

Bilgisayar Programcılığı Uzaktan Eğitim Programı

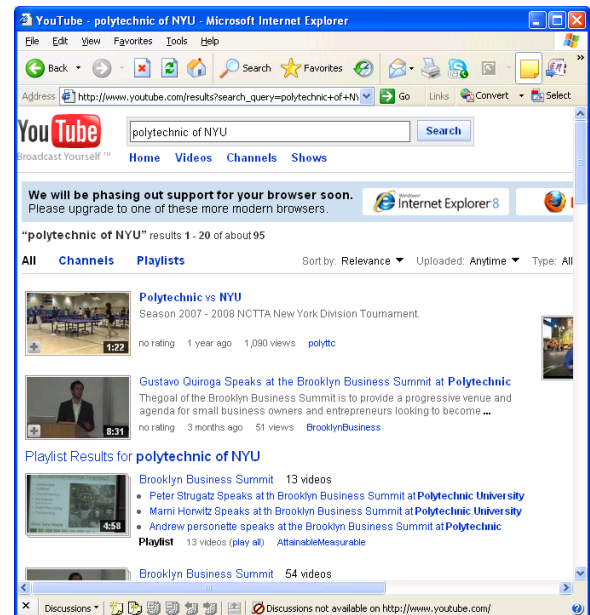
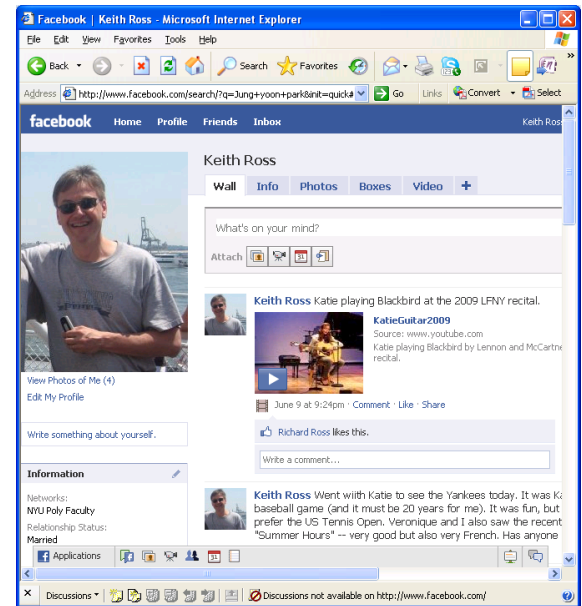
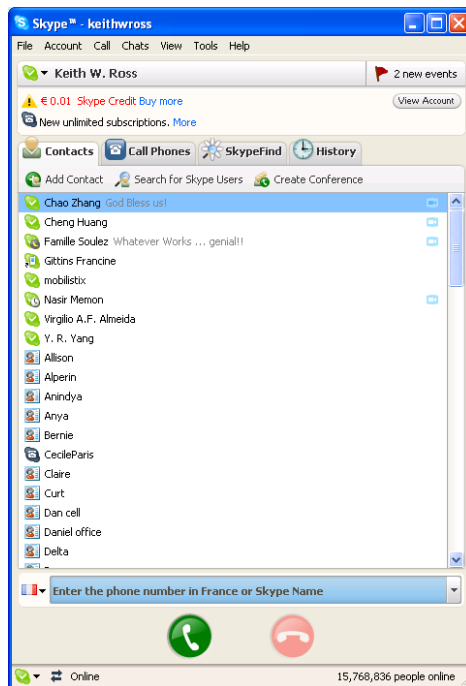
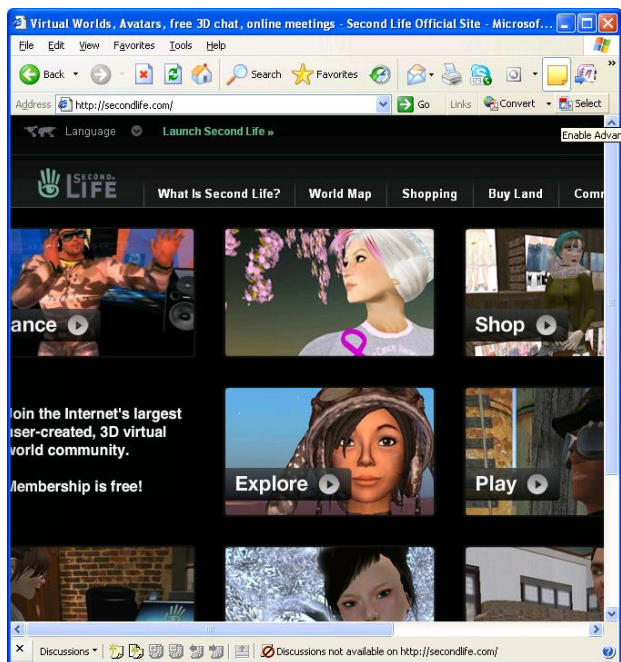
**e-BİLG 121 AĞ TEKNOLOJİLERİNİN
TEMELLERİ**

Öğr. Gör. Bekir Güler

E-mail: bguler@fatih.edu.tr

Hafta 2. Uygulama (Application) katmanı

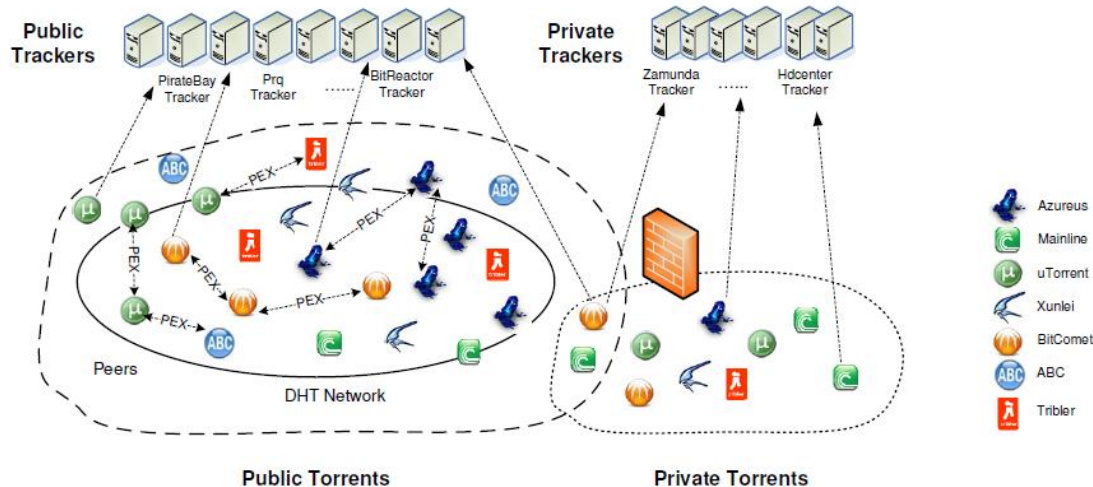
- ❑ 2.1 Ağ uygulamalarının temelleri
- ❑ 2.2 Web ve HTTP



Public
Discovery
Sites



Private
Discovery
Sites



Bazı ağ uygulamaları

- ❑ E-mail
- ❑ Web
- ❑ Anlık mesajlaşma (instant messaging)
- ❑ Uzaktan oturum açma (remote login)
- ❑ P2P dosya paylaşımı
- ❑ Çok kullanıcılı ağ oyunları
- ❑ Depolanmış video klipleri
- ❑ Sosyal ağlar
- ❑ IP üzerinden ses
- ❑ Gerçek zamanlı video konferansı

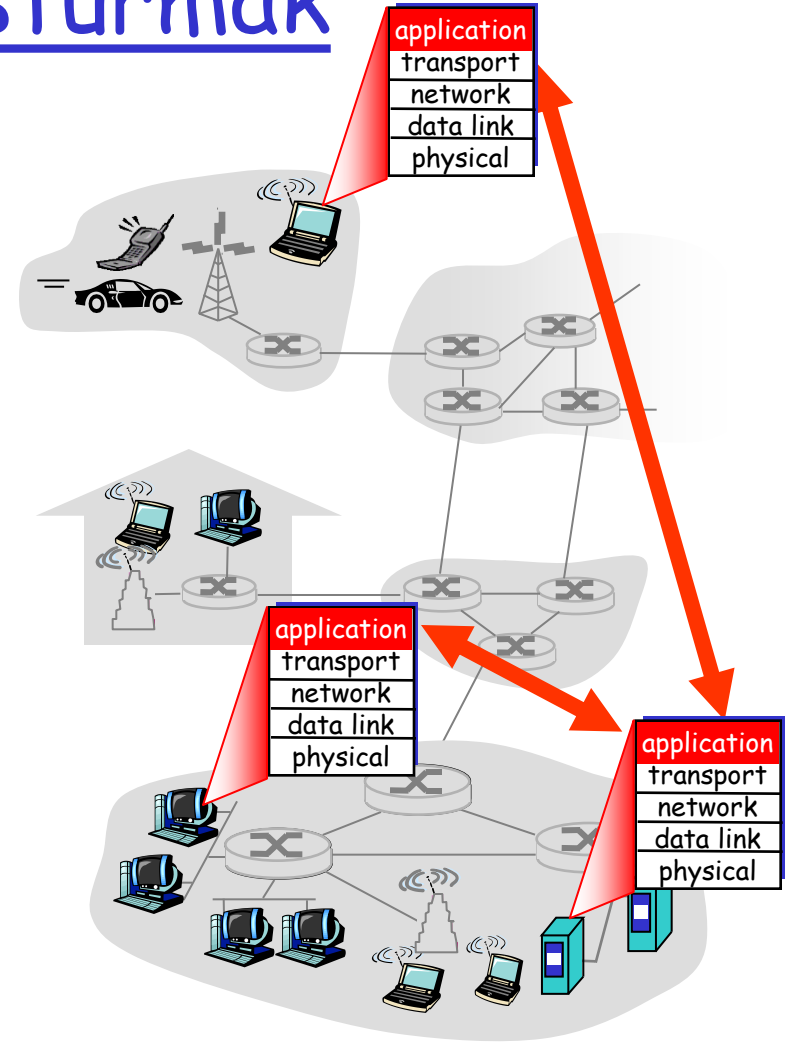
Bir ađ uygulaması oluřturmak

Program yazılır

- ❖ Farklı son kullanıcı sistemlerinde alıřtırılır
- ❖ Ađ zerinden iletiřim kurulur
- ❖ rneđin, web server yazılımı ađ zerinden web browser yazılımı ile bađlantı kurar

Temel ađ cihazları iin program yazmaya gerek yoktur

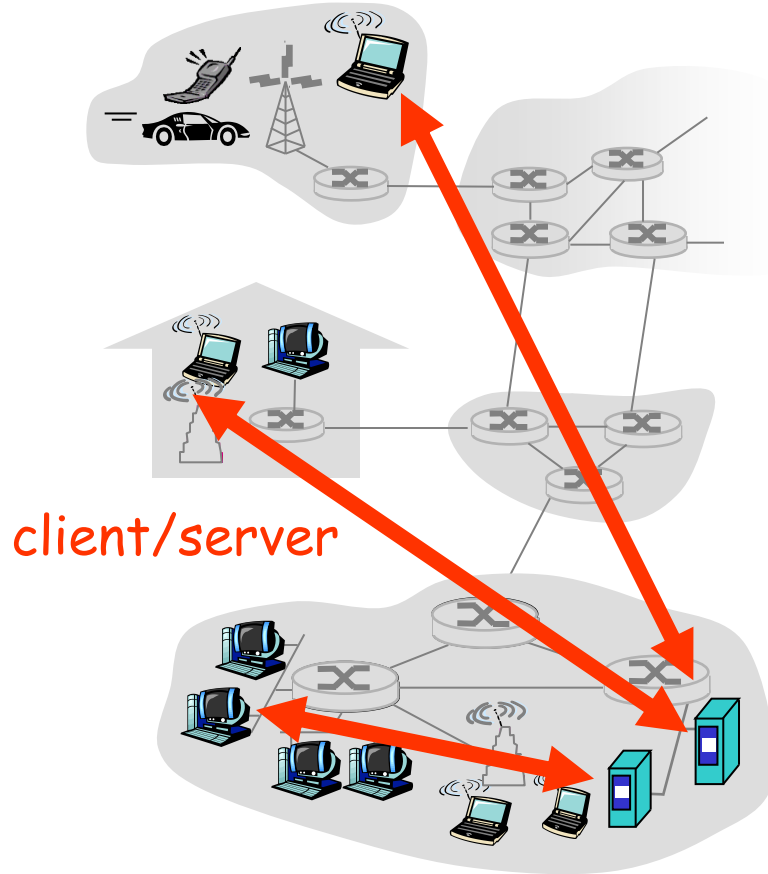
- ❖ Temel ađ cihazları kullanıcı programlarını alıřtırmaz



2.1 Uygulama mimarisi

- ❑ İstemci (Client) - Sunucu (server)
 - ❖ Veri merkezleri barındırır
- ❑ Peer (eş)-to-peer (P2P)
- ❑ İstemci-sunucu ve P2P karışım

Client-server mimarisi



server:

- ❖ Ana bilgisayar her zaman açık
- ❖ Sabit IP adresi
- ❖ Sunucu sayısı ihtiyaca göre değiştirilebilir

client:

- ❖ Sunucuyla iletişim
- ❖ Belli zamanlarda bağlanabilir
- ❖ Dinamik IP adresleri alabilir
- ❖ Birbirleri ile direk iletişim kurmaz

Google veri merkezleri

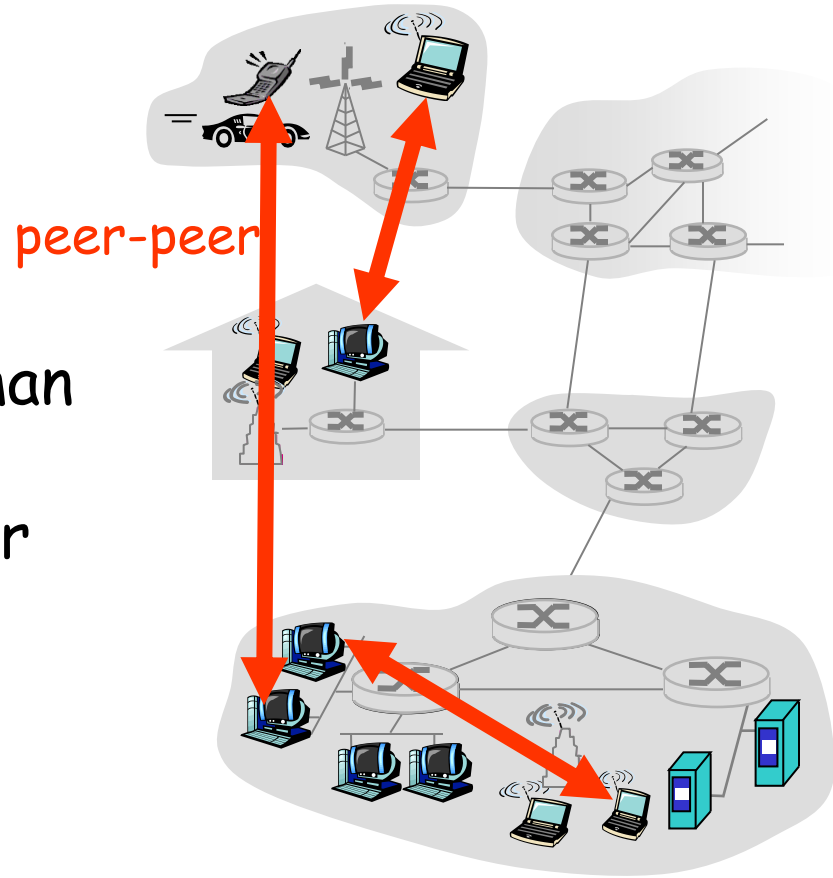
- ❑ Tahmin edilen veri merkezi maliyeti: 600 milyon \$
- ❑ Google, 2007'de yeni veri merkezlerine 2.4 milyar \$ harcadı
- ❑ Her bir veri merkezi 50-100 megawatt güç kullanıyor



P2P mimarisi

- ❑ Sunucu her zaman açık değil
- ❑ Rasgele uç sistemlerle doğrudan iletişim
- ❑ Bilgisayarlar zaman zaman bağlanabilir ve IP adreslerini değiştirebilir

Ölçeklenebilir, fakat yönetmesi zor



Client-server ve P2P karışımı

Skype

- ❖ IP üzerinden P2P uygulaması
- ❖ Merkezi sunucu: uzaktaki bilgisayarın adresini tutuyor
- ❖ Client-client bağlantı: sunucu olmadan direk bağlantı

Anlık mesajlaşma (Instant messaging)

- ❖ İki kullanıcı arasında P2P kullanarak mesajlaşma
- ❖ Merkezi servis: istemcinin varlığını ve yerini algılar
 - Kullanıcı, IP adresini merkezi sunucuya kayıt eder
 - Kullanıcı arkadaşlarının IP adresini bulmak için merkezi sunucuyla bağlantı kurar

İletişim işlemleri

İşlem (Process): bir bilgisayarda çalışan programlardır

- ❑ Aynı bilgisayarda, işlemler kendi arasında haberleşebilir (OS tarafından gerçekleştirilir).
- ❑ Farklı bilgisayarlardaki işlemler mesajların değişimi ile haberleşirler

Client işlemi: iletişimi başlatan işlem

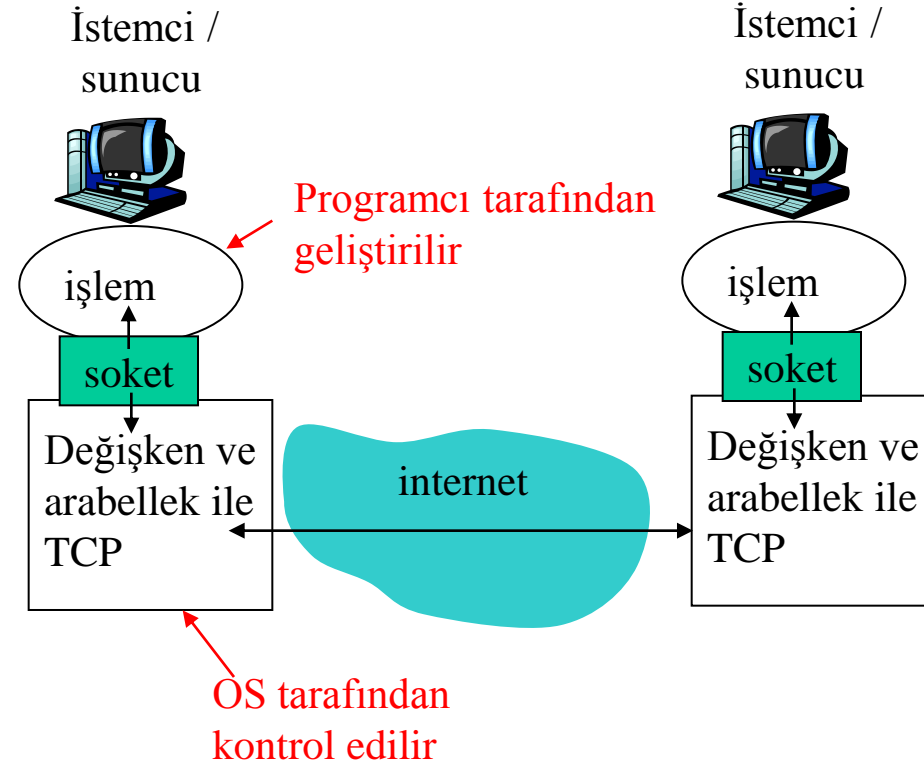
Server işlemi: bağlantıyı bekleyen işlem

Not: P2P mimarisinde client ve server işlemleri vardır

Soket (socket)

IP adresi ve port numarası soketi oluşturur

- ❑ İşlem, mesajları soketinden gönderir/alır
- ❑ Soket üzerinden iletim
 - ❖ Gönderen işlem, mesajı kapının dışına koyar
 - ❖ Taşıma katmanı, mesajı alıcı işlem soketine getirir



İşlemleri adreslemek

- ❑ Mesajları almak için işlemin bir tanımlayıcısının olması gerekir
- ❑ Ağda bilgisayarların ve aygıtların 32-bit benzersiz IP adresi vardır
- ❑ Bilgisayarda **işlemleri adreslemek** için **tanımlayıcı**, hem **IP adresi** hem de **port numarası** kullanılır.
- ❑ Örnek port numaraları:
 - ❖ HTTP server: 80
 - ❖ Mail server: 25
- ❑ **Uygulama:** komut satırından `ipconfig` komutunu kullanarak IP adresinizi öğreniniz (Windows)

Taşıma hizmeti, uygulamanın ihtiyacını karşılar

Veri kaybı

- ❑ Bazı uygulamalarda (örneğin, ses) az veri kaybını önemli değildir
- ❑ Bazı uygulamalar %100 verinin güvenli iletimini gerektirir (örneğin, dosya transferi)

Zamanlama

- ❑ Bazı uygulamalarda (örneğin, internet telefonu, etkileşimli oyunlar) az bir gecikme kabul edilebilir

Aktarılan veri miktarı

- ❑ Bazı uygulamalarda (örneğin, multimedya) minimum bir işin üretilmesi gerekir

Güvenlik

- ❑ Şifreleme (Encryption), veri bütünlüğü, ...

Ortak uygulamaların taşıma hizmeti gereksinimleri

Uygulama	Veri kaybı	Aktarılan veri mik.	Zaman duyarlı
Dosya transferi	Kayıp olmaz	esnek	Hayır
e-mail	Kayıp olmaz	esnek	Hayır
Web belgeleri	Kayıp olmaz	esnek	Hayır
gerçek zamanlı ses/video	Kayba dayanıklı	ses: 5kbps-1Mbps video:10kbps-5Mbps	evet, 100 mili sn
stored audio/video	Kayba dayanıklı	Yukarısı ile aynı	evet, birkaç sn
interactive games	Kayba dayanıklı	Birkaç kbps	evet, 100 mili sn
instant messaging	Kayıp olmaz	esnek	Evet ve hayır

İnternet taşıma protokolleri hizmetleri

TCP hizmeti:

- ❑ *Bağlantı yönelimli:* istemci ve sunucu işlemleri arasında kurulum gerekir
- ❑ Gönderen ve alan işlemler arasında *güvenilir veri aktarım* yapılır
- ❑ *Akışın denetimi:* veri akışı denetlenir
- ❑ *Tıkanıklık denetimi:* ağ aşırı yoğun olduğunda gönderen veriyi azaltır

UDP hizmeti:

- ❑ Gönderen ve alıcı arasında güvenli olmayan veri transferi
- ❑ Sağlamaz: bağlantı kurulumu, güvenilirlik, akış kontrolü, tıkanıklık denetimi, zamanlama ve minimum aktarılan veri garantisi

S: Neden UDP var?

İnternet uygulamaları: Uygulama, taşıma protokolleri

	Uygulama	Uygulama katmanı protokolü	Ulaşım protokolü
Uzaktan	e-mail	SMTP	TCP
	terminal erişimi	Telnet	TCP
	Web	HTTP	TCP
	dosya aktarımı	FTP	TCP
	Multimedya akışı	HTTP (örnek, Youtube), RTP	TCP veya UDP
	İnternet telefonu	SIP, RTP, özel (örnek, Skype)	Genellikle UDP

2.2 Web ve HTTP

Bazı tanımlar

- ❑ Web sayfası nesnelerden oluşur
- ❑ Bir nesne HTML dosyası, JPEG resmi, Java uygulaması, ses dosyası... olabilir
- ❑ Web sayfası, başvuru nesneleri içeren **temel HTML dosyasıdır**
- ❑ Her bir nesnenin, bir **URL** adresi vardır
- ❑ Örnek URL:

`www.fatih.edu.tr/btp/index.html`

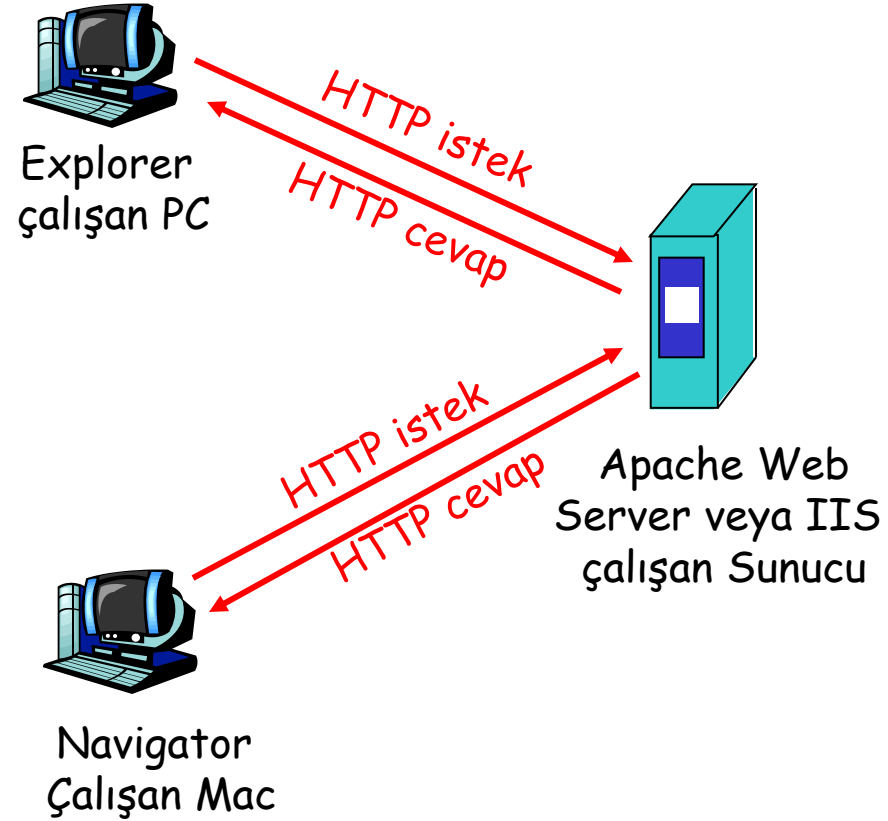
Ana bilgisayar

Yol adı

HTTP genel bakış

Köprü Metni Aktarım Protokolü (hypertext transfer protocol -HTTP)

- ❑ Web uygulama katman protokolüdür
- ❑ client/server modeli
 - ❖ *client*: web nesnelerini ister, alır ve tarayıcıda gösterir
 - ❖ *server*: Web sunucusu, istenen nesneleri gönderir



HTTP genel bakış (devamı)

TCP kullanılır:

- ❑ Client, server'a port 80 üzerinden TCP bağlantısını başlatır (socket oluşturur)
- ❑ Server, client'tan TCP bağlantısını kabul eder
- ❑ HTTP mesajları, web tarayıcı(HTTP client) ve Web server(HTTP server) arasında alış-verişi yapılır
- ❑ Web server TCP bağlantısı kapatılır

HTTP durum bilgisi tutmaz

- ❑ Server, client istekleri hakkında bilgi tutmaz

Durum bilgisini tutmak karmaşıktır

Eğer geçmiş durum bilgisi tutulursa server/client çöktüğünde geçmiş durum bilgisi kullanılarak veri kurtarılır

HTTP bağlantıları

Kalıcı olmayan HTTP bağlantısı

- En çok bir nesne TCP bağlantısı üzerinden gönderilir.

Kalıcı HTTP bağlantısı

- Client ve server arasındaki bir bağlantı üzerinden birden çok nesne gönderilebilir

Kalıcı olmayan HTTP bağlantısı

Kullanıcının aşağıdaki URL'ye girdiğini kabul edelim

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

10 resme bağlantı metin içerir

1a. HTTP client, HTTP server'a
`www.someSchool.edu` port 80
üzerinden TCP bağlantısını
başlatır

1b. `www.someSchool.edu` da
bulunan HTTP server 80.
porta, TCP bağlantılarını
bekler. Client gelen isteği
kabul eder

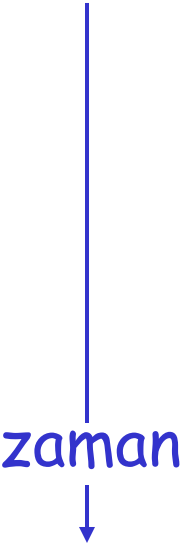
2. HTTP client, HTTP *istek*
mesajını (URL içeren) TCP
bağlantı soketine gönderir.
Mesaj client'ın
`someDepartment/home.index`
adresinden nesneleri istediğini
gösterir

3. HTTP server, istek mesajını
alır, istenen nesneleri içeren
cevap mesajını oluşturur ve
soketine gönderir

zaman




Kalıcı olmayan HTTP bağlantısı(devamı.)



5. HTTP client, html dosyasını içeren cevap mesajını alır ve html'i gösterir. html dosyasını ayrıştırarak 10 jpeg nesnesinin referanslarını bulur

6. 1-5 adımlar her 10 jpeg nesnesi için tekrarlanır



4. HTTP server, TCP bağlantısını kapatır.

Kalıcı olmayan HTTP bağlantısı:

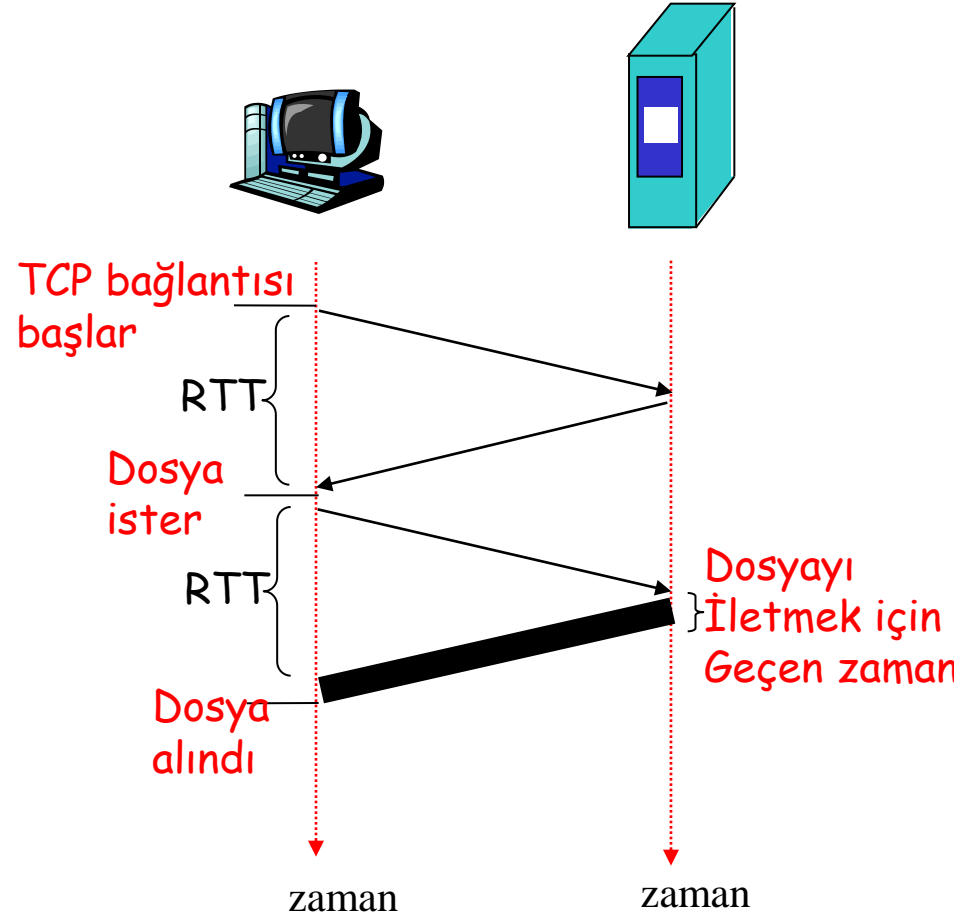
Cevap zamanı

**RTT (Retrieval-to-time-
alınma zamanı):** küçük bir paketin, client'tan server'a gitmesi ve gelmesi için geçen zaman

Tepki süresi (Response time):

- Bir RTT, TCP bağlantısını başlatır
- Bir RTT, HTTP istek ve dönen HTTP cevabın ilk birkaç baytı için
- Dosya iletim süresi

toplam= 2RTT + iletime zamanı



Kalıcı HTTP bağlantısı

Kalıcı olmayan HTTP bağlantı sorunları:

- ❑ Bir nesne için 2 RTT gerekir
- ❑ Her TCP bağlantı işletim sistemine yük getirir
- ❑ Web tarayıcılar, istenen nesneleri getirmek için paralel TCP bağlantıları açarlar

Kalıcı HTTP bağlantısı

- ❑ Server cevap gönderdikten sonra bağlantıyı açık bırakır
- ❑ client/server arasındaki sonraki HTTP mesajlar açık bağlantı üzerinden gönderilir
- ❑ Client, başvuru bir nesne bulduğunda kısa sürede nesne için istek gönderir
- ❑ Bir RTT ile istenen nesnelerin hepsi getirilir

HTTP istek mesajı

- HTTP mesajları: *request (istek), response (cevap)*
- **HTTP istek mesajı:**
 - ❖ ASCII (okunabilir biçimde)

İstek satır
(GET, POST,
HEAD komutları)

Başlık
satırları

satırbaşı,
boş satır
mesajın bittiğini
gösterir

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

Metot türleri

HTTP/1.0

- ❑ GET
- ❑ POST
 - ❖ Veriyi sunucuda mesajın gövde kısmına yükeler
- ❑ HEAD
 - ❖ İstenilen nesnelerin cevap mesajı dışında bırakılması sunucuya sorulur

HTTP/1.1

- ❑ GET, POST, HEAD
- ❑ PUT
 - ❖ Dosyayı, gövde metninde belirtilen URL alanına yükler
- ❑ DELETE
 - ❖ URL alanında belirtilen dosyayı siler

HTTP cevap mesajı

Durum satırı
(protokol
durum kodu
durum ifadesi)

Başlık satırı

veri, örneğin,
istenen
HTML dosyası

HTTP/1.1 200 OK

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 veri veri veri veri ...

HTTP cevap durum kodları

Client cevap mesajları

200 OK

- ❖ istek başarılı oldu, bu mesajdaki sonraki nesneyi iste

301 Moved Permanently

- ❖ İstenen nesne taşındı, yeni konumu bu mesajda sonra belirtilecek (Location:)

400 Bad Request

- ❖ İstek mesajı sunucu tarafından anlaşılır değil

404 Not Found

- ❖ İstenen belge sunucuda bulunmuyor

505 HTTP Version Not Supported

Kullanıcı-sunucu tanımlama bilgileri (cookies)

Birçok büyük web sitesi
cookie kullanır

Dört bileşeni:

- 1) HTTP cevap mesajının
cookie başlık satırı
- 2) HTTP istek mesajında
cookie başlık satırı
- 3) cookie dosyası kullanıcı
bilgisayarında tutulur ve
kullanıcının web
tarayıcısı tarafından
yönetilir
- 4) Web sitesinde veri
tabanı bulunur

Örnek:

- ❑ Ayşe her zaman internete
PC'den giriş yapıyor
- ❑ Bir e-ticaret sitesini ilk kez
ziyaret etti
- ❑ Başlangıç HTTP istekleri
siteye ulaştığında, site
aşağıdakileri oluşturur:
 - ❖ Benzersiz ID (kullanıcıyı
tanımak için verilen
numara)
 - ❖ ID'yi Web sitesindeki
veri tabanına kayıt eder

Cookies (devamı)

client

server



cookie dosyası



1 hafta sonra:



Normal http istek msg

Normal http cevap
cookie: 1678

Normal http istek msg
cookie: 1678

Normal http cevap msg

Normal http istek msg
cookie: 1678

Normal http cevap msg

Amazon server
Kullanıcı için 1678
numaralı ID
oluşturdu

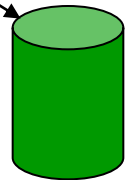
Cookie ile
ilgili işlem

Cookie ile
ilgili işlem

Girdi
oluşturma

giriş

giriş



Veri tabanı

Cookies (devamı)

Cookie hangi bilgileri getirir:

- ☐ Yetkilendirme(Authorization)
- ☐ Alışveriş kartları
- ☐ Öneriler
- ☐ Kullanıcı oturum durumu (Web e-mail)

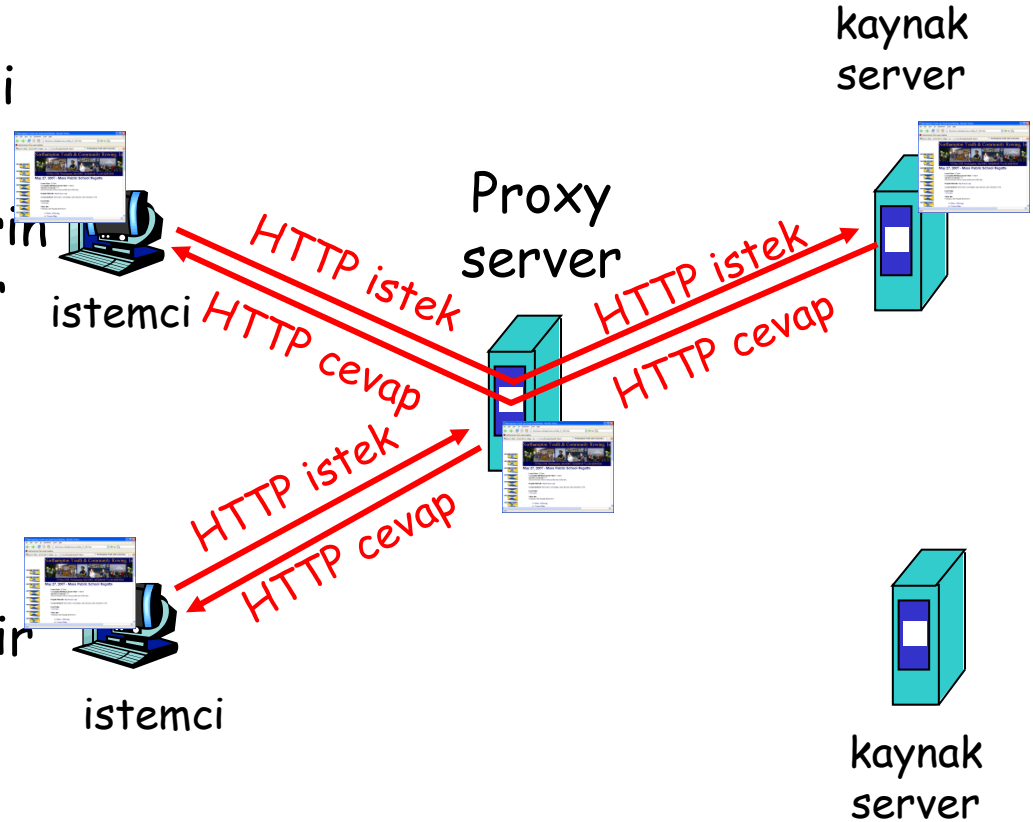
Cookie ve gizlilik:

- ☐ Cookie, sitelerin sizin hakkınızda bilgi almasına izin verir

proxy sunucusu ve önbellekleri

Amaç: Client isteklerini kaynak sunucu olmadan karşılamak

- ❑ Kullanıcı, web tarayıcısını web'e önbellek ile erişeceğini ayarlar
- ❑ Web tarayıcı, HTTP isteklerini hepsini önbelleğe yönlendirir
 - ❖ Nesne önbellekte ise : nesne döndürülür
 - ❖ Değilse, nesne kaynak sunucudan istenir sonra nesne istemciye gönderilir



Proxy Sunucusu: Bir şirkette interneti yerel ağa dağıtan bilgisayara proxy sunucusu denir

Web önbellekleri(devamı)

- ❑ Önbellek hem client hem de server gibi davranır
- ❑ Genellikle önbellekler ISP tarafından yüklenir (üniversite, şirket, ISP)

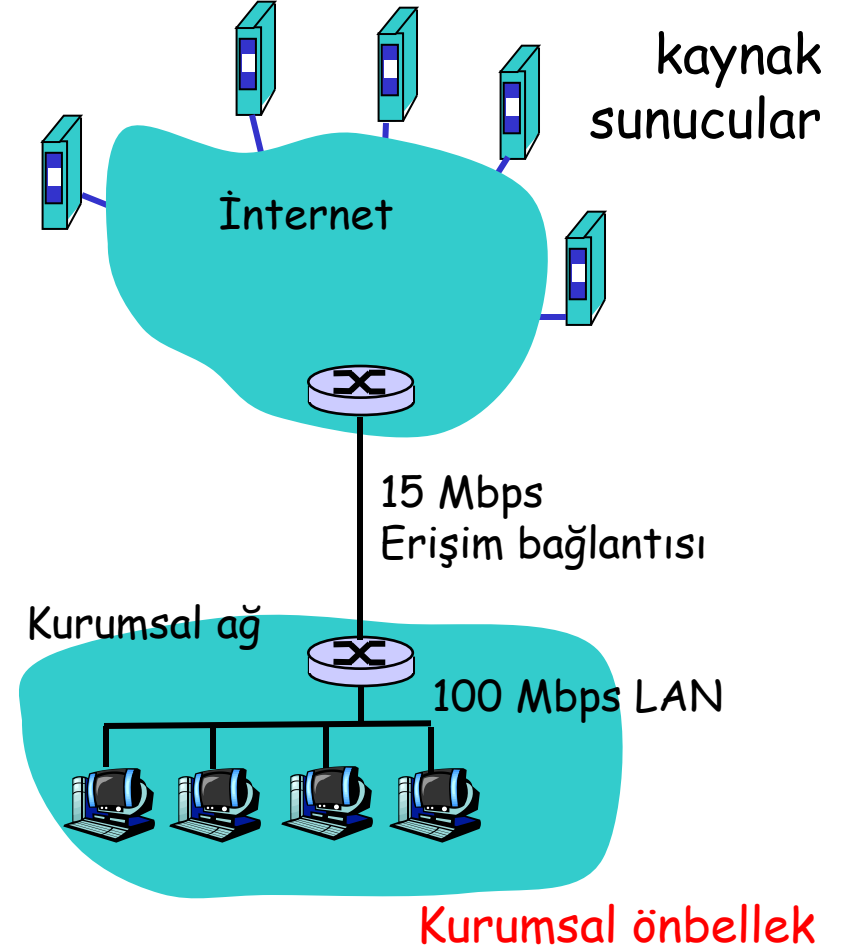
Neden Web önbellekleri?

- ❑ Client isteklerine cevap zamanını azaltır
- ❑ İnternet trafiğini azaltır

Önbellek örneđi

Gecikmeler

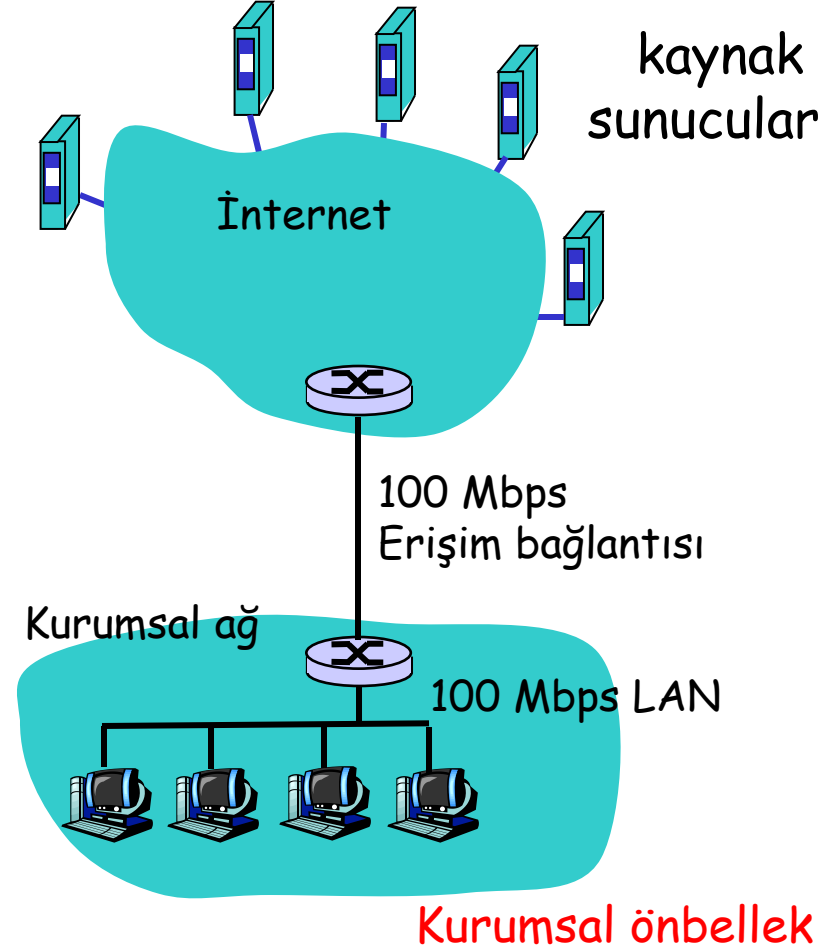
- LAN gecikmesi
- Eriřim bađlantısı gecikmesi
- İnternet gecikmesi



Önbellek örneđi(devamı)

Olası çözüm 1

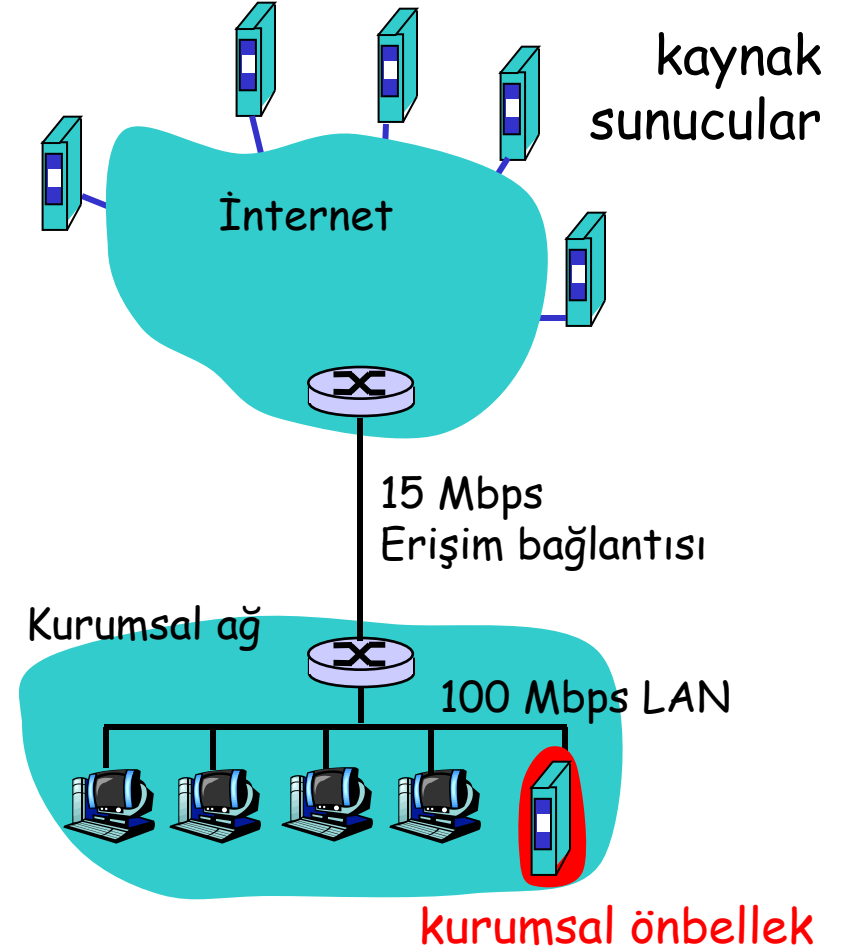
- Erişim bağlantısının bant genişliğini arttırmak, 100 Mbps olsun



Önbellek örneği(devamı)

Olası çözüm 2: önbellek yüklemek

- ❑ Önbellekte bulunan istekleri hemen karşılanacak
- ❑ Önbellekte olmayanlar kaynak sunucu tarafından karşılanacak
- ❑ Erişim bağlantısının kullanımı azalacak
- ❑ Sonuç olarak toplam gecikmelerde de azalma olur



Koşullu GET

