Outline

- Definition
- Point-to-point network denial of service
  - Smurf
- Distributed denial of service attacks
  - Trin00, TFN, Stacheldraht, TFN2K
- TCP SYN Flooding and Detection

Denial of Service Attack Definition

- An explicit attempt by attackers to prevent legitimate users of a service from using that service
- Threat model – taxonomy from *CERT*
  - Consumption of network connectivity and/or bandwidth
  - Consumption of other resources, e.g. queue, CPU
  - Destruction or alternation of configuration information
    - Malformed packets confusing an application, cause it to freeze
  - Physical destruction or alternation of network components

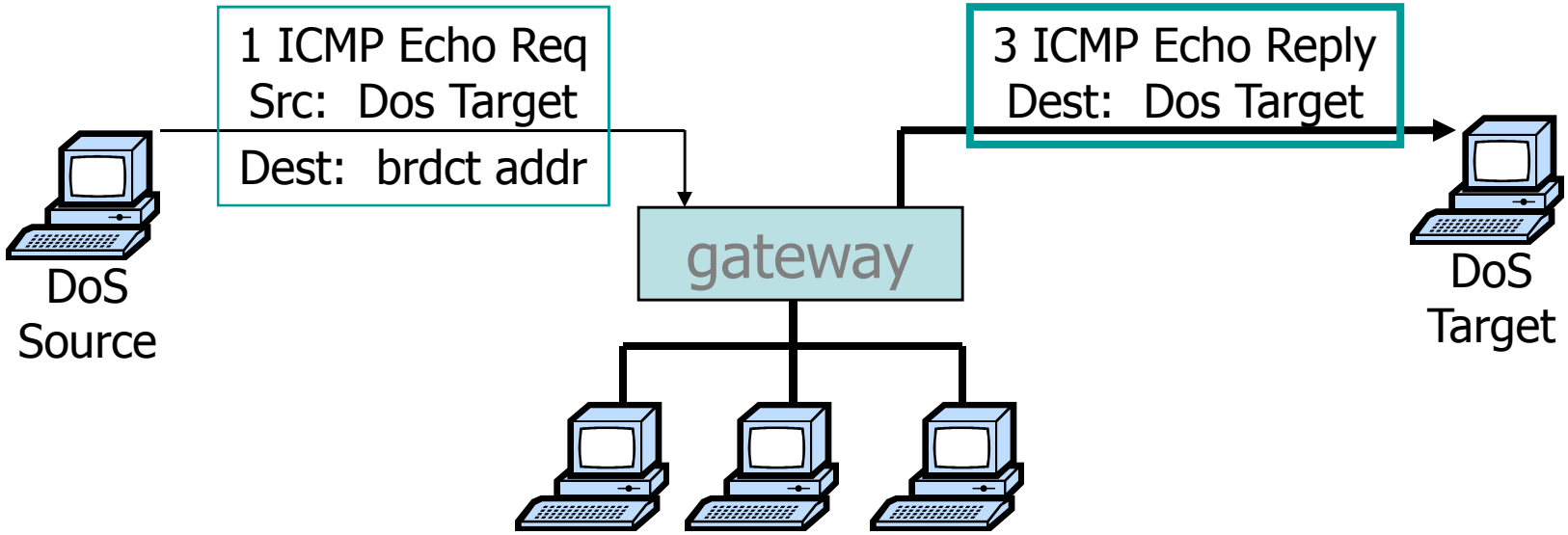
Status

- DoS attacks increasing in frequency, severity and sophistication
  - *32%* respondents detected DoS attacks (1999 CSI/FBI survey)
  - Yahoo, Amazon, eBay and MicroSoft DDoS attacked
  - About *4,000* attacks per week in 2000
  - Internet's root DNS servers (9 out of 13) attacked on Oct 2002

Two General Classes of Attacks

- Flooding Attacks
  - Point-to-point attacks: TCP/UDP/ICMP flooding, Smurf attacks
  - Distributed attacks: hierarchical structures
- Corruption Attacks
  - Application/service specific

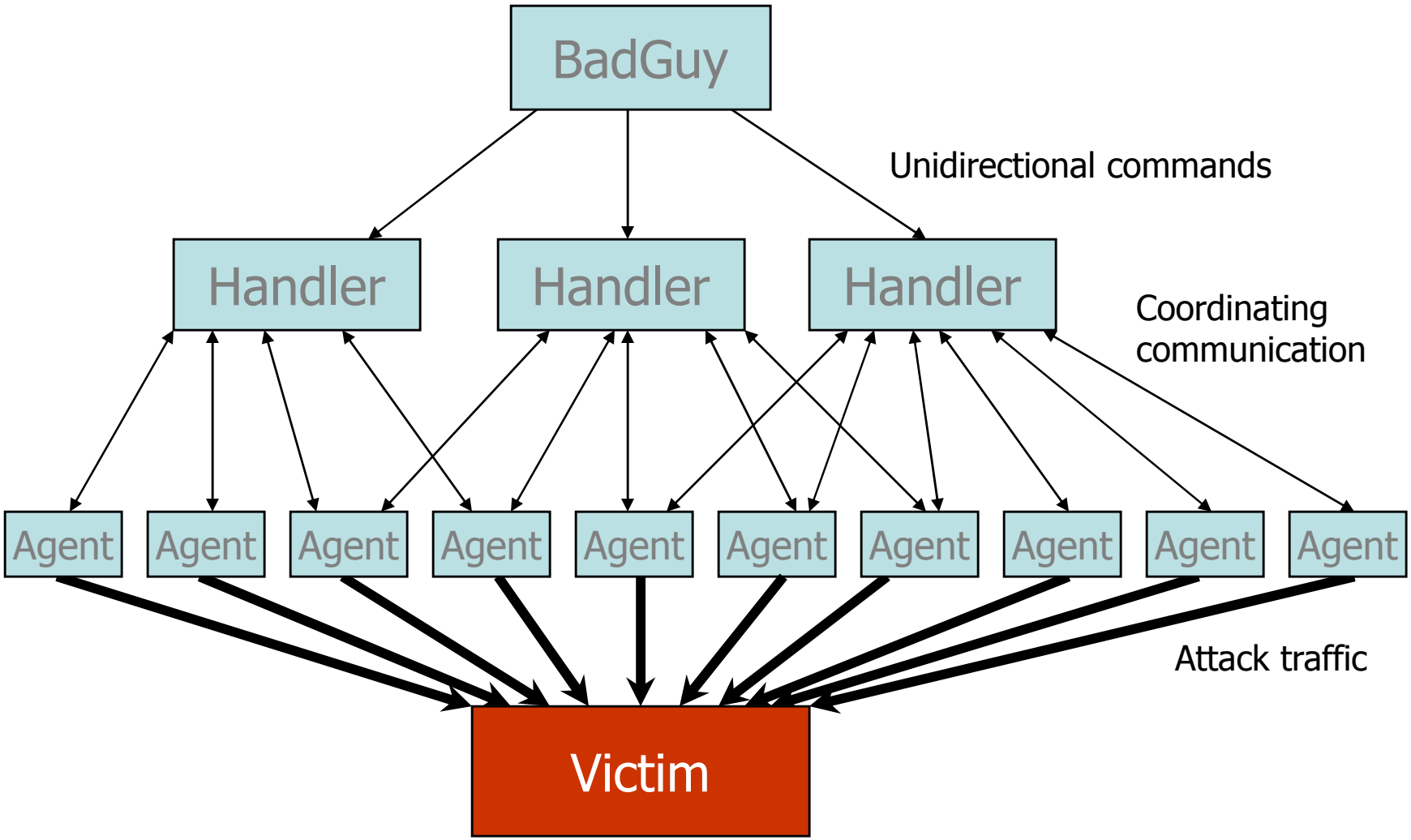
Smurf DoS Attack



- Send ping request to brdcst addr (ICMP Echo Req)
- Lots of responses:
  - Every host on target network generates a ping reply (ICMP Echo Reply) to victim
  - Ping reply stream can overload victim

Prevention: reject external packets to brdcst address.

DDOS


BadGuy
Unidirectional commands
Handler
Handler
Handler
Coordinating communication
Agent
Agent
Agent
Agent
Agent
Agent
Agent
Agent
Agent
Agent
Attack traffic
Victim

Attack using Trin00

- In August 1999, network of > 2,200 systems took University of Minnesota offline for 3 days
  - scan for known vulnerabilities, then attack with UDP traffic
  - once host compromised, script the installation of the DDoS master agents
- According to the incident report
  - Took about 3 seconds to get root access
  - In 4 hours, set up > 2,200 agents

Can you find source of attack?

- Hard to find BadGuy
  - Originator of attack compromised the handlers
  - Originator not active when DDOS attack occurs
- Can try to find agents
  - Source IP address in packets is not reliable
  - Need to examine traffic at many points, modify traffic, or modify routers

Source Address Validity

- Spoofed Source Address
  - random source addresses in attack packets
  - Subnet Spoofed Source Address
    - random address from address space assigned to the agent machine's subnet
  - En Route Spoofed Source Address
    - address spoofed en route from agent machine to victim
- Valid Source Address
  - used when attack strategy requires several request/reply exchanges between an agent and the victim machine
  - target specific applications or protocol features

Attack Rate Dynamics

Agent machine sends a stream of packets to the victim

- Constant Rate
  - Attack packets generated at constant rate, usually as many as resources allow
- Variable Rate
  - Delay or avoid detection and response
  - Increasing Rate
    - gradually increasing rate causes a slow exhaustion of the victim's resources
  - Fluctuating Rate
    - occasionally relieving the effect
    - victim can experience periodic service disruptions

Outline

- Definition
- Point-to-point network denial of service
  - Smurf
- Distributed denial of service attacks
  - Trin00, TFN, Stacheldraht, TFN2K
- TCP SYN Flooding and Detection

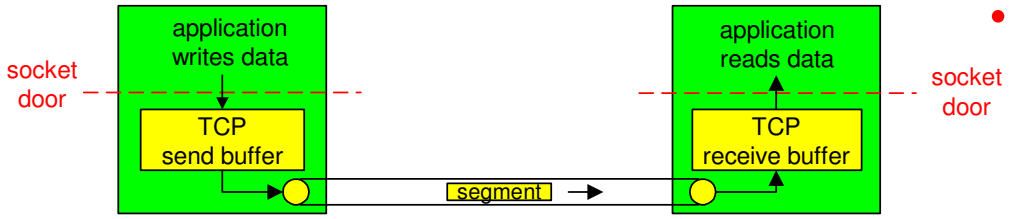
SYN Flooding Attack

- 90% of DoS attacks use TCP SYN floods
- Streaming spoofed TCP SYNs
- Takes advantage of three way handshake
- Server start "half-open" connections
- These build up… until queue is full and all additional requests are blocked

TCP: Overview

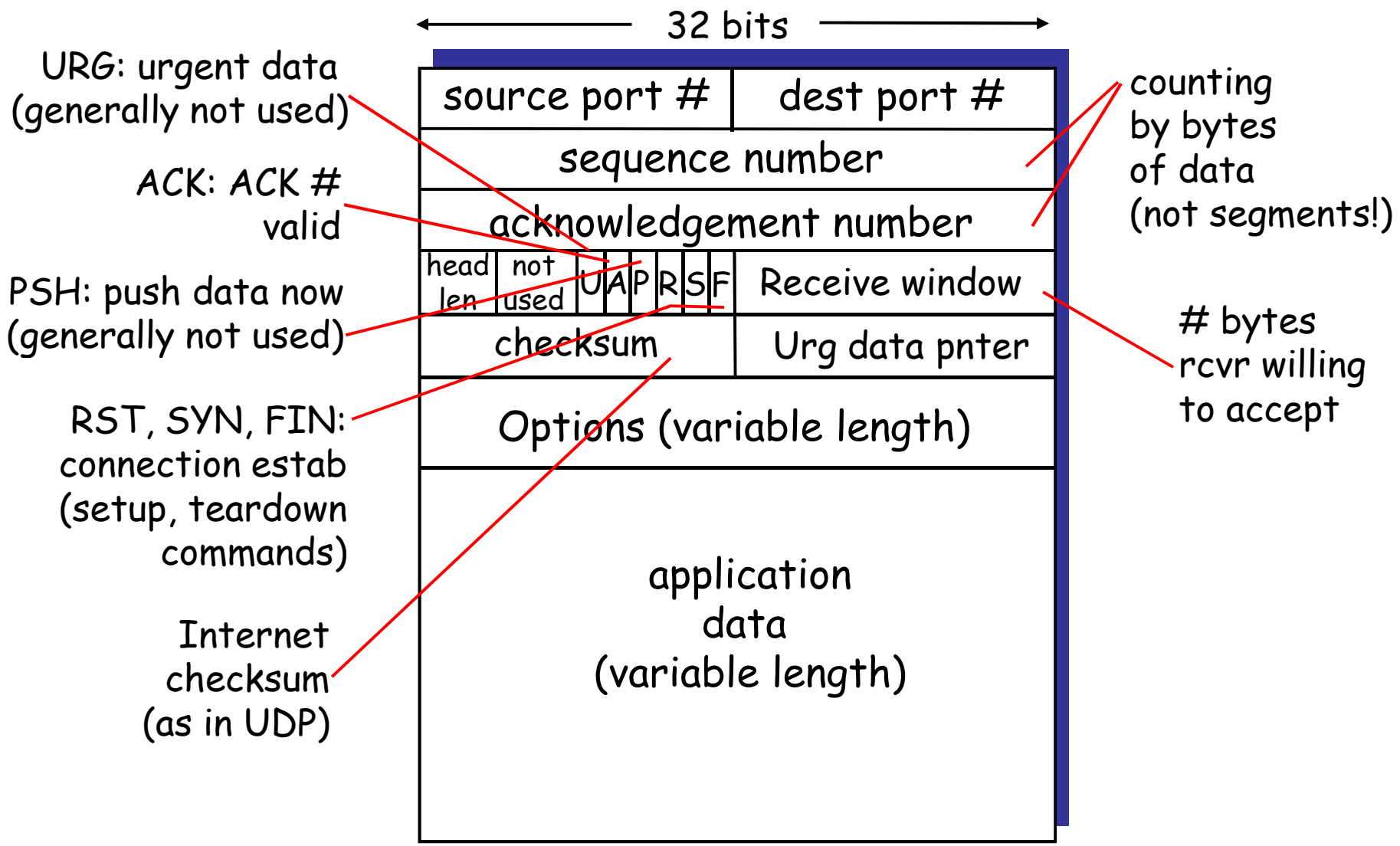
RFCs: 793, 1122, 1323, 2018, 2581

- point-to-point:
  - one sender, one receiver
- reliable, in-order *byte steam:*
  - no "message boundaries"
- pipelined:
  - TCP congestion and flow control set window size
- *send & receive buffers*



- full duplex data:
  - bi-directional data flow in same connection
  - MSS: maximum segment size
- connection-oriented:
  - handshaking (exchange of control msgs) init's sender, receiver state before data exchange
- flow controlled:
  - sender will not overwhelm receiver

TCP segment structure



TCP Connection Management

Recall: TCP sender, receiver establish "connection" before exchanging data segments

- initialize TCP variables:
  - seq. #s
  - buffers, flow control info (e.g. **`RcvWindow`**)
- *client:* connection initiator
- *server:* contacted by client

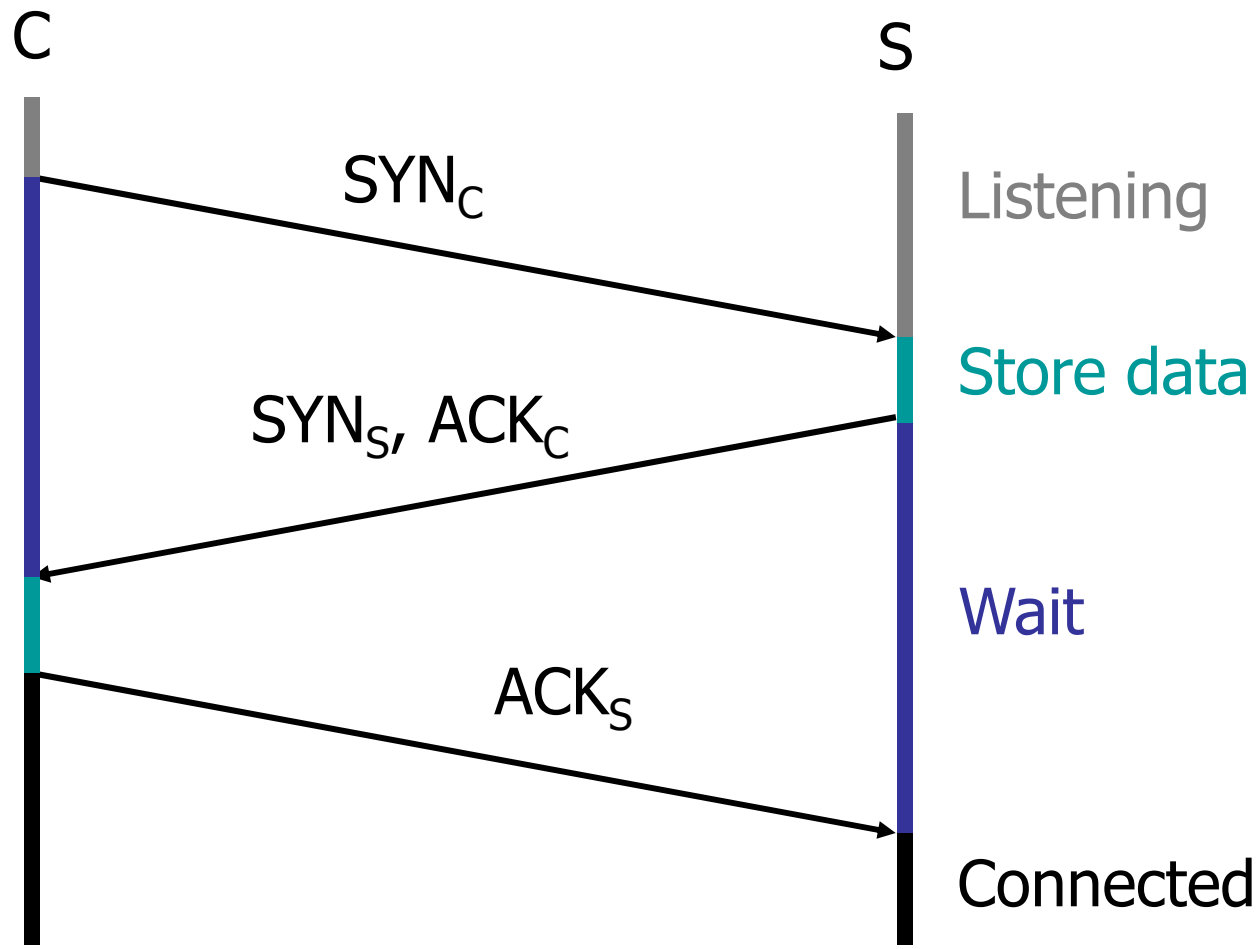
Three way handshake:

Step 1: client host sends TCP SYN segment to server
- specifies initial seq #
- no data

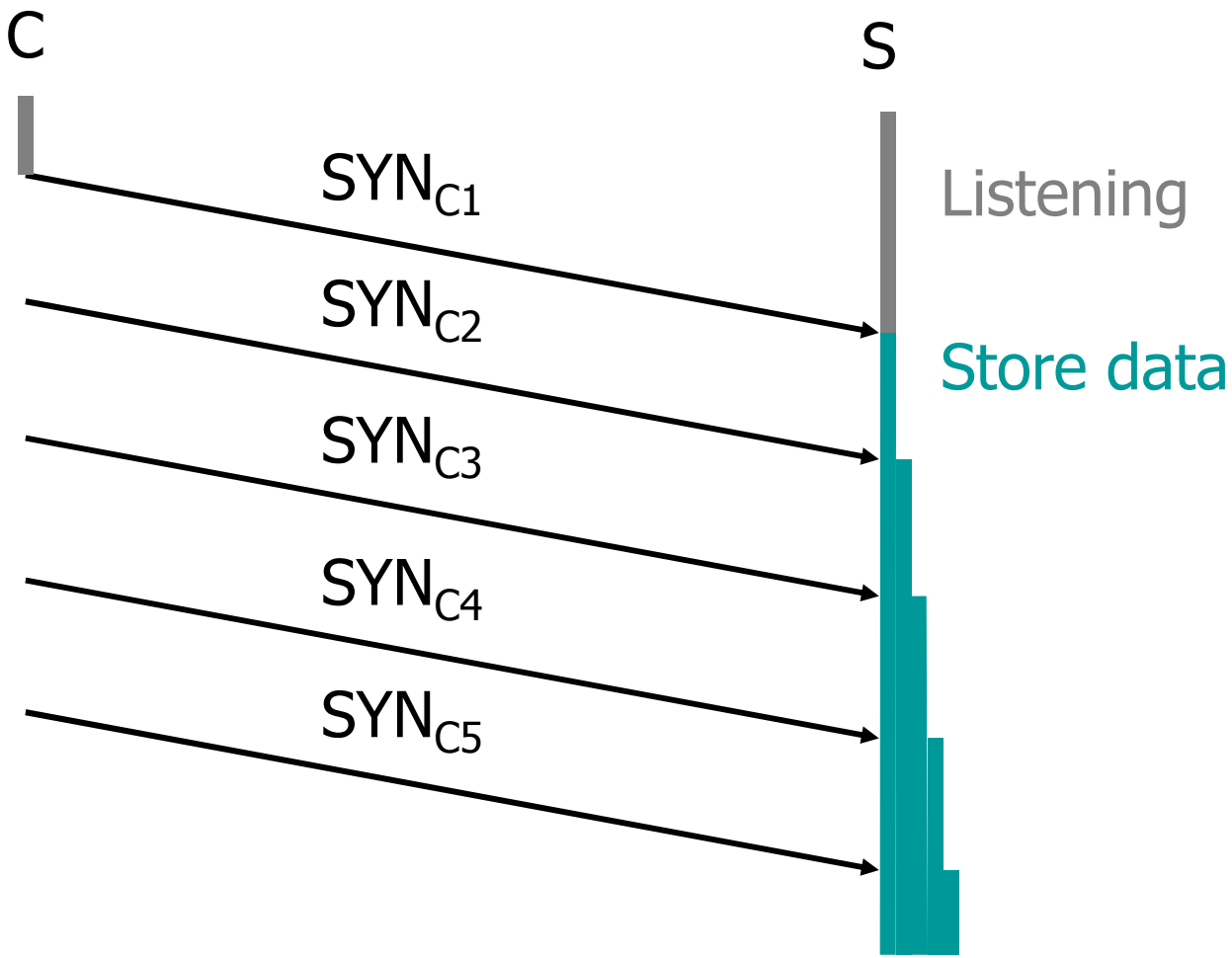
Step 2: server host receives SYN, replies with SYNACK segment
- server allocates buffers
- specifies server initial seq. #

Step 3: client receives SYNACK, replies with ACK segment, which may contain data

TCP Handshake



SYN Flooding



TCP Connection Management: Closing

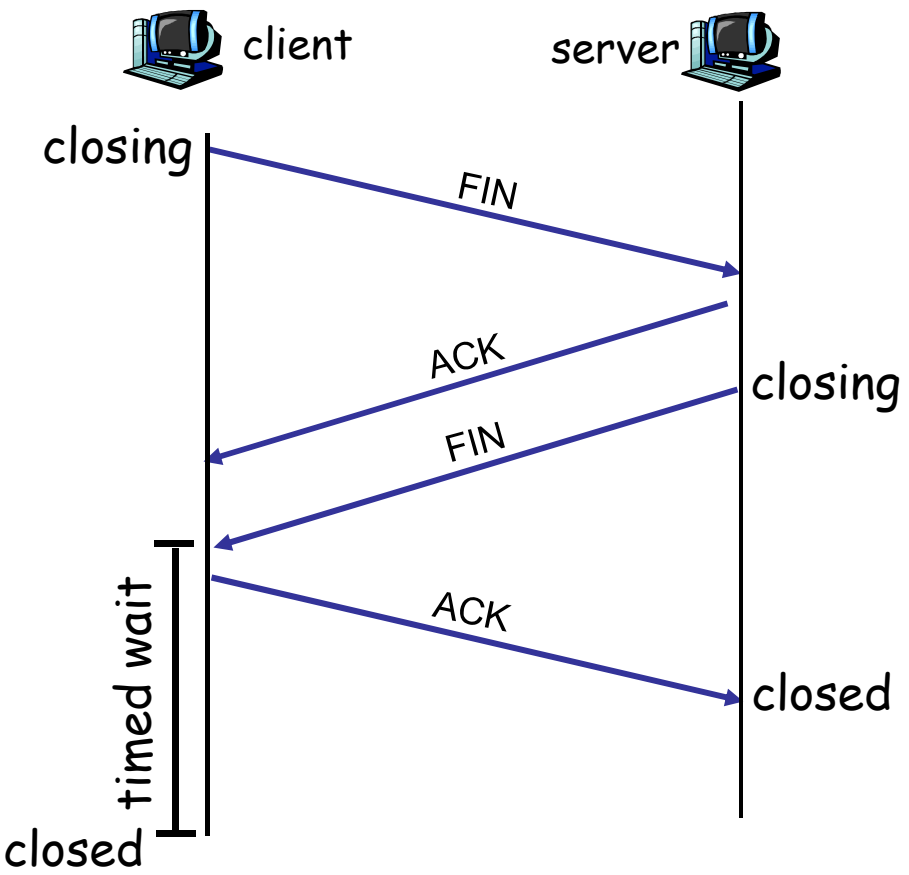
Step 1: client end system sends TCP FIN control segment to server

Step 2: server receives FIN, replies with ACK. Closes connection, sends FIN.

Step 3: client receives FIN, replies with ACK.

- Enters "timed wait" - will respond with ACK to received FINs

Step 4: server, receives ACK. Connection closed.



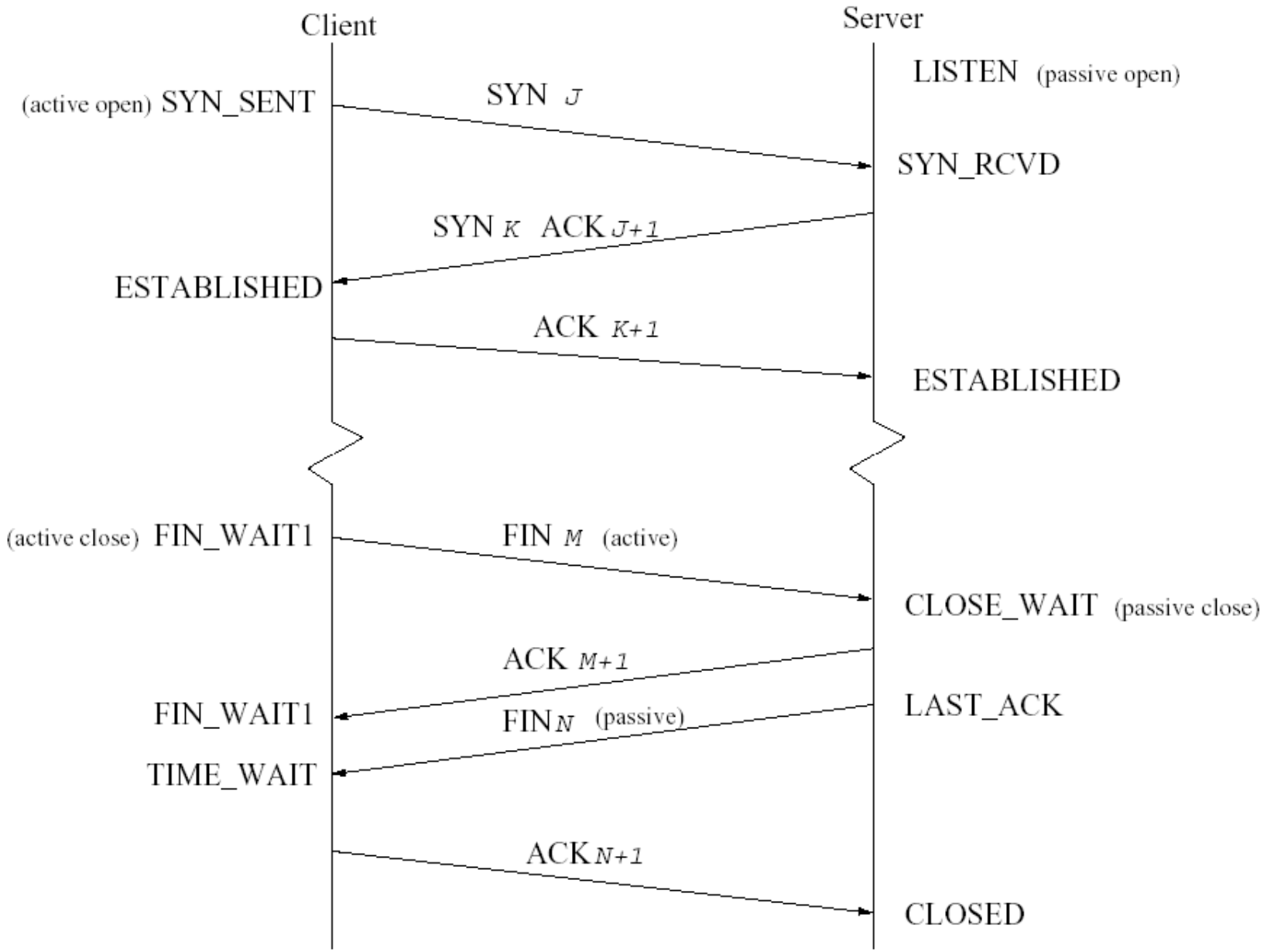
Flood Detection System on Router/Gateway

- Can we maintain states for each connection flow?
- Stateless, simple detection system on edge (leaf) routers desired
- Placement: First/last mile leaf routers
  - First mile – detect large DoS attacker
  - Last mile – detect DDoS attacks that first mile would miss

Detection Methods (I)

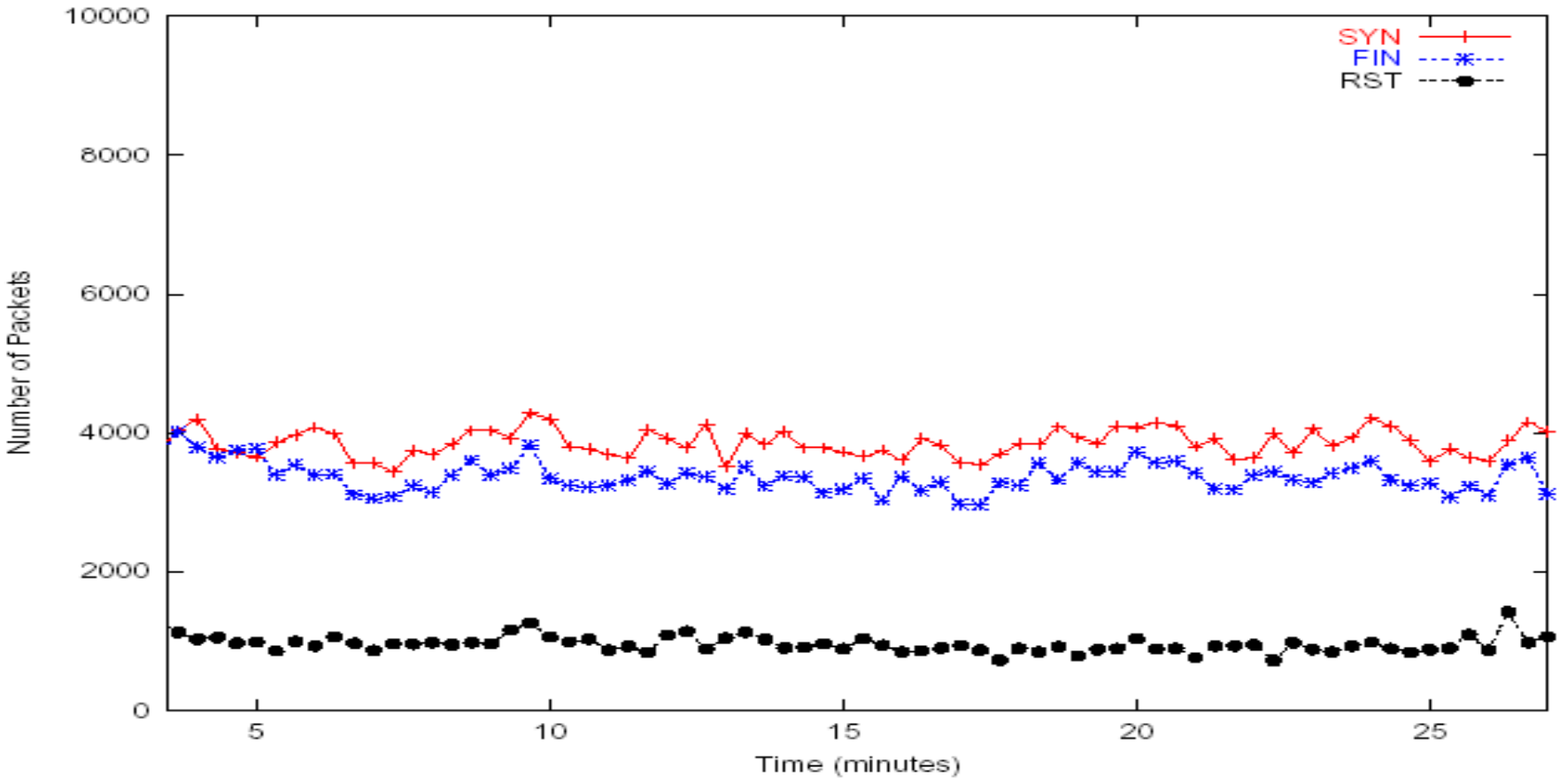
- Utilize SYN-FIN pair behavior
- OR SYNACK – FIN
- Can be both on client or server side
- However, RST violates SYN-FIN behavior
  - Passive RST: transmitted upon arrival of a packet at a closed port (usually by servers)
  - Active RST: initiated by the client to abort a TCP connection
  - So SYN-RST_{active} pair is also normal

SYN - FIN Behavior



SYN - FIN Behavior

- Generally every SYN has a FIN
- We can't tell if RST is active or passive
- Consider 75% active



Vulnerability of SYN-FIN Detection

- Send out extra FIN or RST with different IP/port as SYN
- Waste half of its bandwidth

Detection Method II

- SYN – SYN/ACK pair behavior
- Hard to evade for the attacking source
- Problems
  - Need to sniff both incoming and outgoing traffic
  - Only becomes obvious when really swamped

False Positive Possibilities

- Many new online users with long-lived TCP sessions
  - More SYNs coming in than FINs
- A major server is down which would result in 3 SYNs to a FIN or SYN-ACK
  - Because clients would retransmit the SYN