

## The Application Layer:

- Application layer protocols
- Transport Layer services required by application layer protocols

**Note:** Many of the lecture slides are based on presentations that accompany *Computer Networking: A Top Down Approach*, 6<sup>th</sup> edition, by Jim Kurose & Keith Ross, Addison-Wesley, 2013.

# Application Layer Protocols

- An application layer protocol defines
  - Types of messages exchanged,
    - e.g., request, response
  - Message syntax:
    - what fields are included in a messages
    - how fields are delineated
  - Message semantics
    - meaning of information in fields
  - Rules for when and how processes send & respond to messages

# Application Layer Protocols

## Public-domain protocols:

- defined in RFCs
- allow for interoperability
  - e.g., HTTP, SMTP, etc.

## Proprietary protocols:

- defined by the application
  - e.g., Skype
  - e.g., the app you will create for Project #1

# “Lower” layer services required by *Application Layer* protocols

## Data integrity

- some apps (e.g., audio) can tolerate some loss
- other apps (e.g., file transfer, telnet) require 100% reliable data transfer

## Timing

- some apps (e.g., Internet telephony, interactive games) require low delay to be “effective”

## Bandwidth

- some apps (e.g., multimedia) require a certain amount of bandwidth to be “effective”
- other apps (“elastic apps”) make use of whatever bandwidth they get

## Security

- How big a problem is this?

# “Lower” layer services required by common applications

Application	Allowed Data loss	Bandwidth	Time Sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
web documents	no loss	elastic	no
streaming audio/video	loss-tolerant	audio: 5kbps-1Mbps video: 10kbps-5Mbps	yes, 100' s
stored audio/video	loss-tolerant	same as above	msec
interactive games	loss-tolerant	few kbps and up	yes, few secs
instant messaging	no loss	elastic	yes, 100' s msec yes and no

# Recall (Lecture #2): What services do Internet transport protocols provide?

## TCP service:

- *connection-oriented*: setup required between client and server processes
- *reliable transport* between sending and receiving process
- *flow control*: sender won't overwhelm receiver
- *congestion control*: throttle sender when network overloaded
- *does not provide*: timing, minimum bandwidth guarantees

## UDP service:

- “unreliable” (best effort) data transfer between sending and receiving process
- does not provide: connection setup, reliability, flow control, congestion control, timing, or bandwidth guarantee

**Discussion question:** Why is there UDP?

# Internet applications: application and transport protocols

Application	Application layer protocol	Underlying transport protocol
e-mail	SMTP [RFC 2821]	TCP
remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	proprietary (e.g. RealNetworks)	TCP or UDP
Internet telephony	proprietary (e.g., Vonage, Dialpad)	typically UDP
Domain name services	DNS	UDP

Much more later on transport layer protocols ...

- Application Layer Protocols
  - Responsibilities
  - Examples
- Transport Layer services required by application layer protocols
  - Reliability, timing, security, etc.
  - Examples