

## 2.2 The Web and HTTP

### 2.2.1 Overview of HTTP (HTTP'ye genel bakış )

### 2.2.2 Non-Persistent and Persistent Connections( Kalıcı ve kalıcı olmayan HTTP bağlantısı)

### 2.2.3 HTTP Message Format

### 2.2.4 User-Server Interaction: Cookies

### 2.2.5 Web Caching

!!! (işin hikaye kısmı) !!!

1990'ların başında, sahneye yeni bir uygulama geldi - World Wide Web.

Kullanıcılar için en cazip olanı, Web'in isteğe bağlı olarak çalışmasıdır. Kullanıcılar istediklerini, istedikleri zaman alırlar. Bu, kullanıcıları yayınlamaya zorlayan geleneksel yayın radyo ve televizyonundan farklıdır.

Web ve protokolleri, YouTube, Web tabanlı e-posta (Gmail gibi) ve Instagram ve Google Haritalar da dahil olmak üzere çoğu mobil İnternet uygulaması için bir platform görevi görür.

Formlar, JavaScript, Java uygulamaları ve diğer birçok cihaz sayfalar ve sitelerle etkileşimde bulunmamızı sağlar.

Fotoğraflar ve videolar duyularımızı harekete geçirir.

Hyperlinks ve arama motorları bir bilgi okyanusunda gezinmemize yardımcı olur.Web sayfası nesnelerden oluşur

### 2.2.1 Overview of HTTP ( Http'ye genel bakış )

Web'in uygulama katmanı protokolü olan Hyper Text Transfer Protocol (HTTP) Web'in kalbindedir. \*[RFC 1945] ve \*[RFC 2616] 'da tanımlanmıştır.

### \*RFC Nedir?

TCP/IP standartları, Request For Comment(RFC) denilen dökümanlar içinde yayınlanmıştır. RFC'lerde internetin işleyişi açıklanır. TCP/IP standartları RFC'lerde açıklanır fakat tüm RFC'ler standartları tanımlamaz.

HTTP iki programda tanımlanır.Birisi client programı diğeri de server programı,bunlar HTTP mesajlarını alıp vererek birbirleriyle konuşurlar.

HTTP bu mesajların yapısını ve client ile server'ın mesajları nasıl değiştirdiğini tanımlar.

HTTP den önce bazı Web terminolojilerini gözden geçirelim.

Web page (document de denir) nesnelerden(object) oluşur.

Nesne tek bir URL ile adreslenebilen HTML dosyası, JPEG,Java uygulaması veya video gibi bir dosyadır. Çoğu Web sayfası temel olarak bir HTML dosyasından ve birkaç nesneden oluşur.

Örneğin bir Web sayfasında HTML metni ve 5 JPEG görüntüsü varsa Web sayfasında 6 nesne bulunur.Temel HTML dosyası diğer nesnelerin URL'lerine reference eder. Her URL nin 2 bileşeni vardır.

-nesneyi barındıran server'ın host adı ve nesnenin yol adı(path name)

www.someschool.edu/someDept/pic.gif

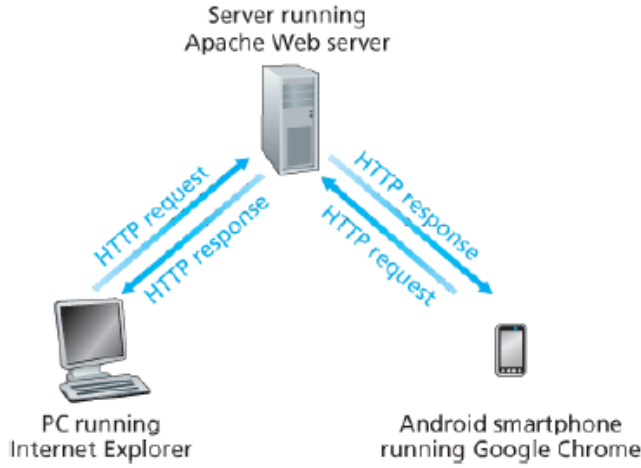
host name

path name

Web tarayıcıları HTTP nin client tarafına uygulandığından Web bağlamında tarayıcı ve client kelimelerini birbirinin yerine kullanabiliriz.

HTTP nin server tarafını uygulayan web sunucuları her biri bir URL ile adreslenebilen Web nesneleri barındırır.Popüler web serverları arasında Apache ve Microsoft Internet Information Server bulunmaktadır.

HTTP, Web clientlerinin Web serverlarından Web sayfalarını nasıl istediğini ve serverların Web sayfalarını client'e nasıl aktardığını tanımlar. Genel fikir şekil 2.6 da gösterilmiştir.



Bir kullanıcı bir Web sayfası istediğinde (örneğin, bir \*hyperlink'i tıkladığında) tarayıcı, sayfadaki nesneler için HTTP istek mesajlarını server'a gönderir.

**\*Hyperlink:** web tarayıcınızda 'başka bir sayfaya geçiş yapmak' için kullanılan programlı komutlardır. Her web sayfası, sizi ilgili bazı web sayfalarına veya resim ya da dosyalara gönderen onlarca hyperlink ile doludur. Mouse işaretçisi işaret parmağına dönüştüğünde bir şeyin hyperlink olduğunu anlayabilirsiniz.

HTTP, TCP yi temel transport protokolü olarak kullanır.(UDP yerine)

HTTP client önce serverla bir TCP bağlantısı başlatır.

Bağlantı kurulduktan sonra tarayıcı ve server işlemleri TCP'ye soket arabirimleri üzerinden erişir.

Bölüm 2.1'de açıklandığı gibi, client tarafında soket arayüzü, client process ile TCP bağlantısı arasındaki kapıdır; server tarafında server process ile TCP bağlantısı arasındaki kapıdır. Client soket arayüzüne HTTP istek(request) mesajları gönderir ve soket arayüzünden HTTP yanıt(response) mesajları alır.

Benzer şekilde, HTTP server soket arabiriminden request mesajları alır ve soket arabirimine response mesajları gönderir. Client soket arayüzüne bir mesaj gönderdiğinde, mesaj müşterinin elinden çıkar ve TCP'nin “elindedir”.

Bölüm 2.1'den TCP'nin HTTP'ye güvenilir bir veri aktarım hizmeti verdiğini hatırlayın.

Bu, bir client process tarafından gönderilen her HTTP request mesajının sonunda server’a sağlam geldiğini gösterir.

Benzer şekilde, server process tarafından gönderilen her HTTP response mesajı eninde sonunda client’e bozulmadan ulaşır.

Burada katmanlı bir mimarinin en büyük avantajlarından birini görüyoruz: HTTP'nin kayıp veriler(lose data) veya TCP'nin ağdaki verilerin kaybından veya yeniden sıralanmasından(reordering) nasıl kurtulduğuna dair endişeleri yoktur.

Bu TCP'nin işi ve protokol yığınının(protocol stack) alt katmanlarındaki protokollerdir.

Server’ın, clientle ilgili herhangi bir durum bilgisi(state inf.) saklamadan istenen dosyaları clientlere gönderdiğini unutmamak önemlidir.

Belirli bir client birkaç saniye içinde aynı nesneyi iki kez isterse, server nesneyi cliente sunduğunu söyleyerek yanıt vermez; bunun yerine, server daha önce yaptıklarını tamamen unuttuğu için nesneyi yeniden gönderir.

Bir HTTP server clientler hakkında hiçbir bilgi tutmadığından, HTTP'nin durum bilgisi olmayan \*(Stateless(unutkan)) bir protokol olduğu söylenir.

**\*Stateless Protocol:** Durum Bilgisi Olmayan Protokoller, clientim server’a request gönderdiği ve mevcut duruma göre server’ın respons’u geri gönderdiği ağ protokolleri türüdür. Serverın, oturum(session) bilgilerini veya iletişim kuran her ortak hakkındaki durumu birden fazla request için tutması gerekmez.

Ayrıca Web'in Bölüm 2.1'de açıklandığı gibi client-server uygulama mimarisini kullandığını da belirtiyoruz.

Bir Web server her zaman açıktır ve sabit(fixed) bir IP adresidir ve potansiyel olarak milyonlarca farklı tarayıcıdan gelen requestlere hizmet eder.

### 2.2.2 Non-Persistent and Persistent Connections( Kalıcı ve kalıcı olmayan HTTP bağlantısı)