Network 101 — Hacker'ın gözünden network yapısı ve modern web (Kısım 1)

Mert Telli
May 16 · 35 min read

KISIM 1 — Network Kavramlarına Giriş

Öncelikle konuya girmeden, yani bir hacker'ın gözünden modern web kavramını anlatmadan önce bilgisayar ağları (computer network) konusuna ufak bir giriş yapalım, terminolojik bir bilgi aktaralım ve daha sonrasında ise Bir hacker'ın gözünden modern web nasıl gözüküyor, günümüzde modern web mimarisinde hangi bileşenler mevcut bunlardan bahsedeceğiz.

> *Alıntı yapılan kaynaklar, bölümler içinde ve kaynakça bölümünde belirtilecektir (herhangi bir haksızlık yapmamak adına yazıda kullandığım kaynakları eksiksiz bir şekilde belirtmeye çalıştım), konu hakkında daha detaylı bilgi almak isteyen kişiler kaynaklar üzerinde okuma/araştırma yapabilirler. İyi okumalar*

. . .

Bilgisayar Ağları nedir?

> ***Bilgisayar ağı**, küçük bir alan içerisindeki veya uzak mesafelerdeki bilgisayarların ve/veya iletişim cihazını iletişim hatları aracılığıyla birbirine bağlandığı, dolayısıyla bilgi ve sistem kaynaklarının farklı kullanıcılar tarafından paylaşıldığı, bir yerden başka bir yere veri aktarımının mümkün olduğu iletişim sistemi. (Kaynak: https://tr.wikipedia.org/wiki/Bilgisayar_ağı)*

gibi kabaca bir tanım yapabiliriz. Geçmişten bugüne ARPANET, USENET, X.25 gibi bir çok ağ oluşumu mevcut bulundu. ARPANET'te kullanılan paket anahtarlama (packet switching) teknolojisi o dönemde ve yahutta biraz daha eski bir dönemde kullanılan devre anahtarlama (circuit switching) teknolojisinin olduğu bir ortamda devrim etkisi yarattı ve günümüzdeki koşulların oluşmasının temelini oluşturdu. (Paket anahtarlama kavramı için kaynak: https://tr.wikipedia.org/wiki/Paket_anahtarlama)

Ağ Çeşitleri Nelerdir?

1. LAN (Local Area Network — Yerel Alan Ağları)

- Bir yerel bilgisayar ağı (LAN, Local Area Network), bir departman, çalışma grubu gibi aynı fiziksel lokasyondaki bilgisayar veya diğer bilgi işleme aygıtlarını birbirlerine bağlayan yüksek hızlı bir haberleşme sistemidir.
- Yerel alan ağları (LAN'lar) aynı çalışma ortamında birbirleriyle ilgili işlerde çalışan bir topluluk içinde veri alış verişi ve bilgisayarların CPU, disk gibi kaynaklarının ve yazıcı, çizici gibi cihazların paylaşması amacıyla geliştirilmiştir.
- LAN'Iarda temel özellik, sistemlerin aynı ortamda veya birbirlerine yakın mesafede almasıdır. Bu nedenle sistemler arasında kullanılacak kabloların, seçiminde büyük esneklik vardır ve kablolama alt yapısı bir kez kurulduktan sonra maliyetsiz bir iletişim ortamı sağlar.
- *Ethernet, Token Ring, Token Bus, 100VG-AnyLAN, ATM ve FDD*I, LAN uygulanmalarında kullanılan teknolojilerdir.
- Yerel alan ağ toplojileri ağ cihazlarının bağlantı ve organizasyonlarının şeklini belirler. *Bus*, *Halka(Ring)*, *Yıldız(Star)* ve *ağaç(Tree)* olmak üzere dört temel topoloji mevcuttur

- Bir yerel ağ şebekesinde *unicast*, *multicast* ve *broadcast* olmak üzere üç türlü iletim yöntemi vardır:

> ** **Unicast** iletimde tek bir paket bir istasyondan diğer istasyona iletilir. Pakette kaynak ve varış adresleri biulunur. Buna göre ilgili ağa gelen paket varış istasyonuna iletilir.*
> ** **Multicast** iletim yönteminde ise, bir veri paketi ağadaki özel bir altküme düğümlere iletilir. Multicast adresini kullanan paket ağa geldiğinde, kopyaları ağdaki özel düğümlere iletilir.*
> ** **Broadcast** iletim yönteminde ise, bir veri paketi ağadaki bütün düğümlere iletilir. Broadcast adresini kullanan paket ağa geldiğinde, kopyaları bütün düğümlere iletilir.*

- Yerel alan Ağlarında çoğunlukla kullanılan cihazlar; *repeaterlar, Hublar, LAN Extender, Bridgler, LAN Anahtarları* ve *Routerlardır*.
- Daha detaylı bilgi için bkz. (en.wikipedia.org/wiki/Local_area_network)

2. MAN (Metropolitan Area Network — Metropol Alan Ağı)

- MAN'lar, LAN'ların şehir çapındaki büyük türleridir ve LAN'larla benzer teknolojileri kullanırlar.
- Bir MAN veri ve ses haberleşmesi sağlayabileceği gibi yerel kablolu TV ağına da bağlantılı olabilir.
- MAN'larda anahtarlama elemanları bulunmaz; bağlantı bir ya da iki kablo ile sağlanır ve yayın türü iletim yapılır.
- *Head end* olarak adlandırılan aygıt, iletişimde kullanılan 53 oktet'lik frameleri (burada hücre adı verilir) art arda üretir.
- Üretilen hücreler veriyolu boyunca yol alır ve ağa bağlı bilgisayarlar tarafından kullanılmazlarsa veriyolunun diğer ucundaki sonlandırma direnci (TR) tarafından yutulurlar.
- MAN'ları diğer ağlardan ayıran en önemli özellik MAN'lar için uygulanan *DQDB (Distributed Queue Dual Bus)* yöntemidir.
- MAN hakkında daha fazla bilgi için bkz. (en.wikipedia.org/wiki/Metropolitan_area_network)

3. WAN (Wide Area Network — Geniş Alan Ağı)

- WAN teknolojileri komple bir ağın önemli bir parçasını oluşturur.

- Bilindiği gibi komple bir ağ, LAN'lardan, uzak kullanıcılardan ve bunların birbirleriyle haberleşmeleri veya merkez noktaya erişebilmeleri için WAN bağlantıları içerir.
- WAN teknolojisi denildiğinde akla hemen *çevirmeli (dial-up) modem*, *kiralık hat, X.25, FR, ISDN, xDSL* gelir; ancak *ATM, B-ISDN ve SMDS* gibi teknolojilerde WAN uygulamalannda boy göstermektedir.
- WAN teknolojileri birçok açıdan sınıflanır ve bu sınıflamalar projelendirme aşamasında kullanılması gereken teknolojiyi ortaya çıkarır.
- Sınıflamalardan yoğun kabul gören üç tanesi bağlantı durumuna, anahtarlama yöntemine ve topolojik yapısına göre yapılır:

> • ***Bağlantı Durumuna Göre:***
> ** Noktadan noktaya*
> ** Çoklu Bağlantı teknolojisi*
> • ***Anahtarlama Yöntemine Göre:***
> ** Devre anahtarlama*
> ** Paket anahtarlama*
> ** Hücre anahtarlama*
> • ***Topolojik Yapısına Göre***
> ** Hiyerarşik topoloji*
> ** Örgü topolojisi*

Ağ Topojileri Nelerdir?

Topoloji, ağı toplayan düğümler, kablolar ve bağlanırlık aygıtlarının düzenlemesini tanımlar. İki kategori, topolojilerin temelini inceleyebiliriz.

1. **Fiziksel Topoloji:** Ağ iletim ortamının güncel yerleştirilmesini tanımlar.
2. **Mantıksal topoloji:** Sinyalin mantıksal yolunu, ağ düğümleri arasında tanımlar.

- Bu iki kategori arasındaki fark başka bir deyişle, fiziksel topoloji ağ bakışının yolunu tanımlar ve mantıksal topoloji de düğümler arasında veri geçiş yolunu tanımlar. Topoloji çeşitlerini şöyle sıralayabiliriz:
- Bus Topolojisi, Ring (Halka) Topolojisi, Star (Yıldız) Topolojisi, Mesh Topolojisi, Dairesel topoloji, Ağaç ve Örgü Topolojileri gibi topolojiler mevcut. (Kaynak: http://www.muratkara.com/network/)

Client/Server mimarisi ve p2P (peer-to-peer) mimari kavramları üzerine de bahsedecek olursak:

Ağlar, genellikle iki geniş ağ kategorisi olan Peer-to-Peer ağ ile Client-Server ağından birisinden yararlanırlar. Bunları:

1- Client-Server Mimarisi:

- Web bazen bir istemci-sunucu iletişim modeli (client-server model of communications) olarak adlandırılır.
- Client-Server modelinde iki tür aktör vardır: clientlar ve serverlar.
- Server, normalde günde 24 saat, haftada 7 gün etkin olan ve istekte bulunan herhangi bir clienttan gelen sorguları dinleyen bir bilgisayar temsilcisidir.
- Client, yanıt kodları(response codes), resimler, metin dosyaları ve diğer veriler biçiminde serverdan istekte bulunan (makes requests) ve yanıtlar alan (receives responses) bir bilgisayar temsilcisidir.(computer agent)

1.1 Client (İstemci):

- Client makineleri, günlük yaşamın her yerinde gördüğünüz masaüstü bilgisayarlar, dizüstü bilgisayarlar, akıllı telefonlar ve tabletlerdir. (Bu makineler işletim sistemi, işlem hızı, ekran boyutu, kullanılabilir bellek ve depolama ile ilgili çok çeşitli özelliklere sahiptir.)
- En bilinen senaryoda, web sayfaları için client requestleri bir web browserından gelir.
- Ancak bir client bir web browserdan daha fazlası olabilir.
- Kelime işlemcinizin (word processor) yardım sistemi çevrimiçi kaynaklara eriştiğinde, bir istemcidir. (HTTP kullanarak bir game serverla iletişim kuran bir iOS oyunu gibi)
- Bazen bir server web programı client olarak bile çalışabilir. (Örneğin, PHP web sitelerimiz Flickr ve Microsoft gibi servis sağlayıcılardan web servisleri kullanacaktır; bu gibi durumlarda PHP uygulamamız client olarak görev yapar.)
- Bir clientın temel özelliği, *URL'leri* kullanarak belirli kaynaklar için belirli serverlara istekte bulunabilmesi ve ardından yanıtı beklemesidir. (Bu istekler bir şekilde

server tarafından işlenir)

1.2 Server (Sunucu):

- Bu modeldeki server, client-server modelinin *merkezi deposu/veri havuzu (central repository)*, komut merkezi (command center) ve merkezi hub'ıdır. (central hub)
- Web uygulamalarını barındırır (host eder), kullanıcı ve program verilerini depolar ve güvenlik yetkilendirme görevlerini (security authorization tasks) yerine getirir.
- Bir server binlerce veya milyonlarca client isteğine hizmet edebileceğinden, serverlara olan talepler yüksek olabilir. (Örneğin, görüntü veya video verilerini depolayan bir site, kullanıcıların taleplerini karşılamak için birçok terabayt depolama alanı gerektirir.)
- Bir serverın temel özelliği, requestleri dinlemesi ve bir tanesinin alınmasının ardından bir mesajla yanıt vermesidir.(response)
- Client ve server arasında bilgi alışverişi, *istek-yanıt döngüsü (request-response loop)* tarafından özetlenir.

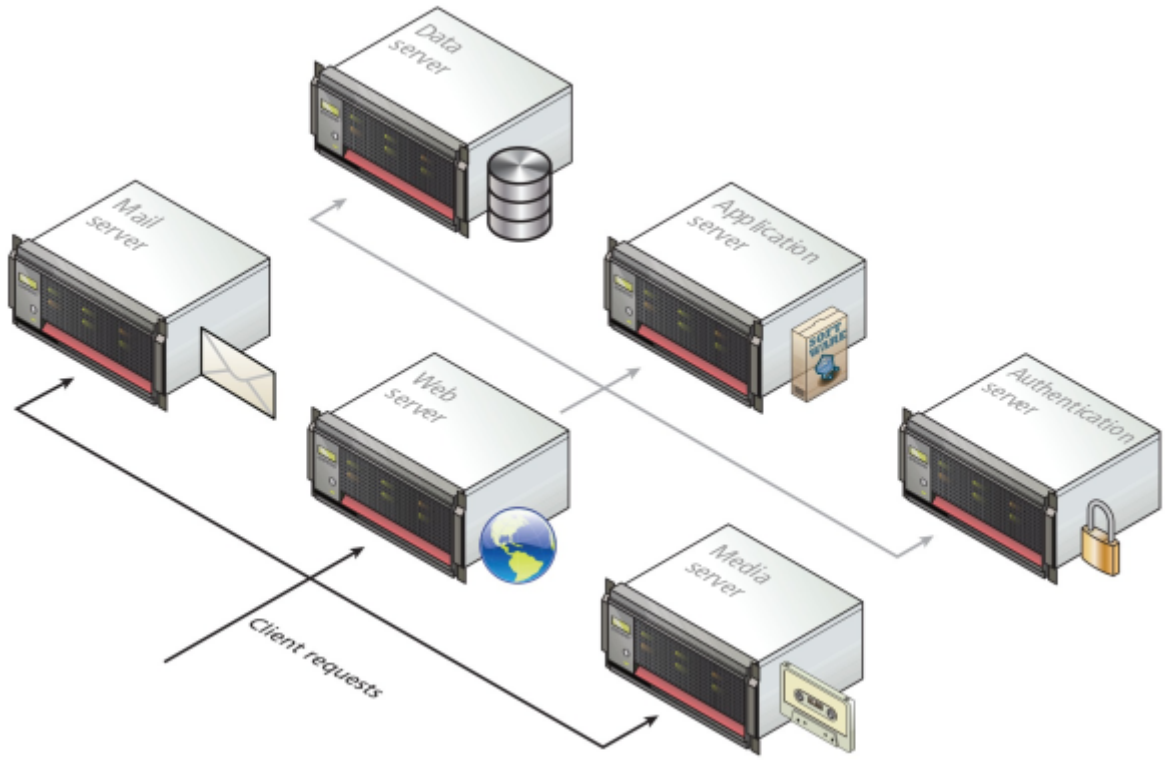
1.2.a Server Türleri

- Server kavramı genelde güçlü tek bir makie olarak algılanır.
- Bununla birlikte, gerçek dünyadaki çoğu web sitesi genellikle tek bir server makinesinden değil, birçok server tarafından sunulur.
- Bir web sitesinin işlevselliğini Şekil'de gösterildiği gibi birkaç farklı sunucu türü arasında bölmek yaygındır. Bunlar:
- ***Web servers:*** Web server, HTTP requestlerine hizmet veren bir bilgisayardır. Bu genellikle *Apache* veya *Microsoft IIS (Internet Information Services)* gibi web server yazılımı çalıştıran bir bilgisayarı ifade eder.
- ***Application Servers:*** Application server, *PHP*, *ASP.NET*, *Ruby on Rails* veya başka bir web geliştirme teknolojisinde oluşturulabilen web uygulamalarını barındıran (hosts) ve yürüten (executes) bir bilgisayardır.
- ***Database Servers:*** Database Server, web uygulamaları tarafından kullanılan *MySQL*, *Oracle* veya *SQL Server* gibi bir *Veritabanı Yönetim Sistemi (Database Management System — DBMS)* çalıştırmaya ayrılmış bir bilgisayardır.

- ***Mail Servers:*** Mail server, genellikle *SMTP (Simple Mail Transfer Protocol)* kullanan posta istekleri (mail request) oluşturan ve karşılayan (creating and satisfying) bir bilgisayardır.

- ***Media Servers:*** Media Server (gerçek zamanlı aktarım sunucusu (streaming server) olarak da bilinir), görüntü ve video isteklerine hizmet vermeye adanmış özel bir server türüdür. Video içeriğinin clientlara akışını sağlayan özel bir yazılım çalıştırabilir.

- ***Authentication Servers:*** Authentication Server, web uygulamalarının en yaygın güvenlik gereksinimlerini karşılar. Bu, *LDAP (Lightweight Directory Access Protocol — Basit Dizin Erişimi Protokolü)* veya *Active Directory* gibi yerel ağ kaynaklarıyla (local networking resource) etkileşimi içerebilir.



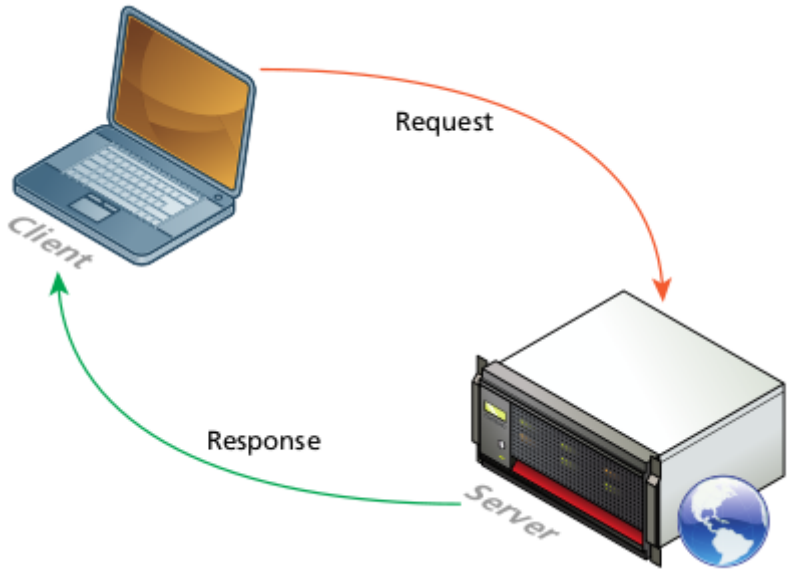
Server Türlerini gösteren figür

1.3 The Request-Response Loop (İstek Yanıt Döngüsü):

- Client-Server modelinde, *request-response döngüsü,* serverda requestleri almak ve response olarak veri iletmek için en temel mekanizmadır.

- Client, servera bir request başlatır ve Şekil'de gösterildiği gibi HTML dosyası, görüntü veya başka veriler gibi bazı kaynakları içerebilecek bir response alır.

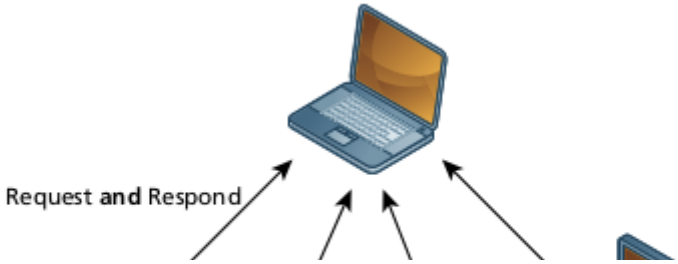
- Bu response ayrıca *request* veya *response codes*, *cookieler* ve diğer veriler gibi sağlanan kaynak hakkında başka bilgiler de içerebilir.

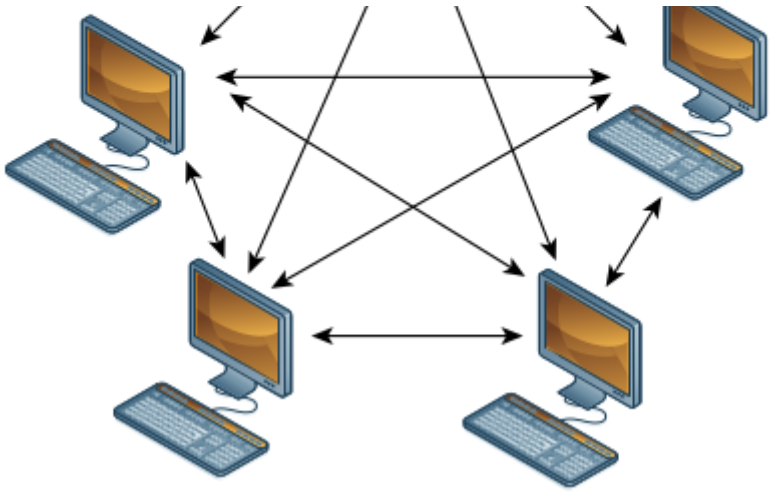


Request-Response döngüsünü gösteren bir figür.

2- Peer-to-Peer Mimari:

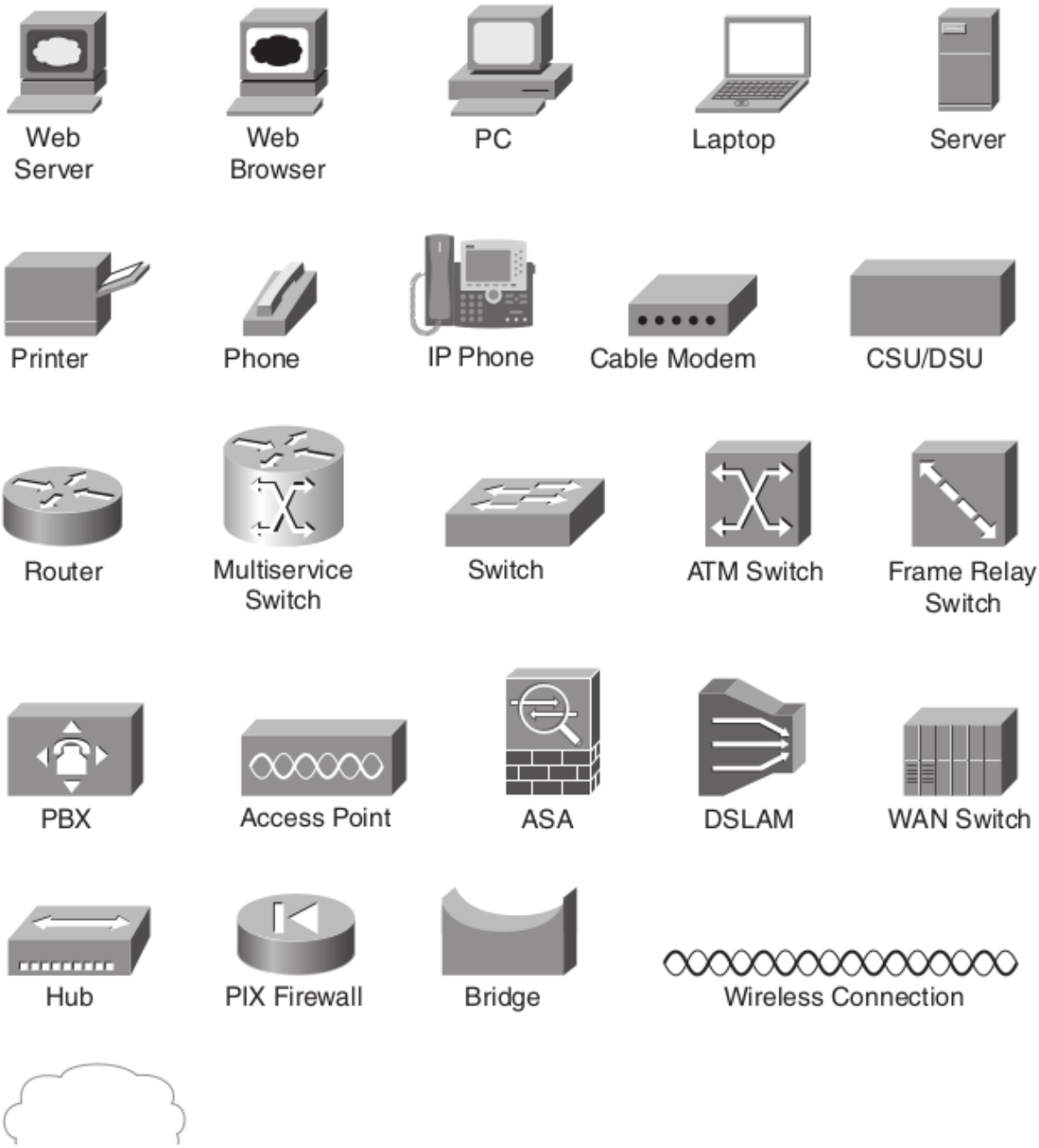
- Client-Server modelini farklı bir *ağ topolojisi (network topology)* ile karşılaştırmak sizin anlamanıza yardımcı olabilir.
- Her bilgisayarın işlevsel olarak özdeş (functionally identical) olduğu Şekil'de gösterilen *p2p modelde*, her *node* doğrudan birbirleriyle veri alışverişinde bulunabilir.
- Böyle bir modelde her eş, bilgi yükleyip indirebilen bir istemci ve sunucu gibi davranır.
- Her ikisinin de 7/24 bağlanmasına gerek yoktur ve her bilgisayar işlevsel olarak eşit olduğunda, eşler arasında daha az fark vardır.
- Client-Server modeli, bunun aksine server için net ve farklı roller tanımlar.
- Görüntülü sohbet ve bit torrent protokolleri p2p modelin örnekleridir.

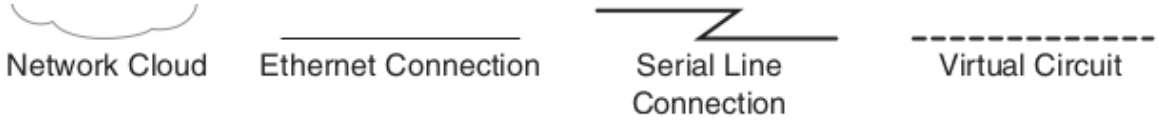




p2p modelini gösteren bir figür

Ağ Cihazları/Kavramlarına Kısa Bir Bakış





(Kaynak: Cisco Ağ Teknolojileri Yönetimi Kitabı)

Router: Yönlendirici, aynı ağ iletişim kurallarını kullanan iki bilgisayar ağı arasında *veri çerçevelerinin(data frame)* iletimini sağlayan ağ donanımıdır. Yönlendirme için OSI yedi katman modelinin üçüncüsü olan ağ katmanı (Network Layer) kullanılır. Genellikle bu iş için özel üretilmiş donanımlar varsa da birden çok arayüzü olan bilgisayarlar da yazılım desteğiyle yöneltici olarak çalışabilirler. (Kaynak: https://tr.wikipedia.org/wiki/Yönlendirici)

Switch: Switch, bilgisayarların ve diğer ağ öğelerinin birbirlerine bağlanmasına olanak veren ağ donanımlarından biridir. OSI yedi katman modelinin 2. katmanında ve yeni dağıtıcılar IP routing yapabildiği için 3. katmanda da çalışır. (Kaynak: https://tr.wikipedia.org/wiki/Ağ_anahtarı)

Hub: *Star Topolojisi* yapısındaki ağlarda merkezi bağlantıyı sağlayan cihazdır. Üzerindeki port sayısına göre isimlendirilir ve bu portlara makineler takılır. (Kaynak: https://tr.wikipedia.org/wiki/Hub_(bilgisayar))

Bridge: İki bilgisayar ağını birine bağlayan ağ öğelerinden birine verilen addır. Bu işlem OSI yedi katman modelinin ikincisi olan veri bağ katmanında gerçekleşir. Bu özelliği sebebiyle çalışma prensibi 3. katmanı kullanan yönelticiden ve 1. katmanı kullanan yineleyiciden(repeater) farklıdır. (Kaynak: https://tr.wikipedia.org/wiki/Ağ_köprüsü)

Repeater: Uzaktaki bilgisayara gönderilen veri paketinin gönderim sırasında kayıp yaşanmaması için belirli mesafelere veri kaybını önlemek için repeater ismi verilen cihazlar konularak bu verileri tekrarlatarak canlı tutan ağ donanımlarından biridir. (Kaynak: https://www.olkando.com/tekrarlayici-repeater-nedir/)

Modem: Tanım olarak "*Modülator*" ve "*Demodülator*" kelimelerinin birleşiminden üretilmiştir. Çevirge ya da Modem, bilgisayarların genel ağa bağlantısını sağlayan ve bir bilgisayarı uzak yerlerdeki bilgisayar(lara) bağlayan aygıttır. (Kaynak: https://tr.wikipedia.org/wiki/Modem)

Web Kavramı

- Web internetin kendisi değildir, sadece bir alt kümesidir.

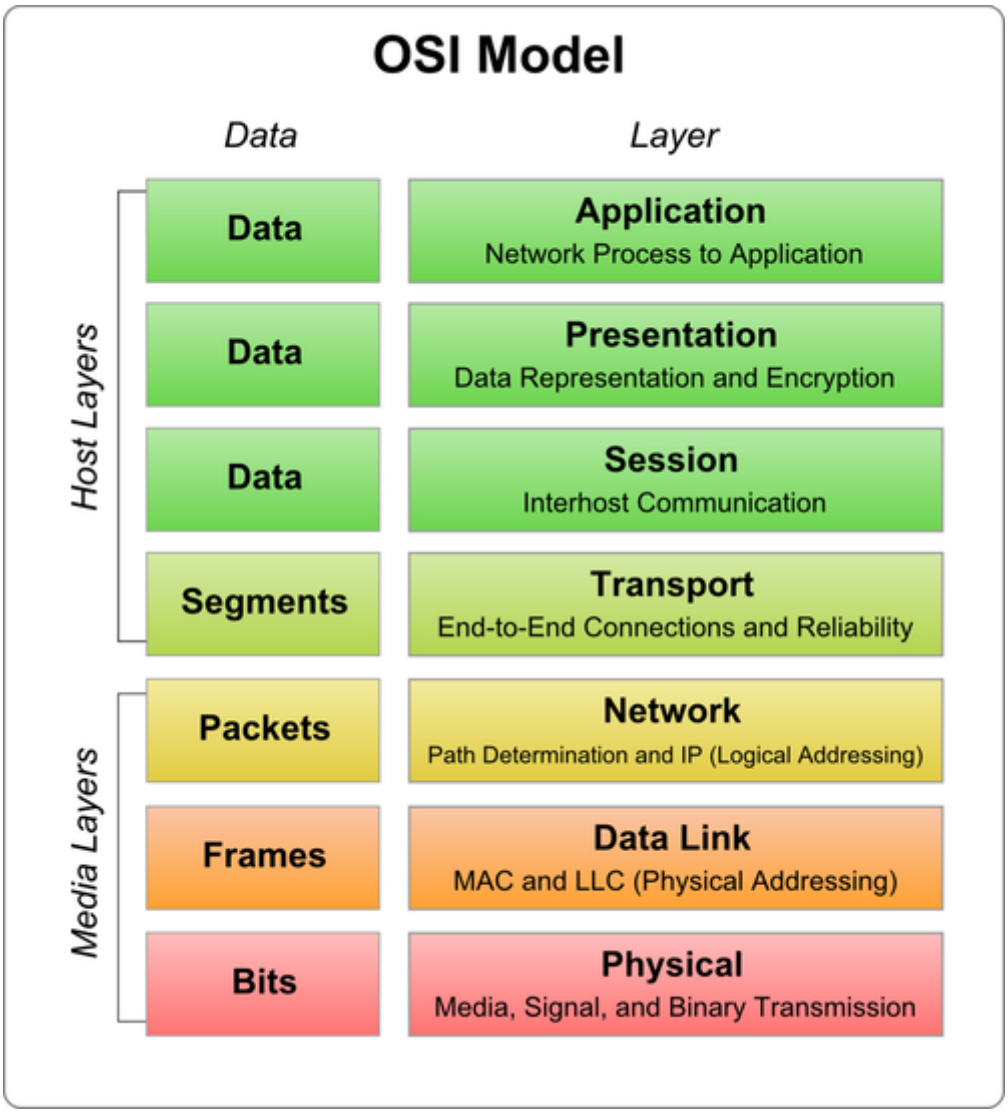
- ***Web 1.0:*** Statik Web sayfaları bu Web kavramının içersinde mevcuttu. Kullanıcılar web siteleriyle etkileşim halinde olamıyorlardı. Webmaster adı verilen bir admin web sayfalarını yayınlıyor ve periyodik olarak güncelliyordu.

- ***Web 2.0:*** *Backend* ve *Frontend* gibi kavramların ortaya çıktığı, dinamik sitelerin içersinde bulunduğu kavram. Facebook, Twitter, Youtube vb. siteler bu kavramın bir getirisidir.

- ***Web 3.0 — Semantic Web (Anlamsal Web):*** Türkçe karşılığı ***Anlamsal Ağ*** olan Semantik Web, online içeriklerin sadece insanlar tarafından değil, yazılımlar tarafından da kolayca anlaşılabilir, kullanılabilir ve yorumlanabilir olmasını amaçlayan bir web projesi olma özelliği taşıyor. (Kaynak: https://www.mediaclick.com.tr/blog/semantik-web-nedir)

Referans Modelleri

OSI Referans Modeli



- 1974 Yılında ISO (International Standarts Organization) tarafından, farklı bilgisayar firmalarının ürettikleri bilgisayarlar arasındaki iletişimi standartlaştırabilmek adına ve farklı sistemler arasındaki uyumsuzluk sebepleri ile ortaya çıkan iletişim sorununu ortadan kaldırmayı amaçlayarak tasarlanmıştır.
- OSI Referans modelinde, iki bilgisayar sistemi arasında yapılacak olan iletişim problemini çözebilmek için 7 katman mevcuttur. (7 katmanlı mimari olarak da bilinir.)
- Her bir katman, iletişimin nasıl yapılacağını ve diğer katmanlar ile etkileşimi açıklar.
- Bu 7 katmanın en altında yer alan iki katman yazılım ve donanım, üstteki beş katman ise genelde yazılım yolu ile çözülmüştür.
- OSI modeli, bir bilgisayarda çalışan uygulama programının, iletişim ortamı üzerinden başka bir bilgisayarda çalışan diğer bir uygulama programı ile olan iletişiminin tüm adımlarını tanımlar.
- En üst katmanda görüntü ya da yazı şeklinde yola çıkan bilgi, alt katmanlara indikçe makine diline dönüşür ve sonuç olarak 1 ve 0'lardan ibaret elektrik sinyalleri halini alır.
- OSI referans modeli bir ağ uygulaması değildir. OSI sadece her katmanın görevini tüm detayları ile tanımlar.
- Fiziksel bağlantılar (1 ve 2. katmanlar): Bu katmanlar üst katmanlara (3–7) fiziksel bağlantı sağlarlar ve verinin ağ ortamından iletilmesinden sorumludurlar.
- İletişim (3 ve 4. Katmanlar): Bu katmanlar fiziksel ortamdan bağımsız olarak gönderici ya da alıcı tarafından verinin doğru olarak gönderildiğini/alındığını garanti eden katmanlardır.
- Servisler (5, 6 ve 7. katmanlar): Bu katmanlar kullanıcıya bilgisayar ağı servisleri sağlarlar. Bu servislerden bazıları, dosya ve yazı servisi, elektronik posta, terminal emülasyonu, format çevrimi, login denetimi ve diğerleridir.
- OSI modeli detaylı bir tanımlama ya da protokol olmadığından, veri iletişim protokollerini tartışırken referans model olarak kullanılabilir.
- Veri iletişim protokollerinin amacı farklı birimler üzerinden uygulama verilerinin taşınmasıdır. Tüm veri iletişim protokollerinde temel amaç budur. Veri iletişim

protokolleri farklı yöntemlerle geliştirilseler bile aynı işlevi yerine getirmektedirler.

Katmanları ise şu şekilde betimleyebiliriz:

1. **Physical Layer (Fiziksel Katman)**

- 1 ve 0 sinyallerinin taşındığı katmandır.
- Bu katmanın görevi dijital sintali taşımaktır.
- *CAT-5*, *CAT-6* ve *Fiber kablolar* ile sinyalin zayıfladığı yerde sinyali yeniden güçlendirerek ileten *repeater(tekrarlayıcı)* bu katmanda görev alır.
- Ayrıca en ilkel ağ donanımlarından olan *HUB* cihazı da bu katmanda görev alır.
- Bu katmandaki veri birimi Bit'tir. Yani, veri bu katman için sıradan bit dizisi olarak algılanır.
- Bitlerin taşıdığı bilgi bu katmanda yorumlanmaz.

2. **Data-Link Layer (Veri Bağlantı Katmanı)**

- Bu katman network içi iletişimi sağlar.
- *Switch (Ağ anahtarı)* cihazı bu katmanda görev alır.
- Bu katmanda switch, network içi iletişimi *frame'*leri kullanarak sağlar.
- Switch, üzerinde bulunan *MAC Table*'a göre frame'leri ilgili yerlere yönlendirir. (MAC Tablosunda fiziksel portlar ve ilgili porta bağlı olan cihaz/cihazların MAC adresleri liste halinde bulunur.)
- *MAC Adresleri* 6 Byte (48 Bit)'dan oluşan adreslerdir.
- Bilgisayarlarda bulunan *ethernet* ya da *kablosuz ağ kartlarının* üzerinde bulunan MAC adresleri *unique(eşsiz)* yapıdadır.
- Bu katmanda görev alan switch'in çalışma prensibi şu şekildedir:

> ** Switch, bir bilgisayardan frame geldiği zaman öncellikli olarak hangi fiziksel porttan bu frame'in geldiğine bakar.*
>
> ** Gelen frame içerisindeki source(kaynak) MAC adresini alır ve MAC tablosuna fiziksel portu ve karşısına da o porta bağlı bilgisayarın MAC adresini yazar.*

> ** Daha sonra, gelen frame'in içeriğine bakarak hedef (destination/target) MAC adresi tespit eder.*
>
> ** Hedef MAC adresin kaydı, MAC Tablosunda var ise hedef MAC Adresi ile eşleştirilmiş fiziksel porta frame gönderilerek hedefteki cihaza ulaştırılması sağlanır.*
>
> ** Eğer adres tabloda yok ise frame'in alındığı port dışındaki tüm portlara frame gönderilerek doğru adres bulunmaya çalışılır.*

- Özetle, bu katmanda switch cihazı network içi iletişimi sağlamak için framelerin haberleşmesini, üzerinde bulunan MAC Table ile sağlar.
- Bilgisayarınızda bulunan MAC adresini/adreslerini öğrenmek için;

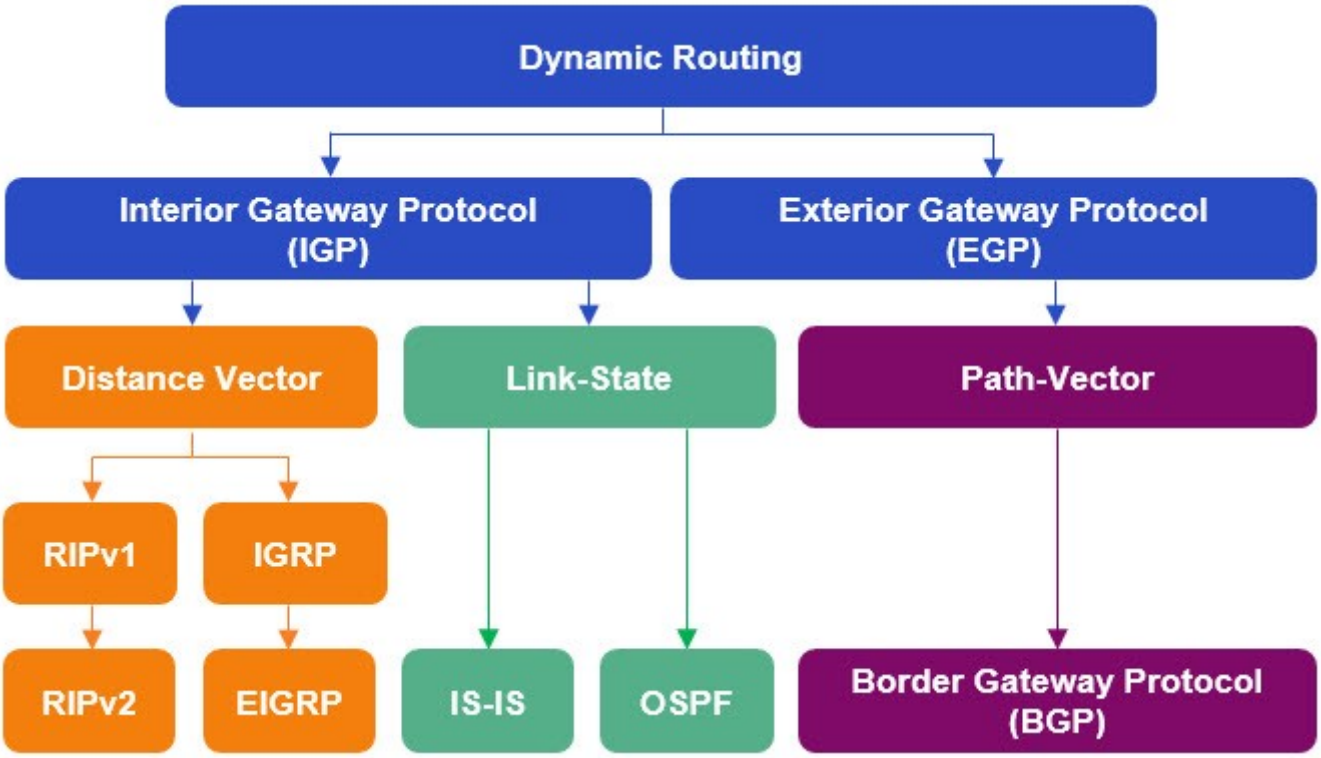
> ** Windowsta; cmd üzerinden **getmac** komutunu çalıştırabilirsiniz.*
>
> ** GNU/Linux dağıtımlarında ise, terminal üzerinden **ifconfig** komutunu çalıştırabilirsiniz.*

- Bu katmanda giden framelerde bozulma olup olmadığının anlaşılması için bütünlük kontrolü yapılarak iletişimde hata olup olmadığı tespit edilir.
- Ayrıca, gönderilen ve alınan lojik işaret bloklarına *Frame(çerçeve)* denir ve Framelerin içerdiği bit sayısının alt ve üst sınırları standartlarla belirlenmiştir.
- *Ethernet*, *HDLC*, *Wi-Fi*, *Token ring*, *FDDI*, *PPP*, *L2TP* vb gibi kavramlar bu katmanda geçer.

3. **Network Layer(Ağ Katmanı)**

- Bu katman, networkler arası iletişimi sağlamakla görevli katmandır.
- *Router(yönlendiri)* adlı network cihazı bu katmanda görev alır.
- Router cihazının görevi, farklı ağları birbirine bağlamaktır.
- *IP, ICMP (Internet Control Message Protocol), ARP(Address Resolution Protocol)* gibi protokoller bu katmanda görev alır.
- ARP protokolünün görevi; IP adresinden MAC adresinin çözümlenmesini sağlamaktır.

- Router, *Routing Table*'a (Yönlendirme Tablosu) göre bir paketin hedefe gönderilmesi için en kısa/en kullanışlı yolu belirler.(Çeşitli *Routing Protokollerini* kullanarak bu işlemi gerçekleştirir. Örneğin *OSPF* vb.)



Dinamik Yönlendirme Protokolleri

- Ayrıca, routerlar, *paketleri filtreleyebilir* ve *paket anahtarlaması (packet switching)* yapabilir.
- Routing tablosunda IP ve Router üzerinde bulunan *interface* eşleşmesi ya da IP adresleri ve hedefe giderken paketin bir sonraki noktadan geçmesi için ihtiyaç duyulan bir sonraki noktanın IP adresi eşleşmesi bulunur.
- *IP adresleri* internetteki cihazların birbirleriyle haberleşmesi için kullanılan adreslerdir.
- *IPv4* kapsamında yer alan IP adresleri, 4 adet 8-bitlik kısımdan oluşur yani 32 bitlik (4 Byte) adreslerdir.
- ***Private IP Address:*** Özel IP adresleri iç ağda (local ağda) kullanılan IP adresleridir.
- ***Public IP Address:*** İnternette ya da kabaca Dış dünyada kullanılan, *ISP (Internet Service Provider — Internet Servis Sağlayıcı)* tarafından kullanıcıya verilen IP adresleridir.

- Bu iki IP adresi arasındaki farkı anlamandırabilmek için şöyle bir betimle yapmak yanlış olmaz; modemin dış dünyaya bakan ayağında ISP'nin verdiği gerçek IP adresini bulunurken, evde kullanılan makinelerde ise modemin lokal ağa bakan ayağında bulunan IP aralığındaki IP adreslerini yani Private IP adresleri kullanılır. (Bunun sebebi IPv4 içindeki IP adreslerinin daha tasarruflu şekilde kullanılmak istenmesinden kaynaklanır.)

- Ayrıca burada *NAT(Network Address Translation)* adı verilen bir yapı vasıtası ile lokal ağda bulunan bilgisayarların IP adreslerini, Public adreslerine dönüştürme işlemi gerçekleştirilir. Bunun sebebi lokal ağdaki tüm makinelerin dış dünyada aynı Public (reel) IP adresini kullanması durumudur. (NAT kavramı adına daha fazla bilgi edinmek için bkz. https://medium.com/@gokhansengun/nat-network-address-translation-nedir-ve-nasıl-çalışır-a2c8b6291de8)

- Routerlar üzerinde farklı ağların birbirlerine bağlanabilmesi için farklı interfacelere ihtiyaç duyulmaktadır.

- Maliyetin yüksek olduğu durumlarda tek bir interface *alt interfacelere (subinterface)* bölünerek router interface'ine bağlı olan switch üzerinde farklı ağlar yapılandırabilmek için *VLAN (Virtual Local Area Network — Sanal Yeral Alan Ağları)* oluşturulabilir.

- Her bir VLAN farklı bir ağı temsil etmiş olur.

- *ICMP* ağdaki hataları kontrol etmek amacıyla kullanılır.

> ***ping*** *ip_adresi komutuyla hedefe, ICMP paketleri gönderilerek ilgili kontroller sağlanabilir.*

- Ayrıca veri birim olarak *paket(packet)* kavramı mevcuttur.

- *IP, IPv4, IPv6, ICMP, ARP, IGMP, IPX* vb kavramlar bu katmanda görev alırlar.

4. Transport Layer (İletim Katmanı)

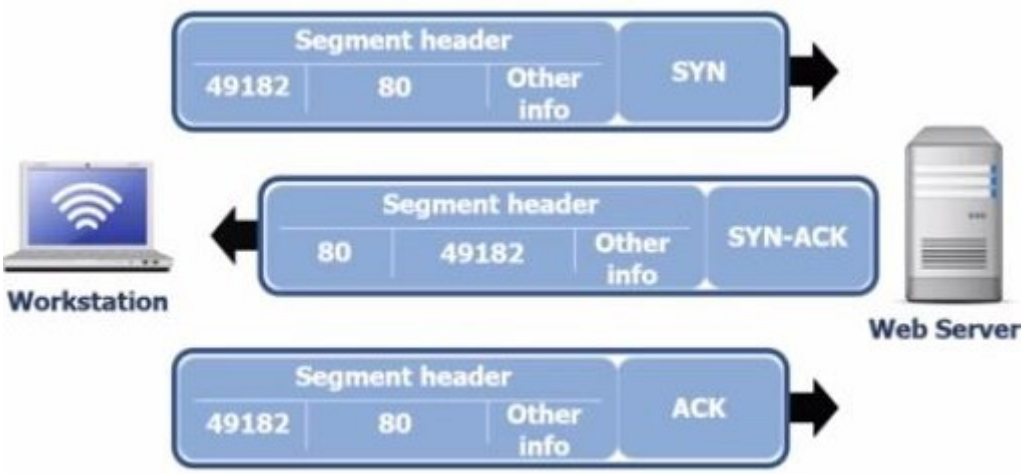
- *Uçtan-uca iletimi* sağlayan katmandır.

- *Port*, *network servislerinin* bağlantı kurabilmesi için sistem üzerinde açılması gereken sanal kapıdır. (Bir bilgisayarda 2**16 adet yani 65536 adet port mevcuttur)

- Bilgisayarınızdaki açık olan portları görüntüleyebilmek için şu komutları kullanabilirsiniz:

> ** Windowsta, cmd üzerinden **netstat -an** komutunu kullanabilirsiniz.*
>
> ** GNU/Linux'da terminal üzerinde **netstat -tuna** komutunu kullanabilirsiniz.*

- *TCP* ve *UDP* protokolleri bu katmanda görev alır.
- *TCP (Transmission Control Protocol)* 3'lü el sıkışma yöntemini kullanarak transfer edilen verinin karşı tarafa gidip gitmediğini, veri kaybı yaşanıp yaşanmadığını kontrol eder.
- ***TCP 3-Way Handshake:*** TCP protokolü kullanılarak client ve server arasında gerçekleşen bu olay sonucu bağlantı gerçekleştirilir. (Aksi durumlar hariç) Bu mekanizma client'ın SYN paketi ile başlar, karşı taraftan da SYN+ACK paketinin gelmesi halinde (ACK paketi client'ın SYN paketini karşılarken SYN paketi de client tarafından bir ACK paketi almak adına gönderilir) client'ında bu paketlere ACK işaretli paketle yanıt vermesi sonucunda bağlantı sağlanır.



TCP 3'lü El Sıkışma Mekanizması

- Veriler karşılıklı olarak iletilirken iki taraftanda veri alındığında karşı tarafa verinin alındığına dair ACK paketi gönderilir.
- TCP protokolü bu bakımdan veri aktarımı konusunda güvenlidir. (Buradaki güvenli sözcüğü herhangi bi encryption işlemini ifade etmek için değil paket iletiminin sağlanığı sağlanmadığının kontrolünün gerçekleştirilmesinden ötürüdür.) Ayrıca TCP'ye *connection-oriented* protokol de denir.
- *UDP (User Datagram Protocol)'*nde ise bir onaylama mekanizması yoktur.

- Ses ağları gibi hız gerektiren yerlerde sıklıkla UDP kullanılır.
- Buradan yapabileceğimiz bir çıkarımla TCP bağlantısı, UDP bağlantısına oranla daha yavaş gerçekleşir.
- TCP hakkında daha fazla bilgi için bkz. https://medium.com/@gokhansengun/tcp-nasıl-çalışır-1–484612c5264f ve https://medium.com/@gokhansengun/tcp-nasıl-çalışır-2-dfa21d9a730d adreslerini inceleyebilirsiniz.
- Ayrıca, eğer veri frameden büyük ise iletim katmanında küçük parçalara ayrılırarak sıra numarası verilir. İletim katmanının oluşturduğu bilgi bloklarına *Segment* denir. Bunlar son alıcıya sırası bozulmuş olarak
  gelirse düzgün olarak sıralanırlar.
- *TCP*, *UDP*, *SCTP*, *DCCP* vb kavramlar bu katmanda görev alırlar

5- Session Layer (Oturum Katmanı)

- İki bilgisayarın/makinenin arasında oturumun açılması devan ettirilmesi ve sonlandırılmasından sorumlu katmandır.
- *SMB (Server Message Blog), NFS (Network File System* — Ağda birden fazla makine üzerinde bulunan çeşitli dosyaların tek bilgisayardaymış gibi kullanılmasına olanak tanıyan protokol*)* gibi kavramlar bu katmanda görev alır.
- *NFS, SMB, ISO 8326, ISO 8327, ITU-T T.6299* vb. kavramlar bu katmanda görev alır

6- Presentation Layer (Sunum Katmanı)

- Verinin formatının belirlendiği katmandır.
- Bilginin karakter set çevrimi veya değiştirilmesi, şifreleme vs. görevlerini bu katman üstlenir.
- Veri sıkıştırma, şifreleme, EBCDIC-ASCII dönüşümü ve ters dönüşüm bu katmanda gerçekleştirilir.
- Bu katman için *ASCII kodları* ve jpeg örnek verilebilir.
- *ISO 8822, ISO 8823, ISO 8824, ITU-T T.73, ITU-T X.409* vb. kavramlar bu katmanda görev alır.

7- Application Layer (Uygulama Katmanı)

- En üst katmandır, yani kullanıcıya en yakın katmandır.
- Uygulamaların çalıştığı katmandır.
- Kullanıcının etkileşimde bulunduğu uygulama programları doğrudan bu katmanla iletişim içindedir.
- Bu katman için dosya aktarımı, elektronik mektuplaşma, uzaktan dosya erişimi, ağ yönetimi, terminal protokolleri gibi standartlar geliştirilmiştir.
- Spreadsheet, kelime işlemci, banka terminali programları vs. bu katmanın parçalarıdır.
- *HTTP, DNS, SMTP, FTP, TFTP, UUCP, NNTP, SSL, SSH, IRC, SNMP, SIP, RTP, Telnet* vb. kavramlar bu katmanda görev alırlar.

OSI Referans Modeli Nasıl Çalışır?

Bu katmanların nasıl çalıştığını bir örnek üzerinde açıklayalım. Bir kelime işlem programı kullanıldığını ve bu programın resume.txt adındaki dosyayı uzaktaki sunucunun home kataloğundan almak istediğini varsayalım.Bu durumda işlem adımları aşağıdaki şekilde olacaktır.

• Uygulama katmanı bir request ile resume.txt dosyasının istendiğini anlar ve sunun katmanına bunu iletir.
• Sunum katmanı bu isteğin şifreli olup olmadığını ve bir veri tipi dönüşümü olup olmadığını belirler. İhtiyacı olan bilgiyi ekleyerek paketi oturum katmanına iletir.
• Oturum katmanı, dosyanın getirilmesi için hangi uygulamanın ve uzak sistemin hangi servisinin kullanılacağına karar verir. Uzak sistemin servis bilgisini ekleyerek paketi iletim katmanına gönderir.
• İletim katmanı uzak sistem ile garantili bir bağlantının olmasını ve eğer birden fazla frame gerekli ise paketi framelere ayırma işlemine hazırlar. Framelere sıra numarasını (sequence number)ekleyip ağ katmanına iletir.
• Ağ katmanı aldığı frame kendi ve diğer sistem adreslerini ekler ve veri ulaştırma katmanına iletir.
• Veri bağlantı katmanı, blokları bağımsız framelere ayırır. Ethernet paketlerinin header kısımlarına MAC adreslerini yerleştirir. Frame'in sonuna denetim dizisini koyar.

Topolojinin yapısına göre bu düzenlemeyi yapar.
• Fiziksel katman veriyi kaynaktan hedef sisteme sayısal darbeler halinde iletir.

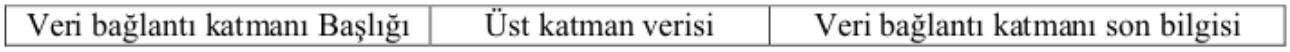
Diğer sistemde verinin alınması

• Uzak sistemdeki Veri Bağlantı Katmanı iletilen çerçeveyi okur. Varış adresinin kendisi olup olmadığına bakar. Eğer kendisi ise *CRC denetimini* yaparak uygun ise Network katmanına transfer eder.
• Network katmanı frame'i analiz ederek varış adresinin kendisi olduğunu anlar. Bu analizden sonra bu seviyedeki bilgiyi ayırır ve kalanı iletim katmanına gönderir.
• İletim Katmanı, kaynak sistem tarafından kaydedilen bilgiyi analiz ederek, bir sıra numarası bulursa veriyi kuyruğa atarak, bütün bilginin tamamlanmasını bekler. Eğer alınamayan veri var ise sıra numarasını kullanarak kaynak sistemin yeniden göndermesini sağlar. Daha sonra veriyi
oturum katmanına iletir.
• Oturum katmanı alınan veriyi alır ve geçerli bir bağlantıdan geldiğini kontrol eder. Daha sonra sunum katmanına iletir.
• Sunum katmanı alınan verideki dönüşüm ve çözümleme işlemini yaparak, sunum katmanı bilgisini ayırır ve uygulama katmanına iletir.
• Uygulama katmanı sistemde çalışan doğru sürecin işlem yapmasını garanti eder. Bu bir dosya isteği olduğu için dosya erişiminde sorumlu olan sürece görevi devreder.

Bağlı ağlar arasında iletilen veri ve kontrol bilgisi değişik formatta olabilir. Bu bilgi formatlarını belirtmekte kullanılan terimler bağlantılı ağ endüstrisinde uyumlu değildir, ancak birbirine dönüştürülebilir. Ortak bilgi formatları, *Çerçeve(frame)*, *Paket(Packets)*, *datagram*, *segment*, *iletiler(Messages)*, *hücre(Cells)* ve *veri birimleri(data units)* dir.

Veri birimi, değişik bilgi birimlerini gösteren genel bir terimdir.

- ***Çerçeve(Frame):*** Veri bağlantı katmanının varlığını simgeleyen kaynak ve varıştan oluşan bilgidir. Bir frame, veri bağlantı katmanı başlığı(header ve muhtemelen bir son bilgisi) ve üst katman verisinden oluşur. Başlık ve son bilgisi, veri bağlantı katmanının belirten kontrol bilgisini, üst katman bilgisi ise, veri bağlantı katmanı, başlık ve son bilgisi içinde saklanır.



Data-Link katmanına ait frame bilgisinin formatı

- ***Paket(Packet):*** Bir paket ağ katmanını belirten, kaynak ve varış bilgisini içerir. Bir paket, ağ katmanı başlığı(header ve muhtemelen bir son bilgisi) ve üst katman verisinden oluşur. Başlık ve son bilgisi, ağ katmanının belirten kontrol bilgisini, üst katman bilgisi ise, ağ katmanı, başlık ve son bilgisi içinde saklanır.

Ağ katmanı Başlığı	Üst katman verisi	Ağ katmanı son bilgisi

Network katmanı paketini oluşturan birim bilgileri

- ***Datagram:*** Bağlantısız servislerdeki ağ katmanında, başlangıç ve varış oluşumlarınii içeren birim bilgilerdir.
- ***Segment:*** İletim katmanındaki başlangıç ve varış oluşum bilgilerini içeren birim veridir.
- ***İleti(Message):*** Ağ katmanının üst katmanlarındaki(sıkça uygulama katmanında) kaynak ve varış oluşumlarını içeren bilgillerdir.
- ***Hücre(Cell):*** Bağlantı katmanındaki, kaynak ve varış oluşumlarını içeren sabit uzunluktaki birim veridir. Hücreler, çoğunlukla, *ATM* ve *SMDS (Switched Multi-megabit Data Service)* gibi anahtarlamalı ağlarda kullanılır. Bir hücre *header* ve (veri) *payload* bilgisinden oluşur. Örneğin
  ATM'de Header 5 byte, data ise 48 bayt olmak üzere toplam hücre uzunluğu 53 byte'tan oluşur.

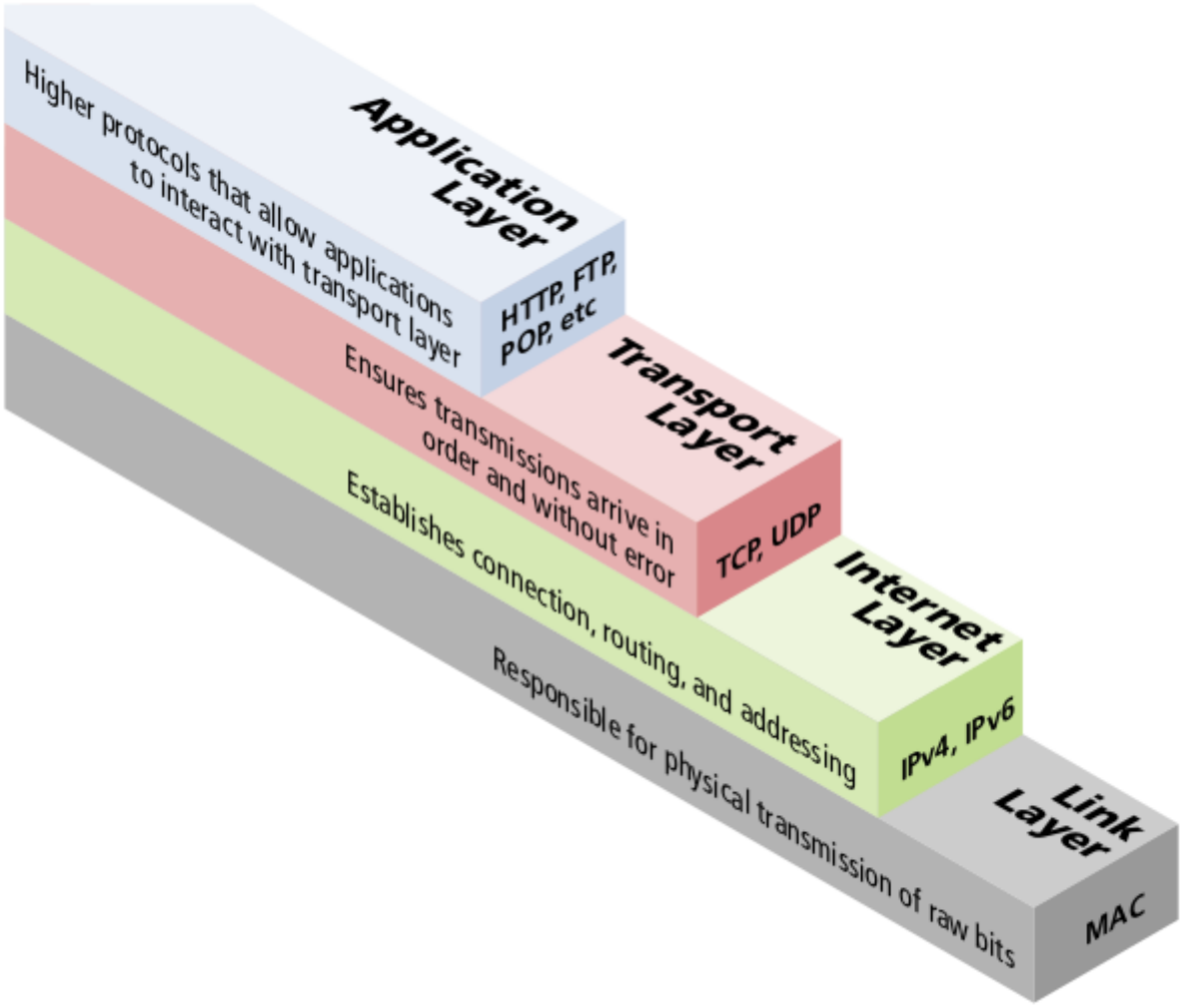
Hücre Başlığı (5 Byte)	Veri (48 Byte)

Bir ATM hücresinin yapısı

(Kaynak: http://www.muratkara.com/network/Network_DersNotu.pdf)

- Neticede, TCP/IP teknolojisinin OSI'ye göre teknik olarak üstün olduğu anlaşılmıştır ve birkaç yıllık süre içerisinde OSI'yi geliştirme ve büyütme çabaları son bulmuştur. (Kaynak: Computer Network and Internet, Sixth Edition, Douglas E. Comer, sayfa 13)

TCP/IP Referans Modeli



(Kaynak: Fundamentals of Web Development, First Edition, Randy Connolly, Ricardo Hoar)

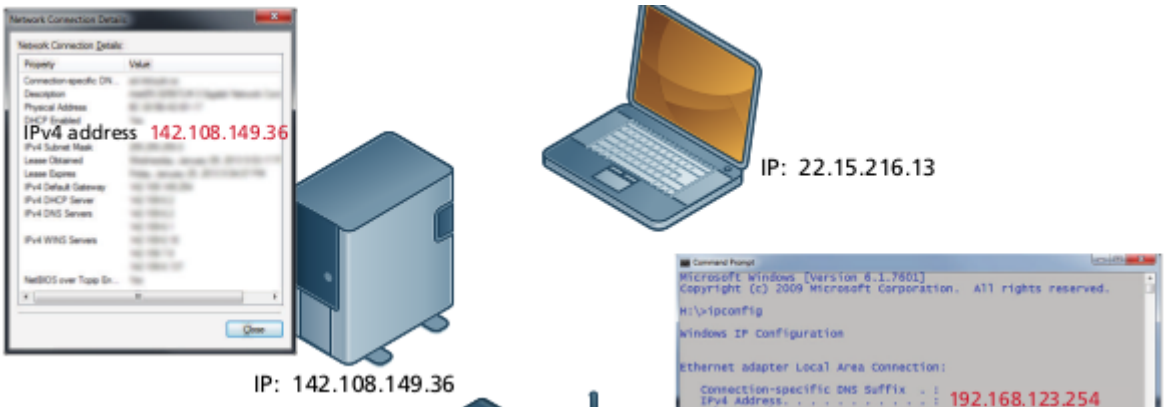
- TCP / IP Internet protokolleri başlangıçta *dört katmanlı bir yığın (four-layer stack)* olarak soyutlanmıştır.
- Daha sonraki soyutlamalar onu beş veya yedi katmana ayırır. Yine de üst katmana odaklandığımız için, Şekil'de gösterilen en eski ve en basit dört katmanlı ağ modelinden bahsedeceğiz.
- Katmanlar bilgileri bir düzey yukarı veya aşağı iletir.
- Alt katmanlar, sinyalleri ağlar üzerinden iletmenin daha temel yönlerini ele alarak daha yüksek katmanların bir client'ın ve server'ın nasıl etkileşimde bulunacağını düşünmesini sağlar. Yani alt katmanlar fiziksel iletişimi gerçekleştirirken üst katmanlar sadece iletişime odaklanmalı.
- Web, tüm katmanların çalışmasını gerektirir, ancak web geliştirmede en yüksek katman olan uygulama katmanına (application layer) odaklanmamız gerekir.

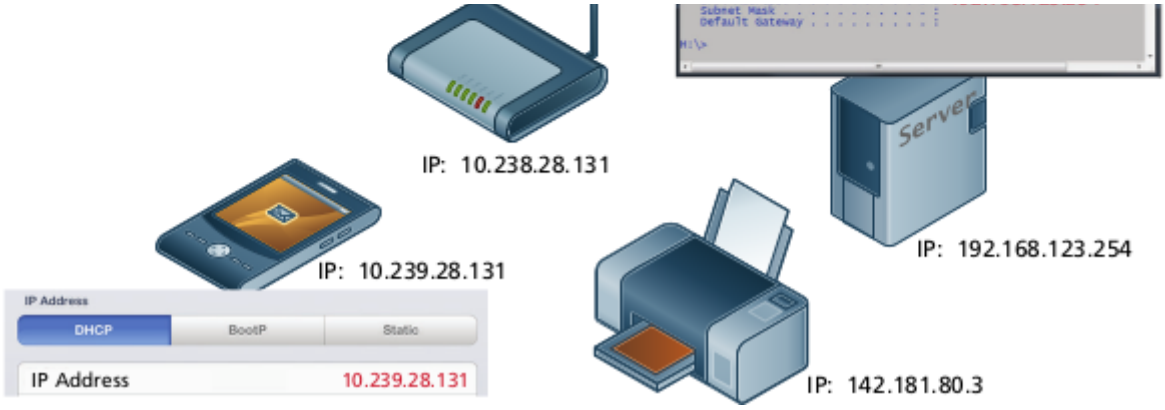
1- Link Layer (Bağlantı Katmanı):

- Bağlantı katmanı, hem *ortam/media* (*wires, wireless — Kablolu, kablosuz*) boyunca *fiziksel iletimden (physical transmission)* hem de *mantıksal bağlantılar (logical links)* oluşturmaktan sorumlu olan en düşük katmandır.
- *Paket oluşturma(packet creation), iletim(transmission), alım(reception), hata tespiti (error detection), çarpışmalar (collisions), hat paylaşımı (line sharing)* ve daha fazlası gibi sorunları ele alır.
- Burada, bazen İnternet bağlamında kullanılan bir terim olan, *MAC (Media Access Control)* adresleri mevcuttur.
- MAC Adresleri, *ağ donanımına (network hardware)* atanan ve fiziksel ağ düzeyinde (physical networking level) kullanılan benzersiz 48 veya 64 bit tanımlayıcılardır. (unique 48- or 64-bit identifiers)

2- Internet Layer (Internet Katmanı):

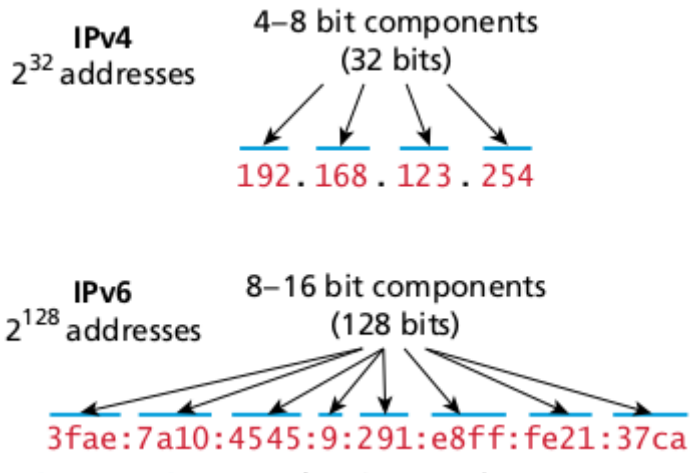
- İnternet katmanı, (bazen IP Katmanı/IP Layer olarak da adlandırılır) paketleri ağlar arasındaki iletişim ortakları (communication partners) üzerinden yönlendirir.
- İnternet katmanı "en iyi çaba" iletişimini ("best effort" communication) sağlar.
- İletiyi hedefe gönderir, ancak yanıt beklemez ve mesajın/iletinin bozulmadan elinize ulaşacağına dair garanti sağlamaz
- İnternet, İnternet üzerindeki hedefleri tanımlamak için *İnternet Protokolü (IP)* adreslerini kullanır.
- İnternet'e bağlı her cihazın, onu benzersiz bir şekilde tanımlayan (uniquely identify), IP adresi (IP address) vardır.





IP adresleri ve Internet

- IP adreslerinin ayrıntıları bir web geliştiricisi için önemli olabilir.
- Belirli bir *web requestinin* IP adresini *izlemesi (track)*, *kaydetmesi (record)* ve *karşılaştırması (compare)* gereken durumlar vardır. (Örneğin, çevrimiçi anketlerin, aynı adresin birden fazla oy kullanmadığından emin olmak için IP adreslerini karşılaştırması gerekir.)
- İki tür IP adresi vardır: *IPv4* ve *IPv6*.
- IPv4 adresleri, orijinal TCP/IP protokolünün IP adresleridir.
- IPv4, dört adet 8 bitlik tamsayı olarak uygulanır/implemente edilir.
- İşaretsiz bir 8 bitlik tam sayının maksimum değeri 255 (an unsigned 8-bit integer's maximum value is 255) olduğundan, dört adet 8-bitlik tam sayı birlikte yaklaşık 4,2 milyar benzersiz IP adresini kodlayabilir.



IPv4 ve IPv6 Karşılaştırması

- *Public IP adresiniz* genellikle size *internet servis sağlayıcınız (Internet Service Provider -ISP)* tarafından atanacaktır.

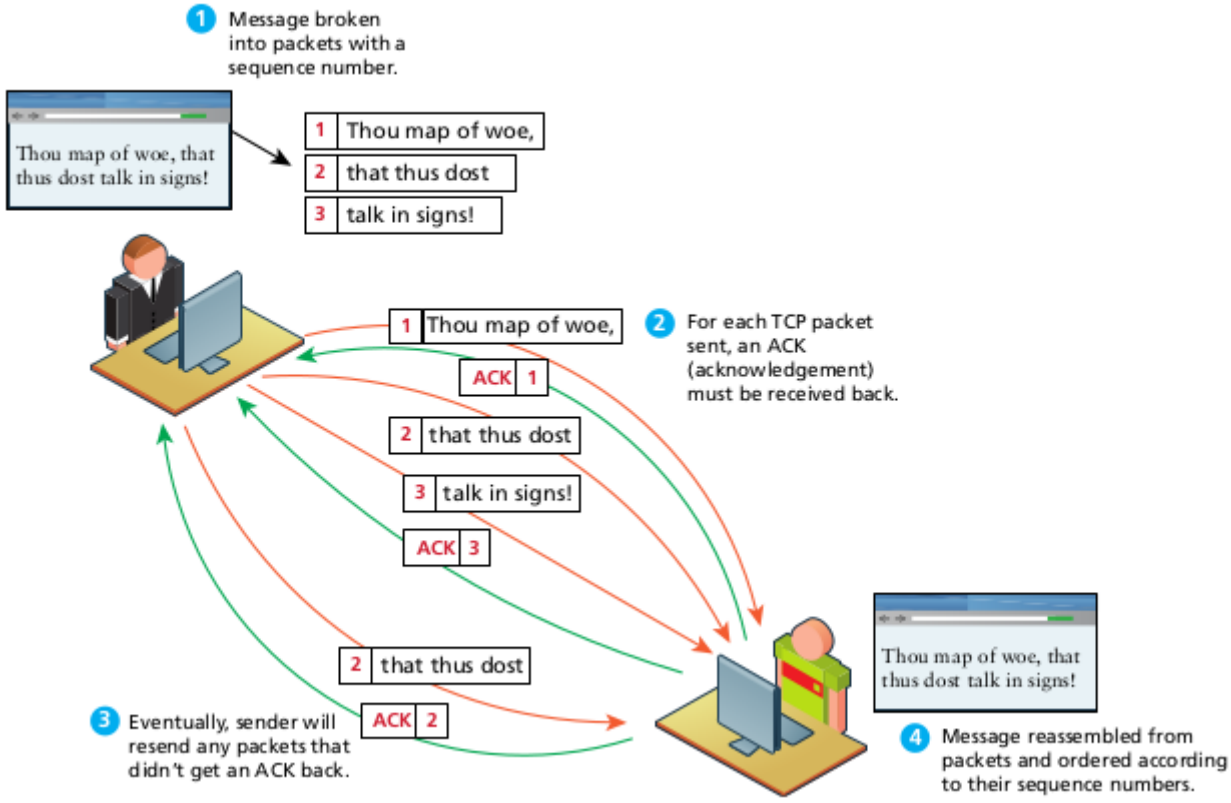
- Yerel bir ağda (local network), bilgisayarlar aralarında tek bir harici IP adresi (Public IP address) paylaşabilir. Yani Yerel ağdaki makineler yerel ağ dışında tek bir Public IP adresini kullanır.
- Örneğin 192.168.0.0 ila 192.168.0.255 aralığındaki IP adresleri tam olarak bu yerel alan ağı kullanımı için ayrılmıştır. Bu IP adreslerine ise *Private IP adresleri* denir.
- Bu nedenle bağlantınızın yalnızca lokal ağ tarafından bilinen bir lokal IP'si (Internal IP/Private IP) (Örneğin 192.168.0.15 gibi) ve dış dünya için adresiniz olan başka bir genel IP (External IP/Public IP) adresi olabilir.
- IP adreslerinin uzunluğu 32 Bit olduğundan maks. Host sayısı 4,2 milyar ile sınırlandırılmıştır.
- IPv6, IPv4 versiyonunda ki IP adresi sayısının yetersiz kalma ihtimaline karşı geliştirildi.
- IPv6 sürümünde IPv4'teki sayının milyar milyar katından fazla olan benzersiz adres (unique addresses) için sekiz adet 16 bitlik tam sayı kullanır. (Bu 16 bitlik tamsayılar uzunluklarından dolayı normalde hexadecimal(onaltılık) tabanda yazılırlar.)

3- Transport Layer (İletim Katmanı):

- Taşıma katmanı, *transmisyonların(transmissions)* hatasız ve düzenli olarak gelmesini sağlar. Bu, birkaç mekanizma ile gerçekleştirilir.
- İlk olarak, veri, *TCP'ye (Transmission Control Protocol — İletim Kontrol Protokolü)* göre biçimlendirilmiş *paketlere* bölünür/ayrılır. Bu paketlerdeki verilerin boyutu 0 ila 64K arasında değişebilir, ancak pratikte tipik paket veri boyutu yaklaşık 0.5 ila 1K arasındadır.
- Her veri paketinde bir *sıra numarası (sequence number)* içeren bir *başlık(header*) vardır, böylece *alıcı(receiver)*, veriler ne zaman gelirlerse gelsin orijinal mesajı tekrar sıraya koyabilir.
- İkincisi, her paketin gönderiminden sonra gönderene geri bildirim yapılır (*ACK Paketi — acknowledged)*, böylece bir paketin kaybedilmesi durumunda, *verici (transmitter)*, bir paket için ACK gelmediğinden o paketin kaybolduğunu fark

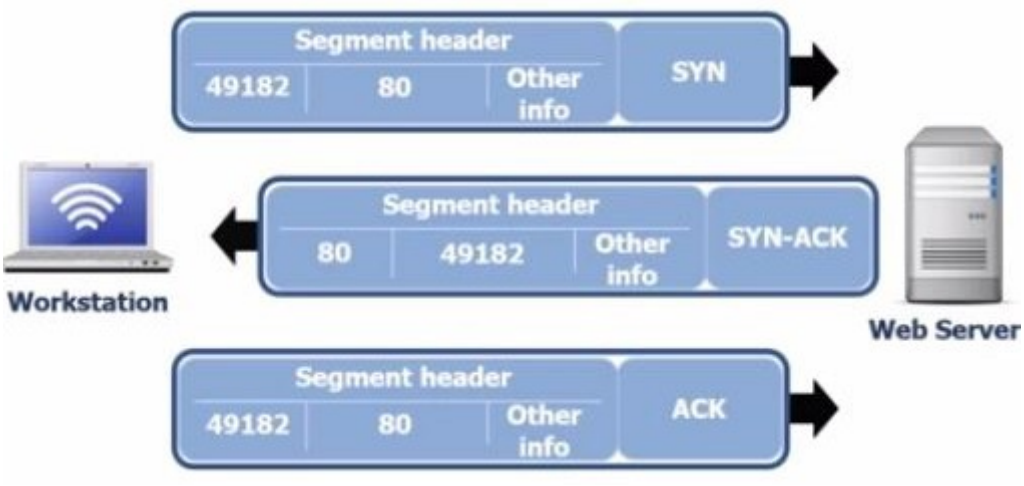
edecektir. Bu paket tekrar iletilir *(retransmitted)* ve sırasız (out of order) olmasına rağmen, Şekil'de gösterildiği gibi varış yerinde yeniden sıralanır.

- Bu, gönderilen mesajların geleceğini ve bunun düzenli olacağını garanti ettiğiniz anlamına gelir.
- ***NOT (UDP — User Datagram Protocol):***
- Bazen paketlerin garantili iletimini istemeyiz. Örneğin, bir futbol oyununun canlı çok noktaya yayınını düşünün. Milyonlarca abone oyunu yayınlıyor olabilir ve kaybedilen her paketi takip edip yeniden iletemeyiz. Feed'deki küçük bir veri kaybı kabul edilebilir ve müşteriler yine de oyunu görecektir.
- Bu senaryolarda TCP yerine *UDP (User Datagram Protokolü — Kullanıcı Datagram Protokolü)* adı verilen bir İnternet protokolü kullanılır.
- UDP hizmetlerine/servislerine diğer örnekler arasında *Voice Over IP (VoIP)*, birçok çevrimiçi oyun ve *DNS (Domain Name System — Alan Adı Sistemi)* sayılabilir.



- Ayrıca burada gerçekleşen bir olay da *3-Way Handshake (3'lü el sıkışma)* mekanizmasıdır.

- ***3-Way Handshake:*** TCP protokolü kullanılarak client ve server arasında gerçekleşen bu olay sonucu bağlantı gerçekleştirilir. (Aksi durumlar hariç) Bu mekanizma, client'ın SYN paketi ile başlar, karşı taraftan da SYN+ACK paketinin gelmesi halinde (ACK paketi, client'ın SYN paketini karşılarken SYN paketi de client tarafından bir ACK paketi almak adına gönderilir) client'ında bu paketlere ACK işaretli paketle yanıt vermesi sonucunda bağlantı sağlanır.



TCP 3'lü El Sıkışma Mekanizması

4- Application Layer (Uygulama Katmanı):

- Uygulama katmanı ile, çoğu web geliştiricisine tanıdık gelen protokoller seviyesindeyiz.
- Uygulama katmanı protokolleri, *süreçler arası iletişimi (process-to-process communication)* uygular ve altındaki katmanlardaki düşük düzeyli paket (low-level packet) ve IP adresi protokollerine kıyasla daha yüksek bir soyutlama düzeyindedir.
- Birçok uygulama katmanı protokolü vardır. Web geliştiricileri için yararlı olanlardan bazıları şunlardır:
- **HTTP (Hypertext Transfer Protocol):** HTTP web iletişimi için kullanılır.
- **SSH (Secure Shell Protocol):** SSH, sunuculara uzaktan komut satırı bağlantılarına izin verir.
- **FTP (File Transfer Protocol):** FTP, bilgisayarlar arasında dosya aktarımı için kullanılır.

- **POP / IMAP / SMTP:** E-posta aktarma ve depolama (transferring and storing) için e-posta ile ilgili protokoller.
- **DNS (Domain Name System):** *DNS, Alan adlarını(Domain Names*) IP adreslerine *çözümlemek(resolving)* için kullanılan Alan Adı Sistemi protokolü.

OSI Modeli vs TCP/IP Modeli

- İki protokolde iletişimi katmanlarla tanımlamaktadır.
- Katmanlar görevleri kullanım görevlerine göre tanımlanmıştır.
- OSI modelindeki 7 katmana karşılık TCP/IP modelinde 4 katman belirlenmiştir.
- OSI modeli daha çok iletişimde standartı belirtmeye yönelmekteyken TCP/IP ise daha çok uygulamaya yönelmektedir.
- TCP/IP ve ilgili protokollerin kullanımı hızla artmaktadır. Buda TCP/IP'nin OSI modeline göre daha uygulanabilir bir model olduğunu göstermektedir.
- TCP/IP nin daha uygulanabilir olasının şu nedenlerle söyleyebiliriz.
  1. TCP/IP mevcuttur ve uygulamaları çalışmaktadır.
  2. TCP/IP protokol ailesini kullanan geniş bir ürün yelpazesi mevcuttur.
  3. *IAB (Internet Advisory Board)* tarafından sağlanan iyi-kurulmuş ve fonksiyonel bir yönetim yapısı vardır.
  4. Dokümanlara kolay ulaşım sağlar.
- OSI modeli üç merkezde açıklana bilir. *Servisler, Arayüzler, Protokoller*.

1. Her katman bazı servisler içerir. Bu servisler katmana nasıl erişileceğini ve katmanın nasıl çalışacağını tanımlar.
2. Katmanların arayüzleri ise erişim işlemini nasıl çalışacağını belirler. Bu bazı parametreler ile sağlanır.
3. Son olarakta katman protokoller kullanır. Bu protokoller o katmanın görevlerini belirler

Diğer Ağ Terimleri/Kavramları

Default Gateway

- Default Gateway, kabaca lokal ağınızın internete çıkış noktasının IP adresidir.

- Örneğin evde modem(ya da router) kullanarak internete çıkıyorsanız modeminizin iç ağ tarafında (lokal ağ) almış olduğu IP adresi kullanıcı için default gateway olarak adlandırılır.

- (Daha fazla bilgi için bkz. firatboyan.com/default-gateway-varsayilan-ag-gecidi-nedir.aspx ve networkingguides.com/what-default-gateway/ ayrıca lifewire.com/what-is-a-default-gateway-817771 adreslerinden faydalanabilirsiniz.)

IP Subnetting

- Lokal ağda kullanılan IP adreslerini, Private IP adresleri olarak adlandırmıştık.

- Bu IP adreslerini şu tabloda gösterelim:

> ***10.0.0.0–10.255.255.255*** *adresleri arasındaki IP adresleri: A sınıfı Private IP Adresleri*
>
> ***172.16.0.0–172.31.255.255*** *adresleri arasındaki IP adresleri: B sınıfı Private IP Adresleri*
>
> ***192.168.0.0–192.168.255.255*** *adresleri arasındaki IP adresleri: C sınıfı Private IP Adresleri*

- 192.168.1.12/25 gösterindeki /25 ifadesi *Subnet Mask* olarak adlandırılır.

- IP Adresleri ve ağ maskesi birlikte kullanılarak cihazların ağ adresleri tespit edilir ve aynı ağda olup olmadıkları anlaşılır.

- Subnet Mask, binary düzende yazıldığında, içerisindeki 1'lerin toplamı IP adresinin sağına yazılır. (192.168.1.12/25 örneğinde olduğu gibi. Burada 25 sayısı, 1'lerin toplam sayısıdır.)

- Örneğin /25, 255.255.255.128 subnet mask anlamına geliyor. (255.255.255.128 ifadesini binary düzlemde yazdığınızda, 11111111.11111111.11111111.10000000 gibi bir ifade elde ederiz ki bu ifadedeki 1'lerin toplam sayısı 25tir.)

- Buradan, IP adresini ve subnet mask'ı binary düzende yazıp *AND* işlemine tabii tuttuğumuzda ise sonuç bize *Ağ Adresini* verir.

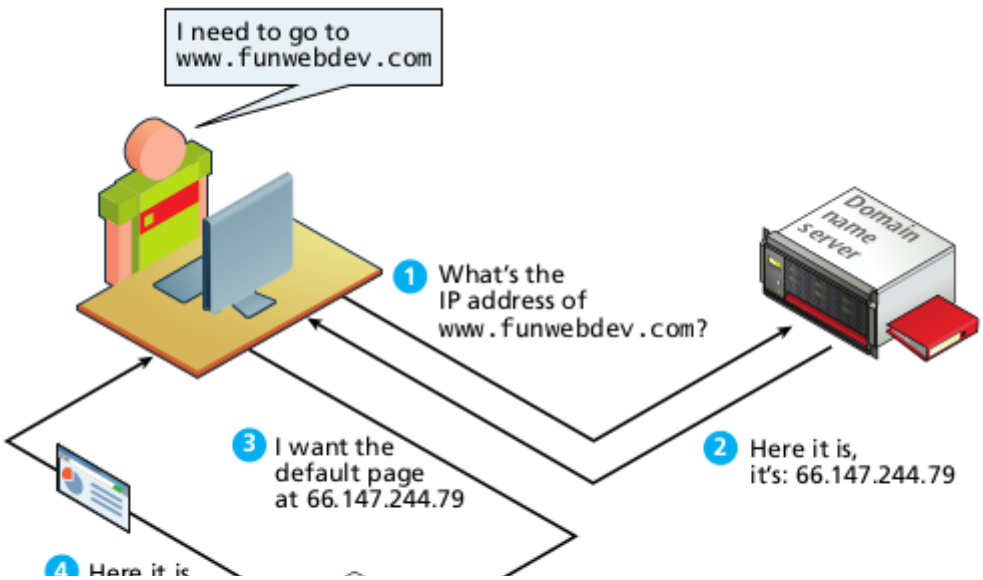
> ***IP Adresi*** *→ 11000000.10101000.00000001.00001100*

> ***Subnet Mask*** *→11111111.11111111.11111111.10000000*
>
> ***Network Address*** *→11111111.11111111.11111111.10000000*

- İki farklı makinede yapılan bu işlem sonucunda ağ adresleri aynı çıkan makineler aynı ağda, farklı çıkan makineler ise farklı ağda algılanır.
- Burada kullanmak için ***ipcalc*** aracı mevcuttur.

DNS (Domain Name System — Alan Adı Sistemi)

- İnsanlar uzun sayı dizilerini hatırlamaktan hoşlanmazlar. *Etki alanları (domain)* yerine IP adreslerini hatırlamanız gerektiğinde internetin ne kadar tatsız olacağını düşünebilirsiniz. (Google.com yerine 173.194.33.32 yazmanız gerekir. Facebook'u ziyaret etmek için 69.171.237.24 yazmanız gerekiyorsa, sosyal ağların daha az popüler bir eğlence olması muhtemeldir.)
- ARPANET günlerinde bile, araştırmacılar IP adreslerine domainler atadı.
- İnternet hostlarının sayısı azdı, bu yüzden birkaç yüz domain ve IP adresinin bir listesi, Stanford Araştırma Enstitüsü'nden bir host dosyası (hosts file) olarak gerektiği gibi indirilebilir. (Pro Tip'e bakınız). Bu *anahtar/değer çiftli (key-value pairs) domain names* ve IP adresleri, kişilerin, IP adresi yerine domain adını kullanmasına izin verdi.
- İnternet'teki bilgisayar sayısı arttıkça, bu host dosyasının daha iyi, daha *ölçeklenebilir (scalable)* ve *dağıtılmış (distributed)* bir sistemle değiştirilmesi gerekiyordu. Bu sistem *DNS (Domain Name System)* olarak adlandırılır ve Şekil'de en basitleştirilmiş biçimde gösterilir.



DNS'e genel bakış

- DNS, İnternetin core sistemlerinden biridir. (DNS, e-posta için de kullanılır)
- DNS sisteminin avantajlarından biri, bir serverin domain adını, IP konumundan (IP Location) ayırarak bir site, sitenin adını değiştirmeden başka bir konuma taşınabilir. Bu, sitelerin ve e-posta sistemlerinin hizmeti kesintiye uğratmadan daha büyük ve daha güçlü tesislere geçebileceği anlamına gelir.
- Tüm *request-response döngüsü* bir saniyeden az sürebileceğinden, tüm web ve e-posta uygulamalarınızda, DNS requestlerinin gerçekleştiğini unutabiliriz.
- DNS sisteminin bilinmesi ve anlaşılması, web sistemlerinin geliştirilmesi (developing), güvence altına alınması (securing), devreye alınması (deploying), sorunlarının giderilmesi (troubleshooting) ve sürdürülmesinde(maintaining) başarı için esastır.
- ***PRO TIP:***
- Bu ilk günlerin kalıntıları yani hosts dosyası(hostların dosyası hosts file) modern bilgisayarlarda hala bulunur. *(UNIX sistemler tipik olarak etc/hosts konumundadır)* Bu dosyanın içinde domain name eşleşmelerini (mapping) aşağıdaki biçimde görürsünüz:

> *127.0.0.1 Localhost* ***SomeLocalDomainName.com***

- Bu mekanizma sıklıkla, kendi bilgisayarlarımızda, adres çubuğundaki(address bar) gerçek domain adlarıyla(real domain names) web siteleri geliştirmemize yardımcı olmak için kullanılır.
- Aynı hosts dosya mekanizması (hosts file mechanism), kötü niyetli bir kullanıcının belirli bir domain için hedeflenen trafiği (traffic destined) yeniden yönlendirmesine (reroute) de izin verebilir.
- Kötü niyetli bir kullanıcı 123.56.789.1'de bir sunucu çalıştırdıysa, bir kullanıcının hostlarını facebook.com'u kötü amaçlı sunucularına işaret edecek şekilde değiştirebilir(modify). Son client daha sonra tarayıcısına facebook.com yazacak ve

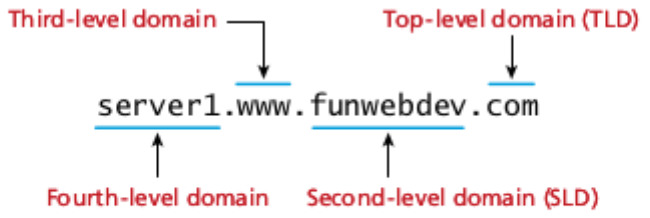
bu trafiği meşru facebook.com serverlarına yönlendirmek (routing) yerine programcının kimlik avı (phish) yapabileceği veya verileri çalabileceği kötü amaçlı siteye (malicious site) gönderilecekti.

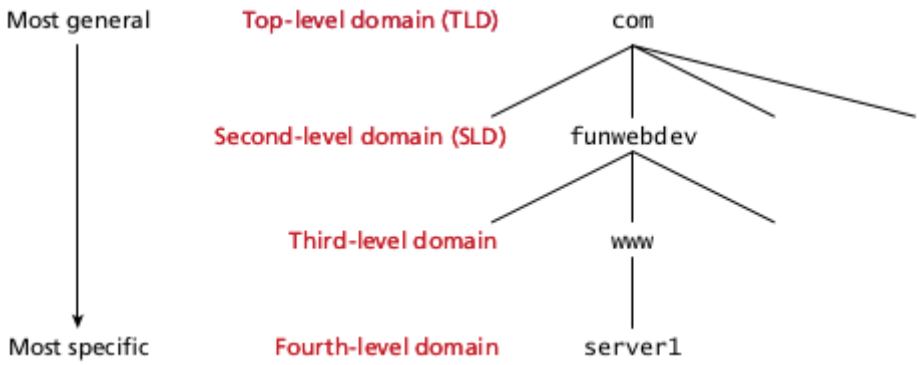
> *123.456.678.1* ***facebook.com***

- Bu nedenle, birçok sistem yöneticisi (system administrators) ve çoğu modern işletim sistemi, yönetici parolası (administrator password) olmadan bu dosyaya erişime izin vermez.

1. **Name Levels(İsim Seviyeleri)**

- Bir domain name birkaç bölüme ayrılabilir. Bir hiyerarşiyi temsil ederler, en sağdaki kısımlar internet adlandırma hiyerarşisinin "*üst (top)*" kısmındaki köke en yakın olanlardır. Tüm domain namelerinin en az bir *üst düzey domain(TLD top-level domain)* adı ve *ikinci düzey domain (SLD second-level domain)* adı vardır.
- Çoğu web sitesi ayrıca *üçüncü düzey (third-level)* bir *www alt domaini (subdomain)* ve belki de diğerlerini barındırır/sağlar(maintain).
- Şekilde, dört seviyeli bir domain (domain with four levels.) göstermektedir.
- Domain name'in en sağ kısmına (en sağ periyodun sağında) en üst düzey domain(top-level domain) denir. Bir domain name'in en üst düzeyi (top level) için iki geniş kategoriyle (two broad categories) sınırlıyız. Artı üçüncüsü diğer kullanım için ayrılmıştır (reserved). Onlar:





Domain seviyeleri

- ***Generic top-level domain (gTLD):***
- TLD'ler .com, .net, .org ve .info'yu içerir.
- TLD'ler .gov, .mil, .edu ve diğerlerini içerir. Bu domainlerin sahiplik gereksinimleri olabilir ve bu nedenle yeni ikinci düzey domainlerin yeni bir adres edinmeden önce sponsordan izin alması gerekir.
- ***Country code top-level domain (ccTLD):***
- TLD'ler arasında .us, .ca, .uk ve .au bulunur.
- Bu kodlar temsil ettikleri ülkelerin kontrolü altındadır, bu yüzden her biri farklı şekilde yönetilir.
- Bazı ülkeler kendi dillerinde batı dışı karakterler kullandığından, *uluslararasılaştırılmış üst düzey domain name(IDN — internationalized top-level domain name)* kavramı da son yıllarda büyük bir başarıyla test edilmiştir. (Bazı örnek IDN'ler, sırasıyla http://παράδειγμα.δοκιμή, http://例 え. テ ス ト ve http://لاثم.رابتخإ adreslerinde test alanlarına sahip Yunanca, Japonca ve Arapça domainlerini (diğerleri arasında) içerir.)
- ***arpa:***
- *.arpa domaini*, ilk atanan üst düzey domaindir. Hala atanır ve *ters DNS aramaları (reverse DNS lookups)* için kullanılır (yani, bir IP adresinin domain name'ini bulma).
- Funwebdev.com gibi bir domainde, ".com" en üst düzey domaindir, funwebdev ise ikinci düzey domain olarak adlandırılır.
- Kayıt şirketi tarafından dayatılanların dışında ikinci düzey domainlerde çok az kısıtlama vardır. Uluslararası domain name'ler dışında A-Z, 0–9 karakterleri ve "-" karakteri ile sınırlandırılmıştır. Domain name'ler büyük/küçük harfe duyarlı olmayan karakterler (case-insensitive characters) olduğundan, a-z de birbirinin yerine kullanılabilir.
- İkinci düzey bir domainin sahibi, isterse subdomainlere sahip olmayı seçebilir; bu durumda, bu subdomainler, temel(base) hostname eklenir. Örneğin, exam-

answers.webdevfun.com adresini domain name olarak oluşturabiliriz; burada exam-answers subdomaindir.

- ***NOT:***

- İstersek alt alt alanlar (sub-subdomains) oluşturabiliriz. Subdomainlerin her bir düzeyi, hostname'in önüne eklenir. Bu, üçüncü seviyeye, dördüncü seviyeye, vb. izin verir. Bu ağdaki bilgisayarları domain içinde tek tek tanımlamak için kullanılabilir.

2. **Address Resolution (Adres Çözümleme):**

- Domain adları, kullanıcıların bir web sitesine başvurması(reference) için kesinlikle daha kolay bir yol olsa da, sonunda browserınızın web sitesinden herhangi bir kaynak talep edebilmesi için web sitesinin IP adresini bilmesi gerekir.

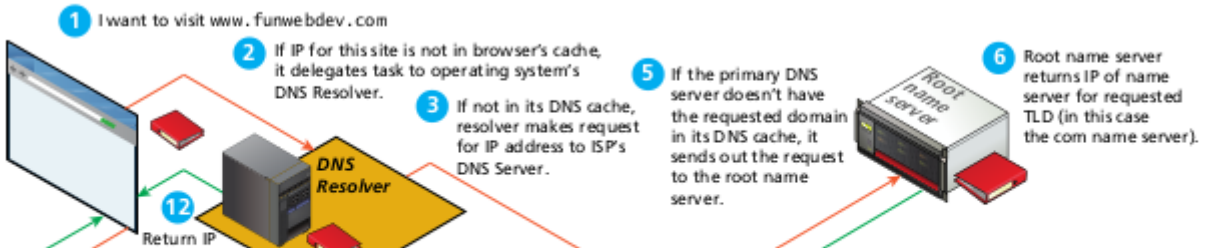
- DNS, yazılımın bu sayısal IP adresini keşfetmesi için bir mekanizma sağlar. Bu işleme burada *adres çözümleme (address resolution)* denir.

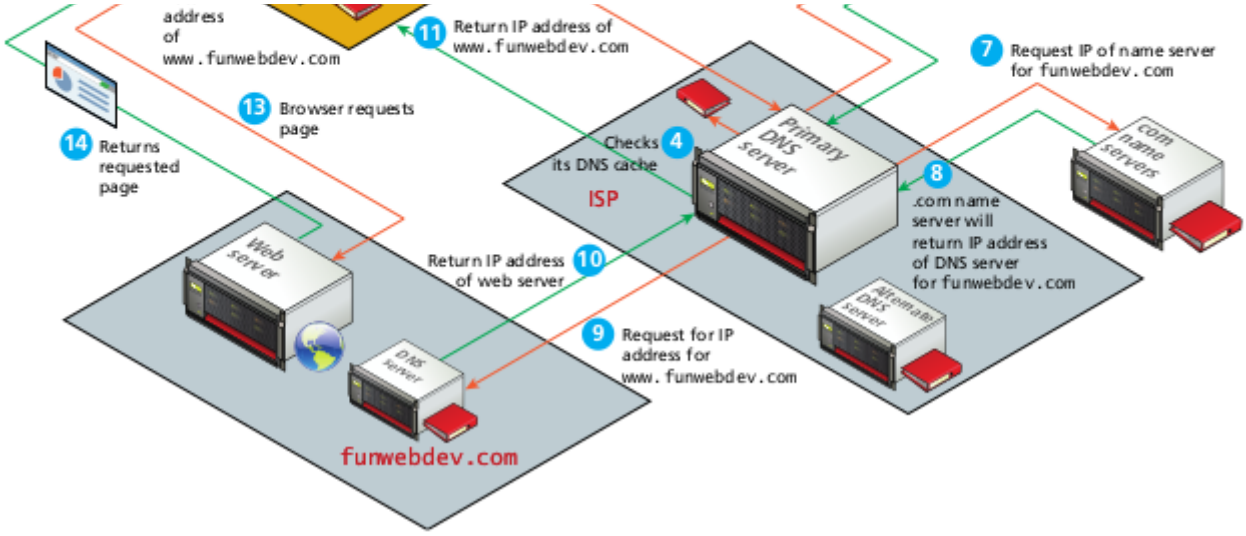
- İlk şekilde gösterildiği gibi, bir domain adı istediğinizde, *domain name server* olarak adlandırılan bir bilgisayar o domainin IP adresini döndürür. Bu IP adresiyle, browser daha sonra söz konusu domain için web serverdan bir kaynak için istekte bulunabilir.

- İlk şekil, *adres çözümleme sürecine (address resolution process)* ilişkin net bir genel bakış sağlarken, oldukça basitleştirilmiştir (simplified). Adres çözümlemesi sırasında gerçekte ne olduğu, aşağıdaki şekilde görüldüğü gibi daha karmaşıktır.

- DNS bazen *name serverlarının dağıtılmış veritabanı sistemi (distributed database system of name servers)* olarak da adlandırılır.

- Bu sistemdeki her server, domainlerle ilgili soruları yanıtlayabilir ya da domainlerle ilgili soruların yanıtlarını arayabilir veya bu yolla önbelleğe alabilir. (caching) Bir clientın bakış açısından bu, benzersiz bir adı (unique name) bir numarayla eşleyen(mapping) telefon defteri gibidir.





domain adı adres çözümleme süreci

1. *Çözümleme işlemi (Resolution process)* kullanıcının bilgisayarında başlar. www.funwebdev.com domaini istendiğinde (belki bir bağlantıyı tıklayarak veya bir URL yazarak) browser, domainin önbelleğinde(cache) önceden IP adresi olup olmadığını kontrol ederek başlar. Eğer mevcut ise, şemada 13. adıma atlayabilir.

2. Browser istenen sitenin IP adresini bilmiyorsa, görevi, bir yazılım aracısı olan *DNS çözümleyicisine (DNS Resolver)* devredecektir. (DNS Resolver işletim sisteminin bir parçasıdır.) DNS çözümleyici ayrıca sıkça istenen domainleri önbelleğini tutar; istenen domain önbelleğinde ise, işlem adım 12'ye atlar.

3. DNS çözümleyici sıkça istenen domainlerin önbelleğini tutar; istenen domain önbelleğinde değilse, dış yardım istemelidir; bu durumda yardım istenilen yer, DNS requestlerini işleyen özel bir server olan yakındaki bir *DNS Serverdır*. Bu, İnternet servis sağlayıcınızdan (ISP) veya üniversitenizden veya genellikle kurumsal IT departmanınızdan bir bilgisayar olabilir. Bu yerel DNS sunucusunun adresi, bilgisayarınızın işletim sisteminin ağ ayarlarında saklanır. Bu server, domain name/IP adresi çiftlerinin daha önemli bir önbelleğini tutar. İstenen domain önbelleğinde ise, işlem adım 11'e atlar.

4. Local DNS serverının, önbelleğinde domaine ait IP adresi yoksa, yanıt için diğer DNS serverlarından istemesi gerekir. Neyse ki, domain sisteminde yerleşik bir fazlalık (redundancy) vardır. Bu, genel olarak, verilen herhangi bir DNS Requesti için yanıtları olan birçok serverın olduğu anlamına gelir. Bu fazlalık sadece yerel düzeyde (Local level) değil, (örneğin, yukarıdaki şekilde, ISP'nin birincil DNS Serverı (primary DNS server) ve alternatif bir serverı da vardır) küresel düzeyde de mevcuttur.

5. Local DNS server, alternatif bir DNS serverından gelen requestin yanıtını bulamazsa, ilgili *TLD name serverından (Top-level domain name server — TLD Name Server)* alması gerekir. Funwebdev.com için bu .com'dur. Local DNS serverımızın önbelleğinde uygun TLD name serverlarının adreslerinin bir listesi zaten olabilir. Böyle bir durumda, işlem adım 7'ye atlayabilir.

6. Local DNS serverı istenen TLD serverının adresini zaten bilmiyorsa (örneğin, yerel DNS serverı ilk başlatıldığında bu bilgilere sahip olmaz), o zaman bir *root name serverdan* bu bilgileri istemesi gerekir. DNS root name serverları, TLD name serverlarının adreslerini depolar.

7. İstenen domain için *TLD ad sunucusunun* adresini aldıktan sonra, local DNS artık TLD ad sunucusundan istenen etki alanının adresini isteyebilir. Domain kayıt işleminin bir parçası olarak, domain'in DNS sunucularının adresi TLD name serverlarına gönderilir, bu nedenle 8. adımda yerel DNS serverına döndürülen bilgiler budur.

8. Kullanıcının local DNS serverı artık talep edilen domaini (www.funwebdev.com) DNS Serverından (second-level name server da denir) isteyebilir; söz konusu domain için web Serverının doğru IP adresini alması gerekir. Bu adres kendi önbelleğinde (cache) saklanacak, böylece bu domain için gelecekte yapılacak requestler daha hızlı olacaktır. Bu IP adresi, son olarak, adım 11'de gösterildiği gibi, istekte bulunan bilgisayardaki DNS çözümleyicisine (DNS Resolver) döndürülebilir.

9. Browser, 12. adımda gösterildiği gibi, istenen domain için doğru IP adresini alır. **Not:** Local DNS serverı IP adresini bulamadıysa, başarısız bir yanıt döndürür ve bu da browserın bir hata mesajı (error message) görüntülemesine neden olur.

10. Şimdi istenen IP adresini bildiğine göre, browser nihayet isteği web serverına gönderir (send out the request to the web server), bu da web serverının istenen kaynakla yanıt vermesine neden olur. (Adım 14)

- Bu işlem aşırı karmaşık görünebilir, ancak uygulamada DNS serverlarının sonuçları önbelleğe alması nedeniyle çok hızlı bir şekilde gerçekleşir. Server funwebdev.com'u çözdüğünde (resolves), funwebdev.com'daki sonraki kaynak istekleri daha hızlı olacaktır, çünkü root serverlarda yeniden başlamak yerine IP adresi için yerel olarak saklanan cevabı kullanabiliriz.

- Sistem genelinde önbelleğe almayı kolaylaştırmak için, tüm DNS kayıtlarında (DNS records), name serverını istemeden önce sonucun ne kadar süre önbelleğe alınacağını öneren bir *yaşam süresi (TTL — Time To Live)* alanı bulunur. Bu mekanizma DNS sisteminin verimliliğini ve yanıt süresini geliştirse de, değişikliklerin tüm serverlar arasında yayılmasını geciktirmenin bir bedeli vardır. Yöneticilerin (administrators), bir DNS girdisini (DNS entry) güncelledikten sonra, tüm client ISP önbelleklerine (caches) yayılmasını beklemesi gerekir.

- **NOT:**

- Her web geliştiricisi, ad serverlarını siteyi barındıran(hosting) web serverına işaret etme uygulamasını/pratiğini anlamalıdır. Çoğu kullanıcı, web alanı (web space) satın aldıkları şirketin domaini kaydettirdikleri yerle aynı olması gerekmediğinin farkında değildirler.

URL (Uniform Resource Locators)

- Clientın serverdan dosyayı nasıl isteyeceğini bilmesi için bir adlandırma mekanizması gereklidir.

- Web için bu adlandırma mekanizması *URL'dir (Uniform Resource Locator*).

- Şekilde gösterildiği gibi, iki gerekli bileşenden oluşur: bağlanmak için kullanılan protokol ve bağlanmak için domain (veya IP adresi).

- URL'nin isteğe bağlı bileşenleri *yol/path* (bu, serverda erişilecek dosya veya dizini tanımlayan), bağlanılacak port(bağlantı noktası), bir sorgu dizesi (query string) ve bir parça tanımlayıcıdır(fragment identifier).

URL componentleri

1.1 Protocol (Protokol):

- URL'nin ilk kısmı, kullandığımız protokoldür. (Bir kaç application katmanı protokolünden bahsettik)

- Bu protokollerin çoğu bir URL'de görünebilir ve hangi uygulama protokollerinin kullanılacağını tanımlayabilir.

- Örneğin, ftp://example.com/abc.txt istendiğinde, port 21'den bir *FTP isteği* gönderilirken http://example.com/abc.txt 80 numaralı porttan iletilir.

1.2 Domain (Etki Alanı):

- Domain, kaynak istediğimiz serverı tanımlar.
- DNS sistemi büyük/küçük harfe duyarlı olmadığından (case insensitive), URL'nin bu kısmı büyük/küçük harfe duyarlı değildir.
- Alternatif olarak, domain için bir IP adresi kullanılabilir.

1.3 Port (Bağlantı Noktası/Kapı):

- Opsiyonel olan port özelliği, IANA yetkilisi tarafından tanımlanan varsayılanlar dışındaki portlara bağlantılar belirlememizi sağlar.
- Port, temel TCP/IP protokolü ve bağlanan bilgisayar tarafından kullanılan bir tür yazılım bağlantı noktasıdır (software connection point).
- IP adresi bir bina adresine benziyorsa, port numarası binanın kapı numarasına benzer.
- Port özelliği üretim sitelerinde (production sites) yaygın olarak kullanılmamasına rağmen, istekleri bir test serverına yönlendirmek, bir stres testi yapmak veyahutta Internet filtrelerini atlatmak için kullanılabilir.
- Herhangi bir Port belirtilmezse, bir URL'nin protokol bileşeni hangi Portun kullanılacağını belirler.
- Portun sözdizimi, domainden sonra iki nokta üst üste eklemek ve ardından bir tam sayı port numarası belirtmektir. Bu nedenle, örneğin 888 numaralı porttaki serverımıza bağlanmak için URL'yi http://funwebdev.com:888/ olarak belirtiriz.

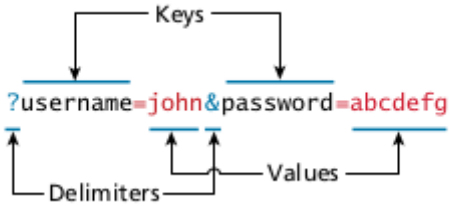
1.4 Path (Yol):

- Yol(Path), daha önce bir bilgisayar dosya sistemi kullanan herkese tanıdık bir kavramdır.
- Web sunucusunun kökü , o serverda bir yerdeki bir klasöre karşılık gelir.
- Birçok Linux serverında yol(path) /var/www/html/ veya benzer bir şeydir (Windows IIS makineleri için genellikle /inetpub /wwwroot/ şeklindedir).

- Path büyük/küçük harfe duyarlıdır (case sensitive), ancak Windows sunucularında büyük/küçük harfe duyarlı olmayabilir.
- Path opsiyoneldir. Bununla birlikte, bir klasör veya bir alanın en üst düzey sayfası istenirken, web serverı size hangi dosyayı göndereceğine karar verecektir. (Apache serverlarında genellikle index.html veya index.php'dir. Windows serverları bazen Default.html veya Default.aspx kullanır. Varsayılan adlar (default names) her zaman yapılandırılabilir ve değiştirilebilir.)

1.6.5 Query String (Sorgu Dizesi):

- HTML formları ve server-side programming hakkında daha fazla bilgi edindiğimizde sorgu dizeleri daha anlamlı bir hal alacaktır.
- Bunlar, clientdan servera kullanıcı formu girişi gibi bilgileri aktarmanın bir yoludur.
- URL'lerde, bunlar "&" simgeleriyle ayrılmış ve öncesinde "?" sembolü İle anahtar/değer çiftleri olarak kodlanmıştır.
- Bir kullanıcı adı ve parolayı kodlayan bir sorgu dizesinin bileşenleri aşğıda gösterilmiştir.



Sorgu dizesi bileşeni

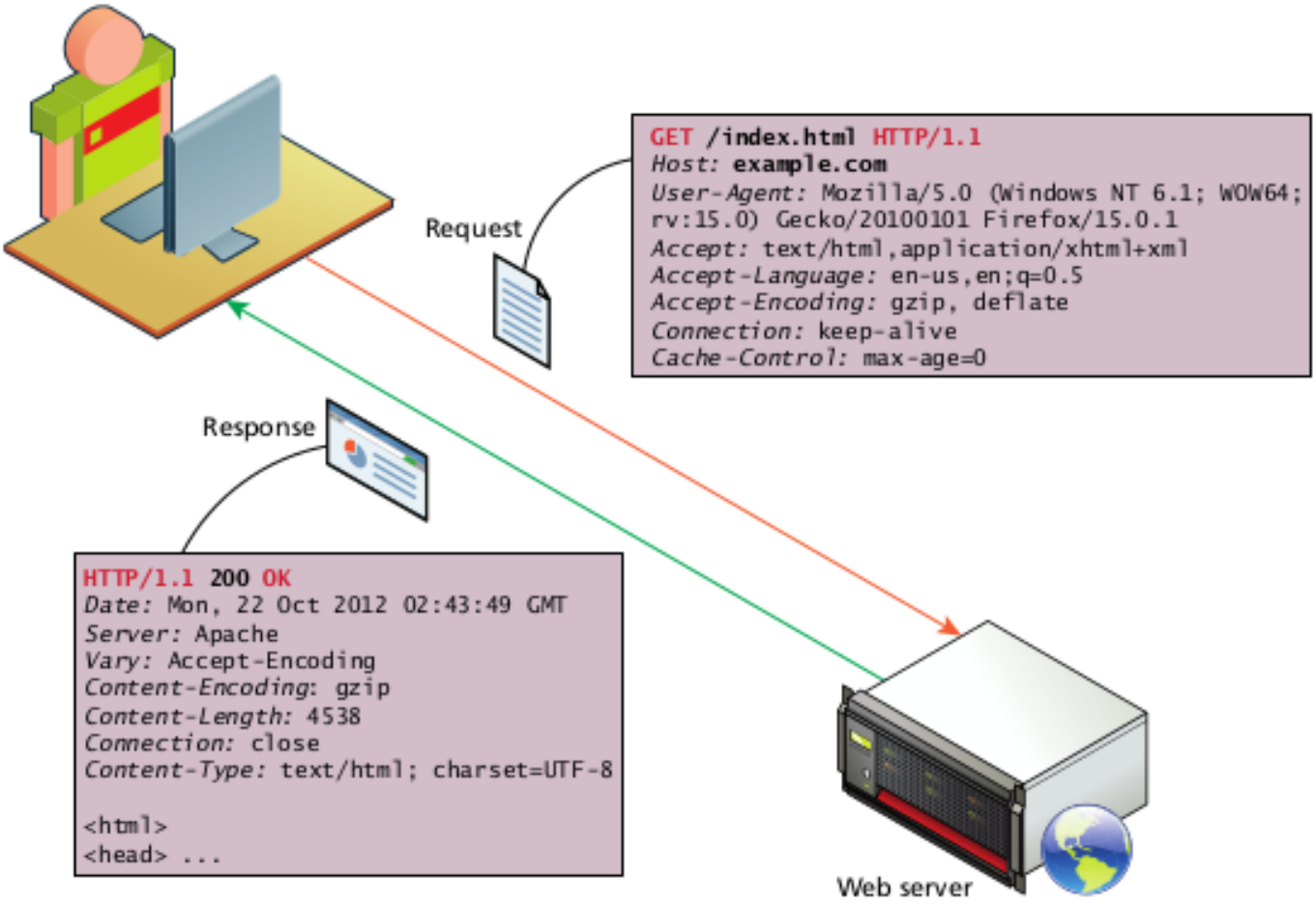
1.6 Fragment (Parça):

- Bir URL'nin son kısmı olan fragment opsiyoneldir.
- Bu, sayfanın bir bölümünü istemenin bir yolu olarak kullanılır.
- Browserlarlar URL'deki fragmenti görür, HTML'deki fragment etiketi çapasını (fragment tag anchor) arar ve web sitesini aşağı doğru kaydırır.
- Birçok eski web sitesinde, fragmentler ve her bölümdeki "back to top"(başa dön) bağlantıları kullanılarak, bu sayfadaki içeriğe bağlantıların bulunduğu bir sayfa

bulunur.

HTTP (Hypertext Transfer Protocol)

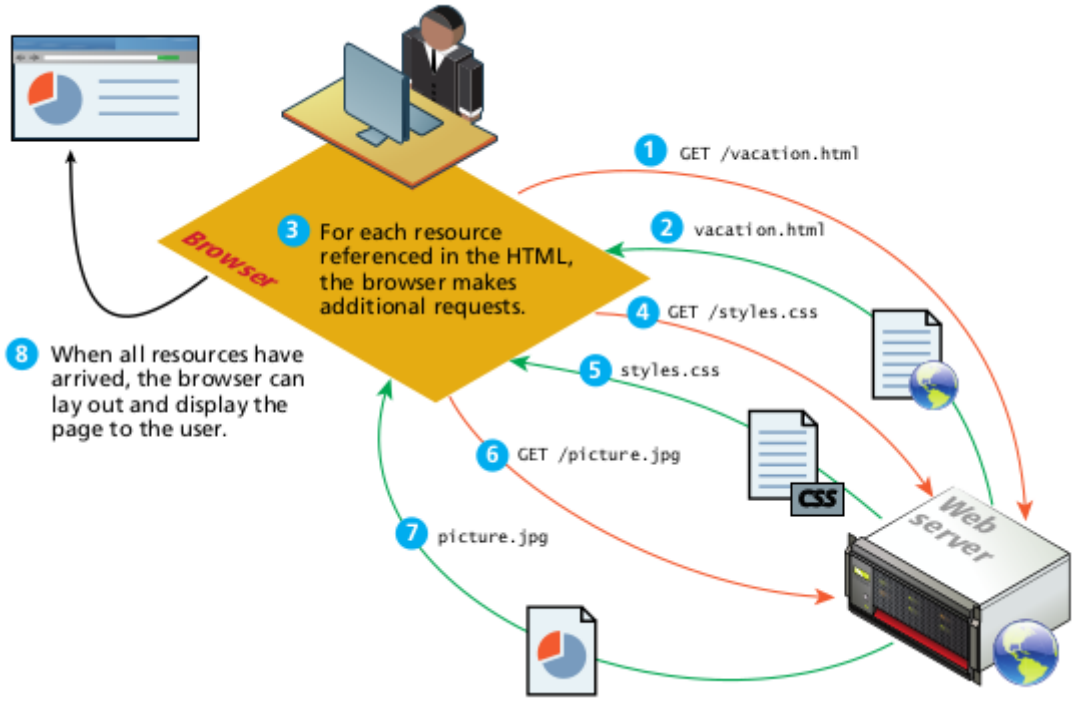
- TCP/IP modelinde, en üst seviyeye ulaşıncaya kadar her biri daha düşük olanlara dayanan birkaç protokol katmanı vardır, uygulama katmanı, SSH (Secure Shell), FTP (File Transfer Protocol) ve World Wide Web'in protokolü, yani HTTP (Hypertext Transfer Protocol) gibi birçok farklı hizmet türüne izin verir.
- HTTP web'in önemli bir parçalarından biridir.
- HTTP, 80. portta (default olarak) bir TCP bağlantısı kurar. Sunucu isteği bekler ve sonra aşağıdaki şekilde gösterildiği gibi bir *yanıt kodu (response code)*, *başlıklar (headers)* ve *isteğe bağlı bir mesajla (dosyaları içerebilir)* yanıt verir.



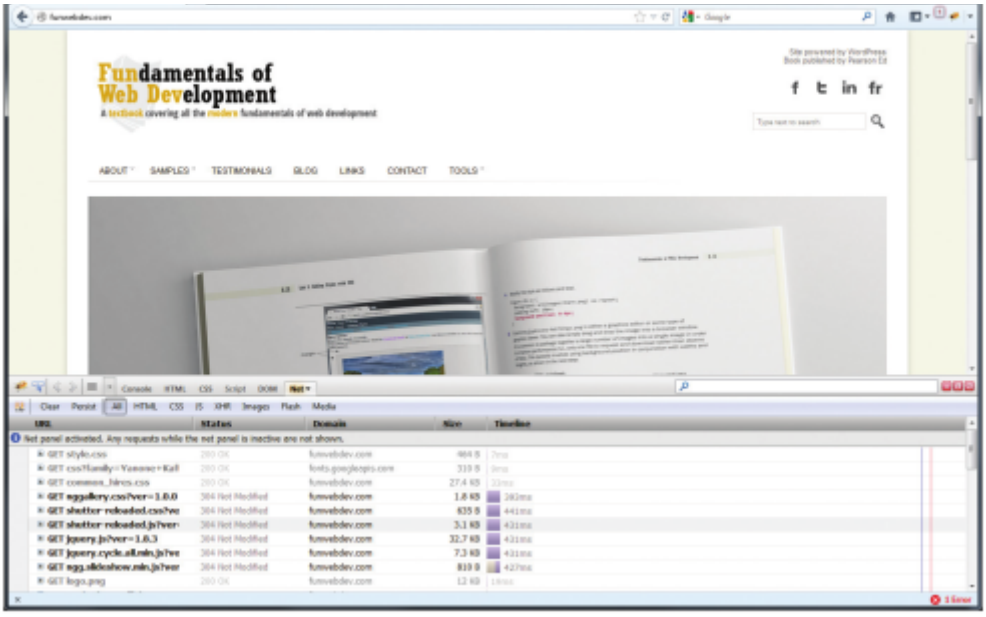
HTTP illüstrasyonu

- Bir web sitesi için kullanıcı deneyimi, geleneksel masaüstü yazılımı kullanıcı deneyiminden (UX -User Experience) farklıdır. Kullanıcılar yazılım indirmez; bir URL'yi ziyaret ederler.

- Bir web sayfasının tamamını bir HTTP cevabında (single HTTP response), döndürmek mantıklı değildir.
- Gerçekte, tek bir web sayfasını görme deneyimi, clientın *baştaki HTML sayfasını (initial HTML Page)* isteyen browserı tarafından kolaylaştırılır ve daha sonra döndürülen HTML içinden referans alınan *resim(images), stil sayfaları (style sheets) ve komut dosyaları (scripts)* gibi tüm kaynakları bulmak için ayrıştırılır(parse).
- Aşağıdaki şekilde gösterildiği gibi, yalnızca tüm dosyalar alındığında sayfa kullanıcı için tam olarak yüklenir.



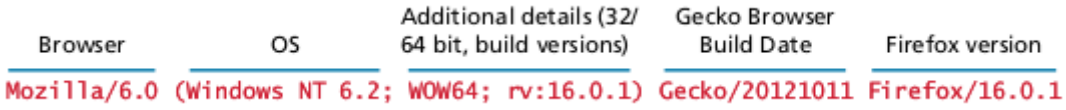
- Tek bir web sayfası düzinelerce dosyaya başvurabilir ve birçok HTTP requesti ve response'u gerektirebilir.
- Tek bir web sayfasının, muhtemelen farklı domainlerden, birden fazla kaynak gerektirmesi, birlikte çalışmamız ve farkında olmamız gereken gerçektir.
- Modern browserlar, geliştiriciye belirli bir sayfanın HTTP trafiğini anlamlandırmasına yardımcı olabilecek toollar sağlar.
- Aşağıdaki şekilde, geçerli bir sayfa için istenen kaynakları ve her bileşen için yükleme sürelerinin dökümünü listeleyen Firefox eklentisi FireBug'dan (HTML / JavaScript hata ayıklayıcısı) bir ekranı göstermektedir.



1.1 Headers (Başlıklar):

- Headerlar client tarafından request içinde gönderilir ve serverdan response olarak alınır.
- Bunlar, HTTP işleminin (HTTP transaction) parametrelerini kodlar(encode eder), yani serverın ne tür bir yanıt göndereceğini tanımlar).
- Headerlar, HTTP'nin en güçlü yönlerinden biridir.
- Düzinelerce başlık olmasına rağmen, her bir taleple hangi tür bilgilerin gönderildiğini anlamanız için gerekli olanlardan birkaçını ele alacağız.
- *İstek başlıkları (Request Headers)*, client makinenizle ilgili verileri içerir (kişisel bilgisayarınızda olduğu gibi).
- Web geliştiricileri bu bilgileri analitik nedenlerle ve site özelleştirmesi (site customization) için kullanabilirler. Bunlardan bazıları:
- ***Host:*** Host Headerı, HTTP 1.1'de tanıtıldı ve birden çok web sitesinin aynı IP adresinde barındırılmasını sağlar. Farklı domainlere yönelik requestler aynı IP'ye ulaşabileceğinden, host headerı, servera bu IP adresindeki hangi domainle ilgilendiğimizi söyler.
- ***User-Agent:*** User-Agent stringi, modern web geliştirmede en çok başvurulan başlıklardan biridir. Bize, kullanıcının ne tür bir işletim sistemi ve tarayıcı çalıştırdığını söyler. Aşağıdaki şekilde, bir örnek stringi ve içinde kodlanmış bileşenleri gösterir. Bu dizeler, farklı stil sayfaları (style sheet) arasında geçiş

yapmak ve sitenin ziyaretçileri hakkındaki istatistiksel verileri kaydetmek için kullanılabilir.



User-Agent Componenti

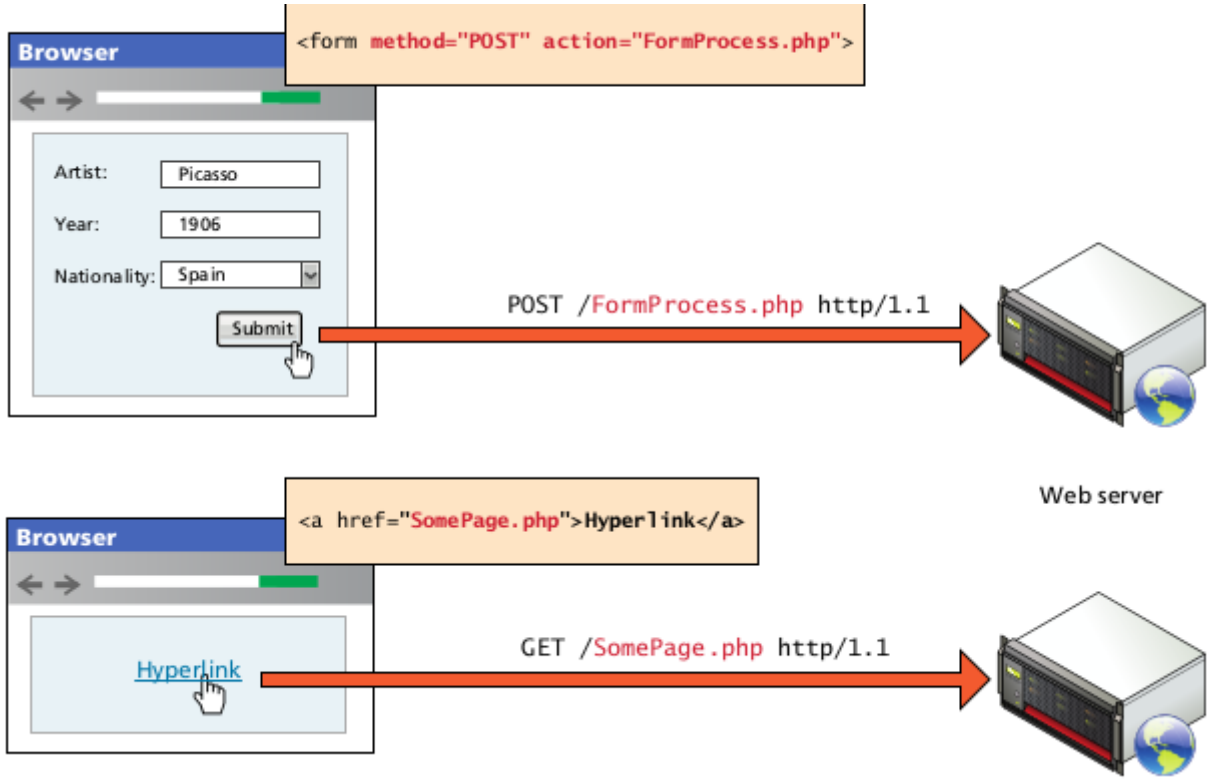
- ***Accept***: Accept (kabul) headerı, servera clientın yanıtta ne tür ortam(media) türleri alabileceğini bildirir. Server bu kısıtlamalara uymalı ve client tarafından kabul edilemez veri türlerini iletmemelidir.
- **Accept-Encoding:** Accept-Encoding headerları, iletimden önce verilerde ne tür değişiklikler (modifications) yapılabileceğini belirtir. Burada, bir browser belirli algoritmalarla sıkıştırılmış dosyaları açabileceğini veya "*indirebileceğini(deflate)*" belirtebilir. *Sıkıştırılmış iletim bant genişliği kullanımını azaltır,* ancak yalnızca client içeriği deflate edip, içeriği görebiliyorsa yararlıdır.
- ***Connection:*** Bu header, serverın bağlantıyı açık tutması veya yanıttan sonra kapatması (close it after response) gerektiğini belirtir. Server isteğe uysa da, response connection headerı, client açık kalmasını istese bile bir oturumu sonlandırabilir.(terminate a session)
- ***Cache-Control:*** Önbellek başlığı(Cache Header), clientın önbellek mekanizmalarını (caching mechanisms) denetlemesine olanak tanır. Bu header, örneğin, verileri yalnızca belirli bir yaştan daha büyük değilse indirmeyi, önbelleğe alındığında (cached) asla yeniden indirmemeyi veya her zaman yeniden indirmeyi belirtebilir. Cache-Control başlığının doğru kullanımı bant genişliğini büyük ölçüde azaltabilir.
- *Yanıt üstbilgileri (Response Headers)* isteği yanıtlayan sunucu hakkında ve gönderilen veriler hakkında bilgi içerir. Bunlardan bazıları:
- ***Server:*** Server headerı, clienta server hakkında bilgi verir. Serverın hangi tür işletim sistemini çalıştırdığını ve kullandığı web sunucusu yazılımını (web server software) içerebilir.
- **NOT:**

- Server Headerı, hackerlara altyapınız (infrastructure) hakkında ek bilgi sağlayabilir.
- Örneğin, bir eklentinin(plugin) güvenlik açığından etkilenen bir sürümünü(vulnerable version) çalıştırıyorsanız ve Server Headerınız, yalnızca headera dayalı olarak taranabileceğiniz ve daha sonra saldırıya uğrayabileceğiniz herhangi bir clienta bu bilgileri bildirebilir.
- Bu nedenle, birçok yönetici/administrators bu alanı olabildiğince az bilgi ile sınırlar.
- ***Last-Modified:*** Son Değiştirme (Last-Modified), istenen kaynağın en son ne zaman değiştiği hakkında bilgi içerir. Değişmeyen statik bir dosya her zaman dosyayla ilişkilendirilen aynı son değiştirilmiş zaman damgasını (last modified timestamp) iletir. Bu, önbellek mekanizmalarının (Cache-Control Request header, gibi), dosyanın yeni bir kopyasını indirmeye veya yerel olarak önbelleğe alınmış bir kopyasını kullanmaya karar vermesine olanak tanır.
- ***Content-Length:*** İçerik Uzunluğu, response body'sinin (iletinin/mesajın) ne kadar büyük olacağını belirtir. İstekte bulunan browser daha sonra verileri almak için uygun miktarda bellek ayırabilir. Last-Modified headerının her requesti değiştirdiği dinamik web sitelerinde, bu alan önbelleğe alınmış bir kopyanın "*tazeliğini(freshness)*" belirlemek için de kullanılabilir.
- ***Content-Type:*** Accept Request Headerına eşlik etmek için Content-Type Response Headerı, browsera iletinin/mesajın gövdesine ne tür veri eklendiğini bildirir. Bazı ortam türü değerleri (media-type values) text/html, image/jpeg, image/png, application/xml ve diğerleridir. Gövde verileri ikili (binary) olabileceğinden, ne tür bir dosyanın eklendiğini belirtmek önemlidir.
- ***Content-Encoding:*** Client dosyaları açabiliyor olsa (a client may be able to gzip decompress files) ve Accept-Encoding headerında belirtilmiş olsa da, server dosyayı kodlamayı (encode) seçebilir veya seçmeyebilir. Her durumda, server, clienta içeriğin nasıl kodlandığını belirtmelidir, böylece gerekirse decompress edilebilir.
- **NOT:**
- İletimden önce sayfaları sıkıştırmak bant genişliğini azaltsa da, bunun için *CPU döngüleri (CPU cycles)* ve *bellek (memory)* gerekir. Meşgul serverlarda, dinamik

içeriği sıkıştırılmadan iletmek bazen verimli olabilir ve bu CPU döngülerini isteklere yanıt vermek için tasarruf ettirebilir.

1.2 Request Methods (istek/talep Metodları):

- HTTP protokolü, her biri farklı bir amaca ve özelliğe sahip birkaç farklı istek türü (types of requests) tanımlar.
- En yaygın istekler, *HEAD requesti* ile birlikte *GET ve POST requestidir*.
- *PUT, DELETE, CONNECT, TRACE* ve *OPTIONS* gibi diğer requestler nadiren kullanılır.
- HTTP isteğinin en yaygın türü GET requestidir. Bu istekte, belirli bir URL'de bulunan bir kaynağın alınması istenir. Bir bağlantıya tıkladığınızda, browserınıza bir URL yazdığınızda veya bir bookmarka tıkladığınızda, genellikle bir GET requesti yaparsınız.
- Veriler ayrıca bir GET isteği yoluyla da iletilebilir.
- Diğer yaygın request yöntemi, *POST requestidir*.
- Bu yöntem normalde verileri bir HTML formu (HTML Form) kullanarak servera iletmek için kullanılır (ancak, bir veri giriş formu (data entry form) bunun yerine GET yöntemini kullanabilir).
- Bir POST requestingde, veriler requestin headerı üzerinden iletilir ve bu nedenle GET gibi uzunluk sınırlamalarına tabi değildir.
- Ayrıca, veriler URL'ye iletilmediğinden, veri iletmenin daha güvenli bir yolu olduğu görülmektedir. (pratikte tüm post verileri şifrelenmemiş (unencrypted) olarak iletilir ve neredeyse GET verileri kadar kolay okunabilir).
- Aşağıdaki şekil, bir GET ve bir POST requestini gösterir.
- Bir HEAD requesti, bir GET requesitine benzer, ancak response, tam bir GET requestinde alınacak gövdeyi değil, yalnızca header bilgilerini içerir.
- Örneğin arama motorları, bir sayfanın, kaynağın gövdesi için gereksiz requestlerde bulunmadan yeniden dizine eklenip eklenmeyeceğini (reindexed) belirlemek ve bant genişliğinden tasarruf etmek için bu requesti kullanır.



GET Request vs Post Request

1.3 Response Codes (Yanıt Kodları):

- Response kodları, Response headerının bir parçası olarak server tarafından döndürülen tam sayı değerleridir.
- Bu kodlar, requestin, başarılı olup olmadığını (successful), hataların olup olmadığını (had errors), izin gerektirip gerektirmediği (requires permission) ve daha fazlası dahil olmak üzere durumunu tanımlar.
- Aşağıdaki tabloda en yaygın response kodları listelenmiştir.
- Kodlar, response kategorisini belirtmek için ilk basamağı kullanır.
- 2## kodları başarılı responselar içindir (for successful responses), 3## yönlendirmeyle ilgili responselar içindir (for redirection-related responses), 4## kodları client hatalarıyken (codes are client errors), 5## kodları server hatalarıdır. (codes are server errors.)

Code	Description
200: OK	200 yanıt kodu, isteğin başarılı olduğu anlamına gelir.
301: Moved Pemanently Kalıcı olarak taşındı	İstemciye istenen kaynağın kalıcı olarak taşındığını bildirir. Bunun gibi kodlar, arama motorlarının, kaynağının yeni konumunu yansıtacak şekilde veritabanlarını güncellemelerine olanak tanır. Normalde bu kaynak için yeni konum yanıtta döndürülür.

304: Not Modified Değiştirilmemiş	Client, uygun Cache-Control headerlarına sahip bir kaynak isterse, yanıt sunucudaki kaynağın istemci önbelleğindekinden daha yeni olmadığını söyleyebilir. İstemcinin, kaynağın önbelleğe alınmış bir kopyasını kullanmasını beklediğimizden böyle bir yanıt yalnızca bir başlıktır.
307: Temporary Redirected Geçici Yönlendirme	Bu kod 301'e benzer, ancak yeniden yönlendirme geçici olarak düşünülmelidir.
400: Bad Request Geçersiz İstek	Başlıklar veya HTTP isteği ile ilgili bir şey HTTP protokolüne doğru şekilde uymuyorsa, 400 yanıt kodu istemciyi bilgilendirir.
401: Unauthorized Yetkisiz	Bazı web kaynakları protecteddır ve kullanıcının kaynağa erişmek için kimlik bilgileri (credentials) sağlamasını gerektirir. İstemci 401 kodu alırsa, isteğin yeniden gönderilmesi gerekir ve kullanıcının bu kimlik bilgilerini sağlaması gerekir.
404: Not Found Bulunamadı	404 kodları web kullanıcıları tarafından bilinen tek koddur. Birçok tarayıcı, istenen kaynak bulunamadığında 404 kodlu bir HTML sayfası görüntüler.
414: Request URI too long İstek URI'ı çok uzun	URL'lerin, yürürlükteki sunucu yazılımına bağlı olarak değişen bir uzunluk sınırlaması vardır. 414 yanıt kodu büyük olasılıkla çok fazla verinin URL yoluyla gönderilmeye çalışıldığı anlamına gelir.
500: Internal Server Error Dahili Sunucu Hatası	Bu hata, sunucunun bir hatayla karşılaştığını söylemek dışında istemciye neredeyse hiçbir bilgi sağlamaz.

HTTP Response Kodları

Ayrıca burada yazıyı daha fazla uzatmamak adına bahsetmemiş olsamda(Yazının diğer partlarında ufak notlar halinde bahsedeceğim) aşağıdaki kavramlar hakkında araştırma yapabilirsiniz.

1. **Firewall**

- Aşağıdaki linklerden Firewall kavramı hakkında bilgi edinebilirsiniz.
- medium.com/@gokhansengun/firewall-nedir-çeşitleri-nelerdir-ve-nerelerde-kullanılır-55c56eed4ffa
- lifewire.com/what-is-a-firewall-how-does-it-work-4692441
- cisco.com/c/en/us/products/security/firewalls/what-is-a-firewall.html#~types-of-firewalls

2. **DHCP (Dynamic Host Configuration Protocol)**

- medium.com/@gokhansengun/dhcp-nedir-ve-nasıl-çalışır-ad7bed1ef468
- lifewire.com/what-is-dhcp-2625848
- httpeducba.com/what-is-dhcp/

3. **Load Balancer**

- https://medium.com/@gokhansengun/load-balancer-nedir-ve-ne-işe-yarar-32d608f98ef9

4. **Encapsulation**

- http://bilgisayarkavramlari.sadievrenseker.com/2007/12/17/kapsulleme-encapsulation/

> Ayrıca eksik gördüğünüz veya eklemek istediğiniz bir husus olduğunda iletişime geçmeniz beni mutlu edecektir.

. . .

Kaynakça:

- Uygulamalarla Siber Güvenliğe Giriş, 2.Basım, M. Alparslan Yıldız.
- Fundamentals of Web Development, 1st Edition, Randy Connolly, Ricardo Hoar
- Computer Networks, 5th Edition, Tanenbaum
- Computer Network & Internet, 6th Edition, Douglas E. Comer
- Cyrptography & Network Security, 4th Edition, William Stallings
- CCNA Routing and Switching Study Guide, Todd Lammle
- Hacking: The Art of Exploitation, 2nd Edition, Jon Erickson
- gokhansengun.com/ adresindeki yazılar.
- mehmetince.net/ adresindeki yazılar
- twitch.tv/mdisec kanalı
- nmap.org/nsedoc/lib/smb.html#script-args
- samba.org/cifs/docs/what-is-smb.html#What_Is_SMB

- blog.varonis.com/cifs-vs-smb/
- webisyo.com/bilgisayar-aglari-hakkinda-genis-bilgi-lan-man-wan/
- muratkara.com/network/Network_DersNotu.pdf

Computer Network Osi Model Tcp Ip DNS Http

About Help Legal

Get the Medium app