Uygulama katmanı ve güvenlik protokolları (Uygulama+Sunum+Oturum)

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

Ağı tanıyabilen (Network aware application) bazı son-kullanıcı uygulama programları, uygulama katmanı servis ve 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-Hizmetleri)

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 programları , kullanıcıya mesajları oluşturmak için bir yol bir araç sunar.
- Uygulama katmanı servisleri ağa bir arayüz oluşturmak için vardır.
- Protokoller bu hizmetlerin 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 katmanında tanımlı protokoller), 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..**

TCP/IP 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.

OSI- Uygulama katmanı (Application Layer)

Uygulama katmanı, Kullanıcı proseslerinin, uygulama katmanı protokolleri yardımı ile ağa erişmek için gerekli alt yapıyı sağlar.

Kullanıcı-Ağ etkileşimi için ilk basamaktır. HTTP (HyperText Transfer Protocol), Telnet, FTP, ... uygulama protokollerinden birkaçıdır.

OSI- Sunuş Katmanı (Presentation layer)

- Gönderilen/alınan bilginin söz dizimi ve anlambilimsel yapısıyla (semantics) ilgilenir. Yani üst/alt katmandan gelen bilgilrti/verileri veri/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, karakterleri farklı kodlamalarla (UNİCODE, ASCII) 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/açılması, kriptografi/dekriptografi v.b işlemler bu katmanda yapılır.
- SMB, AFP, NCP, MIDI, HTML, GIF, TIFF, JPEG, ASCII, EBCXDR, ASN.1DIC

OSI: Oturum Katmanı (Session Layer)

- Bu katman yardımı ile farklı bilgisayarlardaki kullanıcı prosesleri arasında oturumlar kurulması sağlanır.
- Bu işlem oturumların kurulmasını, yönetilmesini ve bitirilmesini içerir.
- İletişimin senkronizasyonunu sağlar. **Yani** bir iletişimin kopmasından sonra iki tarafın kaldıkları yerden iletişime devam edebilmeleri için bir sağlama noktası (checkpoint) kullanma).
- Oturumlara farklı kalitede servisler de sunabilir.

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

Bazı İnternet 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

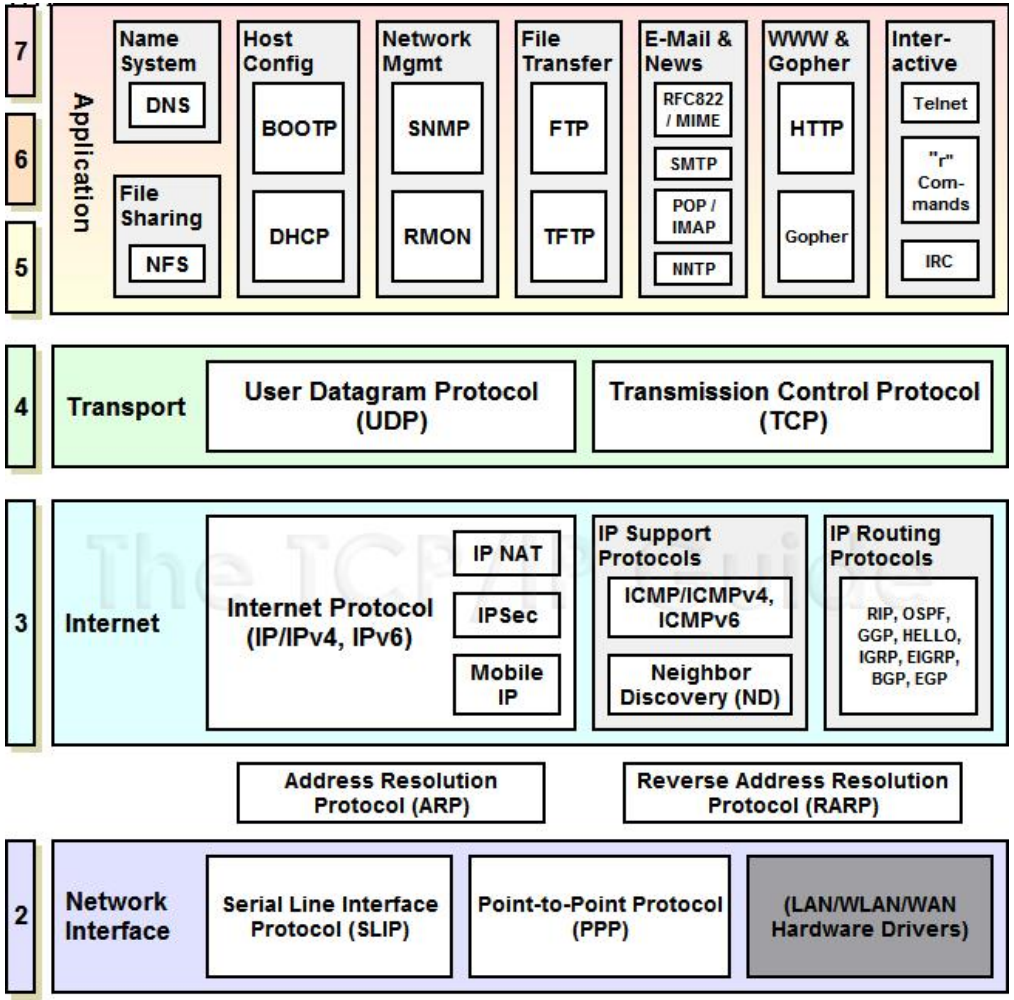
Yazılan programlar

- 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

Uygulama Katmanı protokolleri



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.

Uygulama Mimarileri

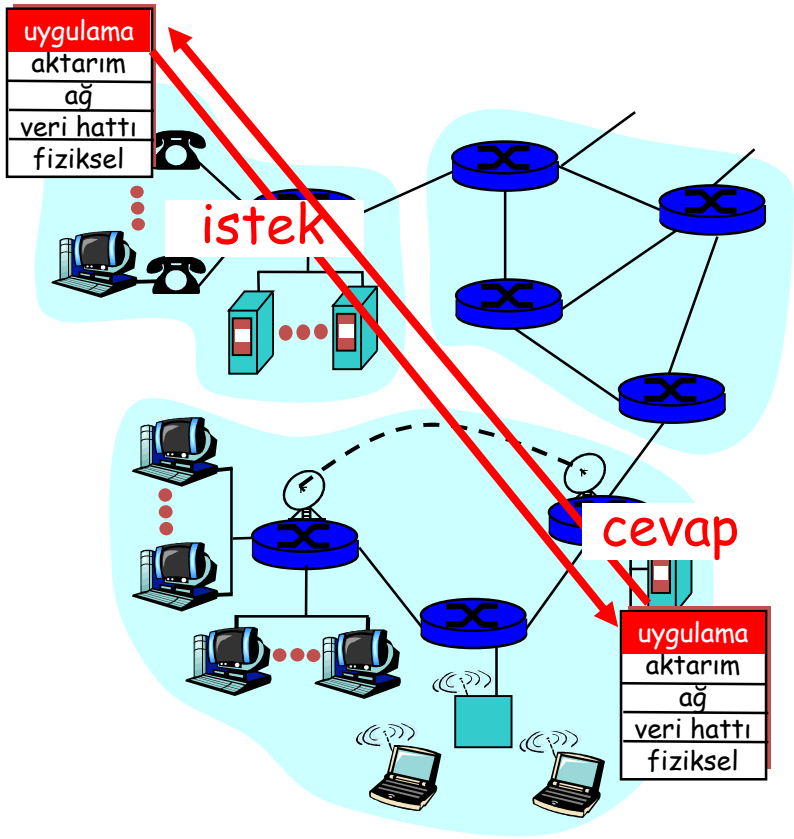
- Client-Server (İstemci –Sunucu)
- Peer-To-Peer (Eş düzey)
- Hibrid (C-S, P2P)

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

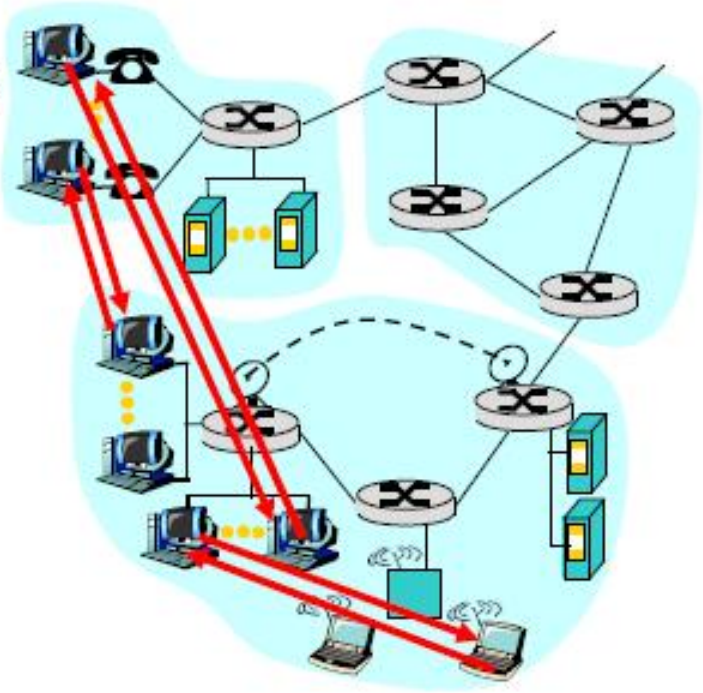


Sunucu (Server):

- 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

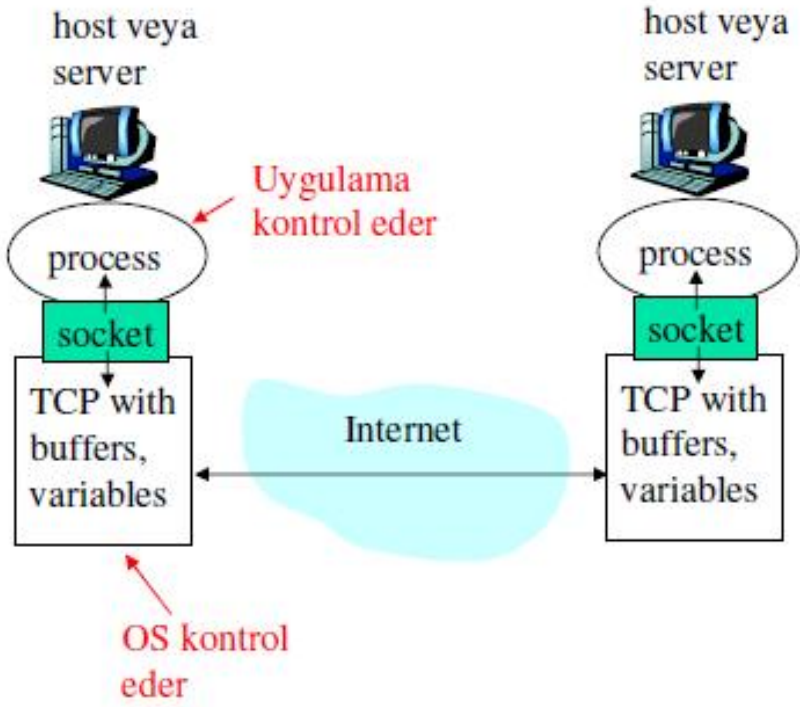
Client process: iletişimi başlatan process

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

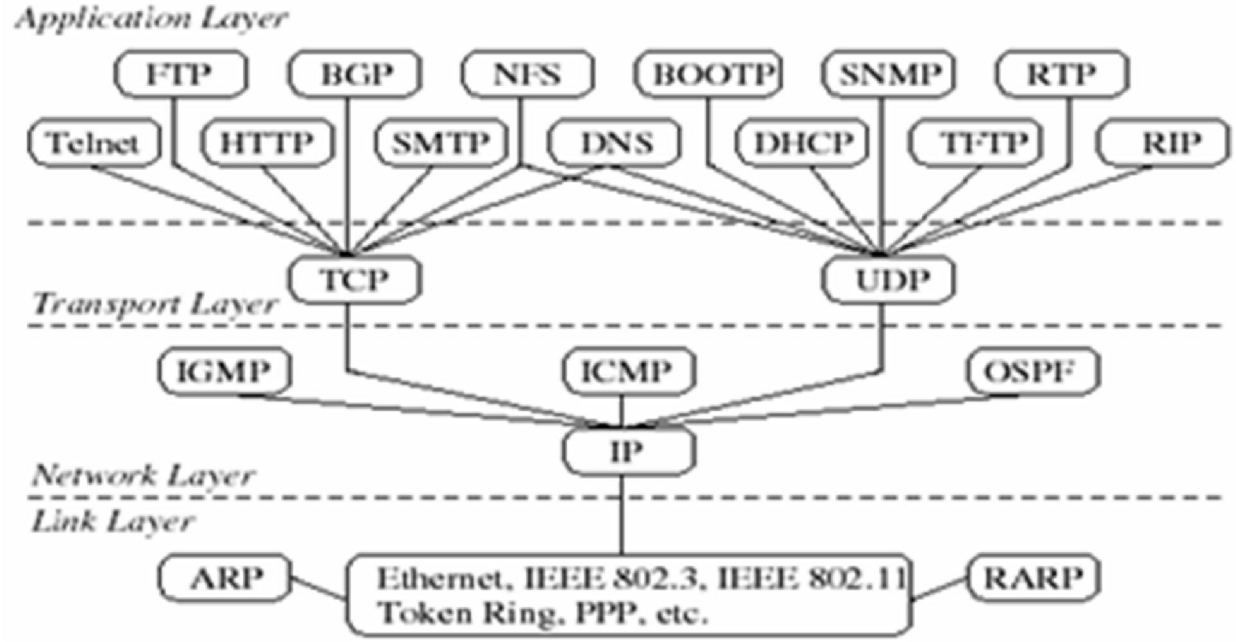


- API: (1) transport protokol seçer; (2) Parametre belirler

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

Protocols in Different Layers

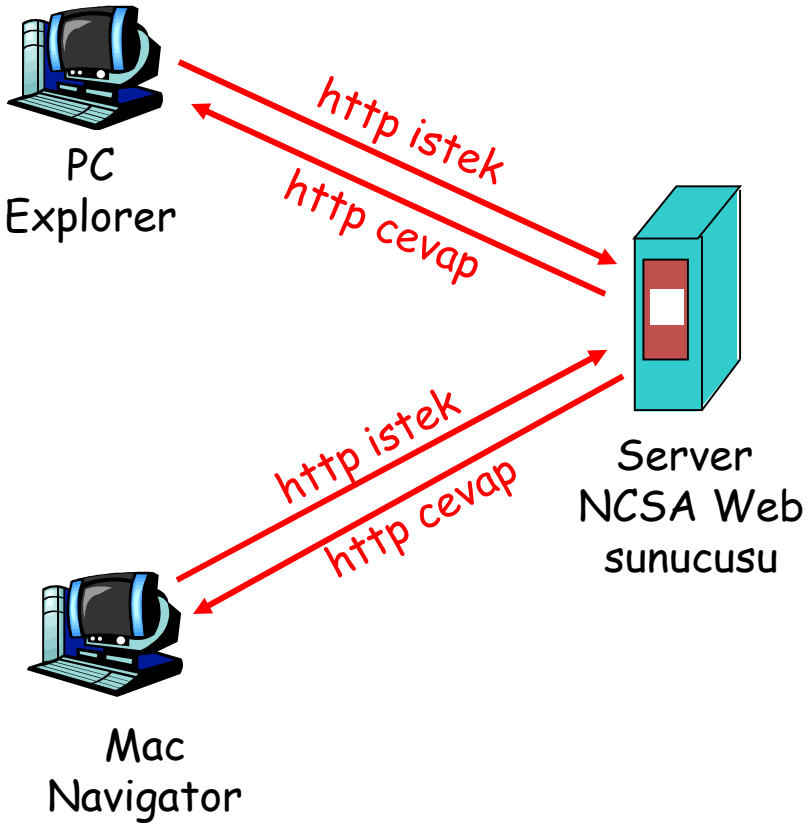


Web: http protokolu

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 bilmediğimiz 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?

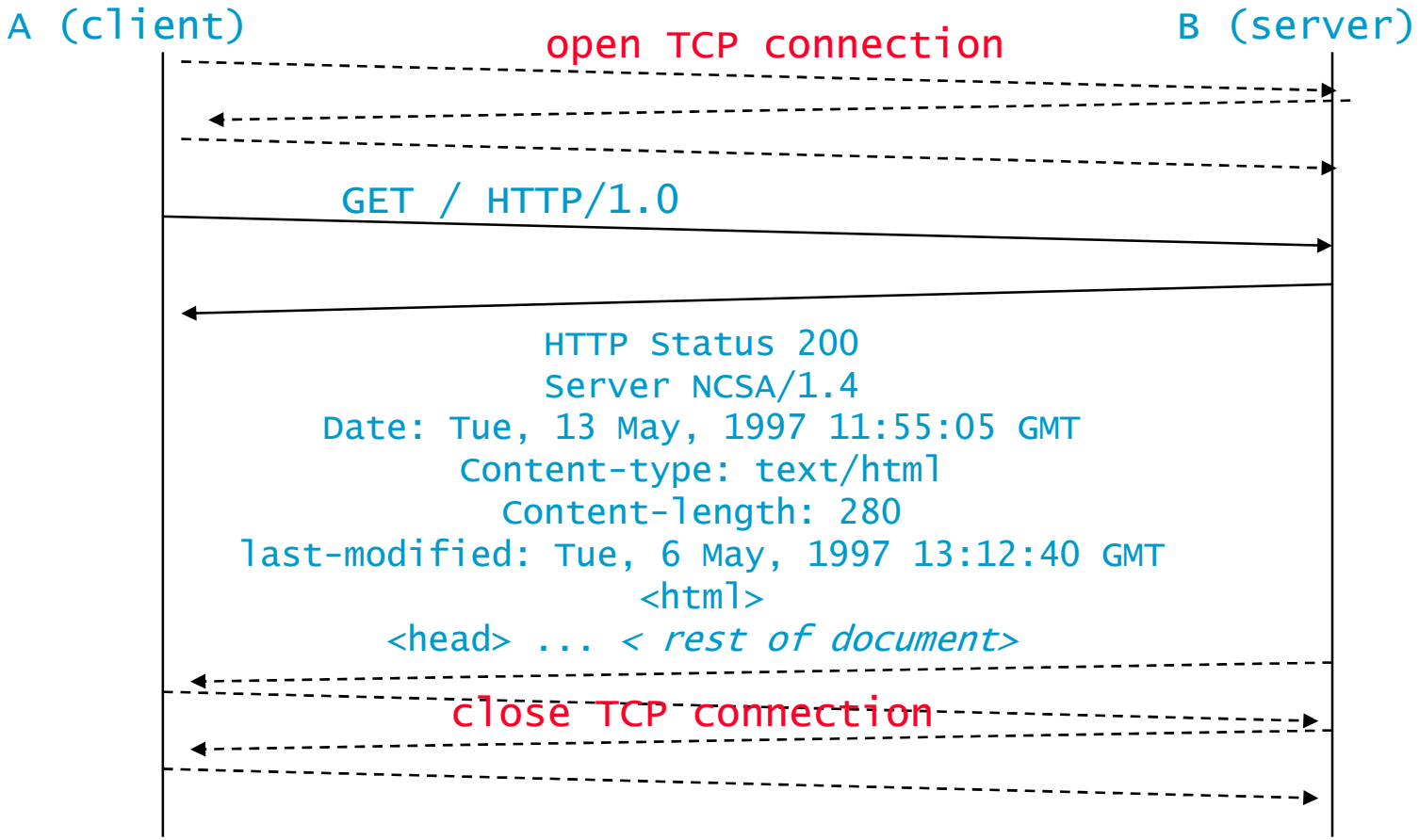
http://www.mbe.com.tr/mbe/yapı.html

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"

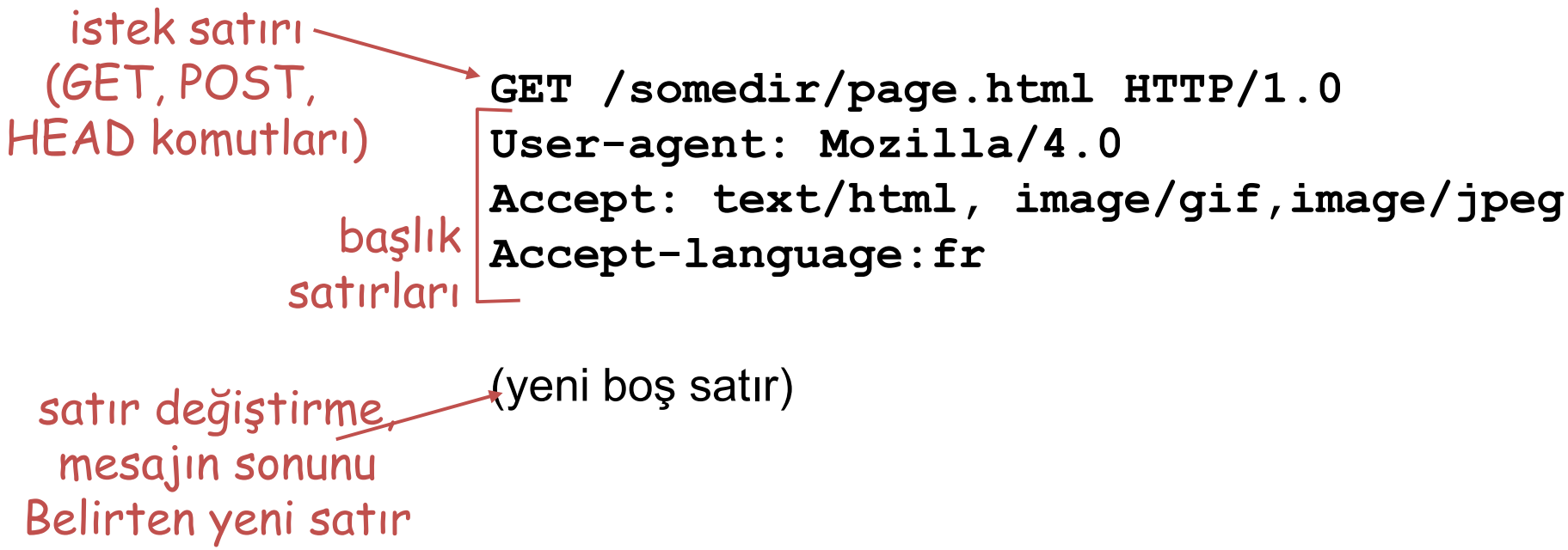
Çoğu zaman yalnızca tam domain ismi kullanılır. Web sunucular domain ismi ile çağırılan web sayfalarında otomatik olarak "index.html, default.html, home.htm, index.htm" sayfalarından hangisi varsayılan olarak belirlenmişse o dosyayı getirir. Bu nedenle çoğu zaman dosya adı yazmadan yalnızca http://www.mbe.com.tr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

HTTP uses TCP



http mesaj formatı

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



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)

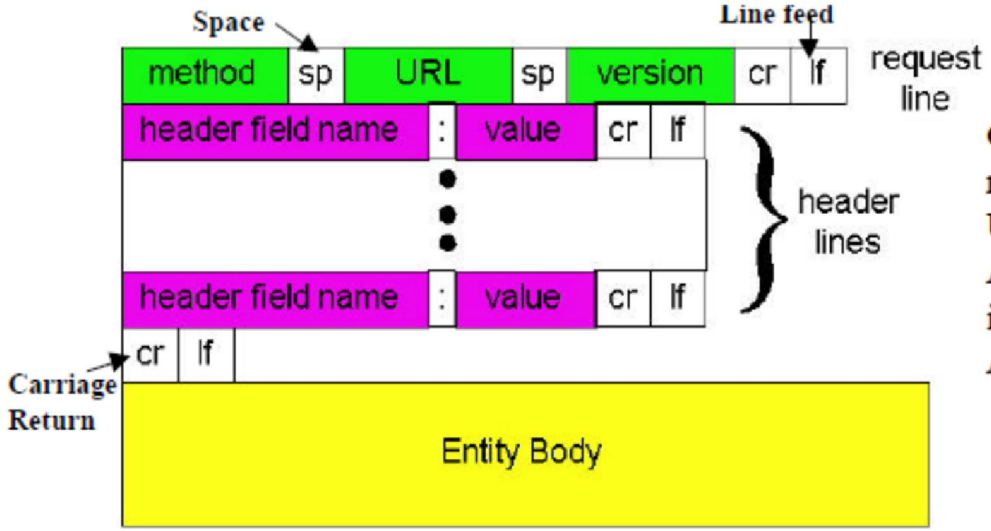
- **Typical Response Message From A Server:**

  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 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

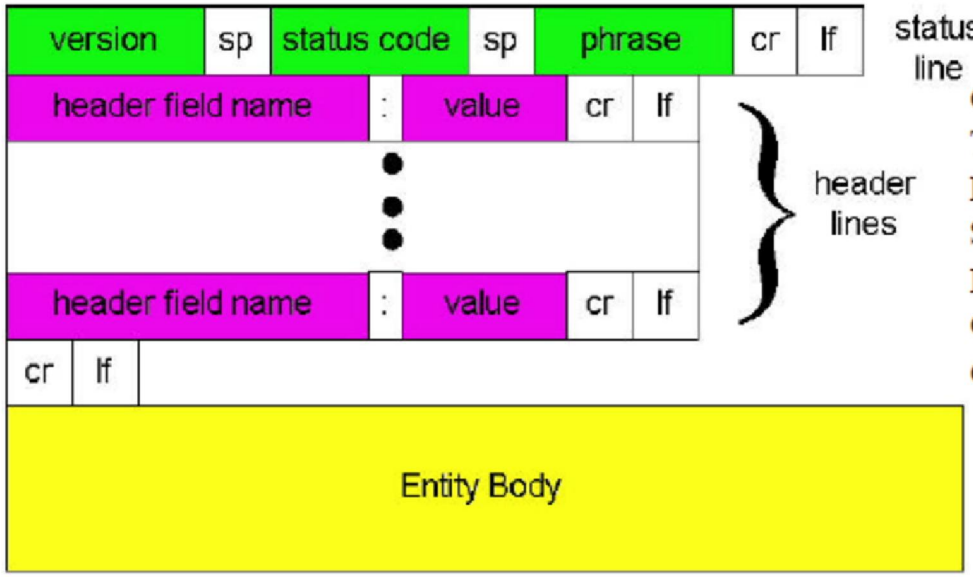
505 HTTP Version Not Supported

HTTP Message Formats:
General Format of A Response Message

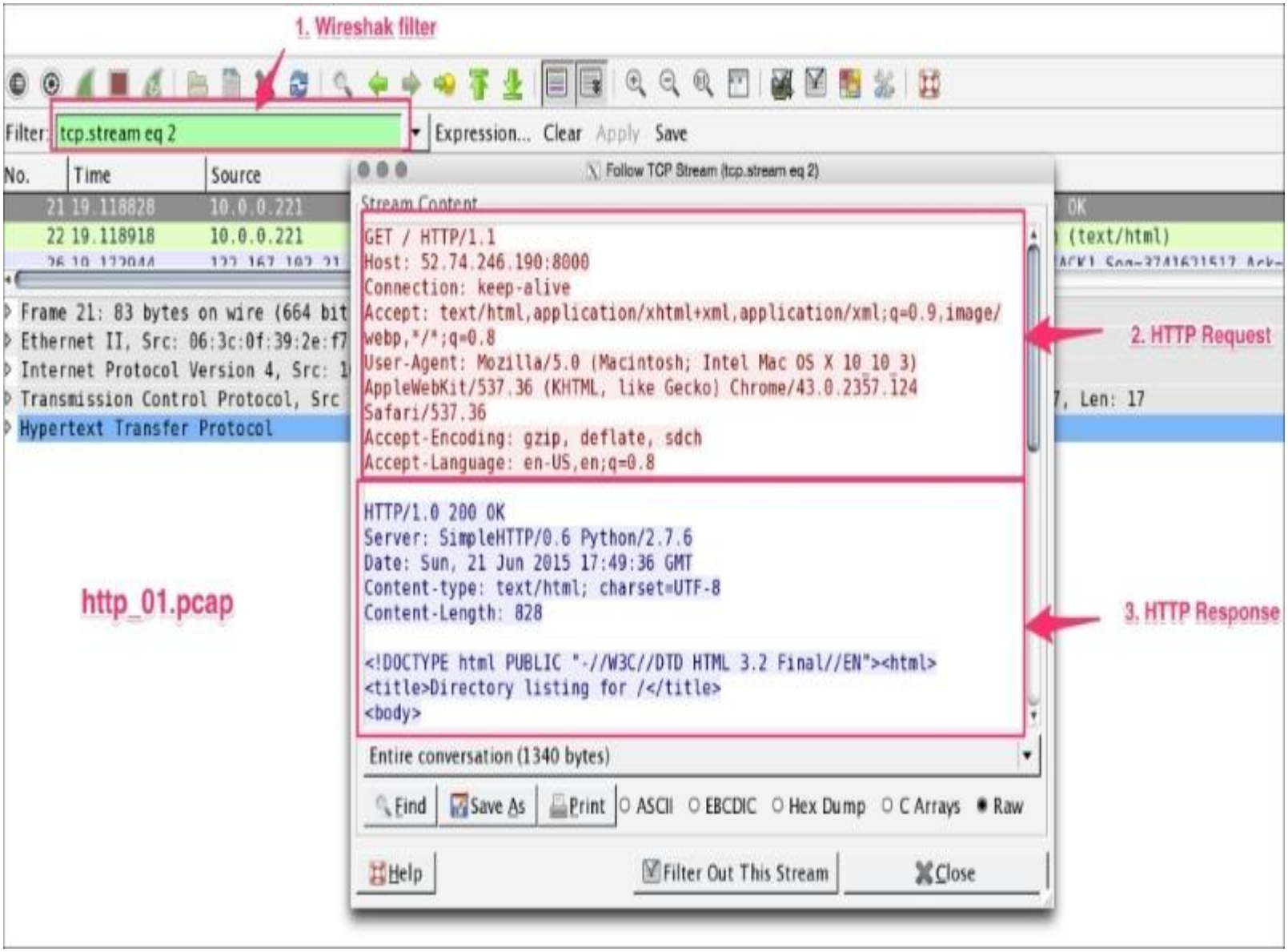
Version: HTTP version number used (e.g. HTTP/1.0 or HTTP/1.1).
Status code and associated phrase indicate the result of the request. Some example status codes and associated phrases include:

- **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.

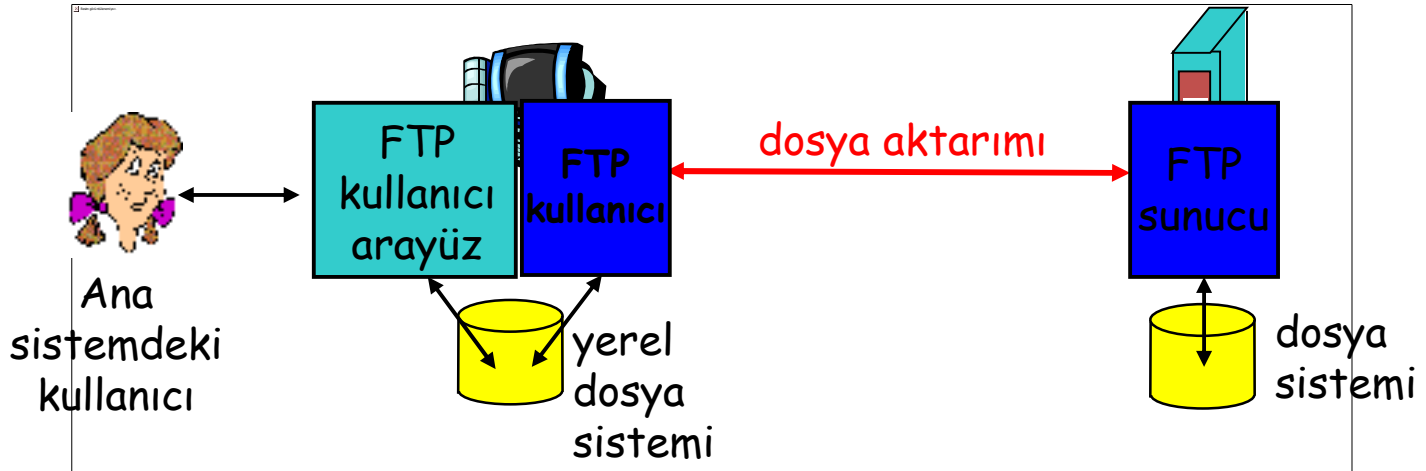


Connection: close for non-persistent TCP connections.
Date: Current GMT date
Server: Server type used
Last-Modified: of object
Content-Length: of object
Content-Type: MIME Type/sub-type of object


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
(text/html)
Len: 17

ftp: dosya transfer protokolu

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

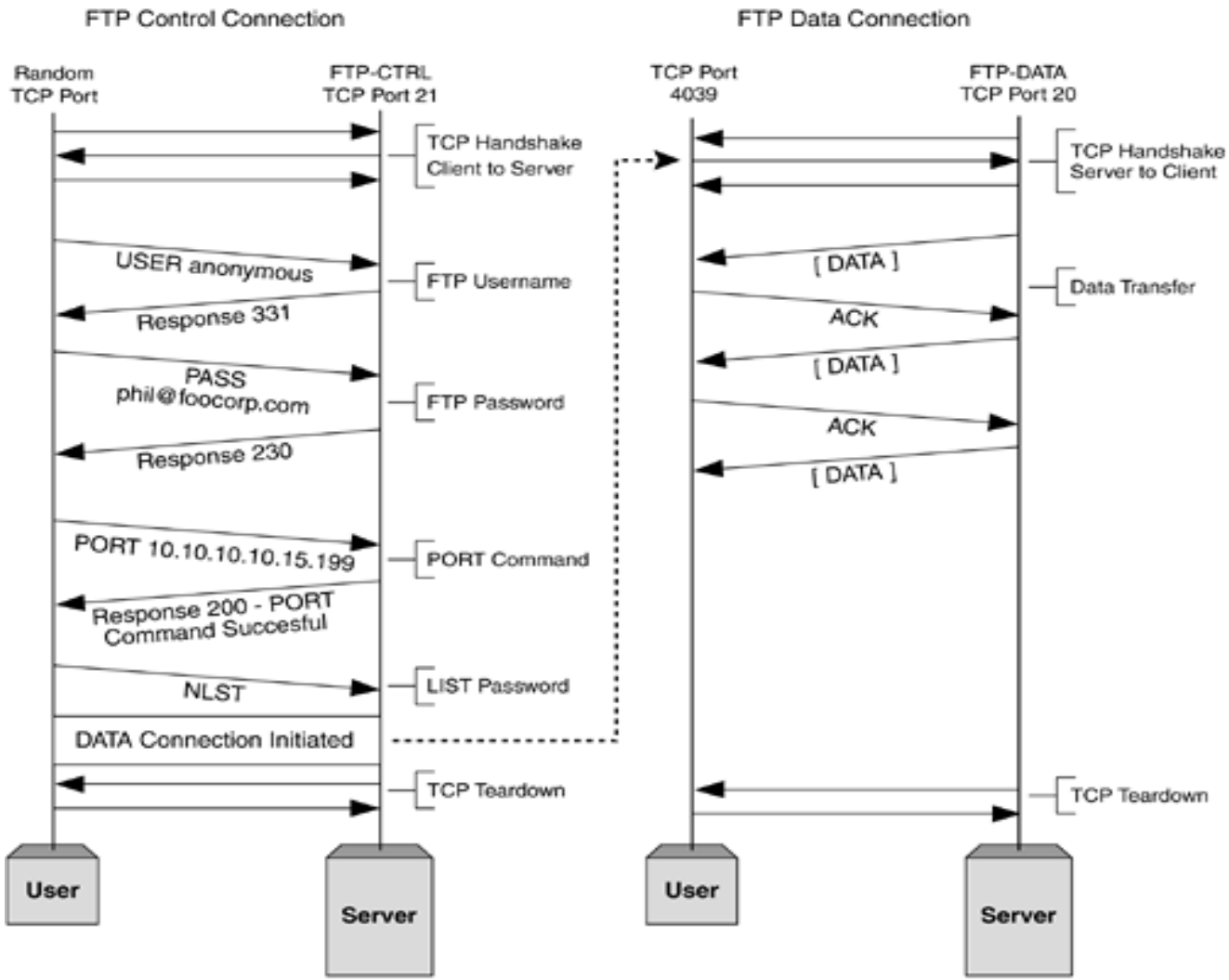


- Ana sisteme veya ana sistemden dosya aktarımı
- Kullanıcı/sunucu modeli
  - *kullanıcı:* transferi başlatan taraf (uzak dosya sistemine ya da sisteminden)
  - *sunucu:* uzaktaki ana sistem (remote host)
- ftp: RFC 959
- ftp sunucu: port 21

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)



In an active FTP connection, the client will use the Control connection to tell the server, via a PORT command, which IP address and TCP port it should establish the Data connection to. The server then opens a Data connection to that IP address and port using the well-known Port 20 as the source.

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**

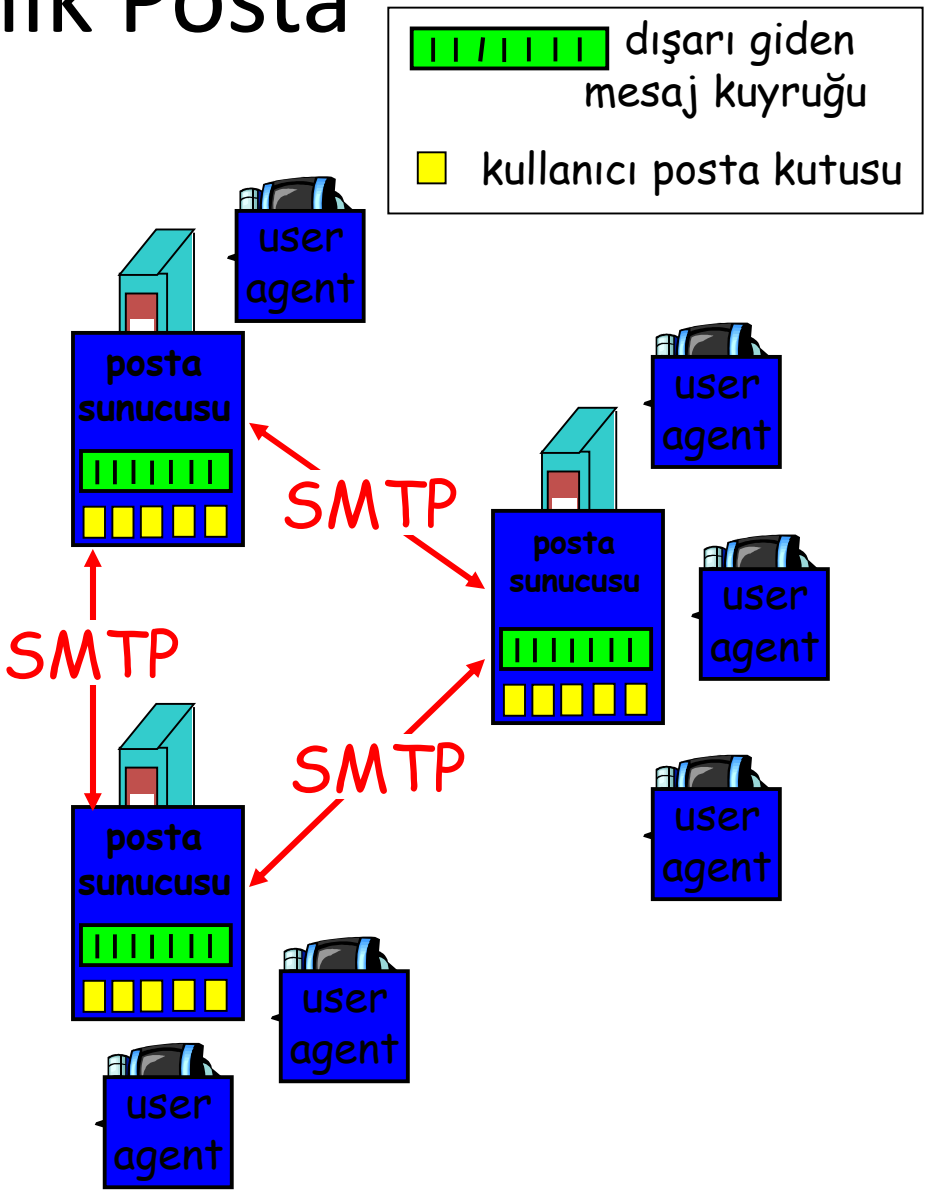
E-posta

- 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:

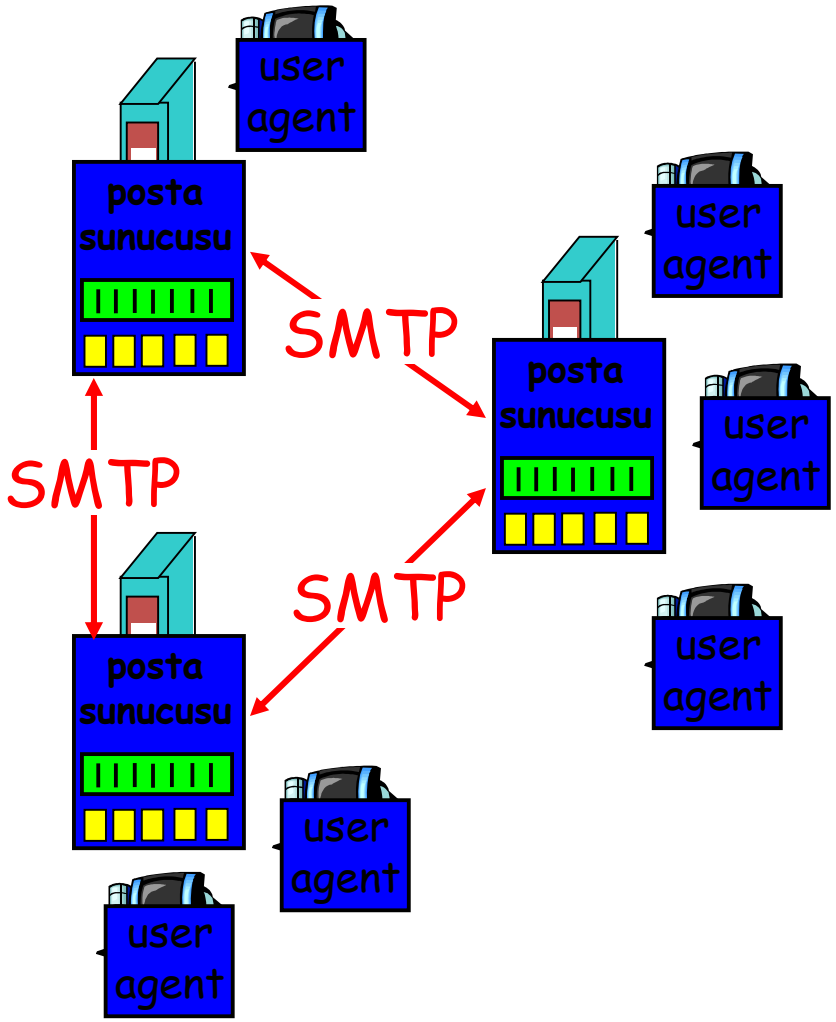
- kullanıcılar
- posta sunucuları
- Basit posta akatarım (simple mail transfer) protokolu"
- User Agent (Kullanıcı arayüzü)
- "posta okuyucusu"
- Posta mesajlarını düzenleyen, yazan, okuyan
- örneğin, Eudora, Outlook, elm, Netscape Messenger
- giden, gelen mesajları sunucuda saklama



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

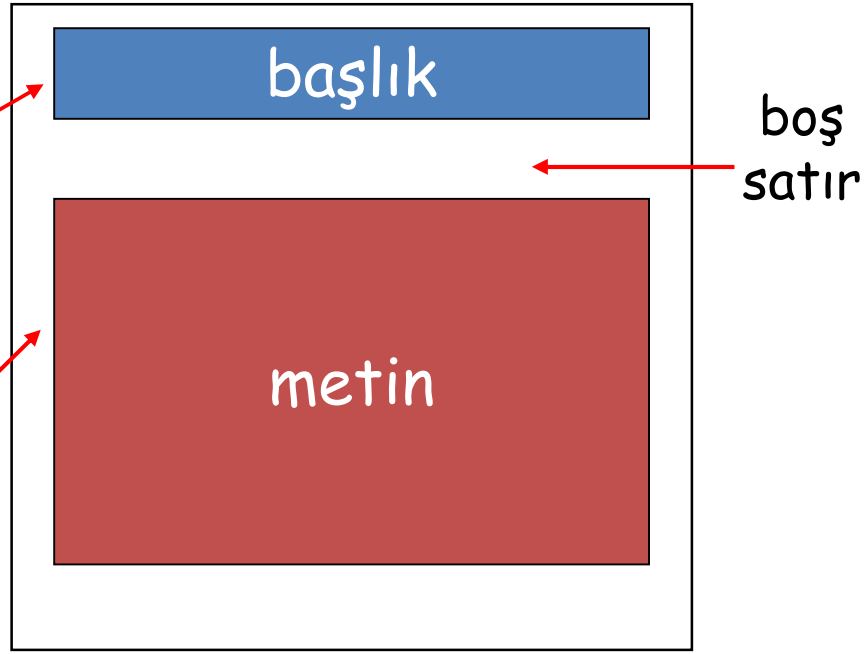
```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250  Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like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

Posta mesaj formatı

smtp: e-posta mesajlarini degistirmek uzere protokol

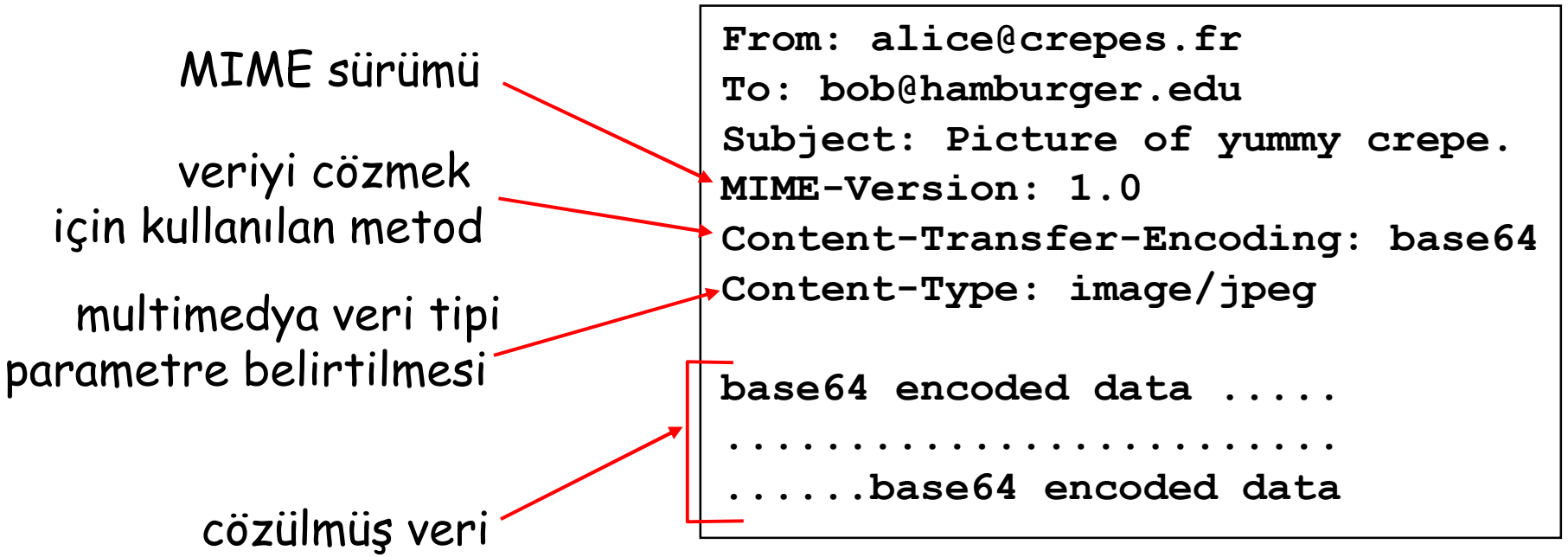
RFC 822: metin mesaj formati icin standart:

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

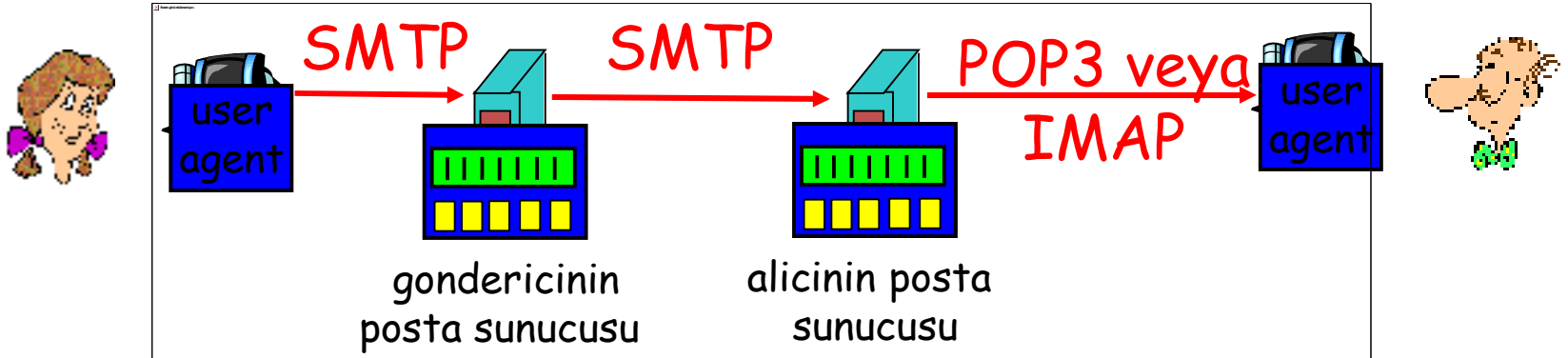


Mesaj formatı: multimedya uzantıları

- 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?

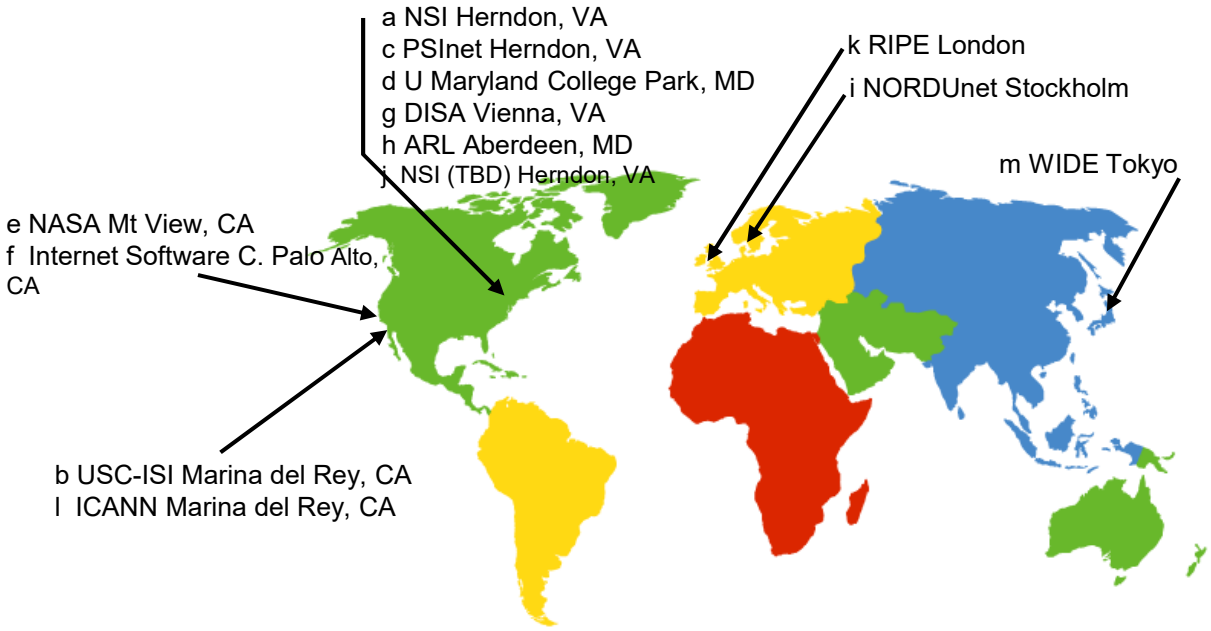
- tek noktada hata oluşması
- trafik hacimi
- uzak merkezi veri tabanı
- bakım

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

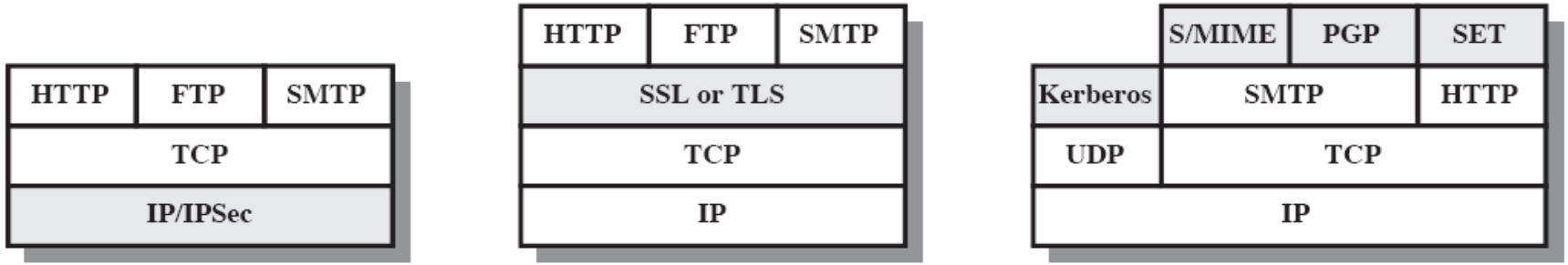


dünya genelinde
13 adet root isim servis sağlayıcı

TCP /IP ve OSI

- Katmanlı ağ modelinde, herbir katmandaki veri, doğrudan açık text formundadır.
- Ağda bulunduğu sürece değiştirilip değiştirilmediği, kimin gönderdiği veya kimin aldığı garantisi olmayan bir yapıdadır.
- Paketin ele geçirildiği anda içeriğinin rahatlıkla okunabildiği bir yapılanma sözkonusudur.
- Dolayısıyla bu TCP/IP yapılarına ek olarak herbir katmanda güvenlik protokollarıda oluşturulmuştur.

- Application Layer (Uygulama Katmanı):
  - PGP (Pretty Good Privacy)
  - S/MIME (Secure/Multipurpose Internet Mail Extension)
  - S-HTTP (Secure-HTTP)
  - HTTPS (Hypertext Transfer Protocol over Secure Socket Layer)
  - SET (Secure Electronic Transactions)
  - KERBEROS
- Transport Layer (Transport Katmanı):
  - SSL
  - TLS
- Network Layer (Netwok Katmanı):
  - IPSec
  - VPN
- Data Link Layer (Veri Bağı Katmanı):
  - PPTP
  - RADIUS
  - TACACS+



(a) Network Level

(b) Transport Level

(c) Application Level

Uygulama Katmanı;

- telnet (port23), mail (port25), finger(port 75), http (80.port) v.b son kullanıcı uygulamaları için kullanılan uygulama katmanı protokollarına uygun olarak, ağ üzerinden veri göndermek ve almak için servislerin sağlandığı katman «UYGULAMA» katmanıdır.
- Transport katmanıyla etkileşimi
  - İşletim sistemine bağımlıdır,
  - Soket arayüzleri ile sağlanır

Uygulama Katmanı Güvenliği

- Avantajları:
  - Çok esnektir.
  - Kullanıcı seviyesinde çalışır→Kullanıcı kimlik bilgilerine erişim kolaydır.
  - Verilere tam erişim→ inkar edilemezliğin sağlanması kolaydır ve güvenlik ayrıntı düzeyi küçüktür.
  - Uygulama tabanlı güvenlik sağlar.
- Dezavantajları:
  - Son kullanıcı bilgisayarlarında geçekleştirilir.
  - Herbir uygulama için farklıdır→
    - Pahalıdır
    - Hata yapma olasılığı yüksektir.
- Sıkça kullanılan Uygulama katmanı güvenlik protokolları:
  - PGP (Pretty Good Privacy)
  - S/MIME (Secure/Multipurpose Internet Mail Extension)
  - S-HTTP (Secure-HTTP)
  - HTTPS (Hypertext Transfer Protocol over Secure Socket Layer)
  - SET (Secure Electronic Transactions)
  - KERBEROS

HTTPS

- HTTPS, HTTP’nin güvenli uzantısıdır. TLS/SSL sertifikası yüklemiş olan websiteleri sunucu ile güvenli bir bağlantı kurmak için HTTPS protokolü kullanabilirler.

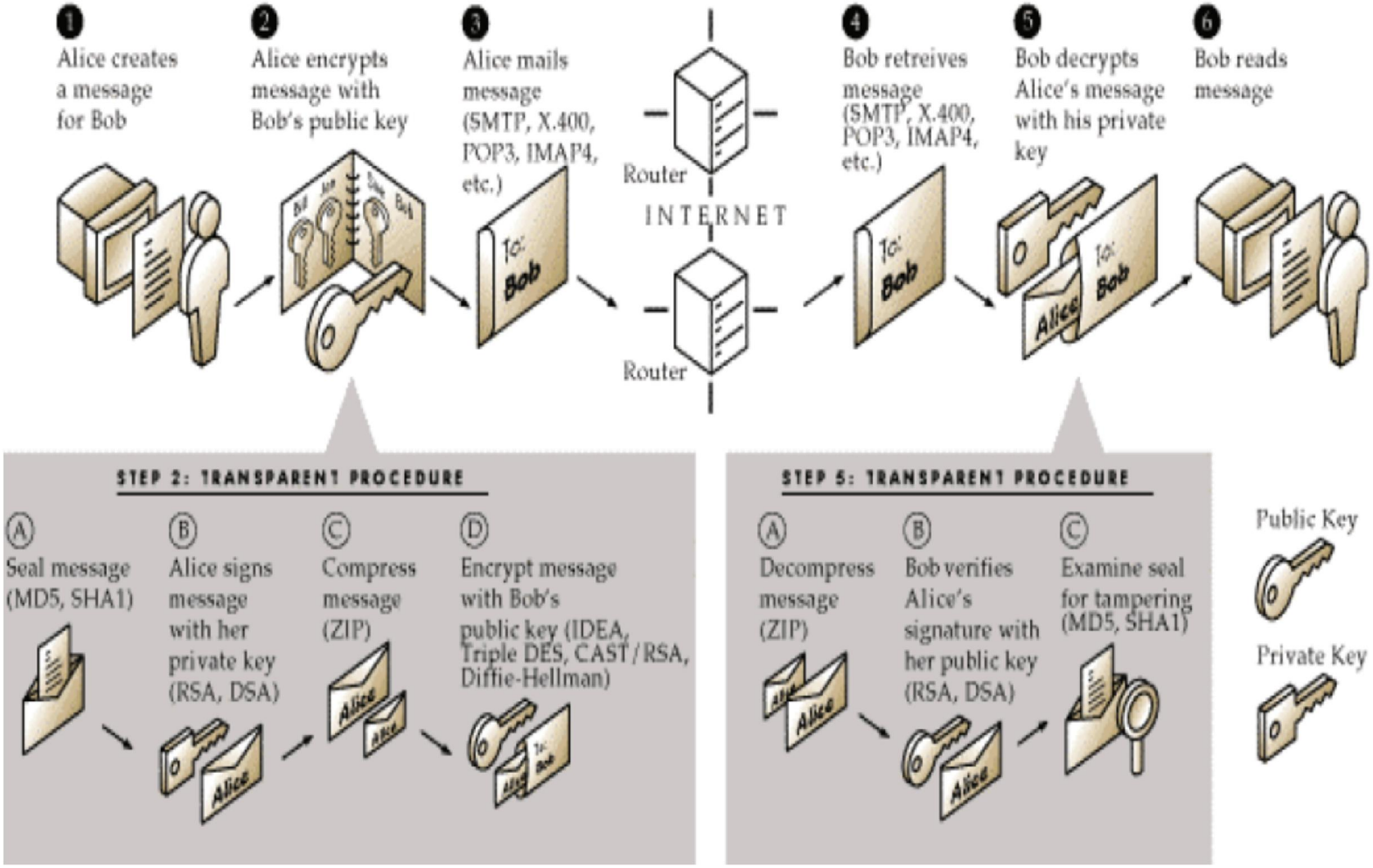
- Webmasterlar tarafından websiteleri güvende tutmak ve insanların ödemelerini sağlıklı şekilde yapabilmeleri için kullanılan SSL/TLS sertifikalarıyla çoğu insan tarafından tanınmaktadır.

- Bir websitesinin bu sertifikayı kullanıp kullanmadığını, adres çubuğundaki URL adresinin hemen yanında bulunan yeşil kilit ikonundan anlayabilirsiniz.

Pretty Good Privacy (PGP) – e-mail gizliliğini sağlamak için

- Şifreleme *; güvenlik, gizlilik* için hayati önem taşımaktadır. PGP ise e-mail gizliğini sağlayan en popüler protokoldur.
- Pretty Good Privacy (PGP), Phil Zimmermann tarafından geliştirilmiştir. Public/private key hibrit kriptosistem temellidir. PGP genellikle dosya imzalama; metin, eposta, dosya, hard disk ve dizin şifreleme gibi iletişim güvenliğini artırma yöntemlerinde kullanılır. **Daha çok ücretsiz bir e-posta güvenlik yazılımı olarak kullanılır.**
- **PGP, kullanıcıları arasında güven çemberi (halkası) oluşturarak çalışır. İki kişi ile başlayan güvenlik, her kullanıcı tarafından tutulan public key / isim çiftleri ile önemli bir halka oluştururlar.**
- PGP 'nin içerdiği güven mekanizmaları karmaşıkdır. Güvenli otorite kavramına karsı çıkan bir yaklaşımın ürünü olduğu için, isteyen herkes, açık anahtarları onaylayan bir otorite olabilir. Ancak kullanıcılar güvendiği kişileri kendileri seçerler.
- **Kullanıcıların açık anahtarları PGP açık anahtar sunucularında tutulur ve istem bazında bu sunucular tarafından dagıtılır. Bu sunucular aslında birer veritabanı sunucularıdır ve hiçbir sekilde güvenlik garantisi vermezler. Yani bir açık anahtarın bu sunucularda bulunması, söz konusu açık anahtarın bahsi geçen kisiye ait oldugunun kanıtı degildir. Ancak çogu kullanıcı bu gerçegin farkında olmadan sunucudan indirdigi açık anahtarları dogruymus gibi kullanmaktadır**.

PGP çalışma mekanizması

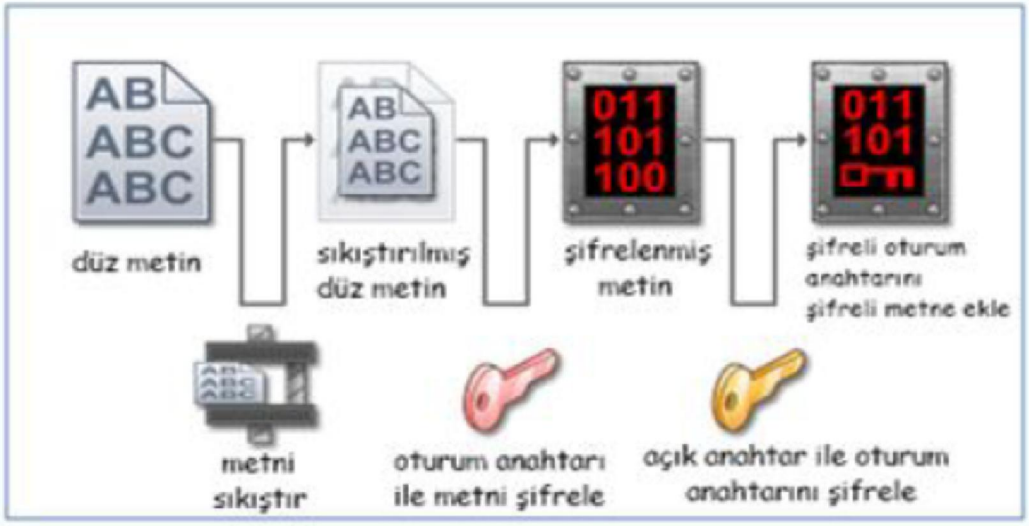


PGP Şifreleme Yönteminin Mekanizması

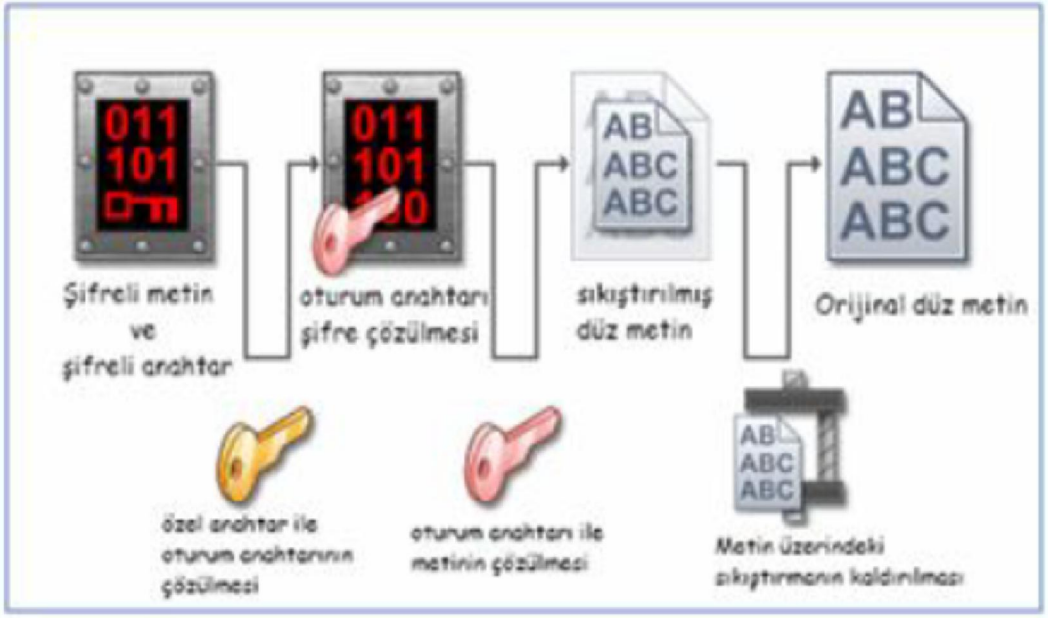
Bir orijinal metnin PGP ile şifreleme ve çözüm süreci aşağıdaki gibidir;

- PGP ile düz bir metinin şifrelenmesi durumunda, PGP öncelikle düz metin üzerinde sıkıştırma operasyonu gerçekleştirir. Sıkıştırma operasyonun sağladığı yararlar; veri aktarımı zamanının azalması, diskte kapladığı alanın azalması, ve daha önemlisi şifreleme güvenliğinin artmasıdır. Şifre analiz yöntemlerinin çoğu, şifreyi kırmak için düz metin içindeki paternlerden yararlanır. Sıkıştırma ise bu düz metinlerdeki paternleri azaltarak, şifre analiz yöntemlerine karşı savunmayı artırmış olur.
- Sıkıştırma işleminden sonra, PGP ***tek seferlik kullanımı olan bir oturum anahtarı (session key***) oluşturur. Bu oturum anahtarı rastgele oluşturulmuş bir anahtardır. Kullanıcı mouse hareketleri ve klavye tuş basılmalarına göre üretilen rastgele değerler ile oluşturulan bu anahtar, çok güvenli ve düz metinleri hızlı bir şekilde şifreleyen anahtar olarak kabul edilir. Bu işlem yukarıda bahsedilen klasik şifreleme ayağını oluşturur.
- **IDEA klasik şifreleme** metoduyla oturum anahtarı kullanılarak metin şifrelemeye tabi tutulur. Bu işlem sayesinde klasik şifrelemenin hızından yararlanılmış olunur.
- Veri şifreleme işlemi gerçekleştirildikten sonra, oturum anahtarı RSA açık anahtarlı şifreleme yöntemi ile şifrelenir. Bu sayede açık anahtarlı şifrelemenin güvenliliğinden de yararlanılmış olunur. Bu şifrelenmiş oturum anahtarı da şifreli metin ile karşı tarafa yollanır.

Şifrelenmiş oturum anahtarını ve şifreli metni alan alıcı taraf, kendi özel anahtarını kullanarak, açık anahtarlı şifreleme metoduyla oturum anahtarının şifresini çözer ve oturum anahtarını açığa çıkarır.
Oturum anahtarı elde edilince, klasik metotla oturum anahtarı kullanılarak şifrelenmiş metin çözülür ve orijinal metni açığa çıkarılır. (Şekil 2)



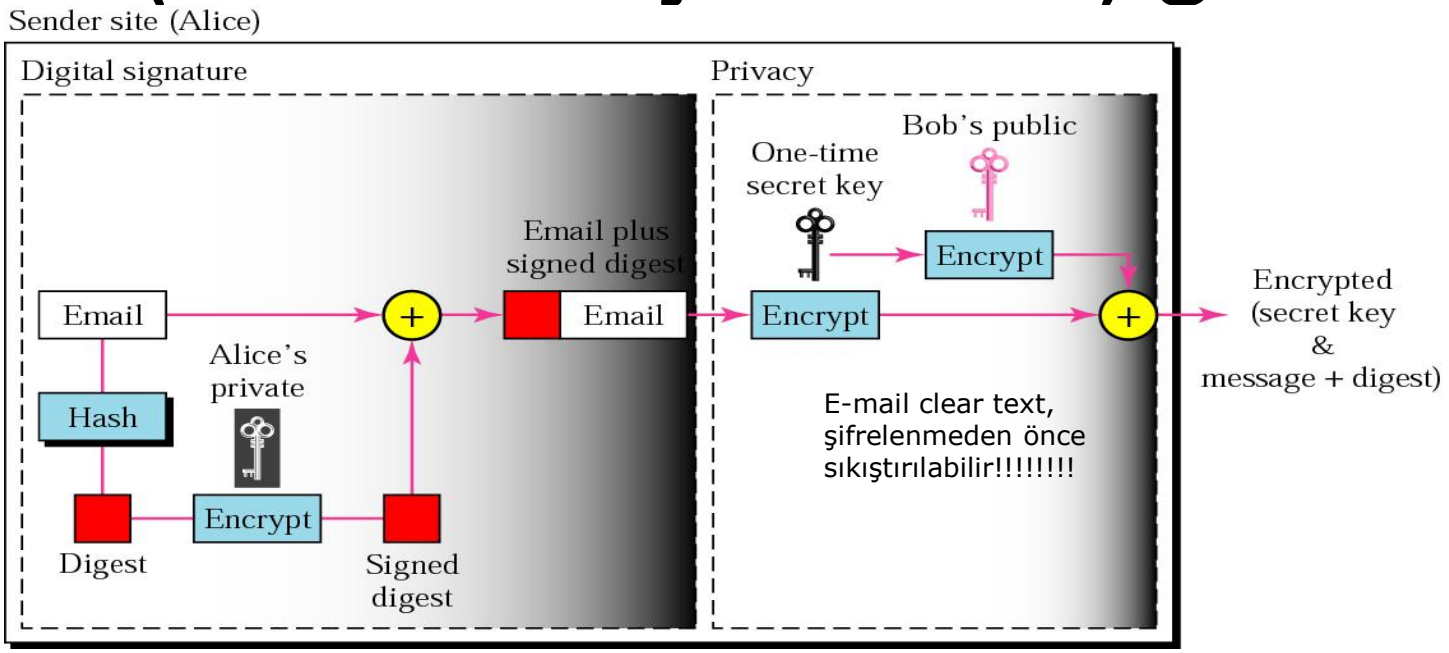
Şekil 1 - PGP Şifreleme Mekanizması



Şekil 2 - PGP Şifre Çözme Mekanizması

PGP iki farklı şifreleme algoritmasını bir araya getirerek, geleneksel şifreleme algoritmalarının (tek anahtar kullanma- simetrik) hızı ile açık anahtarlama algoritmalarının artılarını bir araya getirir. Geleneksel şifreleme algoritmaları, açık anahtarlama şifreleme algoritmalarına göre çok daha hızlı kabul edilir. Açık anahtarlama yöntemleri ise anahtar dağıtımına ve veri transferi problemlerine çözümleri ile ön plana çıkar. Bu iki mekanizmayı beraber kullanarak PGP, güvenlikten taviz vermeden performansı artırmıştır.

PGP (e-imza ve şifreleme) gönderici

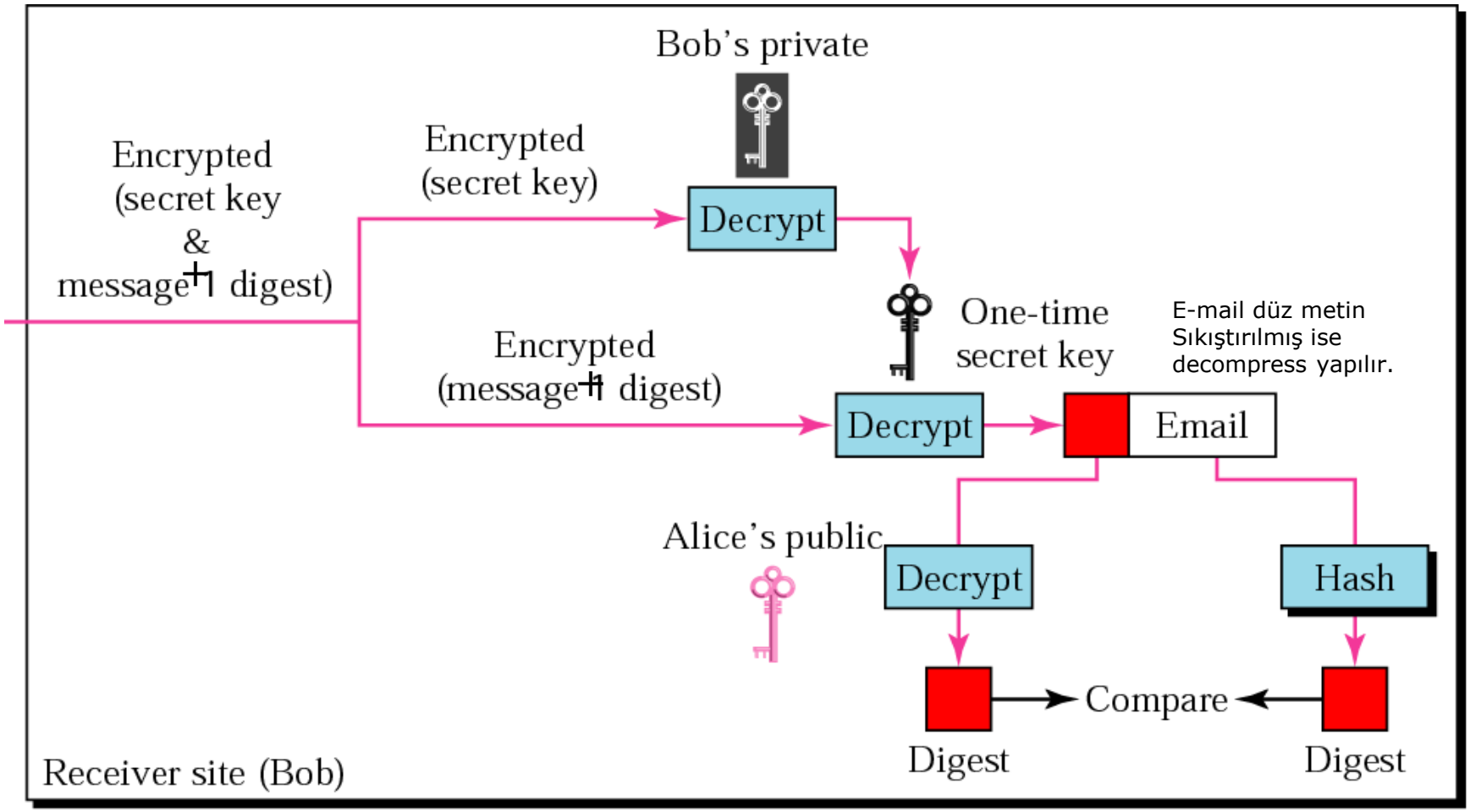


•Pretty Good Privacy (PGP), bir e-posta gönderilirken 4 açıdan güvenlik sağlar (Kimlik Doğrulama +inkar edememe+ Bütünlük + Gizlilik).

•PGP, dijital imzayı (karma ve açık anahtarlı şifrelemenin bir kombinasyonu) , mail bütünlüğünü, kimlik doğrulamayı ve tanıdık olmayanları belirlemek için kullanır.

•Gizlilik sağlamak için gizli bir anahtar ve açık-anahtar şifreleme kombinasyonunu kullanır.

PGP alıcı taraf



PGP’nin Güvenliği

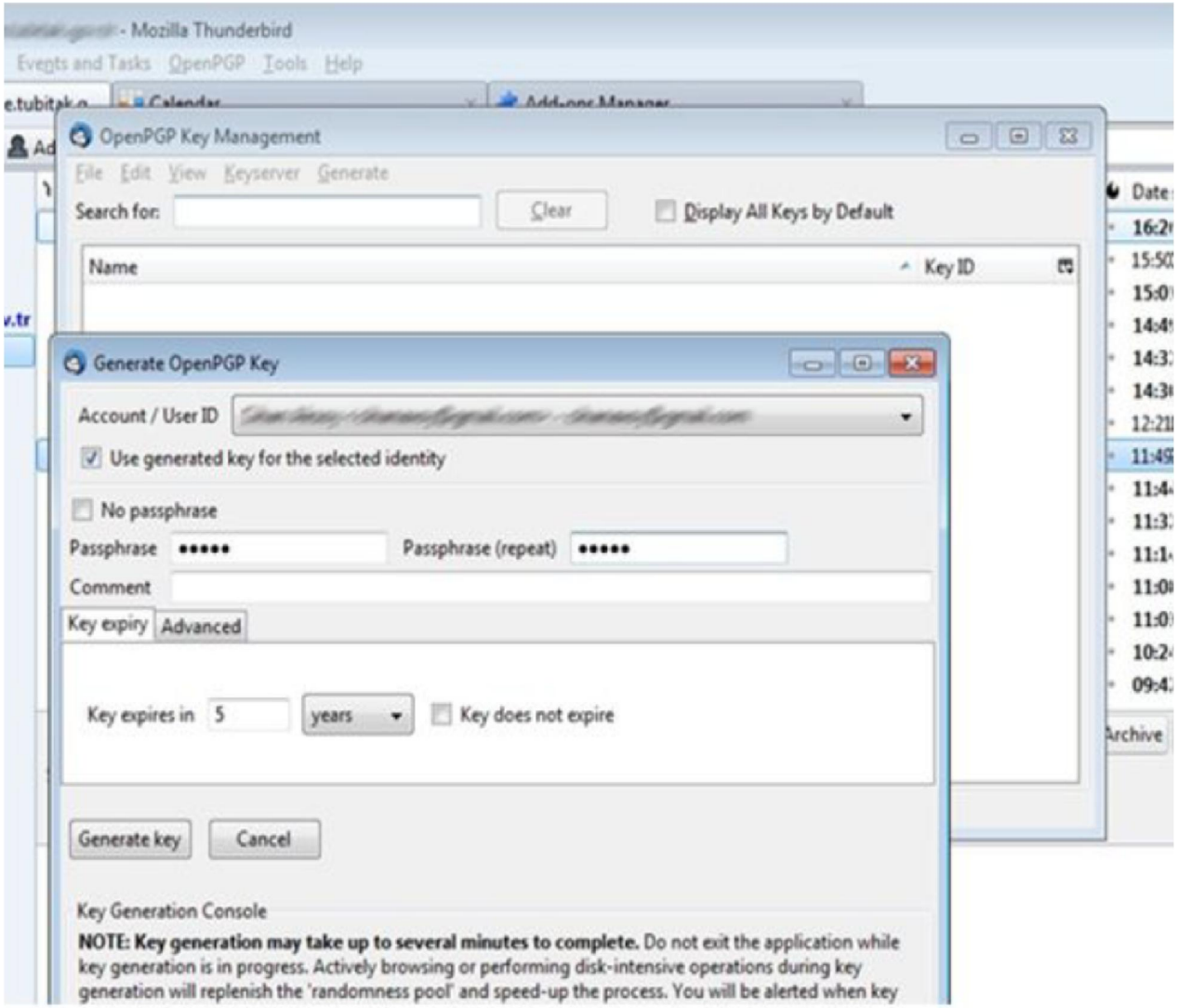
- PGP çok güvenli bir şifreleme algoritması olarak bilinir. Fakat yinede PGP yi zora sokabilecek bazı işlemler bulunmuş ve önlenilmiştir. Bunlar aşağıdadır.ı
- Mesaj şifrelenmiş, imzalanmış bile olsa internet dinlenilerek mesaj kopyalanabilir. Daha sonra bu mesajı tekrar alıcıya gönderilir. Kopyalanan mesajın üzerindeki şifre ve imza doğru olduğu için alıcı mesajın gönderenden geldiğini zannedecektir. Bu durumu engellemek için PGP’de mesajın başına **time stamp** konur. Böylece alıcı mesaj ne zaman yollanıldığı hakkında bili sahibi olarak güvenliğini sağlamış olur.
- Gönderen çok gizli bir mesajı şifreleyip yolladıktan sonra mesajın yalın hali hala bilgisayarında mevcut olacaktır. Gönderen normal olarak bu mesajı silmek isteyecektir. Fakat bilgisayarlardaki silme işlemleri geri dönüşümlü işlemlerdir. PGP bu sorunu şifrelemeden sonra mesajın içeriğini tamamen “0” ile doldurarak tan çözüm bulmaktadır.
- PGP protokolünün veri trafiğinin korunmasıyla bir ilgisi yoktur. PGP’nin amacı sadece dosya ve e-maillerin korunmasıdır.

PGP aracının kullanımı

1-İlgili WEB sitesinden freee PGP yazılımını indirin ve bilgisayara kurun.

2- Kendi imzanızı oluşturun: OpenPGP menüsünden “Key Management” seçip, En sağdaki “Generate” menüsünden kendi imzanızı oluşturabilirsiniz. İmza oluşturma safhasından sonra, bir **private**, bir **public key** oluşacaktır.

3- Public Key’inizi belirlenen key sunucularına gönderin. Public Key’inizi key sunuculara göndermek için; yine Key Management penceresinde, kendi imzanızın üzerine sağ tıklayıp, gelen menüden “Upload Public Keys to Keyserver” seçeneği ile bu işlemleri yapabilirsiniz.



Şekil 4 - Genel ve gizli şifre üretimi

Günümüzde kullanılan birçok eposta sistemi, PGP eposta şifreleme yöntemine entegre olabilmektedir.

"Enigmail" eklentisi ile Thunderbird, "CryptoAnywhere" eklentisi ile Microsoft Outlook ve bilgisayarınıza kuracağınız PGP yazılımlarıyla ücretsiz eposta hizmeti veren gmail, hotmail, yahoo gibi popüler eposta yönetim ara yüzleri ve hizmet sağlayıcıları PGP eposta şifreleme hizmeti sunmaktadır.

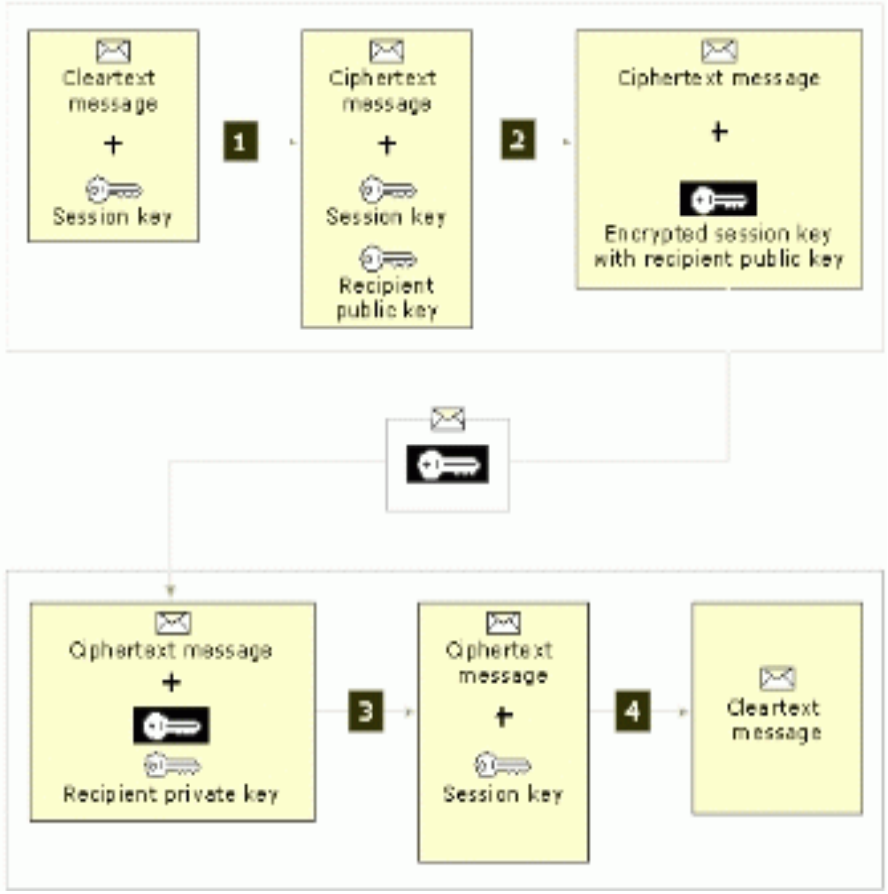
Mail amaçlı kullanılan PGP için Thunderbird mail istemci programına yukarıda bahsedilen "Enigmail" eklentisi kuruldu.

Bu eklenti anahtar yönetimi, mail şifreleme ve mail şifresi çözme yönetimlerini yapar. Ancak PGP mekanizmasını işletmez. PGP mekanizmasını işletmesi adına bir PGP sürümü bilgisayarda bulunmalıdır. Bu çalışmada "GnuPG" adlı sürüm kullanılmıştır.

Mail gönderilecek kişinin public anahtarı elde edilmiş olmalıdır.

Secure/Multipurpose Internet Mail Extension (S/MIME)

- S/MIME Internet'te güvenli mail yollamak için kullanilan bir protokoldür. S/MIME bir e-mail içeriğinin nasıl düzenlenmesi gerektiğini belirleyen standart bir formattır. S/MIME bildiğimiz mail formatına sayısal imza ve şifreleme özelliklerini eklemiştir. Sertifikanızı kullanarak güvenli e-mail alıp yollayabilmeniz için kullandığınız e-mail yazılımının S/MIME protokolünü desteklemesi gerekir.
- S/MIME, açık anahtarların sahiplerinin dogrulugunu garantileme mekanizması olarak güvenli sertifika otoritelerini (Certification Authority– CA) kullanmaktadır.

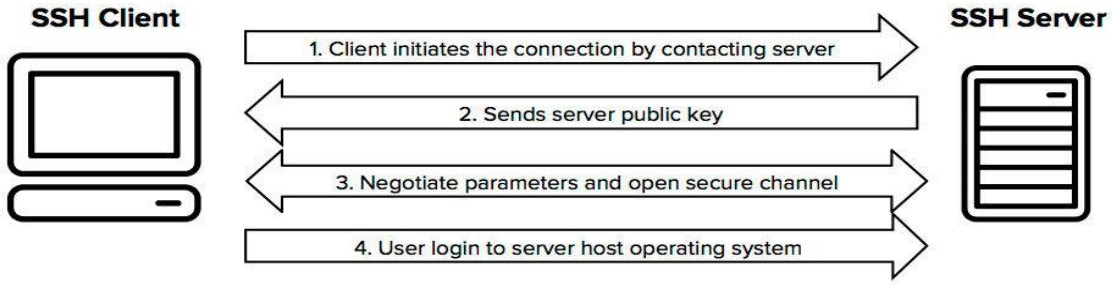


SSH (SECURE SHELL)

TELNET, PUTTY v.b uzak bir sunucuya bağlanma işlevini sağlayan küçük yazılımlar, taşıdıkları veri itibariyle güvensizdirler. Bunun için SSH, şifreli olarak işleyen bir ağ protokolü oluşturulmuştur. SSH yani "Secure SHELL" üzerinden bağlantı gerçekleştirmek istediğinizde kullanıcı adı ve şifreler açık metin olarak değil şifrelenmiş olarak iletilir.

SSH Nasıl Çalışır?

- .Ünitelerden biri SSH sunucusu diğeri ise SSH istemcisi olarak işaretlenir. Bağlantı aşamasında istemcinin uzaktaki makineye, yani SSH sunucusuna bağlanıp kimlik doğrulaması gerçekleştirmesi gerekir. Bu doğrulama aşamasında açık anahtarlı şifreleme (public key encryption) kullanılır. Onaylama sonucu kullanıcıya sistemi kullanmasına izin verilir. SSH kullanımı için otomatik olarak açık-gizli anahtar çifti üreterek ve parolayı kullanarak yetki sahibi olmak mümkün



SSH'ın Kullanımı

- SSH tünellemeyi de destekleyen bir protokoldür.. Dosya transfer protokolü (SFTP) ya da güvenli kopyalama protokolüyle (SCP) gerçekleştirilen dosya transferinde SSH istemci-sunucu modeli kullanılır. Standart TCP portu olan 22 SSH bağlantısı (standart TCP portu) için atanmıştır. SSH, cloud sistemlere bağlanma problemini kolaylaştırdığı için oldukça önemlidir. SSH tüneli sanal makinelere erişimde güvenlik duvarı (firewall) sayesinde güveli bir yol sağlayabilmektedir.