

Bilgisayar Ağları Computer Networks

Hazırlayan: M. Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Bu dersin sunumları, "James Kurose, Keith Ross, Computer Networking: A Top-Down Approach 6/e, Pearson, 2013." kitabı kullanılarak hazırlanmıştır.

İçerik

- ▶ Web ve HTTP
- ▶ FTP

Web ve HTTP

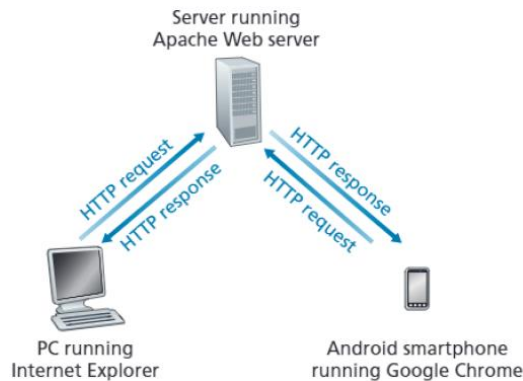
- ▶ İnternet 1990'lı yılların başında **dosya transferi, haber gönderme/alma, e-posta gönderme/alma** için kullanılıyordu.
- ▶ Web 1990'lı yılların başında geliştirildi ve İnternet'in yaygınlaşmasında en önemli aşamalardan birisidir.
- ▶ **HyperText Transfer Protocol (HTTP)**, Web'in uygulama katmanı protokolüdür.
- ▶ HTTP iki kısımdan oluşur: **istemci** ve **sunucu**.
- ▶ Bir **Web sayfası nesnelerden oluşur**.
- ▶ **Bir nesne bir dosyadır** (HTML, jpeg, Java applet, video clip).
- ▶ Tüm **nesneler URL ile adreslenir**.
- ▶ Her **URL adresi** iki kısımdan oluşur: **host adı** ve **nesne adı**.

`http://www.someSchool.edu/someDepartment/picture.gif`

3

Web ve HTTP

- ▶ **Web sunucular, nesneleri bulundurur ve istemcilere gönderir.**
- ▶ Popüler Web sunucular: **Apache** ve **Microsoft Internet Information Server**.
- ▶ HTTP, **istemci ile sunucu arasındaki iletişimi tanımlar**.



4

Web ve HTTP

- ▶ **HTTP**, ulaşım katmanında **TCP** kullanır.
- ▶ **HTTP istemci**, sunucuyla **TCP bağlantısı başlatır** (port 80, 8080).
- ▶ **İstemci** ve **sunucu process'ler TCP soket açar**.
- ▶ Bağlantı kurulduktan sonra, **istemci** ve **sunucu** kendi **TCP soket arayüzlerine erişir**.
- ▶ İstemci ve sunucu arasında **nesne aktarımı bitince soketler kapatılır**.
- ▶ HTTP, **TCP ile güvenilir iletişim yapar**.
- ▶ **HTTP** protokolü **stateless** çalışır.
- ▶ **HTTP**, **istemci process'ler hakkında geçmiş bilgisi tutmaz**.

5

Web ve HTTP

Non-persistent ve persistent bağlantı

- ▶ **Non-persistent** bağlantıda, istemci sunucu arasında **tüm nesneler farklı TCP bağlantısı** ile aktarılabilir.
- ▶ **Persistent** bağlantıda, istemci sunucu arasında **tüm nesneler tek TCP bağlantısı** ile aktarılabilir.
- ▶ **HTTP**, günümüzde varsayılan olarak **persistent bağlantı kullanır**.
- ▶ **HTTP** istemci ve sunucu **non-persistent mod ile çalıştırılabilir**.
- ▶ HTTP **non-persistent bağlantı** ile bir Web sayfasındaki **tüm nesneler için ayrı ayrı nesne alma süresi gerekir**.

6

Web ve HTTP

Non-persistent ve persistent bağlantı

- ▶ Günümüzdeki **Web tarayıcılar, 5-10 arası eş zamanlı TCP bağlantısı açar.**
- ▶ Her bağlantı ile bir request-response işlemi yönetilir.

IE 6 and 7:	2
IE 8:	6
IE 9:	6
IE 10:	8
IE 11:	8
Firefox 2:	2
Firefox 3:	6
Firefox 4 to 46:	6
Opera 9.63:	4
Opera 10:	8
Opera 11 and 12:	6
Chrome 1 and 2:	6
Chrome 3:	4
Chrome 4 to 23:	6
Safari 3 and 4:	4

7

Web ve HTTP

Non-persistent ve persistent bağlantı

- ▶ Kullanıcı aşağıdaki URL adresini girmiş olsun ve sayfada 10 JPEG referansı olsun.
`http://www.someSchool.edu/someDepartment/home.index`
- ▶ Non-persistent bağlantı ile 11 tane TCP bağlantısı açılır.

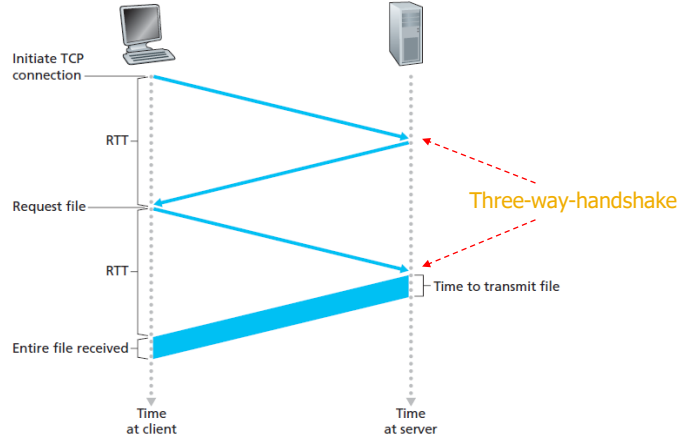
1. **HTTP istemci**, `www.someSchool.edu` HTTP sunucu ile port 80 üzerinden **TCP bağlantısını başlatır.**
2. **HTTP istemci**, açılan soket ile yol adını içeren HTTP istek mesajı gönderir. `/someDepartment/home.index`.
3. **HTTP sunucu**, gelen mesajı alır ve `/someDepartment/home.index` nesnesini encapsulate ederek gönderir.
4. **HTTP sunucu**, TCP protokolünden bağlantıyı kapatmasını ister.
5. **HTTP istemci**, mesajı alır içeriğini açar ve HTML dosyanın içinde 10 tane JPEG referansı görür.
6. **İlk dört adım 10 kez tekrar edilir.**

8

Web ve HTTP

Non-persistent ve persistent bağlantı

- ▶ **Round-trip-time (RTT)**, küçük bir paketin istemci sunucu arasında **gidip gelme süresidir**.
- ▶ **Response time** = $2 * RTT + \text{File Transmit Time}$



9

Web ve HTTP

Nonpersistent HTTP

- ▶ **Her nesne için 2 RTT süre gerekir.**
- ▶ İşletim sistemi, **her TCP bağlantısı için kaynak ayırır.**

Persistent without pipelining

- ▶ İstemci, **önceki cevabı aldıktan sonra yeni bir istek yapar.**
- ▶ **Bir RTT süresi her nesne için gerekir.**

Persistent HTTP

- ▶ Sunucu, cevap gönderdikten sonra bağlantıyı açık tutar.
- ▶ **Diğer HTTP mesajları aynı bağlantıdan gönderilir.**

Persistent with pipelining

- ▶ HTTP 1.1 de varsayılandır.
- ▶ **İstemci bir nesne ile karşılaşınca hemen istek gönderir.**
- ▶ **Bir RTT süresi art arda istenen her nesne için gerekir.**

10

Web ve HTTP

HTTP mesaj formatı

- ▶ İki tür HTTP mesajı vardır: **request** ve **response**.

HTTP request mesajı:

request line
(GET, POST, HEAD, PUT, DELETE) → GET /somedir/page.html HTTP/1.1

header lines
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr

- ▶ **GET metodunda** istenen **nesne URL'de görünür**, **POST metodunda görünmez**.

```
/test/demo_form.asp?name1=value1&name2=value2
```

```
POST /test/demo_form.asp HTTP/1.1  
Host: w3schools.com  
name1=value1&name2=value2
```

11

Web ve HTTP

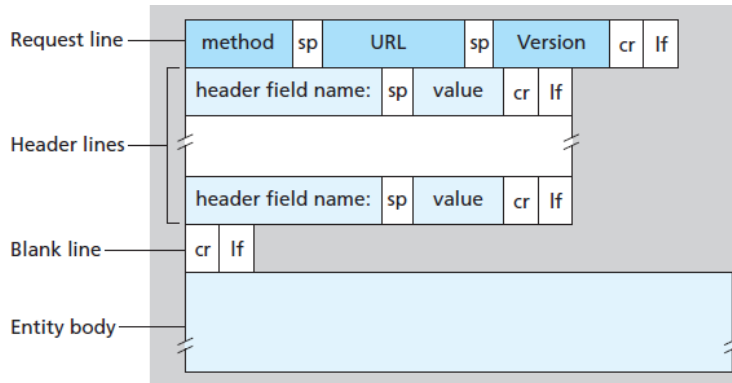
HTTP request mesajı:

- ▶ **Entity body** kısmı **GET metodunda boştur**, **POST metodunda** kullanılır ve kullanıcının girdiği kelimeler yazılır (form verileri, arama kelimeleri, vs.).
- ▶ **HEAD metodu**, geliştiriciler tarafından debug amacıyla kullanılır.
- ▶ **PUT metodu**, sunucuya nesne upload etmek için kullanılır (HTTP/1.1).
- ▶ **DELETE metodu**, sunucudan nesne silmek için kullanılır (HTTP/1.1).

12

Web ve HTTP

HTTP request mesajı:



13

Web ve HTTP

HTTP response mesajı:

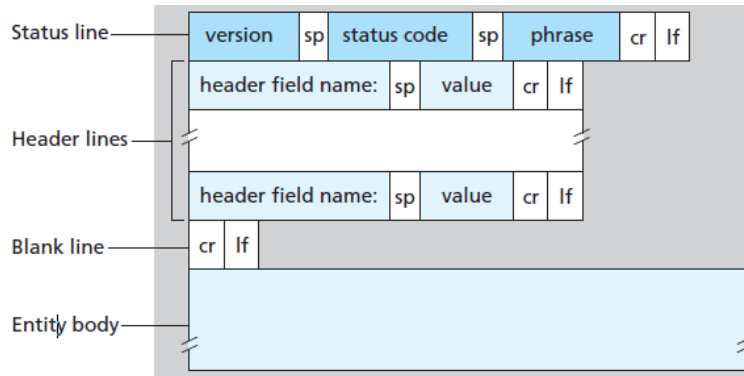
status line → HTTP/1.1 200 OK
header lines →
Connection: close
Date: Tue, 09 Aug 2011 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html
entity body → (data data data data data ...)

- ▶ **Last-modified** önbellekleme için kullanılır.
- ▶ **Entity body** kısmı gönderilen nesneyi bulundurur.

14

Web ve HTTP

HTTP response mesajı:



15

Web ve HTTP

HTTP response mesajı:

► HTTP response mesajı örnek durum kodları

200 OK

- İstek başarılı, istenen nesne mesaj ile gönderildi.

301 Moved Permanently

- İstenen nesne taşınmış, yeni URL mesajın içerisindeki başlıkta verilir (**Location: ...**).

400 Bad Request

- İstek mesajı sunucu tarafından anlaşılamadı.

404 Not Found

- İstenen doküman sunucu üzerinde bulunamadı

505 HTTP Version Not Supported

- İstek HTTP versiyonu sunucu tarafından desteklenmiyor.

16

Web ve HTTP

Cookie'ler

- ▶ **HTTP sunucu** kullanıcılar için durum geçmişi tutmadan **stateless** çalışır.
- ▶ **Web siteleri**, kullanıcı erişimini kısıtlamak veya kullanıcıya özel içerik sunabilmek amacıyla **kullanıcıları ayırt etmek ister**.
- ▶ **HTTP**, kullanıcıları takip etmek için **cookie**'leri kullanır (RFC6265).
- ▶ Günümüzdeki önemli **e-ticaret siteleri cookie'lerle kullanıcıları takip eder**.
- ▶ E-ticaret siteleri kullanıcılara özel öneri oluşturmak, kampanya sunmak için cookie'lerden faydalanabilir.

17

Web ve HTTP

Cookie'ler

- ▶ **Web tarayıcıların minimum gereksinimleri:**
 - ▶ En az **300 cookie**
 - ▶ **Her cookie için en az 4096 byte**
 - ▶ **Her domain için en az 20 cookie**
- ▶ Farklı tarayıcıların kendine özgü sınırlamaları vardır:

Browser	Cookie count limit per domain	Total size of cookies
Chrome	180	4096
Firefox	150	4097
Internet Explorer	50	5117
Opera	60	4096
Safari	600	4097

18

Web ve HTTP

Cookie'ler

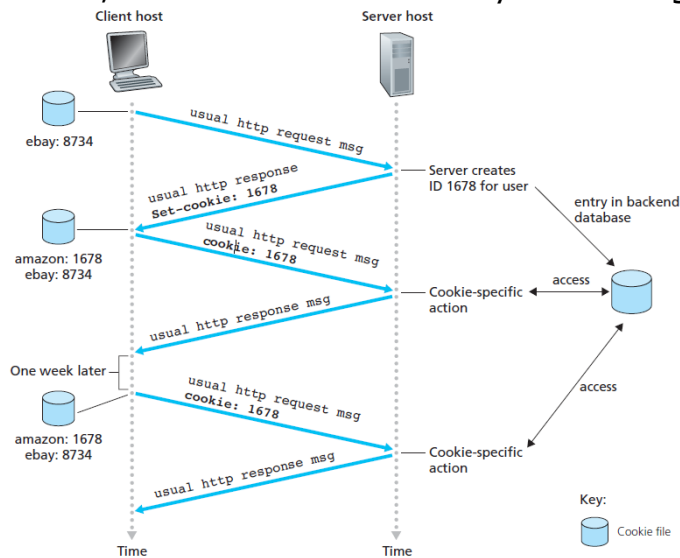
- ▶ Cookie teknolojisinin **dört bileşeni vardır**:
 - ▶ **Cookie başlık satırı** (HTTP istek mesajında)
 - ▶ **Cookie başlık satırı** (HTTP cevap mesajında)
 - ▶ **Cookie dosyası** kullanıcı bilgisayarında saklanır ve kullanıcı browser'ı tarafından kullanılır
 - ▶ Web sitesinde **back-end veritabanı**

19

Web ve HTTP

Cookie'ler

- ▶ HTTP sunucu, **set-cookie**: ile kullanıcıya cookie değeri atar.



20

Web ve HTTP

Cookie'ler

Cookie'ler ne sağlar

- ▶ Kullanıcı için **yetkilendirme** yapılabilir.
- ▶ Kişiyi özel **alışveriş kartları** düzenlenebilir.
- ▶ Kişiyi özel **kampanyalar** düzenlenebilir.
- ▶ Kişiyi özel **öneriler** sunulabilir.
- ▶ Kişiyi **özelleştirilmiş Web sayfaları** sunulabilir.

Cookie'ler ve gizlilik

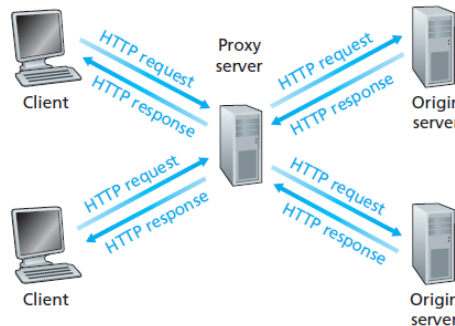
- ▶ Cookie'ler Web sitesinin **kullanıcı hakkında** çok sayıda **bilgiyi öğrenmesini sağlar.**
- ▶ Kullanıcı Web sitesine **kimlik bilgilerini** ve **e-posta** adresini verebilir.
- ▶ Üçüncü parti firmalara **bilgiler satılabilir.**
- ▶ **Reklam şirketleri** bilgi edinebilir.

21

Web ve HTTP

Web önbellekleme

- ▶ **Ağdaki HTTP isteklerini** orijinal web sunucu yerine **Web cache (proxy server)** karşılar.
- ▶ Web cache diske sahiptir ve **son alınan nesnelerin kopyasını tutar.**
- ▶ **Web cache** orijinal sunucudan **istek yaparken client, isteği karşılar**ken **server** durumundadır.

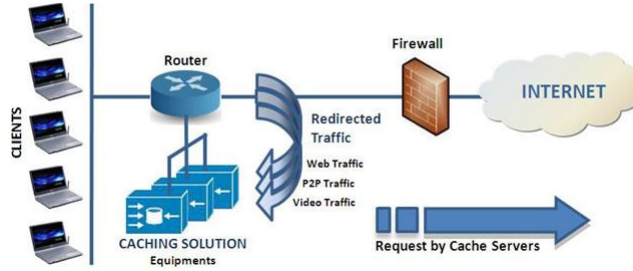


22

Web ve HTTP

Web önbellekleme

- ▶ Web cache **ISP tarafından kurulur.**
- ▶ Web cache kurulmasının iki nedeni vardır:
 - ▶ İstemcinin **cevap alma süresini çok kısaltır.**
 - ▶ Kurumsal erişimlerde **İnternet trafiğini çok azaltır.**
- ▶ Web cache **kurumsal ağlarda maliyeti düşürür.**
- ▶ Web cache tüm **İnternet'teki toplam trafiği de azaltır** ve **uygulamaların servis kalitesini artırır.**



23

Web ve HTTP

Web önbellekleme

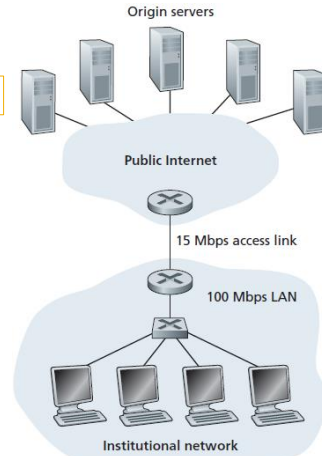
- ▶ Her **nesnenin boyutu 1Mbit** olsun ve kurum tarayıcılarından eşzamanlı ortalama **15 istek** yapılsın.
- ▶ Bu durumda **ağda ve erişim linkinde trafik olmaz.**
- ▶ **LAN** tarafındaki **trafik yoğunluğu:**

$$(15 \text{ requests/sec}) \cdot (1 \text{ Mbits/request}) / (100 \text{ Mbps}) = 0.15$$

- ▶ **Erişim** linkindeki **trafik yoğunluğu:**

$$(15 \text{ requests/sec}) \cdot (1 \text{ Mbits/request}) / (15 \text{ Mbps}) = 1$$

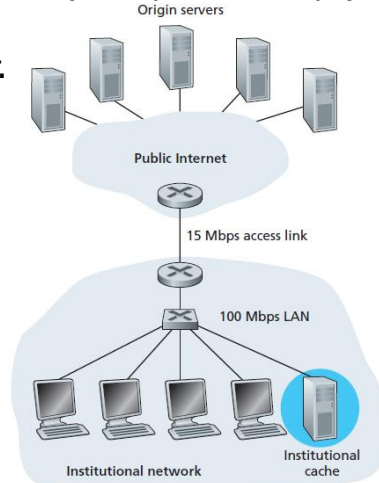
- ▶ **Trafik yoğunluğu çok artarsa?**



Web ve HTTP

Web önbellekleme

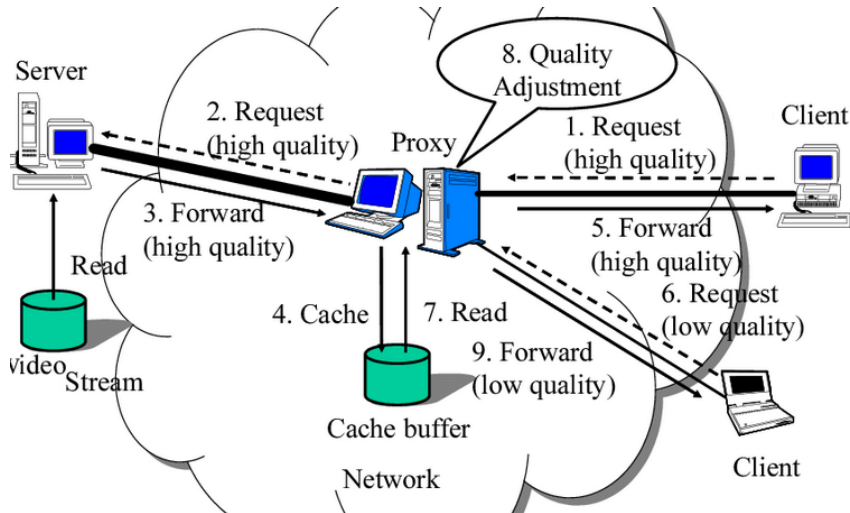
- ▶ Trafik yoğunluğu çok artarsa:
 1. **Erişim linki bant genişliği artırılır** (15Mbps->100Mbps) (Maliyeti yüksek çözümdür).
 2. Kurumsal **Web cache** oluşturulur.
- ▶ **Hit rate:** Web cache tarafından **karşılanan istek oranıdır**.
- ▶ Hit rate 0,4 olursa, isteklerin %40'ı Web cache tarafından karşılanır.
- ▶ Web cache gecikmesi= 0,01sn
Orijinal sunucu gecikmesi= 2,00sn + 0,01sn = 2,01sn
Ortalama gecikme =
 $0,4 * 0,01 + 0,6 * 2,01 = 1,21\text{sn}$



Web ve HTTP

Web önbellekleme

- ▶ Web cache **multimedya isteklerinde kaliteyi düzenleyerek** cevap oluşturabilir.



Web ve HTTP

Şartlı get

- ▶ Web cache içindeki **nesnenin güncel olup olmadığının kontrol edilmesi gerekir.**
- ▶ HTTP, şartlı get (**conditional GET**) ile güncelliği kontrol eder.

İlk istek

```
GET /fruit/kiwi.gif HTTP/1.1
Host: www.exotiquecuisine.com
```

İlk cevap

```
HTTP/1.1 200 OK
Date: Sat, 8 Oct 2011 15:39:29
Server: Apache/1.3.0 (Unix)
Last-Modified: Wed, 7 Sep 2011 09:23:24
Content-Type: image/gif

(data data data data data ...)
```

İkinci istek

```
GET /fruit/kiwi.gif HTTP/1.1
Host: www.exotiquecuisine.com
If-modified-since: Wed, 7 Sep 2011 09:23:24
```

İkinci cevap

```
HTTP/1.1 304 Not Modified
Date: Sat, 15 Oct 2011 15:39:29
Server: Apache/1.3.0 (Unix)

(empty entity body)
```

27

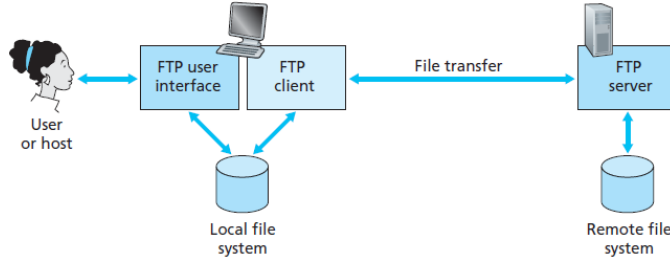
İçerik

- ▶ Web ve HTTP
- ▶ FTP

28

FTP

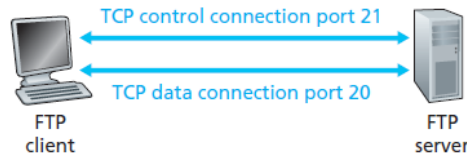
- ▶ **Kullanıcının** uzaktaki host'a erişmek için **kullanıcı adı** ve **şifreye** sahip olması gereklidir.
- ▶ Kullanıcı, user agent ile FTP iletişimi yapar.
- ▶ **FTP** protokolü **reliable iletişim yapar** ve ulaşım katmanında **TCP kullanır**.



29

FTP

- ▶ FTP dosya transfer için iki TCP bağlantısı kullanır:
 - ▶ **Kontrol bağlantısı**
 - ▶ **Veri bağlantısı**



- ▶ **Kontrol bağlantısı**, kullanıcı adı, şifre, uzak dizin komutları, "put" ve "get" dosya komutlarını iletir.
- ▶ **Veri bağlantısı**, dosya aktarımını yapar.
- ▶ **FTP**, kontrol bilgilerini ayrı TCP bağlantısı ile gönderir (**out-of-band**).
- ▶ **HTTP**, request ve response başlık satırlarını dosya transferi ile aynı TCP bağlantısı ile gönderir (**in-band**).

30

FTP

- ▶ FTP, ilk önce TCP ile sunucuya **kontrol** bağlantısını **port 21** üzerinden başlatır.
- ▶ **Sunucu** dosya transferi için bir komut aldığı anda, **istemci ile bir TCP başlatır.**
- ▶ FTP **kontrol bağlantısı tüm oturum boyunca açık tutulur.**
- ▶ FTP **veri bağlantısı her dosya transferi için yeniden oluşturulur.**

31

FTP

Örnek komutlar:

- ▶ Kontrol bağlantısı üzerinden **ASCII metin gönderilir.**
- ▶ **USER username**
- ▶ **PASS password**
- ▶ **LIST:** aktif dizindeki dosyaların listesi alınır.
- ▶ **RETR filename:** dosya alınır (get).
- ▶ **STOR filename:** uzaktaki bilgisayara dosya gönderilir (put).

Örnek return kodları

- ▶ **Durum kodu ve deyim** (HTTP'deki gibi)
- ▶ **331 Username OK, password required**
- ▶ **125 Data connection already open; transfer starting**
- ▶ **425 Can't open data connection**
- ▶ **452 Error writing file**

32

Ödev

- ▶ Python programlama diliyle, login, dosya upload/download, izin oluşturma/silme, dosya adı değiştirme ve lokal/uzak izin ve dosya listeleme gibi temel işlemlere sahip bir FTP programı geliştiriniz.