# CSE398: Network Systems Design

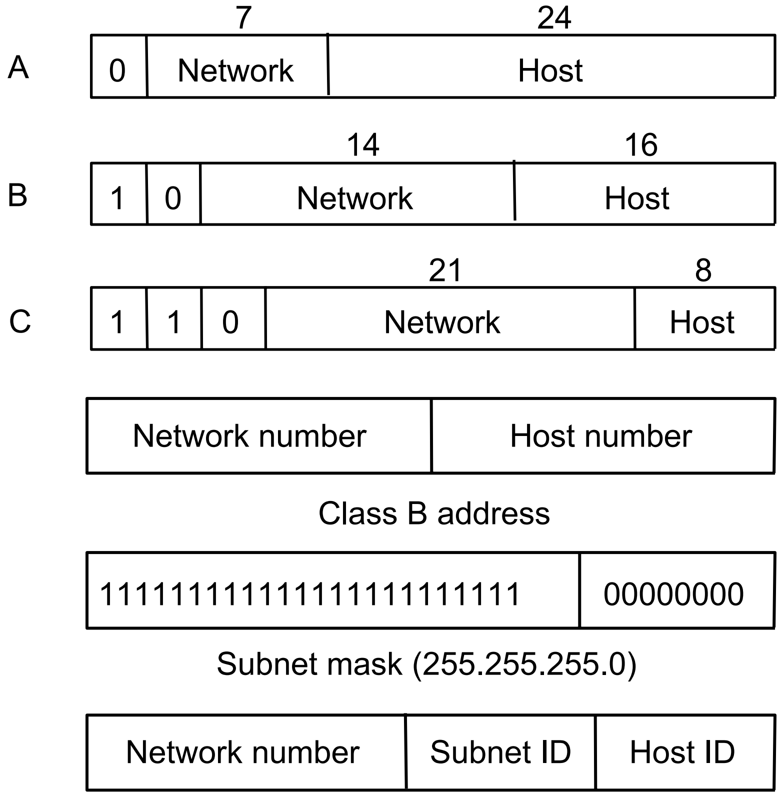
Instructor: Dr. Liang Cheng
Department of Computer Science and Engineering
P.C. Rossin College of Engineering & Applied Science
Assistant Professor, Lehigh University

January 26, 2005

# Outline

- Recap
  - Switching and forwarding
  - IP (Internet Protocol)
- IP
- UDP and TCP
- Summary and homework

# IP Service Model



Class B address

Subnet mask (255.255.255.0)

Subnetted address

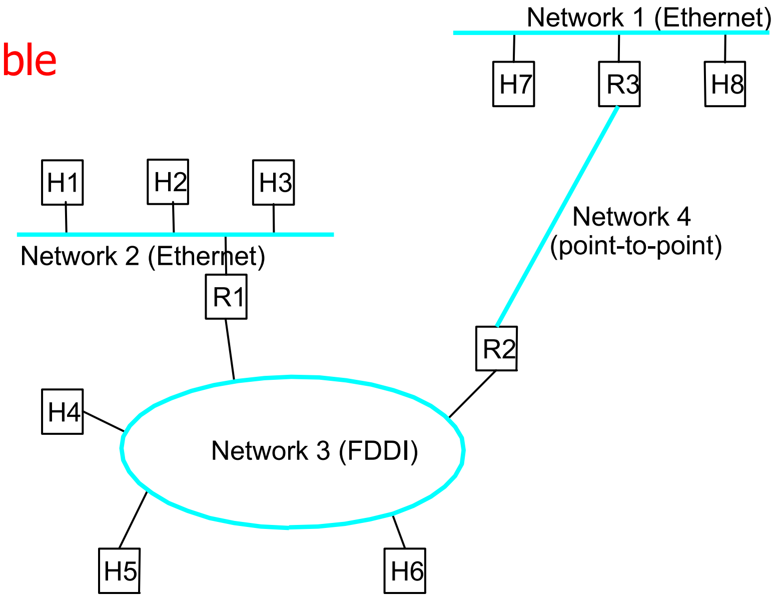
- Connectionless (datagram)
- Best-effort/unreliable service
  - Lost, delivered out of order
  - Duplicated, or delayed
- Addressing
  - Hierarchical: network + host
    - Class A, B, C and D
      - Multicast
  - Dot notation
    - 10.3.2.4, 128.96.34.15,
    - 192.12.69.77, 224.54.93.3
  - Subnetting
    - Add another level to address/routing hierarchy: *subnet*
    - *Subnet masks* define variable partition of host part: e.g., 255.255.255.128

# Packet Routing/Forwarding

- Strategy
  - Every datagram contains destination’s address
  - If directly connected to destination network, then forward to host
  - If not directly connected to destination network, then forward to some router
  - Forwarding table maps network number into next hop
  - Each host has a default router
  - Each router maintains a routing/forwarding table
- Example (R2)

| Network Number (Subnet# & Mask) | Next Hop |
|---|---|
| 1 (128.96.34.0 & 255.255.255.0) | R3 |
| 2 (128.96.34.128 & 255.255.255.128) | R1 |
| 3 (128.96.33.0 & 255.255.255.128) | interface 1 |
| 4 (128.96.33.128 & 255.255.255.128) | interface 0 |

- Classless interdomain routing (CIDR)
  - 128.96.34.0/24
  - 128.96.34.128/25
  - Example: 128.96.34.135
    - Longest match

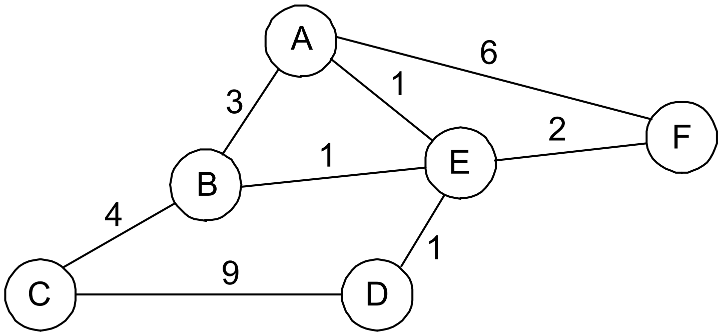


# Address Translation

- Map IP addresses into physical addresses
  - Destination host
  - Next hop router
- ARP (Address Resolution Protocol)
  - Table of IP to physical address bindings
  - Broadcast request if IP address not in table
  - Target machine responds with its physical address
  - Table entries are discarded if not refreshed

# Routing Overview

- Forwarding vs. routing
  - Forwarding: to select an output port based on the destination address and routing table
  - Routing: the process by which the routing table is built
- Network as a graph
  - Graph nodes are routers
  - Default router
  - Graph edges are links
  - Cost: delay, $, …



- Problem: find lowest cost path between two nodes
- Factors
  - Static topology (Wireless)
  - Dynamic load

# Routing Algorithm Classification

- Decentralized information based routing
  - Router knows physically-connected neighbors, link costs to neighbors
  - Iterative process of computation, exchange of info with neighbors
  - “Distance vector” algorithms
- Global information based routing
  - All routers have complete topology, link cost info
  - “Link state” algorithms

# Outline

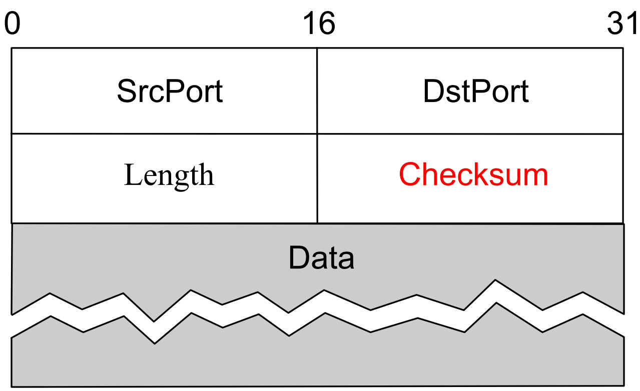
- Recap
- IP
- UDP and TCP (layer 4)
- Summary and homework

# Process-to-Process Channel

- Host-to-host packet delivery service ⇒ process-to-process communication channel
- Underlying best-effort network
  - Drop messages
  - Re-orders messages
  - Delivers duplicate copies of a given message
  - Limits messages to some finite size
  - Delivers messages after an arbitrarily long delay
- Expected end-to-end services
  - Guarantee message delivery
  - Deliver messages in the same order they are sent
  - Deliver at most one copy of each message
  - Support arbitrarily large messages
  - Support synchronization
  - Allow the receiver to flow control the sender
  - Support multiple application processes on each host

# UDP

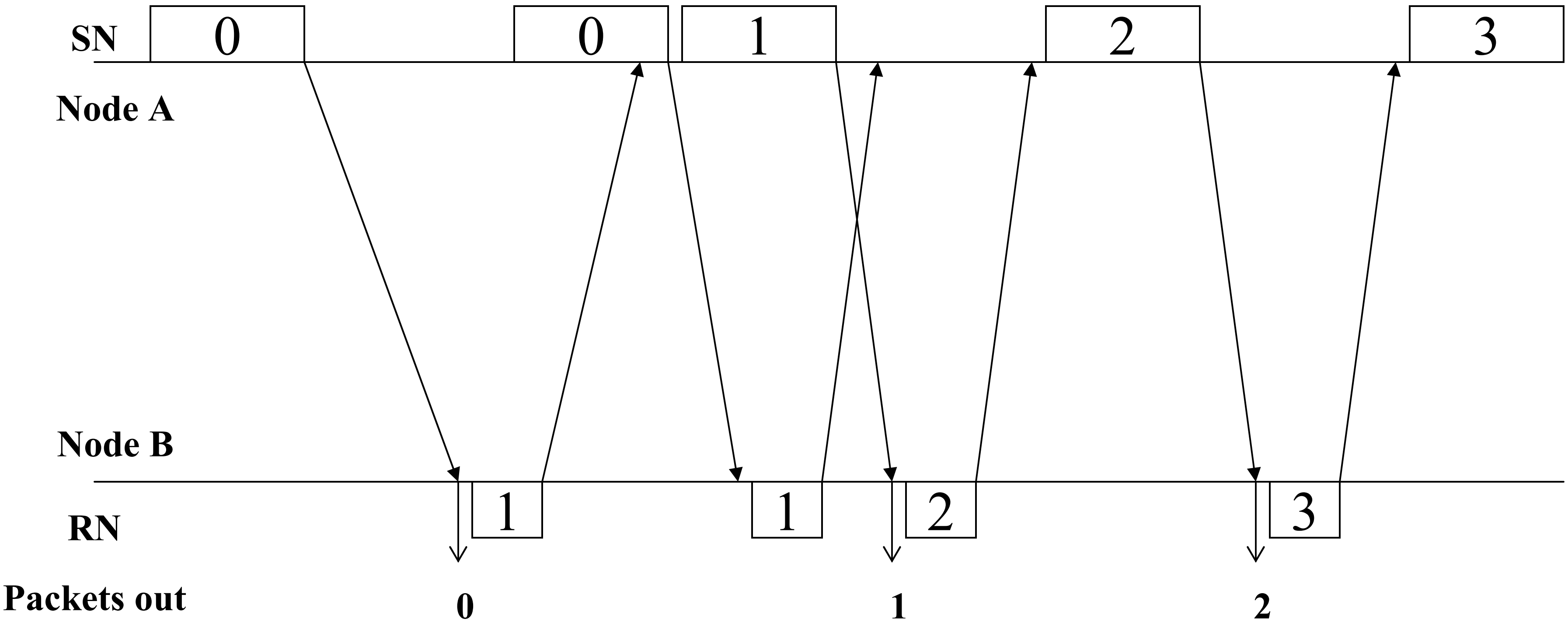
- Unreliable and unordered datagram service
- No flow control
- Header format
- Simple de/multiplexor
- Endpoints identified by ports
  - Servers have *well-known* ports: **`/etc/services`**



# Stop-and-wait Retransmission

- Node A
  - Set $SN$ to 0
  - Accept a packet from the next higher layer, assign number $SN$ to the new packet
  - Transmit the $SN_{th}$ packet in a segment containing $SN$ in the sequence number field; if timeout then retransmit
  - If $RN>SN$, increase $SN$ to $RN$ and go to step 2
- Node B
  - Set $RN$ to 0, then repeat step 2 and step 3
  - An error-free segment from A with $SN=RN$, then release the received segment to the higher layer and increment $RN$
  - At arbitrary time, but within bounded delay after receiving any error-free data segment from A, transmit a segment to A containing $RN$ in the request number field

# Stop-and-wait ARQ Protocol



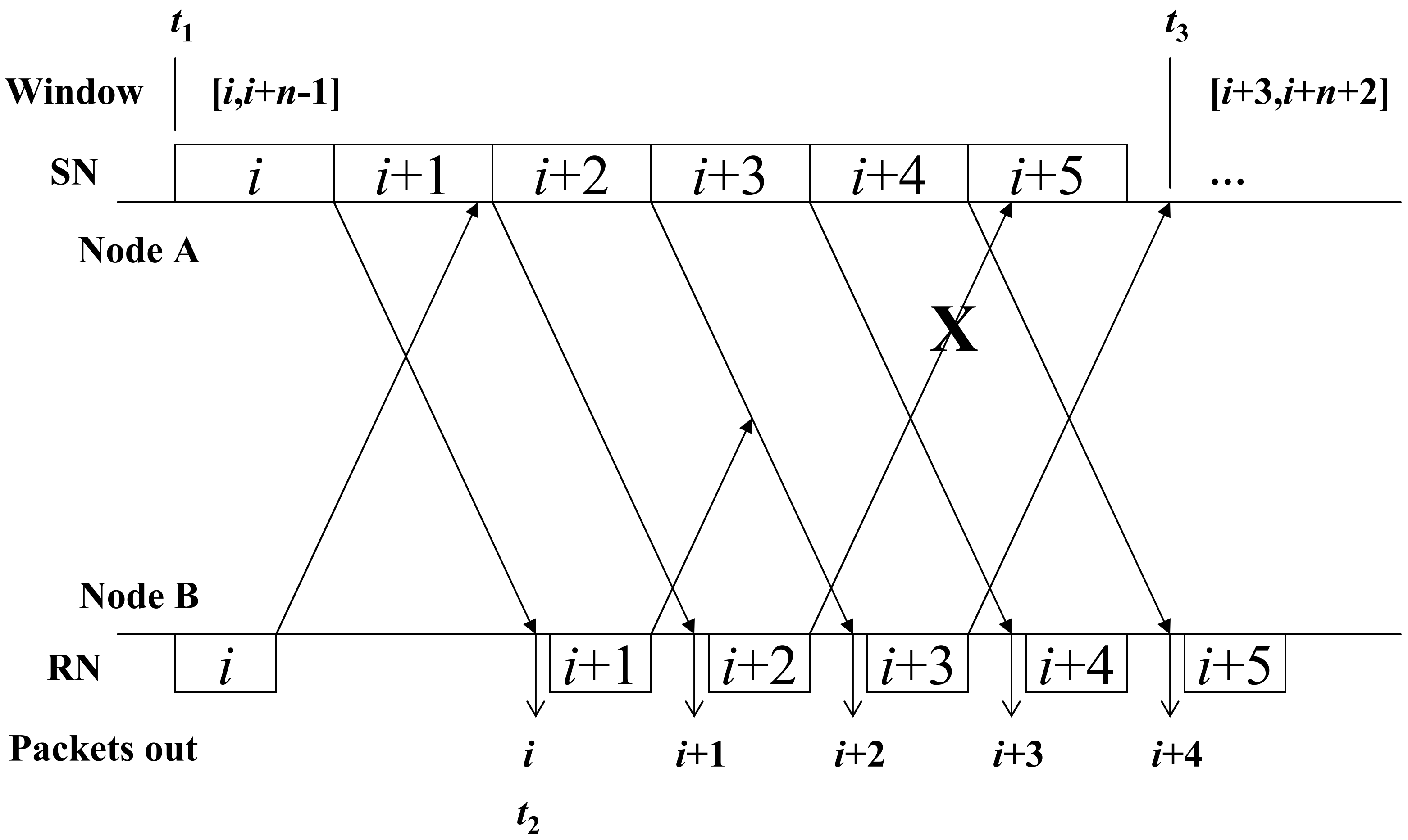
# Go-back-n ARQ Protocol

- Difference from stop-and-wait ARQ
  - Sender: several (*n*) successive packets *can* be sent without waiting for the next packet to be requested
  - Receiver: RN acknowledges all packets prior to RN and requests transmission of packet RN
- Sliding Window ARQ: $SN\max - SN\min \leq n$
  - *SN*min: smallest-numbered packet that has not been acknowledged
  - *SN*max: sequence number to be assigned to the new packet arriving from the higher layer

# Go-back-n ARQ Protocol

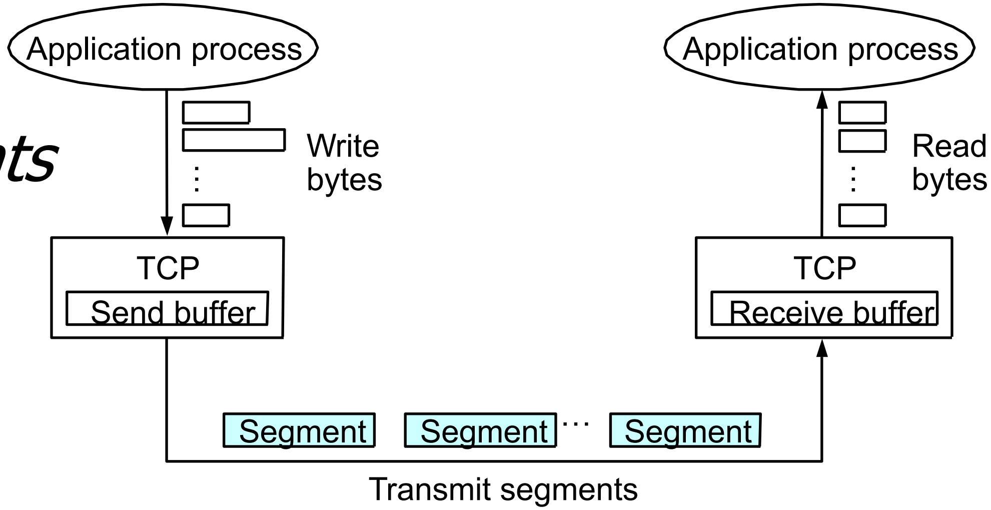
- Node A
  - Set the integer variable $SN$max and $SN$min to 0
  - Do step 3,4,and 5 repeatedly in any order
  - If $SN$max< $SN$min+$n$, and if a packet is available from the higher layer, accept the packet, assign number $SN$max to it, and increment $SN$max
  - If an error-free packet is received from B with $RN$>$SN$min, set $SN$min=$RN$
  - If $SN$min<$SN$max and no packet is in transmission, choose $SN$, $SN$min<$SN$<$SN$max; transmit the $SN$th packet with $SN$. If timeout then retransmit the whole window
- Node B
  - Set the integer $RN$ to 0 and repeat 2 and 3 forever
  - Whenever an error-free packet is received from A with $SN$=$RN$, release the received packet to the higher layer and increment $RN$
  - At arbitrary times, after receiving any error-free packet from A, transmit a packet to A with $RN$

# Go-back-n ARQ Protocol



# TCP Overview

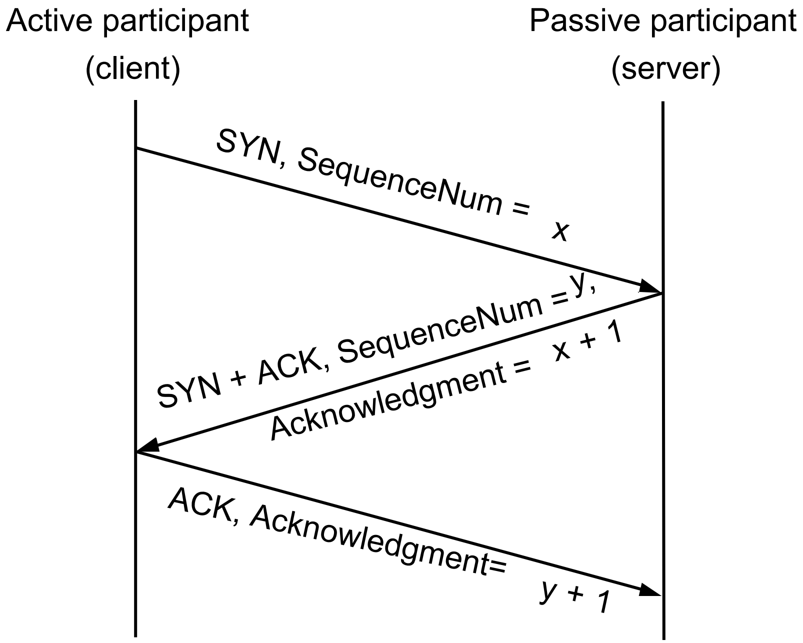
- Connection-oriented
- Byte-stream
  - App writes bytes
  - TCP sends *segments*
  - App reads bytes
- Full duplex



- Flow control
  - Keep sender from overrunning receiver
- Congestion control
  - Keep sender from overrunning network

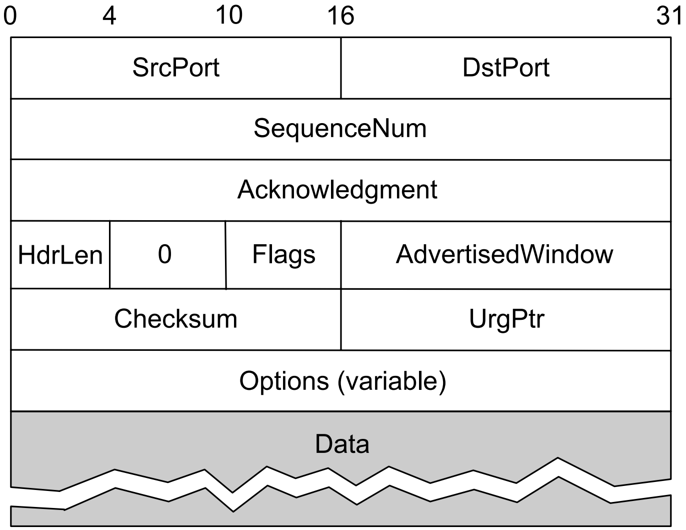
# Establish & Terminate Connection

- Three-way handshake

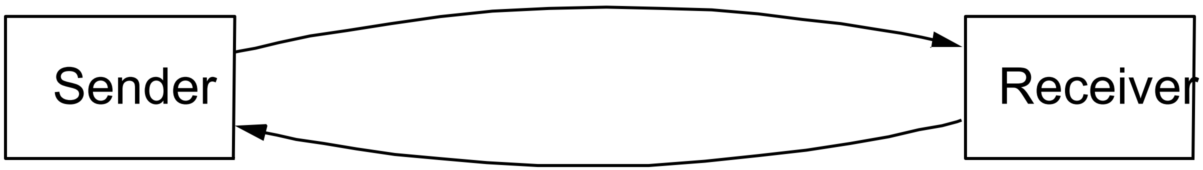


# Flow Control

- TCP segment format
- Connection is identified by 4-tuple:
  - **`(SrcPort, SrcIPAddr, DsrPort, DstIPAddr)`**
- Flow control
  - **`Acknowledgment, SequenceNum, AdvertisedWinow`**
- Flags
  - **`SYN, FIN, RESET, PUSH, URG, ACK`**



Data(SequenceNum)



Acknowledgment +
AdvertisedWindow

# Congestion Control

- AIMD
- Slow start
- Fast retransmit and fast recovery

# Outline

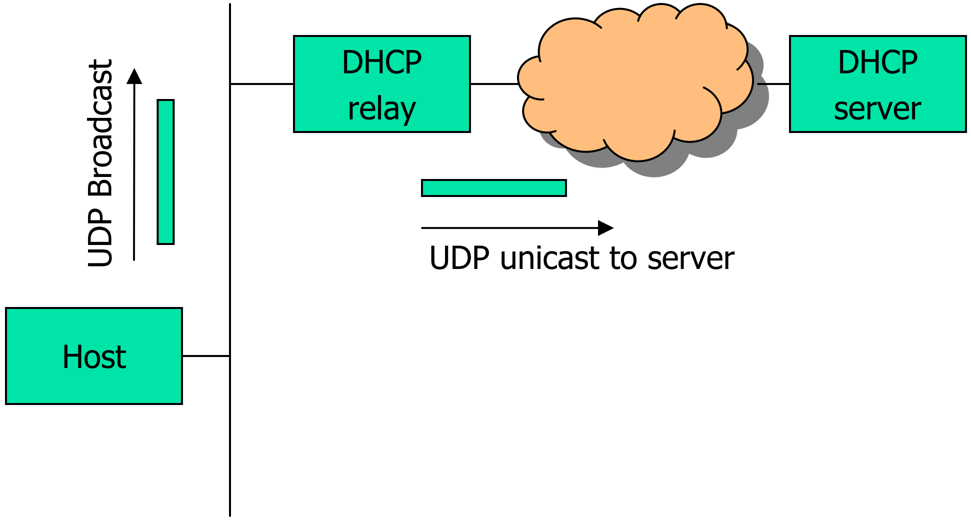
- Recap
- IP
- UDP and TCP
- Summary and homework

# A Review Question

- DHCP
  - UDP or TCP?
  - Unicast, broadcast, and/or multicast?

# DHCP

- DHCP (Dynamic Host Configuration Protocol)
- Save network administrators' configuration efforts: address pool
- Boot: DHCPDISCOVER to 255.255.255.255
- Relay agent
- Lease renew



# Lab on 01/31

- Traffic monitoring and throughput measurement
- Location: PL112

# Homework (due on Jan. 31st before the class)

- 2.2. The following table is a routing table using CIDR. Address bytes are in hexadecimal. The notation "/12" in C4.50.0.0/12 denotes a netmask with 12 leading 1 bits, that is FF.F0.00.00. Note that the last three entries covers every address and thus serve in lieu of a default route. State to what next hop the following will be delivered. (a) C4.5E.13.87 (b) C4.5E.22.09 (c) C3.41.80.02 (d) 5E.43.91.12 (e) C4.6D.31.2E (f) C4.6B.31.2E.

| NetMaskLength | NextHop |
|---|---|
| C4.50.0.0/12 | A |
| C4.5E.10.00/20 | B |
| C4.60.00.00/12 | C |
| C4.68.00.00/14 | D |
| 80.00.00.00/1 | E |
| 40.00.00.00/2 | F |
| 00.00.00.00/2 | G |