek Çalışma

Örnek- A

1) Aynı lokalde bulunan 192.168.10.17 ip adresli bilgisayar, 192.168.10.18 ip adresili bilgisayara ulaşmak istesin ve her ikisinin subnet maskı: 255.255.255.240 olsun

192.168.10.17 ip'li bilgisayar, 192.168.10.18 ip'li bilgisayara ulaşmak istesin.

192.168.10.17 ip'li bilgisayarın ağ adresi:

İp : 192.168.10.17 = **11**000000. **10101**000. 0000**101**0. 000**1**0001

Subnet mask : 255.255.255.240 = **11**111111. **111**11111. 1111**111**1. 111**1**0000

Ağ Adresi 11000000. 10101000. 00001010. 00010000

Onluk sistemdeki karşılığı= **192.168.10.16**

192.168.10.16 ip'li bilgisayarın ağ adresi:

İp : 192.168.10.18 = **11**000000. **10101**000. 0000**101**0. 000**1**0010

Subnet mask : 255.255.255.240 = **11**111111. **111**11111. 1111**111**1. 111**1**0000

Ağ Adresi **11**000000. **10101**000. 0000**101**0. 000**1**0000

Onluk sistemdeki karşılığı= **192.168.10.16**

192.168.10.17 ve 192.168.10.18 ip adresli bilgisayarlar aynı ağ adresine sahipler. Dolayısıyla aynı ağdalar demektir. O halde ağ geçidine gitmeye gerek yok. Haberleşme sağlanabilir.

2) Farklı lokalde bulunan 192.168.10.17 ip adresli bilgisayar, 192.168.10.33 ip adresili bilgisayara ulaşmak istesin ve her ikisinin subnet maskı: 255.255.255.240 olsun

192.168.10.17 ip adresli bilgisayarın ağ adresini bulmuştuk : **255.168.10.16**

…17 ip adresli bilgisayar, …33 ip adresli bilgisayara gitmek istediğinde …33 ip'li bilgisayarı ve kendi subnetini ADN (ve) işlemine tabi tutarak hangi ağ adresinde olduğunu çözecek:

İp Adresi : 192.168.10.33 = **11**000000. **10101**000. 0000**101**0. 00**1**00001

Subnet mask : 255.255.255.240 = **11**111111. **111**11111. 1111**111**1. 11**1**10000

Ağ Adresi **11**000000. **10101**000. 0000**101**0. 00**1**00000

Onluk sistemdeki karşılığı= **192.168.10.32**

….17 ip'li PC …33 ip'li PC'nin ağ adresini buldu ve bu ağ adresi ile kendi ağ adresi aynı olmadığını gördü. O halde direk ağ geçidine gider router de gerekli yardımı sağlar.

Not: Router farklı ağ adreslerine sahip bilgisayarların haberleşmesini sağlar. Aynı ağ adresine sahip olan bilgisayarlar, router'a müracat edip haberleşme ihtiyacını hissetmezler.

Örnek –B

1) Şimdi yine farklı lokallerde bulunan …17 ip li PC, …33 ip li PC'ye gitmek istesin. Fakat bu kez ağ maskeleri 255.255.255.0 olsun

İp Adresi : 192.168.10.17 = **11**000000. **10101**000. 0000**101**0. 00010001

Subnet mask : 255.255.255.0 = **11**111111. **111111**11. 1111**111**1. 00000000

Ağ Adresi **11**000000. **10101**000. 0000**101**0. 00000000

Onluk sistem: **192.168.10.0**

İp Adresi : 192.168.10.33 = **11**000000. **10101**000. 0000**101**0. 00100001

Subnet mask : 255.255.255.0 = **11**111111. **111111**11. 1111**111**1. 00000000

Ağ Adresi =**11**000000. **10101**000. 0000**101**0. 00000000

Onluk sistem: **192.168.10.0**

…17 ip li PC, …33 ip li PC'ye gitmek istedi ve …33 ip li PC'nin ağ adresini hesapladı ve aynı olduğunu gördü. Aynı lokalde olduğunu zannetti fakat yanıldı. Çünkü bu adresler farklı alt ağlardalardı. Aynı ağ adresi çıktığı içinde router'a müracat etti.

BİR IP ARALIĞINI ALT AĞLARA BÖLME

Bilgi:

2^n: İki üzeri n diye okunur.

>=: Büyük eşit

Ortak Ağ Maskesi ve Alt Ağların Ip Sayısını Bulma

Ortak ağ maskesi bulunurken (**2^n>=alt ağ sayısı**) formülü ile bulunur. Alt ağ sayısı bizim ip aralığını kaç alt ağa böleceğimize göre değişir.

Bu formül ile n değeri bulunur. Bu n değeri alt ağ bitini verecektir. Bulunan bu bit ip in dahil olduğu sınıfın alt ağ maskesine eklenir ve yeni ağ maskesi (subnet mask) bulunur

Alt ağlardaki ip sayısı (**2^m-2**) formülüyle bulunur. m değeri oktette kalan 0 (sıfır) sayısı kadardır.

Örnek Çalışma

192.168.0.0/24 ip aralığını 2 subnet (alt ağ) olacak şekilde ayarlayalım.

2^n>=2 ⇨ burada 2 değerini 1 verir. O halde ağ maskesine bir tane 1 ekleyeceğiz.

Yükardaki ip C sınıfı bir ip olduğu için ağ maskesi 255.255.255.0 dır.

Ağ maskesinin ikilik sistemde yazılışı

11111111.11111111.11111111.00000000

\+ **1** **.**

11111111.11111111.11111111.**1**0000000 Onluk Sis.= 255.255.255.128

Son oktetten kalan sıfır sayısı 7 dir. Yani m değeri m = 7 dir.

2^m-2 ⇨ 2^7-2 = 126

Tek subnette 126 ip, 2 subnet olduğu için 252 ip atanabilir.

İlk Subnet: 192.168.0.1 den 192.168.0.126 kadar

192.168.0.0 Network ID olur ve kullanılmaz

192.168.0.127 Broadcast adresi olur ve kullanılmaz.

İkinci Subnet: 192.168.0.129 dan 192.168.0.254 e kadar

192.168.0.128 Network ID olur (Kullanılmaz)

192.168.0.255 Broadcast adresi olur. (Kullanılmaz)

NOT: IP hesapları CCNA sınavına hazırlanan öğrenciler için son derece önemlidir. CCNA sınavlarında ip hesaplarıyla direk ilgili yada içerişinde ip hesaplarını içeren bol sayıda soru çıkmaktadır.

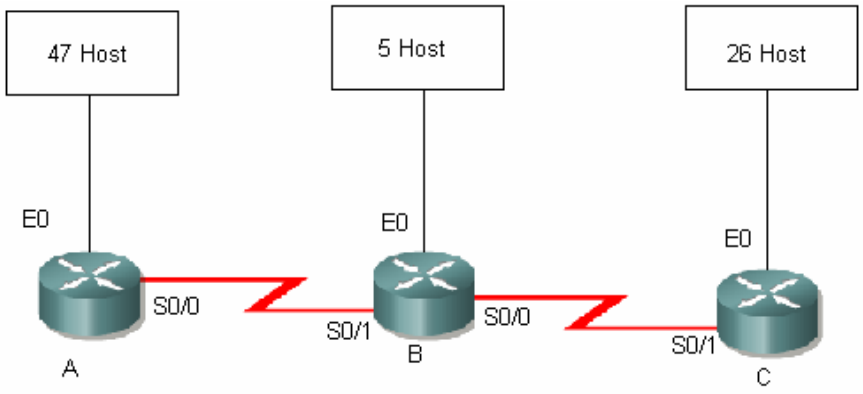
NOT: Routerın Ethernet interface' ine ip adresi atanırken önerilen networkün ilk ip adresini atamaktır.

CLASSFULL - CLASSLESS IP ADRESLERİ

Classfull adreslerde subnet masklar ip adresinin hangi sınıfa ait olduğuyla direk ilgilidir. İp adreslerinin ilk oktetleri sınıflarını belirlerler ve her sınıf için subnet mask belirlenmiş durumdadır. Örnek vermek gerekirse 10.x.x.x gibi bir ip adresi A sınıfı bir ip adresidir ve Classfull olarak çalışan bir sistem de bu adresin subnet maskı her zaman 255.0.0.0 olacaktır. Routing protokoller anlatılırken detaylı değinilecek **Rip ve Igrp protokolleri** Classfull protokollerdir ve subnet maskı sınıflarına göre kendileri belirlerler.

Classless adreslerde ise subnet mask, sınıftan bağımsızdır. Söyle ki 10.x.x.x gibi bir ip adresine istendiğinde 255.255.255.0 gibi bir subnet mask verilebilir ve Classless olan sistemlerde bunu algılarlar. **Ospf, Eigrp gibi protokoller classless'** dır. **Classless adreslemeye VLSM (Variable Length subnet Mask) veya CIDR (Classless Inter Domain Routing) denir.**

IP SUBNETTING Örnek Çalımsa



Elimizde 192.168.1.0 networkü var ve bu networkün 192.168.1.0 /25 lik kısmı daha sonra kullanılmak üzere ayrılmış durumda. Kalan ip adreslerini uygun şekilde dağıtmamız gerekiyor.

A, B, C Routerlarının Ethernet Interface'lerine bağlı 3 network ve router'ların birbirleriyle bağlantısında oluşan 2 (2'şer useable ip gereken) network olmak üzere elimiz toplam 5 network var.

Burada ilk yapmamız gereken host sayılarına bakarak kaçar ip içeren networklere böleceğimize karar vermek.

A Routerı için 64,

C Routerı için 32,

B Routerı için 8 ve

Routerlar arasında ki networkler için 4'er ip içeren gruplar olmalı. Dolayısıyla A routerı için 192.168.1.128 /26 networkü kullanılmalı. Çünkü 192.168.1.0 dan 192.168.1.127 ' ye kadar olan ip ler daha sonra kullanılmak üzere ayrılmış durumda.

C routerı için 192.68.1.192 /27

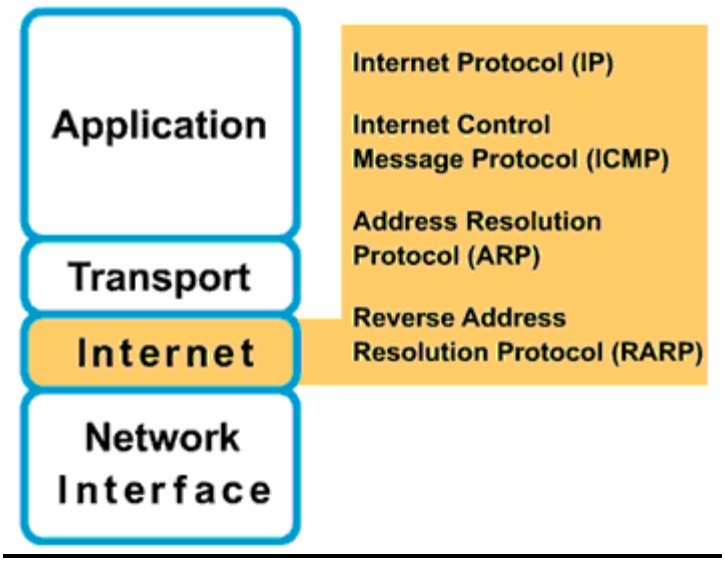
B Routerı için 192.168.1.224/29

diğer networkler içinde 192.168.1.232 /30 ve 192.168.1.236 /30 networkleri kullanılmalıdır.

Network aralıklarımızı detaylı incelersek;

Son Oktet	Yeri	İçerdiği Ip Sayısı	Network Adresi	S.M.
0 – 127	Ayrılmış	128	192.168.1.0	/25
128-191	A Routerı	64	192.168.1.128	/26
192-223	C Routerı	32	192.168.1.192	/27
224-231	B Routerı	8	192.168.1.224	/29
232-235	A-B Arası	4	192.168.1.232	/30
236-239	B-C Arası	4	192.168.1.236	/30

ICMP (INTERNET CONTROL MESSAGE PROTOCOL)



Üçüncü katman yani Internet yada Network katmanı olarak adlandırdığımız katman IP bazından yönlendirmenin yapıldığı katmandır. IP data iletimi ve yönlendirme için belki de en iyi çözümdür.

Fakat IP ile ilgili datanın iletimi sırasında herhangi bir sebeple fail olma durumu olduğunda bu durumu kontrol edecek hata mesajlarına sahip olmaması gibi bir sorun vardır. Sözgelimi yanlış konfigürasyonlar, donanımsal arızalar yada Routing Table' lar ile ilgili sorunlarda IP bir hata mesajı döndürmez.

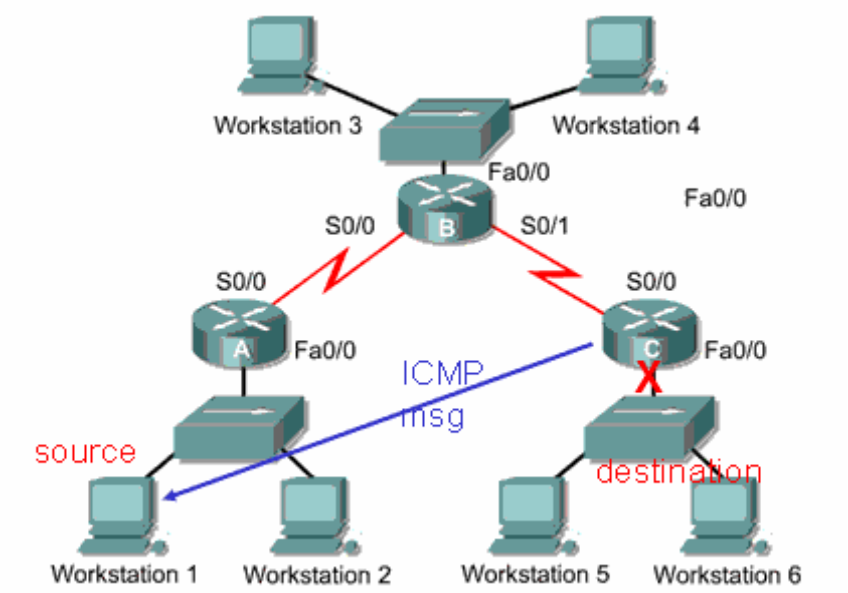
ICMP, IP' nin bu açığını gidermek üzere geliştirilmiş bir protokoldür. Bahsedilen durumlarda ICMP ilgili bir masajı dondurur ve problem çözümlerde Network Mühendislerine yardımcı olur.

Ancak burada IP bazında iletimin güvenilir olmadığını, ICMP mesajları ile bu güvenilirliğin sağlandığını söylemek yanlış olur. Datanın güvenilir şekilde iletilmesi bir üst katman olan Transport katmanı ve bu katmanda çalışan protokoller tarafından sağlanmalıdır.

Genel olarak ICMP mesajları iki ana başlık altında incelenebilir.

1. Hata Raporlama mesajları

2. Durum Kontrol mesajları



Sözgelimi Workstation1 den Workstation6 ya bir data gönderildiğini ve bu C Routerinin da Fa0/0 interface' inin down olduğunu varsayalım. Bu durumda C routeri datanın ulaştırılamadığı ile ilgili bir mesajı geri döndürecektir. Burada bu bilgi sadece kaynağa yani Workstation1 e gönderilecektir.

ICMP mesajları kendi frame yapısına sahip değildir. Bu mesajlar IP ile enapsule edilmiş frameler içerişine gömülmüşlerdir. Dolayısıyla ICMP mesajları tarafından oluşturulmuş hata mesajları kendi ICMP mesajlarını yaratmazlar.

ICMP mesajları Type' lardan ve Code' lardan oluşur.

```
Type 3: Destination Unreachable

Codes
0  Net Unreachable
1  Host Unreachable
2  Protocol Unreachable
3  Port Unreachable
4  Fragmentation Needed and Don't Fragment was Set
5  Source Route Failed
6  Destination Network Unknown
7  Destination Host Unknown
8  Source Host Isolated
9  Communication with Destination Network is Administratively Prohibited
10  Communication with Destination Host is Administratively Prohibited
11  Destination Network Unreachable for Type of Service
12  Destination Host Unreachable for Type of Service
13  Communication Administratively Prohibited
14  Host Precedence Violation
15  Precedence cutoff in effect
```

(Type 3, Hedefe ulaşılamıyor mesaj code'ları)

Ping ve Trace

Ping ve Trace komutları network mühendislerine bir çok problemin teşhisinde yardımcı olar. Her iki komutta ICMP Echo Reuest ve ICMP Echo Reply mesajları ile çalışır.

Ping komutu ile ping isteğini gönderen cihaz ICMP Echo Request' te bulunur. ICMP mesajlarındaki Echo Request Type'i 8 ve Code' u 0' dir.

Hedef ip adresi Echo Request mesajını aldığında gönderen cihaza Echo Reply ICMP mesajını gönderir. Bu mesajın Type'i 0 ve Code'u da 0' dir.

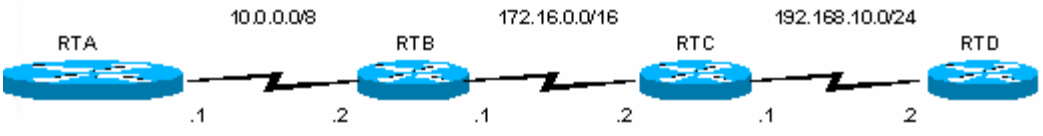
Trace komutu ise kaynak ve hedef ip adresleri arasında ki olası problemleri anlamaya yarar. Burada olası problemler dememizin sebebi kaynak ve hedef ip adresleri arasından birden fazla yol varsa her defasından farklı yollar izlenebilir.

Trace komutu bilgisayarlarda, tracert (ip adresi)

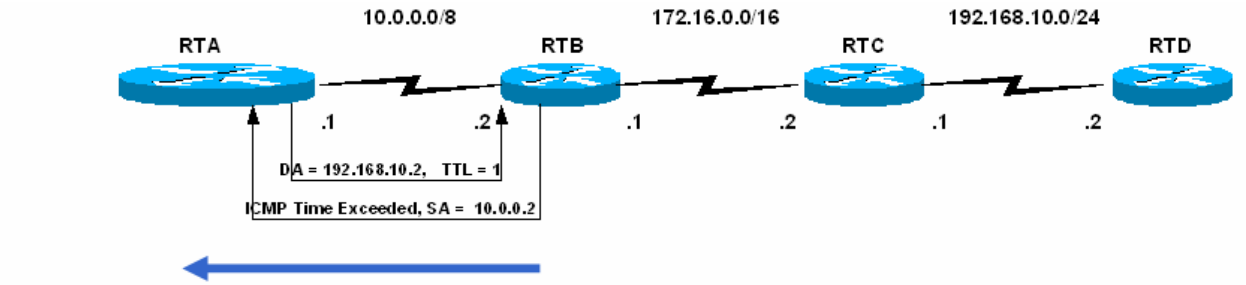
Routerlarda, traceroute (ip adresi)

şeklinde kullanılır. Traceroute çalışırken ping (ICMP Echo) mesajlarını kullanır.

Traceroute Örneği

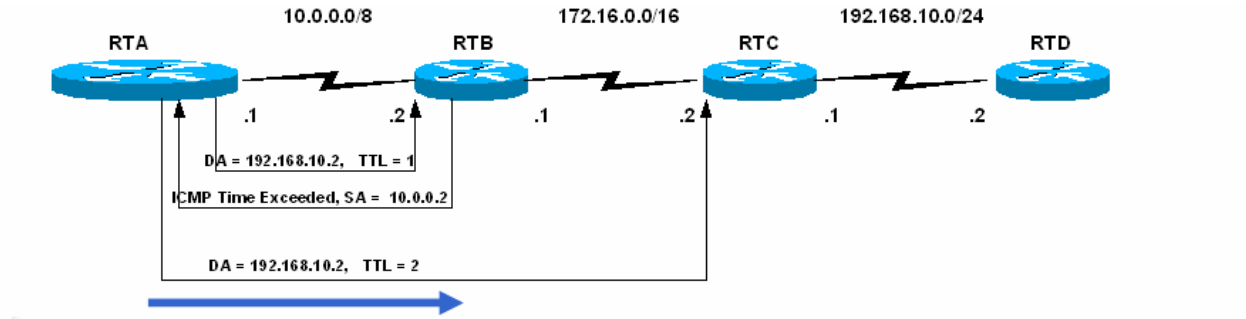



```
RTA# traceroute 192.168.10.2
```



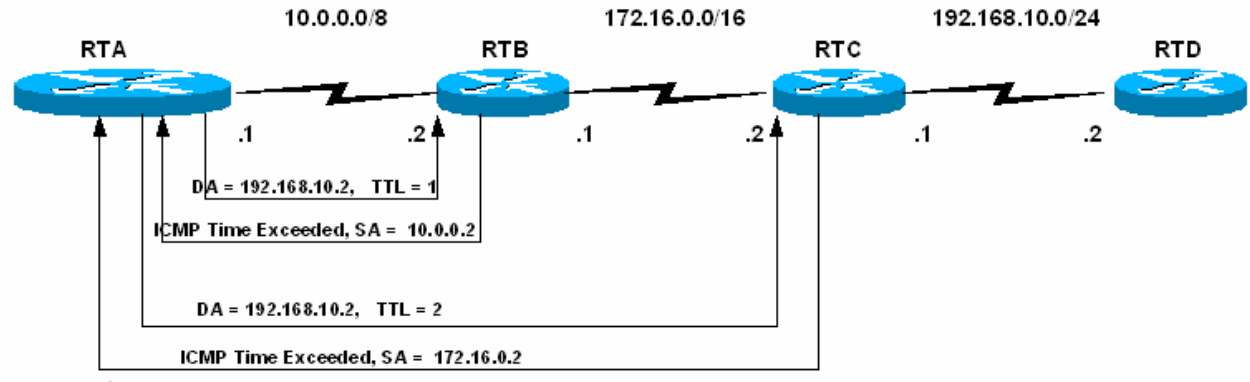
Data Link Header (Layer 2)			IP Header (Layer 3)	ICMP Message - Time Exceeded					DataLink Tr.
Data Link Destination Address	Data Link Source Address	....	**Source IP Add.** 10.0.0.2 **Dest. IP Add.** 10.0.0.1 **Protocol field** 1	**Type** 11 **Code** 0	Chk sum	ID	Seq . Num.	Data	FCS

Traceroute başladıktan sonra IP başlığındaki TTL değerini 1 yaparak ICMP Echo Requestte bulunur. RTB bu isteği aldığı zaman TTL değerine bakar ve bu değer 1 ise bir sonraki Routera gönderir, 0 ise İstek Zaman Asimi mesajını geri gönderir.Bu durumda RTA İstek zaman asimi mesajını aldıktan sonra TTL değerini 1 artırarak yani 2 yaparak yeni bir Echo Requestte bulunur.



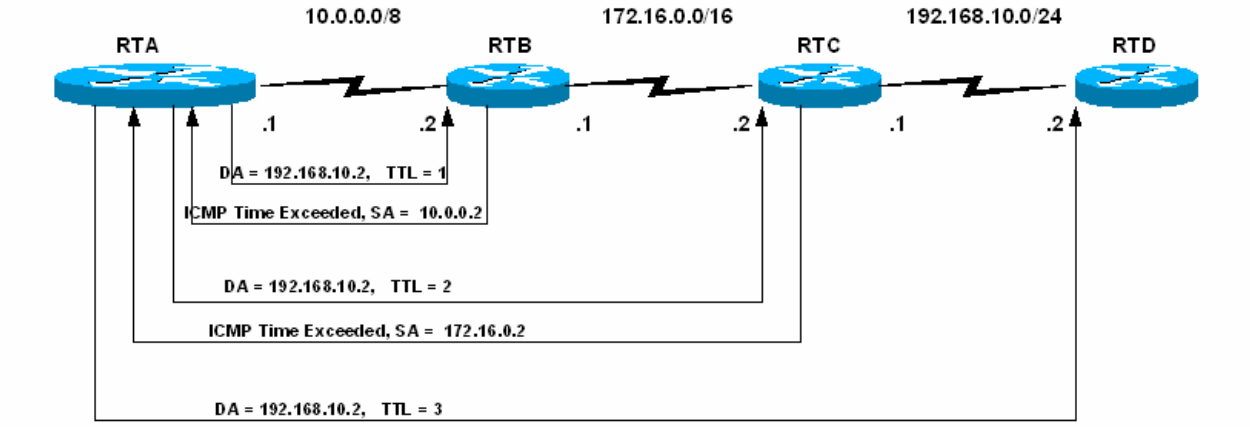
Data Link Header (Layer 2)			IP Header (Layer 3)	ICMP Message - Echo Request (trace)					UDP (Layer 4)	DataLink Tr.
Data Link Destination Address	Data Link Source Address	.....	**Source IP Add.** 10.0.0.1 **Dest. IP Add.** 192.168.10.2 **Protocol field** 1 **TTL** 2	Type 8 Code 0	Chk sum	ID	Seq. Num	Data	**DestPort** 35,000	FCS

Artık RTB mesajı alıp TTL değerine baktığında 0 değil 1 görecek ve dolayısıyla bu isteği RTC routerina gönderecektir. RTB ile yaşananlar bu sefer RTC ile de yaşanacak ve TTL değeri 0 olarak gelen Echo Requeste RTC İstek zaman asimi mesajını geri gönderecek. Burada RTC nin döndüreceği istek zaman asimi mesajında source ip adresi olarak RTC' nin adresi görünecektir.



Data Link Header (Layer 2)			IP Header (Layer 3)	ICMP Message - Time Exceeded					DataLink Tr.
Data Link Destination Address	Data Link Source Address	....	**Source IP Add.** 172.16.0.2 **Dest. IP Add.** 10.0.0.1 **Protocol field** 1	**Type** 11 **Code** 0	Chk sum	ID	Seq. Num.	Data	FCS

```
RTA# traceroute 192.168.10.2
Type escape sequence to abort.
Tracing the route to 192.168.10.2

  1 10.0.0.2 4 msec 4 msec 4 msec
  2 172.16.0.2 20 msec 16 msec 16 msec
```



a Link Header yer 2)			IP Header (Layer 3)	ICMP Message - Echo Request (trace)					UDP (Layer 4)
a Link tination lress	Data Link Source Address	.....	**Source IP Add.** 10.0.0.1 **Dest. IP Add.** 192.168.10.2 **Protocol field** 1 **TTL** 3	Type 8 Code 0	Chk sum	ID	Seq. Num	Data	**DestPort** 35,000

Bu sefer RTA TTL değerini 3 e çıkararak yeni bir Echo Requestte bulunacaktır.Dolayısıyla paket RTD routerina kadar gidebilecektir. Burada TTL değerini 0 olarak alan RTD hedef ip adresi kendine direk bağlı olan networkte bulunduğu için artık istek

zaman asimi mesajı göndermez, ICMO Port Unreachable Mesajını geri dondurur. (Type=3, Code=3) RTA routeri port unreachable mesajını trace ettiği network olarak algılar.

```
RTA# traceroute 192.168.10.2
Type escape sequence to abort.
Tracing the route to 192.168.10.2

1 10.0.0.2 4 msec 4 msec 4 msec
2 172.16.0.2 20 msec 16 msec 16 msec
3 192.168.10.2 16 msec 16 msec 16 msec
```

ROUTER

Network katmanında bulunan ve temel işlevi farklı networklere erişimde en iyi yol seçimini (Best Path Determination) yapan cihaza Router denir.

Bir router, üzerinde taşıdığı **routing table** denilen bir tablo sayesinde, bağlı olduğu herhangi bir segment üzerindeki tüm adresleri bilir. Router'ın bir tarafında bir ATM WAN'ı ve diğer bir tarafında da bir ofis içi Ethernet LAN'ı olabilir. Kısaca router iki farklı network yapısını yada iki farklı network segmentini birleştirmek için kullanılır.

Router'lar sadece üzerlerinde tam bir adres olan veri paketlerinin iletilmesini sağlar. Bazı durumlarda paketin tüm network'teki bilgisayarlara ulaşması için, bilgisayarlar, header'ında bir adres olmayan veri paketleri atarlar. Bu tip veri paketlerinin kısıtlı bir bant genişliğine sahip WAN'a çıkması, router tarafından engellenir. Router'lar, farklı veri aktarım teknolojisi kullanan network'leri (ATM, Ethernet, gibi..) birleştirebilirler.

ROUTER BİLEŞENLERİ

RAM: Random Access Memory' nin kısaltmasıdır. Routerın running-configuration adı verilen ve çalıştığı andaki konfigürasyonunu içeren bilgileri bulundurur. Bazı kaynaklarda RAM' a Dinamik RAM anlamında DRAM, running-configuration dosyasına da active-configuration denir. Router kapatıldığında yada yeniden başlatıldığında RAM' de bulunan bilgiler silinir.

ROM: Read Only Memory' nin kısaltmasıdır. Yani sadece okunabilir kesinlikle silinemez ve değiştirilemez. ROM' un ayrı başlıklarda incelenmesi gereken bileşenleri vardır. Bunları söyle sıralayabiliriz;

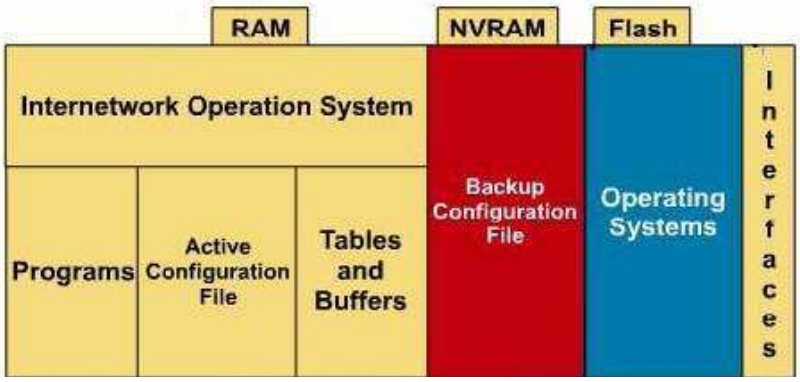
Post; Router' ın power tuşuna basıldığı anda devreye girer ve donanım testini gerçekleştirir.

MiniIOS; Konsoldan giriş yapılarak ulaşılabilecek, IOS' de bir sorun ile karsılaştığımızda sorun çözmemize yetecek kadar içeriğe sahip bölümdür. Burada TFTP servera erişilerek çeşitli yüklemeler yapılabilir.

Boostrap; Router' ın çalışmasını sağlayan bir yazılımdır. Microsoft işletim sistemlerindeki "boot.ini" dosyasına benzetilebilir.

ROM Monitör; Router' ın BIOS' u gibi düşünülebilir. Düşük seviyede hata ayıklama ve özelliklede ileride detaylı anlatacağımız şifre kırma işlemlerinde kullanılır. Kısaca Rommon olarak adlandırılır.

FLASH: Silinebilir, değiştirilebilir, yeniden yüklenebilir (EEPROM) bir hafıza kartıdır. IOS burada bulunur. Flash üzerine yüklemeler yapmak için TFTP Server adındaki programdan faydalanılır.

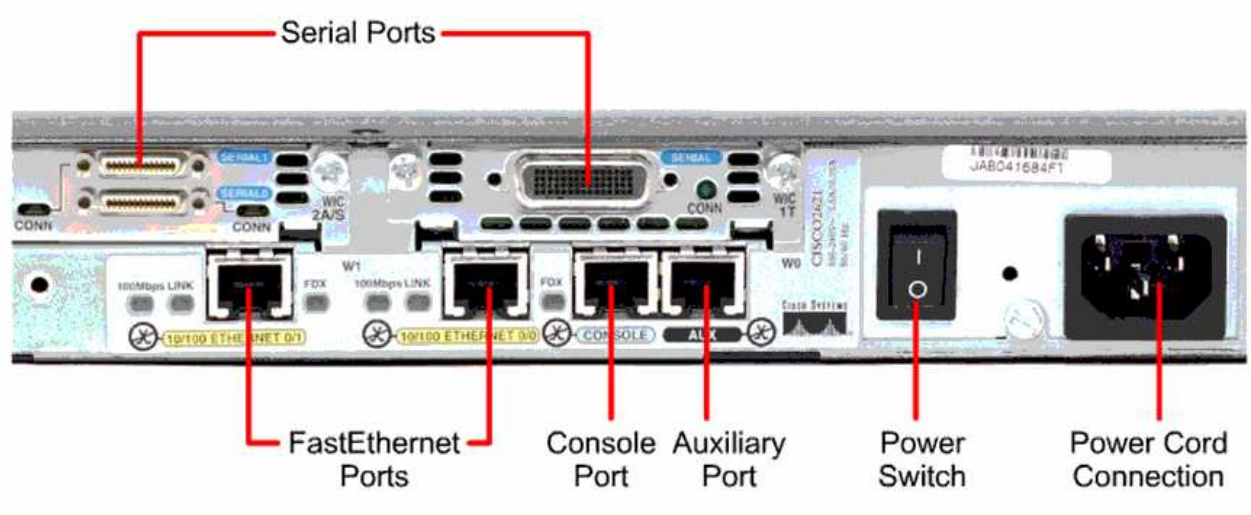


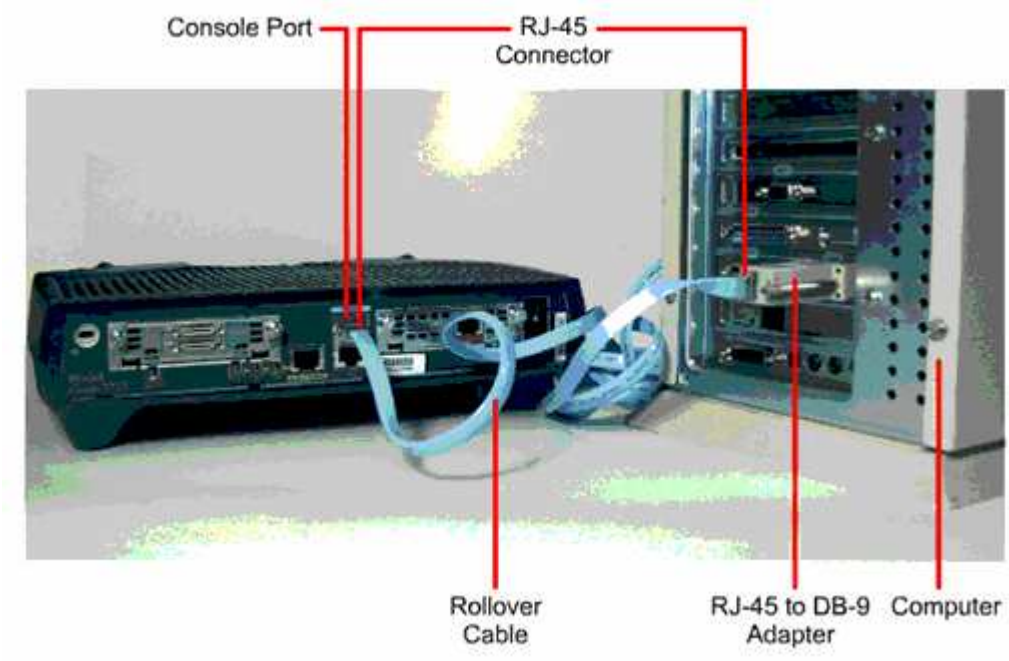
NVRAM: Non-Voltalite Ram' in kısaltmasıdır. Yani kalıcı, silinmez bir RAM' dir. Startup-Configuration denen başlangıç konfigürasyon dosyaları burada bulunur. Router açıldığından buradaki dosyayı alıp RAM' de çalışmasını sağlar. NVRAM boş ise konfigürasyon için bir sihirbaz kullanmayı isteyip istemeyeceğimizi soracaktır.

```
         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: _
```

CPU: İşlemci.

INTERFACE: Router'a erişmek yada çeşitli fiziksel bağlantıları yapmak için kullanılan fiziksel arabirimlerdir. CCNA eğitimleri boyunca kullanılacak interfaceleri "Serial Interface" ve "Ethernet Interface" ler olarak sınıflandırabiliriz. Bu interfaceler default olarak kapalı durumdadır.





ROUTER TEMEL ARAYÜZLERİ

Şimdi de bir Router'da bulunan temel arayüzleri ve nerede kullanıldıklarına bir göz atalım.

- AUI (Attachment Unit Interface): 15 pin'lik bir arayüzdür ve bir harici transceiver ile Enhernet ağlara bağlanabilir.

- Seri Arayüzler: Senkron WAN bağlantıları için kullanılırlar. 2400 Kbps ile 1.544 Mbps arasında bir veri hızına destek verirler. Serial 0, serial 1 gibi isimlerle isimlendirilirler..

- BRI Portları: Basic Rate ISDN portu, uzak bağlantılarda ISDN network'ünü kullanmamıza imkan verir. Genellikle asıl bağlantının yanında yedek bir bağlantı olarak kullanılır. Ayrıca Dial on Demond (DOR) özelliği ile eğer asıl link'in yükü çok artarsa bu bağlantıya yardımcı olmak için devreye girebilir.

Konsol Portu: Router'a yerel olarak bağlanıp konfigüre etmek için kullanılan porttur. Varsayılan veri iletim hızı 9600 bps'dir. Bu portu kullanmak için **rollover kablo** kullanılır. Bu kablonun her iki ucunda RJ 45 konnektor bağlanmıştır. Daha sonra bu konnektörlerin bir tanesi PC'nin seri portlarına bağlanabilmesi için RJ45 - 9 pin seri veya RJ45-25 pin seri dönüştürücüsüne takılarak PC'nin seri portlarından birisine takılır. Kullanılan rollover kablonun her iki uçtaki konnektörlere bağlantı şekli ise şöyle olmalıdır; Bir uçtaki konnektördeki kablo sırası 1-8 ise diğer uçtaki konnektöre bağlantı sırası ise 8-1 olmalıdır.

- AUX Portu: Router'ı konfigüre etmek için her zaman router'ın yanına gitmek zahmetli bir iştir. Router'ı uzaktan konfigüre etmek için bir modem aracılığıyla Router'ın bu portuna bağlantı kurulup gerekli işlemler yapılabilir.

DTE ve DCE

DTE ve DCE kavramları network'teki cihazları işlevsel olarak sınıflandırmamızı sağlar. DTE cihazları genellikle end-user cihazlardır. Örneğin PC'ler, yazıcılar ve router'lar, DTE cihazlardır. DCE cihazları ise DTE'lerin servis sağlayıcıların ağlarına ulaşabilmek için kullandıkları modem, multiplexer gibi cihazlardır. DCE'ler DTE'lere clock işaretini sağlarlar.

Cisco Router'ların seri interface'leri DTE veya DCE olarak konfigüre edilebilir. Bu özellik kullanılarak WAN bağlantıları simüle edilebilir. Bunun için birbirine bağlı Router'ların interface'lerinden bir tanesini DCE diğer Router'ın interface'sini ise DTE olarak kabul ediyoruz. Ardından DCE olarak kabul ettiğimiz interface'in DTE olan interface clock sağlaması gerekiyor. DCE olarak kullanabileceğimiz interface'de "**clock rate**" komutunu kullanarak bir değer atamamız gerekiyor. Aksi halde bağlantı çalışmayacaktır. Örneğin;

RouterA(conf-if)#clock rate 64000

Ayrıca clock rate parametresinin yanında "**bandwidth**" parametresininde girilmesi gerekiyor. DCE ve DTE olarak konfigüre edilecek interface'lerde tanımlanan "bandwidth" değerinin aynı olması gerekiyor. Eğer bandwidth değerini belirtmezseniz varsayılan değeri olarak 1,544 Mbps alınır. Bandwidth'e atadığınız değer sadece yönlendirme protokolü tarafından yol seçimi için kullanılır. Örnegin;

RouterA(conf-if)#bandwidth 64

HYPERTERMİNAL

Router'ı konfigüre etmek için kullanılan bir terminal emülasyon yazılımıdır. Bu yazılım Win 95/98 ve Win NT ile birlikte geldiği için en çok kullanılan terminal emülasyon programıdır. Şimdi bu programı kullanarak Router'a nasıl bağlantı kurulacağını anlatalım. PC'nin herhangi

bir seri portuna taktığımız (COM1 veya COM2) DB-9-RJ45 dönüştürücüye rollover kabloyu takıyoruz. Ardından hyperterminal programını (hypertrm.exe) Start-Programlar-Donatılar'dan çalıştırıyoruz. Karşımıza çıkan "Connection Description" başlıklı pencerede kuracağımız bağlantıya bir isim veriyoruz. Ardından karşımıza çıkan "Connect to" penceresinde ise bağlantının kurulacağı seri port seçiliyor. Bağlantıyı kuracağımız seri portu seçtikten sonra bu portun özelliklerinin belirlendiği bir pencere ile karşılaşıyoruz. Uygun değerleri girdikten sonra hyper terminal penceresindeki "Call" butonuna basıp Router'a bağlantıyı sağlamış oluyoruz.

IOS (INTERNETWORKİNG OPERATİNG SYSTEM)

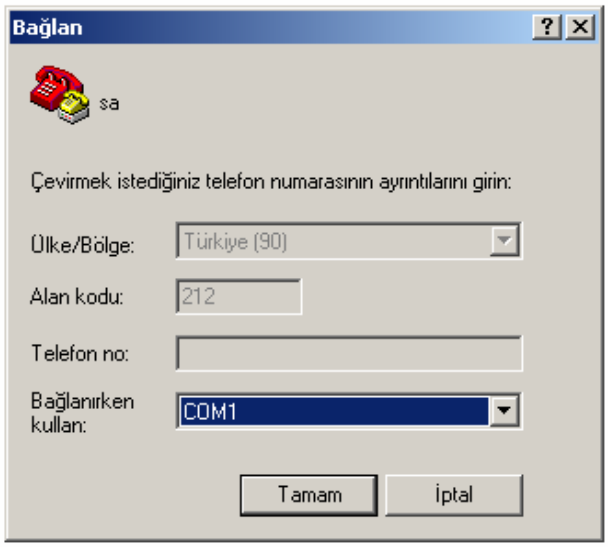
Adından da anlaşılacağı gibi IOS, Router ve Switch' lerin yönetilmesinde kullanılan işletim sistemidir. IOS bize CLI (Command Line Interface) denen text görünümünde bir arayüz sağlar.

Bu arayüze erişmenin üç temel yolu vardır. **Consol Port, Auxilary Port yada Telnet** vasıtasıyla erişmek mümkündür.

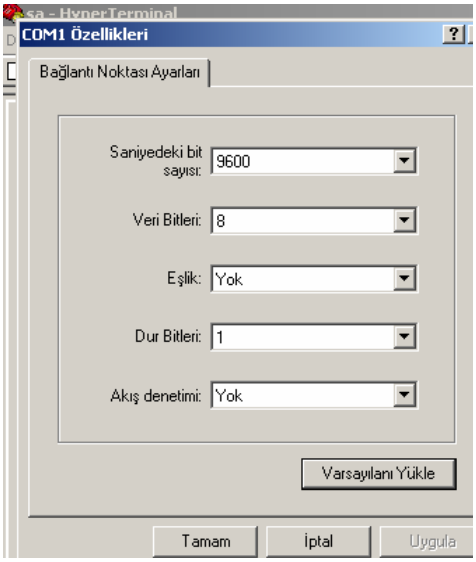
Consol port ile erişmek için, **Roll Over** denen, her iki ucu RJ45 ile sonlandırılmış ve bilgisayarımızın com portundan girilmesi için bir dönüştürücüye sahip özel kablolar kullanılır. Bunlara Konsol kablosu da denir. Hyper Terminal yardımıyla CLI' e erişilebilir.

Auxilary Port veya **Yardımcı portta** denilen bu port modem aracılığı ile asenkron çevirmeli bağlantı kullanarak erişmemizi sağlar.

Buraya herhangi bir isim verip geçiyoruz.



Burada COM1'in seçili olduğuna dikkat edin.



Burada “Varsayılanı Yükle” dedikten sonra Tamam’ a basıyoruz ve routerımıza erişimimiz tamamlanıyor.

Telnet ile Router’ımıza erişebilmemiz için öncelikle Telnet oturumunun aktif hale getirilmesi gerekir. Bunun için Telnet ve enable şifreleri verilmelidir. Bu şifrelerin nasıl verileceğini daha detaylı inceleyeceğiz.

ROUTER’ IN KURULMASI

Router’ın açılması sırasında router konfigürasyon dosyasını arar. Eğer herhangi bir konfigürasyon dosyası bulamazsa sistem konfigürasyon işlemi başlar. Bu işlem sırasında aşağıdaki sorulara “Yes” diye cevap verirseniz Router’ı soru temelli konfigüre edebilirsiniz.

- Continue with configuration dialog? [yes/no]

- Would you like to see the current interface summary? [yes/no]

Bu konfigürasyon türünde router size bir takım sorular sorar ve sizden bu soruların cevaplarını ister. Sorulan soruların varsayılan cevapları soru sonundaki köşeli parantezlerin ([]) içinde verilmiştir. Varsayılan cevapları kabul ediyorsanız yapmanız gereken tek şey Enter’a basmaktır. Eğer soru cevap tabanlı konfigürasyondan herhangi bir zamanda çıkmak istiyorsanız o zaman **Ctrl+C** tuşlarına basmanız yeterlidir.

Eğer yukarıda sorulan sorulara “No” diye cevap verirseniz Router’ı konfigüre edeceksiniz demektir. Bu durumda komut satırı aşağıdaki şekildedir.

Router>

Yani ilk düştüğünüz mod “user exec” moddur. Varsayılan olarak konfigüre edilmemiş tüm Router’ların adı Router’dır ve “privileged exec” moda geçmek için herhangi bir şifre tanımlanmamıştır. Router üzerinde herhangi bir konfigürasyon değişikliği yapmak istiyorsak privileged moda geçmemiz gerekiyor. Bunun için komut satırına aşağıdaki komutu yazalım.

Router>**enable**

Komutu yazdıktan sonra Enter’a basarsanız privileged moda geçersiniz. Bu sırada komut satırının şeklinin değiştiğine dikkat edin. Komut satırı şu şekli almıştır;

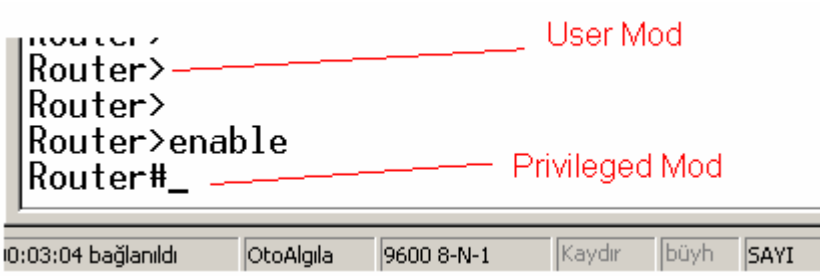
Router#

Privileged exec moddan, user exec moda geri dönmek için ise “disable” komutunu kullanabilirsiniz. Router’da tamamen bağlantıyı koparmak için ise “logout”, “exit” veya “quit” komutlarını kullanabilirsiniz**.**

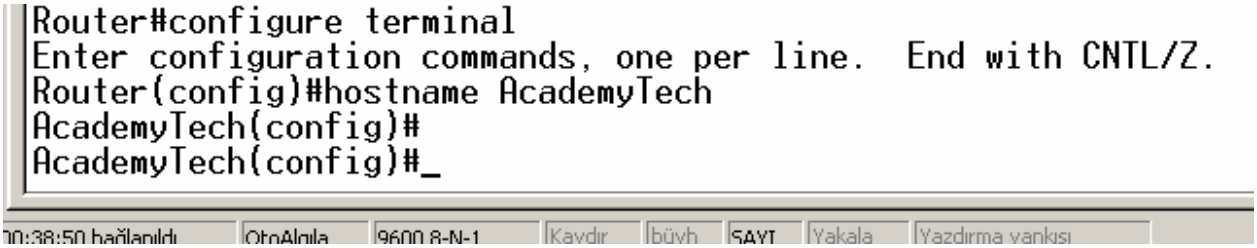
ROUTER ÇALIŞMA MODLARI

User Mod: Router’ ı açıp arayüze eriştiğimiz anda karşımıza çıkan moddur. Bir sonraki modlara geçiş için kullanılır. Bu modda sadece bilgi görüntüleyebilirsiniz. Yani herhangi bir konfigürasyon değişikliği yapamazsınız. Herhangi bir değişiklik yapmak istiyorsanız **privileged exec** moda geçmeniz gerekiyor. User exec moddan privileged moda geçmek için **enable** komutu kullanılır. Bu komutu yazıp enter’a basarsanız router sizden şifre girmenizi isteyecektir. Doğru şifreyi girdikten sonra Router üzerinde istediğiniz ayarları gerçekleştirebilirsiniz.

Privileged Mod: User modda iken “enable” yazıp entera bastığımızda bu moda geçeriz. Bu moda enable moda denir ve önerilen davranış bu moda geçerken şifre konulmasıdır. Zira bir kullanıcı bu moda geçtikten sonra Router’a tamamen hakim olur.



Global Configuration Mod: Config Mod diye de anılan bu moda geçmek için enable moda iken "**configure terminal**" yazılır ve entera basılır. Bu modda yapılan değişiklikler bütün Router'ı etkiler. Örneğin bu modda iken bir router'a isim verilebilir. Bu mod, ileride detaylı anlatacağımız alt modlara ayrılır.



ROUTER KOMUT SATIRI İŞLEMLERİ

Cisco IOS'lar kullanıcılara birçok bakımdan kolaylıklar sunarlar. Örneğin Cisco IOS'lar komut kullanımı sırasında kullanıcılara geniş bir yardım seçeneği sunar. Mesela komut satırındayken **?** karakterine basarsanız bulunduğunuz modda kullanabileceğiniz tüm komutlar bir liste halinde karşınıza çıkacaktır. Eğer sıralanan komutlar ekrana sığmıyorsa ekranın alt kısmında **–More-** diye bir ifade belirecektir. Burada space tuşuna basarsanız sonraki komutları bir ekrana sığacak şekilde görebilirsiniz. Yok eğer varolan komutları teker teker görmek istiyorsanız Enter tuşuna basmanız gerekir.

Bunun haricinde Cisco IOS'lar komut bazında da yardım sağlıyor. Şöyleki; farzedelimki siz sh harfleriyle başlayan komutları listelemek istiyorsunuz. Bunun için komut satırına sh? yazarsanız sh ile başlayan tüm komutlar listelenecektir. Ayrıca kullandığınız komutun parametreleri hakkında bilgi almak içinde komutu yazdıktan sonra bir boşluk bırakıp ? karakterine basın. Örneğin show komutuyla birlikte kullanılabilecek parametreleri görmek için show ? ifadesini yazmalısınız.

Cisco IOS'un kullanıcılara sağladığı diğer önemli bir kolaylık ise komutların syntax'ını tam yazmaya gerek kalmadan komutu anlayarak zaman kazandırmasıdır. Örneğin show komutunu kısaltılmış hali sh'dir. Yani siz komut satırından sh girerseniz IOS bunun show komutu olduğunu anlayacaktır. Komutların kısaltılmış halini belirleyen kural ise o komutun komut listesinde tek (unique) olarak tanımlayabilecek karakter dizisini belirlemektir. Ayrıca komutun kısaltılmış halini yazdıktan sonra Tab tuşuna basarsanız IOS bu komutu, kısaltılmamış haline tamamlayacaktır. Örnegin show komutunu yazmak için sh yazıp Tab tuşuna basarsanız IOS bu komutu show şeklinde tamamlayacaktır. Ayrıca IOS varsayılan olarak yazdığınız son 10 komutu hafızasında tutar. Bu sayıyı "history size" komutunu kullanarak 256'ya kadar arttırabilirsiniz.

Komut yazımı sırasında karşılaşabileceğiniz hata mesajları ve açıklamaları aşağıdaki tabloda verilmiştir.

HATA MESAJI	AÇIKLAMA
%Incomplete command	Yazdığımız komutun tamamlanmadığını, eksik parametre girildiğini belirtir.
%Invalid input	Bu hata mesajıyla birlikte ^ karakteri kullanılır ve bu karekter yanlış girilen omutun neresinde yanlış yapıldığını gösterir.
%Ambiguous command	Girilen komut için gerekli karakterlerin tamamının girilmediğini belirtir. Kullanmak istediğiniz komutu ? karakterini kullanarak tekrar inceleyin.

Aşağıgaki tabloda ise komut satırında kullanılabilecek kısayol tuşları ve fonksiyonlarını bulabilirsiniz.

Kısayol	İşlevi
Ctrl+A	İmleç'i komut satırının başına taşır.
Ctrl+E	İmleç'i komut satırının sonuna taşır.
Ctrl+N veya (↓)	Router'a son girdiğiniz komutlar arasında gezinmemizi sağlar.
Ctrl+F veya (→)	İmleç'i komut satırında bir karakter sağa götürür.
Ctrl+B veya (←)	İmleç'i komut satırında bir karakter sola götürür.
Ctrl+Z	Konfigürasyon modundan çıkartıp exec moda geri döndürür.
Ctrl+P veya (↑)	Router'a girdiğiniz son komutu gösterir.

ROUTER CONFİGURASYON KOMUTLARI

Router üzerinde yapmış olduğunuz değişikliklerin kalıcı olması için bu değişikliklerin konfigürasyon dosyasına yazılması gerekir. Aşağıdaki tabloda Router üzerindeki konfigürasyon ayarlarını görmek, kaydetmek veya silmek için kullanılabilecek komutları bulabilirsiniz.

IOS 10.3 ve öncesi	IOS 11.3 ve öncesi	IOS 12.0	Açıklama
Write terminal	Show running-config	More system: startup-config	Router üzerinde çalışan konfigürasyonu gösterir.
Show configuration	Show startup-config	More NVRAM: startup-config	NVRAM'da bulunan ve Router boot ederken kullanılan konfigürasyonu gösterir.
Write erase	Erase startup-config	Erase NVRAM	NVRAM'de bulunan ve Router boot ederken kullanılan konfigürasyon dosyasını siler.
Write memory	Copy runnig-config startup-config	Copy system: running–config	Router üzerinde yapmış olduğumuz konfigürasyon ayarlarının kalıcı olması için NVRAM'daki konfigürasyon dosyasını yazar.
Write network	Copy running-config TFTP	Copy system: running-config FTP; TFTP	Çalışan konfigürasyonunu FTP veya TFTP server'a kaydetmek için kullanılır.

IOS'UN YEDEKLENMESİ VE GERİ YÜKLENMESİ

Cisco IOS'ların yedeklenmesi ve yedekten geri yüklenmesi için kullanılan komutlar aşağıdaki tabloda listelenmiştir.

Komut	Açıklama
Copy flash tftp	Router'ın flash'ındaki IOS'un yedeğini TFTP server'a kopyalar.
Copy tftp flash	TFTP server'da bulunan bir IOS imajını flash'a kopyalamak için kullanılır.
Copy running-config tftp	Router üzerinde çalışan konfigürasyonu TFTP sunucuna kopyalar.
Copy tftp running-config	TFTP sunucunda bulunan bir konfigürasyon dosyasını router'a yükler.

ROUTER CONFİGURASYONU -I

Şimdi sıra geldi şimdiye kadar teorisiyle ilgilendiğimiz Router'ı konfigüre edip basitçe yönlendirme yapabilecek duruma getirmeye. Bunun için ilk önce Router'a login oluyoruz. Ardından privileged exec mode geçmeniz gerekiyor. "enable" yazıp bu mode giriyoruz. Ardından router'a onu konfigüre edeceğimizi belirten "configure terminal" komutunu veriyoruz. (Bu komutun kısa yazılışı ise "config t"dir.) Şimdi gönül rahatlığı içinde Router'ı konfigüre etmeye başlayabiliriz. İlk önce Router'ımıza bir isim vererek başlayalım. Bunun için "**hostname**" komutunu aşağı şekilde giriyoruz. (Router'ın komut satırının nasıl değiştiğine dikkat edin!)

Router(config)#hostname RouterA

Bu komutu girdikten sonra komut satırı aşağıdaki gibi olacaktır.

RouterA(config)#

Router'ımıza bağlanan kullanıcılara bir banner mesajı göstermek isteyebiliriz. Bunu gerçekleştirmek için "**banner motd**" komutunu aşağıdaki şekilde kullanmalıyız.

RouterA(config)#banner motd#turkmcse.com Router'ına hoşgeldiniz#

Burada komuttan sonra kullandığımız # karakterlerinin arasına mesajımızı yazıyoruz.Bunun haricinde tanımlanabilecek bannerlar ise şunlardır;**Exec banner**,**Incoming banner** ve **Login banner.**

Sıra geldi Router'ımıza bağlantı sırasında kullanıcılara sorulacak şifreleri belirlemeye. Cisco Router'larda beş farklı şifre bulunur. Bunlardan ikisi privileged mod'a erişim için tanımlanırken, bir tanesi konsol portu, bir tanesi AUX portu ve diğeride Telnet bağlantıları için tanımlanır. Bu şifrelerden "**enable secret**" ve "**enable password**", privileged mod'a geçmek için kullanılırlar ve aralarındaki fark "enable secret"'in şifrelenmiş bir şekilde saklanmasıdır. Yani konfigürasyon dosyasına baktığınızda "enable secret" şifresinin yerinde şifrelenmiş halini görürsünüz. Ama aynı dosyada "enable password"'u ise açık bir şekilde şifreleme yapılmadan saklandığını görürsünüz. Bu da sizin konfigürasyon dosyanızı ele geçiren birisinin "enable password" şifresini kolayca okuyabileceğini ama "enable secret" şifresinden bir şey anlamayacağı anlamına gelir. "Enable password" şifresi ise "enable secret" şifresi tanımlanmamışsa veya kullanılan IOS eski ise kullanılır. "Enable secret" şifresinin konfigürasyon dosyasına yazılırken kullanılan şifrelemenin derecesini ise "**service password-encryption**" komutu ile belirleyebilirsiniz. Şimdi sırasıyla bu beş şifrenin nasıl tanımlandığını anlatalım; "Enable secret" ve "enable password" şifreleri aşağıdaki şekilde tanımlanır.

RouterA(config)#enable password cisco

RouterA(config)#enable secret istanbul

Burada turkmcse ve istanbul bizim koyduğumuz şifrelerdir.

Eğer Router'ın konsol portuna şifre koymak istiyorsanız

RouterA(config)#line console 0

RouterA(config-line)#login

RouterA(config-line)#password cisco

Router'ın AUX portuna şifre koymak için:

RouterA(config)#line aux 0

RouterA(config-line)#login

RouterA(config-line)#password istanbul

Router'ın Telnet bağlantılarında soracağı şifreyi ise şöyle belirleyebilirsiniz:

RouterA(config)#line vty 0 4

RouterA(config-line)#login 17

RouterA(config-line)#password turkiye

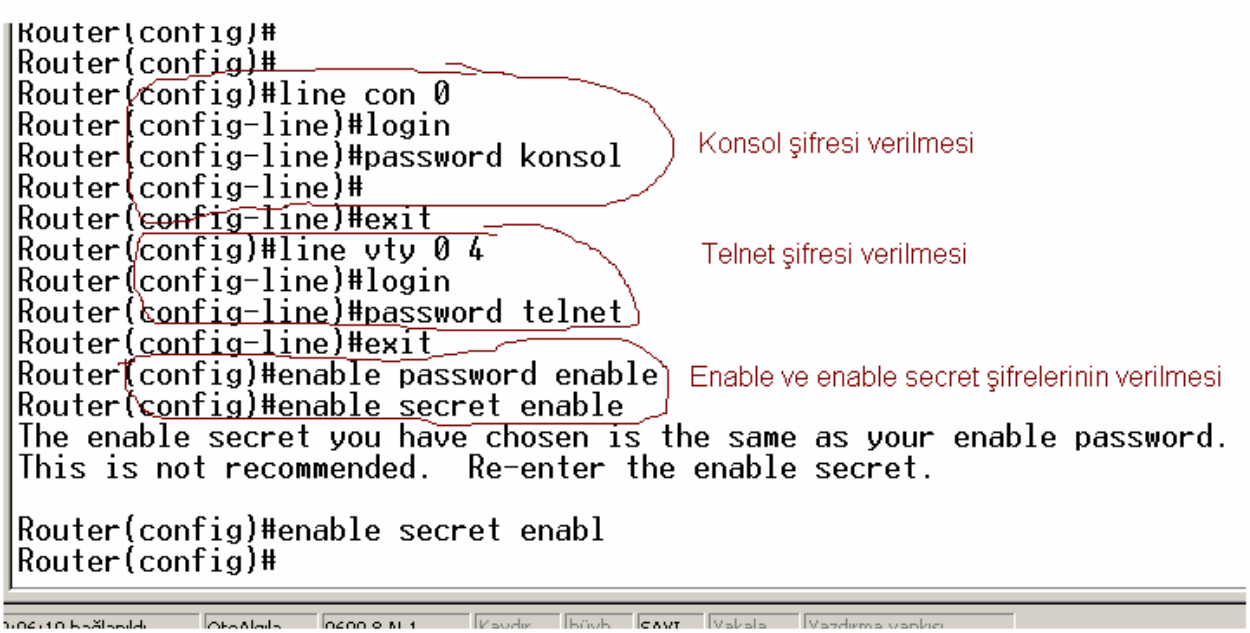
Burada telnet portlarının tamamına aynı şifre verilmiştir. Bu portların herbirisine farklı şifreler atanabilir. Fakat router'a yapılan her telnet isteğine router, o zaman kullanımda olmayan bir port'u atadığı için bağlantıyı kuran kişinin tüm bu telnet portlarına atanmış şifreleri bilmesi gerekir. Bu yüzden telnet portlarına ayrı ayrı şifre atamak iyi bir yaklaşım değildir.

Bunun haricinde Router'a yapılan konsol bağlantılarının, kullanıcı herhangi bir işlem yapmadan ne kadar süre aktif kalacağını da **"exec-timecut**" komutuyla belirleyebiliriz

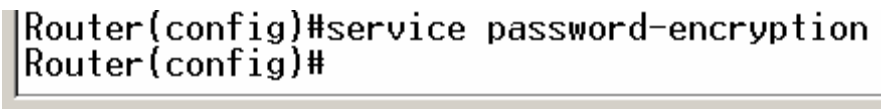
ENABLE, TELNET VE KONSOL ŞİFRELERİ VERME

Enable şifresi Global Configuration modda verilirken konsol ve telnet şifreleri line Configuration mod denilebilecek alt modlarda verilebilir. Enable şifre "**enable secret**" komutu kullanılarak 5. leveldan şifrelenebilirken telnet ve konsol şifrelerinde bu mümkün değildir. Fakat 7. leveldan şifrelenebilirler ve bunun için gerekli komutumuz "**service-password encryption**" dır.

Bir Router' a "enable secret" ve "enable" şifreleri, aynı olmamak şartıyla birlikte verilebilir. Bu durumda "enable secret" şifresi geçerli olacaktır.



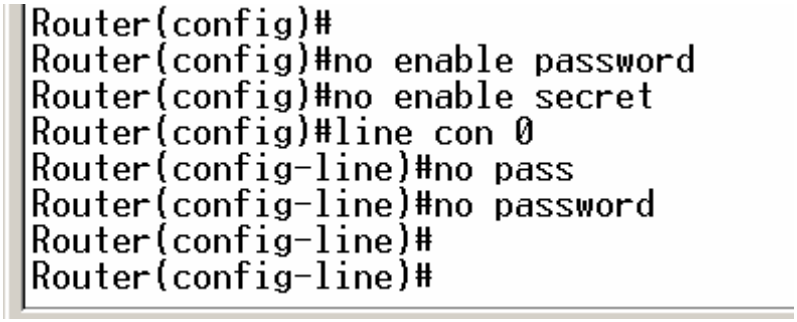
(Dikkat edilirse enable ve enable secret şifrelerinin aynı olmasına izin verilmiyor)



(Şifrelerin 7. leveldan encrypted edilmesi)

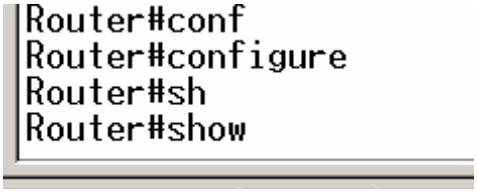
Şifre verirken kullandığımız "login" komutu dikkatinizi çekmiştir. Default olarak şifresiz kabul edilen Router' a bu komut ile artık kendisine şifre vasıtasıyla erişileceği bilgisini vermiş oluyoruz. Bütün komutlar basına "no" yazılarak geçersiz hale getirilebilir.

“ no enable secret” gibi bir komut ile enable secret şifresini kaldırabiliriz.

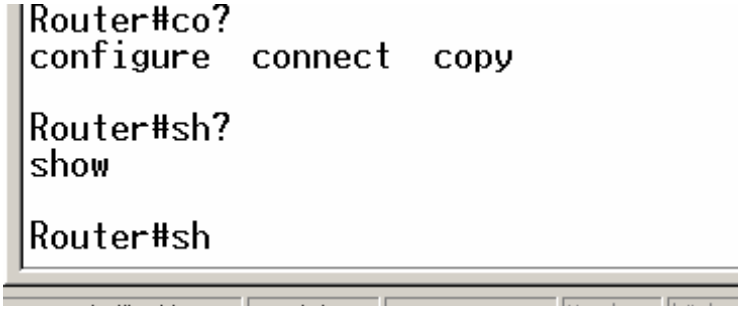


Yardım Alma

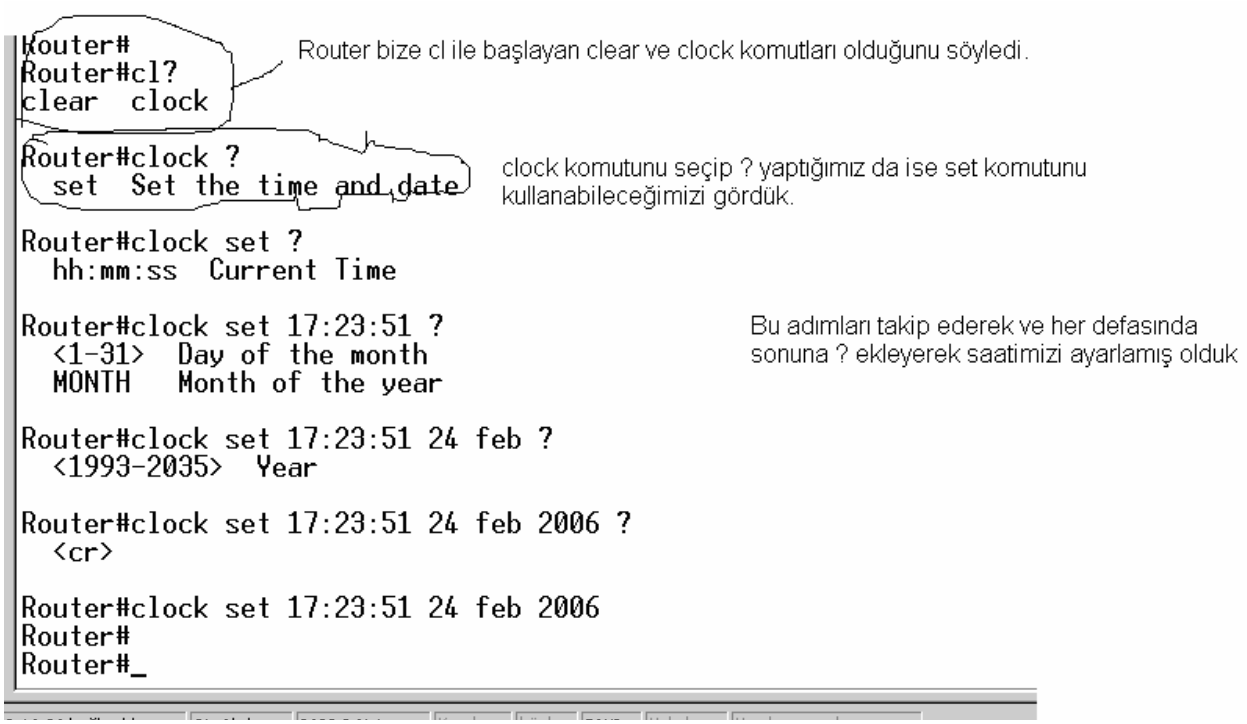
Router konfigürasyonu sırasında kullanacağınız komutun ilk harflerini yazıp tab tuşuna bastığınızda, yazdığınız komut bulunduğunuz mod için geçerliyse ve o harflerle başlayan başka bir komut yoksa, Router sizin için komutu tamamlayacaktır.



Ve yine devamını hatırlamadığınız komutlar için sonuna “?” koymak süretiyle yardım alabilirsiniz.

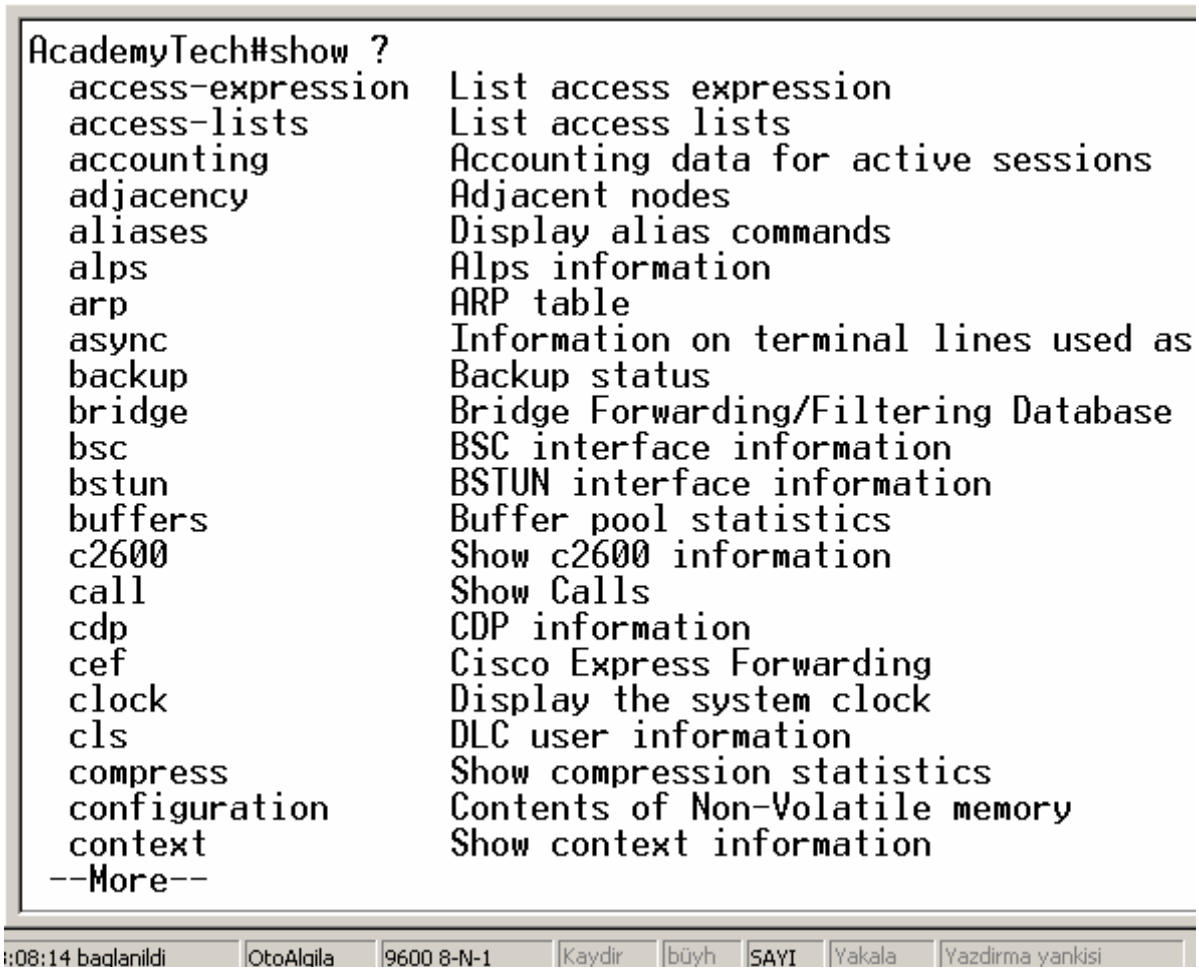


Konuyu tam olarak kavrama da AcademyTech laboratuarlarında sıkça uyguladığımız bir çalışma da (Clock uygulaması) aşağıda detaylı bir şekilde gösterilmiştir.



Show Komutları

Show komutu Router ile ilgili bir çok şeyi görüntüleme de bize yardımcı olur. Show komutları Enable Moda çalışır ve yardım alındığında görünecektir ki bir çok uygulaması vardır.



Görünende çok daha uzun bir listeyi Routerlarda inceleyebilirsiniz. Burada önemli ve bizlere CCNA eğitimi boyunca yardımcı olacak belli baslı show komutları, yeri geldikçe gösterilecektir.

Konfigürasyon Dosyaları

Routerın açılış konfigürasyonunun tutulduğu **startup-config** ve çalışan konfigürasyonunun tutulduğu **running-config** adı altında iki dosyası vardır. Bir router' ın running-config ve startup-config dosyalarını **"show**" komutu ile görebilir, "**copy**" komutu ile birbirleri üzerine kopyalayabilir, "**erase**" komutu ile silebiliriz.

Startup-Config: NVRAM'da bulunur, yeni alınmış bir Router için üzerinde hiçbir bilgi bulunmaz. Ve böyle bir Router açılışta startup ve running konfigürasyonunun bir sihirbaz yardımıyla yapıp yapmayacağımız sorusunu sorar. Bu sihirbaz gereksiz ve boşa zaman harcatan bir çok soru ile doludur ki önerdiğimiz ve uyguladığımız konfigürasyonu manuel yapmaktır.

```
Router#show startup-config
```

Running-Config: RAM'da bulunur ve Router'ın çalıştığı andaki konfigürasyonunu tutar. Router kapatıldığında buradaki bilgiler gider.

```
Router#show running-config
Building configuration...
```

Bir Router yeniden başlatıldığı zaman startup-config dosyası dolu ise, IOS tarafından bu dosya alınıp RAM'a aktarılır ve dolayısıyla o artık Running-config olmuştur. Bir router' ın running-config ve startup-config dosyalarını "show" komutu ile görebilir, "copy" komutu ile birbirleri üzerine kopyalayabiliriz.

```
Router#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
```

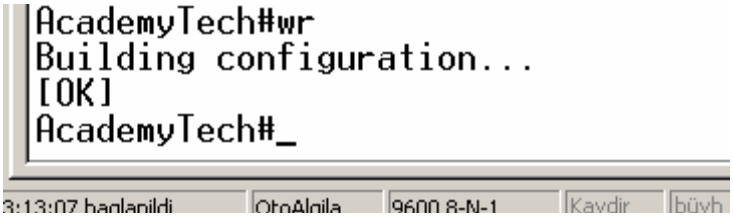
```
Router#erase nvram:
Erasing the nvram filesystem will remove all files! Continue? [confirm]
[OK]
Erase of nvram: complete
Router#
Router#
```

Write Komutu

Kopyalama ve silme işlemlerinde "Write" komutu da kullanılabilir. Write komutu ile birlikte kullanılabilecek komutlar aşağıdadır.

```
AcademyTech#write  ?
  erase     Erase NV memory
  memory    Write to NV memory
  network   Write to network TFTP server
  terminal  Write to terminal
  <cr>

AcademyTech#write  _
```

NOT: Kısaca "wr" yazmak Running Konfigürasyonumzu NVRAM'a kayıt edecektir.



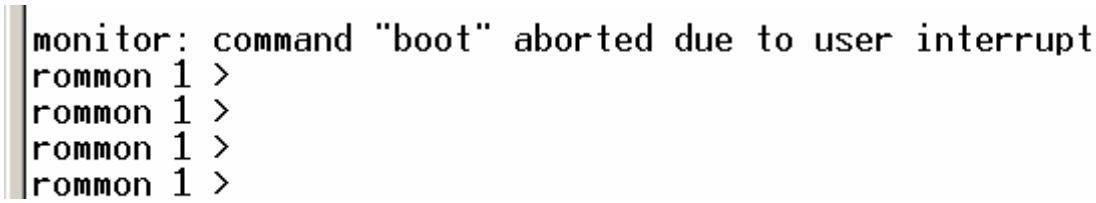
ŞİFRE KIRMA

Routerın şifrelerini unuttuğunuzu yada ikinci el bir Router aldığınızı ve bu router' ın konfigürasyon dosyalarının hala üzerinde olduğunu dolayısıyla şifrelerini bilmediğinizi varsayalım. Böyle bir durumda şifreyi değiştirmek ve istersek eski konfigürasyonun bozulmamasını da sağlayarak bunu yapmak mümkündür. Bu ilk bakışta bir güvenlik açığı gibi görünse de, bu işlemin yapılabilmesi için konsoldan router' a bağlanmamız, dolayısıyla fiziksel olarak router'ın yanında olmamız gerekeceği için açık denilemez. Zira fiziksel olarak erişilebilen bir router' ın şifreleriyle oynayabilmenin bir sakıncası yoktur.

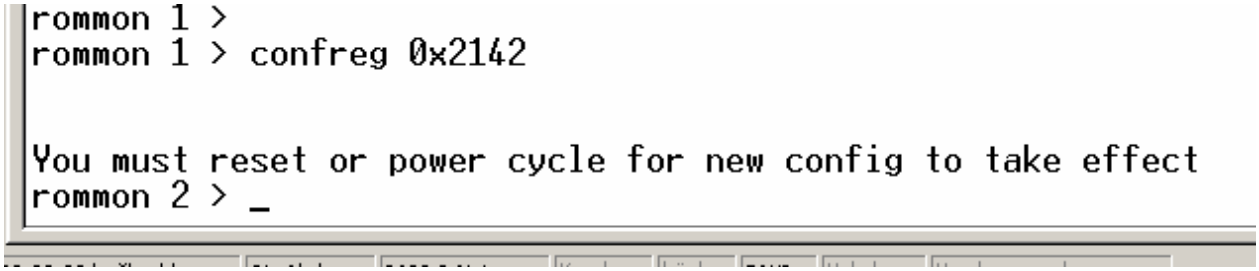
Adım adım şifre kırma işlemini inceleyecek olursak;

1. Router açılırken Ctrl+Break tuşlarına basılarak Rom Monitöre girilir. Burada

"Router>" yerine "rommon>" ifadesiyle karsılaşacağız.



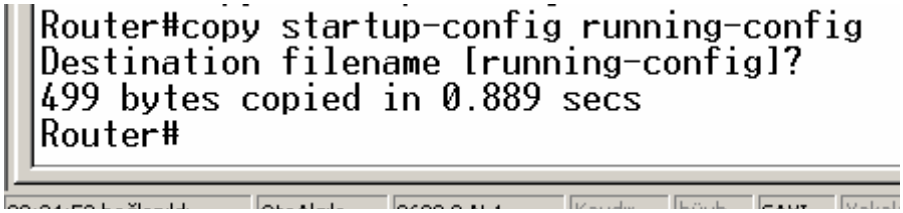
2. "confreg" komutu ile başlangıç register' ı değiştirilir ve NVRAM yerine direk RAM' dan çalışmaya başlaması sağlanır. Bu sayede mevcut konfigürasyon NVRAM' da bulunmaya devam ederken Router RAM'dan sıfır konfigürasyon ile açılacaktır. 0x2102 olan register 0x2142 olarak değiştirilmelidir.



3. Router yeniden başlatılır. Açıldığında Router' ın herhangi bir şifre sormadığını göreceksiniz.

4. Enable moda geçilir. Bu moda geççikten sonra artık istediğimiz her şeyi

yapabileceğimize göre, eski konfigürasyonu kaybetmek istemiyorsak, "copy startup-config running-config" komutu ile o dosyayı alır ve şifreleri değiştirip yeniden NVRAM' a kaydederiz.



Bundan sonra istediğimiz değişiklikleri yapıp running-config dosyasını tekrar Startup-config üzerine yeni haliyle kopyalayabiliriz.

5. Son olarak Rom Monitör' e girip değiştirdiğimiz register' ı eski haline getirip (0x2102) getirip Router' ımızı yeniden başlatabilir ve eski konfigürasyon ve yeni şifreyle router'ın açıldığını görebiliriz.

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#conf
Router(config)#config-register 0x2102
Router(config)#_
```

TEMEL ROUTER KONFİGÜRASYONU -II

Bir router' ın çalışması için şifre vermekten çok daha fazlası gerekir. En temel gereklilik ise Router' ın interface' lerine ip adresi atamaktır. Router' ın interfaceleri default olarak shutdown durumdadır ve bunun kaldırılması gerekir ki bu da ip adresinin atadıktan sonra ilgili interface' e "no shutdwon" komutu vermek ile mümkündür.

Bir router' ın interfacelerinden herhangi birine ip adresi atamanın diğerinde farkı yoktur. Yapılacak işlemler sırasıyla interface konfigürasyon moduna geçmek, ip adresini subnet maskı ile birlikte yazmak ve "no shutdown" ile interface' i aktif hale getirmektir.

Eğer bu interface için bir açıklama eklemek istiyorsanız bunu aşağıdaki gibi "**description**" komutunu kullanarak yapabilirsiniz.

RouterA(config-if)#description Pazarlama Grubunun LAN bağlantısı

Konfigüre ettiğiniz interface'in işlevselliğini yerine getirebilmesi için aktif (up) olması gerekiyor. Varsayılan olarak bütün interface'ler pasif **(administratively disabled)**'dir. Bunun için ise aşağıdaki komutu kullanmalısınız.

RouterA(config-if)#no shutdown

Ayrıca Cisco'nun 7000 veya 7500 serisi router'larında VIP(Versatile Interface Processor) kartları varsa bunun için aşağıdaki formatta bir komut kullanarak interface tanımlamalısınız;

Interface tip slot/port adaptör/port numarası

Örneğin;

RouterA(config)#interface ethernet 2/0/0

Debug İşlemi

Router üzerinde hata ayıklamak için kullanılabilecek komutlar mevcuttur. Bu komutların başında "**debug**" komutu gelir.

RouterA#debug all

Unutulmaması gereken bir nokta da debug işleminin Router'ın kaynaklarını bir hayli fazla kullandığıdır. Bu yüzden debug işlemi bitirildikten sonra "undebug all" veya "no debug all" komutlarından bir tanesi kullanılarak Router'a debug yapmaması gerektiği bildirilmelidir.

CDP (Cisco Discovery Protocol)

Data Link katmanında çalışan bu protokol Cisco tarafından geliştirilmiştir ve fiziksel olarak birbirine bağlı tüm Cisco cihazlarının birbirleri hakkında bilgi sahibi olmalarını sağlar. IOS 10.3 veya daha yukarı versiyon çalıştıran Router'larda CDP default olarak aktifdir ve otomatik olarak komşu Router ve switch'ler hakkında bilgi toplar. Bu bilgiler arasında cihaz ID'si ve cihaz tipi gibi bilgilerde bulunur. CDP kullanılarak öğrenilen bilgileri privileged mod'da "**show cdp neighbors**" komutunu kullanarak görebilirsiniz. Bu komutu kullandığınızda fiziksel olarak bağlı olduğunuz cihazların isimlerini, portlarını, cihaz tiplerini(router,switch vs.) ,sizin router'ınıza hangi interface'inin bağlı olduğunu,bu cihazların hangi platforma ait olduğunu,holdtime değerini interface isimlerini görebilirsiniz. CDP ile toplanmış bilgileri daha ayrıntılı bir şekilde görmek istiyorsanız "**show cvp neighbor detail**" komutunu kullanmalısınız. Bu komutun çıktısında ise show cdp neighbors komutunun çıktısında bulunan bilgilere ek olarak cihazda kullanılan IOS versiyonu, IP adresleri gibi bilgileri bulabilirsiniz.

Eğer CDP protokolünün Router üzerinde çalışmasını istiyorsanız o zaman global konfigürasyon modunda iken **"no CDP run**" komutunu girmelisiniz. Ayrıca CDP'yi interface bazında da pasif yapabilirsiniz. Bunun için interface konfigürasyon modunda iken "**no CDP enable"** komutunu girmelisiniz.

Örnek bir çalışma olarak Router'ımıza şu ip adreslerini atayalım.

Ethernet Interface Ip adresi : 192.168.1.1 / 24

Serial (0/0) Interface Ip Adresi: 192.168.2.1 /24

Serial (0/1)Interface Ip Adresi : 192.168.3.1 /24

```
Router(config)#interface et
Router(config)#interface ethernet 0/0
Router(config-if)#ip addr
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#
```

(Ethernet 0/0 interface'ine ip adresi verildi)

```
Router(config)#interface serial 0/0
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
```

(Serial 0/0 interface' ine ip adresi verildi)

```
Router(config)#interface serial 0/1
Router(config-if)#ip address 192.168.3.1 255.255.255.0
Router(config-if)#no shut
Router(config-if)#no shutdown
```

(Serial 0/1 interface' ine ip adresi verildi)

Buradaki 0/0, 0/1 gibi ifadeler standart olmamakla birlikte Router' ımızın üzerinde yazıyor olmalı. Eğer yazmıyorsa, Router'ımıza "show running-config" komutunu verip hangi interface' in hangi numaraya sahip olduğunu öğrenebiliriz.

Router' ımıza gerekli şifreleri verip interfacelerine de gerekli ipleri atadıktan sonra "Show running-config" ile göreceğimiz text ifade şu şekilde olacaktır.

```
Router#sh running-config
Building configuration...
Current configuration : 526 bytes
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
hostname Router
memory-size iomem 10
ip subnet-zero
interface Ethernet0/0
ip address 192.168.1.1 255.255.255.0
half-duplex
interface Serial0/0
ip address 192.168.2.1 255.255.255.0
no fair-queue
interface Serial0/1
ip address 192.168.3.1 255.255.255.0
!
ip classless
ip http server
dial-peer cor custom
!gatekeeper
shutdown
line con 0
line aux 0
line vty 0 4
!
end
```

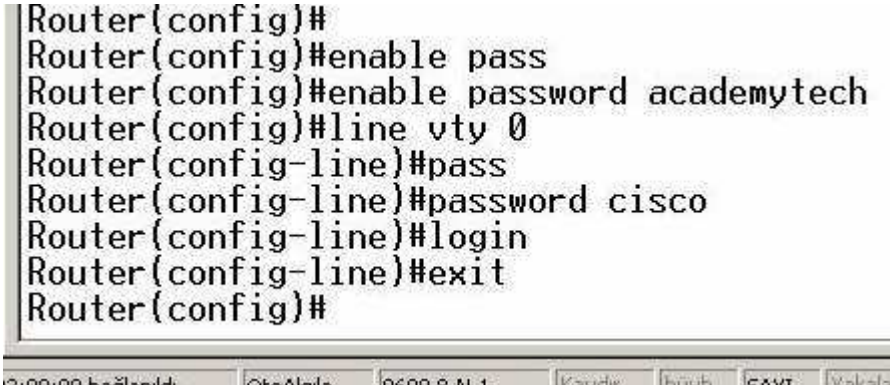
ROUTER'A TELNET İLE BAĞLANMA

Router üzerinde bir konfigürasyon yapılacak olması mutlaka Router'a fiziksel olarak erişmeyi yani Konsol' dan bağlanmayı gerektirmez. Router'a Telnet ile de bağlanılabilir.

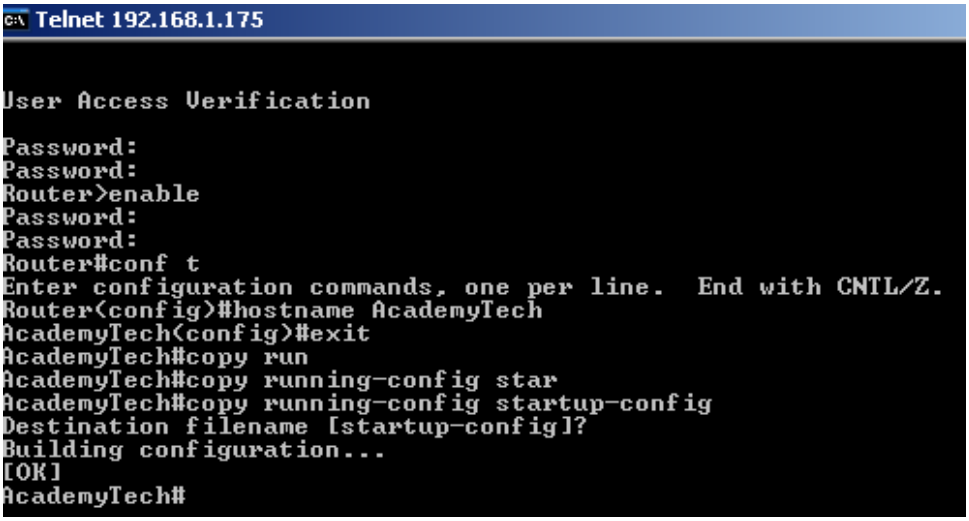
Tüm Cisco Router ve switch'ler Telnet isteklerine cevap verecek şekilde, üzerlerinde Telnet server servisi çalışır vaziyette gelirler. Bunun yanında tüm Cisco Router'ları ve bazı switch'ler Telnet istemci programı ile birlikte gelir ve ağ yöneticilerinin Router'ları uzaktan yönetmesini sağlar. Privileged modda iken herhangi bir Router'a bağlanmak için **"telnet**" veya

“connect” komutlarını kullanabilirsiniz. Bu komutlar parametre olarak bağlantının kurulacağı Router’ın IP adresini veya host ismini alır. Eğer parametre olarak host ismi kullanılmışsa Router’da DNS ayarlarının yapılması gerekir. Ya da Router’daki host tablosuna **“ip host**” komutunu kullanarak bu host’a ait kayıt girilmelidir.

Fakat bunun için bazı şartların yerine gelmesi gerekir. Öncelikle Router’ ın Ethernet interface’ i up olmalıdır ve Telnet, Enable şifreleri verilmiş olmalıdır. Telnet şifresi verilmediğinde “Password Required, but none set” şeklinde bir hata mesajı alınacak ve bağlan gerçekleştirilemeden kaybolacaktır.



(Telnet ve Enable Şifrelerinin Verilmesi)



Görüldüğü gibi şifreler verildikten sonra bağlantı gerçekleştirilebilir ve her türlü konfigürasyon yapılabilir.

Örneğin aşağıdaki komutla adı RouterB ve IP adresi 10.3.10.1 olan router’ın kaydı host tablosuna girilmektedir.

RouterA(config)#ip host RouterB 10.3.10.1

Eğer router’ın isim çözümleme işini host tablosuyla değilde DNS sunucu ile halletmek istiyorsanız o zaman Router’a DNS sunucunun adresini **“ip name-server**” komutunu kullanarak belirtmelisiniz.

RouterA(config)#ip name-server 10.3.9.2

Router’ın komut satırında herhangi bir şeyi örneğin bir komutu yanlış veya eksik yazarsanız router bunun bir isim olduğunu farzedip DNS sunucuyu arayacak ve bu ismi çözmeye çalışacaktır. Bu işlemde bir hayli zaman alacaktır. Böyle bir durumda beklememek için **Ctrl+Shift+6** tuş kombinasyonuna bastıktan sonra **X** tuşuna basıp bu işlemi sonlandırabilirsiniz.

Bunun haricinde bu tuş kombinasyonu uzak sistemlere yapılan telnet bağlantısını askıya alıp kendi router'ınıza geri dönmek içinde kullanılır.

Bir telnet otrumunu kapatmak için "**disconnect**", "**exit**", "**quit**" veya "**logout**" komutlarını kullanabilirsiniz. Eğer birden fazla Router'a Telnet ile bağlanmışsanız bu bağlantıları "**show session**" komutunu kullanarak görebilirsiniz.

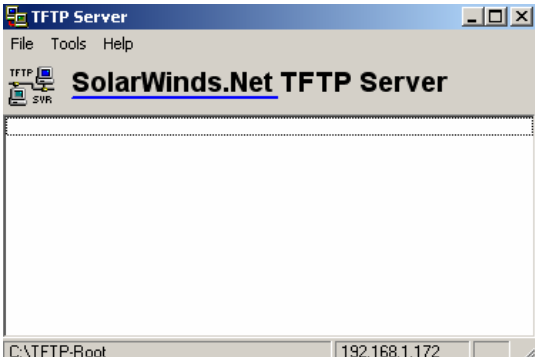
Şekilde ki gibi bir tabloyla karşılaşıldığında anlaşılması gereken gerekli şifrelerin verilmemiş olduğudur. Önceki bölümlerde öğrendiğimiz gibi şifreleri verdikten sonra bağlantımızı gerçekleştirebiliriz.

TFTP SERVER'A YEDEK ALMA

Konfigürasyonu yapılmış bir Router'ın startup ve running-config dosyalarının yedeklerini almak akıllıca bir harekettir. Bu TFTP Server sayesinde mümkün. Ve yine TFTP sayesinde Flash' ın yedeği alınabilir, güncellemesi yapılabilir.

TFTP Server normal bir PC'ye yükleyeceğimiz UDP protokolünü kullanan ufak bir programdır. Bu program network üzerinden TFTP isteklerini karşılamak için devamlı networkü dinler.

TFTP Server' a yedek alınabilmesi için kurulu olduğu bilgisayarın ip adresini, flas' ın yedeği alınacaksa onun tam adını bilmek gerekir. Flash' ın tam adını "Show version" komutu ile öğrenebiliriz. "copy" komutu bundan sonrasını kendisi halledecektir.



Copy startup-config tftp:

Copy running-config tftp:

Copy flash tftp

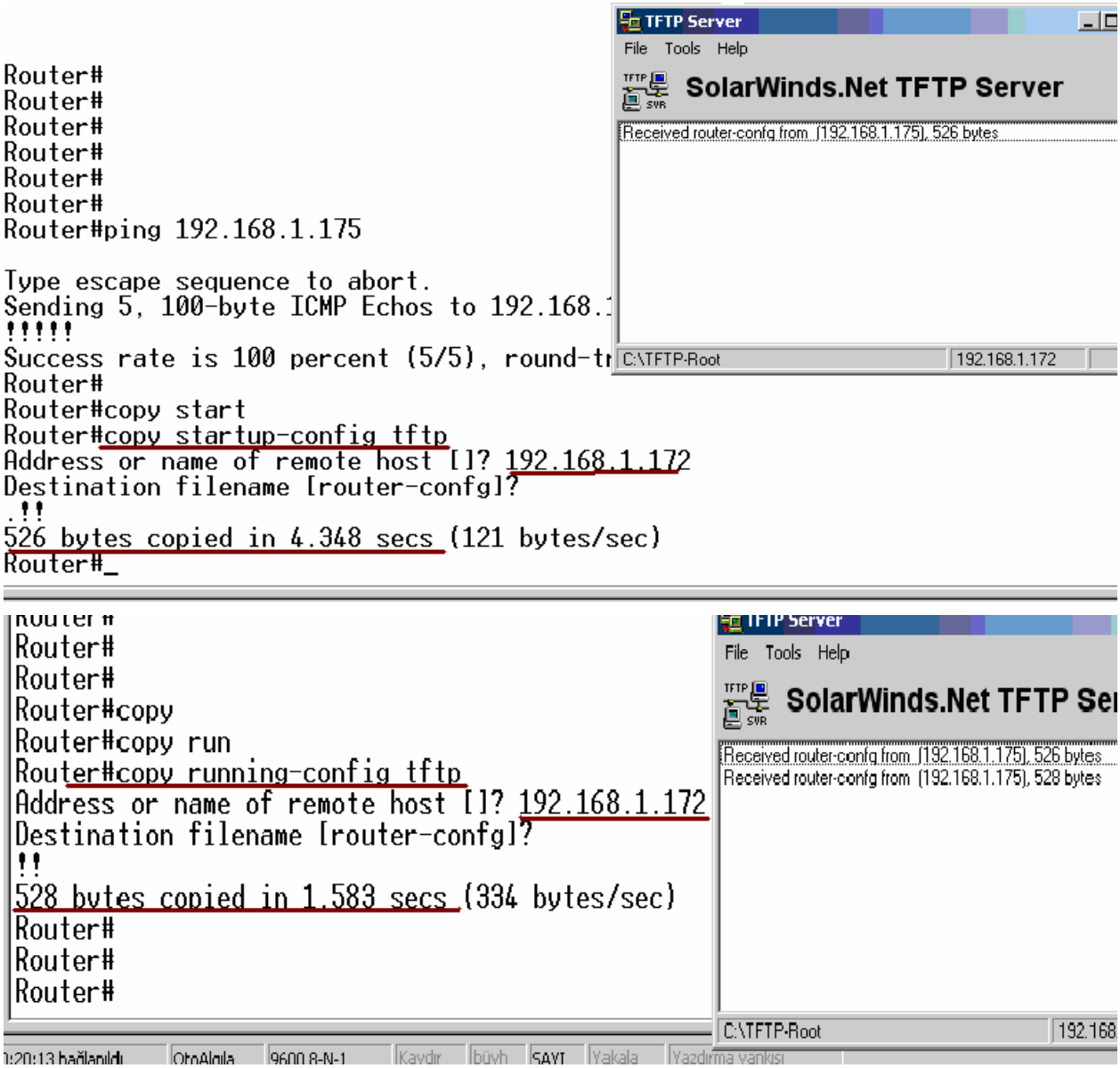
Gibi bir komut yazdığımız da bize ilk olarak TFTP Server'ın ip adresi ve şayet Flas' ın yedeğinin alacaksak onun tam adını soracaktır. Ve bütün bunlar yapılırken TFTP Server çalışıyor durumda olmalı.

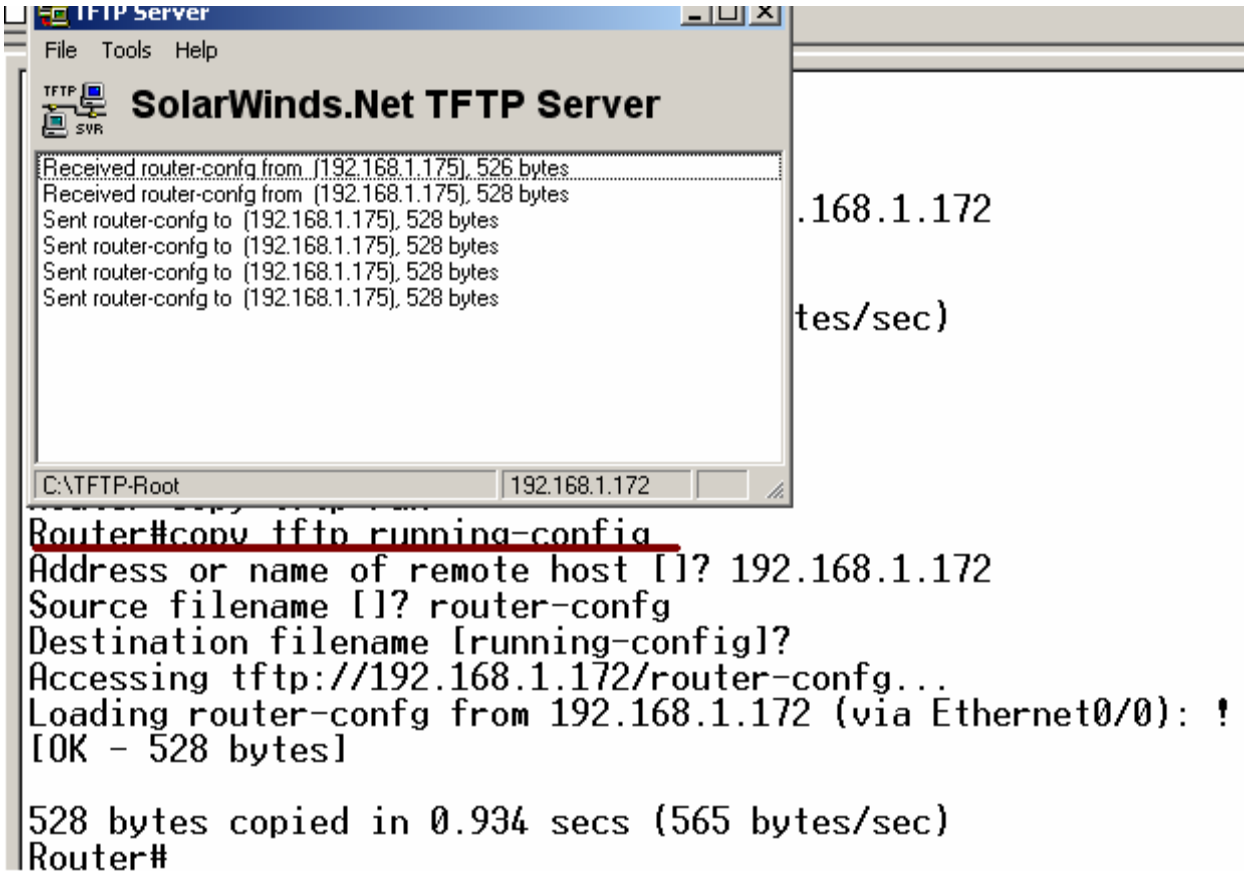
TFTP Serverdan geri yüklemelerde ise komut tam tersi yazılarak çalıştırılacaktır.

Copy tftp startup-config

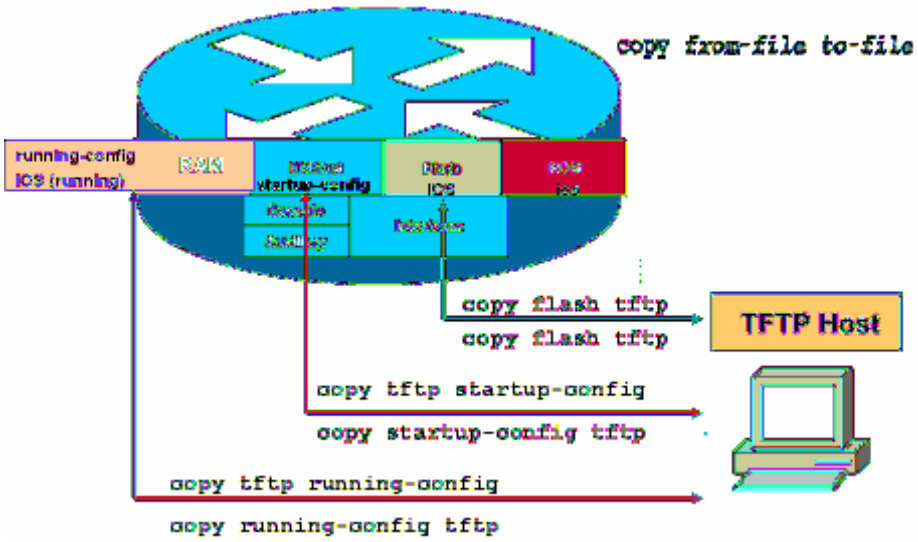
Copy tftp running-config

Copy tftp flash


TFTP Server
File Tools Help
SolarWinds.Net TFTP Server
Received router-config from [192.168.1.175], 526 bytes
C:\TFTP-Root
192.168.1.172
Router#
Router#
Router#
Router#
Router#
Router#
Router#ping 192.168.1.175
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1
!!!!!
Success rate is 100 percent (5/5), round-tr
Router#
Router#copy start
Router#copy startup-config tftp
Address or name of remote host []? 192.168.1.172
Destination filename [router-config]?
!!!
526 bytes copied in 4.348 secs (121 bytes/sec)
Router#_
Router#
Router#
Router#
Router#copy
Router#copy run
Router#copy running-config tftp
Address or name of remote host []? 192.168.1.172
Destination filename [router-config]?
!!
528 bytes copied in 1.583 secs (334 bytes/sec)
Router#
Router#
Router#
TFTP Server
File Tools Help
SolarWinds.Net TFTP Ser
Received router-config from [192.168.1.175], 526 bytes
Received router-config from [192.168.1.175], 528 bytes
C:\TFTP-Root
192.168



Copy Komutları Özet



IOS YEDEK ALMA VE YÜKLEME

TFTP Server kullanarak IOS' in yedeği alınabilir veya IOS yüklenebilir. Bunun için Sistem Imağe File' ın tam dosya adı bilinmelidir ve bu "show version" komutu ile öğrenilebilir. Alınan bütün yedekler gibi IOS' in yedeği de TFTP Server tarafından TFTP-Root klasörünün altına atılır.

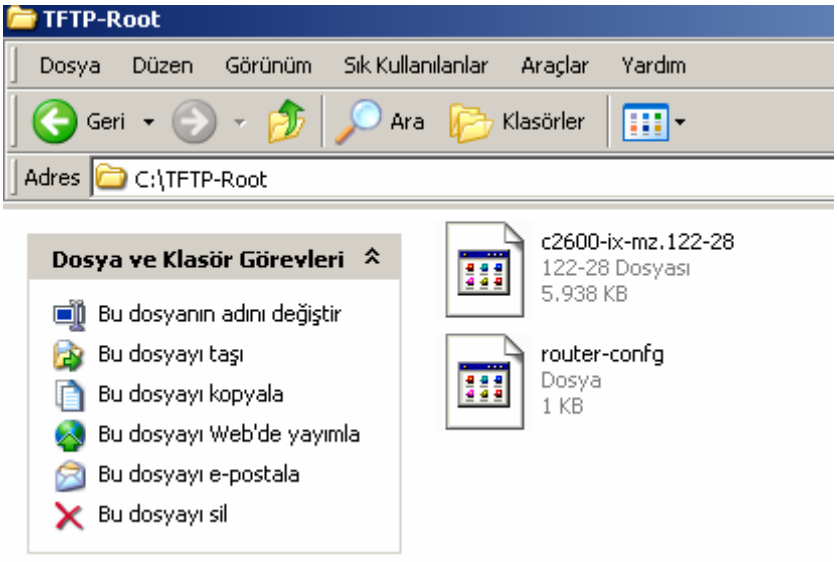
```
ROM: System Bootstrap, Version 11.3(2)XA3, PLATFORM SPECIFIC RELEASE SOFTWARE (f
c1)
ROM: C2600 Software (C2600-IX-M), Version 12.2(28), RELEASE SOFTWARE (fc5)

Router uptime is 34 minutes
System returned to ROM by power-on
System image file is "flash:c2600-ix-mz.122-28"

cisco 2610 (MPC860) processor (revision 0x202) with 36864K/4096K bytes of memory
.
Processor board ID JAB024903E2 (2074409390)
M860 processor: part number 0, mask 49
Bridging software.
X.25 software, Version 3.0.0.
1 Ethernet/IEEE 802.3 interface(s)
2 Serial network interface(s)
 --More--
```

Yedek alırken startup-config ve running-config' den farklı olarak dikkat edilecek tek konu hedef dosya adıdır ve şekilde belirtildiği gibi tam adı olmalıdır.

```
Router#copy flash tftp
Source filename []? flash:c2600-ix-mz.122-28
Address or name of remote host []? 192.168.1.172
Destination filename [c2600-ix-mz.122-28]?
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
.....................................................................
6080092 bytes copied in 46.256 secs (131444 bytes/sec)
Router#_
```

00:37:21 bağlanıldı | OtoAlgıla | 9600 8-N-1 | Kaydır | büyh | SAYI | Yakala | Yazdırma yankısı

("copy flash tftp" komutuyla yedek alınması)



(Yedekleri alınan dosyalar TFTP-Root klasörünün altında)

IOS olmadığında router ethernet interface' ine ip adresi veremeyeceğimizde IOS' in konsoldan yüklenmesi gerekmektedir. Bunun için kullanılacak iletişim kuralı "xmodem"' dir ve konsol hızı 115200 bps' a çıkarılmalıdır. Tabii ki bu işlemlerin tamamı Rom Monitör (Rommon) kullanılarak yapılabilir.

Bunun için Router açılırken Ctrl+Break tuşlarını basılara Rommon' a girilir ve konsol hızı 115200 bps' a çıkartılır. (Flash tamamen boş ise, IOS yoksa CTRL+Break tuşlarına

basmaya da gerek yoktur. Zira IOS olmadığı zaman Router direk Rommon' dan açılır.) Bu durumda ilk bağlantımız 9600 bps ile yapıldığı için kopacaktır. Hyper Terminal' de bağlantı hızı 115200' e çıkarılarak yeniden bağlanılır. Rommon açılıp komut satırında "confreg" yazıldığında router bize değiştirmek istediğimiz bölümleri sıralayacak ve burada sadece konsol hızı için evet deyip uygun hızı seçeceğiz. Ve router' ı yeniden başlatmamız istenecek.

---

rommon 1 > confreg

do you wish to change the configuration? y/n [n]: y

enable "diağnostic mode"? y/n [n]: n

enable "use net in IP bcast address"? y/n [n]: n

enable "load rom after netboot fails"? y/n [n]: n

enable "use all zero broadcast"? y/n [n]: n

disable "break/abort has effect"

enable "ignore system config info"? y/n [n]: n

change console baud rate? y/n [n]: 7

change console baud rate? y/n [n]: y

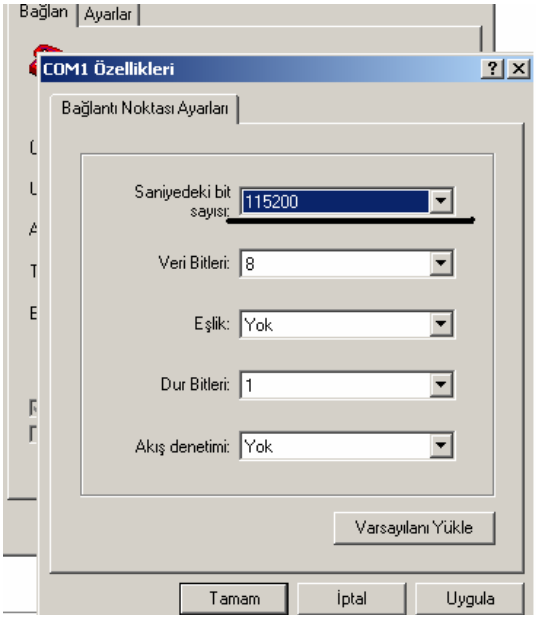
enter rate: 0 = 9600, 1 = 4800, 2 = 1200, 3 = 2400

4 = 19200, 5 = 38400, 6 = 57600, 7 = 115200 [0]: 7

change the boot characteriştics? y/n [n]: n

You must reset or power cycle for new config to take effect

Artık routerımız 115200 konsol hızıyla açılacak ve Xmodem iletişim kuralı kullanılarak Flash' ın yüklemesi yapılabilecektir. Bunun için Hyper Terminal' in "Dosya Gönder" özelliğinden faydalanacağız.



Roter'ı yeniden başlattığımız da konsol hızını 115200' e çıkarmalıyız...

```
rommon 3 > xmodem -c c2600-ix-mz.122-28.bin
Do not start the sending program yet...
         File size           Checksum   File name
   6080092 bytes (0x5cc65c)   0xd773    c2600-ix-mz.122-28

WARNING: All existing data in bootflash will be lost!
Invoke this application only for disaster recovery.
Do you wish to continue? y/n  [n]:  y
Ready to receive file c2600-ix-mz.122-28.bin ...
-
```

Komut Satırı

```
rommon 2 > xmodem ?
Do not start the sending program yet...
         File size           Checksum   File name
   6080092 bytes (0x5cc65c)   0xd773    c2600-ix-mz.122-28

WARNING: All existing data in bootflash will be lost!
Invoke this application only for disaster recovery.
Do you wish to continue? y/n  [n]:  y
Ready to receive file c2600-ix-mz.122-28.bin ...
```

Dosya adı ve yeri seçildi

Dosya Gönder

Klasör: C:\TFTP-Root

Dosya adı: C:\TFTP-Root\c2600-ix-mz.122-28 — Gözat...

İletişim Kuralı: Xmodem

Gönder | Kapat | İptal

Xmodem İletişim Kuralı Seçildi

```
monitor: command "c
rommon 2 > xmodem ?
Do not start the se
         File size
   6080092 bytes (0

WARNING: All existi
Invoke this applica
Do you wish to cont

Download aborted -
rommon 3 > xmodem -
rommon 3 > xmodem -
Do not start the se
         File size
   6080092 bytes (0

WARNING: All existing data in bootflash will be lost!
Invoke this application only for disaster recovery.
Do you wish to continue? y/n  [n]:  y
Ready to receive file c2600-ix-mz.122-28.bin ...
```

sa için Xmodem dosya gönderilmesi

Gönderilen: C:\TFTP-Root\c2600-ix-mz.122-28

Paket: 3720 Hata denetimi: CRC

Deneme: 0 Toplam deneme: 0

Son hata:

Dosya: ▪▪▪ 465k / 5938K

Geçen: 00:01:56 Kalan: 00:22:48 Üretilen iş: 4097 cps

İptal | cps/bps

Şu anda yükleme yapılıyor

IOS yüklenmesi tamamlandıktan sonra Routerımız açabiliriz. Fakat yapmamız gerekenler henüz bitmiş değil. Routerımızın konsol hızı hala 115200' de. İlk açtığımızda Hyper Terminal ile bu hızı göz önüne alarak bağlanıp, standardı sağlama için konsol hızını tekrar 9600 bps olarak değiştirmemiz gerekir.

Bunun için Konsol-line konfigürasyonuna girip "speed" komutuyla gerekli düzenlemeyi yapmalıyız.

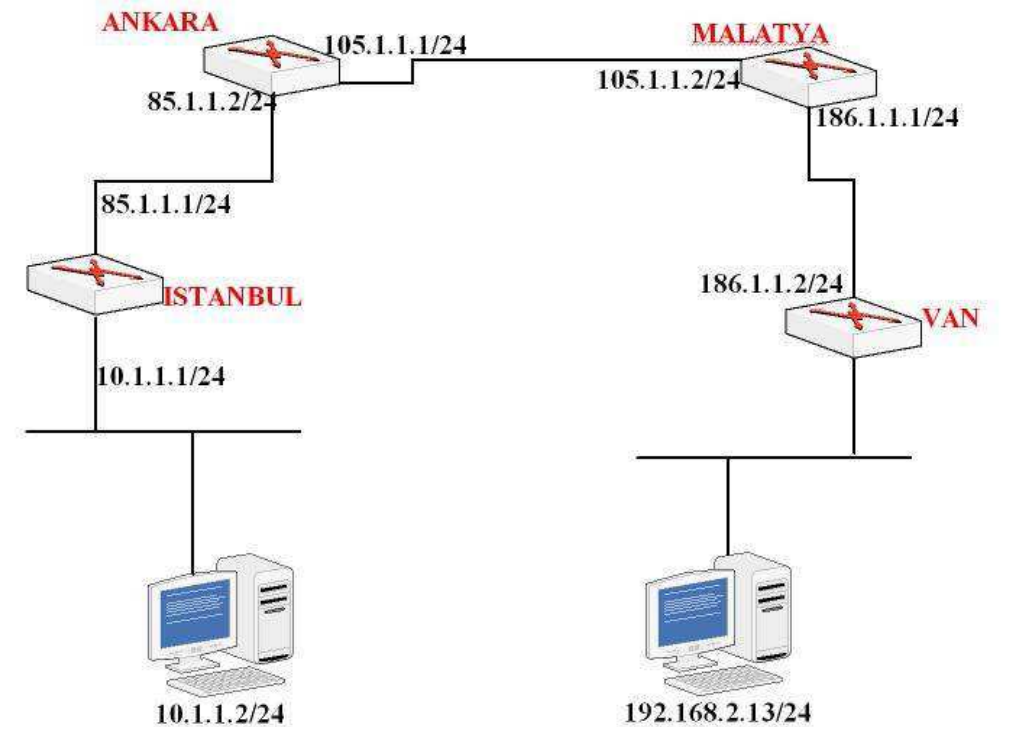
Ve bağlantımız kesildi çünkü Hyper Terminal ile bağlantımızı oluştururken konsol hızı olarak 115200 bps' ı seçmiştik. Bunu da eski haline getirmemiz gerekir.

```
Router(config)#
Router(config)#line con 0
Router(config-line)#speed ?
  <0-4294967295>  Transmit and receive speeds

Router(config-line)#speed 9600
_
```

ROUTING GİRİŞ

Routing en basit ifadeyle bir uzak networke gitmek için gereken yol bilgisinin Router' lar tarafından sağlanmasıdır. Routerlar kendilerine gelen paketlerde, hedef ip adresi olarak, nerede olduğunu ve nasıl gidileceğini bildikleri bir networkten adres bulunduğunda, hedefe yönlendirme yaparlar. Aksi takdirde paketi yok ederler.

Aşağıdaki senaryoyu biraz incelersek daha iyi fikir sahibi olabiliriz.



Burada routerlar üzerinde hiçbir yönlendirme konfigürasyonu yapmadığımızda, 10.1.1.2 bilgisayarından 192.168.2.13 bilgisayarına ping atarsak başarısız oluruz.

Peki neden ?

Çünkü İstanbul Router' ı 192.168.2.13 bilgisayarının bulunduğu network hakkında hiçbir bilgiye sahip değil. Router' lar üzerlerinde konfigürasyon yapılmadığından sadece kendileri (interface' lerine) direk bağlı olan network' leri bilirler. Bu durumda İstanbul Router' ının sadece 10.1.1.0 ve 85.1.1.0 network' lerini bildiğini söyleyebiliriz.

Eğer Router'ın gideceği ip numarası directly connected değil ise Router'a gideceği ip adresine nereden ulaşacağını belirtmemiz gerekir.

1. Routing işlem, Bir paketin bir Networkdeki bir aygıttan diğer Networkdeki bir aygıta gönderilmesidir.

2. Routerlar destination adrese sahiptirler.

3. Routerlar; bütün uzak Networklerin olası yollarını (routes) bilirler.

4. Routerlar; Uzak Networklerin en iyi(en kısa) yolunu kendileri seçerler. Bunu seçerken o anki duruma bakarlar ve belli bir kriter yoktur. O anki hattın yoğunluğuna bakabilir, aradaki mesafeye bakabilir…. En iyi yolu kendisi seçmektedir.

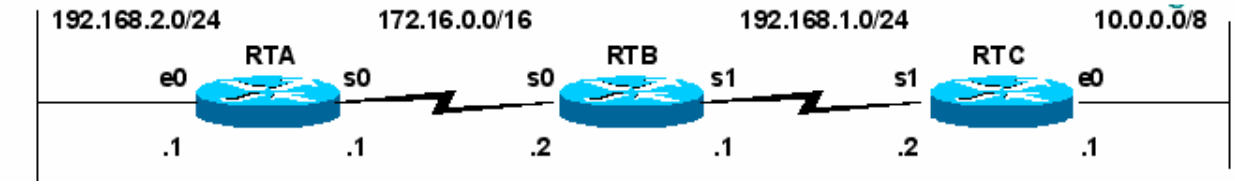
5. Routerler uzak Networklerin adreslerini oluşturdukları bir "Routing" tablosunda tutarlar. Bu bilgiler manuel olarak yada otomatik olarak tutulur. Manuel olarak tutulmasına **Static Routing** , Otomatik olarak tutulmasına **Dynamic Routing** denir.

Bu senaryo da İstanbul Router' ının bilmediği networkler uygun tanımlamalar yapılarak Router' a öğretilebilir. Peki İstanbul Router' ına bütün tanımlamaları yaptıktan sonra uzak bilgisayara ping atabilir miyiz ?

Hayır…

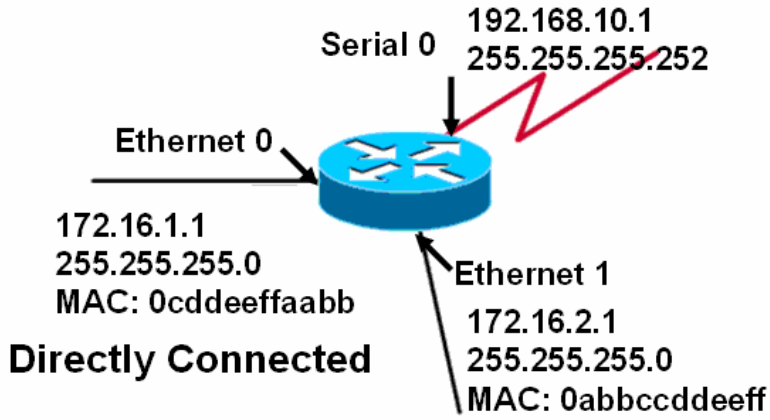
Biz sadece İstanbul Routerında Static Routing yaptık. Malatya Routerında hiçbir işlem yapmadığımızdan dolayı Malatya Routerı ping işlemine cevap vereceği ip adresine nasıl ulaşacağını bilemediği için Ping işlemi gerçekleşmeyecektir. (Ping işlemi iki yönlüdür, paket hedefe gider ve gelir.)

Daha önceden belirttiğimiz gibi Routerlar için Directly Connected networklerine herhangi bir yönlendirme yazmaya gerek yoktur. İki Routeri birbirine bağladığınızda ve interfacelerini uygun şekilde konfigure edip up durumuna getirdiğinizde Routing Tabla' larda o networkler ile ilgili bilgileri görürüz.



Böyle bir networkte interfaceleri up duruma getirdiğimizde Routing Tabla' lar aşağıdaki gibi olacaktır.

```
RTA#show ip route
Codes: C - connected,.....................................
C    172.16.0.0/16 is directly connected, Serial0
C    192.168.2.0/24 is directly connected, Ethernet0

RTB#show ip route
Codes: C - connected,.....................................
C    172.16.0.0/16 is directly connected, Serial0
C    192.168.1.0/24 is directly connected, Serial1

RTC#show ip route
Codes: C - connected,.....................................
C    10.0.0.0/8 is directly connected, Ethernet0
C    192.168.1.0/24 is directly connected, Serial1
```



ROUTING BASICS

Routerların temel işlevi yönlendirmek yapmaktır. Bunu yaparken Router Routing Tabla' ında bulunan bilgilerle hareket eder. Routing table' ı bizler static olarak tanımlayabildiğimiz gibi Routing Protokoller vasıtasıyla oluşmasını da sağlayabiliriz.

Anlaşılacağı gibi Routing işlemi iki ana başlık altında toplanabilir.

1. Static Routing

2. Dynamic Routing

Static Routing **Ip Route** komutu ile gerçekleştirilirken Dynamic Routing **Routing protokoller** yardımıyla gerçekleşir.

Static Routing özellikle küçük ölçekli networklerde kullanıldığında ideal bir çözüm olarak karşımıza çıkabilir fakat büyük ölçekli networklerde çalışmaya başladığımız andan itibaren hata yapma olasılığımız artacaktır.

Dynamic Routing ise konfigürasyonu çok çok kolay olduğu için, mantığı anlaşıldığı andan itibaren birçok fayda sağlayacaktır.

STATİC ROUTİNG

Az önce de bahsettiğimiz gibi static Routing **"ip route**" komutu ile Global Configuration modda yapılır ve küçük ölçekli networklerde ideal çözümdür.

Static Routing yapılırken hedef network adresi, subnet maskı ve bizi o hedefe götürecek bir sonraki routerın ip adresi bilinmelidir. Burada bir sonraki router ile ilgili bir kavram ortaya çıkıyor; **"next hop**". Bunlar bilindiğinde komut şu şekilde kullanılacaktır.

Router(config)#ip route [hedef adres][subnet mask][Next Hop] [distance]

Bu komut yönlendirme tablosundan silinmek istendiğinde ise basına "no" ifadesini yazmak yeterli olacaktır. Distance ifadesi seçimlik olup gerektiği durumlarda Routingler arasında önceliği belirlemeye yarayan Administrative Distance değerini değiştirmek için kullanılır. Static Routing için Administratice Distance default olarak "1" dir.

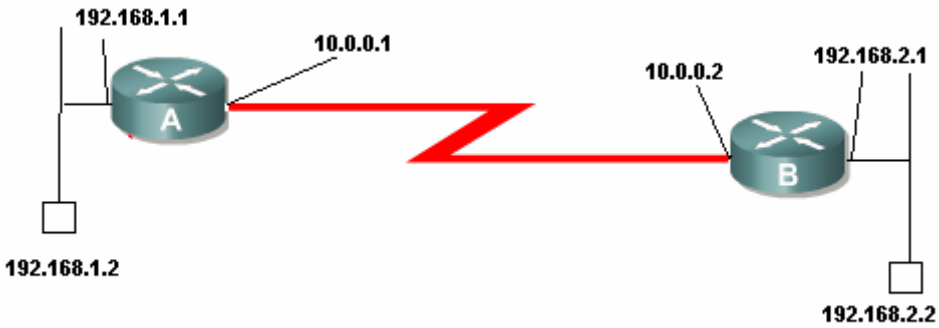
Default Administrative Distance değerleri şunlardır:

Yönlendirme Kaynağı	Varsayılan AD Değeri
Direkt fiziksel bağlantı	0
Statik yönlendirme	1
RIP	120
IGRP	100
EIGRP yönlendirme özeti	5
Internal EIGRP	90
External EIGRP	170
OSPF	110
Bilinmeyen yönlendirme	255

Router'da tanımlanmış statik kayıtları görmek için privileged modda iken "**show IP route**" komutunu kullanmalıyız. Karşımıza çıkan listedeki kayıtların başında bulunan **C** harfi fiziksel olarak birbirine bağlı ağlara olan yönlendirmeyi, **S** harfi yönlendirmenin statik olduğunu **S*** işareti ise kaydın default yönlendirme olduğunu gösterir.

Default yönlendirmenin router'larda çalışabilmesi için "**ip classless**" komutunun girilmesi gerekir. Ayrıca statik bir kaydı yönlendirme tablosunda silmek için **"no ip route**" komutunu parametreleriyle birlikte kullanmanız gerekir.

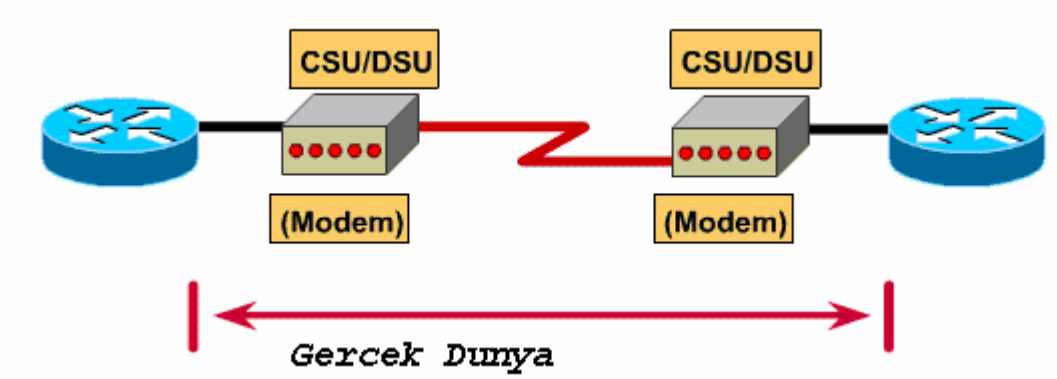
Static Routing, i örnek bir çalışma ile inceleyecek olursak;



192.168.1.0 ve 192.168.2.0 10.0.0.0 networklerimiz var. Bütün subnet masklarımız 255.255.255.0 olsun. Bu durumda Static Routing islemi her iki router için şu şekillerde gerçeklestirilmelidir. A Routerının serial 0 adresi 10.0.0.1ve DCE iken B routerının serial 0 adresi 10.0.0.2' dir.

NOT: Cisco Router'ların seri interface'leri DTE veya DCE olarak konfigure edilebilir. Bu özellik kullanılarak WAN bağlantıları simüle edilebilir. Bunun için birbirine bağlı Router'ların interface'lerinden bir tanesini DCE diğer Router'ın interface'sini ise DTE olarak kabul ediyoruz. Ardından DCE olarak kabul ettigimiz interface'in DTE olan interface clock sağlaması gerekiyor. DCE olarak kullanabileceğimiz interface'de "clock rate" komutunu kullanarak bir değer atamamız gerekiyor. Aksi halde bağlantı çalışmayacaktır. Örneğin;

RouterA(conf-if)#clock rate 64000



(CSU/ DSU)

Artık konfigurasyonumuza geçebliriz.

Router A için;

```
H(config)#
A(config)#ip route 192.168.2.0 255.255.255.0 10.0.0.2
A(config)#_
```

Router B için;

```
B(config)#
B(config)#ip route 192.168.1.0 255.255.255.0 10.0.0.1
B(config)#
B(config)#
B(config)#
```

Burada Next hop olarak her iki konfigurasyonda da bir sonraki routerin ip adresi seçildi. Zaten sistem de 2 tane Router olduğu için bir sorun yaşamadık. Bu noktada hedef networke ulaşmak için birden fazla Router geçildiği zaman next hop olarak hangisi seçilmelidir sorusu aklımıza gelebilir.

Next Hop olarak o routerlardan herhangi biri seçilebilir, burda önemli olan konfigurasyonlar bittiği zaman Routerimizin next hop adresine nasıl ulaşacağını bilip bilmediğidir.

Routerların konfigurasyonları ve problem çözümü aşamasında running-config dosyalarının incelenmesi önemlidir, Çünkü bu dosyada yaptigimiz her konfigurasyon adimini görebiliriz.

Şimdi topolojimizdeki A routeri için running-config dosyalarına bir göz atalım.

---

hostname A

!

enable secret 5 1gZBQ$yyxVv/2B4uq7pROiHGRhg/

!

!

memory-size iomem 10

ip subnet-zero

!

voice-port 1/0/0

!

voice-port 1/0/1

!

!

interface Ethernet0/0

ip address 192.168.1.1 255.255.255.0

!

interface Serial0/0

ip address 10.0.0.1 255.255.255.0

no fair-queue

clockrate 64000

!

interface BRI0/0

no ip address

shutdown

isdn x25 static-tei 0

!

ip classless

ip route 192.168.2.0 255.255.255.0 10.0.0.2

ip http server

!

!

line con 0

line aux 0

line vty 0 4

!

no scheduler allocate

end

NOT: CCNA Sınavlarında DCE ve DTE olacak interface belirtilmektedir ve konfigürasyon sorularında DCE olan interface' lere Clock Rate verilmelidir.

"Show ip route" komutu ile yönlendirme tablosunu görebiliriz.

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

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.0.0.0 is directly connected, Serial0/0
C    192.168.1.0/24 is directly connected, Ethernet0/0
S    192.168.2.0/24 [1/0] via 10.0.0.2
RouterA#_
```

(RouterA için yönlendirme tablosu)

Yönlendirme tablosunda "S" ile başlayan satırlar Statik bir yönlendirme yapıldığını ve şekilden hareketle bu yönlendirmenin 192.168.2.0 network' üne, 10.0.0.2 next hop'undan giderek olduğunu söyler. Bu tabloda C ile başlayan satırlar ise A router' ının interfacelerine direk olarak bağlanmış networkleri gösterir ve bu networklere **"Directly Connected**" networkler denir. Roterlar kendi Directly Connected networklerini bilirler ve bu networkler ulaşmak için yönlendirme yapılmasına gerek yoktur.

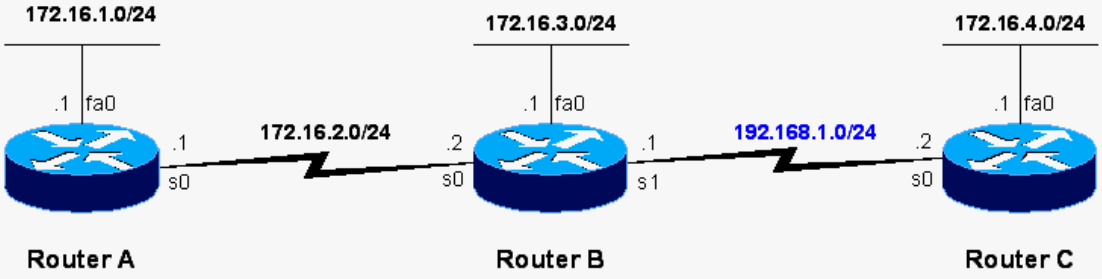
Senaryomuzda hiçbir yönlendirme yapmasaydık bile A router'ınan ethernet interface' ine bağlı bir bilgisayardan B router' ının serial interface' ine ping atabilirdik. Bunun için tek yapmamız gereken sey, o bilgisayarda Default Gateway' i (Varsayılan Ağ Geçidi) 192.168.1.1 olarak konfigüre etmektir.

```
RouterA#show ip interface brief
Interface                  IP-Address      OK? Method Status                Prot
ocol
Ethernet0/0                192.168.1.1     YES NVRAM  up                    up

Serial0/0                  10.0.0.1        YES manual up                    up
```

("show ip interface brief" komutu ile interface' leri durumunun görüntülenmesi)

Routing Table

Routing Table konfigurasyonlarımız ve projelerimiz sırasında problem teşhisimiz açısından çok önemlidir. İyi bir network yöneticisi running-config ve routing table' a hakim olmalıdır. İşimizi Routing olduğu durumda Routing Table bir numaralı yardımcımız olacaktır.




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

Gateway of last resort is not set

RouterB#
```

İşte örnek bir network topolojisi B routerinin Roting Table'i. Routing Table göründüğü üzere bomboş. Burada Directky Connected networklerin bile görünmemesinden dolayı interfaceler ile ilgili bir sorun olduğundan bahsedilebilir. Sorunun ne olduğu ile ilgili bilgiyi "show ip interfaces brief" komutu ile görüntüleyebiliriz. Şu anda anlamamız gereken nokta, eğer bir interface sebebi ne olursa olsun "down" ise o interface bağlı network Routing Table, da görünmez!

```
RouterB(config)#inter s 0
RouterB(config-if)#ip add 172.16.2.2 255.255.255.0
RouterB(config-if)#end
```

```
RouterB(config)#interface s 1
RouterB(config-if)#ip add 192.168.1.1 255.255.255.0
RouterB(config-if)#no shutdown


RouterB(config)#interface fastethernet 0
RouterB(config-if)#ip add 172.16.3.1 255.255.255.0
RouterB(config-if)#no shutdown
```

Router B için interfaceleri up duruma getirdikten sonra Artık en azından Rouring Table' imizda Directly Connected networklerimizi görmemiz gerekir.

```
RouterB#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
<text omitted>

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.2.0 is directly connected, Serial0
C       172.16.3.0 is directly connected, FastEthernet0
C    192.168.1.0/24 is directly connected, Serial1
RouterB#
```

Burada 172.16.0.0 networkune Parent route ve o networkun subnetworku olan 172.16.2.0 - 172.16.3.0 networklerine Child Route denir.

Bilindiği gibi Static Route yazılırken Routerin interface' i yada Next Hop ip adresi kullanılabilir. Routerin interface' i kullanıldığında o static route satırı interface ile direk bağlı bir network gibi görünecektir.

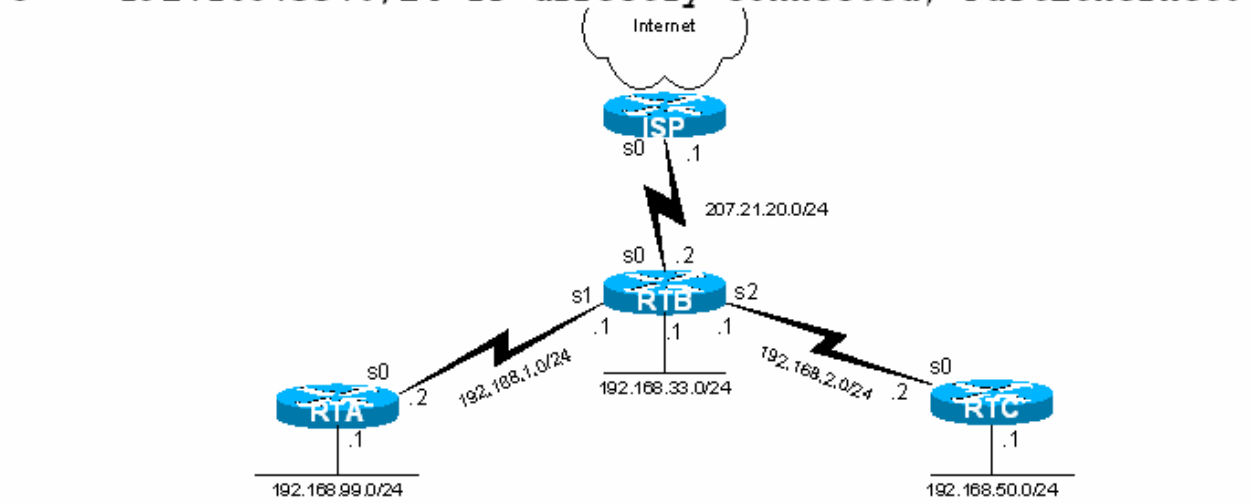
```
RouterB(config)#ip route 172.16.1.0 255.255.255.0 serial 0
RouterB#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
<text omitted>

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 3 subnets
S       172.16.1.0 is directly connected, Serial0
C       172.16.2.0 is directly connected, Serial0
C       172.16.3.0 is directly connected, FastEthernet0
C    192.168.1.0/24 is directly connected, Serial1
RouterB#
```

Next Hop ip adresi kullanıldığında ise Routing Table o networke verilen ip adresi ile ulaşabileceğini gösteren satır yer alacaktır.

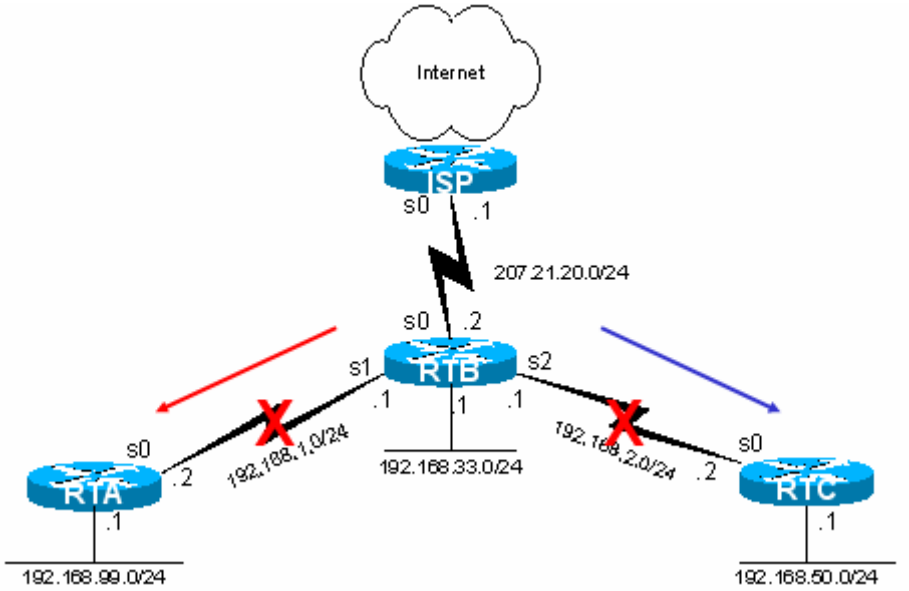
```
RouterB(config)#ip route 172.16.1.0 255.255.255.0 172.16.2.1
RouterB#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
<text omitted>

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 3 subnets
S       172.16.1.0/24 [1/0] via 172.16.2.1
C       172.16.2.0 is directly connected, FastEthernet1
C       172.16.3.0 is directly connected, FastEthernet0
C    192.168.1.0/24 is directly connected, Serial1
RouterB#
```

Interfaceler up oldğu sürece Routing Tablelarda bozulma olmayacaktır.

```
C    207.21.20.0/24 is directly connected, Serial0
S    192.168.99.0/24 [1/0] via 192.168.1.2
S    192.168.50.0/24 is directly connected, Serial2
C    192.168.1.0/24 is directly connected, Serial1
C    192.168.2.0/24 is directly connected, Serial2
C    192.168.33.0/24 is directly connected, FastEthernet0
```



Örneğin şekildeki yapı içerişinde RTB Routeri için bütün interfaceler up durumdayken Routing Table sorunsuz görünüyor. 192.168.1.0 ve 192.168.2.0 networklerinin bulunduğu interfacelerin bir an için down olduğunu düşünelim.



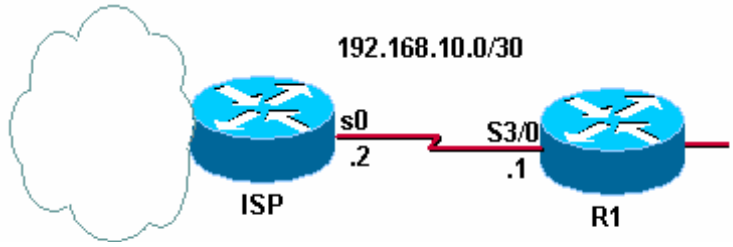
Bu durumda Routing Table RTB için şu şekilde olacaktır.

```
RTB#show ip route
C    207.21.20.0/24 is directly connected, Serial0
C    192.168.33.0/24 is directly connected, FastEthernet0
```

Söz konusu Interfacelere direk bağlı olan networkler ve o interfaceleri kullanarak yazılan Static Route satırları Artık Routing Table' da yoklar.

Default Routing

Burada ISP Routeri ile bağlanılan networke (Internet) R1 üzerinde default route yazılarak ulaşılabiliecektir.



Default hedefi bilinmeyen paketleri yönlendirmek için yazilabilecek Route satırıdır şeklinde tanımlanabilir.

R1(config)#ip route 0.0.0.0 0.0.0.0 Serial3/0 yada

R1(config)#ip route 0.0.0.0 0.0.0.0 192.168.10.2

Şeklinde yazılabilir.

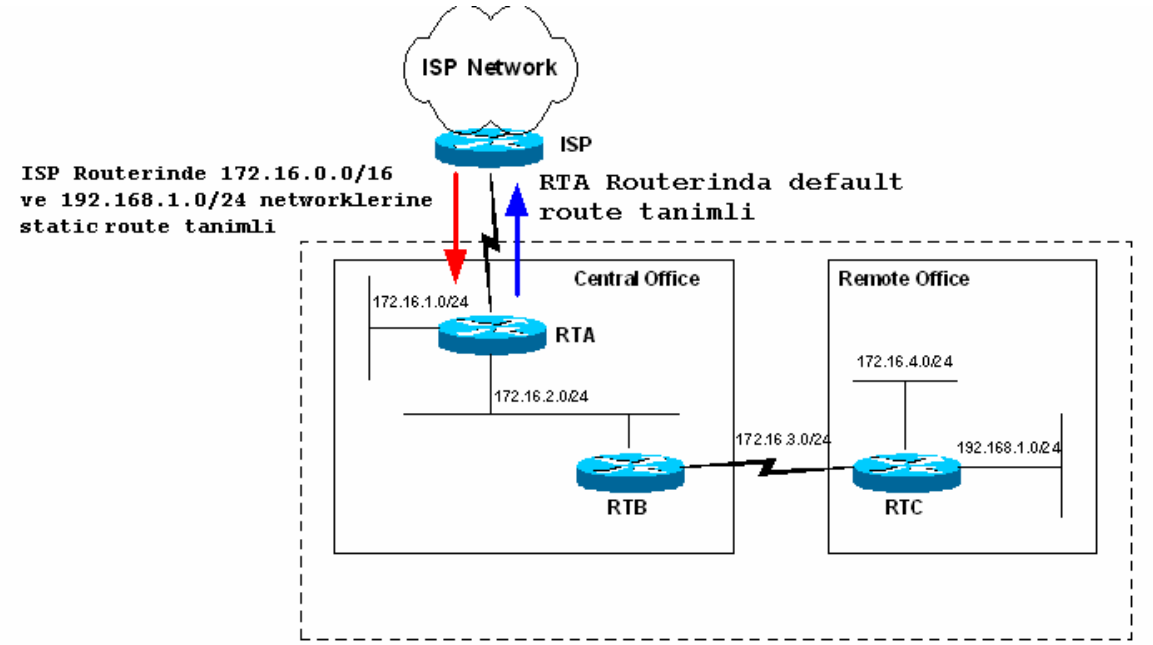
Routing Table' da aşağıdaki gibi görünecektir.

```
     192.168.10.0/30 is subnetted, 1 subnets
C       192.168.10.0 is directly connected, Serial3/0
S*   0.0.0.0/0 is directly connected, Serial3/0
```

Exstra

Default tanımlamak bazen sorunlar ile birlikte gelebilir. Çünkü üzerinde Default tanımlı ve bu route satırı ile paketleri internete gönderen bir router, sisteminde bulunan diğer networklere olan yolu down olduğunda o networklere gelen paketleri de default router satırına göre değerlendirecektir.

Örnek üzerinde incelemek gerekirse;



Böyle bir yapı içerişinde RTB ve RTC arasındaki bağlantının down olduğunu düşünürsek Routing Table lar update edildikten sonra RTA ve RTB routerları 172.16.4.0 ve 192.168.1.0 hedef networklerine giden yolları bilmedikleri için hedefinde bu networkler bulunan paketleri default route satırından hareketle ISP routerina gönderecekti.

ISP Routeri da kendine gelen bu paketleri üzerinde tanımlı statik route satırlarından hareketle tekrar geri gönderecek ve bu sebeple bir döngü oluşmasına sebep olacaktır. Bu döngü IP başlığındaki TTL (Time – to - live) alani sıfırlanana kadar devam edecektir.

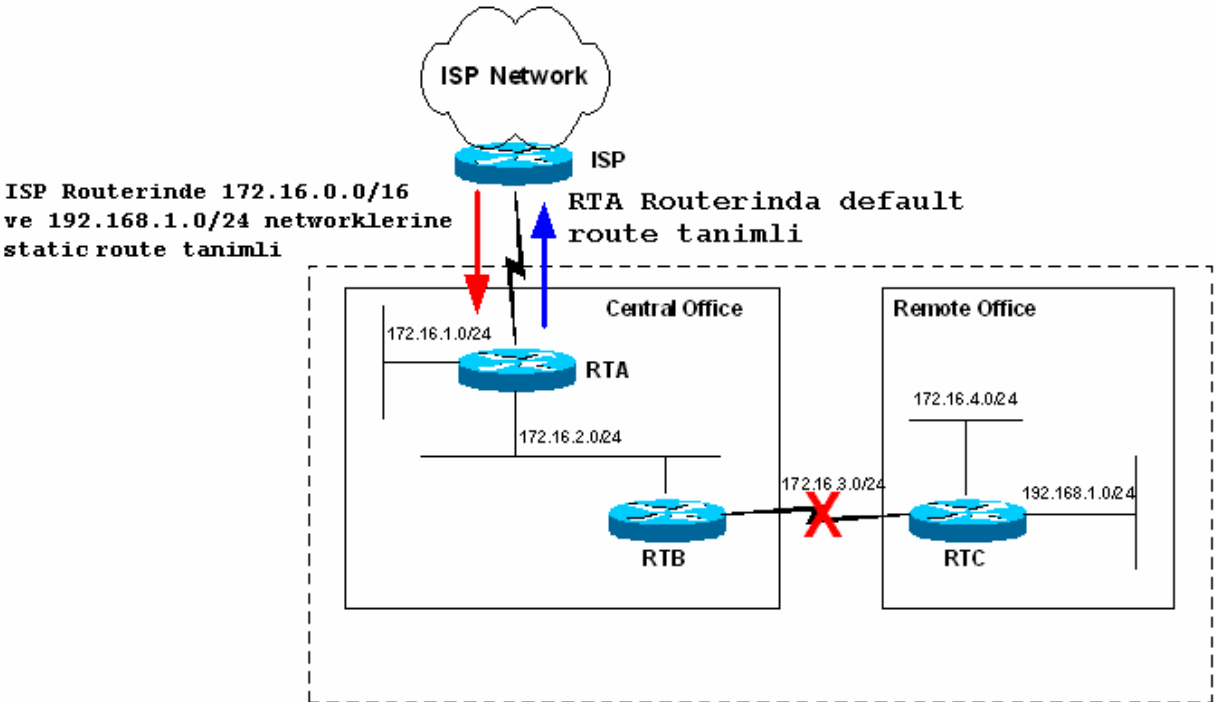
Bunun için kullanılacak çözüm RTA üzerinde Discard Route denen tanımlamayı yapmaktır.

Discar Route routing table' da bir eşleşme olmadığında ve default route' un işletilmesi istenmediğinde kullanılır ve paketler null0 ' a gönderilir.

Örneğin;

RTA(config)#ip route 172.16.0.0 255.255.0.0. null0

Satırı ile RTA Routeri kendisine gelen hedefinde 172.16.0.0 networku bulunan paketleri drop edecektir.



Böyle bir durumda bir başka çözümde "no ip classless" komutunu kullanmaktır.

Bu komut kullanıldıktan sonra Router Söz gelimi hedef ip adresi 172.16.4.9 olan bir paket için routing table' ina bakacak ve en uygun yolu arayacak. Bu durumda parent network olarak 172.16.0.0. networkunu ve bu networkun altında bilinen 172.16.1.0, 172.16.2.0 networklerini bulacak.

"no ip classless" ile konfigure edilmiş bir router her ne kadar parent networklerde 172.16.0.0 olsa da 172.16.4.9 ip adresini içeren 172.16.4.0 networku bilinen networkler arasında olmadığı için paketi drop edecektir.

Fakat bu çözüm önerilen bir çözüm değildir. Örneğimiz içerişide de var olan Örneğin 192.168.1.0 gibi bir networkte ise yaramaz. Çünkü bu network herhangi bir parent networkun subnetworku değildir.

Dolayısıyla bu network için null0 kullanılmalıdır.

DYNAMİC ROUTİNG

Static Routing ile çalışmalarımız sırasında Router' a ihtiyacı olan balıkları verdik ama artık balık ihtiyacı arttı yani networkler büyümeye başladılar. Dolayısıyla artık onlara balık tutmayı ögretmenin zamanı da geldi. :=)

Dynamic Routing' te Static Routing' de olduğu gibi sabit bir tanımlama yapmak yerine her Router'a kendi Directly Connected networklerini, çeşitli Routing Protokoller ile tanımlıyoruz. Ve ilgili Routing protokolün çalışma mantığına göre en iyi yol seçimi (Best Path Determination) Router tarafında gerçeklestiriliyor.

Burada bahsettigimiz Routing Protokolleri üç başlık altında incelememiz mümkün.

1. Distance Vector Protokoller (RIP, IGRP)

2. Link State Protokoller (OSPF)

3. Hybrid Protokoller (EIGRP)

Distance Vector protokoller routing table update mantığıyla çalışırlar. Yani belirli zaman aralıklarında sahip oldukları network bilgilerini komşu routerlarına gönderirler ve komşu routerlarından da aynı bilgileri alırlar. Bu döngünün sonunda her router sistemdeki bütün networkler öğrenmiş olur ve uygun yol seçimini yapar.

Link State Protokoller ise sürekli bir update yapmak yerine, komşu routerlarının up olup olmadıklarını anlamak için küçük "Hello" paketleri gönderirler. Sadece gerektigi zamanlarda, yeni bir router ortama eklendiğinde veya bir router down olduğunda, sadece o bilgi ile ilgili update gerçekleştirirler.

Hybrid Protokoller hem Distance Vector hem de Link State protokollerin bazı özelliklerini taşır. Bu gruba üye olan EIGRP Cisco tarafından ortaya çıkarılmıştır ve sadece Cisco routerlarda çalışır.

Her gruba üye olan protokoller ile ilgili detaylı bilgi ilerleyen başlıklar altında verilecektir.

DİSTANCE VECTOR PROTOKOLLER

RIP (RIPv1)

Rip (Routing Information Protocol) en iyi yol seçimi yaparken tek kriter olarak hop sayısına bakar. Rip tanımlanarak oluşturulmuş bir networkte maksimum hop sayısı 15' dir ve 16. hop' tan sonra Destination Unreachable hatası verecektir.

Rip ile tanımlanan routerlar her 30 saniyede bir kendisinde tanımlı olan networkleri komşu routerlarına iletirler. Burada dikkat edilmesi gereken bir konu, RIP ile tanımlanan bir networkün bağlı bulunduğu interface' i, aynı zaman da routing update gönderilecek bir interface olarak seçiyor olmamızdır.

Rip classfull bir routing protokoldür. Yani konfigürasyon sırasında subnet mask girilemez ve subnet masklar update sırasında ip adresinin sınıfına ait subnet mask seçilerek gönderilir.

Rip konfigürasyonu diğer bürün routing protokoller de olduğu gibi oldukça basittir. (Bütün subnet maslar 255.255.255.0)

RIP üç farklı sayaç (timer) kullanarak performansını ayarlar. Bu sayaçlar şunlardır;

- Route Update Timer: Router'ın komşularına, yönlendirme tablosunun tümünü göndermesi için beklediği zaman aralığı. Tipik olarak 30 sn.'dir.

- Route İnvalid Timer: Bir yönlendirmenin, yönlendirme tablosunda geçersiz olarak kabul edilmesi için geçmesi gereken zaman aralığı. 90 sn.'lik bu zaman aralığında yönlendirme tablosundaki bir yönlendirme kaydıyla alakalı bir güncelleme olmazsa o kayıt geçersiz olarak işaretlenir. Ardından komşu router'lara bu yönlendirmenin geçersiz olduğu bildirilir.

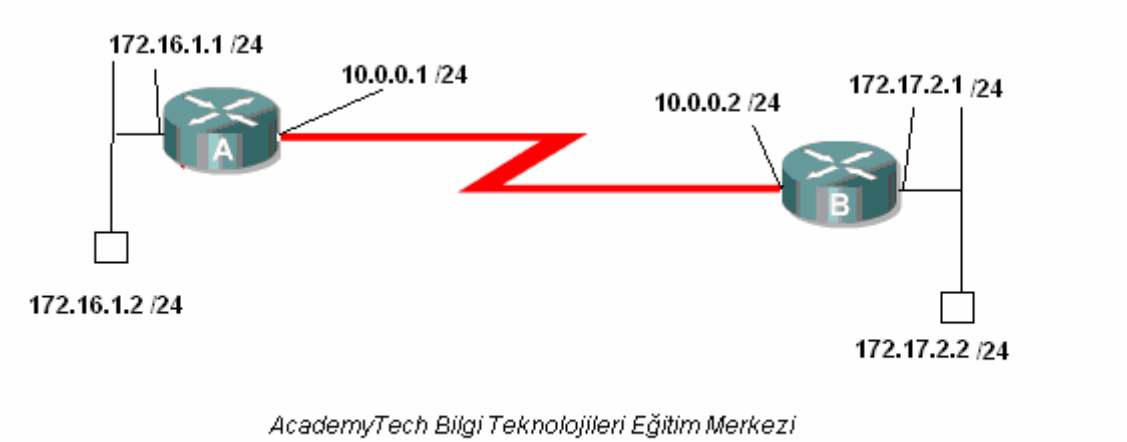
- Route Flush Timer: Bir yönlendirmenin geçersiz olması ve yönlendirme tablosundan kaldırılması için gereken zaman aralığı(240 sn.).

RIP'ı router üzerinde çalıştırmak için global konfigürasyon modunda "**router rip**" komutunu girmeliyiz.

RouterA(config)#router rip

Ardından router'a hangi network'e ait olduğunu bildiren "network" komutunu girmeliyiz.

RouterA(config-router)#network 172.16.0.0



Bu senaryoyu Rip ile konfigüre edecek olursak;

```
RouterA(config)#router rip
RouterA(config-router)#net
RouterA(config-router)#network 172.16.1.0
RouterA(config-router)#net
RouterA(config-router)#network 10.0.0.0
RouterA(config-router)#exit
RouterA(config)#_
```

(RouterA Rip Konfigürasyonu)

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

Gateway of last resort is not set

R    172.17.0.0/16 [120/1] via 10.0.0.2, 00:00:17, Serial0/0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.1.0 is directly connected, Ethernet0/0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.0.0.0 is directly connected, Serial0/0
RouterA#_
```

(RouterA Yönlendirme Tablosu)

```
RouterB(config)#router rip
RouterB(config-router)#network 172.17.2.0
RouterB(config-router)#net
RouterB(config-router)#network 10.0.0.0
RouterB(config-router)#
RouterB(config-router)#exit
RouterB(config)#
```

(RouterB Rip Konfigürasyonu)

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

Gateway of last resort is not set

     172.17.0.0/24 is subnetted, 1 subnets
C       172.17.2.0 is directly connected, Ethernet0/0
R    172.16.0.0/16 [120/1] via 10.0.0.1, 00:00:00, Serial0/0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.0.0.0 is directly connected, Serial0/0
RouterB#_
```

(RouterB Yönlendirme tablosu)

Routing Table' ımıza "**Show ip route**" komutu ile baktığımız da başında R harfi bulunan satırlar görüyoruz. Buradan çıkartacağımız anlam şu: Bu satırlarda belirtilen networklerin bilgisi Rip protokol sayesinde başka routerlardan update yoluyla gönderildi.

Yine Routing Table dikkatli izlendiğinde köşeli parantez içindeki [120/1] gibi ifadeler görünüyor. Burada 120 Rip protokol için Administrative Distinct denen ve routing protokoller arasında ki önceliği belirleyen değerdir. Diğer ifade da "n" gibi bir sayıdır (burada 1) ve hedef networke ulaşmak için asılacak hop sayısıdır.

```
RouterA#sh ip interface brief
Interface                  IP-Address      OK? Method Status                Prot
ocol
Ethernet0/0                172.16.1.1      YES manual up                    up

Serial0/0                  10.0.0.1        YES manual up                    up

Serial0/1                  unassigned      YES unset  administratively down down

RouterA#ping 172.17.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.17.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/33 ms
RouterA#_
```

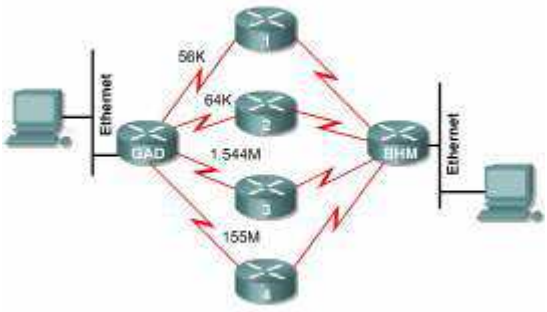
Rip protokolü updatelerini broadcast adresi olan 255.255.255.255 ip' sinden yapar.

"debug ip rip" komutunu verdiğimizde bunu açıkca görürüz.

```
RouterA#debug ip rip
RIP protocol debugging is on
RouterA#
00:37:18: RIP: sending v1 update to 255.255.255.255 via Ethernet0/0 (172.16.1.1)

00:37:18: RIP: build update entries
00:37:18:       network 10.0.0.0 metric 1
00:37:18:       network 172.17.0.0 metric 2
00:37:18: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (10.0.0.1)
00:37:18: RIP: build update entries
00:37:18:       network 172.16.0.0 metric 1_
```

Rip Load Balancing

Load Balancing tam olarak yükü birden fazla yol arasında dağıtmak demektir.



(Routerlar metrikler eşit olduğu için load balancing yapar)

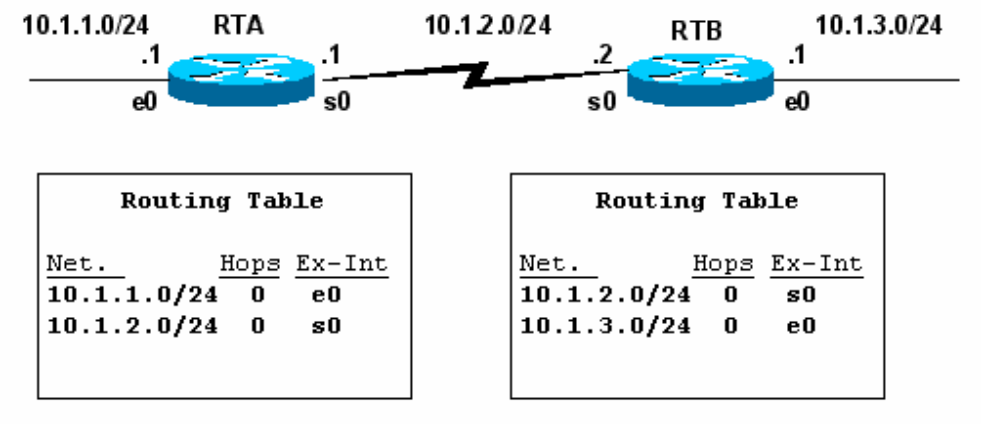
Mantıksal olarak düşündügümüz de Rip' in load balancing yapma ihtimali her zaman vardır. Çünkü referans olarak bir tek hop sayısına bakar. Oysa diğer protokoller de load balancing ihtimali en iyi yol seçimi sırasında bir çok kriter göz önüne alındığı için mucize derecesinde zayıf bir ihtimaldir. Fakat ileride değineceğimiz IGRP ve EIGRP protokollerinde fazladan bir komut kullanarak Routerın load balancing yapması sağlanabilir.

Split Horizon

Bir Router kendi directly connected networkünü başka bir router'dan da öğrenirse öğrendiği bilgiyi çöpe atar.

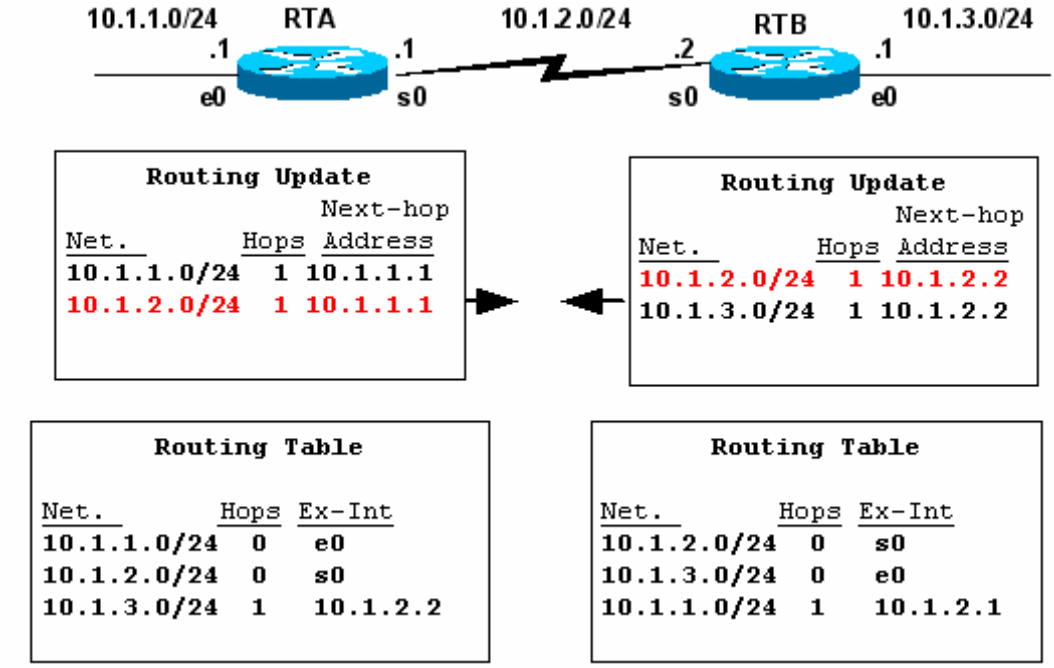
Ayrıca router'ın ağ üzerinde herhangi bir değişiklik olduğunu anladığında bu değişikliği, öğrendiği interface hariçindeki interface'lerden yayınlamasını sağlar. Böylece router'lar değişikliği sadece bir yönde yayınlarlar.

Aşağıdaki örnek ile Split Horizon kuralını detaylı anlayabiliriz.



İki adet Routerimiz var ve başlangıca Routing Table' lar şekildeki gibi oluşmuş durumda yani sadece Directly Connected networkleri biliyorlar.

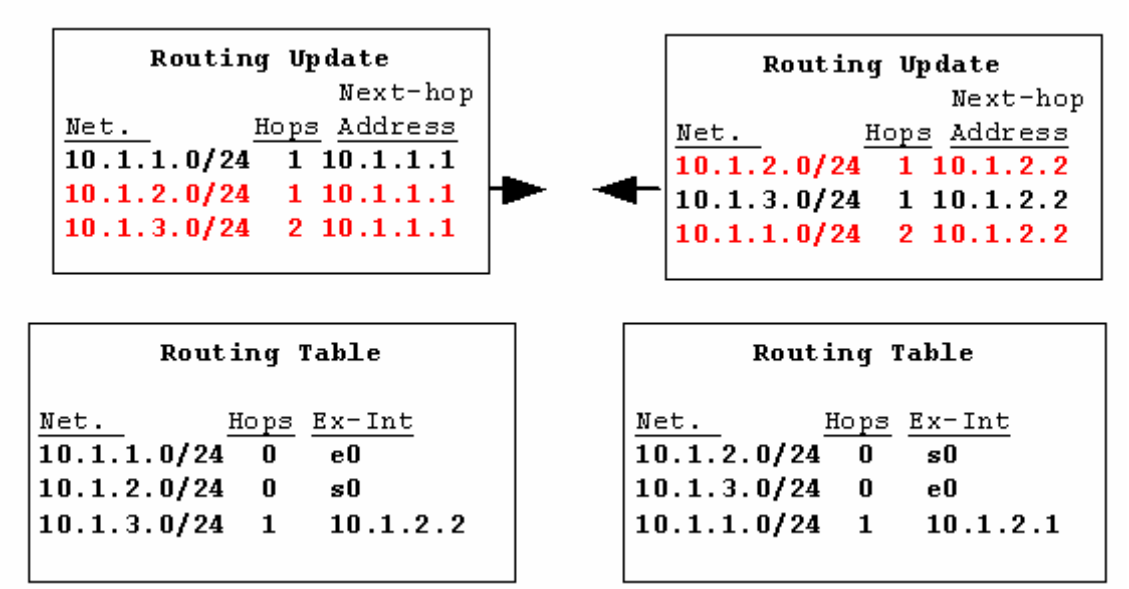
Split Horizon Disable edildiği zaman Routerlar Routing Table' larında ki bütün networkleri ve herhangi bir interfacelerinden öğrendikleri bütün networkleri update edeceklerdir.



Routerlar şekilde gösterilen updateleri komşu Routerlarına yapacaklar. Burada kırmızı ile gösterilmiş Directly Connected networklerinde update edildiğine dikkat edin. Bu update Split Horizon, un disable olmasının sonucudur.

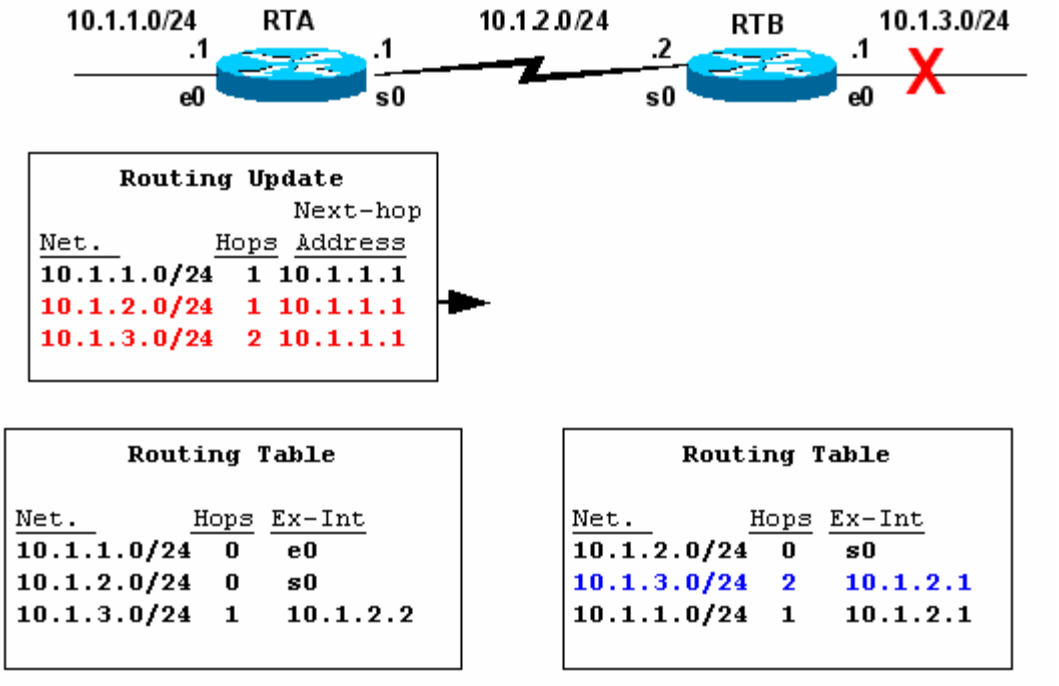
Routing Table incelendiğinde bir sorun yok gibi görünüyor. Gerçekten de yok, Çünkü split horizonun disable olmasından kaynaklanan updateler daha yüksek metriğe sahip olduğu için Routing table' larda yer almadı.

Şimdi bir sonraki updatelere bakalım.

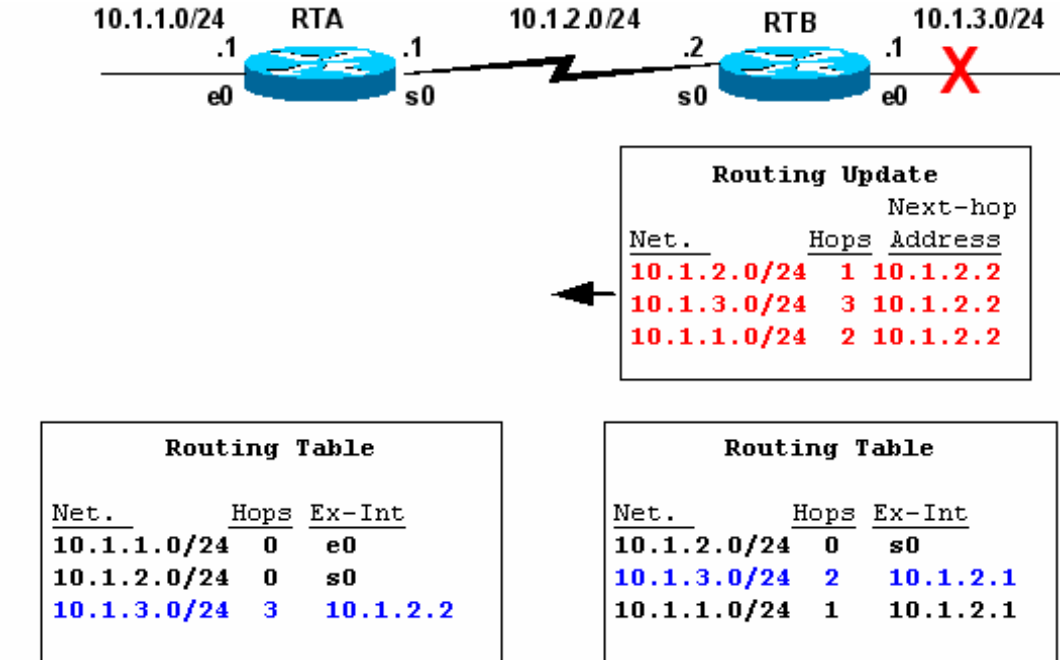


Burada da aslında update edilmemesi gereken networkler update edilmiş olmasına rağmen bir sorun yok Çünkü o networkler daha büyük metrik ile update ediliyor. Örneğin RTA Routeri 10.1.3.0 networkunu serial 0 interfaceinden aldığı için split horizon disable edilmemiş olsaydı o interfaceden geriye update etmeyecekti.

Bu ana kadar bir sorun olmadı ama bir an için RTB Routerina bağlı olan 10.1.3.0 networkunun down olduğunu varsayalım.

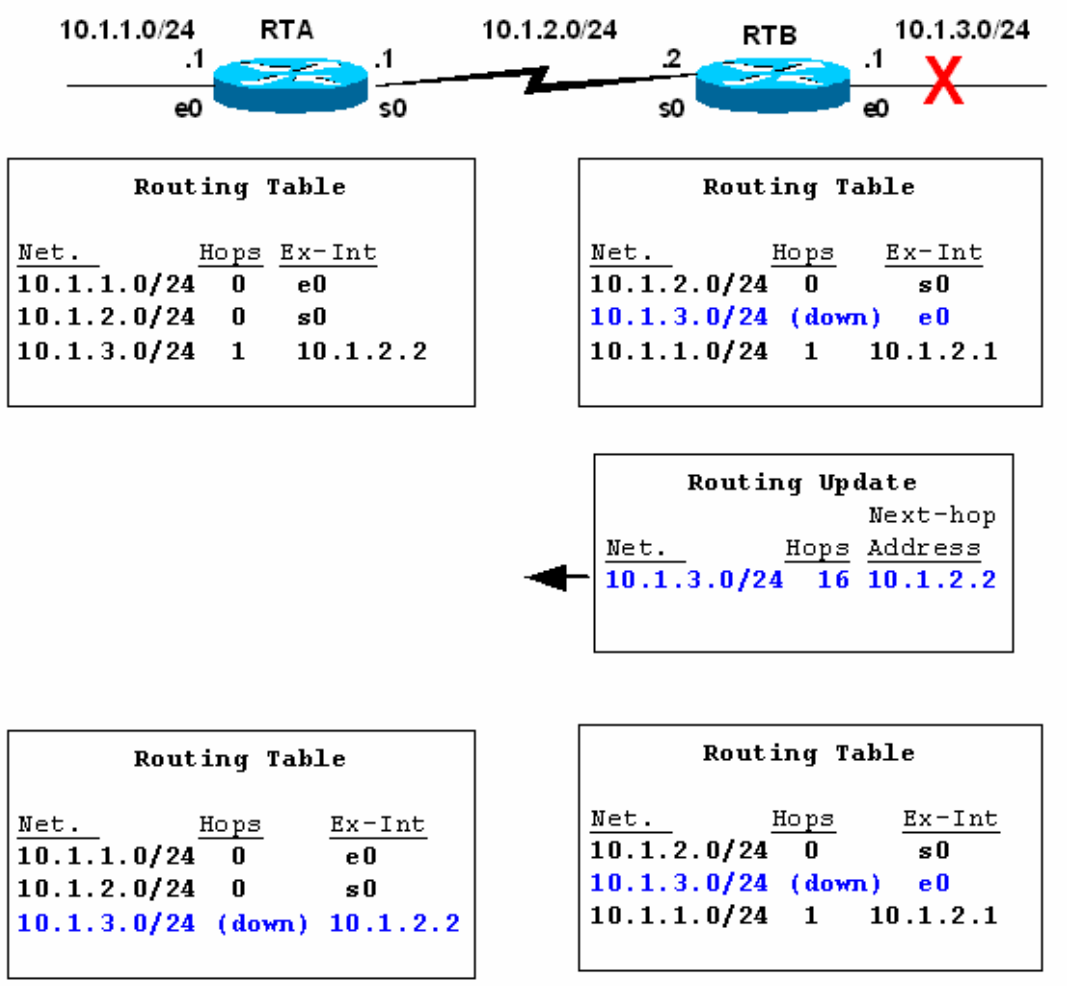


Bu durumda RTB routerina RTA routerindan 10.1.3.0 netwroku kendisine bağlı olan network down olduğu için daha küçük metrik ile geliyormuş gibi olacak ve RTB routerinin Routing Table' inda şekildeki gibi yer alacak. Şu anda RTB Routeri bir süre önce kendisine direk bağlı olan networke diğer router üzerinden 2 hop geçerek gidebileceğini sanıyor. RTB updatelerini üstelik yanlış olan Routing Table,' ina dayanarak yapacaktır.



Ve bu updatelerden sonra RTA routerinin Routing Table, ida 10.1.3.0 networkune 3 hop ile gidilebileceği kanısında. Bu döngü ta ki hop sayısı 16 oluncaya kadar devam edecektir. (Rip maximum 16 ho ilerleyebilir)

Bu döngünün engellenmesi Split Horizon ile mümkündür.



Split Horizon enable olduğunda RTB routeri anında Triggered Update gönderir komşu routerina ve bu update bilgisi Söz konusu networkun 16 hop ile ulaşılacağı şeklindedir ki rip Söz konusu olduğunda bu RTA nin da o networku down olarak varsayacağı anlamına gelir.

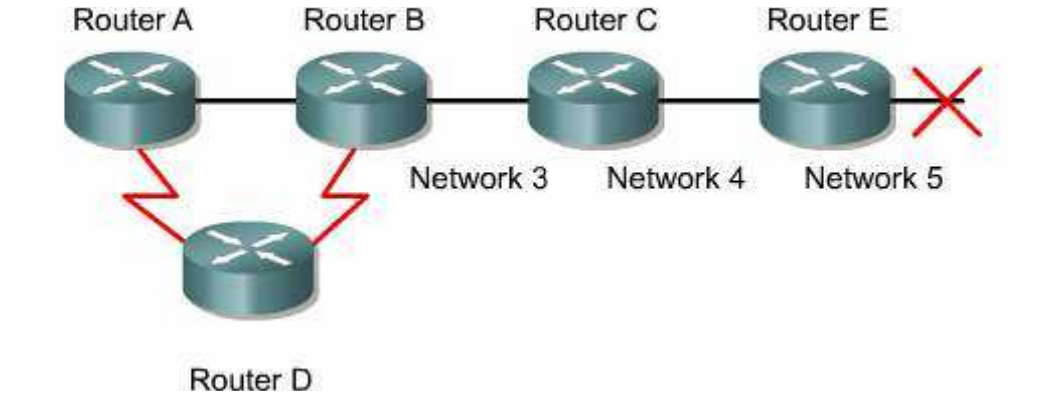
Dolayısıyla her ikir router da 10.2.3.0 networku için Hol Down Timer' i başlatırlar.

Bu çalışma yapısına Split Horizon with Poisen Reverse denir ki Routerlar, da default olarak enable durumdadır. Disable edilme gereken zamanlar da ki CCNA 4 içerinde bu konudan bahsedeceğiz, aşağıdaki komut kullanılabilir.

```
Router(config-if)# no ip split-horizon
```

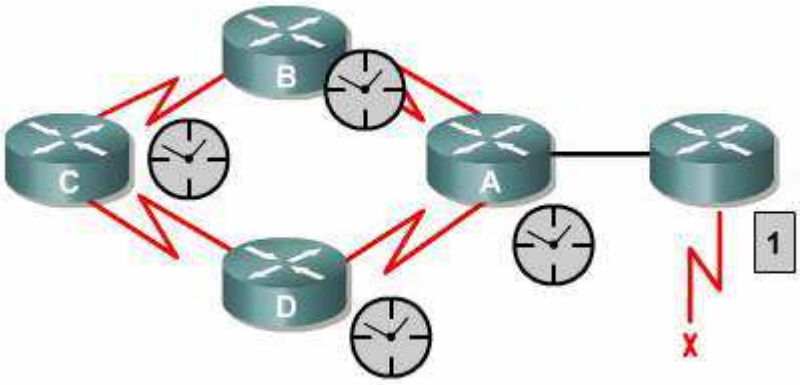
Route Poisoning

Router'ların yönlendirme tablosuna hop count değer 16 olarak yazılan bir yönlendirmedir ve hedef adresin erişilemez olduğunun router'lar arasında bilinmesini sağlar.



Holddown Timers

Bu teknikte hold-down sayıcılar router'ın komşusundan aldığı ulaşılamaz bir ağa ait güncelleme ile başlar. Eğer aynı komşudan aynı ağa ait daha iyi bir metrik değerine sahip bir güncelleme bilgisi alırsa hold-down kaldırılır. Fakat hold-down değeri dolmadan aynı komşudan daha düşük bir metrik değerine sahip bir güncelleme gelirse bu kabul edilmez.



Triggered Updates

Routing Table, da bir değişiklik olduğu anda Routerlar tarafinda gönderilen updatelerdir. Topoloji değiştiği anda bunu farkeden Router periodic update süresini beklemeden değişikliği komşu Routerlarına bildirir.

Triggered Updateler Route Poisoning ile tümleşik çalışırlar.

Extralar

Timers Basic ve uptede timer komutları ile Rip update, holdwoen v.s süreleri değiştirilebilir.

Router(config-router)#**timers basic** update invalid holddown flush

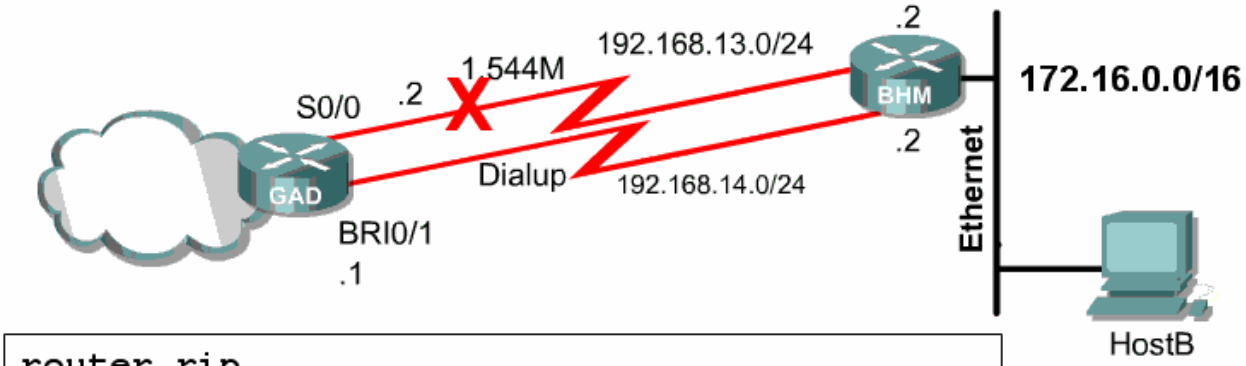
Router(config-router)#**update-timer** seconds

Rip ve Floating Static Route

Floating Static Routelar backup route olarak tanımlanmış route' lardir. Bu Route' lar reel olarak çalışan route lara göre daha yüksek bir Administrative Distance ile konfigure edilmelidir. Rip ile çalışan bir Floating Static Route için Administrative Distance değeri 120' den büyük olmalıdır.

Bu durumda Rip sorunsuz çalıştığı sürece Floating Static Route Routing Table' da görünmeyecek ancak Rip devre dışı kaldığında çalışmaya başlayacaktır.

Ve Rip tekrar aktif olarak çalışmaya başlarsa devre dışı kalacaktır. WAN bağlantısının sürekli up olmasını isteyen musterile için ideal çözümdür. Alternatifi olarak Örneğin ISDN bağlantıları Floating Static Route ile backup için önerilir.




```
router rip
   network 192.168.14.0
ip route 172.16.0.0 255.255.0.0 bri0/1 130
```

Örneğimiz de iki nokta arasında 1,5 Mbitlik bir bağlantı var ve bu bağlantı Rip ile konfigure edilmiş. Aynı iki nokta arasında dial-up bir bağlantı var bu da Floating Static Route ile konfigure edilmiş ve Administrative Distance için Ripinkinden daha büyük olan "130" seçilmiş.

(Burada ki bri 0/1 portu ISDN bağlantıları için kullanılan porttur, CCNA 4 içerişinde detaylı olarak anlatılacaktır.)

Dolayısıyla Rip ile çalışan hat down olduğunda dial-up bağlantı devreye girecek ve hat tekrar aktif olduğunda devreden çıkacaktır.

IGRP (INTERİOR GATEWAY ROUTİNG PROTOCOL)

IGRP Cisco tarafından geliştirilmiş bir uzaklık-vektör algoritmasıdır. Bu yüzden network'te IGRP çalıştırmak için tüm router'ların Cisco olması gerekir. IGRP'de maksimum hop count değeri 255 dir ve RIP'te tanımlanabilecek maksimum hop count olan 15'den çok daha büyük bir değerdir. Bunun haricinde IGRP, RIP'ten farklı olarak en iyi yolu seçerken kullanılan metric değeri için varsayılan olarak, hattın gecikmesi **(delay)** ve band genişliğini **(bandwidth)** kullanır. Bunun haricinde güvenilirlik **(reliability),** yük **(load)** ve MTU**(Maximum Transmission Unit**) değerleri de metric hesabında kullanılabilir.

IGRP yol seçimi yaparken K1' den K5' e kadar 5 ayrı değere bakar. Burada kullanılan en etkin değer bant genişliğiyle ifade edilen K1 değeridir.

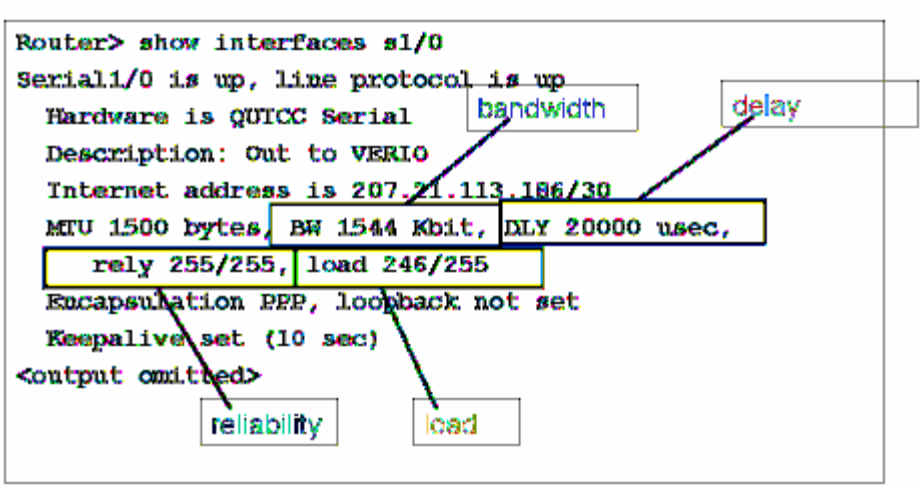
K1: Bant Genişliği

K2: Yük

K3: Gecikme

K4: Güvenilirlik

K5: MTU (Maximum Transmission Unit)



Burada büyük çoğunlukla etki eden değer bant genişliği değeridir. Routerlar bant seri interface'lerindeki genişliklerini anlayamazlar bu yüzden bizim verdiğimiz yada default olan değerleri kullanırlar. Default olarak bir Cisco Router' in seri interface' i 1,5 M.bit olarak çalışır, daha doğrusu hesaplarınıbu değer ile yapar. Bu 1,5 Mbit ile çalışıldığı anlamına gelmez.

Metric değerlerinin anlamlı olması için gerçek bant genişliği interfacelere atanmalıdır. Bunun için "**bantwith** bantgenişliği (kbit)" komutu kullanılır.

```
Router(config-if)# bandwidth kilobits
```

IGRP AD VE TİMERS

IGRP' nin Administrative Distance' i 100 ' dür ve dolayısıyla aynı routerda Rip ile birlikte kullanılacak olursa önceliğe sahip olacak, Router en iyi yol seçimini IGRP mantığından hareketle yapacaktır.

IGRP performans kontrolü için aşağıdaki sayaçları kullanır.

- Update Timer: Hangi sıklıkla yönlendirme güncelleme mesajlarının gönderileceğini belirler. Varsayılan olarak 90 sn.'dir.

- Invalid Timer: Router'ın herhangi bir yönlendirme kaydını geçersiz olarak işaretlemesi için ne kadar beklemesi gerektiğini belirtir. Varsayılan olarak update timer değerinin üç katıdır.

- Holddown Timer: Holddown periyodunu belirtir ve varsayılan olarak update timer değeri artı 10 sn.'dir.

- Flush Timer: Bir yönlendirmenin, yönlendirme tablosundan ne zaman süre sonra kaldırılacağını belirtir. Varsayılan değer ise update timer değerinin yedi katıdır.

IGRP default olarak 90 saniyede Routing Table' ini komşu Routerlarına 255.255.255.255 broadcast adresi üzerinden **update** eder.

Yine default olarak 3x90 yani 270 saniye sonra hala update gelmeyen networklerini **invalid** varsayar fakat bu network bilgisini Routing Tablesindan silmez, ayrıca bu network ile ilgili daha büyük metrikli updateleri kabul etmez.

Daha büyük metriğe sahip updateleri ancak **Holddown Timer** süresinin sonunda kabul eder ki bu süre 280 saniyedir. Artık bu noktadan sonra IGRP ile kofigure edilmiş Router kaybettigi network bilgisini silmese bile daha büyük metrik ile gelebilecek updateleri kabul edecektir.

Kaybettigi networkun bilgisini ise **Flush Timer** süresinin sonunda silecektir. Bu süre de default olarak 630 saniyedir.

"show ip protocols" komutu ile bu süreler görüntülenebilir.

```
RouterB#show ip protocols
Routing Protocol is "igrp 101"
 Sending updates every 90 seconds, next due in 51
seconds
 Invalid after 270 seconds, hold down 280, flushed
after 630
 Outgoing update filter list for all interfaces is
 Incoming update filter list for all interfaces is
 Default networks flagged in outgoing updates
 Default networks accepted from incoming updates
 IGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
 IGRP maximum hopcount 100
 IGRP maximum metric variance 1
 Redistributing: igrp 101
 Routing for Networks:
  192.168.2.0
  192.168.3.0
```

Tıpkı Rip' te olduğu gibi timers basic komutu ile default olan bu süreler değiştirilebilir.

Takrar default değerlere dönülmek istendiğinde ise **"no timers basic** " komutu

kullanılmalıdır.

Router (config-router) #**royter igr 100**

Router (config-router) **#timers basic update invalid bolddown**

Flash [sleeptime]

Router (config-router) **#no timers basic**

IGRP maksimum hop sayısı yonunden Rip'e göre üstündür, maksimum 255 hopa kadar çalışır. Fakat Cisco özel olması yüzünden dezavantajlıdır, farklı üreticilere ait routerların olduğu sistemlerde kullanılamaz.

IGRP LOAD BALANCİNG

Her Routing protocol eşit metrikli yollara Yük dağıtımı yapar ancak IGRP konuşan Routerlardan eşit olmayan yollar için load balancing yaptirilabilir. (Bu durum EIGRP tarafindan da desteklenmektedir.)

Bunun için **"variance"** komutu kullanılır.

Örnek;

Router(config)#router igrp 102

Router(config-router)#network 10.1.1.0

Router(config-router)#network 192.168.1.0

Router(config-router)#network 172.16.1.0

Router(config-router)#variance 2

Burada Router variance ile belirtilmiş sayısı alıp en küçük metrik değeri ile çarpar ve o değerin altında metriğe sahip yollar arasında load balancing yapar.

IGRP Konfigurasyonu

IGRP' de tıpkı Rip gibi classfull bir routing protokoldür.

IGRP konfigürasyonu Rip' in ki ile büyük ölçüde aynıdır. Burada tek fark aynı sistemde çalıştığımızı belirtmek için kullanacağımız **Autonomus System** numarasıdır. Kısaca **AS** denebilir. **Bütün Routerlarda aynı AS kullanılmaz ise routerlar arasında iletişim olmaz.**

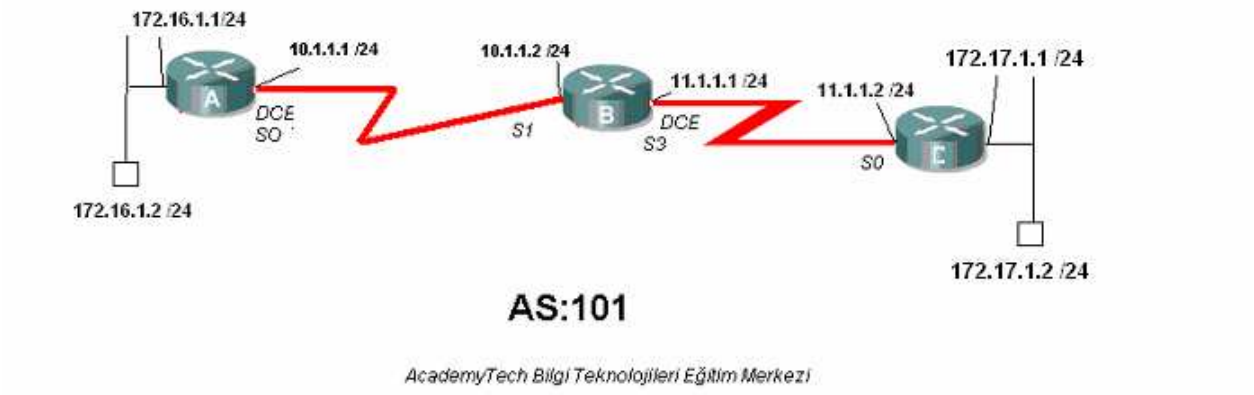
Router üzerinde IGRP'yi çalıştırmak için aşağıdaki komutu girmeniz gerekiyor.

RouterA(config)#router igrp 10

RouterA(config-router)#network 172.16.0.0

Yukarıdaki komutta router'a autonomous system (AS) numarasının 10 olduğunu ve bağlı bulunduğu ağın IP numarası bildiriliyor.

Örnek bir senaryo ile konfigürasyonu yapmak gerekirse;



Konfigürasyon yapılırken DCE ve DTE uçlar düzgün belirlenmeli ve gereken yerlere "**clock rate**" verilmelidir.

```
A(config)#router igrp 101
A(config-router)#net
A(config-router)#network 172.16.1.0
A(config-router)#network 10.1.1.0
A(config-router)#_
```

(AS)

```
C(config)#router igrp 101
C(config-router)#net
C(config-router)#network 172.17.1.0
C(config-router)#network 11.1.1.0
C(config-router)#
```

(A ve C router' larının konfigürasyonu)

Routing Table incelendiğinde administatice distinct' in 100 olduğu görünecektir. Yine dikkat edilirse metrik değerlerinin Rip' inkinden çok faklı ve yüksek değerlerde olduğu gözden kaçmaz. İşte bu metrik değerleri bahsettigimiz K1' den K5' e kadar değerlerle hesaplanmıştır.

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

Gateway of last resort is not set

I    172.17.0.0/16 [100/91056] via 10.1.1.2, 00:00:13, Serial0/0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.1.0 is directly connected, Ethernet0/0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Serial0/0
I    11.0.0.0/8 [100/90956] via 10.1.1.2, 00:00:13, Serial0/0
A#
```

(A router' ı Yönlendirme Tablosu)

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

Gateway of last resort is not set

     172.17.0.0/24 is subnetted, 1 subnets
C       172.17.1.0 is directly connected, Ethernet0/0
I    172.16.0.0/16 [100/10576] via 11.1.1.1, 00:01:13, Serial0/0
I    10.0.0.0/8 [100/10476] via 11.1.1.1, 00:01:13, Serial0/0
     11.0.0.0/24 is subnetted, 1 subnets
C       11.1.1.0 is directly connected, Serial0/0
C#
```

(C Router'ı yönlendirme tablosu)

```
B(config)#router igrp 101
B(config-router)#net
B(config-router)#network 10.1.1.0
B(config-router)#network 11.1.1.0
B(config-router)#
B(config-router)#_
```

00:02:09 bağlanıldı | OtoAlgıla | 9600 8-N-1 | Kaydır | büyk

B Router'ı Konfigürasyonu ve Yönlendirme Tablosu

```
B#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile,
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external t
       E1 - OSPF external type 1, E2 - OSPF external type 2, E -
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - ca
       U - per-user static route, o - ODR

Gateway of last resort is not set

I    172.17.0.0/16 [100/89056] via 11.1.1.2, 00:00:07, Serial3
I    172.16.0.0/16 [100/8576] via 10.1.1.1, 00:00:57, Serial1
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Serial1
     11.0.0.0/24 is subnetted, 1 subnets
C       11.1.1.0 is directly connected, Serial3
B#
```

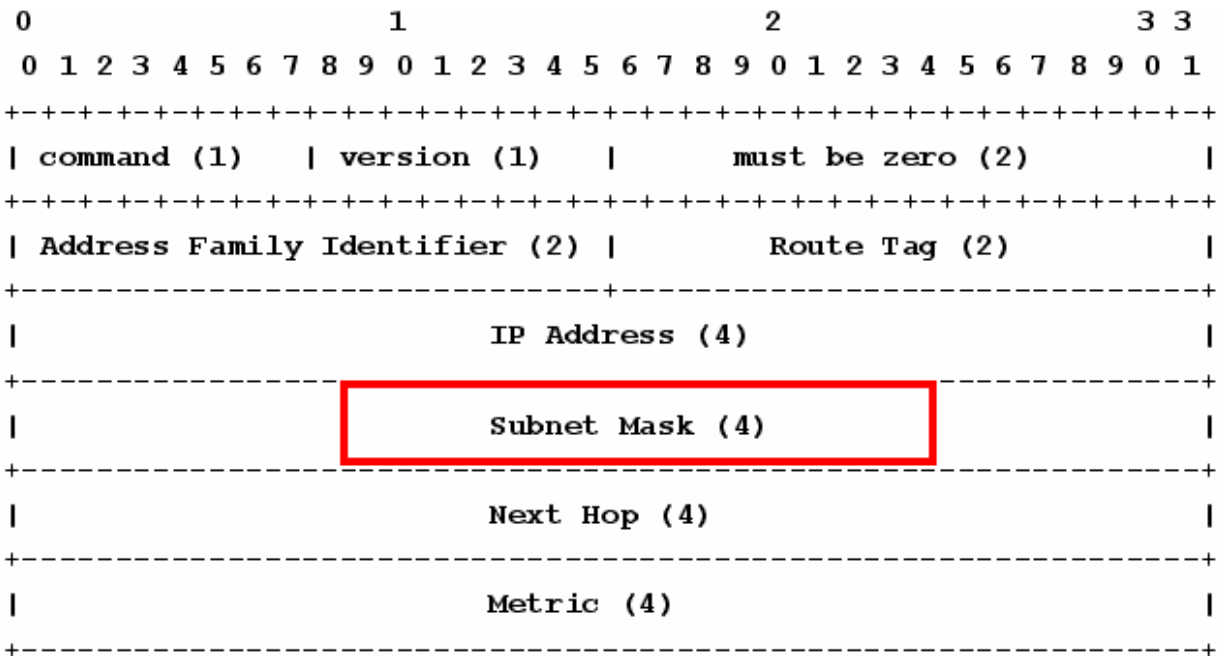
IGRP' de tıpkı Rip gibi updatelerini 255.255.255.255 broadcast adresinden yapar.

KOMUT	AÇIKLAMA
Show protocol	Her bir interface'in Network katmanı adresini ve interface'lerin aktif (up) mi yoksa pasif(down) mi olduğunu gösterir.
Show ip protocol	Router'da çalışan yönlendirme protokolleri hakkında özet bilgi verir.
Debup ip rip	Router tarafından gönderilen ve alınan yönlendirme güncellemelerinin konsol portuna da yollanmasını sağlar. Böylece yönlendirme işlemlerini izleyebilirsiniz. Eğer telnet ile router'a bağlıysanız bu güncellemeleri izleyebilmek için "**terminal monitor**" komutunu kullanmalısınız.
Debug ip igrp (events/transactions)	Eğer events parametresi ile kullanılırsa ağ üzerindeki IGRP yönlendirme bilgileri hakkında özet bilgi sunar. Transactions parametresi ile birlikte kullanılırsa komşu router'lara yapılan güncelleme istekleri ile broadcast mesajları hakkında bilgi verir.

RIPv2

Rip protokolünün classfull olması ve uygulamada sorunlar çıkarması sebebiyle geliştirilmiş ve Classless olan versiyonu çıkarılmıştır: Ripv2. Classless olmasının yanında bir önemli farkta Ripv2 nin updatelerini broadcast adresinden değil 224.0.0.9 multicast adresinden göndermesidir.

UDP 520 nolu portu kullanan RIP version 2 çalıştığında update paketleri şu şekildedir.



Burada ki subnet mask bilgisi protokolun Classless çalışmasını sağlar.

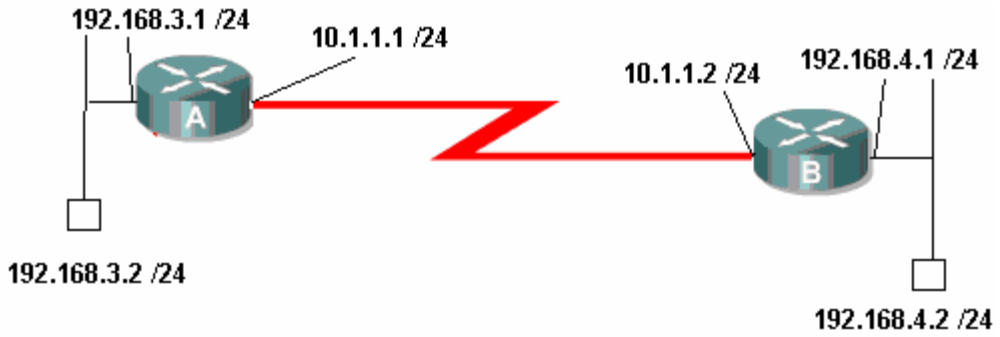
```
ISP#debug ip rip
RIP protocol debugging is on
ISP#01:23:34: RIP: received v2 update from 192.168.4.22 on Serial1
01:23:34:      172.30.100.0/24 -> 0.0.0.0 in 1 hops
01:23:34:      172.30.110.0/24 -> 0.0.0.0 in 1 hops
ISP#
01:23:38: RIP: received v2 update from 192.168.4.26 on Serial0
01:23:38:      172.30.2.0/24 -> 0.0.0.0 in 1 hops
01:23:38:      172.30.1.0/24 -> 0.0.0.0 in 1 hops
ISP#
01:24:31: RIP: sending v2 update to 224.0.0.9 via Ethernet0 (10.0.0.1)
01:24:31:      172.30.2.0/24 -> 0.0.0.0, metric 2, tag 0
01:24:31:      172.30.1.0/24 -> 0.0.0.0, metric 2, tag 0
01:24:31:      172.30.100.0/24 -> 0.0.0.0, metric 2, tag 0
01:24:31:      172.30.110.0/24 -> 0.0.0.0, metric 2, tag 0
01:24:31:      192.168.4.24/30 -> 0.0.0.0, metric 1, tag 0
01:24:31:      192.168.4.20/30 -> 0.0.0.0, metric 1, tag 0
```

Subnet Mask Bilgisi

multicast

Konfigürasyonda ise tek küçük fark “version 2” komutunun verilecek olmasıdır. Bu komut router konfigürasyon alt moduna geçildiğinde verilmelidir.

Rip protokolü sırasında üzerinde çalıştığımız senaryoyu burada da uyguladığımız da aradaki farkları daha iyi anlayacağız.



Bu topoloji de ip adreslerimizi ilgili interface' lere atadıktan sonra konfigürasyon Ripv2 için şu şekilde olacaktır.

```
RouterA#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RouterA(config)#router rip
RouterA(config-router)#ver
RouterA(config-router)#version 2
RouterA(config-router)#net
RouterA(config-router)#network 10.1.1.0
RouterA(config-router)#net
RouterA(config-router)#network 192.168.3.0
RouterA(config-router)#exi
RouterA(config)#exit
RouterA#
```

(Router A için Konfigürasyon)

```
RouterB#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RouterB(config)#router rip
RouterB(config-router)#ver
RouterB(config-router)#version 2
RouterB(config-router)#net
RouterB(config-router)#network 10.1.1.0
RouterB(config-router)#net
RouterB(config-router)#network 192.168.4.0
RouterB(config-router)#exit
RouterB(config)#exit
```

(Router B için Konfigürasyon)

Her iki router için konfigürasyonlar tamamlandığında networkler arasında iletişim sağlanmış olacaktır. Bu iletişim tabiki Router' ların Routing Table' larında bulunan bilgilere dayanarak olacaktır.

Routing Table' lar artık çok iyi bildiğiniz gibi "**show ip route**" komutu ile görüntülenebiliyor.

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

Gateway of last resort is not set

R    192.168.4.0/24 [120/1] via 10.1.1.2, 00:00:18, Serial0/1
     10.0.0.0/30 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Serial0/1
C    192.168.3.0/24 is directly connected, Ethernet0/0
RouterA#_
```

00:20:15 bağlanıldı | OtoAlgıla | 9600 8-N-1 | Kaydır | büyh | SAYI | Yakala | Yazdırma yankısı

(RouterA için Routing Table)

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

Gateway of last resort is not set

C    192.168.4.0/24 is directly connected, Ethernet0/0
     10.0.0.0/30 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Serial0/1
R    192.168.3.0/24 [120/1] via 10.1.1.1, 00:00:12, Serial0/1
RouterB#
```

00:24:43 bağlanıldı | OtoAlgıla | 9600 8-N-1 | Kaydır | büyh | SAYI | Yakala | Yazdırma yankısı

(RouterB için Routing Table)

Routign Table' lar dikkatle incelendiğinde uzak networklere giderken kullanılacak yollar metrik ifadeleriyle birlikte görüntülenebiliyor. Ripv2' de tıpkı Ripv1 gibi metrik hesabında hop sayısını kullandığı için buradaki metrikler aynı zaman da hop sayısınıa eşittir.

```
RouterA#debug ip rip
RIP protocol debugging is on
RouterA#
00:28:32: RIP: sending v2 update to 224.0.0.9 via Ethernet0/0 (192.168.3.1)
00:28:32: RIP: build update entries
00:28:32:       10.0.0.0/8 via 0.0.0.0, metric 1, tag 0
00:28:32:       192.168.4.0/24 via 0.0.0.0, metric 2, tag 0
00:28:32: RIP: sending v2 update to 224.0.0.9 via Serial0/1 (10.1.1.1)
00:28:32: RIP: build update entries
00:28:32:       192.168.3.0/24 via 0.0.0.0, metric 1, tag 0
00:28:37: RIP: ignored v2 update from bad source 192.168.4.1 on Ethernet0/0
00:28:37: RIP: received v2 update from 10.1.1.2 on Serial0/1
00:28:37:      192.168.4.0/24 via 0.0.0.0 in 1 hops
00:28:59: RIP: sending v2 update to 224.0.0.9 via Ethernet0/0 (192.168.3.1)
00:28:59: RIP: build update entries
00:28:59:       10.0.0.0/8 via 0.0.0.0, metric 1, tag 0
00:28:59:       192.168.4.0/24 via 0.0.0.0, metric 2, tag 0
00:28:59: RIP: sending v2 update to 224.0.0.9 via Serial0/1 (10.1.1.1)
00:28:59: RIP: build update entries
00:28:59:       192.168.3.0/24 via 0.0.0.0, metric 1, tag 0_
```

0:23:57 bağlanıldı | OtoAlgıla | 9600 8-N-1 | Kaydır | büyh | SAYI | Yakala | Yazdırma yankısı

(RouterA için debug)

Updateler multicast 224.0.0.9 adresinden gönderiliyor ve alınıyor. Oysa Rip version 1' de updateler broadcast 255.255.255.255 adresinden yapılıyordu.

İsterseniz şimdi tekrar Rip version 1' e geçip, bir de oradaki updateleri inceleyelim. Geçiş her iki Router'da da "version 2" ifadesini kaldırarak yapılabilir. Her zaman olduğu gibi kaldırmak istediğimiz bir komut olduğunda basına **"no"** yazmamız yeterli olacaktır. Örneğin A Router' ı Rip version 1' e şu şekilde geçer:

```
RouterA(config)#router rip
RouterA(config-router)#no ver
RouterA(config-router)#no version 2
RouterA(config-router)#_
```

00:25:05 bağlanıldı | OtoAlgıla | 9600 8-N-1 | Kaydır | büyh | SAYI | Yak

```
RouterA#debug ip rip
RIP protocol debugging is on
RouterA#
00:32:39: RIP: sending v1 update to 255.255.255.255 via Ethernet0/0 (192.168.3.1
)
00:32:39: RIP: build update entries
00:32:39:       network 10.0.0.0 metric 1
00:32:39:       network 192.168.4.0 metric 2
00:32:39: RIP: sending v1 update to 255.255.255.255 via Serial0/1 (10.1.1.1)
00:32:39: RIP: build update entries
00:32:39:       network 192.168.3.0 metric 1
```

00:27:40 bağlanıldı | OtoAlgıla | 9600 8-N-1 | Kaydır | büyh | SAYI | Yakala | Yazdırma yankısı

(RouterA için dedbug)

Her iki versiyonun update' leri arasında ki fark artık daha iyi anlaşılmıştır.

Networkler arasıdaki iletişimi komut satırında kullanabileceğimiz **"tracert**" komutu ile inceleyebiliriz.

```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Cisco>ping 192.168.4.2

32 bayt veri ile 192.168.4.2 'ping' ediliyor:

192.168.4.2 cevabı: bayt=32 süre=21ms TTL=126
192.168.4.2 cevabı: bayt=32 süre=18ms TTL=126
192.168.4.2 cevabı: bayt=32 süre=18ms TTL=126
192.168.4.2 cevabı: bayt=32 süre=18ms TTL=126

192.168.4.2 için Ping istatistiği:
    Paket: Giden = 4, Gelen = 4, Kaybolan = 0 (0% kayıp),
Mili saniye türünden yaklaşık tur süreleri:
    En Az = 18ms, En Çok = 21ms, Ortalama = 18ms

C:\Documents and Settings\Cisco>tracert 192.168.4.2

En çok 30 atlamanın üstünde 192.168.4.2'e giden yolu izlemek

  1     1 ms     1 ms     1 ms  192.168.3.1
  2    22 ms    22 ms    22 ms  10.1.1.2
  3    26 ms    26 ms    26 ms  192.168.4.2

İzleme tamamlandı.

C:\Documents and Settings\Cisco>
```

Ben bu komutu kullanırken 192.168.3.2 ip adresine sahip bilgisayarı kullandım. 1.adımda ping paketim varsayılan ağ geçidi olarak tanımladığım RouterA' nin Ethernet interface' ine, ikinci adımda bir sonraki Router' ın Serial interface' ine ve üçüncü adımda da hedefe ulaştı.

Her iki Router için Running-config dosyasının incelenmesi fayda sağlayacaktır.

```
RouterA#show running-config
Building configuration...
Current configuration : 567 bytes
!
version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname RouterA
!
!
memory-size iomem 10
ip subnet-zero
!
!
interface Ethernet0/0
ip address 192.168.3.1 255.255.255.0
!
interface Serial0/0
no ip address
shutdown
no fair-queue
!
interface Serial0/1
ip address 10.1.1.1 255.255.255.252
!
router rip
version 2
network 10.0.0.0
network 192.168.3.0
!
ip classless
ip http server
!
!
line con 0
```

```
line aux 0
line vty 0 4
!
no scheduler allocate
end

RouterB#show running-config
Building configuration...
Current configuration : 578 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname RouterB
!
memory-size iomem 10
ip subnet-zero
!
interface Ethernet0/0
ip address 192.168.4.1 255.255.255.0
half-duplex
!
interface Serial0/0
no ip address
shutdown
!
interface Serial0/1
ip address 10.1.1.2 255.255.255.252
clockrate 64000
!
!
version 2
network 10.0.0.0
network 192.168.4.0
!
```

ip classless

ip http server

!

!

gatekeeper

shutdown

!

!

line con 0

line aux 0

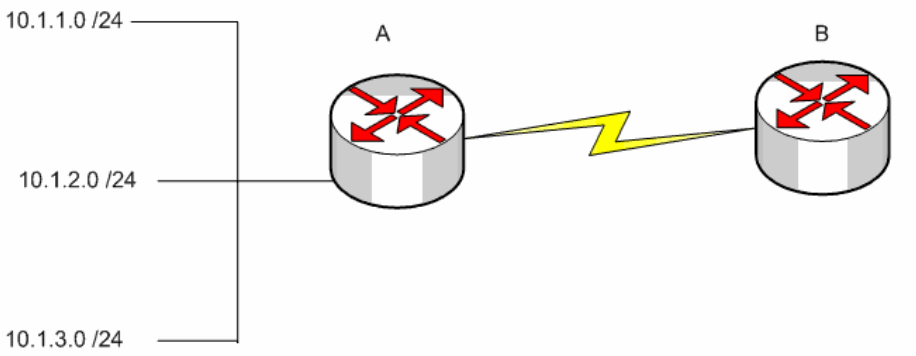
line vty 0 4

!

end

RipV2 Auto summary

Ripv2' nin Auto summarization özelliği vardır ve default olarak açık durumdadır.



Örneğin şekildeki yapı içerişinde Ripv2 ile konfigure edilmiş A routeri yine Ripv2 ile konfigure edilmiş B routerina 10.0.0.0 netrworkunu update edecektir.

Bu çalışma mantığı içerişinde default olarak açık olan auto summarization özelliği kapatılmadığı takdir de Ripv2 ninde sanki Classfull muş gibi çalıştığı söylenir.

Auto şummarizxation özelliği "no auto-summary" komutu ile kaldırılabilir.

A(config)#router rip

A(config-router)#network 10.1.1.0

A(config-router)#network 10.1.2.0

A(config-router)#network 10.1.3.0

A(config-router)#version 2

A(config-router)#no auto summary

Konfigurayonun bu hali ile Artık A routeri summary update yerine bütün networkleri update edecektir ve B Routeri Routing Table' inde bütün networkler yer alacaktır.

Extra

Split Horizon kuralının enable olmadığı durumlarda ripv2 ile konfigure edilmiş ve update edilecek networklerin üzerine yazacak ve interface' e uygulanacak **"ip summary-address**" komutu kullanılabilir.

int s1

ip address 10.1.1.1 255.255.255.0

ip summary-address rip 10.2.0.0 255.255.0.0

no ip split-horizon

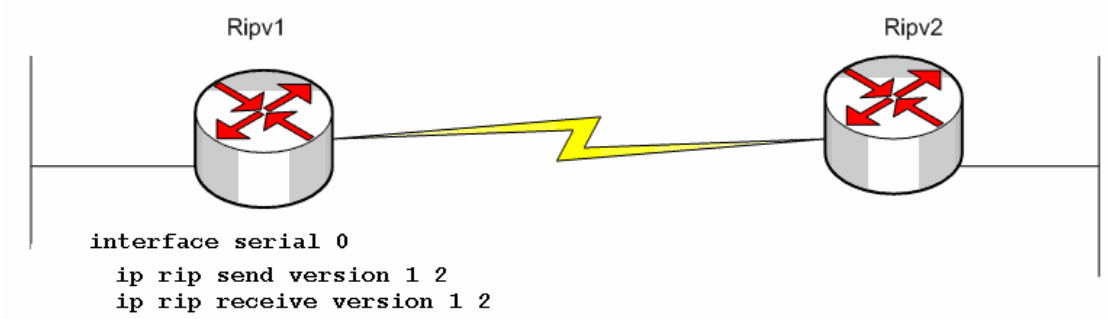
router rip

network 10.0.0.0

Örneğin bu uygulamada 10.2.0.0 update' i rip tarafindan özetlenen 10.0.0.0 update' inin üzerine yazacaktır.

RİPV1 VE RİPV2 HABERLESMESİ

Ripv1 ve Ripv2 konfgirasyonlar sistem de bulnan routerlar arasında sadece Ripv1, Ripv2 paketlerini alıp göndermek yada her ikisinide alıp göndermek üzere konfigure edilebilir.

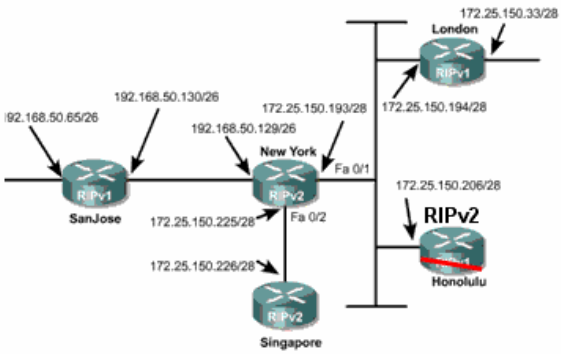


Burada istenirse farklı uygulamalar yapılabilir.

Örneğin;

ip rip receive version 1

komut satırı ile Söz konusu Routerin sadece Version 1 updatelerini almasi sağlanabilir.




```
NewYork

interface fastethernet0/0
  ip address 192.168.50.129 255.255.255.192
  ip rip send version 1
  ip rip receive version 1

interface fastethernet0/1
  ip address 172.25.150.193 255.255.255.240
  ip rip send version 1 2
  ip rip receive version 1 2
interface fastethernet0/2
  ip address 172.25.150.225 225.255.255.240

router rip
  version 2
  network 172.25.0.0
  network 192.168.50.0
```

CNAP Slaytlarından alınan bu şekilde durum daha iyi anlaşılacaktır. Burada Newyork Router i Ripv2 ile konfigure edilmiş. Ve interfacelerine sırasıyla şu şekilde konfigure edilmiş diğer routerlar bağlı;

Fa0/0: Ripv1

Fa0/2:Ripv2

Fa0/1:Ripv1 ve Ripv2

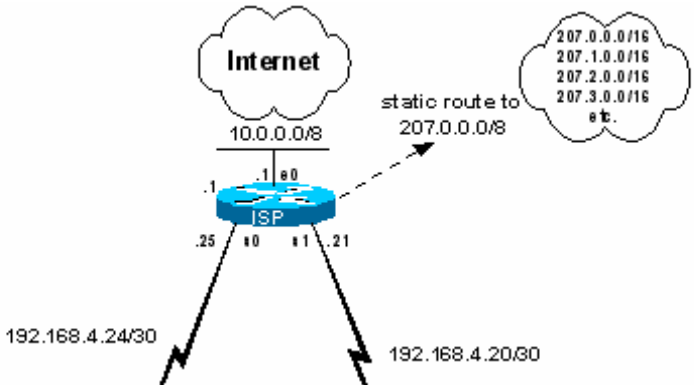
Bu durumda Fa0/0 interface ine hem London Routerindan aldığı v1 updatelerini göndermeli hemde ondan v1 updatelerini almalıdır.

Fa0/1 interface inde ise konfigurasyona bakıldığında hem v1 hem de v2 updatelerini göndermek üzere hem de almak üzere konfigure edildiği anlaşılmıştır. Çünkü bu interface' e hem Ripv1 ile hem de Ripv2 ile konfigure edilmiş Routerlar bağlanmıştır.

(Multiaccess)

Fa0/2 için zaten özel bir konfigurasyona gerek yoktur.

Ripv2 ve Default Routing



ISP Routeri üzerinde Default Route tanımlanması aşağıdaki gibi olacaktır.

```
ISP
router rip
   redistribute static
   network 10.0.0.0
   network 192.168.4.0
   version 2
   no auto-summary
   default-information originate

ip route 207.0.0.0 255.0.0.0 null0
ip route 0.0.0.0 0.0.0.0 10.0.0.2
   etherenet0
```

Ripv2 Authentication

Ripv2 konuşan Routerların updateleri sırasında authentication sağlanabilir. Bunun için Global Konfigurasyon modunda **"key**" komutu kullanılmalıdır.

Router(config)#key chain Hayrullah

Router(config-keychain)#key 1

Router(config-keychain-key)#key-string Kolukisaoglu

Authentication sağlanacak Routerlar için password aynı olmalıdır ancak key adı değişitirilebilir. Key oluşturulduktan sonra interface'e uygulanmalıdır.

Router(config)#interface fastethernet 0/0

Router(config-if)#ip rip authentication key-chain Hayrullah

Router(config-if)#ip rip authentication mode md5

Burada ki "ip rip authentication mode md5" komutunun kullanımı opsiyöneldir. Authentication bilgilerinin encrypted halde gönderilmesini sağlayan bu komut kullanılmadığında da interface default olarak text halinde authentication bilgilerini gönderecektir.

ACCESS LİSTS (ERİŞİM LİSTELERİ)

Access list'ler sistem yöneticilerine, ağdaki trafik üzerinde geniş bir kontrol imkanı sunar. Ayrıca access list'ler router üzerinden geçen paketlere izin vermek veya reddetmek içinde kullanılır. Bunun haricinde telnet erişimleri de access list'ler kullanılarak düzenlenebilir. Oluşturulan access list'ler router'daki interface'lerin herhangi birisine giren veya çıkan trafiği kontrol edecek şekilde uygulanabilir. Eğer herhangi bir interface'e bir access list atanmışsa router bu interface'den gelen her paketi alıp inceleyecek ve access list'te belirtilen işlevi yerine getirecektir. Yani ya o paketi uygun yöne iletecek yada paketi yönlendirmeden yok edecektir.

1. Access List' lerde kriterler satır satır belirtilmiştir. Gelen isteklerin kriterlere uyup uymadıkları sırayla belirlenir.

2. İlk eşleşen kriterin bulunduğu satıra gelindiğinde o satırda ki aksiyon (deny yada permit) gerçekleştirilir.

3. Paket bütün satırları geçmiş ve herhangi bir kriterle eşleşme olmamışsa “bütün paketleri yoket” (implicit deny all) kuralı uygulanır.

Access List’ ler 3 Başlık altında incelenirler:

1. Standart ACL

2. Extended ACL

3. Named Acl

3 başlık dememize rağmen aslında iki başlık gibi düşünülmelidir. Çünkü Named Accces Listler hem standart hem de Extended olarak kullanılabilirler.

Access Listler arasındaki bu ayırım Acces List Numaraları ile yapılır. Access Listler şu numaraları alabilirler;

Access List Numarası	Açıklama
1-99 arası	IP standart access list
100-199 arası	IP extended access list
1000-1099 arası	IPX SAP access list
1100-1199 arası	Extended 48-bit MAC address access list
1200-1299 arası	IPX summary address access list
200-299 arası	Protocol type-code access list
300-399 arası	DECnet access list
400-499 arası	XNS standart access list
500-599 arası	XNS extended access list
600-699 arası	Appletalk access list
700-799 arası	48-bit MAC address access list
800-899 arası	IPX standart access list
900-999 arası	IPX extended acess list

Access List’ ler oluşturulurken dikkat edilmesi gerekenler şunlardır;

1. Oluşturulduktan sonra mutlaka bir interface ile ilişkilendirilmelidir aksi takdir de aktif olmayacaktır.

2. Kriterler satır satır uygulanacağı için listeler oluşturulurken en belirgin kriterden en genel kritere doğru Yükarıdan başlayarak organize edilmelidir.

3. Listeden satır silmek ve satır eklemek sadece Named ACL’ lerde mümkündür. Diğer listelerde silme isleminde satır değil listenin tamamı silinir. Bu durumda araya satır eklenmek isteniyorsa liste bir yazı editörüne aktarılıp değişiklik orada yapılmalıdır.

4. Standart Access Listler mümkün olduğu kadar hedefe, Extended Access Listler mümkün olduğu kadar kaynağa yakın olmalıdır.

5. Access Listlerin en sonudan görünmeyen bir satır oluştugunu ve bu satırında diğer satırlardaki herhangi bir kritere uymayan istekleri yok ettigini söylemistik.

6. Dolayısıyla mutlaka ve mutlaka bir Access List grubunda “permit” aksiyonu olmalıdır.

7. Access Listler sadece Router üzerinden giden veya gelen trafiği düzünlemek için kullanılabilirler. Router’ ın sebep olduğu trafik için kullanılamazlar.

8. Access Listler’ den satır çıkaramasanız ve satır eklediğinizde de o satır en son satır olarak yerini alır. Dolayısıyla kritelerinizi yeniden düzenlemek bu şekilde imkansızdır. (Named Access List’ ler hariç) Bu durumda yapılması gereken Access List’ i bir text editörüne kopyalayıp gerekli değişiklikleri yaptıktan sonra ger kopyalamaktır.

Access Listler oluşturulurken subnet Mask yerine Wild Card Mask denilen ve subnet Maskın 255' e tamamlanmasıyla elde edlien bir maske kullanılır. Örneğin 255.255.128.0 subnet maskının wild-card maskı 0.0.127.255 olacaktır.

Tek bir host belirtmek için kullanılacak;

Ip adresi: 192.168.1.2

Wild-Card Mask: 0.0.0.0

STANDART ACCESS LİSTLER

Bu tür access list'te IP paketlerinin sadece kaynak (source) adreslerine bakılarak filtreleme yapılır. Izin verme yada yasaklama bütün protokol kümesi için geçerlidir.

Router(config)#acces-list {Access list numarası} {permit / deny} {kaynak} {mask}

Şeklinde kullanılır. Burada ki "permit" izin vermek için, "deny" yasaklamak için kullanılır.

Daha sonra uygulanacak olan interface' gidilerek

"ip Access-group {numarası} in/out" komutuyla interface ile ilisiklendirilir. Burada ki in ve out komutlara istege göre içeriden dışarıya (in) ve dışarıdan içeriye (out) olan trafiği kısıtlamak için kullanılır.

Örneğin networkümüz de bulunan 192.168.1.100 ip adresine sahip bilgisayarın dışarıya çıkışını önlemek istersek komut satırında;

Router(config)#access-list 1 deny 192.168.1.100 0.0.0.0

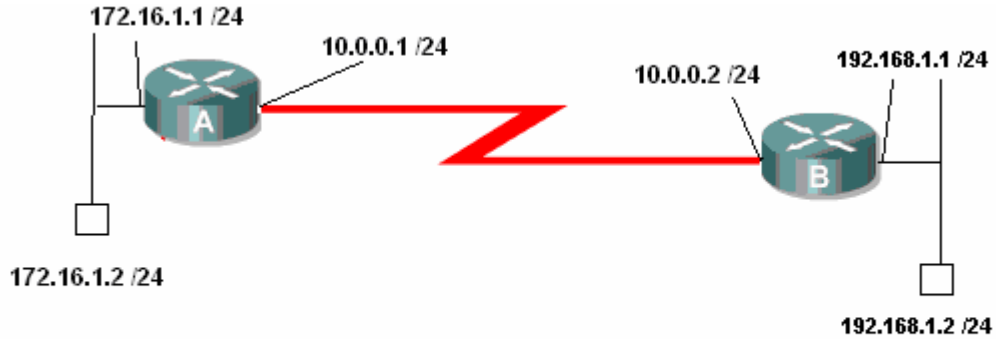
Router(config)#access-list 1 permit any

Router(config)#interface Ethernet 0/0

Router(config-if)#ip Access-group 1 in

Yazmalıyız. Burada 1. satırda ilgili hosta "deny" uygulandı, 2.satırda diğer hostların "implicit deny all" kuralı ile yok edilmemeleri için kalan hostlara "permit uygulandı, 3 ve 4.satırlarda ise oluşturulan Access list Ethernet interface' ile ilişkilendirildi. "

Access listlerde "{ip adresi wild-card mask}" yerine "host {ip adresi}" kullanılabilir.Fakat networklere bir aksiyon uygulanacaksa Wild-Card Mask kullanılmalıdır.



Örnek senaryomuz da A Router' ının Ethernet interface' ine bağlı 172.16.1.0 networkünde yer alan 172.16.1.2 bilgisayarının Ethernet interface2 inden dışarı çıkmasını engelleyelim fakat diğer bilgisayarlar bundan hiçbir şekilde etkilenmesin.

Bu durumda A Router' ın şu konfigürasyon yapılmalıdır;

```
A(config)#access-list 1 deny 172.16.1.2 0.0.0.0
A(config)#access-list 1 permit any
A(config)#int
A(config)#interface ethernet 0/0
A(config-if)#ip access-group 1 in
A(config-if)#
```

1:05:10 bağlanıldı | OtoAlgıla | 9600 8-N-1 | Kaydır | büyh | SAYI | Yakala | Yazdırma yan

Bu konfigürasyon yapıldığı andan itibaren 192.168.1.2 bilgisayarı sadece kendi LAN' ı ile haberleşebilecek, Router üzerinden kesinlikle dışarıya çıkamayacaktır. İkinci satırda yer alan "Access-list 1 permit any" satırı ile diğer bilgisayarların bu kısıtlamadan etkilenmesi engellenmiş oldu. Konfigürasyon sırasında 2. satır yazılmamış olsaydı gelen paketler / istekler tamamen yok edilecekti.

EXTENDED ACCESS LİSTLER

Bu tür access listler de kaynak ile birlikte kullanılan protokol, hedef ip adresi ve hedef port numarası da kısıtlanabilir.

Örneğin 192.168.1.100 bilgisayarının 212.1.1.8 bilgisayarına 80. porttan erişememsini, aynı bilgisayara 25. porttan erişebilmesini, diğer bilgisayarlar için herhangi bir kısıtlama olmamasını istiyoruz. (Söz konusu portlar TCP çalışır) Bu durumda komut satırına;

Router(config)#access-list 101 deny tcp host 192.168.1.100 host 212.1.1.8 eq 80

Router(config)#access-list 101 permit tcp host 192.168.1.100 host 212.1.1.8 eq 25

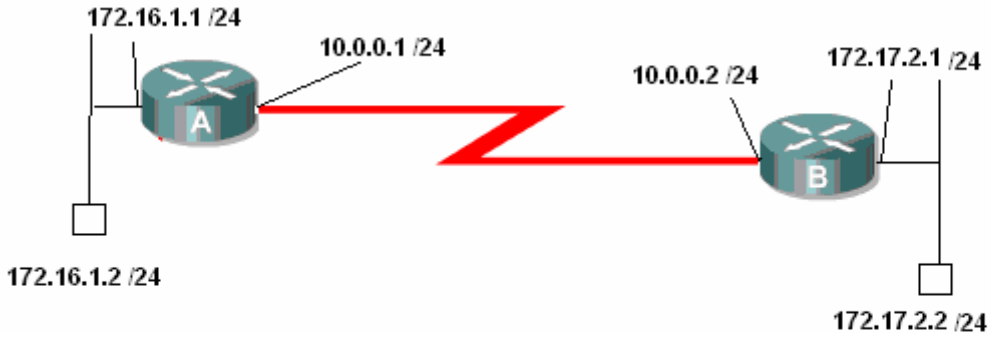
Router(config)#access-list 101 permit ip any any

Yazmak ve gerekli interface'e uygulamak yeterli olacaktır. Burada 1 ve ikinci satırlarda 192.168.1.100 ip adresine sahip bilgisayarın 212.1.1.8 ip adresine sahip uzak bilgisayara, 80. porttan erişmemesini fakat 25. porttan erişmesini sağlamış oluyoruz. 3. satır ile de diğer bilgisayarların "implicit deny all" kuralı ile yok edilmelerini önlemiş olduk.

Yine burada host 192.168.1.100 yerine Wild-Card Mask kullanarak "192.168.1.100 0.0.0.0" yazabilirdik.

Bir senarayo üzerinde çalışmak gerekirse;

Elimizde şekildeki gibi birbirlerine bağlanmış 2 farklı network var. A Router' ımızın Ethernet interface' ine bağlı networkte bulunan 172.16.1.2 bilgisayarı üzerinde bazı kısıtlamalar yapmak istiyoruz;



1. 172.16.1.2 bilgisayarı 172.17.1.2 bilgisayarına 3389. porttan erişemesin.

2. 172.16.1.2 bilgisayarı 172.17.1.2 bilgisayarına 80. porttan erişebilsin.

3. 172.16.1.2 bilgisayarı 172.17.1.2 bilgisayarına 80. porttan erişebilsin.

4. 172.16.1.0 networkünde bulunan diğer bilgisayarlar uzak networkteki diğer bilgisayarlara istedikleri portttan erişebilsinler.

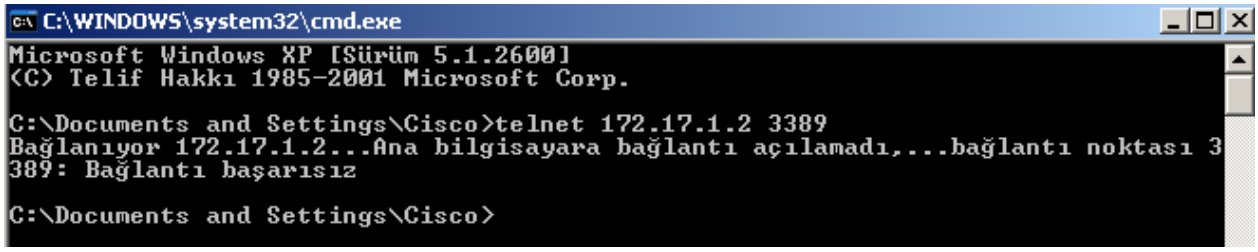
Böyle bir durumda A Router' ı üzerinde yapılacak konfigürasyon şu şekilde yapılmalıdır:

```
A(config)#
A(config)#access-list 101 deny tcp host 172.16.1.2 host 172.17.1.2 eq 3389
A(config)#access-list 101 permit tcp host 172.16.1.2 host 172.17.1.2 eq 80
A(config)#access-list 101 permit tcp host 172.16.1.2 host 172.17.1.2 eq 25
A(config)#access-list 101 permit ip  any any
A(config)#interface et
A(config)#interface ethernet 0/0
A(config-if)#ip access-group 101 in
A(config-if)#
```

1:10:06 bağlanıldı | OtoAlgıla | 9600 8-N-1 | Kaydır | büyh | SAYI | Yakala | Yazdırma yankısı

Senaryo için belirlediğimiz istekleri satır satır konfigüre ettik. Access List' in 4. satırındaki komut ile kalan bilgisayarların çıkmasına izin verilirken 1,2 ve 3. satırlarda 172.16.1.2 bilgisayarının uzak networkte ki 172.17.1.2 bilgisayarına doğru olan trafiğinde çeşitli kısıtlamalar ve izinler uygulandı. Devam eden satırlar da ise oluşturduğumuz Acces List ilgili interface' imizle eşleştirildi.

Yaptığımız düzenlemelerin düzgün çalışıp çalışmadığını test etmek isteyebiliriz. Bu durumda bize Telnet yardımcı olacaktır. Telnet ile uzak bilgisayara yasaklanan bir port üzerinden erişmek istediğiniz de bağlantının başarısız olduğuna dair bir satır karşımıza gelecek, izin verilen bir porttan erişmek istediğimizde tamamen boş bir sayfa anında açılacaktır. Eğer anlatıldığı gibi durumlar ile karşılaşılmamışsa Access List' lerin oluşturulması yada uygulanmasıyla ilgili bir problem var demektir.



(172.17.1.2 uzak bilgisayarına 3389. porttan bağlanılamıyor.)

(172.17.1.2 bilgisayarına 80. porttan Telnet ile bağlanılması)

NAMED ACCESS LİSTLER

Diğer Access Listlerden sadece konfigürasyon sırasında farklılık gösterir. Named Acces listler acces-list numarası vermek yerine akılda kalması da kolay olacak, isimler kullanılır. Named Access List' lerde satırlar tek tek silinebilir veya yeni satır eklenebilir. Çünkü listenin Standart ve Extended olmasına göre uygun modlar oluşturulur ve konfigürasyon bu modlar altında yapılır.

Extended Access List' te üzerinde çalıştığımız aynı senaryoyu Named Access List ile konfigüre etmek istersek komut satırına;

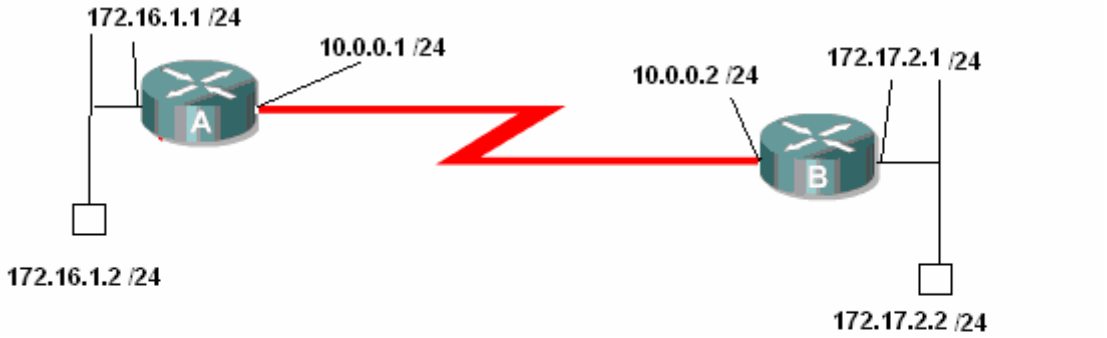
Router(config)# ip Access-list extended AcademyTech

Router(config-ext-nacl)# deny tcp host 192.168.1.100 host 212.1.1.8 eq 80

Router(config-ext-nacl)# permit tcp host 192.168.1.100 host 212.1.1.8 eq 25

Router(config-ext-nacl)# permit tcp any any

Yazmamız yeterli olacaktır. Burada 1.satırda belirtilen AcademyTech bizim belirleyeceğimiz bir isimdir ve Access list' lerin standart numaraları yerine kullanılır.Bu konfigürasyonda hatalı bir satır yazıldığında basına "no" yazılarak satır iptal edilebilir.



Böyle bir senaryo da 172.16.1.2 bilgisayarının uzak networkteki 172.17.1.2 bilgisayarının 80 ve 25. portlardan erişmemesini, 3389. porttan erişebilmesini, diğer bilgisayarlar için herhangi bir kısıtlama olmamasını Named Access List ile yapmak istediğimizde konfigürasyon şu şekilde tanımlanmalı;

```
A(config)#ip access-list extended AcademyTech
A(config-ext-nacl)#deny tcp host 172.16.1.2 host 172.17.1.2 eq 80
A(config-ext-nacl)#deny tcp host 172.16.1.2 host 172.17.1.2 eq 25
A(config-ext-nacl)#permit tcp host 172.16.1.2 host 172.17.1.2 eq 3389
A(config-ext-nacl)#permit ip  any any
A(config-ext-nacl)#exit
A(config)#interface ethernet 0/0
A(config-if)#ip access-group AcademyTech in
A(config-if)#
A(config-if)#
```

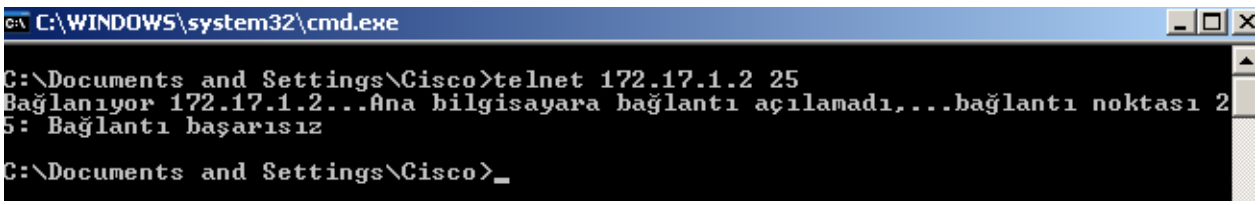
Named Access List

Burada Access List' in1. satırında extenden Acces List kullanılacağı ve bu Access List' in isminin AcademyTech olacağı belirtildi, 2. ve 3. satırları ile uzak networkteki 172.17.1.2 bilgisayarına 80 ve 25. portlardan erişilmesi, 172.16.1.2 bilgisayarı için yasaklanmış oldu.

4 ve 5. satırlarda ise gerekli izinler verildi. 4. satırdaki komut örnek olması için komut satırına yerleştirildi. Normal sartlarda bu satır kullanılmayabilir, çünkü 5. satırda ki ifade ile zaten 172.16.1.2 bilgisayarı da diğer izinleri elde ediyor.

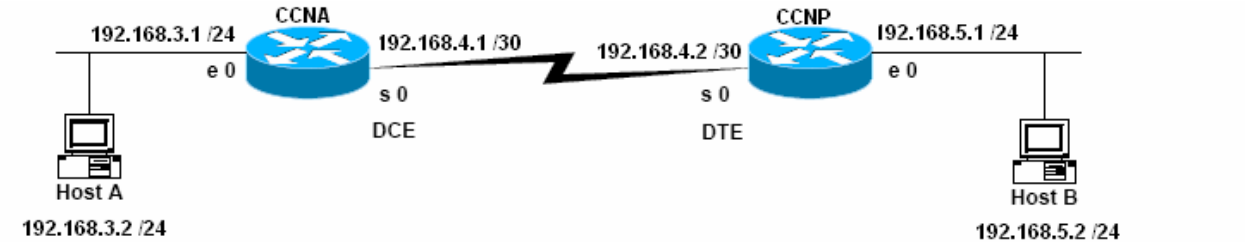
Telnet ile Acces List' leri test edersek;

(172.16.1.2'den 172.17.1.2'ye Telnet ile 3389. porttan bağlanma)



(25. porttan bağlanma denemsi başarısız.)

ACL Uygulamaları -1



Host B' den cikan paketlerin 192.168.3.0 networkune erişmesini engellemek.

CCNA Routerinda;

CCNA(config)#access-list 1 deny 192.168.5.2 0.0.0.0

CCNA(config)#access-list 1 permit any

CCNA(config)#inter serial 0

CCNA(config-if)#ip access-group 1 in

Veya;

CCNP Routerinda;

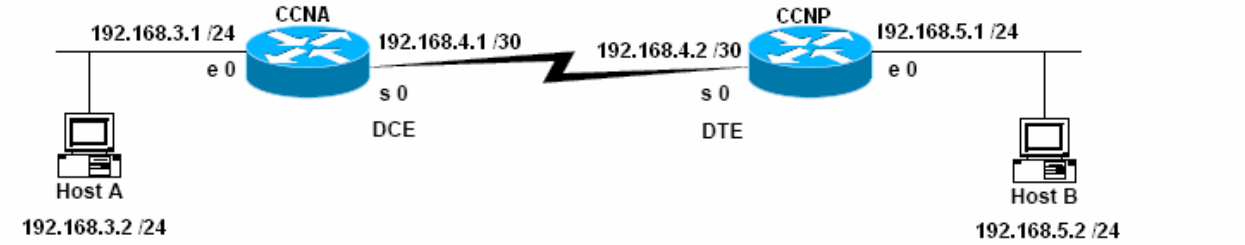
CCNP(config)#access-list 1 deny 192.168.5.2 0.0.0.0

CCNP(config)#access-list 1 permit any

CCNP(config)#inter ethernet 0

CCNP(config-if)#ip access-group 1 in

ACL Uygulamaları -2



192.168.5.0 networkunun tamaminin 192.168.3.0 networkune erişmesini engellemek.

CCNA Routerinda;

CCNA(config)#access-list 1 deny 192.168.5.2 0.0.0.255

CCNA(config)#access-list 1 permit any

CCNA(config)#inter serial 0

CCNA(config-if)#ip access-group 1 in

Veya;

CCNP Routerinda;

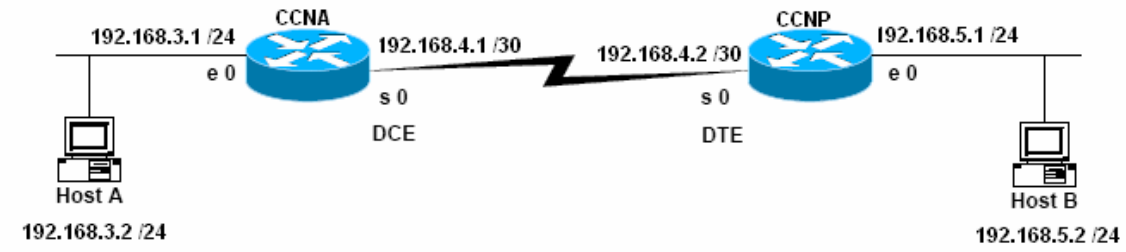
CCNP(config)#access-list 1 deny 192.168.5.2 0.0.0.255

CCNP(config)#access-list 1 permit any

CCNP(config)#inter ethernet 0

CCNP(config-if)#ip access-group 1 in

ACL Uygulamaları -3



HostA da bulunan FTP Server ve Web Server'a 192.168.5.2 bilgisayarinin erişmesini engellemek. (Kalan trafik akisi normal devam etmeli)

CCNA(config)**#access-list 101 deny tcp host 192.168.5.2 host 192.168.3.2 eq 80**

CCNA(config)#**access-list 101 deny tcp host 192.168.5.2 host 192.168.3.2 eq 21**

CCNA(config)#**access-list 101 permit ip any any**

CCNA(config)#

CCNA(config)#**inter serial 0**

CCNA(config-if)#**ip access-group 101 in**

CCNA(config-if)#

Veya;

CCNP(config)#**access-list 101 deny tcp host 192.168.5.2 host 192.168.3.2 eq 80**

CCNP(config)#**access-list 101 deny tcp host 192.168.5.2 host 192.168.3.2 eq 21**

CCNP(config)#**access-list 101 permit ip any any**

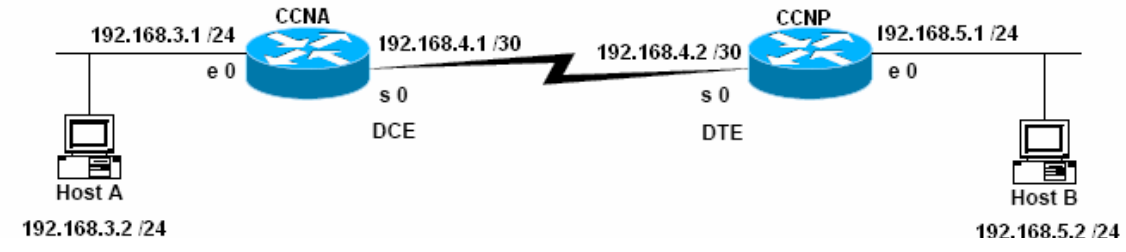
CCNP(config)#

CCNP(config)#**inter ethernet 0**

CCNP(config-if)#**ip access-group 101 in**

CCNP(config-if)#

ACL Uygulamaları -4



192.168.5.0 networkunun 192.168.3.0 networkune ping atmasini yasaklamak, 192.168.3.0 networkunde 192.168.3.1 dışındaki telnet isteklerini yasaklamak. (Kalan trafik akisi devam etmeli)

CCNA(config)#**ip access-list extended Kolukisaoglu**

CCNA(config-ext-nacl)#**deny icmp 192.168.5.0 0.0.0.255 any echo**

CCNA(config-ext-nacl)#**permit tcp 192.168.5.0 0.0.0.255 host 192.168.3.1 eq telnet**

CCNA(config-ext-nacl)#**deny tcp 192.168.5.0 0.0.0.255 any eq telnet**

CCNA(config-ext-nacl)#**permit ip any any**

CCNA(config)#**inter serial 0**

CCNA(config-if)#**ip access-group Kolukisaoglu in**

CCNA(config-if)#

ACCESS LİSTS VE DİSTRİBUTE LİST

Routing protokoller ile çalışırken bazı networklerin update edilmemesini isteyebiliriz. Bunun için passive interface komutu bir çözümdür ancak burada o interface' den hic bir update yapılmayacaktır. Oysa Access Listler ile birlikte oluşturulacak Distribute List' ler ile hangi networklerin update edileceğine hatta hangi networklerin update' inin alınacağına karar verebiliriz.

Durumu örnek çalışma ile özetleyeceğim.

Örnek topolojide her router için 3'er adet loopback interface oluşturdum ve konfigurasyon içinde bu loopbacklarıde Ripv2 içerişinde tanıttım. Başlangıçta Routing Table' lar A ve B için sırasıyla şu şekilde oluştu.

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

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 3 subnets
C       10.3.3.0 is directly connected, Loopback1
C       10.2.2.0 is directly connected, Loopback0
C       10.1.1.0 is directly connected, Ethernet0/0
     13.0.0.0/24 is subnetted, 3 subnets
R       13.3.3.0 [120/1] via 211.2.1.2, 00:00:01, Serial0/0
R       13.2.2.0 [120/1] via 211.2.1.2, 00:00:01, Serial0/0
R       13.1.1.0 [120/1] via 211.2.1.2, 00:00:01, Serial0/0
     211.2.1.0/30 is subnetted, 1 subnets
C       211.2.1.0 is directly connected, Serial0/0
Router#
```

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

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 3 subnets
R       10.3.3.0 [120/1] via 211.2.1.1, 00:00:20, Serial0/0
R       10.2.2.0 [120/1] via 211.2.1.1, 00:00:20, Serial0/0
R       10.1.1.0 [120/1] via 211.2.1.1, 00:00:20, Serial0/0
     13.0.0.0/24 is subnetted, 3 subnets
C       13.3.3.0 is directly connected, Loopback1
C       13.2.2.0 is directly connected, Loopback0
C       13.1.1.0 is directly connected, Ethernet0/0
     211.2.1.0/30 is subnetted, 1 subnets
C       211.2.1.0 is directly connected, Serial0/0
Router#
```

Her iki Router da Loopback adresler Directly Connected ve Ripv2 ile update edilmiş olarak görünmekteydi.

A routerinda iki adet acces list yazdım ve bunlar Ripv2 konfigurasyonuna Distribute list komutu ile bağladım.

```
 version 2
 network 13.0.0.0
 network 211.2.1.0
 distribute-list 10 out
 distribute-list 20 in
 no auto-summary
!
ip http server
ip classless
!
!
access-list 10 deny   13.1.1.0 0.0.0.255
access-list 10 permit any
access-list 20 deny   10.2.2.0 0.0.0.255
access-list 20 permit any
!
line con 0
line aux 0
line vty 0 4
!
!
end

Router#
```

10 numaralı access list ile 13.1.1.0 networkunun, 20 numaralı access list ile 10.2.2.0 networkunu yasaklamak için gereken satırları yazdıktan sonra “in” ve “out” olarak Rip’e uyguladım.

Yazılan satırların tam Türkçesi şu şekildedir: 10.2.2.0 networkunu içeriden dışarıya gönderme, 13.1.1.0 netwrokune ait update' i dışarıdan içeriye alma.

Bu durumda Routing Table'lar şu şekillerde değişti.

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

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 3 subnets
C       10.3.3.0 is directly connected, Loopback1
C       10.2.2.0 is directly connected, Loopback0
C       10.1.1.0 is directly connected, Ethernet0/0
     13.0.0.0/24 is subnetted, 2 subnets
R       13.3.3.0 [120/1] via 211.2.1.2, 00:00:15, Serial0/0
R       13.2.2.0 [120/1] via 211.2.1.2, 00:00:15, Serial0/0
     211.2.1.0/30 is subnetted, 1 subnets
C       211.2.1.0 is directly connected, Serial0/0
Router#
```

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

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 2 subnets
R       10.3.3.0 [120/1] via 211.2.1.1, 00:00:24, Serial0/0
R       10.1.1.0 [120/1] via 211.2.1.1, 00:00:24, Serial0/0
     13.0.0.0/24 is subnetted, 3 subnets
C       13.3.3.0 is directly connected, Loopback1
C       13.2.2.0 is directly connected, Loopback0
C       13.1.1.0 is directly connected, Ethernet0/0
     211.2.1.0/30 is subnetted, 1 subnets
C       211.2.1.0 is directly connected, Serial0/0
Router#_
```

EIGRP (ENHANCED INTERRİOR GATEWAY ROUTİNG PROTOCOL)

Cisco daha önce geliştirdiği IGRP' nin yetersiz kalması ve RIP'in RIPv2'ye yükseltilmesiyle boş durmamış, EIGRP' yi geliştirmiş ve bu protokolü sınıflandırmada da, hem Distance Vektör hem de Link State protokollerin özelliklerini taşıdığı için Hybrid başlığı altına yerleştirmiştir.

Bütün Routing protokolleri gibi EIGRP' de Routing update mantığı ile çalışır fakat Rip ve IGRP' den farklı olarak belirli zaman aralıklarında tüm networklerin bilgisini göndermektense küçük hello paketleri yollayarak komşu routerlarının up olup olmadıklarını

kontrol eder. Komşu routerlardan gelen Acknowledgement paketleriyle o routerın hala up olduğu kabul eder.

Hello ve Acknowledgement mesajları dikkate alındığında burada TCP gibi bir protokolün kullanılması gerekliliği ortaya çıkar. Fakat bu işlemler sırasında EIGRP yine Cisco'nun geliştirdiği ve RTP (Reliable Transport Protocol) protokolünü kullanır. Çalışma mantığı TCP ile aynıdır.

Gerektigi zamanlarda, sözgelimi yeni bir router eklendiğinde veya bir router down olduğunda, "ADD" yada "DELETE" bilgilerini yollar.

Bir router ortama dâhil olduğunda öncelikle bir Query paketi yollar ve bu paketlerden gelen Reply' lar ile komşu routerları hakkında bilgi edinir ve topoloji tablosunu oluşturur.

Buraya kadar anlattıklarımızla EIGRP' nin 5 farklı paket ile çalıştığını söyleyebiliriz.

EIGRP Paketleri

Hello	Acknowledgement
Update	Query
Reply	

EIGRP Hello paketlerini 224.0.0.10 multicast ip adresi üzerinden gönderir. T1 ve üzeri bant genişliklerinde 5 saniye de bir gönderilen bu paketler T1 den daha düşük bant genişliklerinde 60 saniyede bir gönderilr. (Hold Time=3 X hello interval)

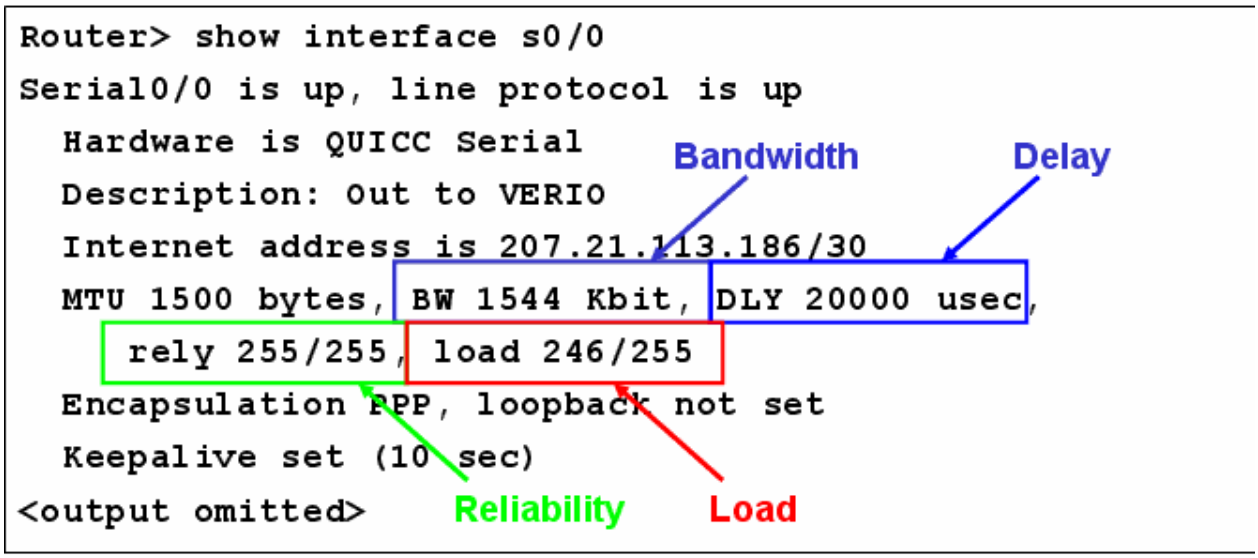
Acknowledgement paketleri data içermeyen paketlerdir ve Güvenli iletişimi sağlar. Hello paketlerinin multicast olmasına karşın Acknowledgement paketleri unicast çalışırlar.

Update paketleri sistemdeki bir router yeni bir network bulduğunda ya da kaybettiginde, metrik hesabinda bir değişiklik olduğunda ve successor değiştiğinde gönderilir. Bu aksiyonlardan biri gerçeklestiginde EIGRP konuşan bir Router bütün komşularını multucastupdate gönderir.

Query paketleri bir router herhangi bir şekilde, yeni, özel bir bilgiye ihtiyaç duyulduğunda gönderilir. Sözgelimi successor' i down olan ve Feasible succesr' i bulunmayan bir router Query paketleri gönderir ve cevaplar Reply paketleri ile doner. Query paketleri multicast iken Reply paketleri unicasttir.

EIGRP metrik Hesabi

EIGRP metrik hesabında K1 vw K3 değerlerini kullanır. (Banswidth ve Delay)



EIGRP ve IGRP bant genişliklerini aynı formul ile hesaplarlar.

```
metric = [K1 * bandwidth + (K2 * bandwidth)/(256 - load)+(K3 * delay)]* [K5/(reliability + K4)]
```

Fakat EIGRP için K2, K4 ve K5 default olarak 0 sayilir.

EIGRP Table'ları

EIGRP çalışma mantığı içerişinde bütün komşularını Neighbpr Table' da ve hedef networke olan bütün yolları da Topology Table' da tutar. Bu bilgiler ışığında en iyi yol seçimini yapar.

```
RouterC#show ip eigrp neighbors
IP-EIGRP neighbors for process 44
H   Address         Interface   Hold Uptime   SRTT   RTO  Q  Seq
                                (sec)         (ms)       Cnt Num
0   192.168.0.1     Se0           11 00:03:09 1138   5000  0  6
1   192.168.1.2     Et0           12 00:34:46    4    200  0  4
```

Neighbor Table da komşu routerların network katmanı adresleri (ip adresleri), Q ile gösterilen ve sıradan gönderilmeyen bekleyen paket sayısını ifade eden bir değer (ki bu değer 0 dan büyük ise router da olası bir problemden bahsedilebilir), SRTT ile gösterilen ve komşu routerlara gönderilen ve alınan paketler için gecen ortalama süreyi gösteren bir değer ve Hold Time değeri bulunur.

```
RouterB#show ip eigrp topology
IP-EIGRP Topology Table for process 44
Codes: P - Passive, A - Active, U - Update, Q - Query, R -
   Reply, r - Reply status
P 206.202.17.0/24, 1 successors, FD is 2195456
         via 206.202.16.1 (2195456/2169856), Ethernet0
P 206.202.18.0/24, 2 successors, FD is 2198016
         via 192.168.0.2 (2198016/284160), Serial0
         via 206.202.16.1 (2198016/2172416), Ethernet0
```

EIGRP hedef networklere gitmek için kullanacağı yolların bilgisini ise Topology Table'inda saklar. Bu table da bulunan bilgilere dayanarak successor ve Feasible successor' u secer.

Routing Table ise successor (best route) olarak seçilen yolun bulunduğu yerdir.

```
RouterB#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -
   BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * -

Gateway of last resort is not set
C    10.1.1.0 is directly connected, Serial0
D    172.16.0.0 [90/2681856] via 10.1.1.0, Serial0
D EX 192.168.1.0 [170/2681856] via 10.1.1.1, 00:00:04, Serial0
```

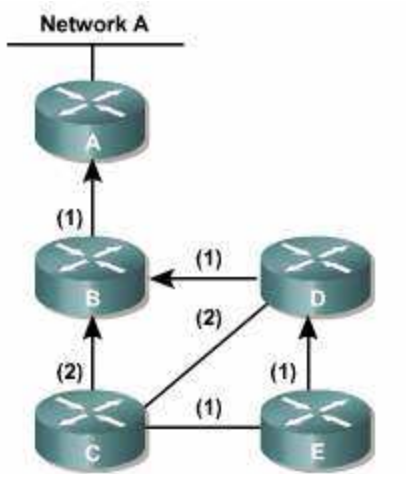
EIGRP harici bir protokolden gelen update bilgileri Routing Table'inda EX (external) olarak isaretler.

EIGRP topolojini oluştururken Dual Algoritmasını kullanır. Bu algoritma ile kendisine bir en iyi yol (Successor) bir de yedek sayılabilecek en iyi ikinci yol (Feasible successor) seçer.

Successor seçerken tek dayanağı mümkün olan yollara ait metrik toplamlarının (Her biri Feasible Distance olarak adlandırılır.) en küçügünü kullanır. Feasible Distance' ları eşit olan birden fazla yol var ise en düşük Reported Distance' a sahip olan yolu seçer. Burada Reported Distance' dan kasıt adından anlasılacağı gibi bir sonraki router için geçerli olan Feasible Distance' dır.

Burada bir önemli kuralda, Feasible successor seçilen yola ait Reported Distance değeri, successor seçilen yolun Feasible Distance' ından küçük olmalıdır, aksi takdirde loop başlar.

Örnek üzerinde açıklamak gerekirse;



(Parantez 5çindeki değerler metrik değerleridir.)

C Routerından Net A ya gidilme istendiğinde topoloji söyle olacak;

Next Hop	FD	RD	Topoloji
B	3		Successor
D	4	2	FS
E	4	3	

En iyi yol B routerı üzerinden gidilen yoldur, çünkü metrik değerleri toplandığında en küçük değere (Feasible Distance) sahiptir.

Feasible Distance' ları eşit olan D ve E routerları üzerinde gidilen yollar için Reported Distance' ı küçük olan (D) Feasible successor seçilir. (Burada D routerı için RD değerinin B routerı FD değerinden küçük olduğuna dikkat edin)

D Router'ından Net A ya gidilme istendiğinde topoloji söyle olacak;

Next Hop	FD	RD	Topoloji
B	2		Successor
E	5	4	
C	5	3	

Burada görüldügü gibi Feasible successor seçilemiyor çünkü Reported Distance değerleri hem E hem de C routerı için B routerının Feasible Distance' ından büyük.

(Feasible successor' a default route' da denmektedir.)

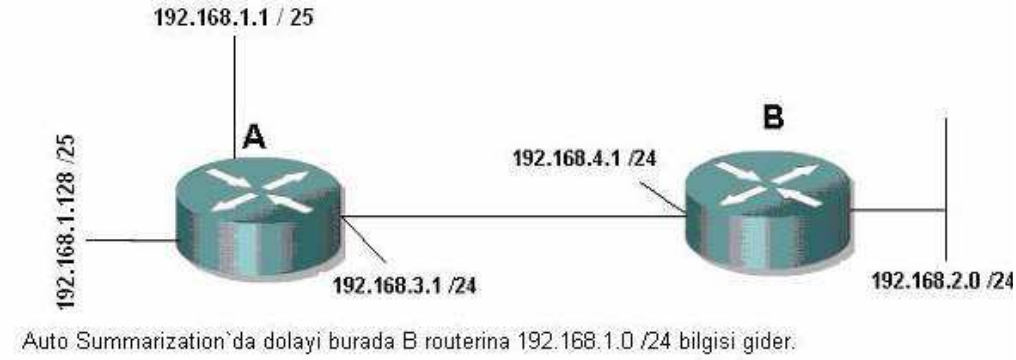
Not: EIGRP IPX ve AppleTalk networklerini de destekler ve bu networklere ait Neigbor, Topology ve Routing table' ları ayrı ayrı tutar.

Auto summarization

Auto summarization ve Load Balancing özellikleri detaylı olarak incelenmelidir. (Auto summarization özelliği Ripv2' de de vardır.)

Sözgelimi elimizde, interfacelerinde sırasıyla s0=192.168.1.1, s1=10.1.1.0 / 25 ve s2=10.1.1.128 / 25 networkleri olan bir router (Router A) var ve s0 interface'inden başka bir routera (Router B) bağlı.

Routerlar EIGRP ile konfigure edildiği zaman A routera B routerına Auto summarization yapacak ve 10.1.1.0 / 24 networkü bilgisini update edecektir. Bu istenmeyen bir durum ise "no auto-summarization" komutu ile özellik kaldırılabilir.



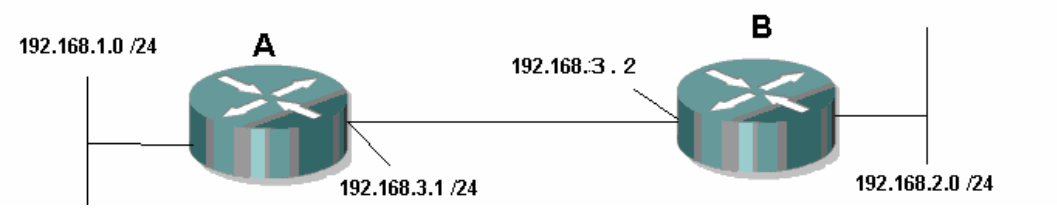
Auto summarization özelliği "no auto-summary" komutu ile kaldırılabilir.

Router(config)#router eigrp 34

Router(config-router)#no auto-summary

EIGRP KONFİGURASYONU

EIGRP de Tıpkı IGRP gibi konfigure edilir.



Router A

RouterA(config)#router eigrp 34

RouterA(config-router)#network 192.168.1.0

RouterA(config-router)#network 192.168.3.0

RouterA(config-router)#no auto-summary

Router B

RouterB(config)#router eigrp 34

RouterB(config-router)#network 192.168.2.0

RouterB(config-router)#network 192.168.3.0

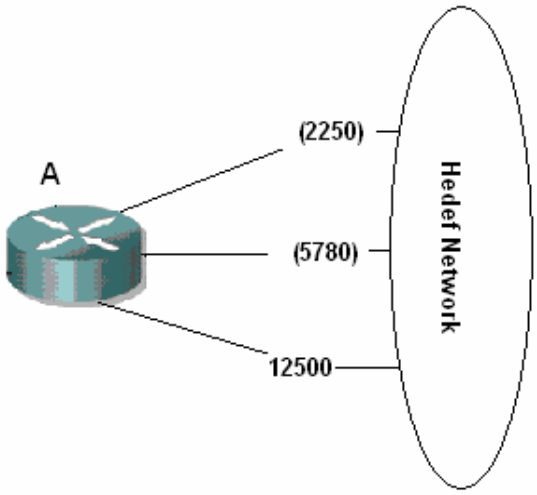
RouterB(config-router)#no auto-summary

Load Balancing

Rip söz konusu olduğunda, metrik hesabı tamamen hop sayısına bağlı olduğundan aynı metriğe sahip birden fazla yol olması ve bu yollar arasından router ın load Balancing yapması ihtimaller arasındadır. Fakat EGRP' yi de içene alan diğer bütün protokoller de metrik hesabı birçok değerle birlikte yapıldığı için, aynı metriğe sahip birden fala yolun olması çok çok zor bir ihtimaldir.

Bu durumda load Balancing imkânsızdır. Fakat EIGRP "variance n" komutu ile load balancing yapılmasına izin verir. (Bu özellik IGRP' de de vardır.)

Bu komutta n ile belirtilen bölüm, bizim belirleyeceğimiz bir sayıdır. Ve komut işletilmeye başladığında EIGRP en düşük metrik değerini alır, n ile çarpar ve çıkan sonucun altında yer alan bütün metrik değerlerine sahip yollar arasında load Balancing yapmaya başlar.



Burada variance 3 gibi bir komut kullanirsak,, bu komut en dusuk metric degeri olan 2250` yi 3 ile carpacak ve cikan sonucun (6750) altinda metric degerlerine sahip yollar (2250 ve 5780 metricli yollar) arasinda load balancing yapacaktir.

Örnek Konfigurasyon;

Router(config)#router eigrp 14

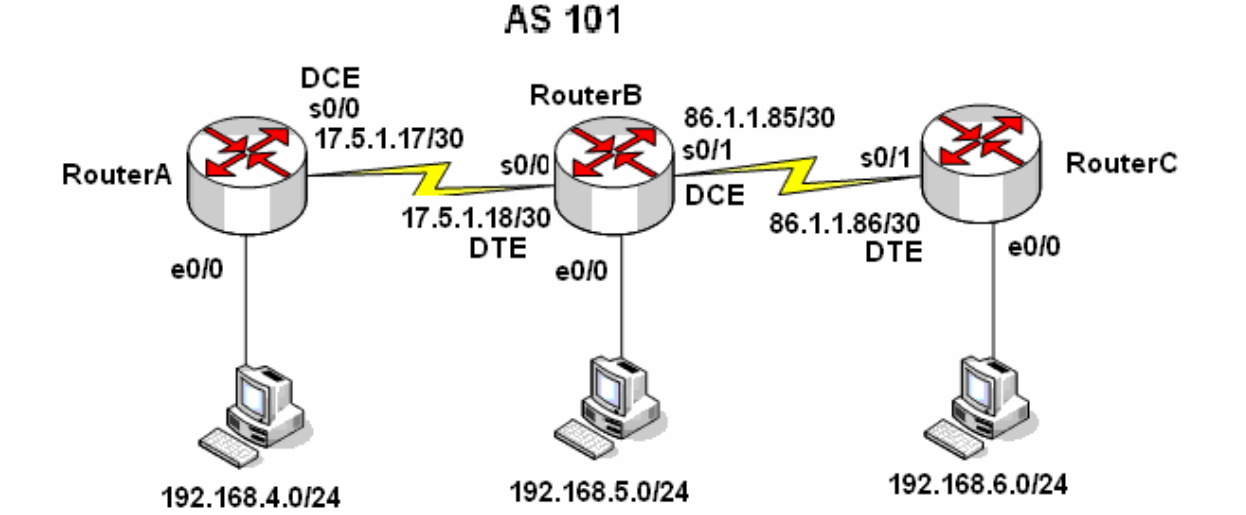
Router(config-router)#network 10.1.1.0

Router(config-router)#network 10.2.1.0

Router(config-router)#network 10.3.1.0

Router(config-router)#variance 2

EIGRP Laboratuar ÇALIŞMASI



Burada yapılan çalışmada AS olarak 101 seçilmiştir.

Laboratuar ortamında clock üretimini sağlayacak DCE kabloların takıldığı interfacelere uygulama içerişinde clock rate komutu verilmiştir.

Auto summarization özelliği kapatılmıştır.

Her bir Router, dan Runnin-config dosyaları, Routing Table'ları, Neighbor Table'ları ve Topology Table' ları alınmıştır.

RouterA#show running-config

Building configuration...

Current configuration : 642 bytes

!

version 12.1

service timestamps debug uptime

service timestamps log uptime

no service password-encryption

!

hostname RouterA

!

!

memory-size iomem 10

```
ip subnet-zero
!
!
interface Ethernet0/0
ip address 192.168.4.1 255.255.255.0
!
interface Serial0/0
ip address 17.5.1.17 255.255.255.252
clockrate 64000
!
interface BRI0/0
no ip address
shutdown
isdn x25 static-tei 0
!
router eigrp 101
network 17.5.1.16 0.0.0.3
network 192.168.4.0
no auto-summary
no eigrp log-neighbor-changes
!
ip classless
!
!
line con 0
line aux 0
line vty 0 4
!
no scheduler allocate
end
```

RouterA#

```
RouterB#show running-config
Building configuration...
Current configuration : 640 bytes
!
version 12.2
```

```
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname RouterB
!
memory-size iomem 10
ip subnet-zero
!
interface Ethernet0/0
ip address 192.168.5.1 255.255.255.0
half-duplex
!
interface Serial0/0
ip address 17.5.1.18 255.255.255.252
!
interface Serial0/1
ip address 86.1.1.85 255.255.255.252
clockrate 64000
!
router eigrp 101
network 17.5.1.16 0.0.0.3
network 86.1.1.84 0.0.0.3
network 192.168.5.0
no auto-summary
!
ip classless
!
!
dial-peer cor custom
!
gatekeeper
shutdown
!
line con 0
line aux 0
line vty 0 4
```

!
end

RouterB#
RouterC#sh running-config
Building configuration...
00:29:43: IP-EIGRP: Neighbor 192.168.4.1 not on common subnet for Ethernet0/0
(192.168.6.1 255.255.255.0)
Current configuration : 620 bytes
!
version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname RouterC
!
!
memory-size iomem 10
ip subnet-zero
!
interface Ethernet0/0
ip address 192.168.6.1 255.255.255.0
!
interface Serial0/0
no ip address
shutdown
no fair-queue
!
interface Serial0/1
ip address 86.1.1.86 255.255.255.252
!
router eigrp 101
network 86.1.1.84 0.0.0.3
network 192.168.6.0
no auto-summary
no eigrp log-neighbor-changes

!

ip classless

!

line con 0

line aux 0

line vty 0 4

!

no scheduler allocate

end

RouterC#

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

Gateway of last resort is not set

     17.0.0.0/30 is subnetted, 1 subnets
C       17.5.1.16 is directly connected, Serial0/0
     86.0.0.0/30 is subnetted, 1 subnets
D       86.1.1.84 [90/2681856] via 17.5.1.18, 00:14:16, Serial0/0
C    192.168.4.0/24 is directly connected, Ethernet0/0
D    192.168.5.0/24 [90/2195456] via 17.5.1.18, 00:01:44, Serial0/0
D    192.168.6.0/24 [90/2707456] via 17.5.1.18, 00:13:29, Serial0/0
RouterA#
```

39:45 baglanildi | OtoAlgila | 9600 8-N-1 | Kaydir | büyh | SAYI | Yakala | Yazdirma yankisi

```
RouterA#sh ip eigrp neighbors
IP-EIGRP neighbors for process 101
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq Type
                                        (sec)         (ms)       Cnt Num
0   17.5.1.18               Se0/0         13 00:10:14  607  3642  0  23
RouterA#sh ip eigrp top
RouterA#sh ip eigrp topology
IP-EIGRP Topology Table for AS(101)/ID(192.168.4.1)

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

P 86.1.1.84/30, 1 successors, FD is 2681856
        via 17.5.1.18 (2681856/2169856), Serial0/0
P 17.5.1.16/30, 1 successors, FD is 2169856
        via Connected, Serial0/0
P 192.168.4.0/24, 1 successors, FD is 281600
        via Connected, Ethernet0/0
P 192.168.6.0/24, 1 successors, FD is 2707456
        via 17.5.1.18 (2707456/2195456), Serial0/0
RouterA#
```

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

Gateway of last resort is not set

     17.0.0.0/30 is subnetted, 1 subnets
C       17.5.1.16 is directly connected, Serial0/0
     86.0.0.0/30 is subnetted, 1 subnets
C       86.1.1.84 is directly connected, Serial0/1
D    192.168.4.0/24 [90/2195456] via 17.5.1.17, 00:12:37, Serial0/0
C    192.168.5.0/24 is directly connected, Ethernet0/0
D    192.168.6.0/24 [90/2195456] via 86.1.1.86, 00:11:48, Serial0/1
RouterB#_
```

```
IP-EIGRP neighbors for process 101
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq Type
                                        (sec)         (ms)       Cnt Num
1   86.1.1.86               Se0/1         14 00:12:44  672  4032  0  14
0   17.5.1.17               Se0/0         12 00:13:36   21   200  0  14
RouterB#show ip eigrp topology
IP-EIGRP Topology Table for AS(101)/ID(192.168.5.1)

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

P 86.1.1.84/30, 1 successors, FD is 2169856
        via Connected, Serial0/1
P 17.5.1.16/30, 1 successors, FD is 2169856
        via Connected, Serial0/0
P 192.168.4.0/24, 1 successors, FD is 2195456
        via 17.5.1.17 (2195456/281600), Serial0/0
P 192.168.5.0/24, 1 successors, FD is 281600
        via Connected, Ethernet0/0
P 192.168.6.0/24, 1 successors, FD is 2195456
        via 86.1.1.86 (2195456/281600), Serial0/1
RouterB#
```

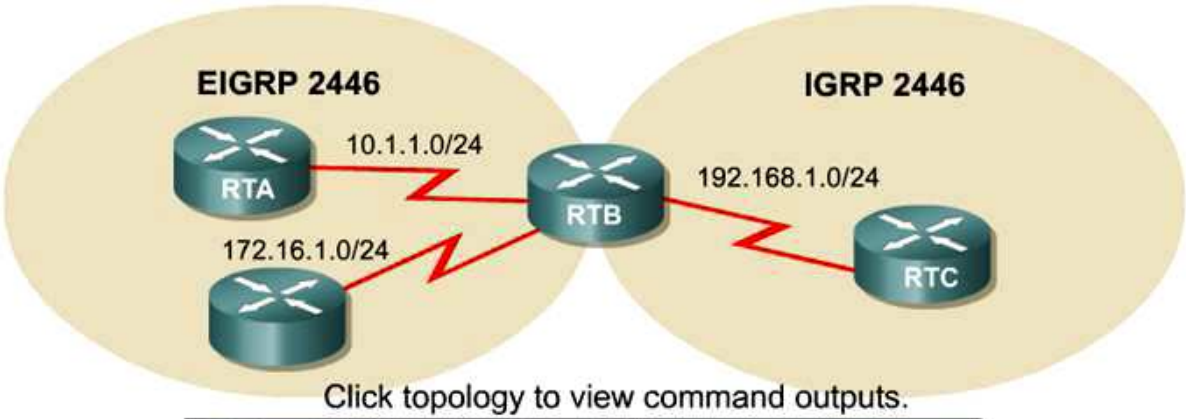
```
RouterC#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP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

Gateway of last resort is not set

     17.0.0.0/30 is subnetted, 1 subnets
D       17.5.1.16 [90/2681856] via 86.1.1.85, 00:00:46, Serial0/1
     86.0.0.0/30 is subnetted, 1 subnets
C       86.1.1.84 is directly connected, Serial0/1
D    192.168.4.0/24 [90/2707456] via 86.1.1.85, 00:00:46, Serial0/1
D    192.168.5.0/24 [90/2195456] via 86.1.1.85, 00:00:46, Serial0/1
C    192.168.6.0/24 is directly connected, Ethernet0/0
RouterC#
```

```
RouterC#show ip eigrp neighbors
IP-EIGRP neighbors for process 101
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq Type
                                        (sec)         (ms)       Cnt Num
0   86.1.1.85               Se0/1         14 00:05:50   28   200  0  21
RouterC#
```

```
RouterC#show ip eigrp topology
IP-EIGRP Topology Table for AS(101)/ID(192.168.6.1)

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

P 86.1.1.84/30, 1 successors, FD is 2169856
         via Connected, Serial0/1
P 17.5.1.16/30, 1 successors, FD is 2681856
         via 86.1.1.85 (2681856/2169856), Serial0/1
P 192.168.4.0/24, 1 successors, FD is 2707456
         via 86.1.1.85 (2707456/2195456), Serial0/1
P 192.168.5.0/24, 1 successors, FD is 2195456
         via 86.1.1.85 (2195456/281600), Serial0/1
P 192.168.6.0/24, 1 successors, FD is 281600
         via Connected, Ethernet0/0
RouterC#_
```

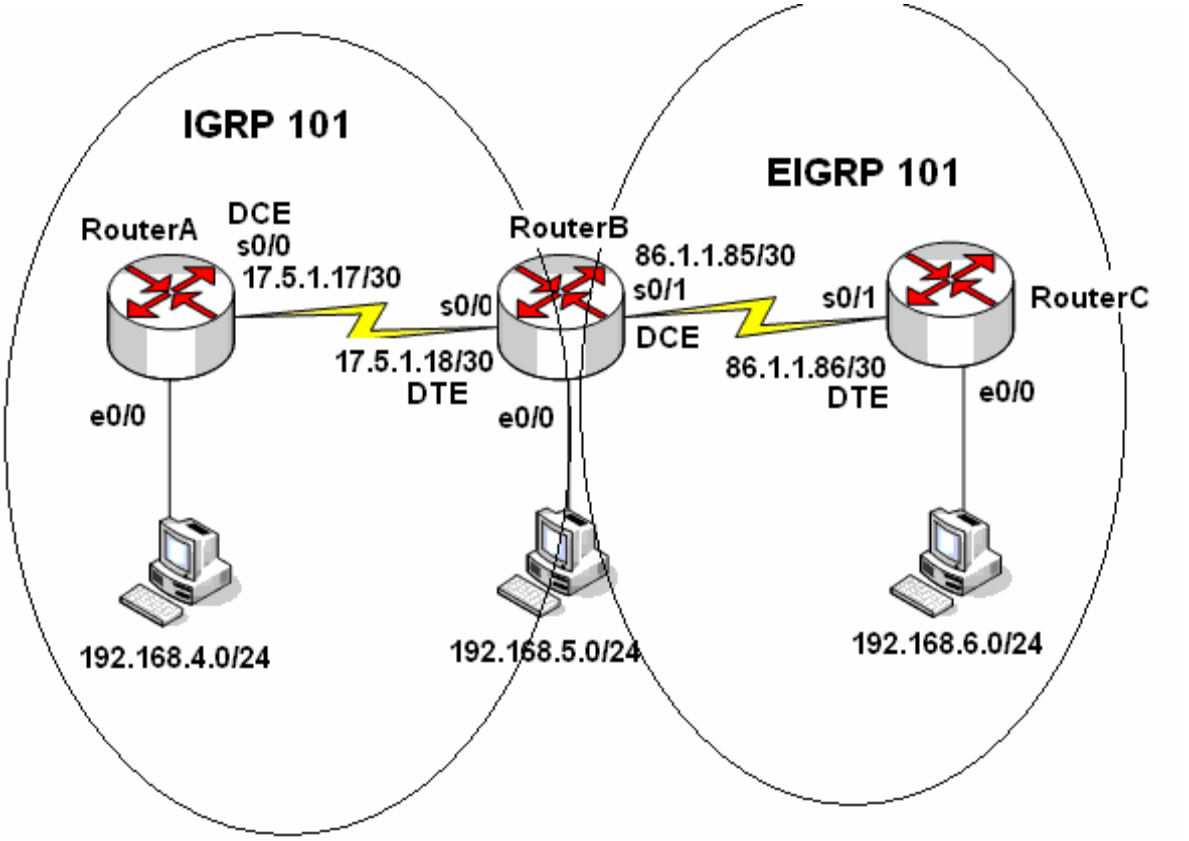
```
:32:29 baglanildi | OtoAlgila | 9600 8-N-1 | Kaydir | büyh | SAYI | Yakala | Yazdirma yankisi
```

EIGRP VE IGRP BİRLİKTE ÇALIŞMASI




```
RTB(config)#router igrp 2446
RTB(config-router)#network 192.168.1.0
RTB(config)#router eigrp 2446
RTB(config-router)#network 10.1.1.0
RTB(config-router)#network 172.16.1.0
```

IGRP ve EIGRP aynı AS içerişinde birbirleriyle haberleşirler. Burada özel olarak dikkat edilecek tek nokta EIGRP konuşan Routerların Routing Tebler' larında IGRP konuşan Routerlara giden yolları External olarak etiketlemiş olmasıdır.



Hem IGRP hem de EIGRP için AS numarası 101 seçilmiştir.

Router B üzerinde hem IGRP hem EIGRO konfigurasyonları yapılmıştır.

Bütün Routerların Routing Teble' ları ve B routerinin running-config dosyası incelenmek üzere alınmıştır.

RouterB#show run

```
Building configuration...
Current configuration : 670 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname RouterB
!
memory-size iomem 10
ip subnet-zero
!
interface Ethernet0/0
ip address 192.168.5.1 255.255.255.0
half-duplex
!
interface Serial0/0
ip address 17.5.1.18 255.255.255.252
!
interface Serial0/1
ip address 86.1.1.85 255.255.255.252
clockrate 64000
!
router eigrp 101
network 86.1.1.84 0.0.0.3
network 192.168.5.0
no auto-summary
!
router igrp 101
network 17.0.0.0
network 192.168.5.0
!
ip classless
!
dial-peer cor custom
!
```

gatekeeper

shutdown

line con 0

line aux 0

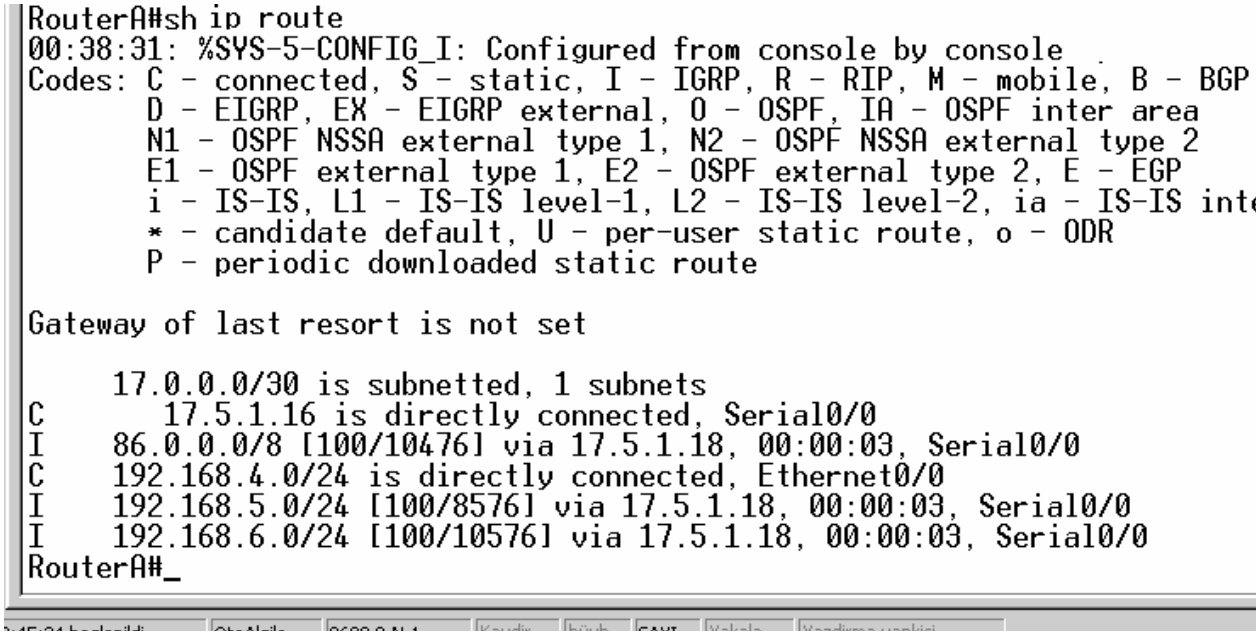
line vty 0 4

!

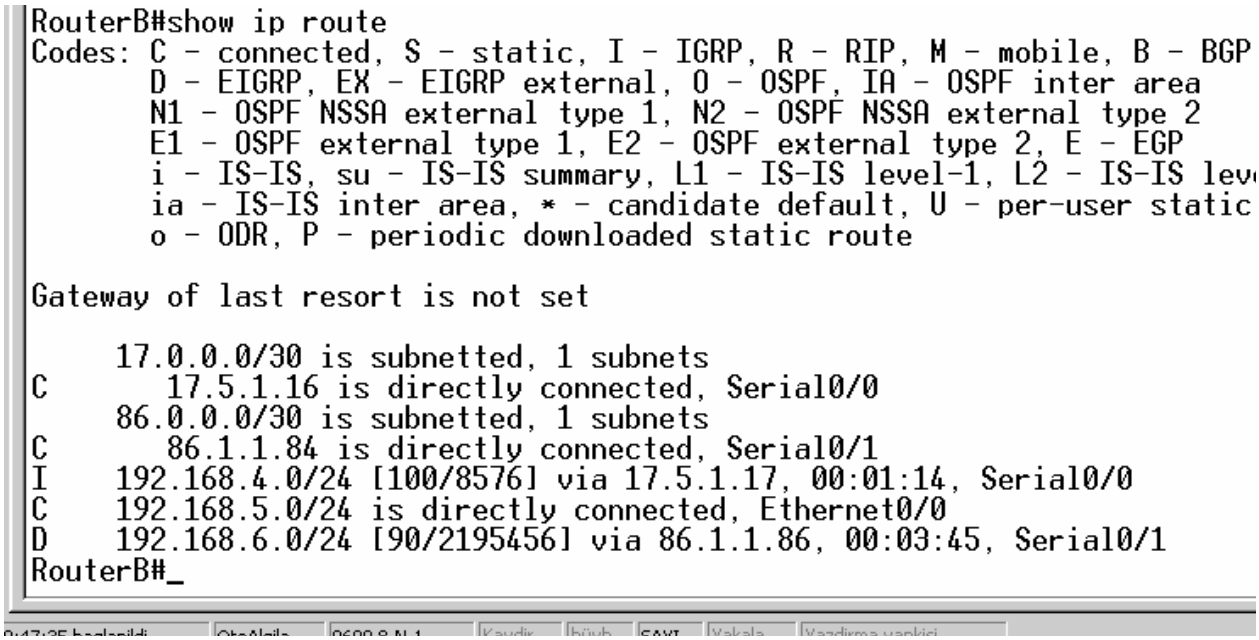
end

RouterB#

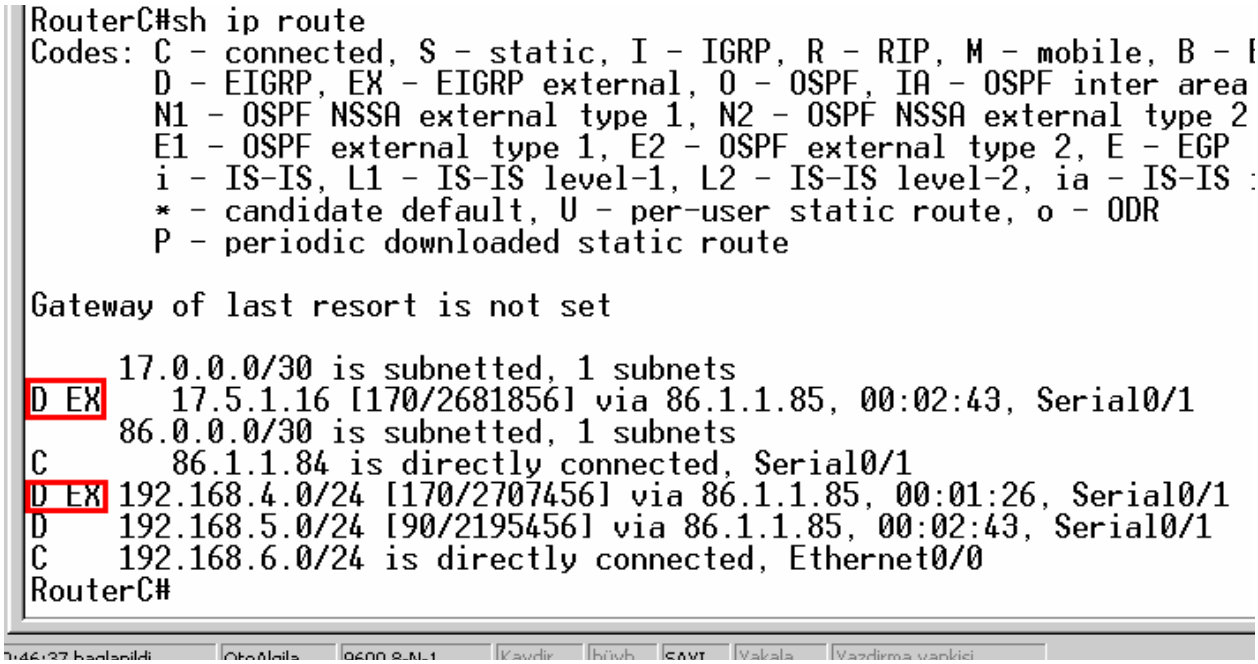
RouterA Routing Table'i



RouterB Routing Table'i



RouterC Routing Table'i



OSPF (OPEN SHORTEST PATH FİRST)

OSPF Link State Protocol olup, ulaşılmak istenen networke giden en kısa yolu Dijkstra algoritması kullanarak tespit etmektedir.

"Hello" protokolü ile OSPF çalışan routerlar komşularını keşfederler. Hello paketleri her 10 saniye de bir gönderilir ve bu paketlerden alınan sonuçlara göre OSPF database oluşturulur.

OSPF metrik için Cost adı verilen değeri kullanırlar. Standart bir tanımı yapılamamakla birlikte Cisco Routerlar da ön görülen OSPF metrigi bant genişliği ile ters orantılıdır.

(cost= 10.000.000 / bantgenişliği)

Bu protokolde, networkteki yönlendirme bilgilerini kendisinde toplayıp, diğerlerine dağıtacak bir router vardır. Bu routera Designated Router denir ve DR olarak kısaltılır.

DR aktif olmadığı durumlarda Backup Designated Router devreye direr. (BDR)

HELLO PAKET İÇERİGİ (Type 1)

Router ID: Router da konfigüre edilen en yükse IP adresidir.

Network Mask: Router ID' yi belirleyen interface'in ağ maskesidir.

Area ID: Hello paketi gönderen routerın interface'inin alan kimliğidir. Hello paketindeki bilgilerin geçerli olabilmesi için bu paketi alan routerın interface'i ile aynı olmalıdır.

Router Priority: Routerın DR veya BDR seçimini belirlemektedir.

Hello Aralıgı: Hello paketleri arasındaki süredir ve 10 saniyedir.

Router(config-if)#ip ospf hello-interval n komutuyla değiştirilebilir. n bizim belirleyeceğimiz birim saniye olan bir değerdir. Burada dikkat edilmesi gereken bir konu ise

birbirine bağla olan iki interface ‘ inde hello zaman aralığının eşit olması gerektigidir. Aksi takdirde komşuluk ilişkisi kurulamaz.

Ölüm Aralıgı (Dead Interval): Komşu router ile bağlantının koptuğunu belirten süredir. (Hello Aralıgının 4 katıdır.)

DR IP adresi: Mevcut DR ip adresidir. Bu adresi öğrenen Routerlar, OSPF mesajlarını bu ip adresine gönderirler.

BDR IP Adresi: Mevcut BDR ip adresidir. DR aktif olmadığı zaman OSPF mesajları bu ip adresine gönderilir.

Komşu Router ID’leri: Komşuluk tablosunda bulunan routerların ip adresleridir. Router kendi ip adresini bu alanda görürse database paylaşımı gerçekleştirilir.

Authentication Information: Kimlik doğruluma tipi ve bilgisini içerir.

Stub Area Flağ: Hangi tip LSA (Link State Advertisement) mesajlarının gönderileceği ve alınacağı bilgisini içerir.

Hello paketleri dışında OSPF konuşan Routerların birbirlerine gönderdikleri 4 ayrı paket şekli daha vardır. Bunlar;

Type2: DBD yani Database Descriptiin paketleri olarak bilinir ve Routerların Link durumları hakkında özet bilgiler içerir.

Type3: LSR yani Link State Request paketleri olarak bilinir. Routerlar DBD paketleri ile öğrendikleri bilgilerin detayı için diğer Routerlara LSR paketleri gönderirler.

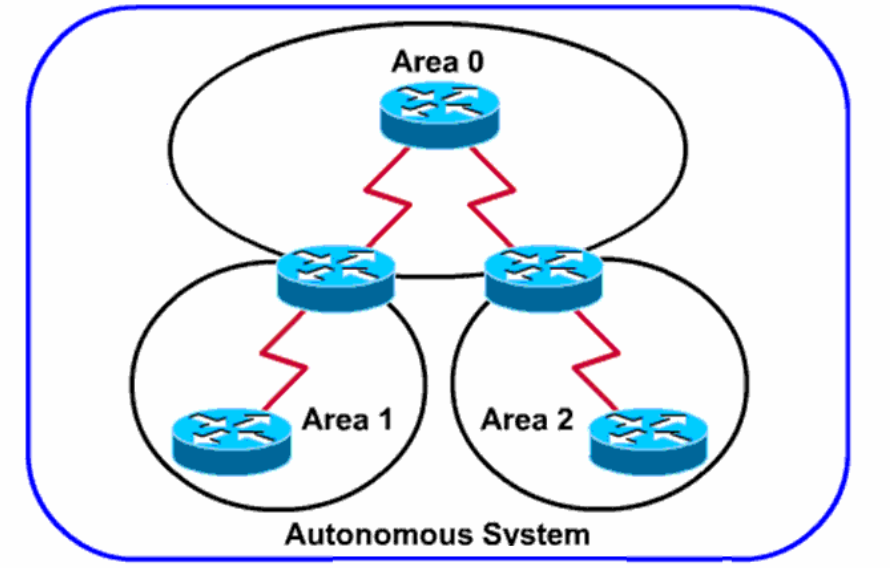
Type4: LSU yani Link State Update paketleri olarak bilinir. LSR ile istenen Link State Advertisements (LSAs)paketlerini tasir.

Type5: LSA yani Link State Acknowledgement paketleridir ve routerlar arasında paketlerin aldığı onay bilgisini taşır.

OSPF Area

Ospf çalışma mantığı arealar üzerine kurulmuştur ve bu sayede bir dizayn hiyerarşisi sağlanabilmektedir. Bu hiyerarşik yapının convergence’ i hızlandırdığı da söylenebilir.

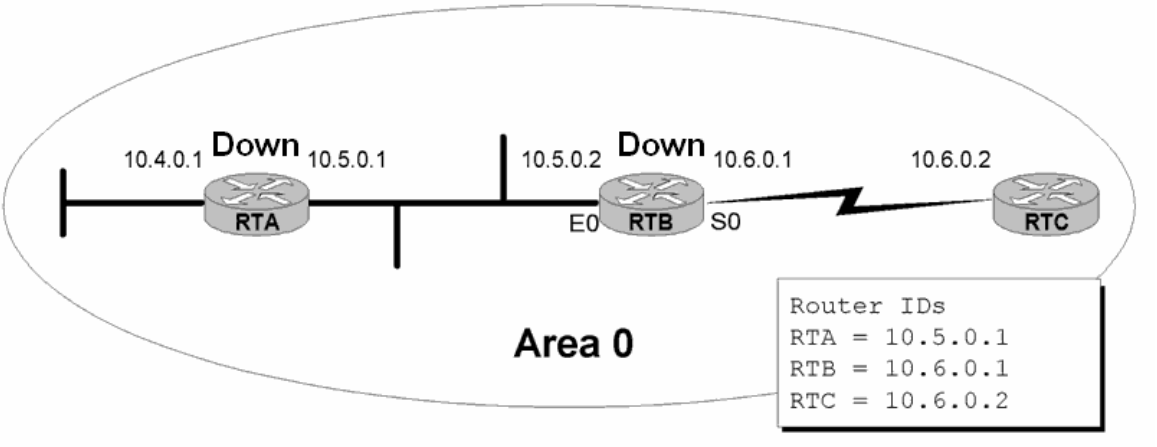
Ospf in merkezi area 0’ dir. Area 0 backbone area olarak adlandırılır ve farklı arealar olduğunda o arealar içinde area 0 ile konuşan interface’ e sahip routerlar olmalıdır.



OSPF KOMŞULUGU

OSPF ile konfigüre edilmiş routerlar 7 adım ile diğer routerlar ile komşuluk kurarlar. Bu adımlar şunlardır;

Down, Hello paketinin alınamadığı durumdur. Yeni bir router networke katıldığında down durumdadır. Routerlar networkteki varlıklarını duyurmak için 224.0.0.5 multicast adresini kullanarak Hello paketleri gönderir.



Init, Diğer routerlardan cevap bekleme adımıdır.

Two-Way, Diğer routerların gönderdikleri Hello mesajlarının Komşu Router ID alanında kendi 5P adreslerini gördükleri durumdur. Artık iki router komşuluk bağı kurmustur.

Exstart, Karsılıklı iki router arasında paket alıs verişinin yapıldığı andır. Bu adımda iki Router dan biri master diğeri slave rolü üstlenir. Burada seçim sadece iletişimi başlatacak routerı belirlemek için kullanılır, bu seçim herhangi birine bir üstünlük sağlamaz.

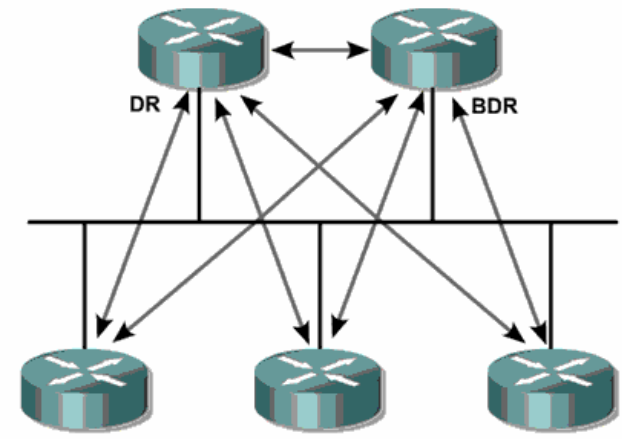
Exchange, Routerların bilgi alısverişi yaptıkları adımdır.

Loading, Exchange adımı ile elde edilen yeni yollar / networkler hakkındaki bilgileri ilgili routerlardan alma adımıdır.

Full, Yönlendirme bilgilerinin senkron hale getirilmesi durumudur.

DR ve BDR Seçimi

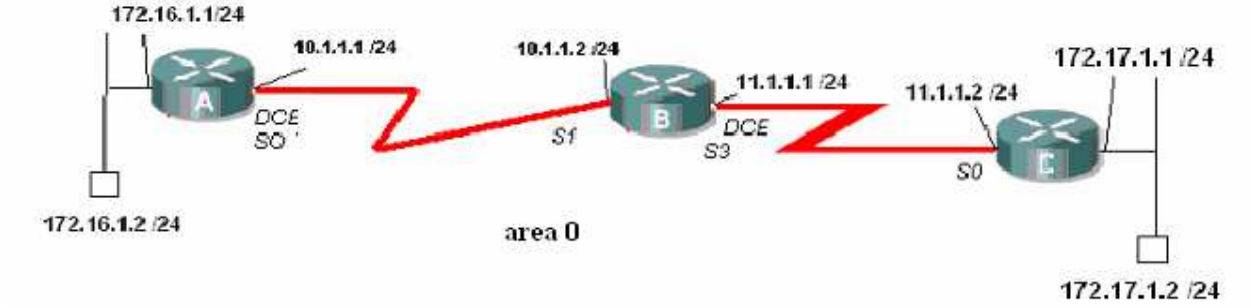
Multi-acces networklerde isler biraz daha farklı yurur. Bu networklerde Two Way halindeyken ortamda bütün trafiği yönetecek bir router seçilir ki buna Designated Router (DR) denir. Ve yine Bacup Designated Router (BDR) denen ve DR ‘ in yedegi olan bir router daha seçilir.



DR ve BDR seçimleri Router ID' ler ile yapılir. En yüksek Router ID' ye sahip router DR ve ikinci en yüksek ID, ye sahip router BDR seçilir.

Router ID bir routerin aktif olan interfacelerindeki en yüksek ip adresidir. Burada loopback adreslerin bir ayrıcaligi vardır. Eğer bir Routerda loopback adresi tanımlanmışsa o routerin ID' si loopback ip'sidir. Ip adrsinin küçük veya büyük olması durumu değiştirmez.

SİNGLE AREA OSPF KONFİGÜRASYONU



Router A Konfigürasyonu;

```
A(config)#router ospf 1
A(config-router)#network 172.16.1.0 0.0.0.255 area 0
A(config-router)#network 10.1.1.0 0.0.0.255 area 0
A(config-router)#exit
A(config)#
```

Router B Konfigürasyonu;

```
B(config)#router ospf 1
B(config-router)#network 10.1.1.2 0.0.0.255 area 0
B(config-router)#network 11.1.1.2 0.0.0.255 area 0
B(config-router)#exit
B(config)#_
```

Router C Konfigürasyonu;

```
C(config)#router ospf 1
C(config-router)#network 11.1.1.0 0.0.0.255 area 0
C(config-router)#network 172.17.1.0 0.0.0.255 area 0
C(config-router)#exit
C(config)#
```

(A Router' ının Routing Table' ı)

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

Gateway of last resort is not set

     172.17.0.0/24 is subnetted, 1 subnets
O       172.17.1.0 [110/943] via 10.1.1.2, 00:45:03, Serial0/0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.1.0 is directly connected, Ethernet0/0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Serial0/0
     11.0.0.0/24 is subnetted, 1 subnets
O       11.1.1.0 [110/933] via 10.1.1.2, 00:45:03, Serial0/0
A#_
```

(B Router' ının Routing Table' ı)

```
B#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     172.17.0.0/24 is subnetted, 1 subnets
O       172.17.1.0 [110/879] via 11.1.1.2, 00:43:08, Serial3
     172.16.0.0/24 is subnetted, 1 subnets
O       172.16.1.0 [110/74] via 10.1.1.1, 00:43:08, Serial1
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Serial1
     11.0.0.0/24 is subnetted, 1 subnets
C       11.1.1.0 is directly connected, Serial3
B#
```

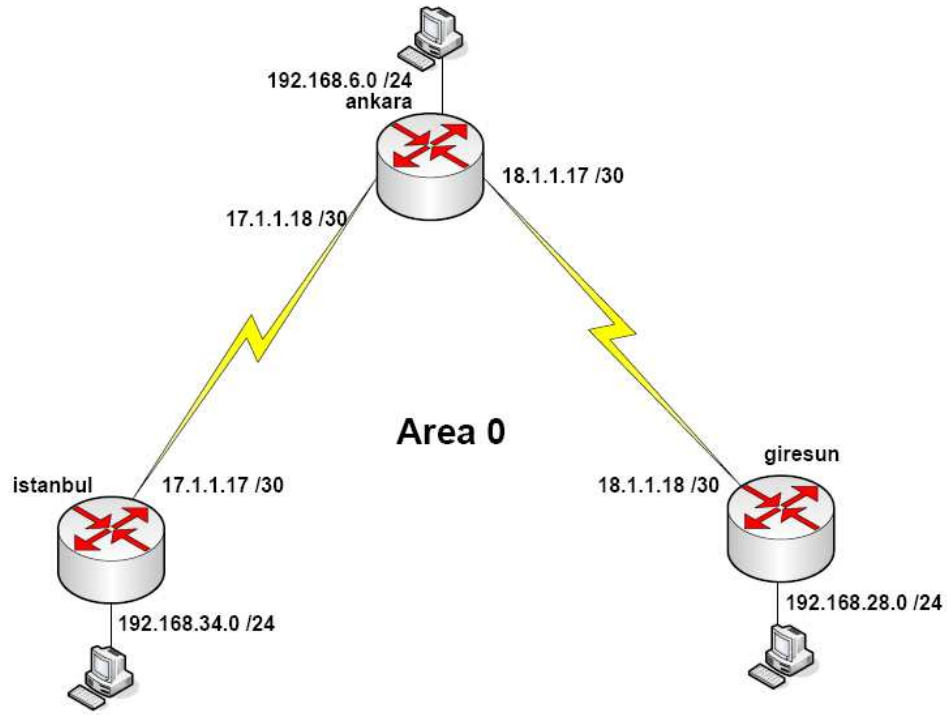
(C Router' ının Routing Table' ı)

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

Gateway of last resort is not set

     172.17.0.0/24 is subnetted, 1 subnets
C       172.17.1.0 is directly connected, Ethernet0/0
     172.16.0.0/24 is subnetted, 1 subnets
O       172.16.1.0 [110/138] via 11.1.1.1, 00:43:54, Serial0/0
     10.0.0.0/24 is subnetted, 1 subnets
O       10.1.1.0 [110/128] via 11.1.1.1, 00:43:54, Serial0/0
     11.0.0.0/24 is subnetted, 1 subnets
C       11.1.1.0 is directly connected, Serial0/0
C#_
```


(Hello Paketleri)

```
C#debug ip ospf events
OSPF events debugging is on
C#
01:15:44: OSPF: Rcv pkt from 172.16.1.1, Ethernet0/0, area 0.0.0.0 : src not on
the same network
01:15:48: OSPF: Rcv hello from 11.1.1.1 area 0 from Serial0/0 11.1.1.1
01:15:48: OSPF: End of hello processing
01:15:54: OSPF: Rcv pkt from 172.16.1.1, Ethernet0/0, area 0.0.0.0 : src not on
the same network
01:15:58: OSPF: Rcv hello from 11.1.1.1 area 0 from Serial0/0 11.1.1.1
01:15:58: OSPF: End of hello processing
01:16:04: OSPF: Rcv pkt from 172.16.1.1, Ethernet0/0, area 0.0.0.0 : src not on
the same network
01:16:08: OSPF: Rcv hello from 11.1.1.1 area 0 from Serial0/0 11.1.1.1
01:16:08: OSPF: End of hello processing
01:16:14: OSPF: Rcv pkt from 172.16.1.1, Ethernet0/0, area 0.0.0.0 : src not on
the same network
01:16:18: OSPF: Rcv hello from 11.1.1.1 area 0 from Serial0/0 11.1.1.1
01:16:18: OSPF: End of hello processing
01:16:24: OSPF: Rcv pkt from 172.16.1.1, Ethernet0/0, area 0.0.0.0 : src not on
the same network
```

OSPF Laboratuar Çalışmaları




```
!
router ospf 101
 network 17.1.1.16 0.0.0.3 area 0
 network 192.168.34.0 0.0.0.255 area 0
!
ip classless
no ip http server
!
!
!
line con 0
 transport input none
line aux 0
line vty 0 4
!
no scheduler allocate
end

istanbul# sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route

Gateway of last resort is not set

C    192.168.34.0 is directly connected, Ethernet0
     17.0.0.0/30 is subnetted, 1 subnets
C       17.1.1.16 is directly connected, Serial1
O    192.168.6.0 [110/64] via 17.1.1.18, 00:03:10, Serial1
     18.0.0.0/30 is subnetted, 1 subnets
O       18.1.1.16 [110/64] via 18.1.1.17, 00:03:00, Serial1
O    192.168.28.0 [110/192] via 17.1.1.18, 00:03:31, Serial1

istanbul#
```

```
!
router ospf 101
 network 192.168.6.0 0.0.0.255 area 0
 network 17.1.1.16 0.0.0.3 area 0
 network 18.1.1.16 0.0.0.3 area 0
!
ip classless
no ip http server
!
!
!
line con 0
 transport input none
line aux 0
line vty 0 4
!
no scheduler allocate
end

ankara# sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route

Gateway of last resort is not set

     17.0.0.0/30 is subnetted, 1 subnets
C       17.1.1.16 is directly connected, Serial1
C    192.168.6.0 is directly connected, Ethernet0
     18.0.0.0/30 is subnetted, 1 subnets
C       18.1.1.16 is directly connected, Serial0
O    192.168.34.0 [110/64] via 17.1.1.17, 00:06:50, Serial1
O    192.168.28.0 [110/64] via 18.1.1.18, 00:05:11, Serial0

ankara#
```

```
!
router ospf 101
 network 192.168.28.0 0.0.0.255 area 0
 network 18.1.1.16 0.0.0.3 area 0
!
ip classless
no ip http server
!
!
!
line con 0
 transport input none
line aux 0
line vty 0 4
!
no scheduler allocate
end

giresun# sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route

Gateway of last resort is not set

     18.0.0.0/30 is subnetted, 1 subnets
C       18.1.1.16 is directly connected, Serial1
C    192.168.28.0 is directly connected, Ethernet0
     17.0.0.0/30 is subnetted, 1 subnets
O       17.1.1.16 [110/128] via 18.1.1.17, 00:06:00, Serial1
O    192.168.6.0 [110/64] via 18.1.1.17, 00:06:00, Serial1
O    192.168.34.0 [110/192] via 18.1.1.17, 00:04:00, Serial1

giresun#


istanbul#ping 192.168.6.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.6.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
istanbul#ping 192.168.28.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.28.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
istanbul#


istanbul#sh ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.6.1       1     FULL/           00:16:30    17.1.1.18         Serial1

istanbul#
```

```
ankara#sh ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.34.1      1     FULL/           00:16:03    17.1.1.17         Serial1
192.168.28.1      1     FULL/           00:16:03    18.1.1.18         Serial0

ankara#
```

```
giresun#
giresun#sh ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.6.1       1     FULL/           00:17:01    18.1.1.17         Serial1

giresun#
```

```
giresun#show ip ospf interface
Serial1 is up, line protocol is up
  Internet Address 18.1.1.18/30 , Area 0
  Process ID 101, Router ID 192.168.28.1, Network Type , Cost: 64
  Transmit Delay is 1 sec, State , Priority 1
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
  Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 18.1.1.17
  Suppress hello for 0 neighbor(s)
Ethernet0 is up, line protocol is up
  Internet Address 192.168.28.1/24 , Area 0
  Process ID 101, Router ID 192.168.28.1, Network Type , Cost: 10
  Transmit Delay is 1 sec, State , Priority 1
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)

giresun#
```

OSPF Özet

Link State bir protokodur.

Hizli yayilma özelliğine sahiptpr.

VLSM (Variable Length subnet Mask) ve CIDR (Classless Inter Domain Routing) destegi vardır.

Metric hesabi tamamen bant genişliği genişliği üzerine kuruludur.

Distance Vector protokollerin aksine periyodik updateler yapmaz, gerektiginde yani networkte değişiklik olduğu zaman update yapar.

Area 0 Backbone area olarak adlandirili ve diğer bütün arealar ancak area 0 üzerinde birbirleriyle konusabilirler.

Komşu Routerlarına 10 saniye aralıklar ile gönderdiği Hello paketleri ile komşuluk ilişkilerini başlatır devam ettirir. Non-Broadcast Multi Access (NBMA) networklerde 30 saniyedir.

Dead Interval Hello Interval, in 4 katidir. Routerların komşuluk ilişkisi kurabilmeleri için Hello ve Dead Intervallarının aynı olması gerekir. Hello ve Dead Interval aralıkları değiştirilebilir.

```
Rtr(config-if)# ip ospf hello-interval seconds
Rtr(config-if)# ip ospf dead-interval seconds
```

Broadcast Multi Access ve Non-Broadcast Multi Access networklerde bütün trafiği DR denen router yönetir, BDR ile yedeklenmistir. Bu networklerde

Routerlar sadece birbirlerine Hello paketleri gönderirken diğer bütün paketler DR üzerinden gerçeklesir.

Konfigurasyonu oldukca basittir.

```
Rtr(config)# router ospf process-id
Rtr(config-router)#network address wildcard-mask area area-id
```

Aşağıdaki show komutları ile olaylar görüntülenebilir.

```
Router# show ip route
Router# show ip ospf
Router# show ip ospf interface
Router# show ip ospf neighbor
Router# show ip ospf database
Router# debug ip ospf adj
Router# debug ip ospf events
```

Router# debug ip ospf adj

04:19:46: OSPF: Rcv hello from 201.0.0.1 area 0 from

FastEthernet0 192.168.20.1

04:19:46: OSPF: 2 Way Communication to 201.0.0.1 on

FastEthernet0, **state 2WAY**

04:19:46: OSPF: End of hello processing

04:20:22: OSPF: end of Wait on interface FastEthernet0

04:20:22: OSPF: DR/BDR election on FastEthernet0

04:20:22: OSPF: Elect BDR 200.0.0.1

04:20:22: OSPF: Elect DR 200.0.0.1

04:20:22: OSPF: Elect BDR 201.0.0.1

04:20:22: OSPF: Elect DR 200.0.0.1

04:20:22: DR: 201.0.0.1 (Id) BDR: 200.0.0.1 (Id)

04:20:23: OSPF: Rcv DBD from 201.0.0.1 on FastEthernet0 seq

0x2657 opt 0x2 flağ

0x7 len 32 mtu 1500 **state EXSTART**

04:20:23: OSPF: NBR Negotiation Done. We are the SLAVE

04:20:23: OSPF: Send DBD to 201.0.0.1 on FastEthernet0 seq

0x2657 opt 0x2 flağ 0 x2 len 92

04:20:23: OSPF: Rcv DBD from 201.0.0.1 on FastEthernet0 seq

0x2658 opt 0x2 flağ

0x3 len 72 mtu 1500 **state EXCHANGE**

<text omitted>

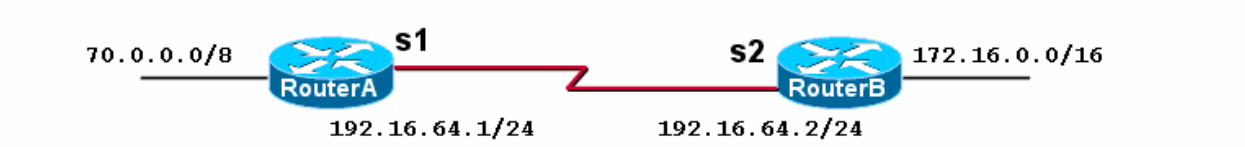
04:20:23: OSPF: Synchronized with 201.0.0.1 on FastEthernet0,

state FULL

Extralar

Authentication Konfigurasyonu yapılabilir.

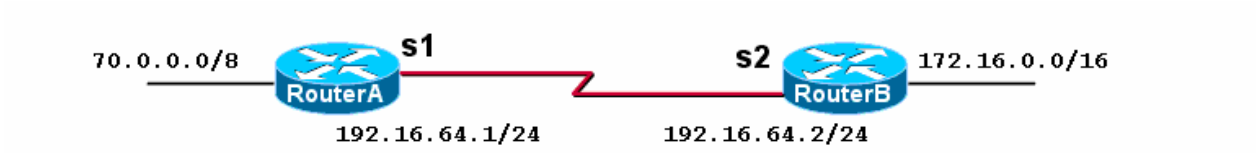
(Basit)




```
RouterA
interface Serial1
  ip address 192.16.64.1 255.255.255.0
  ip ospf authentication-key secret
  !
router ospf 10
  network 192.16.64.0 0.0.0.255 area 0
  network 70.0.0.0 0.255.255.255 area 0
  area 0 authentication
```

```
RouterB
interface Serial2
 ip address 192.16.64.2 255.255.255.0
 ip ospf authentication-key secret
!
router ospf 10
 network 172.16.0.0 0.0.255.255 area 0
 network 192.16.64.0 0.0.0.255 area 0
 area 0 authentication
```

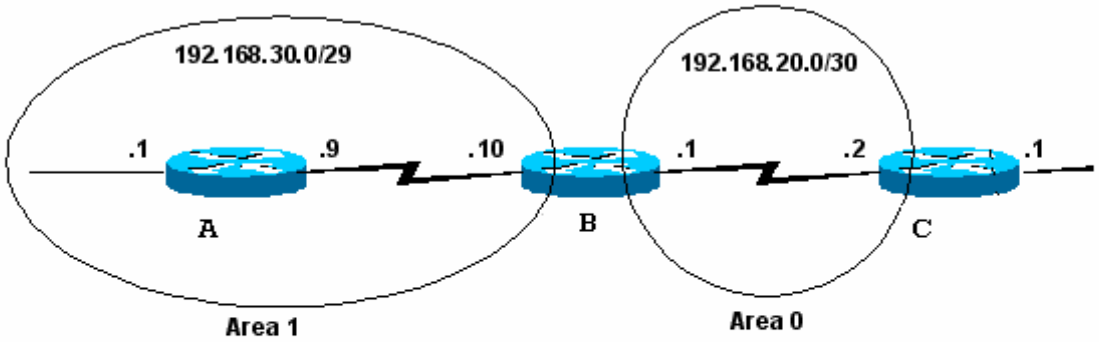
(MD5)




```
RouterA
interface Serial1
  ip address 192.16.64.1 255.255.255.0
  ip ospf message-digest-key 1 md5 secret
  !
router ospf 10
  network 192.16.64.0 0.0.0.255 area 0
  network 70.0.0.0 0.255.255.255 area 0
  area 0 authentication message-digest
```

```
RouterB
interface Serial2
 ip address 192.16.64.2 255.255.255.0
 ip ospf message-digest-key 1 md5 secret
!
router ospf 10
 network 172.16.0.0 0.0.255.255 area 0
 network 192.16.64.0 0.0.0.255 area 0
 area 0 authentication message-digest
```

Iki farklı Area Area Borde Router (ABR) denen Routerlar ile haberlesebilirler.



B

```
router ospf 20
   network 192.168.30.0 0.0.0.255 area 1
   network 192.168.20.0 0.0.0.255 area 0
```

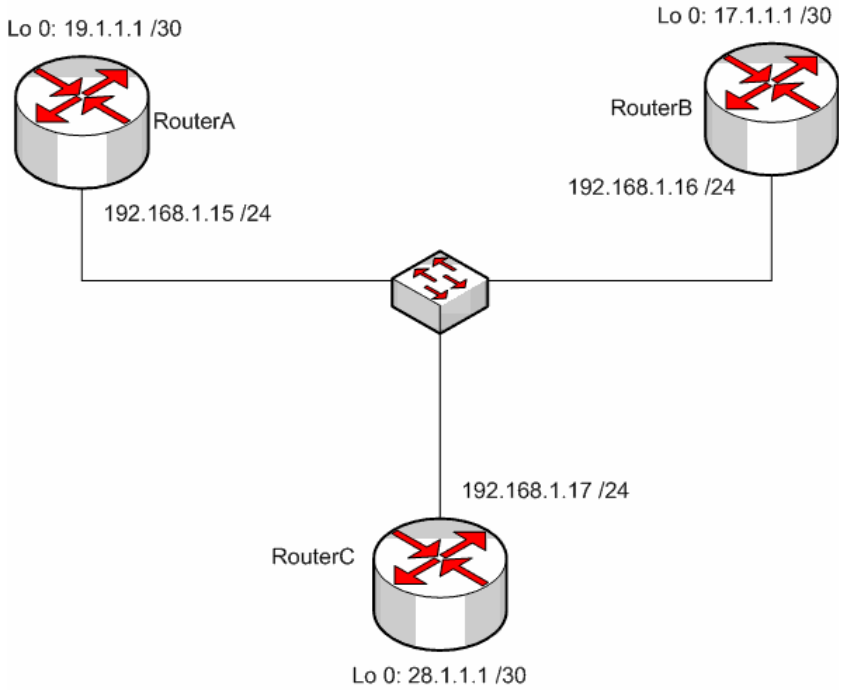
Burada B Routeri Area Border Router dir ve interfacelerinden biri area 0' da bir diğeri area 1' dedir.

Default Route yapılabilir.

Bunun için static default route OSPF konfigurasyonu içine gomulmelidir.

```
Router(config)# ip route 0.0.0.0 0.0.0.0 serial0

Router(config)# router ospf 1
Router(config-router)# default-information originate
```

OSPF DR-BDR Seçimi Lab. ÇALIŞMASI



Bu çalışma içerişinde DR ve BDR seçimlerinin anlaşılmasi amaclanmistir. Laboratuar imkanlirinin elverdiği olcude tasarlanan senaryo daher router aynı Ethernet networkune bağlanmış ve her Router üzerinde Loopback adresleri tanımlanmıştır.

Routerlarda OSPF konfigurasyonu yapılirken Loopback betworklerde tanitilmistir.

Konfigurasyon ve convergence tamamlandiktan sonra Routing Table' lar sağidaki gibi oluşmutur.

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

Gateway of last resort is not set

     17.0.0.0/32 is subnetted, 1 subnets
O       17.1.1.1 [110/11] via 192.168.1.16, 00:00:11, Ethernet0/0
     19.0.0.0/30 is subnetted, 1 subnets
C       19.1.1.0 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, Ethernet0/0
     28.0.0.0/32 is subnetted, 1 subnets
O       28.1.1.1 [110/11] via 192.168.1.17, 00:00:11, Ethernet0/0
RouterA#
```

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

Gateway of last resort is not set

     17.0.0.0/30 is subnetted, 1 subnets
C       17.1.1.0 is directly connected, Loopback0
     19.0.0.0/32 is subnetted, 1 subnets
O       19.1.1.1 [110/11] via 192.168.1.15, 00:02:05, Ethernet0/0
C    192.168.1.0/24 is directly connected, Ethernet0/0
     28.0.0.0/32 is subnetted, 1 subnets
O       28.1.1.1 [110/11] via 192.168.1.17, 00:02:05, Ethernet0/0
RouterB#
```

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

Gateway of last resort is not set

     17.0.0.0/32 is subnetted, 1 subnets
O       17.1.1.1 [110/11] via 192.168.1.16, 00:03:01, Ethernet0/0
     19.0.0.0/32 is subnetted, 1 subnets
O       19.1.1.1 [110/11] via 192.168.1.15, 00:03:01, Ethernet0/0
C    192.168.1.0/24 is directly connected, Ethernet0/0
     28.0.0.0/30 is subnetted, 1 subnets
C       28.1.1.0 is directly connected, Loopback0
RouterC#_
```

Routing Table' ların ardindan OSPF database'i ve Ospf komşuları incelenmistir. Bu incelemede DR ve BDR' lar detaylı görulebilmektedir.

```
RouterA#sh ip ospf database

            OSPF Router with ID (19.1.1.1) (Process ID 123)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
17.1.1.1        17.1.1.1        321         0x80000002 0x0043A2 2
19.1.1.1        19.1.1.1        255         0x80000004 0x0007D7 2
28.1.1.1        28.1.1.1        321         0x80000003 0x00B50D 2

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.1.17    28.1.1.1        257         0x80000002 0x003819
RouterA#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
17.1.1.1          1   FULL/BDR        00:00:34    192.168.1.16    Ethernet0/0
28.1.1.1          1   FULL/DR         00:00:33    192.168.1.17    Ethernet0/0
RouterA#
```

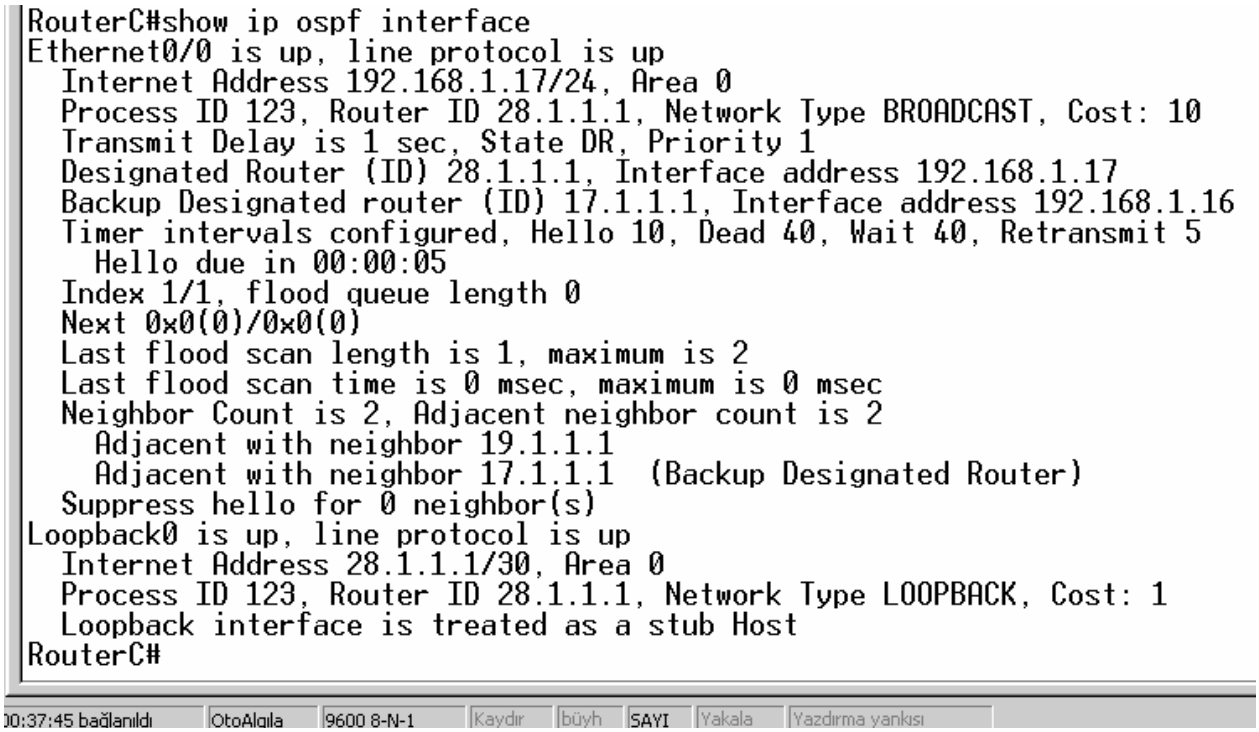
RouterA' dan alınan bu görüntude komşu routerlar ve bu routerlar ile olan ilişki tespit edilebilmektedir. Örneğin 28.1.1.1 ID' sine sahip Router ile Full komşuluk ilişkisi kurulmus ve DR olarak kabul edilmiştir. (Bunun Böyle olacağını zaten biliyorduk zira 28.1.1.1 ortamdaki en yüksek ID)

```
RouterB#show ip ospf database

            OSPF Router with ID (17.1.1.1) (Process ID 123)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
17.1.1.1        17.1.1.1        383         0x80000002 0x0043A2 2
19.1.1.1        19.1.1.1        318         0x80000004 0x0007D7 2
28.1.1.1        28.1.1.1        383         0x80000003 0x00B50D 2

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.1.17    28.1.1.1        319         0x80000002 0x003819
RouterB#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
19.1.1.1          1   FULL/DROTHER    00:00:36    192.168.1.15    Ethernet0/0
28.1.1.1          1   FULL/DR         00:00:37    192.168.1.17    Ethernet0/0
RouterB#_
```

```
RouterC#show ip ospf database

       OSPF Router with ID (28.1.1.1) (Process ID 123)


                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
17.1.1.1        17.1.1.1        435         0x80000002 0x43A2   2
19.1.1.1        19.1.1.1        370         0x80000004 0x7D7    2
28.1.1.1        28.1.1.1        435         0x80000003 0xB50D   2

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.1.17    28.1.1.1        370         0x80000002 0x3819
RouterC#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
19.1.1.1          1   FULL/DROTHER    00:00:31    192.168.1.15    Ethernet0/0
17.1.1.1          1   FULL/BDR        00:00:32    192.168.1.16    Ethernet0/0
RouterC#
```

```
RouterA#sh ip ospf interface
Loopback0 is up, line protocol is up
  Internet Address 19.1.1.1/30, Area 0
  Process ID 123, Router ID 19.1.1.1, Network Type LOOPBACK, Cost: 1
  Loopback interface is treated as a stub Host
Ethernet0/0 is up, line protocol is up
  Internet Address 192.168.1.15/24, Area 0
  Process ID 123, Router ID 19.1.1.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DROTHER, Priority 1
  Designated Router (ID) 28.1.1.1, Interface address 192.168.1.17
  Backup Designated router (ID) 17.1.1.1, Interface address 192.168.1.16
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:05
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 1
  Last flood scan time is 4 msec, maximum is 4 msec
  Neighbor Count is 2, Adjacent neighbor count is 2
    Adjacent with neighbor 17.1.1.1  (Backup Designated Router)
    Adjacent with neighbor 28.1.1.1  (Designated Router)
  Suppress hello for 0 neighbor(s)
RouterA#
```

```
RouterB#sh ip ospf interface
Loopback0 is up, line protocol is up
  Internet Address 17.1.1.1/30, Area 0
  Process ID 123, Router ID 17.1.1.1, Network Type LOOPBACK, Cost: 1
  Loopback interface is treated as a stub Host
Ethernet0/0 is up, line protocol is up
  Internet Address 192.168.1.16/24, Area 0
  Process ID 123, Router ID 17.1.1.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 28.1.1.1, Interface address 192.168.1.17
  Backup Designated router (ID) 17.1.1.1, Interface address 192.168.1.16
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:07
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 2, Adjacent neighbor count is 2
    Adjacent with neighbor 19.1.1.1
    Adjacent with neighbor 28.1.1.1  (Designated Router)
  Suppress hello for 0 neighbor(s)
RouterB#_
```



Show ip ospf interface komutu ile aldığımiz görüntülere baktigimiz da 17.1.1.1 ID' li routerin BDR seçildiğini oysa daha yüksek ID' ye sahip 19.1.1.1 ID' li routerin DROther olarak kaldığını göruyoruz ki bu karmasik bir durum.

OSPF konuşan routerlar ortama daha yüksek ID' ye sahip bir Router katildiğinda onu Drother olarak alirlar, yeniden bir DR – BDR seçimine gitmezler. Bu Cisco' nun bir bug' idir. Anlaşılan o ki örneğimiz de 17.1.1.1 ID'li router ortama daha önce katilmis ve 19.1.1.1 ID' li router up olmadan BDR secmimi tamamlanmis.

Cisco' nun bu bug' ini asmak için interfacelerden en azından birini down – up yapmamız gerekecek.

```
Router#debug ip ospf adj
00:50:38: OSPF: DR/BDR election on Ethernet0/0
00:50:38: OSPF: Elect DR 28.1.1.1
00:50:38: OSPF: Elect BDR 0.0.0.0
00:50:38:       DR: 28.1.1.1 (Id)   BDR: none
00:50:38: OSPF: Remember old DR 17.1.1.1 (id)
00:50:39: OSPF: Reset old DR on Ethernet0/0
00:50:39: OSPF: Build router LSA for area 0, router ID 28.1.1.1, seq 0x80000010
00:50:39: OSPF: No full nbrs to build Net Lsa for interface Ethernet0/0
00:50:46: OSPF: 2 Way Communication to 19.1.1.1 on Ethernet0/0, state 2WAY
00:50:46: OSPF: Neighbor change Event on interface Ethernet0/0
00:50:46: OSPF: DR/BDR election on Ethernet0/0
00:50:46: OSPF: Elect BDR 19.1.1.1
00:50:46: OSPF: Elect DR 28.1.1.1
00:50:46:       DR: 28.1.1.1 (Id)   BDR: 19.1.1.1 (Id)
00:50:46: OSPF: Send DBD to 19.1.1.1 on Ethernet0/0 seq 0x1692 opt 0x42 flag 0x7 len 32
00:50:46: OSPF: Rcv DBD from 19.1.1.1 on Ethernet0/0 seq 0x1692 opt 0x52 flag 0x2 len
112  mtu 1500 state EXSTART
00:50:46: OSPF: NBR Negotiation Done. We are the MASTER
00:50:46: OSPF: Send DBD to 19.1.1.1 on Ethernet0/0 seq 0x1693 opt 0x42 flag 0x3 len 112
00:50:46: OSPF: Database request to 19.1.1.1
00:50:46: OSPF: sent LS REQ packet to 192.168.1.15, length 12
00:50:46: OSPF: Rcv DBD from 19.1.1.1 on Ethernet0/0 seq 0x1693 opt 0x52 flag 0x0 len 32
mtu 1500 state EXCHANGE
00:50:46: OSPF: Send DBD to 19.1.1.1 on Ethernet0/0 seq 0x1694 opt 0x42 flag 0x1 len 32
00:50:46: OSPF: Rcv DBD from 19.1.1.1 on Ethernet0/0 seq 0x1694 opt 0x52 flag 0x0 len 32
mtu 1500 state EXCHANGE
00:50:46: OSPF: Exchange Done with 19.1.1.1 on Ethernet0/0
00:50:46: OSPF: Synchronized with 19.1.1.1 on Ethernet0/0, state FULL
00:50:46: %OSPF-5-ADJCHG: Process 123, Nbr 19.1.1.1 on Ethernet0/0 from LOADING to
FULL, Loading Done
00:50:46: OSPF: Build router LSA for area 0, router ID 28.1.1.1, seq 0x80000011


00:50:46: OSPF: Build network LSA for Ethernet0/0, router ID 28.1.1.1
00:50:48: OSPF: Rcv DBD from 17.1.1.1 on Ethernet0/0 seq 0x1C03 opt 0x52 flag 0x7 len
32  mtu 1500 state INIT
00:50:48: OSPF: 2 Way Communication to 17.1.1.1 on Ethernet0/0, state 2WAY
00:50:48: OSPF: Neighbor change Event on interface Ethernet0/0
00:50:48: OSPF: DR/BDR election on Ethernet0/0
00:50:48: OSPF: Elect BDR 19.1.1.1
00:50:48: OSPF: Elect DR 28.1.1.1
00:50:48:       DR: 28.1.1.1 (Id)   BDR: 19.1.1.1 (Id)
00:50:48: OSPF: Send DBD to 17.1.1.1 on Ethernet0/0 seq 0xB19 opt 0x42 flag 0x7 len 32
```

```
00:50:48: OSPF: First DBD and we are not SLAVE
00:50:48: OSPF: Rcv DBD from 17.1.1.1 on Ethernet0/0 seq 0xB19 opt 0x52 flag 0x2 len 92
mtu 1500 state EXSTART
00:50:48: OSPF: NBR Negotiation Done. We are the MASTER
00:50:48: OSPF: Send DBD to 17.1.1.1 on Ethernet0/0 seq 0xB1A opt 0x42 flag 0x3 len 132
00:50:48: OSPF: Database request to 17.1.1.1
00:50:48: OSPF: sent LS REQ packet to 192.168.1.16, length 12
00:50:48: OSPF: Rcv DBD from 17.1.1.1 on Ethernet0/0 seq 0xB1A opt 0x52 flag 0x0 len 32
mtu 1500 state EXCHANGE
00:50:48: OSPF: Send DBD to 17.1.1.1 on Ethernet0/0 seq 0xB1B opt 0x42 flag 0x1 len 32
00:50:48: OSPF: Rcv DBD from 17.1.1.1 on Ethernet0/0 seq 0xB1B opt 0x52 flag 0x0 len 32
mtu 1500 state EXCHANGE
00:50:48: OSPF: Exchange Done with 17.1.1.1 on Ethernet0/0
00:50:48: OSPF: Synchronized with 17.1.1.1 on Ethernet0/0, state FULL
00:50:48: %OSPF-5-ADJCHG: Process 123, Nbr 17.1.1.1 on Ethernet0/0 from LOADING to
FULL, Loading Done
00:50:48: OSPF: Neighbor change Event on interface Ethernet0/0
00:50:48: OSPF: DR/BDR election on Ethernet0/0
00:50:48: OSPF: Elect BDR 19.1.1.1
00:50:48: OSPF: Elect DR 28.1.1.1
00:50:48:       DR: 28.1.1.1 (Id)   BDR: 19.1.1.1 (Id)
00:50:52: OSPF: Build network LSA for Ethernet0/0, router ID 28.1.1.1
RouterC(config)#
```

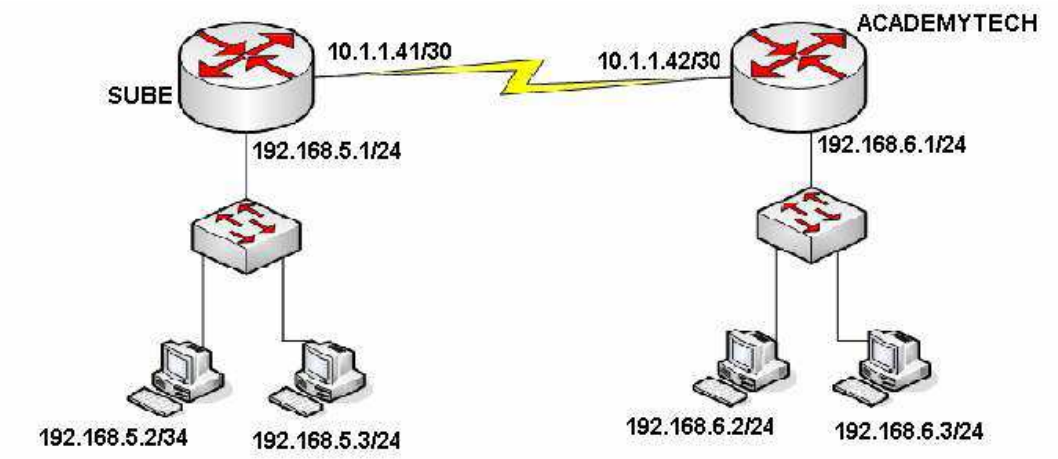
ROUTİNG PROTOKOLLERE GENEL BAKIS

Bu bölümde Routing Protokolleri genel olarak inceleyeceğiz. Routing Protokolleri genel olarak üç grup halinde inceleyebiliriz.

Distance Vector Protokoller	Link State Protokoller	Hybrid Protokoller
Rip	OSPF	EIGRP
Ripv2		
IGRP		

Bütün Routing Protokollerin anlatımı sırasında hep söylediğimiz gibi, Routing Protokoller update mantığıyla daha açık bir ifadeyle sahip oldukları veritabanlarını (Routing Table) paylasarak çalışırlar.

Biz sadece Routerlarımızın kendilerine direk bağlı olan networkleri protokoller vasıtasıyla tanıtırız. Protokol cinsine göre, belli zaman aralıklarından Routerlar arasında veritabanı paylaşımı gerçeklesir ve bir süre sonra bütün Routerlar sistemdeki bütün networkleri öğrenmiş olarak Routing Table' larını son haliyle oluştururlar.



Routing Protokollerin karşılaştırılması sırasında şekildeki topolojiden hareketle konfigürasyonlar yapılacak ve karşılaştırmalar gerçeklestirilecektir.

Routerlar en iyi yol seçimi yaparken (Best Path Determination) referans olarak Routing Table' larında ki bilgileri alırlar. Dolayısıyla Routing Protokoller kullanarak oluşturulan Routing Table' ların sistem başladıktan belirli bir zaman sonra son halini alacak olması bir dezavantaj olarak görülebilir. Bunun yanında networklerin giderek büyüdükleri göz önüne alınırsa bir kez Routing Protokoller ile konfigüre ettigimiz Routerlar ileride eklenecek networkleri biz müdahale etmeden öğrenebileceklerdir ki buda önemli avantajlarındandır.

Burada Update sürelerini baz alarak Routing Protokolleri karşılaştırabiliriz.

RIP	RIPv2	IGRP	EIGRP	OSPF
30 sn.	30 sn.	90 sn.	Gerektiğinde	Gerektiğinde

Rip, Ripv2 ve IGRP' de updateler belirli zaman aralıklarında yapılırken, EIGRP ve OSPF için update gerektiginde yani sistem üzerinde bir değişiklik olduğunda, yeni bir network eklendiğinde veya bir network down olduğunda yapılır. Ve burada yine aklımızda tutmamız gereken konu EIGRP ve OSPF gerektiginde yaptıgı updatelerde sadece değişen durum ile ilgili bilgi gönderirken diğerleri tüm Routing Table' larını her seferinde gönderirler. Bunun yanında EIGRP 5 ve OSPF 10 saniye aralıklarla komşu routerlarının up olup olmadıklarını kontrol etmek için küçük paketler gönderirler fakat bunlar hattı çok az mesgul ederler.

Burada Routing Protokollerin update yaparken kullandıkları broadcast ya da multicast adreslerde karşılaştırılabilir. Hatırlayacağınız gibi "debug" komutunu kullanarak protokollerin aldıkları veya gönderdikleri paketleri izleyebiliyorduk.

```
ACADEMYTECH#debug ip rip
RIP protocol debugging is on
ACADEMYTECH#
00:29:02: RIP: sending v1 update to 255.255.255.255 via Ethernet0/0 (192.168.6.1
)
00:29:02:      network 192.168.5.0, metric 2
00:29:02:      network 10.0.0.0, metric 1
00:29:02: RIP: sending v1 update to 255.255.255.255 via Serial0/0 (10.1.1.42)
00:29:02:      network 192.168.6.0, metric 1
```

```
ACADEMYTECH#debug ip igrp events
IGRP event debugging is on
ACADEMYTECH#
00:33:35: IGRP: sending update to 255.255.255.255 via Ethernet0/0 (192.168.6.1)
00:33:35: IGRP: Update contains 0 interior, 2 system, and 0 exterior routes.
00:33:35: IGRP: Total routes in update: 2
00:33:35: IGRP: sending update to 255.255.255.255 via Serial0/0 (10.1.1.42)
00:33:35: IGRP: Update contains 0 interior, 1 system, and 0 exterior routes.
00:33:35: IGRP: Total routes in update: 1
```

```
ACADEMYTECH#debug ip ospf events
OSPF events debugging is on
ACADEMYTECH#
00:43:30: OSPF: Rcv hello from 192.168.5.1 area 0 from Serial0/0 10.1.1.41
00:43:30: OSPF: End of hello processing
00:43:34: OSPF: Rcv pkt from 192.168.5.1, Ethernet0/0, area 0.0.0.0 : src not on
 the same network
00:43:40: OSPF: Rcv hello from 192.168.5.1 area 0 from Serial0/0 10.1.1.41
00:43:40: OSPF: End of hello processing
```

(OSPF Hello paketleri)

Routing Protokollerden bahsederken bahsettigimiz konulardan biri de bazı protokollerin VLSM (Variable Length subnet Mask) destegi verirken bazılarının vermemsiydi. Bundan kastettigimiz sey protokollerin Classless veya Classfull olmalarıdır. Anladığınız gibi Classfull bir protokolde kullanacağımız network adreslerinde subnet maskı biz belirleyemeyiz, protokol o adresin ait olduğu sınıfa göre subnet maskını kabul eder. Classless protokollerde ise subnet mask tamamen bizim kontrolümüzdedir.

	Rip	Ripv2	IGRP	OSPF	EIGRP
VLSM Desteği	Yok	Var	Yok	Var	Var

```
 no ip directed-broadcast
 no ip mroute-cache
 no fair-queue
!
interface TokenRing0/0
 no ip address
 no ip directed-broadcast
 shutdown
 ring-speed 16
!
interface Serial0/1
 no ip address
 no ip directed-broadcast
 shutdown
!
router ospf 101
 network 10.1.1.40 0.0.0.3 area 0
 network 192.168.6.0 0.0.0.255 area 0
!
ip classless
!
!
line con 0
 --More--
```

(OSPF Runnin-Config, VLSM Destegi)

Burada Rip ve IGRP kullanırken Örneğin 10.1.1.0 /30 gibi bir network tanımlamamız mümkün değildir. Bu protokoller söz konusu adres A sınıfı olduğu için subnet maskı 255.0.0.0 olarak kabul edeceklerdir.

Routing Protokoller metrik hesaplarında farklı kritelere bakarlar. Rip tamamen hop sayısına bakarken IGRP bahsettigimiz K1'den K5'e kadar olan değerlere büyük ölçüde bant genişliğini baz alarak bakar. Tıpkı değişik kriterlere göre metrik hesabı yapıldığı gibi Routing Protokollerin çalışacakları maksimum hop sayıları da farklı farklıdır.

	Rip	Ripv2	IGRP	OSPF	EIGRP
Metric Hesabı	Hop	Hop	K1-K5	Bantwidth	K1, K2
Max. Hop Sayısı	15	15	255	Sınırsız	224

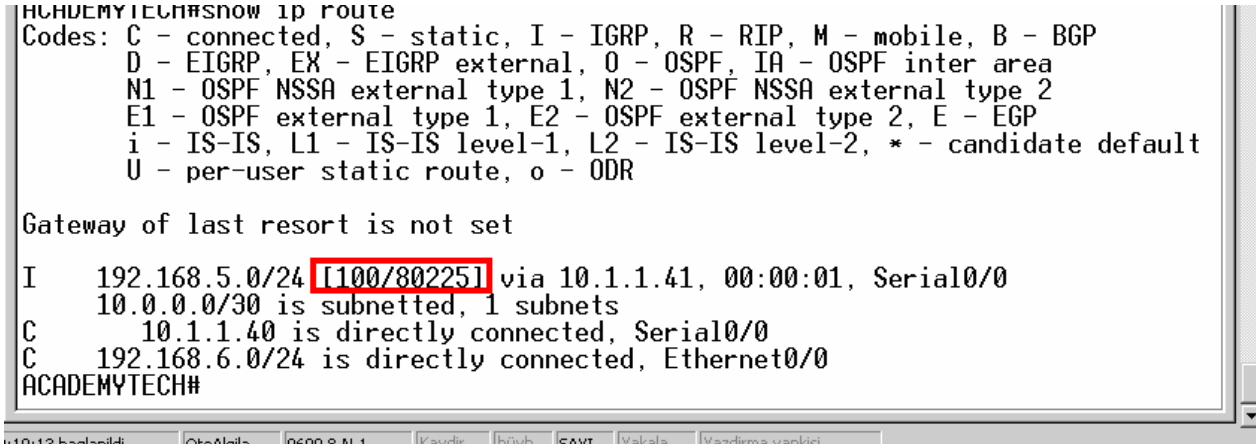
Routing protokoller Autonomous System numaralar kullanıp kullanmadıklar ve bir Area mantığı içine girerek hiyerarsik bir yapı oluşturup oluşturmadıklarına göre de incelenebilirler.

	Rip	Ripv2	IGRP	OSPF	EIGRP
Autonomous System	Yok	Yok	Var	Var	Var
Area	Yok	Yok	Yok	Var	Yok

Routing protokollerde Administrative Distance ve metrik hesaplarında, Routing Table'larına bakıldığında detaylı bilgi sahibi olunabilir.

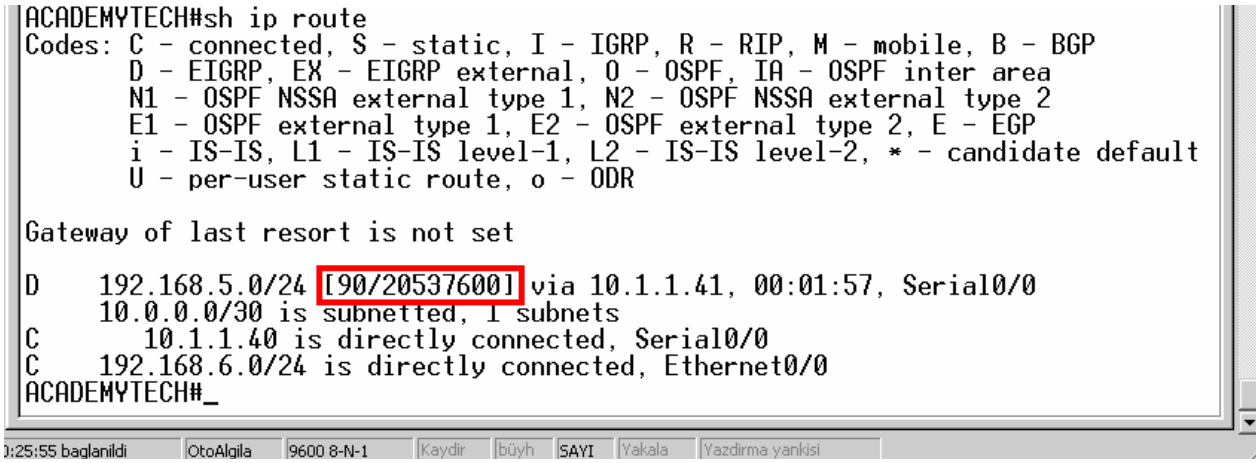
```
ACADEMYTECH#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

R    192.168.5.0/24 [120/1] via 10.1.1.41, 00:00:05, Serial0/0
     10.0.0.0/30 is subnetted, 1 subnets
C       10.1.1.40 is directly connected, Serial0/0
C    192.168.6.0/24 is directly connected, Ethernet0/0
ACADEMYTECH#
```

Rip için Routing Table görüntülendiğinde Administrative Distance' ının 120 olduğu görülmektedir. Parantez içerişindeki bir diğer ifade ("1") metrigi yani Rip için hop sayısını belirtmektedir. Routing Table' da Rip protokolüyle öğrenilen Networkler "R" harfi ile belirtilirler.

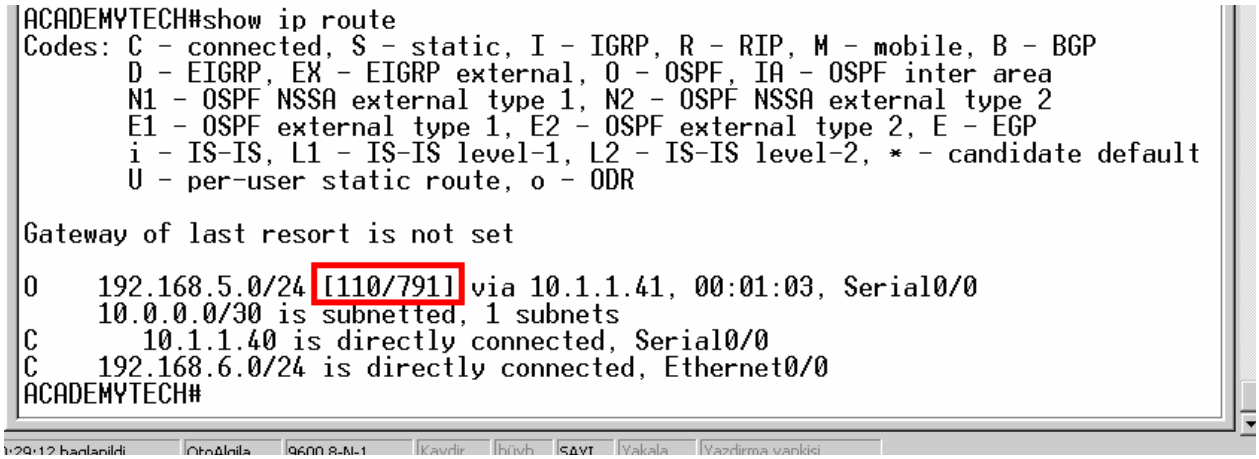


Igrp için Routing Table görüntülendiğinde Administrative Distance' ının 100 olduğu görülmektedir. Parantez içerişindeki bir diğer ifade ("80225") metrigi belirtir. K1 'den K5'e kadar olan kriteler baz alınarak hesaplanmıştır.

OSPF ve EIGRP Routing Table' ları asağıdadır.



(D: EIGRP, Administrative Distance=90)



(O: OSPF, Administrative Distance=110)

CİSCO ÖZEL PROTOKOLLER

IGRP ve EIGRP Cisco özel protokollerdir. Diğer bütün protokoller ise publicdir. Dolayısıyla sistemimizde Cisco dışında üreticilere ait Router' larda varsa IGRP ve EIGRP bizim için doğru seçim olmayacaktır.

IGRP ve EIGRP Cisco tarafindan uretildiklerinden aynı AS içinde birbirleriyle haberlesebilirler. Fakat bu durumda sistemin IGRP konuşan network bilgiler EIGRP konuşan Routerlarda External EIGRP olarak etiketlenir ve bu networkler için Administrative Distance 170' dir.

IPX- APPLETALK DESTEGİ

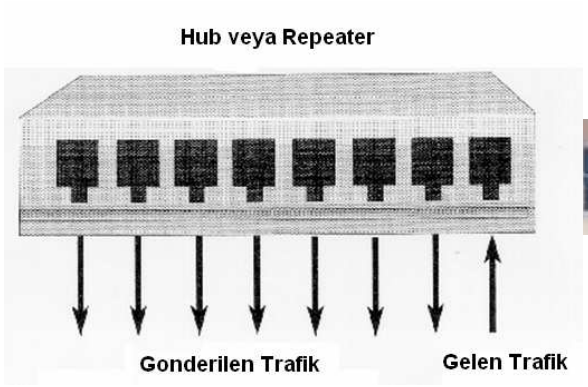
Cisco özel bir protokol olan EIGRP Ip dışında IPX ve AppleTalk networklerini de desteklemesiyle diğer protokollerden ayrılabilir.

Router'da bulunan IPX yönlendirme tablosundaki kayıtları görmek için ise **"show ipx route"** komutu kullanılır. Bunun haricinde Router üzerinde IPX protokolünü izlemek için kullanılabilecek bazı komutlar aşağıdaki tabloda listelenmiştir.

Komut	Açıklama
Show ipx server	Cisco router üzerindeki SAP tablosunun içeriğini gösterir. Netware'deki "display servers" komutuna eşdeğerdir.
Show ipx traffic	Router tarafından alınan ve gönderilen IPX paketlerinin sayısı ve tipi hakkında özet bilgiler gösterir.
Show ipx interfaces	Router interface'lerindeki IPX durumunu, IPX parametrelerini gösterir.
Show protocols	Router interface'lerinin IPX adresini ve frame tipini gösterir.
Debug ipx	ipx konfigürasyon hatalarını belirlemek için kullanılır ve bu komut ile ipx ve sap güncellemelerini gösterir.

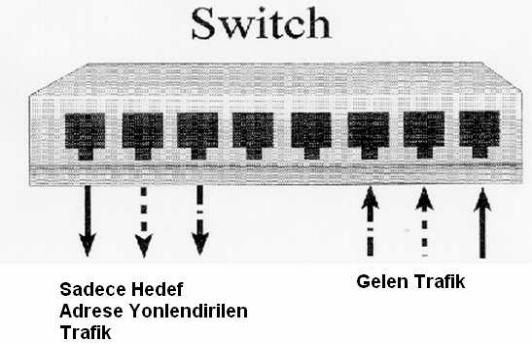
LAYER 2 SWİTCHİNG

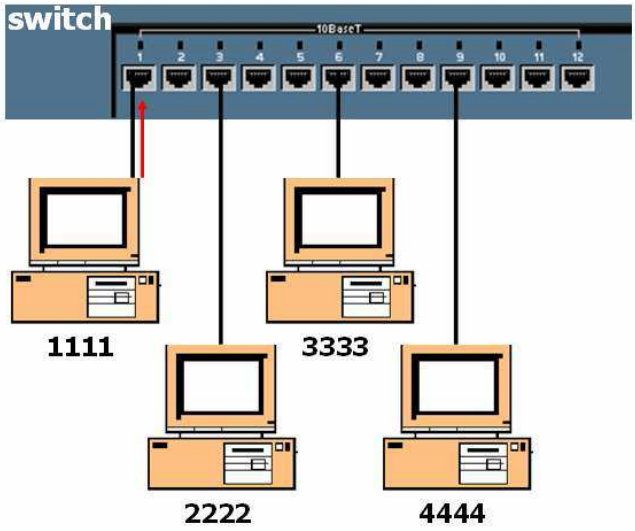
Hub'lar ile çalıştığımızda , hub aslında çok portlu bir repeater olduğu için ağdaki tüm bilgisayarlar aynı çakısma alanı içinde olacaklardır. Bu alana "collision domain" denire. Dolayısıyla network performansında düsme olacaktır.



Bu problemin çözümü olarak networklerde Switch adı verilen cihazlar kullanılmaya başlanmıstır.

Switch OSI 2. katmanda yani Data Link Layer Katmanında çalışır. Bir portuna bağlı bilgisayar veya bilgisayarları gönderdikleri framelerden source MAC adreslerini okuyarak tanır. Bir portundan gelen veri paketini hub'lar gibi tüm portlara dağıtmak yerine sadece veri paketi üzerinde yazan "alıcı MAC adresine" sahip portuna yollar. Paketler direk hedefe gönderildiği için de network üzerinde çarpısmalar (collision) meydana gelmez.





Şekildeki yapı sistemin yeni başlatildiğini varsayarsa. Switch 1111 MAC adresine sahip bilgisayarin gönderdiği frame' I alacak ve buradaki source ma cadres alanından okuduğu değeri ma adresi tablosuna yazacak.

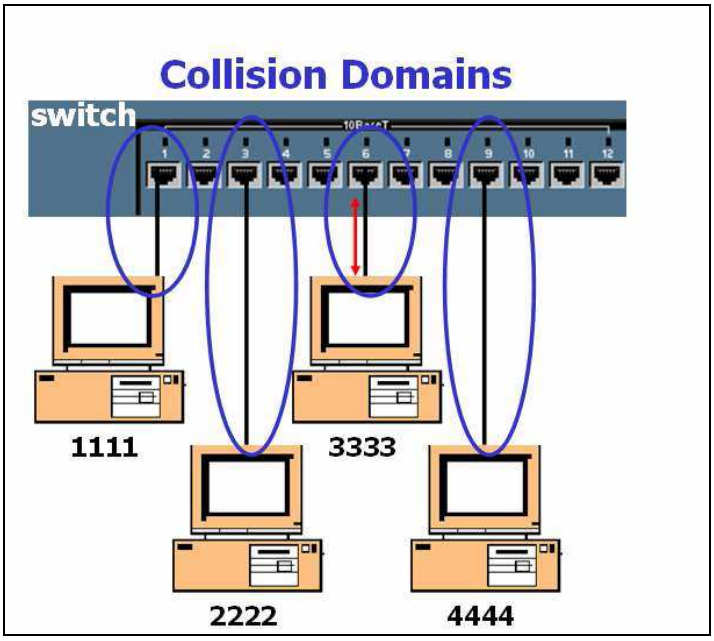
Şu an için MAC adresi tablosunda ki tek girdi 1111 MAC adresi olduğu için hedef mac adresinin hangi portta olduğunu bilmediğimizi soyleyebiliriz. Bu durumda frame bütün portlardan flood edilecektir.

Bu şekilde zamanla switch bütün portlarında ki bilgisayarların mac adreslerini onları cache' in 300 saniye tutmak üzere mac adresi tablosuna yazacaktır. Artık hedef MAC adreslerine göre frameleri yönlendirebilir.

Switche 300 saniye boyunca ilgili MAC adresinin bulunduğu porttan bir istek gelmez ise o adres tablodan silinecektir.

Şimdi sözgelimi 3333 mac adresine sahip bilgisayardan 1111 mac adresine sahip bilgisayar bir istek gönderildiğini varsayalım. Bu durumda switch mac adresi tablosundan 1111 mac adresli bilgisayarin 1. portunda olduğunu group istegi sadece o porta fonderecektir. Bu sayede olası collision' lar engellenmiştir.

İşte bu sebeple Switchin her bir portu bir Collision Domain' dir denilebilir.



SWİTCH KONFİGURASYONU

Switch konfigurayonu bir çok yonde Router ile aynıdir. Switch acildiğinda user moddadir ve ‘enable’ yazılara Enable moda gecilebilir.

Switch>enable

Switch#

Tıpkı Routerda olduğu gibi Switch de yaptigimiz konfigurayonları görüntüleyip sorun cozmemizde bize yardimci olacak show komutları vardır.

```
Switch#show running-config
Building configuration...

Current configuration:
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Switch
!
!<OUTPUT OMITTED>
!
```

```
Switch#show 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
1002 fddi-default        active
1003 token-ring-default active
1004 fddinet-default     active
1005 trnet-default       active

VLAN Type  SAID     MTU   Parent RingNo BridgeNo
---- ----- ------  ----- ------ ------ --------
```

Switch>enable

Switch#conf t

Switch(config)#line con 0

Switch(config-line)#password hayrullah

Switch(config-line)#login

Switch(config-line)#exit

Switch(config)#line vty 0 4

Switch(config-line)#password hayrullah

Switch(config-line)#login

Switch(config-line)#exit

VLAN 1 yönetim VLAN'idir ve switch için verilecek ip adresi bu VLAN'da, default gateway adresi Global Configuration modda verilmelidir. 1900 serişi switchlerde ise durum biraz farklıdır.

Switch(config)#interface vlan 1

Switch(config-if)#ip address 192.168.1.10 255.255.255.0

Switch(config-if)#exit

Switch(config)#ip default-gateway 192.168.1.1

1900 serişi switchlerde ise durum biraz farklıdır. Bu switchlerde ip adresi ve default gateway adresi Global Configuration modda verilir.

Switch(config)#ip adress 192.168.1.10 255.255.255.0

Switch(config)#ip default-gateway 192.168.1.1

Switche web browser ile erişilebilir. Bunun için şu konfigurasyon yapılmalıdır.

```
Switch#configure terminal
Enter configuration commands, one per line.  End
with CNTL/Z.
Switch(config)#ip http ?
  access-class    Restrict access by access-class
  authentication  Set http authentication method
  path            Set base path for HTML
  port            HTTP port
  server          Enable HTTP server
Switch(config)#ip http server
Switch(config)#ip http port ?
  <0-65535>  HTTP port
Switch(config)#ip http port 80
Switch(config)#
```

MAC ADDRESS TABLE

Daha önce de belirttigimiz gibi switchler aldikları framelerdeki source mac adres alanında ki bilgiler ile MAC adres tablolarini oluştururlar ve bu tablolara göre frameleri filtrelerler.

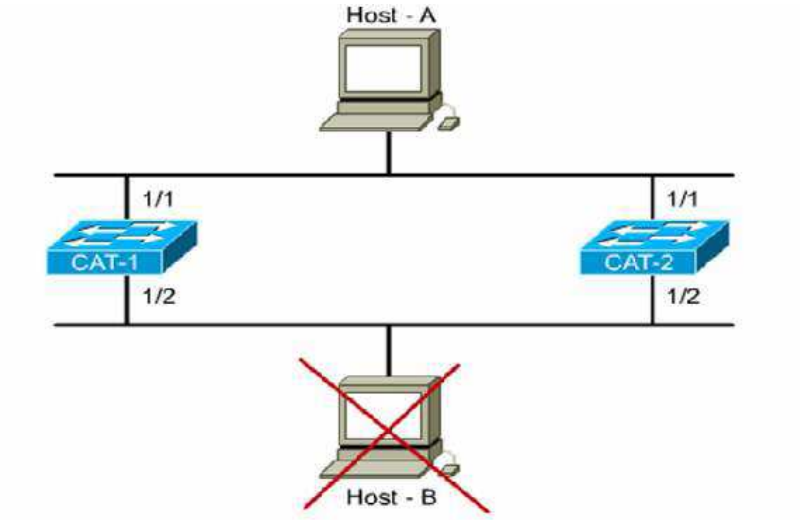
Bununla birlikte switchlere MAC adreslerinin static olarak atanmasi da mümkündür. Güvenligi artirmak için yapılabilecek bu uygulamayla aynı zamanda 300 saniyelik max. Ağe süresi de geçersiz olacaktır.

```
Switch(config)#mac-address-table static
0010.7a60.1884 interface FastEthernet0/5 VLAN1
Switch(config)#no mac-address-table static
0010.7a60.1884 interface FastEthernet0/5 VLAN1
```

SPANNİNG TREE PROTOCOL (STP)

STP Layer 2 cihazların haberlesmi sırasında dogabiliecek olası döngüleri (loop) onleyen bir protokoldur. STP yapısı geregi kullandiği algörltma (Spanning Tree Algörithm) ile döngülere neden olmayacak bir topoloji oluşturur.

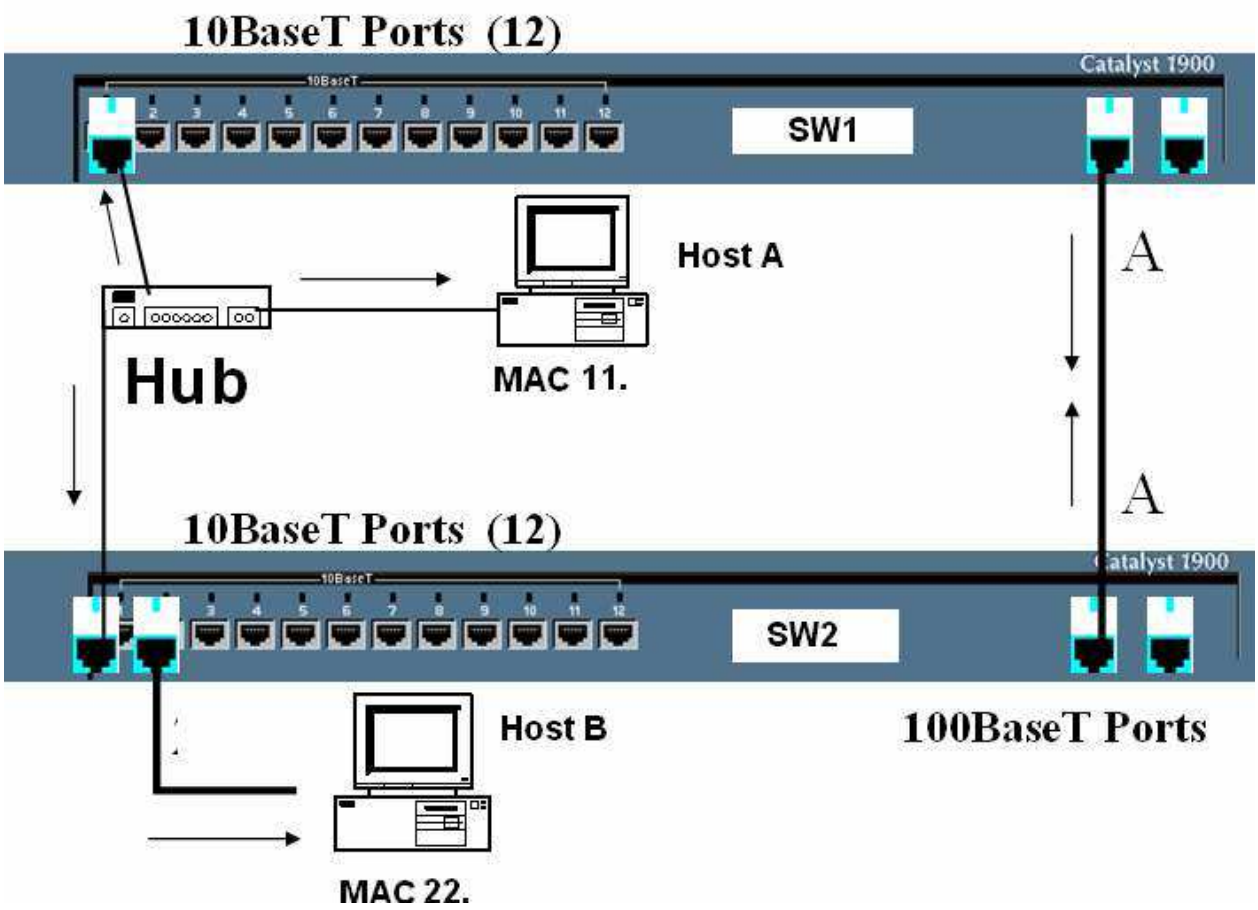
Ethernet Frame' leri TTL alanina sahip olmadıkları için STP ve loop yaratmayacak bir dizayn önemlidir. Aksi takdirde oluşacak döngüler switch kapatılana kadar devam edecektir.



Şekildeki gibi bir yapıda, switchler kendilerine gelen ve hedefi bilinmeyen paketleri diğer bütün portlardan flood edeceklerine göre ciddi sorunlar yasanacaktır. 1/1 portlarından frame' i alan her iki switch de flood edecek ve hemen sonrasında yine her iki switch aynı frame;leri ½ portlarından alacak, devaminda neler olacağını kestirebiliyorsunuzdur sanirim.

Hattları daha da zor durumda birakacak paketler ise Broadcastlerdir. Bilindiği gibi Switvhler Broadcast gecirirler, Dolayısıyla kendilerine gelen broadcaastleri bütün portlarına gönderirler.

Aşağıdaki şekilden hareketle Host A nin Söz gelimi bir ARP Request' te bulunduğunu varsayalım. Arp Request frameleri broadcast olduğundan 1. portlarından bu frameleri alan her iki switchde diğer bütün portlarından bu frame'i iletecektir.



Her iki switch arasında ki A ile gösterilmiş bağlantıdan da broadcastler yayinlanacak dolayısyla her iki switch de bu broadcast frame leri bu kez farklı portlardan olmak üzere, yeniden alacaklar ve yeniden flood edecekler. (Bu durum Broadcast Storm olarak bilinir.)

Bu vebunun gibi bir çok nedenle dogabilecek sorunlar için yardimimiza STP kosacaktır.

STP nin amaci genel olarak networklerdeki olası loop ları onlemek ve bunun için de her hedefe sadece bir yolun aktif olarak çalışmasını sağlamaktir. Bunun içinde STP Spanning Tree Algörithm'i (STA) kullanır.

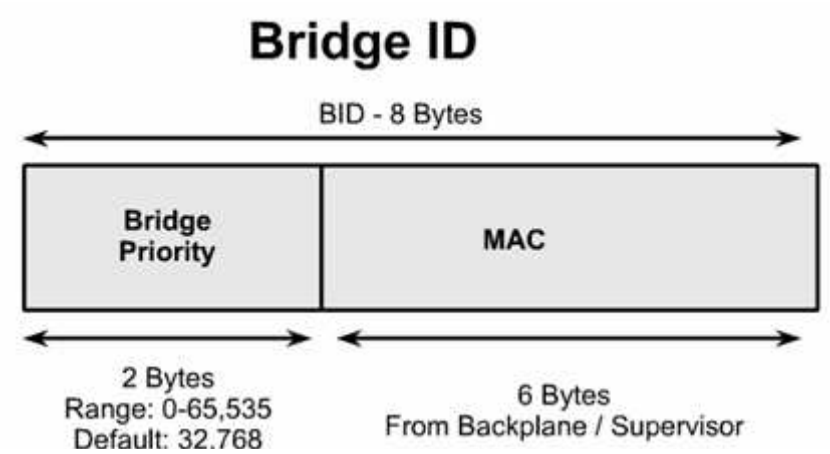
STA networkte bir referans noktasi oluşturur ve bu referans noktasından hareketle, birden fazla alternatif yol varsa, en iyi yol seçimini yapar. Bu referans noktasina Root Bridge denir.

Peki Root Bridge nedir, nasıl seçilir, kim secer...

Aslında ortamdaki bütün switchler Root Bridge'dir. Yani kendilerini oyle sanarlar.

Ortamdaki en küçük Bridge ID' ye sahip bridge Root Bridge'dir.

Bridge ID Bridge Priority ve MAC adresinden oluşur, 8 Byte'tir. Bütün switchlerin Priority'si default olarak 32768' dir.



Switchlerin default priority' leri değiştirilmediğinde hepsi eşit olacağından MAC adreslerine bakilabilir, bu durum da en küçük MAC adresine sahip Bridge Root olacaktır. (Switchlerin efendışı.)

STP hesaplamaları sırasında en iyi yol seçiminin yapılmasini sağlayacak kriter de Path Cost' tur. 1000/ Bandwith ile hesaplansa da IEEE çok kullanılan bant genişlikleri için costları yayinlamistir.

– 4 Mbps 250 (cost)

– 10 Mbps 100 (cost)

– 16 Mbps 62 (cost)

– 45 Mbps 39 (cost)

– 100 Mbps 19 (cost)

– 155 Mbps 14 (cost)

– 622 Mbps 6 (cost)

– 1 Gbps 4 (cost)

– 10 Gbps 2 (cost)

(Bir çok hesaptan kurtulduk sanirim)

Bütün bu öğrendiklerimizin isiginda switchlerin kriter olara aldikları 4 adimi sıralayabiliriz.

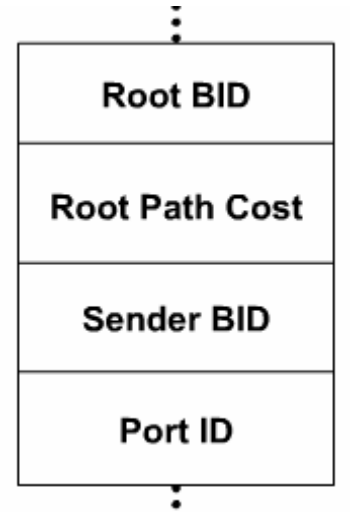
Step 1 - Lowest BID

Step 2 - Lowest Path Cost to Root Bridge

Step 3 - Lowest Sender BID

Step 4 - Lowest Port ID

Bridge bütün bu haberlesmeler için BPDU (Bridge Protocol Data Unit) mesajlarını kullanır, bu msajları bahsettigimi 4 adima göre değerlendirir. Bridge sadece kendisine gelen en iyi BPDU' yu tutar ve her yeni BPDU için 4 adimi tekrarlar. Gelen BPDU' lar arasında daha iyisi varsa onu alıp diğerini silecektir.



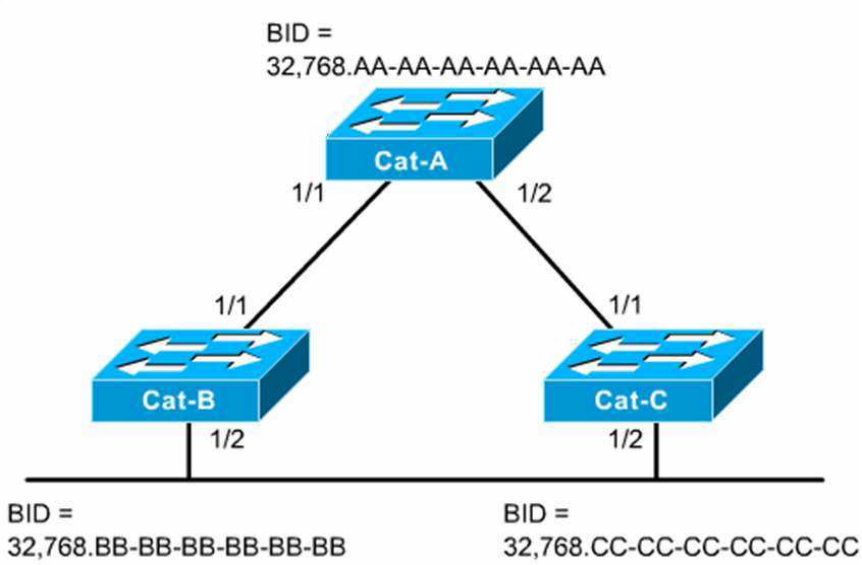
(BPDU Mesaj içeriği)

STP 3 Basamak ile yapısını olsturur.

Step 1 Root Bridge Seçilir

Step 2 Root Portlar Seçilir

Step 3 Designated Portlar seçilir.

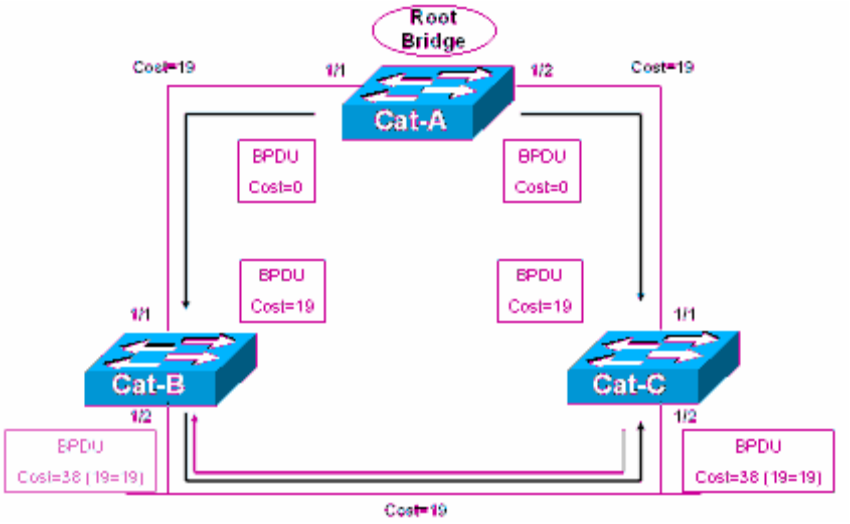


Şekilde ki gibi bir yapı üzerindenhareket ederek bu adimları açıklamaya çalışalım. Ilk adimda Root Bridge seçilmesi gerekir ve bunu için BPDU Mesajları gönderilir. Öncede soylediğimiz gibi her Switch kendisini Root Bridge varsayacağı için BPDU mesajlarındanki Root Bridge ID alanina kendi Idlerini yazar. (Root War başladi)

Çok gecmeden durumun oyle olmadığını anlarlar aslında ortamda ki CAT-A Switchinin gerçek Root olduğunu öğrenip BPDU larına bu Swirchi Root olarak eklerler. Root Bridgein portları her zaman Designated porttur ve sürekli forward durumundadir.

Root Bridge seçildiğine göre ikinci adima, Diğer Switchler için Root Portların seçilmesine geçebliriz.

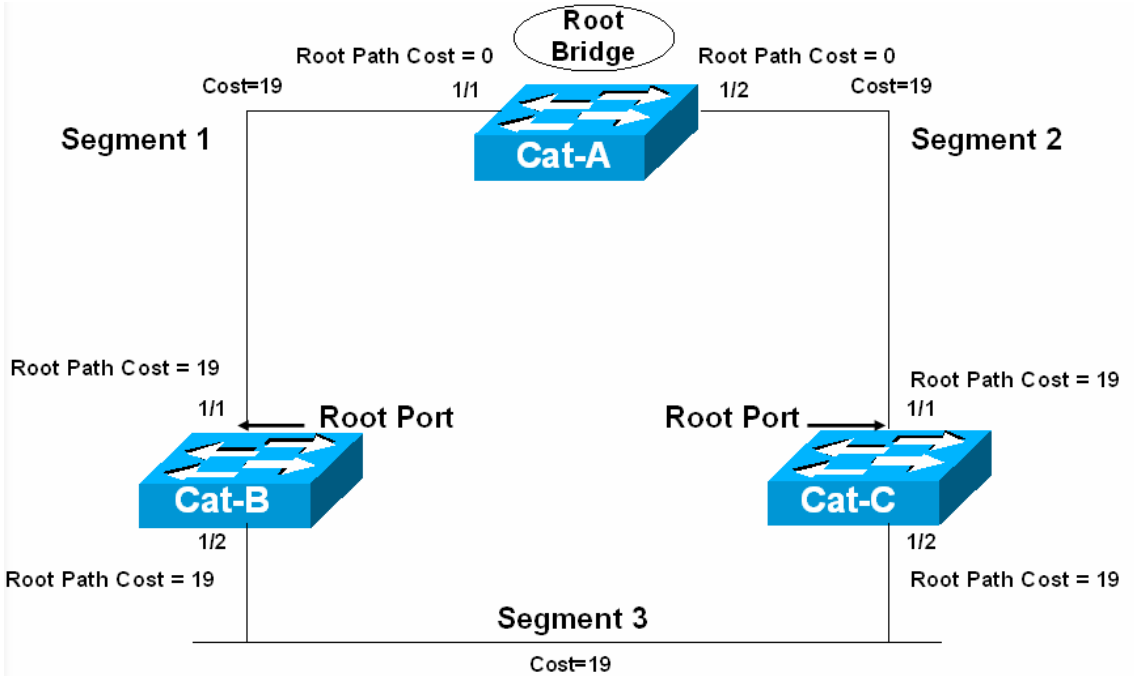
Switchlerin Root Bridge'e en yakin portları Root Porttur. Buraa yakinliktan kast ettigimiz sey aslında Root Bridge olan costtur.



Şekilde costları incelediğimiz zaman rahatlikla Cat-B ve Cat-C' nin 1/1 portlarının Root Port olduğunu soyleyebiliriz. Bu arada her Switch için sadece 1 tane Root Port , her segment içinde sadece 1 tane root port olmali.

Root Portlar seçildiğine göre Designated portlara geçebliriz.

Aşağıdaki şekilde de göruleceği gibi Segment 3' e dahil olan sadece 2 switch var ve bu switclerin birer tane seçilebilen root portları bu segmentte değiller. Bu yüzden bu segment için bu iki Switch portlarından biri Designated Port olarak seçilmeli.



Yine şekilde görulduğu gibi Her iki Switchi birbirlerine bağlayan portların Root Path Costları eşit. Root Bridge ID' lerde eşit olduğuna göre 3. adima geçebliriz.

Hatirlamak için Switchlerin BPDU paketleri içeriğinde sırasıyla inceledikler 4 adimi tekrar sıralamakta fayda var.

Step 1 - Lowest BID

Step 2 - Lowest Path Cost to Root Bridge

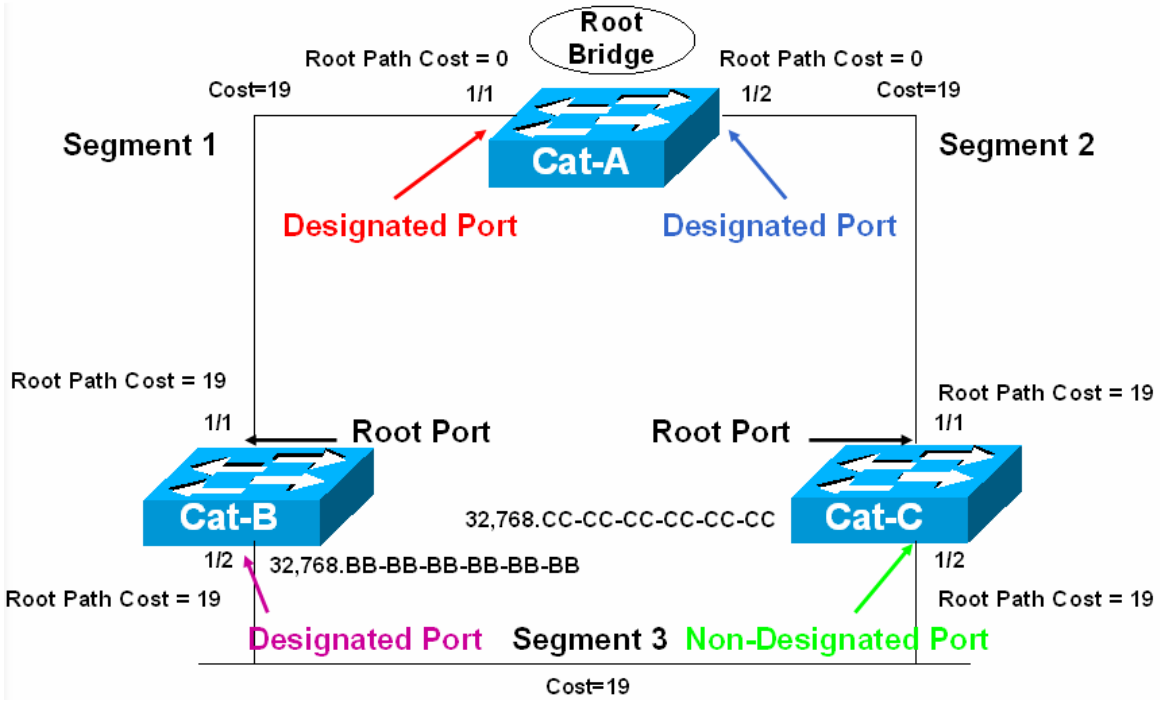
Step 3 - Lowest Sender BID

Step 4 - Lowest Port ID

Yani Lowest Sender BID… Bu durumda her iki Switch için BID değerleri karsilastirilip küçük olan Switche ait portun Designated Port olduğunu soyleyebiliriz.

Sonuc olarak Designated Port Forward duruma yani iletime gececek Non-Designated Port Block durumda kalacaktır.

Örnek olması açısından MAC Adresleri ve Priority değerleri, bu bilgilerden hareketler Portların durumu aşağıdaki şekilde özetlenmistir.



SPANNİNG TREE PORT DURUMLARI

Spanning Tree yapısı içerişinde portlar 5 ayrı durumda bulunabilirler.

1. Forwarding : Datalar gönderilir ve alinir.
2. Learning : Bridge Table oluşturulur.
3. Listening :Aktif topology oluşturulur.
4. Blocking : Sadece BPDU'lar alinir.
5. Disabled : Yönetimsel olarak down durumdadır.

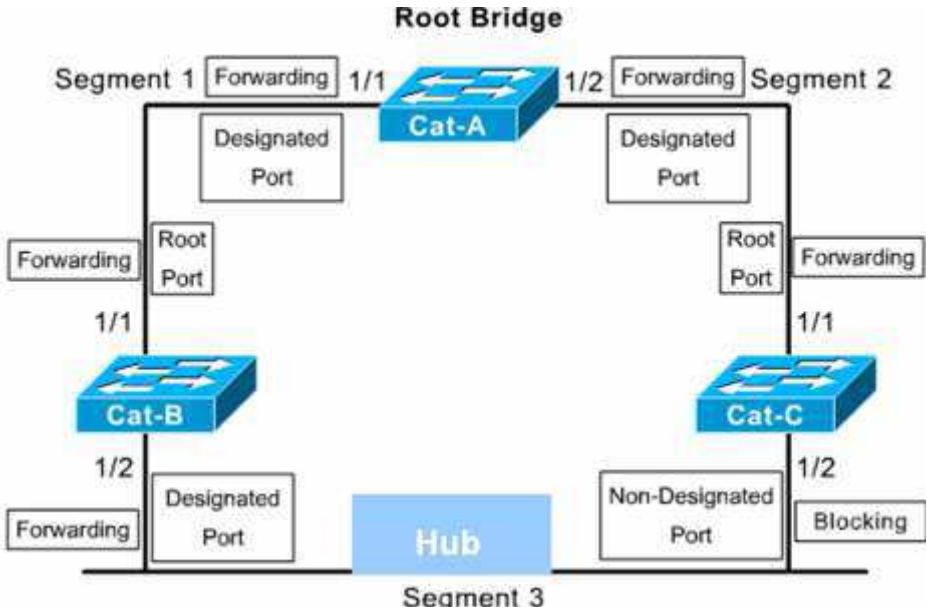
STP TİMERS

Hello Time: Root Bridge tarafindan gönderilen BPDU mesajları zaman aralığıdir. Default olarak 2 saniyedir.

Forward Delay: Portların Forward duruma gecmeden önce Listening ve Learning adimlarında gecen süredir, 15 saniyedir.

Max. Ağe: Bir BPDU' nun saklanma süresidir, 20 saniyedir. 20 saniye boyunca daha önce aldığı en iyi BPDU mesaji tekrarlanmazsa Max. Ağe dolmus olur ve port Listening Moda geçer.

Bir örnek ile STP zaman aralıklarını inceleyelim.



Şekildeki duruma göre Başlangıçta Cat-C' nin ½ portu Blocking durumda ve yanlizca BPDU mesahlarını dinliyor.

Şimdi Cat-B' nin ½ portunun down olduğunu varsayalım. Bu durumda Cat-C Artık BPDU mesajlarını alamayacaktır. Cat-c 20 saniye boyunca Blocking durumda kalacak ve 20 saniyenin sonunda Max. Ağe' e ulaşıldiği için durumunu değiştirecek, 15 saniye sürecek Listening mod ve yine 15 saniye sürecek Leraning Modun ardindan Forwarding duruma gececektir.

Yani 20 sn. max ağe + 15 sn. Listening + 15 sn. Learning modda kalacak Dolayısıyla Cat-B ½ portu down olduktan 50 saniye sonra Cat-C ½ portu devreye girecekyir.

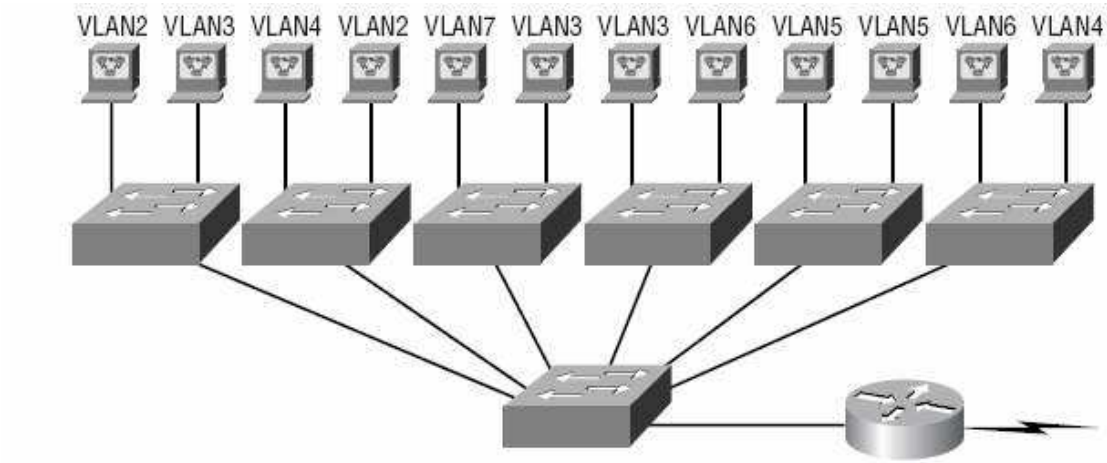
Fakat burada Cat-B' nin MAC Adres Table' inin silinmemesinden ve toplam 300 saniye botunca da Cache de kalacak olmasından dolayı bir sorun var gibi görünüyor.

Bu sorunda Root Bridge tarafindan gönderilecek TCN BPDU (Topology Change Notification BPDU) ile giderilecektir. Ortamdaki switch portlarının durumunda bir değişiklik olduğunda gönderilen bu mesaj ile switchler MAC adres Table' larının yasam sürelerini 15 saniyeye cekerler.

VLANS (VİRTUAL LOCAL AREA NETWORKS)

Virtual Local Area Network switch üzerinde yapılan mantıksal biri gruplama şekliden tanımlanabilir. VLAN oluştururken bilgisayarların fiziksel durumlarına, yerlerine bakmak yerine işlevine yada departmanına göre düzenlemeler yapılır.

Örneğin bir networkte Muhasebe bölümü bir VLAN' da İnsan Kaynakları başka bir VALN' da bulundurulabilir ve bu sayede iki departman arasındakiiletişim engellenmiş olur.



Pazarlama	VLAN2	172.16.20.0/24
İnsan Kaynakları	VLAN3	172.16.30.0/24
Teknik Büro	VLAN4	172.16.40.0/24
Muhasebe	VLAN5	172.16.50.0/24
İYönetim	VLAN6	172.16.60.0/24
Satış	VLAN7	172.16.70.0/24

Her VLAN ayrı bir Broadcast domain olur ve dolayısıyla Broadcast' ler kontrol altına alınabilir. Network üzerinde kullanılan hemen hemen her protokol Broadcast oluşturur ve bu broadcast' lerin miktarı Network performansını olumsuz etkileyebilir. Bunu önlemenin iki yolu vardır:

- Router kullanımı
- Switch Kullanımı

Sistem içerişinde uzak networkler varsa Router kullanımı uygun bir çözüm olabilir ama Local Area Network düşünüldüğünde Switch kullanmak ve VLAN' lar oluşturmak daha ucuz dolayısıyla daha mantıklı bir çözüm olacaktır.

VLAN' lar Switch portlarının Network yöneticileri tarafından atanmasıyla oluşturulur ki buna Static VLAN denir. Sistem de bulunan cihazların bir veritabanına girilmesi ve switchler tarafından otomatik oalrak atanmasıyla oluşan VLAN' lara ise Dinamik VLAN denir.

Static VLAN' lar hem daha güvenlidir hem de yönetimi ve bakımı Dinamik VLAN' lara göre daha kolaydır.

Default olarak bir switch üzerindeki bütün portlar VLAN1' dedir.

VLAN konfigürasyonu Switch modeline göre farklılık gösterebilir. Önemli olan mantığını anlamaktır, komutlar kullanılan switch içerişinde yardım alınarak yapılabilir. (Biz hem Cisco1900 hem de Cisco 2950 serişi switchleriçin konfigürasyon komutlarını vereceğiz fakat konfigürasyon çalışması yaparken Cisco1900 serişi Switchler üzerinde çalışacağız.)

VLAN oluşturmak komutlardan bağımsız olarak anlatmak gerekirse iki adımdan oluşur.

1. VLAN Oluşturulur

2. Portlar VLAN' lara üye edilirler.

1900 Switch İçin VLAN Oluşturma:

Switch#configure terminal

Switch(config)#vlan 2 name satis

Switch(config)#vlan 3 name muhasebe

Switch(config)#vlan 4 name yönetim

Switch(config)#exit

Switch#

2950 Switch İçin VLAN Oluşturma:

Switch#configure terminal

Switch(config)#vlan 2

Switch(config-vlan)#name satis

Switch(config)#vlan 3

Switch(config-vlan)#name muhasebe

Switch(config)#vlan 4

Switch(config-vlan)#name yönetim

2950 Seri switchler de her VLAN kendi alt modunda konfigüre ediliyor.

NOT: VLAN1 silinemez, değiştirilemez veya yeniden adlandırılamaz. VLAN' lar oluşturulduktan sonra artık ikinci adıma geçebiliriz. Bu adımda Switch portları VLAN'lar ile eslestirilecek. Tabi burada VLAN üyeliginin Static yada Dinamic olduğu da belirtiliyor.

1900 Seri Switchler için VLAN Üyeligi:

Switch#configure terminal

Switch(config)#interface Ethernet 0/2

Switch(config-if)#vlan-membership static 2

Switch(config-if)#exit

Switch(config)#interface Ethernet 0/3

Switch(config-if)#vlan-membership static 3

Switch(config-if)#exit

Switch(config)#interface Ethernet 0/4

Switch(config-if)#vlan-membership static 3

Switch(config-if)#exit

Switch(config)#

2950 Seri Switchler için VLAN Üyeligi:

Switch#configure terminal

Switch(config)#interface Ethernet 0/2

Switch(config-if)#switchport Access vlan 2

Switch(config-if)#exit

Switch(config)#interface Ethernet 0/3

Switch(config-if)# switchport Access vlan 3

Switch(config-if)#exit

Switch(config)#interface Ethernet 0/4

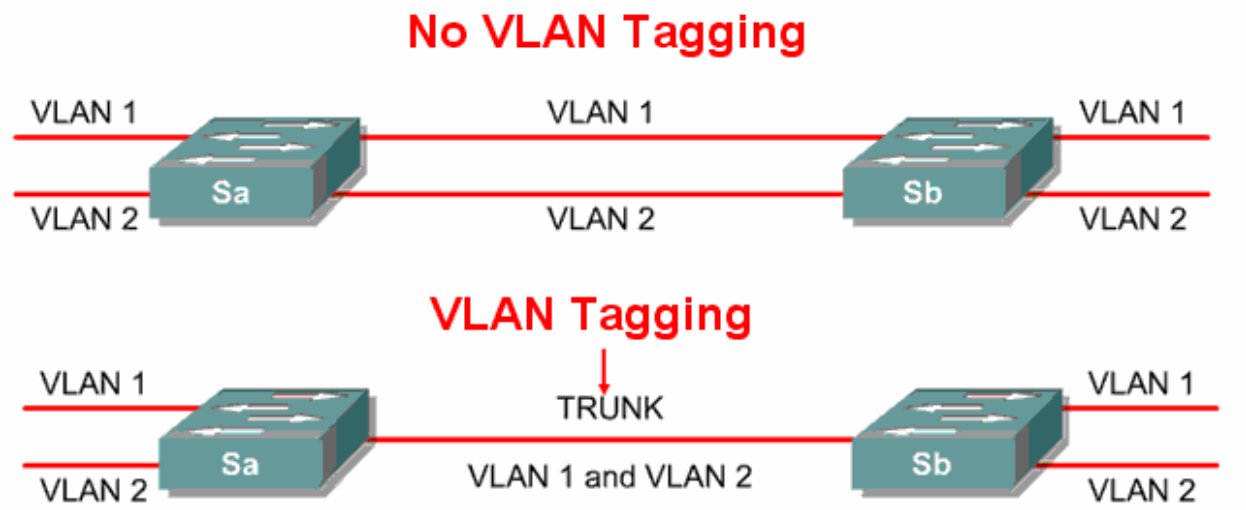
Switch(config-if)# switchport Access vlan 4

Switch(config-if)#exit

Switch(config)#

TRUNK VE TRUNK KONFİGÜRASYONU

Trunk bağlantılar cihazlar arasında VLAN'ları tasımak amacıyla kullanılırlar ve VLANların tümünü yada bir kısmını tasımak üzere biçimlendirilebilirler. Sadece Fast yada Gigabit Ethernet üzerinde destegi vardır. Cisco switch'ler trunk bağlantı üzerindeki VLAN'ları tanımak için iki ayrı yöntem kullanır: **ISL** ve **IEEE802.1q.**



Bir Switch üzerindeki bir porta trunk ing konfigürasyonu şu şekilde olur:

1900 Seri Switch için:

Switch#configure terminal

Switch(config)#interface fastethernet 0/20

Switch(config-if)#trunk on

Switch(config-if)#exit

2950 Seri Switch için:

Switch#configure terminal

Switch(config)#interface fastethernet 0/20

Switch(config-if)#switchport mode trunk

Switch(config-if)#exit

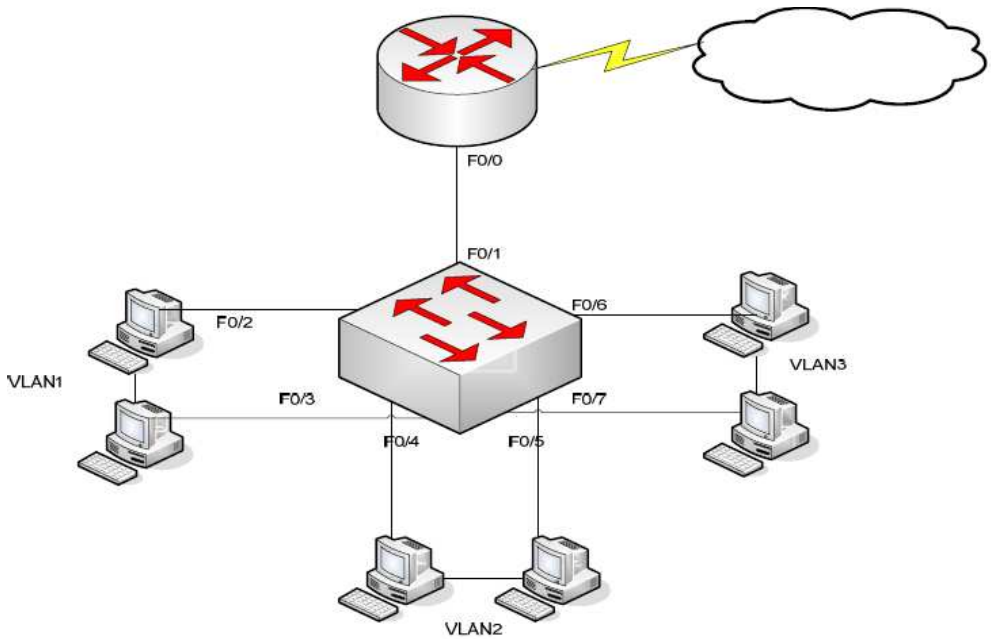
Inter-Switch Link (ISL): Cisco switch'ler tarafından kullanılır. Bu yöntem "external tağging" adı verilen, paketin orijinal boyunu değiştirmeyen, ancak 26 byte'lık bir ISL başlığını pakete ekleyerek, cihazlar arasında VLAN tanınmasını sağlayan bir yöntemdir. Ayrıca paketin sonuna paketi kontrol eden 4-byte uzunlugunda FCS (frame check sequence) alanı ekler. Paket bu eklentilerden sonra sadece ISL tanıyan cihazlar tarafından tanınabilir.

IEEE 802.1q: IEEE tarafından geliştirilen bu standart yöntem, farklı markadan switch yada router arasında, bir bağlantı üzerinden çok VLAN tasımak amacıyla kullanılır. Gelen paket üzerine tanımlanan standarda uygun bir başlık yerleştirilir ve cihazlar arasında pakete ait VLAN'ın tanınması sağlanır.

VLAN'LAR ARASINDA YÖNLENDİRME

Bir VLAN'a bağlı cihazlar kendi aralarında iletişim kurabilir, broadcast'lerini gönderebilirler. VLAN'ların network'ü fiziksel olarak böldükleri varsayıldığı için VLAN'lar arasında cihazların iletişim kurabilmesi ancak 3. katman bir cihaza yardımıyla olacaktır.

Bu durumda yapılacak bir router üzerinde her VLAN için bir bağlantı eklemek ve Router üzerinde gerekli konfigürasyonları yaparak iletişimi sağlamaktır.



Böyle bir topoloji üzerinde çalıştığımızı varsayalım:

Switch#configure terminal

Switch(config)#interface fastethernet 0/1

Switch(config-if)#trunk on

Switch(config-if)#exit

Switch(config)#interface fastethernet 0/2

Switch(config-if)#vlan-membership static 1

Switch(config-if)#exit

Switch(config)#interface fastethernet 0/3

Switch(config-if)#vlan-membership static 1

Switch(config-if)#exit

Switch(config)#interface fastethernet 0/4

Switch(config-if)#vlan-membership static 2

Switch(config-if)#exit

Switch(config)#interface fastethernet 0/5

Switch(config-if)#vlan-membership static 2

Switch(config-if)#exit

Switch(config)#interface fastethernet 0/6

Switch(config-if)#vlan-membership static 3

Switch(config-if)#exit

Switch(config)#interface fastethernet 0/7

Switch(config-if)#vlan-membership static 3

Switch(config-if)#exit

İlgili portları ilgili VLAN'lara atadık ve Router'a bağlantının sağlandığı FastEthernet0/1 portunda trunking'i aktif hale getirdik. Şimdi sıra Router üzerinde gerekli konfigürasyonu yapmaya geldi. Bunun için Router'ın Fastethernet 0/0 interface'i altında sanal interface'ler oluşturmak, bu sanal interfacelereip adresleri atamak ve encapsulation standardını belirlemek gerekir.

NOT: Gerçek Interface'in ip adresi olmamalı.

Router#configure terminal

Router(config)#interface fastethernet 0/0

Router(config-if)#no ip address

Router(config-if)#no shutdown

Router(config-if)#interface fastethernet 0/0.1

Router(config-subif)#encapsulation isl 1

Router(config-subif)#ip address {ip adresi} {subnet maskı}

Router(config-subif)#exit

Router(config-if)#interface fastethernet 0/0.2

Router(config-subif)#encapsulation isl 2

Router(config-subif)#ip address {ip adresi} {subnet maskı}

Router(config-subif)#exit

Router(config-if)#interface fastethernet 0/0.3

Router(config-subif)#encapsulation isl 3

Router(config-subif)#ip address {ip adresi} {subnet maskı}

Router(config-subif)#exit

Burada öncelikle 3 adet VLAN için Router üzerinde 3 adet sanal interface oluşturuldu. Hemen arkasından kullandığımız switchin 1900 serişi olduğunu varsayarak encapsulation metodunu belirledik ve o sanal interfacein hangi VLAN ile bağlantılı olduğunu belirledik.

2950 Seri Switchler 802.1q metodunu desteklediği için bu switchlerden konfigürasyonumuz:

Router(config-if)#interface fastethernet 0/0.1

Router(config-subif)#encapsulation dot1q 1

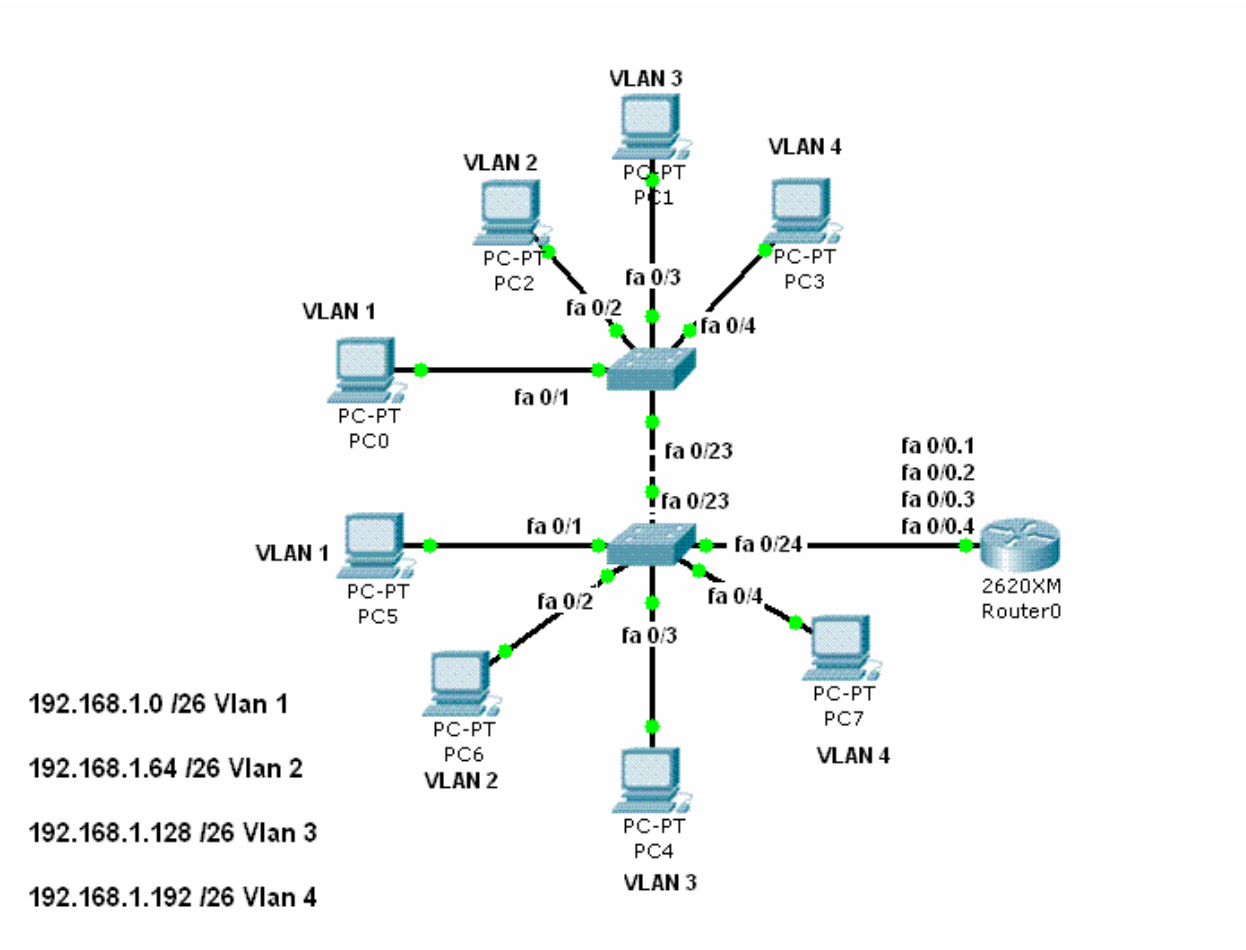
Router(config-subif)#ip address {ip adresi} {subnet maskı}

Şeklinde olacaktı. (802.1q = dot1q)

İslemimiz VLAN üzerindeki ip adreslerinden birini (önerilen ilk useable ip adresini) sanal interface'e vererek tamamlandı.

NOT: Cisco 1900 serişi switchler sadece isl encapsulation metodunu desteklerler 2950 serişi switchler ise sadece 802.1q' yu destekler. Bu yüzden bu iki switch arasında trunking gerçeklestirilemez.

Laboratuar ÇALIŞMASI



Laboratuar çalışmamız da VLAN 1 de dahil olmak üzere SwitchA ve SwitchB ye bağlı 4 adet VLAN var. SwitchA ve SwitchB fa0/23 portlarından birbirlerine bağlanmış ve bu portlarda trunk uygulanmistir.

Aynı şekilde SwitchB fa0/24 portundan Routera bağlanmış ve bu porttada trunk uygulanmistir.

Encapsulation dot1q kullanılmistir.

Switch ve Router running-config dosyaları aşağıdakidadir.

SwitchA#show running-config

!

version 12.1

!

hostname SwitchA

!

interface FastEthernet0/1

switchport mode access

!

interface FastEthernet0/2

switchport access vlan 2

switchport mode access

!

interface FastEthernet0/3

switchport access vlan 3

switchport mode access

!

interface FastEthernet0/4

switchport access vlan 4

switchport mode access

!

!

interface FastEthernet0/23

switchport mode trunk

!

interface FastEthernet0/24

switchport mode access

!

!

interface Vlan1

ip address 192.168.1.11 255.255.255.192

!

ip default-gateway 192.168.1.1

!

line con 0

!

end

SwitchA#show vlan

VLAN	**Name**	**Status**	**Ports**
1	default	active	Fa0/1, 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 Fa0/24, Gig1/1, Gig1/2
2	egitim	active	Fa0/2
3	muhasebe	active	Fa0/3
4	yönetim	active	Fa0/4
1002	fddi-default	active	
1003	token-ring-default	active	
1004	fddinet-default	active	
1005	trnet-default	active	

SwitchA#

SwitchB#show running-config

version 12.1

!

hostname SwitchB

interface FastEthernet0/1

switchport mode access

!

interface FastEthernet0/2

switchport access vlan 2

switchport mode access

```
!
interface FastEthernet0/3
switchport access vlan 3
switchport mode access
!
interface FastEthernet0/4
switchport access vlan 4
switchport mode access
!
interface FastEthernet0/23
switchport mode trunk
!
interface FastEthernet0/24
switchport mode trunk
!
!
interface Vlan1
ip address 192.168.1.12 255.255.255.192
!
ip default-gateway 192.168.1.1
!
line con 0
!
end
```

SwitchB#show vlan

VLAN	**Name**	**Status**	**Ports**
1	default	active	Fa0/1, 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 Fa0/24, Gig1/1, Gig1/2
2	egitim	active	Fa0/2

```
3    muhasebe    active    Fa0/3
4    yönetim     active    Fa0/4
1002 fddi-default        active
1003 token-ring-default  active
1004 fddinet-default     active
1005 trnet-default       active
-----
SwitchB#
Router#show running-config
!
version 12.2
!
hostname Router
!
interface FastEthernet0/0
no ip address
interface FastEthernet0/0.1
encapsulation dot1Q 1 native
no ip address
!
interface FastEthernet0/0.2
encapsulation dot1Q 2
ip address 192.168.1.65 255.255.255.192
!
interface FastEthernet0/0.3
encapsulation dot1Q 3
ip address 192.168.1.129 255.255.255.192
!
interface FastEthernet0/0.4
encapsulation dot1Q 4
ip address 192.168.1.193 255.255.255.192
!
interface Serial1/0
no ip address
shutdown
ip classless
!
```

line con 0

!

end

Router#

VLAN TRUNKİNG PROTOCOL (VTP)

VTP Vlan konfigutasyonun bütün networke yayilmasini sağlayan bir mesajlasma protokoludur.

VTP Layer 2 framelerini kullanır ve VLAN' ların bütün network içinde yönetilmesini, silinmesini, eklenmesini yada yeniden adlandirilmasini sağlar. Dolayısıyla VTP networkteki bütün switchlerin ve VLAN konfigurasyonlarının merkezi bir şekilde yönetilmesini sağlar.

VTP protokolunun çalışma prensibi içinde ortamda VTP Server ve VTP clientlarbulunur. Aynı domain de bulunan VTP Clientlar , serverdan VLAN bilgilerini alirlar.

o VLAN'lar VTP Server da oluşturulur.

o VLAN bilgileri client switchlere gönderilir.

o Aynı domain içinde bulunan switchler VLAN bilgilerini alirlar.

o Bu gelismeden sonar Artık client switchlerde portlar VTP Server da oluşturulanVLAN' lara atanabilir.

o VTP Client olarak konfihure edilen switchlerde VLAN oluşturulamaz.

o farklı domainlerde bulunan switchler VLAN bilgilerini paylasmazlar.

Switchler VTP bilgilerini almamak üzerede configure edilebilirler, Bu switchler VLAN bilgilerini Trunk portlarından gönderirken kendisine gelen bilgileri almaz ve kendi VLAN database' ini yapılandirmaz. Switchleri vu şekilde ÇALIŞMASI VTP Mode Transparent olarak adlandirilmistir. Bu modda çalışan Switchler VTP domaine katilmazlar.

Güvenlik açısından VTP domainlerine password verilebilir. Bu durumda password o domain de bulunan bütün switchlerde configure edilmelidir.

Gönderilen VTP mesafları VTP database' inden revision numarası ile birlikte tutulurlar, her mesaj ile bu numara artirilir. Daha büyük bir revision numarası ile gelen bilgiler switchler tarafindan daha yeni olarak Kabul edilir ve gelen VLAN bilgileri eskilerinin üzerine yazılır.

Buraya kadar anlattiklarımızin isiginda VTP domainlerinde switchlerin 3 ayrı modda çalışabileceklerini soyleyebiliriz.

o VTP Server

o VTP Client

o VTP Transparent

Konfigurasyon:

Switch# vlan database

Switch(vlan)# vtp domain domain-name

Switch(vlan)# vtp {server | client | transparent}

Switch(vlan)# vtp password password

Switch(vlan)# vtp v2-mode (version2)

Örnek

Switch# vlan database

Switch(vlan)# vtp domain corp

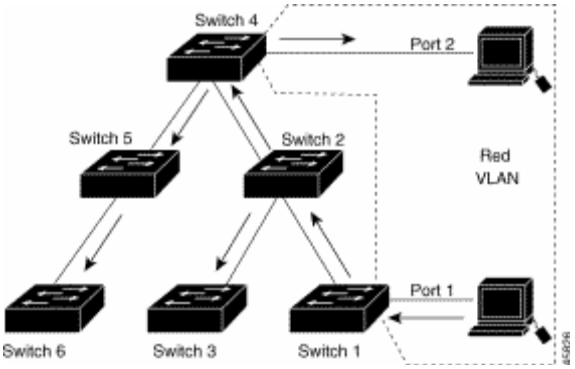
Switch(vlan)# vtp client

VTP PRUNİNG (BUDAMA)

VTP Pruning networkteki broadcast, multicast, unknown unicast gibi gereksiz flood edilen paketleri azaltarak network bant genişliği kullanimini artirir. Cisco Switchlerde default olrak disable durumdadır.

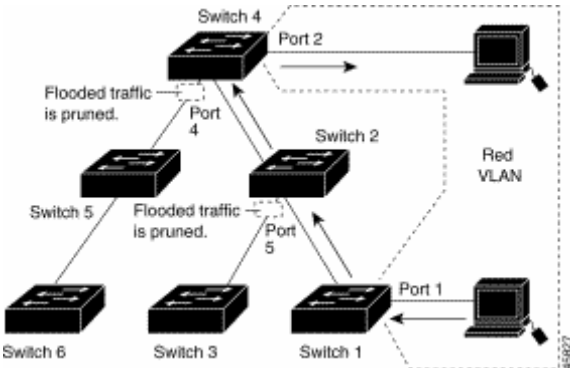
VLAN1' de VTP Pruning enable edilemezken diğer VLAN' larda edilebilir ve VTP Server da Pruning enable edildiğinde ise bütün domainde (tabi ki VLAN 1 dışında) enable olur.

Aşağıdaki şekilde Switch1'in 1. portu ve Switch4' un 2. portu Red VLAN'1 uye durumdalar. Hoslardan birinden gönderilen broadcast trunk portlarından bütün switchlere gider.



Red VLAN'a uye portları olmayan Switch 2-3-5-6 ' da aynı şekilde bu broadcast alacaktır.

Bunu onlemek için VTP Pruning enable edilebilir.



Switch4 un 4. portu ve Switch2' nin 5. portunda Red VLAN trafiği budanmistir. (VTP Pruning enable)

Konfigurasyon

Switch# vlan database

Switch(vlan)# vtp pruning

Belirli bir VLAN ise

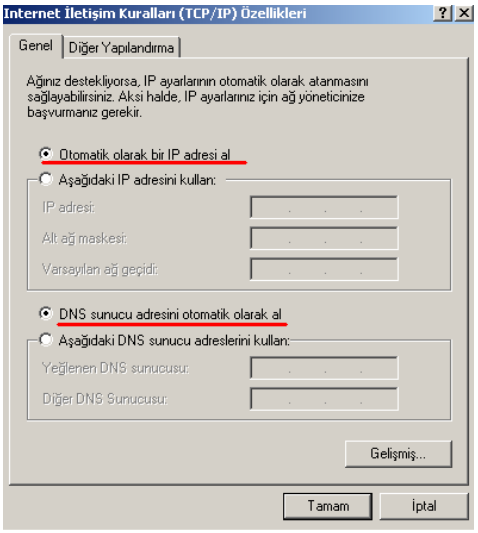
Switch(config-if)#switchport trunk pruning vlan remove vlan

Komutuyla pruning dışında birakilabilir.

DHCP (DYNAMİC HOST CONFİGURATİON PROTOCOL)

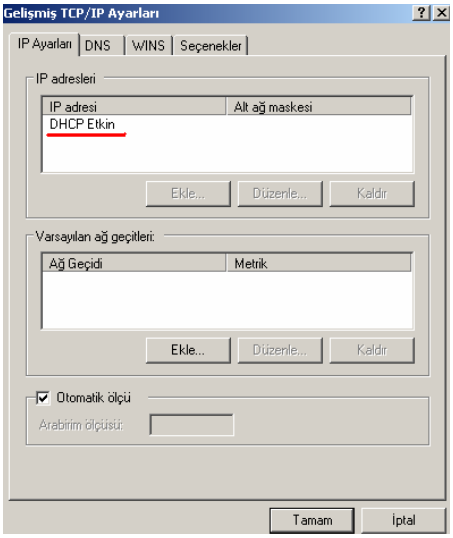
DHCP , DHCP kullanmak üzere yapılandırılmıs bilgisayarlara merkezi ve otomatik olarak ip adresi atanması ile TCP/IP bilgilerinin yapılandırılmasını ve bunların yönetilmesini sağlar. DHCP' nin uygulanması manuel olarak ip adresinin verilmesi nedeniyle ortaya çıkan bazı problemlerin azalmasını sağlar.

Bir bilgisayarı DHCP kullanmak üzere yapılandırmak için bilgisayarın TCP/IP konfigürasyonunda "Otomatik olarak ip al" seçenegini aktif etmek yeterlidir. İstendiğinde DNS sunucunun ismi de otomatik olarak DHCP Server' da alınabilir bunun için de " DNS sunucu adresini otomatik olarak al " seçenegi aktif hale getirilmelidir. Bu işlemler yapıldıktan sonra bilgisayarlar DHCP istemci durumuna gelecektir.



(Bir bilgisayarın DHCP istemci olarak ayarlanması)

Bu ayarlar yapıldıktan sonra TCP/IP konfigürasyonunun gelismis sekmesine baktığımızda DHCP' nin etkin olduğunu görebiliriz.



DHCP istemci DHCP Server ile haberlesmeye geçmesi ve ip adresini elde etmesi birkaç adımlık bir haberlesme ile sağlanır, Bu birkaç adımı basit bir şekilde inceleyecek olursak:

- İstemci bilgisayar başlangıçta DHCP Server adresini bilmediği için broadcast yolu ile ip adres istegini ortama yayar.

- İstegi alan DHCP Server, uygun olan bir ip adresini istemciye kiralama teklifinde bulunur. (Ip adresleri DHCP Server' lar tarafından belirli sürelerle istemcilere kiralanırlar, tamamen verilmezler)
- İstemci ip adres bilgilerini alır.
- DHCP Server veritabanında ip adresinin kiralandığı ve kiralama süresi bilgilerini yazar.

Özellikle büyük işletmelerde IP konfigürasyonu ile ilgili çıkabilecek sorunların çözülmesinde yada olası değişikliklerin düzenlenmesinde DHCP Server ile TCP/IP konfigürasyon bilgilerini dağıtmak akıllıca bir çözüm olacaktır.

Bunun için bir bilgisayarı DHCP Server atamak yeterli olabileceği gibi istendiğinde Router' larda gerekli konfigürasyonlar yapıldığında DHCP hizmeti verebilirler.

DHCP Server kullanarak istenirse oluşturulacak ip havuzundan ip adresleri rast gele dağıtılabilir yada MAC adreslerine bazı ip adresleri reserve edilebilir ve istenirse bazı ip adreslerinin hiçbir şekilde dağıtılmaması sağlanabilir.

CİSCO ROUTER' IN DHCP SERVER OLARAK KONFİGÜRE EDİLMESİ

Cisco Router' larda DHCP server default olarak çalışır durumdadır. Herhangi bir nedenle daha önceden DHCP Server devre dısı bırakıldıysa;

Router(config)# service dhcp

komutu ile DHCP Server aktif hale getirilebilir. Yiene istendiği zaman basına "no" konularak devre dısı bırakılabilir.

Router(config)# no service dhcp

Router' ın DHCP hizmeti verebilmesi için, hangi aralıklarda hangi networke ait ip adreslerinin dağıtılacağı bilgisinin Router' a bildirilmesi gerekir.

Bunun için şu komutlar yazılmalı:

Router(config)#ip dhcp pool poolismi

Router(Config-dhcp)# network ip_aralığı mask subnet_maski

Örneğin:

Router(Config)# ip dhcp pool Academytech

Router(Config-dhcp)#network 192.168.0.0 mask 255.255.0.0

İstersek bu networkteki bazı ip adreslerinin yada bir ip adres aralıgının istemci bilgisayarlara dağıtılmasını engelleyebiliriz. Bunun için **"ip dhcp excluded"** komutunu kullanmalıyız. Komutun genel kullanımı şu şekildedir;

Router(config)#ip dhcp excluded-address Başlangıç_ipsi bitis_ipsi

Örneğin ilk örnekte belirttigimiz ip adres aralıgına ait adreslerden 192.168.1.1 ' den 192.168.1.10 ' a kadar olan ip adreslerinin dağıtılmamasını istersek;

Router(config)#ip dhcp excluded-address 192.168.1.1 192.168.1.10

Komutunu yazmamız gerekir.

Bununla birlikte DNS ip adresi, etki alanı adı, NetBios Server ip adresi ve Default Gateway gibi adresleri de konfigürasyonunu Yaptığımız da Router ile dağıtabiliriz. Bu komutların genel kullanımı ise söyledir:

Router (config-dhcp)#domain-name academytech.com

Router (config-dhcp)#dns-server dns_server_ip_adresi

Router(config-dhcp)#netbios-name-server server_ip_adresi

Router(config-dhcp)#default-router routerin_ip_adresi

İstenirse ip adreslerinin reserve edilebileceğinden bahsetmistik. Bunun için ip adresi reserve edeceğimiz bilgisayarın MAC adresini bilmemiz gerekir. Örneğin MAC Adresi 00-11-2F-B2-12-B2 olan bir bilgisayara 192.168.1.100 ip adresini reserve edelim. Bu durumda yeni bir havuz oluşturmalıyız:

Router(config)#ip dhcp pool Academytech-Lab

Router(config-dhcp)#host 192.168.1.100 mask 255.255.0.0

Router(config-dhcp)#client-identifier 0100-11-2F-B2-12-B

Burada MAC adresinin başında yer alan “ 01 ” ifadesi network kartının Ethernet için tasarlandığı anlamına gelir.

Ip adreslerinin dağıtırken olası çakısmaları önlemek için gerekirse Router’ ın ip adreslerini kiraya vermeden önce kullanımda olup olmadığını denetlemesini sağlayabilir ve kira süresini de konfigüre edebiliriz.

Router(config)# ip dhcp ping packets ping_sayısı

Router(config-dhcp)#lease gün saat dakika

Ayrıca:

Router# show ip dhcp binding reserve_edilmiş_adres

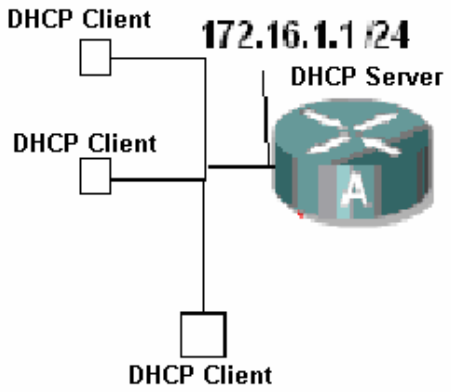
Reserve etigimiz ip adresleri hakkında bilgi,

Router# show ip dhcp conflict

Komutu ile dhcp’ de çakısan ip adreslerini görüntüleyebilir,

Router# show ip dhcp server statistics

Komutu ile dhcp server hakkında istatistiksel bilgileri alabiliriz. Şimdi örnek olucak bir konfigürasyon yapalım.



DHCP Client' lara DHCP Server tarafından otomatik olarak 172.16.1.1 / 24 networkünde ip adresleri dağıtılacak.

Senaryoyu biraz daha geliştirmek için 172.16.1.2 – 172.16.1.5 arasındakiip adreslerinin dağıtılmamasını istediğimizi de düşünelim.

```
Router(config)#ip dhcp pool Academytech
Router(dhcp-config)#network 172.16.1.0 255.255.255.0
Router(dhcp-config)#exit
Router(config)#ip dhcp ex
Router(config)#ip dhcp excluded-address 172.16.1.2 172.16.1.5
Router(config)#
```

Burada ip havuzumu oluşturduk ve dağıtılmasını istemediğimiz ip aralıgını router' a bildirdik.

Söz gelimi etki alanı adımız "AcademyTech", Dns Server'ın ip adresi:"172.16.1.2" ve Default Gateway'da 172.16.1.1 olsun. Bu bilgilerinde DHCP tarafından dağıtılmasını istersek konfigürasyona şu şekilde devam etmeliyiz:

```
Router(config)#ip dhcp pool Academytech
Router(dhcp-config)#domain-name AcademyTech
Router(dhcp-config)#dns-server 172.16.1.2
Router(dhcp-config)#default-router 172.16.1.1
Router(dhcp-config)#exit
Router(config)#_
```

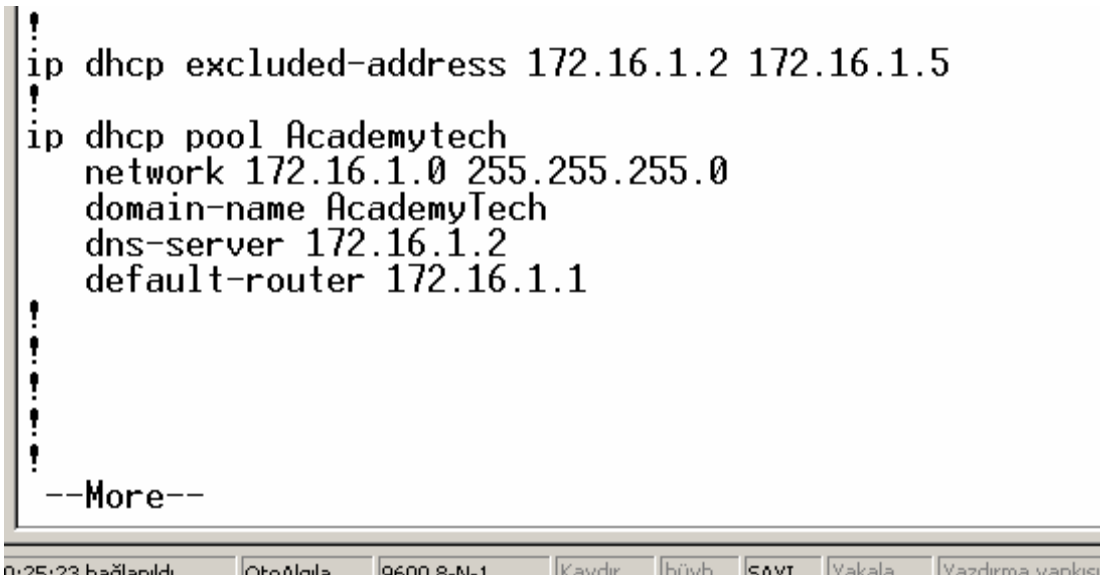
00:16:15 bağlanıldı | OtoAlgıla | 9600 8-N-1 | Kaydır | büyh | SAYI | Yakala | Yazdırma

Artık bilgisayarlarımızı DHCP Client olarak ayarladıktan sonra ip adreslerinin bizim router üzerinde Yaptığımız konfigürasyona uygun olarak alacaklarır.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Sürüm 5.1.2600]
(C) Telif Hakkı 1985-2001 Microsoft Corp.

C:\Documents and Settings\Cisco>ipconfig /all

Windows IP Yapılandırması

        Ana Bilgisayar Adı  . . . . . : JUPITER-11
        Birincil DNS Soneki . . . . . :
        Düğüm Türü . . . . . : Bilinmiyor
        IP Yönlendirme Etkin . . . . . : Evet
        WINS Proxy Etkin . . . . . : Evet
        DNS Soneki Arama Listesi. . . . : AcademyTech

Ethernet bağdaştırıcı Academytech:

        Bağlantıya özgü DNS Soneki .  . . : AcademyTech
        Açıklama  . . . . . : SiS 900 PCI Fast Ethernet Bağdaştırı
cısı
        Fiziksel Adres. . . . . : 00-0D-87-17-0D-01
        Dhcp Etkin. . . . . : Evet
        Otomatik Yapılandırma Etkin. . . : Evet
        IP Adres. . . . . : 172.16.1.6
        Alt Ağ Maskesi. . . . . : 255.255.255.0
        Varsayılan Ağ Geçidi. . . . . : 172.16.1.1
        DHCP Sunucusu . . . . . : 172.16.1.1
        DNS Sunucusu. . . . . : 172.16.1.2
        Kira Sağlanan. . . . . : 21 Mayıs 2006 Pazar 14:26:14
        Kira Bitişi . . . . . : 22 Mayıs 2006 Pazartesi 14:26:14

C:\Documents and Settings\Cisco>
```

Bilgisayarın ip konfigürasyonunda görüldügü gibi bizim istediğimiz şekilde bir çalışma oldu.

Running Konfigürasyona baktığımız da ise DHCP ile ilgili şu bilgileri göreceğiz:



DHCP kullanmaktan vazgeçtigimiz andan itibaren DHCP hizmetini devre dışı bırakabiliriz.

```
Router(config)#no service dh
Router(config)#no service dhcp
```

NETWORK ADDRESS TRANSLATİON

Internette gideceğimiz yeri bulmak için IP adresleri kullanırız. Ama her IP adresini internet ortamında kullanamıyoruz. Bazı özel IP adresleri vardır. Bu adresler, daha doğrusu IP adres aralıkları kendi yerel aglarımızda kullanmamız için ayrılmıstır. Bunlar Address Allocation for Private Internets (özel internetler için adres payı) diye tanımlanır, kısaca Private Adresses (özel adresler) diyoruz. İnternettekullandıklarımıza da Public (Halka Açık) Addresses diyoruz.

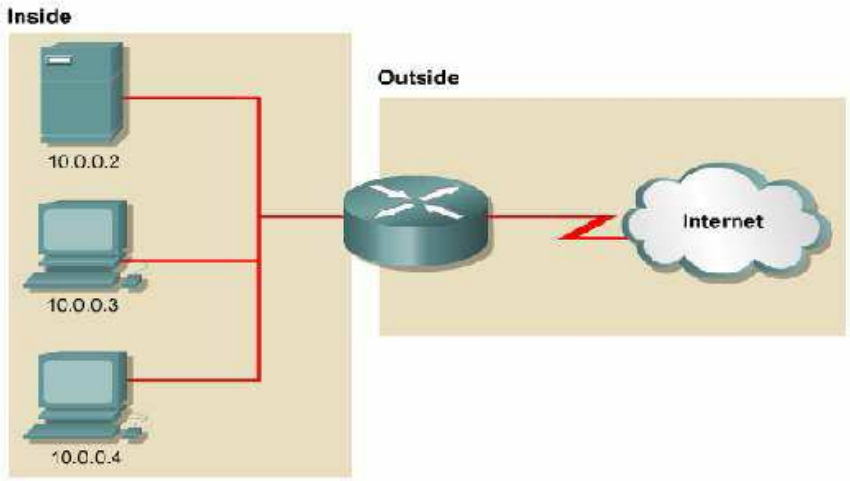
Özel IP adresleri RFC 1918 ile belirlenmistir ve;

10.0.0.0 ile 10.255.255.255

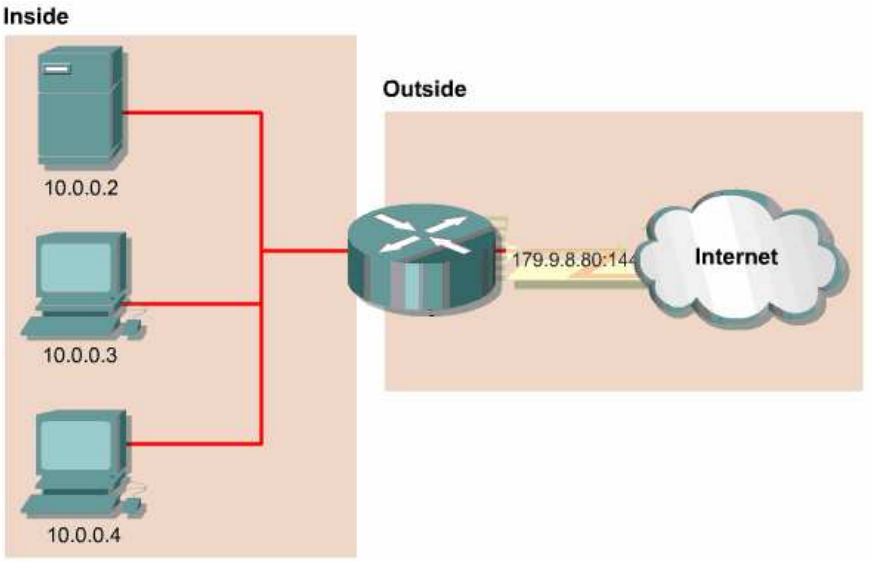
172.16.0.0. ile 172.31.255.255

192.168.0.0 ile 192.168.255.255 arasındadır.

İç networkümüzde kullanmamız için ayrılan bu ip adresleri internette kullanılamazlar ve biz de bu ip adresleri ile internete erişemeyiz. Dolayısıyla internet ortamına girerken public bir ip adresine sahip olmamız gerekli. Bu durumda bize NAT yani Network Address Translation yardımcı oluyor ve NAT konfigürasyonu yapıldıktan sonra iç networkümüzdeki herhangi bir ip adresine sahip bilgisayar dışarı çıkarken bizim istediğimiz bir ip adresine dönüsüyor, mesela modemin ip adresine.



Bu topolojide 10.0.0.0/ 24 networküne ait bilgisayarlar internete erişecekler. Karşımıza "inside" ve "outside" olmak üzere iki kavram çıkıyor. Yine topolojiden anlasılacagı gibi inside iç networkümüz ve outside da dıs network yani internet yada hedef network. Bu kavramlar önemli zira NAT konfigürasyonu sırasında Adres dönüstürme isleminde inside ve outside olarak kullanılacak interface' ler belirlenmelidir.

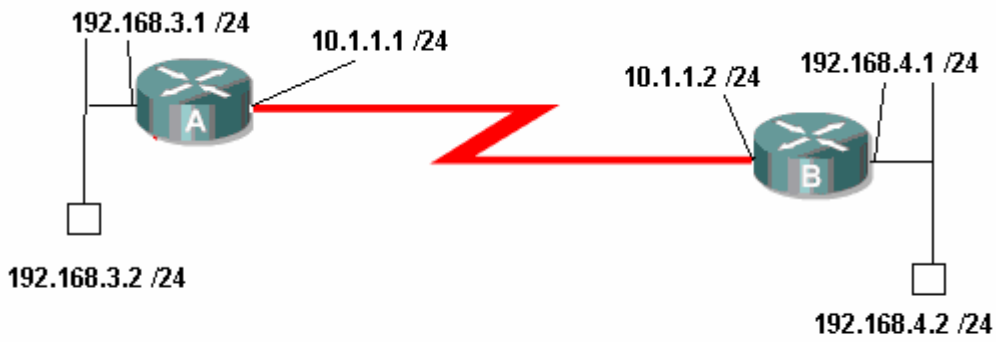


(Topoloji ve ip adresleri dikkatle incelendiğinde NAT islemi görülecektir.)

NAT' ı 3 başlık altında inceleyebiliriz:

- Static NAT : Birebir iç bloktaki IP adreslerini dıs IP adreslerine çevirme.
- Dynamic NAT: Bir havuz yaratarak dinamik olarak içerdeki adresleri bu havuzdaki dıs IP bloklarıyla esleme
- Overloading: Bütün makinaları makina sayısına oranla daha az IP adresiyle dışarıya çıkarma

NAT KONFİGÜRASYONU



Böyle bir senaryoda Static NAT uygulaması yapacak olursak Routerlar şu şekilde konfigüre edilmeli.

```
RouterA(config)#ip nat inside source static 192.168.4.2 10.1.1.1
RouterA(config)#interface ethernet 0/0
RouterA(config-if)#ip nat in
RouterA(config-if)#ip nat inside
RouterA(config-if)#exit
RouterA(config)#interface seri
RouterA(config)#interface serial 0/1
RouterA(config-if)#ip nat out
RouterA(config-if)#ip nat outside
RouterA(config-if)#exit
RouterA(config)#_
```

0:41:30 bağlanıldı | OtoAlgıla | 9600 8-N-1 | Kaydır | büyh | SAYI | Yakala | Yazdırma yankısı

(RouterA için konfigürasyon)

```
RouterB(config)#ip nat inside source static 192.168.4.2 10.1.1.2
RouterB(config)#inter
RouterB(config)#interface ethernet 0/0
RouterB(config-if)#ip nat inside
RouterB(config-if)#ip nat inside
RouterB(config-if)#exit
RouterB(config)#interface serial 0/1
RouterB(config-if)#ip nat out
RouterB(config-if)#ip nat outside
RouterB(config-if)#exit
RouterB(config)#_
```

0:43:26 bağlanıldı | OtoAlgıla | 9600 8-N-1 | Kaydır | büyh | SAYI | Yakala | Yazdırma yankısı

(RouterB için konfigürasyon)

Bu yapılan konfigürasyon ile A routerı için iç networkte bulunan 192.168.3.2 ip adresinin Router Serial interface' inden çıktıktan sonra 10.1.1.1 ip adresine, B Routerı için iç networkte bulunan 192.168.4.2 ip adresinin Router Serial isnterface' inden çıktıktan sonra 10.1.1.2 adresine dönüsmesini saglamıs olduk.

```
RouterA#sh ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 10.1.1.1           192.168.4.2        ---                ---
RouterA#_
```

(Router A için Ip NAT Translations Tablosu)

Benzer bir senaryo üzerinde Dynamic NAT uygulayabiliriz. Konfigürasyonda bazı farklılıklar olacaktır. Basta da bahsettigimiz gibi Dynamic NAT için “outside” tarafında bir ip adres havuzu oluşturulmalı ve “inside” tarafta bir Access list yazılmalıdır.

RouterA#configure terminal

RouterA(config)# ip nat pool AcademyTech 10.1.1.1 10.1.1.5 netmask 255.255.255.0

RouterA(config)# access-list 1 permit 192.168.3.0 0.0.0.255

RouterA(config)#ip nat inside source list 1 pool AcademyTech

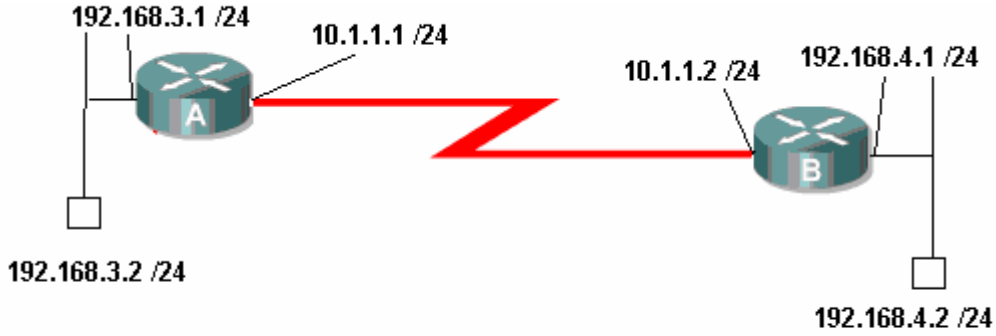
RouterA(config)#interface ethernet 0

RouterA(config-if)#ip nat inside

RouterA(config-if)#exit

RouterA(config)#interface serial 0

RouterA(config-if)#ip nat outside

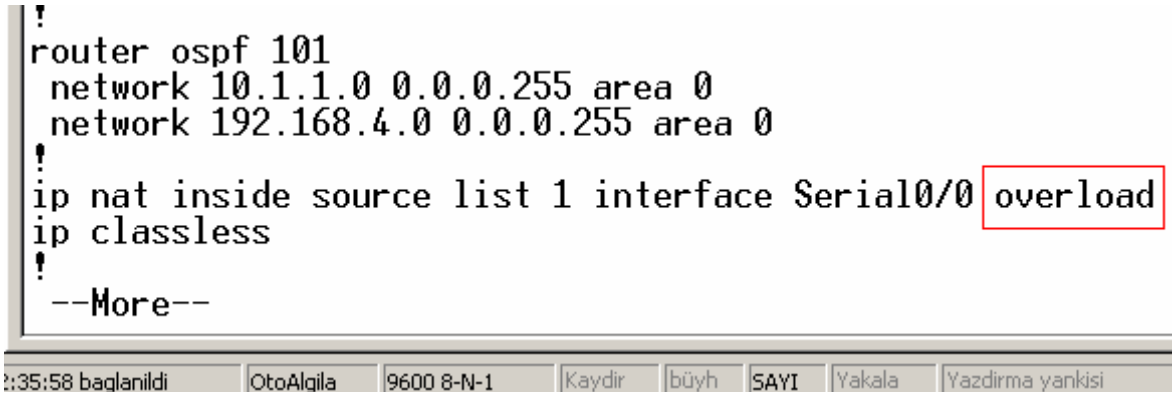


Overload uygulamasında tüm bir network aynı ip adres üzerinden çıkarılabilir. Burada ip adresi belirtmek yerine değişken interface kullanmak gerekir.

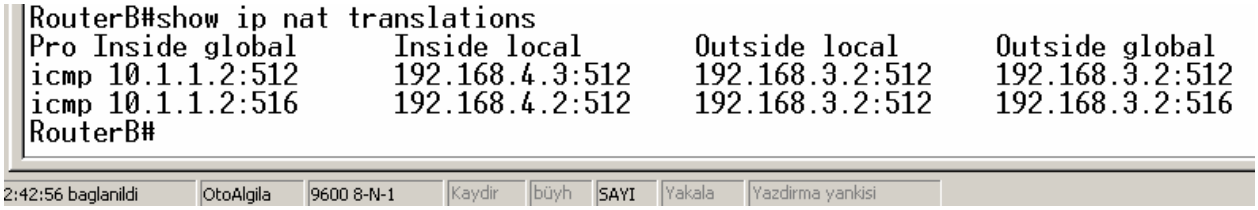
Router B üzerinde NAT konfigürasyonumuz şu şekilde yapılacaktır:

```
RouterB(config)#access-list 1 permit 192.168.4.0 0.0.0.255
RouterB(config)#ip nat inside source list 1 ?
  interface  Specify interface for global address
  pool       Name pool of global addresses

RouterB(config)#ip nat inside source list 1 interface serial 0/0
RouterB(config)#interface et
RouterB(config)#interface ethernet 0/0
RouterB(config-if)#ip nat inside
RouterB(config-if)#exit
RouterB(config)#interface ser
RouterB(config)#interface serial 0/0
RouterB(config-if)#ip nat out
RouterB(config-if)#ip nat outside
RouterB(config-if)#
```

Overload uygulamasında da interface' lerin "inside" ya da "outside" oldukları belirtilmelidir.



(Overload islemi Running Konfigürasyondan incelenebilir)



(IP Nat çevrimlerinin görüntülenmesi)

WAN TEKNOLOJİLERİ

WAN yani Wide Area Network Teknolojilerinin anlaşılmasi için bazı terimlerin önceden bilinmesi fayda saglayacaktır. Bu terimleri kisaca şu şekilerde tanımlayabiliriz.

Customer Premises Equipment (CPE) : Musteri tarafindan kullanılan cihazlara genel olarak verilen addir.

Demarcation (Demarc) : Servis saglayinin hizmet saglayacagi ve bu hizmet ile ilgili desteklerini şurdureceği, musteriye en yakin noktadir. Bu noktadan sonra oluşabiliecek olası hatalar ile ilgili servis saglayaci sorumluluk kabul etmez, musterinin kendisinin çözüm bulmasi gerekir.

Synchronous: Seri bağlantılarda her iki noktada ki cihazların birbirlerine data gönderimi sırasında hizlarını eşitlemeye çalıştıkları durumdur şeklinde anlatilabilir. Bu tarz bir iletimde bizim için önemli olan nokta upload ve download hizlarının eşit olmasıdır.

Asynchronous: Dial-Up modemler örnek olarak gösterilebilir. Baglarılarda ki her nokta veri iletim hizlarının eşit olduğunu kabil eder ama eşit olmadığı durumlarda eşitlemek için bir çalışma yapmaz. Bu durumda upload ve download hizları da birvirinden farklı olacaktır.

Data Services Unit/ Channel Services Unit (CSU/DSU) : DTE olan musteri ekipmanında clock üretimi saglayacak cihazlardir. WAN aslında DTE networklerin DCE networkler üzerinden birbirlerine baglanan LAN; lar toplulugudur şeklinde tanımlanabilir. Bu durumda Örneğin DTE olan Routerlarda data iletimini başlatacakDCE bir cihaza ihtiyac olacaktır, Örneğin modem.

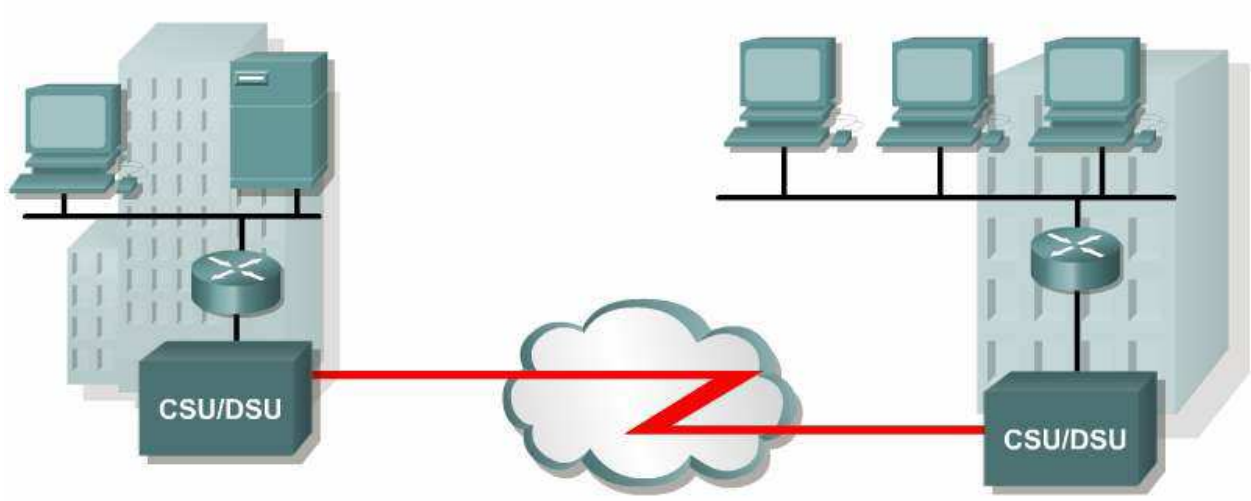
Lanoratuar ortamlarında kullandiğimiz özel kablolar ile DCE cihazlar yerine clock üretimini routerlara yaptiriyorduk. Gerçek dunyada uctan uca DTE ve DCE olan kablolar ile bağlantı saglamak imkansiz olacagi açık olduğuna göre mevcut hatlar üzerinden iletimin saglanabilmesi için DSU/CSU cihazlara ihtiyac vardır.

T1 circuit router

WAN BAĞLANTILARI

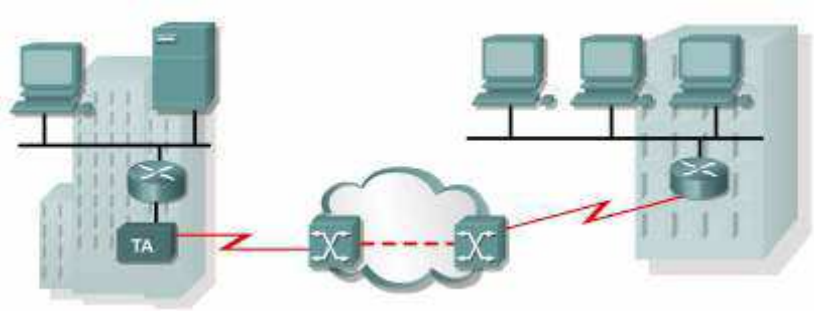
Kiralık Hatlar (Leased Lines): Kiralık hatlar tek bir firmaya atanmis, noktadan noktaya bağlantının saglandiği senkron iletim hatlarıdir. Senkron iletisim kullanılmasından dolayı upload ve download hizları eşittir. 45 Mbps' e kadar hiz desteklemektedir. Bağlantı kurulduktan sonra hat devamli aktif durumdadır.

Bu tur bağlantılarda daha sonra detaylı inceleyeceğimiz HDLC, PPP veya SLIP protokolleri kullanılır.



Devre Anahtarlama (Circuit Switching): Asenkron iletisim cesididir ve düşük bant genişliğine ihtiyac duyulan durumlarda önerilebilir. Bu bağlantılarda Artık neredeyse tamamen vazgecilen Dial-ip modemler veya ISDN hatları kullanılır.

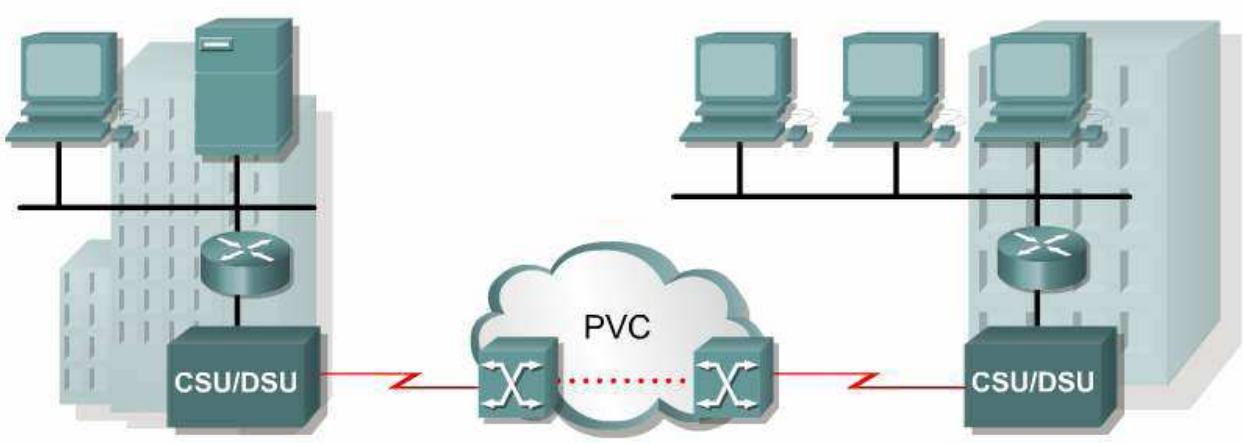
Burada modemler arasında bağlantı kurulduktan sonra hattin sürekli aktif kalmasi maliyeti artiracai için pek tercih edilmeyecektir ama Söz gelimi zaten var olan bir hatta yedek olması ve o hat koptugunda devreye girip, hat tekrar aktif olduğundan devreden cikmasi saglanabildiğinde son derece kullanisli olacaktır.



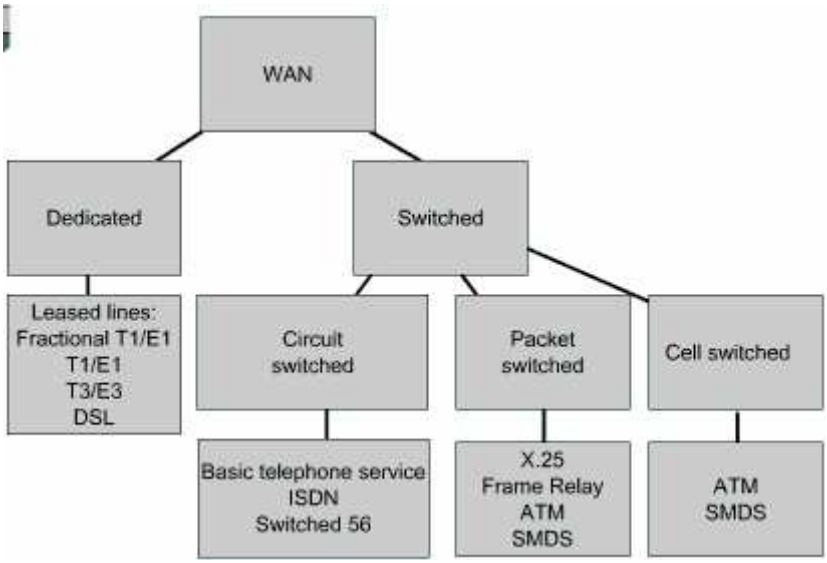
Asenkron iletisim Sözkonusu olduğu için upload ve download hizları eşit değildir. HDLC, PPP ve SLIP protokolleri kullanılabilir.

Paket Anahtarlama (Paket Switching): Geniş alan aglarında sabit miktarlarda datanin gönderilmesi durumunda en uygun çözüm kiralık harlar olacaktır. Fakat agimizda belirli zaman aralıklarında yüksek datalar gönderilirken bazen çok daha az data gönderimi Sözkonus ise bant genişliğinin paylasimi esasina göre tarasarlanmis Packet Switching kullanılabilir.

Servis saglayicilar burda bir miktar bant genişliğini garanti ederken, garanti etmediği ama mümkün olduğunda kullanmasina izin verdiği daha yüksek bir bant genişliğide saglarlar.



Bu tur baglatilarda detaylı inceleyeceğimiz Frame-Relay ile birlikte X.25 ve ATM protokolleri kullanılmaktadir.



WAN terimleri ve açıklamaları aşağıdaki tabloda verilmiştir.

Terim	Açıklama
Costumer premises equipment (CPE)	Müşterinin sahip olduğu ve kendi binasında bulundurduğu Cihazlar için kullanılır.
Demarcation(demarc)	Servis sağlayıcı firmanın sorumluluğunun bittiği nokta.Bu nokta müşterinin CPE 'sine bağlantının sağlandığı noktadır.
Local loop	Demarc'ların ,en yakın anahtarlama ofisine bağlantılarını sağlar.
Central Office (CO)	Müşterilerin ,servis sağlayıcısının networkune katıldığı nokta.POP(Point of Presence) olarak da bilinir.
Toll network	Servis sağlayıcının networkündeki trunk hatları.

HDLC

HDLC; IBM tarafından geliştirilmiş standart bit tabanlı bir protokoldür. HDLC (High-Level Data-Link Control) ; data link katman protokolüdür. Cisco' nun geliştirdiği HDLC ile diğer üretici sirketlerin geliştirdiği HDLC iletisim kuramaz. Bu diğer üreticiler içinde geçerlidir. Yani bütün HDLC protokollerine üreticisine özeldir diyebiliriz.

HDLC ; adres alanı , çerçeve alanı, kontrol dizisi alanı (FCS), ve protokol tür alanını içeren çerçevelemeyi tanımlar. HDLC hata düzeltimi aynen Ethernet gibi yapar. HDLC alt bilgisinde FCS alanını kullanır. Alınan çerçevede hatalar oluşmuş ise çerçeveyi düzeltmeden iptal eder.

HDLC Çerçevelemesi :

HDLC ISO frame					
Flag	Address	Control	Data (Payload)	FCS	Flag
1 byte	1 byte	1 or 2 bytes	1500 bytes	2 (or 4) bytes	1 byte

İki router'ı HDLC kullanarak haberlestirmek için asagıdaki komut satırları kullanılır.

Serial Interface de encapsulation 'ı HDLC olarak ayarlamak :

```
Router(config)#int ser 0
Router(config-if)#encapsulation hdlc
Router(config-if)#
```

Status

Yaptığımız konfigürasyonu görmek için show interface serial0 kullandık :

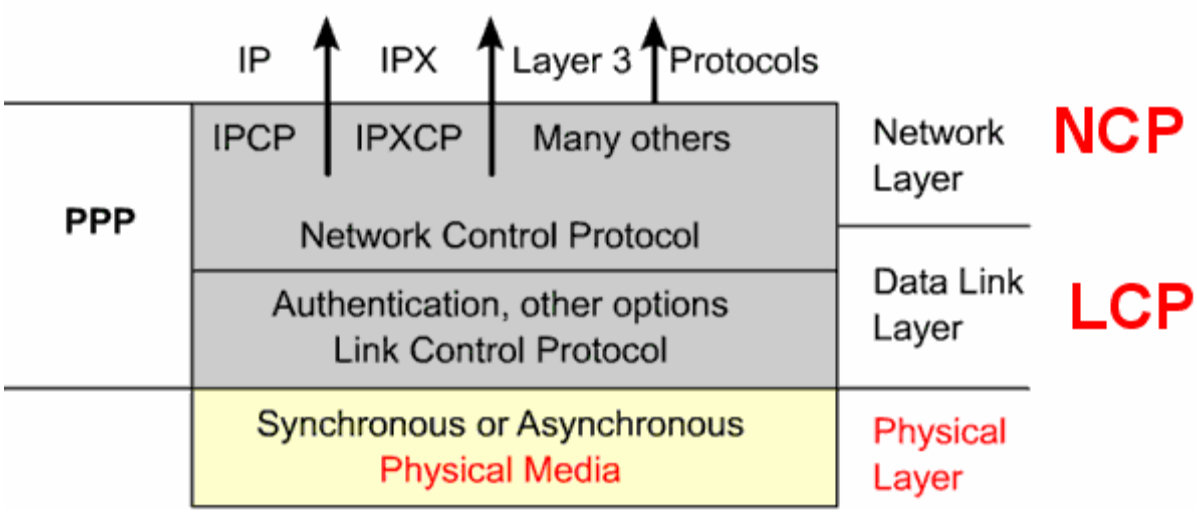
```
Router#sh int serial0
Serial0 is up, line protocol is up
  Hardware is HD64570
  Internet address is 172.19.1.13/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 1000 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of show interface counters never
```

PPP

PPP HDLC protokolune göre bütün üretici firmaların Routerları tarafindan desteklendiği için daha çok tercih edilen encapsulation metodudur.

PPP' bütün olarak incelemek biraz zordur Çünkü aslında PPP iki tane aly protokolden oluşur. Bunlar soyle sıralayabiliriz;

1. Link Control Protocol (LCP)
2. Network Control Protocol (NCP



LCP point to point bağlantının saglanmasi içinkullanılırken NCP network katmanı protokollerinin konfigurasyonu için kullanılır.

LCP, Authentication, sıkıştırma, hata kontrol ve birden fazla yol arasında load balancing gibi özellikleri saglar.

PPP oturumlar opsiyönel olan seçimler ile birlikte 5 adimda oluşur.

1. Link establishment - (LCPs)
2. Authentication - Optional (LCPs)
3. Link quality determination - Optional (LCPs)
4. Network layer protocol configuration (NCPs)
5. Link termination (LCPs)

```
Router#show interfaces serial0/0
Serial0/0 is up, line protocol is up
  Hardware is HD64570
  Internet address is 10.140.1.2/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
rely 255/255, load 1/255
  Encapsulation PPP, loopback not set, keepalive
set (10 sec)
  LCP Open                    <-- LCP
  Open: IPCP, CDPCP           <-- NCP
  Last input 00:00:05, output 00:00:05, output
hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0
drops
  5 minute input rate 0 bits/sec, 0 packets/sec
```

PPP Authentication

Router ‘ın seri interface ‘lerinde PPP tanımı yapmak için “encapsulation ppp” komutu kullanılır. Baglantının saglandığı her iki uçtaki interface’ lerin ikisinde de PPP aktif yapılmalıdır.

```
Router(config)#interface ser 0
Router(config-if)#encapsulation ppp
Router(config-if)#exit
Router(config)#
```

Ayrıca Router ‘lara “hostname” komutunu kullanarak bir isim verilmelidir.

```
Router(config)#hostname kadiköy
kadiköy(config)#
```

Ve karsı tarafın baglantı yapacagı sırada kullanacagı kullanıcı adı ve sifresi global konfigürasyon modundayken tanımlanmalıdır. Kullanılan sifre tüm router ‘larda aynı olmak zorundadır.Daha sonra bir kimlik doğrulama metodu da belirlemek gerekir. Bunun için öncelikle interface moda girilerek “ppp authentication” komutun kullanılır.

Daha önce bahsettigimiz PAP yada CHAP metodlarından biri seçilir. Dikkat edilmesigereken seçilen metodun her iki router da ortak seçilmesidir. Eğer bir router da PAP diğer router da CHAP seçildiyse iletisim kurulamayacaktır.

```
kadiköy(config)#username mecidiyeköy password academytech
kadiköy(config)#int ser 0
kadiköy(config-if)#ppp authentication chap
kadiköy(config-if)#
```

Eğer authentication metodunu PAP seçmiş olsaydık , interface içerişinde PAP ‘ı aktifetmemiz gerekecekti. Çünkü Cisco IOS 11.1 ve sonrasında PAP default olarakdisable durumdadır.

```
kadiköy(config)#int ser 0
kadiköy(config-if)#ppp pap sent-username mecidiyeköy password academytech
```

CHAP KONFİGÜRASYONU

show run

hostname Mecidiyeköy
username Kadıköy password
academytech

int ser 0/0
ip address 125.1.1.1 255.255.255.0
encapsulation ppp
ppp authentication CHAP

show run

hostname Kadıköy
username Mecidiyeköy password
academytech

int ser 0/0
ip address 125.1.1.2 255.255.255.0
encapsulation ppp
ppp authentication CHAP

PAP KONFİGÜRASYONU

show run

hostname Mecidiyeköy
username Kadıköy password
academytech

int ser 0/0
ip address 125.1.1.1 255.255.255.0
encapsulation ppp
ppp authentication PAP
ppp pap sent-username
Mecidiyeköy
password Academytech

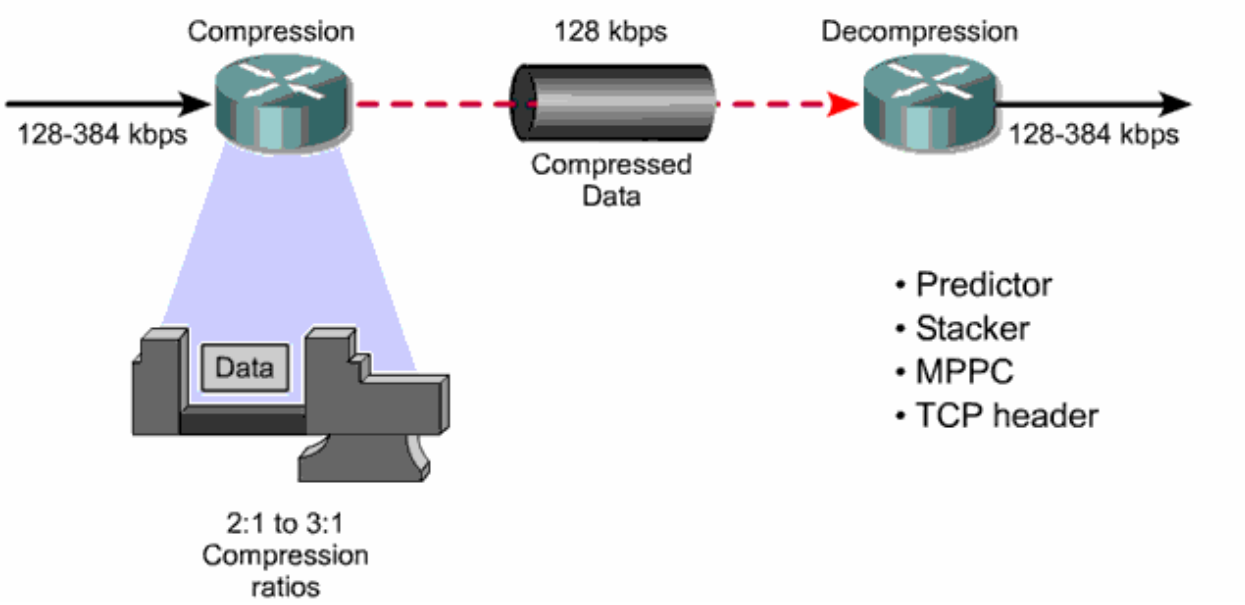
show run

hostname Kadıköy
username Mecidiyeköy password
academytech

int ser 0/0
ip address 125.1.1.2 255.255.255.0
encapsulation ppp
ppp authentication PAP
ppp pap sent-username
Kadıköy
password Academytech

PPP Compression

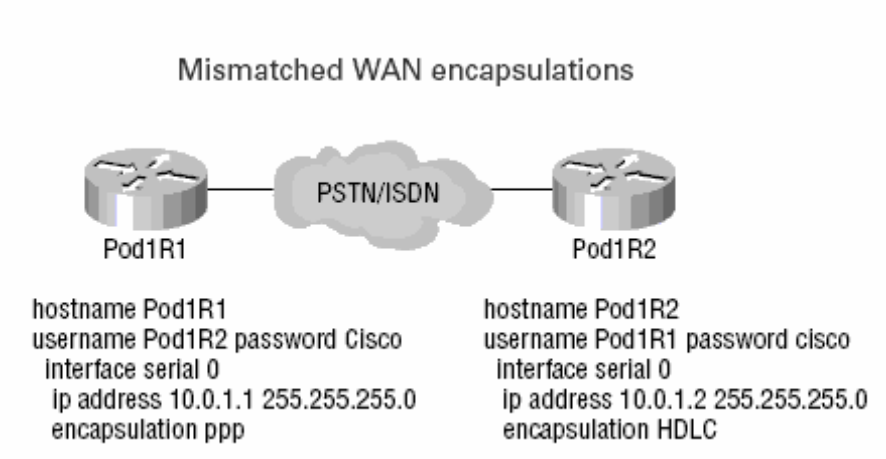
PPP datayi sıkıştırabilme özelliği ile düşük bant genişliğinde dahi Yükse performans saglayabilmektedir.

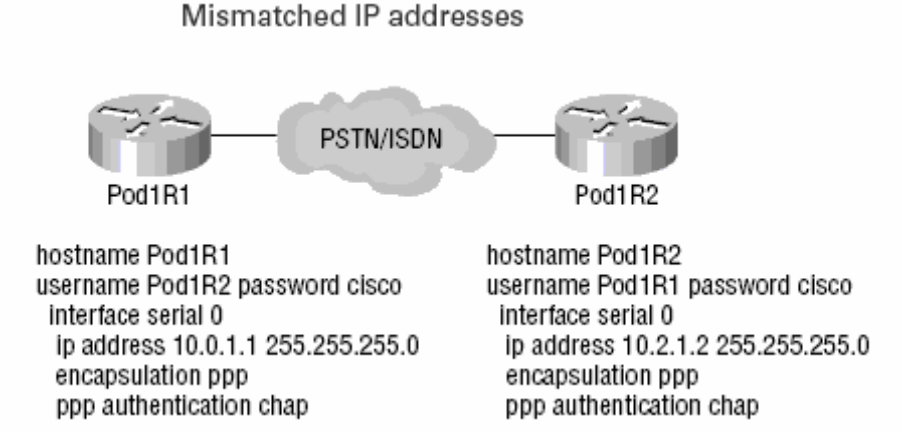


4 farklı Compression tipi vardır.

1. Predictor
2. Stacker
3. MMPC
4. Tcp Header Sıkıştırma

Hatali PPP Konfigurasyon Örnekleri



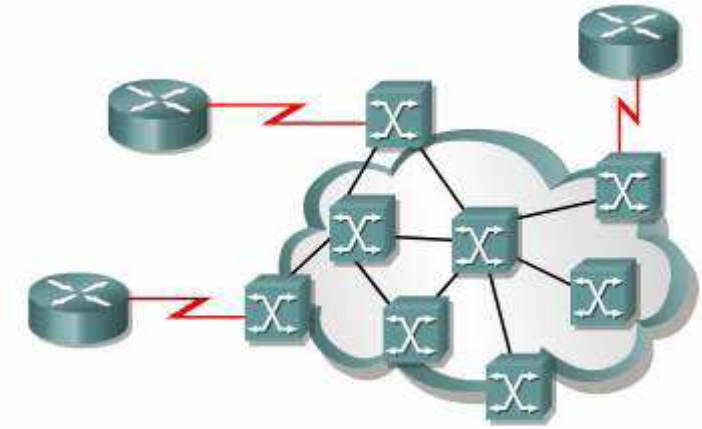


FRAME RELAY

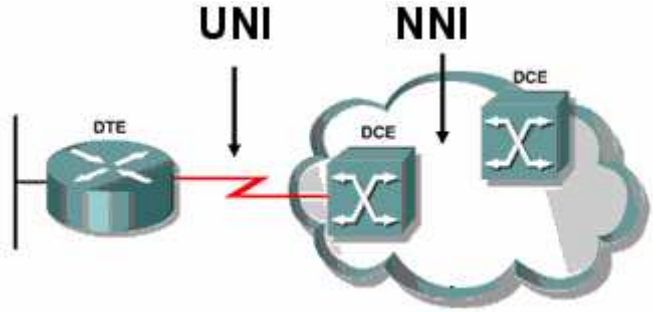
Frame Relay Packet Switching teknolojisyle tümleşik bir WAN servisidir ve OSI referans modelinin Data-Link Katmanında çalışır. Frame Relay HDLN ‘ nin bir alt bileseni olan Link Access Procodure for Frame Relay (LAPF) protokolunu kullanır.

Burada data frameler halinde musterinin DTE cihazlarından diğer nokta veya noktalardaki DTE cihazlarına DCE cihazlar üzeriden tasinir. Burada ki DCE cihazlar ya da DCE network telekom firmalarının sagladiği network ve cihazlardi, kontrolu ve konfigurasyonu bu firmalar tarafindan yapılir.

Frame Relay networklerinde genellikle 56 Kbps ve 2 Mbit arasında babnt genişlikleri kullanılmaktadir fakat 45 Mbit’e kadar desteklenmektedir.



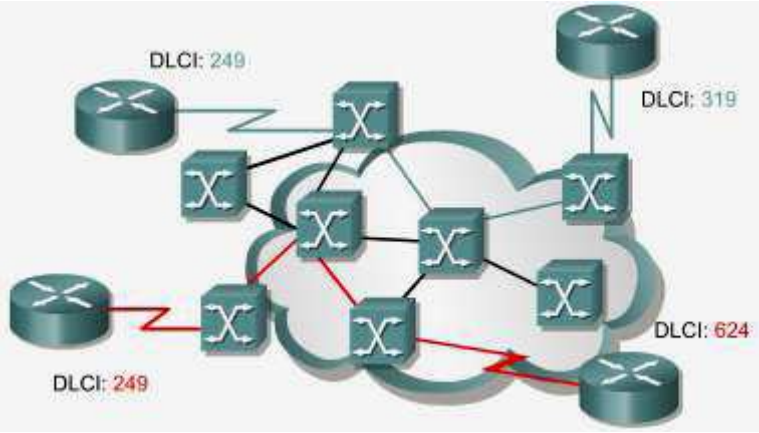
Frame Relay networklerinde musteri ve servis saglayici arasında ki baglatiya User-To Network Interface (UNI), birbirinden farklı servis saglayaicilara ait cihazların baglandikları noktalara ise Network-To-Network Interface (NNI) denir.



Frame Relay Networkleri oluşturulurken servis saglayinin vereceği DLCI numaralirinin tanımlanmasi önemlidir. Çünkü servis saglayici yada telekom bu DLCI numaralarına kendi switchleri üzerinden yol verecek ve iki nokta arasında sanal bir devre oluşturarak bağlantının kurulmasini saglayacaktır.

Burada bahsettigimiz sanal devreler pek kullanılmayan Switched Virtual Circuits (SVCs) ve Permanent Virtual Circuits (PVCs) olmak üzere ikiye ayrılir.

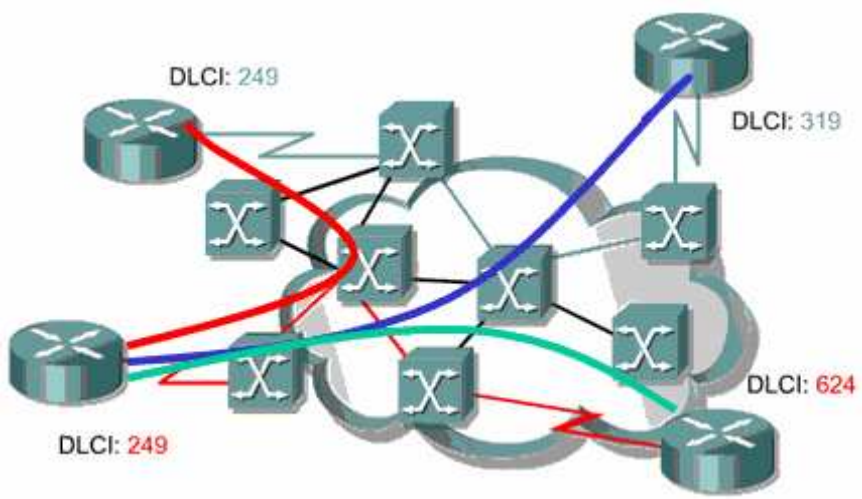
SVCs’ lerde iki nokta arasında kurulan bağlantı için geçerli olan yol dinamik olarak değişmekteyken PVCs’ lerde iki nokta arasında sabir bir yol tanımi yapılir ve manuel olarak değiştirilmediği sürece sürekli o yol kullanılır.



Sözgelimi şekilde kirimiz olarak isaretlenen yol Söz konusu iki nokta arasındaki PVC' yi belirtir. Fakat burada SVC ile bir çalışmadan bahsetseydik mevcut yol yerine alternatif yollardan biri de kullanılabiliyor olacakti.

Çünkü SVC' ler gecici olarak oluşturulurlar ve bu bağlantıların oluşturulmasi için bir cevrim (call setup) gerekmektedir.

Frame Relay konuşan bir router birden fazla nokta arasından birebir bağlantı yapılmasi gereken durumlarda her nokta için ayrı PVC ler oluşturabilir. Merkez nokta üzerinde yapılacak ve her noktaya erişim için farklı olan DLCI numaraları ile bu mümkün olacak ve o dakikadan itibaren merkez Frame Relay router bütün noktalara aynı anda hizmet verebilecektir.



Frame Relay Headers

Frame Relay ile konfigure edilmiş routerlar iki farklı Frame Relay Header'i desteklerler.

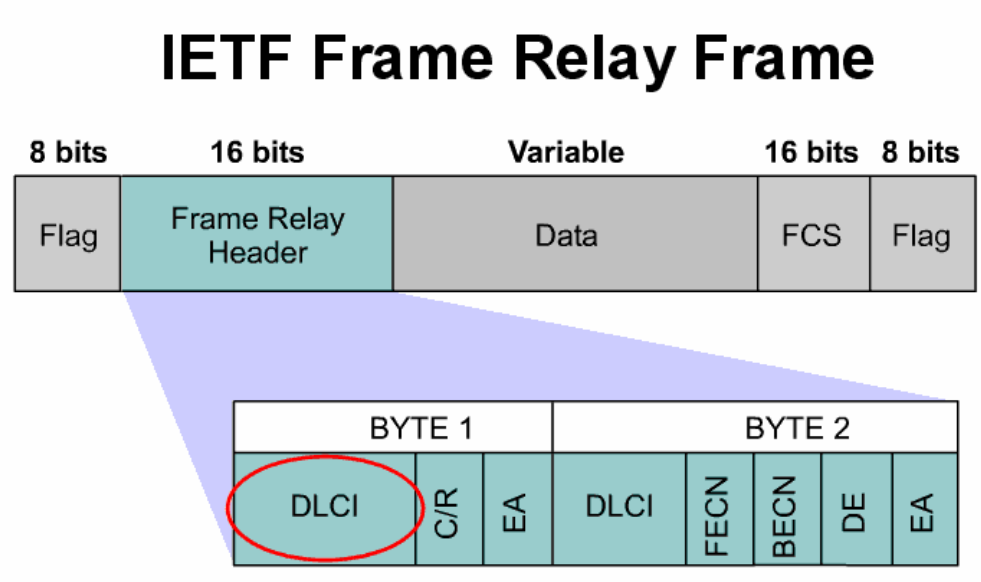
1. Cisco

2. IETF

Cisco adindan da anlaşıldiği gibi Cisco özeldir ve ortam da Cisco dışında üreticilere ait Routerlar varsa kullanılamaz. Bununla birlikte Frame Relay framelerine 4 Byte lik headerler eklediği için önerilen değildir.

IETF ise birden fazla üreticiyi destekler ve framelere Cisco' nun aksine sadece 2 byte' lik headerlar ekler.

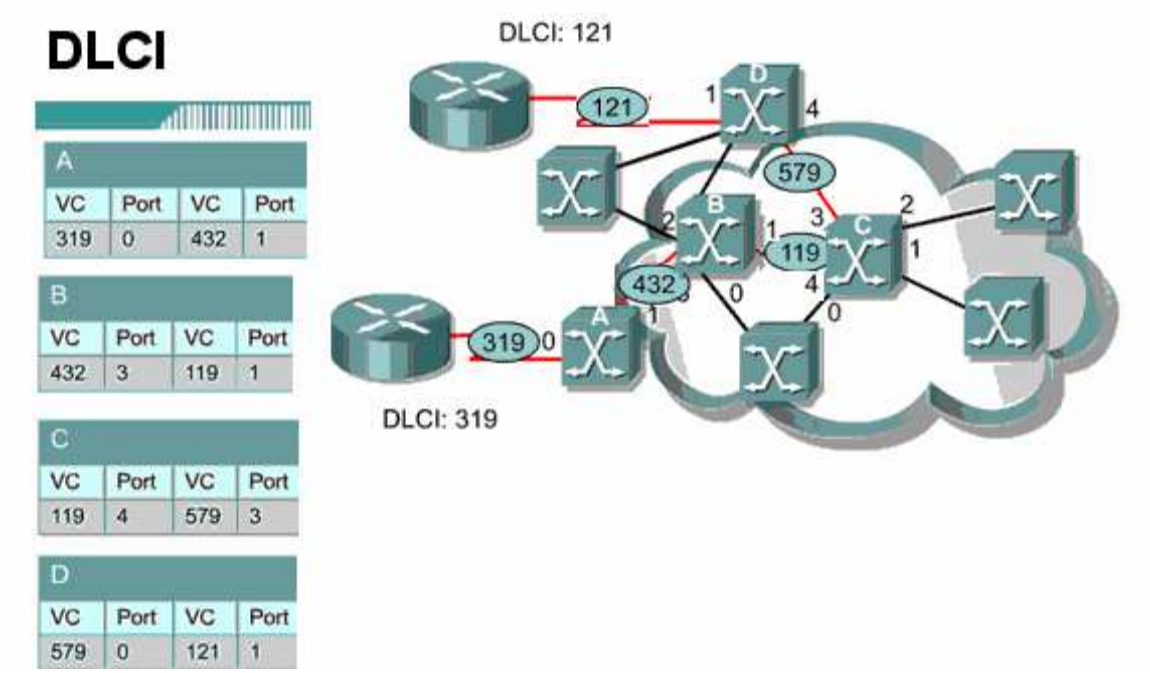
Bu headerların içeriğinde bizim için oneli olacak DLCI' lar vardır.



DLCI

Data Link Connection Identifier' in kisaltmasi olan DLCI musteri cihazi ve frame relay switch arasındaki sanal devreyi tanımlaya yarar.

DLCI numaraları servis saglayicalar tarafindan belirlenen mantiksal adreslerdir denebilir. 0-15 ve 1008 – 1023 arasında ki numaralar özel amaclar için ayrıldığından servis saglayicilar tarafindan 16-1007 arasındaki numaralardan seçilerek atama yapılir.



Şekilde A, B, C ve D switchleri üzerinde ki DLCI yönlendirmeleri incelendiğinde DLCI mantığı daha iyi anlaşılacaktır.

LMI

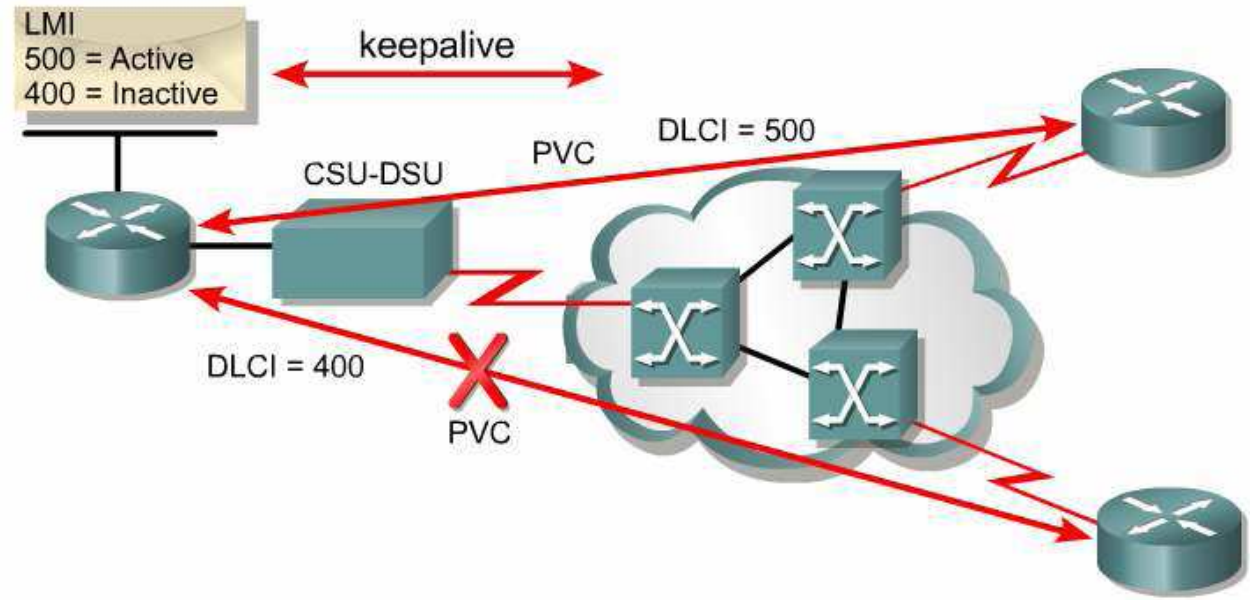
Local Management Interface (LMI) DTE cihazlar ve Frame Relay switchler arasındaki signaling standardidir.

Cisco Routerlar 3 ceşit LMI Type'i destekler.

1. Cisco

2. Ansi

3. q933a



Frame Relay networku ve DTE Router için LMI type aynı olmadığı takdirde çalışmayacaktır. Turkiyede kullanılan LMI Type Ansi' dir. Routerda Cisco IOS 11.2 ve üzeri varsa LMI Type tanımlaya gerek kalmaz, Router Frame Relay networkundeki LMI Type'i algilar.

Frame Relay Switchler konfigure edilen PVC' lerin durumlarını belirtmek için LMI' i kullanırlar. PVC' ler 3 ayi durumda olabilirler.

1. Active State: Routerların data transferi yapabildiği, bağlantının aktif olduğunun belirtildiği durumdur.

2. Inactive State: Frame Relay switch ile Localımiz arasında ki bağlantının aktif olduğu ama uzaktaki Router ile uzaktaki Frame Relay switch bağlantısinin duzgun çalışmadiği durumdur.

4. Deleted State: CPE ve Frame Relay switch arasında herhangi bir servisin çalışmadiği durumdur.

```
1w2d: Serial0/0 (in): Status, myseq 142
1w2d: RT IE 1, length 1 type 0
1w2d: KA IE 3, length 2 yourseg 142, myseq 142
1w2d: PVC IE 0x7, length 0x6, dlci 100, status 0x2, bw0
```

(debug frame-relay lmi)

Burada 0x2 aktif durumu gösterir, diğer durumlar şu şekildedir;

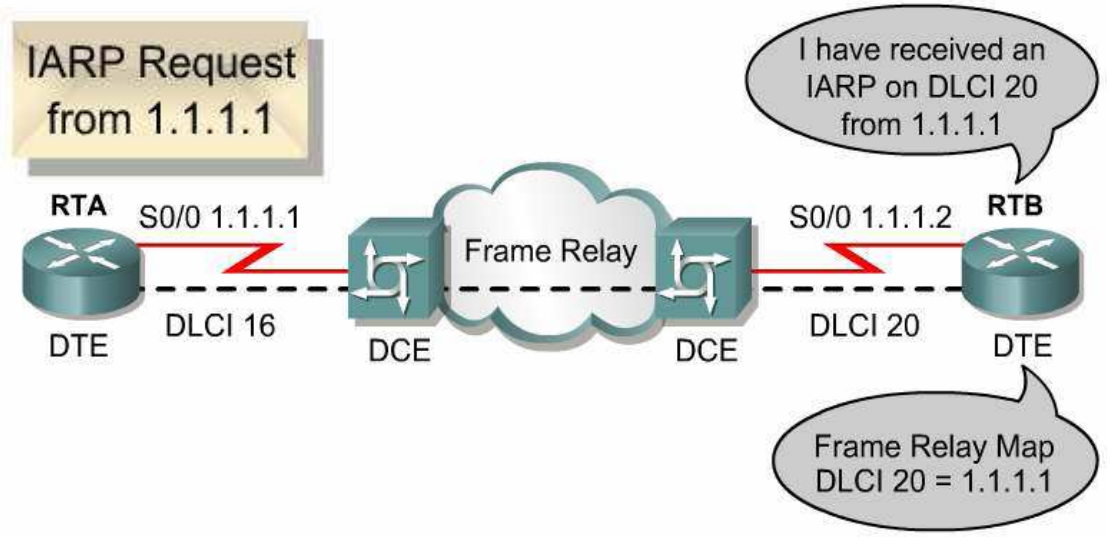
0x0: Inactive

0x4: Deleted

DLCI Mapping

Frame Relay networklerinin konfigurasyonu sırasında önemli bir adimda servis saglayiciların Frame Relay switchlerinde yol verdikleri DLCI numaralarının next hop 3. katman adreslerine map edilmesidir.

Burada mao islemi Dinamik ve static olmak üzere iki şekilde yapılabilir. Static map isleminde frame relay map komutu kullanılır. Dinamik map isleminde ise Inverse ARP protokolu çalışır. Burada Inverse ARP her DLCI için Inverse ARP Request mesaji gönderir ve aldığı cevap ile data-link katman adresi DLCI numarası ve Network Katmanı adresi Next Hop ip adresini map eder.



Kisaca Inverse ARP Lan' lardaki ARP protokolu gibi çalışır.

Static Map

Router (config-if) #frame-relay map protocol protocol-address

dlci [broadcast] [ietf l cisso]

Buradaki ip adresi remote ip adresi DLCI numarası ise local DLCI numarasıdır.

Router(config-if)#frame-relay map ip 10.1.1.1 101 broadcast

Dinamik Map

```
Router(config-if)# frame-relay interface-dlci dlci-number
```

Buradaki DLCI numarası local DLCI'dir.

Router(config-if)#frame-relay interface dlci 100

Frame Relay networklerinde Frame Relay encapsulation kullanılır. Cisco ve Ietf olmak üzere 2 ayrı standardi vardır, default olarak cisco' dur. Sistem de Cisco dışında Routerlar var ise RFC 1490 ile tanımlanmış Ietf standardi kullanılmalıdır.

```
RTB(config)#interface serial 0/0
RTB(config-if)#encapsulation frame-relay
RTB(config-if)#frame-relay map ip 131.108.123.2 48 broadcast
RTB(config-if)#frame-relay map ip 131.108.123.3 49 broadcast ietf
RTB(config-if)#frame-relay map ip 131.108.123.4 50 broadcast
```

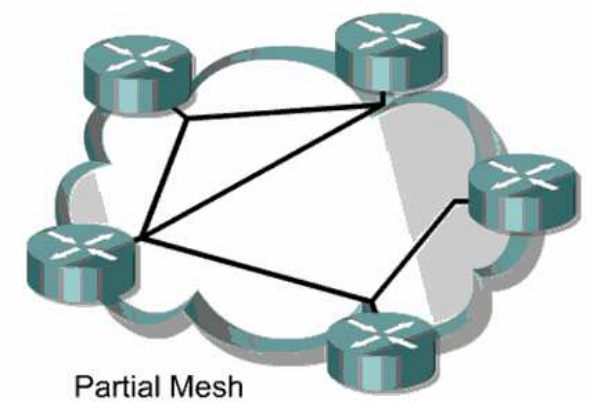
Encapsulation Frame Relay seçildikten sonra Frame Relay ao komutunda encapsulation seçilmeyebilir, bu durumda frame relay encapsulation geçerli olacaktır.

Frame Relay Map satırlarının ikincisinde ki gibi farklı bir encapsulation seçilirse geçerli olan o olacaktır. Örneğimiz de ikinci satır için geçerli olan encapsulation ietf'dir.

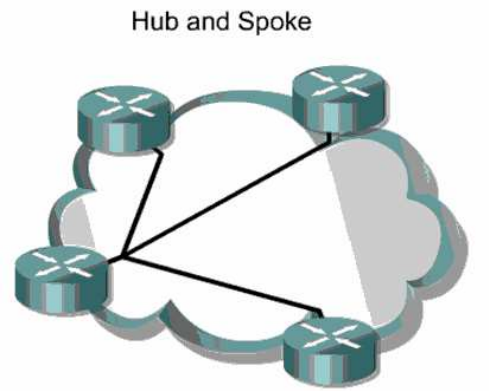
FRAME RELAY TOPOLOJİLERİ

Mesh Topolojide esas olan tüm noktalar arsında ayrı birer PVC olmasıdır. Oldukca pahali bir topolojidir. Fakat bağlantılardan biri down olduğunda bile bir çok alternatif yoldan hedefe ulaşılabilir.

Full Mesh ve Partial Mesh olarak ikiye ayrılır.



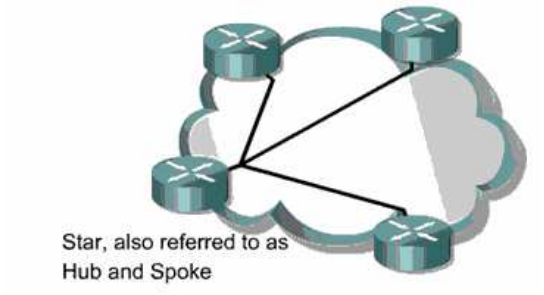
Hun and Spoke Topoloji en çok kullanılan Frame Relay topolojidir ve Star Topoloji olarak da anilir. Bu topoloji genellikle birden fazla uzak networkun merkezi bir routera baglanmasi ile şekillenir.

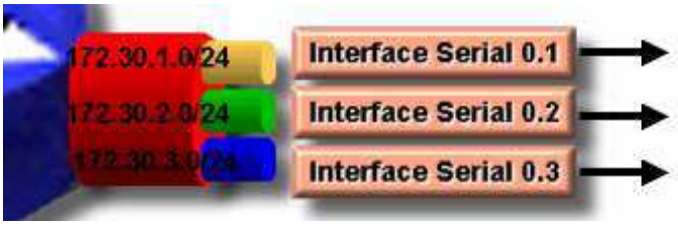


Burada merkez router multipoint bağlantı yada point to poitn bağlantı saglayabilir.

FRAME RELAY SUB-INTERFACE KONFİGURASYONU

Frame Relay networklerinde sözgelimi merkezde olan bir router birden fazla sayiya sahip subeye hizmet verebili, subeyle bağlantıyi sağlanabilir. Burda interfacein altında sub interfaceler oluşturmak gerekir.





Burada her sub-interface farklı bir networke ait ve farklı bir remote bağlantı içinde. Her bağlantı için ayrı PVC' ler mevcut.

Hun and Spoke olarak adlandirilan Frame Relay topolojilerinde kullanian bu yontem ile ilgili uygulama ilerde yapılacaktır. Şimdilik örnek olması açısından cisco.com ' dan alionan konfihurasyonu veriyorum.

```
RTA(config)#interface serial S0/0.1 multipoint
RTA(config-subif)#ip address 1.1.1.1 255.255.255.0
RTA(config-subif)#frame-relay interface-dlci 18
RTA(config-fr-dlci)#exit
RTA(config-subif)#frame-relay interface-dlci 19
RTA(config-fr-dlci)#exit
RTA(config-subif)#exit
RTA(config)#interface serial S0/0.2 point-to-point
RTA(config-subif)#ip address 2.1.1.1 255.255.255.0
RTA(config-subif)#frame-relay interface-dlci 20
RTA(config-fr-dlci)#^Z
```

Hub and Spoke topology Frame Relay networklerinin en çok kullanılan şeklidir. Point to multi point veya sub-interfaceler ile point to point olarak tasarlanabilir. Fakat bu topology, Point to multipoint networklerde routing islemi için Routing Protokoller kullanılmissa Split Horizon kuralından dolayı sorun yasatacaktır.

Cunku Split Horizon kuralı geregi bir Router aldığı update' i aldığı interfaceden geri göndermez. Bu durumda Split Horizon kuralı devre dışı birakilmalıdır.

Router (config-if) #no ip split-horizon

Split Horizon kuralı Link State protokolleri Örneğin OSPF protokolune etkilemez.

FRAME RELAY SHOW KOMUTLARI:

Asagidaki show komutları Cisco' nun CNAP egitimi için ongörduğu program slaytlarından alınmıştır.

```
Router#show frame-relay pvc 110

PVC Statistics for interface Serial0 (Frame Relay DTE)

DLCI = 110, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE,
INTERFACE =    Serial0
input pkts 14055       output pkts 32795    in bytes 1096228
out bytes 6216155      dropped pkts 0       in FECN pkts 0
in BECN pkts 0         out FECN pkts 0      out BECN pkts 0
in DE pkts 0           out DE pkts 0
out bcast pkts 32795   out bcast bytes 6216155
```

```
Router#show frame-relay map
Serial2 (up): IP 131.108.122.2 dlci 20(0x14,0x0440),
dynamic
          CISCO, BW= 56000, status defined, active
```

```
Router#show frame-relay lmi

LMI Statistics for interface Serial0 (Frame Relay DTE)
LMI TYPE =
   CISCO
Invalid Unnumbered info          0 Invalid Prot Disc 0
Invalid dummy Call Ref           0 Invalid Msg Type 0
Invalid Status Message           0 Invalid Lock Shift 0
Invalid Information ID           0 Invalid Report IE Len 0
Invalid Report Request           0 Invalid Keep IE Len 0
Num Status Enq. Sent 113100      Num Status msgs Rcvd 113100
Num Update Status Rcvd 0         Num Status Timeouts 0
```

```
show interface serial 0/0
Serial0 is up, line protocol is up
Hardware is CD2430 in sync mode
MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation FRAME-RELAY, loopback not set, keepalive set (10 sec)
LMI enq sent 112971, LMI stat recvd 112971, LMI upd recvd 0, DTE LMI up
LMI enq recvd 0, LMI stat sent 0, LMI upd sent 0
LMI DLCI 1023 LMI type is CISCO frame relay DTE
FR SVC disabled, LAPF state down
Broadcast queue 0/64, broadcasts sent/dropped 32776/0, interface broadcasts 1
Last input 00:00:00, output 00:00:03, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: weighted fair
<Output Omitted>
```

FRAME RELAY SWİTCH KONFİGURASYONU

Laboratur ortamında Frame Relay ugyulamalri için Frame Relay Switche ihtiyac vardır. Fakat Frame Relay switch olmadığı durumlarda bir Router Frame – Relay Switch olarak konfigure edilebilir.

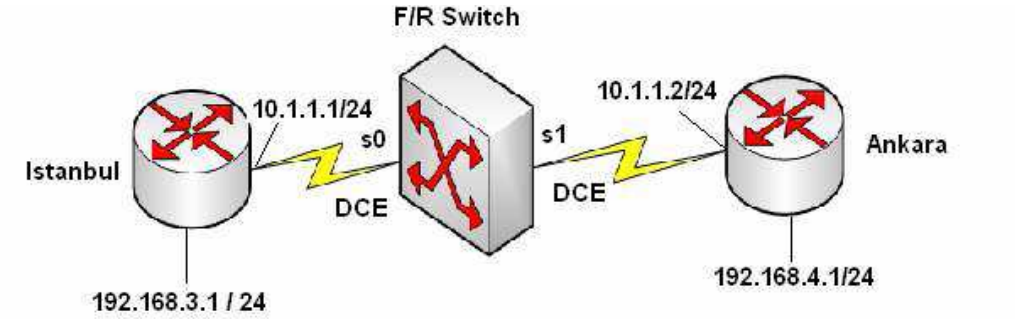
Bunun için Routerlara takilan DTE kablolara DCE kablolalar ile Frame Relay Switch olarak konfigure edilecek Router,a baglanir ve interfacelerine “clock rate” komutu verilir. Burada interfacelere interface type’ in DCE olduğuda söylenir.

Örnek:

```
interface  Serial0
        no ip address
        encapsulation frame-relay
        no fair-queue
        clockrate 64000
        frame-relay lmi-type ansi
        frame-relay intf-type dce
        frame-relay route 100 interface Serial1 101
```

FRAME RELAY POİNT-TO-POİNT KONFİGURASYONU

Frame Relay Konfigurasyonumuzda kullanacagimiz topoloji şu şekildedir;



Burada DLCI numaraları 5stanbul için 100, Ankara için 101`dir ve bir Router laboratuar ortamında Frame Relay Switch olarak konfigure edilmişitr. Ip adresleri atandiktan sonra, Frame Relay çalışma için;

İstanbul Routerinda;

```
Router(config)#interface Serial0/0
Router(config-if)#encapsulation frame-relay
Router(config-if)#frame-relay interface-dlci 100
Router(config-if)# frame-relay lmi-type ansi
```

Ankara Routerinda;

```
Router(config)#interface Serial0/1
Router(config-if)#encapsulation frame-relay
Router(config-if)# frame-relay interface-dlci 101
```

Router(config-if)#frame-relay lmi-type ansi
Konfigurasyonları yapılmıştır.

İstanbul Routeri Running-Config

```
sh running-config
Building configuration...
Current configuration : 636 bytes
!
version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
!
!
!
memory-size iomem 10
ip subnet-zero
!
!
interface Ethernet0/0
ip address 192.168.3.1 255.255.255.0
!
interface Serial0/0
ip address 10.1.1.1 255.255.255.0
encapsulation frame-relay
no fair-queue
frame-relay interface-dlci 100
frame-relay lmi-type ansi
!
interface BRI0/0
no ip address
shutdown
isdn x25 static-tei 0
!
```

ip classless
ip route 192.168.4.0 255.255.255.0 10.1.1.2
ip http server
!
--More--
line con 0
line aux 0
line vty 0 4
login
!
end

Ankara Routeri Running-Config

sh run
Building configuration...
Current configuration : 632 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
memory-size iomem 10
ip subnet-zero
!
!
interface Ethernet0/0
ip address 192.168.4.1 255.255.255.0
half-duplex
!
interface Serial0/0
no ip address
shutdown
!
interface Serial0/1

```
ip address 10.1.1.2 255.255.255.0
encapsulation frame-relay
frame-relay interface-dlci 101
frame-relay lmi-type ansi
!
ip classless
ip route 192.168.3.0 255.255.255.0 10.1.1.1
ip http server
!
!
line con 0
line aux 0
line vty 0 4
login
!
end
```

Frame Relay Switch Routeri Running-Config

```
sh run
Building configuration...
Current configuration:
!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname Router
!
!
frame-relay switching
!
interface Ethernet0
no ip address
shutdown
!
interface Serial0
```

```
no ip address
encapsulation frame-relay
no fair-queue
clockrate 64000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 100 interface Serial1 101
!
interface Serial1
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 101 interface Serial0 100
!
no ip classless
!
line con 0
line aux 0
line vty 0 4
login
!
end
```

İstanbul Routeri Frame Relay PCV ve Frame Relay Map

```
ISTANBUL#show frame-relay pvc

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

             Active     Inactive      Deleted       Static
  Local         1            0            0            0
  Switched      0            0            0            0
  Unused        0            0            0            0

DLCI = 100, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 36             output pkts 33           in bytes 2766
  out bytes 2644            dropped pkts 0           in FECN pkts 0
  in BECN pkts 0            out FECN pkts 0          out BECN pkts 0
  in DE pkts 0              out DE pkts 0
  out bcast pkts 2           out bcast bytes 68
  pvc create time 00:24:43, last time pvc status changed 00:22:03
ISTANBUL#show frame-relay map
Serial0/0 (up): ip 10.1.1.2 dlci 100(0x64,0x1840), dynamic,
             broadcast,, status defined, active
ISTANBUL#_
```

Ankara Routeri Frame Relay PCV ve Frame Relay Map

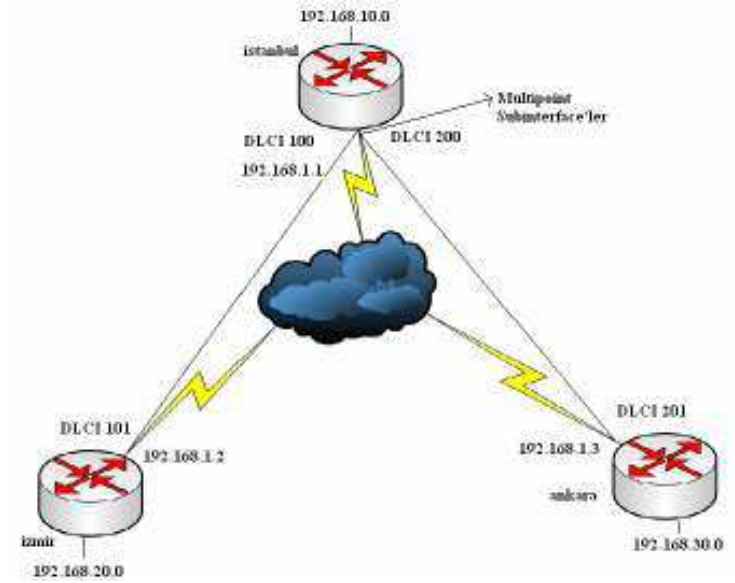
```
ANKARA#show frame-relay pvc

PVC Statistics for interface Serial0/1 (Frame Relay DTE)

             Active     Inactive      Deleted       Static
  Local         1            0            0            0
  Switched      0            0            0            0
  Unused        0            0            0            0

DLCI = 101, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/1

  input pkts 32             output pkts 32           in bytes 2610
  out bytes 2573            dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                  out bytes dropped 0
  in FECN pkts 0            in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0           in DE pkts 0             out DE pkts 0
  out bcast pkts 7          out bcast bytes 469
  pvc create time 00:23:13, last time pvc status changed 00:23:13
ANKARA#show frame-relay map
Serial0/1 (up): ip 10.1.1.1 dlci 101(0x65,0x1850), dynamic,
             broadcast,, status defined, active
ANKARA#_
```

Frame Relay Switch PCV ve Route

```
FRSW#show frame-relay route
Input Intf      Input Dlci      Output Intf     Output Dlci     Status
Serial0         100             Serial1         101             active
Serial1         101             Serial0         100             active
FRSW#_
```

```
Building configuration...

PVC Statistics for interface Serial0 (Frame Relay DCE)

DLCI = 100, DLCI USAGE = SWITCHED, PVC STATUS = ACTIVE, INTERFACE = Serial0

  input pkts 34             output pkts 36           in bytes 2678
  out bytes 2766            dropped pkts 1           in FECN pkts 0
  in BECN pkts 0            out FECN pkts 0          out BECN pkts 0
  in DE pkts 0              out DE pkts 0
  out bcast pkts 0           out bcast bytes 0
  pvc create time 00:28:58, last time pvc status changed 00:20:09
  Num Pkts Switched 34

PVC Statistics for interface Serial1 (Frame Relay DCE)

DLCI = 101, DLCI USAGE = SWITCHED, PVC STATUS = ACTIVE, INTERFACE = Serial1

  input pkts 37             output pkts 34           in bytes 2800
  out bytes 2678            dropped pkts 0           in FECN pkts 0
  in BECN pkts 0            out FECN pkts 0          out BECN pkts 0
  in DE pkts 0              out DE pkts 0
  out bcast pkts 0           out bcast bytes 0
  pvc create time 00:29:01, last time pvc status changed 00:22:59
  Num Pkts Switched 36
FRSW#_
```

FRAME RELAY HUB AND SPOKE MULTİPOİNT KONFİGURASYONU



Routerların Konfigurasyon Dosyaları

Router Istanbul

version 12.0

!

hostname Istanbul

!

interface Serial0

no ip address
encapsulation frame-relay
no frame-relay inverse-arp
!
interface Serial0.1 multipoint
ip address 192.168.1.1 255.255.255.0
no ip split-horizon
frame-relay interface-dlci 100
frame-relay interface-dlci 200
!
interface FastEthernet0
ip address 192.168.10.1 255.255.255.0
no ip directed-broadcast
no keepalive
!
router rip
network 192.168.0.0
!

Router İzmir

version 12.0
!
hostname Izmir
!
interface Serial0
no ip address
encapsulation frame-relay
frame-relay lmi-type ansi
!
interface Serial0.1 point-to-point
ip address 192.168.1.2 255.255.255.0
no ip directed-broadcast
frame-relay interface-dlci 101
!
interface FastEthernet0
ip address 192.168.20.1 255.255.255.0
no ip directed-broadcast

no keepalive
!
router rip
network 192.168.0.0
!

Router Ankara

version 12.0
!
hostname Ankara
!
interface Serial0
no ip address
encapsulation frame-relay
!
interface Serial0.1 point-to-point
ip address 192.168.1.3 255.255.255.0
frame-relay interface-dlci 201
!
interface FastEthernet0
ip address 192.168.30.1 255.255.255.0
no ip directed-broadcast
no keepalive
!
router rip
network 192.168.0.0
!

Frame Relay Switch

version 12.0
!
hostname FrameSwitchE
!
frame-relay switching
!
interface Serial0
no ip address

encapsulation frame-relay
clockrate 56000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 100 interface Serial1 101
frame-relay route 200 interface Serial2 201
!
interface Serial1
no ip address
encapsulation frame-relay
clockrate 56000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 101 interface Serial0 100
!
interface Serial2
no ip address
encapsulation frame-relay
clockrate 56000
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 201 interface Serial0 200

FRAME RELAY MAP VE PVC

FrameSwitchE#show frame-relay route

Input Intf	**Input Dlci**	**Output Intf**	**Output Dlci**	**Status**
Serial0	100	Serial1	101	active
Serial0	200	Serial2	201	active
Serial1	101	Serial0	100	active
Serial2	201	Serial0	200	active

Istanbul#show frame-relay map

Serial0.1 (up): ip 192.168.1.2 dlci 100(0x64,0x1840), dynamic,
broadcast,, status defined, active
Serial0.1 (up): ip 192.168.1.3 dlci 200(0xC8,0x3080), dynamic,
broadcast,, status defined, active

Istanbul#show frame-relay pvc

PVC Statistics for interface Serial0 (Frame Relay DTE)

	Active	Inactive	Deleted	Static
Local	2	0	0	0
Switched	0	0	0	0
Unused	0	0	0	0

DLCI = 100, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.1

input pkts 140 output pkts 77 in bytes 24656
out bytes 7774 dropped pkts 0 in FECN pkts 0
in BECN pkts 0 out FECN pkts 0 out BECN pkts 0
in DE pkts 0 out DE pkts 0
out bcast pkts 69 out bcast bytes 7038
pvc create time 00:31:06, last time pvc status changed 00:30:36

DLCI = 200, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.1

input pkts 128 output pkts 103 in bytes 18760
out bytes 11810 dropped pkts 0 in FECN pkts 0
in BECN pkts 0 out FECN pkts 0 out BECN pkts 0
in DE pkts 0 out DE pkts 0
out bcast pkts 72 out bcast bytes 8594
pvc create time 00:31:07, last time pvc status changed 00:30:37

Izmir#show frame-relay map

Serial0.1 (up): point-to-point dlci, dlci 101(0x65,0x1850), broadcast
status defined, active

Izmir#show frame-relay pvc

PVC Statistics for interface Serial0 (Frame Relay DTE)

	Active	Inactive	Deleted	Static
Local	1	0	0	0
Switched	0	0	0	0
Unused	0	0	0	0

DLCI = 101, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.1

input pkts 58 output pkts 65 in bytes 6252
out bytes 9602 dropped pkts 0 in FECN pkts 0
in BECN pkts 0 out FECN pkts 0 out BECN pkts 0
in DE pkts 0 out DE pkts 0
out bcast pkts 55 out bcast bytes 8562
pvc create time 00:20:34, last time pvc status changed 00:20:34

Ankara#show frame-relay map

Serial0.1 (up): point-to-point dlci, dlci 201(0xC9,0x3090), broadcast
status defined, active

Ankara#show frame-relay pvc

PVC Statistics for interface Serial0 (Frame Relay DTE)

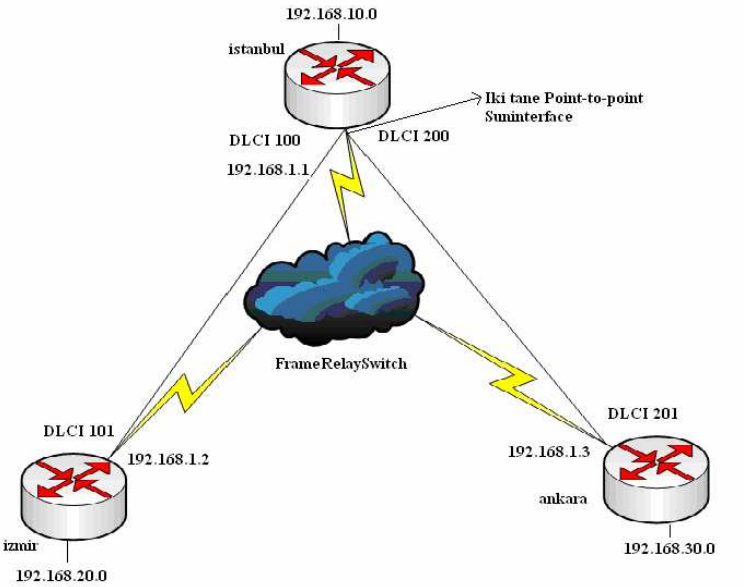
	Active	Inactive	Deleted	Static
Local	1	0	0	0
Switched	0	0	0	0
Unused	0	0	0	0

DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.1

input pkts 59	output pkts 78	in bytes 6484
out bytes 9496	dropped pkts 0	in FECN pkts 0
in BECN pkts 0	out FECN pkts 0	out BECN pkts 0
in DE pkts 0	out DE pkts 0	
out bcast pkts 63	out bcast bytes 7936	

pvc create time 00:19:30, last time pvc status changed 00:19:30

FRAME RELAY HUB AND SPOKE POİNT-TO-POİNT KONFİGURASYONU



Router Konfigurasyon Dosyaları

Router Istanbul

version 12.0

```
!
hostname Istanbul
!
interface Serial0
no ip address
encapsulation frame-relay
no frame-relay inverse-arp
!
interface Serial0.1 point-to-point
ip address 192.168.1.1 255.255.255.0
frame-relay interface-dlci 100
!
interface Serial0.2 point-to-point
ip address 192.168.2.1 255.255.255.0
frame-relay interface-dlci 200
!
interface FastEthernet0
ip address 192.168.10.1 255.255.255.0
no keepalive
!
router rip
network 192.168.0.0
!
```

Router İzmir

```
version 12.0
!
hostname Izmir
!
interface Serial0
ip address 192.168.1.2 255.255.255.0
encapsulation frame-relay
frame-relay map ip 192.168.1.1 101 broadcast
no frame-relay inverse-arp
frame-relay lmi-type ansi
!
interface FastEthernet0
ip address 192.168.20.1 255.255.255.0
```

no keepalive
!
router rip
network 192.168.0.0

Router Ankara

version 12.0
!
hostname Ankara
!
interface Serial0
ip address 192.168.2.2 255.255.255.0
encapsulation frame-relay
frame-relay map ip 192.168.2.1 201 broadcast
!
interface FastEthernet0
ip address 192.168.30.1 255.255.255.0
no ip directed-broadcast
no keepalive
!
router rip
network 192.168.0.0
!

Frame Relay Switch

version 12.0
!
hostname FrameSwitch
!
frame-relay switching
!
interface Serial0
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay lmi-type ansi
frame-relay intf-type dce

frame-relay route 100 interface Serial1 101

frame-relay route 200 interface Serial2 201

!

interface Serial1

no ip address

encapsulation frame-relay

clockrate 64000

frame-relay lmi-type ansi

frame-relay intf-type dce

frame-relay route 101 interface Serial0 100

!

FrameSwitch#show frame-relay route

Input Intf	Input Dlci	Output Intf	Output Dlci	Status
Serial0	100	Serial1	101	active
Serial0	200	Serial2	201	active
Serial1	101	Serial0	100	active
Serial2	201	Serial0	200	active

FRAME RELAY MAP VE PVC

Istanbul#show frame-relay map

Serial0.1 (up): point-to-point dlci, dlci 100(0x64,0x1840), broadcast

status defined, active

Serial0.2 (up): point-to-point dlci, dlci 200(0xC8,0x3080), broadcast

status defined, active

Istanbul#show frame-relay pvc

PVC Statistics for interface Serial0 (Frame Relay DTE)

	Active	Inactive	Deleted	Static
Local	2	0	0	0
Switched	0	0	0	0
Unused	0	0	0	0

DLCI = 100, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.1

input pkts 123	output pkts 140	in bytes 23474
out bytes 25102	dropped pkts 0	in FECN pkts 0
in BECN pkts 0	out FECN pkts 0	out BECN pkts 0
in DE pkts 0	out DE pkts 0	

out bcast pkts 120 out bcast bytes 23022

pvc create time 00:26:26, last time pvc status changed 00:24:46

DLCI = 200, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.2

input pkts 89 output pkts 135 in bytes 14992

out bytes 25487 dropped pkts 0 in FECN pkts 0

in BECN pkts 0 out FECN pkts 0 out BECN pkts 0

in DE pkts 0 out DE pkts 0

out bcast pkts 121 out bcast bytes 23536

pvc create time 00:26:28, last time pvc status changed 00:24:08

Izmir#show frame-relay map

Serial0 (up): ip 192.168.1.1 dlci 101(0x65,0x1850), static,

broadcast,

CISCO, status defined, active

Izmir#show frame-relay pvc

PVC Statistics for interface Serial0 (Frame Relay DTE)

	Active	Inactive	Deleted	Static
Local	1	0	0	0
Switched	0	0	0	0
Unused	0	0	0	0

DLCI = 101, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0

input pkts 155 output pkts 129 in bytes 26714

out bytes 22108 dropped pkts 0 in FECN pkts 0

in BECN pkts 0 out FECN pkts 0 out BECN pkts 0

in DE pkts 0 out DE pkts 0

out bcast pkts 107 out bcast bytes 19820

pvc create time 00:33:33, last time pvc status changed 00:31:23

Ankara#show frame-relay map

Serial0 (up): ip 192.168.2.1 dlci 201(0xC9,0x3090), static,

broadcast,

CISCO, status defined, active

Ankara#show frame-relay pvc

PVC Statistics for interface Serial0 (Frame Relay DTE)

	Active	Inactive	Deleted	Static
Local	1	0	0	0
Switched	0	0	0	0
Unused	0	0	0	0

DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0

input pkts 172 output pkts 108 in bytes 30389

out bytes 14884 dropped pkts 0 in FECN pkts 0

in BECN pkts 0 out FECN pkts 0 out BECN pkts 0

in DE pkts 0 out DE pkts 0

out bcast pkts 87 out bcast bytes 12728

pvc create time 00:37:06, last time pvc status changed 00:35:16

I S D N

ISDN(Integrated Services Diğital Network) var olan telefon agı üzerinden sayısal hizmet vermek için geliştirilen bir teknolojidir. ISDN hat üzerinden ses, görüntü ve veri es zamanlı olarak iletilebilir.

POTS un (Plain Old Telephone Service) aksine ISDN end-to-end dijitaldir. Dolayısıyla ISDN ile birlikte PCM'e (Pulse Code Modulation) ihtiyac yoktur.

ISDN'in Avantajları:

ISDN dial-ip bağlantılardan daha geniş bant genişliği saglar.

Dial-up maodemlerden daha hizli cevrim saglar.

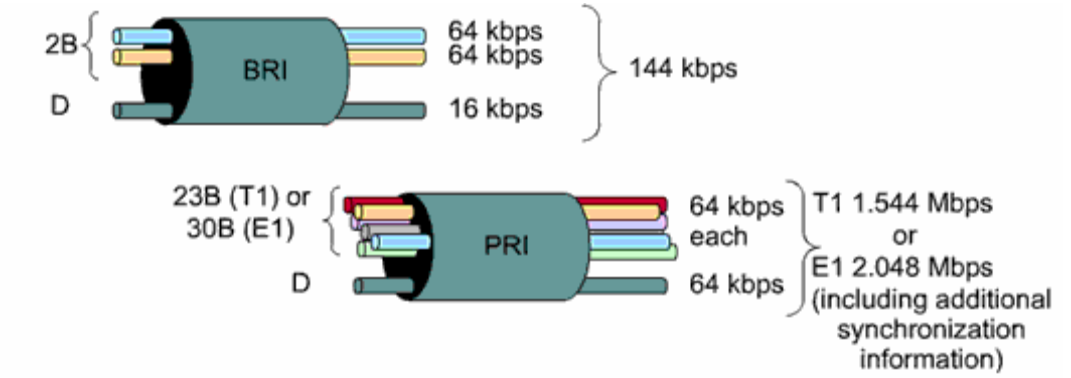
PPP encapsulation ile birlikte kullanılabilir.

ISDN'in Dezavantajları:

ISDN DSL veya kabloya göre daha yavas ve daha pahalidir.

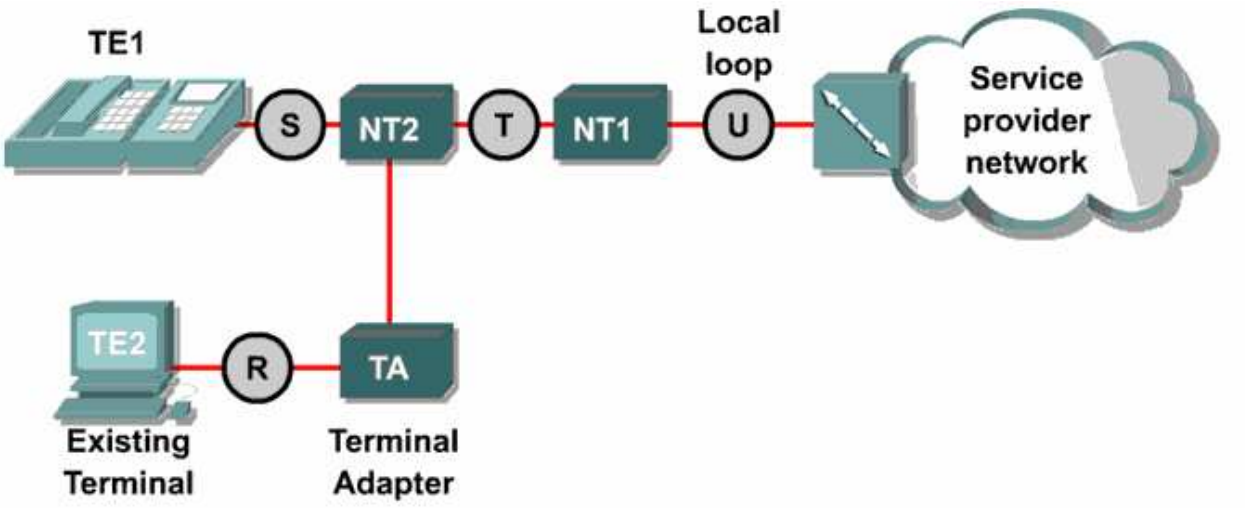
ISDN KANALLARI

ISDN iki tür hat içerir. **BRI (Basic Rate Interface) ve PRI (Primary Rate Interface).** Hem BRI da hem de PRI da B kanalları ve D kanalları mevcuttur. B kanalları veri tasımak için kullanılır. D kanalları kontrol ve isaretleme bilgisi taşır. BRI hatlarda 2 adet 64 Kbps 'lik B kanalı ve bir adet 16 Kbps ' lik D kanalı mevcuttur. T1 çerçevelemesini temel alan PRI ' lar 23 B+D ve E1 çerçevelemesini temel alan PRI 'lar 30 B+D olarak ifade edilir. 23 B+D 'ler Amerika'da ve 30 B+D 'ler ise Avrupa'da kullanılmaktadır.



Arayüz Türü B	Kanalları D	Kanalları	Açıklayıcı Terim
BRI	**2**	**1**	**2B+D**
PRI (T1)	**23**	**1**	**23B+D**
PRI (E1)	**30**	**1**	**30B+D**

ISDN Layer 1



TE1 : Bu sınıftaki cihazlar direkt olarak ISDN agına baglanabilir.

TE2 : Bu sınıftaki cihazlar ISDN standartlarını anlamazlar. ISDN agına baglanabilmeleri için bir terminal adaptör (TA) ‘ e ihityaç duyarlar.

NT1 : Fiziksel katman özelliklerini tanımlar. Cihazları ISDN agına baglar.

NT2 : Servis saglayıcı cihazlardır.

TA : T2 kablolamasını T1 kablolamasına dönüstürür.

(Bri kart)

(Terminal Adapter)

BRI konfigüre ederken her kanal için verilen SPID (Service Profile Identifer) numarasına ihtiyaç vardır. SPID ‘ler kullandığımız telefon numaralarına benzer. Internet Servis Saglayıcısından bize verilen SPID numaralarını **“isdn spid1”** ve “**isdn spid2”** komutlarını kullanarak girebiliriz. Ayrıca konfigüre ederken servis saglayıcının kullandığı switch türünü de router üzerinde belirtmemiz gerekiyor. Kullandığımız router ‘ın ne tür switchlere destek verdiğini görebilmek için **“isdn switch-type** ?” komutu kullanılabilir. (Turkiyede basic-net3 kullanılmaktadir.)

```
Router(config-if)#isdn spid1 spid - numarasi
Router(config-if)#isdn spid2 spid - numarasi
```

PPP ve CHAP authentication kullanımı;

```
Gateway(config)#username ISP password class
Gateway(config)#isdn switch-type basic-dms100

Gateway(config)#interface bri 0
Gateway(config-if)#ip add 10.0.0.3 255.0.0.0
Gateway(config-if)#encapsulation ppp
Gateway(config-if)#ppp authen chap
Gateway(config-if)#isdn spid1 08443  213
Gateway(config-if)#isdn spid2 08132  344
```

```
R1#show isdn status
Global ISDN Switchtype = basic-ni
ISDN BRI0/0 interface
dsl 0, interface ISDN Switchtype = basic-ni
    Layer 1 Status:
ACTIVE
    Layer 2 Status:
TEI = 64, Ces = 1, SAPI = 0, State =
MULTIPLE_FRAME_ESTABLISHED
TEI = 65, Ces = 2, SAPI = 0, State =
MULTIPLE_FRAME_ESTABLISHED
    Spid Status:
TEI 64, ces = 1, state = 5(init)
    spid1 configured, no LDN, spid1 sent, spid1 valid
    Endpoint ID Info: epsf = 0, usid = 70, tid = 1
TEI 65, ces = 2, state = 5(init)
    spid2 configured, no LDN, spid2 sent, spid2 valid
```

```
R2#show interface bri0/0.1
BRI0:1 is up, line protocol is up
  Hardware is BRI
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely
255/255, load 1/255
  Encapsulation PPP, loopback not set, keepalive set
(10 sec)
  LCP Open
  Open: IPCP, CDPCP
  Last input 00:00:01, output 00:00:01, output hang
never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output
drops: 0
```

PRI konfigüre ederken öncelikle PRI kartın takılı olduğu port a girilir. Daha sonra “framing” komutuyla servis saglayıcı tarafından belirlenen frame türü belirlenir. Daha sonra sabit timeslots numaraları konfigüre edilir. T1 için timeslots aralıgı 1-24 ve E1 için timeslots aralıgı 1-31 dir. Kullanılan “linecode” komutuyla fiziksel katmandaki sinyal modeli seçilir. Bu sinyal modellerinden HDB3 Amerika da B8ZS ise Kuzey Amerika da kullanılmaktadır. Son olarak router ‘ın üzerindeki T1/E1 seri interface ‘ine girilir. E1 için 1 ile 31 arası ve T1 için 1 ile 24 arasıdır. Bu frame relay de kullanılan subinterface gibi algılanmamalıdır. Çünkü frame relay da interface serial 0/0.16 şeklinde bit tanımlama bulunmaktaydı fakat PRI da ise interface serial 0/0:23 şeklinde bir tanımlama yapılacaktır. Bu tanımlamayla bir kanal açılacaktır.

PRI T1 konfigürasyonu ;

Router(config)#controller t1 1/0

Router(config-controller)#framing esf

Router(config-controller)#linecode b8zs

Router(config-controller)#pri-group timeslots 1-24

Router(config-controller)#interface serial3/0:23

Router(config-if)#isdn switch-type primary-5ess

Router(config-if)#no cdp enable

PRI E1 konfigürasyonu

Router(config)#controller e1 1/0

Router(config-controller)#framing crc4

Router(config-controller)#linecode hdb3

Router(config-controller)#pri-group timeslots 1-31

Router(config-controller)#interface serial3/0:15

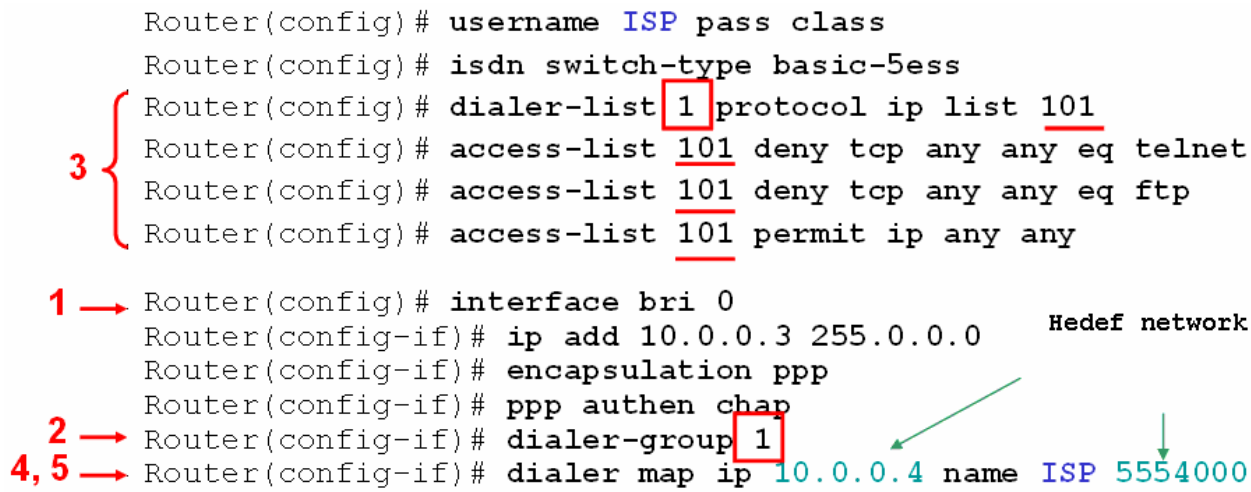
Router(config-if)#isdn switch-type primary-net5

Router(config-if)#no cdp enable

D D R

DDR (Dial-on-Demand Router) iki veya daha fazla Cisco router' ın ISDN dial up baglantı yapmasını saglar. Genellikle PSTN veya ISDN kullanılarak gerçeklesen periyodik network baglantılarında kullanılır. Böylece gerek duyulunca baglantı gerçeklesir ve ödenecek ücret azalacaktır.

DDR baglantı konfigürasyonu yapılırken öncelikle baglantı kurulacak interface içerinde ip adresi tanımlaması yapılır. Daha sonra static bir yönlendirme yapılır. Son olarak "dialer-list" komutu kullanılarak oluşturulan liste hangi tür paketlerin bu baglantıyı aktif yapacagı belirlenir. Ve network baglantısında kullanılacak arama bilgileri konfigüre edilir. Asagidaki çalışma incelendiğinde DDR in çalışma mantığı daha iyi anlaşılacaktır.



3. Routing table ilgili trafiğin bri 0; üzerinden olacağını gösterdiği için bu interface in konfigurasyonu kontrol edilir.

4. Router bu interface deki “dialer-group 1” komutundan aynı id numarasına sahip dilaer-list den bu trafige izin verilip verilmeyeceğinin arastirilmasi gerektigini anlar.

5. Bu trafige izin verilip verilmeyeceği ilgili “dialer-list 1 protocol ip list 101” de belirtilen 101 numaralı access list ile kararlastirilir. 6. Trafige izin verilecek ise next hopu bulmak için dilaer map’ e basvurulur.

7. Dialer map kullanimdaysa data gönderilir, kullanimda değilse call setup islemi başlar.

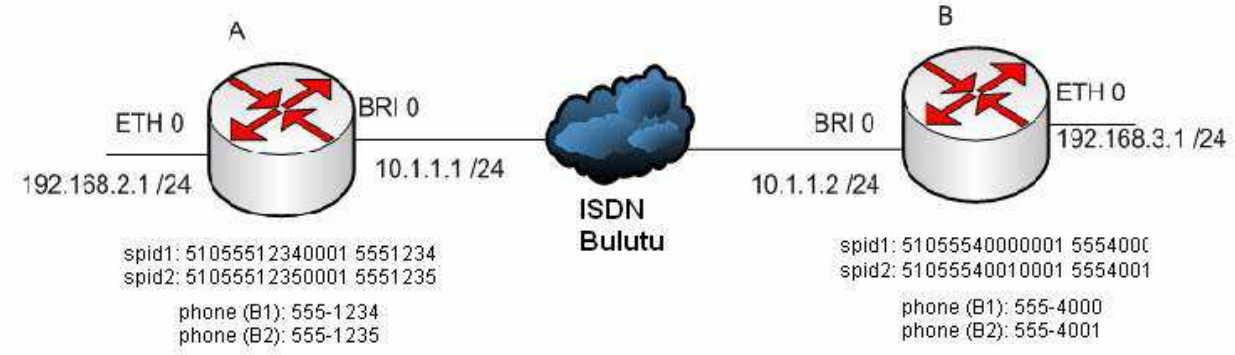
Burada Artık bir kez bağlantı kurulduktan sonra access list ile belirlenen kriterlere uymayan paketler de gönderilecektir. Fakat sadece bu kriterlere uyan paketler konfigurasyona eklenebilecek iddle-time süresini resetleyecektir.

```
hostname Home
!
isdn switch-type basic-5ess
!
username Central password cisco
interface BRI0
 ip address 10.1.0.1 255.255.255.0
 encapsulation ppp
 dialer idle-timeout 180
 dialer map ip 10.1.0.2 name Central 5552000
 dialer-group 1
 no fair-queue
 ppp authentication chap
!
```

DİALER LOAD-THRESHOLD KOMUTU

“dialer load-threshold” komutu BRI interface ‘inin ikinci B kanalının ne zaman aktif olacagını söyler. Parametre olarak 1 ile 255 arası bir değer alır. Eğer 255 kullanılırsa birinci B kanalı %100 kullanıldığında ikinci B kanalı aktif edilir. İkinci bir parametre olarak “in” gelen trafiği , “out” giden trafiği, “either” her ikisinin hesaplanacagını router’ a bildirir. “dialer idle-timeout” komutu en son iletilen paketin ardından ne kadar süre sonra baglantının kopacagını belirtmektedir.

ISDN Konfigurasyon Örneği



ISDN konfigurasyon örneği içerişinde SPID numaraları ve telefon numaraları kullanimistir.

B kanallarının her ikisi birlikte kullanılacagi için her iki kanal için de telefon numaraları ve SPID numaraları verilmistir.

Her iki Router ‘ da ISDN networkune BRI 0 portlarından baglanmistir.

Konfigurasyon içerişinde ppp authentication chap kullanılmistir.

Yönlendirme için IGRP konfigurasyonu kullanılmistir ve IGRP için AS numarası 100 olarak seçilmiştir.

RouterA

```
version 12.0
hostname RouterA
!
enable password cisco
!
username RouterB password 0 cisco
!
ip host RouterB 192.168.3.1
!
isdn switch-type basic-ni
!
interface FastEthernet0/0
ip address 192.168.2.1 255.255.255.0
no ip directed-broadcast
!
interface BRI0/0
ip address 10.1.1.1 255.255.255.0
encapsulation ppp
dialer map ip 10.1.1.2 name RouterB 5554000
dialer-group 1
isdn switch-type basic-ni
isdn spid1 51055512340001 5551234
isdn spid2 51055512350001 5551235
ppp authentication chap
!
router igrp 100
passive-interface BRI0/0
network 10.0.0.0
network 192.168.2.0
```

```
!
ip route 0.0.0.0 0.0.0.0 10.1.1.2
!
dialer-list 1 protocol ip permit
!
end
```

RouterB

```
version 12.0
hostname RouterB
!
enable password cisco
!
username RouterA password 0 cisco
!
isdn switch-type basic-ni
!
interface BRI0
ip address 10.1.1.2 255.255.255.0
encapsulation ppp
dialer map ip 10.1.1.1 name RouterA 5551234
dialer-group 1
isdn switch-type basic-ni
isdn spid1 51055540000001 5554000
isdn spid2 51055540010001 5554001
ppp authentication chap
!
interface FastEthernet0
ip address 192.168.3.1 255.255.255.0
no ip directed-broadcast
!
router igrp 100
passive-interface BRI0
network 10.0.0.0
network 192.168.3.0
!
ip route 0.0.0.0 0.0.0.0 10.1.1.1
```

!
dialer-list 1 protocol ip permit
!
End

RouterA#**show inter bri 0**
BRI0 is up, line protocol is **up (spoofing)**
Hardware is PQUICC **BRI with U interface**
Internet address is **10.1.1.1/24**
MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, loopback not set
Last input 00:00:08, output never, output hang never

show isdn status
RouterA#**show isdn status**
Global ISDN Switchtype = basic-ni
ISDN BRI0 interface
    dsl 0, interface ISDN **Switchtype = basic-ni**
Layer 1 Status:
    ACTIVE
Layer 2 Status:
    TEI = 64, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
    TEI = 65, Ces = 2, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
Spid Status:
    TEI 64, ces = 1, state = 5(init)
        spid1 configured, spid1 sent, **spid1 valid**
        Endpoint ID Info: epsf = 0, usid = 70, tid = 1
    TEI 65, ces = 2, state = 5(init)
        spid2 configured, spid2 sent, **spid2 valid**
        Endpoint ID Info: epsf = 0, usid = 70, tid = 2
Layer 3 Status:
    1 Active Layer 3 Call(s)
Activated dsl 0 CCBs = 1
    CCB:callid=8031, sapi=0, ces=1, B-chan=1, calltype=DATA
The Free Channel Mask: 0x80000002
Total Allocated ISDN CCBs = 1

RouterA#show dialer

BRI0 - dialer type = ISDN

Dial String	**Successes**	Failures	Last DNIS	**Last status**
5554000	1	8	00:02:49	**successful**

0 incoming call(s) have been screened.
0 incoming call(s) rejected for callback.

BRI0:1 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up
Dial reason: ip (s=10.1.1.1, d=192.168.3.1)
Time until disconnect 70 secs
Connected to 5554000 (denver)

BRI0:2 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle

show isdn active
RouterA#**show isdn active**

ISDN ACTIVE CALLS

Call Calling **Type** Number	**Called Number**	**Remote Name**	Seconds Used	Seconds Left	Seconds Idle	Charges Units/Currency
Out	5554000	RouterB	177	62	57	0

RouterA#**debug isdn events**
ISDN events debugging is on

RouterA#**ping Denver**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
00:14:04: ISDN BR0: Outgoing call id = 0x8032, dsl 0
00:14:04: ISDN BR0: Event: Call to 5554000 at 64 Kb/s
00:14:04: ISDN BR0: process_bri_call(): call id 0x8032, called_number 5554000, speed 64, call type DATA
00:14:21474836479: CC_CHAN_GetIdleChanbri: dsl 0
00:14:17179869184: Found idle channel B1
00:14:19335326197: ISDN BR0: received HOST_PROCEEDING call_id 0x8032
00:14:17179869184: ISDN BR0: received HOST_CONNECT call_id 0x8032
00:14:17179869232: %LINK-3-UPDOWN: Interface BRI0:1, changed state to up
00:14:17179869248: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 5554000
00:14:19337989260: ISDN BR0: Event: Connected to 5554000 on B1 at 64 Kb/s
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 32/32/32 ms
RouterA#
00:14:05: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed state to up
00:14:10: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 5554000
RouterB

Kaynaklar

1. Hayrullah KOLUKISAOGLU (CCNA Dökümantasyon Çalısması)
2. Kenan YORULMAZ & Mustafa ÇOPUR (7 Katmanlı OSI Modeli)
3. MEB DSL ve Ağ Teknolojileri Eğitimi
4. Mustafa ÇOPUR (Network Eğitimi Notları)
5. Mustafa ÇOPUR (TCP/IP)
6. MEB Eğitim Teknolojileri Bilgisayar 2003
7. www.cizgi.com.tr
8. www.ciscotr.com/forum

Not: Kaynağın ana omurgasını Hayrullah KOLUKISAOGLU (CCNA Dökümantasyon Çalısması) oluşturmaktadır. Yazım hataları düzeltmeleri, dizgi ve kaynak taraması tarafımdan gerçekleştirilmiştir. Emeği geçen herkese teşekkür ederim.

Ahmet TUNALI

Mart 2008