

# Computer Networks

## Transport Layer Overview

(§6.1.2-6.1.4)



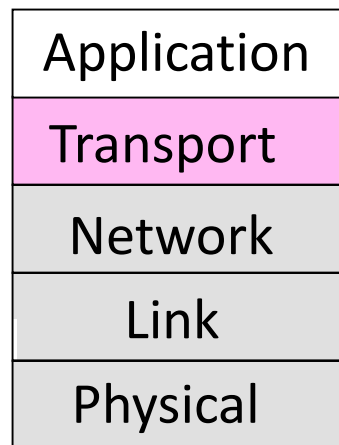
David Wetherall (djw@uw.edu)

Professor of Computer Science & Engineering

UNIVERSITY *of* WASHINGTON

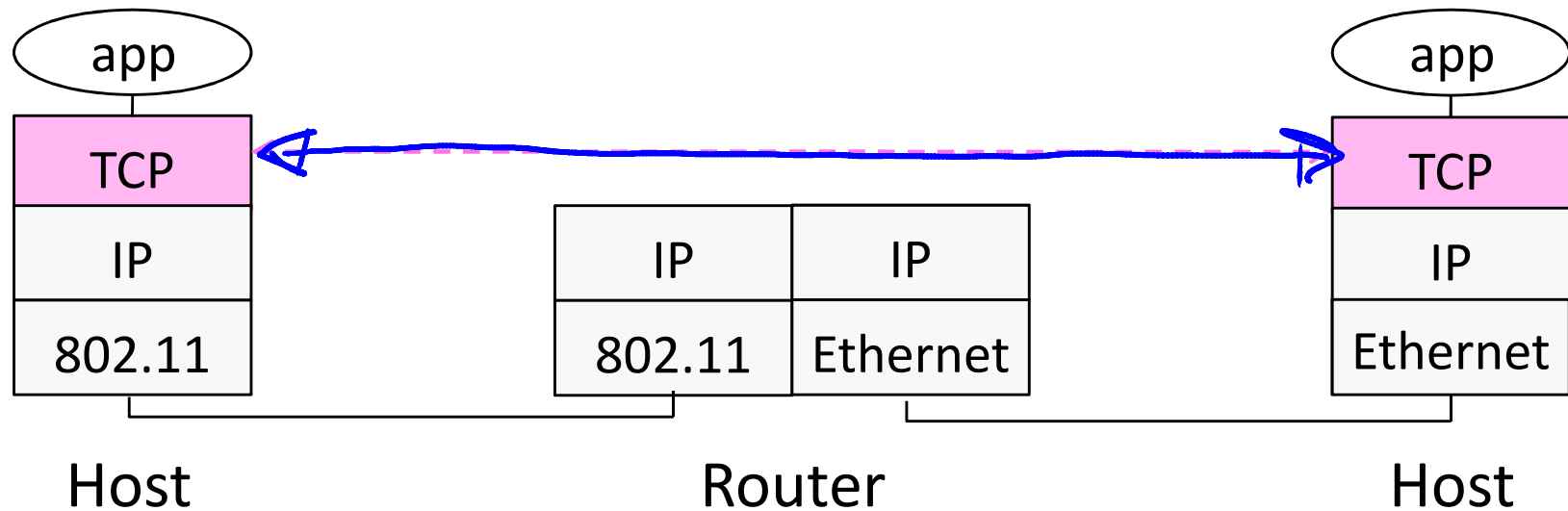
# Where we are in the Course

- Starting the Transport Layer!
  - Builds on the network layer to deliver data across networks for applications with the desired reliability or quality



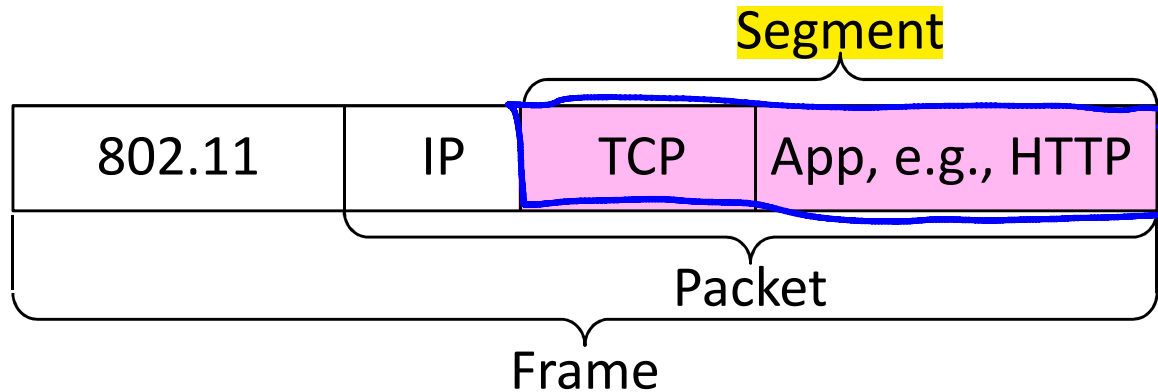
# Recall

- Transport layer provides end-to-end connectivity across the network



## Recall (2)

- Segments carry application data across the network
- Segments are carried within packets within frames



# Transport Layer Services

- Provide different kinds of data delivery across the network to applications

	Unreliable	Reliable
Messages	Datagrams (UDP)	
Bytestream		Streams (TCP)

# Comparison of Internet Transports

- TCP is full-featured, UDP is a glorified packet

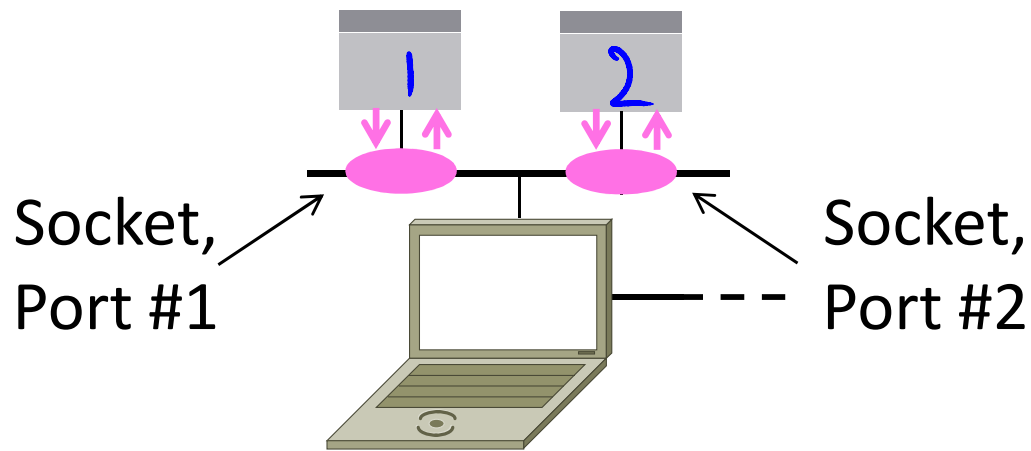
TCP (Streams)	UDP (Datagrams)
Connections	Datagrams
Bytes are delivered once, reliably, and in order	Messages may be lost, reordered, duplicated
Arbitrary length content	Limited message size
Flow control matches sender to receiver	Can send regardless of receiver state
Congestion control matches sender to network	Can send regardless of network state

# Socket API

- Simple abstraction to use the network
  - The “network” API (really Transport service) used to write all Internet apps
  - Part of all major OSes and languages; originally Berkeley (Unix) ~1983
- Supports both Internet transport services (Streams and Datagrams)  
*connection* *connectionless*

# Socket API (2)

- Sockets let apps attach to the local network at different ports





# Socket API (3)

- Same API used for Streams and Datagrams

	Primitive	Meaning
	SOCKET	Create a new communication endpoint
	BIND	Associate a local address (port) with a socket
Only needed for Streams	LISTEN	Announce willingness to accept connections
	ACCEPT	Passively establish an incoming connection
	CONNECT	Actively attempt to establish a connection
To/From forms for Datagrams	<u>SEND(TO)</u>	Send some data over the socket
	<u>RECEIVE(FROM)</u>	Receive some data over the socket
	CLOSE	Release the socket

# Ports

- Application process is identified by the tuple IP address, protocol, and port
  - Ports are 16-bit integers representing local “mailboxes” that a process leases
- Servers often bind to “well-known ports”
  - $<1024$ , require administrative privileges
- Clients often assigned “ephemeral” ports
  - Chosen by OS, used temporarily

# Some Well-Known Ports

Port	Protocol	Use
20, 21	FTP	File transfer
22	SSH	Remote login, replacement for Telnet
25	SMTP	Email
80	HTTP	World Wide Web
110	POP-3	Remote email access
143	IMAP	Remote email access
443	HTTPS	Secure Web (HTTP over SSL/TLS)
543	RTSP	Media player control
631	IPP	Printer sharing

# Topics

- Service models
    - Socket API and ports
    - Datagrams, Streams
  - User Datagram Protocol (UDP)
  - Connections (TCP)
  - Sliding Window (TCP)
  - Flow control (TCP)
  - Retransmission timers (TCP)
  - Congestion control (TCP)
- ✓
- This time
- Next time
- Later

# END

© 2013 D. Wetherall

Slide material from: TANENBAUM, ANDREW S.; WETHERALL, DAVID J., COMPUTER NETWORKS, 5th Edition, © 2011.  
Electronically reproduced by permission of Pearson Education, Inc., Upper Saddle River, New Jersey