

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

- ▶ Ağ uygulamalarının temel prensipleri
- ▶ Ağ uygulama mimarileri
- ▶ Process'lerin haberleşmesi
- ▶ Ulaşım katmanı servisleri
- ▶ Uygulama katmanı protokolleri

Ağ uygulamalarının temel prensipleri

- ▶ **1970-1980** arasında İnternet'te **metin tabanlı uygulamalar** popülerdi (haber grupları, remote access, metin tabanlı e-posta, dosya transferi).
- ▶ 1990'lı yılların ortasında **World Wide Web (WWW)** geliştirildi (e-ticaret, Web arama, Web gezinti yaygınlaştı).
- ▶ 2000'li yılların başında **P2P dosya paylaşımı** ve **anlık mesajlaşma** uygulamaları çok yaygınlaştı.
- ▶ 2000'li yıllarda **ses** ve **video uygulamaları** (Skype, VoIP, YouTube, Netflix) popüler oldu.
- ▶ Aynı yıllarda çok oyunculu **online oyunlar** yaygınlaşmaya başladı.
- ▶ Kısa bir süre sonra da **sosyal medya** uygulamaları (Facebook, Twitter) popüler hale geldi.

3

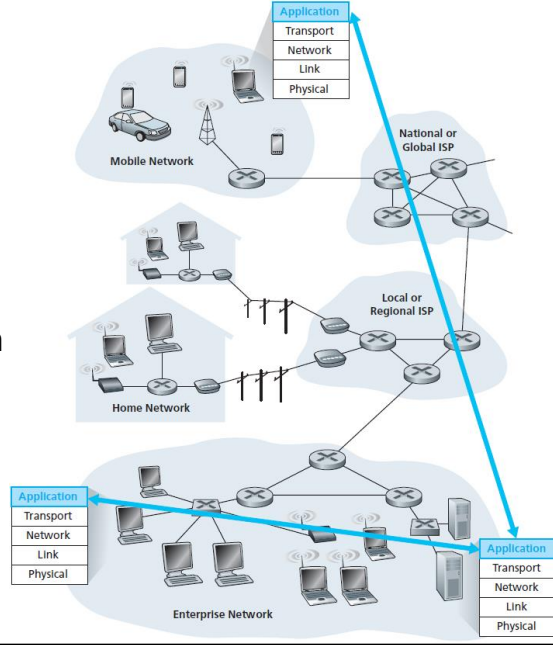
Ağ uygulamalarının temel prensipleri

- ▶ **Application layer'da** uygulamalar için **geliştirilen protokoller**, transport layer protokollerinden servis alırlar.
- ▶ Bir **ağ uygulaması**, iki uç sistem arasında **ağ alt yapısını kullanarak haberleşir**.
- ▶ Bir Web uygulamasında uç sistemlerde **iki ayrı program birbiriyile haberleşir**.
- ▶ Uygulama herhangi bir yüksek seviyeli programlama diliyle (C++, Java, Python) geliştirilebilir.
- ▶ Uygulama geliştirirken **ağın temel bileşenlerinde** (router, link-layer switch) **çalışacak program yazılması gerekmez**.
- ▶ **Geliştirilen uygulamalar application layer'da yazılır**.
- ▶ Yeni uygulamalar için geliştirmenin sadece application layer'da olması **uygulama geliştirme süresini çok kısaltır**.

4

Ağ uygulamalarının temel prensipleri

- ▶ Ağ uygulamaları arasındaki iletişim ağ cihazları kullanılarak uygulama katmanında yapılır.
- ▶ **Mantıksal olarak uygulama katmanındaki programlar doğrudan haberleşiyor gibi düşünülür.**



İçerik

- ▶ Ağ uygulamalarının temel prensipleri
- ▶ Ağ uygulama mimarileri
- ▶ Process'lerin haberleşmesi
- ▶ Ulaşım katmanı servisleri
- ▶ Uygulama katmanı protokolleri

Ağ uygulama mimarileri

- ▶ **Uygulama mimarisi, ağ mimarisinden** (beş katmanlı İnternet mimarisi, vb.) **farklıdır.**
- ▶ Uygulama geliştirici için **ağ mimarisi sabittir** ve uygulamalara bir grup **servis sağlar.**
- ▶ **Uygulama mimarisi, geliştirici tarafından tasarlanır** ve farklı uç sistemlerde uygulamanın **nasıl yapılandırılacağını belirler.**
- ▶ İki temel ağ uygulama mimarisi vardır:
 - ▶ **İstemci-sunucu (client-server) mimarisi**
 - ▶ **P2P (peer-to-peer) mimari**

7

Ağ uygulama mimarileri

İstemci-sunucu mimarisi

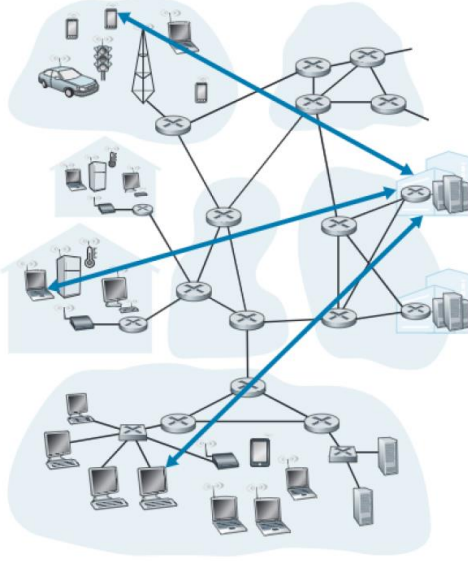
- ▶ İstemci-sunucu mimarisinde, **her zaman çalışır durumda** olan (**always-on host**) **sunucu** bulunur.
- ▶ **Sunucu çok sayıda istemciye servis sunar.**
- ▶ **Web sunucu uygulamaları (IIS, Apache)** always-on host **sunucu bilgisayarlar üzerinde çalışır.**
- ▶ **Web istemci (Browser) client host üzerinde çalışır.**
- ▶ İstemciler birbiriyle doğrudan haberleşmez.
- ▶ **Sunucu host, sabit ve bilinen adrese** (IP adresi veya özel adres) **sahiptir.**
- ▶ **Web, FTP, Telnet** ve **e-posta** istemci-sunucu mimarisine sahip yaygın kullanılan uygulamalardır.

8

Ağ uygulama mimarileri

İstemci-sunucu mimarisi

- ▶ Genellikle bir sunucu, çok sayıdaki istemcinin isteklerini karşılayamaz.
- ▶ Sunucu olarak **çok sayıda host (data center) kullanılabilir.**
- ▶ Bir **data center binlerce sunucu host'tan oluşabilir.**
- ▶ Host'lar dünya üzerinde dağıtık olabilir.
- ▶ Google dünya genelinde 135'den fazla data center'a sahiptir (2021).



9

Ağ uygulama mimarileri

P2P mimari

- ▶ **Host'lar (peers) birbiriyle doğrudan haberleşir.**
- ▶ **Uç sistemler** sunucu kullanmadan **birbiriyle doğrudan haberleşir (peer-to-peer).**
- ▶ Günümüzde, İnternet'te yoğun trafik oluşturan popüler uygulamalar P2P mimariye sahiptir.
- ▶ Yaygın kullanılan P2P uygulamalar:
 - ▶ **Dosya paylaşımı** (BitTorrent)
 - ▶ **İnternet telefon** (Skype)
 - ▶ **IPTV** (PPstream)
- ▶ Bazı uygulamalar hibrit mimariye sahiptir.
- ▶ **Anlık mesajlaşma** uygulamalarında **kullanıcı listesi sunucuda** tutulabilir, **mesaj transferi P2P** yapılır.

10

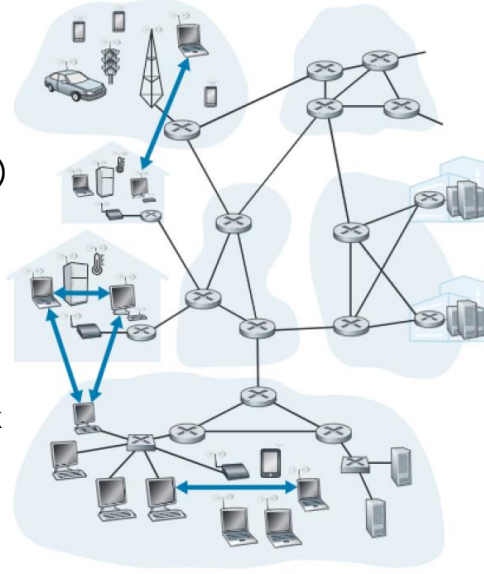
Ağ uygulama mimarileri

P2P mimari

- ▶ P2P uygulamalar **yüksek ölçeklenebilirdir** (self-scalability).
- ▶ P2P mimari **maliyet etkindir** (sunucu ve altyapı maliyeti yok.)

Sorunlar:

- ▶ **ISP friendly:** ISP'ler konut için asimetrik bant genişliği sağlar.
- ▶ **Security:** Yüksek dağıtık ve açık olan yapısı güvenli değildir.
- ▶ **Incentive:** Kullanıcılar **kaynak paylaşımı yapmak** ve **bant genişliği harcamak** için istekli olmalıdır.



11

İçerik

- ▶ Ağ uygulamalarının prensipleri
- ▶ Ağ uygulama mimarileri
- ▶ **Process'lerin haberleşmesi**
- ▶ Ulaşım katmanı servisleri
- ▶ Uygulama katmanı protokolleri

12

Process'lerin haberleşmesi

- ▶ **Process, uç sistemde çalışan programdır.**
- ▶ Uç sistemlerdeki **process'ler haberleşir.**
- ▶ Aynı host üzerindeki process'ler işletim sistemi aracılığıyla haberleşir (**interprocess communication**).
- ▶ **Farklı host'larda çalışan process'ler mesajlar ile ağ üzerinden haberleşir.**
- ▶ Ağ uygulamalarında **iki tür process** vardır:
 - ▶ **Client process:** İletişimi **başlatan process.**
 - ▶ **Server process:** İletişim talebi için **bekleyen process.**

13

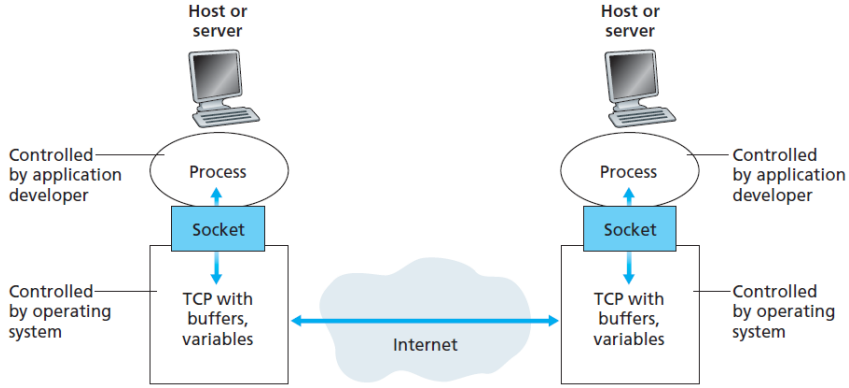
Process'lerin haberleşmesi

- ▶ Web uygulamalarında:
 - ▶ **browser** yazılımı **istemci process**
 - ▶ **web server** yazılımı **sunucu process**
- ▶ P2P uygulamalarda:
 - ▶ **download yapan** yazılım **istemci process**
 - ▶ **upload yapan yazılım sunucu process**
- ▶ P2P uygulamalarda **her peer (process), hem istemci** (talep yaparken) **hem de sunucudur** (talebi cevaplarırken).

14

Process'lerin haberleşmesi

- ▶ Her process ağa göndereceği veya ağdan alacağı mesaj için **socket** kullanır.
- ▶ Bir **socket** uygulama katmanı ile ulaşım katmanı arasında **arayüzdür** (Application Programming Interface - API).



15

Process'lerin haberleşmesi

- ▶ Uygulama **geliştirici, transport layer protokolünü seçer** ve **bazı parametreleri** (buffer size, segment size) **belirler**.
- ▶ Uygulama, transport layer'da seçilen **protokolün sağladığı servisi** (**reliable, unreliable**) tümüyle alır.
- ▶ Bir **alıcı process; host adresi** ve **process adresi** ile adreslenir.
- ▶ İnternet'te **hostlar IP** adresleriyle, **process'ler port** numaralarıyla tanımlanır.
- ▶ Popüler uygulamalar kendilerine ayrılmış port numaralarına (**well-known ports**) sahiptir (Web 80, SMTP 25, FTP 20 ve 21).
- ▶ Geliştirilen protokol ile **istek (request)** ve **cevap (response) mesajları tanımlanır**.
- ▶ Protokol ile **mesaj formatı (syntax)**, **alanların anlamı (semantic)** ve **zamanlama/sıralama (timing) kuralları** tanımlanır.

16

İçerik

- ▶ Ağ uygulamalarının prensipleri
- ▶ Ağ uygulama mimarileri
- ▶ Process'lerin haberleşmesi
- ▶ **Ulaşım katmanı servisleri**
- ▶ Uygulama katmanı protokolleri

17

Ulaşım katmanı servisleri

- ▶ Çoğu ağda, **ulaşım katmanında birden fazla protokol** vardır.
- ▶ **Uygulama geliştirici, ulaşım katmanından beklediği servis kalitesine uygun olan protokolü seçer.**
- ▶ **Ulaşım katmanının** uygulama katmanına **sağladığı servis türleri:**
 - ▶ Reliable data transfer
 - ▶ Throughput
 - ▶ Timing
 - ▶ Security

18

Ulaşım katmanı servisleri

Reliable data transfer

- ▶ **Bir paket, bit bozulması** olduğu için veya **buffer dolu** olduğu için **router** veya **alıcı host'ta atılabilir**.
- ▶ **Web, e-posta** veya **dosya transferi** gibi uygulamalarda **veri kaybı istenmez**.
- ▶ Bir protokol gönderilen verinin **hedefe ulaşmasını garanti ederse**, buna **güvenilir veri aktarımı** (**reliable data transfer**) denir.
- ▶ Güvenilir veri aktarımı yapan protokoller için overhead/payload oranı yüksektir.

19

Ulaşım katmanı servisleri

Reliable data transfer

- ▶ Ulaşım katmanının sağladığı **önemli bir servis türü güvenilir veri aktarımıdır**.
- ▶ Ulaşım katmanı protokolü **güvenilir servis sağlarsa**, gönderici process sadece sokete veriyi gönderir, **alıcıda verinin tamamı alınır**.
- ▶ Ulaşım katmanı **güvenilir servis sağlamazsa**, bazı veriler **hiçbir zaman alıcıya ulaşmayabilir**.
- ▶ Hata toleransına (**loss-tolerant**) sahip uygulamalar **unreliable data transfer** kullanır (multimedia).

20

Ulaşım katmanı servisleri

Throughput

- ▶ İki process arasındaki **throughput** ağdaki dinamik değişime göre **anlık değişim gösterebilir**.
- ▶ **Bazı uygulamalar**, transport layer protokolünün **minimum throughput oranını** sağlamasını **bekler**.
- ▶ **İnternet telefon** için **32kbps throughput** garanti edilmelidir.
- ▶ **Throughput gereksinimi olan uygulamalar** **bandwidth-sensitive** olarak adlandırılır (real-time multimedia).
- ▶ **Bant genişliği hassasiyeti olmayan uygulamalar** **elastic application** olarak adlandırılır (e-posta, file transfer, Web).

21

Ulaşım katmanı servisleri

Timing

- ▶ Timing **bir bitin alıcıya ulaşması için geçen süreyi ifade eder**.
- ▶ Bazı uygulamalar **time-sensitive** olarak adlandırılır. (IP telefon, telekonferans, çok oyunculu oyunlar).
- ▶ **Etkileşimli gerçek zamanlı uygulamaların süre/gecikme hassasiyeti vardır**.

22

Ulaşım katmanı servisleri

Security

- ▶ Transport layer protokolü, **uygulamalara bir veya daha fazla güvenlik servisi sağlayabilir.**
- ▶ **Gönderici** taraftaki transport layer protokolü **veriyi şifreleyebilir.**
- ▶ **Alıcı** taraftaki transport layer protokolü **verinin şifresini çözer.**
- ▶ Transport layer protokolü, **end-point authentication** gibi **servisleri sağlayabilir.**
- ▶ İnternet'te **TCP** ve **UDP** protokolleri uygulama katmanı verisi için **şifreleme servisi sağlamaz.**

23

Ulaşım katmanı servisleri

Security - SSL

- ▶ TCP için **Secure Sockets Layer (SSL)** geliştirilmiştir.
- ▶ **TCP ile SSL, şifreleme ve end-point authentication** sağlar.
- ▶ **SSL**, TCP için geliştirilen ek bir güvenlik servsidir.
- ▶ **Uygulama** SSL kullanırsa, ilkönce **veriyi SSL sokete** gönderir.
- ▶ **SSL, veriyi şifreledikten sonra TCP sokete** gönderir.

24

Ulaşım katmanı servisleri

- ▶ İnternette, **uygulamalara iki temel ulaşım protokolü** sağlanır: **TCP** ve **UDP**.
- ▶ **Her protokol** uygulamalar için **farklı servis sağlar**.
- ▶ Uygulamaların **farklı servis kalitesi (QoS)** gereksinimi vardır.

Application	Data Loss	Throughput	Time-Sensitive
File transfer/download	No loss	Elastic	No
E-mail	No loss	Elastic	No
Web documents	No loss	Elastic (few kbps)	No
Internet telephony/ Video conferencing	Loss-tolerant	Audio: few kbps–1 Mbps Video: 10 kbps–5 Mbps	Yes: 100s of msec
Streaming stored audio/video	Loss-tolerant	Same as above	Yes: few seconds
Interactive games	Loss-tolerant	Few kbps–10 kbps	Yes: 100s of msec
Instant messaging	No loss	Elastic	Yes and no

25

Ulaşım katmanı servisleri

TCP servisleri

- ▶ **Connection-oriented servis:**
 - ▶ Uygulama mesajları iletmeye başlamadan önce, **TCP** process'ler arasında **handshake** yapar.
- ▶ **Reliable data transfer servisi:**
 - ▶ TCP, process'ler arasında **güvenilir iletişim** yapar.
- ▶ **Flow control:**
 - ▶ TCP, alıcı tarafın uyarısıyla **gönderme hızını kontrol** eder.
- ▶ **Congestion control:**
 - ▶ Ağdaki **trafik yoğunluğuna göre** gönderme hızı düzenlenir.
- ▶ **Sağlayamadıkları:**
 - ▶ Zamanlama/gecikme süresi
 - ▶ Minimum bant genişliği garantisi

26

Ulaşım katmanı servisleri

UDP servisleri

- ▶ UDP, gönderici ve alıcı process'ler arasında **güvenilir olmayan iletişim** yapar.
- ▶ UDP, mesajların gitmesini **garanti etmez.**
- ▶ UDP mesajlarının bir kısmı alıcıya ulaştığında **süresi geçmiş olabilir.**
- ▶ **Sağlamadıkları:**
 - ▶ Connection setup
 - ▶ Reliability
 - ▶ Flow control
 - ▶ Congestion control
 - ▶ Timing
 - ▶ Minimum bandwidth

27

Ulaşım katmanı servisleri

- ▶ Popüler İnternet uygulamaları, application layer ve seçilen transport layer protokolleri

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP [RFC 5321]	TCP
Remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
File transfer	FTP [RFC 959]	TCP
Streaming multimedia	HTTP (e.g., YouTube)	TCP
Internet telephony	SIP [RFC 3261], RTP [RFC 3550], or proprietary (e.g., Skype)	UDP or TCP

28

İçerik

- ▶ Ağ uygulamalarının prensipleri
- ▶ Ağ uygulama mimarileri
- ▶ Process'lerin haberleşmesi
- ▶ Ulaşım katmanı servisleri
- ▶ **Uygulama katmanı protokolleri**

29

Uygulama katmanı protokolleri

- ▶ Uygulama katmanı protokolleri, **process'lerin nasıl çalışacağını, birbirlerine mesajları nasıl göndereceğini** tanımlar.
- ▶ Uygulama katmanı protokolleri:
 - ▶ Mesajların **istek** ve **cevap** olarak türlerini belirler (**request, response**).
 - ▶ Mesajların içindeki **alanların dizilimini (syntax)** belirler.
 - ▶ Mesajların içindeki **alanların anlamlarını (semantic)** belirler.
 - ▶ Process'ler arasında **mesaj gönderme** ve **alma kurallarını** tanımlar.

30

Uygulama katmanı protokolleri

- ▶ Uygulama katmanı protokolü, **ağ uygulamasının en önemli bileşenidir.**
- ▶ **Web uygulamasında:**
 - ▶ Web istemci (Firefox, Internet Explorer)
 - ▶ Web sunucu (Apache, IIS)
 - ▶ HTML (doküman formatı standardı)
 - ▶ HTTP (uygulama katmanı protokolü)