CIS 551 / TCOM 401

Computer and Network Security

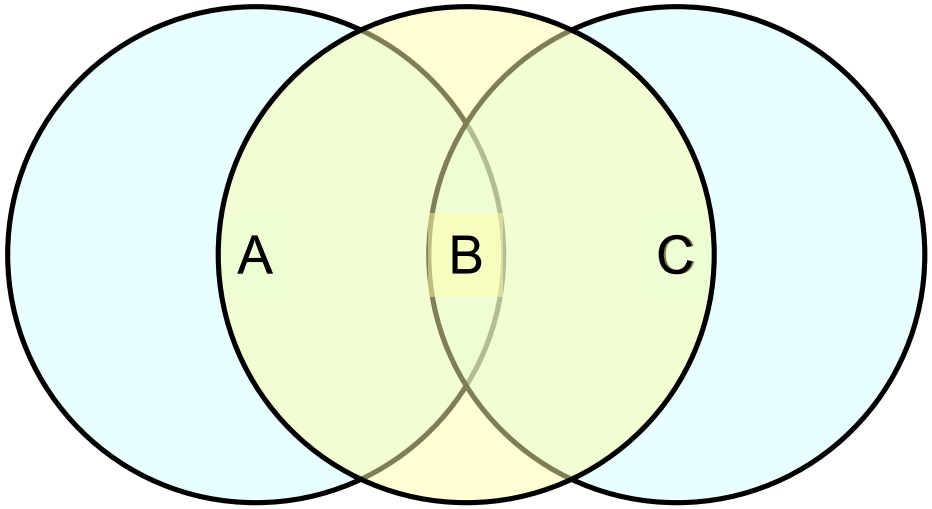
Spring 2008
Lecture 11

Wireless (802.11)

- Spread spectrum radio
  - 2.4GHz frequency band
- Bandwidth ranges 1, 2, 5.5, 11, 22, 54, 248 Mbps
  - 802.11b 11 Mbps
  - 802.11g 54 Mbps
  - 802.11n 248Mbps

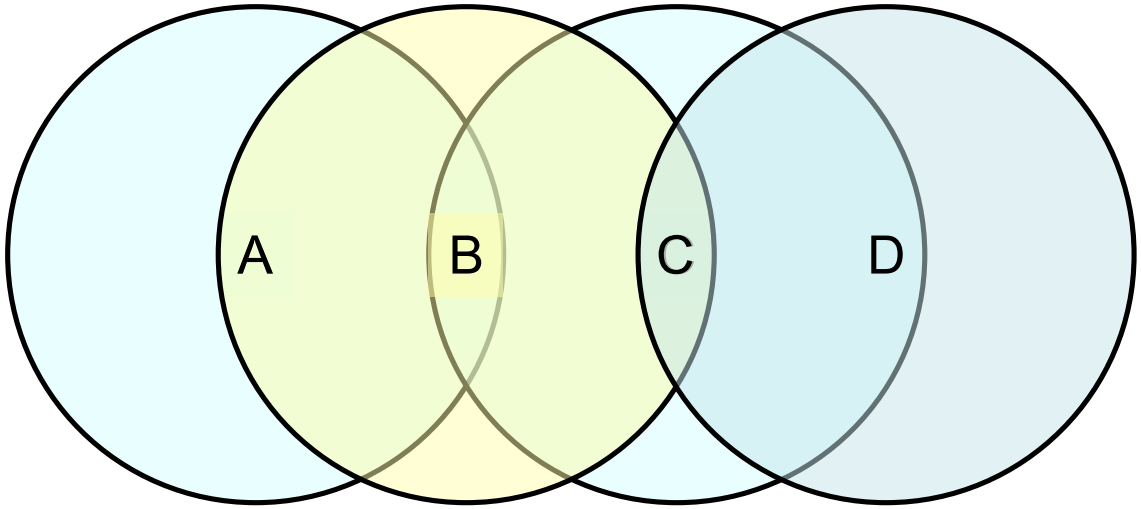
- Like Ethernet, 802.11 has shared medium
  - Need MAC (uses exponential backoff)
- Unlike Ethernet, in 802.11
  - No support for collision detection
  - Not all senders and receivers are directly connected

Hidden nodes



- A and C are *hidden* with respect to each other
  - Frames sent from A to B and C to B simultaneously may collide, but A and C can't detect the collision.

Exposed nodes

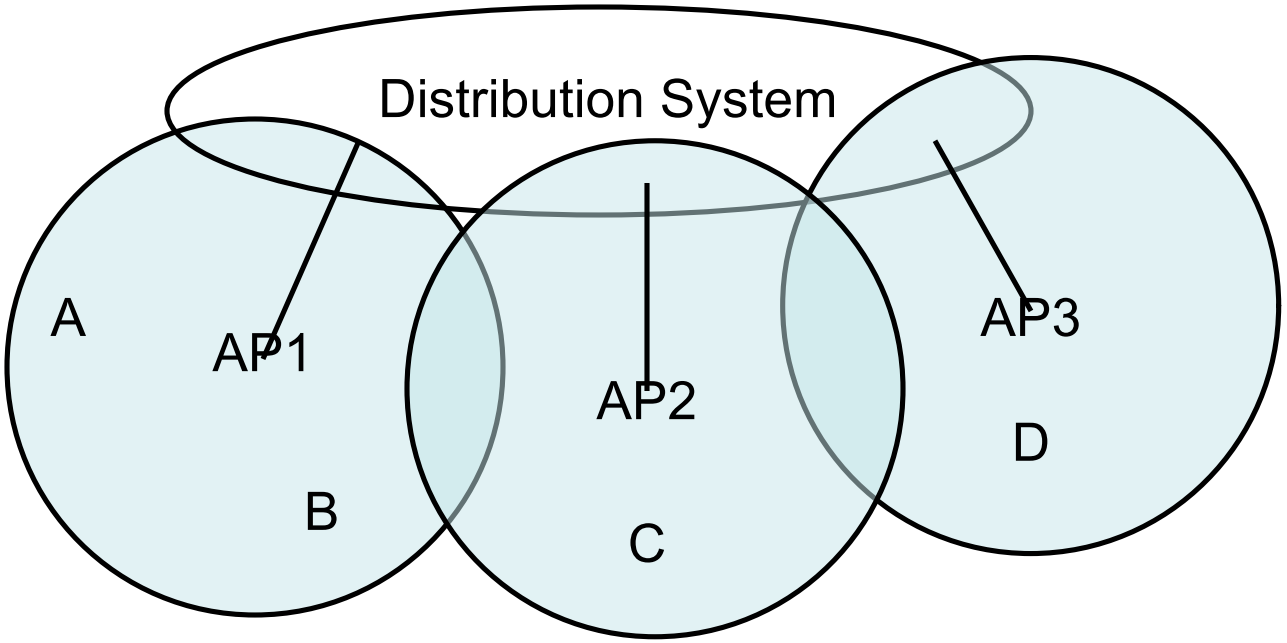


- B is exposed to C
  - Suppose B is sending to A
  - C should still be allowed to transmit to D
  - Even though C—B transmission would collide
  - (Note A to B transmission would cause collision)

Multiple Access Collision Avoidance

- Sender transmits Request To Send (RTS)
  - Includes length of data to be transmitted
  - Timeout leads to exponential backoff (like Ethernet)
- Receiver replies with Clear To Send (CTS)
  - Echoes the length field
- Receiver sends ACK of frame to sender
- Any node that sees CTS cannot transmit for durations specified by length
- Any node that sees RTS but not CTS is not close enough to the receiver to interfere
  - It's free to transmit

Wireless Access Points



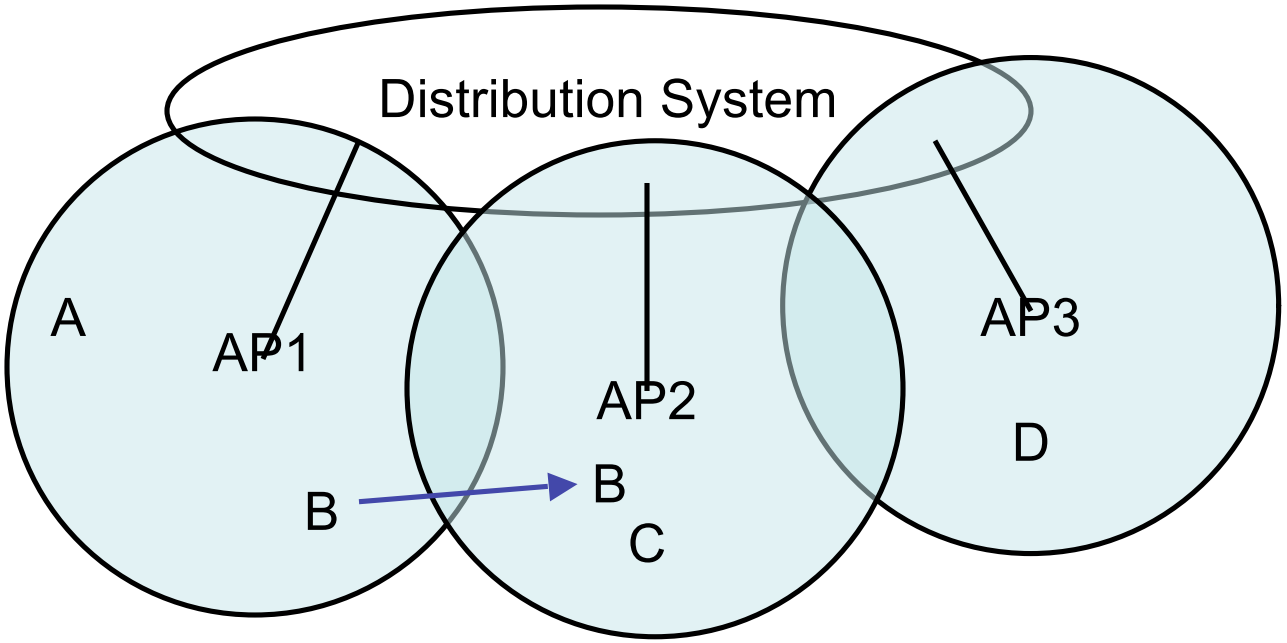
- Distribution System – wired network infrastructure
- Access points – stationary wireless device
- Roaming wireless

Selecting an Access Point

- *Active scanning*
  - Node sends a Probe frame
  - All AP's within reach reply with a Probe Response frame
  - Node selects an AP and sends Association Request frame
  - AP replies with Association Response frame

- *Passive scanning*
  - AP periodically broadcasts Beacon frame
  - Node sends Association Request

Node Mobility



- B moves from AP1 to AP2
- B sends Probes, eventually prefers AP2 to AP1
- Sends Association Request

802.11 Security Issues

- Packet Sniffing is *worse*
  - No physical connection needed
  - Long range:
    - 802.11g indoors: ~38m, outdoors ~140m
    - 802.11n indoors: ~70m, outdoors ~250m
  - Original encryption standards (WEP, WEP2) not that good

- Denial of service
  - Association (and Disassociation) Requests are not authenticated

Wired Equivalent Privacy (WEP)

- Designed to provide same security standards as wired LANs (like Ethernet)
  - WEP uses 40 bit keys
  - WEP2 uses 128 bit keys
- Uses shared key authentication
  - Key is configured manually at the access point
  - Key is configured manually at the wireless device

- WEP frame transmission format:

$$802.11Hdr,\ IV,\ K_{S+IV}\{DATA,\ ICV\}$$

  - S = shared key
  - IV = 24 bit "initialization vector"
  - ICV = "integrity checksum" uses the CRC checksum algorithm
  - Encryption algorithm is RC4

Problem with WEP

- RC4 generates a keystream
  - Shared key S plus IV generates a long sequence of pseudorandom bytes RC4(IV,S)
  - Encryption is: C = P ⊕ RC4(IV,S) ⊕ = "xor"

- IV's are public -- so it's easy to detect their reuse
- Problem: if IV ever repeats, then we have
  - C1 = P1 ⊕ RC4(IV,S)
  - C2 = P2 ⊕ RC4(IV,S)
  - So C1 ⊕ C2 = P1 ⊕ P2
  - Statistical analysis or known plaintext can disentangle P1 and P2

Finding IV Collisions

- How IV is picked is not specified in the standard:
  - Standard "recommends" (but does not require) that IV be changed for every packet
  - Some vendors initialize to 0 on reset and then increment
  - Some vendors generate IV randomly per packet
- Very active links send ~1000 packets/sec
  - Exhaust 24 bit keyspace in < 1/2 day
- If IV is chosen randomly, probability is > 50% that there will be a collision after only 4823 packets

Other WEP problems

- Replay attacks
  - Standard requires the protocol to be stateless
  - Expensive to rule out replay attacks. (The sender and receiver can't keep track of expected sequence numbers)

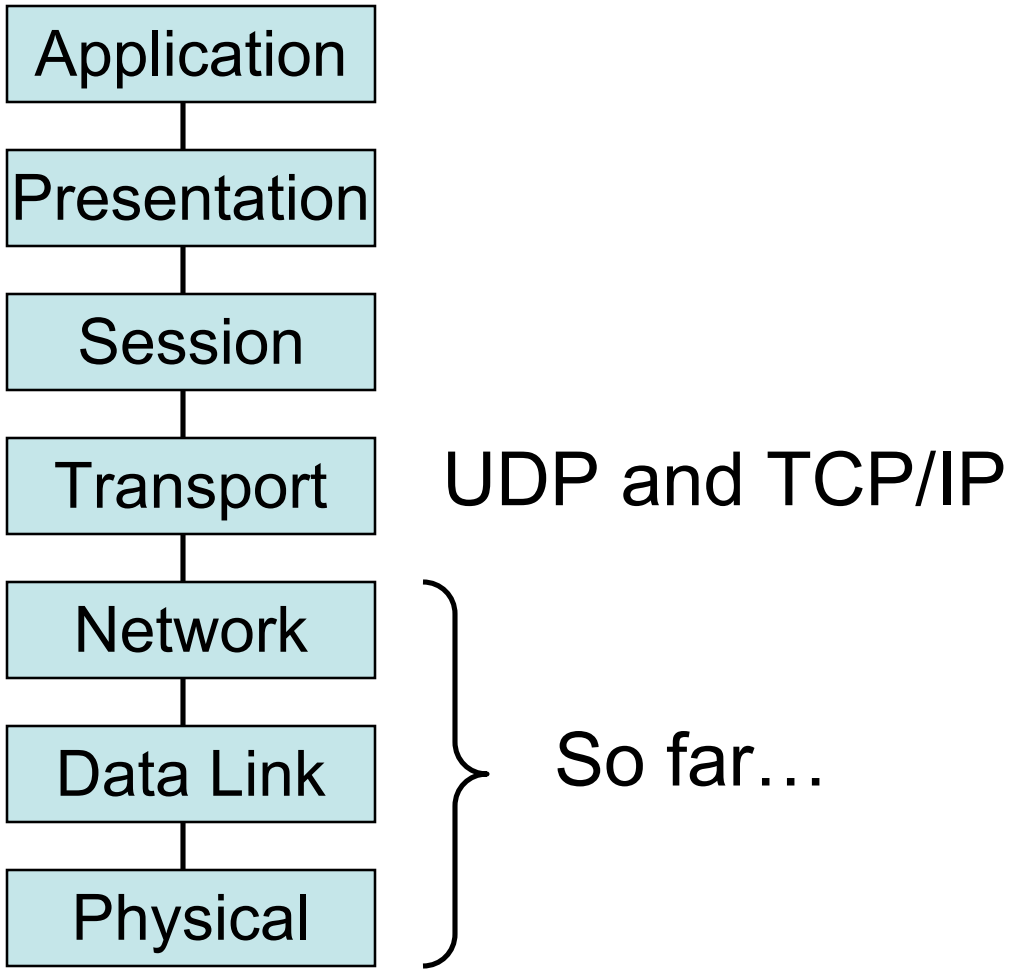
- Integrity violations
  - Attacker can inject or corrupt WEP encrypted packets
  - CRC (Cyclic Redundancy Check) is an error detection code commonly used in internet protocols
  - CRC is good at detecting random errors (introduced by environmental noise)
  - But, CRC is not a hash function -- it is easy to find collisions
  - Attacker can arbitrarily pass off bogus WEP packets as legitimate ones

Wi-Fi Protected Access (WPA)

- Appeared in 2003, standardized as WPA2 in 2004
- Uses AES
  - Advanced Encryption Standard ("government strength")
  - Pre-shared keys (home use) or radius server (enterprise)

- Much better than WEP/WEP2
- More about the crypto soon…

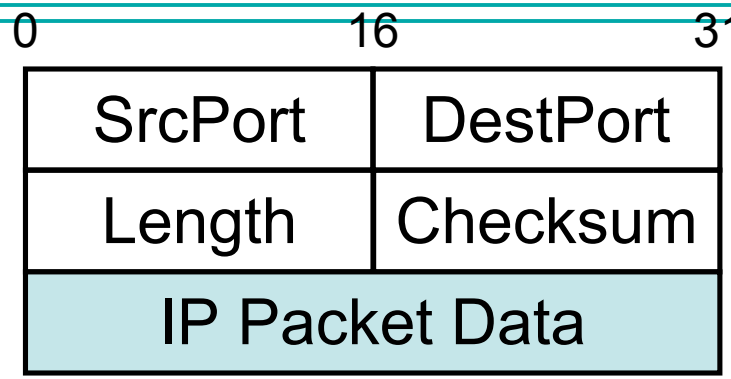
Protocol Stack Revisited



Application vs. Network

Application Needs	Network Char.
Reliable, Ordered, Single-Copy Message Delivery	Drops , Duplicates and Reorders Messages
Arbitrarily large message s	Finite message size
Flow Control by Receiver	Arbitrary Delay
Supports multiple applications per-host	…

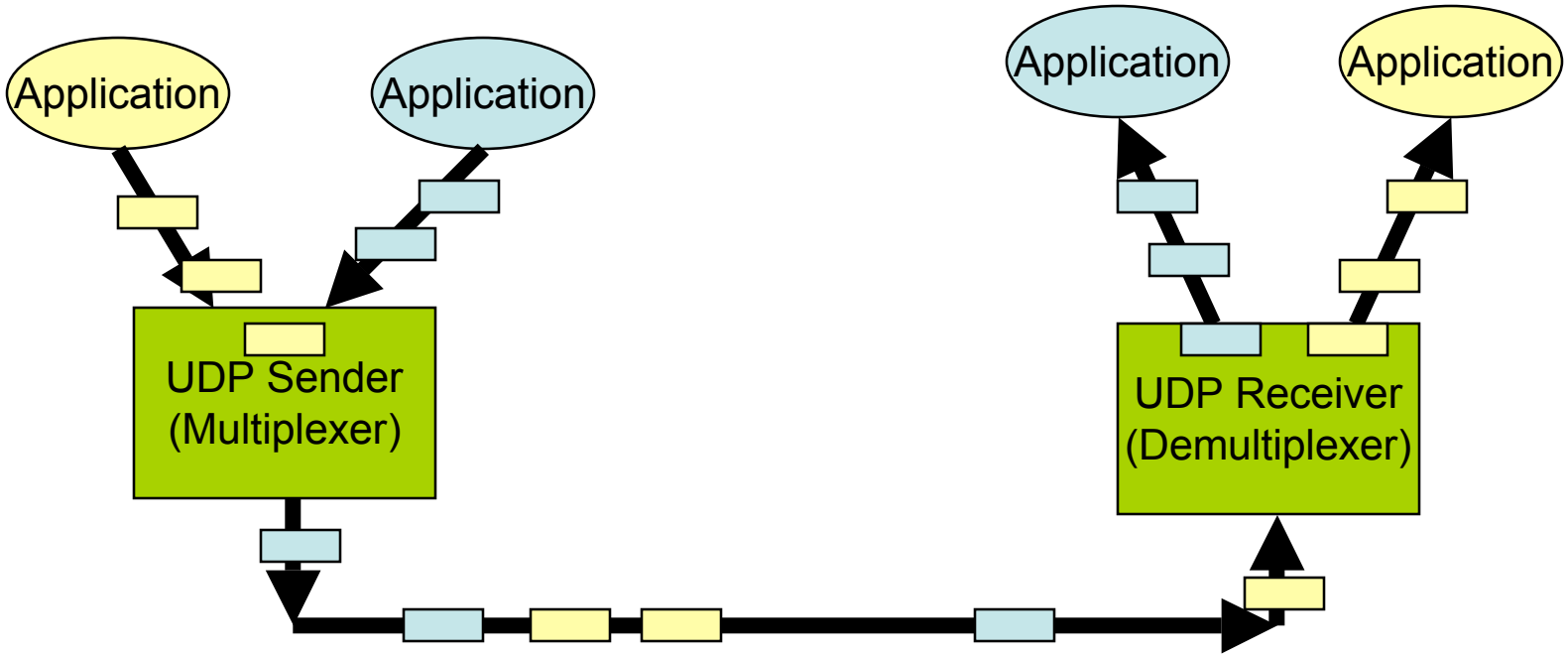
User Datagram Protocol (UDP)



- Simplest transport-layer protocol
- Just exposes IP packet functionality to application level
- *Ports* identify sending/receiving process
  - Demultiplexing information
  - (port, host) pair identifies a network process

UDP End-to-End Model

- Multiplexing/Demultiplexing with Port number



Using Ports

- Client contacts Server at a *well-known port*
  - SMPT: port 25
  - DNS: port 53
  - POP3: port 110
  - Unix talk : port 517
  - In unix, ports are listed in /etc/services
- Sometimes Client and Server agree on a different port for subsequent communication

- Ports are an abstraction
  - Implemented differently on different OS's
  - Typically a message queue

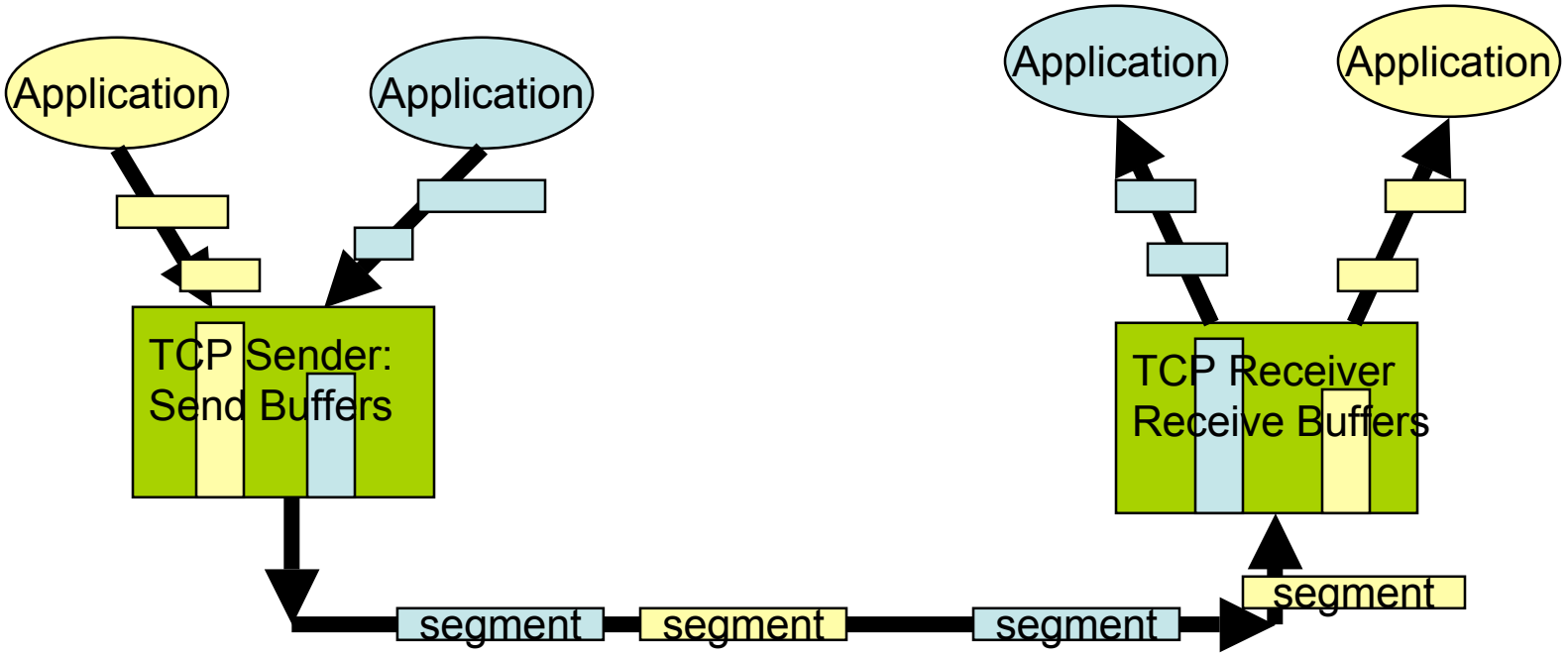
Transmission Control Protocol (TCP)

- Most widely used protocol for reliable byte streams
  - Reliable, in-order delivery of a stream of bytes
  - Full duplex: pair of streams, one in each direction
  - Flow and congestion control mechanisms
  - Like UDP, supports ports

- Built on top of IP (hence TCP/IP)

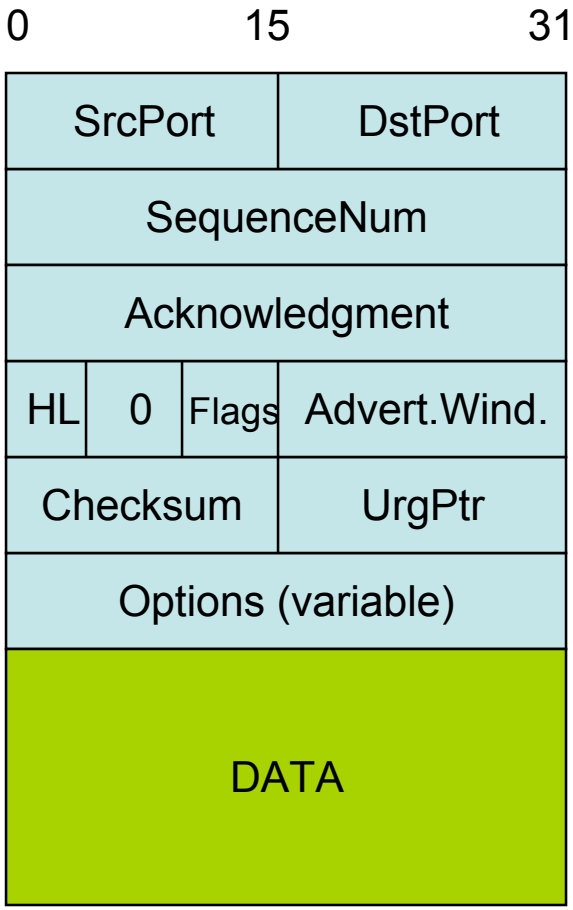
TCP End-to-End Model

- Buffering corrects errors but may introduce delays

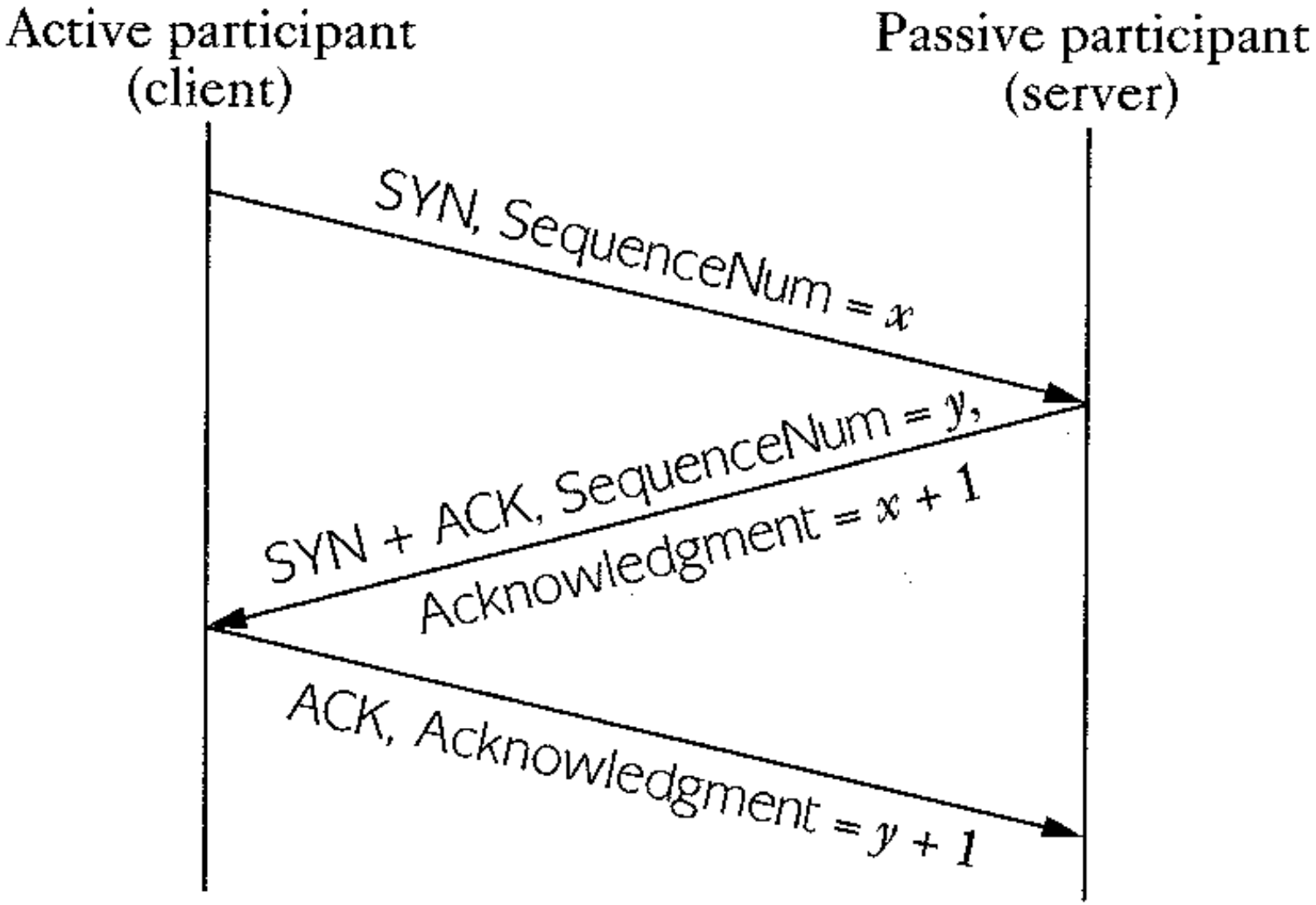


Packet Format

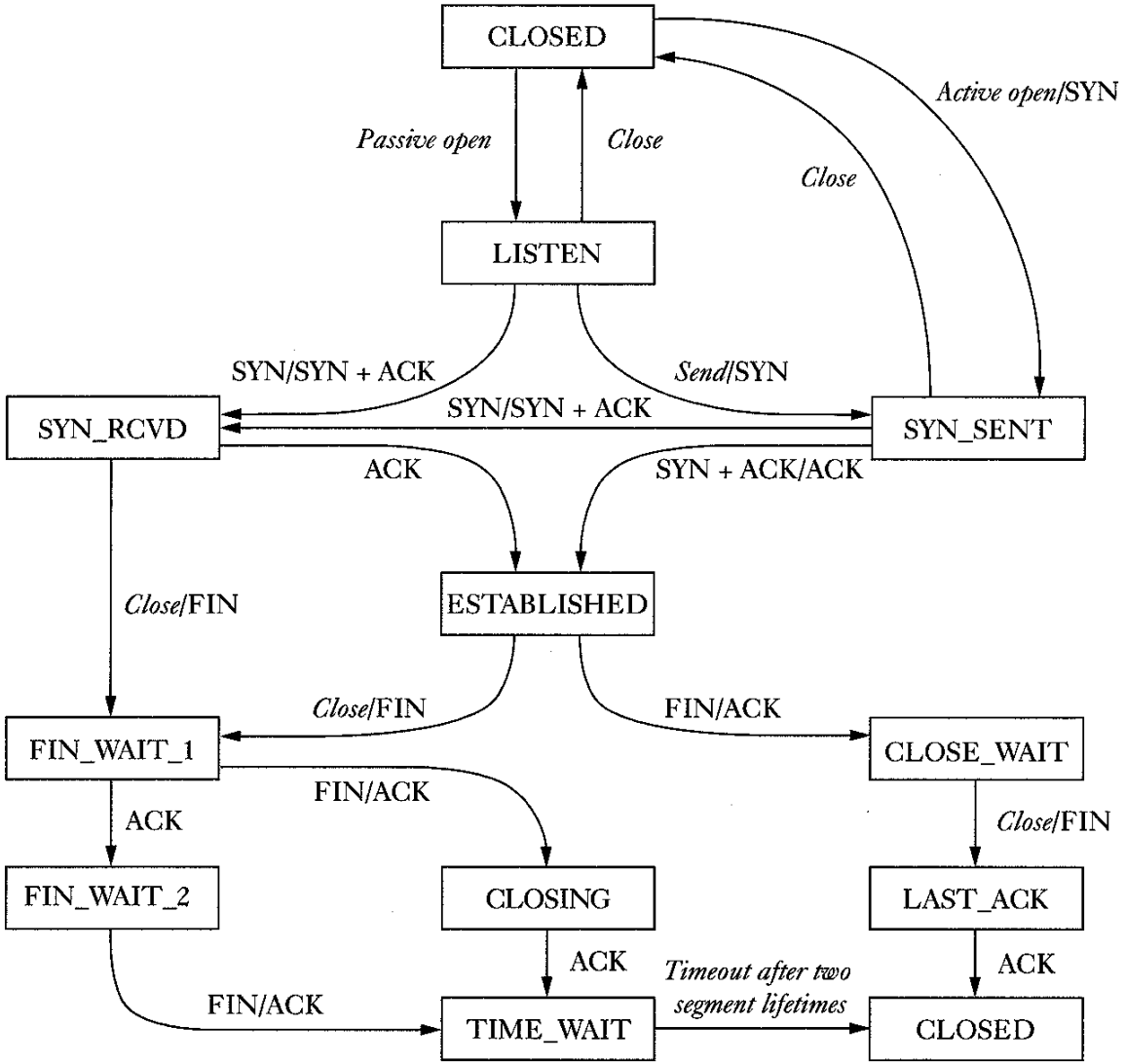
- Flags
  - SYN
  - FIN
  - RESET
  - PUSH
  - URG
  - ACK
- Fields



Three-Way Handshake



TCP State Transitions



TCP Receiver

- Maintains a buffer from which application reads
- Advertises < buffer size as the window for sliding window
- Responds with Acknowledge and AdvertisedWindow on each send; updates byte counts when data O.K.
- Application blocked until read() O.K.

TCP Sender

- Maintains a buffer; sending application is blocked until room in the buffer for its write
- Holds data until acknowledged by receiver *as successfully received*
- Implement window expansion and contraction; note difference between *flow* and *congestion* control

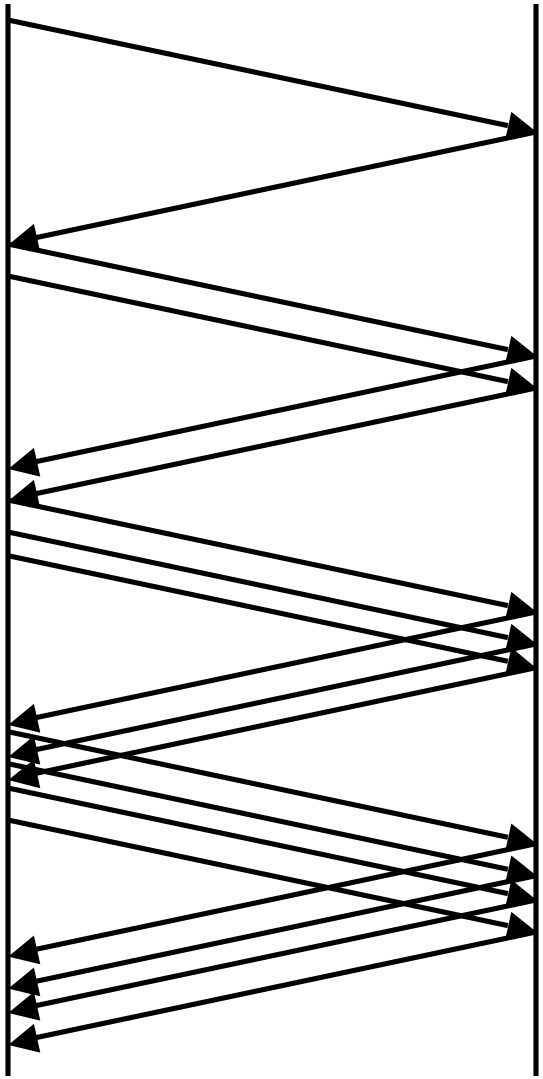
TCP Flow & Congestion Control

- Flow vs. Congestion Control
  - Flow control protects the recipient from being overwhelmed.
  - Congestion control protects the network from being overwhelmed.
- TCP Congestion Control
  - Additive Increase / Multiplicative Decrease
  - Slow Start
  - Fast Retransmit and Fast Recovery

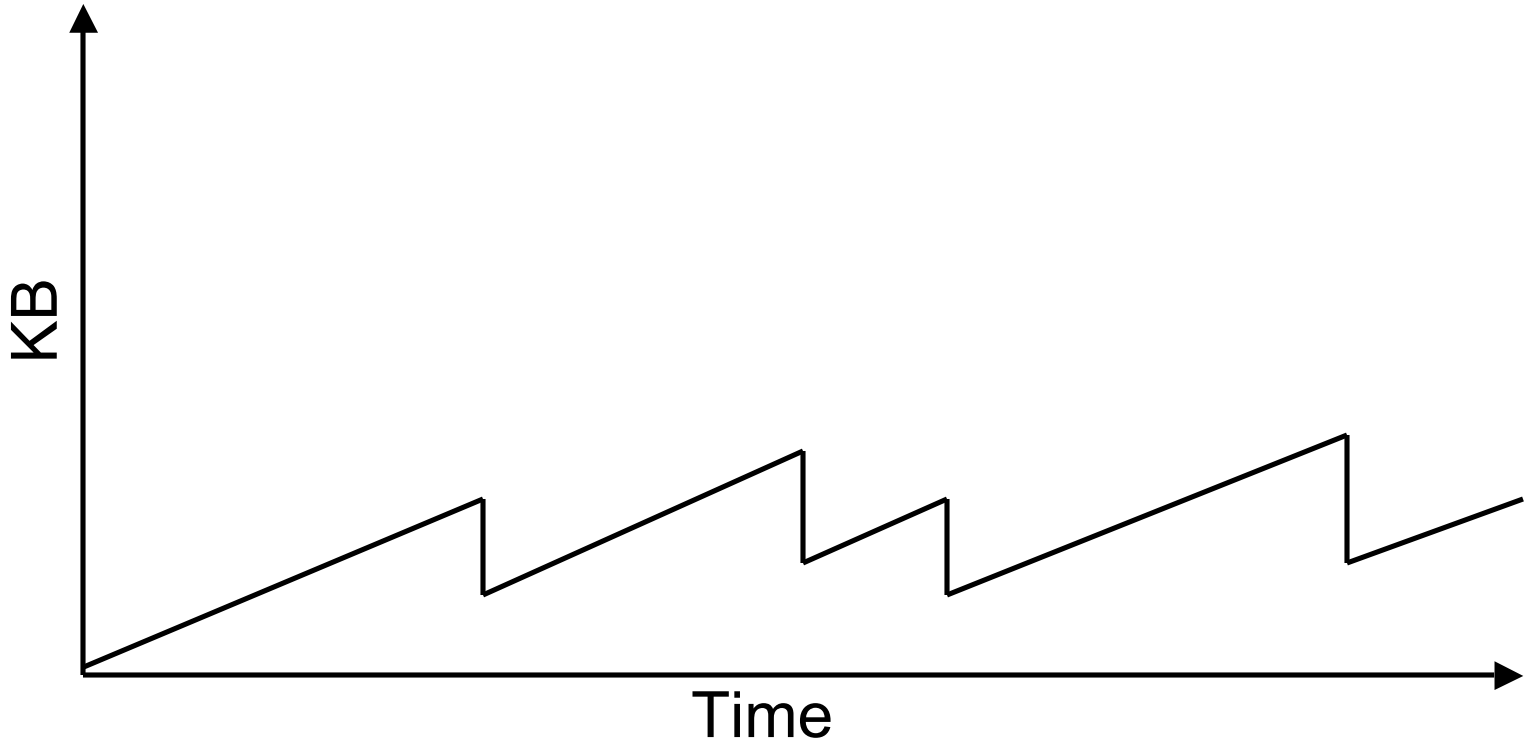
Increase and Decrease

- A value CongestionWindow is used to control the number of unacknowledged transmissions.
- This value is increased linearly until timeouts for ACKs are missed.
- When timeouts occur, CongestionWindow is decreased by half to reduce the pressure on the network quickly.
- The strategy is called “additive increase / multiplicative decrease”.

Additive Increase



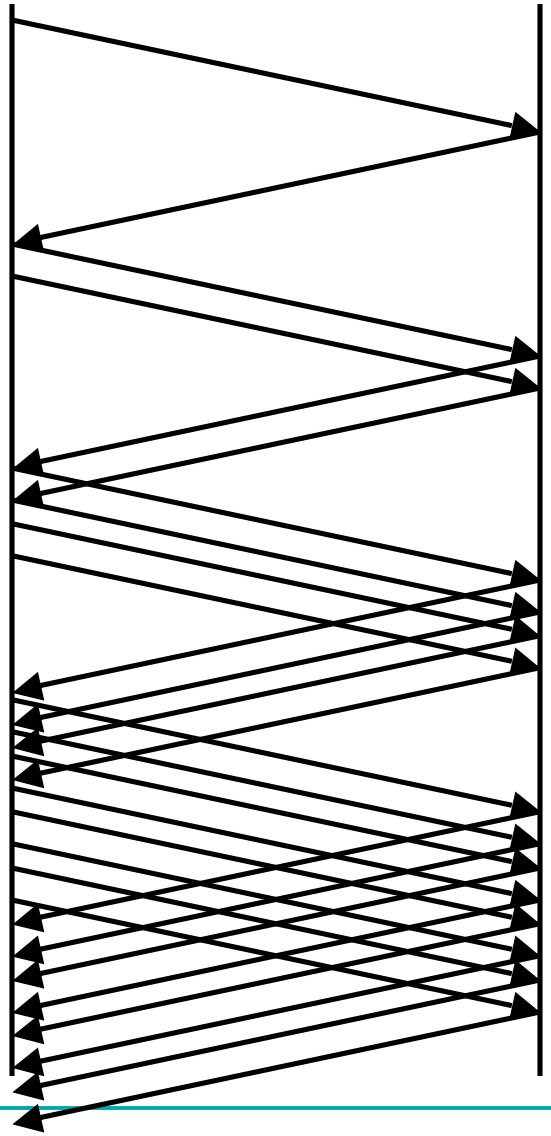
TCP Sawtooth Pattern



Slow Start

- Sending the entire window immediately could cause a traffic jam in the network.
- Begin “slowly” by setting the congestion window to one packet.
- When acknowledgements arrive, double the congestion window.
- Continue until ACKs do not arrive or flow control dominates.

Slow Start



Network Vulnerabilities

- Anonymity
  - Attacker is remote, origin can be disguised
  - Authentication
- Many points of attack
  - Attacker only needs to find weakest link
  - Attacker can mount attacks from many machines
- Sharing
  - Many, many users sharing resources
- Complexity
  - Distributed systems are large and heterogeneous
- Unknown perimeter
- Unknown attack paths

Syn Flood Attack

- Recall TCP's 3-way handshake:
  - SYN --- SYN+ACK --- ACK
- Receiver must maintain a queue of partially open TCP connections
  - Called SYN_RECV connections
  - Finite resource (often small: e.g. 20 entries)
  - Timeouts for queue entries are about 1 minute.

- Attacker
  - Floods a machine with SYN requests
  - Never ACKs them
  - Spoofs the sending address (Why? Two reasons!)

Reflected denial of service

- Broadcast a ping request
  - For sender's address put target's address
  - All hosts reply to ping, flooding the target with responses

- Hard to trace
- Hard to prevent
  - Turn off ping? (Makes legitimate use impossible)
  - Limit with network configuration by restricting scope of broadcast messages

(Distributed) Denial of Service

- Coordinate multiple subverted machines to attack
- Flood a server with bogus requests
  - TCP SYN packet flood
  - > 600,000 packets per second
- Detection & Assessment?
  - 12,800 attacks at 5000 hosts! (in 3 week period during 2001)
  - IP Spoofing (forged source IP address)
  - http://www.cs.ucsd.edu/users/savage/papers/UsenixSec01.pdf
- Prevention?
  - Filtering?
  - Decentralized file storage?