CIS 551 / TCOM 401

Computer and Network Security

Spring 2009
Lecture 8

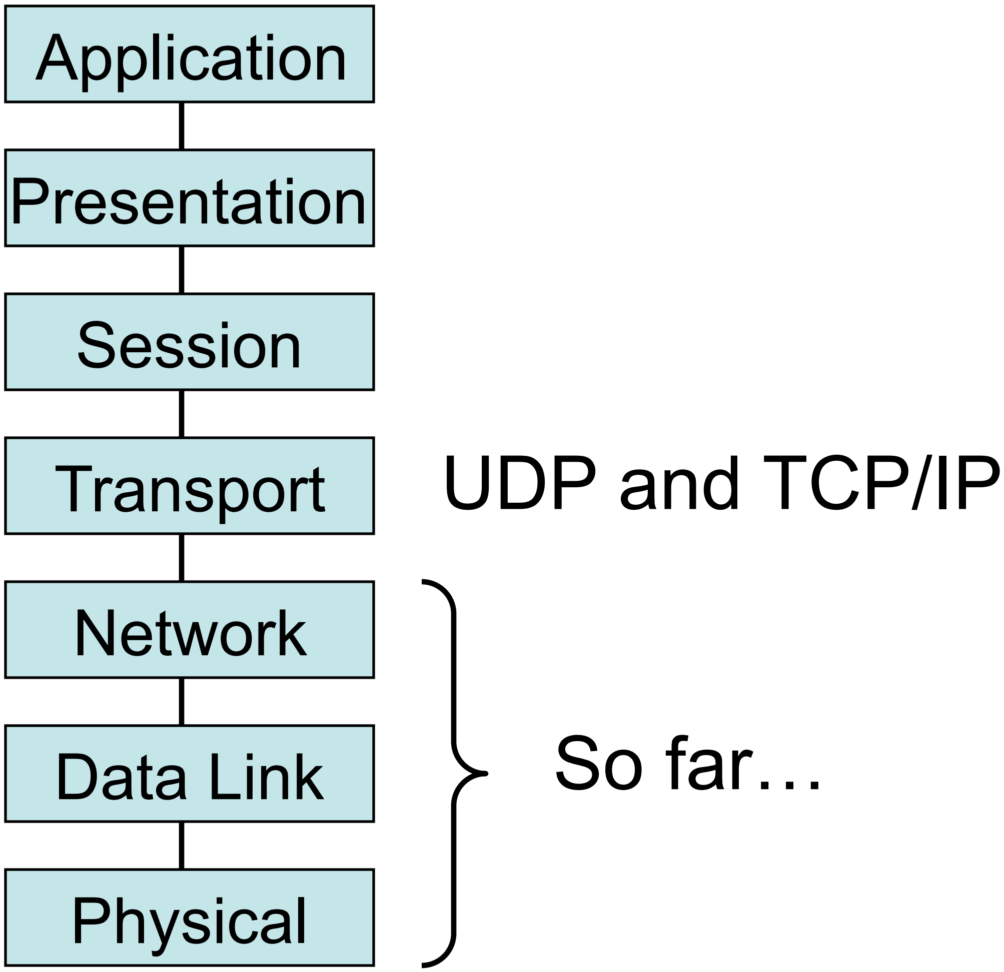
Announcements

- Plan for Today:
  - Networks: TCP
  - Firewalls

- Midterm 1: One week from Today!
  - 2/17/2009
  - In class, short answer, multiple choice, analysis

- Project 2 will be available 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	…

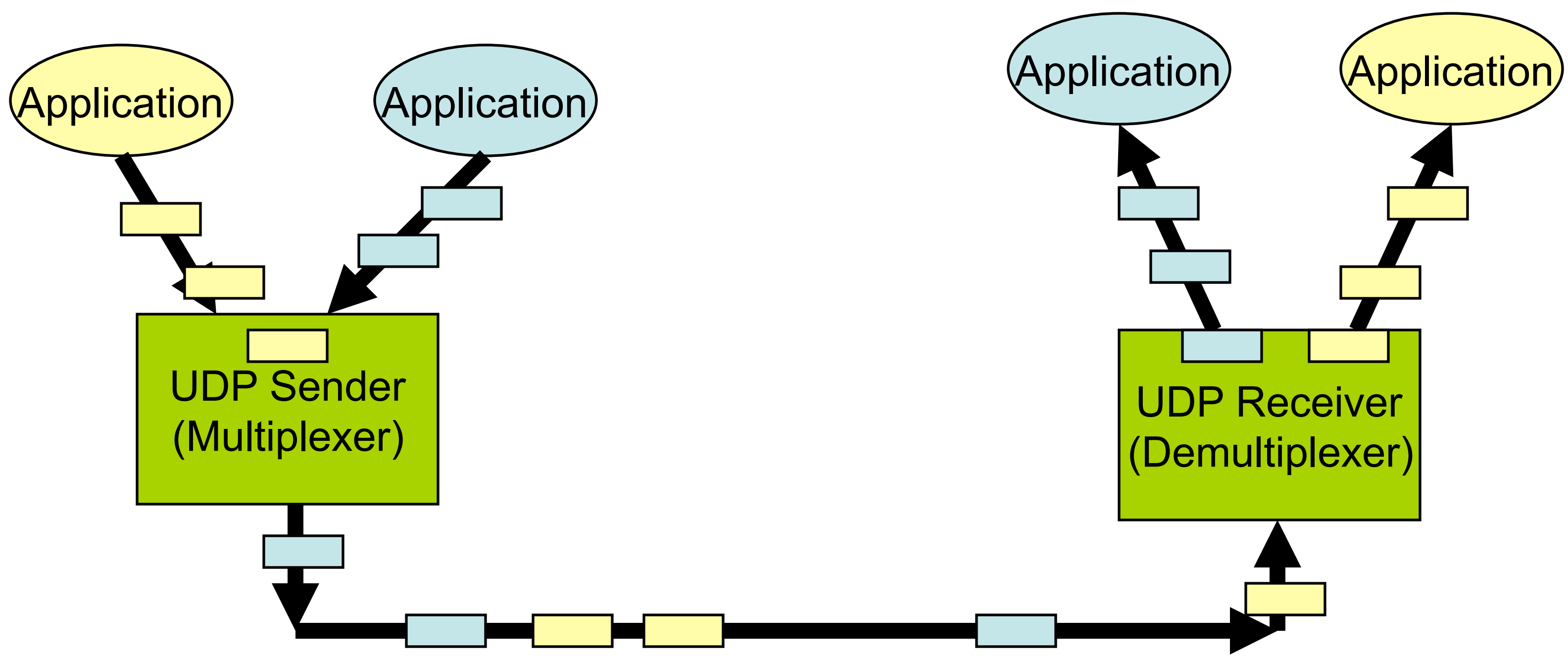
User Datagram Protocol (UDP)

0	16 31
SrcPort	DestPort
Length	Checksum
IP Packet Data	

- 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

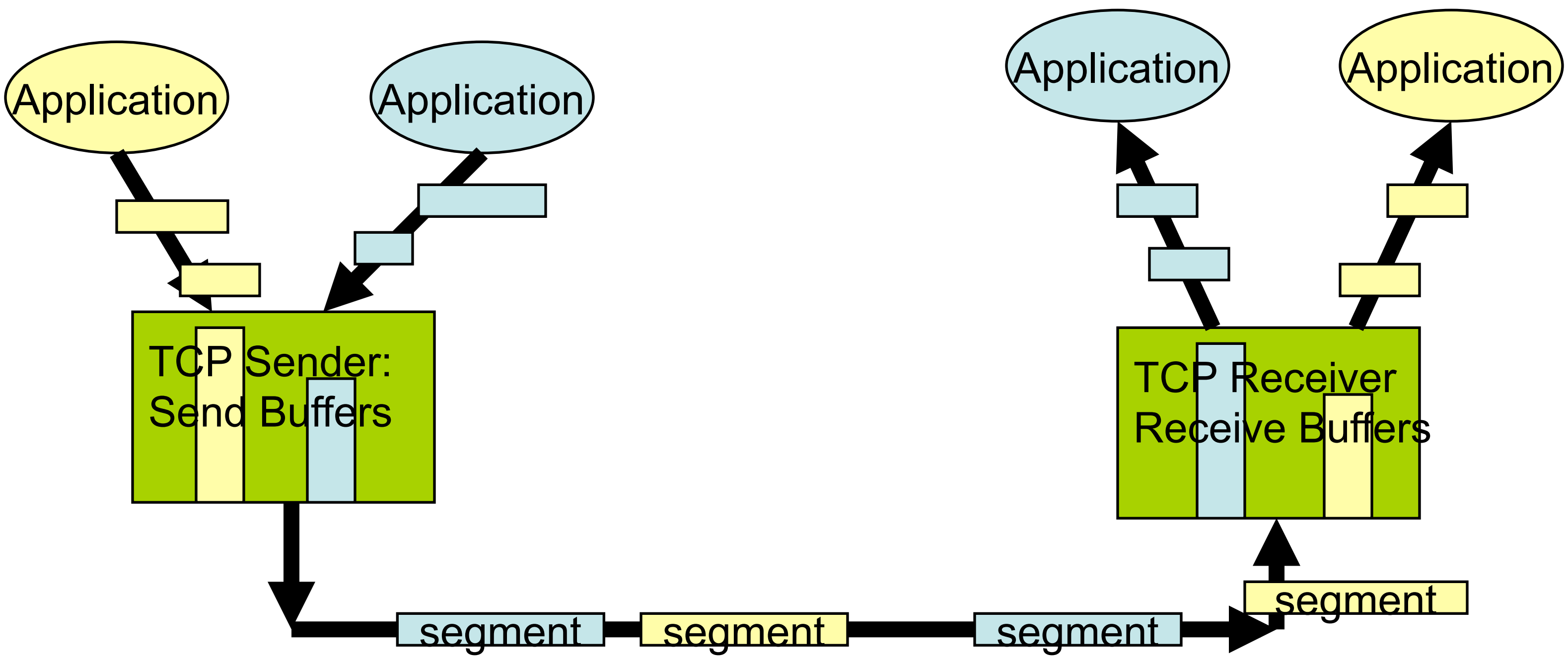
Transmission Control Protocol (TCP)

- 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)

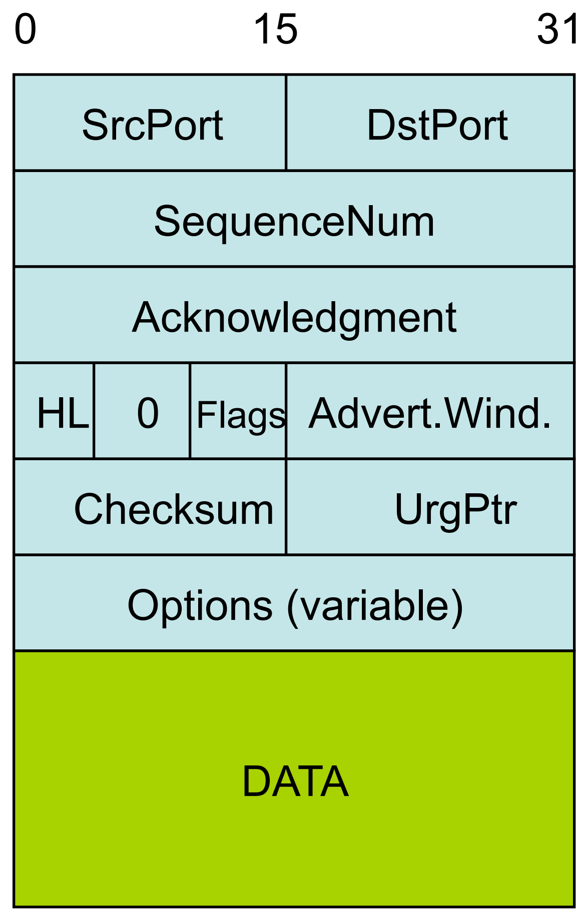
TCP End-to-End Model

- Buffering corrects errors but may introduce delays

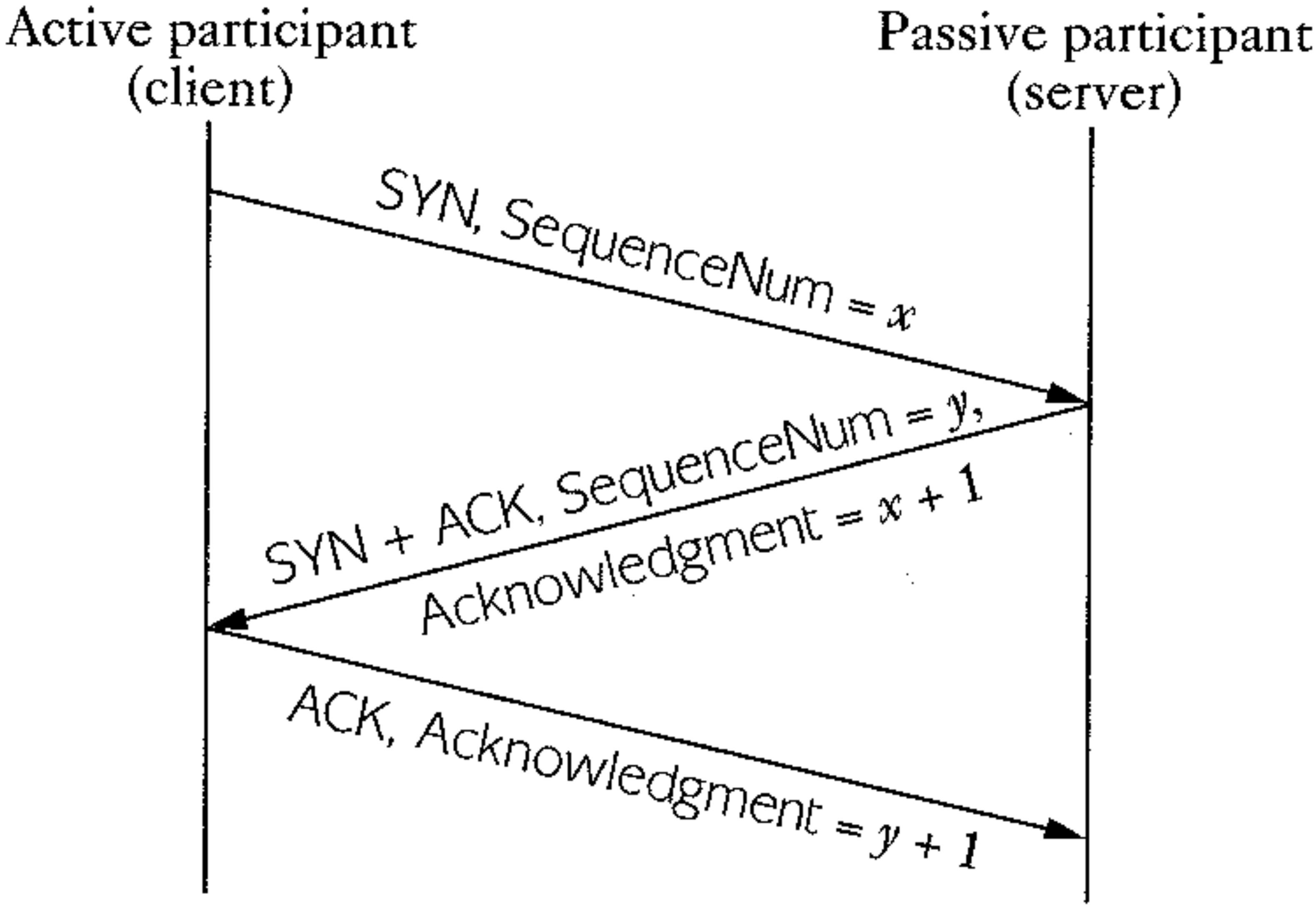


Packet Format

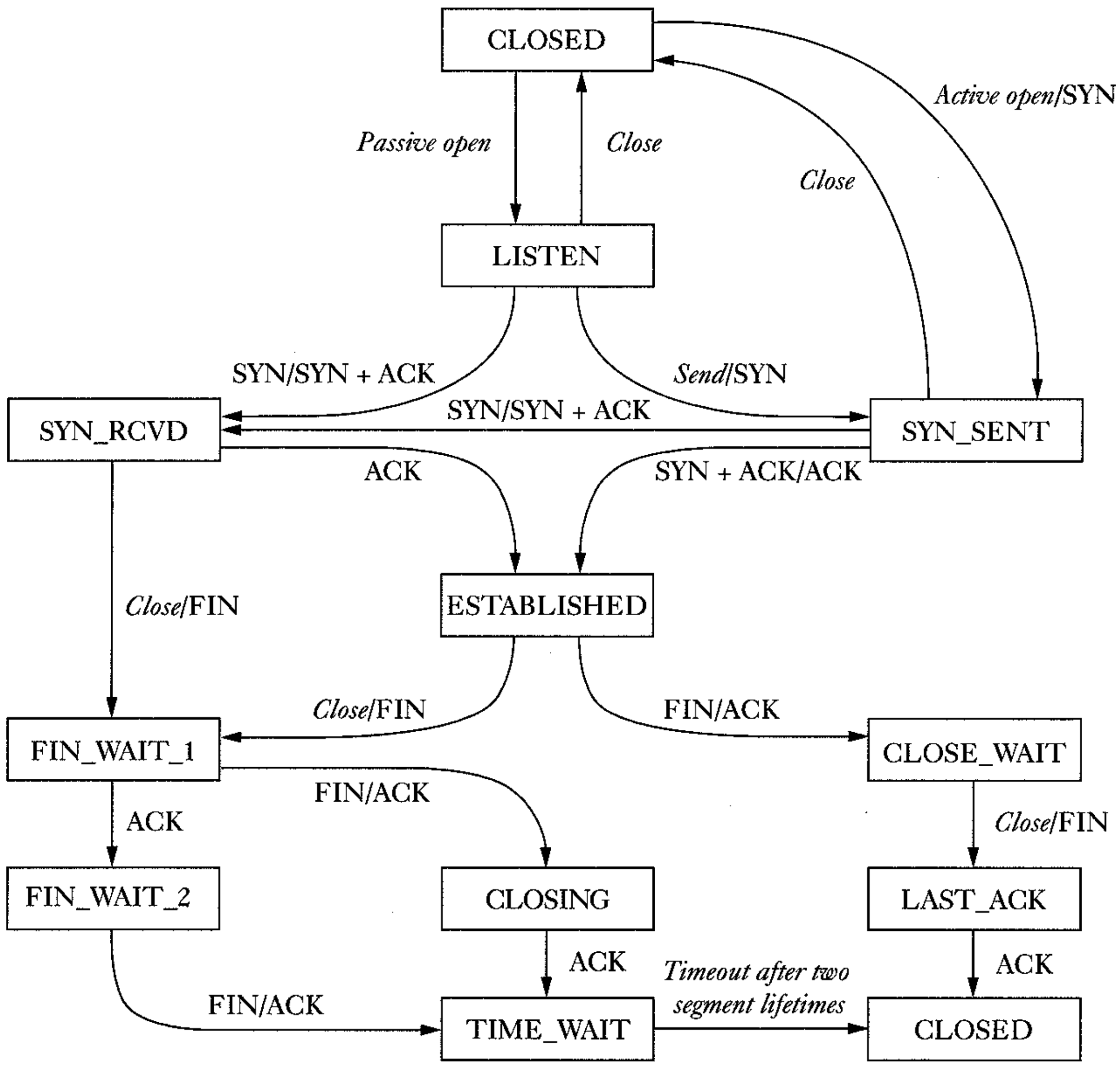
- Flags
  - SYN
  - FIN
  - RESET
  - PUSH
  - URG
  - ACK
- Fields



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

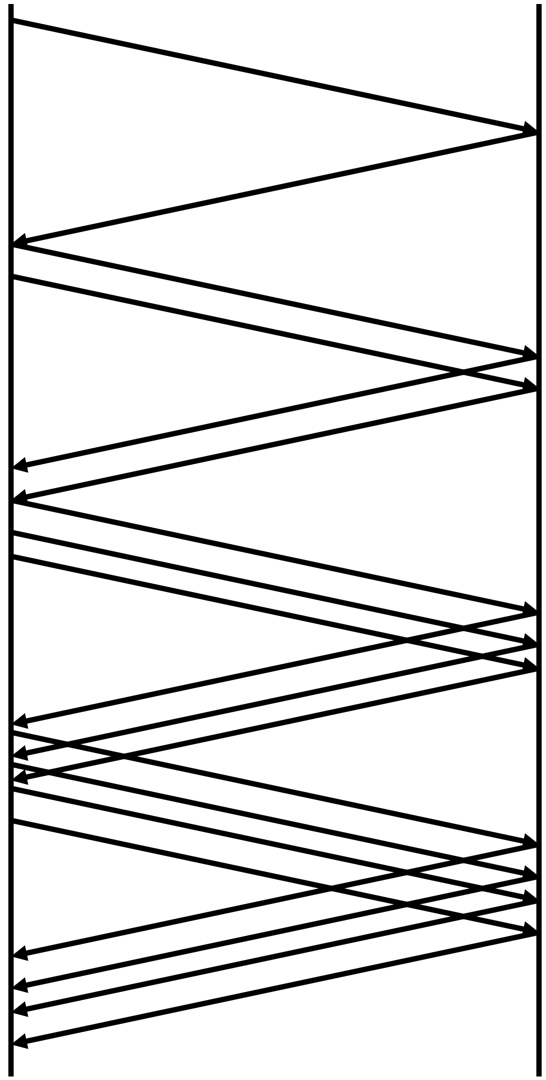
TCP Flow & Congestion Control

- 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

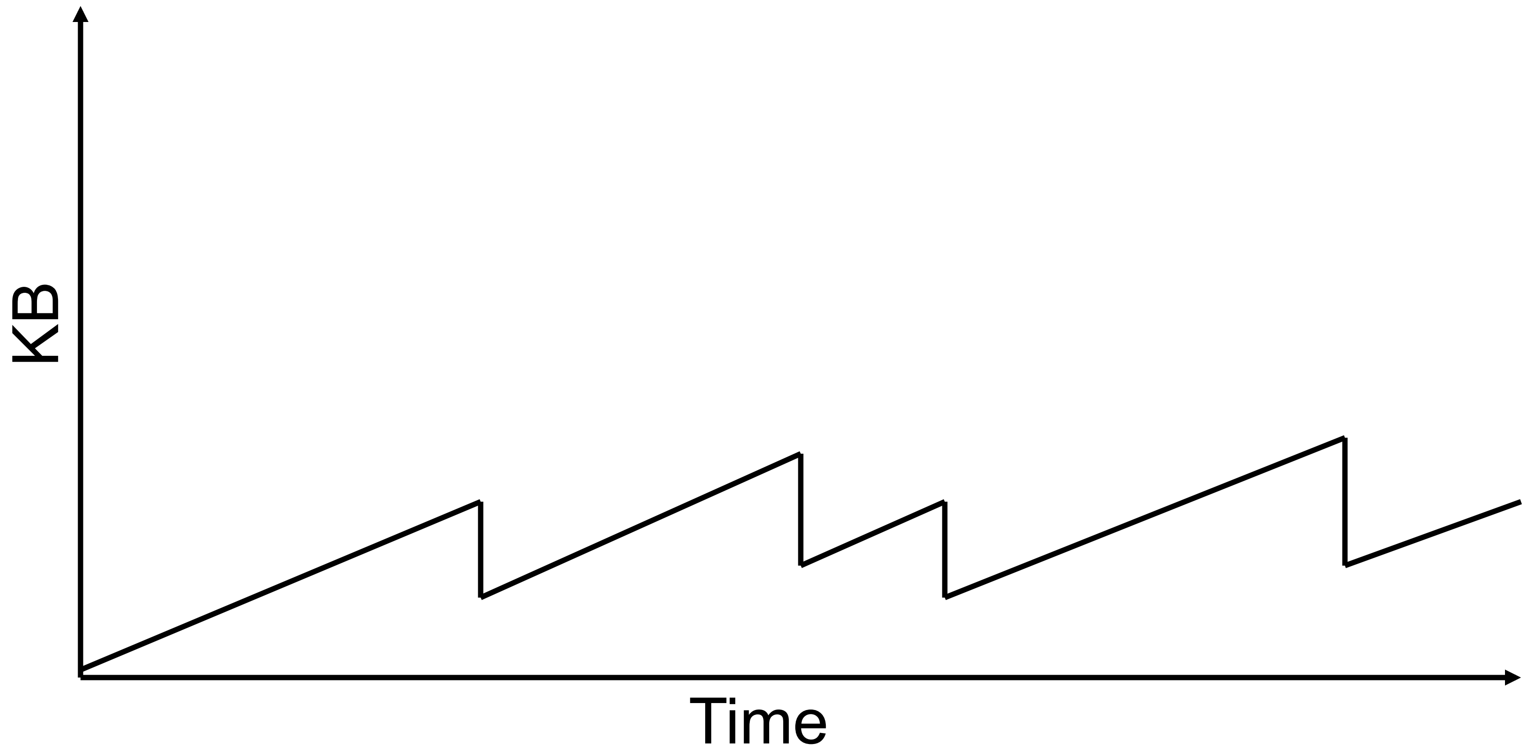
Increase and Decrease

- 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



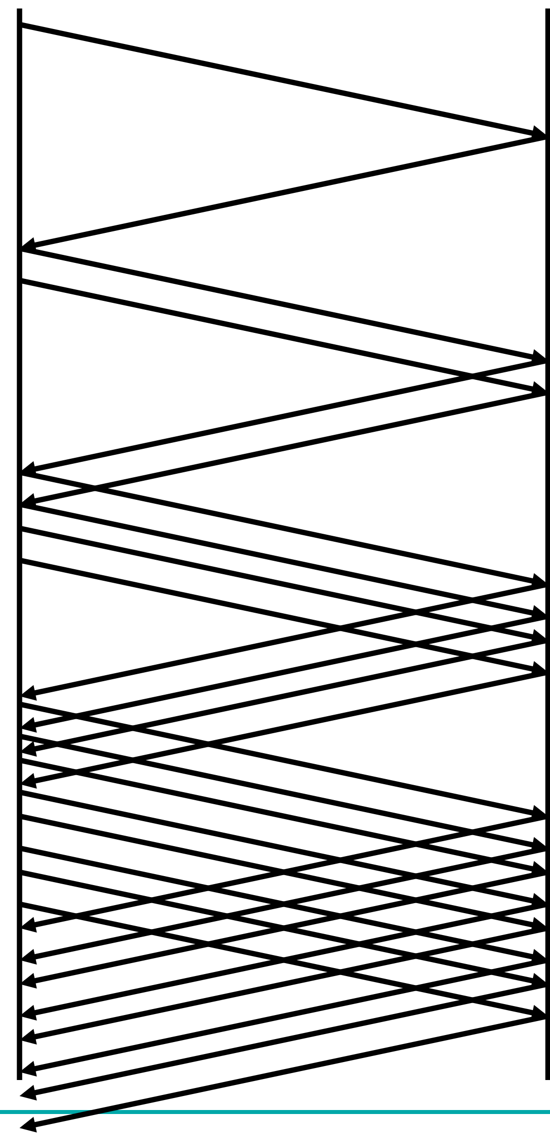
TCP Sawtooth Pattern



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

- ICMP message with an "echo request" is called 'ping'
- 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
- Sometimes called a "smurf attack"

(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

- Feb. 6 2007: 6 of 13 root servers suffered DDoS attack
- Oct. 21 2002: 9 of 13 root servers were swamped
  - Prompted changes in the architecture

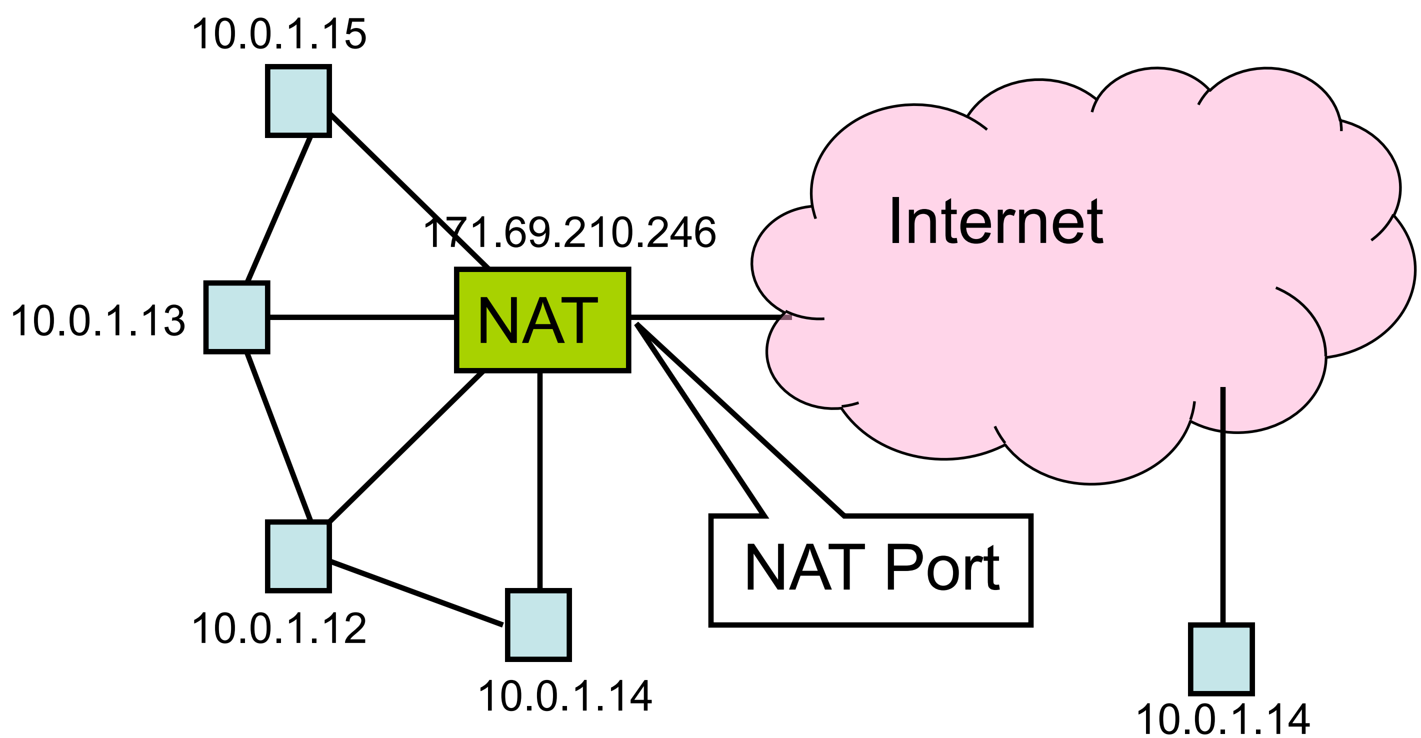
- Prevention?
  - Filtering?
  - Decentralized file storage?

Kinds of Firewalls

- Personal firewalls
  - Run at the end hosts
  - e.g. Norton, Windows, etc.
  - Benefit: has more application/user specific information
- Network Address Translators
  - Rewrites packet address information
- Filter Based
  - Operates by filtering based on packet headers
- Proxy based
  - Operates at the level of the application
  - e.g. HTTP web proxy

Network Address Translation

- Idea: Break the invariant that IP addresses are globally unique



NAT Behavior

- NAT maintains a table of the form:
  <client IP> <client port> <NAT ID>

- Outgoing packets (on non-NAT port):
  - Look for client IP address, client port in the mapping table
  - If found, replace client port with previously allocated NAT ID (same size as PORT #)
  - If not found, allocate a new unique NAT ID and replace source port with NAT ID
  - Replace source address with NAT address

NAT Behavior

- Incoming Packets (on NAT port)
  - Look up destination port number as NAT ID in port mapping table
  - If found, replace destination address and port with client entries from the mapping table
  - If not found, the packet is not for us and should be rejected

- Table entries expire after 2-3 minutes to allow them to be garbage collected

- "Private" IP addresses:
  - 192.168.x.x
  - 172.16.x.x
  - 172.31.x.x
  - 10.x.x.x

Benefits of NAT

- Only allows connections to the outside that are established from *inside.*
  - Hosts from outside can only contact internal hosts that appear in the mapping table, and they're only added when they establish the connection
  - Some NATs support firewall-like configurability

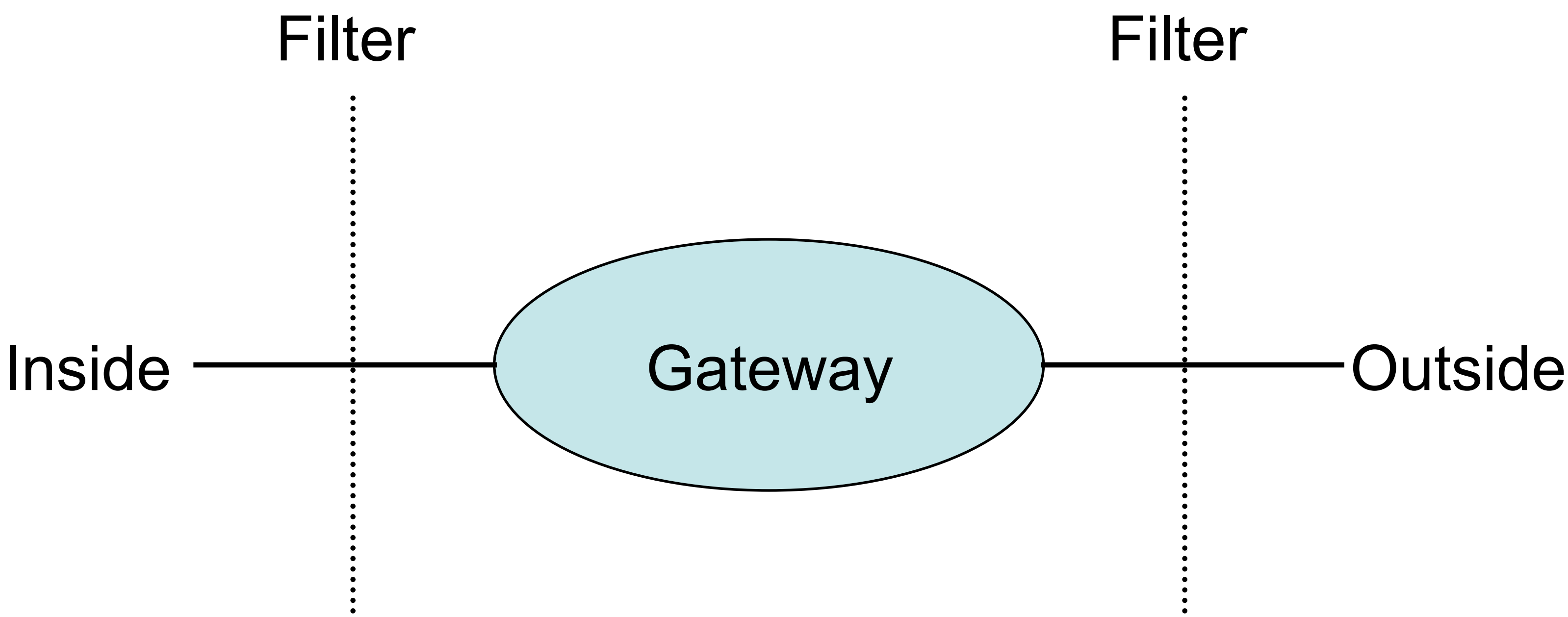
- Can simplify network administration
  - Divide network into smaller chunks
  - Consolidate configuration data

- Traffic logging
- Load balancing
- Robust failover

Drawbacks of NAT

- Rewriting IP addresses isn't so easy:
  - Must also look for IP addresses in other locations and rewrite them (may have to be protocol-aware)
  - Potentially changes sequence number information
  - Must validate/recalculate checksums
- Hinder throughput
- May not work with all protocols
  - Clients may have to be aware that NAT translation is going on
- Slow the adoption of IPv6?
- Limited filtering of packets / change packet semantics
  - For example, NATs may not work well with encryption schemes that include IP address information

Firewalls



- Filters protect against "bad" packets.
- Protect services offered internally from outside access.
- Provide outside services to hosts located inside.

Filtering Firewalls

- Filtering can take advantage of the following information from network and transport layer headers:
  - Source
  - Destination
  - Source Port
  - Destination Port
  - Flags (e.g. ACK)
  - Protocol type (e.g. UDP vs. TCP)

- Some firewalls keep state about open TCP connections
  - Allows conditional filtering rules of the form “if internal machine has established the TCP connection, permit inbound reply packets”

Filter Example

Action	ourhost	port	theirhost	port	comment
block	*	*	BAD	*	untrusted host
allow	GW	25	*	*	allow our SMTP port

Apply rules from top to bottom with assumed *default* entry:

Action	ourhost	port	theirhost	port	comment
block	*	*	*	*	default

Bad entry intended to allow connections to SMTP from inside:

Action	ourhost	port	theirhost	port	comment
allow	*	*	*	25	connect to their SMTP

This allows all connections from port 25, but an outside machine can run *anything* on its port 25!

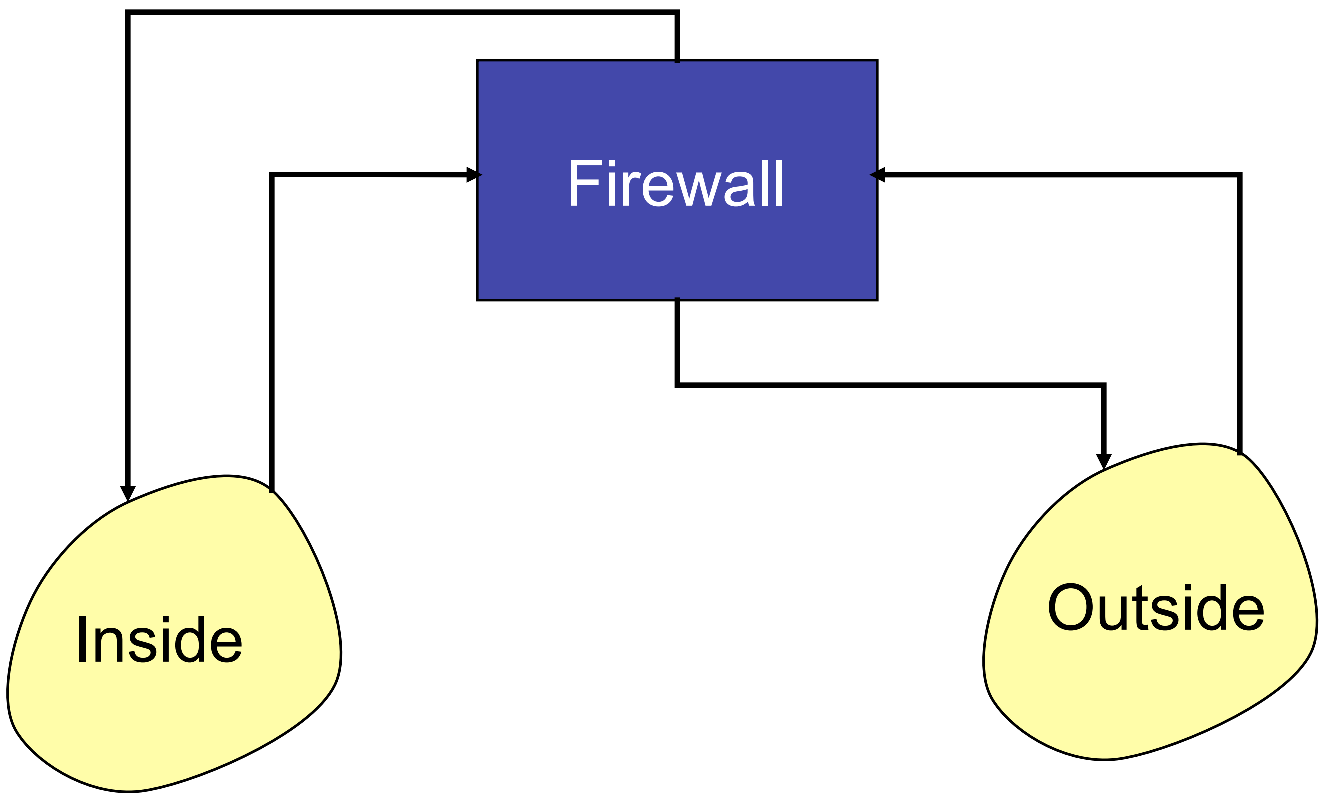
Filter Example Continued

Permit *outgoing* calls to port 25.

Action	src	port	dest	port	flags	comment
allow	123.45.6.*	*	*	25	*	their SMTP
allow	*	25	*	*	ACK	their replies

This filter doesn’t protect against IP address spoofing. The bad hosts can “pretend” to be one of the hosts with addresses 123.45.6.* .

When to Filter?



On Input or Output?

- Filtering on *output* can be more efficient since it can be combined with table lookup of the route.
- However, some information is lost at the output stage
  - e.g. the physical input port on which the packet arrived.
  - Can be useful information to prevent address spoofing.
- Filtering on *input* can protect the router itself.

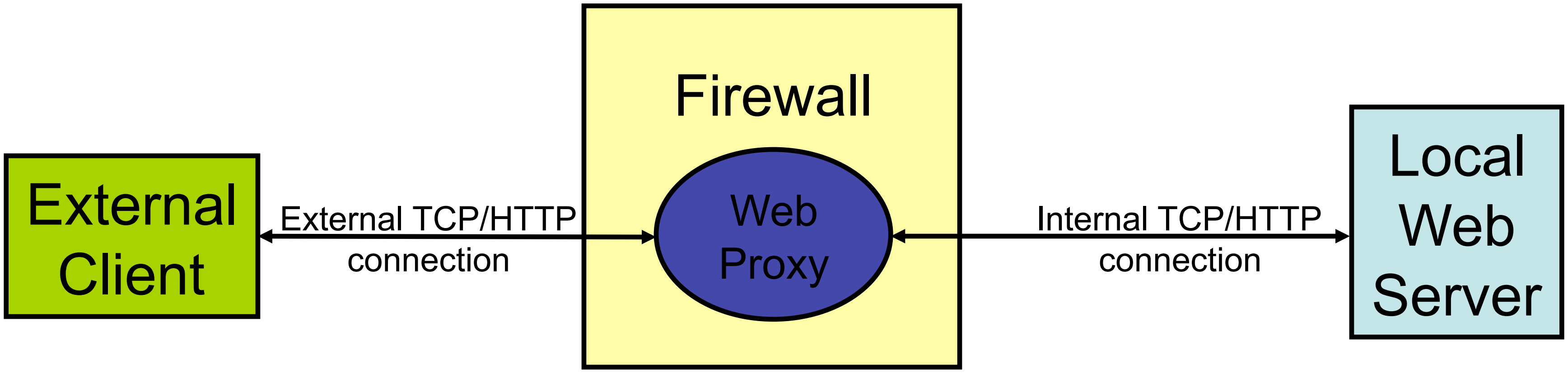
Principles for Firewall Configuration

- General principal: Filter as early as possible
- Least Privileges:
  - Turn off everything that is unnecessary (e.g. Web Servers should disable SMTP port 25)
- Failsafe Defaults:
  - By default should reject
  - (Note that this could cause usability problems…)
- Egress Filtering:
  - Filter outgoing packets too!
  - You know the valid IP addresses for machines internal to the network, so drop those that aren't valid.
  - This can help prevent DoS attacks in the Internet.

Example “real” firewall config script

```
############
# FreeBSD Firewall configuration.
# Single-machine custom firewall setup. Protects somewhat
# against the outside world.
############

# Set this to your ip address.
ip="192.100.666.1"
setup_loopback

# Allow anything outbound from this address.
${fwcmd} add allow all from ${ip} to any out

# Deny anything outbound from other addresses.
${fwcmd} add deny log all from any to any out

# Allow inbound ftp, ssh, email, tcp-dns, http, https, imap, imaps,
# pop3, pop3s.
${fwcmd} add allow tcp from any to ${ip} 21 setup
${fwcmd} add allow tcp from any to ${ip} 22 setup
${fwcmd} add allow tcp from any to ${ip} 25 setup
${fwcmd} add allow tcp from any to ${ip} 53 setup
${fwcmd} add allow tcp from any to ${ip} 80 setup
${fwcmd} add allow tcp from any to ${ip} 443 setup
…
```

Proxy-based Firewalls



- Proxy acts like *both* a client and a server.
- Able to filter using application-level info
  - For example, permit some URLs to be visible outside and prevent others from being visible.
- Proxies can provide other services too
  - Caching, load balancing, etc.
  - FTP and Telnet proxies are common too

Benefits of Firewalls

- Increased security for internal hosts.
- Reduced amount of effort required to counter break ins.
- Possible added convenience of operation within firewall (with some risk).
- Reduced legal and other costs associated with hacker activities.

Drawbacks of Firewalls

- Costs:
  - Hardware purchase and maintenance
  - Software development or purchase, and update costs
  - Administrative setup and training, and ongoing administrative costs and trouble-shooting
  - Lost business or inconvenience from broken gateway
  - Loss of some services that an open connection would supply.
- False sense of security
  - Firewalls don't protect against viruses…