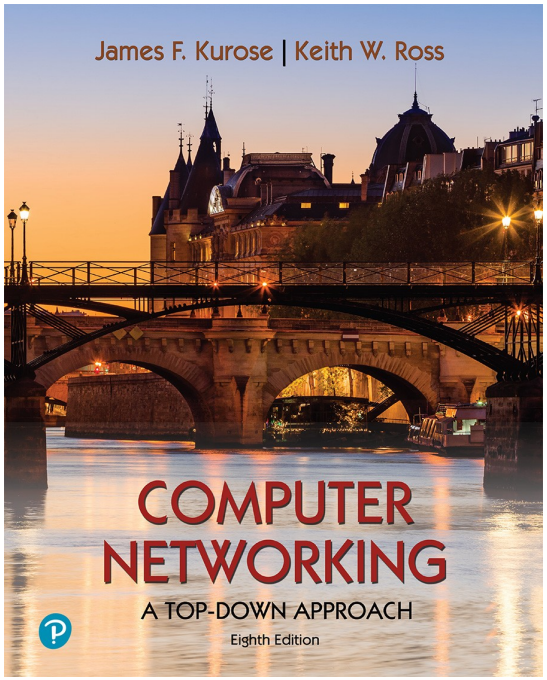
Bölüm 3
Taşıma Katmanı

Doç. Dr. Mehmet Dinçer Erbaş
Bolu Abant İzzet Baysal Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü



Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

Bölüm 3: Taşıma Katmanı

hedefimiz:

- Taşıma katmanı hizmeti prensiplerini anlamak:
  - Çoklama, çoklama çözme
  - Güvenilir veri transferi.
  - Akış kontrolü
  - Sıkışma kontrolü.
- İnternet taşıma katmanı protokollerini öğrenmek:
  - UDP: bağlantısız taşıma
  - TCP: bağlantı-odaklı güvenilir taşıma
  - TCP sıkışma kontrolü

Bölüm 3: konular

3.1 taşıma-katmanı hizmetler

3.2 çoklama ve çoklama çözme

3.3 bağlantısız taşıma: UDP

3.4 güvenilir veri transferi prensipleri

3.5 bağlantı-odaklı taşıma: TCP

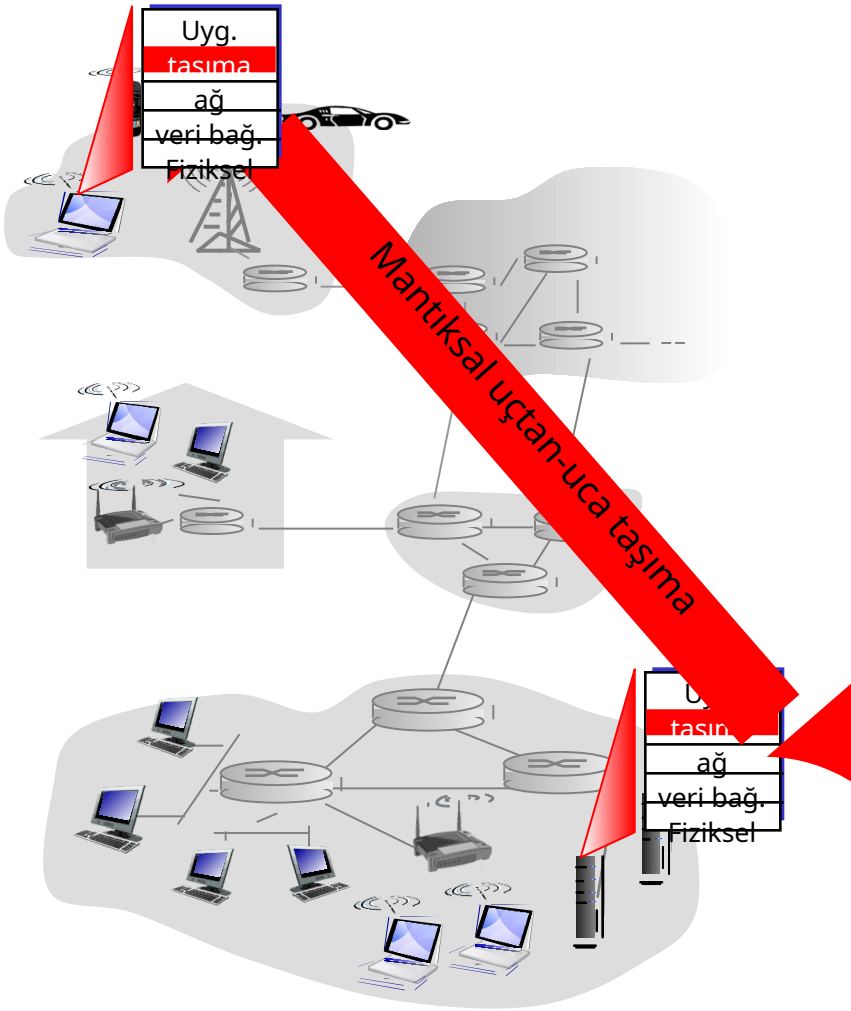
- Segment yapısı
- Güvenilir veri transferi
- Akış kontrolü
- Bağlantı yönetimi

3.6 sıkışma kontrolü prensipleri

3.7 TCP sıkışma kontrolü

Taşıma hizmetleri ve protokolleri

- Farklı cihazlar üzerinde çalışan uyg. Işlemleri arasında mantıksal bağlantı kurar.
- Taşıma protokolleri uç sistemlerde çalışır.
  - gönderen taraf: uyg. mesajını segment ismi verilen parçalara ayırır, ağ katmanına gönderir.
  - alan taraf: segmentleri geri birleştirir, mesajı yeniden oluşturur, uyg. katmanına gönderir.
- Uygulamalara birden fazla protokol imkanı sunar.
  - Internet: TCP ve UDP



Taşıma vs. ağ katmanı

- *Ağ katmanı:* cihazlar arası mantıksal bağlantı.
- *Taşıma katmanı:* işlemler arası mantıksal bağlantı
  - Ağ katmanı servisine dayanır ve ağ katmanı servisini geliştirir.

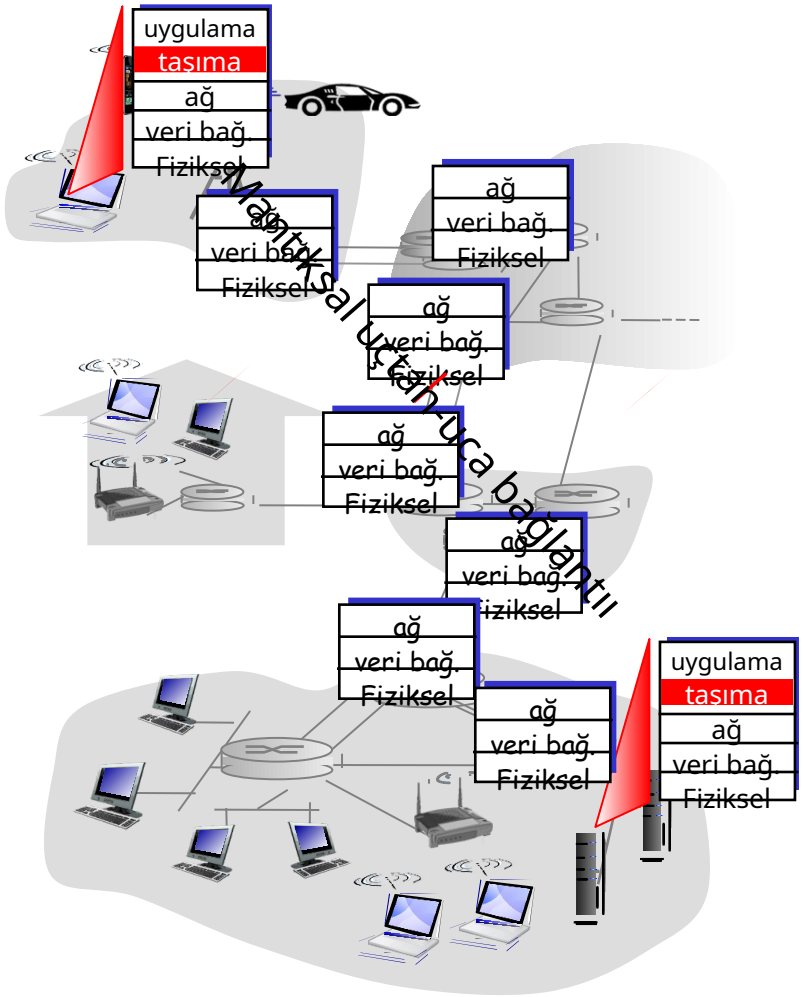
ev benzerliği:

Ayşe'nin evindeki 12 çocuk, Ali'nin eindeki 12 çocuğa mektp göndersin:

- cihazlar = evler
- işlemler = çocuklar
- Uyg. mesajları = zarf içinde mektuplar
- taşıma protokolü = Ev içindeki çocuklara mektup dağıtan Ayşe ve Ali.
- ağ-katmanı protokolü = postal service

Internet taşıma-katmanı protokolleri

- güvenilir, sıralı teslimat (TCP)
  - sıkışma kontrolü
  - Akış kontrolü
  - Bağlantı kurulumu
- Güvenilir olmayan, sırasız teslimat: UDP
  - IP'nin "en-iyi gayret" ya klaşımının devamı gibidir.
- Sağlanmayan hizmetler:
  - Gecikme garantisi
  - Bant genişliği garantisi



Bölüm 3: konular

3.1 taşıma-katmanı hizmetler

3.2 çoklama ve çoklama çözme

3.3 bağlantısız taşıma: UDP

3.4 güvenilir veri transferi prensipleri

3.5 bağlantı-odaklı taşıma: TCP

- Segment yapısı
- Güvenilir veri transferi
- Akış kontrolü
- Bağlantı yönetimi

3.6 sıkışma kontrolü prensipleri

3.7 TCP sıkışma kontrolü

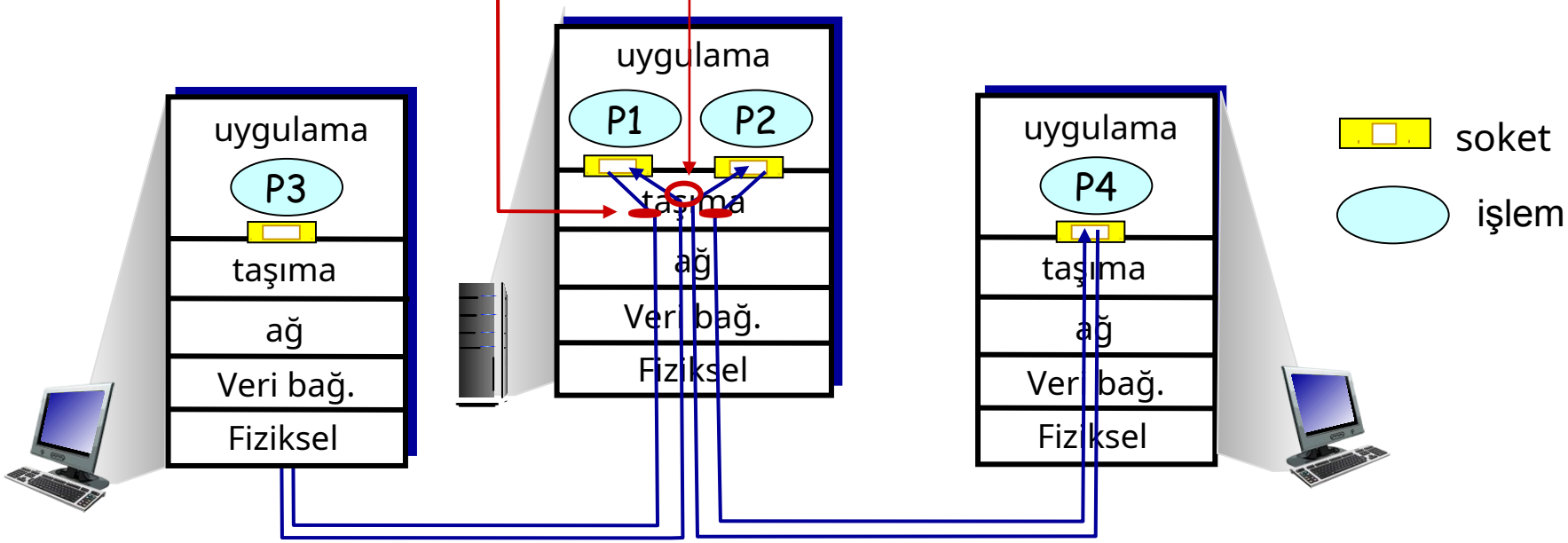
Çoklama/Çoklama çözme

Gönderende çoklama:

(Ing: Multiplexing)
Çoklu soketten veriyi işler, taşıma başlığı ekler (çoklama çözerken kullanılır).

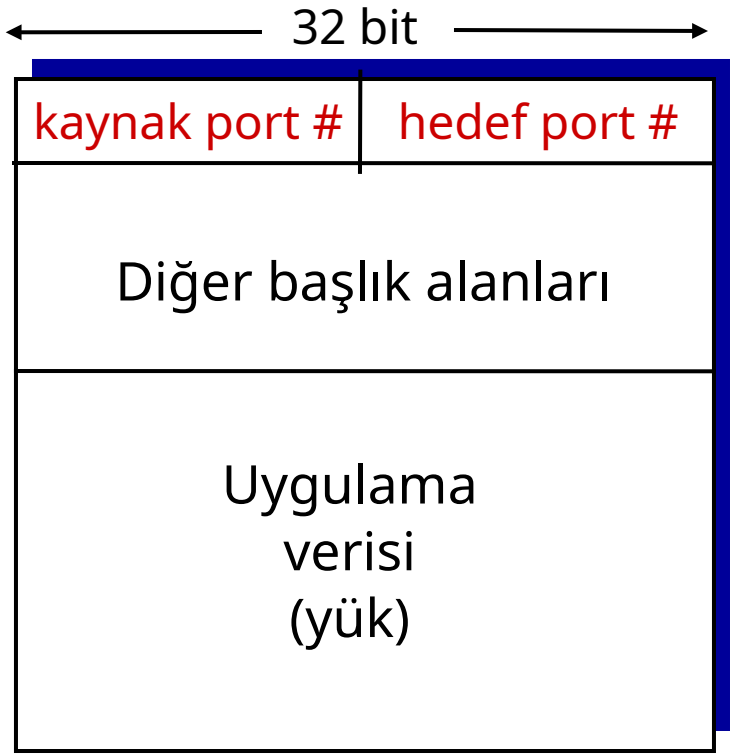
Alıcıda çoklama çözme:

(İng: Demultiplexing)
başlık bilgisini kullanarak alınan segmentleri doğru sokete yollar.



Çoklama çözme nasıl çalışır?

- Cihaz IP datagramı alır.
  - Her datagram kaynak IP adresi ve hedef IP adresi içerir.
  - Her datagram bir taşıma katmanı segmenti içerir.
  - Her segment kaynak ve hedef port numarası içerir.
- Cihaz IP adreslerini ve port numaralarını kullanarak segmenti uygun sokete yönlendirir.



TCP/UDP segment formatı

Bağlantısız çoklama çözme

- *hatırlatma:* oluşturulan soket cihaz-yerel port # ile oluşturulur:

```
DatagramSocket mySocket1
    = new
DatagramSocket(12534);
```

- *hatırlatma:* UDP soketinden datagram göndermek için şunlar belirtilmelidir
  - hedef IP adres
  - hedef port #

- Cihaz UDP segmenti aldığında:
  - Segmentteki hedef port # kontrol eder.
  - UDP segmenti belirtilen port # yönlendirir.

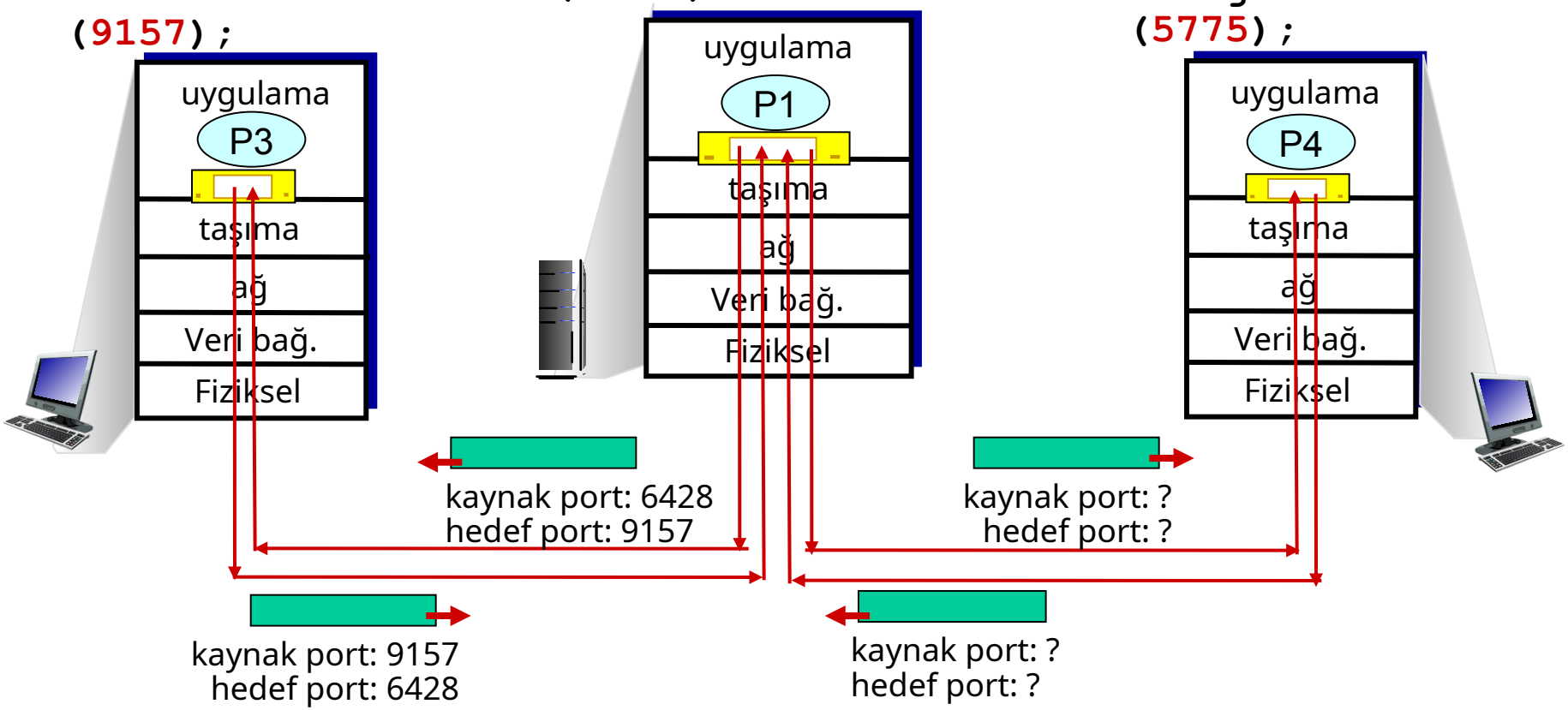
Hedef cihazda aynı hedef IP, port # sahip olan bütün datagramlar (farklı kaynak IP ve/veya kaynak port üzerinden gelseler bile) aynı sokete yönlendirilir.

Bağlantısız çoklama çözme: örnek

```
DatagramSocket
 mySocket2 = new
 DatagramSocket
 (9157);
```

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

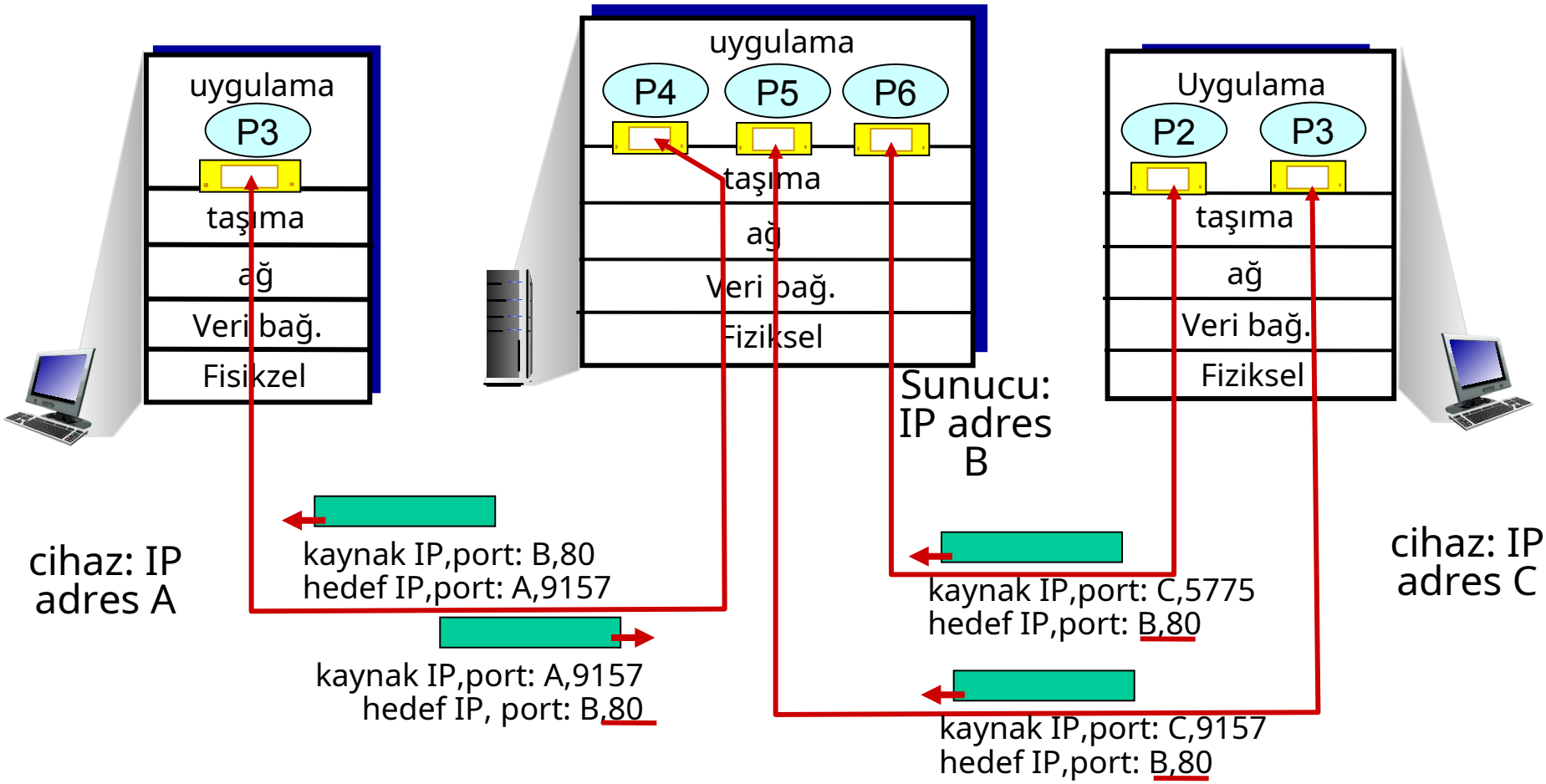
```
DatagramSocket
 mySocket1 = new
 DatagramSocket
 (5775);
```



Bağlantı-odaklı çoklama çözme

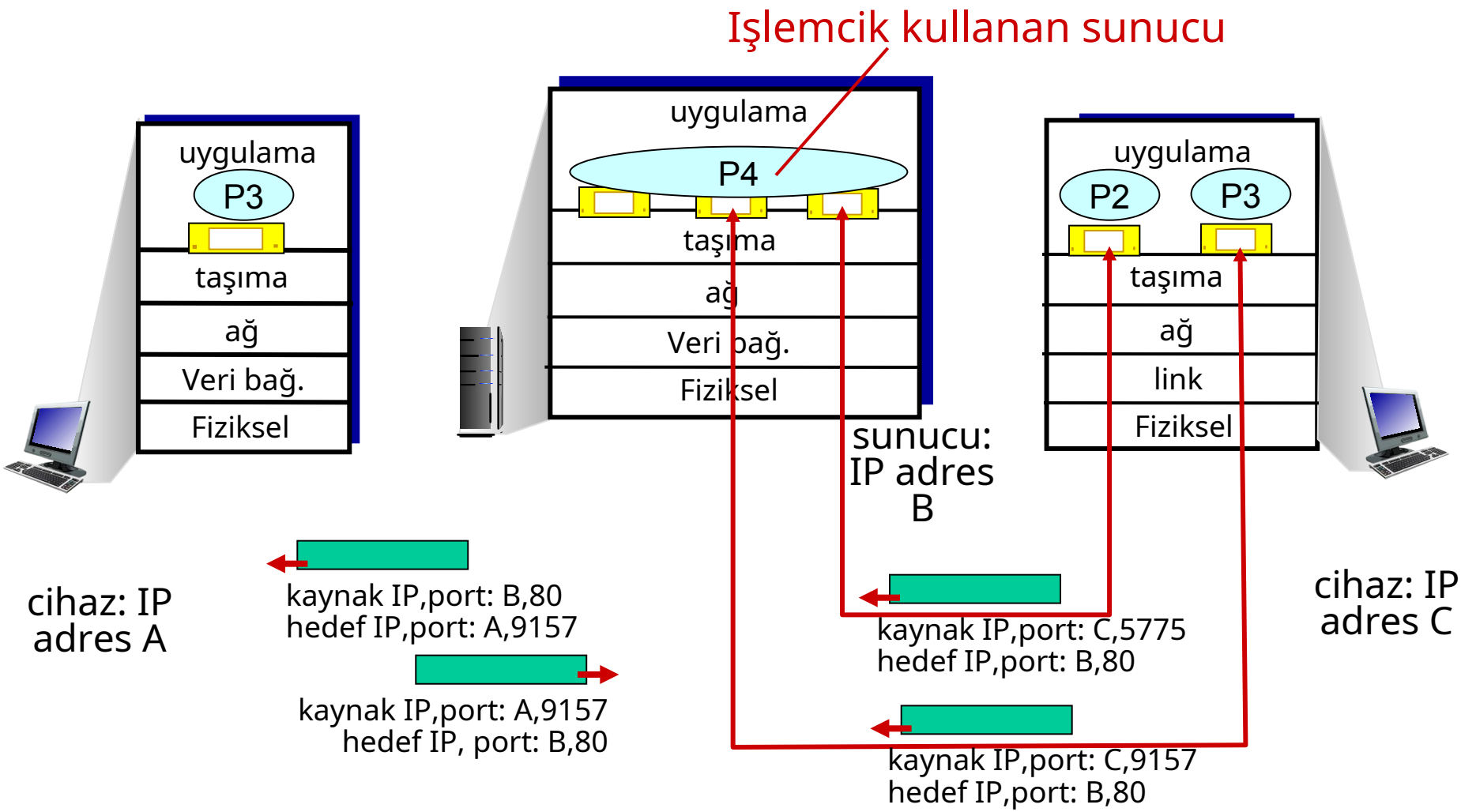
- TCP soketi 4 bilgi ile belirlenir:
  - kaynak IP adresi
  - kaynak port numarası
  - hedef IP adresi
  - hedef port numarası
- Çoklama çöz: alıcı dört değerin tamamını kullanarak segmenti uygun sokete yönlendirir.
- Sunucu cihaz eşzamanlı birçok TCP soketini destekleyebilir:
  - Her soket kendine ait 4 bilgi tarafından belirlenir.
- Web sunucuları bağlanan her istemci için farklı bir soket kullanır.
  - Devamlı olmayan HTTP her istek için farklı soket kullacaktır.

Bağlantı-odaklı çoklama çözme: örnek



Üç segment, herbiri IP adres B, port 80 gönderilmiş, çoklama çözme sonucu farklı soketlere yönlendirilir.

Bağlantı-odaklı çoklama çözme: örnek



Bölüm 3: konular

3.1 taşıma-katmanı hizmetler

3.2 çoklama ve çoklama çözme

3.3 bağlantısız taşıma: UDP

3.4 güvenilir veri transferi prensipleri

3.5 bağlantı-odaklı taşıma: TCP

- Segment yapısı
- Güvenilir veri transferi
- Akış kontrolü
- Bağlantı yönetimi

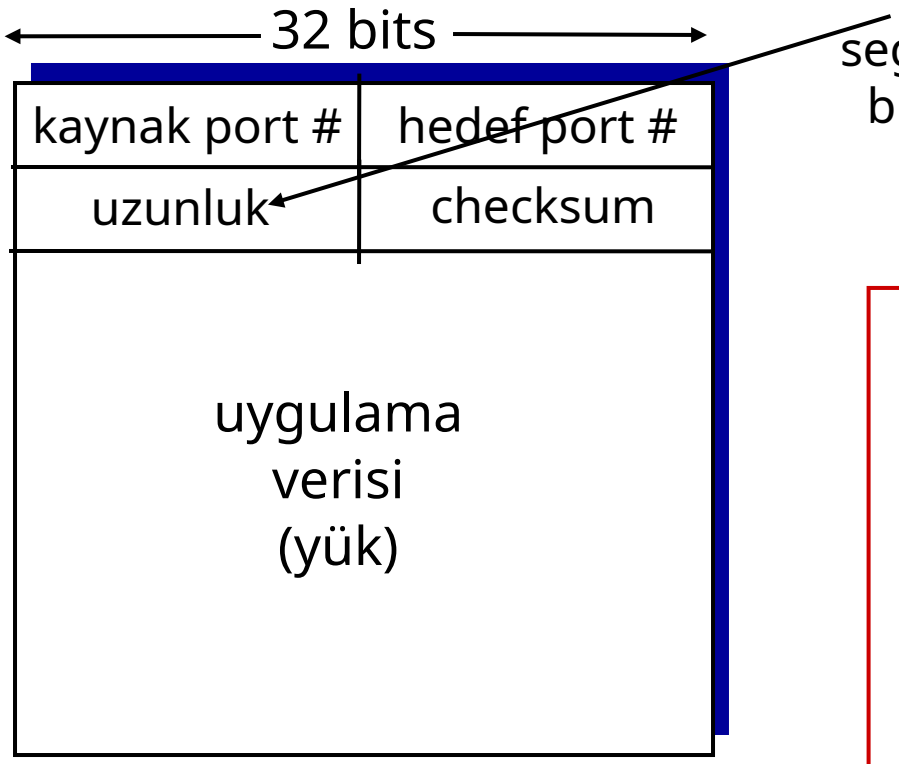
3.6 sıkışma kontrolü prensipleri

3.7 TCP sıkışma kontrolü

UDP: User Datagram Protocol [RFC 768]

- "asgari", "temel" Internet taşıma protokolü
- "en iyi gayret" hizmet, UDP segmentleri:
  - kaybolabilir.
  - Sırası bozulmuş şekilde uygulamaya teslim edilebilir.
- *bağlantısız:*
  - UDP gönderici ve alıcı arasında el sıkışma yoktur.
  - Her UDP segmenti diğerlerinden bağımsız olarak ele alınır.
- UDP kullanımı:
  - Yayın yapan çoklu-ortam uygulamaları. (kayıp tolere edilebilir, hız hassasiyeti)
  - DNS
  - SNMP
- UDP üzerinden güvenilir transfer:
  - Uygulama katmanına güvenilirlik eklenmelidir.
  - Uygulamaya özel hata düzeltme!

UDP: segment başlığı



UDP segment format

uzunluk, UDP segment byte cinsi büyüklüğü, başlık dahil.

Neden UDP var?

- Bağlantı kurulumu yoktur (gecikmeye neden olabilir).
- Basit: gönderen ve alıcıda bağlantı durumu yok.
- Küçük başlık
- Sıkışma kontrolü yok: UDP ile istenildiği kadar paket gönderilebilir.

UDP checksum

Hedef: gönderilen segment üzerinde "hata" te spit etme (e.g., değişen bit)

gönderen:

- Başlık dahil segment içeriğini 16-bit tamsayı olarak alır.
- checksum: segment içeriğinin toplamı (birin tamamlayıcısı toplam)
- Gönderici UDP checksum alanına hesaplanan değeri koyar.

alan:

- Alınan segment için checksum değerini hesaplar.
- Hesaplanan checksum değeri ile gönderilen checksum değeri aynı mı?
  - HAYIR – hata tespit edildi
  - EVET – hata tespit edilmedi. Ancak hata olabilir mi? Sonra bahsedeceğiz.

Internet checksum: örnek

örnek: iki 16-bit tamsayıyı toplayalım

	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
Başa ekle	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
toplam	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

Not: Başta kalan sayı var ise en sona eklenir.