

Bölüm 2

Uygulama Katmanı



Bölüm 2:Uygulama Katmanı

- ❑ 2.1 Ağ uygulamalarının prensipleri
- ❑ 2.2 Web & HTTP
- ❑ 2.3 FTP
- ❑ 2.4 Elektronik posta
 - SMTP, POP3, IMAP
- ❑ 2.5 DNS
- ❑ 2.6 P2P Eşler arası uygulamalar
- ❑ 2.7 TCP ile soket programlama
- ❑ 2.8 UDP ile soket programlama

Bölüm 2: Uygulama katmanı

Amaç:

- Uygulama katmanı ile
 - Taşıma katmanı servisleri
 - client-server
 - peer-to-peer
- Protokoller hakkında bilgi edinmek
 - HTTP
 - FTP
 - SMTP / POP3 / IMAP
 - DNS

Bazı nt uygulamaları

- ❑ E-mail
- ❑ Web
- ❑ messenger
- ❑ Uzak erişim
- ❑ P2P dosya paylaşımı
- ❑ Çok kullanıcılı nt oyunları
- ❑ Video klipler
- ❑ Internet telefonu
- ❑ Video konferans

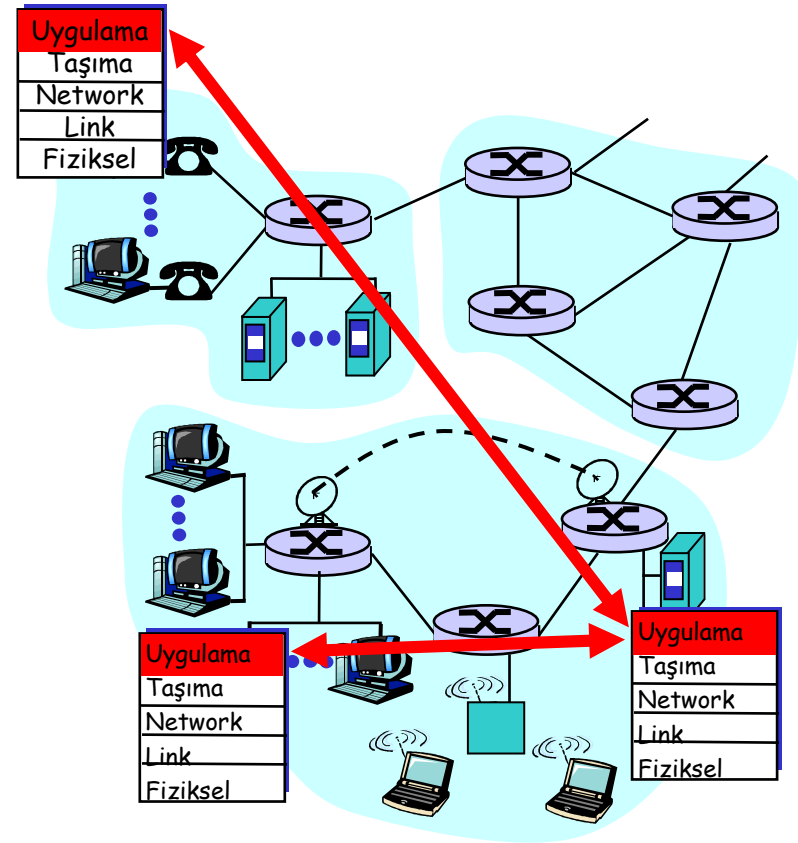
Bir nt uygulaması oluşturmak

- Başka sistemlerde çalışan
- Nt üzerinden haberleşen
- örn Web: Web server yazılımının tarayıcı ile haberleşen

Programlar yazmak.

Nt çekirdeğindeki cihazlar için program yazılmıyor

- Çekirdekteki cihazlarda uygulama katmanı yok
- Bu şekilde daha kolay uygulama yapılabiliyor



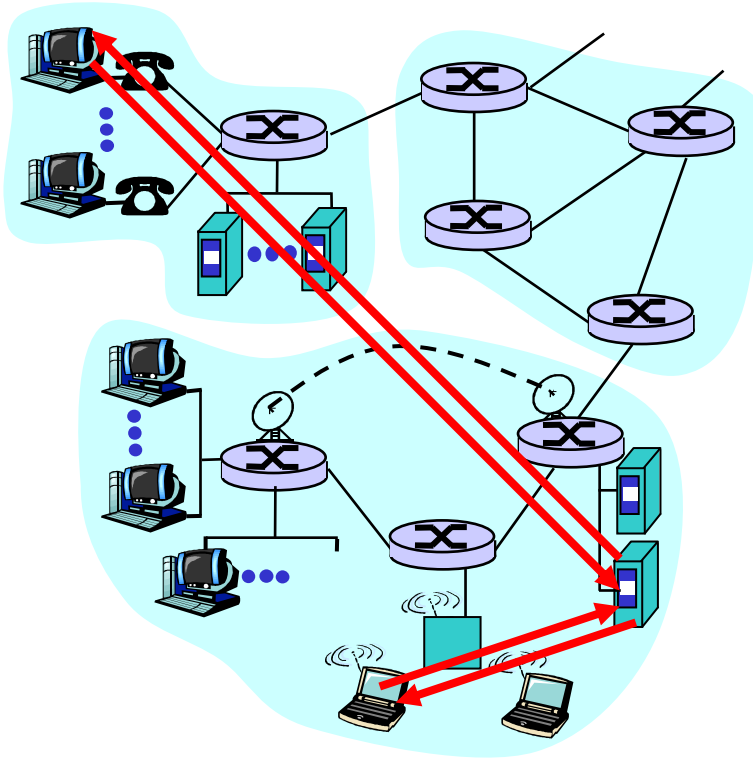
Bölüm 2:Uygulama Katmanı

- ❑ 2.1 Ağ uygulamalarının prensipleri
- ❑ 2.2 Web & HTTP
- ❑ 2.3 FTP
- ❑ 2.4 Elektronik posta
 - SMTP, POP3, IMAP
- ❑ 2.5 DNS
- ❑ 2.6 P2P Eşler arası uygulamalar
- ❑ 2.7 TCP ile soket programlama
- ❑ 2.8 UDP ile soket programlama

Uygulama Mimarileri

- ❑ Client-server
- ❑ Peer-to-peer (P2P)
- ❑ client-server ve P2P birleşimi

Client-server mimarisi



server:

- Her zaman açık
- sabit IP adresi
- Büyüklüğe göre ana sistem kümesi(sunucu çiftliği)

client:

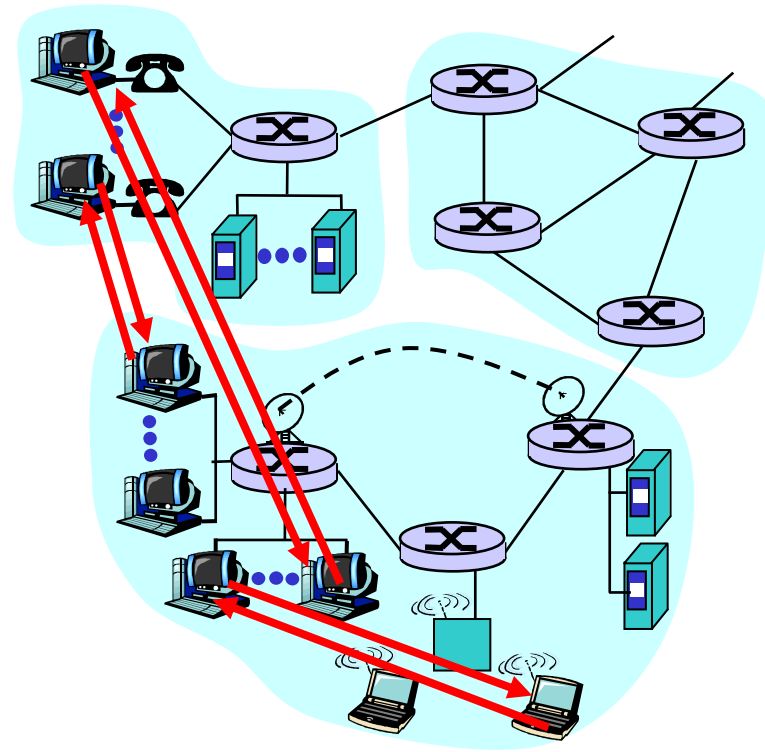
- Serverla haberleşir
- Her zaman açık değil
- değişen IP adresine sahip
- Sunucular birbirleriyle direkt görüşmüyor

Saf P2P mimarisi

- ❑ Her zaman açık bir server yok
- ❑ Uç sistemler direkt haberleşir
- ❑ Ip adresleri değişir
- ❑ örn: Gnutella

Büyür

Kendi kendine
ölçeklenebilir



Süreç iletişimleri

Süreç (Process): uç sistemde çalışan program.

- Aynı uç sistemde 2 süreç **haberleşebilir**
- Farklı uç sistemlerdeki süreçler **mesajlar** sayesinde haberleşirler

Client süreci:

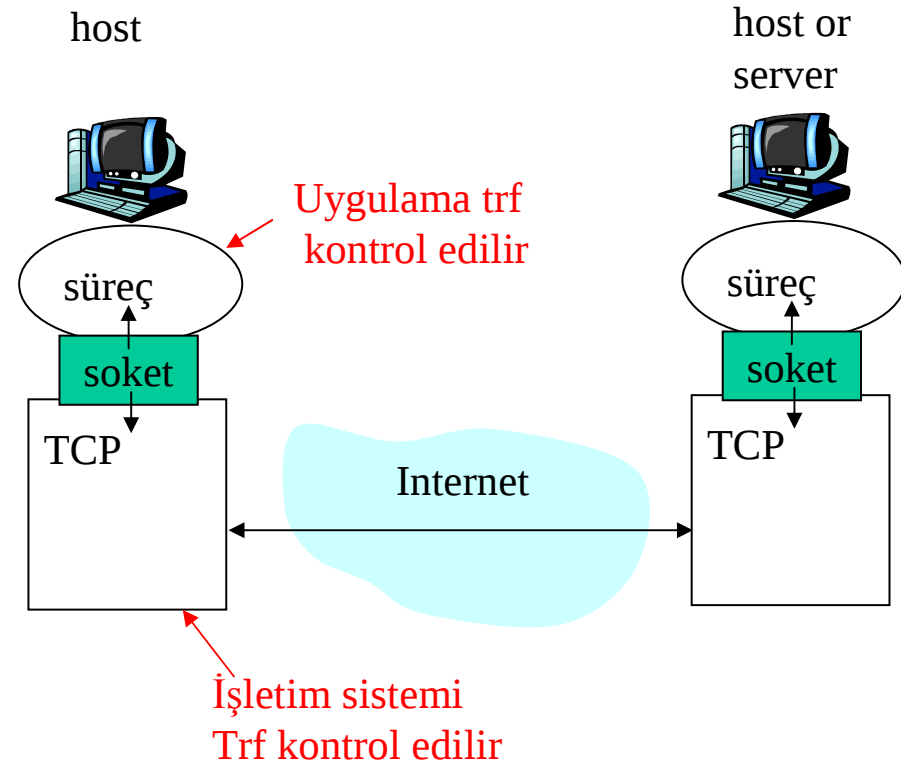
haberleşmeyi başlatan süreç

Server süreci: iletişim kurulmayı bekleyen süreç

- Not:P2P mimarileri uygulamaları client ve server süreçlerine sahiptir(2si ne de)

Soketler

- Süreçler mesajları **soket**lerden gönderir/alır
- Süreç=ev dersek
Soketler kapıya benzer
 - Süreç msj göndermek istediğinde msjı kapıya doğru iter



Adresleme süreci

- ❑ Bir sürecin mesaj alabilmesi için bir adresi olması lazım
- ❑ Bir uç sistemin 32-bit IP adresi vardır
- ❑ **S:** Ip adresi uç sistemde çalışan süreçlerden hangisinin haberleştiğini belirler mi?
- ❑ **C:** Hayır Ip adresi sadece uç sistemi gösterir, süreçleri göstermez
- ❑ Ip adresinin yanında bir de **port numarası** uç sistemdeki süreci belirler
- ❑ Bazı port numaraları:
 - HTTP server: 80
 - Mail server: 25
- ❑ Portlarla ilgili ilerleyen slaytlarda ek bilgi...

Uygulama katmanı protokolleri şunları belirler:

- ❑ Alınan ve gönderilen msjların tipini
- ❑ Msj tiplerinin şekli: msjlardaki alanlar nasıl olacak
- ❑ Msjların alanlarındaki bilgilerin anlamı
- ❑ Gelen msjlara ne zaman ve nasıl cevap verileceği ile ilgili kurallar

Genel protokoller:

- ❑ RFC'de tanımlıdır
- ❑ İşbirliğine izin verir
- ❑ Örn: HTTP, SMTP

özel protokoller:

- ❑ Örn: KaZaA

Uygulamalar trf kullanılabilen taşıma servisleri

Data kaybı

- ❑ Ses gibi datalarda kayıp olabilir
- ❑ Dosya transferi gibi uygulamalar %100 güvenli olmalıdır

Zamanlama

- ❑ İnternet telefonu, oyun vs. gibi uygulamalarda gecikme olmamalıdır

Bant genişliği

- ❑ Esnek uygulamalar ne kadar bant genişliği varsa onu kullanır;
- ❑ Multimedya bant genişliğine duyarlıdır

Taşıma servisleri

	uygulama	Veri kaybı	Bant genişliği	Zamana duyarlı
	Dosya transferi	yok	esnek	hayır
	e-mail	yok	esnek	hayır
	Web	yok	esnek	hayır
Gerçek zamanlı video		olabilir	audio: 5kbps-1Mbps video:10kbps-5Mbps	ewet
Kayıtlı ses,video		olabilir	Yukarıdaki ile aynı	ewet
	oyun	olabilir	Birkaç kb	ewet
	msn	yok	elastik	ewet

Internet taşıma protokolü servisleri

TCP servis:

- ❑ *Bağlantı tabanlı:* client-server süreçleri arasında ilk bağlantı kurulması
- ❑ *Güvenilir taşıma*
- ❑ *Akış kontrolü:* gönderen alıcıyı bunaltmaz
- ❑ *Tıkanıklık kontrolü:* nt te aşırı yükleme olunca gönderen hızını keser
- ❑ *sağlamadığı:* zamanlama, minimum bant genişliği

UDP servis:

- ❑ Güvenilir olmayanda data transferi
- ❑ *Sağlamadıkları:* bağlantı yok, güvenilirlik yok, akış ve tıkanıklık kontrolü yok,
- ❑ zamanlama ve bant genişliği garanti

Internet uygulamaları

Uygulama	Uygulama katmanı protokolü	Taşıma protokolü
e-mail	SMTP [RFC 2821]	TCP
Uzaktan erişim	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
Dosya transferi	FTP [RFC 959]	TCP
multimedya	özel (e.g. RealNetworks)	TCP or UDP
Internet telefonu	özel	UDP

Bölüm 2:Uygulama Katmanı

- ❑ 2.1 Ağ uygulamalarının prensipleri
- ❑ 2.2 Web & HTTP
- ❑ 2.3 FTP
- ❑ 2.4 Elektronik posta
 - SMTP, POP3, IMAP
- ❑ 2.5 DNS
- ❑ 2.6 P2P Eşler arası uygulamalar
- ❑ 2.7 TCP ile soket programlama
- ❑ 2.8 UDP ile soket programlama

Web ve HTTP

Bazı terimler

- ❑ Web sayfası objelerden oluşur
- ❑ Obje HTML sayfası, JPEG resim, Java applet, ses dosyası,...
- ❑ Web sayfası objeleri içeren **taban HTML-dosyasıdır**
- ❑ Her bir objenin **URL si vardır**
- ❑ Örn URL:

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

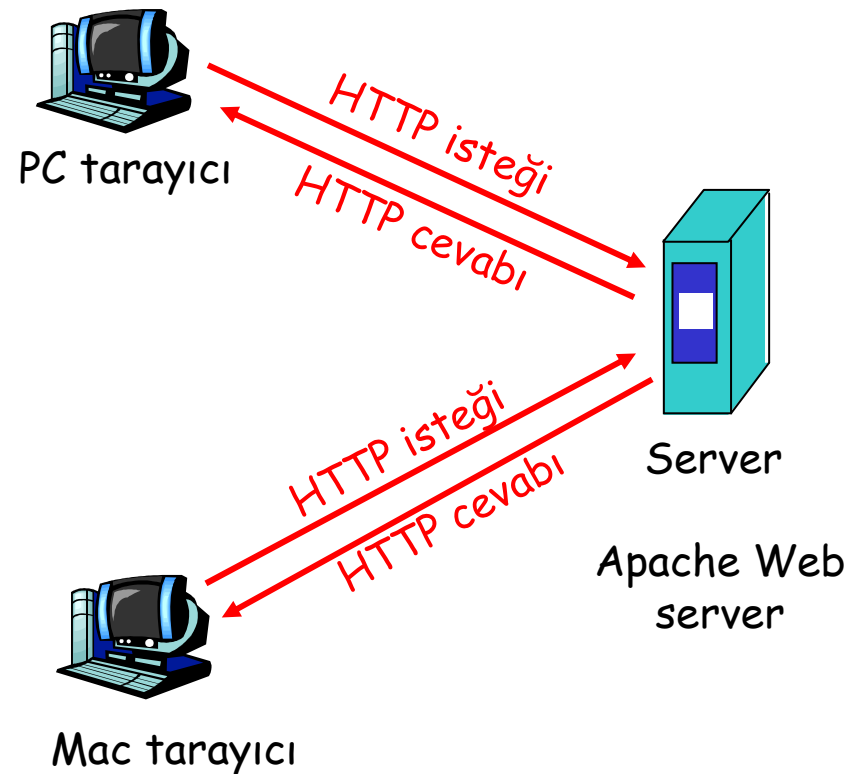
host adı

Yol adı

HTTP

HTTP: hypertext transfer protocol

- ❑ Web'in uygulama katmanı protokolü
- ❑ client/server modeli
 - *client*: web objlerini isteyen, alan ve gösteren tarayıcı
 - *server*: Web server objleri clienta gönderir
- ❑ HTTP 1.0: RFC 1945
- ❑ HTTP 1.1: RFC 2068



HTTP

TCP kullanır:

- ❑ client TCP bağlantısını başlatır (soket açar), port 80 den
- ❑ server clienttan gelen isteği kabul eder
- ❑ HTTP mesajları (uygulama katmanı protokolü mesajları) tarayıcı (HTTP client) ve Web server (HTTP server) arasında değişilir
- ❑ TCP bağlantısı kapatılır

HTTP "durumsuz" dur

- ❑ server eski clientlar hakkındaki bilgileri tutmaz

Bunun yanında

Durumu tutmak zordur!

- ❑ Geçmişin tutulması lazım

HTTP bağlantısı

Kalıcı olmayan bağ. HTTP

- ❑ TCP bağlantısı ile en fazla 1 nesne gönderilir.
- ❑ HTTP/1.0 kalıcı olmayan bağlantılı HTTP yi kullanır

Kalıcı bağ. HTTP

- ❑ Tek bir TCP bağlantısından birçok nesne client server arasında gönderilir
- ❑ HTTP/1.1 kalıcı olan bağlantıyı kullanır

Kalıcı olmayan bağlantılarla HTTP

Kullanıcı URL girer:

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

(text ve 10 jpeg
resim içerir)

1a. HTTP client port 80 den
`www.someSchool.edu` adresli
HTTP serverla TCP bağlantısını
başlatır

1b. `www.someSchool.edu` adresli
HTTP server port 80 den
gelecek bağlantıyı bekler.
Bağlantıyı kabul eder; clienta
bildirir

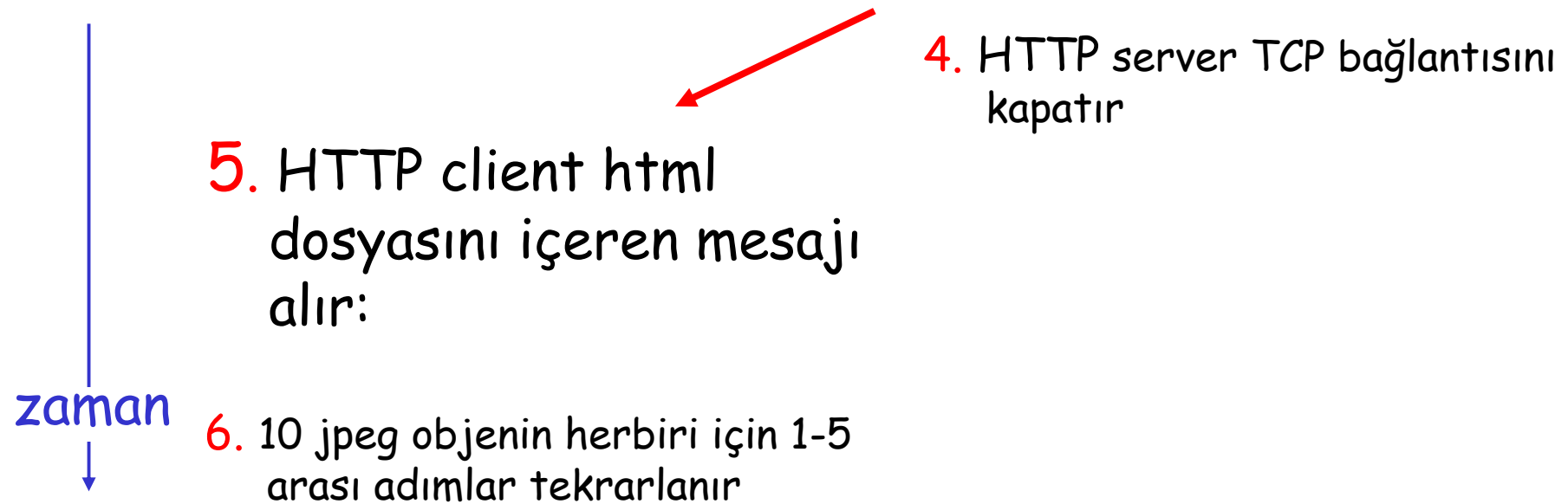
2. HTTP client HTTP URL içeren
istek mesajını TCP bağlantı
soketinden gönderir. Client
aşağıdaki adresi ister
`someDepartment/home.index`

3. HTTP server mesajı alır ve
soketten nesneyi gönderir

zaman



Kalıcı olmayan bağlantılarla HTTP



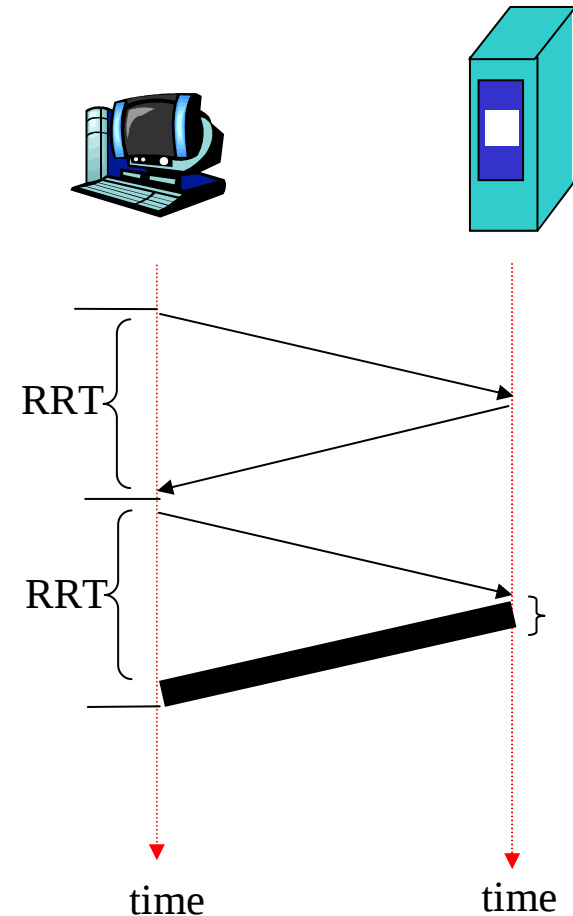
Cevap verme zamanı RRT

RRT: client tan servera ve tekrar serverdan clienta 1 paket gitmesi için gereken zaman.

Cevap verme zamanı:

- ❑ TCP bağlantısını kurmak için 1 RRT
- ❑ HTTP isteği ve ilk bytelerin cevap olarak clienta gelmesi 1 RRT

toplam = 2RRT+gönderme zamanı



HTTP istek mesajı

- 2 çeşit HTTP msjı var: *istek, cevap*
- **HTTP istek mesajı:**
 - ASCII (insanın okuyabileceği format)

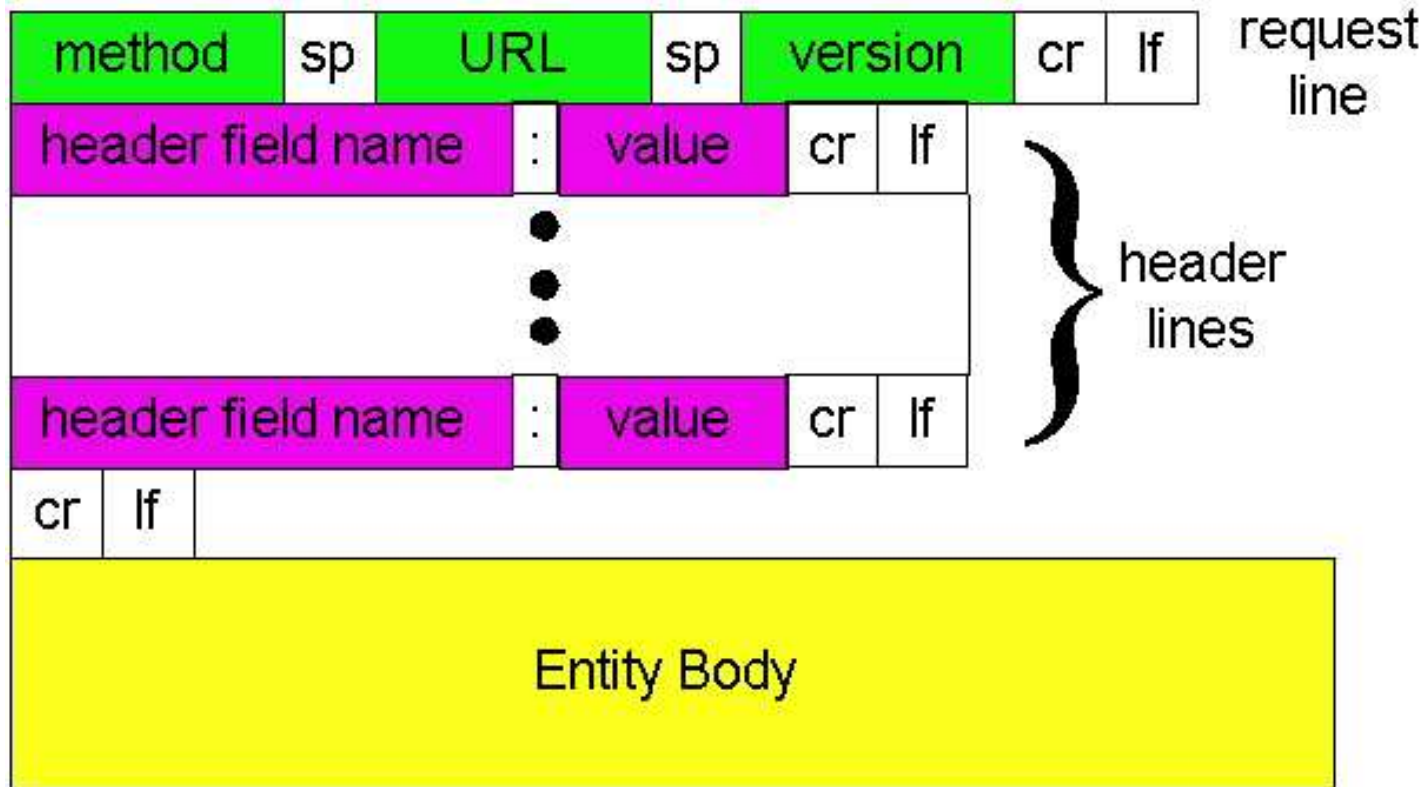
İstek satırı
(GET, POST,
HEAD komutları)

Başlık satırları

\n mesajın
bittiğini gösterir

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

HTTP cevap mesaji: genel format



Form içeriğini upload etmek

Post metodu:

- ❑ Web sayfasında formdan alınan bilgiler olabilir
- ❑ Input servera **entity** bodynin içerisinde gönderilir

URL metodu:

- ❑ GET methodu kullanır
- ❑ Input URL adresinin içinde gönderilir:

`www.somesite.com/search?betul&oktay`

Metod tipleri

HTTP/1.0

- ❑ GET
- ❑ POST
- ❑ HEAD
 - Serverdan cevaba istenen dosyayı eklememesini söylüyor
 - Genelde hata ayıklama için kullanılır

HTTP/1.1

- ❑ GET, POST, HEAD
- ❑ PUT
 - URL alanına dosya upload eder
- ❑ DELETE
 - URL deki dosyayı siler

HTTP cevap mesajı

Durum satırı

HTTP/1.1 200 OK

Başlık satırı

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

İstenilen
datalar

data data data data data ...

HTTP cevabı durum kodları

Serverdaki ilk satır -> clienta cevap .

Bazı örnekler:

200 OK

- İstek kabul edildi, dosya mesajla gönderiliyor

301 Moved Permanently

- İstenilen nesne taşınmıştır, yeni URL cevap mesajının (Location:) kısmındadır

400 Bad Request

- İstek sunucu tarafından anlaşılamadı

404 Not Found

- İstenen belge bu sunucu üzerinde değil

505 HTTP Version Not Supported: istenilen HTTP protokolü sunucu tarafı desteklenmiyor

Kullanıcı taraflı etkileşim: cookiler (kurabiyeler)

Bir çok web sitesi cookie kullanır

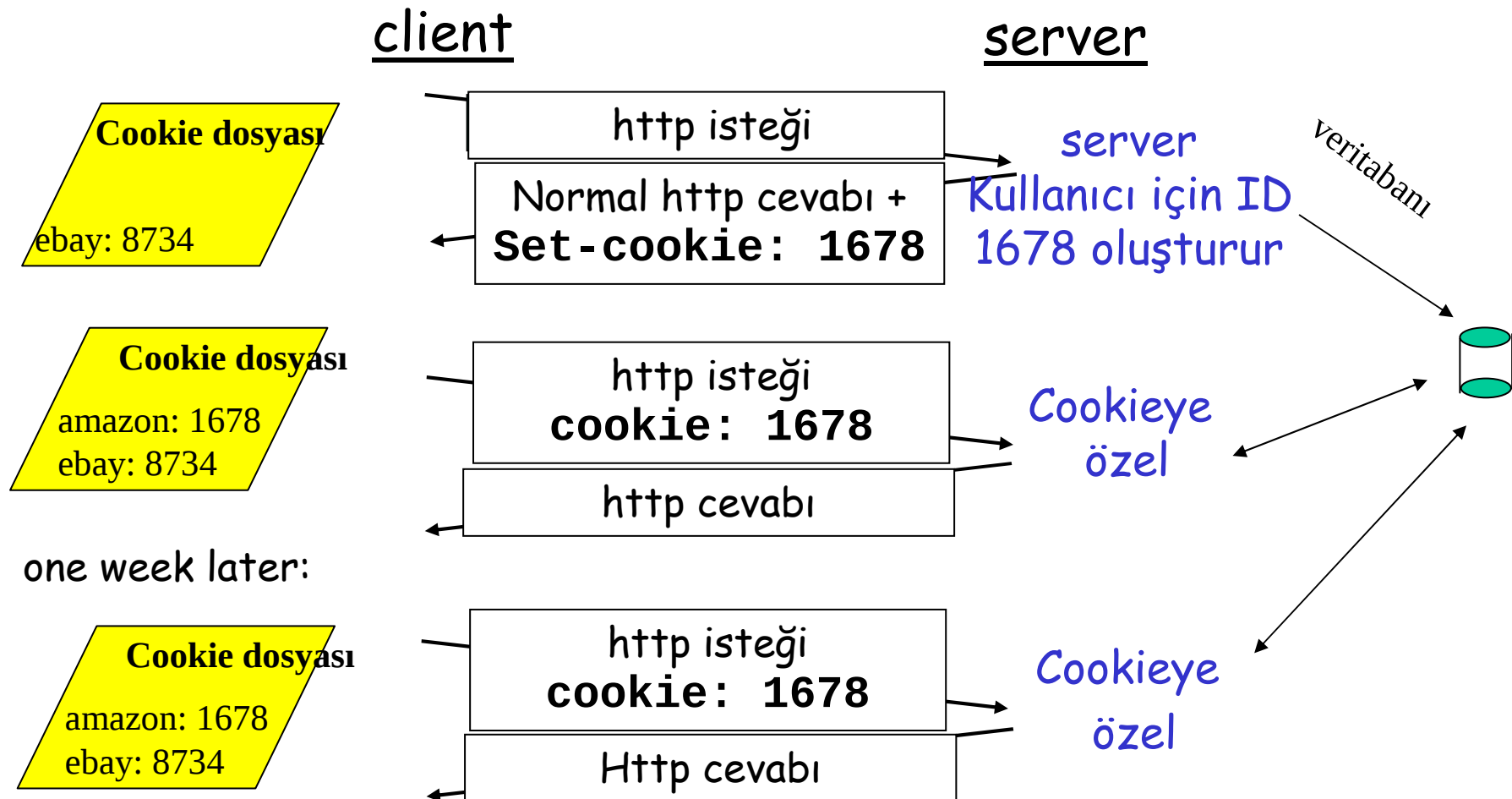
4 bileşeni vardır:

- 1) HTTP cevap msjındaki cookie başlık satırı
- 2) HTTP istek mesajındaki cookie başlık satırı
- 3) Kullanıcının uç sisteminde tutulan bir cookie dosyası
- 4) Web sitesindeki bir veritabanı

Example:

- Susan Internet'e hep aynı bilgisayardan bağlanır
- İlk defa ticari siteyi ziyaret eder
- HTTP isteği siteye ulaşır ve site bir ID oluşturur ve veritabanında da ID ile ilgili bilgileri tutar

Cookieler: "durum"u tutarlar



Cookieler

Cookieler ile:

- ☐ Kimlik kontrolü
- ☐ Alışveriş sepeti
- ☐ tavsiye
- ☐ Kişi oturumu (Web e-mail)

yapılır

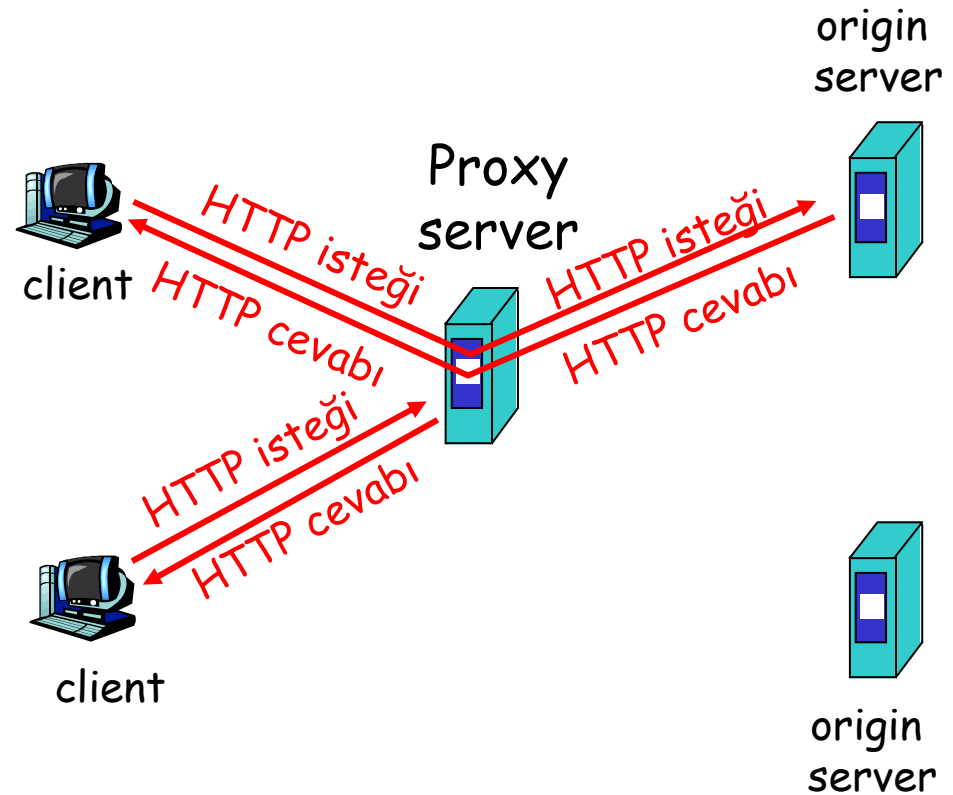
Cookieler ve Kişinin Özeli:

- ☐ Cookielerle özelini öğrenirler
- ☐ Sitelerde email ve isim tutulur
- ☐ Arama motorları
- ☐ Reklam kampanyası

Web cacheleri(tampon belleği) (proxy server)

Amaç: client isteklerini servera gitmeden gerçekleştirmek
Cache?

- Kullanıcı tarayıcıyı açar:
Webe cachten geçerek ulaşılır
- browser tüm HTTP isteklerini cachee yönlendirir
 - İstenen nesne cachee ise nesne clienta gönderilir
 - Yoksa cache nesneyi orjinal serverdan ister ve clienta gönderir



Web cacheleme

- ❑ Cache hem client hem server gibi davranır
- ❑ Cache ISP tarafından kurulur (universite, şirket vs.)

Neden Web cacheleme?

- ❑ Clienta daha hızlı cevap verebilmek için
- ❑ Kurumun dışarı ile bağlantısında fazla trafik oluşturmamak için

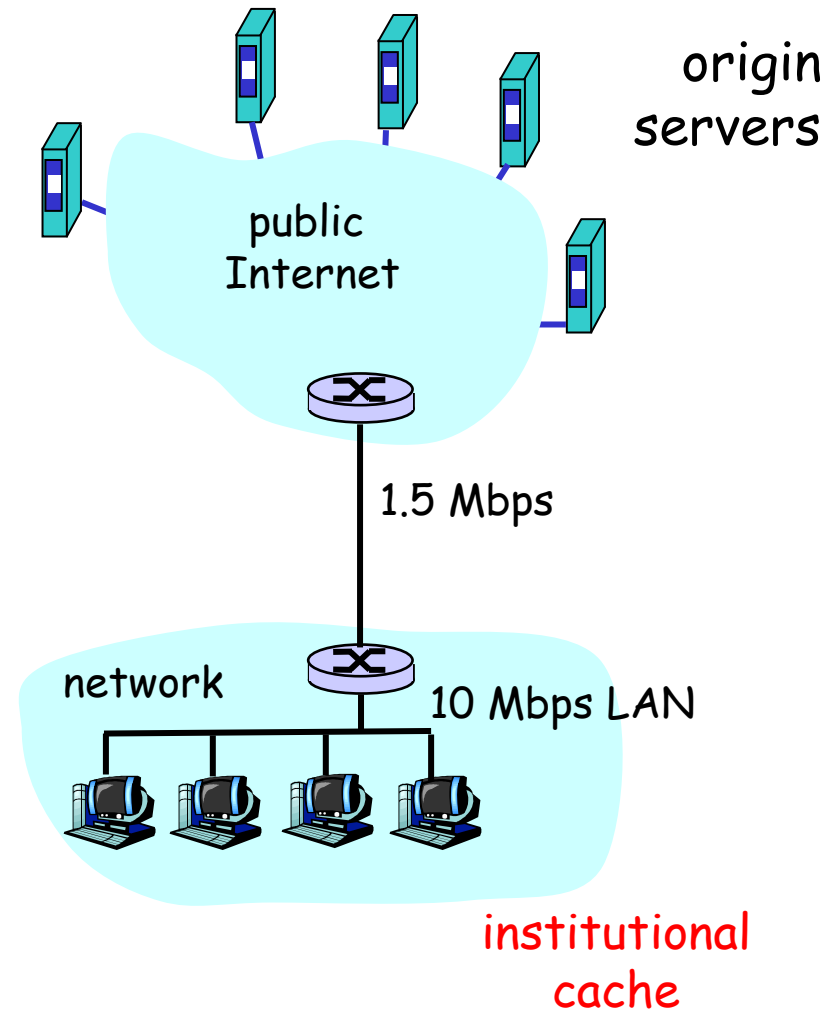
Cache örneği

Assumptions

- ❑ Ortalama nesne = 100,000 bits
- ❑ Clientların orjinal serverdan istekte bulunması = 15 istek/saniye
- ❑ Router ve server arası iletişim = 2 saniye

Consequences

- ❑ LAN = 15%
- ❑ link = 100%
- ❑ Toplam gecikme = Internet gecikmesi +
= 2 saniye +



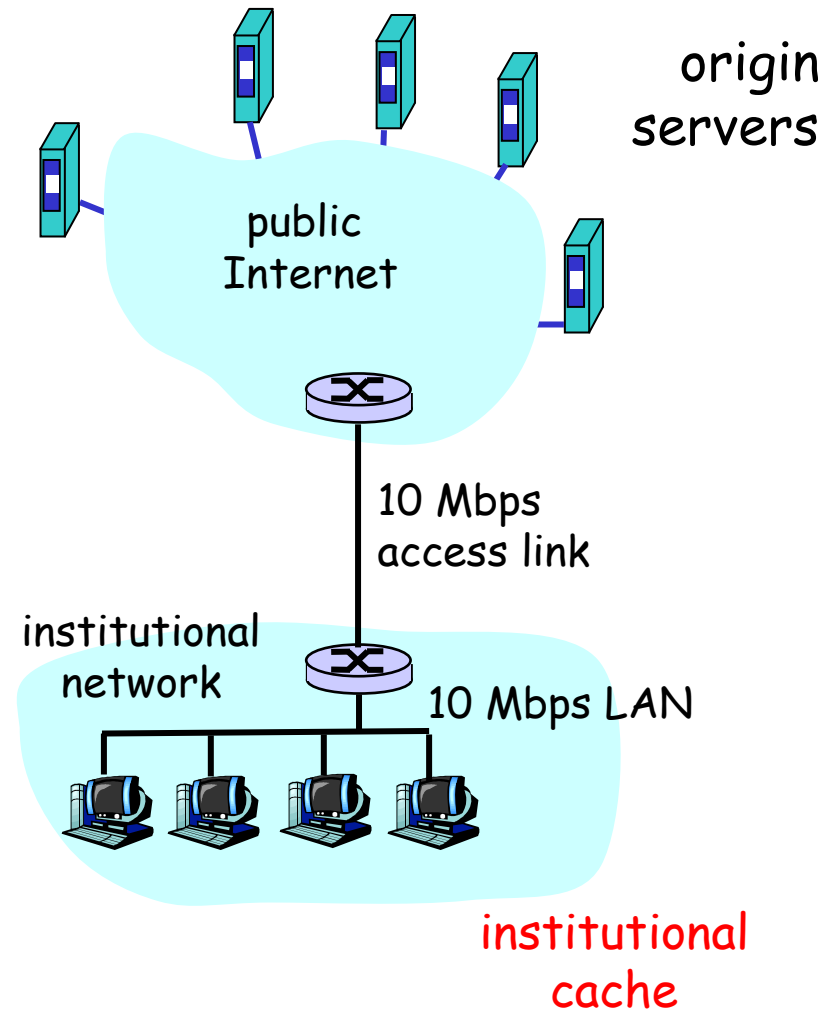
Cache örneği

Çözüm

- ❑ Bant genişliğini arttırmak: 10 Mbps

Consequences

- ❑ LAN = 15% i kullanılır
- ❑ link = 15%i
- ❑ Toplam gecikme = Internet gecikmesi +
- ❑ **Maliyeti arttırmış oluruz**



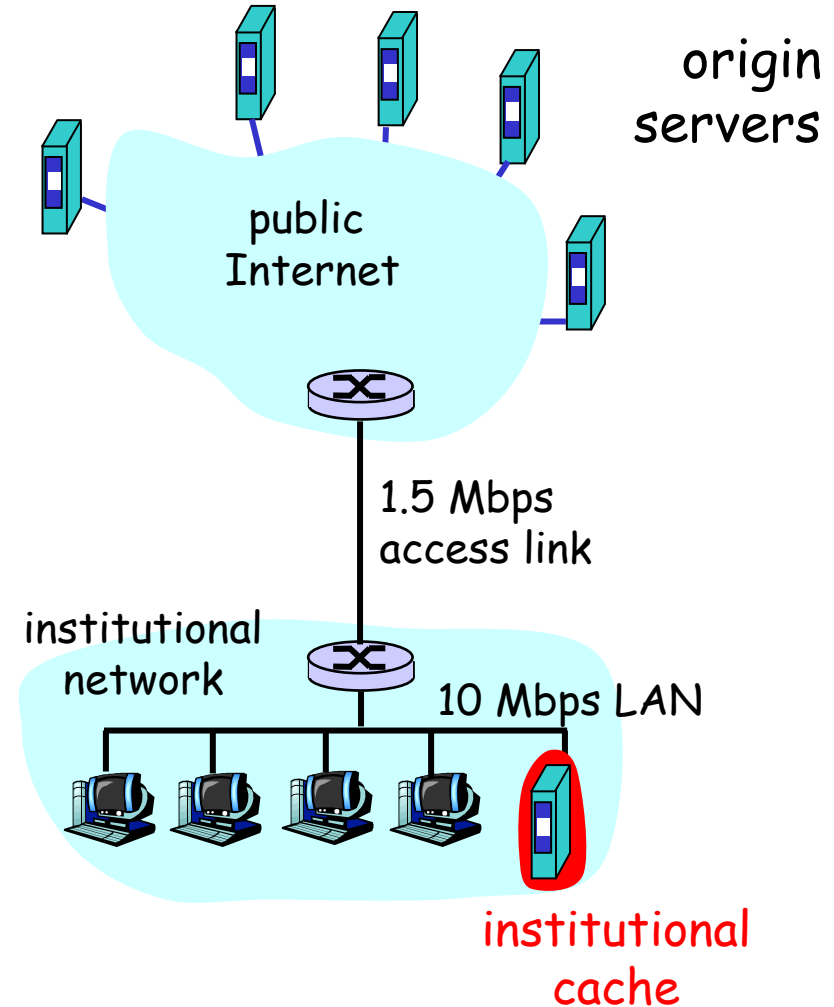
Cache örneği

Cache oluşturmak

- ❑ İstenen verinin cache'de bulunma ihtimali 4

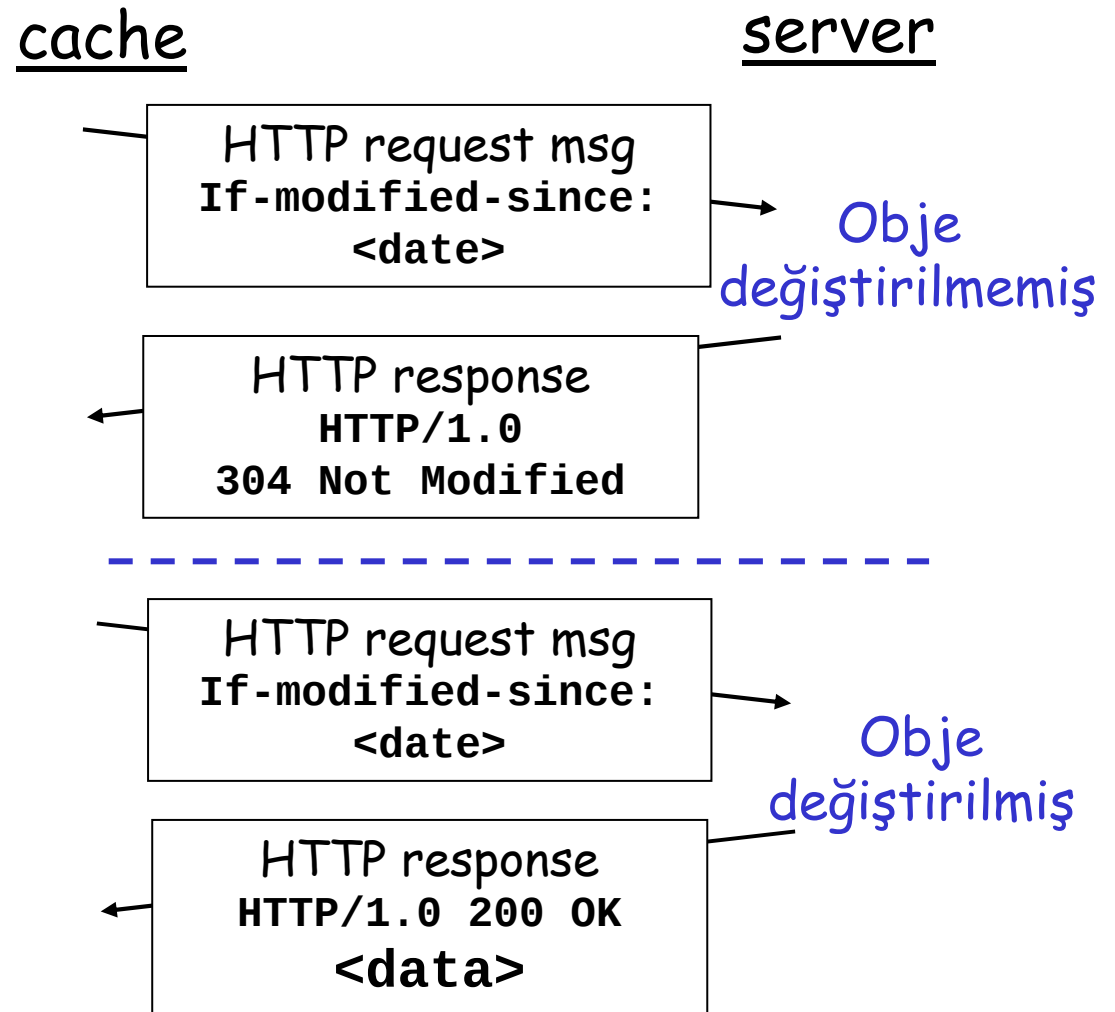
Consequence

- ❑ İsteklerin %40'ı cache'de vardır
- ❑ İsteklerin %60'ı orjinal server tarafından karşılanır
- ❑ Linkin kullanımı %60'a iner.
- ❑ Toplam gecikme azalır.



Conditional GET

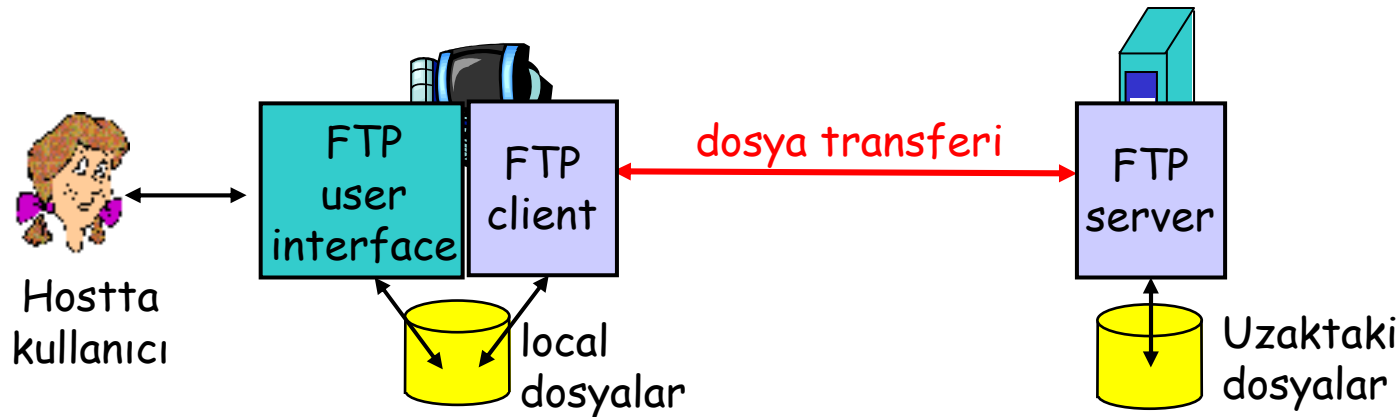
- ❑ **Amaç:** cachede istenen objenin tarihi eski ise alma.
- ❑ **cache:** HTTP isteğinde en son cache gelme zamanı söylenir:
If-modified-since:
<tarih>
- ❑ **server:** eğer obje eski ise aşağıdaki cevap verilir:
HTTP/1.0 304 Not Modified



Bölüm 2:Uygulama Katmanı

- ❑ 2.1 Ağ uygulamalarının prensipleri
- ❑ 2.2 Web & HTTP
- ❑ 2.3 FTP
- ❑ 2.4 Elektronik posta
 - SMTP, POP3, IMAP
- ❑ 2.5 DNS
- ❑ 2.6 P2P Eşler arası uygulamalar
- ❑ 2.7 TCP ile soket programlama
- ❑ 2.8 UDP ile soket programlama

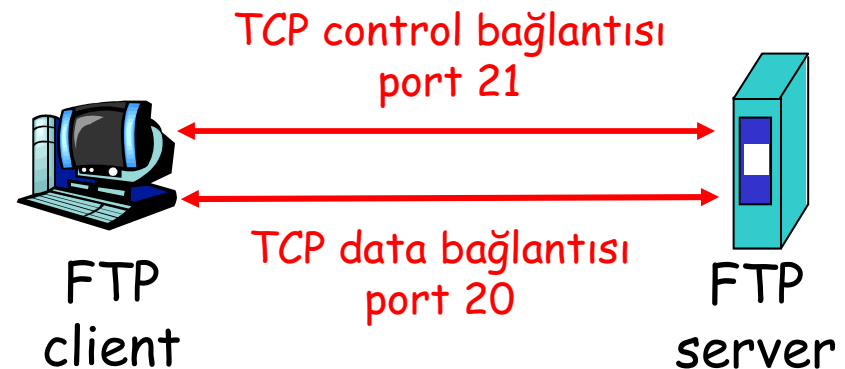
FTP: File Transfer Protocol dosya transfer protokolü



- ❑ Uzaktaki hostla dosya transferi
- ❑ client/server modeli
 - *client*: transferi başlatan taraf (transfer 2 yönlü olabilir)
 - *server*: uzaktaki host
- ❑ ftp: RFC 959
- ❑ ftp server: port 21

FTP: ayrı kontrol ve data bağlantısı

- ❑ FTP client FTP serverla port 21den görüşür,
- ❑ FTP TCP kullanır
- ❑ Client kontrol bağlantısından kimlik bilgilerini gönderir
- ❑ Client kontrol bağlantısından uzaktaki dosyalara bakar
- ❑ Serverla dosya transferi yapılacağı zaman data bağlantısı açılır ve dosya burdan gönderilir.
- ❑ Dosya transferi bitince data bağlantısı kapatılır.



- ❑ Başka dosya transferi için yeni data bağlantısı açılır.
- ❑ Control bağlantısı: "out of band"=bant dışı

FTP komutları ve cevapları

Örnek komutlar:

- ❑ **USER *kullanıcıadı***
- ❑ **PASS *şifre***
- ❑ **LIST** klasördeki dosyaları listeler
- ❑ **RETR *dosyaadı*:**
dosyayı getirir
- ❑ **STOR *dosyaadı*:**
dosyayı servera gönderir

Server cevapları

- ❑ **331 Username OK,
password required**
- ❑ **125 data
connection
already open;
transfer starting**
- ❑ **425 Can't open
data connection**
- ❑ **452 Error writing
file**

Bölüm 2:Uygulama Katmanı

- ❑ 2.1 Ağ uygulamalarının prensipleri
- ❑ 2.2 Web & HTTP
- ❑ 2.3 FTP
- ❑ 2.4 Elektronik posta
 - SMTP, POP3, IMAP
- ❑ 2.5 DNS
- ❑ 2.6 P2P Eşler arası uygulamalar
- ❑ 2.7 TCP ile soket programlama
- ❑ 2.8 UDP ile soket programlama

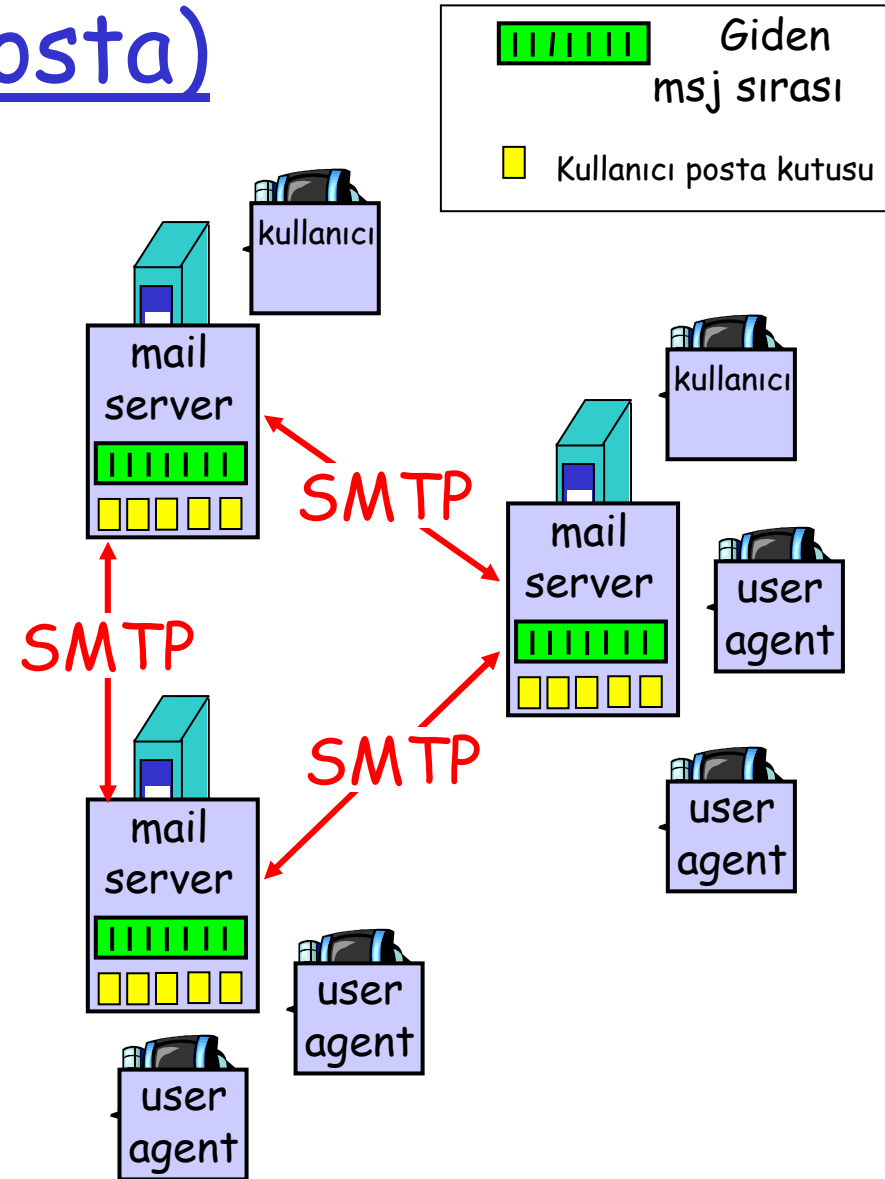
Elektronik Mail (posta)

3 unsur:

- ❑ Kullanıcı temsilcisi
- ❑ Posta sunucusu
- ❑ simple mail transfer protocol: SMTP

Kullanıcı temsilcisi

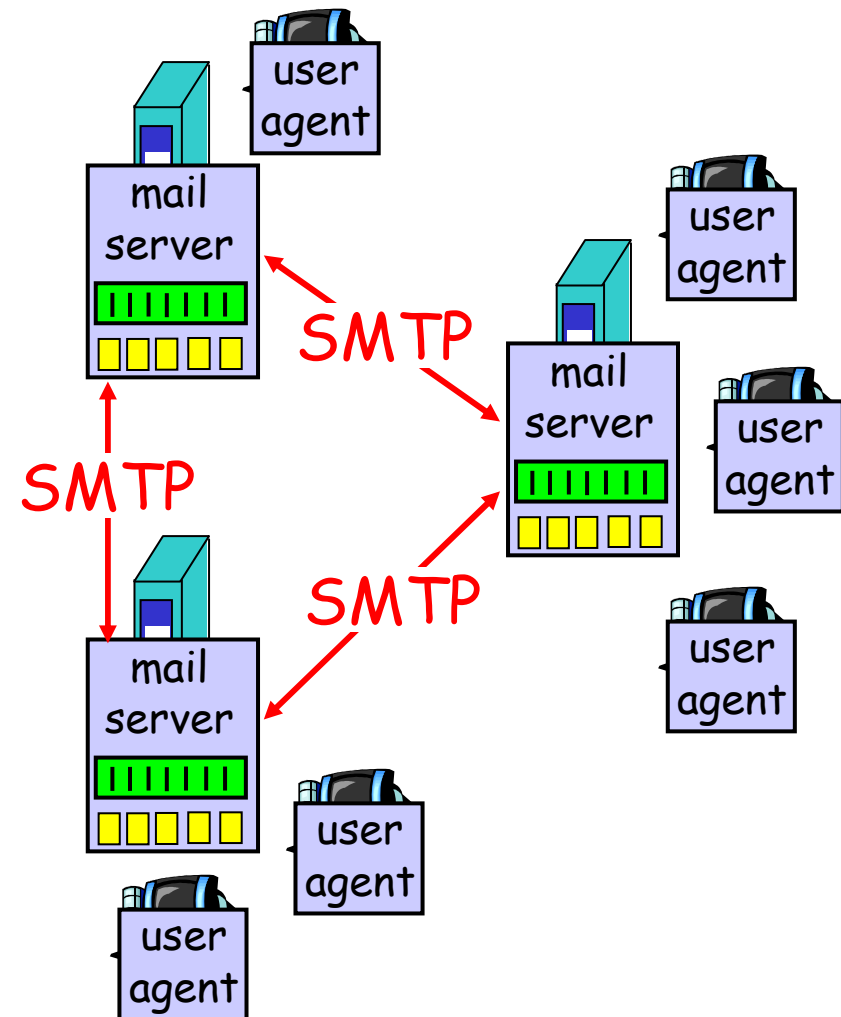
- ❑ Örn: "mail okuyucusu"
- ❑ Mail oluşturma, düzenleme, okuma
- ❑ Örn: Eudora, Outlook, elm, Netscape Messenger
- ❑ Giden ve gelen mailler serverlarda tutulur



Elektronik Mail: mail serverları

Mail Serverları

- ❑ **mailbox** ta kullanıcıya gelen mailler tutulur
- ❑ **message sırası** gönderilecek mailleri saklar
- ❑ **SMTP** protokolü serverlar arasında mail alıp göndermeyi sağlar
 - client: mail gönderen server
 - "server": maili alan server

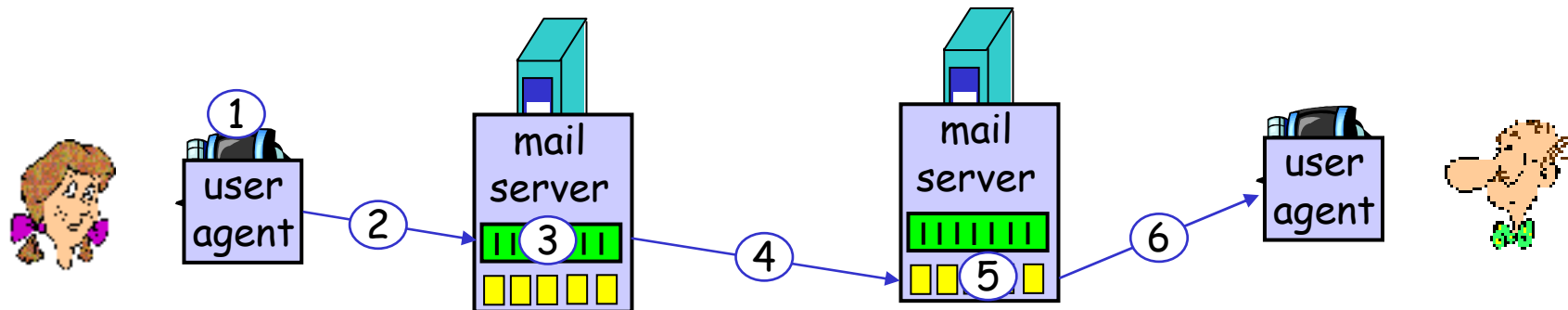


Elektronik Mail: SMTP [RFC 2821]

- ❑ port 25'ten TCP kullanarak güvenli bağlantı ile serverlar arası mail transferini gerçekleştirir.
- ❑ direk transfer: gönderici serverdan alıcı servera
- ❑ Mail transferinde 3 aşama var
 - selamlaşma
 - Mesaj transferi
 - Bağlantının kapatılması
- ❑ Komut ve cevaplarla iletişim
 - komutlar: ASCII text
 - cevap: durum kodu
- ❑ mesajlar 7-bit ASCII koduyla gerçekleşir.

Senaryo: Alice Bob'a mail atıyor

- 1) Alice maili yazar ve "to" bob@someschool.edu
- 2) Alice'in kullanıcı temsilcisi (outlook vs.) maili servera gönderir ve mail serverda sıraya girer
- 3) SMTPnin client tarafı Bobun mail serverı ile TCP bağlantısı kurar
- 4) SMTP client Alice'in mailini TCP bağlantısı üzerinden gönderir
- 5) Bobun mail serverı mail, Bobun mail kutusuna koyar
- 6) Bob kullanıcı temsilcisini kullanarak mailini okur



Örnek SMTP iletişimi

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

Kendiniz SMTPyi deneyebilirsiniz:

- ❑ `telnet servername 25`
- ❑ 220 serverdan cevap
- ❑ HELO, MAIL FROM, RCPT TO, DATA, QUIT komutlarını yaz

SMTP: son

- ❑ SMTP kalıcı bağlantı kullanır
- ❑ SMTP de mail 7-bit ASCII olmalıdır
- ❑ SMTP server alt satır.altsatır kullanarak mailin bittiğini söyler

HTTP:

- ❑ HTTP: çekme
- ❑ SMTP: itme
- ❑ ASCII kodları ile iletişim ve durum kodları
- ❑ HTTP:her nesne kendi isteğine ayrı gönderilir
- ❑ SMTP: msjda birden fazla nesne gönderilebilir.

Mail mesaj formatı

SMTP:

RFC 822: text mesaj formatı:

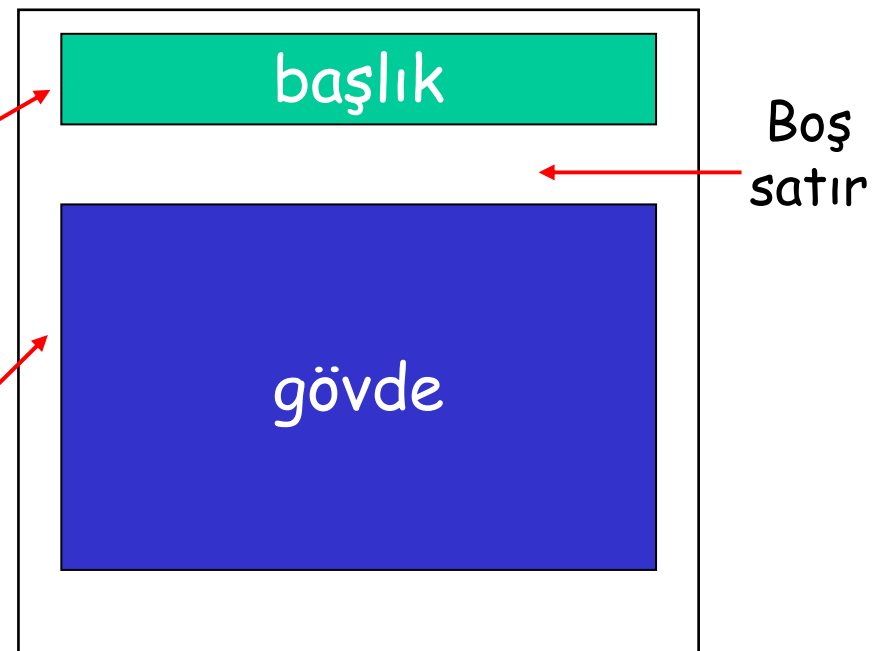
☐ başlıklar

- To: (kime)
- From: (kimden)
- Subject: (konu)

SMTP komutlarından farklı!

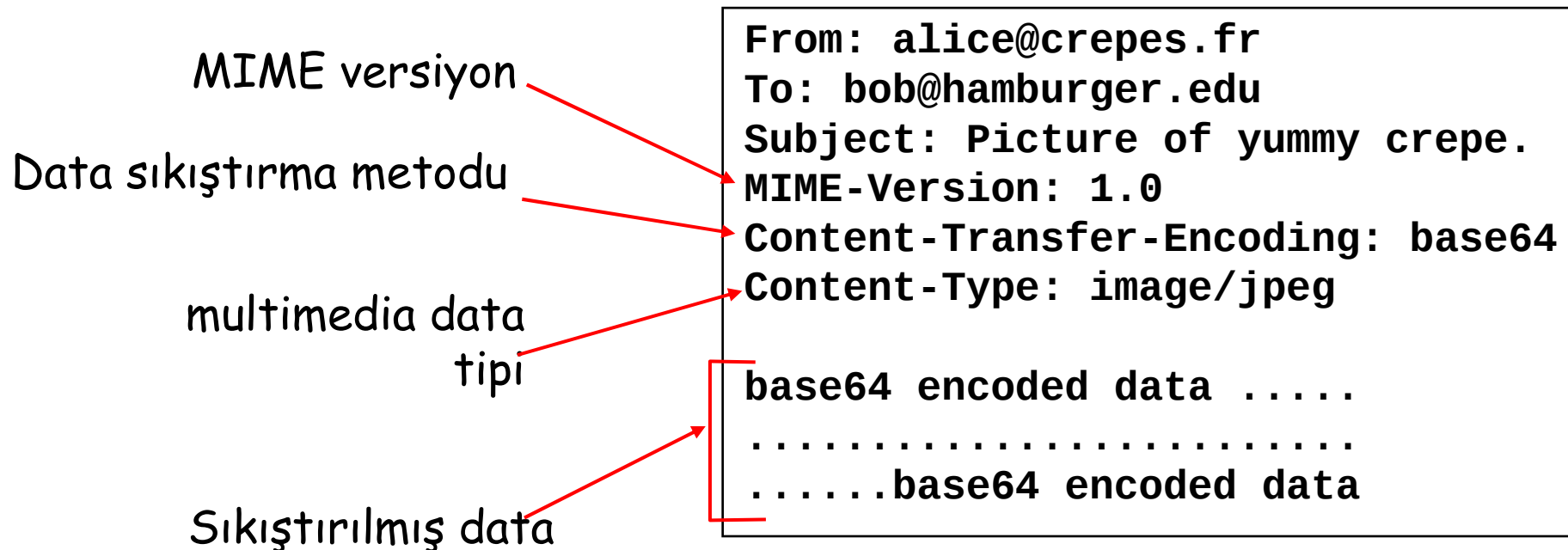
☐ gövde

- Sadece ASCII karakterler

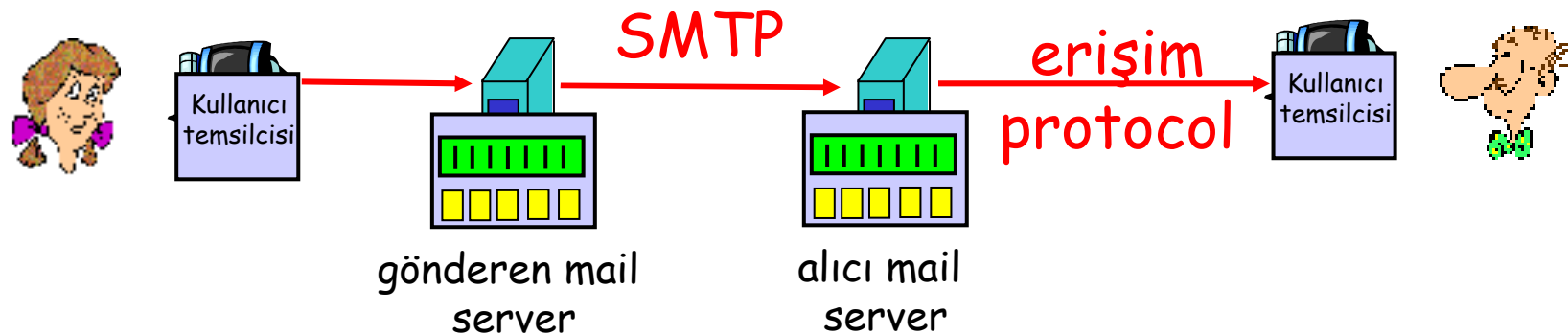


Mesaj formatı: multimedya uzantısı

- ❑ MIME: multimedia mail extension, RFC 2045, 2056
- ❑ Mesaj başlığındaki ekstra satırlar MIME içeriğini belirtir



Posta erişim protokolleri



- ❑ SMTP: alıcı servera iletme ve saklama
- ❑ Mail erişim protokolü: kullanıcının serverdan maili alması
 - POP3: Post Office Protocol [RFC 1939]
 - Kimlik doğrulama (kullanıcı temsilcisi <-->server) ve download
 - IMAP: Internet Mail Access Protocol [RFC 1730]
 - Daha karmaşık
 - Serverda saklanan maillerin işlenmesi
 - HTTP: Hotmail , Yahoo! Mail, vs.

POP3 protocol

Kimlik doğrulama

- ❑ client komutları:
 - user: kullanıcı adı
 - pass: şifre
- ❑ server cevapları:
 - +OK
 - -ERR

işlem fazı, client:

- ❑ list: msj numaralarını listele
- ❑ retr: msjları numaraları ile al
- ❑ dele: sil
- ❑ Quit: çıkış

```
S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on

C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

POP3 ve IMAP

More about POP3

- ❑ Previous example uses "download and delete" mode.
- ❑ Bob cannot re-read e-mail if he changes client
- ❑ "Download-and-keep": copies of messages on different clients
- ❑ POP3 is stateless across sessions

IMAP

- ❑ Tüm mailleri serverda tut
- ❑ Allows user to organize messages in folders
- ❑ IMAP keeps user state across sessions:
 - names of folders and mappings between message IDs and folder name

Bölüm 2:Uygulama Katmanı

- ❑ 2.1 Ağ uygulamalarının prensipleri
- ❑ 2.2 Web & HTTP
- ❑ 2.3 FTP
- ❑ 2.4 Elektronik posta
 - SMTP, POP3, IMAP
- ❑ 2.5 DNS
- ❑ 2.6 P2P Eşler arası uygulamalar
- ❑ 2.7 TCP ile soket programlama
- ❑ 2.8 UDP ile soket programlama

DNS: Domain Name System (internet dizin servisi)

İnsanlar: tanımlayıcılar:

- Tc kimlik no

**İnternetteki hostlar,
routerlar:**

- IP address (32 bit) -
- "isimler", e.g.,
ww.yahoo.com - insanlar
trf kullanılır

Q: Ip adreslerinin isimlere
çevrilmesi ?

Domain Name System:

- *Dağılık veritabanı:* isimler
dağılık isim serverlarında
(name server)
- *Uygulama katmanı protokolü*
 - hostlar, routerlar, name
serverlar adres/isim
çevirmesini yapmak için
haberleşirler
 - not: temel internet
fonksyonu, uygulama
katmanındadır
 - HHTp, SMTP, FTP vs.
DNSi kullanır.

DNS

DNS servisleri

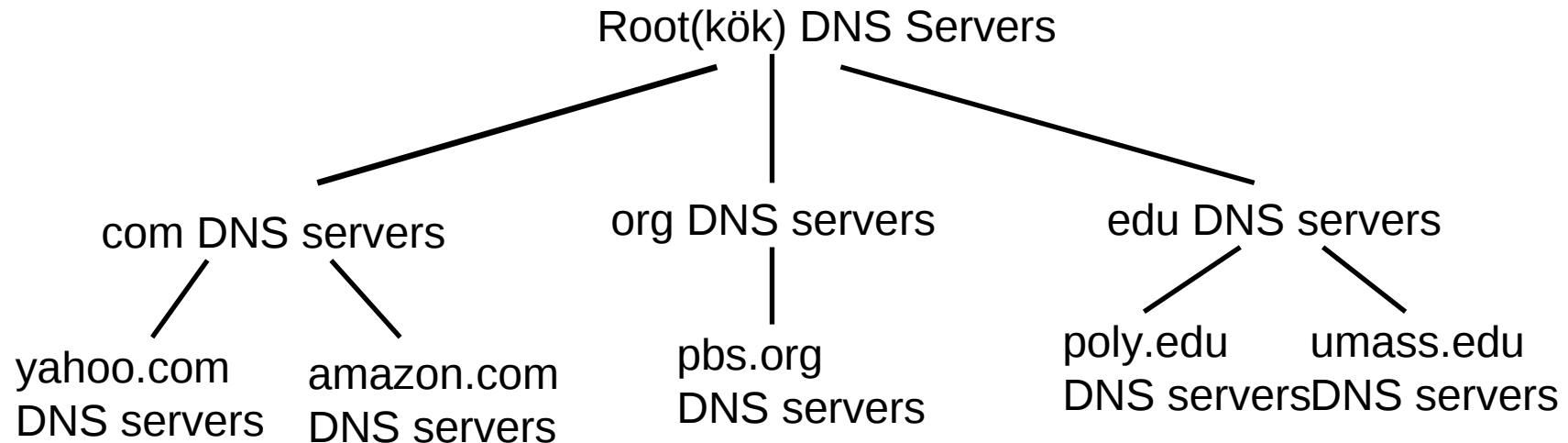
- ❑ Host adı - IP adresi eşlemesi
- ❑ Host lakapları
 - Çift lakap
- ❑ Mail server lakabı
- ❑ Yük dağılımı
 - 1 isim için 1den fazla server (rotasyon)

DNS neden merkezi değil?

- ❑ Çökme tek merkezde olur
- ❑ Trafik hacmi
- ❑ Bazı hostlar için çok uzak olur
- ❑ bakım

ölçeklenemez!

Dağınık, Hiyerarşik Veritabanı

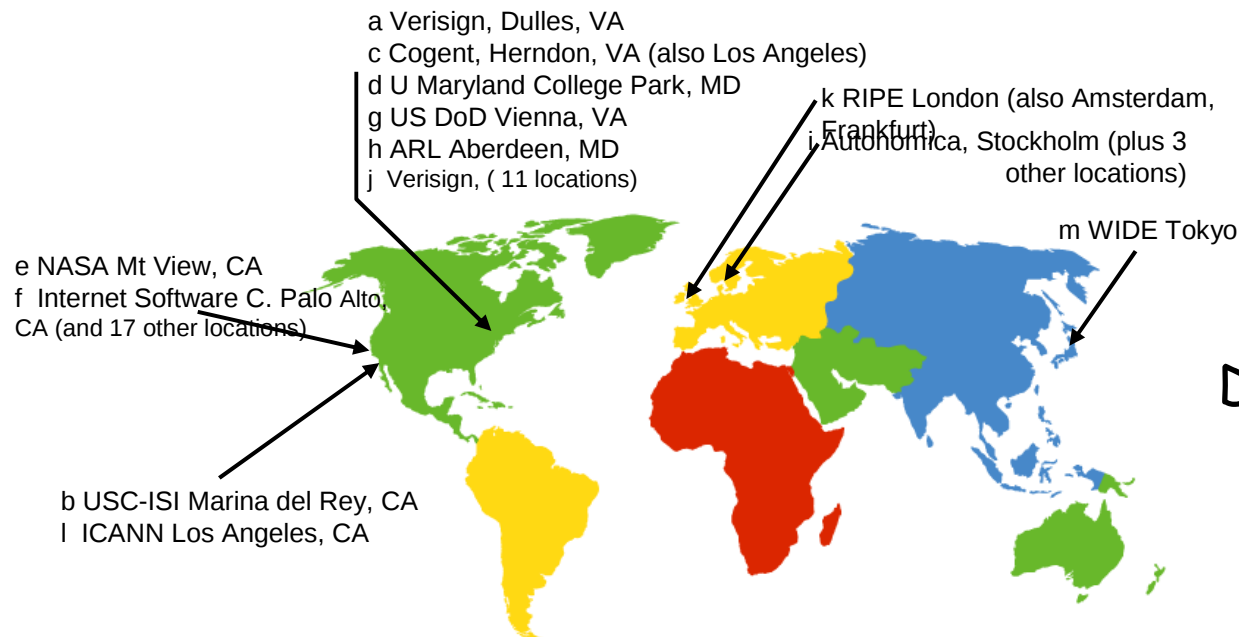


Client www.amazon.com adresinin IP'sini ister; 1. yol:

- ❑ Client root servera com DNS serverını bulma isteğini bildirir
- ❑ Client com DNS serverına amazon.com DNS serverı bulma isteğini bildirir
- ❑ Client amazon.com DNS serverından www.amazon.comun IP adresini ister

DNS: Root name servers (Kök dns sunucuları)

- ❑ İsmi çözemeyen local name serverlar root name serverla kontakt kurarlar
- ❑ root name server:
 - Ip-isim eşlemesi bilinmiyorsa yetkili name serverla görüşür
 - Eşlemeyi alır
 - Eşlemeyi local name servera gönderir



Dünyada 13 tane
root name serveri
var

TLD ve Yetkili Serverlar

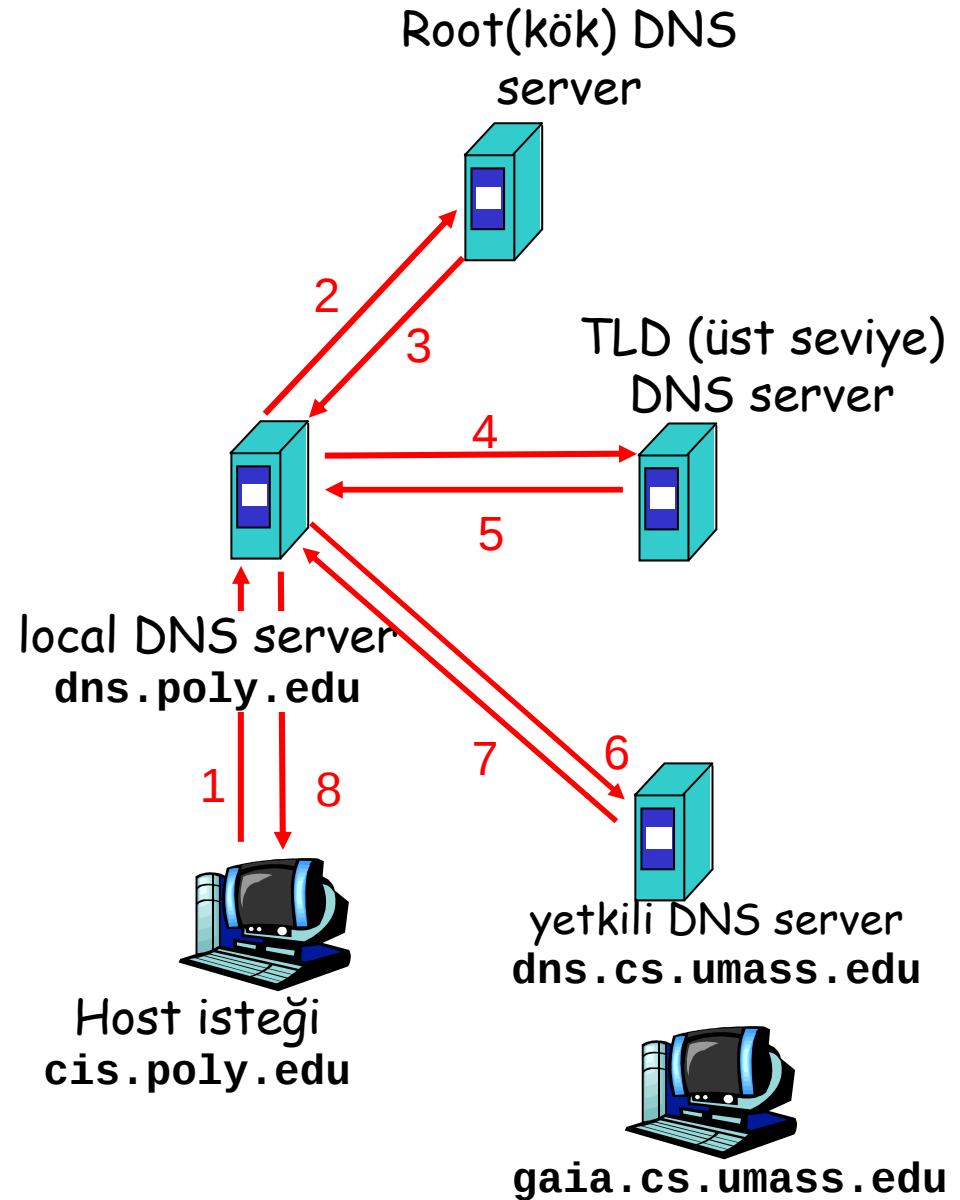
- ❑ **Top-level domain (TLD-üst seviye) serverları:** com, org, net, edu gibi ve ülke domainlerini tutar
 - Network solutions şirketi com TLD
 - Educause şirketi for edu TLD
- ❑ **Yetkili DNS serverları:** kurumun DNS serverları eşlemeleri tutar (örn: Web ve mail).
 - Kurum yada servis sağlayıcısı trf kurulur

Local Name Server

- ❑ Hiyerarşiye fazla bağımlı değil
- ❑ Her ISP (TT, Fatih unv. Vs.) bir local name servera sahip
 - "default name server" olarak da adlandırılır
- ❑ Host DNS sorgusu yaptığı zaman önce local name server sorgulanır
 - Sorgunun cevabı bulunamazsa hiyerarşide bir üst servera sorar

Örnek

- cis.poly.edu adresindeki host gaia.cs.umass.edu nun Ip adresini istiyor



DNS: cache ve güncelleme

- ❑ Server IP-isim eşlemesini öğrenir ve onu cache de saklar
- ❑ Belli bir zaman sonra cachedeki bilgiler silinir
 - Local name serverlarda genellikle TLD serverları cachede tutulur
 - Bu yüzden root serverlardan az istekte bulunulur

DNS kayıtları

DNS: kaynak kayıtlarını (RR) içeren dağıtık veritabanı

RR formatı: (name, value, type, ttl)

□ Type=A

- name hostadıdır
- value IP adrestir

□ Type=NS

- name domain (foo.com) adıdır
- value bu domain için yetkili name serverın Ip adresidir

□ Type=CNAME

- name gerçek addır
www.ibm.com is really
servereast.backup2.ibm.com
- value lakaptır

□ Type=MX

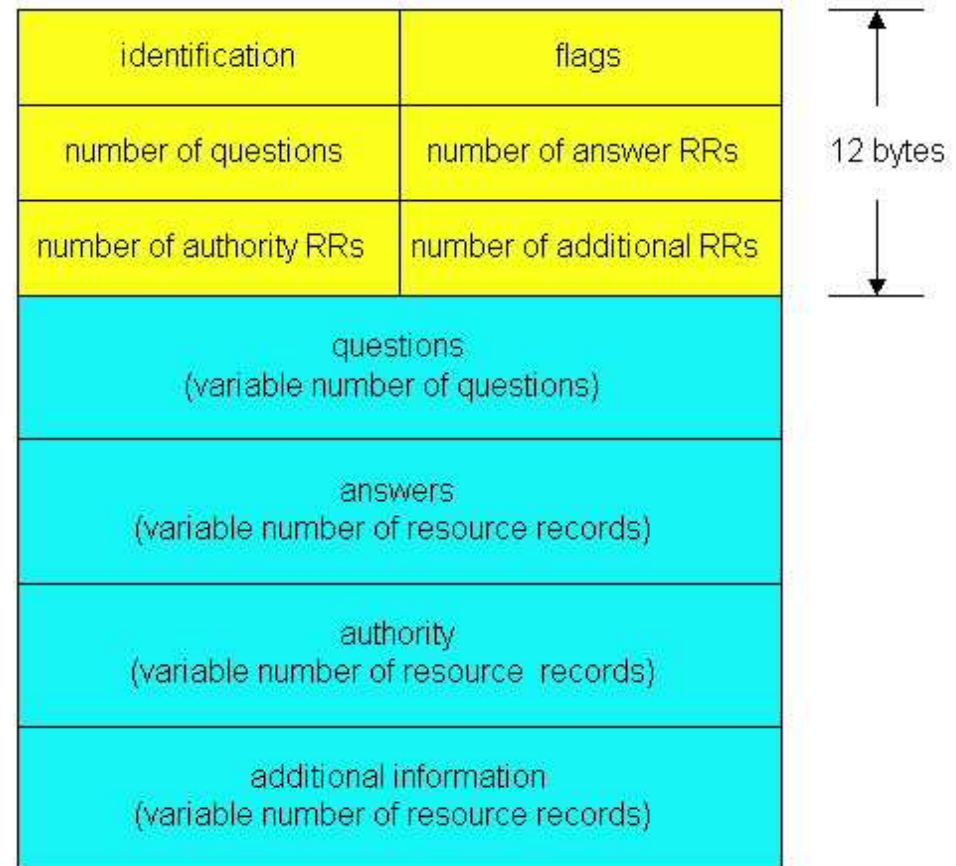
- value namein mail serverının kurallı adıdır

DNS protokol ve mesajlar

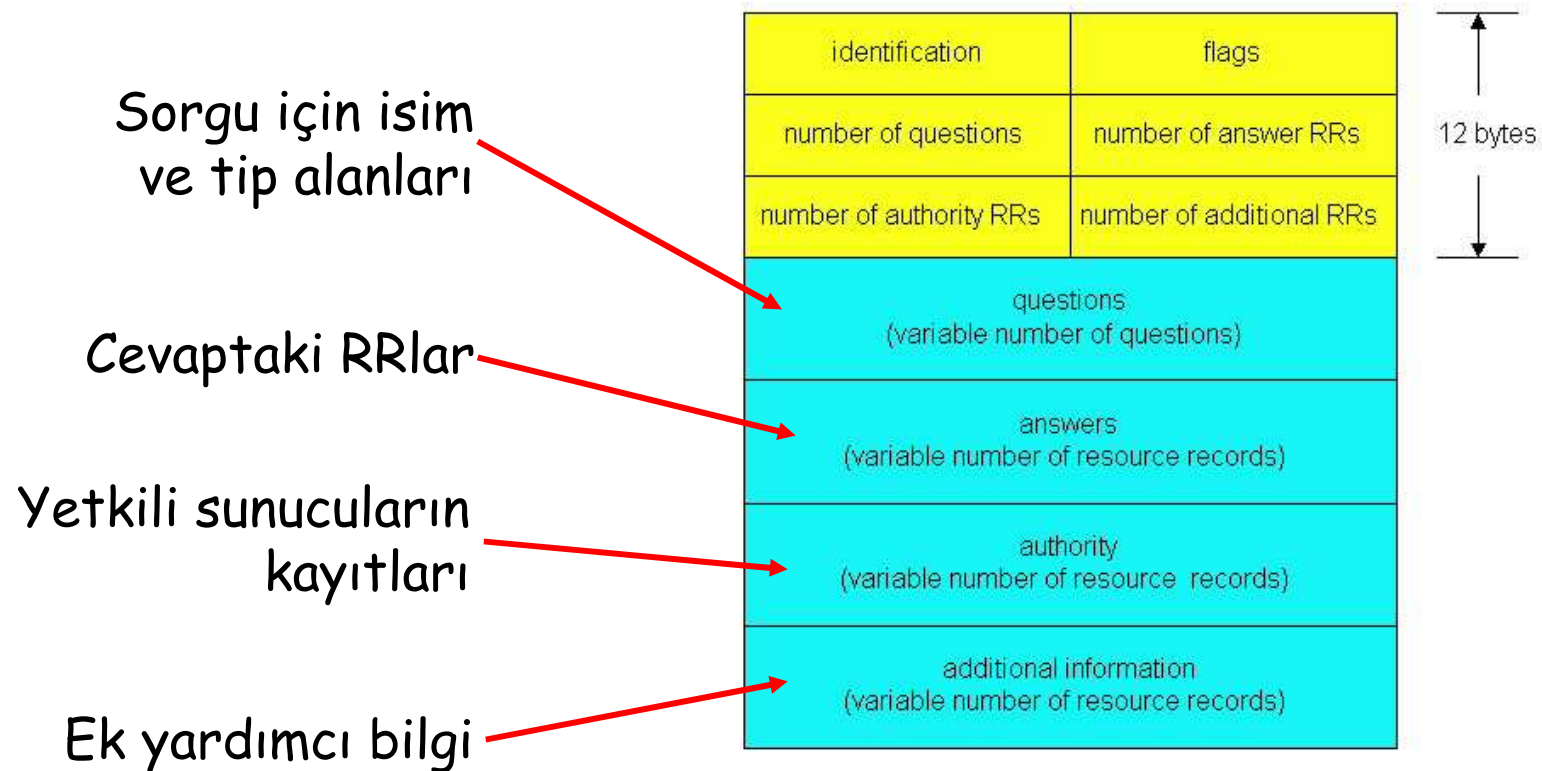
DNS protocol : *istek ve cevaplar aynı formattadır*

Msj başlığı

- **tanımlama:** 16 bit uzunluğunda bir sayı, verilen cevaba da aynı no kopyalanır



DNS protokolü, mesajlar



DNS veritabanına kayıt girmek

- ❑ Örn: "Network Utopia" yı yeniden kurduğunuzu düşünün
- ❑ networkutopia.com için bir kaydedici şirket
- ❑ Yetkili DNS sunucularının Ip adreslerini sunmak gerek
 - Kaydedici trf aşağıdaki bilgiler eklenir:

(networkutopia.com, dns1.networkutopia.com, NS)

(dns1.networkutopia.com, 212.212.212.1, A)

- ❑ A tipi msj www.networkutopia.com için ve
- ❑ MX tip kayıt networkutopia.com mail server için

Chapter 2: Application layer

- ❑ 2.1 Principles of network applications
 - app architectures
 - app requirements
- ❑ 2.2 Web and HTTP
- ❑ 2.4 Electronic Mail
 - SMTP, POP3, IMAP
- ❑ 2.5 DNS
- ❑ 2.6 P2P file sharing
- ❑ 2.7 Socket programming with TCP
- ❑ 2.8 Socket programming with UDP
- ❑ 2.9 Building a Web server

P2P file sharing

Example

- ❑ Alice runs P2P client application on her notebook computer
 - ❑ Intermittently connects to Internet; gets new IP address for each connection
 - ❑ Asks for "Hey Jude"
 - ❑ Application displays other peers that have copy of Hey Jude.
 - ❑ Alice chooses one of the peers, Bob.
 - ❑ File is copied from Bob's PC to Alice's notebook: HTTP
 - ❑ While Alice downloads, other users uploading from Alice.
 - ❑ Alice's peer is both a Web client and a transient Web server.
- All peers are servers = highly scalable!