

# BM 402 Bilgisayar Ağları (Computer Networks)

---

M.Ali Akcayol  
Gazi Üniversitesi  
Bilgisayar Mühendisliği Bölümü

Not: Bu dersin sunumları, ders kitabının yazarları James F. Kurose ve Keith W. Ross tarafından sağlanan sunumlar üzerinde değişiklik yapılarak hazırlanmıştır.

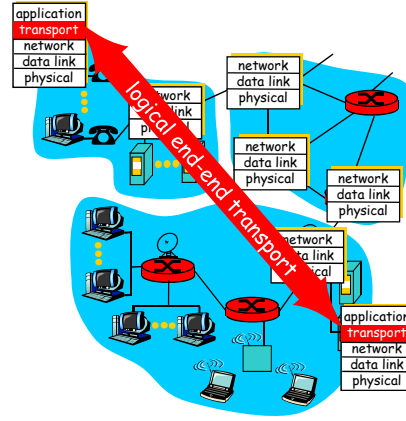
## Ders konuları

---

- Transport Layer Hizmetleri
- Multiplexing ve Demultiplexing
- Bağlantısız Gönderim - UDP

## Transport hizmetleri ve protokoller

- Farklı hostlardaki prosesler arasında *mantıksal bağlantı* (*logical communication*) oluşturur
- transport protokolleri uç sistemlerde çalışır
  - Gönderen taraf: uygulama mesajlarını *segment*'lere böler ve network layer'a gönderir
  - Alıcı taraf: segmentleri birleştirir ve application layer'a gönderir
- Uygulamalar için birden fazla transport protokol bulunmaktadır
  - Internet: TCP ve UDP



3/101

## Transport ve network layer

- *network layer*: hostlar arasında mantıksal bağlantı
- *transport layer*: prosesler arasında mantıksal bağlantı
  - Network layer hizmetlerine güvenir

### Benzetim:

12 çocuk farklı yerde aynı evdeki 12 çocuğa mektup gönderiyor

- prosesler = çocuklar
- Uygulama mesajları = zarflardaki mektuplar
- hostlar = evler
- transport protokol = her evde mektup toplama ve dağıtma yapan kişiler
- network-layer protokol = posta hizmeti

4/101

## Internet transport layer protokolleri

- Güvenilir, sıralı gönderim: TCP

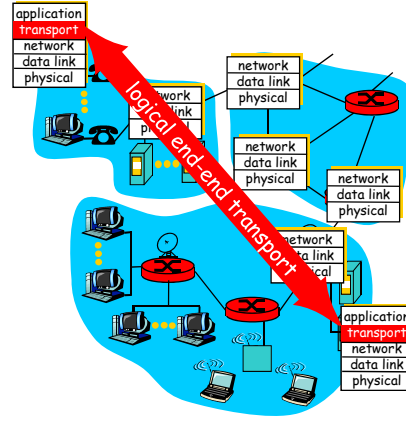
- congestion control
- flow control
- connection setup

- Güvenilir olmayan, sırasız gönderim: UDP

- IP protokolünün best-effort özelliği kullanılır

- Sunulamayan hizmetler:

- delay garanti
- bandwidth garanti



5/101

## Ders konuları

- Transport Layer Hizmetleri
- Multiplexing ve Demultiplexing
- Bağlantısız Gönderim - UDP

6/101


## Multiplexing/demultiplexing


### Demultiplexing alıcı host'ta

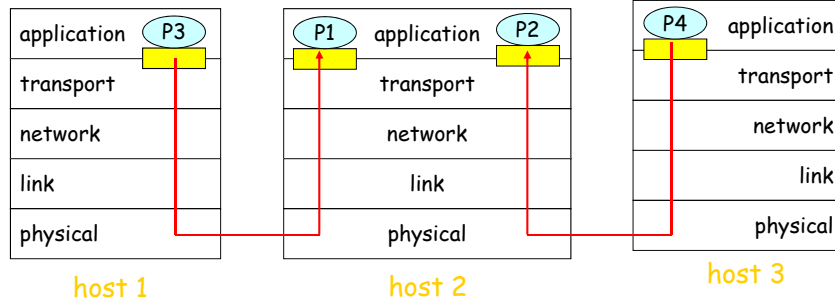
Doğru soketten gelen segmentler alınır

### Multiplexing gönderen host'ta

Birden çok soketten gelen veri toplanır, başlık eklenir (demultiplex için kullanılır)

 = soket

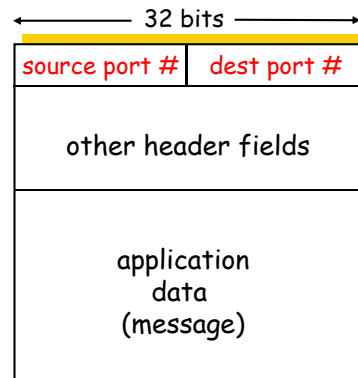
 = proses



7/101

## Demultiplexing işlemi

- host IP datagramlarını alır
  - her datagram kaynak IP adresine ve hedef IP adresine sahiptir
  - her datagram 1 transport-layer segment taşır
  - her segment kaynak ve hedef port numarasına sahiptir (bazı uygulamalar için bilinen port numaraları atanır)
- host IP adreslerini ve port numaralarını kullanarak segmenti uygun sokete yönlendirir.



TCP/UDP segment format

8/101

## Connectionless demultiplexing

- Port numaralarıyla soketler oluşturulur:

```
DatagramSocket mySocket1 = new  
    DatagramSocket(99111);  
DatagramSocket mySocket2 = new  
    DatagramSocket(99222);
```

- UDP soket bir ikiliyle tanımlanır:

(dest IP address, dest port number)

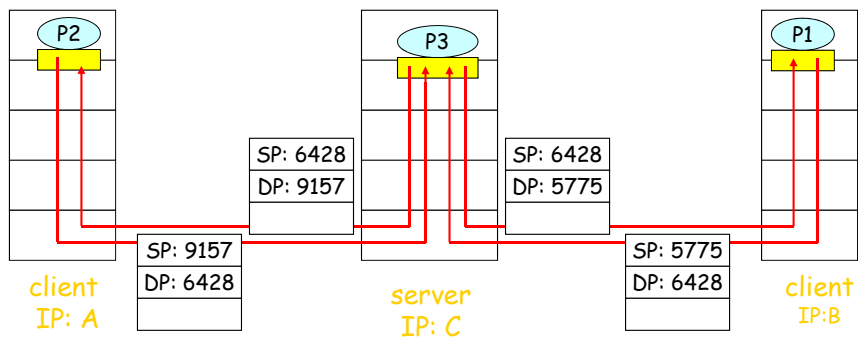
- Host UDP segment aldığı anda:

- Segmentteki hedef port numarası kontrol
- UDP segment port numarasıyla birlikte sokete yönlendirilir
- IP datagramlar aynı sokete kaynak ve hedef IP ve port numaralarıyla yönlendirilir

9/101

## Connectionless demultiplexing - devam

```
DatagramSocket serverSocket = new DatagramSocket(6428);
```



SP dönüş adresini (return address) sağlar

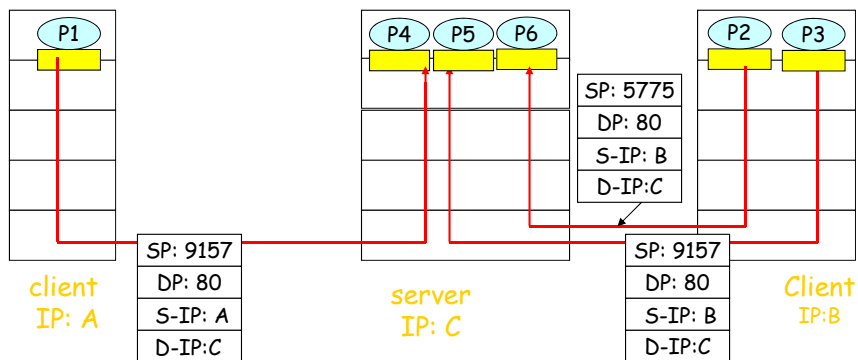
10/101

## Connection-oriented demultiplexing

- TCP socket bir dörtlüyle tanımlanır:
  - source IP address
  - source port number
  - dest IP address
  - dest port number
- Alıcı host bu 4 değeri segmenti uygun sokete yönlendirmek için kullanır
- Server host eşzamanlı çok sayıda TCP socket destekler:
  - her socket kendi dörtlüsüyle tanımlanır
- Web sunucular her bağlanan client için farklı bir sokete sahiptir
  - non-persistent HTTP her istek için farklı bir sokete sahiptir

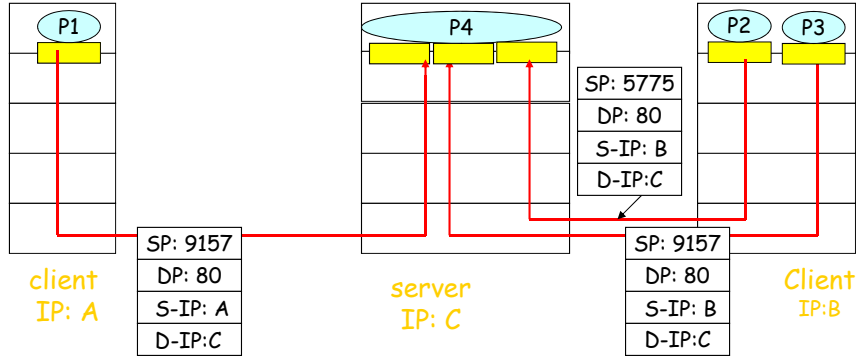
11/101

## Connection-oriented demultiplexing - devam



12/101

## Connection-oriented demux: Threaded Web Server



13/101

## Ders konuları

- Transport Layer Hizmetleri
- Multiplexing ve Demultiplexing
- Bağlantısız Gönderim - UDP

14/101

## UDP: User Datagram Protocol [RFC 768]

- Güvenilir olmayan Internet transport protokolü
- “best effort” servis, UDP segmentleri:
  - Lost (kayıp olabilir)
  - Delivered out of order to app (uygulamaya sırasız gidebilir)
- **connectionless:**
  - UDP alıcı ve gönderici arasında handshaking yapılmaz
  - Her UDP segment diğerlerinden ayrı değerlendirilir

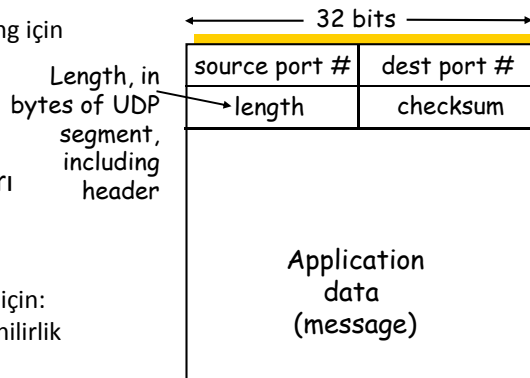
### UDP niçin vardır?

- Bağlantı oluşturulmaz (gecikme olur)
- basittir: alıcı ve gönderici arasında bağlantı durumu yoktur
- Küçük segment başlığı
- Tıkanıklık denetimi yoktur: UDP istediği hızda veri gönderebilir.

15/101

## UDP - devam

- Sıklıkla multimedia uygulamalarında streaming için kullanılır
  - loss tolerant
  - rate sensitive
- diğer UDP kullanımları
  - DNS
  - SNMP
- UDP ile güvenilir transfer için: application layer’da güvenilirlik eklenmelidir.



UDP segment format

16/101



## UDP checksum

Amaç: iletilen segmentte hata algılama

### Gönderici:

- segment içeriklerine 16-bit integer dizisi olarak bakılır
- checksum: segment içeriğinin 1 tümleyeni toplamı alınır
- Gönderici checksum değeri UDP checksum alanına yerleştirir

### Alıcı:

- Alınan segmentte checksum hesaplanır
- hesaplanan checksum gelen checksum değeriyle karşılaştırılır:
  - HAYIR – hata var
  - EVET – hata yok. *Yinede hiçbir şekilde hata yok denilebilirmi?*

17/101

## Internet checksum örnek

- Sayıları toplarken en soldaki bitlerdeki taşma sonuca eklenir
- Örnek: iki 16-bit integer toplanır

		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	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

18/101