UNIX SİSTEMLERDE AĞ ANALİZİ YAPABİLEN BİR UYGULAMA YAZILMASI

Yüksek Lisans Tezi

Mesut AKTOGAN

Danışman: Yrd.Doç.Dr. Erdem UÇAR

Edirne - 2006

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

UNIX SİSTEMLERDE AĞ ANALİZİ YAPABİLEN BİR UYGULAMA YAZILMASI

Mesut AKTOGAN

Yüksek Lisans Tezi
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman: Yrd.Doç.Dr. Erdem UÇAR

EDİRNE - 2006

T.C
TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

UNIX SİSTEMLERDE AĞ ANALİZİ YAPABİLEN BİR UYGULAMA YAZILMASI

Mesut AKTOGAN

Yüksek Lisans Tezi
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Bu tez 25.08.2006 tarihinde Aşağıdaki Jüri Tarafından Kabul Edilmiştir.

Yrd.Doç.Dr. Erdem UÇAR Yrd.Doç.Dr. Yılmaz KILIÇASLAN Yrd.Doç.Dr. Tahir ALTINBALIK

Yüksek Lisans Tezi
Trakya Üniversitesi Fen Bilimleri Enstitüsü

PAKET ANALİZİ YAPAN BİR UYGULAMA GELİŞTİRİLMESİ

ÖZET

Bu çalışmanın amacı, ağdaki trafiğin analizini yaparak gidip-gelen paketlerin yakalanması ve bu paketlerin açılarak içinden istediğimiz verilerin alınıp bir veritabanına yazılmasıdır. Bunun sonucunda veritabanından istediğimiz şekilde sorgu yaparak istediğimiz bilgiyi elde edebiliriz. Mesela istediğimiz saatler arasında internette olan kişileri veya internetten belli bir miktarın üzerinde veri çeken kullanıcıları listeletebiliriz.

Çalışma, temel olarak 2 ana parçadan oluşmaktadır. Birincisi, paketlerin yakalanma kısmıdır. Bu amaçla Libpcap (paket yakalama kütüphanesi) uygulaması araştırılmış, bu uygulamada nasıl kullanılacağı belirlenmiştir. İkincisi ise, yakalanan paketlerden isteğimiz verilerin alınarak bunların veritabanına yazılması. Bu amaçla, bilgisayar bilimlerinin bu alanı üzerinde de çalışma yapılıp uygulaması gerçekleştirilmiştir.

Anahtar Kelimeler: TCP/IP, İnternet Protokolleri, Libpcap, Sniffer, Ağ, UNIX

Yıl: 2005
Sayfa: 60

Master Thesis
Trakya University Graduate School of
Natural and Applied Sciences
Departman of Computer Engineering

DEVELOPING AN APPLICATION THAT MAKES ANALYSE PACKETS

SUMMARY

The aim of this study is to develop an application that analyse network traffic, capture packets on network, open it for take information which we need and wrote them on a database. So we can executes query which we want from database. For example we can list who is using net in a time which we define or who is downloading from net over than we define.

The study is comprises two parts, the part is capture packets. For this purpose, search Libpcap (library of capture packets) apply, ascertain how to use in this application. The second part of the study is take information which we need from captured packets, wrote them on a database. Therefore, this field of computer sciences has also been studied and its application has been realized.

Anahtar Kelimeler: TCP/IP, İnternet Protocols, Libpcap, Sniffer, Network, UNIX

Year: 2005

Page: 60

TEŞEKKÜRLER

Bu çalışmanın gerçekleşmesinde, değerli katkılarından dolayı tez danışmanım Yrd.Doç.Dr. Erdem UÇAR'a Trakya Üniversitesi Bilgisayar Mühendisliği Bölümü Öğretim üyesi sayın Yılmaz KILIÇASLAN'a teşekkür ederim.

Ayrıca çalışma süresince desteklerini esirgemeyen Trakya Üniversitesi Bilgisayar Mühendisliği Bölümündeki bütün araştırma görevlilerine ayrı ayrı teşekkür ederim.

Son olarak da, çalışma süresince her zaman yanımda olan aileme de teşekkür ederim.

ŞEKİLLER LİSTESİ

1. GİRİŞ VE AMAÇ

Birden fazla bilgisayarı, birbirine bağlayarak aralarında bir paylaşım kurmak masraflı bir iştir. Paylaşım, bir bilgisayardaki bilgilerin, başka bir bilgisayara aktarılması olarak açıklanabilir. Bu iki bilgisayar arasında yapılan bilgi alış-verişini yakalamaya "sniffing" denilir. Bir kaç bilgisayarın, bir ağ üzerinde birbirleriyle paylaşıma açık olarak bağlanılmasında kullanılan en popüler yol "ethernet" tir. Ethernet protokolü bir bilgi paketini aynı devreler üzerindeki tüm bilgisayarlara yollayarak çalışır. Gönderilen paketin başlığında, paketin gideceği bilgisayarın adresi yazılıdır. Sadece bu paketteki adres ile adresi tutan makine bu bilgileri alabilir. Her paketi kabul eden bir makine, yani paket başlığındaki adrese aldırmayan bir makine, çok karışık bir hal alacaktır. Bu karışıklık sayesinde, sniffer işlevini yerine getirecektir.

Bu çalışmada amacımız, bir UNIX sistemde çalışacak, ağ'da gidip gelen paketleri yakalayıp bu paketlerin içinden istediğimiz verileri almamıza yarayan bir yazılımın gerçekleştirilmesidir. Bir paketin içinde bize lazım olan veriler ise şunlardır.

1- Paket boyutu

2- Kaynak ip'si

3- Kullanıcı ip'si

4- Protokolu

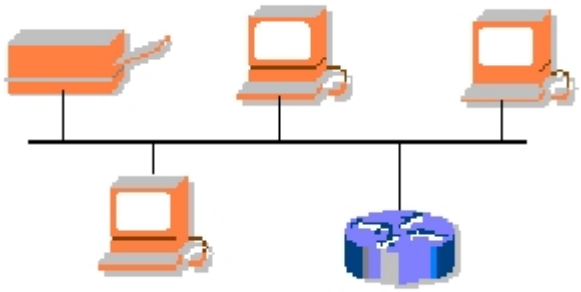
5- Paketin yakalanma zamanı

Böylece kullanıcıların ağ'da ne gibi işlemler yaptığını internet trafiğinin ne kadarını meşgul ettiğini görebileceğiz. Çalışmanın ilk bölümünde ağ kavramlarından bahsedilmiş ağ çeşitleri anlatılmıştır. İkinci bölümde, OSI katmanlarından, ağ topolojilerinden bahsedilmiştir. Üçüncü bölümde, TCP/IP den, protokollerden bahsedilmiştir. Dördüncü bölümde, yeryüzündeki en geniş ağ olan internetin doğuşu, sniffer'ların işleyiş şekli anlatılmıştır. Son olarak da, geliştirilen yazılım hakkında bilgi verilmiştir.

2. TEMEL AĞ KAVRAMLARI

2. 1 Ağ (Network) Nedir?

Ağ (Network) kavramı, var olan kaynakların kullanıcılar tarafından beraber kullanılması, bilgiye ortak ulaşmaları ve buna bağlı olarak da maliyet ve zaman tasarrufu sağlanılması gereksiniminden ortaya çıkmıştır. Bu temel kuraldan hareketle oluşan ağlar günümüzde uzaktaki bilgiye erişim (Web), kişisel iletişim (E-posta, ICQ, IRC, Video-konferans), interaktif eğlence (Web-TV, oyunlar) gibi kavramlarla hayatımızda önemli bir yer kaplamaktadır. Bir ağın oluşabilmesi için minimum iki makineye, bunlara takılı olarak ağ kartlarına ve de bağlantıyı sağlamak içinde kabloya ihtiyaç vardır. Aşağıdaki şekil örnek bir ağı şematik olarak göstermektedir.



Şekil 2. 1 Örnek bir ağ

2. 2 Ağların Gelişimi ve Ağ Teknolojileri

2. 2. 1 Ana Makine (MainFrame) Modeli

Ağ kavramı ilk olarak Ana Makine (MainFrame) teknolojisi ile ortaya çıkmıştır. Ana makinenin kendi işlemcisi (CPU), sabit diski (hard disk), ve bunları kumanda etmek için bir ekranı ve klavyesi ve de terminallere bağlı seri portları vardı. Bu aptal terminaller (dumb terminal) sadece ekran ve klavyeden oluşurdu, yani bir deyişle pasif makinelerdi. Terminallerin yerel bir disk alanları da olmadığı için bilgiyi ana makine üzerinde saklarlardı. Tüm yük ana makinenin üzerindeydi ve bu yüzden çok pahalıydı. En büyük dezavantajı tabii ki güvenilir olmaması, yani ana makinede çıkacak bir sorunun tüm sistemi etkilemesi, terminallerin kendi başlarına işlem yapabilme

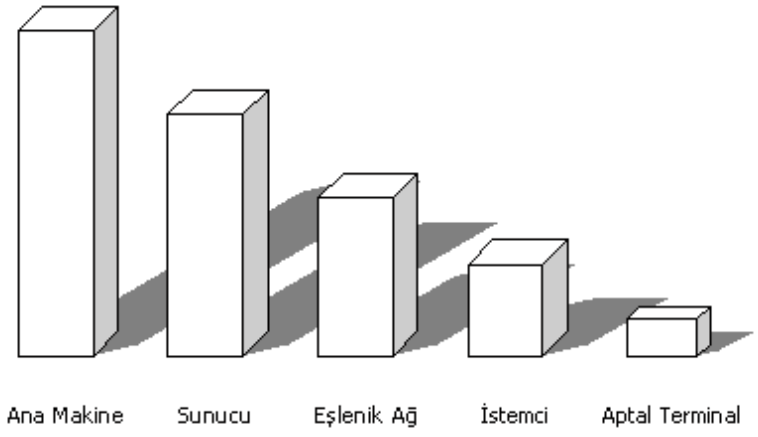
kabiliyetlerinin olmaması idi. Bu önemli sorun halen çok popüler olan İstemci / Sunucu (Client/Server) modelinin doğmasına yol açtı.

2. 2. 2 İstemci / Sunucu (Client / Server) Modeli

İstemci / Sunucu modeli ile pasif terminaller yerine kendi başlarına işlemler yapabilen ve kendi sabit disklerinde programlar saklayabilen makineler geldi. Böylece her istemci kendi başlarına belirli işlemleri yerine getirebilmekte, yetersiz durumda kaldıklarında ise o işe özelleşmiş olan sunuculara başvurmakta idiler.

2. 2. 3 Eşlenik Ağ (Peer to Peer) Modeli

İstemci/Sunucu modelinin gelişmesi ve yaygınlaşması ile birlikte istemcilerin daha ön plana çıktığı, özelleşmiş sunuculara ihtiyaç duyulmayan ağ örnekleri de ortaya çıkmaya başladı. Bu ağlarda makineler yaklaşık özelliklerde idi ve işleyiş olarak birbirlerine üstünlük sağlamıyorlardı. Tamamen Windows 95/98 kullanan ağlar, bu tür ağlara örnek teşkil etmektedir. Aşağıdaki şekil de ağda güç sıralaması şematik olarak gösterilmektedir.



Şekil 2. 2 Ağ Teknolojilerinde Güç Sıralaması

2. 3 Ağ Çalışma Prensipleri

Temel olarak ağlarda iki tip çalışma prensibi vardır:

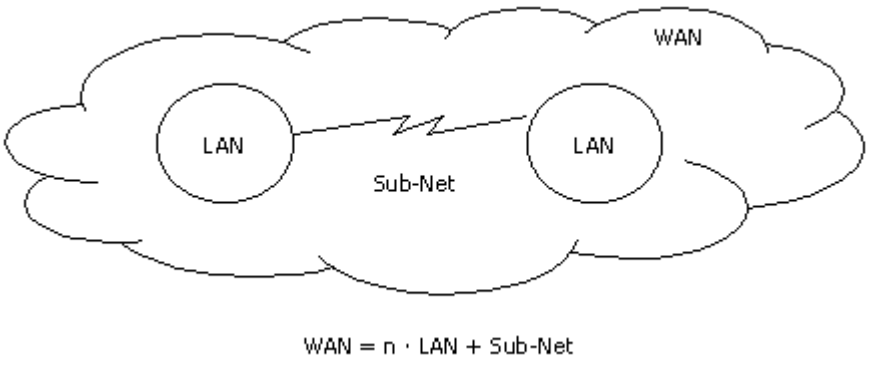
Yayın (Broadcast) : Ağa atılan bir paketin her bilgisayara gönderilmesi. **Noktadan noktaya (Point to Point)** : Ağa atılan bir paketin özel bir noktaya iletilmesi.

Ağların çalışma prensipi genelde yayın tarzındadır. Buna rağmen İnternet omurgası noktadan noktaya çalışmaktadır.

3. BÜYÜKLÜKLERİNE GÖRE AĞLAR

3. 1 Yerel Alan Ağı (Local Area Network): Kurulabilecek en küçük çaplı ağ olmakla birlikte büyüklükleri bir oda veya bir binayla sınırlı kalmayıp 1 km'ye kadar çıkabilmektedir. Örneğin küçük ve orta dereceli kurumların ağları.

3. 2 Geniş Alan Ağı (Wide Area Network): Aralarında 1 km'den fazla mesafe olan LAN'ların birleşmeleriyle meydana gelirler. Türkiye'deki en meşhur WAN'lardan biri Turnet (Türkiye iç omurgası), bir diğeri Ulaknet'tir. Ulaknet Üniversiteler arası ağa verilen isimdir. Wan ağlarının oluşumu aşağıdaki şekilde gösterilmiştir.



Şekil 3. 1 Wan Ağların Oluşumu

3. 3 Metropol Alan Ağı (Metropolitan Area Network): WAN'ların şehir bazında ya da şehirler arası birleştirilmeleriyle oluşur. Fakat günümüzde MAN kavramı kullanılmamakta, yerine WAN terimi tercih edilmektedir.

4. OSI KATMANLARI

Bilgisayar ağlarının ilk günlerinde farklı firmalar kendilerine özel teknolojilerle ağ sistemleri geliştiriyorlar ve satıyorlardı. Kendi başlarına düzgün çalışan bu ağlar ortak çalışma yeteneğine sahip değildi. Herbirinin kendine özel yazılım ve donanımları vardı. Farklı isimlendirme sistemleri ve sürücüler kullanan bu ağları birbirleriyle iletişime geçirmek imkansızdı.

Ağ sistemlerinin bu özel yapısı diğer donanım ve yazılım üreticilerinin bu ağlar için ürün geliştirmesini de imkansız hale getiriyordu.

Bir ağ sistemi satın aldığınızda kablosundan ağ kartına kadar, hub, sürücüler ve ağ işletim sistemi dahil herşeyi üretici firmadan paket olarak çok yüksek bir fiyata almak zorundaydınız. Ayrıca ilerleyen dönemde de bu tek firmaya bağımlı duruma geliyordunuz.

Ağ sistemlerine olan talebin artması ile ağ sistemlerinin işlevlerini tanımlayan ortak bir model oluşturulması gerektiği anlaşıldı.

Bunu gerekli kılan bir diğer unsur ise ağ sistemlerini açıklamakta kullanılan terimlerin üreticiden üreticiye değişiklik göstermesi, ağ üzerinde işlem gören yazılım ve donanım bileşenlerinin ne görev üstlendiklerinin standart halinde olmamasıydı. 1984 yılında International Organization of Standardization (ISO) isimli kurum Open System Interconnection modelini (OSI) ortaya koydu.

OSI Modeli değişmez bir kanun değildir. İsteyen kendi başına bir ağ sistemi tasarlayabilir ve belki de çalışır hale getirir. Ancak OSI modeli referans alınmadıysa diğer ağlarla iletişimi zor olacak değişik üreticiler bu ağ sistemi için donanım ve yazılım üretemeyecekler demektir. (http://www. turkcenet. org/yerel_htm/OSI. htm)

4. 1 Protokoller ve Kavram Karmaşası

Protokol kelimesi günlük yaşamda karşımıza bir yabancı devlet başkanı ülkemizi ziyarete geldiğinde, karşılama töreni ve sonrasında takip edilen kurallar bütünü olarak çıkar. Bu manasıyla protokol nasıl bir devlet başkanının diğerinin konuşmasını

kesmesini ya da yanlış yerde yanlış harekette bulunmasını önlüyorsa, bilgisayar ağlarında da benzer bir işlevi ifade eder.

Bilgisayar ağları söz konusu olduğunda, protokol kelimesi iki aygıt veya yazılımın önceden belirlenmiş kurallar çerçevesinde nasıl haberleşmeleri gerektiğini tanımlar.

Basit bir FTP programı ile dosya gönderirken bile arka planda birçok protokol görev yapar. File Transfer Protocol (FTP) iki bilgisayar üzerinde çalışan iki yazılımın nasıl veri değişimi yapacağını belirler. Transmit Control Protocol (TCP) gönderen sistem üzerinde yollanacak veriyi parçalara bölünmesini ve alıcı sistem üzerinde ise tekrar birleştirilmesini sağlar. Internet Protocol (IP) ise verinin değişik yönlendiriciler üzerinde doğru yolu izleyerek karşı tarafa ulaşmasını sağlar.

4. 2 Yedi Katman

Ağlar ile ilgili birçok dökümanda OSI modeline atıf yapıldığını görürsünüz. OSI modeli sayesinde bir cihazın veya protokolün ağ içinde ne görev üstlendiği daha rahat anlatılabilir.

OSI modeli verinin bir bilgisayar üzerinde bir program'dan, ağ ortamından geçerek diğer bir bilgisayar üzerindeki diğer bir programa nasıl ulaşacağını tanımlar. Model bu süreci 7 katman halinde inceler:

7. Katman - Uygulama
6. Katman - Sunum
5. Katman - Oturum
4. Katman - Taşıma
3. Katman – Ağ
2. Katman - Veri Bağlantısı
1. Katman - Fiziksel

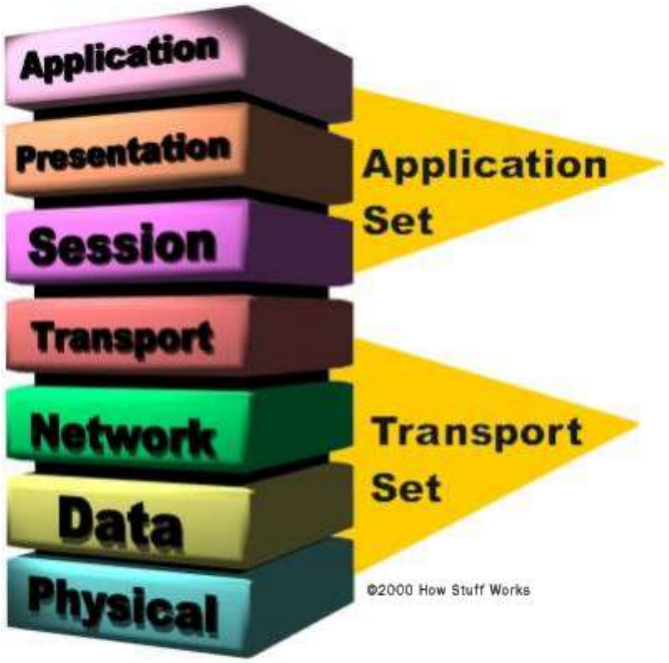
Uygulama katmanı veriyi sunum katmanına sunum ise oturum katmanına aktarırır. Bu şekilde veri fiziksel katmana kadar ulaşır.

Veri alımında ise bu işlem tam tersi şekilde gerçekleşir.

OSI Modelinde her katman çözülmesi gereken problemleri tanımlar. Bu katmanda çalışan aygıt ve protokoller ise bu problemlere çözüm getirir.

7 katmanlı OSI modeli 2 bölümde incelenebilir: Uygulama seti ve Veri Aktarım seti. Application Set (Uygulama seti) uygulamalar yani programlarla ilgili konuları içerir. Genellikle sadece yazılımsaldır. Modelin en üstündeki uygulama katmanı kullanıcıya en yakın katmandır.

Transport Set (Veri Aktarım Seti) veri iletişimi ile ilgili meseleleri tanımlar. Fiziksel ve veri aktarım katmanları hem yazılım hem de donanım olarak görevini yerine getirebilir. Fiziksel katman (en alt katman) fiziksel ağ ortamına (örneğin, ağ kablosuna) en yakın katmandır ve esas olarak bilgiyi kablodan aktarmakla görevlidir. Aşağıdaki şekilde OSI Katmanlarının şematik gösterimi yapılmıştır.



Şekil 4. 1 OSI Katmanları Şekili

4. 3 Katman 1 : Fiziksel Katman (Physical Layer)

1. katman veya fiziksel katman verinin kablo üzerinde alacağı fiziksel yapıyı tanımlar. Diğer katmanlar 1 ve 0 değerleriyle çalışırken, 1. katman 1 ve 0'ların nasıl elektrik, ışık veya radyo sinyallerine çevrileceğini ve aktarılacağını tanımlar. Gönderen tarafta 1. katman bir ve sıfırları elektrik sinyallerine çevirip kabloya yerleştirirken, alıcı tarafta 1. katman kablodan okuduğu bu sinyalleri tekrar bir ve sıfır haline getirir.

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

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

4. 4 Katman 2: Veri Bağlantısı Katmanı (Data Link Layer)

Veri bağlantısı katmanı fiziksel katmana erişmek ve kullanmak ile ilgili kuralları belirler. Veri bağlantısı katmanının büyük bir bölümü ağ kartı içinde gerçekleşir. Veri bağlantısı katmanı ağ üzerindeki diğer bilgisayarları tanımlama, kablonun o anda kimin tarfından kullanıldığının tespiti ve fiziksel katmandan gelen verinin hatalara karşı kontrolü görevini yerine getirir.

Veri bağlantısı katmanı iki alt bölüme ayrılır: Media Access Control (MAC) ve Logical Link Control (LLC).

MAC alt katmanı veriyi hata kontrol kodu (CRC), alıcı ve gönderenin MAC adresleri ile beraber paketler ve fiziksel katmana aktarır. Alıcı tarafta da bu işlemleri tersine yapıp veriyi veri bağlantısı içindeki ikinci alt katman olan LLC'ye aktarmak görevi yine MAC alt katmanına aittir. LLC alt katmanı bir üst katman olan ağ katmanı (3. katman) için geçiş görevi görür. Protokole özel mantıksal portlar oluşturur (Service

Access Points, SAPs). Böylece kaynak makinada ve hedef makinada aynı protokoller iletişime geçebilir. LLC ayrıca veri paketlerinden bozuk gidenlerin (veya karşı taraf için alınanların) tekrar gönderilmesinden sorumludur. Flow Control yani alıcının işleyebileğinden fazla veri paketi gönderilerek boğulmasının engellenmesi de LLC'nin görevidir.

4. 5 Katman 3: Ağ Katmanı (Network Layer)

Bir paket hedefine ulaşmak için bir ağdan diğer bir ağa geçmek zorunda kaldığında başka problemler de baş gösterebilir. Adresleme ağlar arasında farklı olabildiği gibi, bir ağ diğerinden çok geniş olduğu için paketi kabul etmeyebilir veya protokoller farklı olabilir. Heterojen ağların arabağlantılarının sağlıklı birşekilde yapılıp bu problemlerin üstesinden gelme ağ katmanın sorumluluğundadır.

4. 6 Katman 4: Taşıma Katmanı (Transport Layer)

Taşıma katmanı üst katmanlardan gelen veriyi ağ paketi boyutunda parçalara böler. NetBEUI, TCP ve SPX gibi protokoller bu katmanda çalışır. Bu protokoller hata kontrolü gibi görevleride yerine getirir.

Taşıma katmanı alt katmanlar (Transport Set) ve üst katmanlar (Application Set) arasında geçit görevini görür. Alt katmanlar verinin ne olduğuna bakmandan karşı tarafa yollama işini yaparken üst katmanlarda kullanılan donanım ile ilgilenmeden verinin kendisi ile uğraşabilirler.

4. 7 Katman 5: Oturum Katmanı (Session Layer)

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

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

4. 8 Katman 6: Sunum Katmanı (Presentation Layer)

Sunum katmanının en önemli görevi yollanan verinin karşı bilgisayar tarafından anlaşılabilir halde olmasını sağlamaktır. Böylece faklı programların birbilerinin verişini kullanabilmesi mümkün olur.

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

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

4. 9 Katman 7: Uygulama Katmanı (Application Layer)

Uygulama katmanı programların ağı kullanabilmesi için araçlar sunar. Microsoft API'leri uygulama katmanında çalışır. Bu API'leri kullanarak program yazan bir programcı, örneğin bir ağ sürücüsüne erişmek gerektiğinde API içindeki hazır aracı alıp kendi programında kullanır. Alt katmanlarda gerçekleşen onlarca farklı işlemin hiçbirisiyle uğraşmak zorunda kalmaz.

Uygulama katmanı için bir diğer örnek HTTP'dir. HTTP çalıştırılan bir program değil bir protokoldür. Yani, bu bir kurallar dizesidir. Bu dizeye gören çalışan bir Browser (IE mesela), aynı protokolü kullanan bir Web sunucuya erişir.

4. 10 OSI Katmanı İle İlgili Genel Bilgi

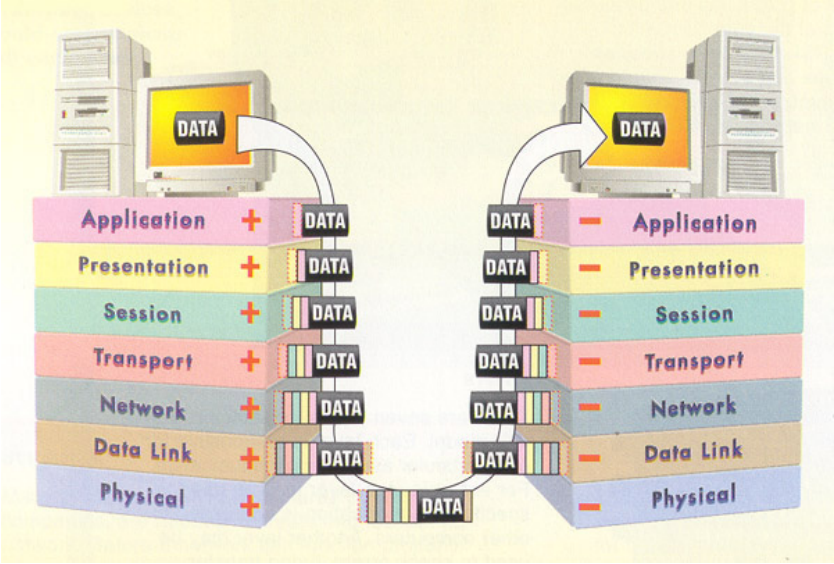
OSI kavramsal bir modeldir. Yani hiç biryerde OSI programı veya OSI donanımı diye bir şey göremezsiniz. Ancak yazılım ve donanım üreticileri bu modelin tanımladığı kurallar çerçevesinde üretim yaparlar ve ürünleri birbiri ile uyumlu olur.

OSI Modeli aygıtların işlevlerini anlamak ve açıklamakta kullanılır. Örneğin HUB dediğimiz cihazlar gelen veriyi sadece bir takım elektrik sinyalleri olarak gören ve bu sinyalleri çoklayıp, diğer portlarına gönderen bir cihazdır. Bu da HUB'ların fiziksel (1. katman) katmanda çalışan cihazlar olduğunu gösterir.

Oysa switch denen cihazlar 2. katmanda çalışırlar. Çünkü 2. katmanda tanımlı MAC adreslerini algılayabilirler ve bir porttan gelen veri paketini (yine elektrik sinyalleri halinde) sadece gerekli olan porta (o porttaki makinanın MAC adresini bildiği için) yollayabilirler.

Yönlendiriciler (router) için ise bazen "3. katman switch'ler" tabirini görebilirsiniz. Çünkü bu cihazlar biraz daha ileri gidip, 3. katmanda veri paketine eklenmiş IP adresi gibi değerleri de okuyabilir ve ona göre veri paketini yönlendirebilir.

OSI Modelinde en üst katmandan yola çıkan ham veri (örneğin A harfi, bir resim, bir ses dosyası vb.), her katmanda o katmanla ilgili bazı ek bilgiler eklenerek bir alt katmana aktarılır. Aşağıdaki sekilde OSI Modeline göre veri iletişiminin, hangi katmanları kullanarak yapıldığı gösterilmiştir.



Şekil 4. 2 OSI Modeline Göre Veri İletişimi

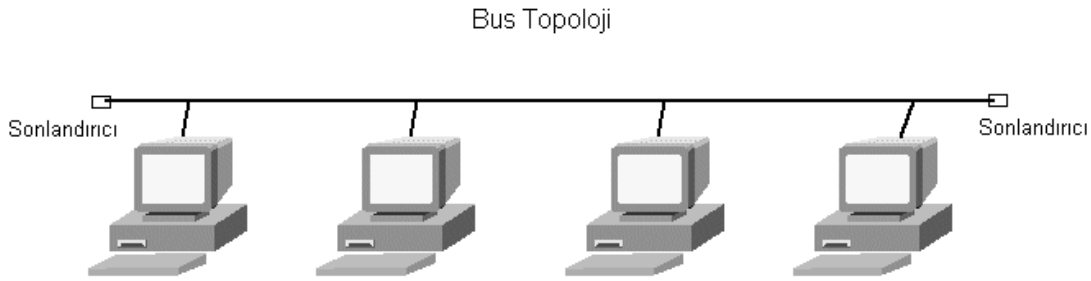
Alıcı bilgisayarda ise, alttan üste doğru her katman karşı taraftaki eş katmanın bilgisini kullanır, gerekeni yapar, bu bilgiyi temizleyip paketi bir üst katmana geçirir.

5. AĞ TOPOLOJİLERİ

Ağ'ın fiziksel yapısı, kablolarla bağlantı şeklidir. Temel olarak 3 topoloji vardır. Bu topolojiler daha sonra geliştirilirek 2 topoloji daha oluşturulmuştur.

5. 1 Kuyruk (Bus) :

Doğrusal bir hat üzerinde kurulmuş bir yapıya sahiptir. Makineler kabloya T-konnektörler aracılığıyla bağlanırlar ve kablonun rezistansını düşürmemek için açıkta kalan iki ucuna sonlandırıcılar takılır. 10 mps hızda çalışır. Bir makinede veya kablonun herhangi bir noktasında oluşan arıza tüm sistemin çalışmasını engeller. Bu dezavantajına rağmen kurulumu en kolay yapı olduğu için tercih edilmektedir. Maksimum kapasitesi 10-12 makine olup, iki makine arası maksimum mesafe ince eş-eksenli (thin coaxial) kablo kullanıldığında 185 m, kalın eş-eksenli (thick coaxial) kablo kullanıldığında 500 metredir. Aşağıdaki şekilde Bus Topoloji şematik olarak gösterilmiştir.

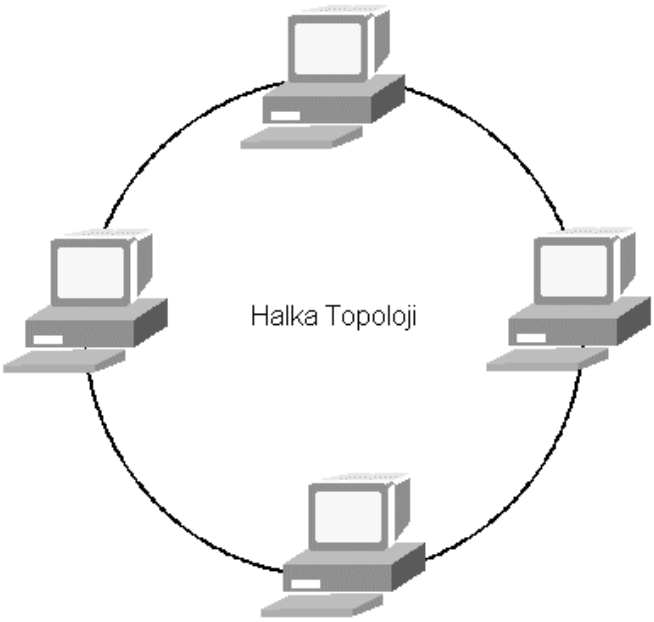


Şekil 5. 1 Bus Topoloji Şekili

5. 2 Zincir (Ring) :

Kuyruk yapısındaki bir ağın sonlandırıcıların çıkarılarak iki ucunun birleştirilmesiyle oluşan ağ yapısıdır. En yaygın uygulaması IBM'e ait olan Token Ring topolojisidir. 4 mps veya 16 mps hızda çalışır. Kuyruk yapısının tüm özelliklerini taşımakla birlikte ağda bulunan düşük hızlı bir kart tüm sistemi yavaşlatır. Zincir yapısında ağda varolduğu düşünülen sanal bir jeton (token) tüm makineleri sırayla dolaşır ve bilgi alışverişi bu şekilde sağlanır.

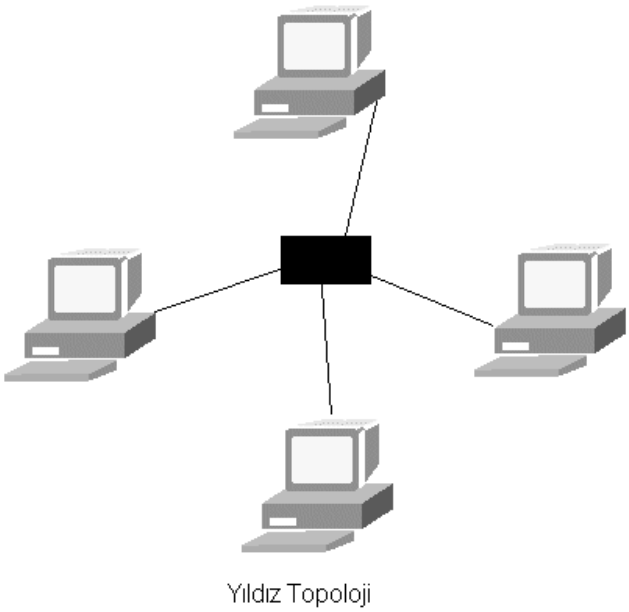
Aşağıdaki şekilde Ring Topoloji şematik olarak gösterilmiştir.



Şekil 5. 2 Ring Topoloji Şekili

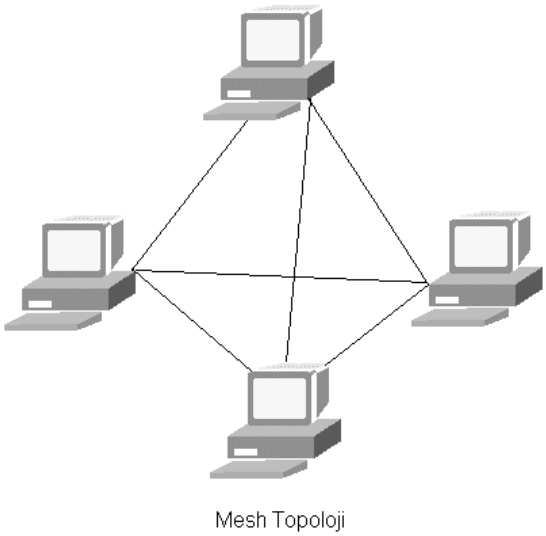
5. 3 Yıldız (Star)

Diğerlerinden farklı olarak, kablo, konnektör ve ağ kartına ek olarak hub, switch gibi diğer cihazlar kullanılarak oluşturulan ağ yapılarıdır. Genelde UTP (Unshielded Twisted Pair) korumasız çift dolanmış ya da STP (Shielded Twisted Pair) korumalı çift dolanmış kablo kullanılarak oluşturulur ve bilgisayarlarla bağlantı cihazının (hub gibi) maksimum mesafesi 100 metredir. Kullanılan çift dolanmış kablonun ve ağ kartının çeşitine göre farklı hızlarda çalışır. Her bilgisayarın bağlantısındaki problem yalnızca onun çalışmasını engellerken, ağdaki diğer cihazlar çalışmalarına devam ederler. Ancak bağlantı cihazlarındaki (hub, switch) problemler, o cihaza bağlanan tüm cihazların çalışmasını engeller. Diğerlerine göre daha güvenilir fakat pahalı çözümler sunar. Aşağıdaki şekilde Star Topoloji şematik olarak gösterilmiştir.



Şekil 5. 3 Star Topoloji Şekili

5. 4 Mesh (ağ) topoloji



Şekil 5. 4 Mesh Topoloji Şekili

Yukarıdaki şekilde Mesh Topoloji şematik olarak gösterilmiştir. Bu topolojide tüm bilgisayarlar diğer bilgisayarlara ayrı bir kablo ile bağlıdır. Bu bağlantı teorik olarak ideal bağlantı tipidir. Ancak aradaki kablo sayısı terminal sayısı arttıkça katlanarak arttığı için gerçek hayatta sadece çok özel durumlarda ve az sayıda bilgisayar arasında kullanılır.

5. 5 Melez (Hybrid) topolojiler

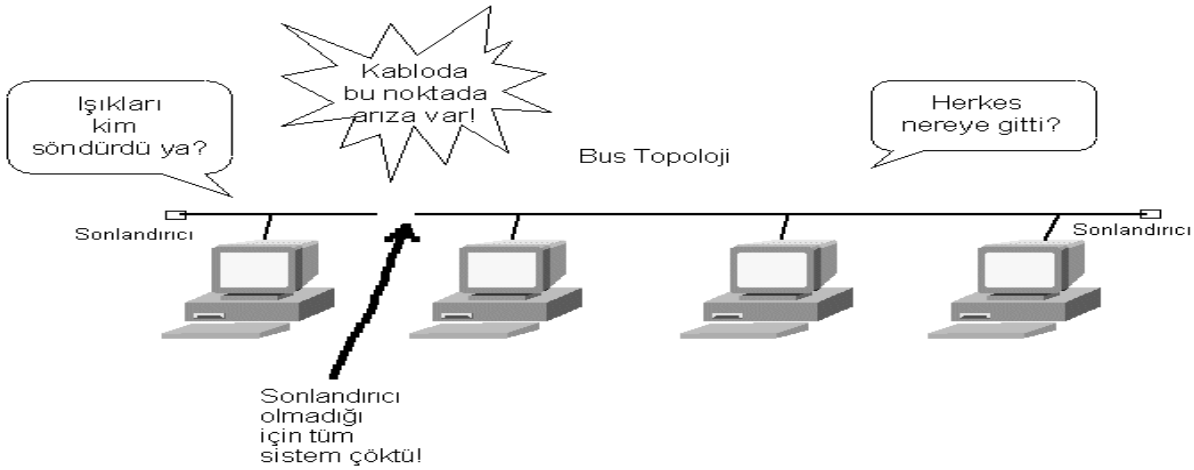
Bu topolojileri başlangıç noktası olarak alıp geliştirilen değişik ağ teknolojileri olduğundan bahsetmiştik. Bu teknolojilerden önemli olanları Token Ring ve Ethernet'tir. Token Ring bir ağ görme ihtimaliniz de çok çok az olduğu için onu bir kenara bırakırsak, elimizde sadece Ethernet kalır. Bugün "ağ kuruyorum" ya da "ağ kurduk süper oldu" diyen birisi %100 Ethernet'ten bahsediyordur. Biz de Ethernet'in kullandığı topolojileri açıklayalım.

Ethernet ilk başta bus topoloji olarak tasarlandı. Koaksiyel bir kablo sırayla tüm bilgisayarları dolaşıyordu. Ethernet ağında bilgisayarlar bu tek kabloya bağlı olduklarını düşünürler. Bir diğer sisteme veri yolladıklarında, veri aslında aynı kabloya bağlı tüm sistemlere ulaşır. Tüm bilgisayarlardan sadece "doğru" olanı bu veriyi alır ve işler.

Ethernet ağında her bilgisayar, daha doğrusu her ağ kartı (bu noktada ethernet kartı diyebiliriz) farklı bir adrese sahiptir (MAC adresi). Veri kablo üzerine yerleştirilirken veri üzerine alıcı ve gönderenin MAC adresleri yazılır. Böylece veriyi alan tüm sistemlerden sadece "doğru" olanı veriyi alır ve işleme koyar, diğerleri kendilerine gelmeyen (gelen ama ait olmayan) veriyi göz ardı eder.

Bu noktada ilk ethernetin hem mantıksal hem de fiziksel olarak bus yapıda çalıştığı anlaşılıyor. Elbette ethernet kullanılacak kablo tipi, maksimum uzunluk ve diğer değerleri de tanımlamıştır.

Zaman içinde fiziksel bus yapı ihtiyaçlara cevap veremez hale gelmiştir. Fiziksel bus yapıda, yani tüm bilgisayarların aynı kabloya bağlandıkları sistemde kablonun bir noktasında oluşan kopukluk veya kısa devre tüm ağı çökertir.



Şekil 5. 5 Melez Topoloji Şekili

Yukarıdaki şekilde Melez Topoloji şematik olarak gösterilmiştir. Ağ'a yeni bir makine eklemek, kablonun bir bölümüne ek yapmak demektir bu işlem sırasında ağ çalışamaz vaziyettedir.

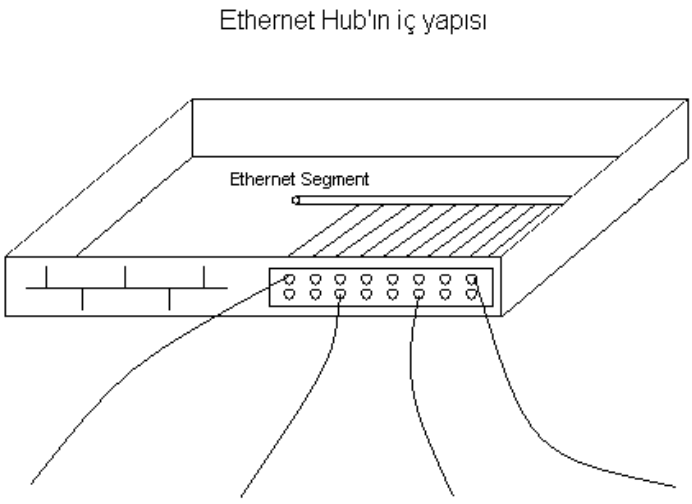
Ağ'da arıza olduğu zaman tüm sistemleri dolaşan tek bir kablonun herhangi bir yerindeki arızayı bulmak çok zahmetlidir.

Yapısal kablolama dediğimiz de, çok fazla sayıda bilgisayarın kullanıldığı binalarda veya kampüslerde gerçekleştirilen kablolama da bus yapı kullanmak mümkün değildir. Çünkü bus yapı ağacın dalları gibi merkezden binanın katlarına oradan da odalara dallanan bir yapıya izin vermez.

Sonuç itibariyle fiziksel bus topolojinin ihtiyaçları karşılamaktan uzak olduğu anlaşılınca yeni bir sistem arayışına gidildi. Çözüm, ethernetin mantıksal topolojişini muhafaza edip fiziksel topolojiyi, yani kablolama yapısını yıldız topoloji ile değiştirmekti. Yıldız topolojide her bilgisayardan ayrı bir kablo merkezi bir kutuya (hub) gider. Kablolardan birinde oluşan arıza sadece o bilgisayarı etkiler.

6. DONANIMIN TOPOLOJİ GELİŞİMİNE UYUMU

Ethernet için yeni fiziksel topoloji yıldız topolojidir. Kullanılan kablo da koaksiyelden UTP'ye dönüşmüştür. Ancak mantıksal olarak ethernet hala bus topoloji kullanır. Böylece yıldız'a geçmeden önce kurulmuş binlerce ethernet ağı devre dışı kalmamış olur. Fiziksel yıldız topolojide kullanılan hub içinde mantıksal bir bus yapı vardır. Bilgisayarlardan birişinin yolladığı veri paketi hub'a ulaşınca, hub bu paketin kopyalarını oluşturup tüm portlarına yollar. Yani bus yapıda olduğu gibi veri paketi diğer tüm bilgisayarlara erişir ve sadece alması gereken bu paketi alır ve işler diğerleri ise siler. Bunu daha iyi anlamak için bir ethernet hub'ı Şekil 6.1 deki gibi temsili olarak gösterebiliriz. Hub'a bağlı bilgisayarlar yıldız topoloji kullanmalarına rağmen, hub içinde aynı bus gibi tek bir hat olduğunu düşünebiliriz. Aşağıdaki şekil de Ethernet'in iç yapısının şematik olarak gösterimi yapılmıştır.



Şekil 6. 1 Ethernet'in İç Yapısının Şekili

Böylece koaksiyel kablolu fiziksel bus ethernet ve utp kablolu fiziksel yıldız ethernet bir arada rahatça kullanılabilir. Çünkü çalışma mantıkları yani mantıksal topolojileri aynıdır.

Zaten hemen hemen tüm ethernet hub'larda bir tane de koaksiyel kablo girişi vardır. Böylece fiziksel yıldıza geçiş ethernet için çok kolay olmuş, zaten en büyük pazar payına sahip ethernet ürünleri, fiziksel yıldızın tartışmasız avantajını da elde edince, günümüzde en yaygın ağ teknolojisi haline gelmiştir.

Ethernetin kullandığı bu melez topoloji bazen star-bus topoloji olarak anılır. Tek melez topoloji star-bus değildir. IBM'in geliştirdiği ve günümüzde popülerliğini kaybeden, ancak zamanında geniş bir kullanım alanı bulmuş olan Token Ring ağ teknolojisi de star-ring melez topolojişini kullanır. Bu sistemde de dışarıdan bakıldığında aynı ethernetin star-bus'ı gibi kablolama yıldız şeklindedir. Her terminalden ayrı bir kablo ethernet'teki hub'ın benzeri bir kutuya girer. Ancak bu kutunun içinde Token Ring ağlarının kullandığı mantıksal bir halka (ring) yapısı mevcuttur.

Geçmişte bir şekilde piyasaya çıkmış fakat tutunamamış bir çok ağ teknolojisinden söz edilebilir. Ancak günümüzde kurulacak bir ağı seçerken çoğu zaman piyasanın bize sunduğu (çoğu zaman da fiyat, performans ve güvenililirlik açısından en iyi olan) teknolojiyi alıp kullanılır.

Günümüzde en yaygın kullanılan ağ tipi ethernet'tir. Ethernet ilk başta hem fiziksel hem de mantıksal olarak bus yapıda tasarlandı. Zaman içinde fiziksel bus ihtiyaçları karşılamayınca, fiziksel yıldız topoloji kullanan, yani hub ve UTP kablo kullanan ethernet geliştirildi. Ancak bu yeni ethernet hem geriye doğru uyumluluk hem de ethernetin temel çalışma mantığı öyle gerektirdiği için mantıksal bus kullanmaya devam ediyor.

6. 1 Fiziksel Katman

Bu katman tamamiyle fiziksel bağlantıdan sorumlu olup, kablo, konnektor gibi parçalardan meydana gelmektedir.

6. 1. 1 Eş-Eksenli Kablo (coaxial / BNC)

Eş-eksenli Kablo, televizyon kablosunun daha esnek ve ince olanıdır. Bakır tellerden ve üzerinde manyetik korumadan ibarettir. İnce ve kalın olmak üzere iki çeşittir. İnce olanının taşıma mesafesi 185m. Kalın olanının ki ise 500 metredir. Bu nedenler kalın eş-eksenli kablolar genelde omurga yapılarında kullanılır.

6. 1. 2 Çift Dolanmış Kablo (twisted pair / UTP-STP)

8 tane çifte dolanmış telden ibarettir. 10 Mbit hızda çalışırken bunların yanlızca 4 tanesi kullanılır. 100 Mbit çalışabilmesi için bu 8 telin belirli bir sıra takip eder durumda bağlanması gerekmektedir. Korumalı (STP) ve korumasız (UTP) olarak iki çeşittir.

CAT3 10 mps.
CAT4 4-16 mps.
CAT5 100 mps.
CAT6 1000 mps.
CAT7 1000 mps.

Bunlar dışında fiber kablo, kablolu TV, telefon hatları veya kiralık hatlar (leased line) fiziksel katmana dahildir.

6. 2 Ağlarda Kullanılan Cihazlar

6. 2. 1 Yükseltici (Repeater) : Kablonun kapasitesinden daha fazla mesafelere bağlantı kurulması gerektiğinde araya bir yükseltici konularak sinyalin güçlendirilmesini sağlayan cihazdır. Aşağıdaki şekil de örnek bir yükseltici şekili gösterilmektedir.

Şekil 6. 2 Yükseltici Şekili

6. 2. 2 Hub: Yıldız yapısındaki ağlarda merkezi bağlantıyı sağlayan cihazdır. Üzerindeki port sayısıyla isimlendirilir ve bu portlara makineler takılır. Hub aslında içerisinde tüm portları birbirine bağlayan kablolardan oluşmuş bir cihazdır ve kablolardan taşınan bilgiyi anlama kapasitesine sahip değildir. Yanlızca bir porttan gelen paketleri diğer bütün portlara yayın (broadcast) şeklinde iletir. Bu yüzden fiziksel katmana dahildir. Aşağıdaki şekil de örnek bir hub gösterilmektedir.

Şekil 6. 3 Hub Şekili

6. 2. 3 Modem: Bilgisayarın dijital sinyallerini analoga çevirerek kablo üzerinden iletilmesini sağlayan cihazdır. 19600, 28800, 57600 Kb hızlarında çeşitli tipleri vardır. Kiralık hatlarda kullanılan modemlere senkron modem, çevirmeli bağlantılarda (dial-up) ise asenkron modem kullanılmalıdır. Aşağıdaki şekil de örnek bir modem şekili gösterilmiştir.

Şekil 6. 4 Modem Şekili

6. 2. 4 Switch: MAC adresleri mertebesinde çalışan bir cihazdır. Portlarına bağlanan makinelerin MAC adreslerini kendi tablosuna kaydeder ve switch içerisindeki data transferi noktadan noktaya gerçekleşir. Switchler hublara göre daha akıllı ve pahalı cihazlardır ve kendi üzerlerinde işlemcileri ve hafızaları vardır. Switch'ler yanlızca

makinelerin direk olarak bağlanması için değil aynı zamanda ağların yükünü azaltmak için kullanılırlar. Diyelim ki birbirine bağlı 4 adet 16 portluk hub var. Bu ağdaki yayın trafiği ve paket çarpışmaları bayağı yüksek olacaktır. Bu durumlarda ağa bir merkezi switch koyup buradan hubları besleme yöntemine gidilmelidir. Böylece her bir hubda oluşan trafik diğer hublara yayın olarak yansımayacak ve lokal kalacak, hublar arası iletişim gerektiğinde ise noktadan noktaya gerçekleşecektir. İyi bir switch yüksek bir hafızaya, portlara aktarım ve portlar arası iletim hızına sahip olmalıdır. Aşağıda da Switch şekili gösterilmiştir.

Şekil 6. 5 Switch Şekili

6. 2. 5 Router (Yönlendirici)

Networkler arası haberleşmenin yapılabilmesi için ara bağlantıyı sağlayacak cihazlara router denir. Routerin bir işlemcisi, epromu ve üzerinde bir işletim sistemi IOS (Internal Operating System) vardır.

Routerlar IP paketlerinin yönlendirilmesinden sorumludur ve bu yüzden üzerlerinde routing tabloları tanımlanmıştır. Routing tabloları iki çeşittir: Statik ve dinamik. Router'ın şekili aşağıda gösterilmiştir.

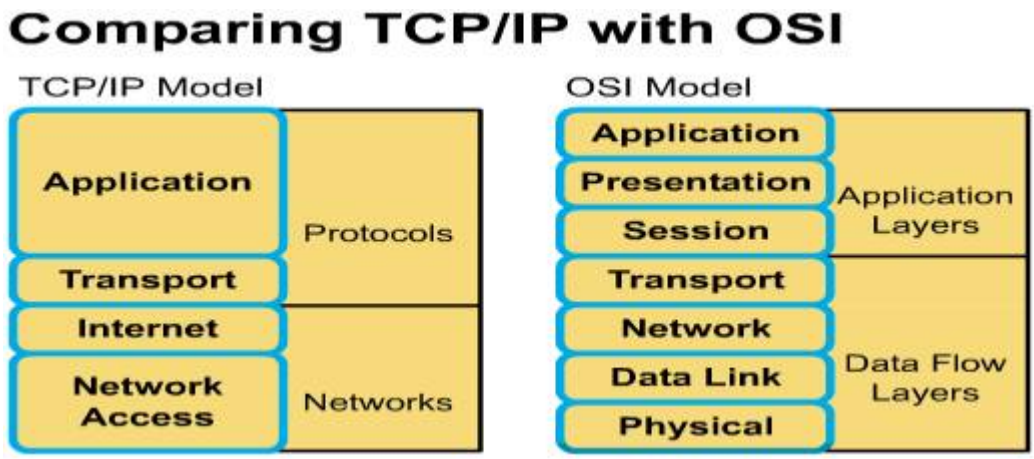
Şekil 6. 6 Router Şekili

7. TCP/IP İle OSI'NİN KARŞILAŞTIRILMASI

TCP / IP MODELİ

- Application Layer : OSI de 7. 6. ve 5. katmanlara karşılık gelir.
- Transport Layer : OSI de 4. katman olan Transport katmanına karşılık gelir.
- Internet Layer : OSI de 3. katmana karşılık gelir.
- Network Access : OSI de 1. ve 2. katmanlara karşılık gelir.

Bu katmanlarda sırasıyla şu işlemler yapılmaktadır. Aşağıdaki şekilde TCP/IP ile OSI'nin karşılaştırlmasının sonuçları gösterilmiştir.



Şekil 7. 1 TCP/IP İle OSI'nin Karşılaştırılması

- FTP : File Transfer Protocol
- HTTP : Hypertext Transfer Protocol
- SMTP : Simple Mail Transfer Protocol
- DNS : Domain Name System
- TFTP :Trivial File Transfer Protocol (Konfigürasyon dosyaları alıp gönderir.)
- TCP : Transmission Control Protocol
- UDP : User Datagram Protocol

TCP/IP Modelinde Application ve Transport Layer'lar Protocol'leri oluşturmaktadırlar. Internet ve Network Access Layer'lar ise Network'ü oluşturur.

OSI Modelinde ise; Application, Presentation ve Session Layer'lar Application Layer yani Uygulama katmanı olarak bilinirler. Diğer 4 katman olan Transport, Network, Data Link ve Physical ise Data Flow Layet yani Data İletim, akım katmanlarıdır.

7. 1 TCP/IP

TCP/IP internette veri transferi için kullanılan iki protokolü temsil eder. Bunlar Transmission Control Protokol (TCP) ve Internet Protocol (IP). Ve bu protokoller de daha genis olan TCP/IP protokol grubuna aittir. TCP/IP'de bulunan protokoller internette veri transferi için kullanılır ve internette kullanılan her türlü servisi sağlarlar. Bunların arasında elektronik posta transferi, dosya transferi, haber grupları, WWW erişimi gibi servisler TCP/IP sayesinde kullanıcılara sunulmaktadır.

TCP/IP protokol grubunu ağ seviyesi protokolleri ve uygulama seviyesi protokolleri olarak iki gruba ayırılabilir.

Ağ seviyesindeki protokoller genellikle kullanıcıya görünmeden sistemin alt seviyelerinde çalışırlar. Örnek olarak IP protokolü kullanıcıyla uzak bir makine arasındaki paket iletimini sağlar. IP ağ seviyesinde diğer protokollerle etkileşimli olarak çalişarak paketlerin hedef adrese gönderilmesini sağlar. Çesitli ağ araçları kullanmadığınız sürece sistemdeki IP trafiğini ve neler dönüp bittiğini anlayamazsınız. Bu araçlar ağda gidip gelen IP paketlerini yakalayabilen sniffer'lardir. Sniffer'lar konusuna ileri ki konularda ayrıntıyla anlatılacaktır.

Uygulama seviyesi protokolleri sistemde daha üst düzeyde çalışırlar ve kullanıcıya görünürler. Örnek olarak Dosya Transfer Protokolünü (FTP) verebiliriz. Kullanıcı istediği bir bilgisayara bağlantı isteğinde bulunur ve bağlantı yapıldıktan sonra dosya transferi işlemini gerçekleştirir. Ve bu karşılıklı transfer işlemleri kullanıcıya belli bir seviyede görünür, giden gelen byte sayısı, meydana gelen hata mesajları . . . gibi.

Kısaca TCP/IP internette veri transferini sağlayan protokoller grubudur.

Burda TCP/IP'nin tarihçesine girmeyeceğim söyleyeceğim tek şey TCP/IP diğer protokollere göre çok fazla avantaja sahip olduğu için çok kısa sürede en yaygın kullanılan protokol haline gelmiştir. Artık internetin belkemiği haline gelen TCP/IP herhalde üzerinde en çok çalışılan ağ protokolüdür.

Artık TCP/IP sadece internet değil bir çok alanda kullanılıyor. Intranet'ler mesela TCP/IP kullanılarak olusturulmaktadir. Bu tip bir sistemde TCP/IP'yi kullanmak diğer protokollere göre avantajlar içerir. En basitinden TCP/IP hemen hemen her türlü sistemde desteklendiği için çok kolay bir şekilde heterojen sistemler kurulabilir. Işte internette tamamen heterojen bir sistem olduğu için TCP/IP en uygun protokoldür.

TCP/IP protokolü günümüzde artık hemen hemen tüm işletim sistemlerinde desteklenmektedir. UNIX, DOS (Piper/IP ile), Windows (TCPMAN ile), Windows 95/98/2000/Me, Windows NT, Machintosh (MacTCP), OS/2, AS/400 OS/400 sistemlerinde TCP/IP desteği gelmektedir. Tabii her sistemin TCP/IP gerçeklemesi farklı olduğundan servis kalitesi de farklılıklar gösterebilir. Ancak temel olarak sunulan servisler aynıdır ve birbiriyle uyumlu olarak çalışırlar. (David, M, 1995)

7. 2 TCP/IP'nin İşleyişi

TCP/IP protokol yığınını kullanarak çalışır. (TCP/IP Stack) Bu yığın iki makine arasındaki veri transferini sağlamak için gereken tüm protokollerin birleşmiş bir halidir. Bu yığın kısaca en üstte "uygulama seviyesi", daha sonra "transport seviyesi", "ağ seviyesi", "datalink seviyesi" ve "fiziksel seviye"'lerden meydana gelir. Bu seviyelerde en üstte yakın olan seviyeler kullanıcıya daha yakındır, alta yakın olan seviyeler ise kullanıcıdan habersiz olarak çalışan seviyelerdir. Örnek olarak en üst düzey olan uygulama seviyesinde FTP, Telnet gibi programlari örnek verebiliriz. Bu programları çalıştırdığınızda diğer sisteme bir bağlantı kurulur ve veri transferi yapılır. Siz sadece yaptığınız işlemlerle ilgili sonuçları ve olayları görürsünüz ancak bir veri gönderdiğinizde bu veri ilk önce sizin bilgisayarınızdaki bu TCP/IP protokol yığınında aşağıya doğru inmek zorundadır. Yani uygulama seviyesinden, ftp'de verdiğiniz bir komut mesela, transport seviyesine, ordan ağ seviyesine ve en sonunda fiziksel seviyeye

iner ve artık diğer bilgisayara ulasmak üzere internet ağında ya da yerel bir ağda uzun yolculuğuna başlar. Gideceği makinenin fiziksel seviyesine ulaşana kadar veriler genellikle bir ya da daha fazla ağ geçidinden geçerler. (tracert komutu belirli bir hedefe hangi geçitlerden geçerek gidileceğini veren komuttur) En sonunda diğer makineye ulaşınca yine uygulama seviyesine ulaşıncaya kadar, bu sefer karşıda çalışan ftp sunucusuna, yine bu TCP/IP protokol seviyelerini bir bir yukarı doğru asmak zorundadır.

Bu arada bu seviyelere ne gerek var diyebilirsiniz. Ancak bu seviyelerin her biri değişik bir görevi üstlenmektedir. Bir seviye fiziksel olarak verilerin gönderilmesi işini yaparken başka bir seviye verileri ufak paket dediğimiz parçacıklara bölerek iletişim işini üstlenir, başka bir seviye ise iletisimde meydana gelebilecek hataları tespit eder. Bu şekilde tüm seviyeler bir uyum içinde çalışır ve her seviye karşı tarafta bulunan yine kendi seviyesindeki protokolle karşılıklı iletişim içindedir. Daha yukarıda ya da daha aşağıdaki bir seviyede ne gibi bir işin yapıldığıyla ve sonuçlarıyla ilgilenmez. (Yıldırımoğlu, M, 2002)

7. 3 TCP/IP Protokolleri

Kısaca TCP/IP protokol yığınının nasıl çalıştığını gördük ve şimdi kullanılan protokollere bir göz atalım.

Ağ seviyesi protokolleri veri transferi işlemini kullanıcıdan gizli olarak yaparlar ve bazı ağ araçları kullanılmadan farkedilemezler. Bu araçlar Sniffer'lardir. Sniffer bir cihaz ya da bir yazılım olabilir ve ağ üzerindeki tüm veri iletişimini izlemeye yarar. Bu araçlarin kullanılış maksadı ağda meydana gelebilecek hataları tespit etmek ve çözmektir. Ancak ileride de göreceğimiz gibi sniffer'lar da hacker ve cracker'lar tarafından kullanılan ölümcül makineler haline gelmistir.

Ağ protokolleri arasında önemli olarak Adres Çözümleme Protokolü (ARP), Internet Mesaj Kontrol Protokolü (ICMP), Internet Protokolü (IP) ve Transfer Kontrol Protokolü (TCP) protokollerini verebiliriz.

7. 3. 1 Donanım Katmanındaki Protokoller

• **ARP** (Address Resolution Protocol, yani Adres Çözümleme Protokolü) protokolü bir IP adresinin hangi ağ kartına (yani MAC adresine) ait olduğunu bulmaya yarar. TCP/IP'de veri gönderiminde gönderilecek bilgisayarın hangisi olduğunu bulmak için kullanılır. Ayrıca IP adresini yeni almış olan bir makine, o IP adresinin sadece kendisinde olduğunu ARP kullanarak teyid eder.

• **RARP** (Reverse ARP, yani Ters ARP) protokolü ARP'ın tersi işlemi yapar, yani hangi MAC adresinin hangi IP adresini kullandığını bulur. Bir TCP/IP ağında RARP'ın çalışacağı garanti değildir, zira RARP bir RARP sunucusuna ihtiyaç duyar.

7. 3. 2 IP Katmanındaki Protokoller

• **ICMP** (Internet Control Message Protocol, yani Internet Yönetim Mesajlaşması Protokolü), hata ve türlü bilgi mesajlarını ileten protokoldür. Örneğin, ping programı ICMP'yi kullanır.

• **RIP** (Router Information Protocol, yani Router Bilgi Protokolü) router'ların yönlendirme tablolarını otomatik olarak üretebilmesi için yaratılmıştır.

• **OSPF** (Open Shortest Path First, yani İlk Açık Yöne Öncelik) aynı RIP gibi router'ların yönlendirme tablolarını otomatik olarak üretebilmesine yarar. OSPF, RIP'ten daha gelişmiş bir protokoldür.

• **IGMP** (Internet Group Messaging Protocol, yani Internet Grup Mesajlaşma Protokolü) bir sistemin internet yayınlarına (multicast) abone olmasına ve aboneliği durdurmasına yarar. Bu yayınlar, UDP üzerinden yapılır ve genelde çoklu ortam (radyo veya video) içerikli olurlar.

• **DHCP** (Dynamic Host Configuration Protocol, yani Dinamik Cihaz Ayar Protokolü) bir TCP/IP ağına bağlanan bir cihaza otomatik olarak IP adresi, ağ maskesi, ağ geçidi ve DNS sunucusu atanmasına yarar.

7. 3. 3 Taşıma Katmanındaki Protokoller

• **UDP** (User Datagram Protocol, yani Kullanıcı Veri Protokolü), IP üzerinden veri yollamaya yarar. Verilerin ulaşacağını garanti etmez ve UDP paketlerinin maksimum boy sınırları vardır. Öte yandan, UDP son derece basit ve bağlantı gerektirmeyen (connectionless) bir protokoldür.

• **TCP** (Transmission Control Protocol, yani Gönderim Kontrol Protokolü), IP üzerinden ulaşma garantili ve harhangi bir boyda veri gönderilmesine imkan tanıyan bir protokoldür. UDP'den farklı olarak, TCP'de iki cihazın iletişim kurabilmesi için önce birbirlerine bağlanmaları gerekmektedir.

7. 3. 4 Uygulama Katmanındaki Protokoller

• **DNS** (Domain Name System, yani Alan Adı Sistemi) alan adı verilen isimler (mesela www. wikipedia. org) ile IP adreslerini birbirine bağlayan sistemdir. Paylaştırılmış bir veritabanı olarak çalışır. UDP veya TCP üzerinden çalışabilir.

• **HTTP** (HyperText Transfer Protocol, yani HiperMetin Yollama Protokolü) ilk başta HTML sayfaları yollamak için yaratılmış olan bir protokol olup günümüzde her türlü verinin gönderimi için kullanılır. TCP üzerinden çalışır.

• **HTTPS** (Secure HTTP yani Güvenli HTTP) HTTP'nin RSA şifrelemesi ile güçlendirilmiş halidir. TCP üzerinden çalışır.

• **POP3** (Post Office Protocol 3, yani Postahane Protokolü 3) e-posta almak için kullanılan bir protokoldür. TCP üzerinden çalışır.

• **SMTP** (Simple Mail Transfer Protocol, yani Basit Mektup Gönderme Protokolü) e-posta göndermek için kullanılır. TCP üzerinden çalışır.

• **FTP** (File Transfer Protocol, yani Dosya Gönderme Protokolü) dosya göndermek ve almak için kullanılır. HTTP'den değişik olarak kullanıcının illa ki sisteme giriş yapmasını gerektirir. Veri ve komut alış verişi için iki ayrı port kullanır. TCP üzerinden çalışır.

• **SFTP** veya FTPS (Secure FTP, yani Güvenli FTP), FTP'nin RSA ile güçlendirilmiş halidir. TCP üzerinden çalışır.

Tüm bu protokoller (ve dahası) sayesinde TCP/IP her geçen gün daha da popülerleşen bir protokol olmuştur. (David, M, 1995)

7. 4 İnetd

İnetd tüm daemon'ların anasıdır. Daemon'lar sistemde devamlı olarak çalışan ve diğer prosesleri dinleyen programlardır. Microsoft DOS platformundaki terminate and stay resident TSR programlarına benzerler. (TSR genellikle virüsler tarafından çok kullanılan bir yöntemdi. Virüs kodunun sürekli hafızada aktif olarak kalabilmesi için TSR metodu kullanılıyordu.) Daemonlar sistem açık olduğu sürece belli bir olayı dinlemek için sürekli çalışır durumdadırlar. İşte süper sunucu olarak ta çağrılan İnetd tüm bu daemonların büyük büyük babasıdır.

Tahmin edebileceğiniz gibi bir sistemde ne kadar çok daemon varsa o kadar çok sistem kaynakları azalacaktır. İşte her türlü işlemi gerçekleştirmek için bir daemonu her zaman çalışır durumda bekletmek ve sistem kaynaklarını yemek yerine bir tane daemon yazmışlar. Bu da İnetd daemonudur. İnetd tüm ağ isteklerini dinler ve bir istek geldiğinde isteğe bakarak hangi servisle ilgili olduğuna karar verir. Daha sonra da ilgili servisi sunan uygulamayı yükleyerek isteği bu uygulamaya yönlendirir. Örnek olarak

bir FTP isteği geldiği zaman İnetd FTP sunucusunu başlatır ve isteğe cevap vermesini ister ve kendisi de başka isteklere cevap vermek üzere dinlemeye devam eder.

İnetd sadece UNIX üzerinde çalışan bir uygulama değildir. Windows ortamında çalışan sürümlerini de piyasada bulmak mümkündür. Hummingbird'ün Exceed ürünü Windows ve OS/2 platformlarI için İnetd'yi sunmaktadır.

İnetd programı normal olarak sistem açildiginda çalışmaya başlar ve sistem yöneticisi tarafından kapatılmadığı sürece sistem kapatılana kadar da çalışmaya devam eder. İnetd programının çalışması /etc/İnetd. conf konfigürasyon dosyası ile tanımlanır. İnetd'nin hangi servisleri sunacağı bu dosya da belirtilir. Bu servisler FTP, Telnet, SMTP, Finger, Netstat. . gibi servislerdir.

7. 5 Portlar

TCP/IP ortamında programların çalıştırılması ve servisler genellikle istemci-sunucu tabanlıdır. Her bağlantı isteği için İnetd bir sunucu çalıştırır ve sunucu da istemciyle haberlesmeye başlar.

Bu işlemi gerçeklestirebilmek için her servise (FTP, Telnet. . gibi) bir numara verilmiştir. İşte istemciler bu numaraları kullanarak karşı bilgisayardaki hangi uygulamayla konusacağını belirtir. Bu numaralar port numaraları olarak adlandırılır. Bir internet sunucusunda binlerce port olabilir. Ancak etkin bir kullanım için iyi bilinen ve her zaman kullanılan servislere standart port numaraları verilmiştir. Sistem yöneticisi istediği servisi istediği port numarasına bağlayabilir ancak normal olarak iyi bilinen port numaraları (well-known ports) kullanmak akıllıca olacaktır. Örnek olarak asağıda bazı servislerin standart port numaraları verilmiştir:

Dosya Transfer Protokolü (FTP) 21

Telnet 23

Simple mail transfer protokol (SMTP) 25

Gopher 70

Finger 79

HTTP 80

NNTP 119

7. 6 Telnet

Telnet uzak sistemlere login olmak ve sistemde komut çalıştırmak için kullanılır. Ankarada bulunan bir kullanıcı İstanbul'da bulunan bir makineye telnet yaparak sanki makinenin başındaymış gibi komutlar çalıştırabilir. Bir telnet oturumu açmak için UNIX komut satirindan ya da DOS komut satırından:

#telnet sunucu_adı

Komutu girilir ve eğer bu sunucuda telnet sunucusu çalışıyorsa kullanıcının karşısına login ekranı gelecektir. Bu ekranda kullanıcı adı ve şifresi girildikten sonra sisteme oturum açılacaktır. Telnet protokolü text tabanlı olup UNIX sisteminde ve çoğu sistemde dahili olarak gelmektedir.

7. 7 IP Adresi

Belli bir ağa bağlı cihazların ağ üzerinden birbirlerine veri yollamak için kullandıkları adrestir. İngilizce'deki Internet Protocol address teriminin kısaltmasıdır.

İnternet'e bağlanan her bilgisayara bir IP adresi atanır, diğer bilgisayarlar bu bilgisayara bu adres ile ulaşırlar. Yani iki farklı cihaz aynı yerel ağda olmasa dahi, IP adresi birbirleri ile iletişim imkanı sağlar.

IP adresleri şu anda yaygın kullanımda olan IPv4 için 32 bit boyunda olup, noktalarla ayrılmış 4 adet 8 bitlik sayıyla gösterilirler. Örneğin: 192. 168. 10. 9

Bir internet sayfası sunucusuna, web tarayıcısına IP adresi yazarak da bağlanılabilir; ancak bu rakamları yazmak pratik olmadığından IP adresine karşılık gelen bir alan adı sistemi kullanılmaktadır. İnternet Servis Sağlayıcılarında bulunan Alan Adı Sunucularından (DNS -Domain Name System) oluşan bir ağ, hangi alan adının hangi IP adresine karşılık geldiği bilgisini eşler ve kullanıcıları doğru adreslere yönlendirir. İnternet'te trafik Başlıca IP adreslerince sağlanmaktadır.

IP adresi, internet'e bağlıyken Windows 98'de Başlat'taki "Çalıştır" satırına "winipcfg" yazarak öğrenilebilir. Daha yeni Windows sürümlerindeyse komut satırında ipconfig yazarak öğrenilebilir. (Postel, J, 1981)

7. 7. 1 Dinamik ve Statik adresler

Dinamik IP adresi, İnternet Servis Sağlayıcı (ISP) tarafından kullanıcıya her internete bağlandığında geçici olarak tayin edilen bir IP adresidir. Büyük bir ihtimalle, sizin IP adresiniz de dinamiktir. İnternet bağlantınızı kesip tekrar bağlanarak bu siteyi yeniden ziyaret ederseniz, IP numaranızın değiştiğini görebilirsiniz. Çoğu bireysel kullanıcının IP adresi bu şekilde dinamiktir

Statik IP adresi, servis sağlayıcı tarafından verilen ve hiç değişmeyen bir adresdir. İnternet'teki her bilgisayarın bir adresi vardır ve bu adres IP numarası ile belirlenir. Örneğin İnternet sitelerinin önemli bir bölümünün adresi statiktir. Pratik açıdan İnternet kullanıcılarının ip adreslerinin statik olmasına pek gerek yoktur. Genelde sunucu görevi gören bilgisayarlar için tercih edilir.

8. İNTERNETİN DOĞUŞU (En Geniş Ağ)

İnternet tam anlamıyla ağlar arası ağdır. Bu kavramı açmak gerekirse büyük küçük binlerce ağın birleşmesinden oluşmuş en büyük ağdır. Bir kişiye, kuruluşa, ülkeye özel değildir.

İnternet kavramı aslında 1969'da savaş sonrasında düşünülen DARPA (Defense Advanced Research Project Agency - İleri Düzey Savunma Araştırmaları Kurumu) isimli basit bir projeden ibaretti. Bu proje büyük bilgisayarları birbirine bağlamayı ve ne olursa olsun bu bağı koparmamayı amaçlıyordu. Klasik bir ağ tarzında, bu ağdaki tek bir bağlantının kopması veya ana sunucunun imha edilmesi durumunda bu ağ çökecektir. Bu yüzden teknisyenler istemci-sunucu modeli yerine her bilgisayarın birbirine eşit özelliklerde olduğu türdeş ağ modeli tercih ettiler. İlk bağlantı California ve Utah'ta olan 4 bilgisayar arasında idi. Yavaş yavaş üniversitelerin de bağlanmasıyla ağ giderek büyümeye başladı. Bu proje daha sonra ARPANET (Advanced Research Projects Agency Network) adını aldı. Sivil kişi ve kuruluşların da bağlanmasıyla tüm Amerika'yı kapsamaya başladı. ARPANET in beklenenden fazla büyümesiyle askeri kısmı MILNET adıyla ayrıldı ve daha sonra da ARPANET gelişerek bügünkü adıyla INTERNET adını aldı. (http://yunus.hacettepe.edu.tr/)

8. 1 İnternet'in Tarihçesi

İnternet'in yaygınlaşmasıyla birlikte TCP/IP kısaltmasını çok sık duymaya başladık. TCP/IP, Transmission Control Protocol/Internet Protocol ifadesinin kısaltması. Türkçesi, İletim Kontrolü/İnternet Protokolü oluyor. Protokol belli bir işi düzenleyen kurallar dizisi demek. Örneğin, devlet protokolü devlet erkanının nerede duracağını, nasıl oturup kalkacağını düzenler. Ağ protokolleri de bilgisayarlar arası bağlantıyı, iletişimi düzenliyor.

TCP/IP'nin kökleri, 1960'ların sonunda 1970'lerin başında Amerikan Savunma Bakanlığı'na bağlı İleri Araştırma Projeleri Ajansının (Advanced Research Projects Agecncy, ARPA) yürüttüğü paket anahtarlamalı ağ deneylerine kadar uzanır. TCP/IP'nin yaratılmasını sağlayan proje ABD'deki bilgisayarların bir felaket anında da ayakta kalabilmesini, birbirleriyle iletişimin devam etmesini amaçlıyordu. Şimdi

baktığımız zaman projenin fazlasıyla amacına ulaştığını ve daha başka şeyleri de başardığını görüyoruz.

Bu projenin ilk aşamasında, 1970'de ARPANET bilgisayarları Network Control Protocol'ünü kullanmaya başladılar. 1972'de ilk telnet spesifikasyonu tanımlandı. 1973'de FTP (File Transfer Protocol) tanımlandı. 1974'te Transmission Control Program ayrıntılı bir şekilde tanımlandı. 1981'de IP standartı yayımlandı. 1982'de Defence Communications Agency (DCA) ve ARPA, TCP ile IP'yi TCP/IP Protokol suiti olarak tanımladı. 1983'de, ARPANET NCT'den TCP/IP'ye geçti. 1984'de Domain Name System (DNS) tanıtıldı.

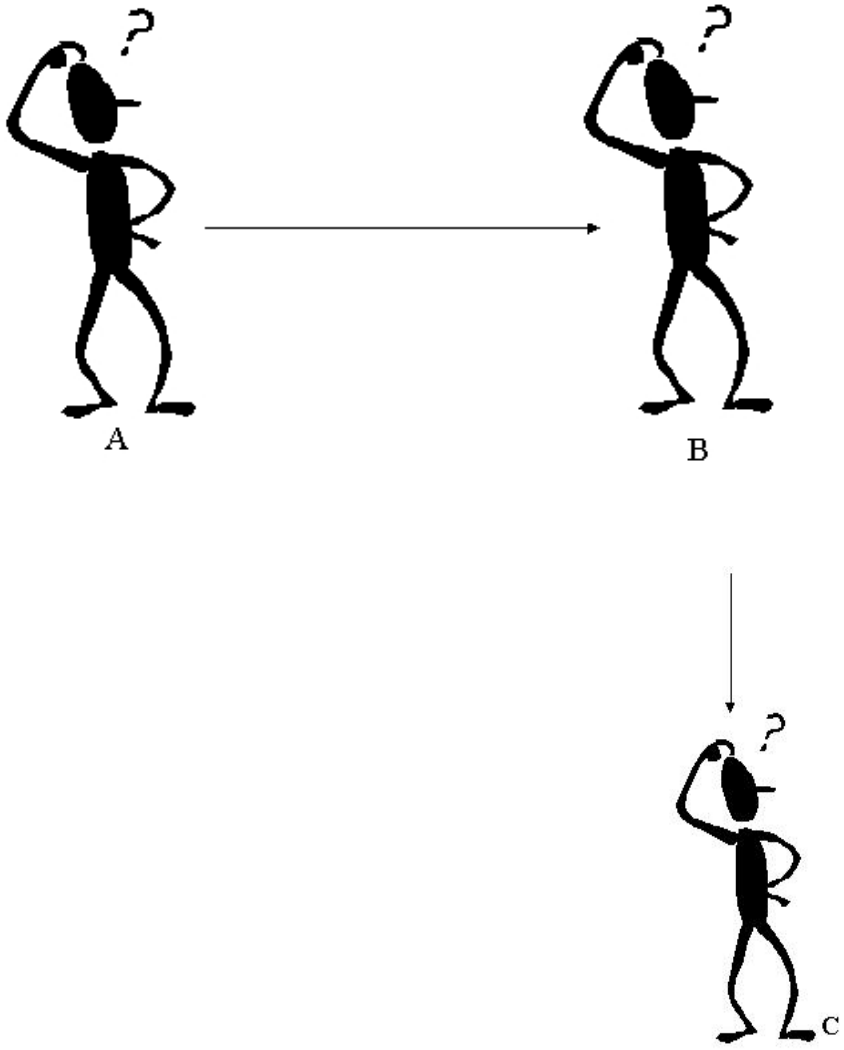
Yukarıda kısaca verdiğimiz tarihçe aynı zamanda Internet'in tarihçesidir. Internet ile TCP/IP ayrılmaz kardeşlerdir. TCP/IP, İnternet'in temelidir.

8. 2 TCP/IP Dünyasında Bir Bilgisayarı Belirleyen 3 Şey

Bunlar bilgisayarın adı, IP adresi, MAC adresi'dir. Bir bilgisayarın MAC adresini ya da IP adresini değil de adını kullanmak daha kolay değil mi? Aksi takdirde, bilgisayarların IP adreslerini, daha da kötüsü MAC adreslerini ezberlemek zorunda kalabilirdik.

Bilgisayar adını kullanmak kolayımıza geliyor ama, ağ üzerinde iletişim gerçekte MAC adresleri üzerinden gerçekleştiriliyor. O zaman bilgisayar adını önce IP adresine çeviren sonra da MAC adresine çeviren mekanizmalar, protokoller olmalı değil mi? IP adresini MAC adresine çeviren protokolü görmüştük (belleği zayıf olanlara anımsatalım; bu protokolün adı ARP idi). Peki, bilgisayar adları IP adreslerine nasıl çeviriliyor? Burada çeşitli seçenekler var. Microsoft'un önerdiği şey Windows Internet Adlandırma Servisidir (WINS). Bu servis ile bir makinayı WINS sunucusu olarak tanımlıyoruz, bütün bilgisayarlar girip adlarını ve IP adreslerini bu sunucuya bildirir. (Aynen yeni eve taşındığımızda hane halkının mahallenin muhtarına kaydolması gibi). Bir bilgisayar, adını bildiği bir bilgisayarın IP adresini bulmak istediği zaman, broadcast yapmak yerine bu sunucuya gidiyor "Şu ağdaki bilgisayarın IP adresi nedir?" şeklinde bir soru soruyor. WINS sunucu da kendi veritabanına bakıp soruyu yanıtlıyor.

9. NETWORKİNG NEDİR?



Şekil 9. 1 Networking Şekili

"Networking" fikri telekomünikasyon kadar eskidir. Taş devrinde yaşayan insanları düşünelim.

Davullar bireyler arasında iletişim olarak kullanılmaktadır. Varsayalım Mağara Adamı A, C'yi taş yuvarlama oyunu için çağırmak istemektedir, fakat C, A'nın davul vuruşunu duyamayacak kadar uzakta yaşamaktadır.

Bu durumda A'nın İkisinin arasında yaşayan B'den C'ye mesajı iletmesini istemeye "networking" denir.

10. SNİFFER NEDİR VE NASIL ÇALIŞIR?

Sniffer denilen şey, network kartınızı seçici olmayan moda geçirerek o kartla ilgili olmasa da gelen paketleri alıp incelememizi sağlayan programlardır. Paylaşım, bir bilgisayardaki bilgilerin, başka bir bilgisayara aktarılması olarak açıklanabilir. Bu iki bilgisayar arasında yapılan bilgi alış-veriş'ini yakalamaya "sniffing" denilir. Bir kaç bilgisayarın, bir ağ üzerinde birbirleriyle paylaşıma açık olarak bağlanılmasında kullanılan en popüler yol "ethernet" dir. Gönderilen paketin başlığında, paketin gideceği bilgisayarın adresi yazar. Sadece bu paketteki adres ile adresi tutan makine bu bilgileri alabilir.

Her paketi kabul eden bir makine, paket başlığındaki adrese aldırmayan makine, çok karışık bir hal alacaktır. Normal bir networkte, account ve şifreler, ethernet üzerinde düzgün bir yazıyla (encrypt edilmemiş) gidip gelirler. Bir ziyaretçi, ethernet üzerindeki herhangi bir makineden root yetkisi elde ederse, sistemi sniffleyerek ağ üzerinde ki diğer makinelerde ne gibi işlemler yapıldığını belirleyip ağ trafiğini analiz edebilir.

Sniffer denilen şey, network kartınızı promiscuous mode'a geçirip o karta o kartla ilgili olmasa da gelen paketlari alıp incelemenizi (ya da genel kullanımını düsünürsek kullanıcıların passwordlerini dahi görebilmenizi) sağlayan bir programdır.

Aynı hub'a bağlıysanız ve bu hub biraz aptalsa (switch değilse diyelim) gelen bir packet'i bütün portlarına yollar. Öyle olunca (10 Mbitlik bir hub'i dusunursek) hub'in bir portu değil tamamı 10 Mbit olur. Ve sizin kartınıza gelen sizle ilgili olmayan paketleri de kartınız promiscuous mode'a geçebiliyorsa görülebilir.

Bunu engellemenin üç yolu vardır.

* Birinci yolu switch kullanmak.

* İkincisi piyasada promiscouos mode'a geçmeyen kartlar var.

* Üçüncüsü de mesela linux kullanıyorsanız kernel'dan drivers/net altında kullandığınız kart ile ilgili olan promisc mode'a geçiren satırları uncomment edip tekrar

derlemeniz. Mesela ne2000 clone'u kullanıyorsanız 8390. c'yi değiştirip flag'in promisc mode'a geçtiğini kontrol eden satırın altındaki outb_p satırını commentlerseniz bunu yapabilirsiniz.

Bu güvenlik için istenen birşeydir. Ama genelde (mesela network adminleri) snifferlari ağı analiz etmek için de kullanabilirler.

10. 1 Promiscious Mode Nedir?

Normalde bir ağ arabirimi sadece hedef adresi kendisini gösteren paketlerle ilgilenir, diğer paketleri önemsemez. Promisc modda ise kendisine gelen her paketi kime yollandığına bakmadan kabul eder, hatırlayacak olursak hub tipi ağ aygıtlarındaki iletişim ortak bir havuzda gerçekleşiyordu yani huba bağlı 8 makinemiz varsa bu 8 makine arasındaki her türlü iletişim diğerleri tarafından da izlenebiliyordu.

3 adet makine birbirine hub ile bağlı olsun. A makinesindeki Temel reis B makinesindekı safinaza "senı sevıyorum" mesajı gonderıyor. Kaba Sakal da Temel Reis ile Safinaz arasındaki geçen yazışmayı izlemek istiyor.

Kaba sakal bulundukları ortamın hub olduğunu bildiği için Ethernet kartını promisc moda geçiriyor ve Temel Reis ile Safinaz arasındaki trafiği dinliyor ve internette yaptığı kısa bir araştırma neticesinde Temel Reis ile Safinaz'ın iletişiminde araya girerek Temel Reis'in Safinaz'a yolladığı paketleri istediği gibi değiştirebiliyor.

Ethernet kartları sıfır yapılandırma ile promisc özelliğine sahip değildirler, ethernet arabirimimizi normal moddan promisc moda geçirmek için ifconfig komutuna promisc parametresini vermemiz yeterlidir.

 # ifconfig

eth0 Link encap:Ethernet HWaddr 00:D0:B7:B6:D1:0C
inet addr:194. 27. 72. 88 Bcast:194. 27. 127. 255 Mask:255. 255. 192. 0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:5228531 errors:0 dropped:0 overruns:0 frame:0
TX packets:4528739 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000

RX bytes:1796789472 (1713. 5 Mb) TX bytes:3725692 (3. 5 Mb)
Interrupt:18 Base address:0x5400 Memory:f6101000-f6101038

 # ifconfig eth0 promisc

 # ifconfig

eth0 Link encap:Ethernet HWaddr 00:D0:B7:B6:D1:0C
inet addr:194. 27. 72. 88 Bcast:194. 27. 127. 255 Mask:255. 255. 192. 0
UP BROADCAST RUNNING **PROMISC** MULTICAST MTU:1500 Metric:1
RX packets:5228715 errors:0 dropped:0 overruns:0 frame:0
TX packets:4528864 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1796807077 (1713. 5 Mb) TX bytes:3737015 (3. 5 Mb)
Interrupt:18 Base address:0x5400 Memory:f6101000-f6101038

Yukarıdaki farklılıktan (PROMISC) da görebileceğimiz gibi ifconfig komutuna promisc parametresini ekleyince ethernet kartımızı promisc moda geçirmiş oluruz. Promisc moddan çıkarmak istediğimizde ise

 # ifconfig eth0 –promisc komutunu vermemiz yeterlidir.

10. 2 UNIX Sistemlerde Çalışan Snifferlara Örnekler

Ethereal	programcı:600 programcısı var
Network Traffic Analyser	programcı: Marko Zivanovic
jNetStream	programcı: Mark
KSnuffle	programcı: Mike Richardson
imsniff	programcı: carlos. fernandez
tcptrack	programcı: Steve Benson
etherdump	programcı: Peter Willis
dietsniff	programcı: Hynek Schlawack

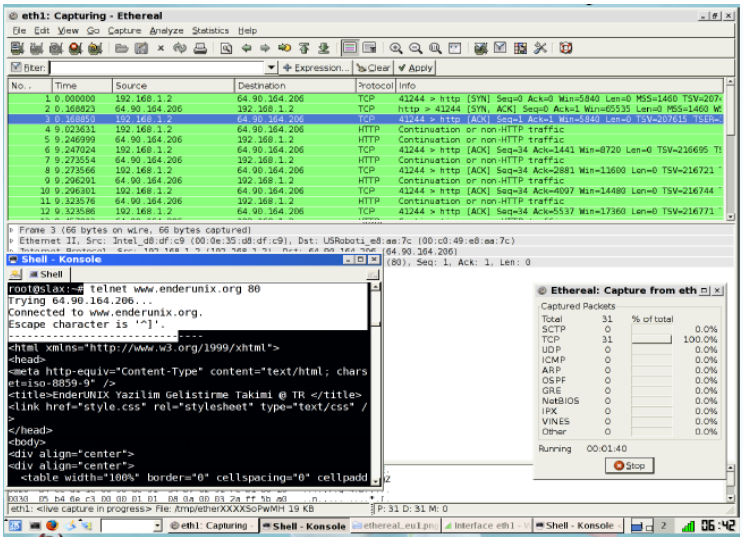
Rkdet	programcı: Andrew Daviel
serialsnoop	programcı: ken restivo
Tvark	programcı: Zak Johnson
Wireshark	programcı: Gerald Combs
Network-I	programcı: ilovevi
tcpick	programcı: Francesco Stablum
angst	programcı: Patroklos G. Argyroudis
Nast	programcı: embyte
Knetdump	programcı: Norbert Weuster
Advanced Packet Sniffer	programcı: Christian Schulte
Ettercap	programcı: ALoR NaGA
Perl Advanced TCP Hijacking	programcı: Bastian Ballmann
Packet Excalibur	programcı: jitsu
Impost	programcı: sickbeatz
Snort	programcı: Martin Roesch

Yukarıda yazılan sniffer'ların hepsi paket yakalamak için libpcap arayüzünü kullanırlar. Daha sonra yakalanan bu paketleri kendilerine göre işleme sokarak programa lazım olan verileri alırlar.

Bu yazılımlardan en çok bilineni ethereal'dir. Bende çalışmamı yapmadan önce ethereal'ı kurup onunla ilgili özellikleri, neler yapabildiklerini inceledim. Böylece kendi yazacağım program için önemli bilgiler edindim.

Ethereal, Windows, Linux, MacOS veya Solaris gibi bir çok işletim sisteminde çalışabilen ve bilgisayara bağlı olan her türlü ağ kartlarındaki (Ethernet kartı veya modem) tüm TCP/IP mesajlarını analiz edebilen bir programdır. (Ramirez, G, 2005)

- 500'nin üzerinde protokolü analiz edebilir
- Paketleri yakalayıp bir dosyaya kaydedebilir
- Daha önceden kaydedilmiş bir dosyayı açabilir
- Gerçek zamanlı analiz yapabilir
- Bir analizi filtre edebilir (örneğin "sadece HTTP mesajlarını göster" gibi)
- Terminal veya kullanıcı arabirimi ile kullanılabilir



Şekil 10. 1 Ethereal'in Çalıştırılmış Şekili

11. YAZILIMIN GERÇEKLEŞTİRİLMESİ

11. 1 Genel Tanıtım

Bu çalışmada gerçekleştirmeye çalıştığımız yazılımın, UNIX sistemde çalışması, belli bir network'teki makinelerin her birinin ağ üzerinde yaptığı işlemleri tespit etmesi gerekmektedir. Bunu yapmak içinde makineler arasında giden paketlerin yakalanıp incelenmesi gerekir. Böyle bir yazılım için (bir UNIX sistemde çalışacağı için) libpcap kütüphanelerini kullanılmıştır. Şayet Windows sistemlerde çalışacak olsa idi libpcap'ın Windows versionu olan winpcap'ı kullanmamız gerekecekti.

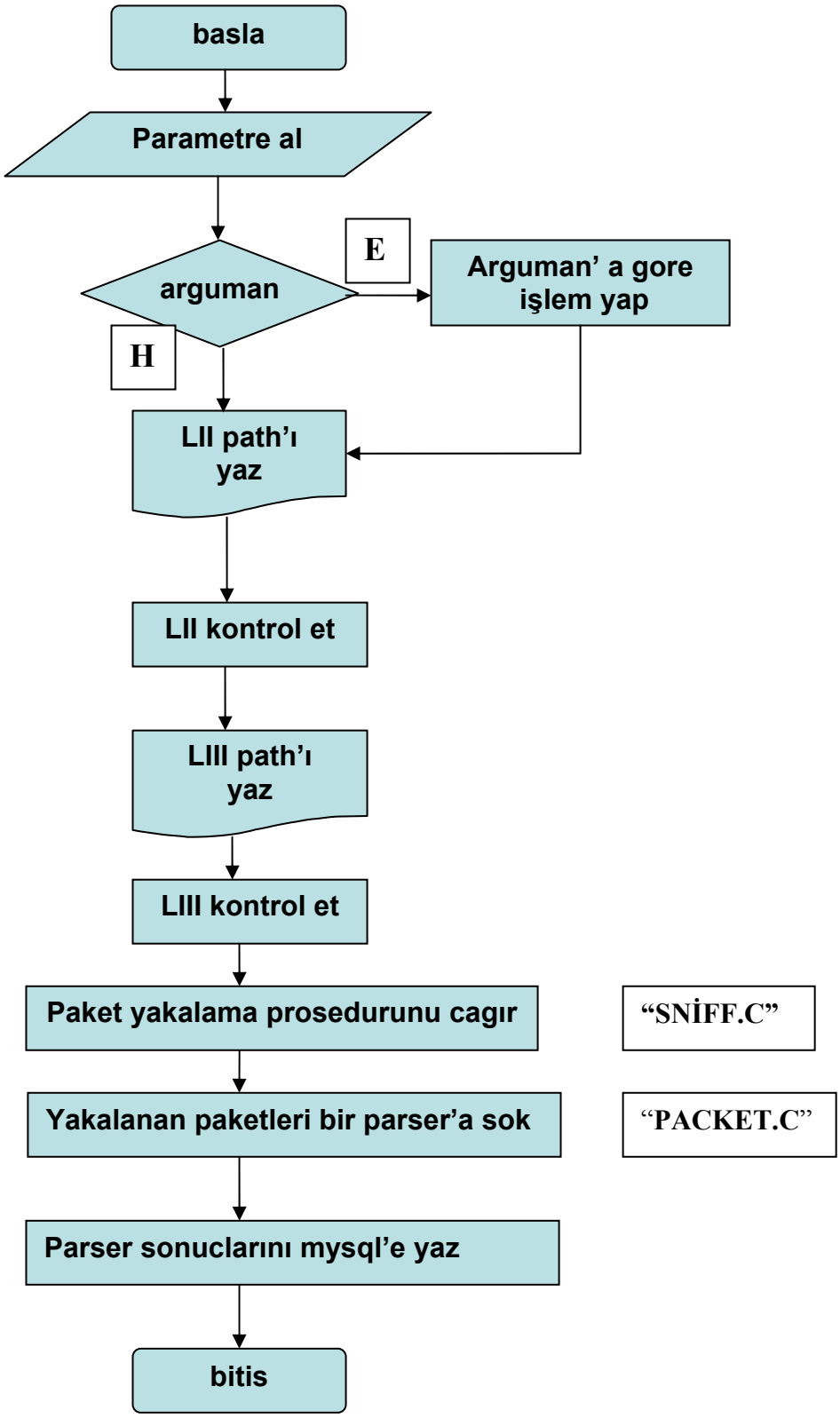
Yazılımı geliştirirken kullandığımız ortam işletim sistemi olarak Red hat Linux ve geliştirme aracı olarak'ta C dili kullanılmıştır. Gidip gelen paketleri yakalamak için piyasadaki diğer sniffer'lar gibi libpcap arayüzü kullanılmıştır.

Libpcap'den alınan paketler yazılmış olan parser'a sokularak istediğimiz veriler (zamanı, paket boyutu, kullanıcı ipkaynak ip, protocol tipi) alınır ve bu veriler mysql veritabanına yüklenir. Daha sonra diğer pakete geçilir. Onun içinde aynı işlemler yapılır. Bu döngü program sonlandırılana kadar devam eder.

11. 2 Yazılımın Genel Veri Akış Diyagramı

Yazılımın genelinde, ilk önce kodun parametre alıp almadığına bakılır, almış ise onunla ilgili işlem yapılır. Sonra KB dizini içindeki LII ve LIII kontrol edilip path'leri yazılır. Ardından libpcap kütüphanesin'deki pcap fonksiyonu kullanılarak paket yakalanmaya başlanır. Yakalanan bu paketler bir parser'a sokularak bize gerekli olan veriler alınır ve mysql içindeki yakalanan isimli veritabanına yazılır.

Aşağıda ki şekilde yazılımın genel akış diyagramı gösterilmiştir.



Şekil 11. 1 Yazılımın Veri Akış Diyagramı

Yukarıda paramatre al kısmında alabileceği parametreler ve yapılacak işlemler şunlardır.

-h bunun sonucunda kullanabileceğiniz opsiyonlar karşınıza gelecektir.

-i paketleri yakalamak için istediğimiz bi aygıtı kullanabiliriz. Örneğin birden fazla ethernet kartımız olduğunda işimize yarar.

-k protokol dosyalarını ekrana basar.

-d debug bilgisini ekranda gösteririr.

-p kullandığımız cihazı promiscous moda almamızı sağlar. Promisc modu sadece kendi makinemize gelen paketlerle değil, bütün paketlerle ilgilendiğimiz zaman kullanabiliriz.

-n yakalanacak olan paket sayısı. Bunu kullanarak şu kadar paketten sonra programı sonlandır diyebiliriz.

-v versionu yazdırır.

Eğer programı herhangi bir opsiyon olmadan kullanırsanız program kendini non-promisc modda çalıştırıp yakaladığı bütün paketleri yorumlamaya başlar. Kullanımıda şu şekildedir.

mesut –h gibi

mesut –n 100 (100 tane paket yakalayınca durmasını sağlar) gibi.

11. 3 Libpcap İle Paket Yakalanması

Libpcap Kütüphanesi (paket alma)

Ağ'dan paket alıp bu paketin bilgilerini değerlendirme işlemi. Sistemden bağımsız bir standart ile bu işlemleri libpcap kütüphanesinin sağladığı imkanlarla gerçekleştirebiliriz.

Libpcap'da paket alma işlemleri belli bir filtreleme yöntemi ile yapılmaktadır. Şöyle ki, ağ'dan alınacak paketlerin protokolu, port bilgileri, ip bilgileri bu filtrelerde tanımlanıp sadece bu kurallara uyan paketler kabul edilir. Böylece, uygulama için gerekli olmayan paketlerin, program içinde tekrar elden geçirilmesi engellenmiş olur.

libpcap'i bir örnek ile anlatmaya çalışalım:

```
int main (int argc, char **argv)
{
        pcap_t *handle;
        char *dev;                          /* dinlemeyi yapacağımız arayüz */
        char errbuf[PCAP_ERRBUF_SIZE];      /* herhangi hatayı geri
döndürdüğümüz string */
        struct bpf_program filter;          /* derlenmiş filtre */
zz
        bpf_u_int32 netmask;                /* netmask adresimiz */
        bpf_u_int32 ip_addr;                /* ip adresimiz */
        struct pcap_pkthdr header;          /* pcap'in bize sağladığı başlık bilgisi */
        const u_char *packet;               /* ve paketimizin kendisi */


/* Eğer belirli bir arayüzümüz yoksa mevcut arayüzü otomatik olarak  sistemden
almamız gerekmektedir. */
          dev = pcap_lookupdev (errbuf) ;

 /* artık bu arayüzün netmask ve ip adres gibi network bilgilerini alalım. */
          pcap_lookupnet (dev, &ip_addr, &netmask, errbuf) ;

     /* ve şimdi arayüzümüzü promiscous modda açıyoruz  pcap_open_live
foksiyonundaki ilk parametre aygıt ismi, ikinci parametre ise bir seferde okunacak paket
büyüklüğü, üçüncü parametre aygıtın promiscous modda mı çalışacağı ile ilgilidir.
Dördüncü parametre is aygıttan okuma ile ilgili süre aşımı değeridir. Ve son olarak
herhangi bir hata elde edersek hata bilgisini koyacağımız arabellek. */

          handle = pcap_open_live (dev, BUFSIZ, 1, 0, errbuf) ;
```

```
        if (!handle) {
                fprintf (stderr, "Arayüz açılamadı: %s", errbuf) ;
                exit (1) ;
        }
    /* belirttiğimiz kurallarla filtreyi derliyoruz */
        pcap_compile (handle, &filter, filter_app, 0, ip_addr) ;
        pcap_setfilter (handle, &filter) ;

    /* networkten bu kurallara uyan bir paket alıyoruz */
        packet = pcap_next (handle, &header) ;

        /* en basit işlem olarak aldiğımız paket üzerinde başlık bilgisini  yazdıralım */
        printf ("%d büyüklüğünde bir paket alındı\n", header. len) ;
        pcap_close (handle) ;
        return 0;
}
```

Görüldüğü üzere, libpcap kütüphanesini kullanarak belli filtrelere göre paket almak belli başlı bir kaç süreç dahilinde olmaktadır. Bunlar kısaca şu şekilde özetlenebilir:

1) Dinleme işleminin yapılacağı network arayüzünün belirlenmesi

Bu iki yolla yapilabilir. Statik olarak arayüzü belirtme suretiyle, diğeri de otomatik olarak sistemin bize sağlamasıyla

2) Network arayüzünün bilgilerinin alınması

3) Network arayüzüyle bağlantı kurulması

4) Filtrelerde belirttiğimiz kuralların derlenip arayüz ile kurulan bağlantıya filtrenin uygulanması

5) Paket alma işlemleri

6) Ve son olarak da arayüze yapılan bağlantının kapatılması

Yukarıdaki işlemlerden 4 numaralı işlemi uygulamaz isek, yani arayüze yaptığımız bağlantı için herhangi kural sağlamaz isek, arayüz ağ'dan bütün paketleri alır. Bir nevi datalink katmanına soket olusturmuş olunur.

Yani paketimiz ethernet başlık bilgisinden itibaren oluşmaktadır. Bu da bizim bu paketi daha esnek kullanmamızı ve daha alt seviyede paket kabul edilmesini sağlar.

Libpcap'in ağ katmanlarının ikinci seviyesinde işlem yapabilme özelliği programcının hem tasınabilir hem de daha alt seviyelerde program yazmasını cok kolaylaştırmaktadır. Sistemlerin birbirinden farklılıklarıyla uğraşmazken, hem de o sistemlerin size sağlayabileceği en esnek paket işlemlerini gerçekleştirebilirsiniz.

11. 4 KB Klasörü İçindeki Protokol Dosyalarının Yazılması

KB dizini içerisine yakalanmasını istediğimiz protokollerin tanımlamasını yapmalıyız.Bu protokollerin yazılması işlemini bir örnekle açıklayalım.

Protocol_Id=6
Yazılması zorunlu bir alandır. Bu protocolun tanımlayıcısı budur. Yani gelen bir paketin bu protocolde olduğunu protokol id'sinden anlarız.
Bu numarayı bir UNIX sistemde adres satırına şurayı"/usr/include/netinet/in. h" yazdığımızda önümüze gelen sayfadan bulabiliriz.

Orada IPPROTO_TCP nın protokol numarasını 0x6 olarak göreceksiniz.

Protocol_Ident=Transmission Control Protocol
Buraya istediğimizi yazabiliriz. Kendimiz nasıl olmasını istiyorsak. Mesela"Yine bir tcp paketi " bile diyebilirdik.
Ama anlaşılır birşey yazmayı öneririm. Aksi taktirde ilerde iletişim problemi olabilir.

Protocol_Length=20
Bu pakette ne kadar byte yer kapladığı.

Asağıdada bu protokolun field ları (alanları) yazıyor.
Field=Field Identification-Number of bits-Startbit-isProtocolIdentifier-will inet_ntoa be applied

Field=Source Port Number-16-0-0-0

Bu field source port number olarak adlandırılmış.
16 bit uzunluğunda, Startbit i 0, protocol ıdentıfier değeri 0, inet_ntoa fonksiyonuna da gerek duymaz. .

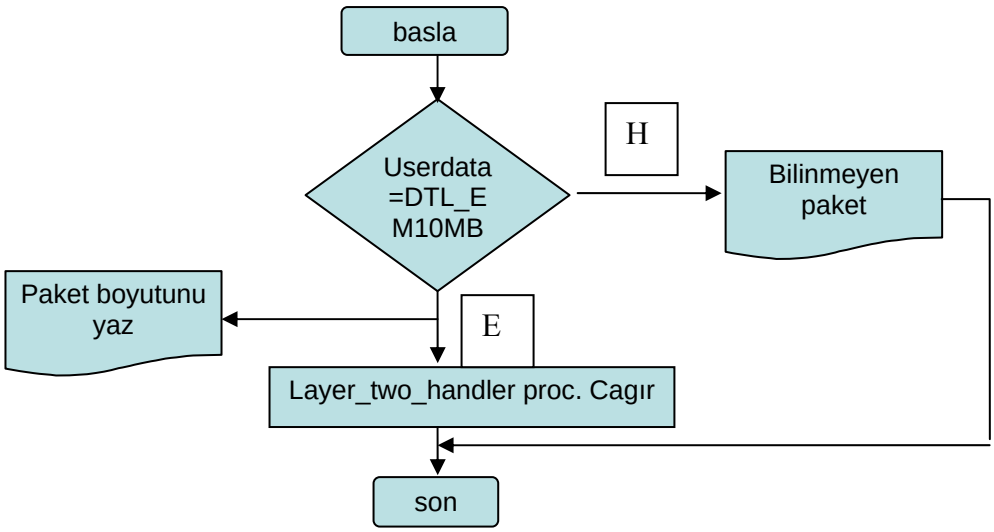
Asağıda KB klasörü içindeki tcp protokolü ile ilgili tanımlanmış bütün veriler gösterilmiştir.

Protocol_Id=6
Protocol_Ident=Transmission Control Protocol
Protocol_Length=20
Field=Source Port Number-16-0-0-0
Field=Destination Port Number-16-16-0-0
Field=Sequence Number-32-32-0-0
Field=Acknowledgement Number-32-64-0-0
Field=Header Length-4-96-0-0
Field=Reserved-6-100-0-0
Field=Flag URG-1-106-0-0
Field=Flag ACK-1-107-0-0
Field=Flag PUSH-1-108-0-0
Field=Flag RESET-1-109-0-0
Field=Flag SYN-1-110-0-0
Field=Flag FIN-1-111-0-0
Field=Window Size-16-112-0-0
Field=TCP Checksum-16-128-0-0
Field=Urgent Pointer-16-144-0-0

11. 5 Yakalanan Paketlerin Parser'a Sokulması

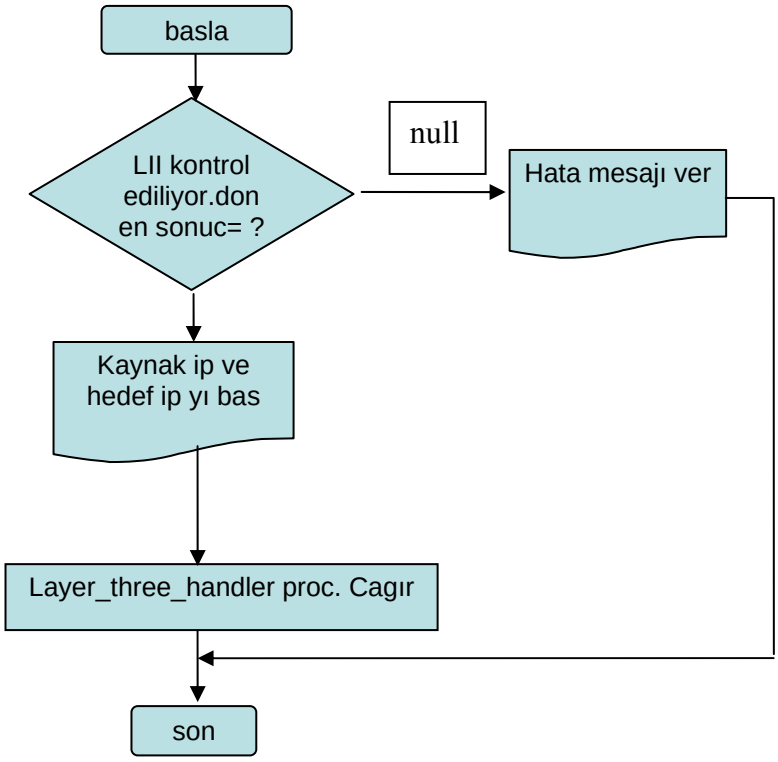
Bu yakalanan paketlerden istediğimiz verilerin alınması için demultiplexing işlemi yapılarak istediğimiz verileri bu paketlerin içinden alırız. Bu işlem yapılarak packet. c'nin içinde yapılır ve ekrana yazdırılır.

Yakalanan paket main_packet_handler fonksiyonuna sokulur. Burada paketin boyutu belli olur bu fonksiyonun içinde layer_two_handler fonksiyonu çağırılır. Bu fonksiyonda paketten kaynak ve kullanıcı ip bilgileri alınır. Bu fonksiyonun içinden de layer_three_handler fonksiyonu çağırılır. Bu fonksiyonda da paketteki protokol tipi belirlenir.



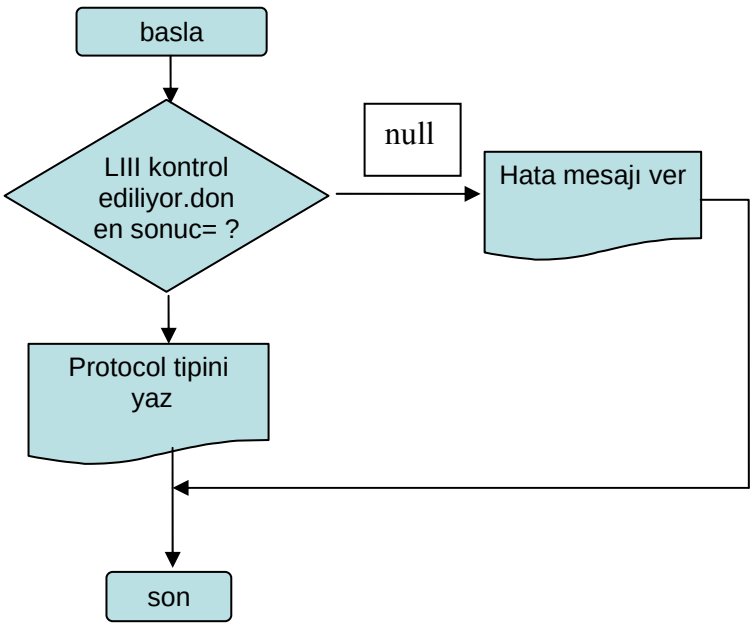
Şekil 11. 2 Main_packet_handler Fonksiyonu İçin Akış Şeması

Bu akış şemasında Main_packet_handler'da paket boyutunun nasıl alınıp, yazdırıldığı gösterilmiştir.



Şekil 11. 3 layer_two_handler Fonksiyonu İçin Akış Şeması

Bu akış şemasında da layer_two_handler ‘da kullanılan makinenin ip’si (kaynak ip) ve kullanıcı ip belirlenip ekrana yazdırılması gösterilmiştir.



Şekil 11. 4 layer_three_handler Fonksiyonu İçin Akiş Şeması

Bu akış şemasında da protokol'un layer_three_handler 'da nasıl yazdırıldığı gösterilmiştir.

11. 6 Elde Ettiğimiz Verilerin Veritabanına Yazılması

Bu programın büyük bir ağda çalıştırılması durumunda verileri sadece ekrana basarak görmemiz mümkün olmaz. Çünkü birbiri arkasına bir sürü veri basılacağı için eski basılanlar terminalden kaybolur ve bunlar uzerinde herhangi bir işlem yapmak veya hangi bilgisayarın nereye bağlandığını görme ihtimali kalmaz. Mesela ağda yoğun trafik yaratan kullanıcının kim olduğunu merak edilse bu kişiyi bulma ihtimali olmazdı. Bunun için bu aldığımız verileri (paket boyutu, kaynak ip, kullanıcı ip, protocolu, işlem zamanı) bir veritabanına yazdırma ihtayacı duyduk. Elde ettiğimiz verileri mysql de yakalanan adlı bir veritabanına atarak programın amacını yapması sağlanmıştır.

Mysql de yine ikinci bir veri tabanı yaratarak buraya da ip'leri sabit olan okuldaki 1400 kullanıcının ip'lerini ve isimlerini bir tabloya ekledik. Bu da bize herhangi bir ip numarasının kime ait olduğunu bu tablodan çekerek görmemize imkan vermiştir. Böylelikle program daha kullanışlı, görsel olarak daha iyi bir hale gelmiştir.

12. TARTIŞMA VE SONUÇ

Sonuç olarak, bu çalışmamızda UNIX sistemde çalışan paket analizi yapabilen ve bu analiz sonucu aldığı bilgileri veritabanına yazmayı amaçlayan bir yazılım gerçekleştirilmesine çalışılmıştır.

Yazılım text tabanlı olarak gerçekleştirilmiştir. Terminalde çalıştırılabilir. Real-time olarak ağ analizi yapabilir. Bu analiz sonucu aldığı verileri veritabanına yazar. Böylece ağ analizi ile ilgili daha etkili bir gözlem yapılabilmektedir.

Ağ analizi günümüzde, önemli bir çalışma alanıdır. Böyle bir sniffer internetten kolaylıkla indirilebilir. Ama bu yazılım onlardan bir yönüyle ayrılmaktadır. Bu da protokol dosyaları için ayrı ayrı C dosyaları hazırlanmamasıdır. Bunun yerine bir dizin içinde LII ve LII klasörleri açılarak bunların içine yakalanmasını istediğimiz protokollerin tanımlamasını yapılmıştır. Yakalanan paketlerin protokollerinin ne olduğu bu paketlerin bir yorumlayıcıya sokularak layer'ların içinde tanımlanan protokollerin ident'ları ile karşılaştırılarak bulunması sağlanmıştır. Böylelikle kod kalabalığından kurtulunmuş olundu.

Bu sebebten dolayı bu yazılıma bir nevi kişiselleştirilebilir network sniffer da denilebilir.

KAYNAKLAR

Cerf V., Kahn R., 1974, *A Protocol for Packet Network Intercommunication.*

Comer, D., 1995, *Internetworking with TCP/IP, Volume I: Principles, Protocols, and Architecture.*

Charles, T., 2004, *Lan Times Guide to Multimedia Networking.*

Çölkesen . R, 1999, *Network TCP/IP UNIX El kitabı.*

David, M., 1995, *TCP/IP networking : a guide to the IBM environment.*

Hornig C., 1984, *A Standard for the Transmission of IP Datagrams over Ethernet Networks, RFC-894.*

Jacobson, V., Braden, R., 1988, *TCP Extensions for Long-Delay Paths, " RFC-1072.*

İntroduction to UNIX / Middle East Technical University Computer Centr (METUCC).

Nagle, J., 1984, *Congestion Control in IP/TCP, RFC-896.*

Orebaugh, A., 2004, *Ethereal Packet Sniffing .*

Paul, G., 1995, *Security Considerations for IP Fragment Filtering, RFC 1858 http://rfc. dotsrc. org/rfc/rfc1858. html.*

Postel, j., 1981, *Internet Protocol (IP), " J. l, RFC-791.*

Ramirez, G., 2005, *Nessus, Snort, and Ethereal Power Tools : Customizing Open Source Security Applications.*

RESNICK, P., 2001, *Internet Message Format, RFC 2822.*

Stanger, J., 2001, *Hack Proofing Linux : A Guide to Open Source Security.*

Stevens, W., 1994, *TCP/IP Illustrated, Volume 1: The Protocols.*

Stine, R., 1990, *FYI: Network Management Tool Catalog, RFC 1147.*

Yıldırımoğlu, M., 2002, *TCP/IP internetin evrensel dili.*

Wellman, B., 1992, *Which types of ties and networks provide which types of support?.*

WEONER P., 1968, *Programming Languages, Information Structures, and Machine Organization.*

http://www. geocities. com/kayseriapk/teknik/ag. htm

http://tr. wikipedia. org/wiki/TCP/IP

http://www. turkcenet. org/yerel_htm/OSI. htm

http://www. cisco. com/univercd/cc/td/doc/cisintwk/ito_doc/introint. htm

http://ogrenci. hacettepe. edu. tr/~b0343623/bağlantılar/tcp-ip. htm

http://www. networksorcery. com/enp/protocol/tcp. htm

http://en. wikipedia. org/wiki/Sniffer

http://www. ethereal. com/

http://en. wikipedia. org/wiki/Packet_sniffing

http://www. mozilla. org/docs/web-developer/sniffer/browser_type. html

http://www. tcpdump. org/

http://sourceforge. net/projects/libpcap/

http://www. stearns. org/doc/pcap-apps. html

http://www. mysql. com/

http://forum. ceviz. net/

http://www. programlama. com/

http://www. linuxnet. com. tr/modules. php?name=Forums

http://www. manyaqforum. com/

http://yunus.hacettepe.edu.tr/

ÖZGEÇMİŞ

Mesut AKTOGAN, 1983 yılında Zonguldak'ta doğdu. İlk ve Ortaöğrenimini Zonguldak'ta tamamladıktan sonra Trakya Üniversitesi Mühendislik Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümünden 2004 yılında mezun oldu. Aynı yıl Trakya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda yüksek lisansa başladı. 2004-2005 yılları arasında Pınarhisar meslek yüksek okulunda söz. öğr. gör. olarak derslere girdi.