Uygulama katmanı ve protokolları (Uygulama+Sunum+Oturum)

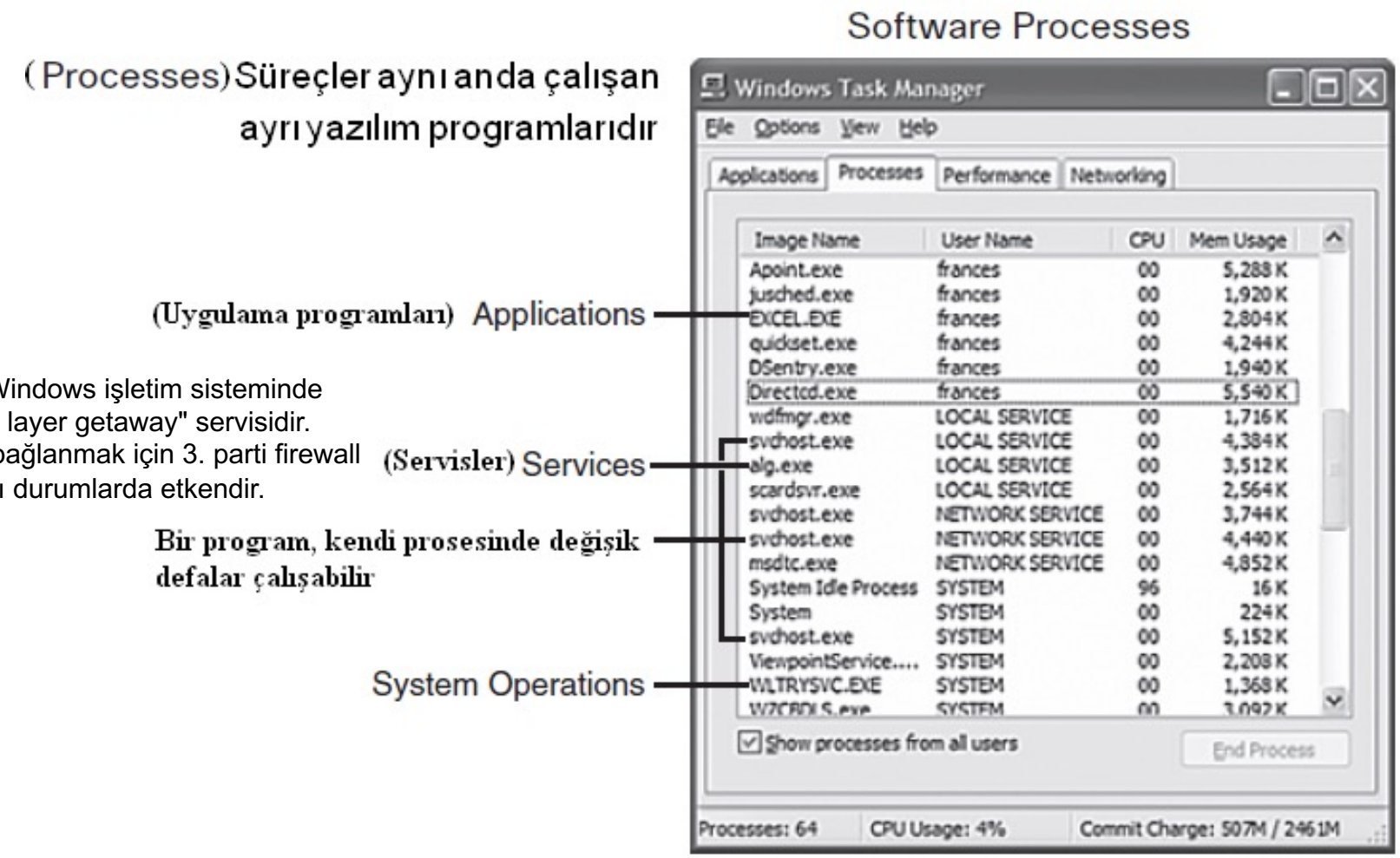
- OSI ve TCP/IP modelindeki uygulama katmanı, networkteki farklı son noktalar arasında (istemci-sunucu veya peer to peer), proseslerin (kullanıcı uygulamalarının) çalışması sürecinde yapılması gerekenlerin tanımlandığı ve uygulandığı yerdir.
- Uygulama katmanındaki, ***uygulama programları***, ***servisler*** ve ***protokollar*** "insan - bilgisayar ağı" arasında bir arayüz sağlamak içindir.

  Uygulama katmanı protokollarının genel fonksiyonları:

  - Mesaj türleri, örneğin talep mesajları ve cevap mesajları.
  - Çeşitli mesaj türlerinin sözdizimi, yani mesajdaki alanlar ve Alanlar nasıl tanımlanacağı
  - Alanların semantiği, yani bilginin anlamlandırılacağı alanı içermesi
  - Bir sürecin ne zaman ve nasıl mesaj göndereceğini belirleyen kurallar ve mesajlara cevap verir

- Bir web tarayıcısı veya bir e-mail penceresini açtığınızda, bir uygulama programı başlatılmış olur. (program hafızasına yüklenen bu program bir executable programdır).
- Hafızaya yüklenmiş herbir uygulama programı (executable program) bir ***prosess*** olarak isimlendirilir.

Uygulama katmanı içinde, ağa erişim sağlayan yazılım programları veya prosesler (süreçler) iki formda bulunur: **Uygulamalar ve Servisler**



alg.exe: Windows işletim sisteminde "aplication layer getaway" servisidir. İnternete bağlanmak için 3. parti firewall kullanıldığı durumlarda etkendir.

Network-Aware Applications (Ağı tanıyan-haberdar - uygulamalar)

Ağı tanıyabilen-farkında olabilen - (Network aware application) bazı son-kullanıcı uygulama programları, uygulama katmanı protokollarını uygulayarak doğrudan doğruya düşük katmanlı protokollarla iletişim kurabilir. E-mail client programları veya web browserler bunlara en iyi örneklerdir.

Application Layer Services
(Uygulama katmanı Servisleri)

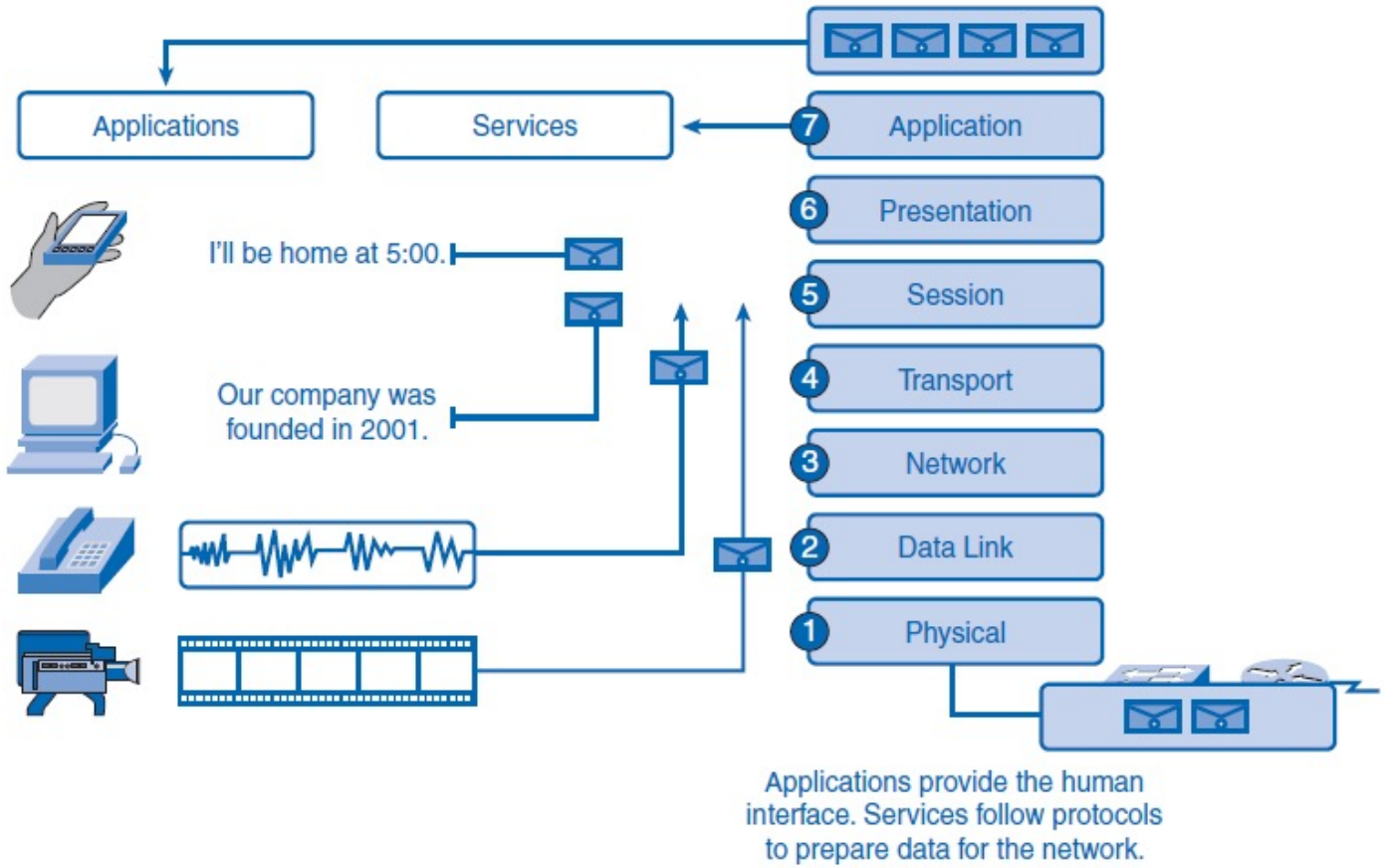
Dosya transferi veya ağ'da yazdırma sıralama işlemlerini sağlayan v.b diğer bazı programlar, ağ kaynaklarını kullanmak için uygulama katmanı servislerine ihtiyaç duyar. Bu servisler, ağ ile kullanıcı arasında bir arayüz oluşturur ve veriyi taransfer için hazırlar. Farklı tip veriler (Text, resim, video v.b) alt katmanlarda işlenebilmesi için farklı servislere (hizmetlere) ihtiyaç duyar.

Her uygulama yazılımı veya ağ servisi, kullanılacak standartları ve veri biçimlerini tanımlamak için protokolleri kullanır. **Bir servis, tanımlanmış bir şeyi yapmak üzere sağlanan bir fonksiyondur. Bir protokol ise; Servis kullanımı kurallarını sağlar.** Çeşitli ağ servislerinin fonksiyonlarını anlamak için, onların işleyişini düzenleyen temel protokolleri bilmek gerekir.

User Applications, Services, and Application Layer Protocols

- Uygulama katmanı kullanıcı (Client) uygulamaları ve servislerini gerçekleştirmek için belirlenmiş standartlara protokol denir.
- Uygulama programları , kullanıcıya; mesajları oluşturmak için bir yol , araç sunar.
- uygulama katmanı servisleri, ağa bir arayüz oluşturmak için vardır.
- Protokolleri ise bu işlemlerin nasıl yapılacağını yöneten kuralları ve formatları sağlamak içindir.
- Bir tek process, bu üç komponenetin hepsini birlikte kullanabilir. Örneğin "Telnet", bir uygulamadır, bir servistir, bir protokoldur.
- **Uygulama Katmanı Protokolleri (Uygulama katmanı için tanımlı olan prokoller), bir üst katmanda bulunan işletim sisteminin kullanıcıya sunduğu program arayüzlerine (web tarayıcı, e-mail gönderici v.b) hizmet verir. Kullanıcıya hizmet veren programın türüne göre uygulama katmanında farklı protokoller çalıştırılır. SMTP, http, SNMP V.b**

Figure 3-5 Interfacing Human and Data Networks



Uygulama Katmanı (Application Layer)

- Uygulama katmanı; kullanıcılar tarafından sıkça kullanılan protokolleri içerir. Örneğin WWW'e erişimi sağlayan HTTP (HyperText Transfer Protocol) bunlardan birisidir. Bir tarayıcı (browser) bir web sayfasını görüntülemek istediğinde sunucuya istediği sayfanın ismini gönderir. Sunucu da cevap olarak o sayfayı geri döndürür.

- Uygulama katmanında 2 önemli fonksiyonu yerine getirmek için yapılması gerekenler açıklanır.

1-Çok değişik uç birimlerin (farklı editör kullanan farklı ekran düzenleri, metin yazma ve silme sistemleri farklı olan) tanınmasının sağlanması. Bunun için bir SANAL AĞ UÇ BİRİMİ oluşturulur. İşte tüm uç birimlerinin tanıyabileceği bu sanal ağ uç birimi oluşturma işlemi protokolları doğurur. Tüm sanal terminal (Uç birimi) yazılımları uygulama katmanında belirlenmiştir.

2-Bu katmanın diğer bir görevi ise uç birimler arasındaki dosya transferinin sağlanmasıdır. Farklı dosya sistemleri, farklı adlandırma v.b değişik özellikler gösterebilir. İşte bu farklı sistemler arasındaki dosya transferinin sağlanması için gerekli protokoller, (e-mail v.b) bu katmanda tanımlı görevleri yapmak içindir.

Uygulama katmanı protokolları

- **Bu protokollar (SMTP, TELNET, HTTP v.b) bir üstte çalışan kullanıcı programlarına hizmet verirler. Uygulama katmanı protokollarının herbiri, biri kullanıcı (Client- hizmet alan) diğeri sunucu (server- hizmet veren) da çalışmak üzere yapılandırılır.**
- **Web Browser, E-mail, Print Services, SIP, SSH and SCP, NFS, RTSP, Feed, XMPP, Whois, SMB; DNS; FTP; TFTP; BOOTP; SNMP;RLOGIN; SMTP; MIME; NFS; FINGER; TELNET; NCP; APPC; AFP; SMB**

- **SMTP (Simple mail transport protocol):** Ağ içerisindeki kullanıcılar arasındaki e-mail alışveriş kurallarını düzenler.
- **SNMP(Simple network managment protocol):** Ağ içerisindeki ağ aktif cihazlarının yönetimi için kullanılan protokol.
- **TELNET :** Uzak bağlantı şeklidir. Sistem üzerindeki bir kullanıcının başka bir sisteme bağlanarak onun terminali gibi o sistemin kullanılmasını sağlar.
- **FTP (File Transfer Protocol):** Bir bilgisayardan başka bilgisayara dosya aktarımı için kullanılan protokol
- **HTTP (hyper Text Transfer Protocol):** WEB sahifelerinin alış-verişini sağlayan protokoldur.
- **DNS(Domain Name Server):** İnternet isimlerini IP noya çeviren protokoldur.

Sunuş Katmanı (Presentation layer)

- Alttaki katmanların bitlerle ve veri paketleriyle ilgilenen yapısının tersine sunuş katmanı gönderilen bilginin sözdizimi ve anlambilimsel yapısıyla (semantics) ilgilenir. Yani alt katmandan gelen verileri bilgi haline dönüştürür.
- Değişik veri yapılarına sahip bilgisayarlar arasındaki bağlantıyı sağlamak için soyut veri yapıları tanımlamak gerekebilir. Sunuş katmanı bu soyut veri yapılarını idare eder ve üst-seviye veri yapılarının (örn. banka kayıtları) tanımlanmasını ve bilgisayarlar arasında alışverişine izin verir.
- Farklı bilgisayarlar, karakterli farklı kodlamalarla kullanıyor olabilirler. Bu farklı gösterime sahip bilgisayarların iletişimini mümkün kılmak için iletişim standart kodlamayla yapılır ve gideceği yerde ise kendi kodlamasına dönüşüm yapılır (örneğin bir taraf ASCII diğer taraf UnICODE kullanabilir).
- Ayrıca veri sıkıştırması, kriptografi v.b işlemler bu katmanda yapılır.

- SMB, AFP, NCP, MIDI, HTML, GIF, TIFF, JPEG, ASCII,
  EBCXDR, ASN.1DIC

Oturum Katmanı (Session Layer)

- Farklı bilgisayarlardaki kullanıcıların biribirleri üzerinde oturum açması hizmetini düzenler.
- Trafiğin tek veya çift yönlü olmasını düzenler.
- İletişimin senkronizasyonunu sağlar. **Yani** bir iletişimin kopmasından sonra iki tarafın kaldıkları yerden iletişime devam edebilmeleri için bir sağlama noktası (checkpoint) kullanma), vs.

- TLS, SSH, X.225, RPC, NetBIOS, ASP, Winsock, BSD

- TCP-IP modelindeki uygulama kartmanı da OSI’nin ilk üç katmanının görevlerini tarif eder.

Bazı Ağ Uygulamaları

- E-posta
- Web
- Instant messaging
- Remote login
- P2P dosya paylaşımı
- Çok kullanıcılı ağ oyunları
- Streaming
- Internet telefon
- Real-time video konferans
- Paralel işlem

Ağ Uygulaması oluşturma

Yazılan programlar

- Farklı uç sistemlerde çalışır
- Ağ üzerinden haberleşir
- örnek, Web: Web server yazılımı browser yazılımı ile haberleşir

Ağ temel elemanlarına yönelik yazılım yapılmaz

- Network core cihazlar application layer'da çalışmaz
- Bu tasarım hızlı uygulama geliştirmeye izin verir

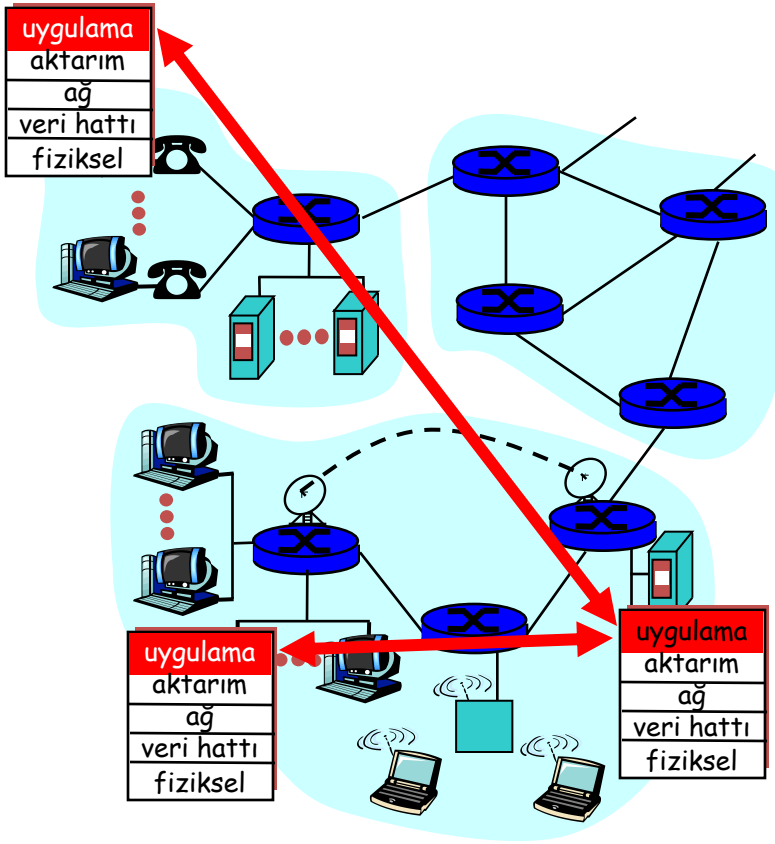
Uygulamalar ve uygulama tabanlı protokoller

Uygulama: haberleşen, dağılmış süreçler'dir.

- "kullanıcı-Client" olarak bilgisayar ağlarında çalışırlar
- uygulamaları yerine getirmek üzere mesaj değişimi
- ör: e-posta, ftp, Web

Uygulama katmanı protokolları

- uygulama katmanının bir parçasıdırlar
- alt katman protokollarınca sağlanan haberleşme servislerini kullanırlar (TCP,UDP)



Uygulama Mimarileri

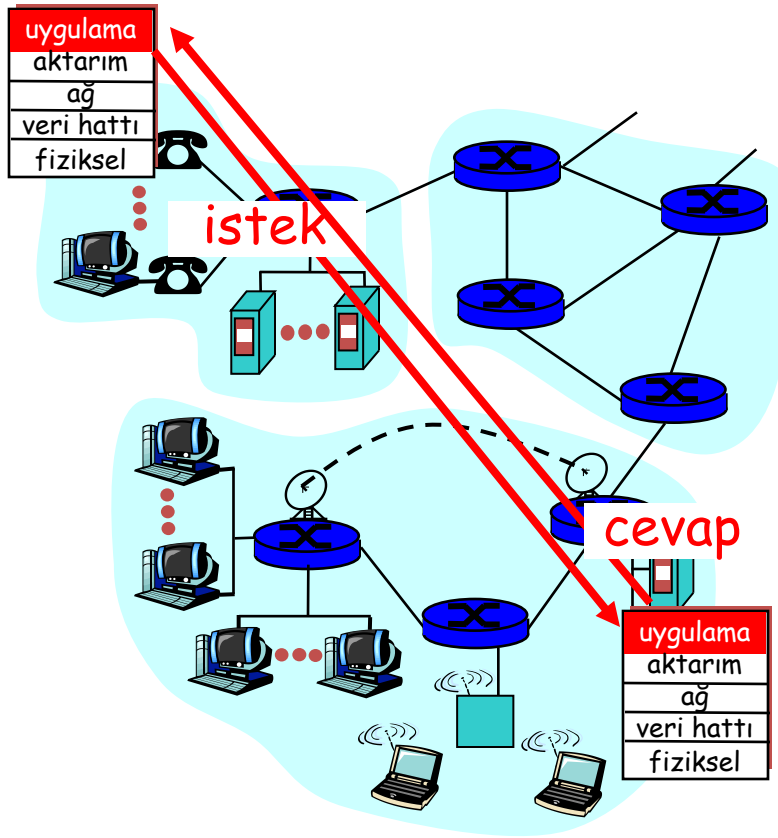
- Client-Server (İstemci –Sunucu)
- Peer-To-Peer (Eş düzey)
- Hibrid (C-S, P2P)

Kullanıcı-Sunucu

Ağ uygulamaları iki kısımdan oluşabilir: kullanıcı(Client) ve servis sağlayıcı (sunucu - server)

Kullanıcı (İstemci-client):

- Sunucu ile ilk irtibatı kurar
  ("ilk konuşan")
- Sunucudan bir servis, hizmet ister,
- İsemcinin IP adresi dinamik olabilir.
- Biribirine doğrudan bağlı değil
- Ör:, WWW sayfası, e-posta gönderme



Sunucu (Server):

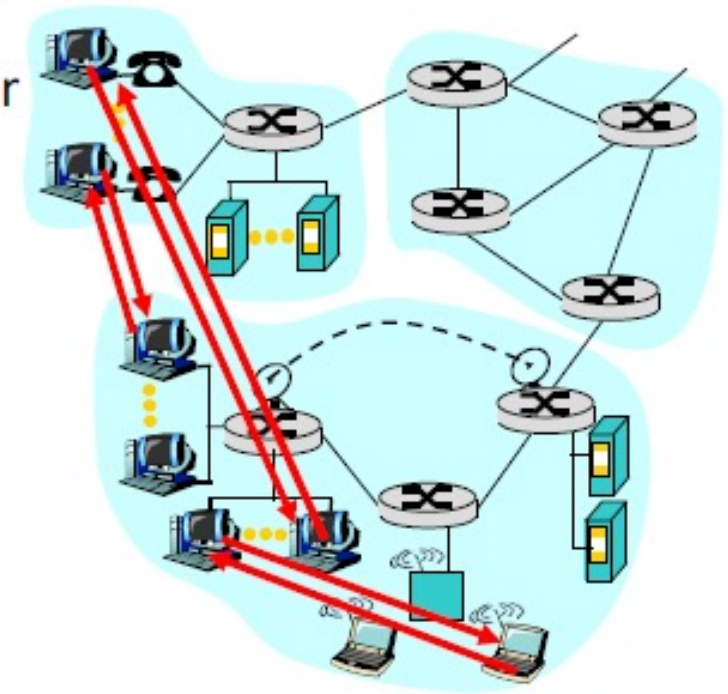
- Kullanıcıya istediği servisi sağlar
- Her zaman açık
- Sunucunun IP adresi sabittir.
- Ölçekleme için çok sayıda sunucu
- Ör:,istenilen WWW sayfasını gönderir, alınan e-postayı alır, saklar

P2P mimari

- Her zaman açık sunucu yoktur
- Uç sistemler doğrudan bağlanır
- Uç sistemler aralıklarla doğrudan bağlanabilir ve IP adres değiştirebilir

Yüksek ölçeklenirdir

Yönetim zordur



Hibrid (C-S, P2P)

Napster

- Dosya transferi P2P
- Dosya arama merkezi:
  - Merkezi sunucuya uç birimler kayıt olur
  - Uç birimler içerik aramayı merkezi sunucuda yapar

Instant messaging

- İki kullanıcı arasında chat P2P yapılır
- Açık olup olmadığı denetimi merkezi:
  - Kullanıcı online olduğunda merkezi sunucuya IP adresi kayıt edilir
  - Kullanıcılar IP adres arayacaklarında merkezi sunucuyla iletişime geçerler

Uygulama oluşturma Süreci
İşlemlerin İletişimi (Process comunications)

Process: host üzerinde çalışan program.

- Aynı host üzerinde, iki process inter-process communication ile haberleşir (OS tanımlar).
- Farklı host'lardaki process'ler mesajlarla haberleşir

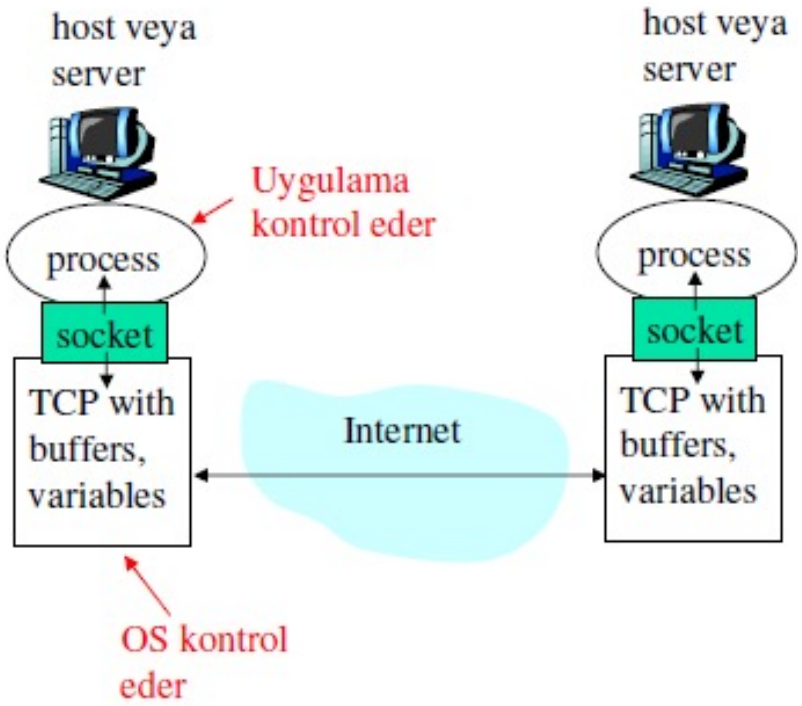
Client process: iletişimi başlatan process

Server process: iletişim başvurusu için bekleyen process

- P2P uygulama mimarileri client ve server işlemlerine sahiptir

Soketler

- Process'ler kendi soketlerine mesaj gönderir veya alır
- soketler kapılara benzer
  - Gönderici process mesajı kapıdan dışarı gönderir
  - Gönderici process kapının diğer tarafındaki transport altyapısına güvenir



- API: (1) transport protokol seçer; (2) Parametre belirler

Process Adresleme

- Mesajı alan process için bir tanımlayıcı gerekir
- Host 32-bit IP adrese sahiptir
- Çok sayıda process aynı host üzerinde açlıştığı için IP adres tanımlayıcı olamaz

- Tanımlayıcı hem IP adresini hemde port numarasını bir process'le ilişkilendirir.
- Örnek port numaraları:
  - HTTP server: 80
  - Mail server: 25

Uygulama katmanı protokolleri (devam)

API: uygulama program arayüzü (application programming interface)

- Uygulama ve aktarım (transport) katmanı arasında arabirimleri tanımlar
- soket: Internet API
  - İki süreç sokete veri göndererek ve soketten veri okuyarak haberleşirler

Soru: haberleşen süreçler birbirlerini nasıl "tanıyabilirler" ?

- IP adres: başka süreçleri de çalıştırabilen ana sistemin (host) IP adresi
- "port numarası" – ana sistemlere gelen mesajların hangi lokal süreçlere gönderilmesi gerektiğini belirlemesinde yardımcı olur

Sürec'in (Process) ihtiyaç duyduğu aktarım (ulaşım) servisi nedir?

Bilgi kaybı

- bazı uygulamalar (ör., audio) kayıplara karşı çok hassas değildir
- diğer uygulamalar (ör., dosya aktarımı, telnet) %100 güvenilir bilgi aktarımı gerektirir

Zamanlama

- bazı uygulamalar (ör., Internet telephony, interaktif oyunlar) efektif olabilmek için küçük gecikmelere ihtiyaç duyarlar

Band genişliği

- bazı uygulamalar (ör., multimedya) en az miktarda band genişliği gerektirir
- diğer uygulamalar ("elastik uygulamalar") ne kadar band genişliği mevcut ise o kadar kullanabilirler

Bazı genel uygulamaların aktarım(transport) servisi gereklilikleri

Uygulama	Bilgi Kaybı	Band genişliği	Zamana duyarlılık
Dosya aktarımı	kayıpsız	elastik	yok
e-posta	kayıpsız	elastik	yok
Web dökümanları	tolerans	elastik	yok
Gerçek zaman audio/video	tolerans	audio: 5Kb-1Mb video:10Kb-5Mb	var, ~100 msec
kaydedilen audio/video	tolerans	yukarıda ki gibi	var, birkaç sec
interaktif oyunlar	tolerans	~ Kbps daha fazla	var, ~100 msec
Finansal uygulamalar	kayıpsız	elastik	Var ve yok

Internet aktarım protokol servisleri

TCP servisi:

- *Bağlantı yönlendirilmiş:* kullanıcı ve sunucu arasında bağlantı kurulması gereklidir
- *Güvenilir aktarım:* gönderen ve alan süreçler arasında güvenilir aktarım
- *akış kontrolu*
- *tıkanıklık kontrolu*
- *sağlamadıkları:* zamanlama, gecikme, minimum (en az) aktarım hızı, band genişliği

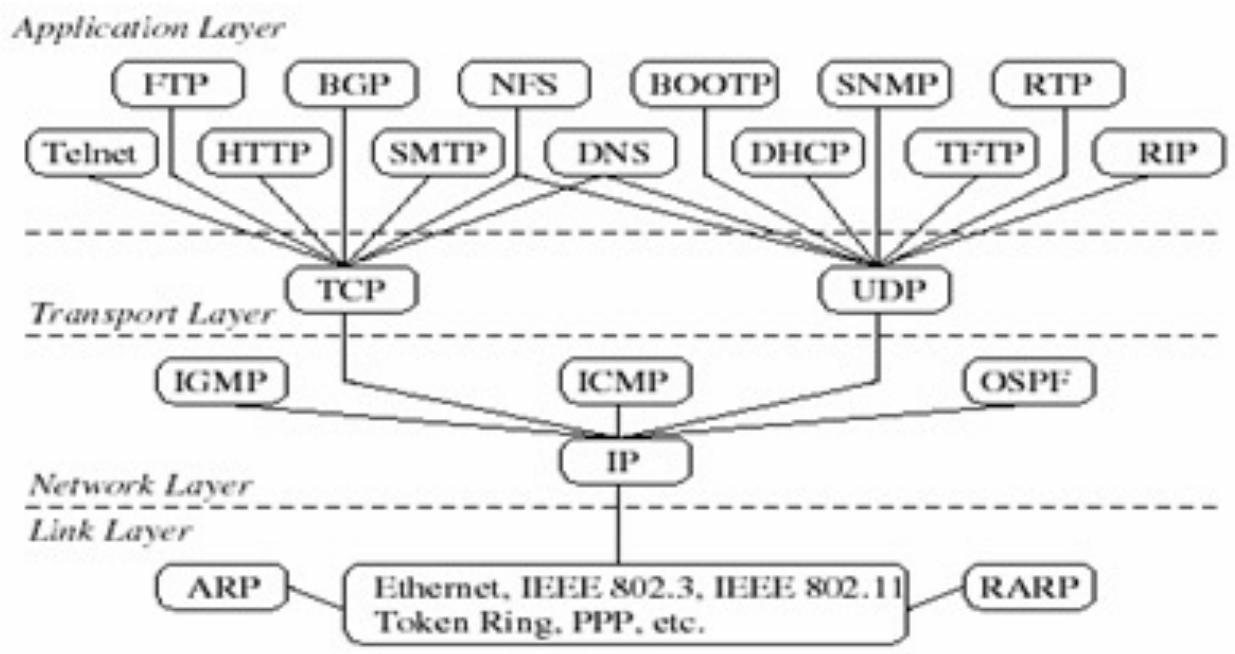
UDP servisi:

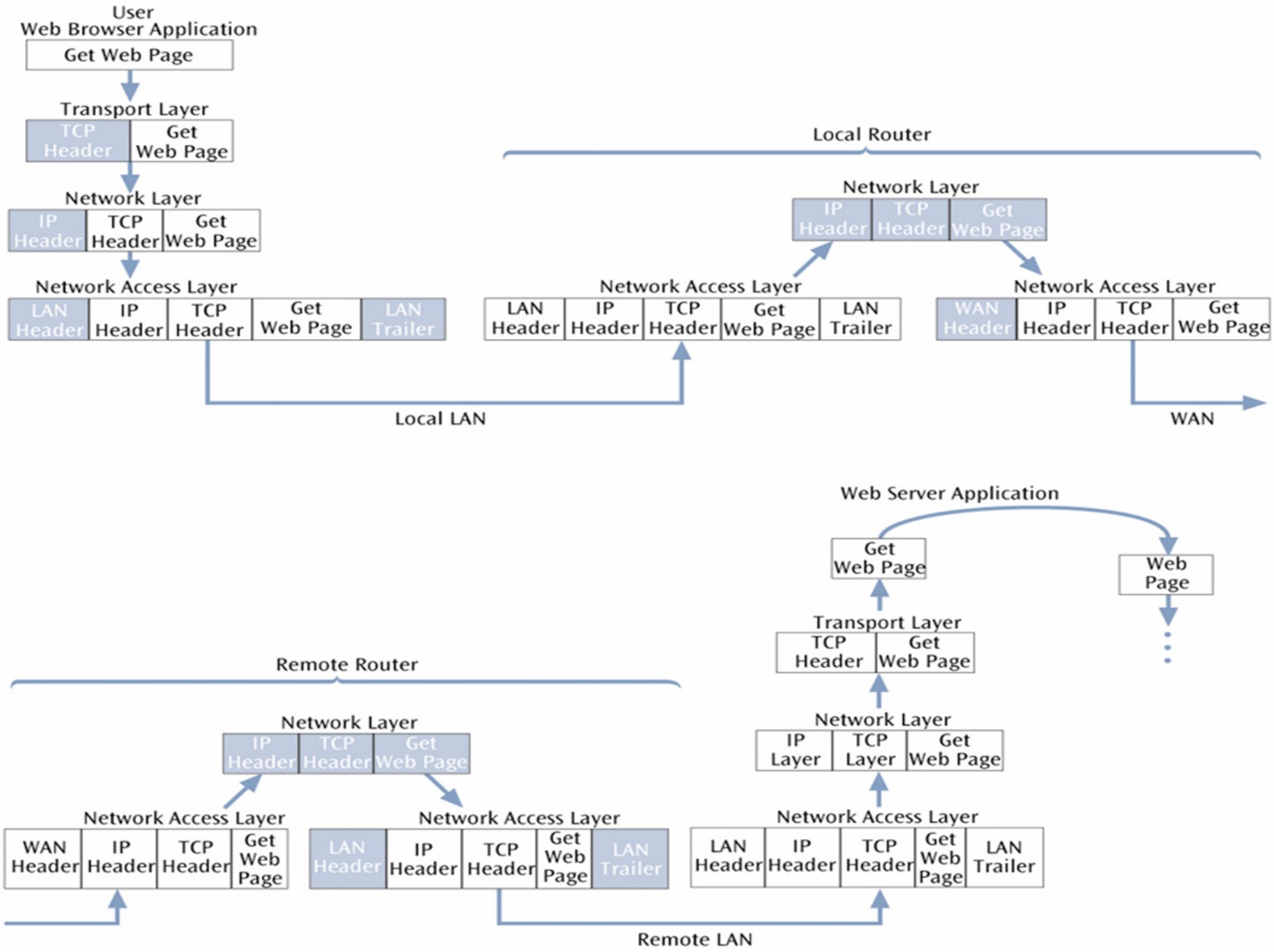
- Alıcı ve gönderen süreçler arasında güvenilir olmayan bilgi aktarımı
- *sağlamadıkları:* güvenilirlik, akış kontrolu, tıkanıklık kontrolu, zamanlama, veya band genişliği

Internet uygulamaları: uygulama, aktarım protokolları RFC(Request of Comments)

Uygulama	Uygulama katmanı protokolu	Aktarım (transport) protokolu
e-posta	smtp [RFC 821]	TCP
uzak terminal erişim	telnet [RFC 854]	TCP
Web	http [RFC 2068]	TCP
Dosya aktarımı	ftp [RFC 959]	TCP
streaming multimedia	özel (ör., RealNetworks)	TCP veya UDP
uzak dosya servis sağlayıcı	NSF	TCP veya UDP
Internet telefon	özel (ör., Vocaltec)	UDP

Protocols in Different Layers



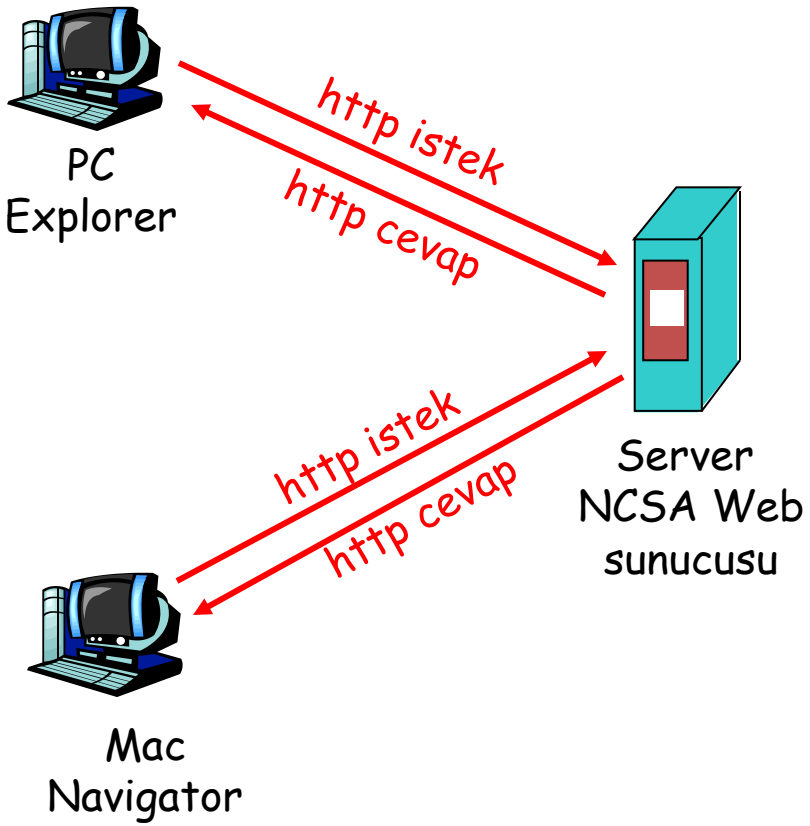

User
Web Browser Application
Get Web Page
Transport Layer
TCP Header
Get Web Page
Network Layer
IP Header
TCP Header
Get Web Page
Network Access Layer
LAN Header
IP Header
TCP Header
Get Web Page
LAN Trailer
Local Router
Network Layer
IP Header
TCP Header
Get Web Page
Network Access Layer
LAN Header
IP Header
TCP Header
Get Web Page
LAN Trailer
Network Access Layer
WAN Header
IP Header
TCP Header
Get Web Page
Local LAN
WAN
Web Server Application
Get Web Page
Web Page
Transport Layer
TCP Header
Get Web Page
Remote Router
Network Layer
IP Header
TCP Header
Get Web Page
Network Layer
IP Layer
TCP Layer
Get Web Page
Network Access Layer
WAN Header
IP Header
TCP Header
Get Web Page
Network Access Layer
LAN Header
IP Header
TCP Header
Get Web Page
LAN Trailer
Network Access Layer
LAN Header
IP Header
TCP Header
Get Web Page
LAN Trailer
Remote LAN

Web: http protokolu

http: hypertext aktarım protokolu

- Web uygulama katmanı protokolu
- kullanıcı/sunucu modeli
- *Kullanıcı:* WWW nesnelerini isteyen, alan ve gösteren "browse" tarayıcı
- *Sunucu:* Web sunucusu isteklere karşılık olarak nesneleri gönderir.

- http1.0: RFC 1945
- http1.1: RFC 2068



http protokolu: (devam)

- http: TCP aktarım servisi:
- kullanıcı, sunucu ile port 80 üzerinden TCP bağlantısını (soket oluşturur) kurar
- sunucu kullanıcının TCP bağlantısını kabul eder
- tarayıcılar arasında (http kullanıcı) ve WWW sunucu (http servis sağlayıcı) arasında http mesajları (uygulama katmanı protokol mesajları) değiştirilir
- TCP bağlantısı kapatılır

- http önceki bağlantılardaki durumları gözönüne almaz
- sunucu daha önceki kullanıcı istekleri hakkında bilgi saklamaz

farkli olarak

- daha önceki durumları gözönünde bulunduran protokoller karmaşıktır!
- geçmiş (durumlar) muhafaza edilmelidir
- sunucu/kullanıcı bağlantısı kopar ise son bağlantı durumları tutarsız olabilir ve yeniden oluşturulmalıdır

URL(Uniform Resource Locators) Kavramı

İnternet üzerindeki sunucu bilgisayarlarda milyonlarca web sayfası, milyonlarca dosyalara nasıl ulaşılacak? Nerde olduğunu bilmediimiz bir sunucudaki web sayfasına nasıl ulaşıyoruz?
Web, web sayfalarını ve diğer kaynakları tanımlamak için URL (Uniform Resource Locators) adında bir şema kullanır. Bir URL şemasında neler bulunur?

http://www.mbe.com.tr/mbe/yapı.html

Bu URL'de bizi World Wide Web birliğindeki bir web sayfasına götüren kısımlar

- Kullanılan protokol HTTP'dir
- Tam domain adı "www.mbe.com.tr"
- Dizin "mbe"
- Alınacak dosya "yapı.html"

Çoğu zaman yalnızca tam domain ismi kullanılır. Web sunucular domain ismi ile çağırılan web sayfalarında otomatik olarak "index.html, default.html, home.htm, index.htm" sayfalarından hangisi varsayılan olarak belirlenmişse o dosyayı getirir. Bu nedenle çoğu zaman dosya adı yazmadan yalnızca http://www.mbe.com.tr yazmamız yeterli olmaktadır.

http örneği

kullanıcının URL girdiğini varsayalım
www.someSchool.edu/someDepartment/home.index

1a. http kullanıcısı TCP bağlantısını http servis sağlayıcısına (süreç) www.someSchool.edu adresinde başlatır. Port 80 http servis sağlayıcısı için kullanılır.

1b. http sunucusu www.someSchool.edu ana sistemdeki http sunucusu port 80 de TCP bağlantısını kabul eder ve kullanıcıyı bilgilendirir.

2. http kullanıcısı http *istek mesajını* (URL de dahil) TCP bağlantı soketine gönderir

3. http sunucusu istek mesajını alır, *cevap mesajını* istenilen dokuman ile (someDepartment/home.index), sokete yollar

(text, 10 adet jpeg Görüntü içerebilir)

zaman

http örneği (devam)

4. http sunucusu TCP bağlantısını kapatır.

5. http kullanıcısı, html dosyası, ve diğer html dosyalarını içeren cevap mesajını alır. html dosyasını ayırarak bağlantılı 10 adet jpeg nesnelerini bulur

zaman

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

- Devamsiz (surekli olmayan) baglantilar: her TCP baglantisinda bir nesne
  - Bazi tarayicilar *eszamanli* olarak bircok baglanti olusturabilir – bir nesne icin bir baglanti
  - surekli baglantilar: bircok nesne bir TCP baglantisinda aktarilabilinir

HTTP Message Examples

- **Typical Request Message From A Client:**

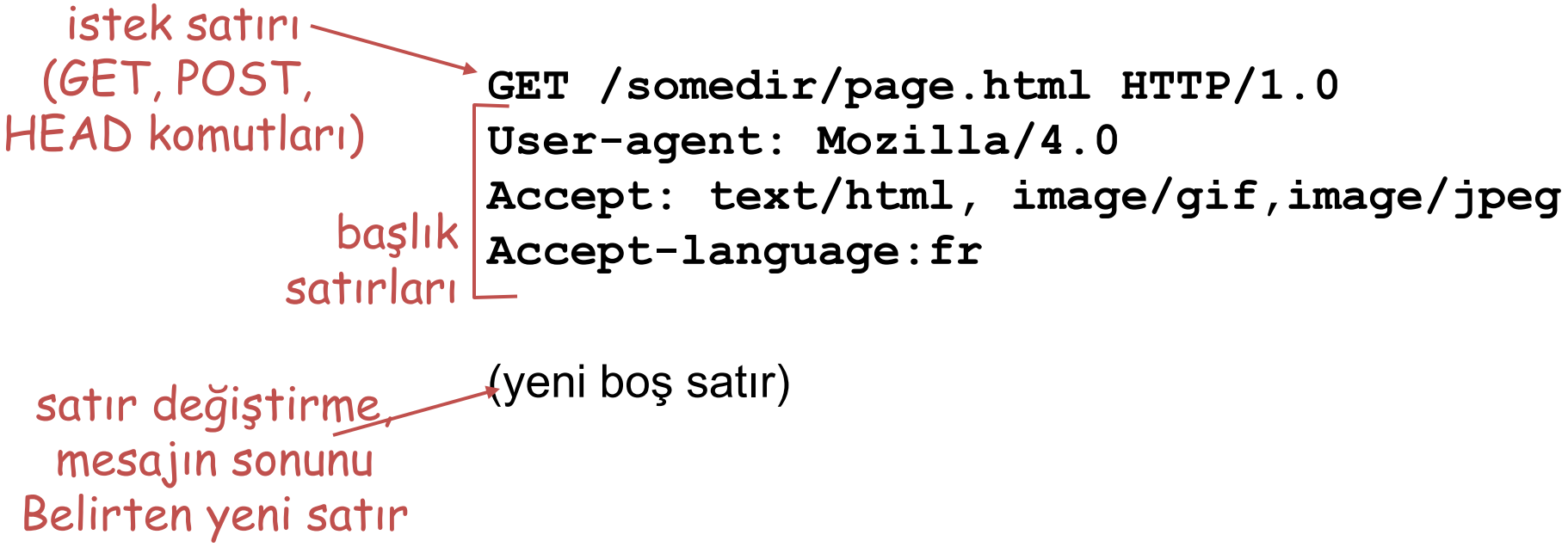
  GET /eecc694-spring2000/index.html HTTP/1.0
  Connection: close
  User-agent: Mozilla/4.72 [en] (Win98; I)
  Accept: text/html, image/gif, image/jpeg
  Accept-language:en
  (extra carriage return, line feed)

- **Typical Response Message From A Server:**

  HTTP/1.0 200 OK
  Connection: close
  Date: Wed, 05 April 2000 12:00:15 GMT
  Server: NCSA/1.5.2
  Last-Modified: Tue, 25 April 2000 11:23:24 GMT
  Content-Length: 20419
  Content-Type: text/html
  data data *and more* data ...

http mesaj formatı: istek

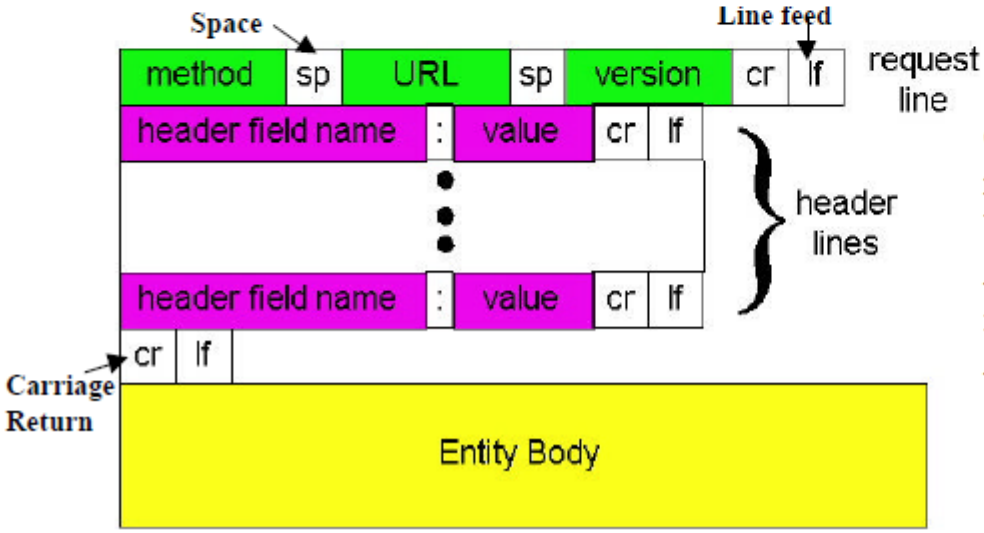
- iki tür http mesajı: *istek*, cevap
- http istek mesajı:
  - ASCII (okunabilir format)



http istek mesajı: genel format

HTTP Message Formats: General Format of A Request Message

- Standart ASCII metninde kodlanmış mesajlar.
- Yöntem: GET, POST ve HEAD. HTTP istek mesajlarının büyük çoğunluğu GET yöntemini kullanır.
- GET yöntemi, tarayıcı, URL 'de tanımlanan nesne ile bir nesne istediğinde kullanılır.
- İstemci kullanıcı bir formu doldurduğunda POST kullanılır.
- URL: TCP bağlantısı zaten sunucuya bağlı olduğundan sunucu ana makine adını eklemenize gerek yoktur.
- Sürüm: Kullanılan HTTP sürüm numarası. (ör. HTTP / 1.0 veya HTTP / 1.1)
- Varlık Gövdesi: GET yönteminde kullanılmaz, POST yöntemine dahil edilen form verileri.



Connection: close, to request non-persistent TCP connections.
User-agent: Browser used.
Accept: type of objects the browser is prepared to accept
Accept-language:

http mesaj formatı: cevap

durum satırı (protokol durum kodu durum cümlesi)

başlık satırları

veri, örneğin, istenilen html dosyası

```
HTTP/1.0 200 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

data data data data data ...
```

Şimdiki örnekte de olmayan bir belge (web sayfası) için yapılan isteğe karşılık gönderilen sunucu cevabıdır.

HTTP/1.1 400 NOT FOUND
Date Wednesday, 28-Feb-07 19:51:28 GMT
Server: Apache/2.0

http cevap durum kodları

Sunucu-> kullanıcı cevap mesajının ilk satırında.

Bazı örnek kodlar:

200 OK

- istek başarılı, istenilen nesne bu mesajın sonrasında

301 Moved Permanently

- istenilen nesne yer değiştirdi, yeni konumu bu mesajın devamında belirtildi (Konum:)

400 Bad Request

- İstek mesajı servis sağlayıcı tarafından anlaşılmadı

404 Not Found

- istenilen doküman bu servis sağlayıcıda bulunamadı

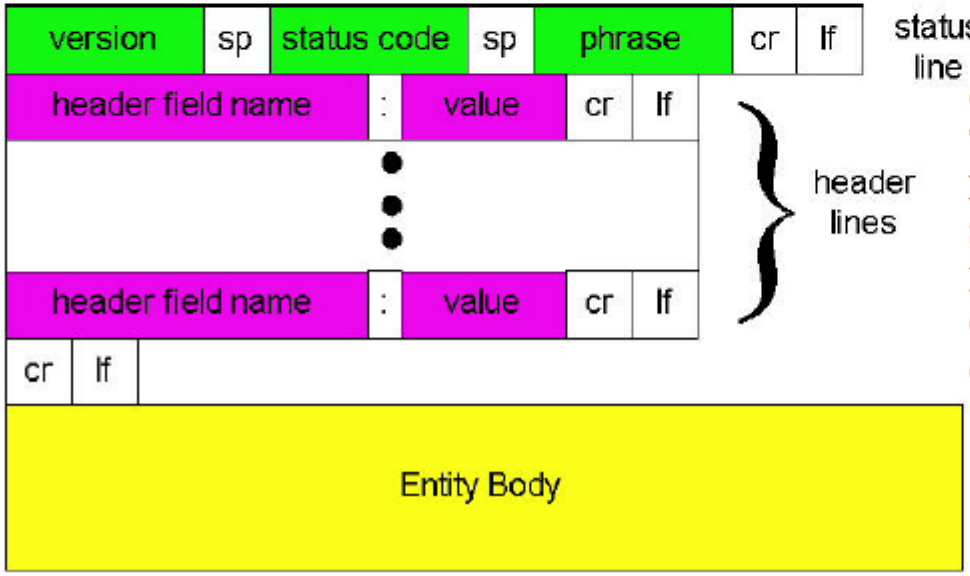
505 HTTP Version Not Supported

HTTP Message Formats:
General Format of A Response Message

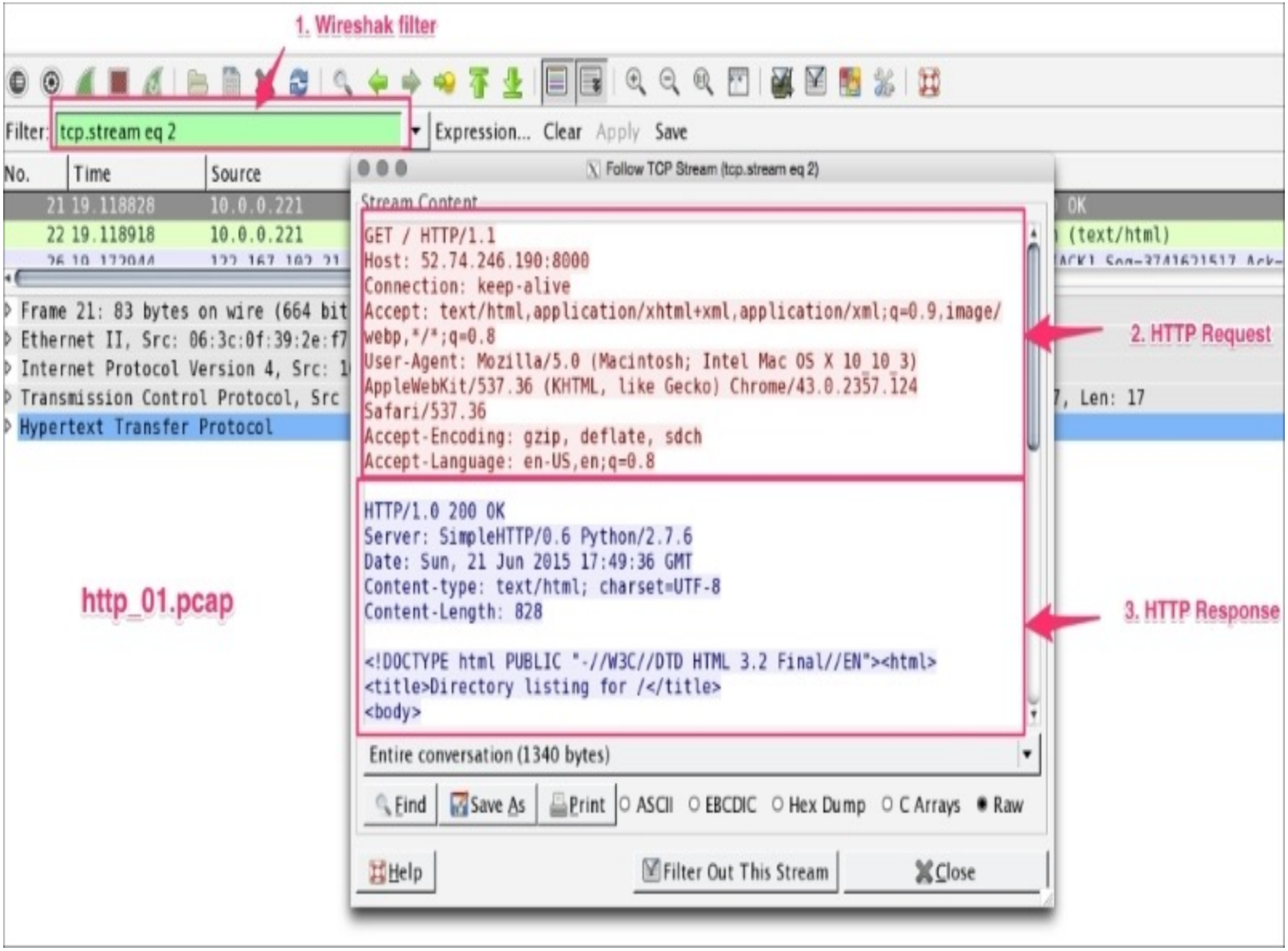
Version: HTTP version number used (e.g. HTTP/1.0 or HTTP/1.1).
Status code and associated phrase indicate the result of the request. Some example status codes and associated phrases include:

- **200 OK: Request succeeded and the information is returned in the response.**
- **301 Moved Permanently: Requested object has been permanently moved; new URL is specified in Location: header of the response message. The client software will automatically retrieve the new URL.**
- **400 Bad Request: A generic error code indicating that the request could not be understood by the server.**
- **404 Not Found: The requested document does not exist**
- **505 HTTP Version Not Supported: The request HTTP protocol version is not supported by the server.**

Entity Body: The requested object if the response is successful.



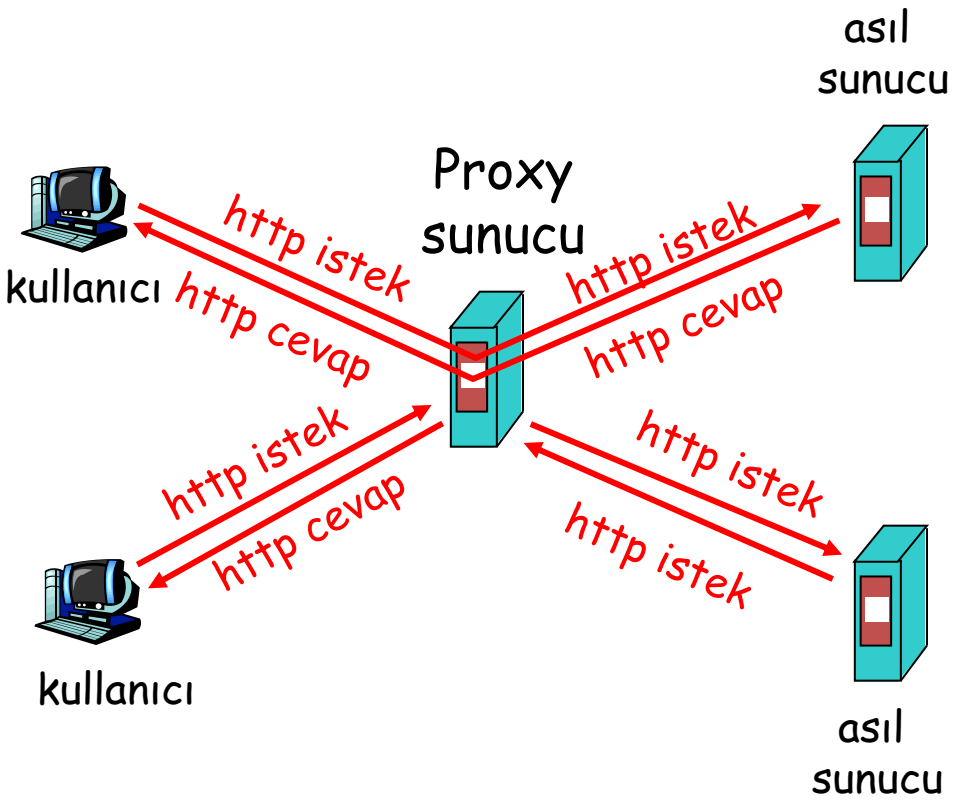
Connection: close for non-persistent TCP connections.
Date: Current GMT date
Server: Server type used
Last-Modified: of object
Content-Length: of object
Content-Type: MIME Type/sub-type of object


1. Wireshak filter
Filter: tcp.stream eq 2
Expression... Clear Apply Save
No. Time Source
21 19.118828 10.0.0.221
22 19.118918 10.0.0.221
Frame 21: 83 bytes on wire (664 bit
Ethernet II, Src: 06:3c:0f:39:2e:f7
Internet Protocol Version 4, Src: 1
Transmission Control Protocol, Src
Hypertext Transfer Protocol
http_01.pcap
Follow TCP Stream (tcp.stream eq 2)
Stream Content
GET / HTTP/1.1
Host: 52.74.246.190:8000
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.124 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
2. HTTP Request
HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/2.7.6
Date: Sun, 21 Jun 2015 17:49:36 GMT
Content-type: text/html; charset=UTF-8
Content-Length: 828
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html>
<title>Directory listing for /</title>
<body>
3. HTTP Response
Entire conversation (1340 bytes)
Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw
Help Filter Out This Stream Close

Web Cep Bellekleri (vekil (proxy) sunucu)

Amaç: kullanıcı isteğini asıl sunucuya erişmeden sağlamak

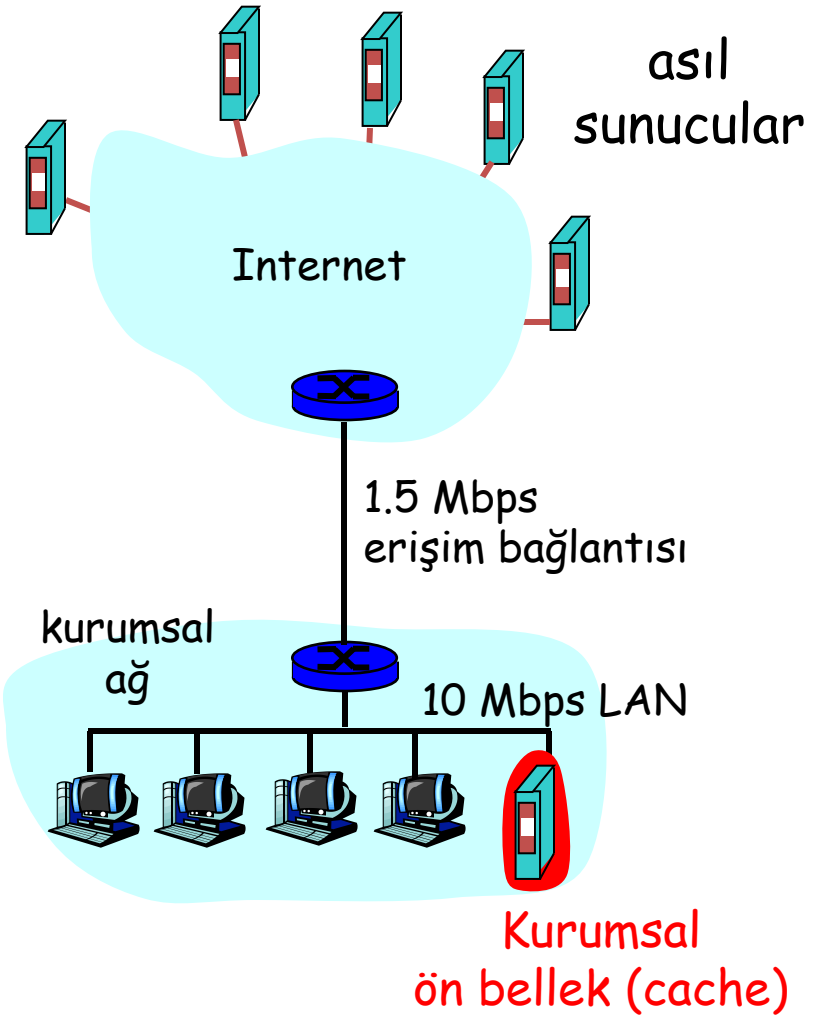
- Kullanıcı tarayıcıyı ayarlar: web cep belleği üzerinden ulaşır
- kullanıcı tüm http isteklerini web cep belleğine gönderir
  - istenilen web cep belleğinde ise: web cep belleği istenilen sağlar
  - değil ise web cep belleği asıl sunucudan ister ve sonra kullanıcıya istenileni gönderir



Neden Web Cep Belleği?

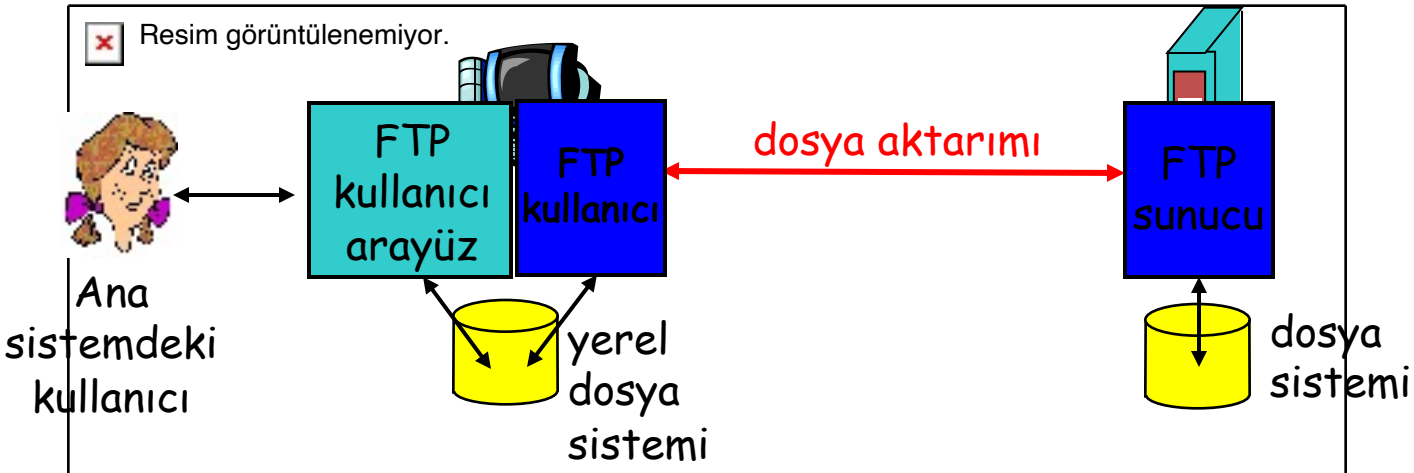
Varsayım: cep bellek kullanıcıya "yakın" ise (örneğin., aynı ağ (network) içinde ise)

- daha az cevap süresi: cep bellek kullanıcıya "daha yakın"
- uzakta olan sunuculara olan trafiği azaltır
  - kurumsal/yerel ISP ağ çıkışındaki bağlantı genellikle dar gecit yapisindadir (bottleneck)



ftp: dosya transfer protokolu

ftp://sunucu_adi/dizin/dosya_adi
ftp://kullanici_adi@sunucu_adi/dizin/dosya_adi



- Ana sisteme veya ana sistemden dosya aktarımı
- Kullanıcı/sunucu modeli
  - *kullanıcı:* transferi başlatan taraf (uzak dosya sistemine ya da sisteminden)
  - *sunucu:* uzaktaki ana sistem (remote host)
- ftp: RFC 959
- ftp sunucu: port 21

ftp: ayrık kontrol, veri bağlantıları

- ftp kullanıcısı ftp sunucusnu port 21 üzerinden aktarım protokolu olarak TCP'yi belirleyerek temasa geçer
- İki paralel TCP bağlantısı açılır:
  - kontrol: kullanıcı ve sunucu arasında komutlar, cevaplar değiştirilir.

    "band kontrolu dışında"
  - veri: sunucudan veya sunucuya dosya verileri
- ftp sunucusu "durumu" korur: kılavuz kütük (directory), önceden doğrulama (authentication)

ftp komutları, cevapları

Örnek komutlar:

- ASCII metin olarak kontrol kanalı üzerinden gönderilir
- **USER *kullanıcı ismi***
- **PASS *şifre***
- **LIST** bulunulan directory içerisinde dosyaların listesini verir
- **RETR dosya ismi** dosyayı (gets) alır
- **STOR dosya ismi** dosyayı ana sisteme (host) saklar (puts)

Örnek dönüş kodları

- durum kodu ve cümlesi (http'de olduğu gibi)
- **331 Username OK, password required**
- **125 data connection already open; transfer starting**
- **425 Can't open data connection**
- **452 Error writing file**

E-posta

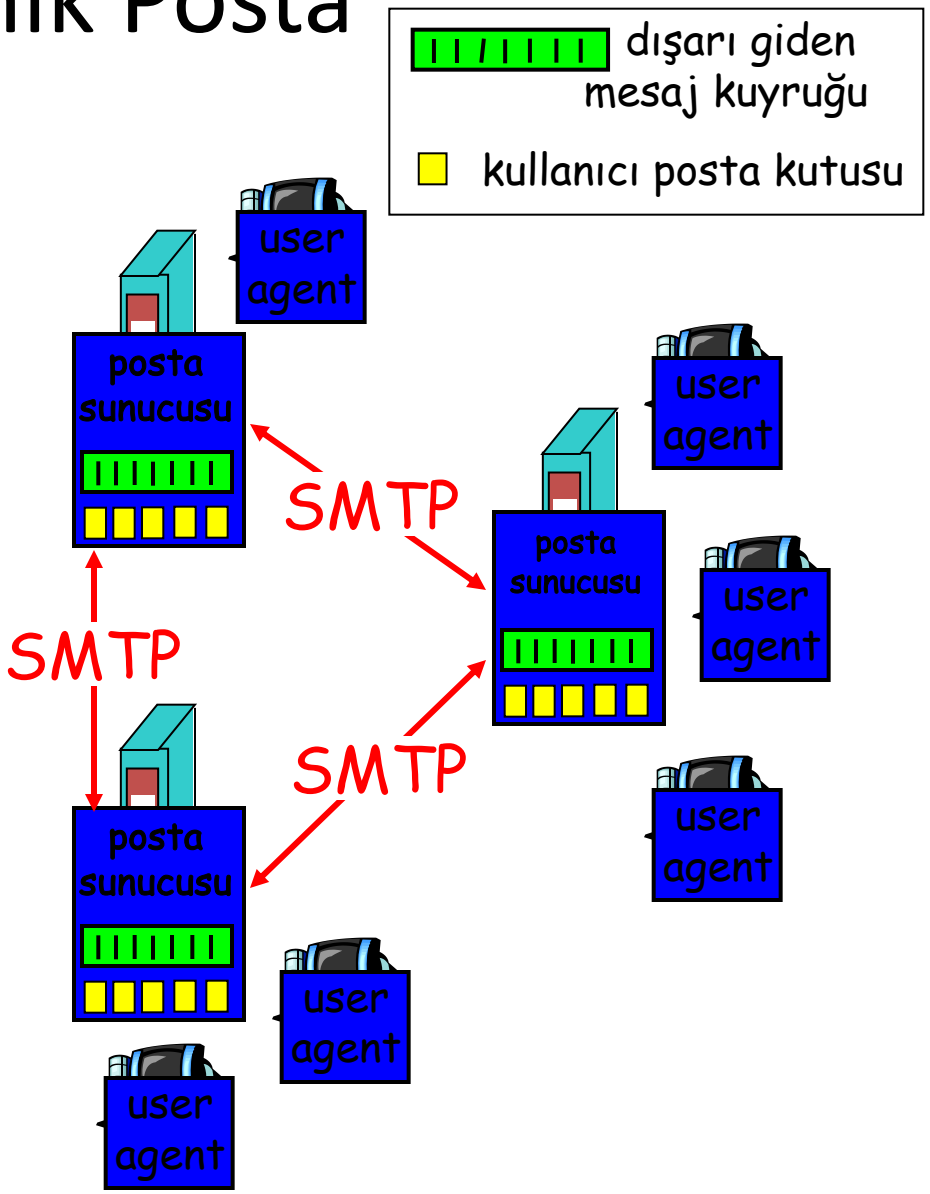
- E-posta, yazma ortamı sunan bir yardımcı program aracılığıyla yazılır; daha sonra uygulama katmanında SMTP protokolüne gönderilir.
- Burada alıcı ve gönderici adresleri yazıldıktan sonra, hazırlanan mektup bir alt katmana, yani ulaşım katmanına gönderilir.

Elektronik Posta

Üç temel bileşen:

- kullanıcılar
- posta sunucuları
- Basit posta akatarım (simple mail transfer) protokolu"

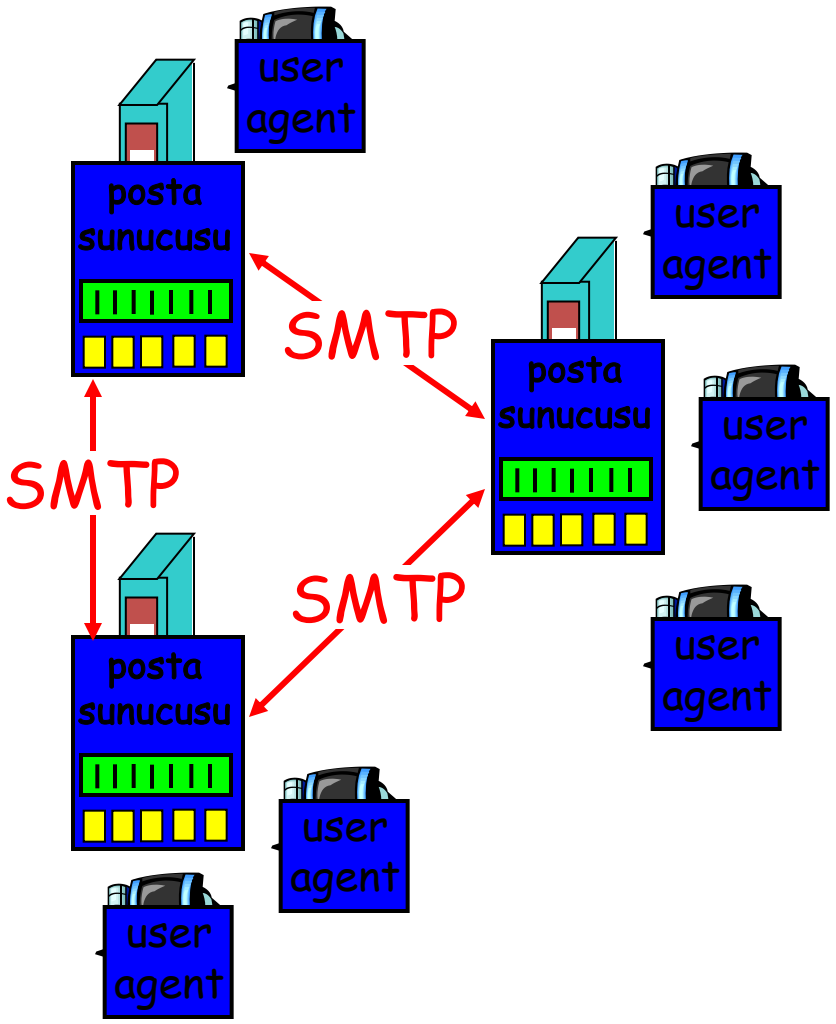
- User Agent (Kullanıcı arayüzü)
- "posta okuyucusu"
- Posta mesajlarını düzenleyen, yazan, okuyan
- örneğin, Eudora, Outlook, elm, Netscape Messenger
- giden, gelen mesajları sunucuda saklama



Elektronik Posta: posta sunucuları

Posta Sunucuları

- posta kutusu kullanıcı için (okunmak üzere) gelen mesajları bulundurur
- mesaj posta mesajları (gönderilmek üzere) kuyruğu
- smtp protokolu e-posta mesajları göndermek için posta servis sağlayıcıları arasında
  - *"kullanıcı":* gönderici posta sunucusu
  - "sunucu": posta alan sunucu



Elektronik Posta: smtp [RFC 821]

- Kullanıcıdan sunucuya eposta mesajlarını güvenilir bir şekilde aktarmak üzere tcp kullanılır, port 25
- doğrudan aktarım: gönderici sunucudan alıcı sunucuya
- Aktarımın üç aşaması
  - el sıkışması (handshaking, (greeting))
  - mesajların aktarılması
  - bitiş
- komut/cevap etkileşimi
  - komutlar: ASCII text
  - cevap: durum kodu ve cümle

Örnek smtp etkileşimi

```
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
```

smtp: özet

http ile karşılaştırma:

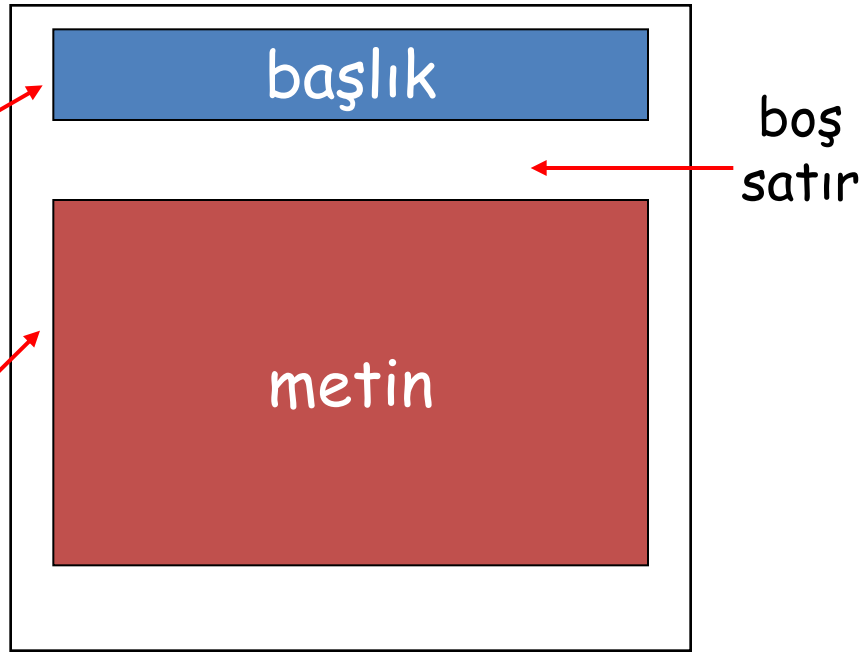
- http: pull
- email: push
- Her ikisi de ASCII komut/cevap etkileşim, durum kodlarına sahiptir
- http: dosya içerisinde birçok nesne farklı bağlantılar ile gönderilir
- smtp: birçok nesnebir tek bağlantı ile gönderilir

Posta mesaj formatı

smtp: e-posta mesajlarini degistirmek uzere protokol

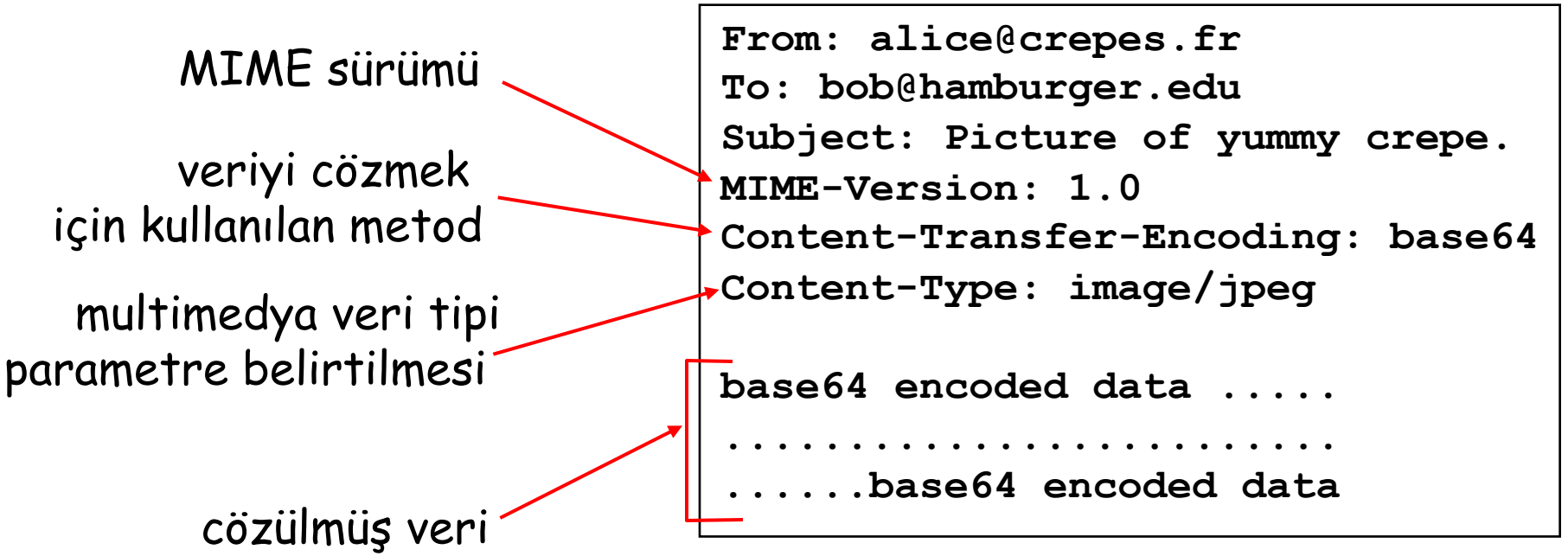
RFC 822: metin mesaj formati icin standart:

- Baslik satirlari, ornegin,
  - To:
  - From:
  - Subject:
  - *smtp komutlarindan farkli*!
- metin kismi
  - "mesaj", ASCII karakterleri kullanarak

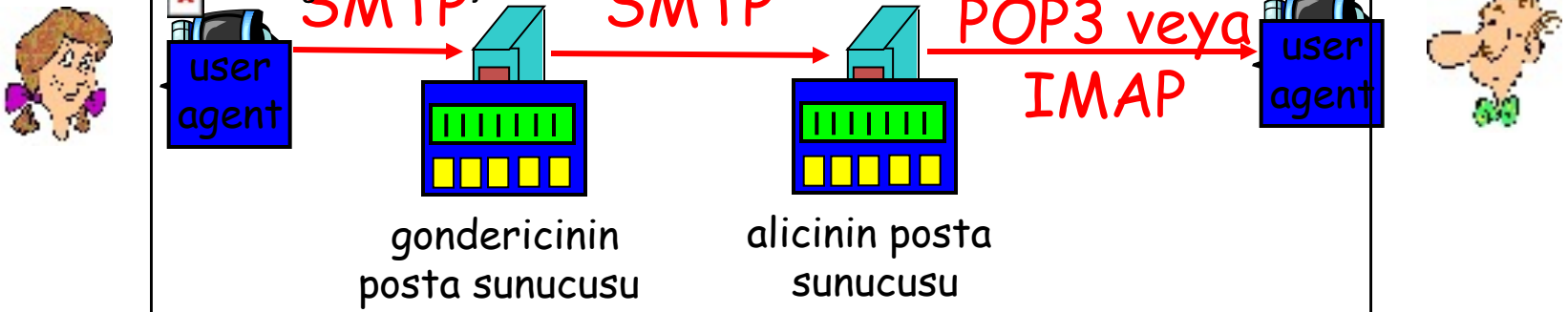


Mesaj formatı: multimedya uzantıları

- MIME: multimedia mail extension, RFC 2045, 2056
- Mesaj başlığındaki ilave satırlar MIME içerik bilgisini verir



Posta erişim protokolleri



- SMTP: alıcının sunucusuna teslimat/saklama
- Posta erişim protokolu: sunucudan yeniden alınması
  - POP: Post Office Protocol [RFC 1939]
    - yetkilendirme (agent <-->server) ve aliş (download)
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - daha fazla özellikler (daha fazla karışık)
    - sunucuda saklanan mesajlarin düzenlenmesi

POP3 protokolu

dogrulama sureci

- kullanıcı komutları:
  - **`user: kullanıcı ismini belirtme`**
  - **`pass:`** şifre
- sunucu cevaplari
  - **`+OK`**
  - **`-ERR`**
- aktarım süreci, kullanıcı:
- **`list:`** mesaj sayılarının listesi
- **`retr:`** meşajları sayısı ile alınması
- **`dele:`** silme

```
S: +OK POP3 server ready
C: user alice
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
```

DNS: Domain Name System

Kişiler: birçok tanımlayıcı:

- Sosyal Güvenlik Numarası, isim, pasaport #

- İnternet anasistemleri, yönlendiriciler(router):
  - IP adresi (32 bit) – veri akışını adreslendirmek için kullanılırlar
  - "isim", örneğin, gaia.cs.umass.edu – kişiler tarafından kullanılırlar

  Soru: IP adresleri ile isimler arasında dönüşüm ?

Domain Name System(Alan İsimlendirme Sistemi):

- *Dağıtılmış veri yapısı*

  birçok *isim sunucusunun* hierarşik (sıra) düzeninde uygulanırlar

- *Uygulama katmanı protoklou*

  ana sistem, yönlendiriciler, isim servis saglayıcıları isimleri *çözmek* üzere haberleşirler (adres/isim dönüşümü)
  - not: çekirdek İnternet fonksiyonu, uygulama katmanı protokolu olarak uygulanır
  - ağ "uç"'larında kompleks yapı

DNS isim servis sağlayıcıları

Neden DNS tek merkezli olamaz?

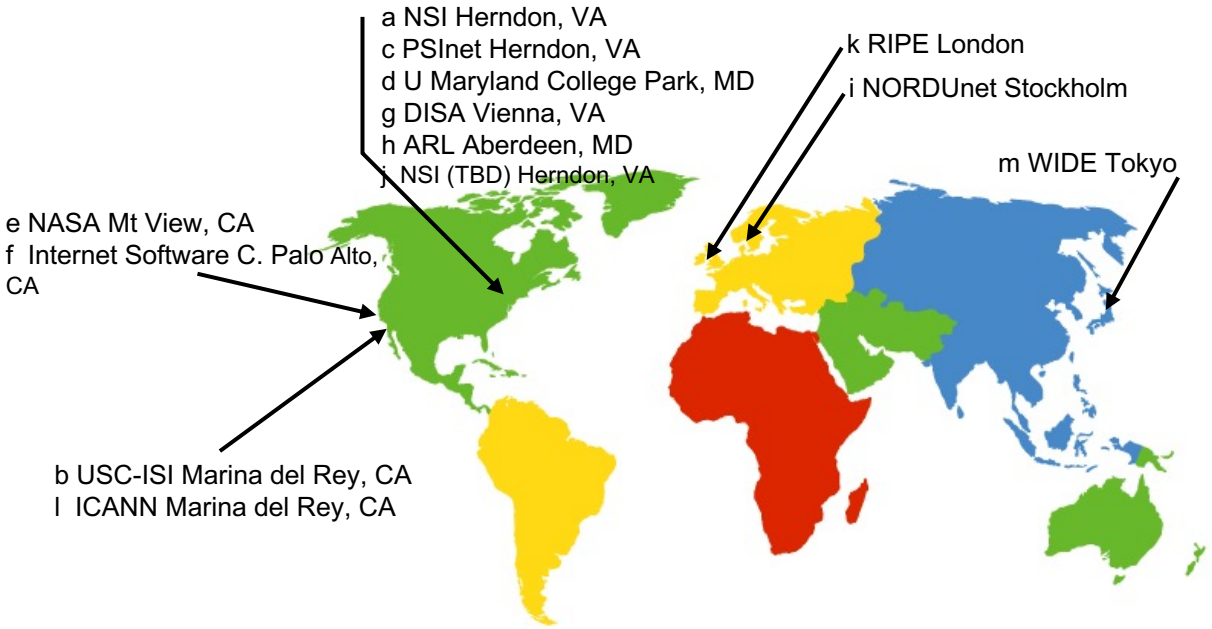
- tek noktada hata oluşması
- trafik hacimi
- uzak merkezi veri tabanı
- bakım

ölçeklendirme yapılamaz*!*

- servis sağlayıcılarının hepsi isim-IP adresleri dönüşümüne sahip değildirler
- yerel isim servis sağlayıcılar:
  - her ISP, şirket yerel *(default)* isim servis sağlayıcıya sahiptir
  - ana sistem DNS isteği ilk olarak yerel isim servis sağlayıcıya gider
- otoriter (authoritative) isim servis sağlayıcıları:
  - ana sistem için: bu ana sistemin IP adreslerini, isim bilgilerini saklar
  - bu ana sistem için isim/adres dönüşümünü gerçekler

DNS: Root isim servis sağlayıcıları

- isim/IP adres dönüşümünü çözemeyen yerel isim servis sağlayıcıları tarafından aranırlar
- root isim servis sağlayıcıları:
  - isim dönüşümü bilinmiyor ise (authoritative) isim servis sağlayıcılarına başvururlar
  - dönüşümü sağlar
  - yerel isim servis sağlayıcılarına dönüşümü gönderir



dünya genelinde
13 adet root isim
servis sağlayıcı