

# Computer Networks and Technologies Vize Notları

---

Bu yazı **MIT** lisanslıdır. Lisanslar hakkında bilgi almak için [buraya](#) bakmanda fayda var.

~ Yunus Emre AK ©

---

## Döküman Renklendirme Yapısı

### PDF Başlığı

---

#### Ana Başlıklar

#### Alt Başlıklar

#### İç Başlıklar

#### En İç Başlıklar

#### Tablo Başlığı

---

#### Bağlantılar

#### Değişmez ifadeler

#### Formüller

#### Önemli notlar

#### Terimsel ifadeler

#### Yorum satırları



# İçerikler

- Temel Terimler
- Network Structure (Ağ Yapısı)
- Network Edge
  - Access Network (Bağlantı Türleri)
  - Physical Media (Fiziksel Veri İşlemleri)
    - Fiber Optik Kablo
  - Radya Bağlantı
    - Satellite (Uydu Bağlantısı)
- Network Core
  - Circuit Switching (Devre Anahtarlama)
  - Packet Switching (Paket Anahtarlama)
  - Packet Yönteminin Circuit Switching Yöntemine Göre Farkı
- Internet Structure (Internet Alt yapısı)
  - Interter Tiers
  - Paketlerin iletimi
  - Packet Delay & Loss (Gecikme ve Kayıp)
  - Packet Switching Delay
  - Internet Delay
  - Protocol Layers (Protokol Katmanları)
  - Internet Protocol Stack (TCP / IP)
    - ISO / OSI Reference Model
  - Encapsulation (Kapsülleme)
- Network Security
  - Kötü Niyetli Uygulamalar
    - Trojan Horse
    - Worm
    - Virus
    - Spyware Malware
  - Kötü Niyetli Saldırıları
    - DoS
    - Packet Sniffing (Paket Yakalama)
    - IP Spoofing (IP Aldatmacası)
- Internetin Geçmişi
- Application Layer (Uygulama Katmanı)
- Network Uygulaması Oluşturmak
- Application Architectures (Uygulama Mimarileri)
  - Client-Server Architecture
  - Peer-to-Peer Architecture (Kişiden kişiye Mimarisi)
  - Client-Server ve Peer-to-Peer Karışımı
- Processes Communicating (İletişim Sistemleri)
  - Socket Yapısı
  - Addressing Processes (İşlemleri Adresleme)
- Transport Service Requirements

- Internet Transport Protocols Services (Taşıma Protokolleri Hizmetleri)
  - TCP (Transmission Control Protocol) Review
  - UDP (User Datagram Protocol) Review
  - Securing TCP (TCP'de Güvenlik)
- Web ve HTTP
- HTTP (Hypertext Transfer Protocol)
  - Temel HTTP Yapısı
  - HTTP Veri Aktarımı
  - HTTP Bağlantıları
    - Non-Persistent HTTP
    - Persistent HTTP
  - HTTP Request Message (İstek Mesajı)
  - HTTP Status Code (Durum Kodları)
  - Cookie (Çerezler)
  - Proxy Server & Cache
    - Conditional GET (Koşullu GET)
- Domain Name System (DNS)
  - DNS Resolution Examples (DNS Çözümleme Örnekleri)
  - DNS Record (DNS Kayıtları)
    - Inserting DNS Record
    - Attacking DNS
- P2P (Peer to Peer)
  - P2p File Distribution (Dosya Paylaşımı)
- Video Streaming and CDNs: context
  - Video Streamin
  - Content Distribution Networks (İçerik Dağıtım Ağları)
- Transport Layer
- Internet Transport Layer Protocols
- Multiplexing (Çoğullama)
- Demultiplexing (Azaltma / Parçalama)
  - TCP / UDP Demux
  - UDP Demux Örneği
  - TCP Demux Örneği
- UDP (User Datagram Protocol)
  - UDP Checksum
- Reliable Data Transfer (RDT)
  - Rdt 1.0
  - Rdt 2.0
    - Rdt 2.0 Kusurları
  - Rdt 2.1
  - Rdt 2.2
  - Rdt 3.0
- Pipelined Protocols
- TCP (Transmission Control Protocol)

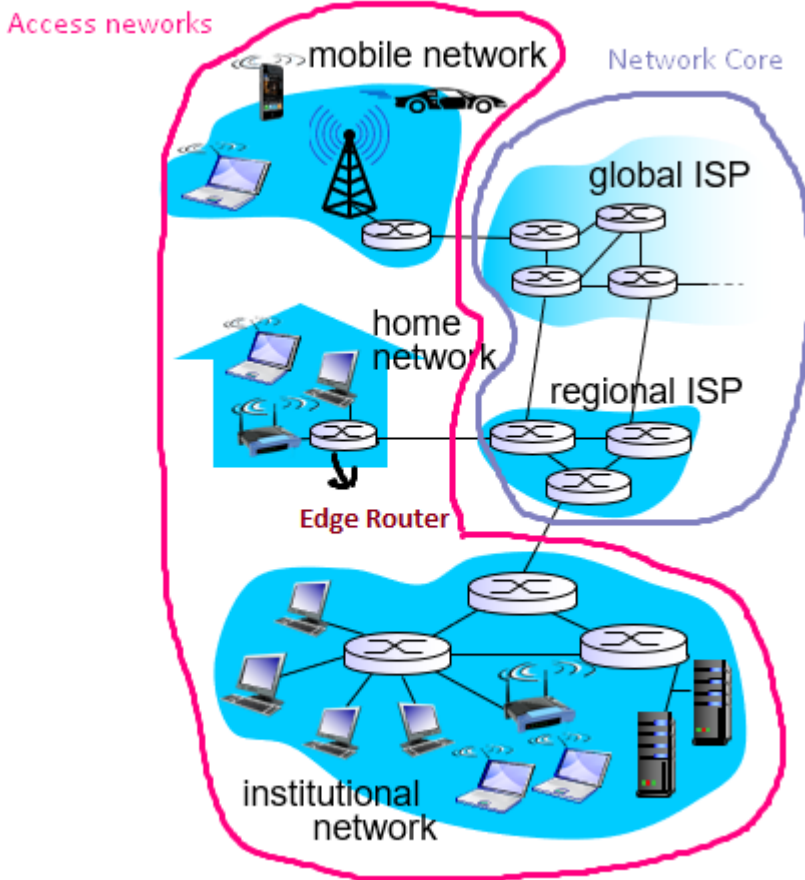
## Temel Terimler

Terim	Açıklama
ISP	İnternet servis sağlayıcıları
Packets	İnternet üzerinde gönderilen veriler
Protocols	<i>Packet</i> aktarım kuralları ve hiyerarşisi
Routers ve Switch	<i>Packet</i> 'ların yönlendirilmesini sağlarlar
Client	Ağa bağlandığımızı araç (bilgisayarımız)
Server	Ağ hizmetini sunan, sunucu
Host	End system, son <i>server</i> ya da <i>client</i>
RFS, IETF	İnternet standartları
Stream	Veri akışı
Upstream	Bizden internete <i>stream</i>
Downstream	İnternette bize <i>stream</i>
Bandwidth	Bant genişliği, saniye aktarılan bit (1sn de olan <i>streaming</i> )
Transmission rate	Saniyede aktarılan bit

## Network Structure (Ağ Yapısı)

Terim	Açıklama
Network Edge	Ağdaki uç noktaları ele alır (bilgisayarlar ve uygulamalar)
Access networks, physical media	Kablolu ve kablosuz iletişim bağlantıları
Network Core	Birbirine bağlı router'lar ve internet (network of network)

- Edge router: İnternete ilk adımın atıldığı yönlendiriciler (routers)



## Network Edge

Bizden internete olan gerçekleşen adımları ele alır.

Yöntem	Açıklama	Örnek
Hosts System	Host'lar arası iletişim	Web, email
Client / Server	Client istekte bulunur, server karşılık verir	Web browsers
Peer to peer	Neredeyse hiç server kullanılmaz	Skype, BitTorrent

## Access Network (Bağlantı Türleri)

- Dial Up: Telefon çalışırken modem, modem çalışırken telefonun çalışmadığı eski bir sistem.
- DSL: *Splitter* ile telefon ve internet eş zamanlı kullanabilmekte.
  - ADSL: Asimetrik anlamındadır, download ve upload hızı farklı olur.
- Wireless LAN: Ev içerisindeki kablosuz ağlar: WiFi
- Wide-Area wireless acces: Mobil operatörler tarafından sunulan ağlar: 3G, 4G, LTE

## Physcial Media (Fiziksel Veri İşlemleri)

Fiziksel verilerin (*bit*'lerin) aktarılmasını ele alır.

- Kablo yapısı TP (twisted pair) iç içe sarmal 2 kablodur.
- Guided: yönetimli (kablo vs ile), unguided: dağınık olarak yayılan (radyo dalgaları) verilerdir.

## Fiber Optik Kablo

- Cam içerisinde bilgiler ışık yoluyla aktarılır
- Işığın farklı frekanslarıyla birden fazla bilgi yollanabilir
- Işık hızıyla iletilir
- Elektromanyetik gürültüden etkilenmez
- Veri kaybı çok düşüktür

## Radya Bağlantı

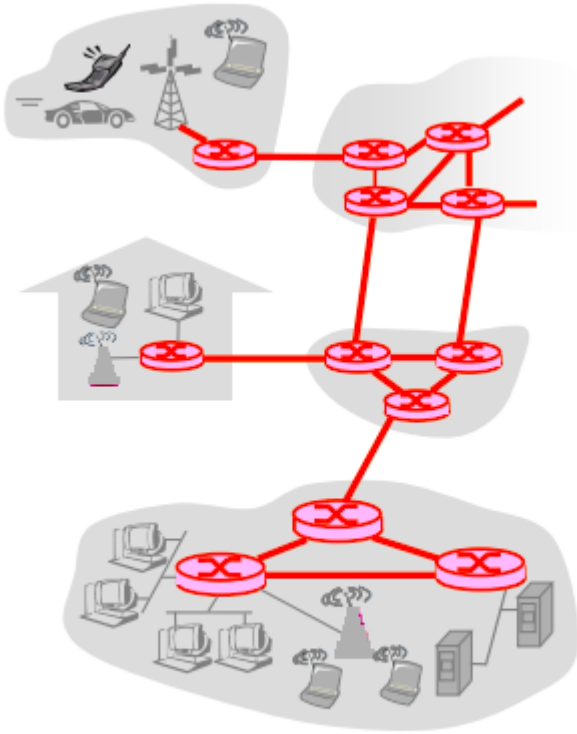
- LAN (WiFi)
- Wide-area (geniş alan bağlantıları) 3G, 4G

## Satellite (Uydu Bağlantısı)

- Gecikmesi çok fazladır. (250ms)

## Network Core

Birbirine bağılı çok sayıda *router*'dan oluşur. Network of network olarak da tabir edilen interneti ele alır.



Aktarım Yöntemi	Açıklama
-----------------	----------

Circuit Switching	Her arama için özel devre kullanılır, telefon ağı gibi
-------------------	--

Packet Switching	Veri ağı ayrı <i>packet</i> 'lar halinde gönderilir
------------------	---

## Circuit Switching (Devre Anahtarlama)

*Bandwidth* parçalara bölünür, her parça sadece kendi sahibi tarafından kullanılır.

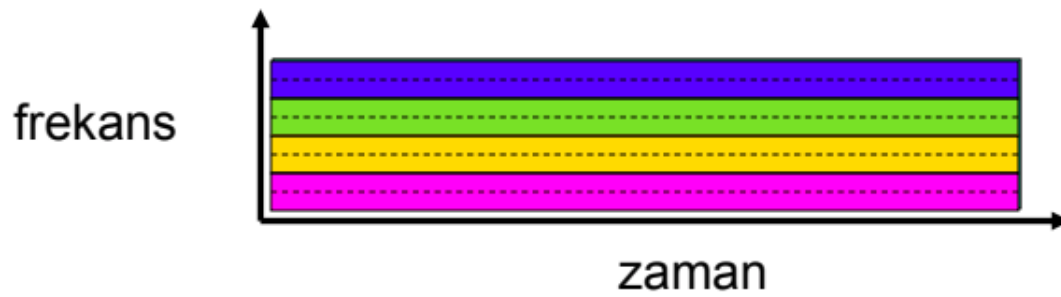
- Genellikle telefon hatlarında kullanılır
- Garantili performans sunar
- Kaynaklar paylaşılmaz, kullanılmayanlar boşta bekler (verimsiz)
- Frekans ve Zaman bölme olarak iki yöntemi vardır. (FDM, TDM)

# Devre anahtarlama: FDM ve TDM

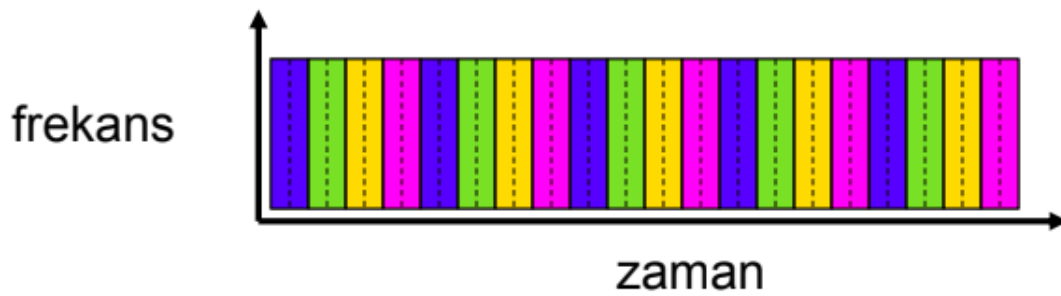
Örnek:

4 kullanıcı ■ ■ ■ ■

FDM



TDM

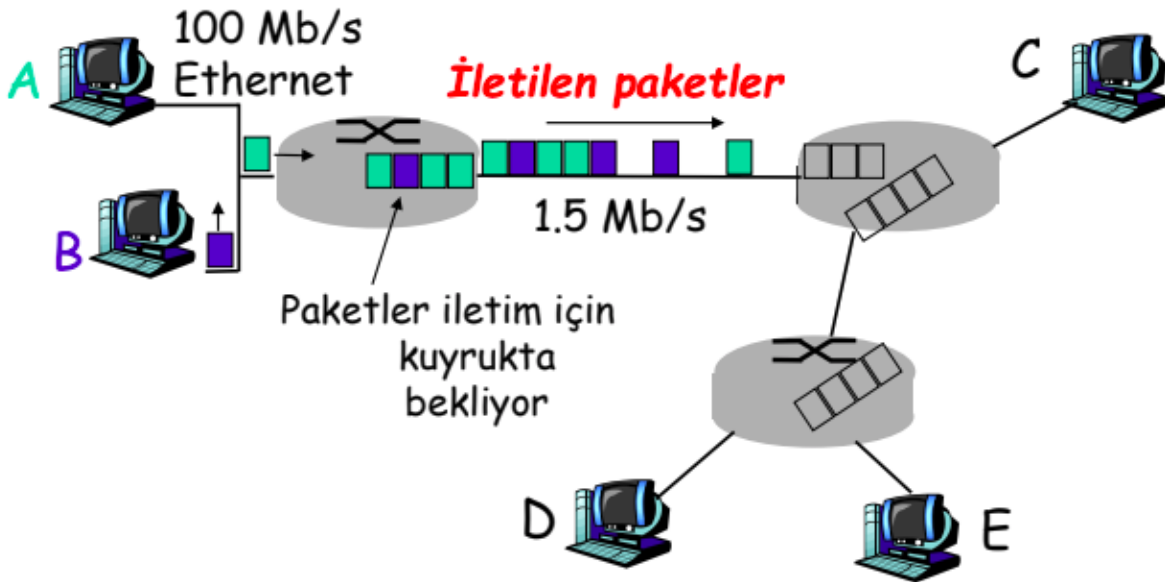




## Packet Switching (Paket Anahtarlama)

Hostlar çok yüksek miktarda gelen veriyi parçalayarak yollarlar, her bir parçaya **packets** denir. Her bir *packet* tam *bandwidth* kullanır ve host tarafından **tamamlanmadan** yollanmaz (storage & forward).

- *Packet*'ların bir sırası yoktur
- Her bilgisayar *packet* iletimi için aynı yolu kullanır
- Kaynaklar boşta kalmaz. (verimli)
- Her bir *packet*  $L$  kadar bit içeriyor ve *transmission rate*  $R$  ise *transmission delay*  $D = L/R$  formülü ile bulunur
- Kaynak çekişmesi olabilir. (olumsuz)
  - Toplam kaynak talebi kullanılamaz aşabilir
  - Trafik sıkışıklığı, *packet*'in kuyruğu ve bağlatıyı kullanmak için beklemesi
  - *Packet*'lar aynı anda bir yönlendiriciye iletilir
    - Buffer'ı yetmezse *packet* kaybı olur



## Packet Yönteminin Circuit Switching Yöntemine Göre Farkı

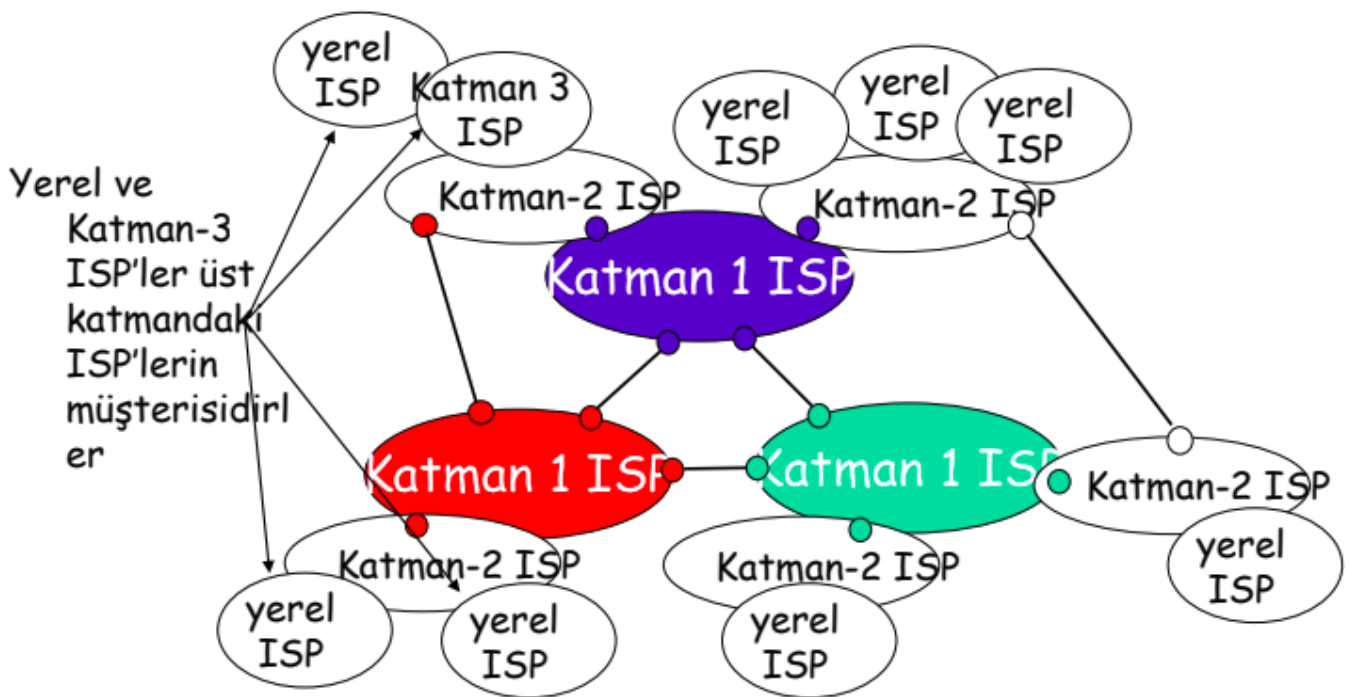
- Basit, arama algoritmalarının kurulmasına gerek yoktur
- Kaynaklar paylaşıldığından ağı daha fazla kullanıcı kullanabilir
- Güvenilir veri transferi ve sıkışıklık için protokellere ihtiyaç vardır.
  - Yoksa verilerinizi çalarlar 😞

## Internet Structure (Internet Alt yapısı)

### Interter Tiers

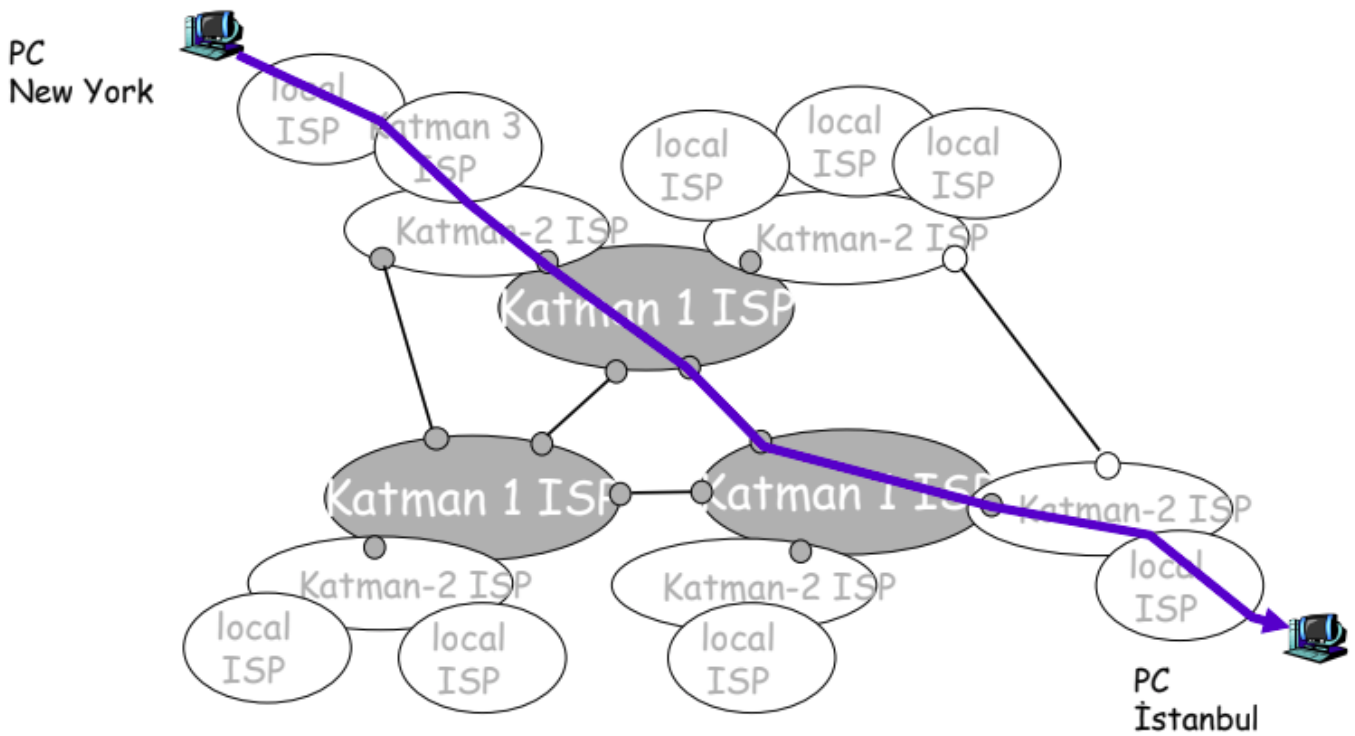
Her bir katman üst katmanının müşterisidir.

Tier (Katman)	Açıklama
Tier-1	Global ISP evrensel servis sağlayıcılarıdır. Birbirlerine bağlıdır Ör: Superonline, TTNNet
Tier-2	Regional ISP bölgesel servis sağlayıcılarıdır. Birbirlerine değil Tier-1'e bağlıdır
Tier-3	Son kullanıcı ağlarıdır, Tier-2'e bağlanırlar



## Paketlerin iletimi

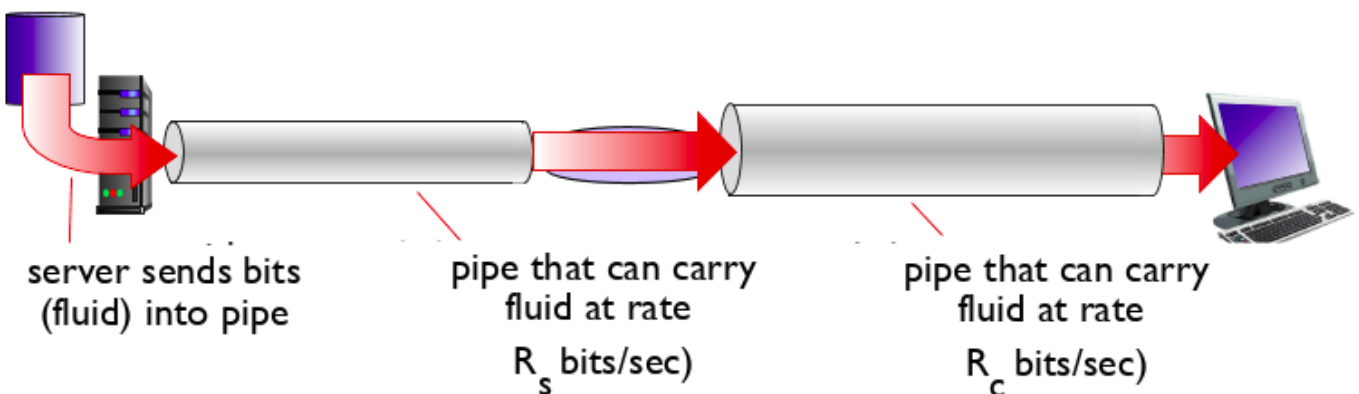
Paketler *tier-3*'ten *tier-1*'e ardından hedef *tier-3*'e doğru yol izlerler.



- Router'lar arası verilerin yayıldığı alana **pipe** denir
- Kalın bağlantılarda (links) veri aktarımı daha fazladır
- İnce alanlara **bottleneck link** denir

❖ **throughput:** rate (bits/time unit) at which bits transferred between sender/receiver

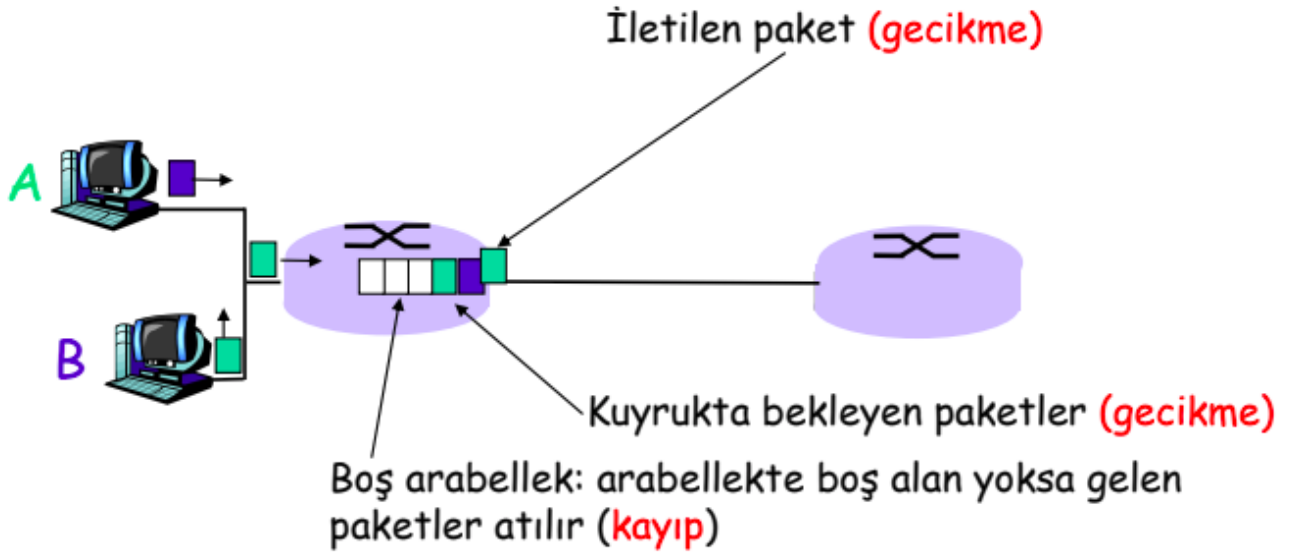
- **instantaneous:** rate at given point in time
- **average:** rate over longer period of time



## Packet Delay & Loss (Gecikme ve Kayıp)

*Packet*'lar *router*'ın *buffer* (arabellek) alanında kuytuktaki beklerler

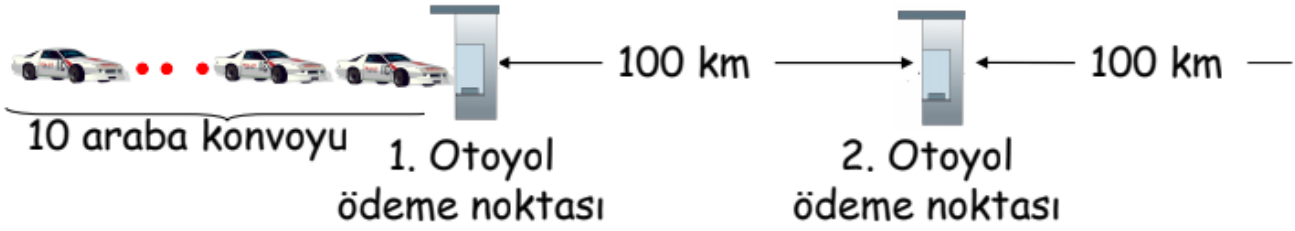
- Gelen *packet* sayısı çıkandan fazla ise fazlalık *packet*'lar *buffer*'a konulur
- *Buffer* yeterli alana sahip değilse *packet* atılır, kayıp *packet*'lar önceki *node*'dan tekrar istenir



## Packet Switching Delay

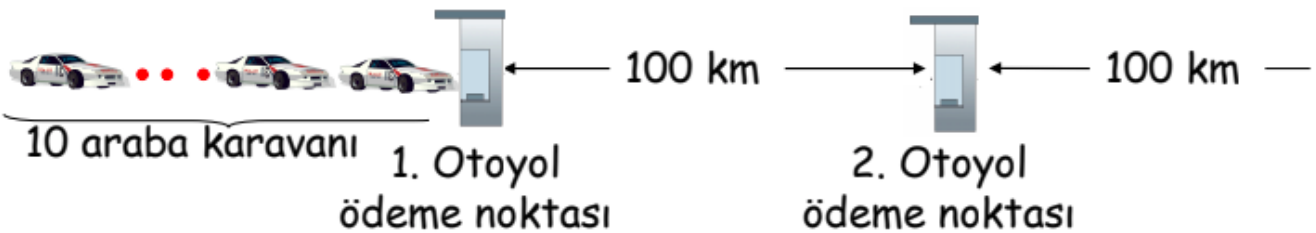
Olay	Açıklama
Nodel Processing	Hatalı bitlerin kontrol edildiği aşama
Queueing Delay	<i>Buffer</i> 'da sıralanmanın olduğu aşama
Transmisson Delay	Yayılım için <i>packet</i> 'ların <i>router</i> 'a iletilmesi
Propagation Delay	<i>Router</i> 'daki paketlerin yayılması

# Konvoy benzetmesi



- Arabaların hızı 100 km/saat
- Bir arabanın geçişi 12 saniye alıyor (iletim zamanı)
- araba~bit; konvoy~ paket
- **S: konvoyun 2. otoyol ödeme noktası önüne dizilmesi ne kadar zaman alır?**
- Tüm konvoyun 1. otoyol ödeme noktasından geçmesi toplam =  $12 \times 10 = 120$  saniye alır
- Son arabanın 2. otoyol ödeme noktasına varması:  $100 \text{ km} / (100 \text{ km/saat}) = 1$  saat
- **C: 62 dakika**

# Konvoy benzetmesi (devamı)



- **"Propagation"**
- Şimdi arabalar 1000 km/saat hızla gidiyorlar
- **"Transmission"**
- Ödeme noktasında bir araba için 1 dakika zaman harcanıyor
- **Q: bütün arabaların ödemeleri bitmeden 2. ödeme noktasına araba ulaşır mı?**
- **Evet!** 7 dakika sonra, 1. araba 2. ödeme noktasında ve 3 araba hala 1. ödeme noktasındadır.
- Paket 1. router'dan tam iletilmeden paketin 1. biti 2. router'a ulaşabilir!

## Internet Delay

Traceroute programı kaynaktan hedefe yol üzerinde bulunan *router*'lardaki gecikmenin ölçümünü sağlar.

- Windows için tracert
- Linux için tracepath

```
yemre@yemre-PC: ~  
File Edit View Search Terminal Help  
yemre@yemre-PC:~$ tracepath  
Usage: tracepath [-4] [-6] [-n] [-b] [-l <len>] [-p port] <destination>  
yemre@yemre-PC:~$ tracepath google.com  
1?: [LOCALHOST] pmtu 1500  
1: csp1.zte.com.cn 0.653ms  
1: csp3.zte.com.cn 0.613ms  
2: csp1.zte.com.cn 0.710ms pmtu 1492  
2: host-92-45-0-231.reverse.superonline.net 1.388ms  
3: 10.36.79.225 5.859ms  
4: 10.40.132.78 5.336ms asymm 6  
5: 10.36.6.37 5.963ms asymm 6  
6: 10.38.218.170 4.345ms  
7: no reply | Buffer*'da sıralanmanın olduğu aşama |  
8: no reply | Yayılım için packet*'ların router*'a iletilmesi |  
9: no reply | Router*'daki paketlerin yayılması |  
10: no reply  
11: no reply  
12: no reply  
13: no reply  
14: no reply  
15: no reply  
16: no reply  
17: no reply  
18: no reply
```

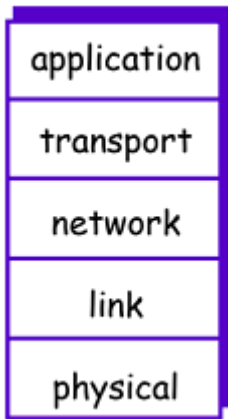
## Protocol Layers (Protokol Katmanları)

Ağ yapıları karmaşıktır. Bilgisayarlar, *routers*, *protocols* ... Katman yapısıyla:

- Karmaşık sistem parçalarının ilişkilerini tanımlamaya olanak sağlar
- Modüler olması sistemin bakımını ve güncelleştirilmesini kolaylaştırır
  - Bir katmandaki servis uygulamasını değiştirmek, sistemi etkilemez

## Internet Protocol Stack (TCP / IP)

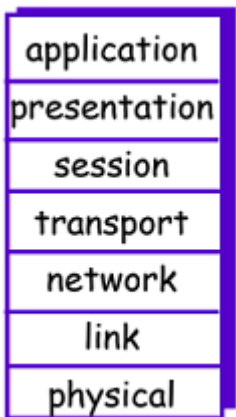
Öge	Açıklama
application	Ağ uygulamalarını destekleyen uygulamalar
transport	Veri aktarımı, TCP, UDP
network	Kaynaktan hedefe <i>datagram</i> 'ları yönlendirir: IP, yönlendirme protokolleri
link	Komşu ağ elemanları arasında veri transferi: PPP, Ethernet
physical	Hattaki bitler ( <i>bits in wire</i> )



## ISO / OSI Reference Model

Internet protocol *stack*'te bu katmanlar yoktur, gerekirse program ile uygulanır

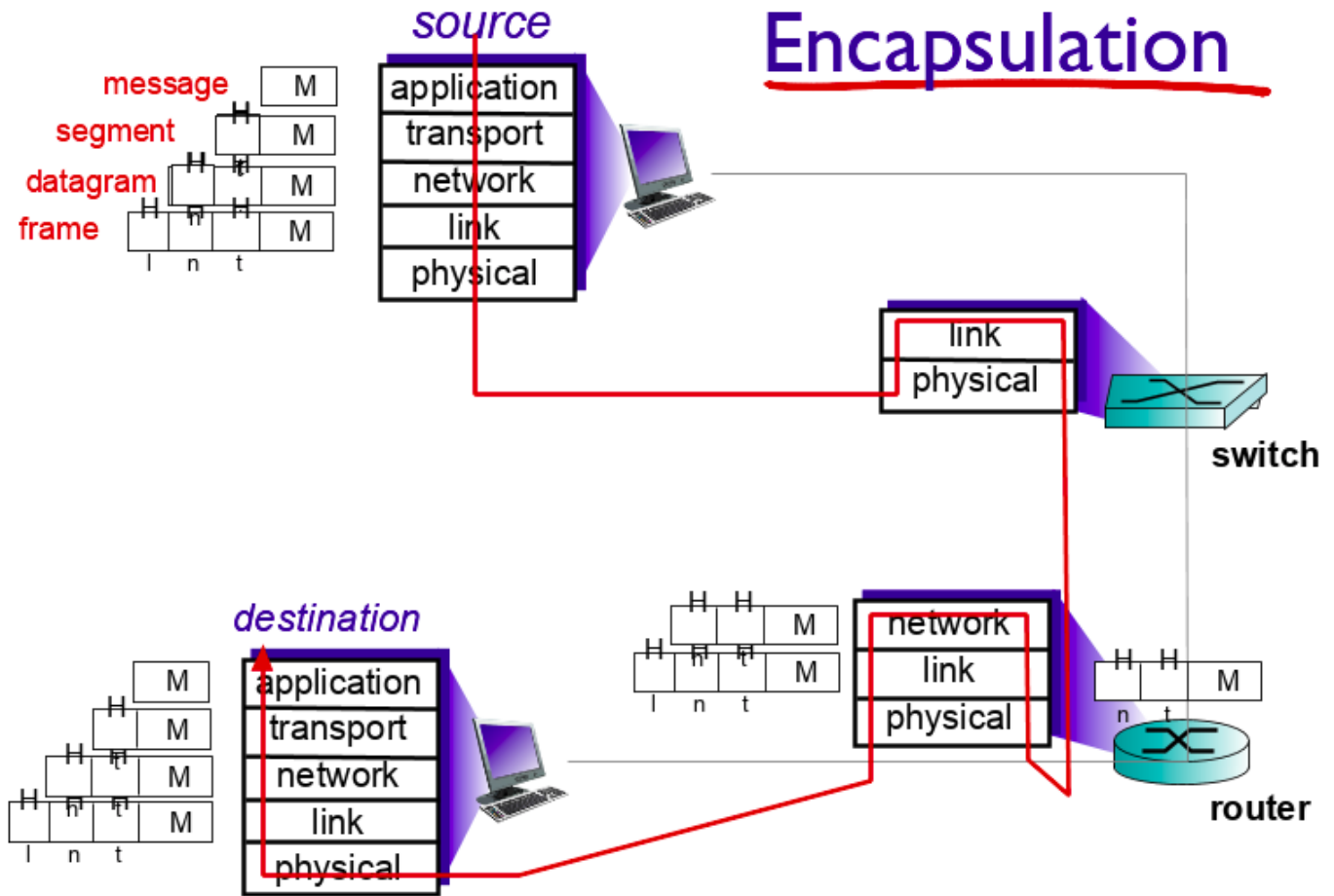
Ek Öge	Açıklama
presentation	Uygulamaların verilerin anlamlarını yorumlamasını sağlar: <i>encryption, compression</i> ...
session	Senkronizasyon, denetim veri değişimi ...



## Encapsulation (Kapsülleme)

Veri transferleri *encapsulation* ile yapılmaktadır.

## Encapsulation





# Network Security

Hiçbir *protocol* güvenlik tedbirleri barındırmaz. 🤖

## Kötü Niyetli Uygulamalar

### Trojan Horse

Faydalı yazılımların gizli bir parçasıdır, web sayfalarında bulunur. (Active-x, plugin)

### Worm

Pasif olarak alınan nesnenin kendini çalıştırması ile bulaşır, çoğalır diğer bilgisayarlara da yayılır.

### Virus

Alınan nesne ile bulaşır (e-posta). Nesne açıldığında *virus* bulaşır, çoğalır diğer bilgisayarlara da yayılır.

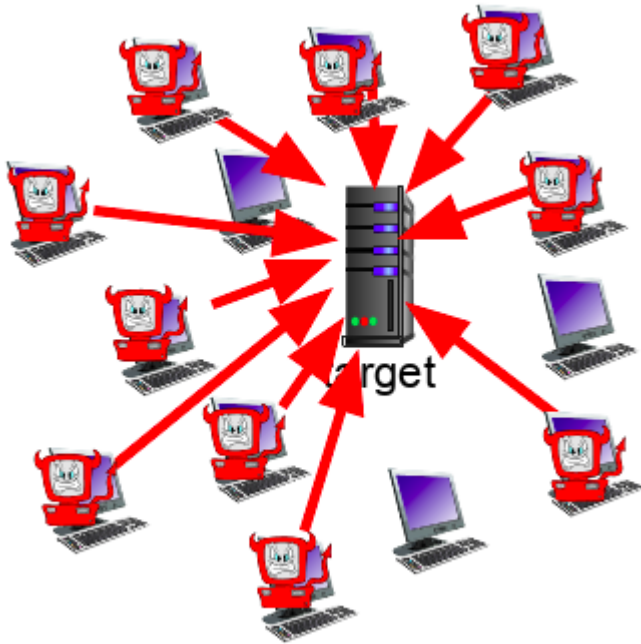
### Spyware Malware

Casus yazılımlar olarak da bilinir. Klavye tuş basımlarını ve girdiğimiz web sitelerinin bilgilerini çalar.

## Kötü Niyetli Saldırıları

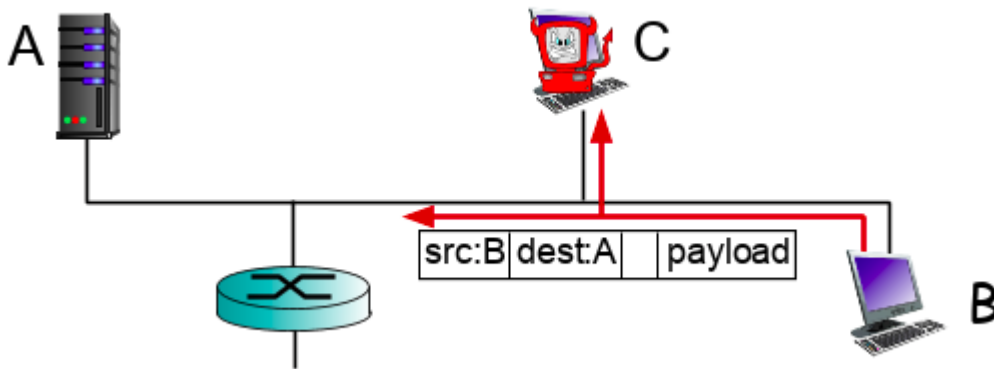
### DoS

Denial of service olarak da bilinir. Saldırganların kaynağa çok fazla *packet* göndererek erişim dışı bırakmasıdır.



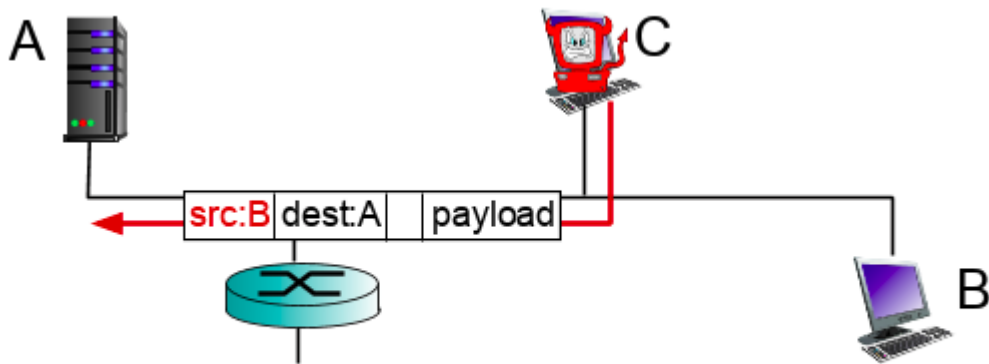
## Packet Sniffing (Paket Yakalama)

Yerel ağa bağlı bir ağ kartından *Wireshark* uygulaması ile başka *packet*'lar de yakalanır.



## IP Spoofing (IP Aldatmacası)

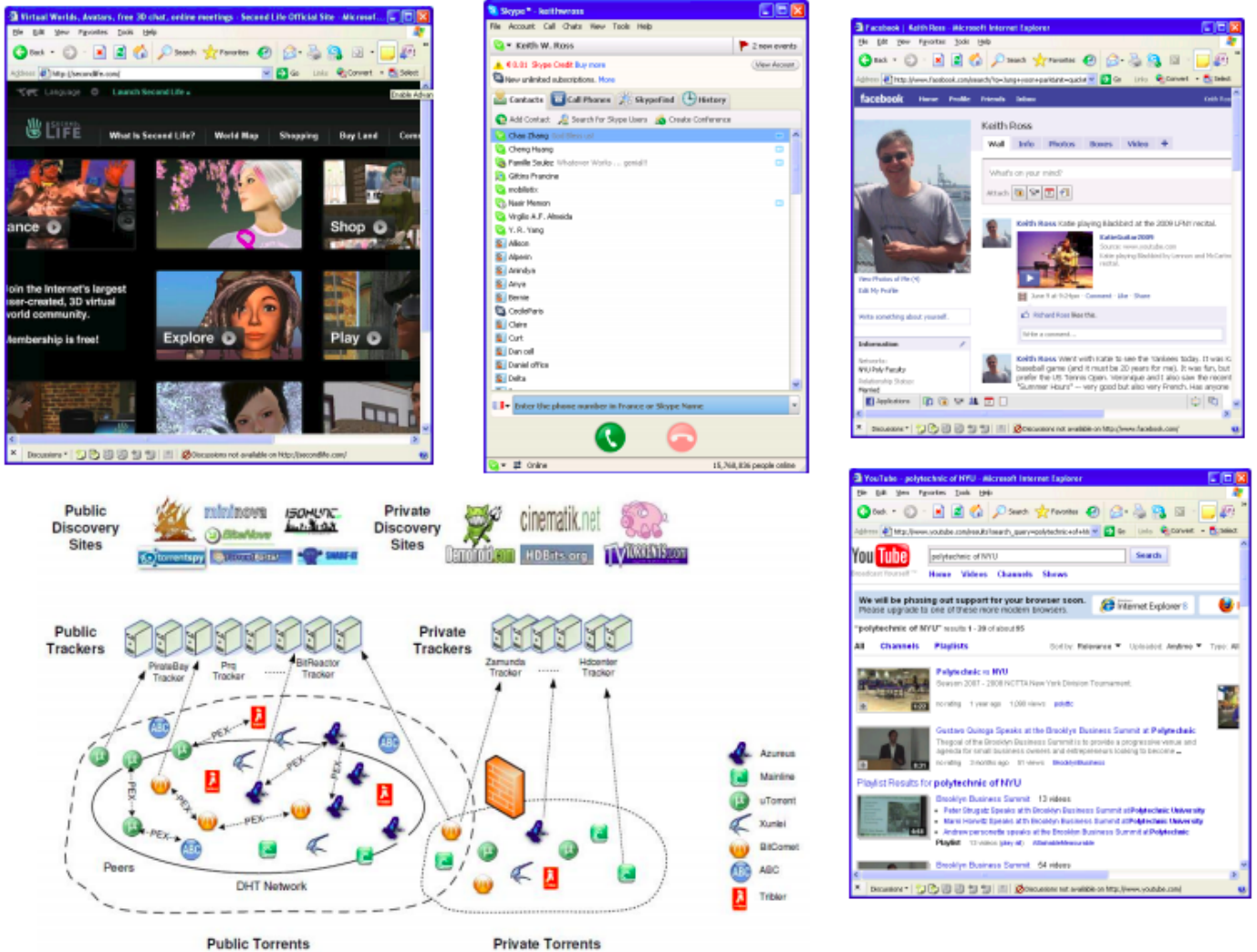
Yanlış IP adresiyle *packet* gönderilir



## İnternetin Geçmişi

Buradaki slaytın 62. sayfasına bakarak erişebilirsin.

## Application Layer (Uygulama Katmanı)



## Network Uygulaması Oluşturmak

Farklı *end systems* (son kullanıcı sistemleri) üzerinde çalıştırılır. Örneğin, web server yazılımı ağ üzerinden web browser yazılımı ile bağlantı kurar

Temel ağ cihazları, kullanıcı programlarını çalıştırmaz. ☹️

## Application Architectures (Uygulama Mimarileri)

2 farklı yapı kullanılmaktadır; client-server, peer-to-peer (P2P)

### Client-Server Architecture

Server Özellikleri:

- Her zaman açık
- *Static IP* (değişmeyen, kalıcı IP)

Client özellikleri:

- Belirlenen zamanlarda *server* ile iletişim kurabilirler
- *Dynamic IP*
- Birbiri ile iletişim kuramazlar

### Pear-to-Pear Architecture (Kişiden kişiye Mimarisi)

- *Server* her zaman açık değildir
- Rastgele *end system*'lerle doğrudan iletişim olur (arada *server* olmaz)
- Bilgisayarlar zaman zaman bağlanabilir ve *dynamic IP* kullanabilirler

Yönetmesi oldukça zordur.

### Client-Server ve Peer-to-Peer Karışımı

#### Skype

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

#### Anlık mesajlaşma (Instant messaging)

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

## Processes Communicating (İletişim Sistemleri)

İşlem	Açıklama
Process	Bir bilgisayarda çalışan programlar, yapılan işlemler
Client Process	İletişimi başlatan <i>process</i> 'ler
Server Process	Bağlantıyı bekleyen <i>process</i> 'ler

### Socket Yapısı

*IP* adresi ve *port* numarasından oluşan, *process*'lerin alınıp / verildiği kısma **socket** adı verilir.

- *Client*, *process*'i kapının dışına koyar.
- *Server*, *process*'i kapıdan içeri alır
- Buradaki kapı olarak adlandırılan *socket*'tir

### Adressing Processes (İşlemleri Adresleme)

Mesajların alınması için *process*'in bir tanımlayıcısı (*identifier*) olması gerekmektedir. 0 ile 1023 arası *well-known ports* olarak bilinmektedir. Tanımlayıcı:

- *IP* adresi, örn: 128.119.245.12
- *Port* numarası, örn: 80

içerir.

Örnek *port* numaraları:

- *HTTP* server: 80
- *Mail* server: 25

Windows için **ipconfig**, linux için **ifconfig** ile IP adresinizi öğrenebilirsiniz.

## Transport Service Requirements

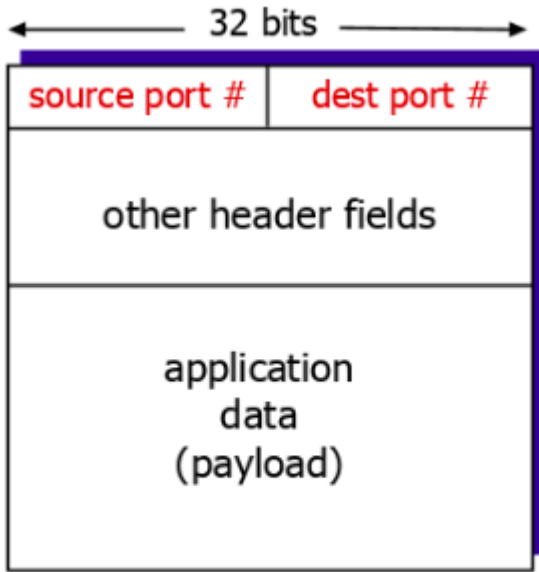
Özellik	Açıklama
Data Integrity	Metin aktarımlarında çok önemlidir, ses gibi verilerin aktarılmasında önemsizdir
Timing	Ses aktarımlarında gecikmenin en az olması gereklidir.
Throughput	Multimedya uygulamaları etkili olmak için daha az veri kullanmayı tercihi eder
Security	Şifreleme ve verinin değiştirilmemesini ele alır

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

## Internet Transport Protocols Services (Taşıma Protokolleri Hizmetleri)

*Protocol*'lerin hiç biri alttaki özellikleri taşımaz, sonradan bunlara uygun sistemler oluşturulur ve entegre edilir.

- Timing (düşük gecikme)
- Min throughput (düşük veri aktarımı)
- Guarantee (garantili taşıma)
- Security (güvenli taşıma)
  - Şifreleme (*encryption*) içermez
  - Socket ve internet verileri olduğu gibi (*cleartext*) gönderilir.



TCP/UDP segment format

## TCP (Transmission Control Protocol) Review

Özellik	Açıklama
Reliable transport	Güvenilir veri aktarımı
Flow control	Veri akışı denetimi
Congestion control	Network aşırı yoğun olduğunda veri akışını azaltır

Detayları *transport layer* altında işlenmekte, [buraya](#) tıklayarak gidebilirsiniz.

## UDP (User Datagram Protocol) Review

UDP yayıncılıkta tercih edilen bir *protocol*'dür. Amacı tamamıyla hızı arttırmak ve maaliyeti düşürmektir.

- *Packet*'in varıp, varmadığıyla ve güvenliğiyle ilgilenmez (*Unreliable transport*), varmazsa tekrar gönderir.
- Tıkanıklık kontrolüne (*congestion control*) ihtiyaç yoktur, olabildiğince hızlı gönderir
- Bağlantı kurmaya gerek yok, zaman kaybına neden olur
- Basitir, *sender* ve *reciver* asla birbiriyle iletişimde değildir
- Olumsuz geri dönüş yoktur.

Detayları *transport layer* altında işlenmekte, [buraya](#) tıklayarak gidebilirsiniz.

## Securing TCP (TCP'de Güvenlik)

TCP'de güvenlik SSL ile sağlanır, uygulamalar **SSL kütüphanesi** yardımıyla TCP ile etkileşir. SSL'in sağladıkları:

- Şifreli (*encrypted*) TCP bağlantısı
- Veri bütünlüğü (*data integrity*)
- Uç sistem doğrulaması (*end-point authentication*)

## Web ve HTTP

- Web sayfası *base HTML* dosyasının referans ettiği objelerden oluşur.
- Web sayfaları objelerden oluşur, bu dosyalar; HTML, JPEG, JAVA applet vs. olabilir.
- Her obje *URL*'ler ile adreslenir.

www.someschool.edu/someDept/pic.gif

host name                      path name



# HTTP (Hypertext Transfer Protocol)

## Temel HTTP Yapısı

*Applicataion Layer* (uygulama katmanı) *protocol*'üdür.



- *Client*: Tarayıcılar, *Server*: Apache Web Server

## HTTP Veri Aktarımı

HTTP, TCP kullanır.

- *Client* TCP bağlantısını başlatır.
  - *Server*'a 80 *port*'unda *socket* oluşturur
- *Server* TCP bağlantısını kabul eder
- *Client* ve *Server* arasında HTTP mesajları aktarılır
- TCP bağlantısı kapatılır

HTTP *stateless* (durumsuz) olarak tanımlanır. Eski istekler (*requests*) hakkında bilgi sahibi değildir.

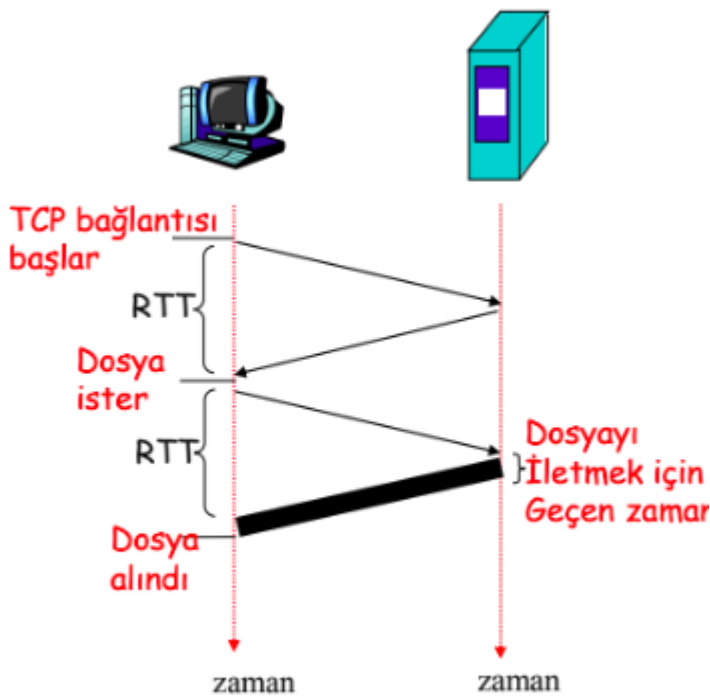
## HTTP Bağlantıları

Bağlantı Türü	Açıklama
non-persistent (kalıcı olmayan)	En fazla bir obje TCP üzerinden gönderilir ardından bağlantı kapatılır
persistent (kalıcı)	Çok sayıda obje TCP üzerinden gönderilebilir

**RTT**, bir *packet*'in *client-server* arasında gidiş geliş süresi

### Non-Persistent HTTP

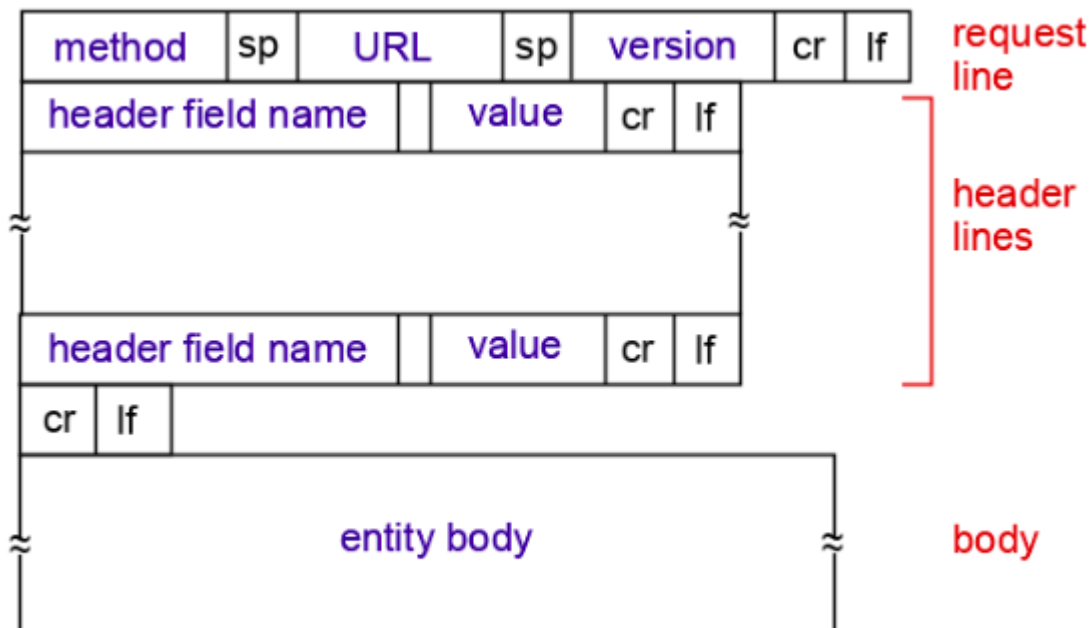
Sunucuyu her defasında açmak için *RTT* kaybı yaşanacaktır, tek bir veri alınacaksa ideal seçimdir



### Persistent HTTP

- Sunucu tek bir seferde açılacak lakin kapatılmak için *request* bekleyecektir, bu da fazladan *RTT* kaybı demektir.

## HTTP Request Message (İstek Mesajı)



- **sp**: Boşluk
- **cr**: \r karakteri
- **lf**: \n, satır sonu karakteri

İstek satır  
(GET, POST,  
HEAD komutları)

Başlık  
satırları

satırbaşı,  
boş satır  
mesajın bittiğini  
gösterir

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

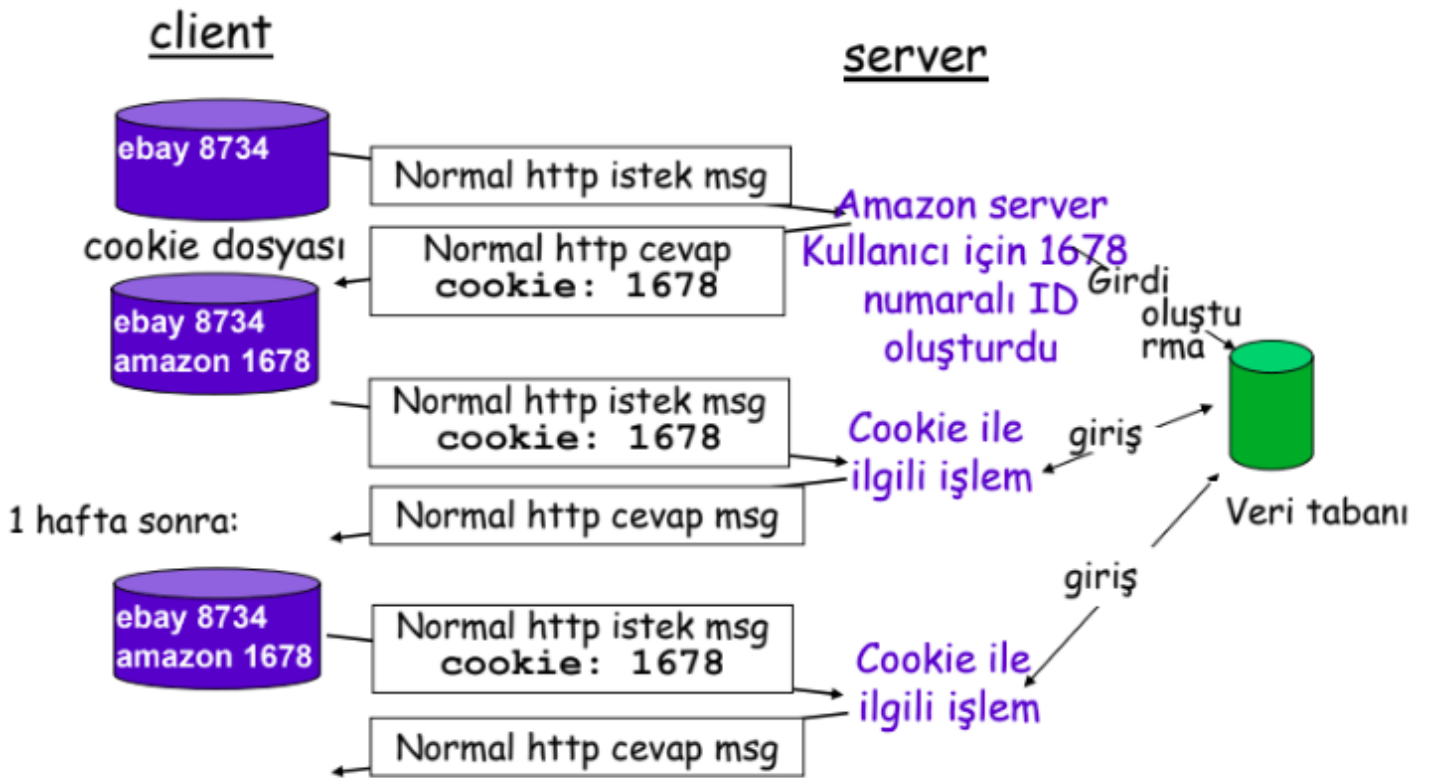
## HTTP Status Code (Durum Kodları)

Status Code	Açıklama
200	OK
301	Moved Permanently
400	Bad Request
404	Not Found
505	HTTP Version not Supported

## Cookie (Çerezler)

Bir websitesine ilk kez girdiğimizde bilgilerimiz **cookie** adıyla *server* veri tabanında saklanır.

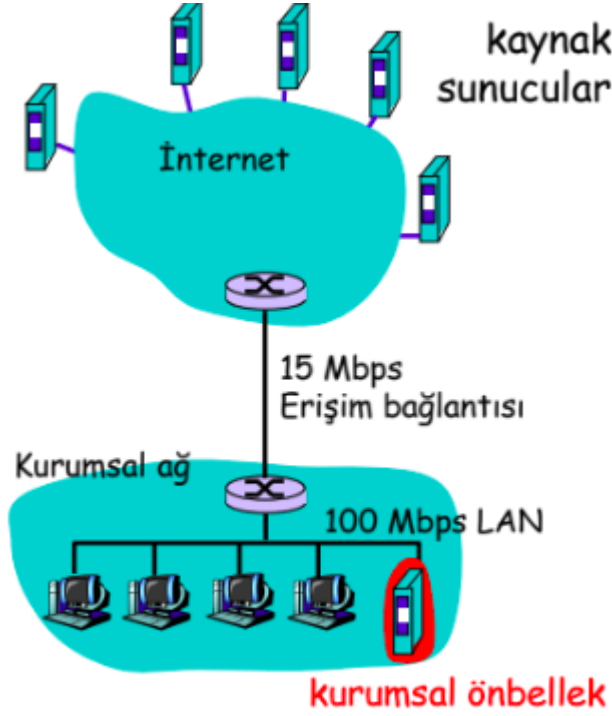
Web siteleri kişisel bilgilerimizi saklarlar. 😊



## Proxy Server & Cache

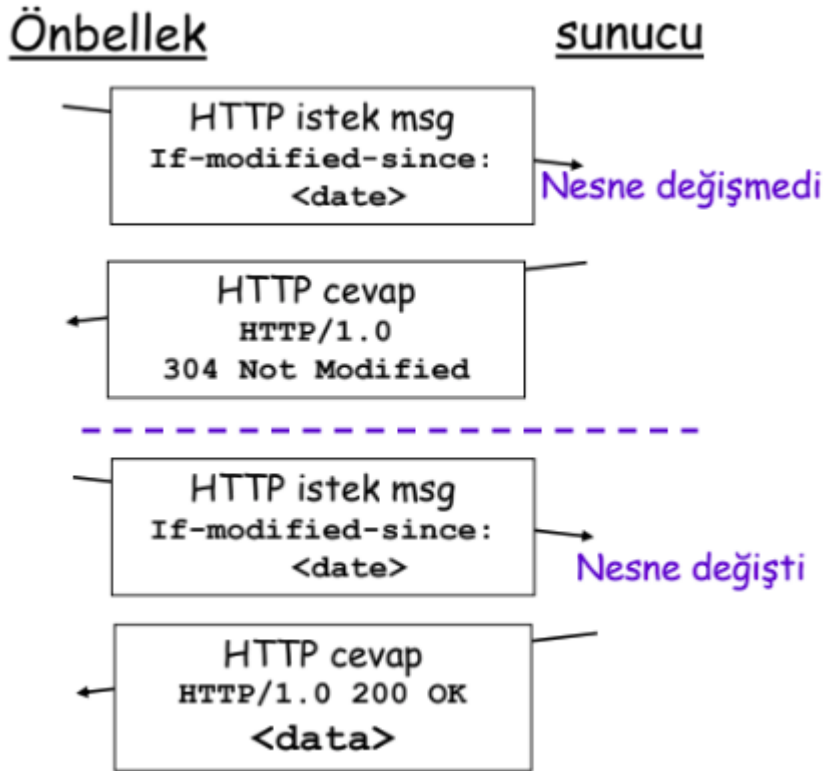
*Client* isteklerini *server* ile uzun süren bağlantılardan kaçınarak hızlıca halletmeyi amaçlar. Belli başlı *server*'lar *cache*'e atılır ve *server*'a istek yollamak yerine yerel ağdaki *proxy server*'a istek yollanarak çok hızlıca işlem halledilir.

LAN (yerel ağ) diğer *network*'lere kıyasla çok hızlıdır.



## Conditional GET (Koşullu GET)

Bu yöntemler *Proxy server* önbelleğinde (*cache*) bulunan verilerin güncel olup olmadığı sorgulanır.

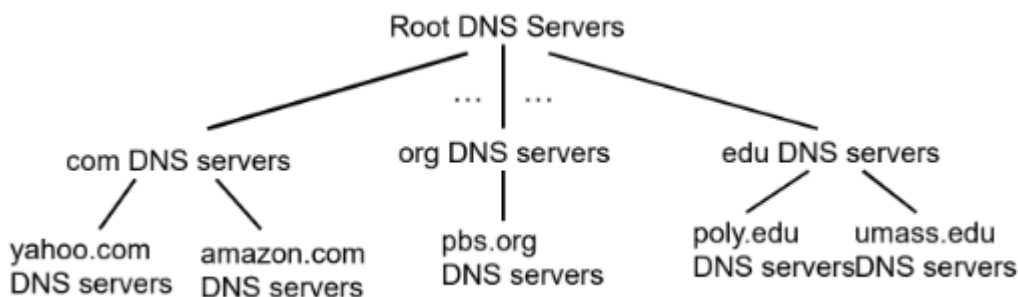


## Domain Name System (DNS)

İnternette adresler IP (192.168.1.1) ile tanımlanır. DNS'ler ile IP'lere isimler (google.com) atanır.

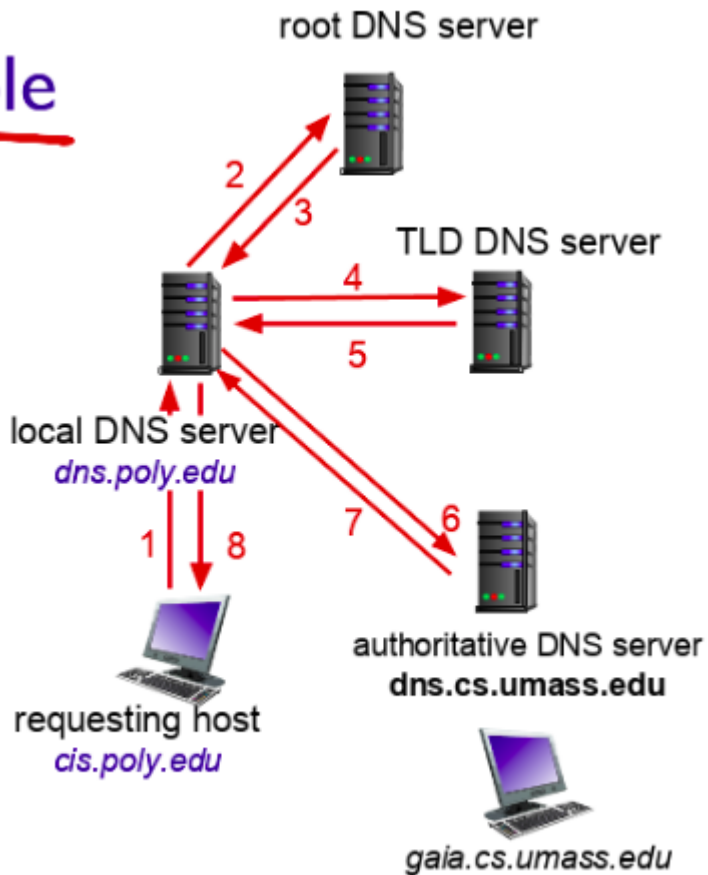
DNS eşleştirilmesi yapıldığında *Local DNS*'de *cache*'e alınır, bundan dolayı TLD sık kullanılmaz.

DNS Türü	Açıklama
Local	DNS hiyerarşisine ait değildir, her istek ilk burada eşleştirilmeye çalışılır
Root	Yerel ( <i>local</i> ) DNS sunucularının çözemedikleri isimler için buraya danışılır
TLD	Top-level domain, <i>com</i> , <i>org</i> , <i>net</i> , <i>tr</i> ... gibi ülke etki alanlarından sorumludurlar
Authoritative	Yetkili isim sunucuları, kurumlardaki sunucuların isimlerini eşleştirir

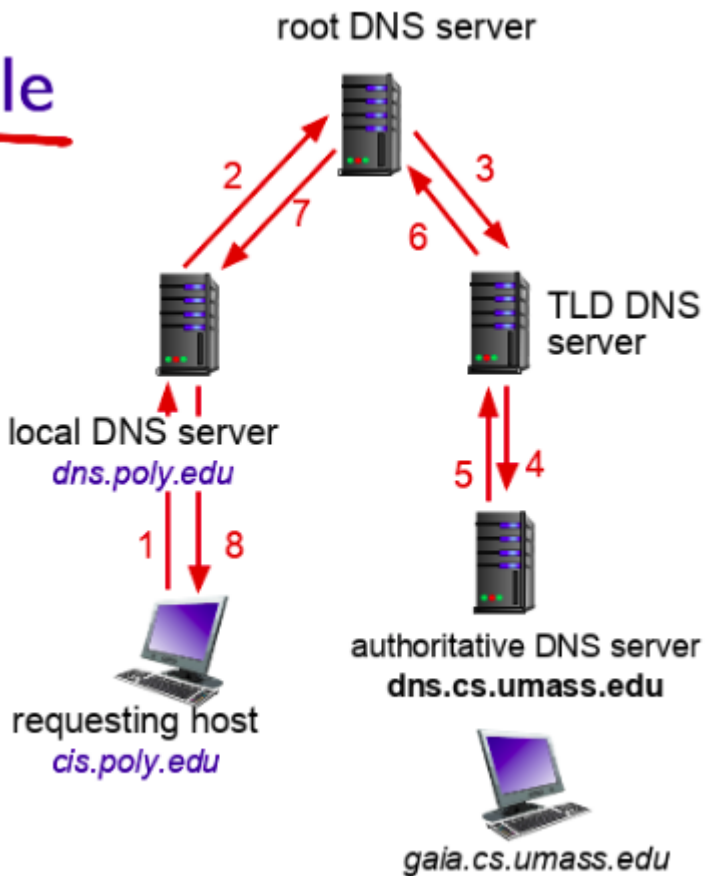


## DNS Resolution Examples (DNS Çözümleme Örnekleri)

ple



ole



## DNS Record (DNS Kayıtları)

Kayıtların formatı (name, value, type, ttl) şeklindedir.

Type	Açıklama
A	name: hostname, value: IP
NS	name: domain, value: hostname
CNAME	name: takma isim, value: domain
MX	name: alakalı isim, value: <i>mailserver</i> ismi

### Inserting DNS Record

- DNS server ismi ve IP adersi belirlenir
- TLD Server'lara alttaki şekilde kayıt edilir

```
(dns1.manolyatekstil.com, 212.212.212.1, A)
(manolyatekstil.com, dns1.manolyatekstil.com, NS) # Nameserver
```

### Attacking DNS

#### DDoS attacks

- ❖ Bombard root servers with traffic
  - Not successful to date
  - Traffic Filtering
  - Local DNS servers cache IPs of TLD servers, allowing root server bypass
- ❖ Bombard TLD servers
  - Potentially more dangerous

#### Redirect attacks

- ❖ Man-in-middle
  - Intercept queries
- ❖ DNS poisoning
  - Send bogus relies to DNS server, which caches

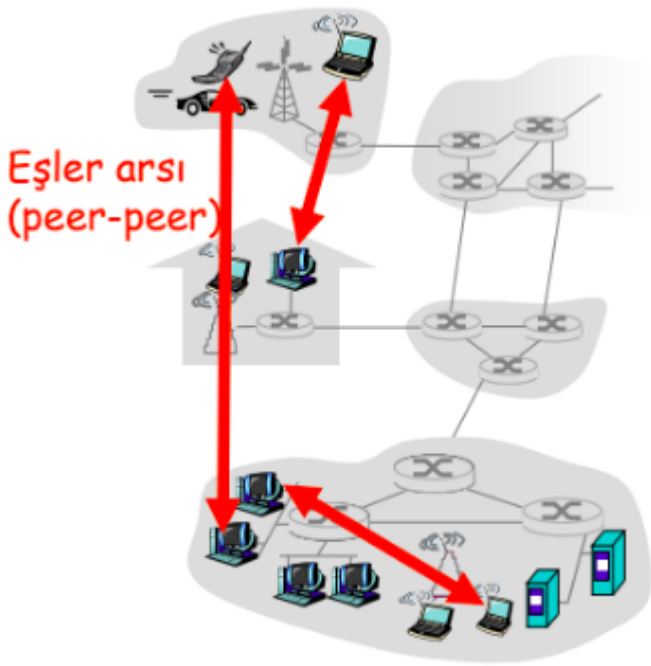
#### Exploit DNS for DDoS

- ❖ Send queries with spoofed source address: target IP
- ❖ Requires amplification

Application Layer 2-74



## P2P (Peer to Peer)

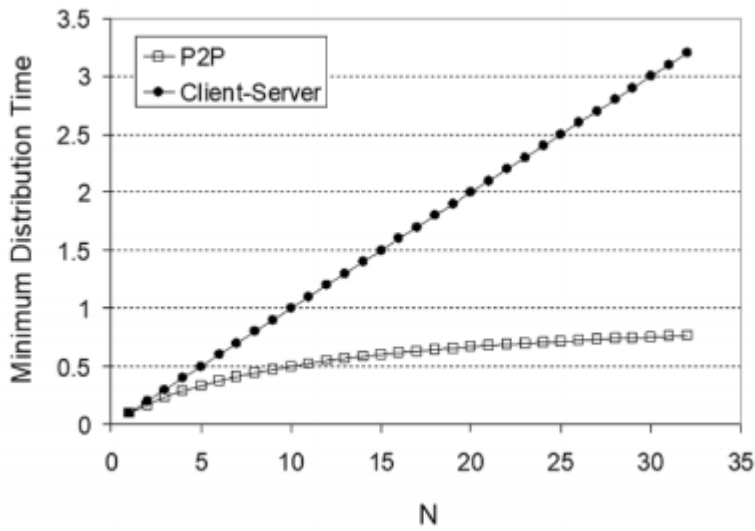


- *Server torrent'e* katılanları izler ve her zaman açık olmaz
- *Network'teki* bilgisayarlar rastgele erişim kurarlar
- Eş bilgisayarlar zaman zaman bağlantı kurarlar ve IP adresleri değişebilir

Terim	Açıklama
Chunk	256KB'lik <i>packet</i> 'lar
Torrent	<i>Chunk</i> alışverişi yapan grup

## P2p File Distribution (Dosya Paylaşımı)

Hızlı veri aktarımı sağlayan bir yapıdır.



- *Chunk*'lar indirilirken aynı zamanda karşıya da yüklenir
- Çok yükleme yapan çok hızlı indirir
- İsteğe bağlı yükleme veya indirme iptal edilebilir

## Video Streaming and CDNs: context

### Video Streamin

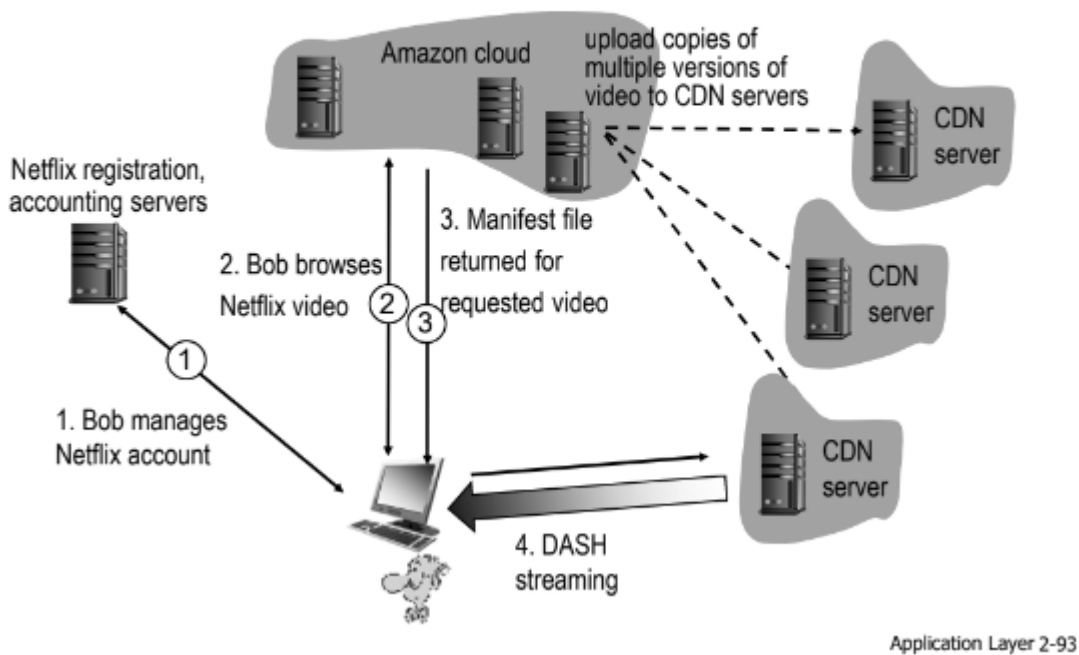
Her video, resin topluluğundan ver resimler de *pixel*'lerden oluşur. Her *pixel* de *bit*'lerden oluşmakta ve bunların aktarımları gerçekleşmektedir. *Bit* sayısını azaltmak için;

Yöntem	Açıklama
spatial (uzaysal)	N tane renk göndermek yerine, rengi ve tekrar etme sayısını gönderir
Temporal (zamansal)	Sadece bir önceki resim ile farklı olan yerleri gönderir

### Content Distribution Networks (İçerik Dağıtım Ağları)

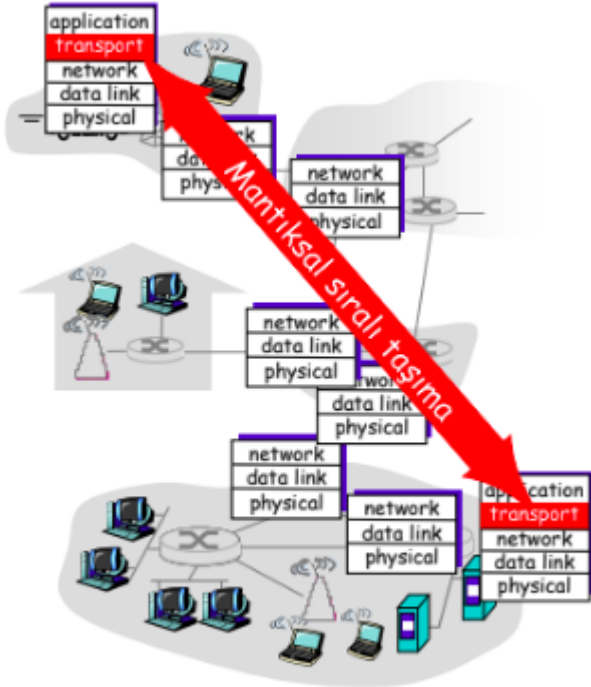
İçerikler kopyalanarak birden fazla *server*'dan akatarılır.

## Case study: Netflix



## Transport Layer

*Network layer*, *host*'lar arası mantıksal iletişimi sağlarken; *transport layer*, ***process***'ler arası mantıksal iletişimi sağlar



## Internet Transport Layer Protocols

Yine, UDP ve TCP protokolleri kullanılır. 😊

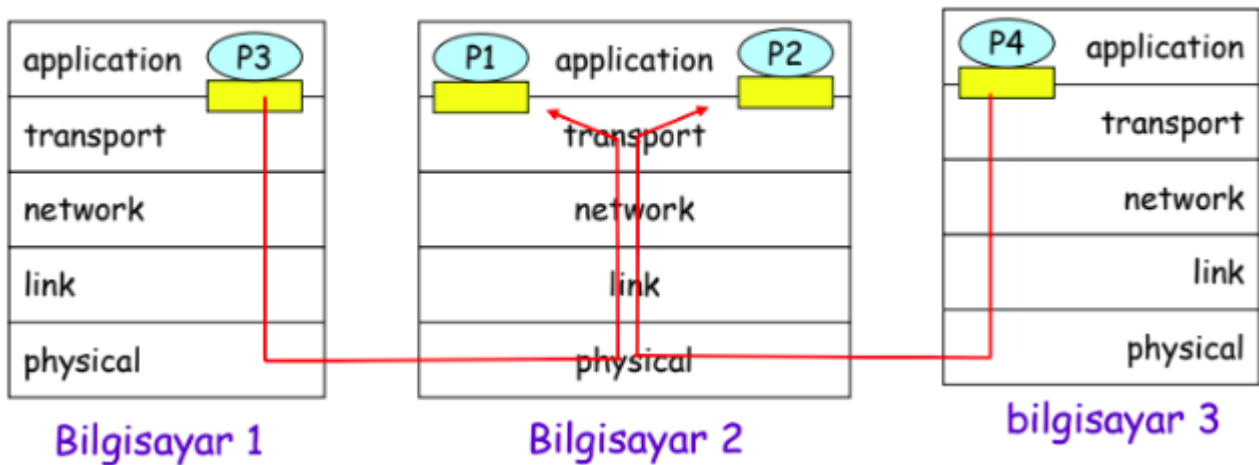
## Multiplexing (Çoğullama)

### Multiplexing Yeri Açıklama

Gönderen bilgisayar Birden çok *socket*'ten verileri toplar, başlık ekliyerek **segment** haline getirir.

Alıcı bilgisayar Alınan *segment*'leri doğru *socket*'e teslim eder

= socket  = işlem



## Demultiplexing (Azaltma / Parçalama)

- Bilgisayarlardan IP *datagram*'ları alınır.
  - *Datagram*'larda *source IP* ve *dest IP* vardır
  - Her *datagram* bir *segment* taşır
  - Her *segment*'in kaynak ve *dest port* numaraları vardır

## TCP / UDP Demux

### UDP Yönelendirme

*Source IP*

### TCP Yönelendirme

*Source IP*

*Destination IP*

*Source port* numarası

*Destination port* numarası

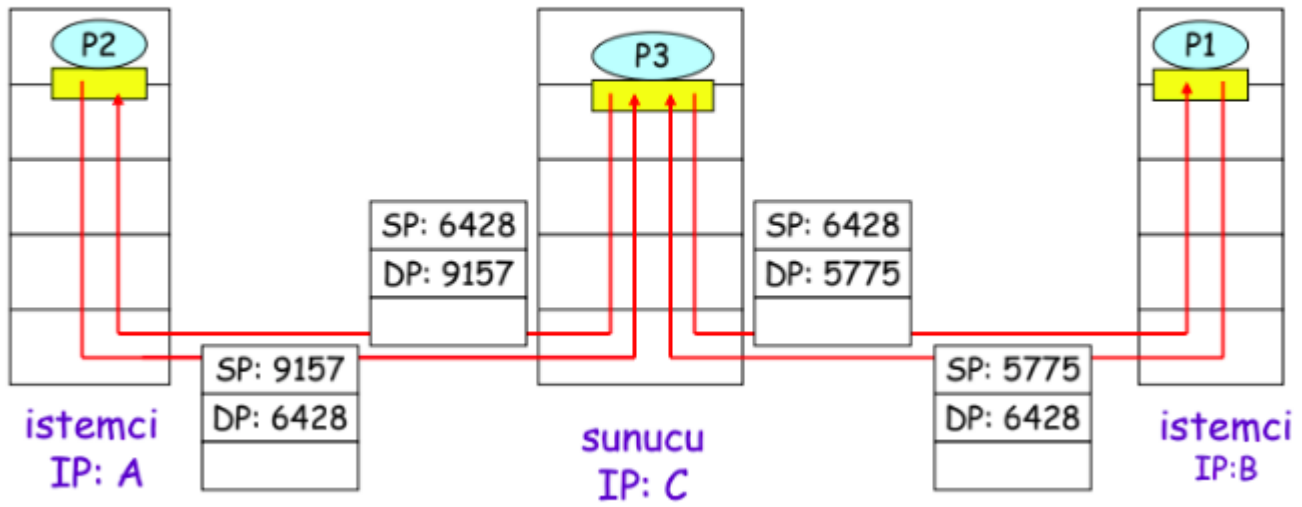
*Destination port* numarası

Socket, *source IP* ve *destination port* numarasından oluşur.

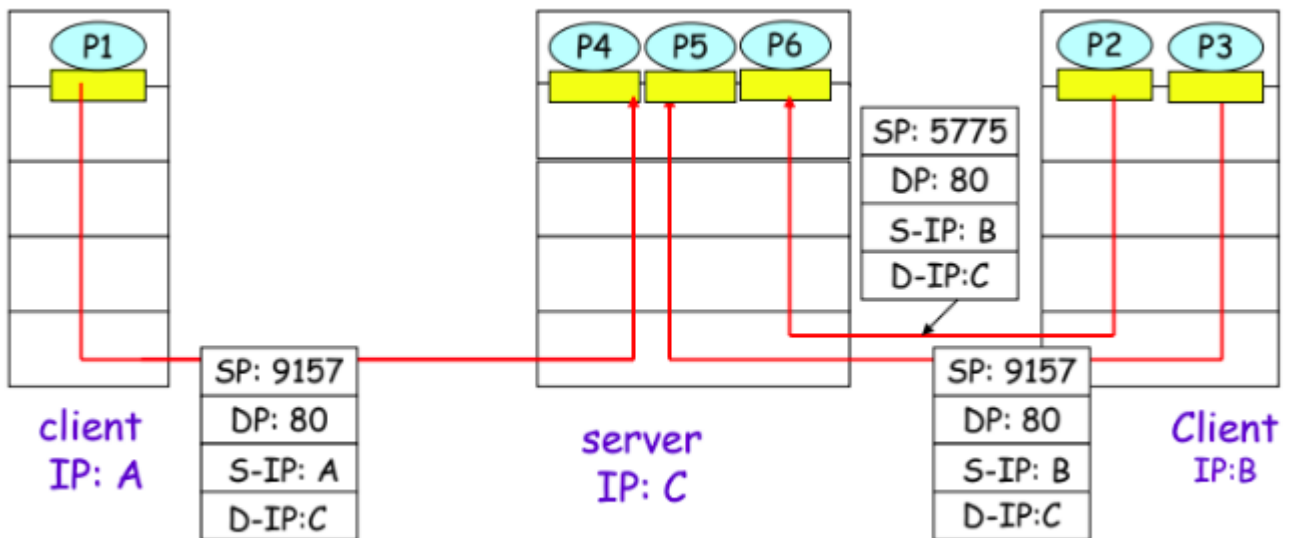
## UDP Demux Örneği

□ SP: Source Port

DP: Destination Port



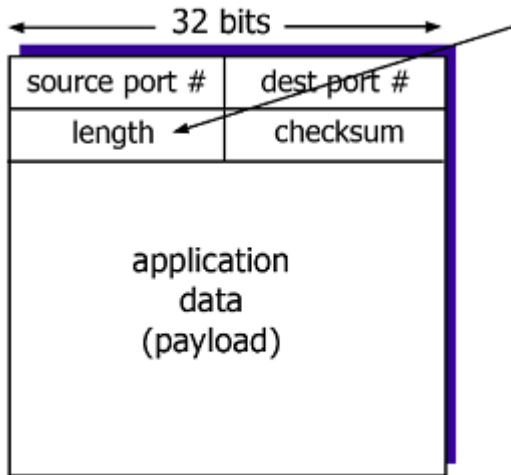
## TCP Demux Örneği



## UDP (User Datagram Protocol)

UDP yayıncılıkta tercih edilen bir *protocol*'dür. Amacı tamamıyla hızı arttırmak ve maaliyeti düşürmektir.

- *Packet*'in varıp, varmadığıyla ve güvenliğiyle ilgilenmez (*Unreliable transport*), varmazsa tekrar gönderir.
- Tıkanıklık kontrolüne (*congestion control*) ihtiyaç yoktur, olabildiğince hızlı gönderir
- Bağlantı kurmaya gerek yok, zaman kaybına neden olur
- Basitir, *sender* ve *reciver* asla birbiriyle iletişimde değildir
- Olumsuz geri dönüş yoktur.



UDP segment format

## UDP Checksum

Aktarılan *segment*'deki hataları algılamak için kullanılan yöntemdir.

example: add two 16-bit integers

	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
wraparound	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1
sum	1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0
checksum	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1

**Note:** when adding numbers, a carryout from the most significant bit needs to be added to the result

# Reliable Data Transfer (RDT)

## Rdt 1.0

Tam güvenli bir kanaldır.

- *Bit* ve *packet* kayıpları yoktur
- *Sender* ve *receiver* verileri güvenli kanaldan (*underlying channel*) alır

## Rdt 2.0

Bitlerde hatalar söz konusu olabilir.

- *Bit* hataları *checksum* ile algılanır.
- *Acknowledgements (ACKs)* paket alındı bilgisi, *negative acknowledgements (NAKs)* paketin hatalı olduğu bilgisi gibi *feedback*'ler vardır.

## Rdt 2.0 Kusurları

- ACK / NAK mesajları bozulması durumunda geçerli *packet* yeniden gönderilir
- *Sender* her gelen *packet*'e *segment* numarası ekler, birden fazla gelen *packet*'ları *receiver* atar
- *Sender* bir *packet* gönderdikten sonra *feedback* için bekler, bu da zamandan kayıp demektir.

## Rdt 2.1

*Sender*:

- *Packet*'lara *segment* numarası ekler.
- ACK / NAK bozuk alınıp alınmadığını kontrol eder

*Receiver*:

- Alınan *packet*'ların eşsiz olup olmadığını kontrol eder
- ACK / NAK mesajlarının *sender* tarafından alınıp alınmadığını bilmez

## Rdt 2.2

NAK içermez, sadece ACK kullanarak rdt 2.1 ile aynı görevi yapar.

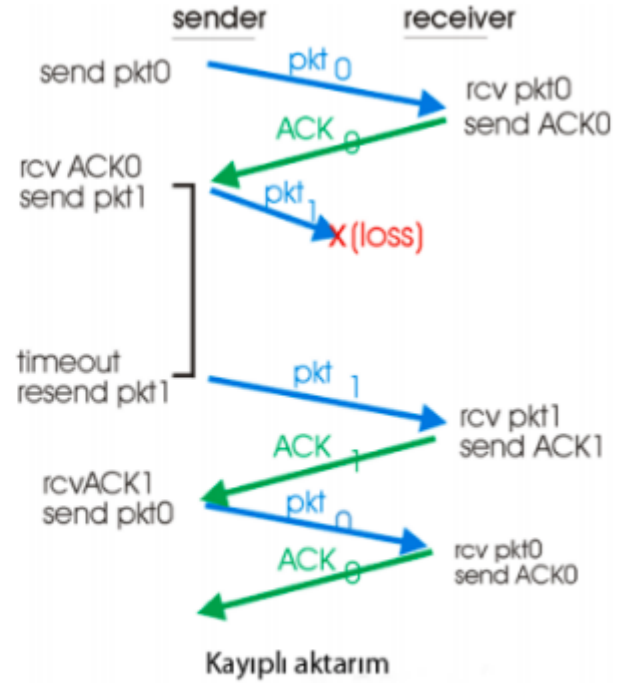
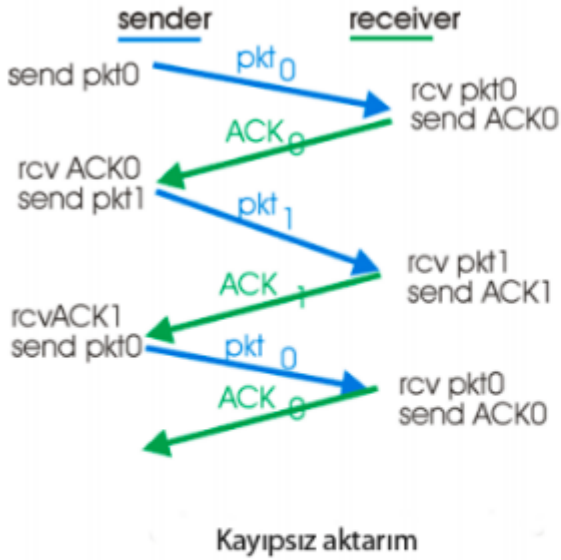
- NAK yerine *packet* başarılı alındığında ACK mesajları gönderilir.
- Çift ACK mesajı NAK gibi kabul edilir, *packet* yeniden gönderilir.



## Rdt 3.0

Rdt 2.2'ye ek olarak:

- *Sender* belli sürede ACK mesajı almazsa (*timeout*) *packet* yeniden gönderilir.
- Eşsiz olmayan *packet*'lar *segment* numaraları ile ayırt edilir.

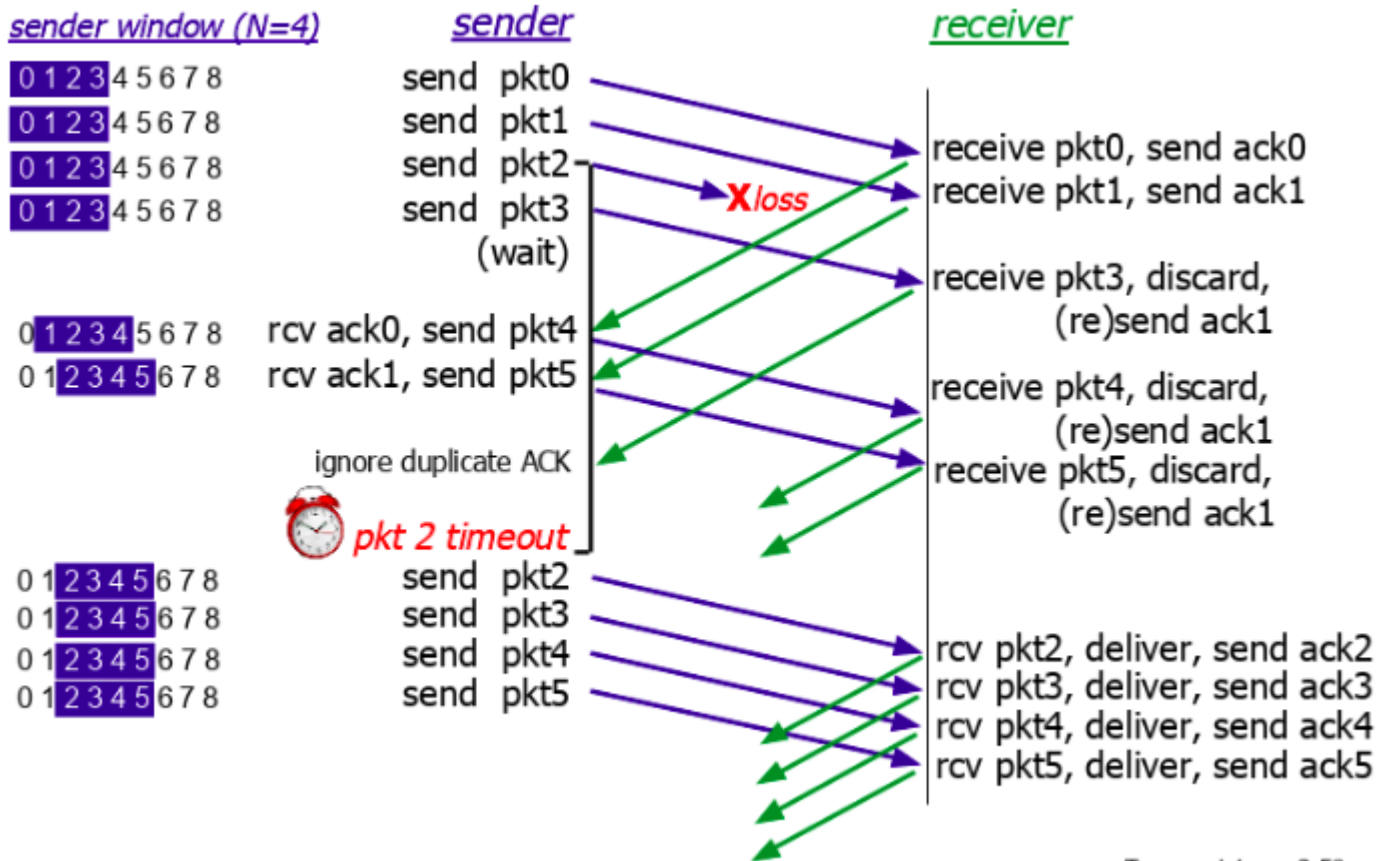


## Pipelined Protocols

Bir *packet* göndermek yerine birden fazla gönderilir.

- *Reciver* aldığı her sağlam *packet* için *ack* gönderir
  - Hatalı *packet*'ler için *ack* gitmez
  - Kaçırılan paketler için en son gönderilen *ack* gönderilir
- Tekrar eden *ack*'lar *sender* tarafından görmezden gelinir ve *packet* yeniden gönderilir
  - Bu yapıya **Go back N (GDN)** adı verilir.

Selective repeat ?



## TCP (Transmission Control Protocol)

Sıkıldım 😞