Last time

- Router internals
  - Input ports, switching fabric, output ports
  - Switching via memory, bus, crossbar
  - Queueing, head-of-line blocking
- Mobility
  - Home, visited networks
  - Home, foreign agents
  - Permanent, care-of addresses
  - Indirect vs. direct routing
- Mobile IP
  - tunneling, agent discovery, registration

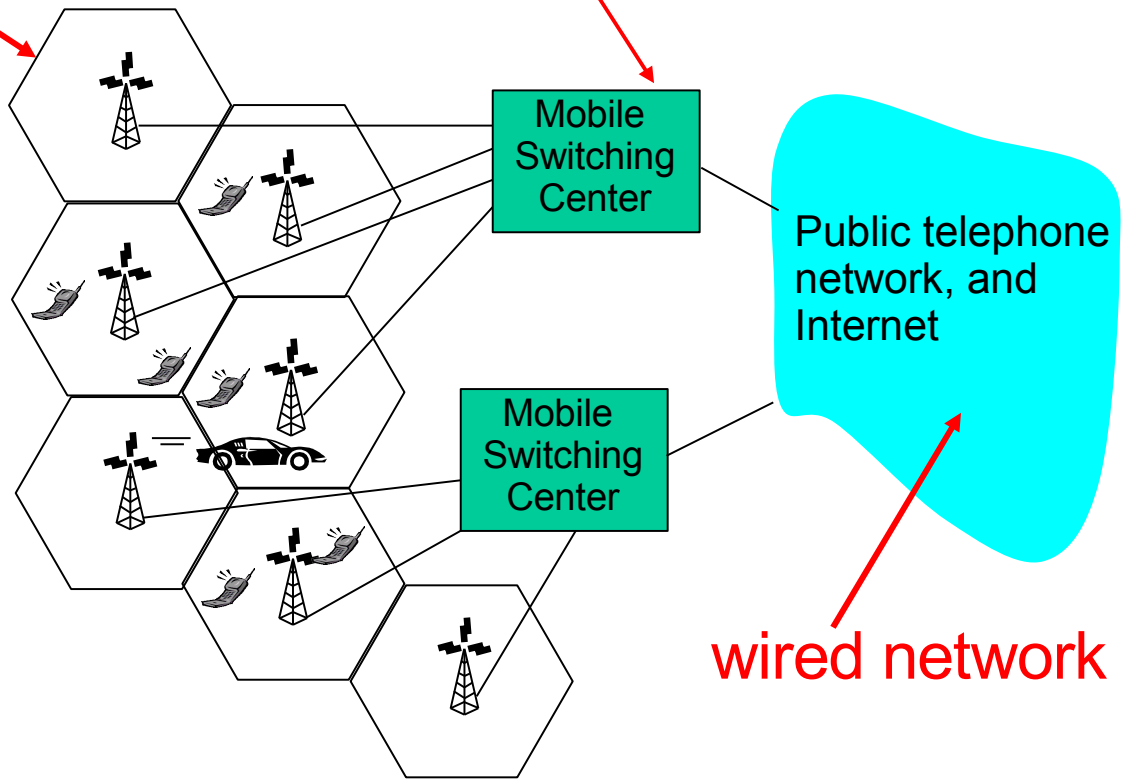
This time

- Mobility in Cellular Networks
- Transport layer
  - Introduction
  - Multiplexing
  - UDP

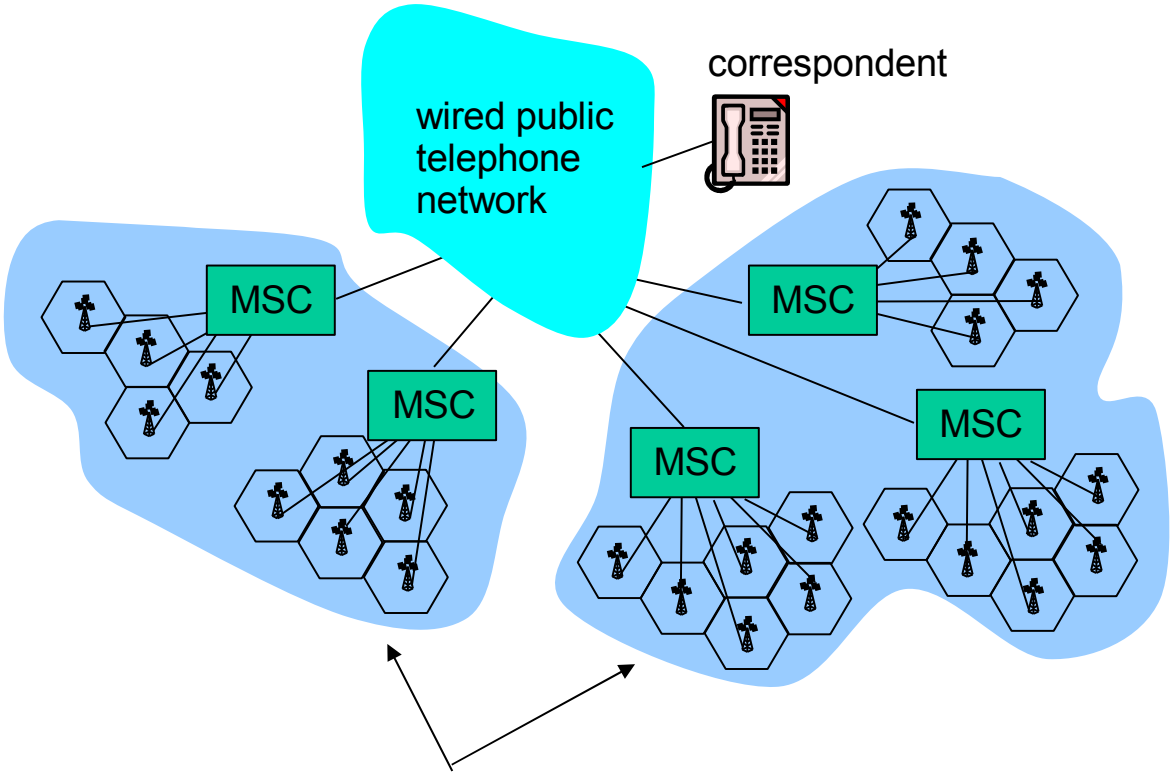
Components of cellular network architecture

MSC
- connects cells to wide area net
- manages call setup
- handles mobility

cell
- covers geographical region
- *base station* (BS) analogous to 802.11 AP
- *mobile users* attach to network through BS
- *air-interface:* physical and link layer protocol between mobile and BS



Components of cellular network architecture

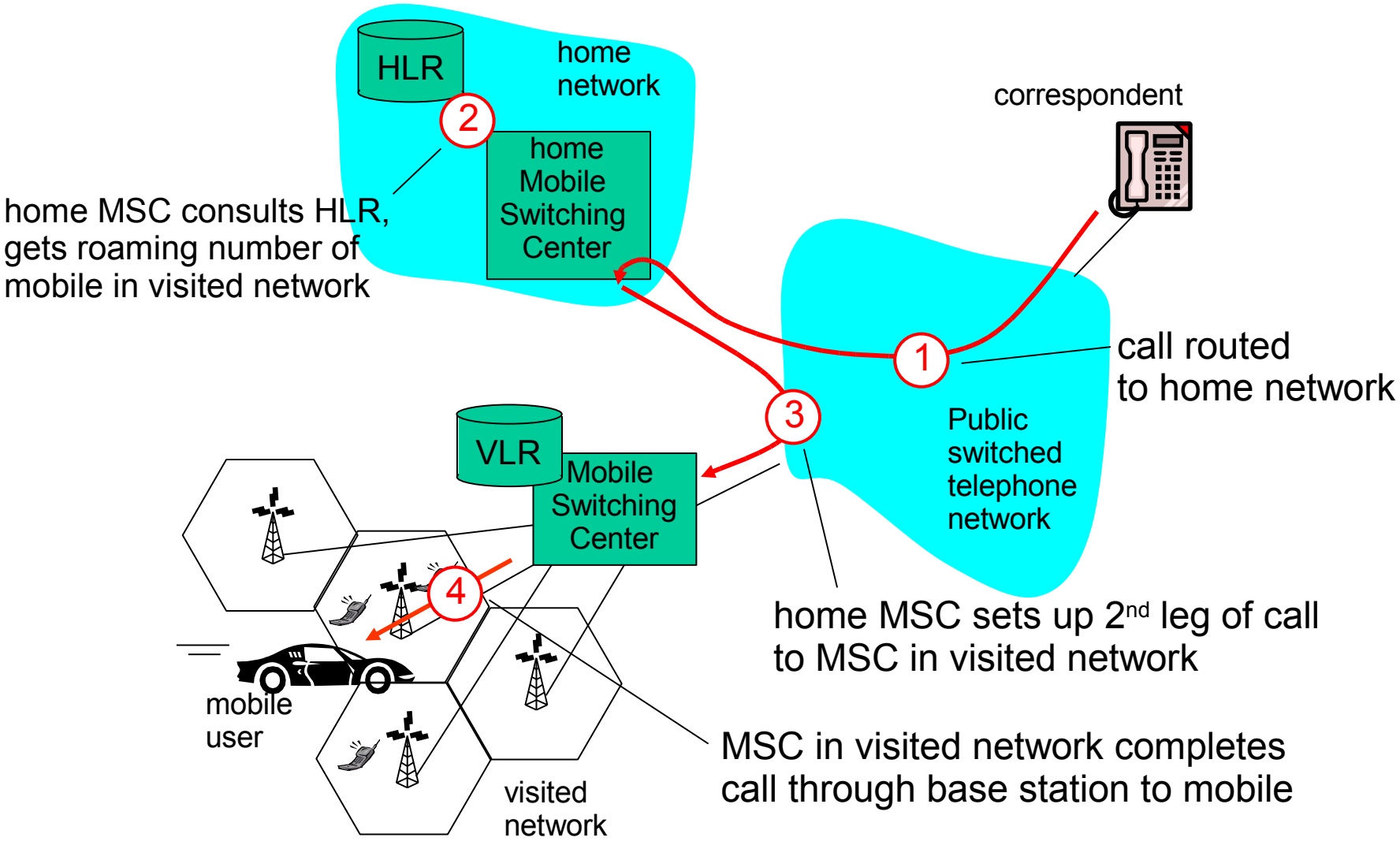


Handling mobility in cellular networks

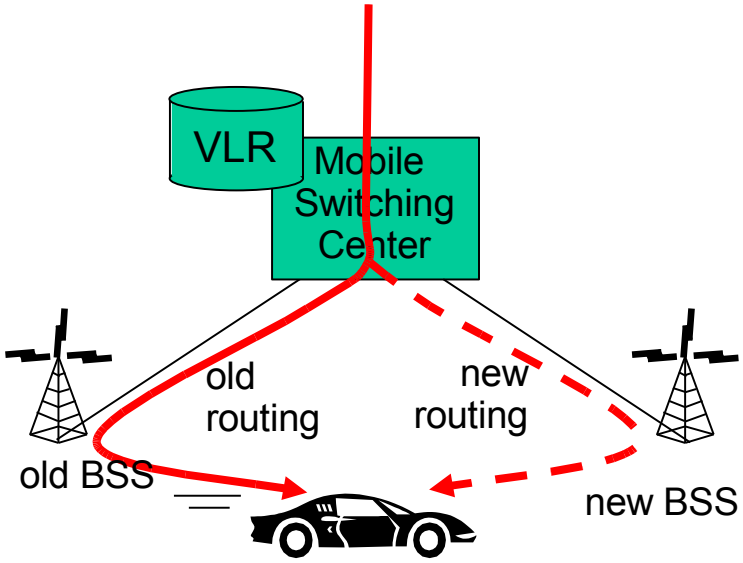
- *Home network:* network of cellular provider you subscribe to (e.g., Sprint PCS, Verizon)
  - *home location register (HLR):* database in home network containing permanent cell phone #, profile information (services, preferences, billing), information about current location (could be in another network)

- *Visited network:* network in which mobile currently resides
  - *visitor location register (VLR):* database with entry for each user currently in network
  - could be home network

GSM: indirect routing to mobile

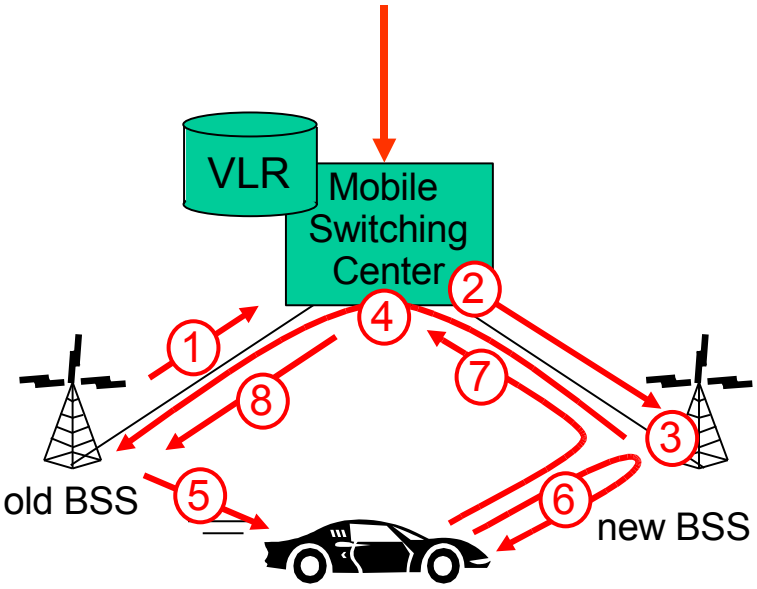


GSM: handoff with common MSC



- Handoff goal: route call via new base station (without interruption)
- Reasons for handoff:
  - stronger signal to/from new BSS (continuing connectivity, less battery drain)
  - load balance: free up channel in current BSS
  - GSM doesn't mandate why to perform handoff (policy), only how (mechanism)
- Handoff initiated by old BSS

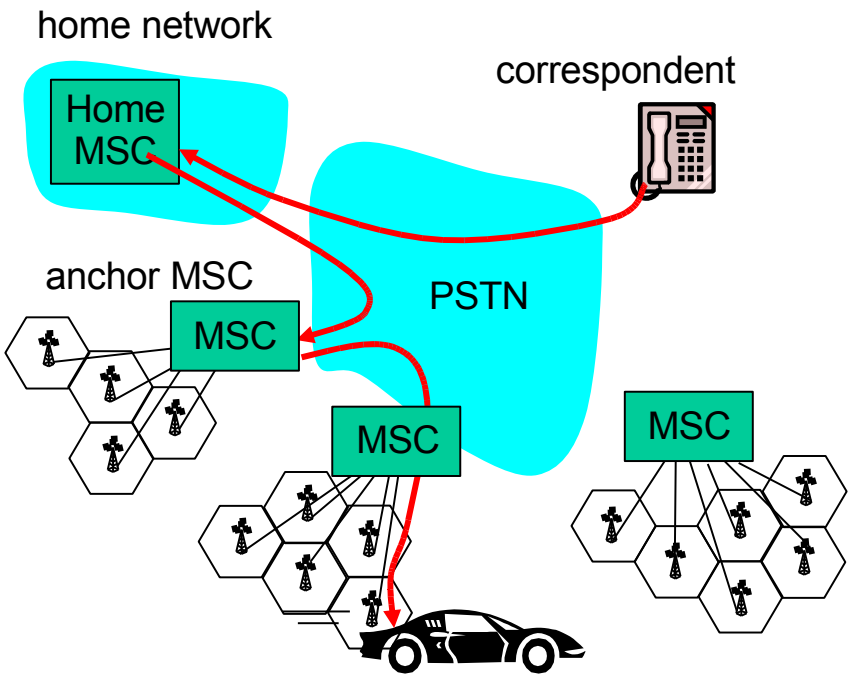
GSM: handoff with common MSC



1. old BSS informs MSC of impending handoff, provides list of 1+ new BSSs
2. MSC sets up path (allocates resources) to new BSS
3. new BSS allocates radio channel for use by mobile
4. new BSS signals MSC, old BSS: ready
5. old BSS tells mobile: perform handoff to new BSS
6. mobile, new BSS signal to activate new channel
7. mobile signals via new BSS to MSC: handoff complete. MSC reroutes call

8 MSC-old-BSS resources released

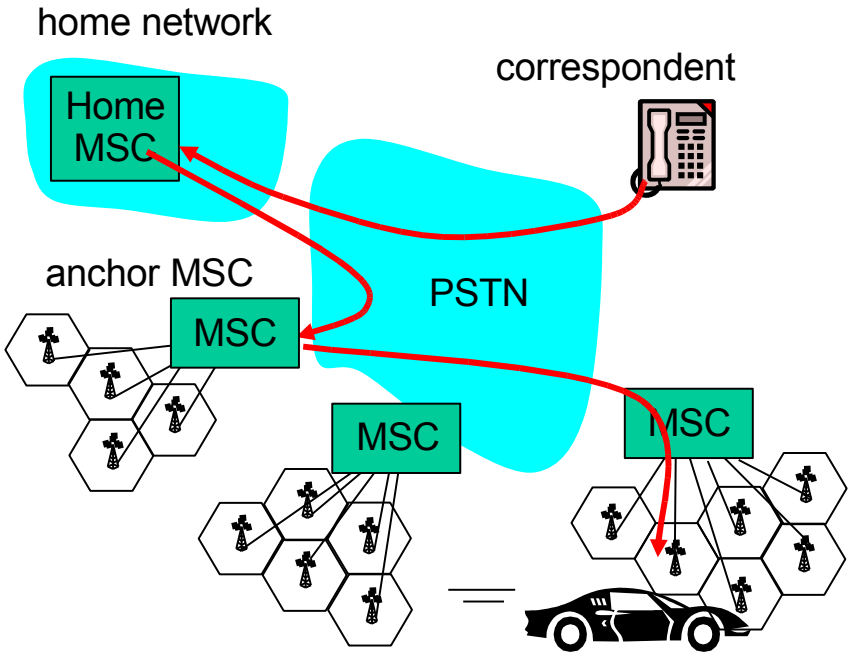
GSM: handoff between MSCs



(a) before handoff

- *Anchor MSC:* first MSC visited during call
  - call remains routed through anchor MSC
- Anchor MSC sends call to *current* visited BSS
- Always at most 3 hops
  - Home MSC
  - Anchor MSC
  - Visited MSC

GSM: handoff between MSCs



(b) after handoff

- *Anchor MSC:* first MSC visited during call
  - call remains routed through anchor MSC
- Anchor MSC sends call to *current* visited BSS
- Always at most 3 hops
  - Home MSC
  - Anchor MSC
  - Visited MSC

Mobility: GSM versus Mobile IP

GSM element	Comment on GSM element	Mobile IP element
Home system	Network to which the mobile user's permanent phone number belongs	Home network
Gateway Mobile Switching Center, or "home MSC". Home Location Register (HLR)	Home MSC: point of contact to obtain routable address of mobile user. HLR: database in home system containing permanent phone number, profile information, current location of mobile user, subscription information	Home agent
Visited System	Network other than home system where mobile user is currently residing	Visited network
Visited Mobile services Switching Center. Visitor Location Record (VLR)	Visited MSC: responsible for setting up calls to/from mobile nodes in cells associated with MSC. VLR: temporary database entry in visited system, containing subscription information for each visiting mobile user	Foreign agent
Mobile Station Roaming Number (MSRN), or "roaming number"	Routable address for telephone call segment between home MSC and visited MSC, visible to neither the mobile nor the correspondent.	Care-of-address

Wireless, mobility: impact on higher layer protocols

- Logically, impact *should* be minimal …
  - best effort service model remains unchanged
  - TCP and UDP can (and do) run over wireless, mobile

- … but performance-wise:
  - packet loss/delay due to bit-errors (discarded packets, delays for link-layer retransmissions), and handoff
  - TCP interprets loss as congestion, will decrease congestion window un-necessarily
  - delay impairments for real-time traffic
  - limited bandwidth of wireless links

Chapter 3: Transport Layer

Our goals:

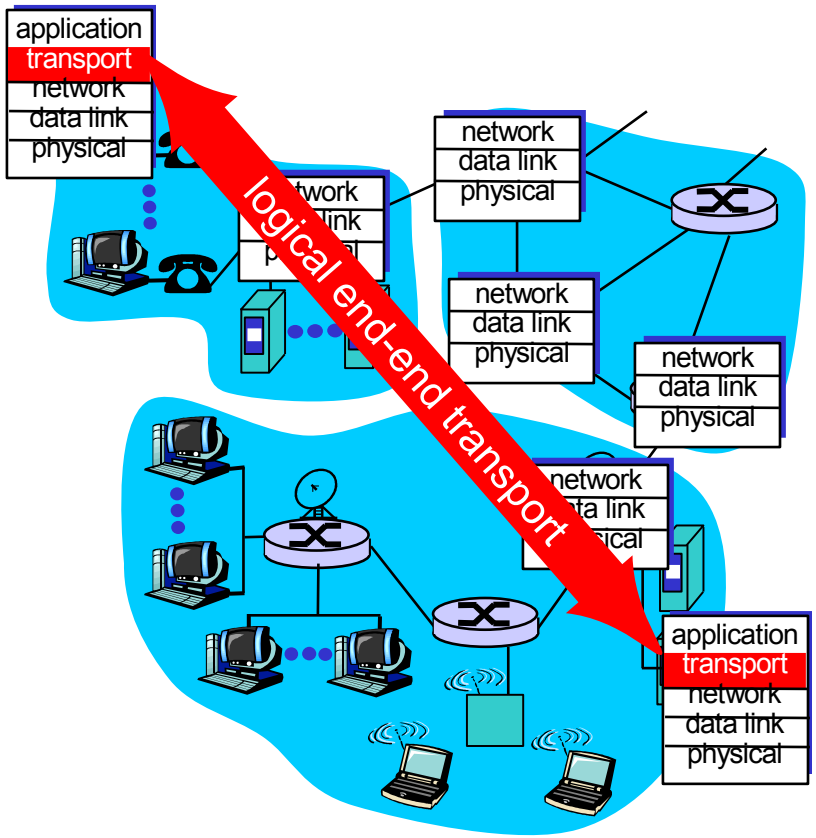
- Understand principles behind transport layer services:
  - multiplexing/ demultiplexing
  - reliable data transfer
  - flow control
  - congestion control
- Learn about transport layer protocols in the Internet:
  - UDP: connectionless transport
  - TCP: connection-oriented transport
  - TCP congestion control

Chapter 3 outline

- 3.1 Transport-layer services
- 3.2 Multiplexing and demultiplexing
- 3.3 Connectionless transport: UDP
- 3.4 Principles of reliable data transfer
- 3.5 Connection-oriented transport: TCP
  - segment structure
  - reliable data transfer
  - flow control
  - connection management
- 3.6 Principles of congestion control
- 3.7 TCP congestion control

Transport services and protocols

- Provide *logical communication* between app processes running on different hosts
- Transport protocols run in end systems
  - sender side: breaks app messages into segments, passes to network layer
  - receiver side: reassembles segments into messages, passes to app layer
- More than one transport protocol available to apps
  - Internet: TCP and UDP



Transport vs. network layer

- *Network layer:* logical communication between hosts
- *Transport layer:* logical communication between processes
  - relies on, enhances, network layer services

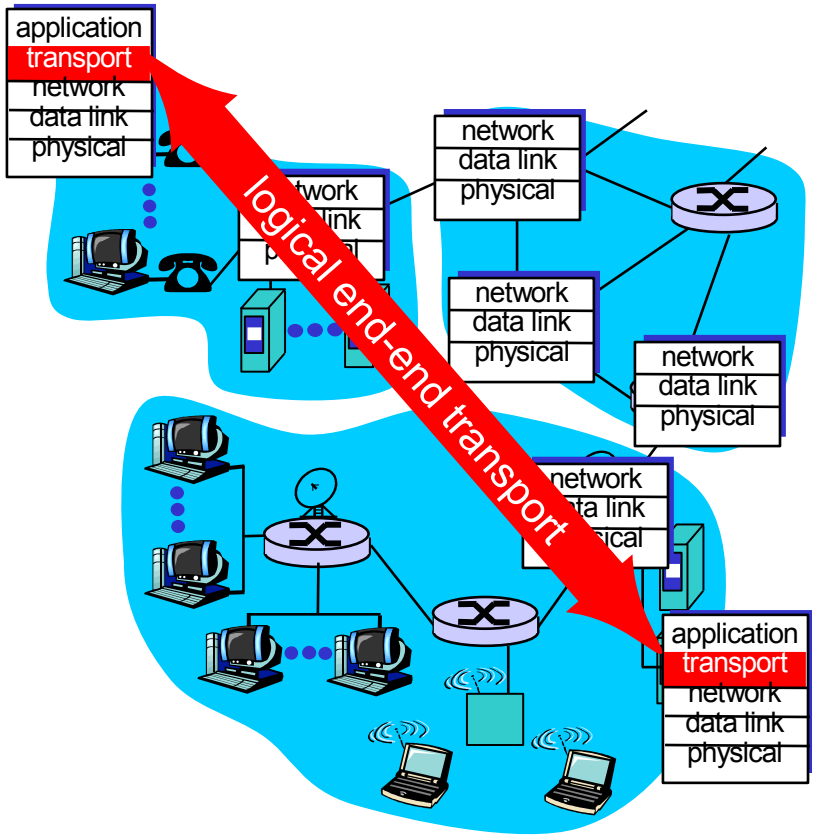
Household analogy:

12 kids sending letters to 12 kids

- processes = kids
- app messages = letters in envelopes
- hosts = houses
- transport protocol = Ann and Bill
- network-layer protocol = postal service

Internet transport-layer protocols

- Reliable, in-order delivery (TCP)
  - congestion control
  - flow control
  - connection setup
- Unreliable, unordered delivery: UDP
  - no-frills extension of “best-effort” IP
- Services not available:
  - delay guarantees
  - bandwidth guarantees



Chapter 3 outline

- 3.1 Transport-layer services
- 3.2 Multiplexing and demultiplexing
- 3.3 Connectionless transport: UDP
- 3.4 Principles of reliable data transfer
- 3.5 Connection-oriented transport: TCP
  - segment structure
  - reliable data transfer
  - flow control
  - connection management
- 3.6 Principles of congestion control
- 3.7 TCP congestion control

Multiplexing/demultiplexing

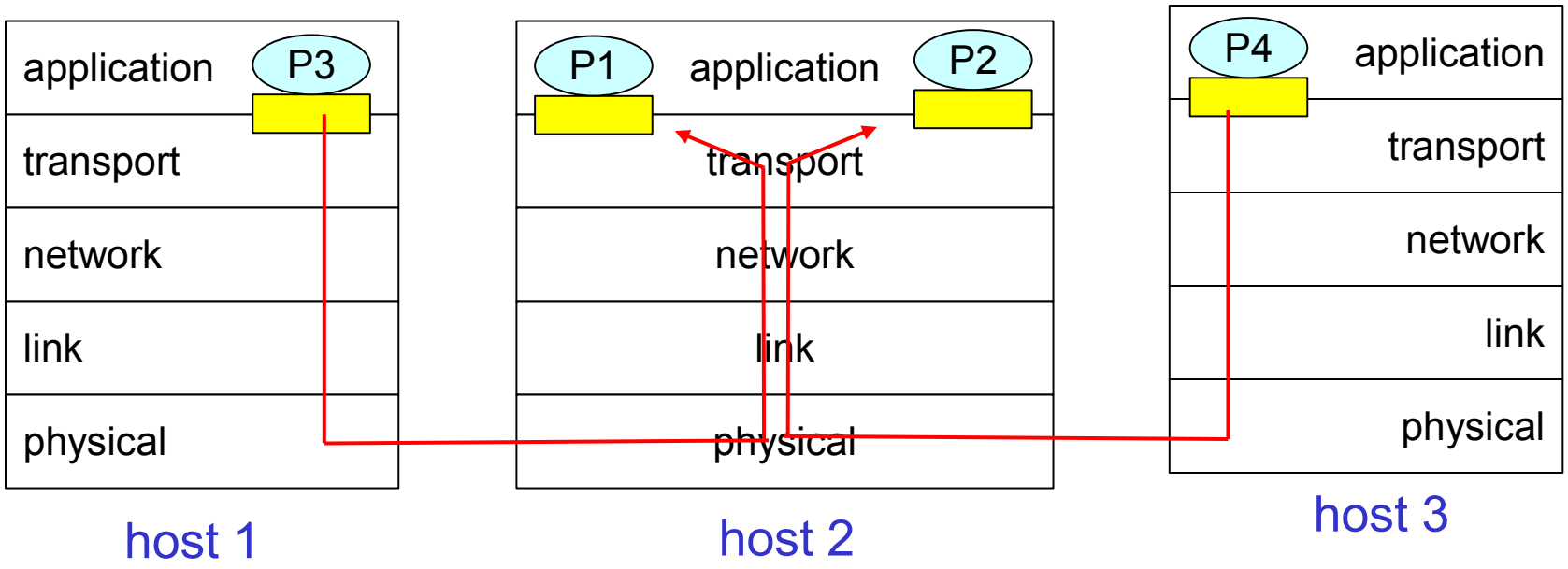
Demultiplexing at rcv host:

delivering received segments to correct socket

Multiplexing at send host:

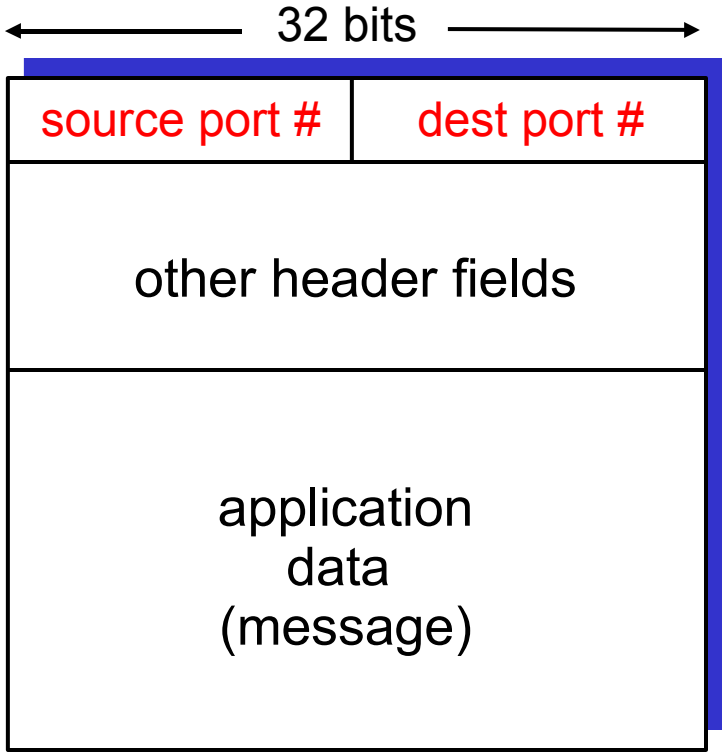
gathering data from multiple sockets, enveloping data with header (later used for demultiplexing)

= socket = process



How demultiplexing works

- Host receives IP datagrams
  - each datagram has source IP address, destination IP address
  - each datagram carries 1 transport-layer segment
  - each segment has source, destination port number
- Host uses IP addresses & port numbers to direct segment to appropriate socket



TCP/UDP segment format

Connectionless demultiplexing

- Create sockets with port numbers:

```
DatagramSocket mySocket1 = new
   DatagramSocket(12534);
DatagramSocket mySocket2 = new
   DatagramSocket(12535);
```

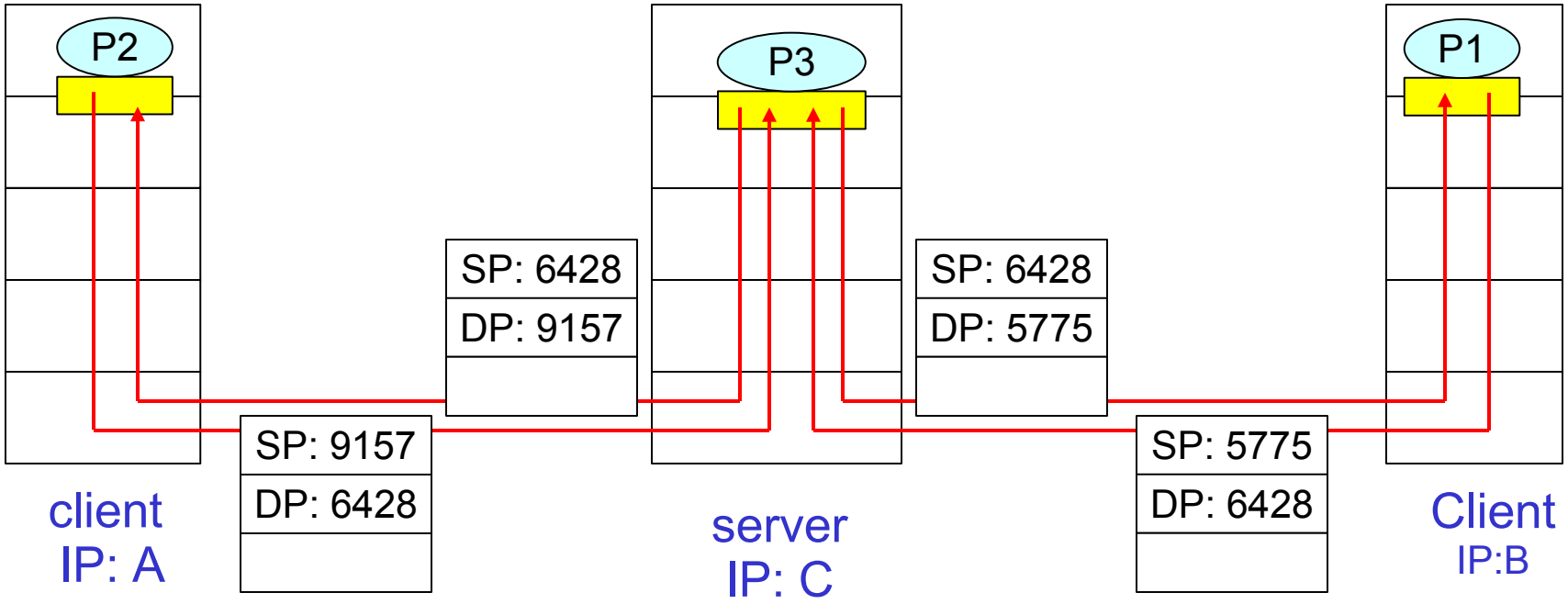
- UDP socket identified by two-tuple:

(dest IP address, dest port number)

- When host receives UDP segment:
  - checks destination port number in segment
  - directs UDP segment to socket with that port number
- IP datagrams with different source IP addresses and/or source port numbers directed to same socket

Connectionless demux (cont)

```
DatagramSocket serverSocket = new DatagramSocket(6428);
```

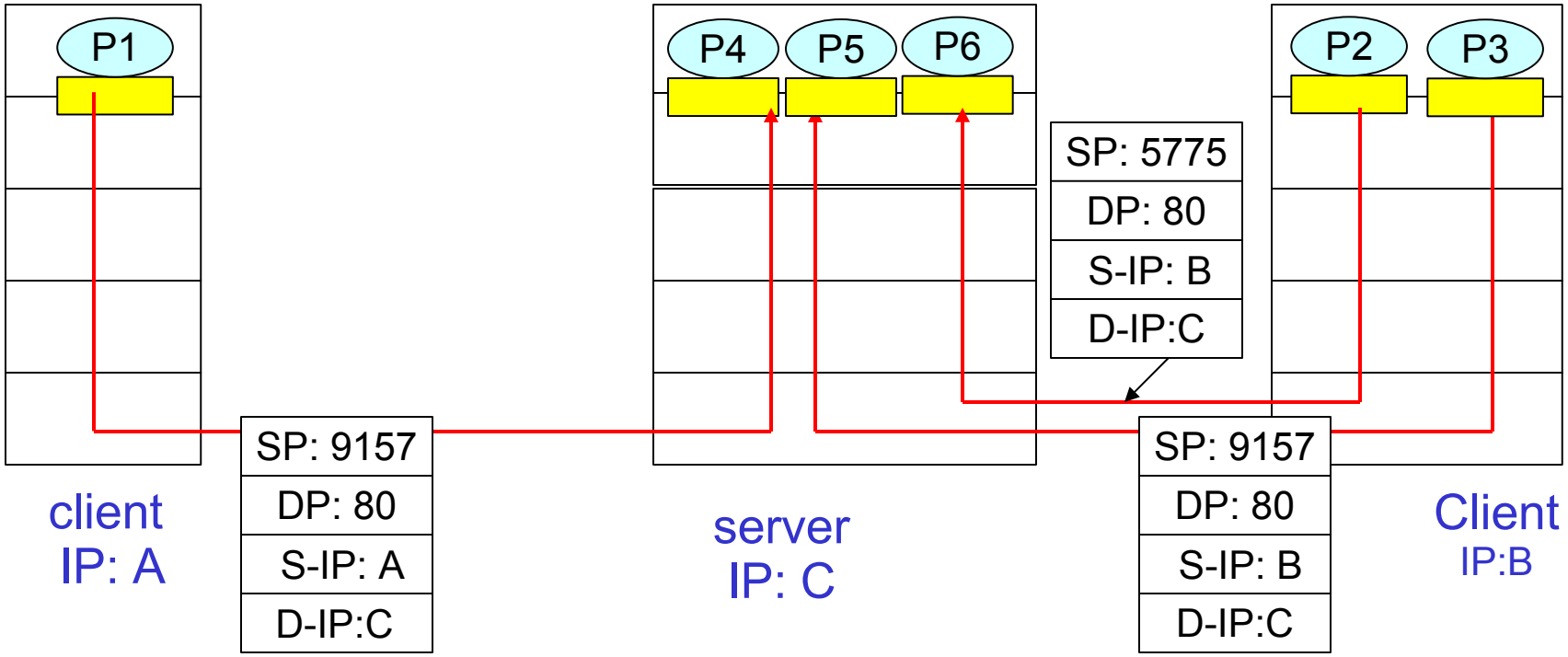


SP provides “return address”

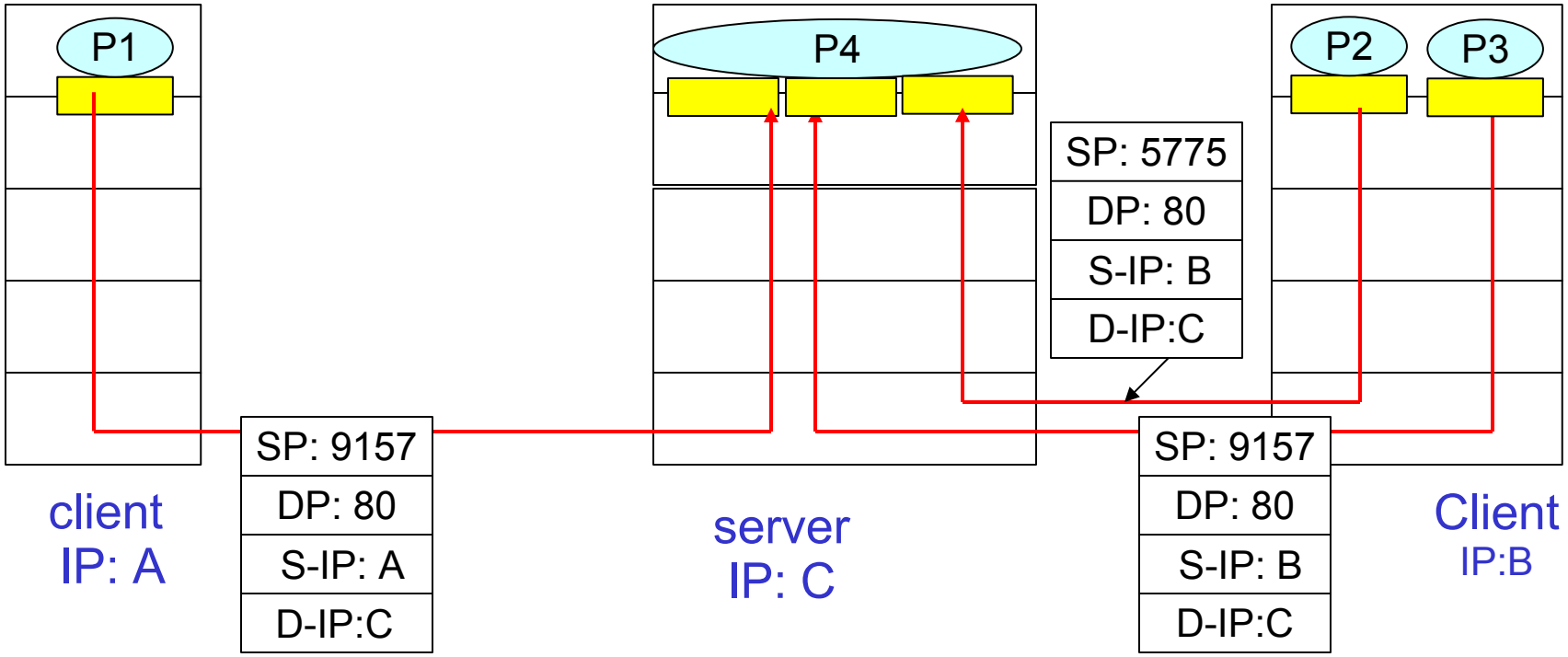
Connection-oriented demux

- TCP socket identified by 4-tuple:
  - source IP address
  - source port number
  - dest IP address
  - dest port number
- Recv host uses all four values to direct segment to appropriate socket
- Server host may support many simultaneous TCP sockets:
  - each socket identified by its own 4-tuple
- Web servers have different sockets for each connecting client
  - non-persistent HTTP will have different socket for each request

Connection-oriented demux (cont)



Connection-oriented demux: Threaded Web Server



Chapter 3 outline

- 3.1 Transport-layer services
- 3.2 Multiplexing and demultiplexing
- 3.3 Connectionless transport: UDP
- 3.4 Principles of reliable data transfer
- 3.5 Connection-oriented transport: TCP
  - segment structure
  - reliable data transfer
  - flow control
  - connection management
- 3.6 Principles of congestion control
- 3.7 TCP congestion control

UDP: User Datagram Protocol [RFC 768]

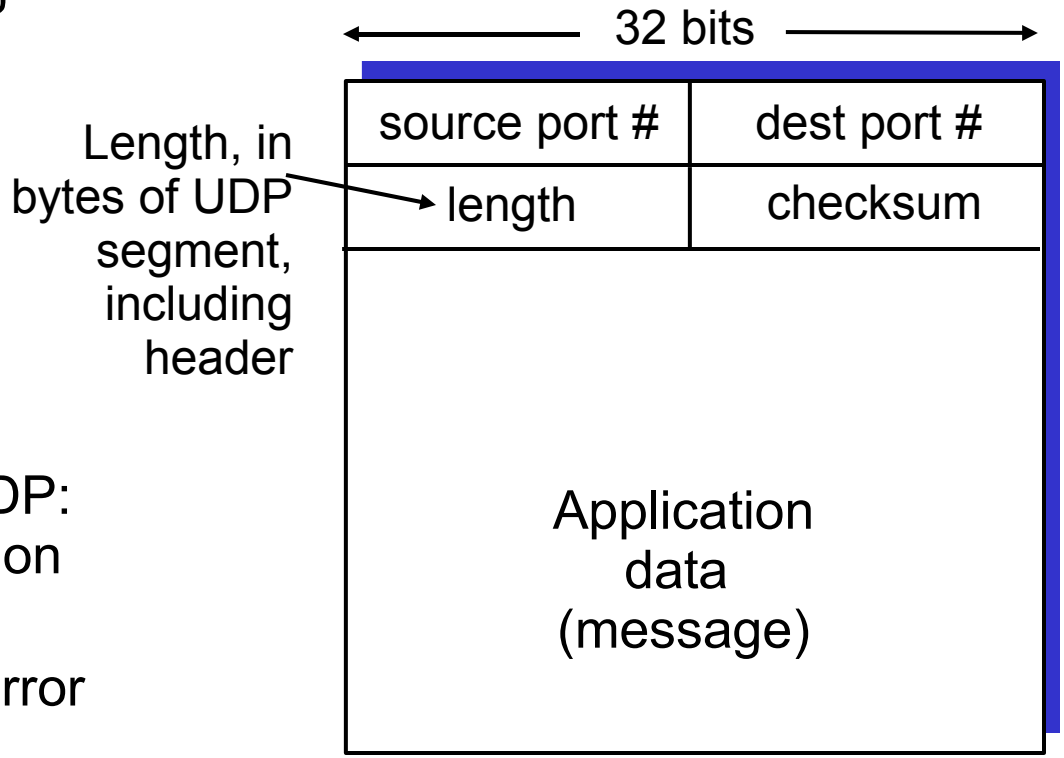
- "No frills," "bare bones" Internet transport protocol
- "Best effort" service; UDP segments may be:
  - lost
  - delivered out of order to app
- *Connectionless:*
  - no handshaking between UDP sender, receiver
  - each UDP segment handled independently of others

Why is there a UDP?

- no connection establishment (which can add delay)
- simple: no connection state at sender, receiver
- small segment header
- no congestion control: UDP can blast away as fast as desired

UDP: more

- Often used for streaming multimedia apps
  - loss tolerant
  - rate sensitive
- Other UDP uses
  - DNS
  - SNMP
- Reliable transfer over UDP: add reliability at application layer
  - application-specific error recovery!



UDP segment format

UDP checksum

Goal: detect “errors” (e.g., flipped bits) in transmitted segment

Sender:

- treat segment contents as sequence of 16-bit integers
- checksum: addition (1’s complement sum) of segment contents
- sender puts checksum value into UDP checksum field

Receiver:

- test checksum of received segment
  - accept or discard segment

Recap

- Mobility in Cellular networks
  - HLR, VLR, MSC
  - Handoff
- Transport Layer
  - Introduction
  - Multiplexing / demultiplexing
  - UDP

Next time

- Reliable Data Transfer
- Midterm review