

# Web ve HTTP

## Web sayfası nesnelere içerir

- ❑ Nesne HTML dosyası, JPEG resim, Java applet, ses dosyası,... olabilir
- ❑ Web sayfası birçok nesneye referansı olan **temel bir HTML** (Hyper Text Markup Language) -**dosyası** içerir
- ❑ Bütün nesnelere bir **URL** (Uniform Resource Locator) ile adreslenebilir
- ❑ Örnek URL:

`www.someschool.edu/someDept/pic.gif`

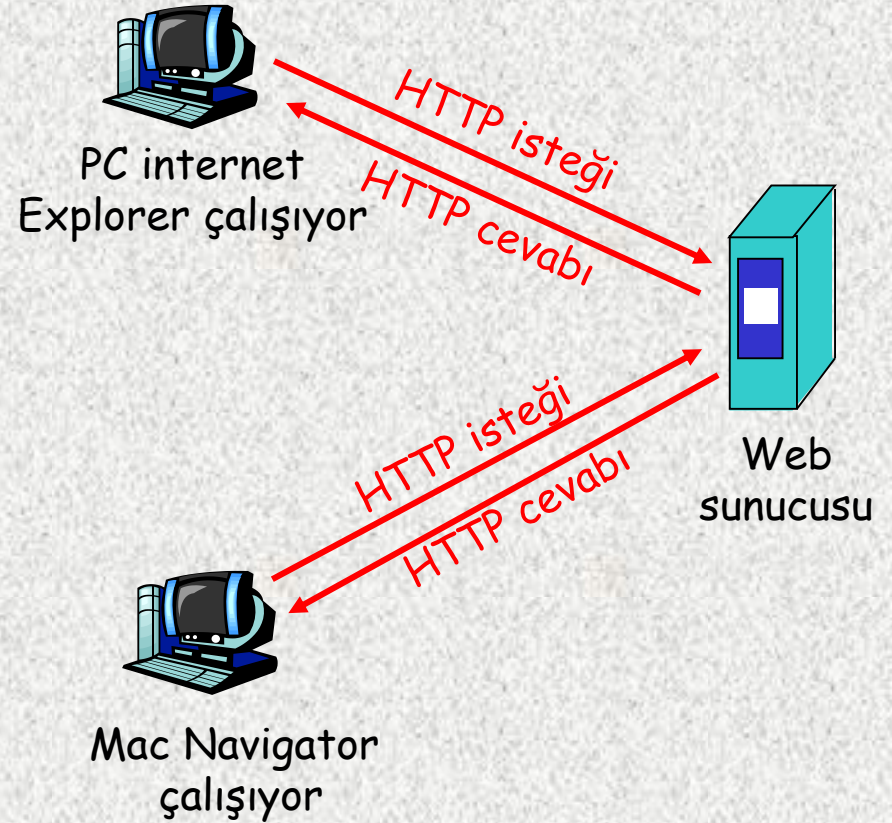
host name

path name

# HTTP

## HTTP: hypertext transfer protocol

- ❑ Web'in uygulama katmanı protokolü
- ❑ istemci/sunucu modeli
  - *istemci*: Web nesnelerini isteyen ve gösteren gezgin
  - *sunucu*: isteklere cevap veren ve nesnelere gönderen web sunucusu
- ❑ HTTP 1.0: [RFC 1945](#)
- ❑ HTTP 1.1: RFC 2068



# HTTP

## TCP kullanır:

- ❑ İstemci, sunucu ile arasında TCP bağlantısı başlatır, port 80
- ❑ sunucu TCP bağlantısını kabul eder
- ❑ HTTP mesajları internet gezgini ve web sunucusu arasında gider gelir
- ❑ TCP bağlantısı kapatılır

## HTTP is "durumsuz"

- ❑ Sunucu istemcinin geçmiş istekleri hakkında bilgi depolamaz

not

### Durum bilgisi tutan protokoller karmaşıktır!

- ❑ Geçmiş durum takip edilmeli
- ❑ Eğer sunucu/istemci çökerse durum bilgileri farklılaşabilir

# HTTP bağlantıları

## Kalıcı olmayan HTTP

- ❑ En fazla bir nesne TCP bağlantısı ile gönderilir.
- ❑ HTTP/1.0 kalıcı olmayan HTTP kullanır.

## Kalıcı HTTP

- ❑ Tek bir TCP bağlantısı üzerinden birden fazla nesne gönderilebilir.
- ❑ HTTP/1.1 kalıcı HTTP kullanır



# Kalıcı olmayan HTTP

Kullanıcı aşağıdaki URL'yi ygirir

`www.someSchool.edu/someDepartment/home.index`

(metin ve  
10 Jpeg'e  
referans içeriyor)

1a. HTTP istemcisi HTTP sunucusuna bağlantı başlatır (www.someSchool.edu, port 80)

1b. Port 80 den TCP bağlantısı bekleyen www.someSchool.edu hostu bağlantıyı kabul eder ve HTTP istemcisine bildirir

2. HTTP istemcisi HTTP *istek mesajını* (URL'yi içeren) TCP bağlantı soketine gönderir. Messaj istemcinin someDepartment/home.index nesnesini istediğini belirtir

3. HTTP sunucusu istek mesajını alır, istenen nesneyi de içeren *cevap mesajını* oluşturarak kendi soketine gönderir.

zaman



# Kalıcı olmayan HTTP

4. HTTP sunucusu TCP bağlantısını kapatır.

5. HTTP istemcisi html dosyasını içeren mesajı alır ve gösterir. Bu dosyayı işlerken 10 tane referans jpeg nesnesi daha olduğunu fark eder

6. 1-5 adımları bu 10 jpeg nesnesi için tekrarlanır

zaman

# Cevaplama süresi modeli

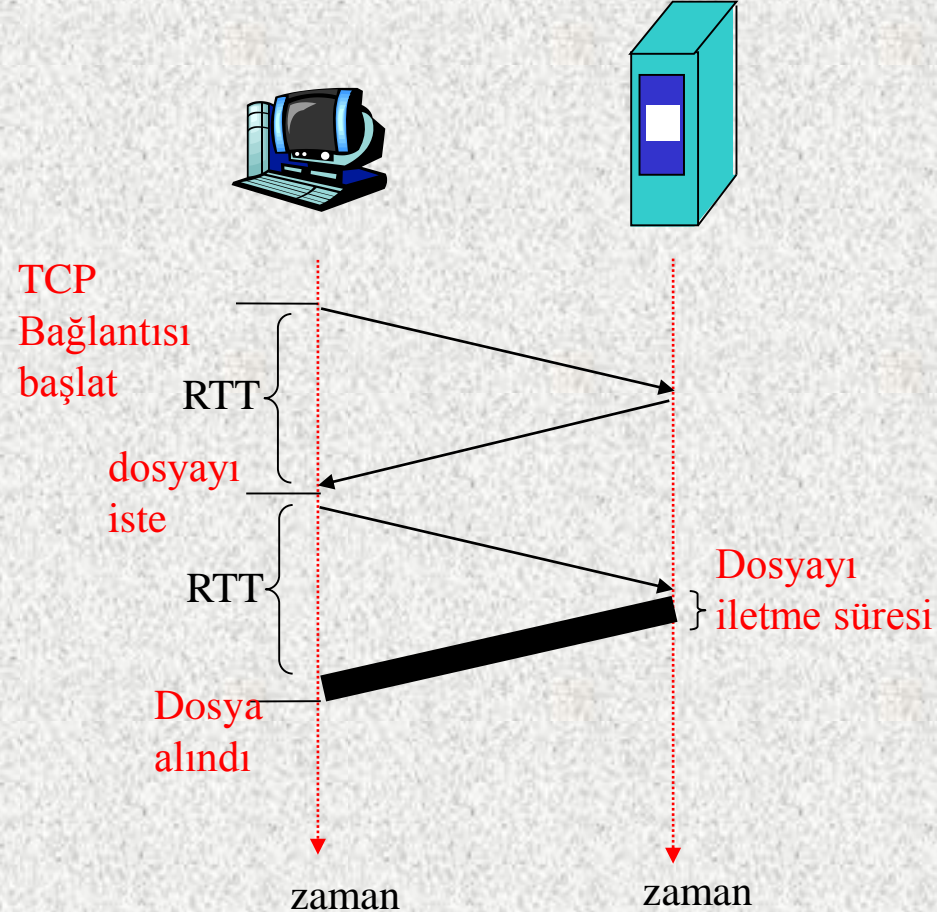
## RTT (Round Trip Time)

**tanımı:** Gönderilen küçük bir paketin sunucuya gidip gelme zamanı.

## Cevaplama zamanı:

- ❑ TCP bağlantı isteği için 1 RTT
- ❑ HTTP isteği ve HTTP cevabının ilk bitlerinin dönmesi için 1RTT
- ❑ Dosya iletim süresi

**toplam = 2RTT+iletim süresi**



# Kalıcı HTTP

## Kalıcı olmayan HTTP sorunları:

- ❑ Her nesne için 2 RTT gerekli
- ❑ İşletim sistemi her nesnede TCP bağlantısını oluşturmak için meşgul olacak
- ❑ Fakat internet gezginleri genelde referans nesnelere için paralel TCP bağlantısı açarlar

## Kalıcı HTTP

- ❑ Sunucu nesneyi gönderdikten sonra bağlantıyı açık tutar
- ❑ sonraki HTTP mesajları aynı bağlantı üzerinden gönderilir

## Ardışık olmayan Kalıcı:

- ❑ İstemci eğer bir önceki isteğe cevap alınmışsa yeni bir istek gönderir
- ❑ Herbir referans nesne için 1 RTT

## Ardışık Kalıcı:

- ❑ HTTP/1.1
- ❑ İstemci gördüğü bütün referans nesnelere için istekte bulunur
- ❑ Bütün referans nesnelere için 1 RTT kadar az bir süre



# HTTP istek mesajı

- ❑ İki tip HTTP mesajı: *istek, cevap*
- ❑ **HTTP istek mesajı:**
  - ASCII (insan tarafından okunabilen biçim)

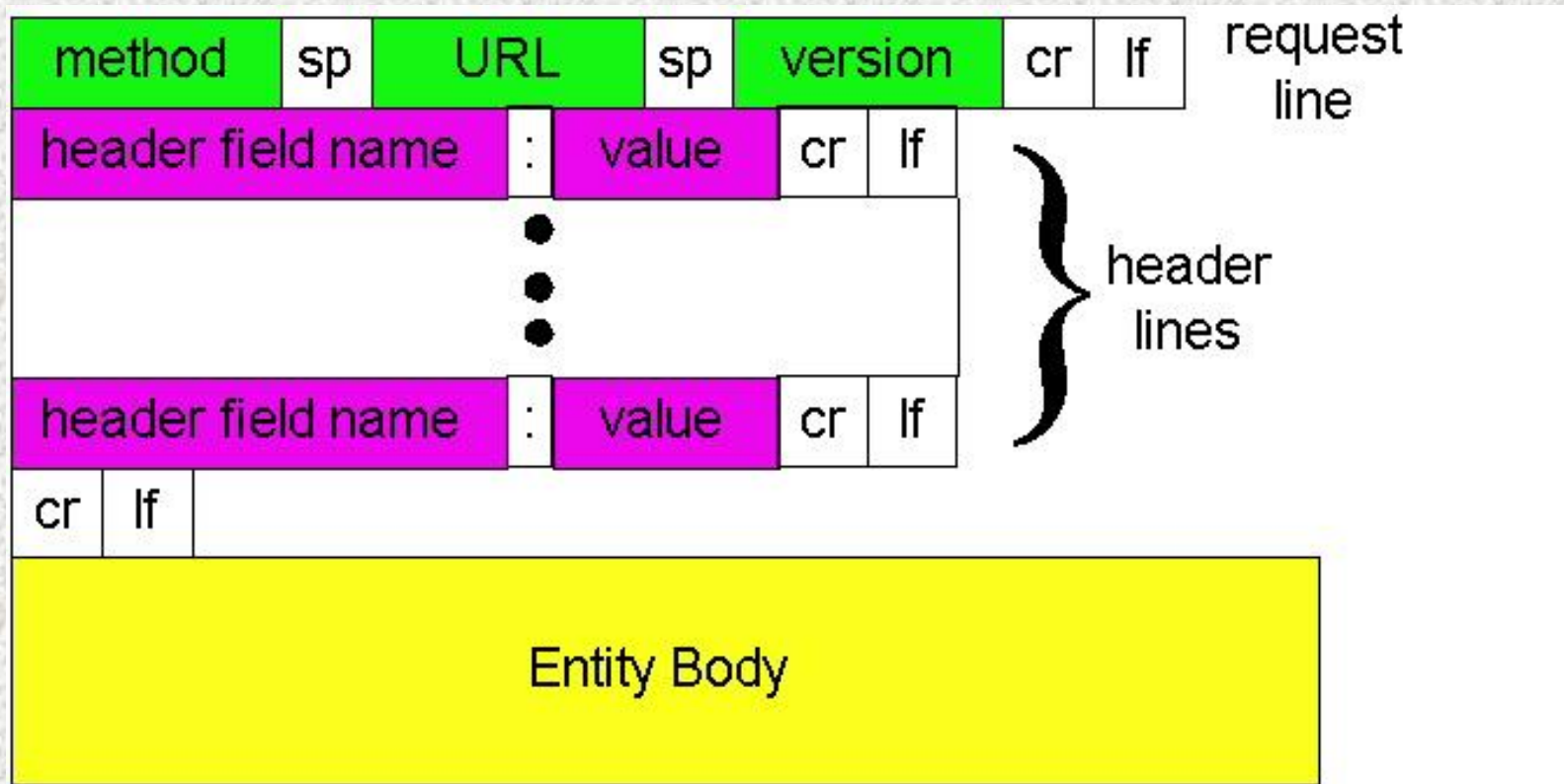
İstek satırı  
(GET, POST,  
HEAD komutları)

Başlık satırları

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: fr
```

Enter satırı,  
Mesajın sonun belirtir (enter)

# HTTP istek mesaji: genel format



# form girişi yükleme

## Post metodu:

- ❑ Web page genelde form girişi içerir
- ❑ Input sunucuya varlık gövdesi kısmında aktarılır

## URL metodu:

- ❑ GET metodunu kullanır
- ❑ Input istek satırının URL alanında aktarılır:

`www.somesite.com/animalsearch?monkeys&banana`

# Method tipleri

## HTTP/1.0

- ❑ GET
- ❑ POST
- ❑ HEAD
  - Sunucudan nesneyi göndermemesini ister. Genelde hata ayıklama için kullanılır

## HTTP/1.1

- ❑ GET, POST, HEAD
- ❑ PUT
  - Dosyayı URL path'ine varlık gövdesi kısmında aktarır
- ❑ DELETE
  - Belirtilen URL'deki dosyayı siler



# HTTP cevap mesajı

Durum satırı  
(protokol  
durum kodu)

HTTP/1.1 200 OK

başlık  
satırları

Connection close

Date: Thu, 06 Aug 1998 12:00:15 GMT

Server: Apache/1.3.0 (Unix)

Last-Modified: Mon, 22 Jun 1998 .....

Content-Length: 6821

Content-Type: text/html

veri, e.g.,  
istenilen  
HTML dosyası

data data data data data ...

# HTTP cevabı durum kodları

sunucu->istemci cevap mesajı ilk satırı.

Birkaç örnek kod:

## **200 OK**

- İstek başarılı, istenen nesne bu mesajda

## **301 Moved Permanently**

- İstene nesne taşındı, yeni yeri bu mesajda

## **400 Bad Request**

- İstek mesajı sunucu tarafından anlaşılamadı

## **404 Not Found**

- İstene doküman sunucuda bulunamadı

## **505 HTTP Version Not Supported**