

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

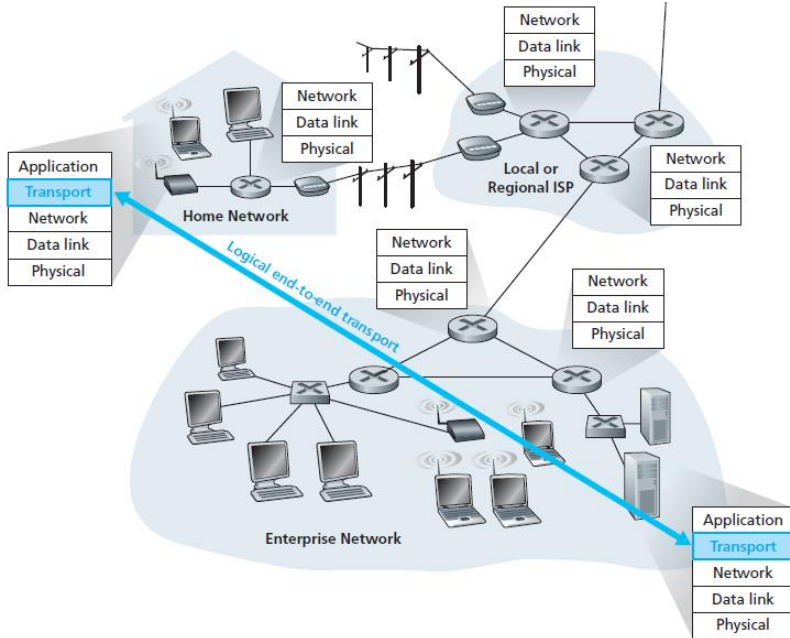
- ▶ **Transport layer servisleri**
- ▶ Multiplexing ve demultiplexing
- ▶ Bağlantısız gönderim: UDP
- ▶ UDP segment yapısı
- ▶ UDP checksum

Transport layer servisleri

- ▶ Uygulama ve ağ katmanları arasında yer alan **ulaşım katmanı (transport layer) ağ mimarisinde merkezi rol oynar.**
- ▶ Ağ katmanında yapılan **host'lar arasındaki iletişimi, host'larda çalışan process'ler arasında iletişime dönüştürür.**
- ▶ Farklı hostlarda çalışan process'ler arasında mantıksal bağlantı oluşturur.
- ▶ Gönderici tarafta TCP uygulama katmanından gelen mesajı alır ve transport layer segment'ine dönüştürür.
- ▶ Ağ katmanına gönderilen **segment IP datagram'ına encapsulate edilir.**
- ▶ Transport layer'da birden fazla protokol kullanılabilir (TCP, UDP).
- ▶ **Her protokol uygulama katmanına farklı servis sağlar.**

3

Transport layer servisleri



4

Transport layer servisleri

Transport ve network layer arasındaki ilişki

- ▶ Transport layer protokol yığnında network layer'ın üstündedir.
- ▶ Network layer **host'lar arasında mantıksal iletişim sağlar.**
- ▶ Transport layer **process'ler arasında mantıksal iletişim sağlar.**
- ▶ Transport layer protokolünün sunacağı **servisi network layer protokolü sınırlar.**
- ▶ Network layer'ın sunmadığı servisler transport layer protokolü ile sağlanabilir.
- ▶ Network layer, **kayıp paket, paket bozulması, duplicate paket sorunlarına çözüm üretmez.**
- ▶ Transport layer **reliable iletişim yapabilir.**

5

Transport layer servisleri

Transport ve network layer arasındaki ilişki

- ▶ Benzetim:
 - ▶ İki farklı konutta 12'şer çocuk yaşamaktadır.
 - ▶ Her konuttaki çocukların hepsi diğer konuttaki tüm çocuklara posta servisini kullanarak her hafta mektup gönderiyor.
 - ▶ Her konutta bir çocuk gidecek mektupları topluyor ve gelenleri dağıtıyor.
 - ▶ **Process'ler** -> Çocuklar
 - ▶ **Uygulama mesajları** -> Zarftaki mektuplar
 - ▶ **Host'lar** -> Konutlar
 - ▶ **Transport layer protokolü** -> Her konutta mektup toplama ve dağıtma yapan çocuklar
 - ▶ **Network layer protokolü** -> Posta servisi
 - ▶ Çocukların sunacağı servisi posta servisi sınırlar.

6

Transport layer servisleri

İnternet transport layer protokolleri

- ▶ İnternet (TCP/IP network) application layer için iki farklı transport layer protokolüne sahiptir.
- ▶ User Datagram Protocol (UDP) güvenilir olmayan (unreliable) ve bağlantısız (connectionless) servis sağlar.
- ▶ Transmission Control Protocol (TCP) güvenilir (reliable) ve bağlantı temelli (connection-oriented) servis sağlar.
- ▶ **Ağ uygulaması geliştirirken bu iki protokolden birisi seçilir.**
- ▶ İnternet literatüründe transport layer paketi **TCP için segment, UDP için datagram** olarak adlandırılır.

7

Transport layer servisleri

İnternet transport layer protokolleri

- ▶ İnternet network layer'ın adı IP'dir (Internet Protocol).
- ▶ **IP best-effort gönderim servisi sağlar** (gitmesini garanti etmez).
- ▶ **IP** transport layer'a **unreliable servis sunar.**
- ▶ **TCP ve UDP'nin en temel görevi** IP'nin host-to-host yaptığı iletişimi **process-to-process şekline dönüştür.**
- ▶ İletişimi process-to-process haline dönüştürmek transport-layer **multiplexing** ve **demultiplexing** olarak adlandırılır.
- ▶ **TCP ve UDP segment için hata denetimi yapar.**

8

Transport layer servisleri

- ▶ **TCP'nin sağladığı servisler:**
 - ▶ Reliable data transfer
 - ▶ Connection-oriented
 - ▶ Hata denetimi
 - ▶ Sıralı gönderim
 - ▶ Akış denetimi
 - ▶ Tıkanıklık denetimi
- ▶ **UDP'nin sağladığı servisler:**
 - ▶ Best-effort servis
 - ▶ Connectionless
 - ▶ Hata denetimi
- ▶ **Transport layer'da sağlanamayan servisler:**
 - ▶ Delay garanti
 - ▶ Bant genişliği garantisi

9

İçerik

- ▶ Transport layer servisleri
- ▶ Multiplexing ve demultiplexing
- ▶ Bağlantısız gönderim: UDP
- ▶ UDP segment yapısı
- ▶ UDP checksum

10

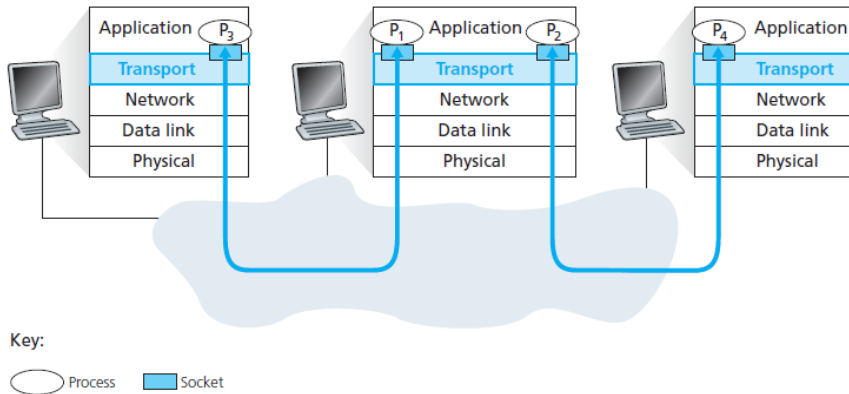
Multiplexing ve demultiplexing

- ▶ Bir host'ta **uygulama katmanında birden fazla process çalışabilir** (3 tane HTTP, 1 tane FTP, 2 tane Telnet, ...).
- ▶ Network layer'dan alınan **paketin doğru process'e yönlendirilmesi gerekir**.
- ▶ Uygulama katmanındaki **her process bir veya birden fazla soket'e sahiptir**.
- ▶ **Alıcı** taraftaki transport layer protokolü aldığı **veriyi soket üzerinden process'e gönderir** (demultiplexing).
- ▶ **Gönderici** taraftaki transport layer protokolü **soketlerden gelen veriyi toplar ve ağ katmanına gönderir** (multiplexing).

11

Multiplexing ve demultiplexing

- ▶ Bir hosttaki process'ler çok sayıda farklı host'taki process'ler ile iletişim yapabilir.

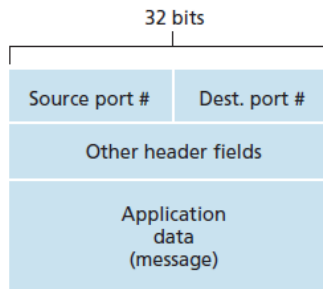


Transport-layer multiplexing and demultiplexing

12

Multiplexing ve demultiplexing

- ▶ **Her soket tekil (unique) tanımlayıcıya sahiptir** (source port number field).
- ▶ **Her segment** hangi sokete gideceğini gösteren **özel alana sahiptir** (destination port number field).
- ▶ Port numarası 16 bittir.
- ▶ **0-1023 arası** bilinen protokoller (HTTP:80, FTP:21) için **reserve edilmiştir** (well-known ports)



Source and destination port-number fields in a transport-layer segment

13

Multiplexing ve demultiplexing

Connectionless multiplexing/demultiplexing

- ▶ Python ile bir UDP soketi aşağıdaki satır ile oluşturulur.

```
clientSocket = socket(socket.AF_INET, socket.SOCK_DGRAM)
```

- ▶ Transport layer otomatik olarak bir port numarası atar (1024-65535).
- ▶ Python ile UDP soketine belirli bir port numarası atanabilir.

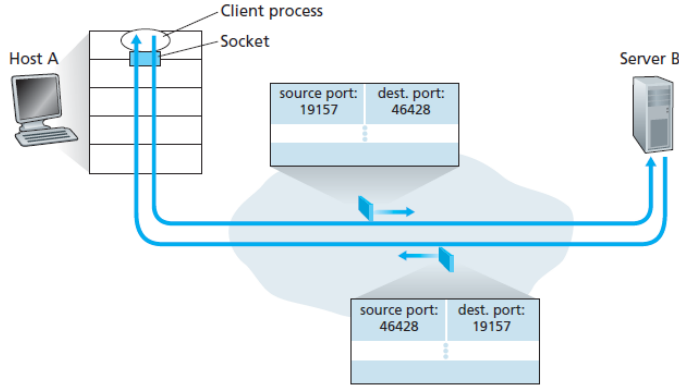
```
clientSocket.bind(('', 19157))
```

14

Multiplexing ve demultiplexing

Connectionless multiplexing/demultiplexing

- ▶ Port numarası atanarak multiplexing/demultiplexing tanımlanmış olur.



The inversion of source and destination port numbers

15

Multiplexing ve demultiplexing

Connection-oriented multiplexing/demultiplexing

- ▶ **TCP socket bir dördlü ile tanımlanır** (source IP, source port, destination IP, destination port).
- ▶ Alıcı host bu dört değer ile segmenti doğru sokete yönlendirir.
- ▶ Server socket **eş zamanlı çok sayıda TCP socketi destekler.**
- ▶ Server beklenen bir sokete **yeni gelen isteği kabul eder ve yeni bir socket açarak yönetir.**

```
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, 12000))
```

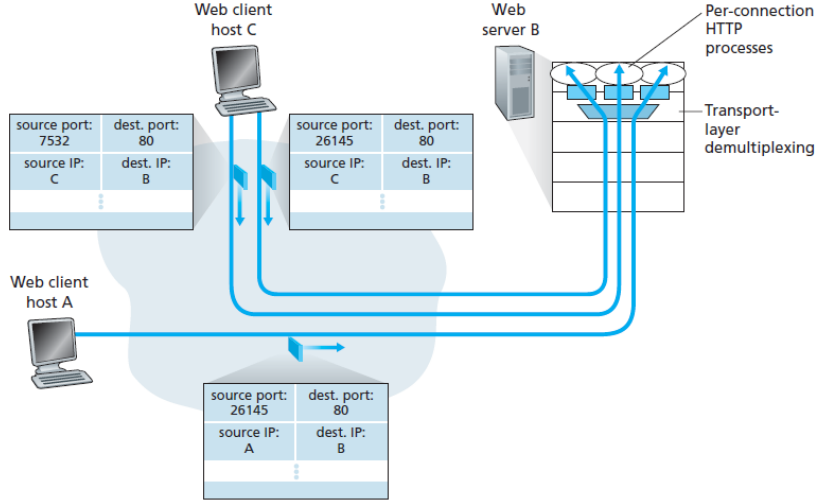
```
connectionSocket, addr = serverSocket.accept()
```

- ▶ Uygulamalar için atanmış belirli port numaraları güvenlik zafiyeti oluşturur mu?

16

Multiplexing ve demultiplexing

Connection-oriented multiplexing/demultiplexing



Two clients, using the same destination port number (80) to communicate with the same Web server application

17

Multiplexing ve demultiplexing

Web sunucu ve TCP

- ▶ **Web sunucu** (Apache Web Server, IIS) **80 portunda çalışır.**
- ▶ İstemciler (browser) sunucuya bir segment gönderdiğinde hedef port 80'dir.
- ▶ Sunucu segment'leri kaynak IP adresleri ve kaynak port numaraları ile birbirinden ayırır.
- ▶ **Web sunucu her bağlantı için bir process oluşturur.**
- ▶ Her yeni process kendisine ait bağlantı soketine sahiptir.
- ▶ Yüksek performanslı Web sunucular **her yeni istemci bağlantısına bir thread (lightweight subprocess)** oluşturur.
- ▶ **HTTP persistent** bağlantı kullanırsa **iletişim süresince aynı soket kullanılır.**
- ▶ **Non-persistent** bağlantı kullanılırsa **her istek/cevap için yeni bir soket oluşturulur.**

18

İçerik

- ▶ Transport layer servisleri
- ▶ Multiplexing ve demultiplexing
- ▶ **Bağlantısız gönderim: UDP**
- ▶ UDP segment yapısı
- ▶ UDP checksum

19

Bağlantısız gönderim: UDP

- ▶ **UDP**, IP protokolüne **basit hata denetimi** ile **multiplexing ve demultiplexing dışında ekleme yapmaz**.
- ▶ **Uygulama** UDP'yi seçerse doğrudan **IP'nin sunduğu servis ile çalışır**.
- ▶ UDP **sadece port numarası ve checksum ekler**.
- ▶ UDP, IP'nin sunduğu serviste değişiklik yapmadan üst katmana sunar.
- ▶ UDP **handshake yapmaz** (**connectionless**).
- ▶ **DNS protokolü sorgu mesajlarını UDP kullanarak gönderir**.

20

Bağlantısız gönderim: UDP

- ▶ Birçok uygulama türü için UDP daha uygundur:
 - ▶ UDP aldığı **veriyi doğrudan ağ katmanına gönderir** (bağlantı kurma, tıkanıklık denetimi vs. yapmaz.)
 - ▶ UDP bağlantı kurmadığı için **gecikme daha azdır ve hızlı gönderim yapar** (TCP daha yavaştır).
 - ▶ UDP **bağlantı durumu tutmaz** (buffer, ACK#, Seq#, CongWindow ve RcvWindow parametreleri gerekmez.).
 - ▶ UDP **daha küçük boyutlu başlık bilgisi içerir** (TCP 20 byte, UDP 8 byte).

21

Bağlantısız gönderim: UDP

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP	TCP
Remote terminal access	Telnet	TCP
Web	HTTP	TCP
File transfer	FTP	TCP
Remote file server	NFS	Typically UDP
Streaming multimedia	typically proprietary	UDP or TCP
Internet telephony	typically proprietary	UDP or TCP
Network management	SNMP	Typically UDP
Routing protocol	RIP	Typically UDP
Name translation	DNS	Typically UDP

Popular Internet applications and their underlying transport protocols

22

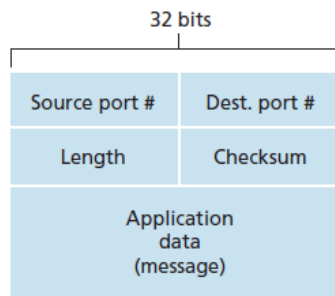
İçerik

- ▶ Transport layer servisleri
- ▶ Multiplexing ve demultiplexing
- ▶ Bağlantısız gönderim: UDP
- ▶ **UDP segment yapısı**
- ▶ UDP checksum

23

UDP segment yapısı

- ▶ **Uygulama verisi UDP'nin data alanına yazılır.**
- ▶ UDP başlık sadece dört alan sahiptir.



UDP segment structure

24

İçerik

- ▶ Transport layer servisleri
- ▶ Multiplexing ve demultiplexing
- ▶ Bağlantısız gönderim: UDP
- ▶ UDP segment yapısı
- ▶ **UDP checksum**

25

UDP checksum

- ▶ UDP checksum ile **uçtan-uca** (iki transport layer arasında) **hata denetimi sağlar.**
- ▶ UDP gönderici segment'teki **16-bit blokları toplar** ve **1 tümleyen alır.**
- ▶ Toplama işleminden taşan bitler sağ tarafa eklenerek tekrar toplanır.
- ▶ Elde edilen **sonuç UDP segment'indeki checksum alanına yazılır.**
- ▶ UDP, **link-by-link** veya **in-memory hatalarına bakmaz.**
- ▶ **UDP, uçtan uca hata denetimi yapar (end-end principle).**

26

UDP checksum

- ▶ Örnek: Üç 16-bit için checksum hesaplanarak segment'e yazılır.

```
0110011001100000
0101010101010101
1000111100001100
```

- ▶ İlk ikisinin toplamı aşağıdadır.

```
0110011001100000
0101010101010101
1011101110110101
```

- ▶ Üçüncünün eklenmesiyle aşağıdaki değer elde edilir.

```
1011101110110101
1000111100001100
10100101011000010
```

- ▶ Sonuç 1 tümleyen alınarak hesaplanır.

```
1011010100111101
```

- ▶ Alıcı segment ile checksum'ı toplar ve 1 tümleyeni hesaplar.
- ▶ **Sonuç 0 ise hatasız** olarak kabul edilir (**hatalı olabilir mi?**).

27

Ödev

- ▶ UDP protokolünde tıkanıklık denetimi yapılmasının gerekliliği hakkında yapılan çalışmalardan bir araştırma ödev hazırlayınız.
- ▶ Transport layer'da UDP kullanıp, application layer'da tıkanıklık denetimi yapmak hakkında detaylı analiz ve değerlendirme yapınız.

28