

Security Topics in Networking and Distributed Systems

CS 590D

Lecture 3

Department of Computer Sciences
Purdue University

Outline

- Attacks on TCP exploiting the handshake protocol.

Analysis of a Denial of Service Attack on TCP, Christoph L. Schuba, Ivan V. Krsul, Markus G. Kuhn, Eugene H. Spafford, Aurobindo Sundaram, Diego Zamboni, Security & Privacy 1997

- Attacks on TCP exploiting the congestion control mechanism

Low-Rate TCP-Targeted Denial of Service Attacks (The Shrew vs. the Mice and Elephants), by Aleksandar Kuzmanovic and Edward W. Knightly, SIGCOM 2003



Transmission Control Protocol - TCP

- Connection oriented protocol for a user process:
 - Established a connection (channel) between two end-points
 - **Reliable**, full-duplex channel: acknowledgements, retransmissions, timeouts, flow-control, congestion control
 - The packets are delivered in the same order in which they were sent.

Establishing the Connection

- Hardware addresses identify network cards
- IP addresses identify hosts
- Names identify hosts in a human friendly way.
- Transport protocols (TCP and UDP) ensure communication between processes.
- How do computers differentiate what data is for which process?

Ports

- Once data reached a computer, the port helps identifying what is the process the data is for.
- In general servers use well-known ports, while clients use ephemeral ports
- Example: port 80 is assigned to web server (HTTP)
- Port numbers:
 - Well-known ports: 0 - 1023
 - Registered ports: 1024 – 49151
 - Dynamic/private ports: 49152 - 65535

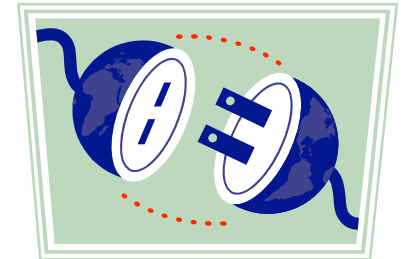
Socket

- Socket: identifies a communication end-point.

socket = (IP address, port number)

- Socket pair: uniquely identifies a TCP connection over the Internet:

socket pair = (local IP address, local IP port, remote IP, remote IP port)

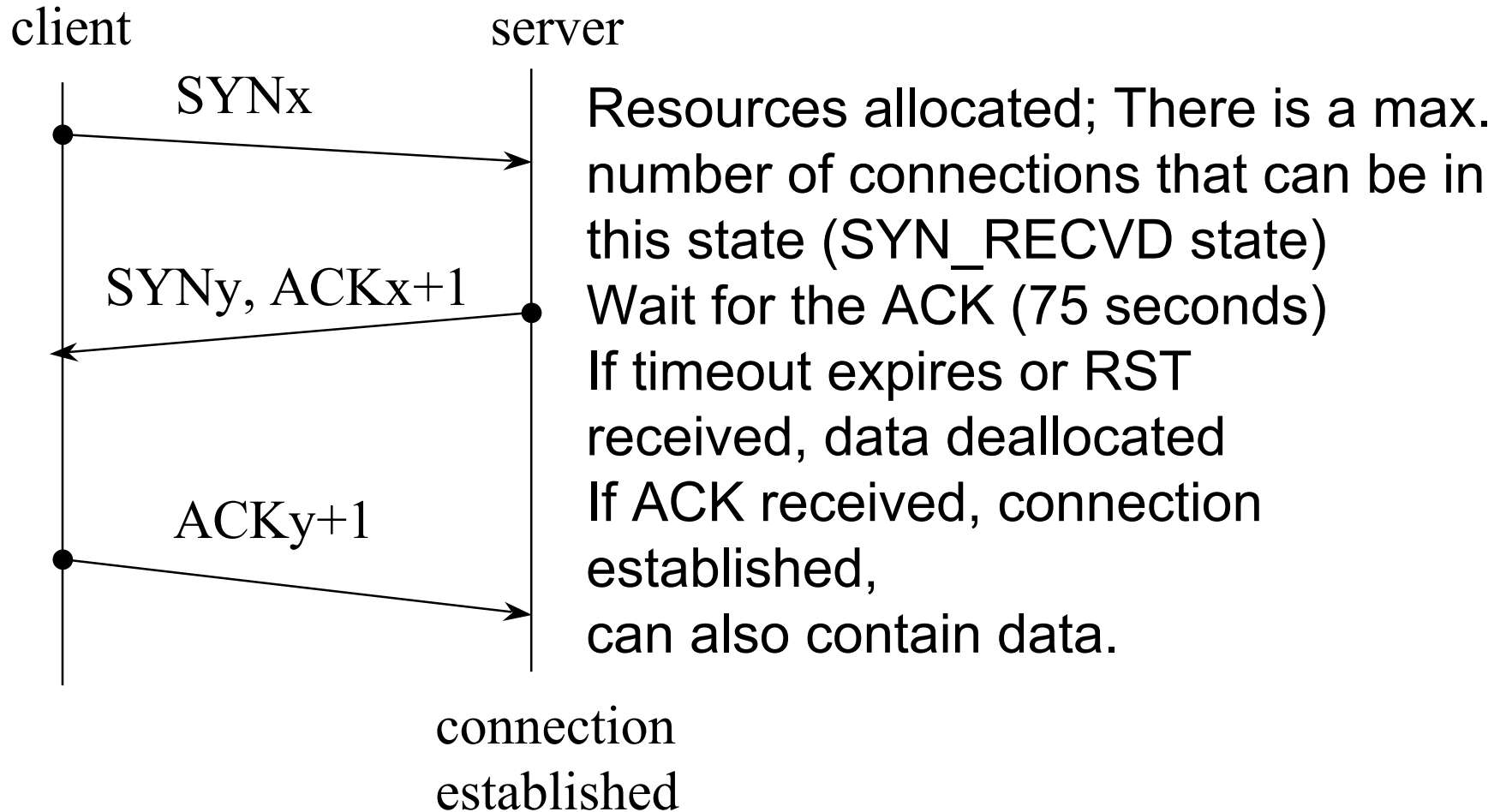


- Information is maintained by OS in the socket structure:

- Protocol information
- State
- Addressing
- Connection queues
- Buffers
- flags

RESOURCES

TCP Handshake



SYN Attack



- An attacker sends many SYN with source address spoofed packets to a target.
- If the limit is reached, target machine will refuse any incoming connections till the timeout expires.
- Spoofed address chosen to be a non-existent one (If the spoofed address belongs to a machine, then SYN+ACK packet will reach that machine and trigger a RST answer that will close the connection).

WHY IS THIS ATTACK POSSIBLE?

Basis of the Attack

- There is no authentication of the source of the packets
- Addresses can be spoofed
- The protocol requires asymmetric allocation of resources

Possible Solutions

- Configuration optimizations
- Infrastructure improvements
- Connection establishment improvements
- Firewall approach
- Active monitoring



Configuration Optimizations

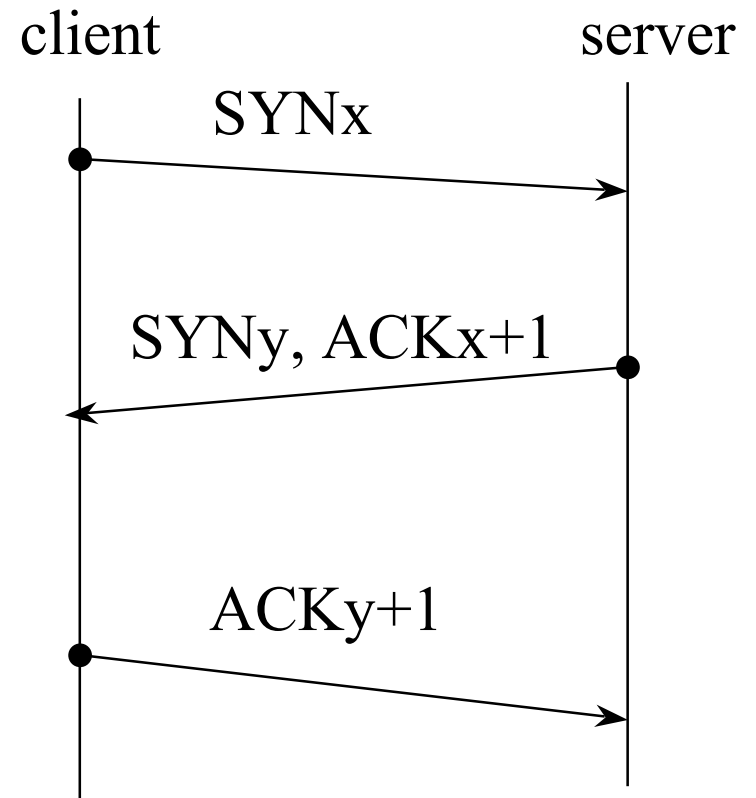
- System configuration
 - Reduce the timeout to 10 seconds
 - Increase the size of the queue
 - Disable non-essential services, reducing the number of ports to be attacked
- Router configuration
 - Block packets that have source addresses from the internal network
 - Block packets to the outside that have source addresses from outside the internal network

Infrastructure Improvements

- If addresses prefixes separate clear the inside from the outside, then router configuration can be improved.
- Example: routers that attach an organization or an ISP to a backbone network.

Connection Establishment Improvements

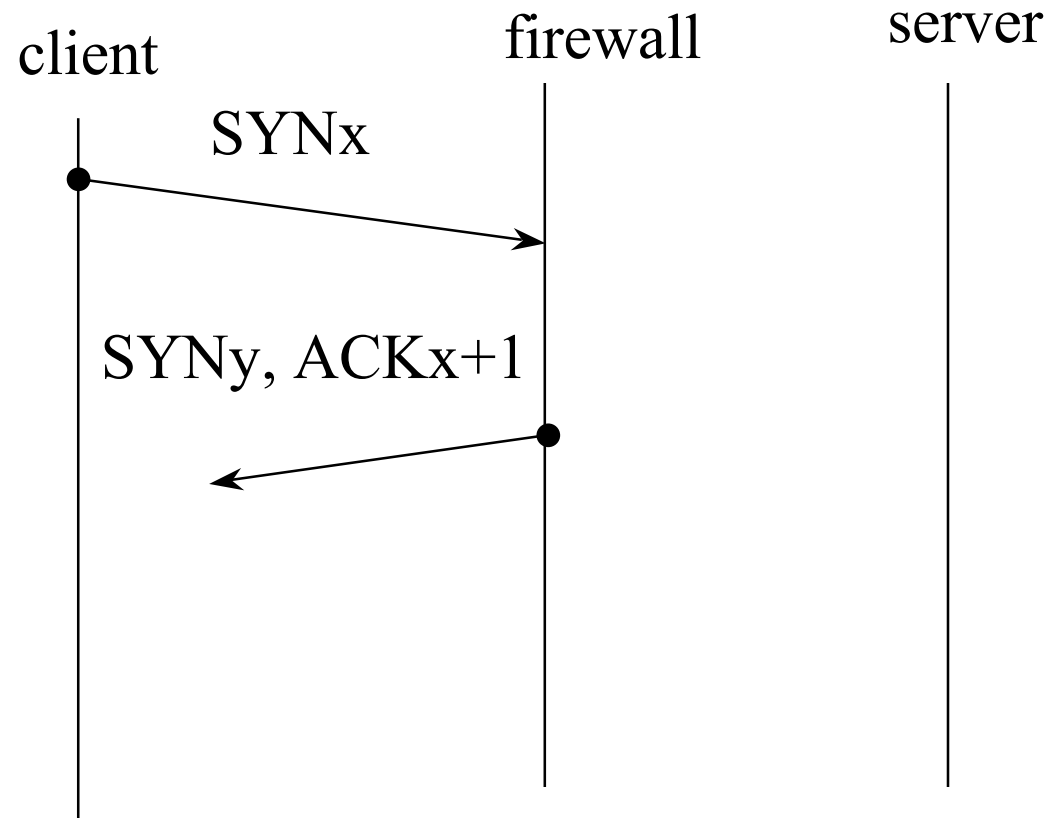
- The handshake protocol requires the sequence y from the 2nd and 3rd step to be the same. WHY?
- One way the destination can recreate y is to use a hash function based on addresses, x and a secret key known only by the destination.



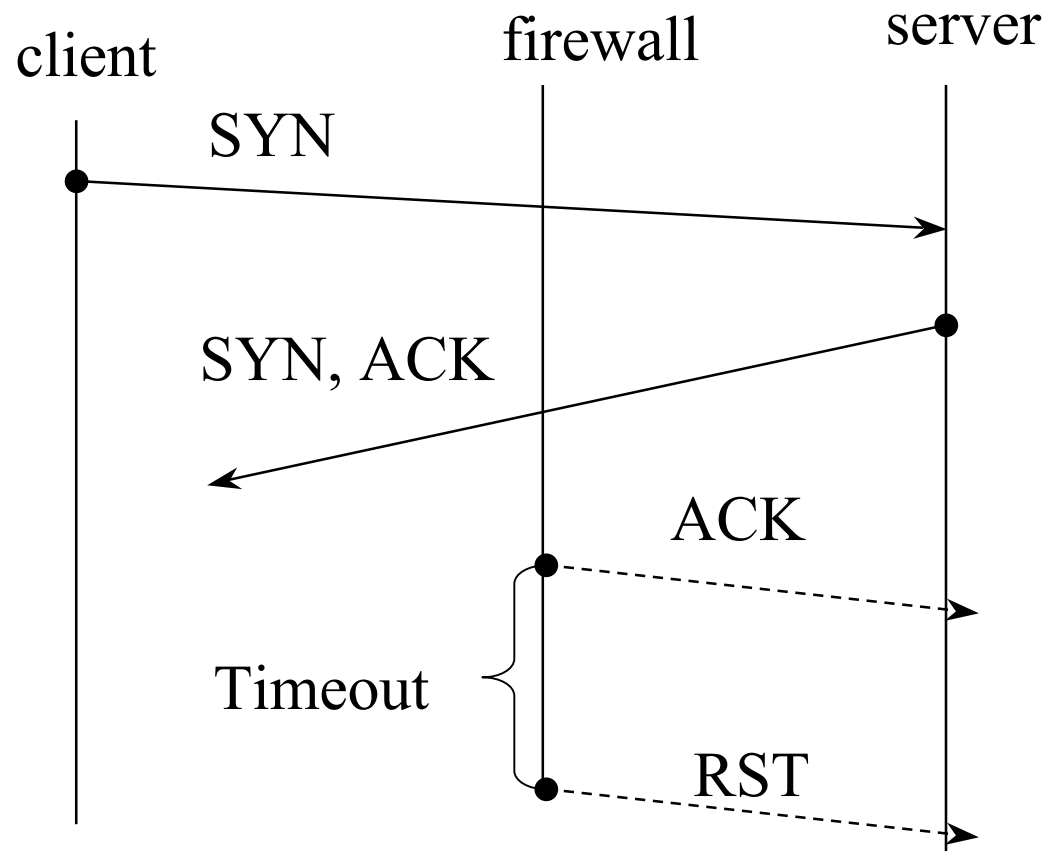
Firewall Approach

- Main idea: each packet for inside network is first examined by the firewall
- Additional delays
- Two approaches:
 - Firewall as a relay
 - Firewall as a gateway

Firewall as a Relay: Attack Scenario



Firewall as a Semi-transparent Gateway: Attack Scenario



Active Monitoring

- Monitor the TCP traffic within a local area network and figure out which ones are illegitimate connection.
- Send RST for the illegitimate connections (this closes the connection).
- Does not require protocol stack modification.
- Monitor can be tricked to classify bad addresses as good addresses

Attack Against TCP Exploiting Its Congestion Control Mechanism

What's Different?

- Traditional attacks require **high-rate** transmission (flood of SYN packets), unusual network traffic, **attackers are relatively easy to detect and filter.**
- TCP can be attacked by using TCP friendly traffic, **low rate**, therefore it can cause **maximal damage without detection.**

TCP Congestion Control

- Source determines how much bandwidth is available for it to send, it starts slow and increases the window of send packet based on ACKS.
- ACKS are also used to control the transmission of packets.
- Uses Additive Increase Multiplicative Decrease (AIMD)
- Uses Retransmission Timeout (RTO) to avoid congestion
- **TCP Fairness:** if k TCP sessions share same bottleneck link of bandwidth B , each should have average rate of B/k

AIMD

- CongestionWindow (cwnd) is set based on the level of congestion

MaxWindow = min (CongestionWindow, AdvertisedWindow)

EffectiveWindow = MaxWindow – (LastByteSent - LastByteAcked)

- If a timeout occurred TCP cwnd is cut in half.
- For each received ACK, **cwnd** is incremented fractionally.

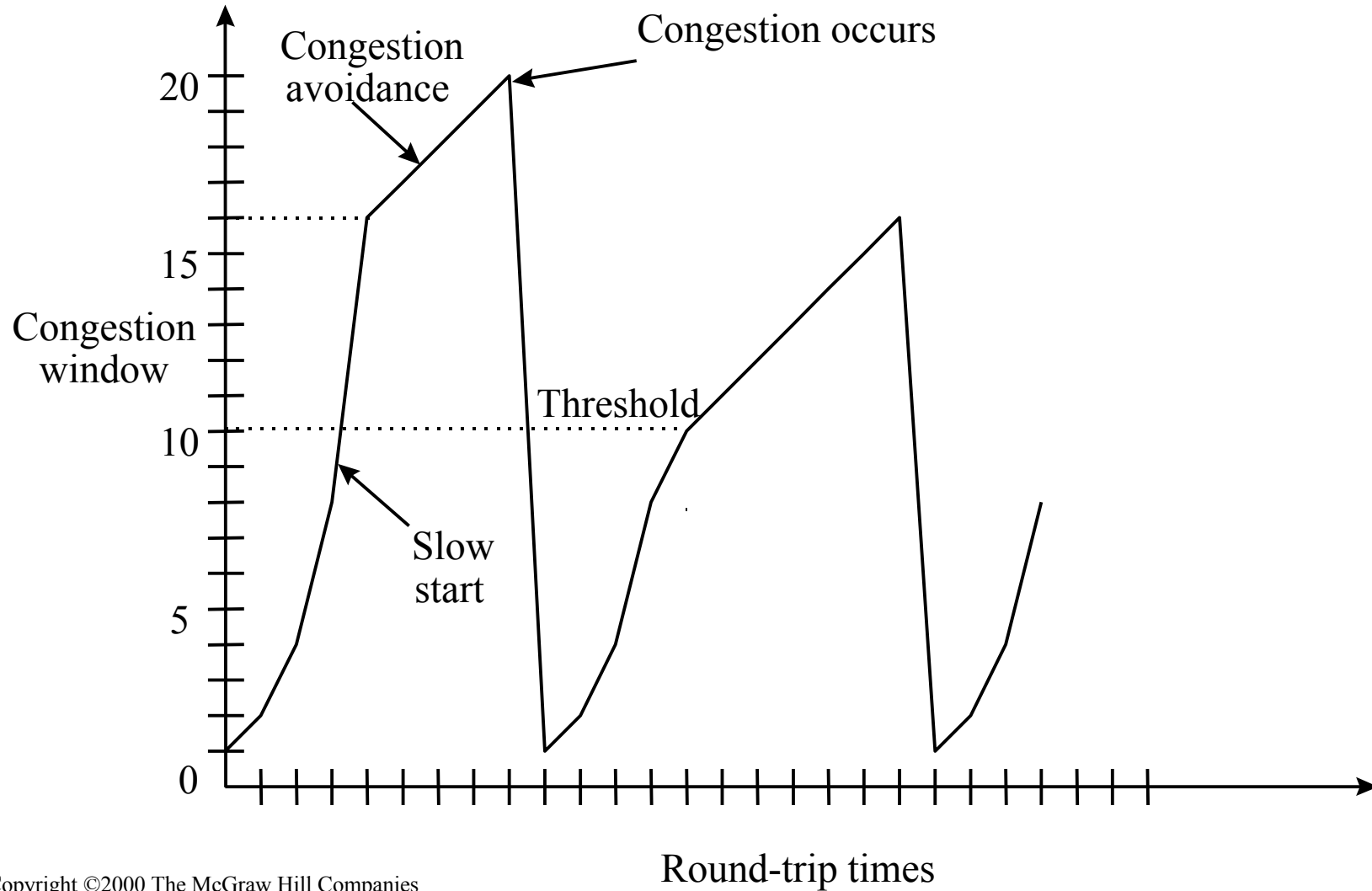
TCP Congestion Control

- If round-trip times RTT is about 10-100 ms, TCP performs AIMD
- If severe congestion occurs retransmission timeout RTO is used:
- If less than 3 duplicate ACKs are received before RTO expires
- $cwnd = 1$ packet
 - $RTO = 2 * RTO$
 - resend the packet
 - If packet successfully received TCP enters slow start.
 - **MIN VAL for RTO is 1 sec.**

Slow Start

- Slow start mechanism provides an initial exponential increase for cwnd.
- $Cwnd = 1$, for every received ACK, cwnd is incremented
- Two slow start situations:
 - At the very beginning of a connection (also known as **cold start**).
 - When the connection goes dead waiting for a timeout to occur.

TCP Congestion Control



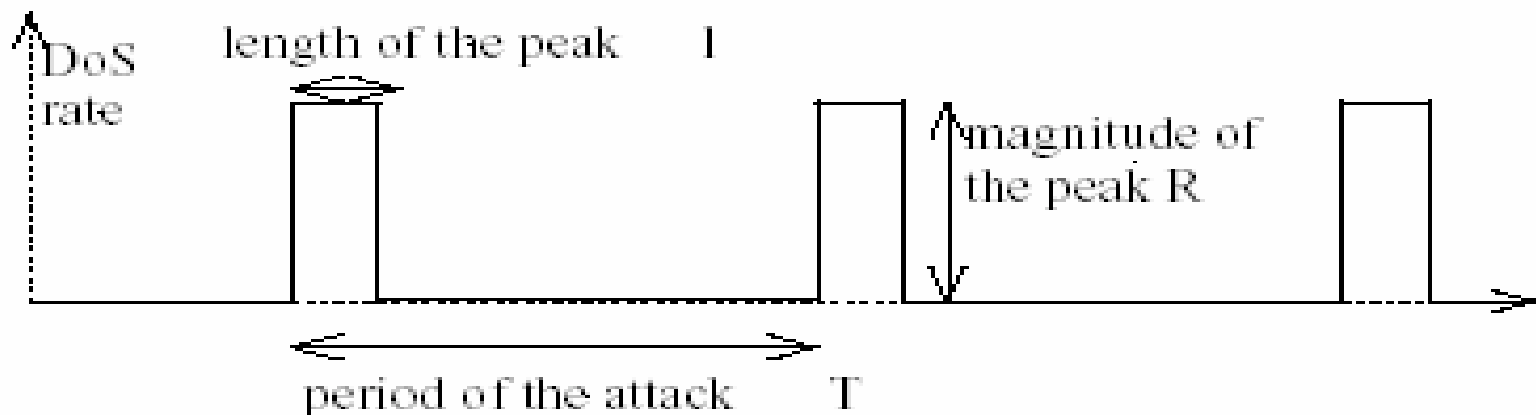
Copyright ©2000 The McGraw Hill Companies

Leon-Garcia & Widjaja: *Communication Networks*

Figure 7.63

The Attack

- All the attacker needs to do is generate a TCP flow to force the targeted TCP connection to repeatedly enter a retransmission timeout state
- Very effective, TCP throughput degrades significantly
- Sending high-rate, RTT scale short duration bursts and repeating periodically at RTO scale period.



Basis of the Attack

- Protocol is homogenous and deterministic
 - protocols react in a pre-defined way
 - tradeoff of vulnerability vs. predictability
- Periodic outages synchronize TCP flow states and deny their service
- Slow time scale protocol mechanisms enable low-rate attacks
 - outages at RTO scale, pulses at RTT scale imply low average rate

Solutions???

- Factors: **randomization, connectivity, accountability**
- Router-Assisted Mechanisms: Routers identify and regulate the misbehaving flows
 - Router-Based algorithms
 - Random early detection with preferential dropping (queue management)
- End-point minRTO Randomization
- They mitigate the attack, but can not eliminate it

Summary

- High rate DoS attacks
 - Cause the server to be loaded and not accept connection anymore
 - Easier to detect
 - Several methods that diminish significantly the attacks
- Low-rate DoS attacks
 - Exploit the vulnerability of the TCP's congestion control algorithm
 - Very difficult to detect because the rate is low
 - Degrade the victim's throughput significantly
 - Existing solutions only mitigate the attack

Next Lecture



- Thursday: DDOS
 - A Framework for Classifying Denial of Service Attacks, by Alefiya Hussain, John Heidemann, and Christos Papadopoulos, SIGCOM 2003
 - The DoS project's trinoo distributed denial of service attack tool, by David Dittrich