Application Layer

Prof. Anja Feldmann, Ph.D.

(Based on slide deck of Computer Networking, 7th ed., Jim Kurose and Keith Ross.)

Application Layer

Goals

- Conceptual, implementation aspects
- Implementation paradigms
  - *Client-server and Peer-to-peer*
  - *Sockets*
- Transport-layer service models

Application Layer

Goals

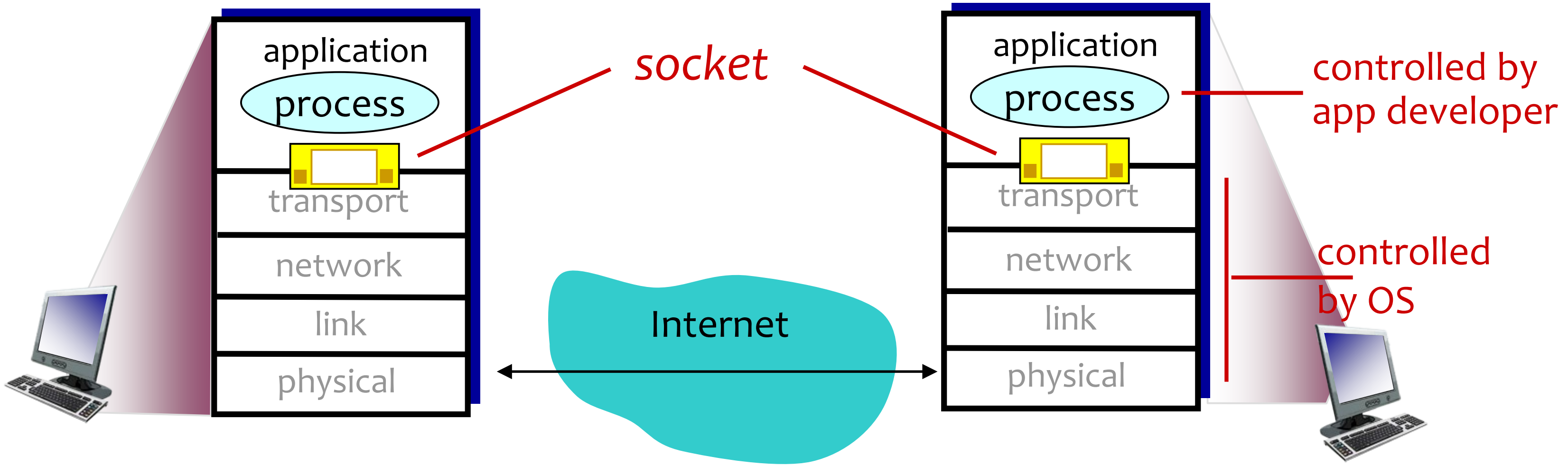
- Conceptual, implementation aspects
- Implementation paradigms
  - *Client-server and Peer-to-peer*
  - **Sockets**
- Transport-layer service models

Sockets

- Process *sends/receives* messages *to/from* its ***socket***

- Socket: analogous to door
  - Sending process
    1. Shoves message out the door
    2. Relies on transport infrastructure on other side of door to deliver message to socket at receiving process

Sockets


application
process
transport
network
link
physical
socket
Internet
application
process
transport
network
link
physical
controlled by
app developer
controlled
by OS

Addressing Sockets

- To receive messages, process ***must*** have ***identifier***
- Host device has unique ***32-bit IP address***

Does IP address of host on which process runs suffice for identifying the process?

No! *Many processes can be running on same host*

Addressing Sockets

- ***Identifier*** includes *both* IP address and port numbers associated with process on host.

Example port numbers: 80 (HTTP server), 25 (mail server)

To send a HTTP messages to a server, we need the IP address and the port number of the HTTP (server) process running on the server

Application Layer

Goals

- Conceptual, implementation aspects
- Implementation paradigms
    - *Client-server and Peer-to-peer*
    - Sockets
- **Transport-layer service models**

Application-Layer Protocol defines …

- ***Types*** of messages exchanged,
  - E.g., *request*, *response*
- Message ***syntax***:
  - *Fields* and *structure*
- Message ***semantics***
  - Meaning of field *X*?
- ***Rules***
  - *When* as well as *how* processes send & respond to messages

Open & Proprietary Protocols

- ***Open*** protocols:
  - Defined in ***RFCs***
  - Allows for interoperability
  - *E.g., HTTP, SMTP*

- ***Proprietary*** protocols:
  - *e.g., Skype*

What transport service does an app need?

- ***Data Integrity***
  - Some apps *(e.g., file transfer, web transactions)* require ***100% reliable*** data transfer
  - Other apps *(e.g., audio)* can tolerate some *loss*
- ***Timing***
  - Some apps *(e.g., Internet telephony, interactive games)* require low delay to be "*effective*"
- ***Throughput***
  - Some apps *(e.g., multimedia)* require minimum amount of throughput to be "*effective*"
  - Other "*elastic apps*" make use of whatever throughput they get
- ***Security***
  - Encryption, data integrity, …

Transport Service Reqs.: Common Apps

application	*data loss*	*throughput*	*time sensitive*
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	no loss	elastic	no
real-time audio/video	loss-tolerant	audio: 5 Kbps-1 Mbps video:10 Kbps-5 Mbps	yes, 100’s ms
stored audio/video	loss-tolerant	same as above	yes, few secs
interactive games	loss-tolerant	few kbps up	yes, 100’s ms
text messaging	no loss	elastic	yes and no

Internet Transport Protocols Services

TCP service:

- *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 throughput guarantee, security
- *Connection-oriented:* Setup required between client and server processes

UDP service:

- *Unreliable data transfer:* Between sending and receiving process
- *Does not provide:* Reliability, flow control, congestion control, timing, throughput guarantee, security, or connection setup

Why bother? Why is there a UDP?

Internet apps: application, 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	HTTP *(e.g., YouTube)*, RTP [RFC 1889]	TCP or UDP
Internet telephony	SIP, RTP, proprietary *(e.g., Skype)*	TCP or UDP

Application Layer

Goals

- Conceptual, implementation aspects
- Implementation paradigms
  - *Client-server* *and* *Peer-to-peer*
  - *Sockets*
- Transport-layer service models