TC.
FIRAT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
ELEKTRİK-ELEKTRONİK BÖLÜMÜ

SWİTCH, ROUTER VE FREWALL KONFİGÜRASYONU

BİTİRME ÖDEVİ

DANIŞMAN
Yrd.Doç.Dr. Hasan H. BALIK

HAZIRLAYAN
Mahmut Ergünöz

ELAZIĞ-2003

TC.
FIRAT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

Switch, Router ve Frewall Konfigürasyonu

Mahmut Ergünöz

Bitirme Ödevi

Elektrik-Elektronik Bölümü

Bu tez,……. tarihinde aşağıda belirtilen juri tarafından oybirliği /oyçokluğu ile başarılı/başarısız olarak değerlendirilmiştir.

Danışman:Yrd. Doç. Dr. Hasan Hüseyin Balık

Üye:

Üye:

Üye:

Bu tezin kabülü Mühendislik Fakültesi Elektrik Elektronik Yönetim Kurulu ‘nun ……. tarih ve kararıyla onaylanmıştır.

İÇİNDEKİLER:

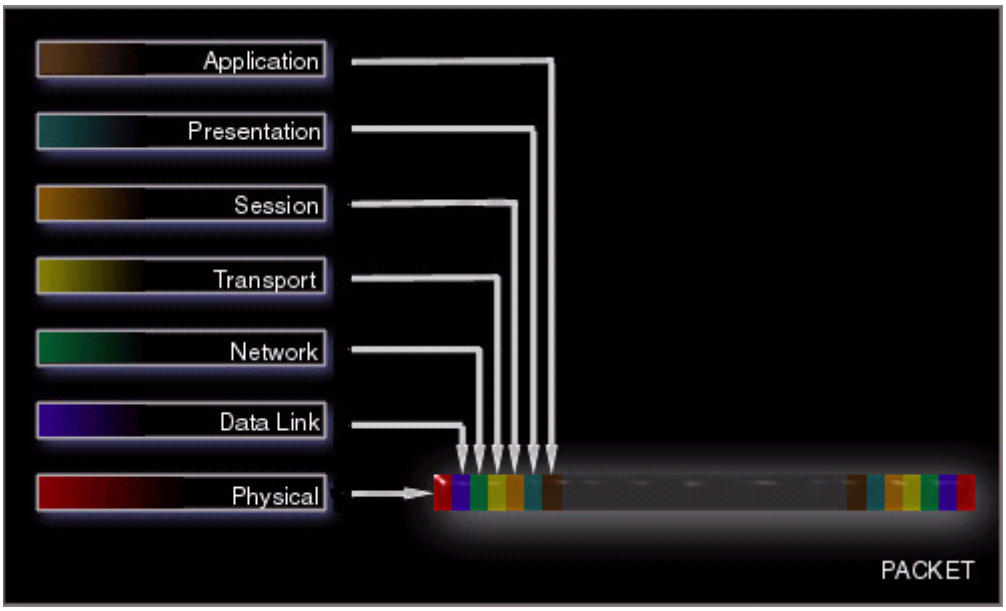
NETWORK DİZAYNI

Network : Bilgisayar ve benzeri sayısal sistemlerin belirli bir protokol altında iletişimde bulunması sağlayan sistemdir. Network, sahip olunan sayısal kaynakların paylaşılması için en önemli araçtır. İki bilgisayar ve HUB ile bir ağ oluşturulduğu gibi milyonlarca bilgisayarı kapsayan İnternet de bir ağdır.

1.1.OSI(Open Systems Interconnection) REFERANS MODELİ

Bir İnternetwork'te kullanıcıdan kullanıcıya bilgi iletişimi yapılırken her katmanın kendine özgü fonksiyonları vardır. OSI referans modeli ISO tarafından tanımlanmış ve ağ uygulamasında kullanılan örnek bir modeldir; her ne kadar pratikte bire bir uygulanmasa da, diğer tüm mimariler OSI başvuru modeli baz alınarak açıklanır.

Bildiğimiz gibi OSI Model 7 katmana sahiptir.



7 Uygulama Katmanı (Application Layer): Kullanıcıların çalıştırdığı uygulama programları bu katmanda tanımlıdır. İnternet Explorer, NetsCape, ftp, e-Mail. İnternet referans modelinin en üst katmanı uygulama katmanıdır. Bu katman kullanıcılar veya programları için fonksiyonlar sağlar ve yüksek derecede gerçekleştirilen uygulamalara özeldir. Kullanıcı uygulamalarının ağ üzerinde haberleşmek için kullandığı hizmetleri sağlar ve kullanıcı ağ erişim süreçlerinin bulunduğu katmandır. Bu süreçler kullanıcıların doğrudan etkileştiği uygulamaları içerdiği gibi, kullanıcıların haberi olmadığı süreçleri de içerir. Bu katman taşıma katmanı protokollerinin veriyi iletmek için kullandığı tüm uygulama protokollerini içerir. Kullanıcı verisini işleyen diğer süreçler, veri kriptolama, kripto çözme, sıkıştırma, sıkıştırılmış veriyi açma da uygulama katmanında bulunur.

Uygulama katmanı birlikte çalışan uygulamalar arasındaki oturumları yönetir. TCP/IP protokol hiyerarşisinde, oturumlar ayrı bir katman olarak tanımlanamazlar ve bu fonksiyonlar ulaştırma katmanı tarafından gerçeklenir. *Oturum* terimini kullanmak yerine TCP/IP birlikte çalışan uygulamaların haberleştikleri yolu (virtual devreyi) tanımlamak için *socket* ve *port'*u

kullanır. Bu katmandaki uygulama protokollerinin çoğu kullanıcı hizmetleri sağlar ve yeni kullanıcı hizmetleri sıkça eklenmektedir. Birlikte çalışan uygulamaların veri alış-verişi için verinin temsil biçimi üzerinde anlaşmaları gerekir. Uygulama katmanı verinin sunuşunun standardize edilmesinden sorumludur.

6 Sunum Katmanı (Presentation Layer):Bilgi iletilirken hangi biçimde yollanacaksa bunun dü- zenlenmesini sağlar; EBCDIC-ASCII dönüşümü-ters dönüşümü Sıkıştırma/Açma, şifreleme çözme gibi işlevleri vardır. Dosyaları programcıların istediği şekilde biçimlendirir. Başarılı bir transfer için datanın iletimden önce standart bir forma dönüştürülmesi gerekir. İletilecek bilgi resim, hareketli resim, MPEG v.b. olabilir. Sunma katmanı kontrol kodlarının, özel grafiklerin ve karakter tablolarının bulunduğu yerdir. Sunma katmanı yazılımı; yazıcıları, ve diğer aygıtları kontrol eder.

5 Oturum Katmanı (Session Layer): Kullanıcıdan kullanıcıya gerekli oturum kurulması yöneltilmesi ve sonlandırılması işlerini kapsar. İletişimin mantıksal sürekliliğinin sağlanması için, iletişimin kopması durumunda bir senkronizasyon noktasından başlayarak iletimin kaldığı yerden devam etmesini sağlar. Oturum katmanı düğümler veya aygıtlar arasındaki dialoğu kontrol eder. Sistemler arasındaki iletişimin hangi modda olacağını (simplex, half-duplex, full-duplex) koordine eder. Bu katman farklı uygulamalara ait dataları diğer uygulamalara ait datalardan ayırır.

4 Ulaşım Katmanı (Transport Layer):Bilginin alıcıda her tür hatadan arındırılmış olarak elde edilebilmesini sağlar. Ulaşım katmanı bilgi blokları oluşturur. Bu bloklara Segmet (bölüm) denir. Ağ katmanının yaptığı işleri yapar. Farkı bu işleri yerel olarak yapar. Ağ yazılımındaki sürücüler taşıma katmanının görevlerini yerine getirirler. Ağda bir arıza olduğu zaman, taşıma katmanı yazılımı alternatif güzergahları arar veya gönderilecek veriyi ağ bağlantısı yeniden kurulasıya kadar bekletir, alınan verilerin doğru biçimde ve sırada olup olmadığını kontrol eder. Bu biçimlendirme ve sıralama yetenekleri, ulaşım katmanı programları farklı bilgisayarlar arasında bağlantı kurdukları zaman önem kazanır. Veri-bağı katmanı, hepsinin orada olup olmadığını öğrenmek için vagonları sayabilir. Ulaşım katmanı da bunları açar ve içinde herhangi bir şeyin eksik olup olmadığını kontrol eder. Farklı bilgisayarlardan oluşan ağlar değişik ulaşım-katmanı protokolleri kullanabilirler. Bunlardan en yaygını TCP'dir. Department of Defence tarafından geliştirilmiş ve şimdi TCP/IP'nin bir parçası olarak birçok firma tarafından pazarlanır.

3 Ağ Katmanı(Network Layer):Veri paketlerinin göndericiden alıcıya ağdaki düğümler (Router Gateway vs) üzerinden geçirilip yönlendirilerek alıcıya ulaşmasını sağlayan işlevlere sahiptir. Bu katman mesajları ağlar arasında yönlendirmeden sorumludur. İki tip aygıt mesajları ağlar arasında yönlendirmeden sorumludur. İlki, iki ağ bağdaştırma kartına sahip olan geçit (gateway) denir. Bu bilgisayar bir ağdan, bir ağ bağdaştırma kartı üzerinden gelen ağ paketlerini kabul eder, ve bu paketleri ikinci bir ağ bağdaştırma kartı ile farklı bir ağa yönlendirir. İkincisi, paketleri bir ağdan farklı bir ağa geçiren o işe özel olarak adanmış bir aygıt olan yönlendirici (router)' dır. Bu iki terim birbirinin yerine kullanılabilir, fakat paketleri iletme yetenekleri arasında belirgin farkları vardır. Burada bilgi bloklarına Paket (Packet) adı verilir. IP Protokolü bu katmana ait bir protokoldür.

2 Veri Bağı Katmanı(Data Link Layer):Gönderilecek bilginin hatalara bağışık bir yapıda lojik işaretlere dönüştürülmesi alıcıda hataların sezilmesi, düzeltilmiyorsa bilginin tekrar istenmesi gibi işlemleri vardır. Gönderilen alınan lojik işaret bloklarına çerçeve (Frame) denir.Fiziksel ve elektriksel bağlantılar yapıldıktan sonra sistem içindeki veri akışını kontrol

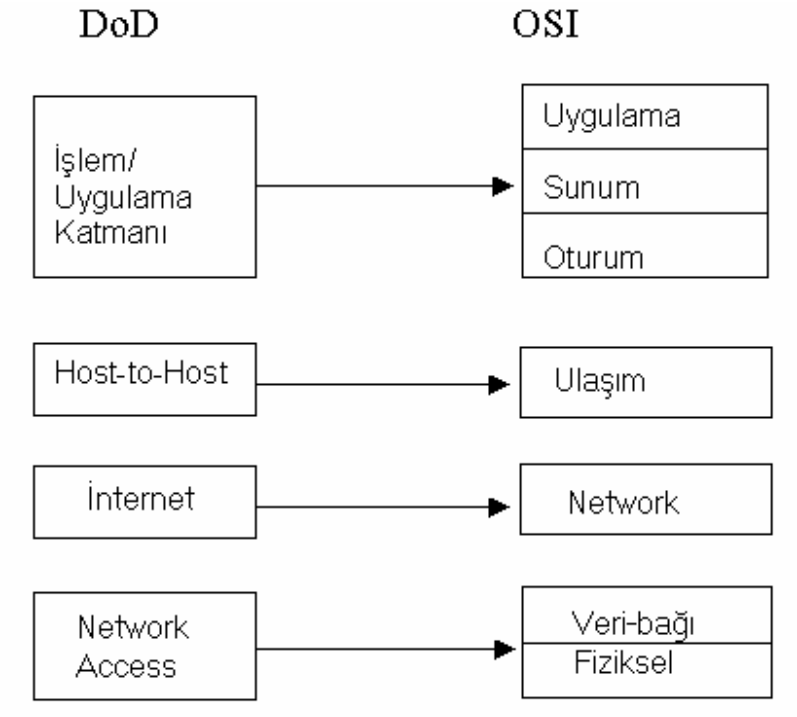
etmelisiniz. Veri-bağı katmanı karakterleri bir dizi halinde birleştirip mesajlar haline getirir ve daha sonra yola koymadan önce kontrol eder. Gönderdikten sonra karşı taraftan "düzgün şekilde geldi" diye bir mesaj gelebilir veya veri doğru gitmediyse yeniden oluşturulabilir. Veri-bağı katmanı High-Level Data Link Control (HDLC), bi-senkron iletişim ve Advanced Data Communications Control Procedures (ADCCP) gibi birçok protokol kullanır. Bunları bilmemiz gerekmez. PC tabanlı iletişim sistem- lerinde, arabirim kartlarının üzerindeki özel devreler bu katmanın fonksiyonlarını yerine getirirler. 1 Fiziksel Katman(Physical Layer):Verinin hat üzerinden aktarılması için gerekli işlemleri kapsar. Bu katmanda tanımlı standartlar taşıyıcı işaretin şekli, verici ve alıcı olan uç noktaların elektriksel ve mekanik özelliklerini belirler. Kablo konnektor standartları bu katmanda yapılır. (UTP,RJ45,RS-232C,V.35,Fiber Optik.....)

İNTERNET PROTOKOL

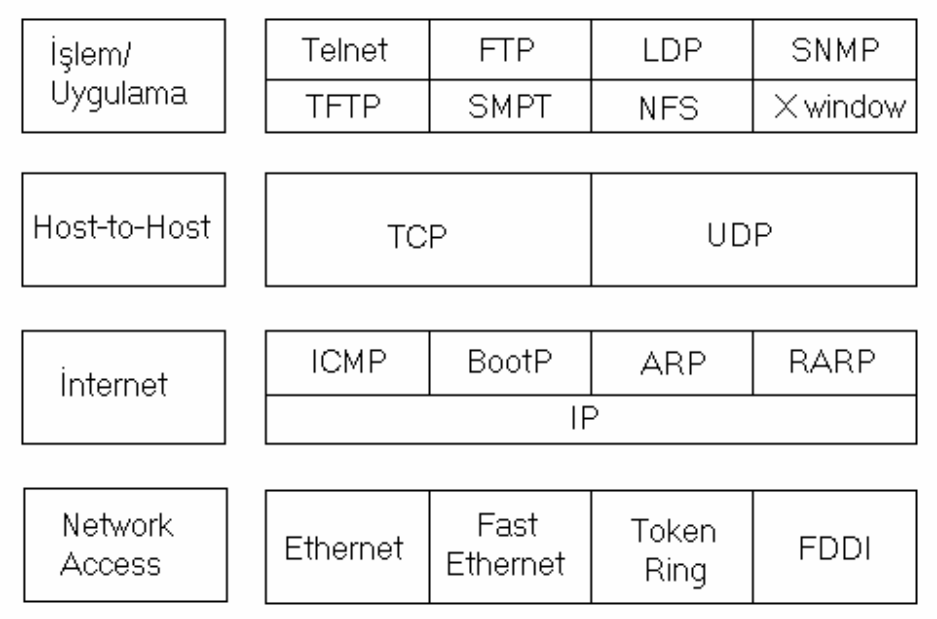
1.2.TCP/IP VE DoD MODELİ

DoD Modeli OSI modelin yoğunlaştırılmış bir versiyonudur. Bu modelde 4 katman baz alınır.

- İşlem/Uygulama katmanı
- Host-to-host katmanı
- Internet katmanı
- Network Access katmanı



Aşağıdaki şekilde TCP/IP ve DoD Modeli protokol kümesi gösterilmiştir.



1.3.İşlem/Uygulama Katmanı Protokolleri

1.3.1.Telnet:

Telnet, Internet ağı üzerindeki bir makinaya uzaktan bağlanmak için geliştirilen bir TCP/IP protokolü ve bu işi yapan programlara verilen genel addır. Bağlanılan makinaya girebilmek (login) için orada bir kullanıcı isminizin (user name) olması gerekir. Bir de telnet erişim programı. Telnet erişim programları, işletim sistemlerinin çoğunda işletim sistemi ile birlikte gelmektedir.

Bazı kütüphane ve herkese açık telnet bazlı web servisleri, buralara telnet ile bağlanıldığında, kullanıcı ismi (numarası) istemeyebilirler; ya da, kullanıcı ismi ve parola olarak ne yazmanız gerektiği bağlandığınızda otomatik olarak karşınıza çıkar.

TELNET aslında ARPANET için geliştirilmiş basit bir terminal emulasyon aracıdır. TELNET ağ-bağımsız bir virtual terminal aracılığıyla kullanıcı koduna sahip olduğu uzak bir TCP/IP yetenekli bilgisayara bağlanabilmeyi sağlar. Kullanıcı uzak TCP/IP bilgisayarındaki standart bağlanma işlemlerini izler ve o bilgisayara ait komutları kullanabilmek için uzak işletim sisteminin karakteristiklerini bilmek zorundadır. TELNET uzak terminallerin ana bilgisayara bağlanmasını, bağlanılan bilgisayarın işletim sistemine sanki yerel bir terminal bağlanıyormuş gibi gösterir. Çoğu zaman TELNET full-duplex modda çalışır, yani ayni anda yollama ve alma yeteneği sağlar.

TELNET protokolünün kullanıcı ve server işlemleri kendi aralarında mantıksal bir sıra izlerler. Kullanıcı TELNET programı, kullanıcı ile server arasında bir passthrough gibi çalışarak veri iletimini sağlar. Makinenin rolüne ve gücüne göre, TELNET'in hem kullanıcı hem de server olarak kullanılması sağlanabilir. Tek görevliliğinden dolayı DOS kullanan mikrobilgisayarlar genellikle TELNET'in kullanıcı tarafını kullanırlar. UNIX ve OS/2 işletim sistemini kullanan bilgisayarlar TELNET'i iki yönlü olarak kullanabilirler. Çünkü bunlar çok-görevli işletim sistemidirler.

1.3.2.FTP

FTP (File Transfer Protocol) Internet'e bağlı bir bilgisayardan diğerine (her iki yönde de) dosya aktarımı yapmak için geliştirilen bir İnternet protokolü ve bu işi yapan uygulama programlarına verilen genel addır. FTP protokolü ile bir başka bilgisayardan bir başka bilgisayara dosya aktarımı yapılırken, o bilgisayar ile etkileşimli-aynı anda (on-line) bağlantı kurulur ve protokol ile sağlanan bir dizi komutlar yardımıyla iki bilgisayar arasında dosya alma/gönderme işlemleri yapılır. Dosya Transfer Protokolü (FTP) bir veri yığınının ASCII, EBCDIC ve binary bir aygıttan diğerine iletimi için kullanılmaktadır. Bir dosyayı FTP ile başka bir TCP/IP ağı üzerindeki kullanıcıya yollamak için o ağdaki bilgisayarda geçerli bir kullanıcı ismi ve şifresi gerekmektedir. Internet 'anonim FTP' ye (anonymous FTP) destek vermekle birlikte bunu dosyayı yollamak için değil sadece okumak için verir. Bu durum, ağ üzerindeki her kullanıcıya postanın yollanmasını sağlayan SMTP yoluyla aşılabilir. Fakat SMTP sadece metin iletebildiği için diğer tip dosyalar gönderilmeden önce metin dosyasına çevrilmelidir. Daha sonra da alici tarafından tekrar eski haline çevrilir. Diğer taraftan elektronik postada kullanılan OSI X.400 standardı, kullanıcıya metin, grafik, teleks, fax, video, ve hatta ses yollamasına izin verir. Elektronik dokuman değişimini (EDI-Electronic Document Interchange)destekler. Ama bu uygulamalar OSI uygulamaları gibi yeterli yaygınlığa ulaşmamıştır.

OSI FTAM (dosya transfer, erişim ve yönetim) protokolü TCP/IP'nin FTP' sinden daha işlevselidir. Görüntü dosya saklama yeteneğinin yanı sıra, FTAM kullanıcısı, tüm dosya yerine dosyanın bir kısmını da gönderebilir. TCP/IP ortamında da ayni düzeyde işlevsellik sağlamak için dosyaları parçalar halinde taşıyabilen NFS (Network File System) FTP yerine kullanılabilir. Bu özelliğinden dolayı NFS'in popülaritesi artmış ve firmalar NFS'i pek çok TCP/IP türüyle entegre etmişlerdir.

1.3.3.TFTP

FTP'nin daha az kapsamlı bir sürümüdür. Küçük boyutlu dosyaların lokal ağlarda iletiminde kullanılır.

1.3.4.NFS

NFS dosya paylaşımında uzman bir protokoldür. Farklı tipte iki dosya sisteminin bir arada çalışmasını sağlar. Örneğin, NT server ile Unix kullanıcı arasındaki bir oturumda iletişimin gerçekleşebilmesi için NFS gereklidir.

1.3.5.SMTP

Basit ağ yönetim protokolü (Simple Network Management Protokol) Elektronik posta hizmeti sunar. Postaların güvenli bir şekilde adreslerine ulaşabilmesi için TCP servislerinden yararlanır.

1.3.6.LPD

İnternet üzerinden yada bir ağ üzerinden yazıcı paylaşımını sağlayan bir protokoldür.

1.3.7.X Window

Grafiksel kullanıcı arayüz tabanlı İstemci/Sunucu uygulamaları geliştirmek için tanımlanmış bir protokoldür.

1.3.8.SNMP

SNMP, ağ üzerindeki bilgisayarların uzaktan izlenmesini ve bazı parametrelerinin değiştirilmesini sağlayan bir protokoldür.İnternet yaygınlaştıkça ağlar arası bağlantıları sağlayan yönlendiriciler ve köprüler önem kazanmaya başlamıştır. SNMP ile yönlendirici ya da köprünün sağlıklı çalışıp çalışmadığını uzaktan izlemek mümkün olmaktadır. SNMP ile bir yönlendiricinin sabit diskinin dolup dolmadığı ya da bir portu üzerindeki trafik miktarı izlenebilir. SNMP iki kısımdan oluşur:

- SNMP yönetim sistemi
- SNMP Ajanları

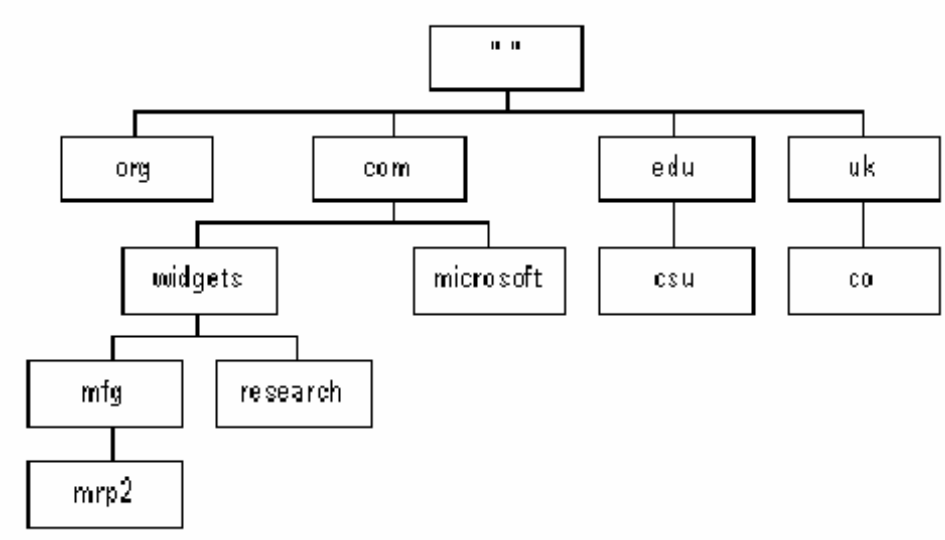
SNMP yönetim sistemi özel bir yazılımdır ve ağdaki yazılım ve donanım unsurlarının SNMP parametrelerini sorgular, bunlardan çeşitli raporlar ya da uyarılar çıkarabilir. SNMP ajanları ise kendilerine sorulduğunda ya da önceden belirlenmiş olaylar gerçekleştiğinde SNMP parametrelerini yönetici sistemlere bildirirler.

1.3.9.DNS

DNS, TCP/IP network' ünde client /server iletişiminin ayrılmaz bir parçasıdır. DNS TCP/IP network' ünde bilgisayar ismini ip adresine , IP adresini de bilgisayar ismine çevirmek için kullanılan networke dağıtılmış bir veritabanıdır. Microsoft Windows 2000 isim çözünürlülüğü için öncelikli metot olarak DNS' i kullanır. Fakat Wins (Windows İnternet Name Service) servisine destek vermeye devam eder.Wins, Windows NT 4.0 ve önceki versiyonların kullandığı bir isim çözme metodudur. Windows 2000 , WINS server servisinin bilgisayar ismi çözünürlülüğü yerine DNS Server servisinin FQDN isim çözünürlülüğü nü kullanır. Clientler (Network'deki üye bilgisayarlar) isim çözünürlülüğü, servislerin bulunduğu domain contoller'lara erişmek ve login işlemleri için DNS Server servisini kullanır.

DOMAIN İSİM UZAYI (Domain name space)

DNS veritabanı ağaç şeklinde yapılandırılmıştır. Her domainin bir ismi vardır ve alt domainler içerebilir. Bir domain ismi ana domaine bağlı olarak o domainin veritabanındaki yerini gösterir. DNS domain'indeki her noktaya ait isimler birbirinden (.) karakteriyle ayrılırlar. Örneğin edu.tr DNS domain ismi tr domain'ine bağlı edu alt domain'ini gösterir.



DNS İSİM SİSTEMİ VE ÇÖZÜMLENMESİ

DNS, İnternetteki Hostları tanımlamak için hiyerarşik bir isimlendirme kullanan dağıtılmış bir veritabanıdır. DNS 1980'lerde İnternette bağlı Hostların sayısının hızla artmasıyla ortaya çıkan problemleri çözmek amacıyla geliştirilmiştir. DNS standartları RFC (Requests For Commend) 1034 ve 1035 ile belirlenmiştir. DNS WINS'e benzemekle birlikte en temel fark WINS'in tamamen dinamik olması, DNS' inse bilgisayar isimleri ile IP adresler arasında mapping işlemini yapacak tamamen statik bir konfigürasyon gerektirmesidir.

DOMAIN İSİMLERİ

Bazı WinNT Server ekranlarında (örneğin User Manager For Domains), domain isminin kullanıcı ismini takip ettiği gözükür. Domain ismi, kullanıcı account'unun nerede oluştuğunu ve tüm domain isteminde nerede bulunduğunu gösterir. Örneğin, **cc** domain'indeki **erdinç** kullanıcısı **cc\erdinç** şeklinde gözükür. Bu şekilde gösteriliş başka bir domainde aynı adlı kullanıcılardan ayırımı sağlar.

1.3.10.Bootp

Bu protokol sadece disket sürücüsü olmayan bilgisayarların IP adresi almalarını sağlar. Networke bağlı disket sürücüsüz bir bilgisayar ilk açıldığında ağa bir BootP isteğini yayın eder. Ağdaki Bootp sunucu bu isteği duyar ve gönderenin MAC adresini kendi tabanında arar. Eğer veri tabanında bu istemci için bir kayıt bulunursa bu istemciye bir IP adresi TFTP protokolünü kullanarak istemciye boot edebilmesi için gereken dosyayı yollar.

1.3.11.DHCP

Her bir bilgisayar açıldığında network üzerinde bir DHCP-server var mı diye mesaj yollar (kendisine otomatik olarak bir ip adresi atansın diye). Atanan bu ip adresleri genellikle kalıcı olmaz, ancak bir süreye bağlı olarak atanmış olurlar (günler, aylar, haftalar boyunca olabilir, ancak internet'e dial-up bağlantıda bu sadece bağlantı süresincedir). Eğer sistem bu süre içinde DHCP-Server'a tekrar başvurursa IP adresinin süresi uzatılır. Ancak uzun bir süre, otomatik olarak IP adresinin atanmış olduğu sistem, DHCP-Server ile bağlantı kurmazsa, onun için atanmış IP adresinin süresi dolmuş kabul edilir ve bir başkasına atanmak üzere

bekletilir. Kendisine verilen IP adresinin süresi geçen sistem yeni bir adres için tekrar başvurur.

1.4.Host-to-Host katmanı protokolleri

1.4.1.TCP

TCP'nin (transmission control protocol-iletişim kontrol protokolü) temel işlevi, uygulama katmanından gelen bilginin segmentler haline dönüştürülmesi, iletişim ortamında kaybolan bilginin tekrar yollanması ve ayrı sıralar halinde gelebilen bilginin doğru sırada sıralanmasıdır. IP ("internet protocol") ise tek tek datagramların yönlendirilmesinden sorumludur. Bu açıdan bakıldığında TCP katmanının hemen hemen tüm isi üstlendiği görülmekle beraber (küçük ağlar için bu doğrudur) büyük ve karmaşık ağlarda IP katmanı en önemli görevi üstlenmektedir. Bu gibi durumlarda değişik fiziksel katmanlardan geçmek, doğru yolu bulmak çok karmaşık bir is halini almaktadır.

Şu ana kadar sadece internet adresleri ile bir noktadan diğer noktaya ulaşılması konusundan bahsettik ancak birden fazla kişinin ayni sisteme ulaşmak istemesi durumunda neler olacağı konusuna henüz bir açıklık getirmedik. Doğal olarak bir segment'i doğru varış noktasına ulaştırmak tek başına yeterli değildir. TCP bu segment'in kime ait olduğunu da bilmek zorundadır. "Demultiplexing" bu soruna çare bulan yontemdir. TCP/IP 'de değişik seviyelerde "demultiplexing" yapılır. Bu işlem için gerekli bilgi bir seri "başlık" (header) içinde bulunmaktadır. Başlık, datagrama eklenen basit bir kaç octet'den oluşan bir bilgiden ibarettir. Yollanmak istenen mesajı bir mektuba benzetecek olursak başlık o mektubun zarfı ve zarf üzerindeki adres bilgisidir. Her katman kendi zarfını ve adres bilgisini yazıp bir alt katmana iletmekte ve o alt katmanda onu daha büyük bir zarfın içine koyup üzerine adres yazıp diğer katmana iletmektedir. Benzer işlem varis noktasında bu sefer ters sırada takip edilmektedir.Bir örnek vererek açıklamaya çalışırsak: Aşağıdaki noktalar ile gösterilen satir bir noktadan diğer bir noktaya gidecek olan bir dosyayı temsil etsin,

.....

TCP katmanı bu dosyayı taşınabilecek büyüklükteki parçalara ayırır:

... ...

Her segment'in başına TCP bir başlık koyar. Bu başlık bilgisinin en önemlileri 'port numarası' ve 'sıra numarası' dır. Port numarası, örneğin birden fazla kişinin ayni anda dosya yollaması veya karsıdaki bilgisayara bağlanması durumunda TCP'nin herkese verdiği farklı bir numaradır. Üç kişi ayni anda dosya transferine başlamışsa TCP, 1000, 1001 ve 1002 "kaynak" port numaralarını bu üç kişiye verir böylece herkesin paketi birbirinden ayrılmış olur. Ayni zamanda varis noktasındaki TCP de ayrıca bir "varış" port numarası verir. Kaynak noktasındaki TCP'nin varış port numarasını bilmesi gereklidir ve bunu iletişim kurulduğu anda TCP karşı taraftan öğrenir. Bu bilgiler başlıktaki "kaynak" ve "varış" port numaraları olarak belirlenmiş olur. Ayrıca her segment bir "sıra" numarasına sahiptir. Bu numara ile karşı taraf doğru sayıdaki segment'i eksiksiz alıp almadığını anlayabilir. Aslında TCP segmentleri değil octet leri numaralar. Diyelim ki her datagram içinde 500 octet bilgi varsa ilk datagram numarası 0, ikinci datagram numarası 500, üçüncüsü 1000 seklinde verilir. Başlık içinde bulunan üçüncü önemli bilgi ise "kontrol toplamı" (Checksum) sayısıdır. Bu sayı segment içindeki tüm octetler toplanarak hesaplanır ve sonuç başlığın içine konur. Karşı

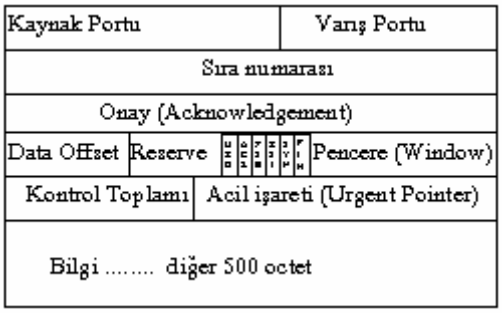
noktadaki TCP kontrol toplamı hesabini tekrar yapar. Eğer bilgi yolda bozulmamışsa kaynak noktasındaki hesaplanan sayı ile varis noktasındaki hesaplanan sayı ayni çıkar. Aksi halde segment yolda bozulmuştur ve bu datagram kaynak noktasından tekrar istenir. Eğer TCP başlığını "T" ile gösterecek olursak yukarda noktalarla gösterdiğimiz dosya aşağıdaki duruma gelir:

T... T... T... T... T...

Başlık içinde bulunan diğer bilgiler genelde iki bilgisayar arasında kurulan bağlantının kontrolüne yöneliktir. Segment'in varışında alıcı gönderici noktaya bir "onay" (acknowledgement) yollar. Örneğin kaynak noktasına yollanan "onay numarası" (Acknowledgement number) 1500 ise octet numarası 1500 e kadar tüm bilginin alındığını gösterir. Eğer kaynak noktası belli bir zaman içinde bu bilgiyi varis noktasından alamazsa o bilgiyi tekrar yollar. "Pencere" bilgisi bir anda ne kadar bilginin gönderileceğini kontrol için kullanılır. Burada amaç her segment'in gönderilmesinden sonra karşıya ulaşıp ulaşmadığı ile ilgili onay (ack) beklenmesi yerine segmentleri onay beklemeksizin pencere bilgisine göre yollamaktır. Zira yavaş hatlar kullanılarak yapılan iletişimde onay beklenmesi iletişimi çok daha yavaşlatır. Diğer taraftan çok hızlı bir şekilde sürekli segment yollanması karşı tarafın bir anda alabileceğinden fazla bir trafik yaratacağından yine problemler ortaya çıkabilir. Dolayısıyla her iki taraf o anda ne kadar bilgiyi alabileceğini "pencere" bilgisi içinde belirtir. Bilgisayar bilgiyi aldıkça pencere alanındaki boş yer azalır ve sıfır olduğunda yollayıcı bilgi yollamayı durdurur. Alici bilgiyi istedikçe pencere artar ve bu yeni bilgiyi karşıdan kabul edebileceğini gösterir. "Acil işareti" ise bir kontrol karakteri veya bir komut ile transferi kesmek vs. amaçlarla kullanılan bir alandır. Bunlar dışında ki alanlar TCP protokolünün detayları ile ilgili olduğu için burada anlatılmayacaktır. TCP bağlantı yönelimli, alındı bildirimli, hata sezme yeteneğinde, uçtan uca veri bütünlüğü sağlayan bir hizmet sunar. TCP hizmeti alıcı ve göndericinin oluşturduğu *socket* adı verilen uç noktaları ile olur. Her socket host'un ip numarası (32 bit) ve port numarasından (16 bit) oluşan 48 bitlik bir numara içerir. TCP hizmeti alabilmek için, alıcı ve verici hosta socketler arası bir bağlantı kurulması gereklidir. Bir socket aynı anda birden fazla bağlantı için kullanılabilir. Bağlantılar socket tanıtıcıları tarafından her iki uçta tanıtılır.

1024'ün altındaki port numaralarına bilinen numaralar denir ve standart hizmetler için rezerv edilmişlerdir. Örneğin ftp hizmeti için 21 nolu port, Telnet için 23 nolu port gibi.

TCP hizmeti noktadan noktaya olup kısmi yayın (multicast), yayın gibi protokolleri desteklemez. Bunlar için IGMP (Internet Group Message Protocol) kullanılabilir. TCP bir byte akımı bağlantısıdır, mesaj akımı değil.



TCP'nin paket yapısını yukarıdaki şekilde inceleyip, paketteki alanların görevlerine bakarsak. Şekildeki format segment yapısını göstermektedir. Her segment sabit formatlı 20 byte uzunluklu bir başlık ile başlar. Sabit başlık opsiyonları tarafından izlenebilir. Opsyonlar'dan sonra, eğer varsa, 65535 – 20 – 20 = 65495 veri byte'ı izleyebilir. Burada çıkarılan ilk 20 byte ip başlığını ve ikinci 20 byte TCP başlığını belirtmektedir. Herhangi bir veri içermeyen segmentler de geçerlidir, ve genel olarak alındı bildirimi ve kontrol mesajlarında kullanılırlar. TCP katmanına gelen bilgi segmentlere ayrıldıktan sonra IP katmanına yollanır. IP katmanı, kendisine gelen TCP segment'i içinde ne olduğu ile ilgilenmez. Sadece kendisine verilen bu bilgiyi ilgili IP adresine yollamak amacındadır.

Kaynak portu ve hedef portu (16 + 16 bit): Hangi üst katman kaynağının ve hedef sürecinin TCP hizmetini alacağını tanımlar.

Dizi Numarası (32 bit): Genellikle o anki mesajın ilk byte'na atanmış numaradır. Belirli koşullar altında, devam edecek iletimde kullanılmak üzere bir başlangıç dizi numarası tanımlayabilir.

Alındı bildirimi numarası (Ackonwledegement - 32 bit): Paket göndericisinin ulaşmasını bekle-diği verinin sıradaki byte'ının dizi numarasını içerir.

TCP başlık uzunluğu(4 bit): TCP başlığındaki 32 bitlik kelimelerin sayısını içerir.

Rezerv(6 bit): Gelecekteki kullanımlar için rezerve edilmiştir.

Bayraklar (6 bit): Çeşitli kontrol bilgisi taşırlar.

Pencere büyüklüğü (16 bit): Göndericinin pencere büyüklüğünü tanımlar.(gelen veri için kullanılabilir buffer uzayıdır.)

Kontrol Toplamı (Checksum – 16 bit): Başlığın iletilirken zarar görüp görmediğini gösterir.

Acil İşaretçisi (16 bit): Paket içersindeki ilk acil veriye işaret eder.

Opsiyonlar (0 veya 32 bit 'in katları): Çeşitli TCP opsiyonlarını tanımlar.

Veri: Üst katman verisini içerir.

1.4.2.UDP

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. Hedefe ulaşan paketler üzerinde sıralama yapıp doğru veri aktarımını sağlayacak mekanizmaya sahip değildir. Uygulamalar güvenli ve sıralı paket dağıtımı gerektiriyorsa UDP yerine TCP protokolü tercih edilmeli. UDP, minimum protokol yükü (overhead) ile uygulama programları arasında basit bir aktarım servisi sağlar.

İnternet protokol takımı bağlantısız bir protokolü de destekler. UDP uygulamalar için kapsüllenmiş ham IP datagramların gönderilmesi için bir yol sağlar, ve datagram'ları bir bağlantı kurmadan gönderir. Birçok sunucu-istemci uygulamasında bir istek ve cevaptan

oluşan UDP, bağlantı kurma ve çözme yerine tercih eder. UDP RFC 768'de tanımlıdır. Bir UDP segment'i data tarafından izlenen 8 byte'lık bir başlığa sahiptir. UDP uzunluk alanı 8 byte'lık başlık ve veriyi içerir.

1.5.İnternet Katmanı Protokolleri

1.5.1.IP

İnternet katmanında en çok bilinen protokol, tüm TCP/IP ağları için basit paket iletme hizmetini sağlayan internet protokolü (IP)'dir. Ağ katmanında kullanılan fiziksel düğüm adreslerine ek olarak, IP protokolü IP adresleri denilen lojik bilgisayar adresleme sistemini kullanır. IP adresleri internet ve yüksek seviyeli protokoller tarafından cihazları tanımlamak için ve ağlar arası yönlendirme için kullanılır. Adres çözünülürlük protokolü (ARP) IP' ye IP adresine uyan bir fiziksel adresi etkinleştirmesini sağlar.

IP veri iletmek için alt ve üst katmanlardaki tüm protokoller tarafından kullanılır, bu şu anlama gelir ki tüm TCP/IP verisi nihai hedefine bağlı olmaksızın alınıp verildiğinde IP üzerinden akar.

IP bağlantısız bir protokoldür, yani, IP veri iletmeden önce uçtan-uca bağlantı kurmak için kontrol bilgisi alıp vermez. Buna tezat olarak, bir bağlantı yönelimli protokol, uzaktaki bilgisayar ile veriyi göndermeden önce hazır olduğunu doğrulamak için kontrol bilgisi alış verişi yapar.

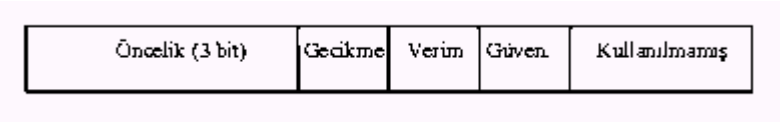
TCP/IP protokolü paket bağlaşmalı bir şebeke olan ARPANET üzerinden veri iletmek için oluşturulmuştur. Bir paket bağlaşmalı şebeke, paketleri bir fiziksel şebekeden diğerine bağlaştırmak için paketlerdeki adresleme bilgilerini kullanır, ve onları nihai hedeflerine taşır. Her paket şebeke içersinde diğer paketlerden bağımsız olarak hareket eder. Datagram IP tarafından tanımlanan paket formatıdır.

Bir IP datagramı bir başlık kısmı ve text kısmı içerir. Başlık 20 byte'lık sabit kısım ve değişken uzunluklu opsiyonel kısmı içerir. Başlık formatı yukarıdaki şekilde görülmektedir. Şimdi sırayla ip datagram başlığındaki bölümleri incelersek;

- **Versiyon (4 bit):** Bu kısım datagramın hangi protokol versiyonundan olduğunu gösterir.
- **IHL (IP header length – 4 bit) :** Başlık kısmının boyu sabit olmadığı için, IHL başlığın 32 bitlik kelimeler halinde kaç tane olduğunu gösterir. Minimum değeri 5

olup opsiyonlar kısmının olmadığını belirtir. Maksimum değeri de 15 olup, başlığı 60 byte ile, böylece de opsiyonlar alanını 40 byte ile sınırlar.

- **Hizmet Türü (Type of Service – 8 bit) :** Bu kısım host'un alt ağa (subnet) ne tür hizmet istediğini söylemesini sağlar. Ses için hızlı iletim, dosya transferi için hatasız iletim gibi. Bu kısmın 8 bitinin kullanımına ilişkin aşağıdaki şekilde de görüldüğü gibidir.



3 bitlik öncelik alanı ve D (Delay), T (Throughtput) ve R (Reliability) bayrakları ve 2 bitlik kullanılmayan kısmı içerir. Öncelik alanı 0 (normal) ve 7 (kontrol paketi) arası öncelik belirtir. D, T ve R'den oluşan diğer 3 bitlik alan Host'u dikkat etmesi gereken konularda (gecikme, verim ve güvenilirlik) uyarır. Teoride bu alana göre yol seçimi (uydu, kiralık hat vb.) yapılır. Pratikte bu alanın içeriği göz önüne alınmaz.

- **Toplam uzunluk (Total length – 16 bit) :** Bu alan datagram içindeki her şeyi, başlık ve verinin her ikisini de içerir. Maksimum uzunluk (2^{16}) 65535 byte' dır. Bu üst sınırın gelecekteki gigabit ağlar ile artması gerekebilecektir.
- **Kimlik (identification – 16 bit) :** Kimlik alanı hedef Host'a yeni ulaşmış bir parçanın hangi datagrama ait olduğunu belirlemesine izin verir. Bir datagramın tüm parçaları aynı kimlik eğerini taşır. Kimlik kısmından sonra kullanılmayan bir bit vardır.
- **DF(Don't fragment – 1 bit) :** Bu kısım, hedef alıcı parçaları birleştirme yeteneğine sahip olmadığı için yönlendiriciler için datagramın parçalanmamasına dair bir emirdir. (DF=1 durumu). Datagramın DF biti ile işaretlenmesi ile gönderici tek parça halinde gönderileceğini bilir. Bu durum yol üzerinde parçalanmamak için optimal olmayan bir rota izlenmesini gerektirebilir.
- **MF (More fragment – 1 bit):** Sonuncusu hariç tüm parçalarda bu bit 1'e ayarlıdır. (MF=1) Bir datagramın tüm parçalarının ulaştığını bilmek için gereklidir. (MF=0, son parçada)
- **Parça Kayıklığı (Fragment offset – 13 bit):** Bu kısım parçanın o anki datagramda nereye ait olduğunu belirtir. Son parça hariç datagramdaki tüm parçalar 8 byte'lık parça biriminin katı olmalıdır. 13 bitlik alan uzunluğu her biri 8 byte'lık 8192 parça birimi sağlar. (8 * 8192 = 65536) değeri 65535 birimlik toplam uzunluk biriminin bir fazlasıdır.
- **TTL (Time to live – 8 bit):** Bu alan paketlerin ömrünü sınırlamak için kullanılan bir sayaçtır. Maksimum 255 sn'ye izin veren, saniyeleri sayan bir sayaç olduğu düşünülür. Her yönlendiricide bu değer bir eksiltilir. Paket kuyruklarda beklediği durumlarda da TTL değeri azaltılır. Paket sonsuz çevrime girip değeri TTL 0 olursa, paket atılır ve göndericisine uyarı mesajı gönderilir.
- **Protokol (8 bit) :** Ağ katmanı tam bir ip datagramı birleştirdiği zaman, paketi ne yapacağını bilme ihtiyacı duyar. Protokol alanı katmana paketi taşıma katmanı sürecine vermesi gerektiğini söyler. Datagramın verileceği süreç, TCP, UDP veya diğerleri olabilir. Protokol numaraları RFC 1700'de tanımlanmıştır.
- **Başlık kontrol toplamı (Header checksum – 16 bit):** Bu alan yalnızca başlığı doğrular. Böyle bir kontrol, bir yönlendiricideki bozuk bellek kelimelerinin ürettiği hataları sezmede yararlıdır. Bu değer yol üzerindeki her adımda, en az bir alan (TTL) değiştiği için, tekrar hesaplanmalıdır.
- **Kaynak adresi (32 bit):**Gönderici adresini içerir.

- **Hedef adresi (32 bit):** Alıcı hedef adresi içerir.

Opsiyonlar (32 bit * 15 = 40 byte 'a kadar): Bu alan aşağıdaki tabloda belirtilmektedir. **Opsiyon**	**Tanımı**
Güvenlik	Datagramın ne kadar gizli olduğunu belirtir.
Değişmez kaynak yönlendirmesi	İzlenmesi gereken tam bir yol verir
Esnek kaynak yönlendirmesi	Atlanmaması gereken yönlendirici listesi verir
Yönlendirme kaydı	Her yönlendiricinin ip adresini pakete eklemesini sağlar
Zaman etiketi	Her yönlendiricinin ip adresini ve zaman etiketini eklemesini sağlar.

Belirttiğimiz başlığa ilişkin bu kısımların ardından *ip* paketindeki veri kısmı gelir.

1.5.2.ICMP

ICMP, TCP / IP protokollerinin işlemesine yardımcı olan bir protokoldür. TCP / IP ile çalışan her Hostta mutlaka ICMP protokolü çalışır. ICMP'nin temel işlevi paket kaynağa, paketinin iletimi sırasında paket üzerinde bir hata meydana gelmesi durumunda, yol üzerindeki bir yönlendirici veya host tarafından paketin sahibinin bilgilenmesini sağlar. ICMP mesajları şu amaçlarla kullanılır:

- Bir yönlendirici paketini TTL süresi dolduğu zaman (TTL = 0) yok eder. Paketin yok edildiğini bir ICMP paketiyle sahibine (yani gönderene) bildirmek amacıyla kullanılır.
- Yönlendirici kendisine gönderilen datagram paketi için yeterli tampon alana sahip değilse bu paket yönlendirici tarafından yok edilir. ICMP paketi gönderen hostu bu durumdan haberdar eder.
- Yönlendirici DF bayrak biti "1" olan bir paketi parçaladığında, hostu bilgilendirir.
- Yönlendirici veya host paketin IP başlığında bir dizilim hatası bulduğunda, hatayı bulan birim tarafından paketi gönderen host ICMP sayesinde bilgilendirilir. (Paket yok edilir)
- Yönlendirici üzerinde geçerli varsayılan yönlendirici tanımı yoksa ve yönlendirici kendisine gelen paketi göndereceği ağın yol bilgisini tablosunda bulamıyorsa, bu yönlendirici tarafından ICMP paketleri aracılığıyla paketi gönderen host bilgilendirilir.
- Yönlendirici kaynak Hosta daha kısa yol olan başka bir yönlendiricinin kullanılmasını önereceğinde bunu ICMP paketleri aracılığıyla yapar.

ICMP paketleri ortamda bir geri besleme sağlarlar. Bu yolla ciddi problemleri, haberleşen birimlere bildirerek bir hata bildirim mekanizması oluştururlar. ICMP mesajı, IP paketlerinin veri bölümünde taşınır. Bu yüzden ICMP paketlerinin dağıtım güvenilirliği, IP paketlerinin dağıtım güvenliliği ile sınırlı kalmaktadır. Buradan ICMP paketlerinin güvenli iletilemeyeceği ve hedefe varmasının garanti edilemeyeceği sonucu çıkarılabilir.

1.5.3.ARP

Bilgisayarların ağda haberleşebilmeleri için birbirlerinin hardware adreslerini bilmeleri gerekir. Adres çözme işlemi Host'un IP adresinden hardware adresinin bulunup eşleştirilmesi işlemidir. ARP hedef bilgisayarın veya gateway in hardware adresini çözmek için lokal broadcast (yayın) kullanır. Hardware adres çözüldüğünde önce IP adres ve hardware adres ARP cache'inde bir kayıt olarak depolanır. ARP broadcastta başlamadan önce istediği hardware adres ve IP adres için daima ilk olarak cache'ini kontrol eder.Bilgisayar ağı üzerindeki hostların birbirleriyle haberleşebilmeleri için birbirlerinin IP adreslerini bilmeleri gereklidir. Bu adres ya kullanıcılardan temin edilir ya da DNS gibi isim servislerinden sorulur.

Veri-bağı katmanında, pakete hedefin fiziksel adresini yerleştirme zorunluluğu vardır. Ethernet protokolü tarafından kullanılan görüşme paketlerinde Kaynak ve Hedef için ayrı ayrı olmak üzere 48 bit fiziksel adres (MAC adress) için ayrılmıştır. Bilgisayarlar, ağ katmanında IP adresleriyle haberleşiyorlar gibi görünseler de gerçekte bu adresler sadece rehberdir ve en son hedef noktayı simgeler. Hedef uçlar arasındaki duraklar (köprüler, yönlendiriciler, switchler ..) aralarındaki haberleşmede IP adresi kılavuz olarak kullanılmakta ve gerçek haberleşme için MAC adreslerinden yararlanmaktadırlar. Gerçek haberleşme fiziksel adresler kullanılarak veri-bağlantı katmanında gerçekleştirilir.

ARP protokolü çalıştığı katmanın bir üst katmanından gelen mantıksal adresi (IP adresi) alır ve kendi tuttuğu veya ana bilgisayara (server'a) sorarak bu mantıksal adresi kullanan hostun MAC adresini elde eder. Yani mantıksal adresi çözümleyerek fiziksel adresi elde eder ve veri-bağlantı katmanına bu adresleri iletir.

ARP isteği (Request):

Aynı ağda yerleşmiş olan hostlar birbirleriyle haberleşmek istediklerinde kaynak host bir ARP isteği oluşturur. ARP protokolü çözümleme işini üstlendiğinde önce kaynak hostun belleğinde (RAM) tuttuğu ve ARP önbelleği (ARP Cache) olarak adlandırılan "IP adresi / MAC adresi" listesine bakar. Herhangi bir şekilde bu listeye karşılığı istenen IP adresinin daha önce işlenip işlenmediğini kontrol eder. Eğer bu listede IP adresini bulursa, IP adresine karşılık düşen MAC adresini okur ve bu adresi veri-bağlantı katmanı protokolüne verir.

Aranan IP adresi ARP Cache 'de bulunamazsa bu durumda ARP protokolü özel bir soru paketi oluşturarak bilgisayar ağına yayınlar.(broadcast) Ağ üzerindeki her host bu ARP paketini alır ve kendi IP adresi ile kontrol eder. Eğer IP adresi kendisine ait değilse bu mesajı çöpe atar, kendisinin ise bu pakete fiziksel adresini yerleştirerek bir ARP cevabı (ARP Response) oluşturur ve doğrudan kaynak hosta geri gönderir. Bu arada kendi ARP Önbelleğine de soru paketini gönderen hostun IP adresini ve fiziksel adresini yerleştirir.

Eğer ARP isteğine bir ARP cevabı gelmezse kaynak hosta “host cevap vermiyor” ya da “böyle bir host yok” şeklinde bir mesaj iletilir. Bu mesaj uygulama katmanı protokollerince hazırlanır.

1.5.4.RARP

RARP (Ters Adres Çözümleme Protokolü-Reverse Adress Resolution Protocol) ARP’ nin yaptığı işin tersini yapar. Yani elinde olan bir fiziksel adresin IP karşılığını bulur. Bu protokol özellikle sabit disksiz hostlar tarafından kullanılır. ARP ile aynı paket formatını kullanır.

RARP protokolünün kullanılabilmesi için ağ üzerinde en az bir tane RARP server olmalıdır. Bazen yedekleme amacıyla ağda ikinci bir RARP server’ın kullanılması gerekebilir. Bu durumda Server’lar birincil ve ikincil olarak konfigüre edilir. RARP Server’larının her ağa konması maliyet açısından problem yaratacağından BOOTP protokolü geliştirilmiştir. RARP ‘in aksine BOOTP, UDP ‘nin servislerini kullanmaktadır.

Bir Yerel IP Adresi Çözümlemesi (Resolving A Local IP Adressing)

Her iki bilgisayar arasında iletişim başlatılmadan önce her iki bilgisayarın IP adresinden hardware adresi çözülmelidir. Adres çözme işlemi ARP request(adres çözme isteği) ve ARP reply (adres çözme işlemi cevabı)den oluşmaktadır. Adres çözme işlemi aşağıdaki ţekilde gerçekleţmektedir.

Bir bilgisayar diğer bir bilgisayar ile konuşmak istediğinden bir ARP requesti başlatılır. IP adresinin local network da olduğu tanımlandığında kaynak host hedef hostun hardware adresi için önce kendi ARP cache ini kontrol eder. Eğer kendi kayıtlarında bulamaz ise o zaman ortamdaki bilgisayarlara bu IP adresi ve hardware adresi kimin diye bir soru üretip kendi IP adresi ve hardware adresini ekleyerek bir istek üretir. Bütün yerel ortamdaki bilgisayarın bu isteği alabilmesi için ARP request ini local bir broadcast olarak herkese yollar. Yerel network deki bütün bilgisayarlar bu broadcast'ı alır ve istenen ip adresin kendi ip adresine uyup uymadığını kontrol eder. Eğer çözülmesi istenen adres kendi adresine uymuyor ise bu isteği yok sayar cevap vermez.

Hedefteki bilgisayarlardan birisi isteğe uyan ip adresi ile kendi adresini eşleştirir ve direk olarak isteği gönderen bilgisayara kendi hardware adresi ile bir ARP reply (adres çözme cevabı) gönderir. Karşı bilgisayar tarafından ip adres ve hardware adres bilgisi gönderildiği zaman isteği gönderen bilgisayar bilgileri ARP cache inde güncelleţtirir. Kaynak bilgisayar hedeften reply aldıktan sonra bağlantı kurulmuş olur.

1.6.IP Adresleme:

IP adresi Networkteki elemanların yerini belirtir. IP adresi yazılım adresidir, ağ kartları üzerinde kodlanmış olan donanım adresi değildir. Bir lokal ağdaki kullanıcıları bulmak için kullanılır. IP bir yerel ağdaki kullanıcı ile farklı bir ağdaki kullanıcının haberleşmesine izin verir. Bilgisayarlar arasında bilgi alışverişi IP paketleri aracılığıyla yapılır. Bilgisayar A, bilgisayar B ye göndermek istediği veriyi paketler halinde gönderir. Bu paketler içerisinde A ve B’ nin adresleri vardır. A'nin gönderdiği paketlere IP-Paketleri, A'nin adresine IP-Adresi denir. Gönderilen paketler, B’ nin adresini taşıdığı için bilgisayar ağları arasında kullanılan Routerler aracılığıyla doğru bilgisayara iletilir. IP, paketleri birbirinden bağımsız olarak gönderilir. B’ nin görevi tüm IP paketlerini alıp, A'nin gönderdiği veriyi kazanmaktır.

Gönderilen bazı IP paketleri B' ye ulaşmayabilir. Her paketin içinde, gönderilecek paket sayısı hakkında bilgi olduğundan, B, kaç tane IP paketi alması gerektiğini bilir. Eksik olan paketler tekrar istenir. TCP eksik olan paketleri tespit eder ve A tarafından tekrar gönderilmesi için gerekli işlemleri yapar.

1.6.1.IP TERMİNOLOJİSİ

Bit: 1 dijit (0 yada 1)
Byte: 8 bit (Bazen 7 bit de olabiliyor)
Octet: Her zaman 8 bit
Network Adres: Uzak bağlantılarda paket yönlendirme dizaynıdır.
Broadcast (Yayın) Adres: Uygulamalarda kullanılan ve kullanıcılara gönderilen bilgilerin hepsi bu kanaldan gönderilir. Örneğin 172.16.0.0 network ünün yayın adresi 172.16.255.255 dir.

1.6.2.HİYERARŞİK IP ADRESLEME YÖNTEMİ

Subnet tiplerini ve subnet içinde yer alacak bilgisayar miktarını tespit etmek amacıyla IP-numaraları A, B, C, D, E sınıflarına ayrılır. Burada A, B, C sınıflarını inceleyeceğiz. D ve E sınıfları test amacıyla kullanılır ve pratikte bir anlamları yoktur. Sınıflar arasında ayrım ilk 8 bit üzerinde yapılır. Sınıfların hangi adres alanlarına sahip oldukları aşağıdaki tabloda yer alıyor.

Sınıf	Sistem	1. byte	2. byte	3. byte	4. byte
A	desimal dual	0-127 0xxx xxxx	0-255 xxxx xxxx	0-255 xxxx xxxx	0-255 xxxx xxxx
B		128-191 10xx xxxx	0-255 xxxx xxxx	0-255 xxxx xxxx	0-255 xxxx xxxx
C		192-223 110x xxxx	0-255 xxxx xxxx	0-255 xxxx xxxx	0-255 xxxx xxxx

Sınıf A: 1.0.0.0 - 127.0.0.0
Net-id: İlk byte 0-127 rakamlarını verecek şekilde seçilir: 0000 0000 - 0111 1111
Host-id: Geri kalan 3 byte (24bit = 2^{24} = 16777216) bilgisayar Host-id olarak kullanılır.
A sınıf subnet sayısı: 1. byte tan 7 bit, 2^7= 128

A sınıfından subnetler teorik olarak 16777216 adet host barındırabilirler. Bu kadar sayıda bilgi-sayarı bir subnet içinde barındırmak mümkün değildir.

Sınıf B: 128.0.0.0 - 191.255.0.0
Net-id: İlk iki byte 128.0-191.255 arasında seçilir: 1000 0000.0000 0000 - 1011 1111.1111 1111
Host-id: Geri kalan 2 byte (16bit = 2^{16}= 65536) bilgisayar Host-id olarak kullanılır.
B sınıf subnet sayısı: 1.byte tan 6, 2. byte tan 8, 2^{14}=16384

Sınıf C: 192.0.0.0 - 223.255.255.0
Net-id: İlk üç byte 192.0.0-223.255.255 rakamlarını verecek şekilde seçilir: 1100 0000.0000 0000.0000 0000 - 1101 1111.1111 1111.1111 1111
Host-id: Geri kalan 1 byte (8bit = 256) bilgisayar Host-id olarak kullanılır.
C sınıf subnet sayısı: 1.byte tan 5, 2. byte tan 8, üçüncü byte tan 8, 2^{21}=2.097.152

Aritmetiksel olarak dünyada 128 + 16384 + 2.097.152 = 2.113.664 adet subnet oluşturulabilir. Bu subnetler kendi aralarında tekrar subnetlere ayrılabilir. Bu rakam çok büyük görünse de, aslında internetin son zamanlarda çok hızlı gelişmesinden dolayı, boş net-id kalmamıştır. Bundan dolayı 1995 yılından beri IPv6 isminde bir adresleme üzerinde çalışılmaktadır.

IPv6 16 byten oluşur. 16 byte = 128 bit = 2^{128} = 3.402823669209e+38 (çok büyük bir rakam). IPv6 şu anda kullanılan IPv4 standardına uygun şekilde tasarlanıyor. Bu yüzden iki standardı paralel çalıştırmak mümkün olacak. Belli bir zaman sonra, tamimiyle IPv6 sistemine geçilmiş olacak.

Ayni subnet içinde yer alan bilgisayarlar arasında paketler direk alınıp, gönderilir. Bilgisayar A paketleri göndermeden önce, paket göndermek istediği bilgisayarın kendi subneti içinde olup olmadığını tespit eder. Bu tespit etme işlemi için subnet mask kullanılır. Aşağıda değişik sınıflar için kullanılan subnet masklar yer alıyor.

Sınıf	1. byte	2. byte	3. byte	4. byte
A	255	0	0	0
B	255	255	0	0
C	255	255	255	0

Bilgisayar üzerinde TCP/IP protokolü kurarken, bir IP numarası yanında, kullanılan sınıfa göre bir subnet mask seçilmesi gerekir. Bilgisayar paket gönderirken subnet maskı nasıl kullanır? Örneğin 205.206.54.26 IP numarasına sahip bir bilgisayarın 205.206.54.27 IP numaralı bir bilgisayara paket göndermek istediğini düşünelim. Bu bilgisayarlar 205.206.54 (C sınıf) Net-id sahip subnet içinde yer alıyorlar.Bu yüzden kullanmak zorunda oldukları subnet mask 255.255.255.0 dir.

Bilgisayar A paket göndermeden önce, diğer Hostun ayni subnet olup olmadığını kontrol etmek amacıyla, Bilgisayar B nin IP adresini AND operatörü aracılığıyla subnet maskla karsılaştırır.

205.206.54.26 11001101.11001110.00110110.00011011
255.255.255.0 11111111.11111111.11111111.00000000

205.206.54.0 11001101.11001110.00110110.00000000

Sonuç 205.206.54.0 olduğundan, A, B ile ayni subnet içindedir ve paketi direk B ye yollar. Eğer AND işlemi sonucu başka bir net-id çıkarsa, paket bu subnete iletilmek için gatewaye gönderilir. Gateway bu paketi alıp, hedef subnete iletir. Her subnetin, diğer subnetlerle bağlantı kurmak için kullandığı gateway denilen bir bilgisayarı vardır. Bu bilgisayar genelde ilk host-id ye sahiptir. Subnet dışındaki bir bilgisayara gitmesi gereken paketler gateway' e yönlendirilir. Yayın adresi üzerinden bir paket subnet içinde yer alan bütün bilgisayarlara gönderilir. Örneğin 205.206.54.255 adresine gönderilen bir paket, subnet içindeki her bilgisayar tarafından alınır.

NOT:A sınıfından olan 127. x.x.x IP numarası, bilgisayar üzerinde TCP/IP kurulumunu kontrol etmek amacıyla kullanılır. 127.0.0.1 numaralı IP adresi ile bilgisayarınız içinde yer alan network kartına ulaşabilirsiniz. Bu adrese gönderilen paketler bilgisayar içinde kalır...

1.7.SUBNETTİNG

A sınıfı bir subnet milyonlarca bilgisayar barındırabilir. Network kurmak için kullanılan topoloji-ler, bu kadar çok bilgisayarın ayni subnet içinde çalışmasına imkan tanımaz. Bu yüzden bu tür büyük subnetlerin, tekrar küçük ağlara, subnetlere bölünmesi gerekir. Subnet terimi büyük bilgisayar ağlarını oluşturan, küçük bilgisayar ağları için kullanılır. Subnetting terimi, büyük bir bilgisayar ağını, küçük parçalara bölmek için kullanılır. Bir bilgisayar ağını küçük parçalara bölmek için çeşitli sebepler vardır:

Çeşitli topolojiye sahip ağların birbirinden ayrılması,
Değişik binalarda ve yerlerde olan ağların birbirinden ayrılması,
Önemli bilgilere sahip bilgisayarların korunması.

Bilgisayar ağları oluşturmak içinde çeşitli topolojiler kullanılır. En yaygın kullanılan topolojiler Ethernet ve Token-Ring dir. Bu topolojilerin barındırabileceği bilgisayar sayısı sinirli olduğu için, subnetting usulü ile, birden fazla subnet oluşturulması gerekir. Subnetting in nasıl yapıldığını bir örnek üzerinde inceleyelim. C sınıfı (209.95.104.x) bir Net-ID' mizin olduğunu düşünelim. Kullanmamız gereken subnet mask 255.255.255.0 dir. Teorik olarak C sınıf IP-numaralarını kullanarak 256 bilgisayar barındırabilecek bir bilgisayar ağı oluşturabiliriz. 256 sayısı fazla olduğu için, 32 bilgisayarın barındığı 8 subnet oluşturmaya karar veriyoruz.

İki ve ikinin katları olarak subnet adedini tespit edebiliriz. Yukarda yer alan örnekte kurmak istediğimiz 8 subnet içinde 3 bite ihtiyacımız vardır.

$2^3 = 8$
111=8

8 subnet oluşturmak için bu 3 biti Host-ID kısmından alarak, Net-ID sine eklememiz gerekiyor. Geri kalan 5 bit ile her bir subnet için 2^5= 32 bilgisayara IP-numarası verebiliriz. Subnet mask üzerinde gerekli değişikliği yaptıktan sonra, C sınıf bilgisayar ağımız, 32 ser bilgisayardan oluşan 8 subnet ihtiva edecektir.

NOT: Yukarıda hesaplamış olduğumuz 32 bilgisayar ve 8 adet subneti pratikte oluşturmak mümkün değildir. Kullanabileceğimiz subnet sayısı: 2^n-2 bağıntısından hesaplanır. Buda 2^3 – 2=6 olarak bulunur. Demek ki gerçekte yukarıdaki net-ID için oluşturabileceğimiz subnet sayısı 6 dır. Ve aynı şekilde subnet başına kullanabileceğimiz bilgisayar sayısı ise yine: 2^n –2 bağıntısından hesaplanır. Buda 2^5–2=30 olarak bulunur. Yani pratikte oluşturabileceğimiz Host sayısı her subnet için 30 dur.

Subnet mask nasıl tespit edilir?
C sınıf bir net-id için 255.255.255.0 subnet mask olarak kullanılır. 8 subnetten oluşacak bilgisayar ağı için aşağıda yer alan subnet mask kullanılır.

1111 1111.1111 1111.1111 1111.1110 000
255 255 255 224

Subnet adedi için gerekli bit sayısı, subnet maska eklenerek, yeni subnet mask elde edilir. Subnetting in nasıl yapıldığını ikinci bir örnek üzerinde inceleyelim:

Net-id: 209.95.104.x
Subnet sayısı: 4
Kullanılan bit sayısı: 2
Bilgisayar adedi: 64 (Geri kalan bit adedi 6, 2^6=64)
Subnet mask: 255.255.255.192 (x.x.x.1100 0000)

Subnet ya da alt ağ kavramı, kurumların ellerindeki Internet adres yapısından daha verimli yararlanmaları için geliştirilen bir adresleme yöntemidir. Pek çok büyük organizasyon kendilerine verilen Internet numaralarını "subnet" lere bölerek kullanmayı daha uygun bulmaktadırlar. Subnet kavramı aslında 'Bilgisayar numarası' alanındaki bazı bitlerin 'Ağ numarası' olarak kullanılmasın-dan ortaya çıkmıştır. Böylece, elimizdeki bir adres ile tanımlanabilecek bilgisayar sayısı düşürülüp, tanımlanabilecek ağ sayısını yükseltmek mümkün olmaktadır.

Nasıl bir ağ yapısının kullanılacağı kurumların ağ yapısına ve topolojisine bağlı olarak değişmekte-dir. Subnet kullanılması bilgisayarların adreslenmesi kontrolü merkezi olmaktan çıkmakta ve yetki dağıtımı yapılmaktadır. Subnet kullanılması, sadece o adresi kullanan kurumu ilgilendirmekte ve bunun kurum dışına hiçbir etkisi de bulunmamaktadır. Herhangi bir dış kullanıcı subnet kullanılan bir ağa ulaşmak istediğinde o ağda kullanılan subnet yönteminden haberdar olmadan istediği noktaya ulaşabilir. Kurum sadece kendi içinde kullandığı geçiş yolları ya da yönlendiriciler üzerinde hangi subnete nasıl gidilebileceği tanımlamalarını yapmak durumundadır.

Bir Internet ağını subnetlere bölmek, subnet maskesi kullanılarak yapılmaktadır. Eğer maske adresteki adres bit'i 1 ise o alan ağ adresini göstermektedir, adres bit'i 0 ise o alan adresin bilgisayar numarası alanını göstermektedir. Konuyu daha anlaşılır kılmak için bir örnek üzerinde inceleyelim:

ODTU kampusu için bir B-sınıfı adres olan 144.122.0.0 kayıtlı olarak kullanılmaktadır. Bu adres ile ODTU 65.536 adet bilgisayarı adresleyebilir. Standart B-sınıfı bir adresin maske adresi 255.255.0.0 olmaktadır. Ancak bu adres alındıktan sonra ODTÜ'nün teknik ve idari yapısı göz önünde tutularak farklı subnet yapısı uygulanmasına karar verilmiştir. Adres içindeki üçüncü octet' inde ağ alanı adreslemesinde kullanılması ile ODTU' de 254 adede kadar farklı bilgisayar ağının tanımlanabilmesi mümkün olmuştur. Maske adres olarak 255.255.255.0 kullanılmaktadır. İlk iki octet (255.255) B-sınıfı adresi, üçüncü octet (255) subnet adresini tanımlamakta, dördüncü octet (0) ise o subnet üzerindeki bilgisayarı tanımlamaktadır.

```
144.122.0.0 ODTU için  kayıtlı adres

255.255.0.0  Standart B-Sınıfı adres maskesi  Bir ağ, 65536 bilgisayar

255.255.255.0 Yeni maske       254 ağ, her ağda 254 bilgisayar
```

ODTU de uygulanan adres maskesi ile subnetlere bölünmüş olan ağ adresleri merkezi olarak bölümlere dağıtılmakta ve her bir subnet kendi yerel ağı üzerindeki ağ parçasında 254 taneye kadar bilgisayarını adresleyebilmektedir. Böylece tek bir merkezden tüm üniversitedeki makinaların IP adreslerinin tanımlanması gibi bir sorun ortadan kaldırılmış ve adresleme yetkisi ayrı birimlere verilerek onlara kendi içlerinde esnek hareket etme kabiliyeti tanınmıştır. Bir örnek verecek olursak: Bilgisayar Mühendisliği bolumu için 71 subneti ayrılmış ve 144.122.71.0 ağ adresi kullanımlarına ayrılmıştır. Böylece, bolum içinde

144.122.71.1 den 144.122.71.254 'e kadar olan adreslerin dağıtımı yetkisi bolumun kendisine bırakılmıştır. Ayni şekilde Matematik bolumu için 144.122.36.0, Fizik bolumu için 144.122.30.0 ağ adresi ayrılmıştır.

C-sınıfı bir adres üzerinde yapılan bir subnetlemeye örnek verecek olursak:

Elinde C-sınıfı 193.140.65.0 adres olan bir kurum subnet adresi olarak 255.255.255.192 kullandığında

```
193.140.65.0    11000001 10001100 01000001 00000000

255.255.255.192 11111111 11111111 11111111 11000000

                 <--------------------------->|<---->
                                              |
                Ağ numarası alanı             |Bilgisayar Numarası
```

elindeki bu adresi dört farklı parçaya bölebilir. Değişik subnet maskeleri ile nasıl sonuçlar edinile-bileceği ile ilgili örnek bir tablo verecek olursak :

IP adres	Subnet	Açıklama
128.66.12.1	255.255.255.0	128.66.12 subnetindeki 1.Host
130.97.16.132	255.255.255.192	130.97.16.128 subnetindeki 4.Host
192.178.16.66	255.255.255.192	192.178.16.64 subnetindeki 2.Host
132.90.132.5	255.255.240.0	132.90.128 subnetindeki 4.5 inci Host
18.20.16.91	255.255.0.0	18.20.0.0 subnetindeki 16.91 inci Host

CİSCO SWİTCH KONFİGÜRASYONU

Bu bölümde Cisco Catalyst Switchlerin temel özelliklerini, Konfigürasyonunu öğreneceğiz. İlerde anlatacağımız VLAN (Virtual LAN) için bir altyapı oluşturacağız.

2.1.LAYER-2 SWİTCHİNG

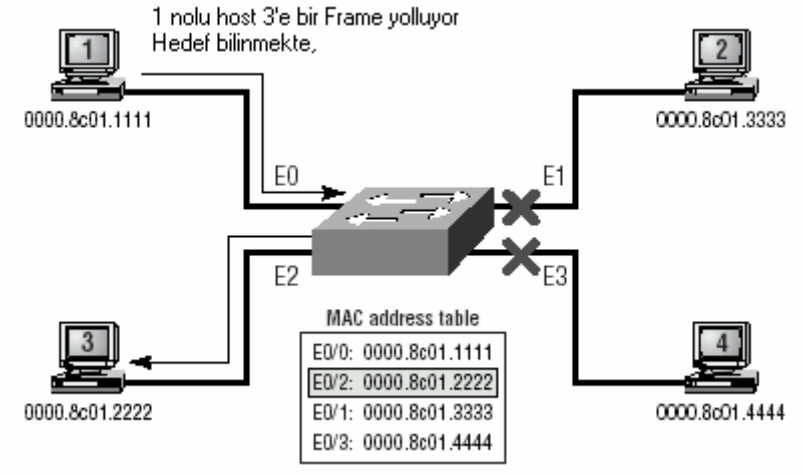
Layer-2 switching donanım tabanlıdır. Networkteki bir kullanıcıyı diğerlerinden ayırt etmek için her bir Host'un Ağ kartındaki MAC adresini kullanır. Switchler Aplication-Specific İntegrated Circuits (ASICs) Kullanarak kendi içinde filtre tablosu kurar ve bu tabloyu muhafaza eder. Layer-2 switching'i çoklu köprü gibi düşünebiliriz. Layer-2 switchler hızlıdır çünkü Network layer başlığına bakmazlar. Bunun yerine Framelerdeki (çerçevelerdeki) Donanım adreslerine (MAC) bakarak bunların gidecekleri yerlere akışını sağlarlar.

2.1.1.SWİTCHLERİN LAYER-2'DEKİ ÜÇ FONKSİYONU

Layer-2 switchlerin üç fonksiyonu vardır.
Adres Öğrenme: Layer-2 switchler arayüzlerden alınmış olan her bir çerçevenin kaynağının MAC adreslerini tutup bu adresleri MAC Database tablosuna kaydeder.
Forward/Filter: Arayüzde bir çerçeve alındığı zaman switch bunun hedef donanım adresine bakar ve MAC Database 'inden bulup çerçeveyi, bağlı olduğu arayüzden hedefe gönderir.
Döngüleri Önlemek: Eğer switchler arasında yapılan çoklu bağlantılar gereğinden fazla büyüklükte olursa bir network döngüsü oluşabilir. Spanning Tree Protocol (STP) bu tür döngüleri durdurmada kullanılır.

2.1.1.1-Adres öğrenme: Switch açıldığı zaman MAC Filtering tablosu boştur. Ekipmanlardan biri bir bilgi gönderdiği zaman ve Arayüzden bir Frame alındığı zaman, Switch kaynak adresini hangi aygıtın MAC adresi olduğunu tayin edip MAC Filtering tablosuna kaydeder.
Eğer aygıt cevap verip geri bir frame yollarsa o zaman switch bu frame den kaynak adresini alacak ve MAC Adres Database e yerleştirip bu adresle interface' teki frame' i karşılaştırır. Şimdi switchte iki adet adres mevcuttur. Artık bir "point-to-point" bağlantı yapabilir ve Frame'ler bu iki nokta arasında güvenli bir şekilde transfer edilebilirler.

Şekilde adım adım adres öğrenme örneğini inceleyelim.



Bu şekilde bir switche 4 adet Host bağlanmıştır.

Switch ilk açıldığı zaman MAC Adres tablosunda hiçbir kayıt yoktur.

1. 1.Host 3.Hosta bir Frame yolladı. Birinci Host'un MAC adresi 0000.8C01.1111 ve 3.Host'un MAC adresi 0000.8C01.2222'dir.
2. Switch E0/1 Arayüzünden Frame'i alır ve MAC Adres Database ine Kaynağın MAC adresini yazar.
3. MAC Database te Frame'in gideceği Hedef adres kayıtlı değil bu yüzden switch bu Frame'i bütün arayüzlerden diğer aygıtlara gönderir.
4. 3. Host Frame'i alır ve 1. Hosta bir Frame yollayarak cevap verir. Switch bu Frame'i E0/3 arayüzünden alır ve bu kaynak donanım adresini MAC Database ünitesine kaydeder.
5. 1. ve 3. Host point-to-point bağlantı yapabilirler ve sadece bu iki aygıt Frame'leri alabilir.

2.1.1.2.- Forward/ Filter Kararı

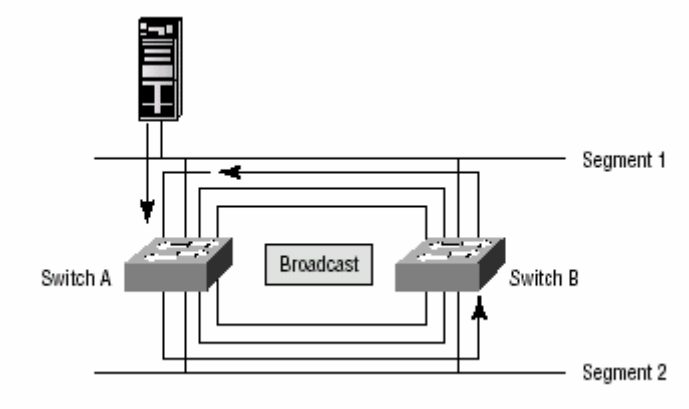
Switch arayüzüne bir Frame geldiği zaman hedef donanım adresi Forward/ Filter Database tablosundaki adreslerle karşılaştırır. Eğer hedef donanım adresi biliniyorsa ve Database tablosunda listelenmiş ise sadece o arayüzden doğru bir şekilde Frame'ler gönderilir. Switch Frame'leri diğer arayüzleri hariç tutarak sadece hedef arayüze gönderir.
Eğer hedef donanım adresi MAC Database tablosunda listelenmemişse, switch bu Frame'i bütün aktif arayüzlerden gönderir. Buda yayın adresinden (broadcast adres) yapılır. Eğer bir aygıt bu yayına cevap verirse; MAC Database tablosu, cevap gelen arayüz ile birlikte güncellenir.

Broadcast ve Multicast Frame'ler

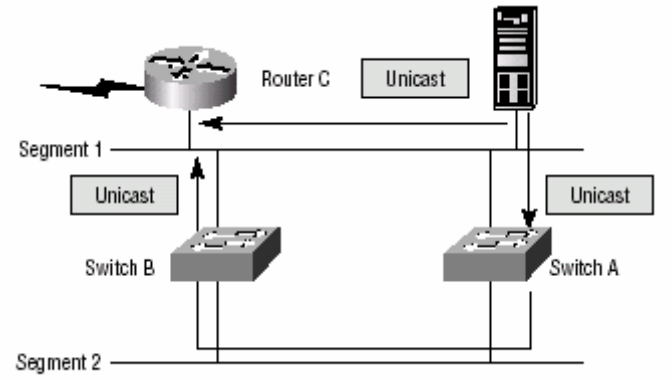
Broadcast ve multicast frame' ler herhangi bir aygıta özgü adresler olamazlar. Kaynak her zaman gönderici aygıtın donanım adresi olacak ve hedef adres ağdaki diğer aygıtları kapsamalıdır. Bu da broadcast adresiyle gerçekleşir.
Kısaca: Switchin arayüzüne bir frame geldi. Switch bu frame' in kaynak adresini MAC adres tablosuna atar ve hedef adresini alıp Forward/Filter tablosundaki adreslerle kıyaslar. Eğer bu adres kayıtlı ise hedef aygıtın hangi arayüzde olduğu biliniyordur ve bu arayüzden direk bağlantı kurulur. Eğer hedef adres switchin Forward/Filter tablosunda yoksa switch bunu broadcast adresinden yollar. Böylece frame ağdaki bütün kullanıcılara aynı anda ulaşır ve bunlardan sadece o adrese sahip olan kullanıcı switche cevap verir. Böylece bağlantı kurulur.

2.1.1.3.-Döngüleri Önlemek:
Switchler arasında birçok link olması bize birçok avantaj sağlar. Switchler bir linkin kopması durumunda Network ün tamamının düşmesine engel olurlar. Link fazlalığında yardımcı olurlar. Amacı böyle durumlarda oluşacak birçok problemi çözmektir. Çünkü Frameler eşzamanlı ve çok fazla olarak broadcasttan yayılır. Network döngüleri oluşur. Şimdi bu durumları inceleyelim.

1. Birden fazla link üzerinden birbirine bağlanmış switchler arasında bir ağ döngüsü oluşur. Eğer döngü ertelenemezse switchler sonsuza kadar bu işlemi devam ettirirler. Bu broadcast üzerinden çok fazla yayın yapılmasına sebep olur. Bunun sonucunda da broadcast hatası meydana gelir.



2. Bir aygıt benzer frame' lerden birçok tane alabilir ve bu frame' ler aynı anda farklı segmentlerden alınmış olabilir. Şekilde birçok segmentten aynı anda birçok frame' in nasıl geldiği gösterilmiştir.



3. MAC Adres tablosu karma karışık olacaktır. Hangi aygıtın nerede olduğu hakkında, switche birden fazla linklerden frameler gelecektir.
4. En büyük problemlerden biri; bir İnternetwork'te birçok döngüler meydana getirmektir. Bunun manası; bu döngüler diğer döngülerin içinde de oluşur. Bu durumda da networkümüz paket anahtarlamada çok kötü performans gösterir.

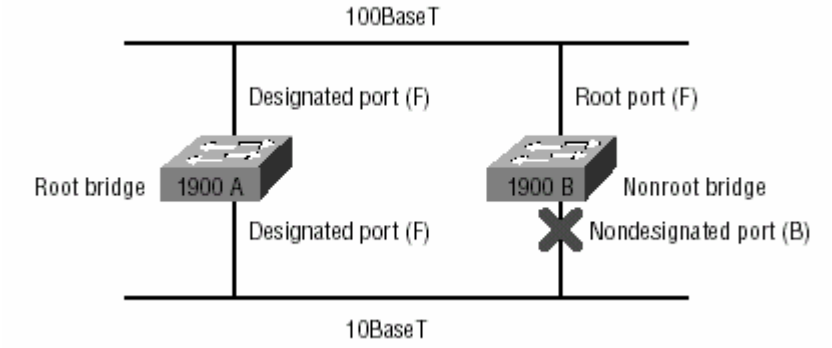
Spanning Tree Protokolü yukarıda sıraladığımız durumlara çözümler getirir.

2.1.2.SPANNİNG TREE PROTOKOLÜ
IEEE' nin 802.1d versiyonu STP yi meydana getirdi. Bütün Cisco switchleri STP' nin 802.1d versiyonu ile çalışır. STP' nin temel görevi ikinci katmanda oluşan network döngülerini

durdurmaktır. STP daima networkteki tüm linkleri ve network içerisinde oluşabilecek gereksiz network döngülerini engeller.

2.1.2.1Spanning-Tree İşlevi

STP Networkteki tüm linkleri bulur ve gereksiz linkleri kapatır. Network içinde meydana gelebilecek network döngülerini belirler. Yöntemi; Network topolojimizde belirlenecek olan temel bir köprü olabilir. Root-Bridge portları (designation port) tayin edilmiş portlar olarak belirler. Forwarding-state genel trafiği inceler ve bu bilgiyi switche gönderir.



Networkümüz deki diğer switchler şekilde görüldüğü gibi tayin edilmemiş köprüyü (nonroot ports) çağırır. Bununla birlikte en önemsiz yolun bağlı olduğu port bu trafikten ayrı tutulur. Yani bloklanır.

2.1.2.2.Root Bridge'in Seçimi

Switchler yada bridgeler Bridge Protocol Data Units (BPDUs) protokolü ile çalıştıkları zaman STP bilgi alış verişinde bulunurlar. BPDUs çoklu Frameleri kullanarak konfigürasyon mesajları gönderir. Her aygıtın Bridge ID si BPDUs kullanan diğer aygıtların hepsine gönderilir.

Bridge ID root portları ve network içindeki temel köprülere karar vermede kullanılıyor Bridge ID 8 byte uzunluğunda ve aygıtın MAC adresinin başına eklenir. Bütün aygıtlarda en çok kullanılan protokol IEEE STP 32768'dir.

2.1.2.3.Root Bridge:

Root bridge'e karar vermek öncelikle Bridge'in ve MAC adresin birleştirilmesiyle olur. Eğer iki switch yada bridge aynı seviyede ve önemde iseler o zaman MAC adres hangi birisinin ID si düşükse ona yazılır. Örneğin benim A ve B diye iki switchim var ve bunlar 32768 standardındalar. Böyle olduğundan MAC adresleri vardır. Eğer switch A nın MAC adresi 0000.0C00.1111.1111 ve switch b nin MAC adresi 0000.0C00.2222.2222 ise switch A root bridge olarak atanır. Aşağıda gösterilen network analizerin çıkışında görülen doküman bir 1900 serili switch ten alınan BPDU birimidir. BPDUs her iki saniyede bir değişiklikleri bildirir. Şunun altını çizelim bu Layer-3 değil Layer-2 switch e ait standartlardır. Burada bir çok başlık görülebilmektedir.

```
  Flags:        0x80  802.3
  Status:       0x00
  Packet Length:64
  Timestamp:    19:33:18.726314 02/28/2000
802.3 Header
  Destination:  01:80:c2:00:00:00
  Source:       00:b0:64:75:6b:c3
  LLC Length:   38
  802.2 Logical Link Control (LLC) Header
    Dest. SAP:    0x42  802.1 Bridge Spanning Tree
    Source SAP:   0x42  802.1 Bridge Spanning Tree
    Command:      0x03  Unnumbered Information
  802.1 - Bridge Spanning Tree
    Protocol Identifier:  0
    Protocol Version ID:  0
    Message Type:         0  Configuration Message
    Flags:                %00000000
    Root Priority/ID:     0x8000  /  00:b0:64:75:6b:c0
    Cost Of Path To Root: 0x00000000  (0)
    Bridge Priority/ID:   0x8000  / 00:b0:64:75:6b:c0
    Port Priority/ID:     0x80  /  0x03
    Message Age:                      0/256 seconds
(exactly 0seconds)
    Maximum Age:                      5120/256 seconds
(exactly 20seconds)
    Hello Time:                       512/256 seconds
(exactly 2seconds)
    Forward Delay:                      3840/256 seconds
(exactly 15seconds)
  Extra bytes (Padding):
    ........              00 00 00 00 00 00 00 00
  Frame Check Sequence:  0x2e006400
```

2.1.2.4.Tayin Edilmemiş Portun Seçimi

Port yada portların belirlenmesi root bridge' lerle haberleşerek belirlenir. Tabi önceliğimiz önceki diyagramda olduğu gibi bağlantıları doğru çizerek bunu kağıda dökmektir.

2.1.3.SPANNİNG TREE PORT DURUMU

STP kullanan switch yada bridge portları 4 farklı durumda çalışır.

Blocking: Blocking Framelerin ilerleyişini durdurur. BPDUs' yi dinler. Bütün portlar taki switch izin verene kadar blocking durumda olur.

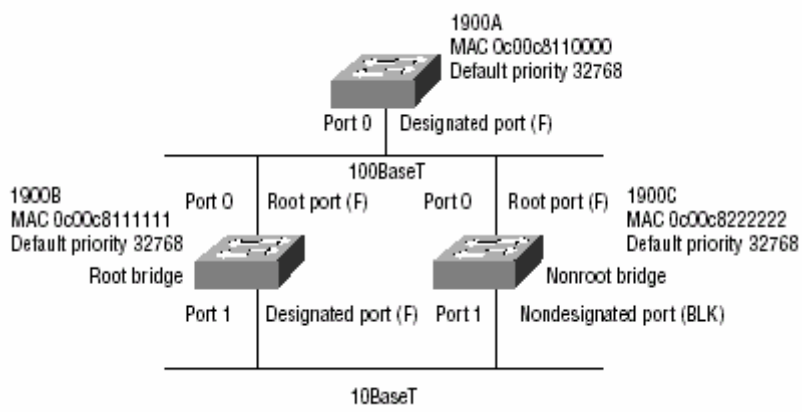
Listening: Network içerisinde data transferi yapılmadan önce network döngüsünün oluşmayacağından emin olununcaya kadar bu konumda bekler.

Learning: MAC adreslerini öğrenerek Filter tablosuna yükler. Ancak yine de framelerin transferini başlatmaz.

Forwarding: Tüm bu işlemlerden sonra network döngülerine karşı güvenlik sağlandıktan sonra bridge nin portundan bilgi alış verişine olanak verir.

STP Örneği:

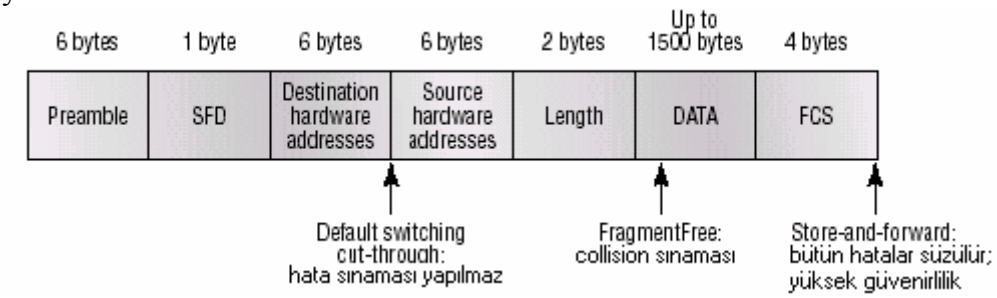
STP' nin bir İnternetwork'te nasıl çalıştığını görme açısından önemli bir örnek. Şekilde 3 switchte 32786 standardına sahiptir. Her switchin MAC adresi belirtilmiştir. Bunlara bakarak root bridge i belirleyebiliriz.



1900 A switch i en düşük değerli MAC adrese sahiptir. Dolayısıyla 1900 A switch i root bridge olarak atanır. 1900 B ve 1900 C nin root portunu belirlemek için bunların link bağlantılarının iyi incelenmesi gerekir. Switchler arası bağlantılara dikkat edilirse root switchin port 0 bağlantısı 100 Mbps bağlantı kullanmıştır ve en hızlı linktir. Burada "root port" switchler arasında port 0 olacaktır. Switchlerde Designated portun seçilmesinde bridge ID kullanılır. Root bridge her zaman olduğu gibi belirtilmelidir. 1900B ve 1900C switchlerine baktığımız zaman bunların ikisi arasında root bridge seçimi yapılırsa 1900B MAC adresi özelliğinden dolayı root bridge ve port1 Designated port olarak tanımlanır. Böylece 1900C switchi nonroot bridge ve port1' ide "nondesignated port" (Block) olarak belirlenir. Tüm bunlardan sonra 1900C switchi port1'ini Block konumunda tutacak ve networkte herhangi bir döngünün olmasına imkan tanımayacaktır.

2.1.4.LAN SWİTCH TİPLERİ

Paket anahtarlama için gizlilik, seçilmiş anahtarlama yöntemine bağlıdır. 3 çeşit anahtarlama yöntemi vardır.



2.1.4.1.Store Ve Forward: "store and forward " anahtarlama LAN switching in 3 tipinden birincisidir. Store and forward anahtarlama metodu ile LAN switchleri bütün frameler üzerindeki karta kopyalanır ve CRC(cyclic redundancy clock) hesaplanır. Çünkü bütün frameler kopyalanır. Framelerin boyu değiştirilerek switchler arası gizlilik sağlanmış olur.
Eğer bir CRC hatası verilirse yani eğer frame çok kısa ise (64 byte kadar CRC eklenir.) yada uzunsa 1518 byte kadar CRC eklenir. O Frame hata verir ve çıkarılır. Eğer frame herhangi bir hata vermezse LAN switch hedef donanım adresini alır ve arayüzün çıkışına gönderir.

2.1.4.2.Cut-Trough(Gerçek Zaman): Cut-Trough anahtarlama LAN Switching in diğer temel anahtarlamalarından birisidir. Bu metot ile LAN Switch sadece hedef adresi (DA)ön belleğine kaydeder. (İlk 6 byte preamble denilen hücre) daha sonra hedefin donanım adresini

MAC Switching tablosuna gönderir, arayüz çıkışı belirlenir. Çıkış arayüz belirlendikten sonra frame gönderilmesine başlanır.
Bir Cut-Trough switch düşük gizlilik sağlar. Çünkü hedef adres bulunduktan ve çıkış arayüz belirlendikten sonra frame gönderilmesine başlanır.
Bazı switchler bir hataya ulaşıncaya kadar port port Cut-Trough anahtarlama ile konfigüre edilebilirler. Bu noktada hata bulunduğu zaman durup; otomatik olarak "Store and Forward" modunu değiştirir. Ne zamanki hata giderildi bütün portlar başlangıçtaki haline döner ve portlar otomatik olarak Cut-Trough moduna geçer.
2.1.4.3.FragmentFree(Modified Cut Trough): FragmentFree, Cut Trough anahtarlamanın biraz değiştirilmiş formudur. Hangi switch Collision penceresinin bitmesini bekliyorsa ilk ilerleme sırasını ona verir. Eğer bir pakette bir hata algılanırsa, büyük oranda bunun sebebinin 64 kbyte lık kısımda olduğu düşünülür. FragmentFree modu Cut-Trough moda göre gizliliği artırmaz ve daha iyi hata sınaması yapar.

2.1.5.BİR SWİTCHE BAĞLANMA VE YÖNETME

Bu bölümde cisco tarafında üretilmiş iki farklı switchi inceleyeceğiz bunlar catalyst 1900 ve catalyst 5000'dir. Catalyst 1900 switchler anahtarlamada Cisco İnternetworking Operation System (IOS) ve Command-Line İnterface (CLI) ile çalıştırılırlar. Bu router konfigürasyonuna çok benzer
Cisco switchlerin çalıştırdığı iki operatör sistem vardır.
IOS-Based: Bu operatör sistemli switchlerde switch konfigürasyonu, cisco router'ların kullandığı operatör programına çok benzer olup switchin CLI arayüzü ile konfigüre edilebilir.

Set-Based: Set-Based' lar CLI konfigürasyon komutu ile kullanılırlar. Set-based CLI operatör sistemi Cisco Switchlerinin 2926 1948G 4000 5000 ve 6000 serilerinde kullanabiliriz. Şimdi 1900 ve 5000 serili catalyst switchleri tanıyalım.

Switch Blok Aygıtlarını Kablolama:

Routere benzer şekilde Cisco Catalyst switchimize console portundan yada Ethernet portundan fiziksel bağlantı yapabiliriz.

Console Portundan Bağlanma:

1900 ve 5000 serilerin her ikisi de bir console konnektörüne sahiptirler. 5000 serililerde bu sadece RS-232 konnektörle sonlandırılmıştır. Bu konnektor, switchlerle birlikte verilmektedir. 1900 serililerde console portu RJ-45 konnektor ile sonlandırılmıştır. Her iki console kabloları yuvarlaktır.

NOT: Console port tan yapılan bağlantılardan bahsetmek için Windows taki hyperterm gibi bir terminal emitör programına ihtiyacımız vardır. Ve şu ayarların yapılması gerekiyor.

- 9600 bps
- 8 data biti
- Parity (eşlik biti yok)
- 1 durdurma (stop) biti
- akış kontrol biti yok

Ethernet Portundan Bağlanma

Catalyst 1900 ve 2800 serili switchler sabit bir port tipine sahiptirler. 5000 serili switchler gibi modüler değildirler. 1900 2800 switchlerde sadece work station için 10Base-T, up linkler için 10Base-T yada FX kullanılır. Her switchin iki hızlı Ethernet up linkleri; 1912 yada 1914 model 10Base-T switch portlarından birini kullanır. 100Base-TX portlar switchlerde A ve B portlar olarak belirtilmişlerdir. Server'a bağlantılarda bu iki port 100 Mbps hızında çalışır. Başka bir switch ile bağlantı yapılırsa Cross Over kablo kullanılmalıdır.

Catalyst 5000 switchler kullandığımız Ethernet kartının ihtiyacına göre herhangi bir portundan 10 yada 100 Mbps hızlarından birinde çalıştırılabilirler. Çok daha yüksek hızlarda kartlar için her zaman ilk slottan iki tane 10Base-TX yada FX portları ile bağlantı kurmak mümkündür. Bütün aygıtların bağlantılarında 1900 2800 yada 5000 serili switchlerde aygıtlar arası kullanılacak kabloların maksimum uzunluğu 100 metre olmalıdır.

NOT:Eğer bir aygıt bir switche herhangi bir portundan bağlanırsa bu portu niteleyen LED yanar ve bu şekilde bekler. Eğer LED yanmazsa ya aygıt kapalıdır. Yada seçilen kabloda bir sorun vardır. Eğer LED yanıp sönüyorsa Auto Speed ve Dublex de bir sorun mevcut demektir. Bunlar ileriki kısımlarda anlatılacaktır.

2.1.5.1.Catalyst 5000 Switchlerin Açılışı

5000 serili switch açıldığında Flash bellekten yazılımı yükler. Ardından bizden güvenlik şifresini girmemizi ister. Şifreyi girdikten sonra Enter tuşuna basarak "console>" şeklinde hazır bekleyen bir komut satırı görürüz. Aşağıda 5000 serili switchin açıldığından itibaren hazır konuma geçtiği ana kadarki işlemleri gösterilmiştir.

```
BOOTROM Version 5.1(2), Dated Apr 26 1999 10:41:04
BOOT date: 08/02/02 BOOT time: 08:49:03
Uncompressing NMP image.  This will take a minute...
Downloading epld sram device please wait ...
Programming successful for Altera 10K10 SRAM EPLD
Updating epld flash version from 0000 to 0600

Cisco Systems Console

Enter password: [press return here]
1997 Mar 22 22:22:56 %SYS-5-MOD_OK:Module 1 is online
1997 Mar 22 22:23:06 %SYS-5-MOD_OK:Module 2 is online

Console>
```

2.1.5.2.Catalyst 1900 Switchlerin Açılışı

1900 switchin console portunda bir bağlantı yaptığımız zaman karşımıza bir menü çıkar. "K" tuşuna basarak komut satırı arayüzüne geçebiliriz. Ve "M" tuşuna basılarak bir menü sistemi vasıtasıyla switchin konfigürasyon edilmesine imkan verilir. "I" opsiyonu ise switchin IP konfigürasyonun yapılmasına imkan verir.

Aşağıda 1900 switchin açılış menüsü görülmektedir.

```
  1 user(s) now active on Management Console.

          User Interface Menu

       [M] Menus
       [K] Command Line
       [I] IP Configuration

  Enter Selection:  K

        CLI session with the switch is open.
        To end the CLI session, enter [Exit].

  >
```

2.1.6.CİSCO IOS VE SET-BASED KOMUTLARI:

Bu kısımda her iki tip switchin konfigürasyonunda kullanılan komutları göreceğiz.

- Password koymak
- Hostname kurmak
- IP adresi ve Subnet maskesi konfigürasyonu
- Arayüzlerin tanıtımı
- Arayüzlerde bir tanımlama ayarı
- Port hızlarının konfigürasyonu
- Dublex portları belirtme
- Konfigürasyonu onaylamak

2.1.6.1.Password Koymak:

İlk düşünmemiz gereken bir password koymak olmalı. Çünkü yetkilendirilmemiş birinin switche bağlanmasını istemeyiz. Hem User moda hem de Privileged moda şifre koyabiliriz.

NOT: seçeceğimiz password 4 karakterden küçük ve 8 karakterden büyük olmamalıdır.

a-)5000 Serili Switchlerde:
5000 serili switchlerde Enablepass ve Usermode password olmak üzere iki password koymak gerekiyor.
Şimdi satır satır bu konfigürasyonu inceleyelim

```
1997 Mar 21 06:31:54 %SYS-5-MOD_OK:Module 1 is online
1997 Mar 21 06:31:54 %SYS-5-MOD_OK:Module 2 is online
Console> en

Enter password:
Console> (enable) set password ?
Usage: set password
Console> (enable) set password [press enter]
Enter old password:
Enter new password:
Retype new password:
Password changed.
```

"enable password" de değişiklik yapabiliriz.

```
Console> (enable) set enablepass
Enter old password:
Enter new password:
Retype new password:
Password changed.
Console> (enable)
```

b-) 1900 Serili Switchlerde:
1900 serili switchlerin IOS yazılımı CLI arayüzünü çalıştırır. Usermode ve Enable modu password için komutlar, router için kullanılan komutlardan farklılık gösterir. Enable mode password komutu benzerdir. Fakat farklı Giriş seviyeleri (Access Level) seçmek Cisco routerler de isteğe bağlı 1900 switchlerde değildir.
Secret (gizli) password koyup Access Level değerini 15'e yapmak süper bir güvenlik sağlayacaktır. Telnet Password' ü Level 15 yada secret password etkin kılınarak aktif hale getirilebilir.
Switche console portundan bağlandığımız zaman çıkan menüde K tuşlayarak CLI moduna geçeriz. Daha sonra ***config t*** komutunu kullanarak Enable mod ve Global konfigürasyon moduna geçeriz.

```
1 user(s) now active on Management Console.

        User Interface Menu

     [M] Menus
     [K] Command Line
     [I] IP Configuration

Enter Selection:  K

      CLI session with the switch is open.
      To end the CLI session, enter [Exit].

#config t
Enter configuration commands, one per line.  End with
CNTL/Z
(config)#enable password ?
  level  Set exec level password
(config)#enable password level ?
  <1-15>  Level number
```

Dikkat edilirse bu programda Level-1 password' ü giren Usermode password' ü yönetebilir. Enable password Secret password' ün kurulmasıdır.

```
(config)#enable password level 1 todd
(config)#enable password level 15 sanfran
(config)#enable secret cisco
(config)#exit
#exit
CLI session with the switch is now closed.
Press any key to continue.
Catalyst 1900 Management Console
Copyright (c) Cisco Systems, Inc.  1993-1998
All rights reserved.
Enterprise Edition Software
Ethernet Address:      00-30-80-CC-7D-00
PCA Number:            73-3122-04
PCA Serial Number:     FAB033725XG
Model Number:          WS-C1912-A
System Serial Number:  FAB0339T01M
Power Supply S/N:      PHI031801CF
PCB Serial Number:     FAB033725XG,73-3122-04
-------------------------------------------------
1 user(s) now active on Management Console.
        User Interface Menu
     [M] Menus
     [K] Command Line
Enter Selection:  K
Enter password:  ****
      CLI session with the switch is open.
      To end the CLI session, enter [Exit].
>en
Enter password:  ****
#
```

2.1.6.2.Hostname Kurmak:

Bir switchteki hostname sadece yöresel anlam taşır. Bunun manası; isim ne olursa olsun, hostname bir networkte herhangi bir fonksiyona sahip değildir.switchte başarılı bir hostname kurulumu yaparsak, daha sonra switche bağlandığımız zaman isminden bu switchin hangi birisi olduğunu anımsayabiliriz.

a-)5000 Serili Switchlerde:

5000 serili switchlerde hostname kurulumu ***set prompt*** komutu ile olur.

```
Cisco Systems Console          Thu Mar 21 1997, 06:31:54

Enter password:
Console> en
Enter password:
Console> (enable) set prompt Todd5000
Todd5000 (enable) set prompt Todd5000>
Todd5000> (enable)
```

Prompt komutunu DOS işletim sisteminden hatırlarsak benzer bir işlevi vardı. Yukarıda dikkat edersek “set prompt” dan sonra ne yazdıysak harf harf aynısı hostname olarak alındı.

b-)1900 Serili Switchlerde:
1900 switchlerde bu komut routerlarda kullanılan komut ile aynı ve eş işlevlidir. ***Hostname*** komutu kullanılarak hostname ataması yapılır.

```
1 user(s) now active on Management Console.

        User Interface Menu

     [M] Menus
     [K] Command Line
     [I] IP Configuration
Enter Selection:  K
Enter password:  ****
      CLI session with the switch is open.
      To end the CLI session, enter [Exit].
>en
Enter password:  ****
#config t
Enter configuration commands, one per line.  End with
CNTL/Z
(config)#hostname Todd1900EN
Todd1900EN(config)#
```

2.1.6.3.IP İle İlgili Ayarlar:

Bir switch üzerinde IP konfigürasyonu kurulumu ile ilgili hiçbir çalışma yapmadık IP adres bilgisi Telnet yolu ile, yönetici yazılımlarla yada farklı VLAN ve diğer network fonksiyonları ile kurulabilir.

a-)5000 Serili Switchlerde:
5000 serili switchlerde IP adres bilgisini kurmakla "İn-hand" mantıksal arayüzünü çağırmış oluruz. Bu da ***set interface sc0*** komutu ile yapılır.

```
Todd5000> (enable) set int sc0 172.16.10.17 255.255.255.0
Interface sc0 IP address and netmask set.
```

Yaptığımız konfigürasyonu ***show interface*** komutunu kullanarak görebiliriz. ***show int*** komutu da aynı işlevi görür. Dikkat edersek yukarda yaptığımız ayar sonucu VLAN1 atanmış.

```
Todd5000> (enable) sh int
sl0: flags=51<UP,POINTOPOINT,RUNNING>
  slip 0.0.0.0 dest 0.0.0.0
sc0: flags=63<UP,BROADCAST,RUNNING>
 vlan 1 inet 172.16.10.17 netmask 255.255.255.0 broadcast
172.16.10.255
Todd5000> (enable)
```

Eğer switchimizde başka bir VLAN tanımlaması istiyorsak, yani VLAN yerine VLAN1 istiyorsak ***set int sc0*** komutu ile bu değişikliği yapabiliriz.
NOT: ileriki uygulamalarda eğer bir yönetici ataması yapmak istersek bunun VLAN1 olarak atanması gereklidir. Cisco switchler VLAN1' i diğer bütün VLAN' lardan farklı olarak algılar.

```
Todd5000> (enable) set int sc0 2
Interface sc0 vlan set.
Todd5000> (enable) sh int
sl0: flags=51<UP,POINTOPOINT,RUNNING>
  slip 0.0.0.0 dest 0.0.0.0
sc0: flags=63<UP,BROADCAST,RUNNING>
 vlan 2 inet 172.16.10.11 netmask 255.255.255.0 broadcast
172.16.10.255
Todd5000> (enable)
```

b-)1900 Serili Switchlerde:

1900 switchlerde IP konfigürasyonu ***ip address*** komutu ile yapılır. ***Show ip*** komutu ise yapmış olduğumuz konfigürasyonu görmede kullanılır.

```
Todd1900EN#sh ip
IP Address: 0.0.0.0
Subnet Mask: 0.0.0.0
Default Gateway: 0.0.0.0
Management VLAN:  1
Domain name:
Name server 1: 0.0.0.0
Name server 2: 0.0.0.0
HTTP server : Enabled
HTTP port :  80
RIP : Enabled
```

Birde switchlerde "Default Gateway" kurulumunun yapılması gerekiyor. Bunun komutu ise ***ip default-gateway*** dir.

```
Todd1900EN#config t
Enter configuration commands, one per line.  End with
CNTL/Z
Todd1900EN(config)#ip address 172.16.10.16 255.255.255.0
Todd1900EN(config)#ip default-gateway 172.16.10.1
Todd1900EN(config)#

Todd1900EN#sh ip
IP Address: 172.16.10.16
Subnet Mask: 255.255.255.0
Default Gateway: 172.16.10.1
Management VLAN:  1
Domain name:
Name server 1: 0.0.0.0
Name server 2: 0.0.0.0
HTTP server : Enabled
HTTP port :  80
RIP : Enabled
Todd1900EN#
```

Default gateway ayarını ***sh ip*** komutunu kullanarak görebiliriz.

2.1.6.4.Switch Arayüzleri:

Switch portlarının nasıl olduğunu anlamada önemlidir. 5000 serili switchler ***slot/port*** komutlarını kullanır.

a-)5000 Serili Switchlerde:

5000 serili switchlerde ***show port*** komutu kullanılarak portlara ait istatistikleri görme imkanımız vardır. 2900, 4000, 5000 ve 6000 serili switchlerin portları aktif hale getirilebilir. Bazı uygulamalarda portların konfigürasyonu gerekebilir. ***Set port enable*** ve ***set port disable*** komutları ile aktif yada pasif port ataması yapılabilir.

```
Todd5000> (enable) show port ?
Usage: show port
       show port <mod_num>
       show port <mod_num/port_num>
Todd5000> (enable) show port 2/1
Port  Name   Status  Vlan  Level    Duplex Speed Type
----- ------------------ ---------- ---------- ------------
 2/1        connect   2     normal   auto  auto 10/100BaseTX

Todd5000> (enable) set port disable 2/1
Port 2/1 disabled.
        Todd5000> (enable) sh port 2/1
Port  Name    Status     Vlan    Level  Duplex Speed  Type
----- ------------------ ---------- ------ ------ -----
 2/1          disabled   1      normal   auto  auto 10/100BaseTX

Todd5000> (enable) set port enable 2/1
Port 2/1 enabled.
Todd5000> (enable) sh port 2/1
Port  Name     Status     Vlan       Level  Duplex  Speed Type
----- ------  --------- ---------- ---------- ------ --- ------
2/1           connect    1       normal   auto  auto 10/100BaseTX
```

NOT: 5000 serili switchlerde bütün güncel konfigürasyonu görmek için ***show config*** komutu kullanılır.

b-)1900 Serili Switchlerde:

1900 serili switchlerde ***show*** yada ***interface*** komutlarından biri ile birlikte ***slot/port*** değeri ile birlikte yazılırsa istenilen konfigürasyon yapılabilir. ***İnterface*** komutu arayüzlere özgü konfigürasyon yapmamıza izin verir. 1900 switchler sadece bir tane slot ihtiva ederler. Buda sıfır (0)=zero olarak tanımlanmıştır.

```
Todd1900EN#config t
Enter configuration commands, one per line.  End with
CNTL/Z
Todd1900EN(config)#int ethernet ?
  <0-0>  IEEE 802.3
Todd1900EN(config)#int ethernet 0?
  /
Todd1900EN(config)#int ethernet 0/?
  <1-25>  IEEE 802.3
Todd1900EN(config)#int ethernet 0/1
Todd1900EN(config-if)#?
Interface configuration commands:

cdp             Cdp interface subcommands
description     Interface specific description
duplex          Configure duplex operation
exit            Exit from interface configuration mode
help            Description of the interactive help system
no              Negate a command or set its defaults
port            Perform switch port configuration
shutdown        Shutdown the selected interface
spantree        Spanning tree subsystem
vlan-membership VLAN membership configuration
```

İnterface leri birbirleri arasında değiştirmek için ***int e 0/#*** komutu kullanılır. İki tane olan Ethernet portunu konfigüre etmek ***interface fastethernet 0*** \# komutu ile olur. Ancak unutmamamız gereken nokta 10Base-T ile Fastethernet portuna bağlantı yapamayız.

```
Todd1900EN(config-if)#int e 0/2
Todd1900EN(config-if)#int e0/3
Todd1900EN(config-if)#exit
Todd1900EN(config)#int fastEthernet ?
  <0-0>  FastEthernet IEEE 802.3
Todd1900EN(config)#int fastEthernet 0/?
  <26-27>  FastEthernet IEEE 802.3
Todd1900EN(config)#int fastEthernet 0/26
Todd1900EN(config-if)#int fast 0/27
Todd1900EN(config-if)# [control+Z]
```

interface'i görmek için;

```
Todd1900EN#sh int e0/1
Ethernet 0/1 is Suspended-no-linkbeat
Hardware is Built-in 10Base-T
Address is 0030.80CC.7D01
MTU 1500 bytes, BW 10000 Kbits
802.1d STP State:  Forwarding     Forward Transitions: 1
[output cut]
 Todd1900EN#sh int f0/26
FastEthernet 0/26 is Suspended-no-linkbeat
Hardware is Built-in 100Base-TX
Address is 0030.80CC.7D1A
MTU 1500 bytes, BW 100000 Kbits
802.1d STP State:  Blocking     Forward Transitions: 0
[output cut]
```

2.1.6.5.İnterface Konfigürasyon Tanımı

Bir interface e isim verilebilir. Örneğin portun bulunduğu aygıtın işleyişine göre isim ataması yapılabilir. Hostname gibi buda sadece yerel tanımlamadır.

a-)5000 Serili Switchlerde:

set port name slot/port komutu ile interface isim ataması yapılabilir. Vereceğimiz isim 21 karaktere kadar olabilir.

```
Todd5000> (enable) set port name 2/1 Sales Printer
Port 2/1 name set.
Todd5000> (enable) sh port 2/1
Port  Name           Status   Vlan Level  Duplex Speed Type
----- -------------- ---------- ---- ----- ------ ------ -----
 2/1  Sales Printer  notconnect 2  normal  auto  auto 10/100BaseTX
```

b-)1900 Serili Switchlerde:

1900 switchler için description komutu kullanılır.

```
Todd1900EN#config t
Enter configuration commands, one per line.  End with
CNTL/Z
Todd1900EN(config)#int e0/1
Todd1900EN(config-if)#description Finance_VLAN
Todd1900EN(config-if)#int f0/26
Todd1900EN(config-if)#description trunk_to_Building_4
Todd1900EN(config-if)#
```

"description" u görmek için ***show int*** yada ***show running-config*** komutları kullanılır.

```
Port monitoring: Disabled
Unknown unicast flooding: Enabled
Unregistered multicast flooding: Enabled
Description: Finance_VLAN
Duplex setting: Half duplex
Back pressure: Disabled
Todd1900EN#sh run
Building configuration...
Current configuration:
hostname "Todd1900EN"
!
ip address 172.16.10.16 255.255.255.0
ip default-gateway 172.16.10.1
!
interface Ethernet 0/1
  description "Finance_VLAN"
!
[output cut]
```

2.1.6.6.Dublex ve Port Hızı Konfigürasyonu:

5000 serili switch üzerindeki 10/1000 çıkışlarının hepsi portun dublex ve hızını otomatikman algılayıp ayarlarlar. 1900 switchler ise sadece 12 yada 24 10Base-T portuna sahiptir ve değiştirilemez. Eğer sadece dublex olarak seçersek 1 yada 2 fastethernet portu gelir.

a-)5000 Serili Switchlerde:
bizim dublex ve hız ayarı yapmamız gerekli değildir. Otomatik olarak ayarlanır. Ancak konumunun algılanması doğru çalışmayabilir. Bunu test etmemiz gerekir.

```
Todd5000> (enable) set port speed 2/1 ?
Usage: set port speed <mod_num/port_num> <4|10|16|100|auto>
Todd5000> (enable) set port speed 2/1 100
Port(s) 2/1 speed set to 100Mbps.
```

Eğer port hızı "aotu" seçeneğini seçersek dublex ve hız her ikisi otomatik olarak tahsis edilir.

```
Todd5000> (enable) set port duplex 2/1 ?
Usage: set port duplex <mod_num/port_num> <full|half>
Todd5000> (enable) set port duplex 2/1 full
Port(s) 2/1 set to full-duplex.
Todd5000> (enable) ^C
```

NOT: CTRL+C DOS ta olduğu gibi devam etmekte olan bir işlemi kesmede kullanılır. Show port komutu ile hızı ve dublex i görebiliriz.

```
Todd5000> (enable) sh port 2/1
Port  Name          Status     Vlan   Level  Duplex Speed Type
----- ------------- ---------- ------- ------ ------ ----- ----
 2/1  Sales Printer notconnect  2    normal full  100 10/100BaseTX
```

b-)1900 Serili Switchlerde:
1900 switchlerde sadece dublex i ayarlayabiliriz. Çünkü bütün portların hızı sabittir. Dublex ayarı ise dublex komutu kullanılarak yapılır.

```
Todd1900EN(config)#int f0/26
Todd1900EN(config-if)#duplex ?
  auto              Enable auto duplex configuration
  full              Force full duplex operation
  full-flow-control  Force full duplex with flow control
  half              Force half duplex operation
Todd1900EN(config-if)#duplex full
```

tabloda 1900 2800 ve 2900 XL switchlerde mevcut farklı dublex opsiyonları ve tanımlamaları verilmiştir.

Dublex opsiyonları

Auto	100Base-TX port olan bütün portları "auto-negoiate" moda set eder.
Full	10 yada 100Mbps portlar full dublex moda girer.
Full-flow-control	sadece 100Base-TX portlarda çalışır.
Half	10Base-T portlar için sadece bu modda çalışır.

Show interface komutu ile dublex konfigürasyonu görülebilir.

```
Todd1900EN#sh int f0/26
FastEthernet 0/26 is Suspended-no-linkbeat
Hardware is Built-in 100Base-TX
Address is 0030.80CC.7D1A
MTU 1500 bytes, BW 100000 Kbits
802.1d STP State:  Blocking     Forward Transitions:  0
Port monitoring: Disabled
Unknown unicast flooding: Enabled
Unregistered multicast flooding: Enabled
Description: trunk_to_Building_4
Duplex setting: Full duplex
Back pressure: Disabled
```

2.1.6.7.IP SINAMASI

IP konfigürasyonu bakımından önemlidir. Telnet te ping program gibi işlevi mevcut 5000serili switchlerde traceroute komutu ile istatistikler alınabilir.

a-)5000 Serili Switchlerde:

Networkteki switchin ping telnet ve traceroute komutları ile IP hizmetleri test edilebilir.

```
Todd5000> (enable) ping 172.16.10.10
172.16.10.10 is alive
Todd5000> (enable) telnet ?
Usage: telnet <host> [port]
       (host is IP alias or IP address in dot notation:
a.b.c.d)
Todd5000> (enable) traceroute
Usage: traceroute [-n] [-w wait] [-i initial_ttl] [-m max_
ttl]
       [-p dest_port] [-q nqueries] [-t tos] host [data_
size]
(wait = 1..300, initial_ttl = 1..255, max_ttl = 1..255
dest_port = 1..65535, nqueries = 1..1000, tos = 0..255
data_size = 0..1420, host is IP alias or IP address in
dot notation: a.b.c.d)
```

b-)1900 Serili Switchlerde:

1900 switchlerde ping komutu kullanılarak bazı istatistikler elde edilebilir.

```
Todd1900EN#ping 172.16.10.10
Sending 5, 100-byte ICMP Echos to 172.16.10.10, time out
is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max
0/2/10/ ms
Todd1900EN#telnet
           ^
% Invalid input detected at '^' marker.
```

2.1.6.8.Switch Konfigürasyonunu Silmek :

Yaptığımız konfigürasyonların hepsi NVRAM e yazılır. Bunları silmek mümkündür.

a-)5000 Serili Switchlerde:

clear config all komutu kullanılarak NVRAM deki kaydedilmiş bütün konfigürasyonlar silinir. ***Erase all*** komutu ile Flash bellekteki bilgileri siler=>DİKKAT

```
Todd5000> (enable) clear config ?
Usage: clear config all
       clear config <mod_num>
       clear config rmon
       clear config extendedrmon
Todd5000> (enable) clear config all
This command will clear all configuration in NVRAM.
This command will cause ifIndex to be reassigned on the
next system startup.
Do you want to continue (y/n) [n]? y
........
..................
System configuration cleared.
```

bir diğer komut;

```
Todd5000> (enable) erase all
FLASH on Catalyst:
Type            Address             Location
Intel 28F016    20000000            NMP (P3) 8MB SIM
Erasing flash sector...
Todd5000> (enable)
Todd5000> (enable) sh flash

File      Version        Sector   Size    Built
--------- -------------- -------- ------- -------
```

Eğer ***erase all*** komutunu yazarsak bu programın akışını CTRL C ile kesme şansımız yoktur. Bu geri yüklemesini anca ***copy tftp flash*** komutu ile flash yazılımı tekrar yüklenir.

b-)1900 Serili Switchlerde:

1900 Switchlerde NVRAM bilgilerinin silinmesi ***delete NVRAM*** komutu ile yapılır. ve delete VTP komutu ile de VTP konfigürasyonu silinir.

```
Todd1900EN#delete ?
  nvram  NVRAM configuration
  vtp    Reset VTP configuration to defaults
Todd1900EN#delete nvram
This command resets the switch with factory defaults.  All
system parameters will revert to their default factory
settings.  All static and dynamic addresses will be
removed.
Reset system with factory defaults, [Y]es or [N]o?  Yes
```

VLAN' LAR (VİRTUAL LAN)

Bundan önceki anlattığımız konularda genelde tekil bileşenleri inceleyip bunların iç dinamiğini, konfigürasyonunu inceledik. Bu bölümde ise grup çalışması, network yönetimi vb. konuların üzerinde duracağız. VLAN larıda network yönetimine doğru çıktığımız ilk basamak olarak düşünebiliriz. VLAN bölümünde bir switch üzerinden birbirlerine bağlı bilgisayarlar ve benzeri sayısal aygıtlar arasında bir çalışma grubu kurma ve bir yönetici ile bunları idare etme gibi somut konular üzerinde duracağız. Bu bölümde şunları inceleyeceğiz.

- VLAN nedir.
- Set-Based ve IOS-Based switchlerde VLAN konfigürasyonu
- VLAN Trunk Protocol Nedir.
- Frame leri etiketlendirme ve teşhis etme metotları

2.2.1.VLAN' LARIN ÖZELLİKLERİ:

VLANs Layer-2 anahtarlama networkünde Broadcast domain leri parçalamakta kullanılır. Bir Layer-2 VLAN anahtarlamalı internetworkte farklı VLANs ların diğerleri ile iletişiminde sabit bir routere ihtiyaç vardır.

Bir internetworkte VLANs oluşturmanın birçok faydası vardır. Normal bir networkte bizim bütün güvenliğimiz password lerdir. Ve bütün kullanıcılar bütün aygıtları görebilir. VLANs oluşturarak layer-2 anahtarlama ile birçok problemin üstesinden gelebiliriz.

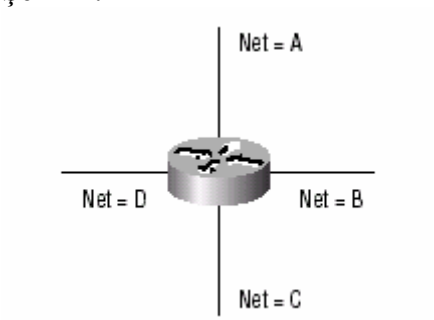
2.2.2.BROADCAST KONTROLÜ:

Broadcast' ler bütün protokollerde bulunur. Uygulamalar internetworkte çalıştırılırlar. VLANs küçük broadcast domain'leri tanımlar. Bazı eski uygulamaları yenileyerek bant genişliği gereksinimini azaltır ve yeni oluşturulan uygulamalarla bant genişliğini ekonomik kullanabilirler. Bu uygulamalar, geniş bağlantı ve multimedya uygulamalarıdır.

2.2.3.ÇÖKMÜŞ OMURGA (COLLAPSED BACKBONE) VE VLAN

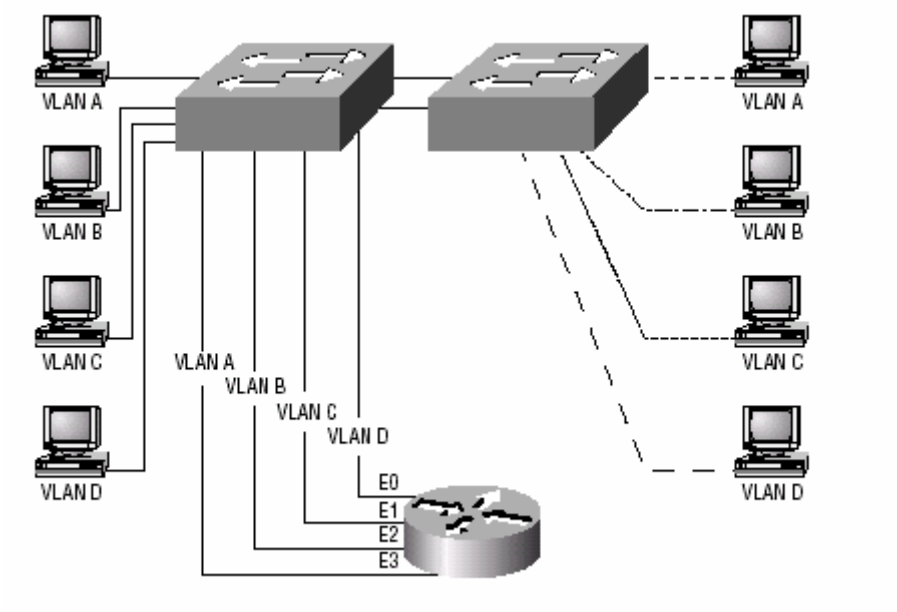
Geleneksel çökmüş omurga denilen yapıyı anlamak bize bir VLAN' ın bir switche nasıl baktığını anlamada yardımcı olur. Şekilde bir routere fiziksel LANs bağlantısı ile oluşturulmuş "çökmüş omurga" örneği gösterilmiştir.

Şekil-2.1



Her bir network tek bir routere bağlıdır ve her networkün kendi mantıksal numarası vardır. Özel bir fiziksel networke bağlanmış her bir düğüm internetwork teki iletişimin sağlanabilmesi için network numaralarını izlemelidir. Şimdi aşağıdaki şekle bakarak anahtarlamaları inceleyelim.

Şekil-2.2



Biz aslında her yerde router lara ihtiyaç duymayız. Router in kullanılabildiği bir uygulamada switchte kullanılabilir. Yukarıdaki şekilde 4 adet VLAN yada broadcast domain vardır. Her bir VLAN içindeki düğümler diğer bir düğüm ile bağlantı yapabilir. Bir VLAN içinde konfigürasyon esnasında düğümler bunu aslında bir "kırılmış omurga" olarak algılar. (şekil 2.1 deki gibi). Şekil 2.1 deki bu kullanıcıların farklı bir düğüme yada farklı bir networke bağlanabilmesi için neye ihtiyaçları vardır? Cevap ya bir routere, yada diğer bir Layer-3 aygıta.

2.2.4.VLAN ÜYELİĞİ:

Bir VLANs oluştururken switch portlarını işaretlememiz gerekir. Statik ve dinamik olmak üzere VLAN port konfigürasyonunun iki tipi vardır. Bir statik VLAN daha az bir çalışma gerektirir. Ancak yöneticinin konfigürasyonu açısında çok daha zordur.
Bir dinamik VLAN' ın daha çok işlevi vardır. Ve kurulumu kolaydır.

Statik VLANs

Bir Statik VLAN da yönetici VLAN a switch portlarını belirtmelidir. Ve yönetici port atamalarını değiştirmedikçe değişmez. Bu VLANs oluşturulurken en tipik yöntemdir. Ve güvenilirdir.

Dinamik VLANs

Eğer yönetici bir çalışma yapmak isterse ve database içindeki bütün aygıtların donanım adreslerini belirlese, internetwork teki kullanıcılar dinamik VLAN görevlendirilerek belirlenebilir. Kullanıcı yazılım programı kullanarak, donanım adresleri, protokoller yada dinamik VLANs oluşturma uygulamaları etkin kılınabilir. Örneğin, MAC adresleri merkezi VLAN yönetici uygulamasında belirlenmiştir. Eğer bir düğüm, tahsis edilmemiş switch portuna takılırsa; VLAN yönetici database donanım adresini tahsisini ve bağlı bulunduğu doğru VLAN portunu konfigüre edebilir.

2.2.5.Statik VLAN Konfigürasyonu:

a-)5000 Serili switchlerde:

Catalyst 5000 serili switchlerde VLANs konfigürasyonu ***set vlan***[***vlan#***][***isim***] komutu ile yapılır.

```
Todd5000> (enable) set vlan 2 name Sales
Vlan 2 configuration successful
Todd5000> (enable) set vlan 2 2/1-2
VLAN  Mod/Ports
---- ----------------------
2     1/1-2
      2/1-2

Please configure additional information for VLAN 2.
Todd5000> (enable)
```

b-)1900 Serili Switchlerde:

```
>en
#config t
Enter configuration commands, one per line.  End with CNTL/Z
(config)#hostname 1900EN
1900EN(config)#vlan 2 name sales
1900EN(config)#vlan 3 name marketing
1900EN(config)#vlan 4 name mis
1900EN(config)#exit
```

VLAN' lar oluşturulduktan sonra ***show vlan*** komutu kullanılarak konfigürasyon görülebilir.

```
1900EN#sh vlan

VLAN Name                Status     Ports
---------------------------------------
1    default             Enabled    1-12, AUI, A, B
2    sales               Enabled
3    marketing           Enabled
4    mis                 Enabled
1002 fddi-default        Suspended
1003 token-ring-defau    Suspended
1004 fddinet-default     Suspended
1005 trnet-default       Suspended
---------------------------------------
VLAN Type           SAID   MTU  Parent RingNo BridgeNo Stp  Trans1 Trans2
------------------------------------------------------------------------
1    Ethernet       100001 1500   0      0      0        Unkn 1002   1003
2    Ethernet       100002 1500   0      1      1        Unkn 0      0
3    Ethernet       100003 1500   0      1      1        Unkn 0      0
4    Ethernet       100004 1500   0      1      1        Unkn 0      0
1002 FDDI           101002 1500   0      0      0        Unkn 1      1003
1003 Token-Ring     101003 1500   1005   1      0        Unkn 1      1002
--More--
1004 FDDI-Net       101004 1500   0      0      1        IEEE 0      0
1005 Token-Ring-Net 101005 1500   0      0      1        IEEE 0      0
------------------------------------------------------------------------
```

Her bir portu istediğimiz VLAN' a ***vlan-membership*** komutunu kullanarak atayabiliriz. VLANs' ı port port konfigüre de edebiliriz.

```
1900EN#config t
Enter configuration commands, one per line.  End with CNTL/Z
1900EN(config)#int e0/2
1900EN(config-if)#v?
vlan-membership
1900EN(config-if)#vlan-membership ?
  dynamic  Set VLAN membership type as dynamic
  static   Set VLAN membership type as static
1900EN(config-if)#vlan-membership static ?
  <1-1005>  ISL VLAN index
1900EN(config-if)#vlan-membership static 2
1900EN(config-if)#int e0/4
1900EN(config-if)#vlan-membership static 3
1900EN(config-if)#int e0/5
1900EN(config-if)#vlan-membership static 4
1900EN(config-if)#exit
1900EN(config)#exit
```

Bu konfigürasyonu ***show vlan #*** komutunu kullanarak görebiliriz.

```
1900EN#sh vlan

VLAN Name                Status     Ports
---------------------------------------
1    default             Enabled    1, 3, 6-12, AUI, A, B
2    sales               Enabled    2
3    marketing           Enabled    4
4    mis                 Enabled    5
1002 fddi-default        Suspended
1003 token-ring-defau    Suspended
1004 fddinet-default     Suspended
1005 trnet-default       Suspended
---------------------------------------
VLAN Type          SAID   MTU  Parent RingNo BridgeNo Stp  Trans1 Trans2
------------------------------------------------------------------------
1    Ethernet      100001 1500   0      0      0        Unkn 1002   1003
2    Ethernet      100002 1500   0      1      1        Unkn 0      0
3    Ethernet      100003 1500   0      1      1        Unkn 0      0
4    Ethernet      100004 1500   0      1      1        Unkn 0      0
1002 FDDI          101002 1500   0      0      0        Unkn 1      1003
1003 Token-Ring    101003 1500   1005   1      0        Unkn 1      1002
1004 FDDI-Net      101004 1500   0      0      1        IEEE 0      0
1005 Token-Ring-Net 101005 1500  0      0      1        IEEE 0      0
------------------------------------------------------------------------
1900EN#sh vlan 2
VLAN Name                Status     Ports
---------------------------------
2    sales               Enabled    2
---------------------------------

VLAN Type       SAID   MTU  Parent RingNo BridgeNo Stp  Trans1 Trans2
------------------------------------------------------------------
2    Ethernet      100002 1500   0      1      1        Unkn 0      0
------------------------------------------------------------------
1900EN#
```

2.2.6.TRUNKİNG

Trunk linkler; iki switch, bir router bir switch yada bir switch ile bir Server arasındaki, point-to-point, 100 yada 1000 Mbps linkleridir. Trunk edilmiş linkler çoklu bağlanmış VLAN' ların trafiğini taşırlar.

2.2.6.1.Trunk Portlarının Konfigürasyonu:

a-)5000 Serili Switchlerde:

5000 serili switchlerde ***set trunk*** komutu kullanılır.

```
Console> (enable) set trunk 2/12 ?
Usage: set trunk <mod_num/port_num>
[on|off|desirable|auto|nonegotiate] [vlans] [trunk_type]
(vlans = 1..1005 An example of vlans is 2-10,1005)
       (trunk_type = isl,dot1q,dot10,lane,negotiate)
Console> (enable) set trunk 2/12 on isl
Port(s) 2/12 trunk mode set to on.
Port(s) 2/12 trunk type set to isl.
Console> (enable) 1997 Mar 21 06:31:54
%DTP-5-TRUNKPORTON:Port 2/12 has become k
```

2/12 portu ISL (Switchler Arası Link Protokolü =İnter-switch Link Protocol) protokolünü kullanarak bir trunk port olur. Biz burada VLANs' a özel bir trunk oluşturmadık. Ancak yapılacak ayarlamalarla bütün VLANs' lar edilir.

```
Console> (enable) set trunk 2/12 on 1-5 isl
Adding vlans 1-5 to allowed list.
Please use the 'clear trunk' command to remove
vlans from allowed list.
Port(s) 2/12 allowed vlans modified to 1-1005.
Port(s) 2/12 trunk mode set to on.
Port(s) 2/12 trunk type set to isl.
```

Trunk portu VLANs' dan çıkarmak için ***clear VLAN*** komutu kullanılır. Bir trunk portun gösterilmesi için bunun kullanabileceğimiz farklı opsiyonlarını izah etmemiz gerekir.

"On" seçeneği: Switch portu sürekli trunk port olarak tanımlanır.

"Off" seçeneği**: "on"** olarak seçilmiş trunk portu kapatır.

"**Auto"**seçeneği: Eğer bir port trunk port yapılmak isteniyor, ancak komşu portlardan biride trunk port olması istenmiyorsa sadece 1 tane trunk port seçilebilir. Bu bütün portları değiştirir. Çünkü "auto" switch portları için hiçbir zaman soru sormaz. Portların her ikisi "auto" seçilmedikçe bu iki port birden trunk port olarak atanmaz.

"Desirable" seçeneği: Eğer bir portu trunk port olarak tanımlamak istiyor ve komşu port ta bir trunk port olarak atanmış ise bunu "desirable" yada "auto" seçerek trunk port atayabiliriz.

"Nonegotiate" seçeneği: Bir portu sürekli trunk port yapar ancak port iletişim için DTP Framelerini kullanamaz. Switch olmayan bir aygıt ile switch bağlantısında bir DTP problemi varsa ***set trunk*** komutu kullanılırken birde ***nonegotiate*** komutu kullanılmalıdır. Bu portun trunk port olmasına izin verir fakat DTP framelerinin geçişine olanak tanımaz.

b-)1900 Serili Switchlerde:

1900 switchler aynı opsiyonlara sahiptir.

```
1900EN#config t
Enter configuration commands, one per line.
End with CNTL/Z
1900EN(config)#int f0/26
1900EN(config-if)#trunk ?
  auto         Set DISL state to AUTO
  desirable    Set DISL state to DESIRABLE
  nonegotiate  Set DISL state to NONEGOTIATE
  off          Set DISL state to OFF
  on           Set DISL state to ON
1900EN(config-if)#trunk auto
```

2.2.6.2.Trunk Linklerinden VLANs Silinmesi:

a-)5000 Serili Switchlerde:

Trunk edilmiş bir VLAN' ı silme ***clear trunk slot/port vlans*** komutu ile olur.

```
Console> (enable) clear trunk 2/12 5-1005
Removing Vlan(s) 5-1005 from allowed list.
Port 1/2 allowed vlans modified to 1-4
```

b-)1900 Serili Switchlerde:

```
1900EN(config-if)#no trunk-vlan ?
  <1-1005>  ISL VLAN index
1900EN(config-if)#no trunk-vlan 5
1900EN(config-if)#
```

2.2.6.3.Trunk Linklerinin Sınanması:

Trunk linklerini sınamak ***show trunk*** komutu kullanılarak yapılır. Eğer birden fazla trunk edilmiş port var ise ***show trunk*** [***port_numarası***] komutu ile istatistikler görülebilir.

```
Console> (enable) sh trunk 2/12
Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 2/12     on           isl            trunking      1
Port      Vlans allowed on trunk
--------  ------------------------------------------------------
 2/12     1-4
Port      Vlans allowed and active in management domain
--------  ------------------------------------------------------
 2/12     1
Port      Vlans in spanning tree forwarding state and not pruned
--------  ------------------------------------------------------
 2/12     1
Console> (enable)
```

1900 switchte benzer komutlar vardır. Ancak sadece Fastethernet 26 ve 27 portlarında çalıştırılabilir.

```
  A  Trunk A
  B  Trunk B
1900EN#sh trunk a
DISL state: Auto, Trunking: On, Encapsulation type: ISL

1900EN#sh trunk ?
  A  Trunk A
  B  Trunk B
1900EN#sh trunk a ?
  allowed-vlans   Display allowed vlans
  joined-vlans    Display joined vlans
  joining-vlans   Display joining vlans
  prune-eligible  Display pruning eligible vlans
  <cr>
1900EN#sh trunk a allowed-vlans
1-4, 6-1004
1900EN#
```

2.2.7.VLAN Trunk Protokol (VTP)

VTP Cisco tarafından bütün VLANs konfigürasyonunu yönetmek amacı ile üretilmiştir. VTP yöneticinin, VLANs eklemesine, çıkarmasına, yeniden adlandırmasına, değişiklik istenen bütün switchlerde bunları gerçekleştirmesine izin verir.

2.2.7.1.VTP Operasyon Modları:

Bir VTP Domain i içindeki operasyonların üç farklı modu vardır; server, client, ve transparent

Server: VTP server modu bütün catalyst switchler için farklıdır. Bir server mod içinde aşağıda gösterilen maddelerin tamamlanabilmesi gerekir.

- VTP Domaininde VLANs oluşturma, ekleme, silme.
- VTP bilgisini değiştirme. Server moddaki bir switchte herhangi bir değişiklik yapmak, bütün VTP Domainine ilan vermektir.

Client: VTP Clientler VTP Serverlardan bilgi alırlar. Ancak herhangi bir değişiklik yapamazlar.

Transparent: VTP Transparent switchler VTP domainine katılmaz. Ancak trunk linkler arasındaki ilanlar VTP de ilerleyebilir. VTP Transparent switchlerin VLANs ekleyebilme ve silebilme yeteneği vardır.

2.2.8.VTP Konfigürasyonu:

VTP Domain konfigürasyonuna başlamadan önce bilmemiz gereken çeşitli seçenekler vardır.

1. Çalıştıracağımız VTP numarasını tekrar gözden geçirmek.
2. Bir switchin var olan bir domaine üye edilmesi yada yeni bir domain oluşturulmasına karar vermek. Eğer var olan bir domaine eklenecekse adının password ünün bulunması gerekir.
3. İnternetworkteki her switch için VTP modu seçilmeli.

2.2.8.1.VTP Versiyonunu Konfigüre Etmek

Cisco switchlerde konfigüre edilebilir VTP' nin iki farklı versiyonu vardır. Versiyon 1 ve versiyon 2. Versiyon 1, bütün switchlerde farklı olan VTP versiyonudur. Örneğin Token Ring Çalıştırmak istiyorsak Versiyon 2 seçilmelidir.

```
Console> (enable) set vtp v2 enable
This command will enable the version 2 function
in the entire management domain.
All devices in the management domain should
be version2-capable before enabling.
Do you want to continue (y/n) [n]? y
VTP domain  modified
Console> (enable)
```

Sadece 1900 switchler versiyon 1 i kullanır.

```
1900EN(config)#vtp ?
  client       VTP client
  domain       Set VTP domain name
  password     Set VTP password
  pruning      VTP pruning
  server       VTP server
  transparent  VTP transparent
  trap         VTP trap
```

2.2.8.2.Domain Konfigürasyonu:

Hangi versiyonun çalıştırılacağına karar verildikten sonra ilk switchte VTP Domain ismi ve passwordu kurulmalıdır. VTP ismi 32 karakter uzunluğunda olabilir. 5000 switchlerde VTP Domain passwordu en az 8 karakter en fazla 64 karakter uzunluğunda olmalıdır.

```
Console> (enable) set vtp domain ?
Usage: set vtp [domain <name>] [mode <mode>]
[passwd <passwd>]
[pruning <enable|disable>]
[v2 <enable|disable>
       (mode = client|server|transparent
        Use passwd '0' to clear vtp password)
Usage: set vtp pruneeligible <vlans>
       (vlans = 2..1000
        An example of vlans is 2-10,1000)
Console> (enable) set vtp domain Globalnet
VTP domain Globalnet modified
Console> (enable)
```

1900 switchlerde "password seçeneğimiz yoktur.

```
1900EN(config)#vtp domain ?
  WORD  Name of the VTP management domain
1900EN(config)#vtp domain Globalnet ?
  client       VTP client
  pruning      VTP pruning
  server       VTP server
  transparent  VTP transparent
  trap         VTP trap
  <cr>
1900EN(config)#vtp domain Globalnet
1900EN(config)#
```

2.2.8.3.VTP Modu Konfigürasyonu:

Bir server gibi birincil switchi oluşturur. Daha sonra bağlı bulunduğu switchleri oluşturur.

```
Console> (enable) set vtp domain
Usage: set vtp [domain <name>] [mode <mode>]
[passwd <passwd>]pruning <enable|disable>]
[v2 <enable|disable>
(mode = client|server|transparent
        Use passwd '0' to clear vtp password)
Usage: set vtp pruneeligible <vlans>
       (vlans = 2..1000
        An example of vlans is 2-10,1000)
Console> (enable) set vtp domain Globalnet mode server
VTP domain Globalnet modified
```

1900 switchlerde ise

```
1900EN(config)#vtp ?
  client       VTP client
  domain       Set VTP domain name
  password     Set VTP password
  pruning      VTP pruning
  server       VTP server
  transparent  VTP transparent
  trap         VTP trap
1900EN(config)#vtp client ?
  pruning  VTP pruning
  trap     VTP trap
  <cr>
1900EN(config)#vtp client
```

2.2.8.4.VTP Konfigürasyonunun Sınanması:

Show vtp domain yada ***show vtp statistics*** komutları kullanılarak VTP bilgisi görüntülenebilir. Ancak ***show vtp domain*** komutu 1900 switchlerde çalıştırılamaz.

```
Console> (enable) sh vtp domain
Domain Name              Domain Index VTP Version Local Mode  Password
-------------------- ------------ ----------- -----------
Globalnet       1            2           server
Vlan-count Max-vlan-storage Config Revision Notifications
---------- ---------------- --------------- -------------
5          1023             1               disabled

Last Updater    V2 Mode  Pruning  PruneEligible on Vlans
--------------- -------- -------- -------------------------
172.16.10.14    disabled disabled 2-1000
Console> (enable)
```

a-)5000 Serili Switchlerde:

show vtp statistics komutu ile VTP alınan ve gönderilen yayın mesajlarının bir özeti görüntülenebilir. Bir konfigürasyon hatası alınırsa;

```
Console> (enable) sh vtp stat
VTP statistics:
summary advts received          0
subset  advts received          0
request advts received          0
summary advts transmitted       5
subset  advts transmitted       2
request advts transmitted       0
No of config revision errors    0
No of config digest errors      0
VTP pruning statistics:
Trunk     Join Transmitted  Join Received  Summary advts received from
                                           non-pruning-capable device
--------  ----------------  -------------  --------------------------
 2/12     0                 0              0
Console> (enable)
```

b-)1900 Serili Switchlerde:

```
1900EN#sh vtp stat
         Receive Statistics                    Transmit Statistics
-------------------------------  -----------------------------
Summary Adverts                0  Summary Adverts             0
Subset Adverts                 0  Subset Adverts              0
Advert Requests                0  Advert Requests            56
Configuration Errors:
  Revision Errors              0
  Digest Errors                0
VTP Pruning Statistics:
Port    Join Received    Join Transmitted  Summary Adverts received
                                           with no pruning support
----    -------------    ---------------  -----------------------
A       0                0                0
B       0                0                0
             1900EN#
```

ROUTER KONFİGÜRASYONU

3.1.CISCO IOS ROUTER

Cisco IOS Catalyst 1900 lerdekine benzer şekilde ve bir çok routerin kullandığı arayüzdür. Cisco Router IOS yazılımı Cisco aygıtlarda aşağıda gösterilen görevleri yerine getirir.

- Network protokolleri ve fonksiyonları taşıma
- Aygıtlar arası yüksek hızlı bağlantı
- Güvenliği sağlamak, girişi kontrol etmek ve network kullanımına izin vermek
- Zengin networklere bağlantılar için network dağılımının güvenilirliğini sağlamak.

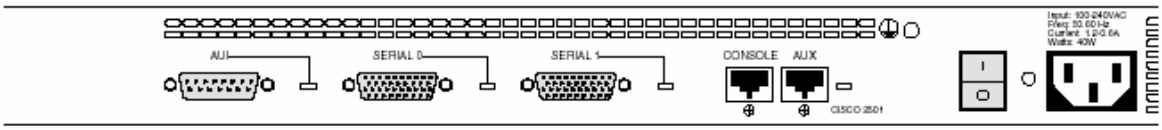
Modem veya Telnet ile bir routerin Console portundan Cisco IOS arayüzüne girebiliriz.

3.1.1.CİSCO ROUTERE BAĞLANMA

Router konfigürasyonu yapmak, yapılan konfigürasyonu teyit etmek yada routere ait istatistikleri görmek için routere bağlanabiliriz. Cisco Routere bağlanmanın farklı yolları vardır ama birinci tercihimiz Console portundan bağlanmaktır.

Console portundan genellikle RJ-45 konnektörle bağlantı yapılır. Routere bağlantı için diğer bir yol ise "auxiliary port" girişidir. Console portu gibidir ve aynı amaçlar için kullanılır. Modem bağlantısı (dial up) ile uzaktan bağlantı yapmakta mümkündür. Üçüncü yol ise Telnet bağlantısıdır. Bir router üzerindeki aktif arayüzlerden (Ethernet yada seri port) birinden bağlantı yapılır. Aşağıda 2051 serili bir router ın arka panel yapısı vardır.

2501 router



2501 Router, WAN bağlantıları için 2 seri arayüze ve Ethernet Networkü bağlantısı için 10 Mbps' lik AUI (Attachment Unit İnterface) arayüzü vardır. Ayrıca RJ-45 konnektörlü bir Console ve bir Auxiliary bağlantı arayüzüne sahiptir.

3.1.2.ROUTER' IN AÇILIŞI

Bir router' ı ilk açtığımız zaman, router POST (power on self test) işlemi yapar. Bu işlem bittikten sonra Flash bellekten Cisco IOS yazılımını çalıştırır. Flash bellek Elektronik olarak silinebilir, programlanabilir sadece okunabilir bellek (EEPROM)'dir. IOS yazılımı geçerli değişiklikleri ile birlikte NVRAM' da saklı olan Startup-config dosyasını çağırıp geçerli konfigürasyona bakar. (Bizim yapmış olduğumuz yada yapacağımız konfigürasyonlar startup-config dosyasında eklenip kaydedilirler.)

Eğer NVRAM de bir geçerli konfigürasyon bulunamazsa Router Setup modda açılış yapar. Bu router' ı adım adım konfigüre etmede bize yardımcı olur.

3.1.3.SETUP MODU:

Setup moda girdiğimiz zaman bize "Basic Management" ve "Extended Setup" olmak üzere iki opsiyon sunulur. Basic Management sadece router bağlantılarına izin verir ve yeterli konfigürasyonu yapmamız mümkündür.

```
          --- System Configuration Dialog ---
Would you like to enter the initial configuration dialog?
[yes/no]: y

At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.
```

Extended Setup bize sistemdeki bütün arayüzlerin konfigürasyonunu sorar.

```
Would you like to enter basic management setup?
[yes/no]: n
First, would you like to see the current interface summary?
[yes]:return
Any interface listed with OK? value "NO" does not have a
valid configuration
Interface     IP-Address     OK? Method Status  Protocol
FastEthernet0/0    unassigned     NO  unset  up        up
FastEthernet0/1    unassigned     NO  unset  up        up
Configuring global parameters:
  Enter host name [Router]: Todd
  The enable secret is a password used to protect access to
  privileged EXEC and configuration modes. This password,
  after entered, becomes encrypted in the configuration.
  Enter enable secret: todd
  The enable password is used when you do not specify an
  enable secret password, with some older software
  versions, and some boot images.
  Enter enable password: todd
% Please choose a password that is different from the
  enable secret
  Enter enable password: todd1
```

Dikkat edersek burada bizden iki password istendi. Bunları ileriki kısımlarda anlatacağız. Ancak şunu bilmeliyiz ki biz gerçekte her zaman Enable secret password' ü kullanacağız. Ve Enable secret ile Enable password birbirinden farklı olmalıdır. Ayrıca Setup Mode passwordu enable secret konfigürasyonunda kullanılmaz. Sonraki password ise Telnet oturumu ile Routere bağlantı için güvenlik şifresidir. Eğer buraya bir password konulmaz ise Telnet bağlantısı ile Router'a herhangi bir bağlantı yapılamaz.

```
The virtual terminal password is used to protect
access to the router over a network interface.
Enter virtual terminal password: todd
Configure SNMP Network Management? [yes]:enter or no
  Community string [public]:enter
Configure DECnet? [no]:enter
Configure AppleTalk? [no]:enter
Configure IP? [yes]:enter
  Configure IGRP routing? [yes]: n
  Configure RIP routing? [no]:enter
Configure bridging? [no]:enter
Configure IPX? [no]:enter
```

Önceki komutlar, eğer emin olmadığımız bir komut varsa bunu nasıl kullanacağımıza ve bir protokolü kolayca konfigüre etmemize yardımcı olur.
Setup mode yerine CLI modu kullanmak bize daha çok esneklik sağlar. Eğer router' da kurulu bir async modem varsa Setup moddan bunu konfigüre edebiliriz.

```
Async lines accept incoming modems calls. If you will have
users dialing in via modems, configure these lines.

  Configure Async lines? [yes]: n
```

Eğer bir router ISDN BRI arayüzü varsa ISDN switchler için konfigürasyon yapılabilir.

```
BRI interface needs isdn switch-type to be configured
Valid switch types are:
[0]  none.....Only if you don't want to configure
BRI.
[1]  basic-1tr6....1TR6 switch type for Germany
[2]  basic-5ess....AT&T 5ESS switch type for the US/Canada
[3]  basic-dms100..Northern DMS-100 switch type for US/
Canada
[4]  basic-net3....NET3 switch type for UK and Europe
[5]  basic-ni......National ISDN switch type
[6]  basic-ts013...TS013 switch type for Australia
[7]  ntt...........NTT switch type for Japan
[8]  vn3...........VN3 and VN4 switch types for France
Choose ISDN BRI Switch Type [2]: 2
```

Bu konfigürasyonlardan sonra interface konfigürasyonu için IP adreslerini ekleyebiliriz.

```
Configuring interface parameters:

Do you want to configure FastEthernet0/0  interface?
[yes]:return
  Use the 100 Base-TX (RJ-45) connector? [yes]:return
  Operate in full-duplex mode? [no]: y and return
  Configure IP on this interface? [yes]:return
    IP address for this interface: 1.1.1.1
    Subnet mask for this interface [255.0.0.0] :
255.255.0.0
    Class A network is 1.0.0.0, 16 subnet bits; mask is /
16

Do you want to configure FastEthernet0/1  interface?
[yes]:return
  Use the 100 Base-TX (RJ-45) connector? [yes]:return
  Operate in full-duplex mode? [no]:y and return
  Configure IP on this interface? [yes]:return
    IP address for this interface: 2.2.2.2
    Subnet mask for this interface [255.0.0.0] :
255.255.0.0
    Class A network is 2.0.0.0, 16 subnet bits; mask is /
16
```

Bu konfigürasyon çok basit anca yinede bir Router' ı çalıştırmak için yeterlidir.
Şu haliyle, Extended Setup yaptığımız konfigürasyonu gösterebilir.

```
The following configuration command script was created:
hostname Todd
enable secret 5 $1$B0wu$5F0m/EDdtRkQ4vy4a8qwC/
enable password todd1
line vty 0 4
password todd
snmp-server community public
!
no decnet routing
no appletalk routing
ip routing
no bridge 1
no ipx routing
!
interface FastEthernet0/0
media-type 100BaseX
full-duplex
ip address 1.1.1.1 255.255.0.0
no mop enabled
!
interface FastEthernet0/1
media-type 100BaseX
half-duplex
ip address 2.2.2.2 255.255.0.0
no mop enabled
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
end
[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.
Enter your selection [2]:0
```

Extended Setup modun ilginç bir yanı program sonlandırıldığı zaman bize seçenekler sunar. Değişiklikleri kaydetmeden CLI Moda geçme([0]), Yaptığımız değişiklikleri kaydedip Geri başa dönme([1]) yada konfigürasyonu NVRAM' e kaydedip Çıkma([2]).

3.1.4.Command-Line İnterface (CLI):

CLI bir Router' ı konfigüre etmek için çok iyi bir arayüzdür. Bize bir çok esneklik verir. CLI moda girmek için "Initial Configuration dialog" sorusuna "no" [N] demek yeterlidir. "No" dedikten sonra Router, bütün arayüzlerinin durumlarını mesajlar şeklinde gösterir.

```
Would you like to enter the initial configuration dialog?
[yes]: n
Would you like to terminate autoinstall? [yes]:return
Press RETURN to get started!
00:00:42: %LINK-3-UPDOWN: Interface Ethernet0, changed
state to up
00:00:42: %LINK-3-UPDOWN: Interface Serial0, changed state
to down
00:00:42: %LINK-3-UPDOWN: Interface Serial1, changed state
to down
00:00:42: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Ethernet0, changed state to up
00:00:42: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0, changed state to down
00:00:42: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1, changed state to down
00:01:30: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Ethernet0, changed state to down
00:01:31: %LINK-5-CHANGED: Interface Serial0, changed
state to administrativelydown
00:01:31: %LINK-5-CHANGED: Interface Ethernet0, changed
state to administratively down
00:01:31: %LINK-5-CHANGED: Interface Serial1, changed
state to administratively down
00:01:32: %IP-5-WEBINST_KILL: Terminating DNS process
00:01:38: %SYS-5-RESTART: System restarted --
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-DS-L), Version 11.3(9),
RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Tue 06-Apr-99 19:23 by dschwart
```

3.2.1.Bir Router' a Bağlanma

İnterface durum mesajları bittikten sonra "return" a basarsak "Router>" şeklinde bir prompt görürüz. Bu kullanıcı modda (User Mode) olduğumuz manasına gelir. Burada istatistikleri görebiliriz. Eğer komut satırına ***enable*** yazarsak; Router'ın privileged moduna geçeriz ve burada konfigürasyonda değişiklikler yapabiliriz.

```
Router>
Router>enable
Router#
```

"Router#" bu prompt bizim şu an privileged modda olduğumuz manasına gelir. Privileged modda konfigürasyonu görebilir ve değiştirebiliriz. User moda geri dönmek için ***disable*** komutu kullanılır.

```
Router#disable
Router>
```

Bu konumda iken ***logout , exit*** yada ***quit*** komutlarından birini yazarak Console dan çıkabiliriz.

```
Router>logout
Router con0 is now available
Press RETURN to get started.
Router>en
Router#logout
Router con0 is now available
Press RETURN to get started.
```

3.2.1.1.Router Modlarına Genel Bir Bakış

CLI' den konfigürasyonda ***config terminal*** (kısaca ***config t***) komutunu girerek router üzerinde global değişiklikler yapabiliriz. Privileged modda ***config*** yazıp ardından ***return*** dersek terminale geçebiliriz.

```
Router#config
Configuring from terminal, memory, or network
[terminal]?return
Enter configuration commands, one per line.  End with
CNTL/Z.
Router(config)#
```

Yaptığımız konfigürasyon değişikliği Dinamik RAM (DRAM) de tutulup buradan çalıştırılır. ***Config t*** komutu kullanılabilir. Yaptığımız konfigürasyonlar ise NVRAM' de saklanır. Bunları çalıştırma komutu ise ***config memory*** yada ***config mem*** dir. Yani biz yaptığımız konfigürasyonu kaydetmedikçe o bilgiler RAM' de tutulur. Eğer biz bunları NVRAM' e kaydedersek bir dahaki açılışta Router bu konfigürasyonlarla güncellenip açılır.

3.2.1.2.CLI Portları

Farklı promptları anlamak önemlidir. Konfigürasyon yaparken hangi modunda olduğumuzu bize söyler. Bu bölümde bir Cisco Router da kullanılan promptları açıklamaya çalışacağız. Router konfigürasyonunda bir komutu yazmadan önce mutlaka promptu incelememiz gerekir.

3.2.1.3.Arayüzler

Bir interface de global değişiklikler yapmak için, global modda ***interface*** komutu kullanmak gerekir.

```
Router(config)#interface ?
  Async              Async interface
  BVI                Bridge-Group Virtual Interface
  Dialer             Dialer interface
  FastEthernet       FastEthernet IEEE 802.3
  Group-Async        Async Group interface
  Lex                Lex interface
  Loopback           Loopback interface
  Multilink          Multilink-group interface
  Null               Null interface
  Port-channel       Ethernet Channel of interfaces
  Tunnel             Tunnel interface
  Virtual-Template   Virtual Template interface
  Virtual-TokenRing  Virtual TokenRing
Router(config)#interface fastethernet 0/0
Router(config-if)#
```

Dikkat edersek promptumuz ilkin "Router(config)#" şeklindeydi ***interface fast ethernet 0/0*** komutu girildikten sonra "Router(config-if)#" olarak değişti ve bu yeni prompt bize bir arayüz konfigürasyonunda olduğumuzu söylüyor.

3.2.2.Line Komutları

Usermode password' ü için ***line*** komutu kullanılır. Promptumuz Router(config-Line)# haline gelir.

```
Router#config t
Enter configuration commands, one per line.  End with
CNTL/Z.
Router(config)#line ?
  <0-70>   First Line number
  aux      Auxiliary line
  console  Primary terminal line
  tty      Terminal controller
  vty      Virtual terminal
Router(config)#line console 0
Router(config-line)#
```

3.2.3.Protokol Yönlendirme Komutları:

RIP, IGRP gibi protokollerin konfigürasyon etmek için (config-router)# promptu kullanılır.

```
Router#config t
Enter configuration commands, one per line.  End with
CNTL/Z.
Router(config)#router rip
Router(config-router)#
```

3.2.4.Düzen ve Yardım Araçları:

Yapacağımız konfigürasyonda Cisco'nun yardımından faydalanabiliriz. Bunun için "?" kullanırız. Nerede olursak olalım ? yazıp Enter a basarsak orada kullanabileceğimiz komutları yazar.

```
Router#?
Exec commands:
  access-enable    Create a temporary Access-List entry
  access-profile   Apply user-profile to interface
  access-template  Create a temporary Access-List entry
  bfe              For manual emergency modes setting
  clear            Reset functions
  clock            Manage the system clock
  configure        Enter configuration mode
  connect          Open a terminal connection
  Copy configuration or image data
  debug            Debugging functions (see also
'undebug')
  disable          Turn off privileged commands
  disconnect       Disconnect an existing network
connection
  enable           Turn on privileged commands
  erase            Erase flash or configuration memory
  exit             Exit from the EXEC
  help             Description of the interactive help
system
  lock             Lock the terminal
  login            Log in as a particular user
  logout           Exit from the EXEC
  mrinfo           Request neighbor and version information
                 from a multicast router
  --More-
```

Daha fazla bilgi için diğer sayfaya geçebilir yada buradan çıkabiliriz. Bir diğer avantajımız; Örneğin bir komutun baş harfini biliyoruz ama sonrasını bilmiyoruz. İşte bu noktada baş harfini yazıp yanına boşluk bırakmadan ? yazıp Enter a basarsak Router bize bütün olasılıkları verir. Örneğin, baş harfi C ile başlayan bir komut arıyoruz.

```
Router#c?
clear  clock  configure  connect  copy
Router#c
```

Gördüğümüz gibi router# promptunda iken C harfi ile başlayan 5 komut çıktı ve benim aradığım komutun Clock komutu olduğunu hatırladım ancak yinede nasıl kullanacağımı bilmiyorum. Aşağıdaki adımları takip edelim

```
Router#clock ?
  set  Set the time and date
Router#clock set ?
  hh:mm:ss  Current Time
Router#clock set 10:30:10 ?
  <1-31>  Day of the month
  MONTH   Month of the year
Router#clock set 10:30:10 28 ?
  MONTH  Month of the year
Router#clock set 10:30:10 28 may ?
  <1993-2035>  Year
Router#clock set 10:30:10 28 may 2000 ?
  <cr>
Router#
```

En son basamağa dikkat edersek Clock set 10:30:10 28 May 2000? Yazdık ve bunun cevabı olarak bize <CR> (Carriage Return) denildi. Bunun manası "tek seçenek" tir. Yani Clock set 10:30:10 28 May 2000 komutunun farklı seçenekleri yoktur tek seçenek kendisidir.
Eğer komutu eksik yazarsak

```
Router#clock set 10:30:10
% Incomplete command.
```

hatasını mesaj verir.
NOT: yön tuşlarından "yukarı" tuşuna basarsak bir önce yazılmış olan komut otomatikman geri yazılır.
Bir de şöyle bir hata mesajı alırsak

```
Router(config)#access-list 110 permit host 1.1.1.1
                                        ^
% Invalid input detected at '^' marker.
```

"^" işareti komut nerede yanlış kullanılmışsa o bölgeye gider.
Eğer şu mesajı alırsak

```
Router#sh te
% Ambiguous command:  "sh te"
```

Bunun manası ya komutun bütün kelimelerini girmedik yada yanlış komut girdik

KOMUT	İŞLEVİ
CTRL+A	Kursörü satır başına taşır.
CTRL+E	kursörü satır sonuna taşır.
CTRL+B	Kursörü bir harf geri hareket ettirir.
CTRL+F	kursörü bir harf ileri hareket ettirir.
ESC+F	bir harf ileri hareket
CTRL+D	tek bir karakter siler
BACKSPACE	tek bir karakter siler
CTRL+R	satırı tekrar gösterir.
CTRL+U	satırı siler
CTRL+W	kelimeyi siler
CTRL+Z	konfigürasyonu sonlandırır ve EXEC moda döner

Aşağıdaki tabloda da bir önce girmiş olduğumuz komutlarla ilgili operasyonlar gösterilmiştir.

KOMUT	İŞLEVİ
CTRL+P yada UP	Son komutu çağırır.
CTRL+N yada DOWN	Önceki irilen komutları çağırır.
Show history	En son girilen 10 komutu görüntüler
Show terminal	Terminal konfigürasyonu gösterir.

Terminal history size	En son girilen kaç komutun tutulması isteniyorsa buraya yazılır. (max 256)

örneğin aşağıda ***show history*** komutu en son girilen 10 komutu görüntülemiştir.

```
Router#sh history
  en
  sh history
  show terminal
  sh cdp neig
  sh ver
  sh flash
  sh int e0
  sh history
  sh int s0
  sh int s1
```

Aşağıda ***show terminal*** komutu kullanılmıştır. Bu bizim terminal bilgilerimizi verir

```
Router#sh terminal
Line 0, Location: "", Type: ""
[output cut]
History is enabled, history size is 10.
Full user help is disabled
Allowed transports are lat pad v120 telnet mop rlogin
nasi.  Preferred is lat.
No output characters are padded
No special data dispatching characters
Group codes:    0
```

Kaç tane komut tutmak istiyorsak bunu (10 dan daha fazla yada az) ayarlayabiliriz.

```
Router#terminal history size ?
  <0-256>  Size of history buffer
Router#terminal history size 25
```

yaptığımız değişiklikleri görelim:

```
Router#sh terminal
Line 0, Location: "", Type: ""
[output cut]
Editing is enabled.
History is enabled, history size is 25.
Full user help is disabled
Allowed transports are lat pad v120 telnet mop rlogin
nasi.  Preferred is lat.
No output characters are padded
No special data dispatching characters
Group codes:    0
```

3.3.Temel Yönlendirme Bilgisinin Gösterilmesi:

Show versiyon komutu, sistem donanımının basit konfigürasyonunu, yazılım konfigürasyonunu, konfigürasyon dosyalarının ismi ve kaynağı ve Boot imajları ile ilgili temel bilgileri verir.

```
Router#sh version
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-JS-L), Version 12.0(8),
RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Mon 29-Nov-99 14:52 by kpma
Image text-base: 0x03051C3C, data-base: 0x00001000
ROM: System Bootstrap, Version 11.0(10c), SOFTWARE
BOOTFLASH: 3000 Bootstrap Software (IGS-BOOT-R), Version
11.0(10c), RELEASE SOFTWARE (fc1)
RouterA uptime is 5 minutes
System restarted by power-on
System image file is "flash:c2500-js-l_120-8.bin"
cisco 2522 (68030) processor (revision N) with 14336K/
2048K bytes of memory.
Processor board ID 15662842, with hardware revision
00000003
Bridging software.
X.25 software. Version 3.0.0.
SuperLAT software (copyright 1990 by Meridian Technology Corp)
TN3270 Emulation software.
Basic Rate ISDN software. Version 1.1.
1 Ethernet/IEEE 802.3 interface(s)
2 Serial network interface(s)
8 Low-speed serial(sync/async) network interface(s)
1 ISDN Basic Rate interface(s)
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read ONLY)
Configuration register is 0x2102
```

Bu komut ile Router ın ne kadar zaman Router ın çalıştırıldığı DRAM 'in boyutu, çalışan 105 dosya isimleri vb. gibi bilgileri verir.

3.3.1.Password' lerin Kurulması:

Cisco Router lerin güvenliği için 5 tane Password vardır. İlk iki password; Privileged modun güvenliği için kurulan Enable password' leridir. Diğer 3 password ise Console Port, Aux port yada Telnet girişleri ile konfigürasyon yapmak istediğimiz zaman bize sorulacak olan güvenlik şifreleridir.

3.3.2.Enable Password leri:

Global konfigürasyon modunda iken Enable password leri kullanabiliriz.

```
Router(config)#enable ?
  last-resort  Define enable action if no TACACS servers
               respond
  password     Assign the privileged level password
  secret       Assign the privileged level secret
  use-tacacs   Use TACACS to check enable passwords
```

Enable Secret ile Enable password' ün her ikisine aynı şifreyi atamak istersek bir uyarı mesajı verir. Tekrar aynı şifreyi girersek bu defa kabul eder.

```
Router(config)#enable secret todd
Router(config)#enable password todd
The enable password you have chosen is the same as your
enable secret. This is not recommended.  Re-enter the
enable password.
```

Line komutu kullanılarak User-mode password' leri koyulabilir.

```
Router(config)#line ?
  <0-4>   First Line number
  aux     Auxiliary line
  console Primary terminal line
  vty     Virtual terminal
```

NOT: VTY Router da Telnet Password' üdür.

3.3.3.Auxiliary Password:

Global konfigürasyon modunda iken Aux passwordu kurmak için ***line aux?*** Komutunu kullanabiliriz. Dikkat edersek sadece bir seçenek vardır. Çünkü bu router da başka aux port yoktur.

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#line aux ?
  <0-0>  First Line number
Router(config)#line aux 0
Router(config-line)#login
Router(config-line)#password todd
```

Login komutunu hatırlamak önemlidir. Login yazıp enter a basıldıktan sonra password kurulabilir.

3.3.4.Console Password:

```
Router(config-line)#line console ?
% Unrecognized command
Router(config-line)#exit
Router(config)#line console ?
  <0-0>  First Line number
Router(config)#line console 0
Router(config-line)#login
Router(config-line)#password toddl
```

Dikkat edersek; sadece tek Console seçeneği var ve diğer önemli nokta password ü kurmadan önce ***login*** komutunu kullandık.

Diğer bazı komutlar:

```
Router(config)#line con 0
Router(config-line)#exec-timeout ?
  <0-35791>  Timeout in minutes
Router(config-line)#exec-timeout 0 ?
  <0-2147483>  Timeout in seconds
  <cr>
Router(config-line)#exec-timeout 0 0
Router(config-line)#logging synchronous
```

3.3.5.Telnet Password:

Router a Telnet girişi için User-Mode password kurmak ***line vty*** komutu ile yapılır. Router larda Cisco IOS un Enterprise sürümü yoksa VTY değeri 0 ile 4 arasındaki değerlerin haricinde olursa çalışmaz. Eğer enterprise sürümü varsa 198 değer atayabiliriz. (0-197) Bunu ezberlememize gerek yok "?" bize yardıma hazırdır.

```
Router(config-line)#line vty 0 ?
<1-197>Last Line Number
<cr>
Router(config-line)#line vty 0 197
Router(config-line)#login
Router(config-line)#password todd2
```

Eğer Telnet bağlantıları için Router a bir VTY password u kurulmamışsa bir hata mesajı alırız ve bağlantı gerçekleşmez çünkü password kurulmamıştır ve bu durumda da **no login** komutunu kullanarak Router' dan password' süz Telnet bağlantısı için izin isteyebiliriz.

```
Router(config-line)#line vty 0 197
Router(config-line)#no login
```

3.3.6.Password' ları Şifrelemek:

Sadece Secret password şifrelenebilir. Dikkat edersek aşağıda ***show running-config*** komutu girildikten sonra Enable Secret haricindeki bütün password ler gözükmektedir.

```
Router#sh run
[output cut]
!
enable secret 5 $1$rFbM$8.aXocHg6yHrM/zzeNkAT.
enable password todd1
!
[output cut]
line con 0
 password todd1
 login
line aux 0
 password todd
 login
line vty 0 4
 password todd2
 login
line vty 5 197
 password todd2
 login
!
end
Router#
```

Password' leri manual olarak şifreleme ***service password-encription*** komutu kullanılarak yapılır.

```
Router#config t
Enter configuration commands, one per line.  End with
CNTL/Z.
Router(config)#service password-encryption
Router(config)#enable password todd
Router(config)#line vty 0 197
Router(config-line)#login
Router(config-line)#password todd2
Router(config-line)#line con 0
Router(config-line)#login
Router(config-line)#password todd1
Router(config-line)#line aux 0
Router(config-line)#login
Router(config-line)#password todd
Router(config-line)#exit
Router(config)#no service password-encryption
Router(config)#^Z
```

Şimdi ***show running-config*** komutunu kullanarak Enable password ve Line password' lerinin şifrelendiğini görebiliriz.

```
Router#sh run
Building configuration...
[output cut]
!
enable secret 5 $1$rFbM$8.aXocHg6yHrM/zzeNkAT.
enable password 7 0835434A0D
!
[output cut]
!
line con 0
 password 7 111D160113
 login
line aux 0
 password 7 071B2E484A
 login
line vty 0 4
 password 7 0835434A0D
 login
line vty 5 197
 password 7 09463724B
 login
!
end
Router#
```

3.3.7.Banner:

Banner' in buradaki kelime manası "manşet"tir. Bu komut ile gerek Telnet bağlantılarında olsun gerek yöneticinin Router'a bağlantılarında görülmesini istediğimiz mesajı burada

yazabiliriz. Banner'in kullanılmasının diğer bir sebebi internetworke bağlanmış olan kullanıcılara bir güvenlik notu gösterilmiş olur.

```
Router(config)#banner ?
  LINE      c banner-text c, where 'c' is a delimiting
            character
  exec      Set EXEC process creation banner
  incoming  Set incoming terminal line banner
  login     Set login banner
  motd      Set Message of the Day banner
Router(config)#banner motd ?
  LINE  c banner-text c, where 'c' is a delimiting
character
Router(config)#banner motd #
Enter TEXT message.  End with the character '#'.
Sized to be in Acme.com network, then you must disconnect immediately.
#
Router(config)#^Z
Router#
00:25:12: %SYS-5-CONFIG_I: Configured from console by
console
Router#exit
Router con0 is now available
Press RETURN to get started.
If you are not authorized to be in Acme.com network, then
you must disconnect immediately.
Router>
```

3.4.Router Arayüzleri:

İnterface konfigürasyonu Router' ın en önemli konfigürasyonudur. İnterfaceler tanıtılmadıkça Router'lar kullanılamazlar. İnterface konfigürasyonu diğer aygıtlarla bağlantı halinde olmasını gerektirir.

```
Router(config)#int serial ?
  <0-9>  Serial interface number
Router(config)#int serial 5
Router(config-if)#
Router(config)#int ethernet ?
  <0-0>  Ethernet interface number
Router(config)#int ethernet 0
Router(config-if)#
```

Fastethernet için;

```
Router(config)#int fastethernet ?
  <0-1>  FastEthernet interface number
Router(config)#int fastethernet 0
% Incomplete command.
Router(config)#int fastethernet 0?
/
Router(config)#int fastethernet 0/?
  <0-1>  FastEthernet interface number
```

Ve bunlarla birlikte medya tipinin de belirtilmesi gerekir. Bunun için ***media-type*** komutu kullanılır.

```
Router(config)#int fa 0/0
Router(config-if)#media-type ?
  100BaseX  Use RJ45 for -TX; SC FO for -FX
  MII       Use MII connector
```

3.4.1.Bir İnterface' i Hazırlamak:

Bir İnterface ***shut down*** komutu ile kapatılır. Ve ***no shut down*** komutu ile kapattığımız interface' i açabiliriz. Eğer interface kapatılıp ***show interface*** komutu girilirse İnterface ve line protokolün kapandığını görürüz.

```
Router#sh int e0
Ethernet0 is administratively down, line protocol is down
[output cut]
```

tekrar eski haline getirmek için;

```
Router#config t
Enter configuration commands, one per line.  End with
CNTL/Z.
Router(config)#int e0
Router(config-if)#no shutdown
Router(config-if)#^Z
00:57:08: %LINK-3-UPDOWN: Interface Ethernet0, changed
state to up
00:57:09: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Ethernet0, changed state to up
Router#sh int e0
Ethernet0 is up, line protocol is up
```

3.4.2.Bir IP Adresinin Bir İnterface Üzerinde Konfigürasyonu:

Bir interface' in IP konfigürasyonunu yapmak için interface konfigürasyon modunda iken ***ip address*** komutu kullanılır.

```
Router(config)#int e0
Router(config-if)#ip address 172.16.10.2 255.255.255.0
Router(config-if)#no shut
```

no shut down komutunu yazmayı unutmayalım.

Eğer interface' e ikinci bir subnet adresi eklemek istersek o zaman ***secondary*** komutunu kullanırız. Bu komut ip adresi yazıldıktan sonra enter' a basılmadan önce yazılır.

```
Router(config-if)#ip address 172.16.20.2 255.255.255.0
secondary
Router(config-if)#^Z
```

yaptığımız konfigürasyonu görelim.

```
Router#sh run
Building configuration...
Current configuration:
[output cut]
!
interface Ethernet0
 ip address 172.16.20.2 255.255.255.0 secondary
 ip address 172.16.10.2 255.255.255.0
!
```

3.4.3.Seri İnterface Komutları:

İnterface, aygıtın CSU/DSU tipini belirlemeli ve hat için clock oluşturmalı. Cisco Router'lar DTE aygıtlarıdır. Bunun manası clock işareti DCE interface sağlar. Ve bu DCE interface ***clock rate*** komutu ile konfigüre edilebilir.

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int s0
Router(config-if)#clock rate ?
        Speed (bits per second)
  1200
  2400
  4800
  9600
  19200
  38400
  56000
  64000
  72000
  125000
  148000
  250000
  500000
  800000
  1000000
  1300000
  2000000
  4000000
  <300-4000000>    Choose clockrate from list above
Router(config-if)#clock rate 64000
%Error: This command applies only to DCE interfaces
Router(config-if)#int s1
Router(config-if)#clock rate 64000
```

Daha sonraki komutumuz ise ***bandwidth*** komutudur. Bütün Cisco Router'lar seri bağlantılarda T1 standardında yada 1544 Mbps hızında çalışırlar. Bir seri linkin bant genişliğini, uzak bağlantılarda IGRP, EIGRP ve OSPF gibi yönlendirici protokoller, en iyi seçimi yaparak kullanılır. Eğer RIP yönlendirme kullanıyorsak, RIP seri linkin bant genişliği ile alakalı değildir.

```
Router(config-if)#bandwidth ?
  <1-10000000>  Bandwidth in kilobits
Router(config-if)#bandwidth 64
```

3.4.4.Hostname:

Hostname yerel bir tanımlamadır.

```
Router#config t
Enter configuration commands, one per line.  End with
CNTL/Z.
Router(config)#hostname todd
todd(config)#hostname Atlanta
Atlanta(config)#
```

3.4.5.Tanımlamalar:

Buda switchlerde olduğu gibi, kolaylık olması açısından her bir interface' i bir isimle nitelendirmek için kullanılır.

```
Atlanta(config)#int e0
Atlanta(config-if)#description Sales Lan
Atlanta(config-if)#int s0
Atlanta(config-if)#desc Wan to Miami circuit:6fdda4321
```

tanımlamaları ***show running config*** yada ***show interface*** komutları ile görebiliriz.

```
Atlanta#sh run
[cut]
interface Ethernet0
 description Sales Lan
 ip address 172.16.10.30 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 description Wan to Miami circuit:6fdda4321
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
Atlanta#sh int e0
Ethernet0 is up, line protocol is up
  Hardware is Lance, address is 0010.7be8.25db (bia
0010.7be8.25db)
  Description: Sales Lan
  [cut]
Atlanta#sh int s0
Serial0 is up, line protocol is up
  Hardware is HD64570
  Description: Wan to Miami circuit:6fdda4321
[cut]
Atlanta#
```

3.5.Konfigürasyonların Kaydedilmesi Ve Görüntülenmesi:

Yaptığımız konfigürasyonlar DRAM' de saklı tutulan **running-config** dosyasıdır. Bu dosyayı NVRAM' e **startup-config** adı altında kaydetmemiz gerekir. DRAM' den NVRAM'e kaydetmek için ***copy running-config startup config*** yada kısaca ***copy run start*** demek yeterlidir.

```
Router#copy run start
Destination filename [startup-config]?return
Warning: Attempting to overwrite an NVRAM configuration
previously written by a different version of the system
image.
Overwrite the previous NVRAM configuration?[confirm]return
Building configuration...
```

Bu mesaj; daha önce kaydedilmiş bir startup-config dosyası vardır. Bunun üzerine kayıt yapılsın mı diye bize soruyor.
Geçerli konfigürasyonu görmek için;

```
Router#sh run
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
ip subnet-zero
frame-relay switching
!
[cut]
```

NVRAM' de kayıtlı startup-config dosyasını görmek için;

```
Router#sh start
Using 4850 out of 32762 bytes
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
!
ip subnet-zero
frame-relay switching
!
[cut]
```

kaydetmiş olduğumuz konfigürasyonu silmek istersek;

```
Router#erase startup-config
Erasing the nvram filesystem will remove all files!
Continue? [confirm]
[OK]
Erase of nvram: complete
Router#sh start
%% Non-volatile configuration memory is not present
Router#
```

show interface komutunun en önemli özelliği Hat ve Data Link protokollerinin durumunu da gösterir.

```
RouterA#sh int e0
Ethernet0 is up, line protocol is up
```

1. parametre fiziksel katmanı simgeliyor. “ is up” ise taşıyıcı algılandığı zaman yazılır. 2. parametre ise Data Link katmanını simgeliyor. “is up” bağlantının kurulup kurulmaması durumlarında yazılır.
Show controllers komutu fiziksel interface’ ler hakkında bilgi görüntüler.

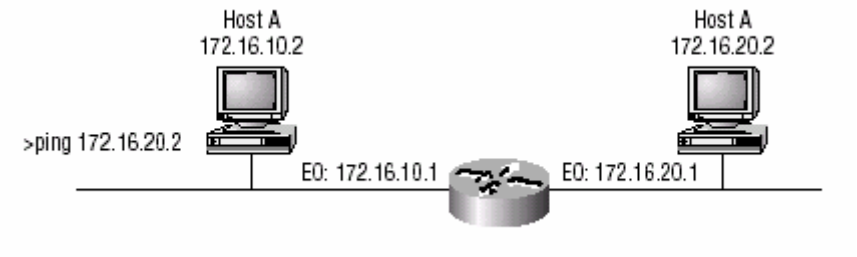
```
Router#sh controllers s 0
HD unit 0, idb = 0x1229E4, driver structure at 0x127E70
buffer size 1524  HD unit 0, V.35 DTE cable
cpb = 0xE2, eda = 0x4140, cda = 0x4000

Router#sh controllers s 1
HD unit 1, idb = 0x12C174, driver structure at 0x131600
buffer size 1524  HD unit 1, V.35 DCE cable
cpb = 0xE3, eda = 0x2940, cda = 0x2800
```

IP YÖNLENDİRME

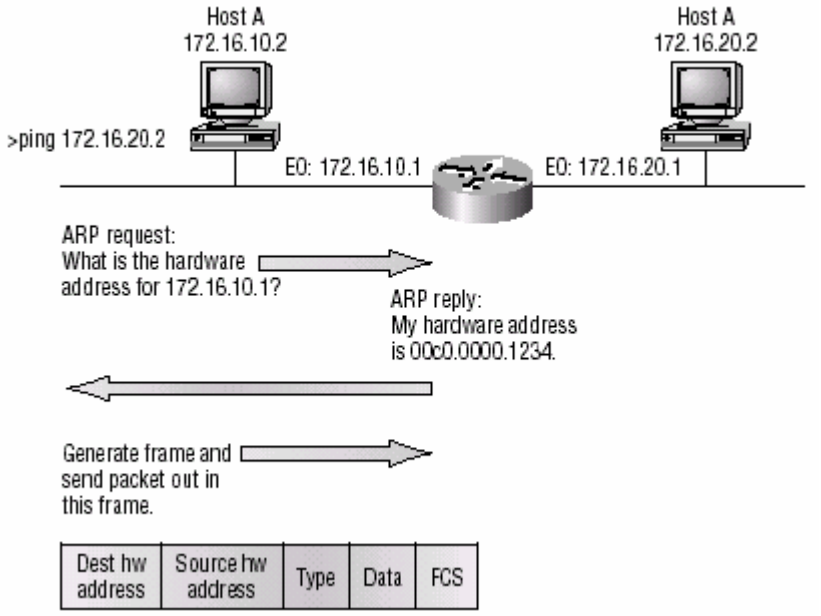
GİRİŞ:

Bu bölümü anlatmadan önce aşağıdaki şekle bakalım. Host A, Farklı networkteki Host B ile bağlantı kurmak istediği zaman adım adım yapılan işlemleri inceleyelim



Bu örnekte Host A, Host B nin IP adresine ping atıyor.

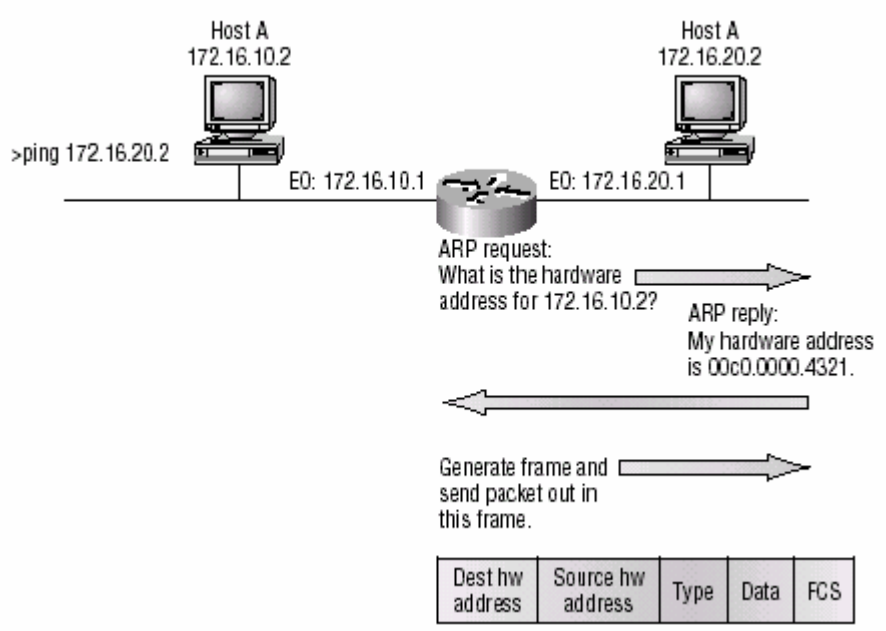
1. Komut satırına ping 172.16.20.2 komutu girildi. Bu durumda Host A, IP ve ICMP network katmanı protokollerini kullanarak bir paket üretir.
2. IP ve ARP protokolleri birlikte çalışıp, bu paketin nereye gideceğine karar vermek için ilkin Host A nın IP adresine ve Subnet Maskesine bakarlar. Sonra hedef IP adresinin Lokal adres içinde olmadığını ve uzaktaki bir kullanıcı olduğunu belirler. Host A bu paketin doğru yere ulaşması için Router'a gönderir.
3. Host A nın Router'a bir paket göndermek için Host A nın Lokal networkteki Router'ın İnterface' ine yazılı, donanım adresini bilmesi gerekir. Hatırlarsak; Network katmanı, paket ve hedef donanım adresini bir alt katman olan data link katmanında çerçevelenmesi ve lokal Hosta iletilmesi için gönderiliyordu. Donanım adresi alınır ve kullanıcı ARP protokolünü çağırır.
4. Eğer IP adres bulunamıyor ve ARP buna karar veremiyorsa Host A nın ARP protokolü , IP adresi 172.16.10.1 olan bir arama yayını yapar. Ve Host B buna cevap yollar.
5. Router lokal networke, Ethernet arayüzü bağlantısının donanım adresini verir. İşte bu noktada lokal networkten Router'a paket akışı için uygun bağlantı kurulmuş olur. Network katmanı, paketleri ICMP protokolü ile yapılandırarak Host A nın paketleri göndermek istediği donanım adresi ile birlikte, kaynak ve hedef IP leri de içeren paketi bir alt katmana (Data Link) iletir.
6. Data link katmanı açılan paketi lokal networkte iletilirken ihtiyacı olan kontrol bilgilerini ekleyerek bir Frame oluşturur. Şekilde Data Link katmanında oluşturulmuş ve onun dışarıya gönderilirken aldığı biçimi görüyoruz



görüldüğü gibi bütün bilgilerin iletiminde şekildeki Routere ihtiyaç vardır. Kaynak ve hedef donanım adresleri, kaynak ve hedef IP adresleri, sonlandırma biti, Data ve FCS bitlerinin hepsi bir frame içinde mevcuttur.

7. Host A nın Data Link katmanı Frame'i fiziksel katmana 0 ve 1 lerden oluşan dijital sinyale çevrilip gönderilmesi için verir.
8. Router, Ethernet 0 arayüzünde toplanmış olan senkronize dijital sinyalleri alıp tekrar Frame i elde eder. Ve CRC yi çalıştırır. Eksik bölüm yada Çakışma (Collision) olup olmadığı FCS kontrol edilerek öğrenilir.

9. Hedef donanım adresi sınamıştır. Bundan sonra Router data paketi ile birlikte Frame i incelemeli ve ne yapması gerektiğini anlaması gerekir. IP bu Frame in içindedir. Ve router de çalışan IP protokolü, paketi alır.
10. IP protokolü, paketin hedef IP adresine bakar. Hedef IP adresi 172.16.20.2, ve Router Ethernet 1 arayüzüne 172.16.20.0 networkünün bağlı olduğuna yönlendirme tablosuna bakarak karar verir.
11. Router paketi ethernet 1 arayüzüne yollar. Router ın bu paketi hedefe gönderebilmesi için bunun tekrar Frame e dönüştürmesi gerekir. İlk olarak Router ARP yi çağırarak Donanım adresinin içerip içermediğini sorar. Eğer içermiyorsa ARP Ethernet 1 çıkışından 172.16.20.2 donanım adresini bulmak için bir yayın yapar. (bu örnekte biliniyor)
12. Host B ise bir ARP tekrarı ile kendi network arayüzünün donanım adresini cevap olarak verir. Routerin Ethernet 1 arayüzü şimdi paket transferi için elverişli hale gelmiştir.



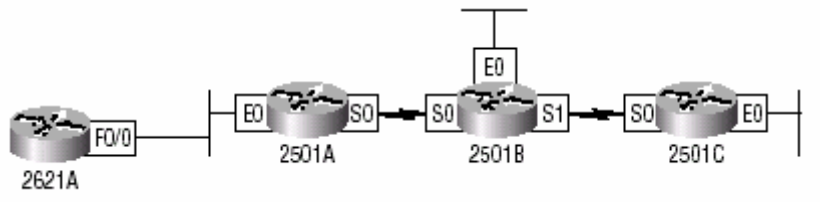
Router in Ethernet 1 arayüzünde oluşturulan Frame, Host B nin network arayüz kartının donanım (hedef) adresi ve Ethernet 1 arayüzünün donanım (kaynak) adresine sahiptir. Bu nokta önemlidir. Dikkat edersek hedef ve kaynak donanım adresleri değişmesine rağmen hedef ve kaynak IP adresi hiç değişmemiştir.

13. Host B Frame alır ve CRC yi çalıştırır. Sınamayı bitirince IP adresini verir. Ve IP adresini kontrol eder. Eğer IP adresi kendisinin IP adresi ise paketin ne amaçlı olduğuna karar vermek için Protokol kümesini çalıştırır.
14. Eğer Host B bir cevap verecekse ICMP paketi oluştururken Host B nin kaynak adresini Host A nın hedef adresini ekler. Aynı İşlemler tekrar cereyan eder. Ancak bu sefer yol boyunca bütün donanım adresleri biliniyor sadece ARP de her arayüzün donanım adresine bakar.

4.1.Daha Büyük Bir Networkte IP Yönlendirme İşlemi:

Bir önceki örnekte; Router direkt networke bağlı idi, aşağıdaki örnek daha fazla bir Router grubundan oluşuyor.

Aşağıdaki şekilde 2500A, 2500B, 2500C ve 2621A Router' ları gösterilmiştir.



Şekilde WAN ile bağlanmış üç adet 2500 serili Router ve 2500A Router' ının çıkışına Ethernet networkü ile bağlı bir 2621 serili Router gözükmektedir. Ayrıca her Router bir ethernet bağlantısına sahiptir.
Her router'ın konfigürasyonun ilk adımı hatasız düzenlenmesidir. Aşağıdaki tabloda her arayüz ve adresler doğru şekilde düzenlenmiştir. Buradaki her networkün IP si C sınıfıdır.

Router	Network Adres	İnterface	Adres
2621A	172.16.10.0	f0/0	172.16.10.1
2501A	172.16.10.0	e0	172.16.10.2
2501A	172.16.20.0	s0	172.16.20.1
2501B	172.16.20.0	s0	172.16.20.2
2501B	172.16.40.0	s1	172.16.40.1
2501B	172.16.30.0	e0	172.16.30.1
2501C	172.16.40.0	s0	172.16.40.2
2501C	172.16.50.0	e0	172.16.50.1

Router da IP konfigürasyonu tamamen basit bir işlemdir. Sadece yapmamız gereken arayüzlere IP adreslerini yazmak ve bu interface leri etkin kılmaktır. (***no shut down)***

İlk olarak 2621A nın IP konfigürasyonunu yapalım

4.1.1. 2621A Routerin Konfigürasyonu:

Burada yapmamız gereken sadece Fastethernet 0/0 arayüzüne IP adresini eklemektir.

```
Router> en
Router#config t
Router (config)#hostname 2621A
2621A(Config)#interface fa0/0
2621A(Config-if)#ip address 172.16.10.1 255.255.255.0
2621A(Config-if)#no shut
```

Oluşturduğumuz IP yönlendirme tablosunu görmek için Privileged modda ***show ip route*** komutunu girmek yeterlidir. Dikkat edersek tek bir adres tanımlıdır. Bunun manası bu router sadece 172.16.10.0 networkünü biliyor.

```
2621A#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, * - candidate default U - per-user
static route, o - ODR, P - periodic downloaded static
route T - traffic engineered route
Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.10.0 is directly connected, FastEthernet0/0
2621A#
```

Yukarda “C 172.16.10.0 directly connected .....” yazılı en alttaki satırda “C” karakterinin manası “bu network e doğrudan bağlantılıdır” demektir.

4.1.2. 2501A Router’ının konfigürasyonu:

Bu Router in konfigürasyonu ethernet 0 ve Serial 0 arayüzlerinin düzenlenmesi ile yapılır

```
Router>en
Router#config t
Router(config)#hostname 2501A
2501A(config)#int e0
2501A(config-if)#ip address 172.16.10.2 255.255.255.0
2501A(config-if)#no shut
2501A(config-if)#int s0
2501A(config-if)#ip address 172.16.20.1 255.255.255.0
2501A(config-if)#no shut
```

Tablomuzda Serial 0 arayüzü 172.16.20.0 ve Ethernet 0 arayüzü de 172.16.10.0 networkü içindedir.

```
2501A#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP,
M - [output cut]
Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.20.0 is directly connected, Serial0
C       172.16.10.0 is directly connected, Ethernet0
2501A#
```

Yukarıda “C” karakteri ile belirtilen iki tane doğrudan bağlantı vardır.

4.1.3. 2501B Routerin konfigürasyonu:

2501B nin konfigürasyonu da benzer şekildedir. Farklı olarak DCE arayüzlü bağlanmış her iki Serial arayüze ***clock rate*** komutu ile çıkış hızını ayarlıyoruz.

```
Router>en
Router#config t
Router(config)#hostname 2501B
2501B(config)#int e0
2501B(config-if)#ip address 172.16.30.1 255.255.255.0
2501B(config-if)#no shut
2501B(config-if)#int s0
2501B(config-if)#ip address 172.16.20.2 255.255.255.0
2501B(config-if)#clock rate 64000
2501B(config-if)#no shut
2501B(config-if)#int s1
2501B(config-if)#ip address 172.16.40.1 255.255.255.0
2501B(config-if)#clock rate 64000
2501B(config-if)#no shut
2501B#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP,
M - [output cut]
Gateway of last resort is not set
     172.16.0.0/24 is subnetted, 3 subnets
C       172.16.40.0 is directly connected, Serial1
C       172.16.30.0 is directly connected, Ethernet0
C       172.16.20.0 is directly connected, Serial0
2501B#
```

Router A ile Router B doğrudan bağlantı yapabilirler. Çünkü aynı WAN networkündedirler. Ancak Router B, 2621A Router i ile doğrudan bağlantı yapamaz. Çünkü 172.16.10.0 network ü hakkında bir bilgi sahibi değildir.

4.1.4. 2501C Router'ının Konfigürasyonu:

2501C nin konfigürasyonu Network IP hariç aynen 2501A nın konfigürasyonu şeklindedir.

```
Router>en
Router#config t
Router(config)#hostname 2501C
2501C(config)#int e0
2501C(config-if)#ip address 172.16.50.1 255.255.255.0
2501C(config-if)#no shut
2501C(config-if)#int s0
2501C(config-if)#ip address 172.16.40.2 255.255.255.0
2501C(config-if)#no shut
```

NOT: 2501C nin Ethernet 0 Arayüzü 172.16.50.0 networküne bağlıdır ve serial 0 arayüzüde 172.16.40.0 WAN network ü içindedir.

```
2501C#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP,
M - [output cut]
Gateway of last resort is not set
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.50.0 is directly connected, Ethernet0
C       172.16.40.0 is directly connected, Serial0
2501C#
```

2501C ile 2501B birbirleri ile doğrudan iletişim yapabilirler çünkü aynı WAN networkü içindedirler.

4.2.Network İçinde IP Yönlendirilmesi:

Bir önceki kısımda Network IP adresleme ile doğrudan konfigüre edilebiliyordu. Peki bir Router uzak bir networke bir paketi nasıl Gönderir? Router sadece Routing Tablosuna bakar ve uzak networklere nasıl bağlanacağını araştırıp paketleri gönderir. Ancak bizim Konfigürasyon yaptığımız Router'lar sadece direkt bağlı oldukları networklerin IP adreslerini Routing tablolarının her birinin içine atarlar. Router bir paket alıp bunun network adresi Routing tablosunda bulamadığı zaman ne olur?
Router hemen broadcast adresini kullanarak uzak networkler için bir arama yayını yapmaz. Paketlerin küçük networkümüzde ilerleyebilmesi için network adreslerini Routing tablosuna eklemek için yapacağımız birkaç farklı konfigürasyon yolu vardır. Ve eğer farklı yönlendirme tiplerini anlarsak bizim yapmış olduğumuz çalışmalarımız en iyi performanslı bir yönlendirme yapabilir.
Üç farklı yönlendirme Tipi vardır.

- Statik yönlendirme
- Default yönlendirme
- Dinamik yönlendirme

4.3. Statik Yönlendirme:

Statik yönlendirme Network yönlendiricisinin manual olarak her bir aygıtı Routing tablosuna yazma işlemidir. Bu yöntemde kullanılan komutlar ve işlevleri aşağıda belirtilmiştir.

ip route :statik yönlendirmenin yapılması için kullanılan komuttur.
destination network :Routing tablosuna eklemek istediğimiz network adresi
mask :network te kullanılacak olan subnet mask
next hop address :sonraki sekme adresi
exit interface :interface den çıkış LAN da çalışmaz.
administrative distance : ne kadar mesafede olduğu
permanent : eğer arayüz kapanırsa Router diğer sekme ile bağlantı kuramaz. Router bunu otomatik olarak Routing tablosundan çıkarır. Permanent kapatılarak Routing tablosuna bir şey yazılaması engellenir.

4.3.1. 2621A nın Konfigürasyonu:

Her bir bağlantı Routing tabloya otomatikman eklenir. 2621A sadece 172.16.10.0 networküne bağlıdır. 2621A için bütün networklere yönlendirmeler için aşağıdaki adreslerin her birinin tanıtılması gerekmektedir.

172.16.20.0 172.16.30.0 172.16.40.0 172.16.50.0

Aşağıda 2621A için Statik yönlendirme kullanılarak bir konfigürasyon yapılmıştır. Böylece 2621A uzak networkleri bulabilir.

```
2621A(Config)#ip route 172.16.20.0 255.255.255.0
172.16.10.2
2621A(Config)#ip route 172.16.30.0 255.255.255.0
172.16.10.2
2621A(Config)#ip route 172.16.40.0 255.255.255.0
172.16.10.2
2621A(Config)#ip route 172.16.50.0 255.255.255.0
172.16.10.2
```

Bu işlemlerden sonra show running-config ve show ip route komutları kullanılarak statik yönlendirme görülebilir. Eğer bağlantı yapılamıyorsa "permanent" parametresi girilmemiş olabilir.

```
2621A#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP,
M - [output cut]
Gateway of last resort is not set
     172.16.0.0/24 is subnetted, 5 subnets
S       172.16.50.0 [1/0] via 172.16.10.2
S       172.16.40.0 [1/0] via 172.16.10.2
S       172.16.30.0 [1/0] via 172.16.10.2
S       172.16.20.0 [1/0] via 172.16.10.2
C       172.16.10.0 is directly connected, FastEthernet0/0
2621A#
```

4.3.2. 2501A nın Konfigürasyonu:

2501A Router'ı 172.16.10.0 ve 172.16.20.0 networklerine direkt bağlıdır.
172.16.30.0 , 172.16.40.0 , 172.16.50.0 networklerine statik yönlendirme yapılması gerekir.

```
2501A(Config)#ip route 172.16.30.0 255.255.255.0
172.16.20.2
2501A(Config)#ip route 172.16.40.0 255.255.255.0
172.16.20.2
2501A(Config)#ip route 172.16.50.0 255.255.255.0
172.16.20.2
```

```
2501A#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP,
M - [output cut]
Gateway of last resort is not set
     172.16.0.0/24 is subnetted, 5 subnets
S       172.16.50.0 [1/0] via 172.16.20.2
S       172.16.40.0 [1/0] via 172.16.20.2
S       172.16.30.0 [1/0] via 172.16.20.2
C       172.16.20.0 is directly connected, Serial0
C       172.16.10.0 is directly connected, Ethernet0
2501A#
```

S ⟶ Static entry C ⟶ Direkt

[1/0] Administrative Distance ve network sekmesini gösterir.
Böylece 2501A nın Routing tablosu tamamlanmış oldu.

4.3.3. 2501B nın Konfigürasyonu:

Bu Router 172.16.20.0 , 172.16.30.0 ve 172.16.40.0 networklerine direkt bağlıdır.
172.16.10.0 ve 172.16.50.0 networkleri Statik Routing tablosuna eklenmelidir.

```
2501B(Config)#ip route 172.16.10.0 255.255.255.0
172.16.20.1
2501B(Config)#ip route 172.16.50.0 255.255.255.0
172.16.40.2
```

```
2501B#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP,
M - [output cut]
Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 5 subnets
S       172.16.50.0 [1/0] via 172.16.40.2
C       172.16.40.0 is directly connected, Serial1
C       172.16.30.0 is directly connected, Ethernet0
C       172.16.20.0 is directly connected, Serial0
S       172.16.10.0 [1/0] via 172.16.20.1
2501B#
```

4.3.4. 2501C nın Konfigürasyonu:

Bu Router 172.16.40.0 ve 172.16.50.0 networklerine doğrudan bağlıdır.
172.16.10 , 172.16.20.0 ve 172.16.30.0 networkleri için Statik yönlendirme yapılması gerekiyor.

```
2501C(Config)#ip route 172.16.10.0 255.255.255.0
172.16.40.1
2501C(Config)#ip route 172.16.20.0 255.255.255.0
172.16.40.1
2501C(Config)#ip route 172.16.30.0 255.255.255.0
172.16.40.1
```

Arayüz konfigürasyonuna bakalım:

```
2501C#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP,
M - [output cut]
Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 5 subnets
C       172.16.50.0 is directly connected, Ethernet0
C       172.16.40.0 is directly connected, Serial0
S       172.16.30.0 [1/0] via 172.16.40.1
S       172.16.20.0 [1/0] via 172.16.40.1
S       172.16.10.0 [1/0] via 172.16.40.1
2501C#
```

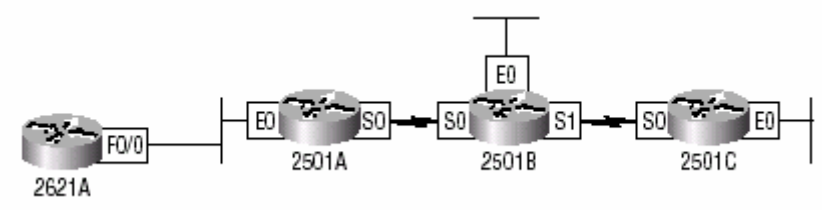
Şu anda bütün Router'lar doğru yönlendirme tablolarına sahiptirler ve kullanıcılar problemsiz bir şekilde iletişim yapabilirler. Ancak bir ekleme yaparsak bunu bütün router' lara teker teker eklememiz gerekir. Bu küçük networklerde kolaydır. Ancak büyük networklerde çok zordur.

4.3.5. Yaptığımız Statik Routing Konfigürasyonunu Sınayalım:
Bu networkteki bütün Router ların konfigürasyonlarını yaptık. Şimdi yaptığımız konfigürasyonu kontrol edelim. Bunun için en iyi yol "ping" programdır.

```
2621A#ping 172.16.50.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.50.1, timeout is
2 seconds:
.!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max
= 64/66/68 ms
2621A#
2501C#ping 172.16.10.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.1, timeout is
2 seconds:
!!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max =
64/67/72 ms
```

4.4. Default Routing:

Default Routing paket göndermede kullanılır. Routing tablosunda bulunmayan networklere paket göndermede kullanılır. Uç noktalarda sadece Default Routing kullanılır.



Yukarıdaki şekilde 2621A ve 2501C uç networklerdir. Örneğin bunu 2501A yada 2501B için yaparsak paketler doğru yönde ilerleyemeyebilir.
2501C Router'ı 2 bağlantıya sahip olup 172.16.50.0 networkünde bir Router yoktur. Bu sadece 172.16.40.1(2501B nin arayüzüne) networküne paket gönderebilir.
2621A sadece 2501A nın 172.16.10.1 arayüzüne paket gönderebilir.

```
2501C(Config)#no ip route 172.16.10.0 255.255.255.0
172.16.40.1
2501C(Config)#no ip route 172.16.20.0 255.255.255.0
172.16.40.1
2501C(Config)#no ip route 172.16.30.0 255.255.255.0
172.16.40.1
2501C(Config)#ip route 0.0.0.0 0.0.0.0 172.16.40.1
```

```
2501C#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M -
[output cut]
 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default U - per-user static route, o - ODR

Gateway of last resort is 172.16.40.1 to network 0.0.0.0
172.16.0.0/24 is subnetted, 5 subnets
C       172.16.50.0 is directly connected, Ethernet0
C       172.16.40.0 is directly connected, Serial0
S*     0.0.0.0/0 [1/0] via 172.16.40.1
2501C#
```

4.5. Dinamik Yönlendirme:

Dinamik yönlendirme Router'larda Routing tablolarını güncelleme ve bunlardan adres öğrenmek için kullanılan protokol işlemleridir. Bu Statik ve Default yönlendirmelerden kolaydır. Yönlendirme protokolleri komşu Router' larla iletişimde kullanılır.

4.5.1.Administrative Distance:

Yönlendirme protokollerinin konfigürasyonu esnasında Administrative Distances (ADs)'i bilmemiz gerekir. ADs 0-255 değerlerinde olabilir. 0 en güvenilir tercihtir. 255'in manası; bu Router üzerinden trafik olmayacaktır.

Aşağıdaki tabloyu inceleyelim

Görevi	Distance değeri
interface'e bağlanma	0
Statik Yönlendirme	1
EIGRP	90

IGRP	100
OSPF	110
RIP	120
Harici EIGRP	170
Bilinmiyor	255

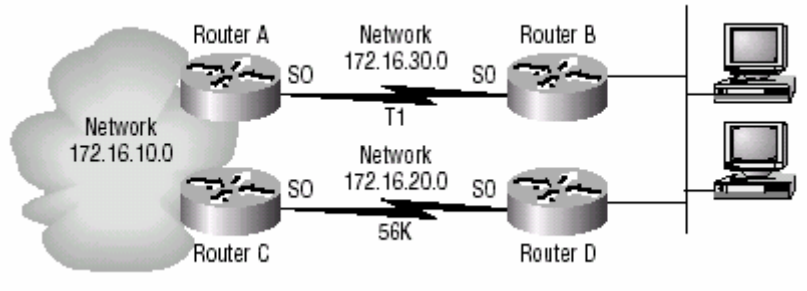
Eğer bir network bağlanacaksa; her zaman için networke interface bağlantısı vardır. Eğer yönetici bir Statik yönlendirme konfigürasyonu yaparsa Router bunu "öğrenilmiş bir yönlendirme" olarak algılar.

4.5.2.Yönlendirme Protokolleri:

Distance-Vektor(uzaklık vektörü), RIP ve IGRP olmak üzere üç farklı yönlendirme protokolü vardır.

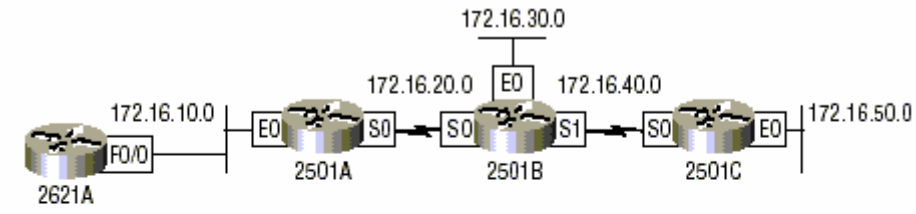
4.5.2.1.Uzaklık Vektörü:

Komşu Router'lar arasındaki yönlendirme tablolarını tamamlamak için kullanılan bir algoritmadır. Router'lar kombineli bir şekilde Routing tablolarını tamamlayıp internetworking haritasını çizerler. Böylece uzak bir networke bağlantı istendiği zaman bu bağlantının hangi düğümler üzerinden yapılacağı biliniyor olur.
Aşağıda farklı bant genişliğinde ve eşit sekme değerinde bir network örneği vardır.



172.16.30.0 network'ü T1 linki 1544 Mbps bant genişliğinde ve 172.16.20.0 network'ü 56K lık bir linke sahiptir. Bu örnek bir router açıldığında, Distance-Vektor yönlendirme protokolünü çalıştırdığı zaman neler olduğunu anlamamıza yardımcı olur.

Aşağıdaki şekilde de görüldüğü gibi 4 router açıldığı zaman Routing tablolarında direkt bağlantılar belirlenir. Daha sonra her bir Router'daki Distance-vektor protokolü çalıştırılır. Şimdi bu protokolü çalıştırmadan önceki Routing tablolarını inceleyelim.



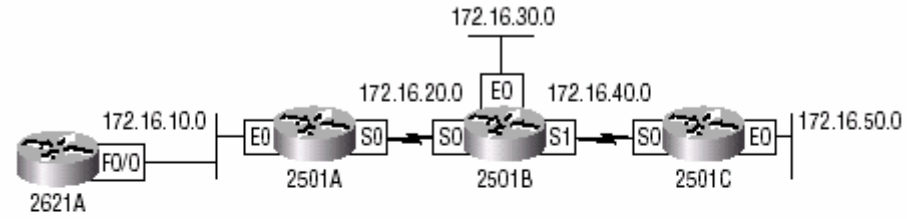
Routing Table		
172.16.10.0	FO/O	0

Routing Table		
172.16.10.0	E0	0
172.16.20.0	S0	0

Routing Table		
172.16.20.0	S0	0
172.16.30.0	E0	0
172.16.40.0	S1	0

Routing Table		
172.16.40.0	S0	0
172.16.50.0	E0	0

Şimdide Distance-vektor protokolü çalıştırıldıktan sonraki Routing tablolarına bakalım:



Routing Table		
172.16.10.0	FO/O	0
172.16.20.0	FO/O	1
172.16.30.0	FO/O	2
172.16.40.0	FO/O	2
172.16.50.0	FO/O	3

Routing Table		
172.16.10.0	E0	0
172.16.20.0	S0	0
172.16.30.0	S0	1
172.16.40.0	S0	1
172.16.50.0	S0	2

Routing Table		
172.16.20.0	S0	0
172.16.30.0	E0	0
172.16.40.0	S1	0
172.16.10.0	S0	1
172.16.50.0	S1	1

Routing Table		
172.16.40.0	S0	0
172.16.50.0	E0	0
172.16.10.0	S0	2
172.16.20.0	S0	1
172.16.30.0	S0	1

4.5.3. RIP (Routing Information Protocol):

RIP gerçek bir Uzaklık vektörü yönlendirme protokolüdür. RIP ile ağdaki değişimin bilgisinin ağ üzerinde yayını önemli ölçüde zaman alan tablo güncellemeleri yapılır. En yakın komşu Router'da tablo değişimleri islenirken hatalı tablolar, daha aşağıdaki Router'larda kalmaya devam eder. Bir RIP güncellemesi sırasında, bir yol tıkanıklığı veya veri kaybı ihtimali artar. RIP' in tablo güncelleme metodu, ağ değişimlerine verilen cevabin gecikmesine sebep olur. Bir RIP yönlendiricisi tüm routing tablosunu kendi komşu Router'lara iletir. Alan Router bu tabloyu, kendinde olan her değerle karsılaştırmak zorundadır. Geniş ağlarda bu işlem CPU gücü ve hafizaya bağlıdır. Bu hesaplama, tablolar değişse de değişmese de her 30 saniyede bir yapılan tablo değişim işlemi sırasında yinelenir. Alan Router karsılaştırma işlemi bitmeden, ağda bir değişim olup olmadığını anlayamaz. RIP Hop Count değerine bakıp en iyi ve en kısa yolun hangisi olduğuna karar verir.

4.5.3.1.RIP Yönlendirme Konfigürasyonu:

Statik Routing bölümü anlatılırken yaptığımız örnek çalışmayı ele alalım. Önce orada yaptığımız konfigürasyonu silelim.bunun için ***no ip route*** komutu kullanılır.

```
2621A#config t
Enter configuration commands, one per line.  End with
CNTL/Z.
2621A(config)#no ip route 172.16.20.0 255.255.255.0
172.16.10.2
2621A(config)#no ip route 172.16.30.0 255.255.255.0
172.16.10.2
2621A(config)#no ip route 172.16.40.0 255.255.255.0
172.16.10.2
2621A(config)#no ip route 172.16.50.0 255.255.255.0
172.16.10.2
```

RIP konfigürasyonu için ***route rip*** komutu kullanılır.

4.5.3.2. 2621A nın Konfigürasyonu:

RIP Protokolünün Default Distance değeri 120 idi.

```
2621A(config)#router rip
2621A(config-router)#network 172.16.0.0
2621A(config-router)#^Z
2621A#
```

4.5.3.3. 2501A nın Konfigürasyonu:

RIP konfigürasyonu yapmak için bunda da statik konfigürasyonu silmemiz gerekir.

```
2501A#config t
Enter configuration commands, one per line.  End with
CNTL/Z.
2501A(config)#no ip route 172.16.30.0 255.255.255.0
172.16.20.2
2501A(config)#no ip route 172.16.40.0 255.255.255.0
172.16.20.2
2501A(config)#no ip route 172.16.50.0 255.255.255.0
172.16.20.2
2501A(config)#router rip
2501A(config-router)#network 172.16.0.0
2501A(config-router)#^Z
2501A#
```

4.5.3.4. 2501B nın Konfigürasyonu:

```
2501B#config t
Enter configuration commands, one per line.  End with
CNTL/Z.
2501B(config)#no ip route 172.16.10.0 255.255.255.0
172.16.20.1
2501B(config)#no ip route 172.16.50.0 255.255.255.0
172.16.40.2
2501B(config)#router rip
2501B(config-router)#network 172.16.0.0
2501B(config-router)#^Z
2501B#
```

4.5.3.5. 2501C nın Konfigürasyonu:

```
RouterC#config t
Enter configuration commands, one per line.  End with
CNTL/Z.
RouterC(config)#no ip route 0.0.0.0 0.0.0.0 172.16.40.1
RouterC(config)#router rip
RouterC(config-router)#network 172.16.0.0
RouterC(config-router)#^Z
RouterC#
```

Şimdide RIP yönlendirme tablolarını inceleyelim.

4.5.3.6. 2621A:

```
2621A#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - [output cut]
Gateway of last resort is not set
     172.16.0.0/24 is subnetted, 5 subnets
R       172.16.50.0 [120/3] via 172.16.10.2, FastEthernet0/0
R       172.16.40.0 [120/2] via 172.16.10.2, FastEthernet0/0
R       172.16.30.0 [120/2] via 172.16.10.2, FastEthernet0/0
R       172.16.20.0 [120/1] via 172.16.10.2, FastEthernet0/0
C       172.16.10.0 is directly connected, FastEthernet0/0
2621A#
```

4.5.3.7. 2501A:

```
2501A#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - [output cut]
Gateway of last resort is not set
     172.16.0.0/24 is subnetted, 5 subnets
R       172.16.50.0 [120/2] via 172.16.20.2, 00:00:11, Serial0
R       172.16.40.0 [120/1] via 172.16.20.2, 00:00:11, Serial0
R       172.16.30.0 [120/1] via 172.16.20.2, 00:00:11, Serial0
C       172.16.20.0 is directly connected, Serial0
C       172.16.10.0 is directly connected, Ethernet0
2501A#
```

4.5.3.8. 2501B:

```
2501B#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - [output cut]
Gateway of last resort is not set
     172.16.0.0/24 is subnetted, 5 subnets
R       172.16.50.0 [120/1] via 172.16.40.2, 00:00:26, Serial1
C       172.16.40.0 is directly connected, Serial1
C       172.16.30.0 is directly connected, Ethernet0
C       172.16.20.0 is directly connected, Serial0
R       172.16.10.0 [120/1] via 172.16.20.1, 00:00:04, Serial0
2501B#
```

4.5.3.9. 2501C:

```
RouterC#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - [output cut]
Gateway of last resort is not set
     172.16.0.0/24 is subnetted, 5 subnets
C       172.16.50.0 is directly connected, Ethernet0
C       172.16.40.0 is directly connected, Serial0
R       172.16.30.0 [120/1] via 172.16.40.1, 00:00:06, Serial0
R       172.16.20.0 [120/1] via 172.16.40.1, 00:00:06, Serial0
R       172.16.10.0 [120/2] via 172.16.40.1, 00:00:06, Serial0
RouterC#
```

4.6. IGRP (Interior Gateway Routing Protocol) Yönlendirme Konfigürasyonu:

RIP protokolüne benzer. Farkı ise her Router bir AS (autonomous system=müstakil sistem) değerinde çalışır. Bu AS değerini RIP deki Hop Count değeri gibi düşünebiliriz. AS değeri 1'den 65535 kadar değerlerde olabilir.

```
RouterA#config t
RouterA(config)#router igrp 10
RouterA(config-router)#network 172.16.0.0
```

4.6.1. 2621A:

```
2621A#config t
Enter configuration commands, one per line.  End with
CNTL/Z.
2621A(config)#router igrp ?
  <1-65535>  Autonomous system number

2621A(config)#router igrp 10
2621A(config-router)#netw 172.16.0.0
2621A(config-router)#^Z
2621A#
```

4.6.2. 2501A:

```
2501A(config)#router igrp 10
2501A(config-router)#netw 172.16.0.0
2501A(config-router)#^Z
2501A#
```

4.6.3. 2501B:

```
2501B(config)#router igrp 10
2501B(config-router)#netw 172.16.0.0
2501B(config-router)#^Z
2501B#
```

4.6.4. 2501C:

```
2501C(config)#router igrp 10
2501C(config-router)#netw 172.16.0.0
2501C(config-router)#^Z
RouterC#
```

4.6.5.Yaptığımız Konfigürasyonları Kontrol Edelim:

- ***Show protokol*** komutu:

```
2501B#sh protocol
Global values:
  Internet Protocol routing is enabled
Ethernet0 is up, line protocol is up
  Internet address is 172.16.30.1/24
Serial0 is up, line protocol is up
  Internet address is 172.16.20.2/24
Serial1 is up, line protocol is up
  Internet address is 172.16.40.1/24
2501B#
```

- ***Show ip protokol*** komutu:

```
2501B#sh ip protocol
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 6 seconds
  Invalid after 180 seconds, hold down 180, flushed after
240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 1, receive any
version
    Interface         Send  Recv   Key-chain
    Ethernet0         1     1 2
    Serial0           1     1 2
    Serial1           1     1 2
  Routing for Networks:
    172.16.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.40.2           120      00:00:21
    172.16.20.1           120      00:00:23
  Distance: (default is 120)
```

Bu RIP 120 değeri içindi. Şimdide RIP 100 değeri için bakalım:

```
Routing Protocol is "igrp 10"
  Sending updates every 90 seconds, next due in 42 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: eigrp 10, igrp 10
  Routing for Networks:
    172.16.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.40.2           100      00:00:47
    172.16.20.1           100      00:01:18
  Distance: (default is 100)
```

- ***Debug ip rip*** komutu:

```
2501B#debug ip rip
RIP protocol debugging is on
2501B#
07:12:56: RIP: received v1 update from 172.16.40.2 on
Serial1
07:12:56:      172.16.50.0 in 1 hops
07:12:56: RIP: received v1 update from 172.16.20.1 on
Serial0
07:12:56:      172.16.10.0 in 1 hops
07:12:58: RIP: sending v1 update to 255.255.255.255 via
Ethernet0 (172.16.30.1)
07:12:58:      subnet  172.16.40.0, metric 1
07:12:58:      subnet  172.16.20.0, metric 1
07:12:58: RIP: sending v1 update to 255.255.255.255 via
Serial0 (172.16.20.2)
07:12:58:      subnet  172.16.40.0, metric 1
07:12:58:      subnet  172.16.30.0, metric 1
07:12:58: RIP: sending v1 update to 255.255.255.255 via
Serial1 (172.16.40.1)
07:12:58:      subnet  172.16.30.0, metric 1
07:12:58:      subnet  172.16.20.0, metric 1
2501B#undebug all
All possible debugging has been turned off
2501B#
```

- ***Debug ip igrp*** komutu:

```
2501B#debug ip igrp ?
  events        IGRP protocol events
  transactions  IGRP protocol transactions
```

- ***Debug ip igrp events*** komutu:

```
2501B#debug ip igrp events
IGRP event debugging is on
07:13:50: IGRP: received request from 172.16.40.2 on
Serial1
07:13:50: IGRP: sending update to 172.16.40.2 via Serial1
(172.16.40.1)
07:13:51: IGRP: Update contains 3 interior, 0 system, and
0 exterior routes.
07:13:51: IGRP: Total routes in update: 3
07:13:51: IGRP: received update from 172.16.40.2 on
Serial1
07:13:51: IGRP: Update contains 1 interior, 0 system, and
0 exterior routes.
07:13:51: IGRP: Total routes in update: 1
2501B#un
07:13:52: IGRP: received update from 172.16.40.2 on
Serial1
07:13:52: IGRP: Update contains 1 interior, 0 system, and
0 exterior routes.
07:13:52: IGRP: Total routes in update: 1
2501B#un all
All possible debugging has been turned off
```

- ***Debug ip igrp transactions*** Komutu:

```
2501B#debug ip igrp transactions
IGRP protocol debugging is on
07:14:05: IGRP: received request from 172.16.40.2 on
Serial1
07:14:05: IGRP: sending update to 172.16.40.2 via Serial1
(172.16.40.1)
07:14:05:       subnet 172.16.30.0, metric=1100
07:14:05:       subnet 172.16.20.0, metric=158250
07:14:05:       subnet 172.16.10.0, metric=158350
07:14:06: IGRP: received update from 172.16.40.2 on
Serial1
07:14:06:       subnet 172.16.50.0, metric 8576 (neighbor
1100)

2501B#un all
All possible debugging has been turned off
2501B#
```

Firewall (Güvenlik Sistemleri)

GİRİŞ:

İnternet sistemi üzerinden bağlanan bilgisayarların birbirlerine ulaşabilmeleri bir güvenlik sorunu oluşturmaktadır. Bu nedenle networkte alınması gereken güvenlik önlemleri, internet bağlantısı devreye girdiğinde daha da artırılmalıdır. Özel kuruluşlar, Üniversiteler, kamu kuruluşları, bilgi paylaşımı, e-ticaret ve benzeri nedenlerle dahili bilgisayar ağlarını internete açmaktadırlar. Organizasyonlar bu sırada kendi ağlarında hayati öneme sahip kaynaklara istenmeyen erişimleri engelleyecek düzenlemelere ihtiyaç duymaktadır. Uygun güvenlik düzenlemeleri olmadan kurum kaynakları Hacker'lar rakip firmalar ve hatta işten çıkarılanların tehdidi altındadırlar.
Güvenlik konusu, özellikle son yıllarda gerçekten büyük önem kazanan ve e-iş projelerinin başarısında kilit rol oynayan unsurların başında gelmektedir. İletişim teknolojilerinin gelişmesi her ne kadar pozitif anlamda oldukça büyük getiriler sağlasa da aynı düzlemde düşünüldüğünde kötü niyetli yaklaşımlara da oldukça fazla imkan sunmaktadır. İşletmeler, gerek ticari gerekse ticari olmayan iletişimlerinde verinin bütünlüğünü korumak, doğru bilginin doğru alıcıya ulaştırıldığından emin olmak, iletişim altyapısına gelecek saldırıları bertaraf etmek ve olası saldırılara yönelik gerekli tedbirleri almak durumundadır. Bu tedbirlerin doğru biçimde alınması ve uygun yöntemlerin kullanılması için belli başlı saldırı tiplerinin ve güvenlik tehditlerinin bilinmesi önem taşımaktadır.

a. **Hacking ve Intrusion :** Bu tür saldırıların temel mantığı, işletmenin sistemlerine yetkisi olmayan birinin sızarak sistemin belli kısımlarının veya tamamının kontrolünü ele geçirmesidir. Bu saldırılar günümüzde yoğun olarak gerçekleştirilmektedir. Bu saldırılardan korunmanın yolu öncelikle sisteme giriş yapan kullanıcıların kimliklerini ve erişim yetkilerini denetleyen bir sistemin oluşturulması ve ayrıca firewall ve intrusion detection sistemlerinin kullanılmasıdır.
b. **Virüsler :** Virüs ve benzeri saldırılar, sistemi enfekte edecek olan zararlı bir kodun sisteme sokulması ve bilinçli veya bilinmeden çalıştırılması ile gerçekleştirilen saldırılardır. Söz konusu olan kod, çeşitli sistem kaynaklarına kendini kopyalayarak sistem içerisinde oldukça zarar verici bazı aktiviteleri tetikleyebilmektedir. Bu aktiviteler sonucunda sistem kısmen veya tamamen devre dışı kalabileceği gibi, sistem içerisindeki bazı bilgiler silinebilir, değiştirilebilir ya da virüs kendini sistem içerisindeki diğer alıcılara göndererek yayılabilir. Son yıllarda Melissa ve Lovebug virüsleri başta olmak üzere görülen birçok vakada oldukça ciddi zararlar ortaya çıkmıştır. Virüs saldırılarından korunmanın en genel yolu bir antivirüs yazılımı kullanmaktır. Günümüzde oldukça başarılı antivirüs yazılımları bulunmaktadır. Ayrıca firewall'lar içerisine de etkili virüs koruma sistemleri entegre edilmektedir. Bunun yanında ağ sisteminin açıklarından faydalanan solucanlar (worms) ve benzeri saldırıların önüne geçmek için doğrulama sistemlerinin de kullanılması gereklidir.
c. **Denial of Service:** Bir sisteme veya bir sisteme bağlı olarak çalışan alt sisteme kaldırabileceğinden daha fazla talepte bulunarak bu sistemin yavaşlamasına ve kilitlenmesine yol açan saldırı çeşididir. Genellikle saldırgan, İnternet üzerinden ele geçirdiği birçok farklı terminal üzerinden belli bir sisteme veya sunucuya çok fazla talepte bulunarak bu sistemin veya sunucunun önce yavaşlamasına, daha sonra devre dışı kalmasına neden olmaktadır. Geçtiğimiz yıl içerisinde Amazon.com' un maruz kaldığı benzeri bir saldırı sonucunda milyon dolarlarca zarar ortaya çıkmıştır.
d. **Veri trafiğine yönelik saldırılar:** Veri trafiğine yönelik saldırılar pasif saldırılar ve aktif saldırılar olarak tanımlanabilir. Pasif saldırılar, iletilen bilginin iletim yolu

boyunca izlendiği ama bilgi bütünlüğüne müdahale edilmediği saldırılardır. Aktif saldırılar ise bilginin iletim yolu içerisinde herhangi bir noktada değiştirilmesi veya iletim yolunun değiştirilerek saldırganın belirlediği farklı bir yoldan iletimin gerçekleşmesi yönünde saldırılardır. Bunun yanında saldırganın başkalarına ait olan makineleri ele geçirerek kimliğini gizlediği ve "spoofing" olarak adlandırılan saldırılar da bu kategoride değerlendirilebilir.

e. **İçten gelen saldırılar:** Buraya dek sözü edilen saldırıların tümünde kötü niyetli kişi veya saldırgan, sistemin dışından biri olarak kabul edilmektedir. Bunun yanında sistemin içerisinde olan ve dışarıdaki birine göre çok daha fazla avantajlara sahip olan bazı saldırganlardan söz etmek de mümkündür. Computer Security Institute'un FBI ile ortak gerçekleştirdiği, Bilgisayar Suçları ve Güvenlik-2001 araştırmasının sonuçlarına göre işletmelerin sistemlerine gelen saldırılardan %4'ünün içeriden, %22'sinin ise hem içeriden hem de dışarıdan geldiği bulgusuna erişilmiştir. Bu saldırılardan korunmanın yolu gelişmiş şifreleme algoritmalarından faydalanmak ve doğrulama sistemlerini hayata geçirmektir.

Görüldüğü üzere güvenlik sistemine yönelik saldırı yöntemleri oldukça çeşitlidir ve bu saldırılardan her biri işletmeye çok ciddi zararlar verebilir. Maddi zararların yanı sıra son derece değerli bilgilerin geri getirilemeyecek şekilde kaybedilmesi de söz konusudur.

Herhangi bir sistemin veya bir bilginin güvenliğinin sağlanması söz konusu olduğunda üç öğenin varlığı sorgulanmalıdır. Bu üç öğe, güvenlik kavramının en önemli fonksiyonlarıdır.

a. **Bilginin gizliliği :** Bilgiyi gönderen tarafın, bu bilginin sadece kendisinin belirlediği alıcı tarafından alındığından ve görüldüğünden emin olmasıdır.
b. **Bilginin bütünlüğü :** Gönderilen bilginin, gönderen tarafın oluşturduğu biçimde alıcıya iletilmesini, iletim sırasında herhangi bir müdahaleye uğramamasını ifade eder. Bilginin bütünlüğünün sağlanması için farklı şifreleme tekniklerinden yararlanılır.
c. **Doğrulama :** Alıcının; gönderilen bilginin, gerçekten gönderen tarafından gönderildiğinden emin olması için, gönderen tarafın kendini alıcıya ispat etmesidir.

Sözü edilen tehlikelerden korunmak ve olası tehlikelere proaktif yaklaşımlarda bulunmak için doğru güvenlik çözümünün seçilmesi ve seçilen çözümün mevcut altyapıya tamamen uyumlu kılınması gerekmektedir. Bu anlamda, günümüzde temel gereksinimlere cevap verecek bir güvenlik çözümünde bulunması gereken başlıca bileşenler şunlardır:

a. **Firewall ve intrusion detection:** Firewall, bir ağ yapısı içerisinde farklı ağ bölgelerinin birbirinden ayrılmasına yarayan güvenlik duvarıdır. Firewall, tanımlanan kurallara ve politikalara bağlı olarak sözü edilen farklı bölgelerin birinden diğerine olan geçişi kontrol eden bir yapıdır. Bir firewall, çeşitli protokollere dayanarak üzerinden akan ağ trafiğini süzer ve belirlenen kurallar çerçevesinde bu trafiğin şekillenmesini sağlar.

Günümüzde kullanılan firewall'lar, ağ trafiğini her noktada izleyebilecek bir yapıda olmalıdır. Çok katmanlı (multi-layer) olarak adlandırılan bu firewall'lar saldırı gelebilecek katmanlar olan ağ katmanı, uygulama katmanı ve anahtarlama katmanı gibi belli başlı katmanlarda ağ trafiğini denetleyebilmektedir. Son yıllarda firewall sistemlerinde kullanılan bir önemli teknoloji de DMZ (De-Militarized Zone) teknolojisidir. DMZ, doğrulanmamış kullanıcıların işletmenin ağına girmesini engelleyen bir sistemdir. İşletmenin dahili ağı ile İnternet arasında bulunan DMZ, İnternet üzerinden gelen tüm

kullanıcılara açıktır fakat yerel ağa herhangi bir bağlantı yapılmasını engeller. Ayrıca ağ içinden İnternete çıkmak isteyen kullanıcılar, DMZ üzerinden geçerek ağın güvenliğinde açık vermeden İnternete erişmiş olurlar. Intrusion detection sistemleri de firewall'a benzer biçimde ağ üzerindeki trafiği gözlemleyen sistemlerdir. Bu sistemler, çeşitli ağ algılayıcıları kullanarak ağ üzerinde gerçekleşen trafiği Denial of Service (DoS) saldırılarına ve yetkilendirilmemiş girişlere karşı izler. Intrusion detection sistemleri daha önceden tanımlanmış saldırı parametrelerine dayanarak ağdaki trafiği analiz eder. Ayrıca bu sistemler saldırıları da izleyerek saldırının ne derece etkili olduğu gibi bazı verileri ana merkeze ulaştırır.

b. **Erişim kontrolü (Access control) :** Herhangi bir ağı veya sistem kaynağını kullanmak isteyen kullanıcının kimliğinin, belirli politikalar ve mekanizmalar üzerinden tanımlanması işlemidir. Genellikle oluşturulan belli erişim kurallarını kullanarak çalışan ve kural tabanlı (rule based) olarak nitelendirilen erişim kontrol sistemleri, kullanıcıların kaynaklara erişimini güvenli bir biçimde sağlamanın yanı sıra erişimi belli politikalar ile düzenleyerek maliyet anlamında da bazı tasarrufların elde edilmesine yönelik kullanılabilmektedir.
c. **Yönetilebilirlik ve ölçeklenebilirlik:** Güvenlik sistemlerinin farklı parçalarının bir çatı altından merkezi yönetimi, bu kaynakların doğru konumlandırılması ve hedeflenen etkinlikte kullanılması açısından oldukça önemlidir. Sistem ve ağ ile ilgili gerek duyulan raporların düzenli ve mümkün olduğunca gerçek zamanlı sunulabilmesi, sistemin ve gerek duyulan kaynakların etkin biçimde izlenebilmesi ve önceden belirlenmiş durumlara dayanan çeşitli uyarıların gerektiği zamanda proaktif bir yaklaşımla yapılabilmesi yönetilebilirlik açısından önemli gereksinimlerdir.

Her geçen gün farklılaşan ve gelişen güvenlik donanım ve yazılımlarını sisteme kolaylıkla entegre edebilecek, işletmenin gelecekte gereksinim duyacağı farklı unsurları rahatlıkla bünyesine alabilecek, artan güvenlik gereksinimlerine mümkün olduğunca cevap verebilecek esnek bir yapı da güvenlik sisteminin önemli özelliklerinin başında gelmektedir.

Firewall her zaman güvenlik nedenleri ile devreye sokulmaz. Gerekçelerden biri de istenilmeyen yerlere erişimi engellemek olabilir. Firewall mantığında hedeflenen makinanızın veri iletişiminde kullandığı portların ve bu portlar üzerinden yapılan veri alışverişinin kontrolüdür. Bu kontrolü sağlamak için bazı portların kapatılması, bazı portların sadece bir tür veri alışverişine izin vermesi, sadece yönlendirme yapması gibi metotlar gereklidir. Portları dinleyip gelen paketleri sorgulamak ve istenilen türdeki veri paketlerini "yakalayıp" bu paketler için tanımlanacak "yaptırım"ları uygulamak da olan **Ipchains** yardımıyla mümkündür.

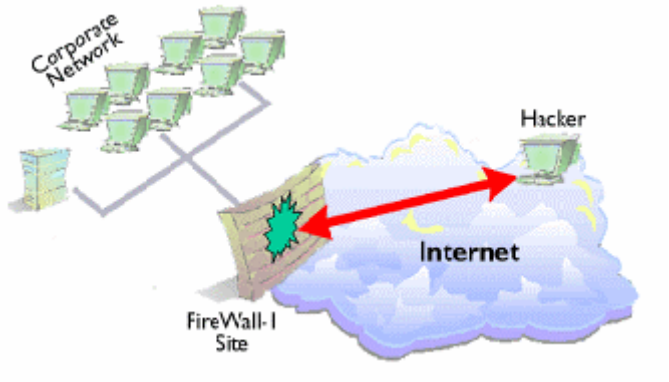
Firewall, İnternet üzerinden bağlanan kişilerin, bir sisteme girişini kısıtlayan/yasaklayan ve genellikle bir İnternet gateway servisi olarak çalışan bir bilgisayar ve üzerindeki yazılıma verilen genel addır. Firewall sistemleri, bu engelleme işini, sadece daha önceden kendisinde tanımlanmış bazı domainlere erişim yetkisi (telnet,ftp, http vb) vererek yaparlar. Günümüzde, Internet Servisi veren makineler oldukça sofistike Firewall sistemleri ile donanmıştırlar.

Firewall sistemlerinin en önemli kullanımında, bir kuruluşun yerel ağındaki tüm bilgisayarlar sanki bir duvar arkasındaymış gibi dış dünyadan erişilmez olurlar. Yerel ağdaki bilgisayarların İnternet bağlantıları firewall yüklü bilgisayar üzerinden olur ve firewall yazılımları, adres çevirme (address translation) özellikleri sayesinde iç ağdaki tüm İnternet bağlantılarının tek bir IP numarasından yapılıyormuş gibi olmasını sağlarlar. Ayrıca, yerel

ağdan dışarıya, dışardan da yerel ağa olan bağlantıları İnternet protokolleri üzerinde sofistike kurallar tanımlayarak kısıtlarlar bu da yüksek ölçüde ağ güvenliği sağlar. Bir ağ firewall'u ağ güvenliği sağlamanın en etkili yoludur. Firewall'lar dahili ağı ve internet gibi harici ağlar arasında bir engeldir. Üzerlerinden geçen tüm trafiği inceleyerek çalışırlar ve sadece izin verilen trafiğin geçişine izin verirler.

Firewall'lar statik veya dinamik olabilir. Dinamik Firewall'lar değişen durum ve şartlara göre kendilerini adapte edebildiklerinden dolayı daha güvenlidirler. Üzerlerinde, güvenlik parametre-lerini belirlemek için kullanıcı arabirimleri vardır. Bir firewall yalnız başına çalışabilen bir sistem olabileceği gibi bir ağ cihazının bir bileşeni de olabilir. Firewall'un ağ üzerindeki mantıksal konumu internet bağlantısına ne kadar yakın olursa, o kadar iyi güvenlik sağlar. Bu durumda ideal durum firewall'un erişim cihazlarından birisi olmasıdır. Erişim cihazı tarafından zaten izlenmesi gereken trafik firewall özellikleri ile belirlenen güvenlik politikasına göre filtrelenerek ağa ulaşmadan güvenlik otomatik olarak sağlanmış olur.

5.1.Erişim Denetimi



Internet'e bağlı kurumların mutlaka dikkate almaları gereken konu erişim denetimidir. Erişim denetimi ile kurum içi network (LAN , WAN da olabilir) ile Internet'in arasındaki erişim kurallarının belirlenmesidir. Yani Internet üzerinden hangi kaynaklara, hangi şartlar altında ve nasıl ulaşılacağı, erişim denetimi ile belirlenir.

Erişim denetimi firewall ile sağlanır. Firewall ile network üzerinden çıkışı ve giriş arasındaki belirlenen kurallar çerçevesinde ayrıca geçen trafiğin loglanması, tanımlanan kurallar ile çeşitli hakların belirlenmesi sağlanabilir. Firewall, bir kurumun güvenlik sisteminin temelini oluşturmaktadır. Güvenlik sistemi içerisinde düşünülecek diğer yapılar firewall'a dayanmaktadır.

Firewall olarak yazılım ve donanım tabanlı seçenekler mevcuttur. İşletim sisteminden bağımsız olması, yüksek performans sunması ve yedeklenebilirlik nitelikleri nedeniyle donanım tabanlı çözümler tercih edilebilir. Dünyanın en büyük Network ekipmanı üreticisi Cisco Systems, donanım tabanlı firewall olarak iki seçenek sunmaktadır:

Cisco PIX Firewall: Sadece Firewall işlevleri için tasarlanmış yüksek performanslı bir donanımdır. Çok sayıda kullanıcının olduğu sistemlerde yüksek performans elde edilmek istendiğinde tercih edilmektedir.

Cisco IOS Firewall: Cisco Router'ları üzerine yüklenerek, Router ile entegre bir firewall elde edilmek istendiğinde kullanılmaktadır. Router ile entegre olması, yönetim ve maliyet açılarından avantaj sağlamaktadır.

Ağ güvenliği konusunda sorumlu kişiler ve ağ yöneticileri güvenlik konusunda su problemlerle karşı karşıyadır :

Yeterli Güvenlik - Basit şifrelerden kompleks firewall'lara kadar değişen farklı güvenlik seviyeleri mevcuttur.
Yetkili Kullanıcılara Saydam İşletim - En bilinçli kullanıcılar dahi, kullanımı zor güvenlik sistemlerini gerektiği gibi kullanmamaktadırlar. Bu yüzden tüm tedbirler mümkün olduğunca göze batmayacak şekilde alınmalıdır.
Yönetim Yükünü En Aza İndirmek - İyi bir güvenlik sistemi kolaylıkla ayarlanmalı ve yönetilebilmelidir. Ayrıca sistemin yönetsel işlevlerinin sıradan kullanıcılar tarafından erişilemez olması gerekmektedir.
Uygun Bütçeler - İhtiyaç duyulan güvenlik sistemlerinin kurulmasında ki en büyük engel maliyettir. Oysa, toplam bir ağ çözümünün parçası olarak ele alındığında, güçlü ağ güvenlik yapılarının pahalı olması şart değildir.

5.2.Bir Firewall 2002'nin özellikleri
Güçlü firewall teknolojisi, PC'den internete yapılan tüm bağlantıları denetler. Otomatik Bloklama ve saldırı tespiti özelliği, otomatik olarak 'Hacker'ların kullanıcı sistemine erişimini engeller.
Uygulama Denetimi özelliği, kullanıcının bilgisi olmadan bilgisayara yüklenen ve çalıştırılan Truva ati ve benzeri zararlı uygulamalara karşı koruma sağlar. Norton Personal Firewall, kullanıcının, güvenilir olmayan web sitelerine vermek istemediği bilgileri filtreler ve böylece kişisel bilgileri gizli tutar. Home Network Wizard, otomatik olarak kullanıcının network'ü üzerindeki sistemleri tespit eder ve böylece kurulumu kolaylaştırır. Güvenlik Yardımcısı, kişisel firewall konfigürasyonunun kolay bir şekilde yapılmasını ve uyumlu çalışmasını sağlar.
Dünyanın en güvenilir antivirüs çözümü Norton AntiVirüs 2002 ile tümüyle entegredir.
Genel Özellikler: Çok yönlü hibrid yapısı aynı anda hız ve güvenlik sağlar Günlük çalışmayı etkilemeden güçlü saldırı tespiti yapar Saldırı riskini ve güvenlik açıklıklarını azaltmak için yükleme esnasında ve daha sonra sürekli olarak sistem sağlamlaştırması yapar Güçlü kimlik doğrulama alternatifleri varolan güvenlik veritabanı kullanımında esneklik sağlar. ICSA sertifikasyonu endüstride lider güvenlik gereksinimleri ile uyumluluğu garanti eder Opsiyon olarak Symantec Enterprise VPN (eski adi ile Power VPN) ile uzaktan erisen kullanıcılar için düşük maliyetli, güvenli bağlantı sunar
5.3.Endüstrideki en güvenli ve hızlı güvenlik duvarı
Symantec Enterprise Firewall, ağlar ve Internet arasındaki güvenli haberleşmeyi garantilemek için en güvenilir ve yüksek performanslı çözümlerden biri ile kurumun değerli bilgilerini ve isletme iletişimini korur. Essiz hibrid yapısı ile güvenlik ve hız sunar, güçlü ve geçirgen güvenlik duvarı ile trafikte herhangi bir yavaşlamaya sebep olmadan istenmeyen kullanıcılara ve saldırganlara karsı koruma sağlar. Uzaktaki ofislere güvenli bağlantı yapmak için opsiyonel olarak Symantec Enterprise VPN (eski adıyla Power VPN) ve kişisel güvenlik duvarı ile Symantec Enterprise Firewall entegrasyonu sağlanabilir. Ödül almış yapısı ile Symantec Enterprise Firewall güvenlik açıklıklarını en aza indirme ve tam güvenlik sağlama ile is ortakları ve müşterilerin şirket kaynaklarına sorunsuz ve kesiksiz bir şekilde bağlandıkları sırada içeriye giren ve dışarı çıkan bilgi üzerinde tam bir kontrol garanti eder. En çok tercih edilen Radius, Digital Certificates, LDAP, ve NT etki alanı kimlik doğrulaması

gibi kimlik doğrulama metotlarını desteklemesi ile ağ yöneticilerine varolan güvenlik veritabanı kullanımında esneklik sağlar. Ve sistemin her zaman çalışır durumda ve ölçeklenebilir olması için, Symantec Enterprise Firewall donanım ve yazılım tabanlı yüksek seviye kullanılabilirlik ve yük dengeleme çözümleri sunar; ayni zamanda Web ve Usenet içerik filtreleme ürünleri ile de entegrasyonu vardır. Windows NT/2000 ve Solaris platformları için geliştirilmiş olan Symantec Enterprise Firewall, konfigürasyonu, yönetimi ve bakimi kolaylaştırmak için çok geniş arayüz ve cihaz seçeneğini destekler. Merkezi bir konsoldan, ağ sorumluları yerel ve uzaktaki güvenlik duvarları üzerinde bulunan güvenlik politikalarını yönetebilir, kayıtları ve yönetim raporlarını elde edebilirler. Symantec Enterprise Firewall günümüzün en siki güvenlik, farklı işletimlerle uyumlu çalışabilme ve endüstrideki sertifikasyon gereksinimleri gibi ihtiyaçları karşılar.

5.4.Cisco Discovery Protocol Neighbor Announcement DoS

Cisco Discovery Protocol (CDP) Cisco Internet Operating System ile birlikte gelen bir ağ komşu kesif protokolüdür. CDP Cisco IOS`un bazı sürümleri ile uygulanmaktadır. Ağın yerel kısmındaki bir hostun bir Cisco router`ın isini sağlıklı yapamayacak duruma düşürmesi ve geniş miktarda CDP trafiği yaratarak router`in trafiği route etmesini engellemesi mümkün. Bu protokol router`lar arasında uzaktaki ağ bölümlerine route edilemez. Bu Cisco router`ların işlemini azaltmaya ve bir denial of service`e yol açabilir.

Çözüm: Cisco tarafından önerilen geçici çözüm: Bu açık için geçici çözüm CDP`yi kapatmaktır. CDP`yi tüm router için kapatmak için aşağıdaki global komutları çalıştırın:

Router# configure terminal Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# no cdp run
Alternatif olarak, CDP belirli bir arabirimde de kapatılabilir.
Router(config)# interface Ethernet0
Router(config-if)# no cdp enable

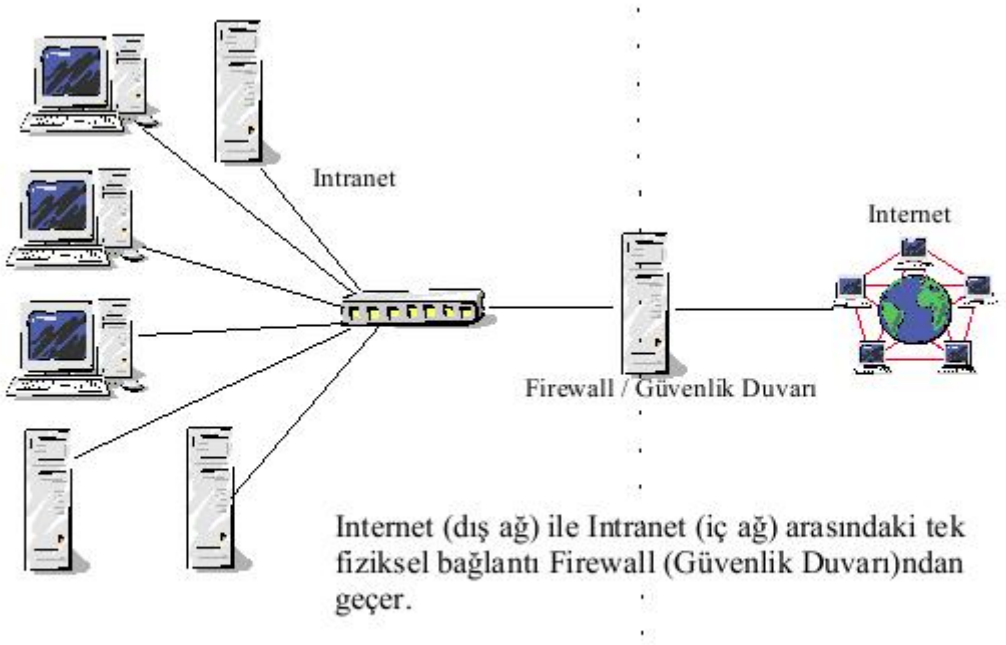
5.5. Güvenlik Duvarı Kavramları

5.5.1. Tabya (Bastion Host)

İdealde, ağınızdaki güvenlik, ağ seviyesinde ve ağdaki her bir makinada uygulanır. Pratikte ise, bu ya yapılamamakta, ya da ihtiyaç duyulan kimi protokollerin güvenli olmadıkları bilinse dahi kullanılmaktadır. Böyle durumlarda güvenlik duvarı, içeride birbirlerine güvenen, az korumalı makinaların olduğu bir ağla, dış dünya arasına yerleştirilir ve aradaki fiziksel bağlantı yalnızca güvenlik duvarı tarafından sağlanır. Dolayısıyla içerideki ağa girmek isteyen her kötü niyetli dış saldırı, önce özel olarak korumalı tasarlanmış güvenlik duvarı makinasını bertaraf etmek zorundadır. Bu makinaya "kale", "nöbetçi kale" anlamına gelen *tabya* da denir. *Tabya*, fiziksel olarak iki farklı ağa bağlıdır: iç ağ (Intranet) ve dış ağ (Internet). *Tabya* iki özelliğe sahiptir:

- Yüksek güvenlikli olmalıdır, buna izinsiz erişim son derece zor hale getirilmelidir.
- İki (bazen üç) fiziksel ağ bağlantısına sahip olmalı ve bu farklı ağlar arasındaki iletişimin nasıl yapılacağına dair karar verebilmelidir.

Şekil 2. İç ağ ile dış ağ arasında güvenlik duvarı



5.5.2. Ağ Adres Çevrimi (NAT), Maskeleme

Günümüzde iç ağların hemen hepsi tahsisli olmayan IP numaraları (`10.0.0.0`, `192.168.0.0` vs.) kullanmaktadır. Bu IP numaraları Internet üzerindeki yönlendiriciler (router) tarafından bilinmez. Dolayısıyla bu ağlardan Internet'teki herhangi bir makinaya bir erişim olduğu zaman Internet'teki makina bu ağa nasıl geri döneceğini bilmez ve pratikte iletişim yapılamaz. Güvenlik duvarı ise, dinamik veya statik olarak Internet'te bilinen ve kendisine yönlendirme yapılabilen bir IP numarasına sahiptir. İç ağdaki makinalara erişim sağlayabilmek için güvenlik duvarı, kendisine iç ağdan gelen her paketin kaynak adresini kendi adresi olarak değiştirir. Kendisine Internet'ten gelen paketlerin de hedef adresini iç ağdaki ilgili makinanın adresi olarak değiştirir ve bu yolla iç ağdaki makinaların Internet üzerindeki makinalarla haberleşmesini sağlar. Bu işleme *IP Maskelemesi* veya *Ağ Adres Çevrimi* (Network Address Translation) denir. NAT yapıldığı zaman, oluşan trafiğin Internet'ten görüldüğü hali, Internet'te bulunan tek bir makinanın bazı Internet alışverişleri yaptığıdır. Internet'e, bu duvarın arkasındaki ağın büyüklüğü, bu ağdaki makinaların cinsi, sayısı, vs. hakkında herhangi bir bilgi gitmez. Dolayısıyla NAT, yalnızca tahsissiz ağlardan Internet'e erişimi sağlamakla kalmaz, ağınızdaki makinalar hakkında bilgi edinilmesini, bize karşı yapılabilecek saldırıları zorlaştırır.

5.5.3. Paket Filtreleme

Yukarıda bahsedilen önlemler ağa belli bir miktar güvenlik sağlar, fakat esas güvenlik, paket filtreleme yöntemlerinden gelir. Bu yöntemler, güvenlik duvarından geçen her IP paketine bakılması ve ancak belli şartlara uyarsa geçişine izin verilmesi şeklinde uygulanır. Örneğin:

- İç ağınızdan kimsenin Internet'te ICQ kullanmasını istemiyorsunuz.
- Dışarıdan içeriye hiç kimsenin telnet yapabilmesini istemiyorsunuz.

Bu hedefleri gerçekleştirmek için *paket filtreleme* yöntemleri kullanılır. *Paket filtreleme*, güvenlik duvarının her fiziksel bağlantısı üzerinde ayrı ayrı ve yöne bağlı (dışarıya çıkışa izin ver, fakat içeriye girişe izin verme) olarak uygulanabilir.

Paket filtreleme de özellikle yapılması gereken minimum, dışarıdan gelip de kaynağını içerisi gibi gösteren (IP spoofing - IP aldatmacası) paketleri ve devam etmekte olan bir

trafiğin parçası imiş gibi gelen paketleri (IP fragments) filtrelemek ve bunların geçişine izin vermemektir. Çoğu saldırı, bu şekilde başlar.
Bu minimumu sağladıktan sonra, dışarıdan içeriye yapılmasına izin verilen erişimleri (telnet yapsınlar mı?, ping yapabilsinler mi?) ve içeriden dışarıya yapılmasına izin verilen erişimleri (kullanıcılarınız dışarıya telnet yapabilsin mi? Web'e erişsinler mi? ICQ yapabilsinler mi?) belirlemek ve güvenlik duvarı üzerindeki filtre protokolleri buna göre oluşturmak gerekir.

5.5.4. Dinamik Filtreleme

Eskiden filtreleme yöntemleri ağırlıklı olarak statikti. Yani genel olarak ağınıza ICQ paketlerinin girmesine izin verip vermeme kararı söz konusu idi. 2.4 Çekirdeği ve bizim aşağıda örneğini verdiğimiz **iptables** uygulaması ile birlikte *dinamik filtreleme* Linux üzerinde kullanılabilir hale geldi. Aradaki fark, paketin sırf protokolüne bakarak karar vermek yerine, güvenlik duvarının bir bağlantıyı hangi tarafın başlattığını takip etmesi ve çift yönlü paket geçişlerine buna göre karar vermesidir. Yani bir telnet bağlantısında her iki taraftan da paketler gelir ve gider. Fakat *dinamik filtreleme* ile, bir telnet bağlantısı iç ağınızdan başlatılmışsa izin verir, başlangıç istemi dış ağdan gelmişse reddedebilirsiniz. *Dinamik filtreleme* özelliği olmayan güvenlik duvarlarının kullanılması pek önerilmez. 2.4 çekirdeği ve **iptables** uygulaması olan her Linux üzerinde *dinamik filtreleme* yapılabilir.

5.5.6. Bazı Internet Servislerinin İç Ağdan Verilmesi

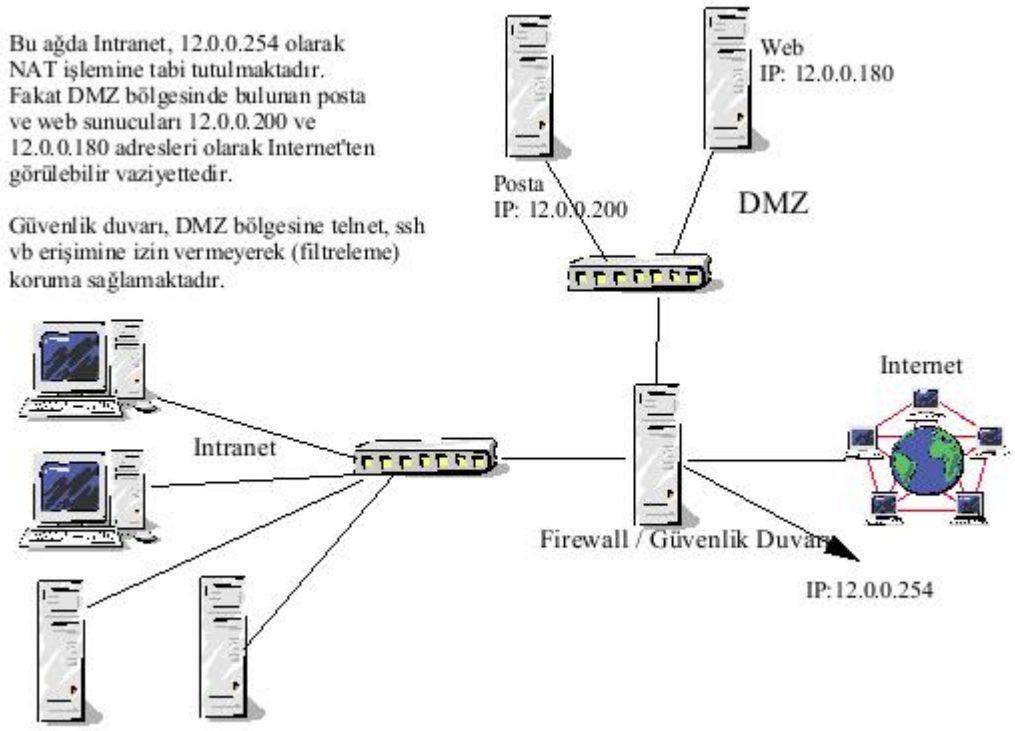
Ağda, internet'ten erişimi olması gereken web, posta gibi sunucular bulunabilir. Bu sunuculara erişimi iki yoldan vermek mümkündür:

- Silahsızlandırılmış bölge uygulaması (DMZ - Demilitarized Zone)
- İç ağda bu servislere doğrudan filtreleme yaparak.

5.5.6.1.Silahsızlandırılmış bölge (DMZ - DeMilitarized Zone)

DMZ, güvenlik duvarı tarafından daha az korunan, daha fazla erişime izin verilen bir bölgedir. Güvenlik duvarına üçüncü bir ağ çıkışı eklenmesi ve Internet'e servis verecek olan makinaların buraya konulması ile oluşturulur. Örneğin DMZ'deki makinalara NAT uygulanmayabilir, tahsisli IP numaralarına sahip olabilirler. Firewall, telnet, SSH gibi kimi protokollerin buraya erişimini filtreleyerek DMZ bölgesindeki makinalara güvenlik sağlar. Dikkat edilecek nokta, DMZ' de bulunan makinaların daha fazla erişime (ve dolayısıyla saldırıya) açık olmasıdır. Buradaki makinalar dikkatli kurulmalı, güvenliğe aykırı protokoller vs. burada yer almamalıdır.

Şekil 3. Silahsızlandırılmış bölge (DMZ)



5.5.6.2.Doğrudan Filtreleme

DMZ oluşturmak için ek ekipman ve IP numarası gerekir. Güvenlik duvarında üçüncü bir ağ birimi, ayrı bir switch, daha fazla adette tahsisli IP numarası, ve iç ağınızda başka herhangi bir görev görmeyecek olan sunucu makinalar gerekir. Eldeki imkanlar buna yetişmeyebilir. Böyle durumlarda, güvenlik duvarındaki filtreleme politikasını değiştirerek iç ağdaki kimi makinalara dışarıdan sınırlı erişim imkanı verilebilir. Örneğin güvenlik duvarı ağ genelinde dışarıdan gelen SMTP protokolünü filtrelerken, sadece posta sunucusuna dışarıdan SMTP protokolü erişimini verebilir. NAT ile birleştirileceğinden, bu dışarıdan bakıldığı zaman sanki güvenlik duvarı posta sunuculuğu yapıyormuş izlenimini verir.

5.6. Vekil (Proxy)

Proxy'nin kelime anlamı vekildir. Yukarıdaki yöntemlerin hepsi, belli kurallara bağlı olarak Internet'teki bir makina ile iç ağdaki bir makina arasında doğrudan alışverişe izin verir. Vekil uygulamaları ise, bu doğrudan alışverişin arasına girer. Dolayısıyla protokol bazlı herhangi bir saldırı, vekil sunucuya yönelik gerçekleşir, iç ağdaki makinayı etkilemez. Örneğin bir http (web) vekili, iç ağdan dışarıya giden bütün web isteklerini toplar. Bu istekleri kendisi yapar, gelen sonuçları iç ağa dağıtır. Örneğin eğer web protokolü yolu ile istemci makinanın bazı bilgilerinin alınması veya bir saldırı yapılması söz konusu olur ise, bundan etkilenen sadece web vekili makina olur, iç ağda web erişiminde bulunan her makina değil. Güvenlik amacı ile proxy kullanımı, ***uygulama temelli güvenlik duvarı*** (application level firewall) olarak adlandırılır.

Proxy servisi, İnternet üzerindeki yerel bir ağ (ya da İnternet'e bağlı bir bilgisayar) ile, dış dünya arasındaki ilişkiyi sağlayan bir yardımcı geçiş (gateway) sistemidir. İki amaç için kullanılabilirler :

1. Bir proxy servisi (sunucusu), sizin adınıza (proxy'nin kelime anlamı VEKİL' dir) sizden aldığı "İnternet'ten bilgi alma" isteklerini yürütür ve sonucu yine size iletir.

Ancak, aynı anda, bu bilgilerin bir kopyası da, proxy sunucusu üzerinde tutulur ve bir dahaki erişimde kullanıcının istediği bilgiler doğrudan ilgili siteden değil de, proxy servisinden gelir; ve iletişim daha hızlı olur. Internet'e erişim için mutlaka bir proxy servisine ihtiyaç yoktur, ancak, en yakın servis noktasındaki proxy servisini kullanmak, İnternet erişimini bir hayli hızlandıracaktır. Özellikle evden modemle internete erişim yapılıyorsa, proxy servislerini kullanmak performansı arttırır. Çünkü, istenilen bilgiler, dış bağlantı hızı daha fazla olan proxy bilgisayarı alır, kullanıcı bu bilgilere daha hızlı erişmiş olur.

2. Firewall-güvenlik sistemlerinin kullanıldığı yerlerde, kullanıcıları çıkışları tek bir makine üzerinden olabilir. Bu durumda proxy servis makinesi sadece bir aracı olarak çalışır.

Proxy servisi kullanmanın avantajı çoktur. Herhangi bir siteden istenilen bir bilgi (web sayfası, ftp dokümanı vb) eğer kullanılan proxy servisinde henüz depolanmamışsa, bu bilginin olduğu siteden alınır ve istemciye iletilir. Ancak, daha sonra başka bir kullanıcı aynı dokümanı istediğinde, bu doküman proxy servisinde depolandığı için, doğrudan oradan alınır ve erişim çok daha hızlı olur.

Proxy servisleri, uluslararası internet bağlantılarındaki yoğunluğu azaltmak, erişimleri hızlandırmak ve ağı daha etkin kullanmak için çok yararlı araçlardır.

En popüler proxy servisleri, Web (http), FTP, Gopher ve Wais internet araçları için tanımlıdır. Akademik Ağ üzerinden çıkış yapanlar ya kendi kurumlarındaki proxy sunucusunu ya da ulaknet'in proxy sunucusunu kullanabilir. Benzer şekilde, TTNet ve diğer servis sağlayıcıların da proxy sunucuları vardır.

Yukarıdakilerin dışında, birçok kuruluşun kendi özel proxy servisleri vardır. Size en yakın proxy servisini kullanmanız çoğunlukla en iyi sonucu verir. Bu yüzden, internet servisini aldığınız yerin proxy servisini öğrenin ve kullanmaya çalışın. Önemli proxy servislerin bir zincir oluşturarak ortak kullanımına yönelik çalışmalar da yapılmaktadır.

5.6.1. Vekillerin Başka Kullanımları

- **Güvenlik amaçlı** - yukarıda bahsedilmiştir.
- **İzin amaçlı** - İç ağdan bazı servislere kimin erişebileceğini belirlemekte kullanılırlar.
- **Performans amaçlı** - Pek çok istemci aynı istekte bulunuyorsa, bunların bir defaya indirgenmesini sağlayarak hem sunucu makinanın üzerindeki yükü, hem de kullanılan bağlantı yükünü hafifletirler.

5.7.FİREWALL KONFİGÜRASYONU

5.7.1.Firewall Konfigürasyonu İçin Kılavuz Bilgiler:

Bütün network aygıtları gibi bir Firewall için giriş için yapacağımız güvenlik önlemi, her zaman password konfigürasyonu ile olur. Aşağıdaki tanımlamaları inceleyelim:

- Firewall girişi için ***enable password*** komutu yada buna tercihen ***enable secret*** komutu kullanılarak 'privileged password' kurulmalıdır.

- Console portuna şifre koyma iki komut ile olur. Önce ***login*** girilir ve ardından ***password 9817*** yazılır. (9817 benim şifrem. Bunun yerine farklı bir şifrede yazılabilir.)
- Console portundan yapılacak girişlerle; giriş kontrol konfigürasyonu da dahil olmak üzere Firewall'un bütün kontrolü ele geçirilebilir. Bunun üzerinde etkili bir koruma yapılmalıdır.
- Bütün sanal çıkış portlarının güvenliği için password yada erişim listelerine müracaat edilir.
- SNMP yada NTP gibi herhangi bir lokal servis etkin kılınmaz. CDP protokolünü kapatmak için ***no cdp run*** global konfigürasyon komutu kullanılır. NTP protokolünü kapatmak içinse ***ntp disable*** komutu kullanılır. Eğer NTP konfigürasyonu gerekirse sadece gerekli olan interface'de bu ayarlama yapılır. Çünkü yukarıda etkin kılınmış her bir servis bir potansiyel güvenlik riskidir. Saldırganlar bu servisleri kullanarak Firewall yada networke giriş yolları bulabilirler.
- Kaynak yönlendirmeyi 'disable' edebiliriz. IP yönlendirmesi için ***ip source-route*** global konfigürasyon komutu kullanılır.
- IP için ***no service tcp-small-servers*** yada ***no service udp-small-servers*** komutları kullanılarak tcp ve udp servisleri kapatılabilir.
- Normalde Router ve Firewall üzerindeki çalıştırılabilir bütün protokoller için Broadcast'ler doğrudan 'Disable' edilmeli. Çünkü veri akışı Broadcast'ten yapılır. Broadcast üzerinden subnet içindeki bütün kullanıcılara ulaşılabilir. O yüzden Broadcast'e büyük saldırılar olabilir.
- Son olarak Firewall, organizasyon içerisinde gizli ve güvenli bir yere saklanmalıdır.

5.7.2.CBAC

Bu bölümde CBAC konfigürasyonu anlatacağız. CBAC trafik denetleme avantajı sağlar.

5.7.2.1.CBAC Ne Yapar:

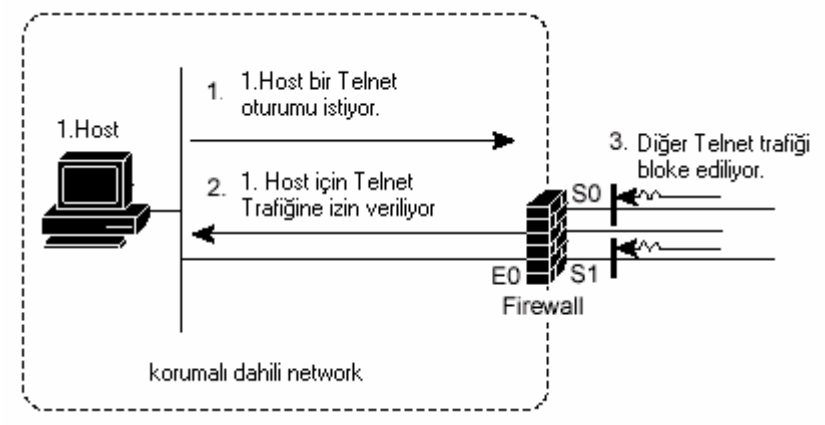
İnternet, intranet yada extranet'lerde kullanılan TCP ve UDP paketlerdeki uygulama katmanı oturum bilgisi CBAC tarafından denetlenir. Sadece istediğimiz güvenlik network içinde bağlantı gerçekleştiği zaman, Firewall içinde özel TCP ve UDP trafiğine izin verilerek CBAC konfigüre edilebilir. CBAC sadece network katmanın değil, ulaşım katmanı bilgisini, uygulama katmanı bilgisini ve TCP ve UDP oturumlarının durumlarını gözden geçirir.

CBAC korumalı bir networkün içerisinden gelen saldırılıları koruyamaz. CBAC sadece dışardan gelip Firewall içerisinden geçen saldırıları sezebilir. CBAC bu saldırılara çok güçlü olmadıkları sürece karşı koyabilir.

5.7.2.2.CBAC Nasıl Çalışır:

CBAC Firewall interface'inde açılan geçici erişim listeleri oluşturur. Bunlar Firewall ile dahili network arasında özel bir trafik meydana geldiği zaman oluşturulur. Şekilde S0 ve S1 sınırında erişim listeleri, erişim listeleri Telnet trafiğini bloke edecek şekilde konfigüre

edilmişlerdir. 1. Host Firewall üzerinden bağlantı yapmak istediği zaman, CBAC 1. Hostun Telnet Oturum isteği için S0 çıkışında bir erişim listesi yaparak, geçici bir oturum oluşturur.



CBAC ile hangi protokolün hangi arayüzde çalıştırılması isteniyorsa bu şekilde konfigürasyon yapılabilir. Ancak özel protokollerin CBAC tarafından denetlenmesi gerekir.

5.7.2.3.CBAC İŞLEMİ:

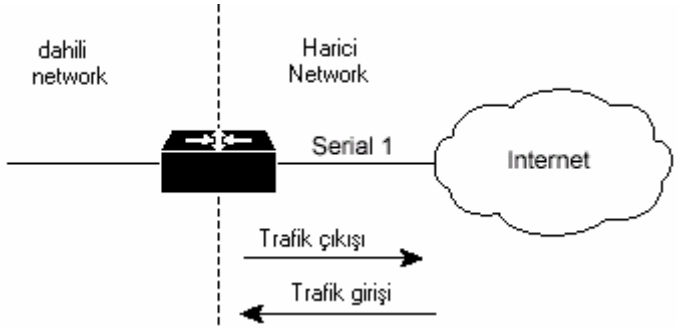
Bu örnekte bir TCP paket dahili networkten Firewall'un harici arayüzünden çıkar. Ve bu paket bir Telnet oturumunun ilk başlangıç paketidir. Ayrıca Telnet oturumu için CBAC de konfigürasyon yapılmıştır. Şimdi TCP paket ilerlerken hangi aşamalardan geçiyor bunlara bakalım.

1. Paket Firewall'un harici arayüzüne ulaşır.
2. Paket arayüzde erişim listesine bakılarak kontrol edilir ve Paketin geçişine izin istenir.
3. Paketin geçişine CBAC izin verir ve paketin bağlantısının konumu hakkındaki bilgi kaydedilir. Bu bilgi yeni bir bağlantı için yeni bir durum tablosunda kayıt oluşturulur.
4. CBAC genişlemeli erişim listesini harici arayüzün eşiğine yerleştirip geçici bir erişim listesi kaydı oluşturur.
5. Bu paket çıkıştaki arayüzden gönderilir.
6. Daha sonra aynı arayüzden bir paket alınır. Bu paket önceki telnet bağlantısının devamıdır ve CBAC den geçiş izni ister.
7. CBAC paketi denetleyip geçişine izin verir ve bağlantı durum tablosu gerekli biçimde güncellenir.
8. herhangi bir eklentide paketlerin kayıtları bu tablolara yazılır.
9. Eğer bağlantı zaman aşımına uğrarsa bağlantı durum tablosu kaydı silinir. Ve bu durum geçici erişim listesine yazılır.

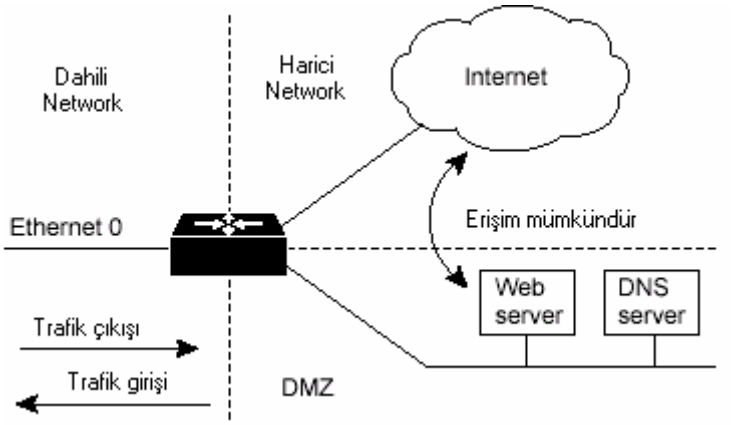
5.7.3.CBAC KONFİGÜRASYONU:

5.7.3.1.Bir Arayüzü Dahili Yada Harici Olarak Seçmek:

Bir Firewall arayüzü konfigürasyonu yapılırken bunun dahili yada harici arayüz olduğunun bilinmesi gerekir. Firewall kabaca temel iki network topolojisinden biri ile kullanılır. Birinci topoloji aşağıdaki şekilde görüldüğü gibi basit bir topolojidir. CBAC, S1 arayüzü için **external** olarak konfigüre edilmiştir. Bu dışarıdan Firewall'a giren özel protokol trafiklerinin dahili networke geçişini engeller.



İkinci topoloji aşağıdaki şekilde gösterilmiştir. Bu topolojide CBAC konfigürasyonu yapılırken Ethernet 0 arayüzü **internal** olarak konfigüre edilmiştir. Bu yöntem ile harici trafiğin DMZ bölgesine girişine izin verilir. Ancak dahili networklere erişimleri engeller.



5.7.3.2.Arayüzde IP Erişim Konfigürasyonu:

CBAC'in düzgün çalışması için interface'de uygun IP erişim listeleri konfigürasyonunun yapıldığından emin olmak gerekir.

5.7.3.3.Global Timeouts ve Thresholds:

CBAC Timeouts ve Thresholds, konfigürasyonu bir oturum için durum bilgisi idaresinin ne kadar süreceğine karar verir. Tabi komutlar yazılırken bunlara özel değişken değerlerle birlikte yazılırlar.

Aşağıdaki tabloda kullanılan komutlar ve değişkenleri belirtilmiştir.

KOMUT	AÇIKLAMA	DEĞİŞKEN
İp inspect tcp synwait-time *saniye*	Bir TCP oturumunun başlangıcından düşmesine kadarki süre	30 saniye
ip inspect tcp finwait-time *saniye*	Firewall'da FIN algılandıktan sonra bekleme süresi	5 saniye
ip inspect tcp idle-time *saniye*	TCP yarı-açık bekleme süresi	3600 saniye (1 saat)
ip inspect udp idle-time *saniye*	UDP yarı-açık bekleme süresi	30 saniye
ip inspect dns-timeout *saniye*	DNS yarı-açık bekleme süresi	500 oturum
ip inspect max-incomplete high *değer*	max mevcut oturum sayısı	400 oturum
ip inspect one-minute low *değer*	minimum mevcut oturum sayısı	500 oturum/dk
İp inspect tcp max-incomplete host *değer* **block-time** *dakika*	Eğer aynı hedeften sürekli paket istenirse buna dikkat edilmeli	50 oturum/0 dk

5.7.3.4.Yarım-açık Oturumlar:

Yüksek miktarda Yarım-açık oturum değeri genelde, DoS saldırılarına imkan verebilir. TCP için Yarım-açık oturum manası, oturum isteği yerine ulaşmadı, UDP için ise "Firewall geri dönüşü olmayan bir trafik sezdi" manasındadır. TCP ve UDP yarım-açık oturumlarının her ikisi de hız miktarını ve bir sayıcı ile bizim atamış olduğumuz değişken değerine kadar sayar. Ne zaman ki sayıcı bu değere ulaşırsa Yazılım yeni bağlantı istekleri için bu yarım-açık oturumu siler.

5.7.4.Uygulama Katmanı Protokolleri Denetim Konfigürasyonu:

Bir uygulama katmanı protokolü için CBAC denetimini belli bir süreliğine devre dışı bırakma konfigürasyonu, aşağıda gösterilen global konfigürasyon komutları ile olur.

ip inspect name *protokol_adı* ***[timeout*** *saniye**]*** yada
ip inspect name *denetim_adı* ***rpc program-number*** *değer* ***[wait-time*** *dakika**][timeout*** *saniye**]***

5.7.4.1.Java Denetimi Konfigürasyonu:

Java aplet'lerini bloke etmek için aşağıdaki komutlar kullanılır.

1. ***ip access-list standard*** *adı* ***permit.....deny.....*** yada

access-list *erişim liste no* ***{deny/permit} source[source-****wildcard**]***

2. ***ip inspect name*** *denetim_adı* ***http[java-list*** *erişim listesi**][timeout*** *saniye**]***

5.7.4.2.CBAC için Konfigürasyon İstatistiklerini ve Durumunu Görüntüleme:

Komut	İşlevi
Show ip inspect name *denetim_adı*	denetim kuralları konfigürasyonunu gösterir.
Show ip inspect config	Bütün CBAC denetim konfigürasyonunu gösterir.
Show ip inspect interfaces	Erişim listesi ile beraber arayüz konf. gösterir.
Show ip inspect all	Bütün CBAC konfigürasyonu ve oturumları gösterir.

5.8.DEBUG KOMUTLARI:

KOMUT	İŞLEVİ
Debug ip inspect function trace	yazılım fonksiyonları hakkında CBAC dan bilgi gösterir.
Debug ip inspect *object-creation*	CBAC ın oluşturduğu yazılım objeleri hakkında bilgi verir.
Debug ip inspect object-deletion	CBAC in sildiği yazılım objeleri hakkında bilgi verir.
Debug ip inspect events	CBAC paket işlemleri bilgileri hakkında bilgi verir.

5.8.1.TCP ve UDP Oturumları Debug Komutları:

Debug ip inspect tcp	TCP paketinin içeriği ile ilgili bilgiler verir.
Debug ip inspect udp	UDP paketinin içeriği ile ilgili bilgiler verir.

5.8.2.Uygulama Katmanı Debug Komutları:

Debug ip inspect *protokol* CBAC hakkında istenilen protokol bilgisi verilir.

Aşağıdaki tabloda ***Debug ip inspect*** komutundan sonra kullanılacak protokollerin kodlanma şekli gösterilmiştir.

CU-SeeMe	**cuseeme**
FTP commands and responses	**ftp-cmd**
FTP tokens	**ftp-tokens**
H.323	**h323**
Java applets	**http**
UNIX R commands (rlogin, rexec, rsh)	**rcmd**
RealAudio	**realaudio**
RPC	**rpc**
SMTP	**smtp**
SQL*Net	**sqlnet**
StreamWorks	**streamworks**
TFTP	**tftp**
VDOLive	**vdolive**

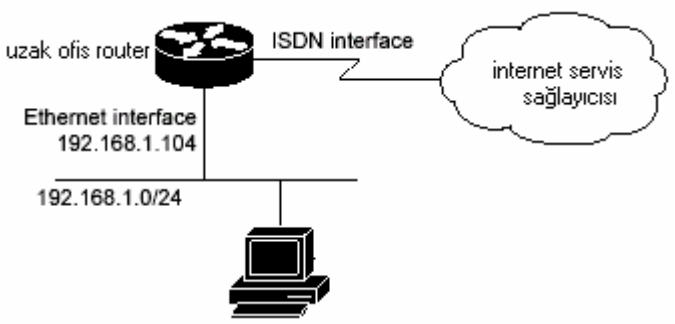
5.9.FİREWALL KONFİGÜRASYON ÖRNEKLERİ:

5.9.1. CBAC Konfigürasyonu Örneği:

```
!-------------------------
!Create the Inspection Rule
!-------------------------
!
!Create the CBAC inspection rule "test", allowing inspection of the protocol traffic
!specified by the rule. This inspection rule sets the timeout value to 30 seconds for
!each protocol (except for RPC). The timeout value defines the maximum time that a
!connection for a given protocol can remain active without any traffic passing through
!the router. When these timeouts are reached, the dynamic ACLs that are inserted to
!permit the returning traffic are removed, and subsequent packets (possibly even valid
!ones) are not permitted.
ip inspect name test cuseeme timeout 30
ip inspect name test ftp timeout 30
ip inspect name test h323 timeout 30
ip inspect name test realaudio timeout 30
ip inspect name test rpc program-number 100000
ip inspect name test streamworks timeout 30
ip inspect name test vdolive timeout 30
!
!-----------------------------
!Create the Access Control List
!-----------------------------
!
!In this example, ACL 105 denies all TCP and UDP protocol traffic. IP traffic is
!permitted to allow access for routing and control traffic. This means that only the
!return traffic for protocols defined in the inspection rule is allow access through
!the interface where this rule is applied.
access-list 105 deny TCP any any
access-list 105 deny UDP any any
access-list 105 permit ip any any
!

!---------------------------------
!Apply the Inspection Rule and ACL
!---------------------------------
!
!In this example, the inspection rule "test" is applied at ATM interface 3/0 for
!connections initiated in the outbound direction; that is, from hosts that are located
!on a local network. ACL 105 is applied at ATM interface 3/0 in the inbound direction;
!that is, return traffic in response to local host initiated connections. If inbound
!traffic at the interface matches an inspection rule, CBAC creates a dynamic access
!list allowing inbound (returning) traffic for that connection. This combination of the
!ACL and CBAC inspection rules means that TCP and UDP traffic that is not part of a
!connection that initiated from a local host is not permitted access through the
!interface.
interface ATM3/0
 ip address 10.1.10.1 255.0.0.0
 ip access-group 105 in
 no ip directed-broadcast
 ip nat outside
 ip inspect test out
 no shutdown
 atm clock INTERNAL
 atm pvc 7 7 7 aal5snap
 map-group atm
```

5.9.2.Uzak Ofisten ISP(internet servis sağlayıcısı) Bağlantıda:



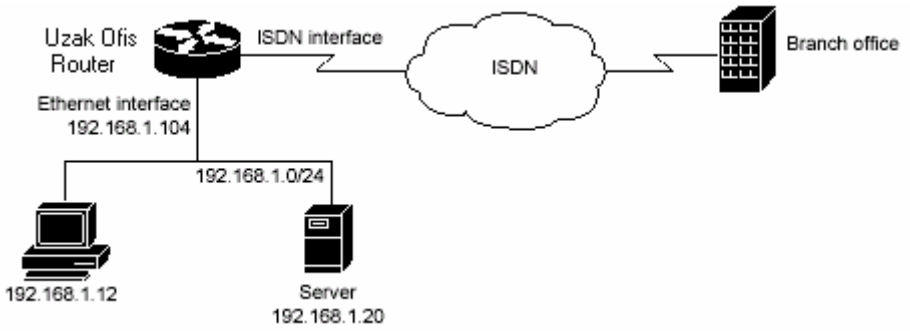
ISP, ISDN arayüzünü bloke etmiştir.

```
!-------------------------------------
!General Cisco IOS Firewall Guidelines
!-------------------------------------
!The following global configuration entries illustrate good security practices.
enable secret 5 <elided>
no ip source-route
no cdp run
!
!---------------------------------
!Create the CBAC inspection rule
!---------------------------------
!Create the CBAC inspection rule STOP to allow inspection of the protocol traffic
!specified by the rule.
ip inspect name STOP tcp
ip inspect name STOP ftp
ip inspect name STOP smtp
ip inspect name STOP h323
ip inspect name STOP rcmd
!
!---------------------------------
!Create Access Control List 105
!---------------------------------
!ACL 105 denies all IP protocol traffic except for specific ICMP control traffic.
!This means that only the return traffic for protocols defined in the
!inspection rule and the specified ICMP traffic is allowed access through the
!interface where this rule is applied.
!
!Deny broadcast messages with a source address of 255.255.255.255; this helps to
!prevent broadcast attacks.
access-list 105 deny ip host 255.255.255.255 any
!
!Add anti-spoofing protection by denying traffic with a source address matching a host
!on the Ethernet interface.
acl 105 deny ip 192.168.1.0 0.0.0.255 any
!
!ICMP traffic is not inspected by CBAC. To control the type of ICMP traffic at the
!interface, add static access list entries. This example has the following ICMP
!requirements: outgoing ping commands require echo-reply messages to come back,
!outgoing traceroute commands require time-exceeded messages to come back, path MTU
!discovery requires "too-big" messages to come back, and incoming traceroute
!messages must be allowed. Additionally, permit all "unreachable" messages to come
!back; that is, if a router cannot forward or deliver a datagram, it sends an ICMP
!unreachable
!message back to the source and drops the datagram.
```

```
access-list 105 permit icmp any any echo-reply
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 time-exceeded
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 packet-too-big
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 traceroute
access-list 105 permit icmp any 192.168.1.0 0.0.0.255 unreachable
!
!Final deny for explicitness. This entry is not required but helps complete the access
!list picture. By default, the final entry in any access list is an implicit deny of IP
!protocol traffic. This ensures that the firewall blocks any traffic not explicitly
!permitted by the access list.
access-list 105 deny ip any any
!
!------------------------------------------------------------
!Configure the interface
!------------------------------------------------------------
!In this example, no ACLs or inspection rules are applied at interface Ethernet0,
!meaning that all traffic on the local network is allowed to go out. This assumes a
!high-level of trust for the users on the local network.
interface Ethernet0
 ip address 192.168.1.104 255.255.255.0
 no ip directed-broadcast
!
!This example uses a dialer profile, so the ACL and CBAC inspection rules are applied
!at the dialer interface, not the physical BRI interface. The dialer pool-member
!command is used to associate the physical interface with a dialer profile.
interface BRI0
 no ip address
 no ip directed-broadcast
 encapsulation ppp

 dialer pool-member 1
 isdn switch-type basic-5ess
!
!------------------------------------------------------------------------
!Create the dialer profile.
!------------------------------------------------------------------------
!Through the dialer profile, the ACL and CBAC inspection rules are
!applied to every pool member. In this example, the ACL is applied in, meaning that it
!applies to traffic inbound from the ISP. The CBAC inspection rule STOP is applied out,
!meaning that CBAC monitors the traffic through the interface and controls return
!traffic to the router for an existing connection.
interface Dialer0
 ip address negotiated
 ip access-group 105 in
 no ip directed-broadcast
 ip inspect STOP out
 encapsulation ppp
 dialer remote-name <ISP router>
 dialer idle-timeout 500
 dialer string <elided>
 dialer pool 1
 dialer-group 1
 ppp authentication callin
 !
!------------------------------------------------------------
!Additional entries
!------------------------------------------------------------
!Configure the router to forward packets destined for an unrecognized subnet of
!a directly connected network.
ip classless
!Route traffic to the dialer interface.
ip route 0.0.0.0 0.0.0.0 Dialer0
!Include a dialer list protocol entry to specify the protocol that triggers dialing.
dialer-list 1 protocol ip permit
!Add a user name (name of the router your are configuring) and password for caller
!identification and password authentication with the ISP router.
username <router host name> password 5 <elided>
```

5.9.3.Uzak Ofisten Branch Ofise Konfigürasyon:




```
!-----------------------------------------
!General firewall configuration guidelines
!-----------------------------------------
!The following global configuration entries illustrate good security practices.
enable secret 5 <elided>
no ip source-route
no cdp run
!
!---------------------------
!Create the Inspection Rule
!---------------------------
!Create the CBAC inspection rule STOP to allow inspection of the specified protocol
!traffic. Create the inspection rule GO to allow inspection of HTTP and SMTP
!traffic. Note that Java applets will be permitted according to access list 51, which
!is defined later in this sample configuration.
ip inspect name STOP tcp
ip inspect name STOP ftp
ip inspect name STOP smtp
ip inspect name STOP h323
ip inspect name STOP rcmd
ip inspect name GO http java-list 51
ip inspect name GO smtp
!
!-------------------------------------------------------------
!Create Access Control Lists 106 and 51
!-------------------------------------------------------------
!ACL 106 permits mail and Web traffic from any host to the specified server. ACL 106
!denies all other ip protocol traffic except for specific ICMP control traffic.
!This means that only the return traffic for protocols defined in the
!inspection rule and the specified ICMP traffic is allowed access through the
!interface where this rule is applied.
!
!Deny broadcast messages with a source address of 255.255.255.255; this helps to
!prevent broadcast attacks.
access-list 106 deny ip host 255.255.255.255 any
```

```
!
!Add anti-spoofing protection by denying traffic with a source address matching a host
!on the Ethernet interface.
access-list 106 deny ip 192.168.1.0 0.0.0.255 any
!
!ICMP traffic is not inspected by CBAC. To control the type of ICMP traffic at the
!interface, add static access list entries. This example has the following ICMP
!requirements: outgoing ping commands require echo-reply messages to come back,
!outgoing traceroute commands require time-exceeded messages to come back, path MTU
!discovery requires "too-big" messages to come back, and incoming traceroute must be
!allowed. Additionally, permit all "unreachable" messages to come back; that is, if a
!router cannot forward or deliver a datagram, it sends an ICMP unreachable message back
!to the source and drops the datagram.
access-list 106 permit icmp any any echo-reply
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 time-exceeded
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 packet-too-big
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 traceroute
access-list 106 permit icmp any 192.168.1.0 0.0.0.255 unreachable
!
!Permit mail and Web access to a specific server.
access-list 106 permit tcp any host 192.168.1.20 eq smtp
access-list 106 permit tcp any host 192.168.1.20 eq www
!


!Final deny for explicitness. This entry is not required but helps complete the access
!list picture. By default, the final entry in any access list is an implicit deny of IP
!protocol traffic. This ensures that the firewall blocks any traffic not explicitly
!permitted by the access list.
access-list 106 deny ip any any
!
!Access list 51 defines the sites for Java applet blocking. If the access list denies a
!site, that site is deemed "hostile" and applets from that site are blocked. If the
!access list permits a site, that site is deemed "friendly" and applets from that
!site are not blocked. Java applet blocking is defined in the inspection rule "GO",
!meaning applets are permitted or denied from the sites defined in the access list. In
!this example, access list 51 permits Java applets from any site (source address).
access-list 51 permit any
!
!-----------------------------------------------------------
!Configure the interface.
!-----------------------------------------------------------
!In this example, no ACLs or inspection rules are applied at interface Ethernet0,
!meaning that all traffic on the local network is allowed to go out. This assumes a
!high-level of trust for the users on the local network.
interface Ethernet0
 ip address 192.168.1.104 255.255.255.0
 no ip directed-broadcast
!
!This example uses a dialer profile, so the ACL and CBAC inspection rules are applied
!at the dialer interface, not the physical BRI interface. The dialer pool-member
!command is used to associate the physical interface with a dialer profile.
interface BRI0
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 dialer pool-member 1
 isdn switch-type basic-5ess
!
```

```
!--------------------------------------------------------------------
!Apply the ACL and CBAC inspection rules at the dialer interface.
!--------------------------------------------------------------------
!Through the dialer profile, the ACL and CBAC inspection rules are
!applied to every pool member. In this example, the ACL is applied in, meaning that it
!applies to traffic inbound from the branch office. The CBAC inspection rule STOP is
!applied out, meaning that CBAC monitors the traffic and controls return traffic to the
!router for an existing connection. The CBAC inspection rule GO is applied in,
!protecting against certain types of DoS attacks as described in this document. Note
!that the GO inspection rule does not control return traffic because there is no ACL
!blocking traffic in that direction; however, it does monitor the connections.
interface Dialer0
 ip address <ISDN interface address>
 ip access-group 106 in
 no ip directed-broadcast
 ip inspect STOP out
 ip inspect GO in
 encapsulation ppp
 dialer remote-name <branch office router>
 dialer idle-timeout 500
 dialer string <elided>
 dialer pool 1
 dialer-group 1
 ppp authentication
!
!----------------------------------------------------------
Additional entries
!----------------------------------------------------------
!Configure the router to forward packets destined for an unrecognized subnet of
!a directly connected network.
ip classless
!Route traffic to the dialer interface.
ip route 0.0.0.0 0.0.0.0 Dialer0
!Include a dialer list protocol entry to specify the protocol that triggers dialing.
dialer-list 1 protocol ip permit
!Add a user name (name of the router your are configuring) and password for caller
!identification and password authentication with the ISP router.
username <router host name> password 5 <elided>
```

5.9.4.İki Arayüz Branch Ofisin Konfigürasyonu:

Yukarıdaki örneğe ek olarak;

Firewall'un iki arayüzü konfigüre edildi.

- Ethernet0 arayüzü Dahili korumalı networke bağlıdır.
- Serial0 arayüzü Frame Relay ile WAN' a bağlıdır.

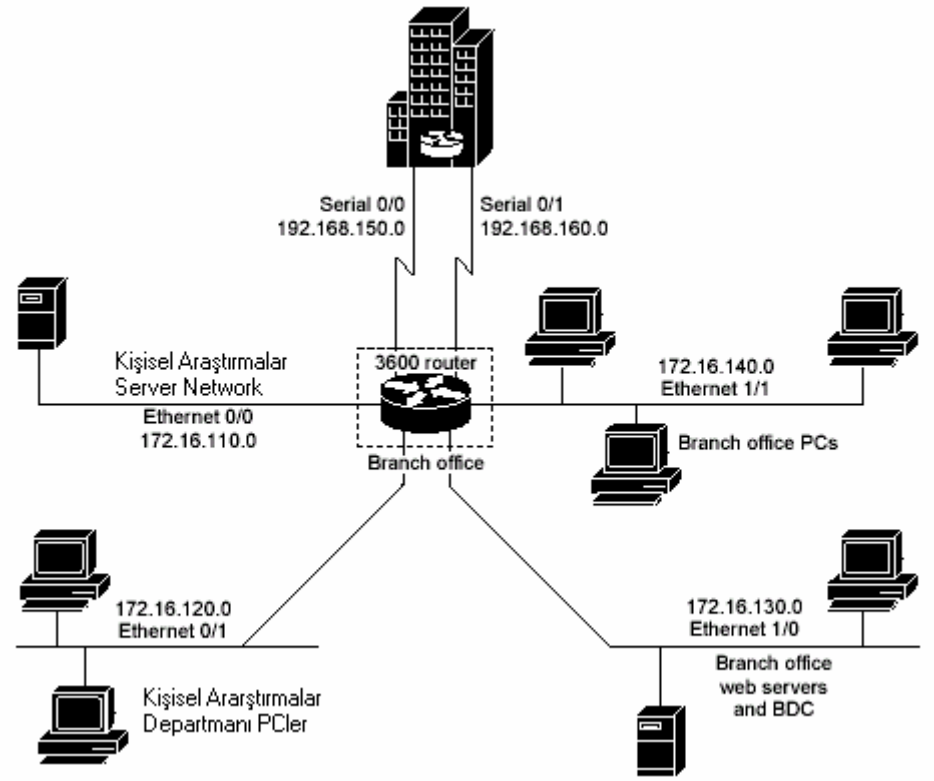
```
!------------------------------------------------------------------------
! This first section contains some configuration that is not required for CBAC,
! but illustrates good security practices. Note that there are no
! services on the Ethernet side. Email is picked up via POP from a server on the
! corporate side.
!------------------------------------------------------------------------
!
hostname user1-examplecorp-fr
!
boot system flash c1600-fw1600-l
enable secret 5 <elided>
!
username user1 password <elided>
ip subnet-zero
no ip source-route
ip domain-name example.com
ip name-server 172.19.2.132
ip name-server 198.92.30.32
!
!
!------------------------------------------------------------------------
!The next section includes configuration required specifically for CBAC
!------------------------------------------------------------------------
!
!The following commands define the inspection rule "myfw", allowing
!the specified protocols to be inspected. Note that Java applets will be permitted
!according to access list 51, defined later in this configuration.
ip inspect name myfw cuseeme timeout 3600
ip inspect name myfw ftp timeout 3600
ip inspect name myfw http java-list 51 timeout 3600
ip inspect name myfw rcmd timeout 3600
ip inspect name myfw realaudio timeout 3600
ip inspect name myfw smtp timeout 3600
ip inspect name myfw tftp timeout 30
ip inspect name myfw udp timeout 15
ip inspect name myfw tcp timeout 3600
!
!The following interface configuration applies the "myfw" inspection rule to
!inbound traffic at Ethernet 0. Since this interface is on the internal network
!side of the firewall, traffic entering Ethernet 0 is actually
!exiting the internal network. Applying the inspection rule to this interface causes
!inbound traffic (which is exiting the network) to be inspected; return traffic will
!only be permitted back through the firewall if part of a session which began from
!within the network.
!Also note that access list 101 is applied to inbound traffic at Ethernet 0.
!Any traffic that passes the access list will be inspected by CBAC.
!(Traffic blocked by the access list will not be inspected.)
!interface Ethernet0
 description ExampleCorp Ethernet chez user1
 ip address 172.19.139.1 255.255.255.248
 ip broadcast-address 172.19.131.7
 no ip directed-broadcast
 no ip proxy-arp
 ip inspect myfw in
 ip access-group 101 in
 no cdp enable
!
intertace Serial0
 description Frame Relay (Telco ID 22RTQQ062438-001) to ExampleCorp HQ
 no ip address
 ip broadcast-address 0.0.0.0
 encapsulation frame-relay IETF
 no arp frame-relay
 bandwidth 56
 service-module 56k clock source line
 service-module 56k network-type dds
 frame-relay lmi-type ansi
```

```
!
!Note that the following interface configuration applies access list 111 to
!inbound traffic at the external serial interface. (Inbound traffic is
!entering the network.) When CBAC inspection occurs on traffic exiting the
!network, temporary openings will be added to access list 111 to allow returning
!traffic that is part of existing sessions.
!
interface Serial0.1 point-to-point
 ip unnumbered Ethernet0
 ip access-group 111 in
 bandwidth 56
 no cdp enable
 frame-relay interface-dlci 16
!
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0.1
!
!The following access list defines "friendly" and "hostile" sites for Java
!applet blocking. Because Java applet blocking is defined in the inspection
!rule "myfw" and references access list 51, applets will be actively denied
!if they are from any of the "deny" addresses and allowed only if they are from
!either of the two "permit" networks.
!
access-list 51 deny   172.19.1.203
access-list 51 deny   172.19.2.147
access-list 51 permit 172.18.0.0 0.1.255.255
access-list 51 permit 192.168.1.0 0.0.0.255
access-list 51 deny   any
!
!The following access list 101 is applied to interface Ethernet 0 above.
!This access list permits all traffic that should be CBAC inspected, and also
!provides anti-spoofing. The access list is deliberately set up to deny unknown
!IP protocols, because no such unknown protocols will be in legitimate use.
!
access-list 101 permit tcp 172.19.139.0 0.0.0.7 any
access-list 101 permit udp 172.19.139.0 0.0.0.7 any
access-list 101 permit icmp 172.19.139.0 0.0.0.7 any
access-list 101 deny   ip any any

!The following access list 111 is applied to interface Serial 0.1 above.
!This access list filters traffic coming in from the external side. When
!CBAC inspection occurs, temporary openings will be added to the beginning of
!this access list to allow return traffic back into the internal network.
!This access list should restrict traffic that will be inspected by
!CBAC. (Remember that CBAC will open holes as necessary to permit returning traffic.)
!Comments precede each access list entry. These entries are not all specifically
!related to CBAC, but are created to provide general good security.
!
!Anti-spoofing.
access-list 111 deny   ip 172.19.139.0 0.0.0.7 any
!Sometimes EIGRP is run on the Frame Relay link. When you use an
!input access list, you have to explicitly allow even control traffic.
!This could be more restrictive, but there would have to be entries
!for the EIGRP multicast as well as for the office's own unicast address.
access-list 111 permit igrp any any
!
!These are the ICMP types actually used...
!administratively-prohibited is useful when you are trying to figure out why
!you cannot reach something you think you should be able to reach.
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 administratively-prohibited
!
!This allows network admins at headquarters to ping hosts at the field office:
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 echo
!
!This allows the field office to do outgoing pings
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 echo-reply
!
!Path MTU discovery requires too-big messages
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 packet-too-big
!
!Outgoing traceroute requires time-exceeded messages to come back
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 time-exceeded
!
```

```
! Incoming traceroute
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 traceroute
!
!Permits all unreachables because if you are trying to debug
!things from the remote office, you want to see them. If nobody ever did
!any debugging from the network, it would be more appropriate to permit only
!port unreachables or no unreachables at all.
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 unreachable
!
!These next two entries permit users on most ExampleCorp networks to Telnet to
!a host in the field office. This is for remote administration by the network admins.
access-list 111 permit tcp 172.18.0.0 0.1.255.255 host 172.19.139.1 eq telnet
access-list 111 permit tcp 192.168.1.0 0.0.0.255 host 172.19.139.1 eq telnet
!
!Final deny for explicitness
access-list 111 deny   ip any any
!
no cdp run
snmp-server community <elided> RO
!
line con 0
 exec-timeout 0 0
 password <elided>
 login local
line vty 0
exec-timeout 0 0
password <elided>
login local
length 35
line vty 1
 exec-timeout 0 0
 password 7 <elided>
 login local
line vty 2
 exec-timeout 0 0
 password 7 <elided>
 login local
line vty 3
 exec-timeout 0 0
 password 7 <elided>
 login local
line vty 4
  exec-timeout 0 0
  password 7 <elided>
  login local
 !
 scheduler interval 500
 end
```

5.9.5.Çoklu Arayüzlü Branch Ofis Konfigürasyonu:




```
! ------------------------------------------------------------------
! This first section contains some configuration that is not required
! for CBAC, but illustrates good security practices.
! ------------------------------------------------------------------
!Add this line to get timestamps on the syslog messages.
service timestamps log datetime localtime show-timezone
!
hostname Router1
!
boot system flash c3600-fw3600-l
!
! Configure AAA user authentication.
aaa new-model
aaa authentication login lista tacacs+ enable
!
enable secret 5 <elided>
ip subnet-zero
!
! Disable source routing to help prevent spoofing.
no ip source-route
!
! Set up the domain name and server IP addresses.
ip domain-name example.com
ip name-server 192.168.55.132
ip name-server 192.168.27.32
!
```

```
! The audit-trail command enables the delivery of specific CBAC messages
! through the syslog notification process.
ip inspect audit-trail
!
! Establish the time-out values for DNS queries. When this idle-timer expires,
! the dynamic ACL entries that were created to permit the reply to a DNS request
! will be removed and any subsequent packets will be denied.
ip inspect dns-timeout 10
!
!--------------------------------------------------------------------
!The next section includes configuration statements required
!specifically for CBAC.
!--------------------------------------------------------------------
! Define the CBAC inspection rule "inspect1", allowing the specified protocols to be
! inspected. The first rule enables SMTP specific inspection. SMTP inspection causes
! the exchange of the SMTP session to be inspected for illegal commands. Any packets
! with illegal commands are dropped, and the SMTP session will hang and eventually
! time out.
ip inspect name inspect1 smtp timeout 300
!
! In the next two lines of inspect1, define the maximum time that each of the UDP and
! TCP sessions are allowed to continue without any traffic passing
! through the router. When these timeouts are reached, the dynamic ACLs that
! are inserted to permit the returning traffic are removed and subsequent packets
! (possibly even valid ones) will not be permitted.
ip inspect name inspect1 udp timeout 300
ip inspect name inspect1 tcp timeout 300
!
! Define the CBAC inspection rule "inspect2", allowing the specified protocols to be
! inspected. These rules are similar to those used in the inspection rule "inspect1,"
! except that on the interfaces where this rule is applied, SMTP sessions are not
! expected to go through; therefore, the SMTP rule element is not applied here.
ip inspect name inspect2 udp timeout 300
ip inspect name inspect2 tcp timeout 3600
!
!--------------------------------------------------------------------
! The next section shows the Ethernet interface configuration statements for each
! interface, including access lists and inspections rules.
!--------------------------------------------------------------------
! Apply the "inspect1" inspection rule to sessions that are initiated in the outbound
! direction (toward the LAN) at Ethernet interface 0/0. All packets in these sessions
! will be inspected by CBAC. Provided that network traffic passes the Access Control
! List (ACL) restrictions, traffic is then inspected by CBAC for access through the
! Cisco IOS Firewall. Traffic blocked by the access list is not inspected by CBAC.
! Access list 110 is applied to outbound traffic on this interface.
```

```
interface Ethernet0/0
description HR_Server Ethernet
ip address 172.16.110.1 255.255.255.0
ip access-group 110 out
no ip directed-broadcast
no ip proxy-arp
ip inspect inspect1 out
no cdp enable
!
! Apply access list 120 to inbound traffic on Ethernet interface 0/1.
! Applying access list 120 to inbound traffic provides anti-spoofing on this interface
! by dropping traffic with a source address matching the IP address on a network other
! than Ethernet 0/1. The IP helper address lists the IP address of the DHCP server on
! Ethernet interface 1/0.
interface Ethernet0/1
description HR_client Ethernet
ip address 172.16.120.1 255.255.255.0
ip access-group 120 in
ip helper-address 172.16.130.66
no ip directed-broadcast
```

```
no ip proxy-arp
no cdp enable
!
! Apply the "inspect2" inspection rule to sessions that are initiated in the outbound
! direction (toward the LAN) at Ethernet interface 1/0. Provided that network traffic
! passes the Access Control List (ACL) restrictions, traffic is then inspected by CBAC
! through the Cisco IOS Firewall. Traffic blocked by the access list is not inspected
! by CBAC. Access list 130 is applied to outbound traffic on this interface.
interface Ethernet1/0
description Web_server Ethernet
ip address 172.16.130.1 255.255.255.0
ip access-group 130 out
no ip directed-broadcast
no ip proxy-arp
ip inspect inspect2 out
no cdp enable
!
! Apply access list 140 to inbound traffic at Ethernet interface 1/1. This
! provides anti-spoofing on the interface by dropping traffic with a source address
! matching the IP address of a network other than Ethernet 1/1. The IP helper address
! lists the IP address of the DHCP server on Ethernet interface 1/0.
interface Ethernet1/1
description Everyone_else Ethernet
ip address 172.16.140.1 255.255.255.0
ip access-group 140 in
ip helper-address 172.16.130.66
no ip directed-broadcast
no ip proxy-arp
no cdp enable
!
!--------------------------------------------------------------------
! The next section configures the serial interfaces, including access lists.
!--------------------------------------------------------------------
! Apply access list 150 to Serial interfaces 0/0. This provides anti-spoofing on the
! serial interface by dropping traffic with a source address matching the IP address
! of a host on Ethernet interface 0/0, 0/1, 1/0, or 1/1.
interface Serial0/0
description T1 to HQ
ip address 192.168.150.1 255.255.255.0
ip access-group 150 in
bandwidth 1544
!
interface Serial1/1
description T1 to HQ
ip address 192.168.160.1 255.255.255.0
ip access-group 150 in
bandwidth 1544
!
! ------------------------------
! Configure routing information.
! ------------------------------
router igrp 109
network 172.16.0.0
network 192.168.150.0
network 192.168.160.0
!
! Define protocol forwarding on the firewall. When you turn on a related command,
! ip helper-address, you forward every IP broadcast in the ip forward protocol
! command list, including several which are on by default: TFTP (port 69),
! DNS (port 53), Time service (port 37), NetBIOS Name Server (port 137),
! NetBIOS Datagram Server (port 138), BOOTP client and server datagrams
! (ports 67 and 68), and TACACS service (port 49). One common
! application that requires helper addresses is Dynamic Host Configuration
! Protocol (DHCP). DHCP information is carried inside of BOOTP packets. The
! "no ip forward protocol" statements turn off forwarding for the specified protocols.
```

```
no ip forward-protocol udp netbios-ns
no ip forward-protocol udp netbios-dgm
no ip forward-protocol udp tacacs
no ip forward-protocol udp tftp
ip forward-protocol udp bootpc
!
! Add this line to establish where router SYSLOG messages are sent. This includes the
! CBAC messages.
logging 192.168.55.131
!
! ----------------------------------------------------------------
! Define the configuration of each access list.
! ----------------------------------------------------------------
! Defines Telnet controls in access list 12.
access-list 12 permit 192.168.55.0 0.0.0.255
!
! Defines SNMP controls in access list 13.
access-list 13 permit 192.168.55.12
access-list 13 permit 192.168.55.19
!
! Access list 110 permits TCP and UDP protocol traffic for
! specific ports and with a source address on Ethernet interface 0/1. The access list
! denies IP protocol traffic with any other source and destination address. The
! access list permits ICMP access for any source and destination
! address. Access list 110 is deliberately set up to deny unknown IP protocols
! because no such unknown protocols will be in legitimate use. Access list
! 110 is applied to outbound traffic at Ethernet interface 0/0. In ACL 110,
! network traffic is being allowed access to the ports on any server on the HR server
! network. In less trusted environments, this can be a security problem; however, you
! can limit access more severely by specifying specific destination addresses in the
! ACL statements.
access-list 110 permit tcp 172.16.120.0 0.0.0.255 any eq smtp
access-list 110 permit tcp 172.16.120.0 0.0.0.255 any eq pop3
access-list 110 permit tcp 172.16.120.0 0.0.0.255 any eq 110
access-list 110 permit udp any any eq 137
access-list 110 permit udp any any eq 138
access-list 110 permit udp any any eq 139
access-list 110 permit icmp any any
access-list 110 deny ip any any!
!
! Access-list 120 permits TCP, UDP, and ICMP protocol traffic with a source address
! on Ethernet interface 0/1, but denies all other IP protocol traffic. Access list
! 120 is applied to inbound traffic on Ethernet interface 0/1.
access-list 120 permit tcp 172.16.120.0 0.0.0.255 any
access-list 120 permit udp 172.16.120.0 0.0.0.255 any
access-list 120 permit icmp 172.16.120.0 0.0.0.255 any
access-list 120 deny ip any any
!
! Access list 130 permits TCP, UDP, and ICMP protocol traffic for specific ports and
! with any source and destination address. It opens access to the web server and to
! all NBT services to the rest of the company, which can be controlled through the
! trust relations on the Windows NT servers. The bootpc entry permits access to the
! DHCP server. Access list 130 denies all other IP protocol traffic. Access list 130 is
! applied to outbound traffic at Ethernet interface 1/0.
access-list 130 permit tcp any any eq www
access-list 130 permit tcp any any eq 443
access-list 130 permit tcp any any eq 110
access-list 130 permit udp any any eq 137
access-list 130 permit udp any any eq 138
access-list 130 permit udp any any eq 139
access-list 130 permit udp any any eq bootpc
access-list 130 permit icmp any any
access-list 130 deny ip any any
!
! Access list 140 permits TCP, UDP, and ICMP protocol traffic with a source address on
```

```
! Ethernet interface 1/1, and it denies all other IP protocol traffic. Access list 140
! is applied to inbound traffic at Ethernet interface 1/1.
access-list 140 permit tcp 172.16.140.0 0.0.0.255 any
access-list 140 permit udp 172.16.140.0 0.0.0.255 any
access-list 140 permit icmp 172.16.140.0 0.0.0.255 any
access-list 140 deny ip any any
!
! Access list 150 denies IP protocol traffic with a source address on Ethernet
! interfaces 0/0, 0/1, 1/0, and 1/1, and it permits IP protocol traffic with any other
! source and destination address. Access list 150 is applied to inbound traffic
! on each of the serial interfaces.
access-list 150 deny ip 172.16.110.0 0.0.0.255 any
access-list 150 deny ip 172.16.120.0 0.0.0.255 any
access-list 150 deny ip 172.16.130.0 0.0.0.255 any
access-list 150 deny ip 172.16.140.0 0.0.0.255 any
access-list 150 permit ip any any
!
! Disable Cisco Discovery Protocol.
no cdp run
!
snmp-server community <elided> ro 13
tacacs-server host 192.168.55.2
tacacs-server key <elided>
!
! ----------------------------------------------------------------------------
! Configures the router console port and the virtual terminal line interfaces,
! including AAA authentication at login. Authentication is required for users defined
! in "lista." Access-class 12 is applied on each line, restricting Telnet access to
! connections with a source address on the network management network.
! ----------------------------------------------------------------------------
line console 0
exec-timeout 3 00
login authentication lista
line aux 0
exec-timeout 3 00
login authentication lista
line vty 0
exec-timeout 1 30
login authentication lista
access-class 12 in
line vty 1
exec-timeout 1 30
login authentication lista
access-class 12 in
line vty 2
exec-timeout 1 30
login authentication lista
access-class 12 in
line vty 3
exec-timeout 1 30
login authentication lista
access-class 12 in
line vty 4
exec-timeout 1 30
login authentication lista
access-class 12 in
!
end
```