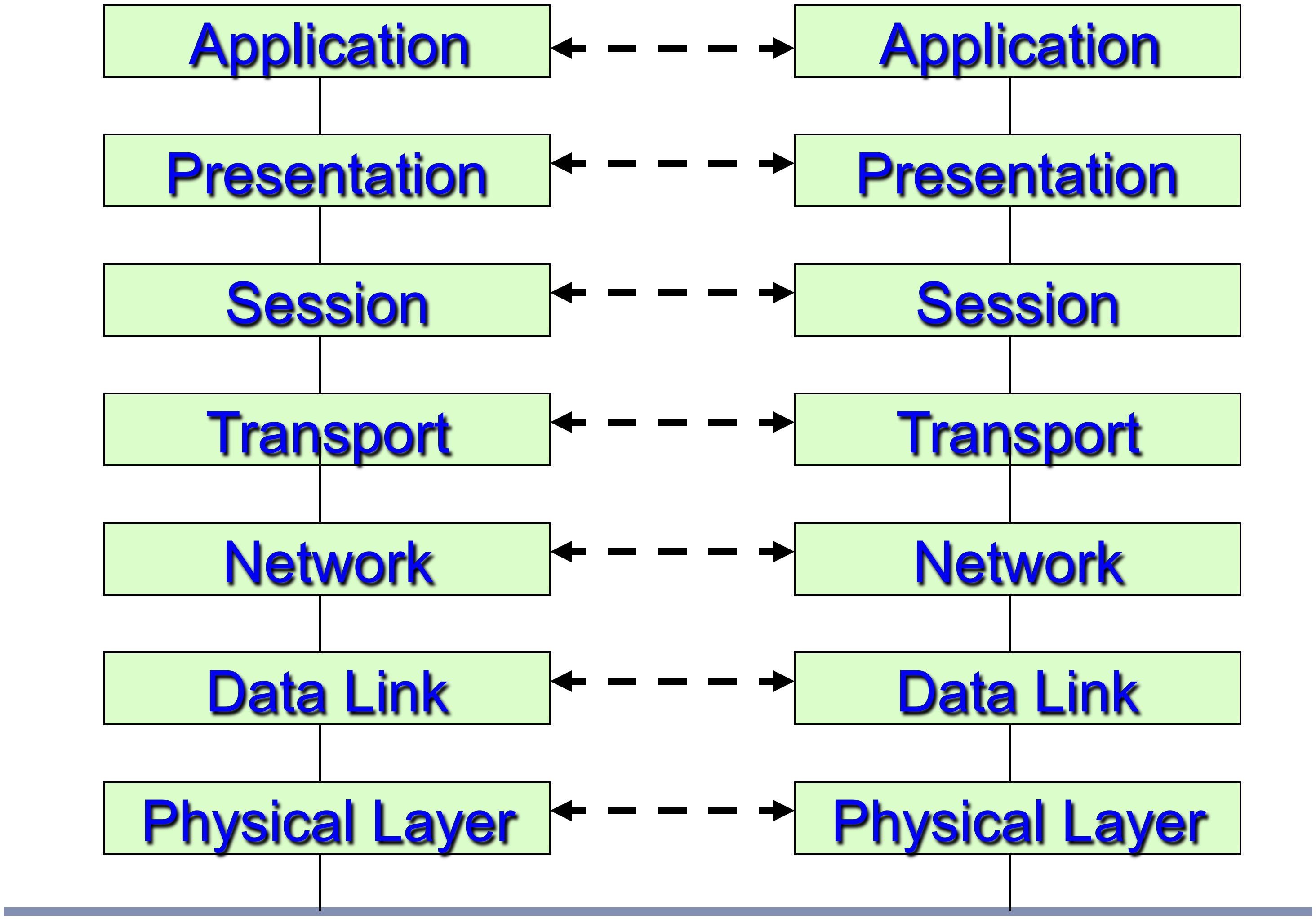
Cristina Nita-Rotaru

CS670: Network security

ARP, TCP

1: Background on network protocols

OSI/ISO Model



Application ↔ Application

Presentation ↔ Presentation

Session ↔ Session

Transport ↔ Transport

Network ↔ Network

Data Link ↔ Data Link

Physical Layer ↔ Physical Layer

A network of networks

- Internet is a "network of networks"
- Autonomous System (AS): a network, a single administrative domain, can span more organizations
- How do those networks connect
  - Internet Exchange (IX)
  - Network Access Points (NAP)
  - Metropolitan Area Exchange (MAE)

Support for network protocols

- Naming services
  - How are the entities that communicate identified: addresses, names
  - Types of entities: network cards, computers, applications, services
- Routing
  - How is the data forwarded from source to destination
    - Within an AS
    - Between ASs
- More specialized services: reliability, security, quality of service, congestion control, fairness, security

Identifiers

- Media Access Control (MAC) addresses in the network access layer
  - Associated with network interface card (NIC)
  - 48 bits or 64 bits
- IP addresses for the network layer
  - 32 bits for IPv4, and 128 bits for IPv6
  - E.g., 128.3.23.3
  - Static or dynamically allocated
- IP addresses + ports for the transport layer
  - E.g., 128.3.23.3:80
- Domain names for the application/human layer
  - E.g., www.neu.edu

Routing

- IP address identifies a computer
  - IPv4 - 32 bits, IPv6 - 128 bits
- Routing protocols: propagate information about routes to reach hosts (IP addresses) or networks (IP prefixes)
- Algorithms:
  - Distance vector protocols
  - Link-state protocols
  - Path vector protocols
- Relative to an AS
  - Inter-routing: RIP, OSPF
  - Intra-routing: BGP

Routing and translation of addresses

- Translation between IP addresses and MAC addresses
  - Address Resolution Protocol (ARP) for IPv4
  - Neighbor Discovery Protocol (NDP) for IPv6
- Routing with IP addresses
  - TCP, UDP, IP for routing packets, connections; IP communication between hosts, TCP and UDP between processes
  - Border Gateway Protocol for routing table updates
- Translation between IP addresses and domain names
  - Domain Name System (DNS)

NATs and their implications

- There are not enough IP addresses
- Solutions: IPv6 or ….Network Address Translation (NAT)
- NAT allows a single device, to act as an agent between the Internet (or "public network") and a local (or "private") network: only a single, unique IP address is required to represent an entire group of computers
- Computers can not communicate directly, STUN client-server protocol allows computers to discover each other behind a NAT (learn their public addresses), but requires presence of STUN server

DHCP

- Dynamic Host Configuration Protocol
  - Used for dynamic allocation of IP addresses
    - used for hosts that run only client applications
  - Allows for host-specific configuration parameters to be delivered from a DHCP server to a host
- DHCP can also be used to convey permanent IP address assignments to hosts
  - Server interfaces need permanent addresses because clients need to be able to reach them
  - Also, router interfaces should have permanent addresses for stability of routing data

Address Resolution Protocol (ARP)

- Interface between Link layer and Network Layer
- Allows hosts to query who owns an IP address on the same LAN
- Owner responds with hardware address
- Allows changes to link layer to be independent of IP addressing

Internet Protocol – IP

- IP is the current delivery protocol on the Internet, between hosts.
- IP provides ‘best effort’, unreliable delivery of packets.
  - IPv4 is the most used routing protocol on the Internet
  - IPv6, a newer version, still under adoption

  - https://www.google.com/intl/en/ipv6/statistics.html#tab=ipv6-adoption

User Datagram Protocol - UDP

- Runs on top of IP
- Connectionless protocol for a user process
- Minimal guarantee
  - No connection established
  - Unreliable transmission: no guarantee that the packets reach their destination
  - Error detection
  - No acknowledgment
  - No flow control

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
  - The packets are delivered in the same order in which they were sent.
  - Close the connection

Ports

- Remember:
  - Hardware addresses identity network cards
  - IP addresses identify hosts
  - Names identify hosts in a human friendly way
- However, transport protocols (TCP and UDP) ensure communication between processes.
- Computers differentiate what data is for which process through port numbers
  - 16-bit numbers

Ports contd.

- 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

IP Multicast

- Provides support for group communication: send to multiple parties
- Groups are specified by reserved IP multicast addresses 224.0.0.0 to 239.255.255.255.
- Unreliable communication
- IGMP is used to dynamically register individual hosts in a multicast group on a particular LAN.
- Network cards recognize IP multicast addresses: hosts that did not subscribe to a particular group will not process those packets (unlike broadcast that is processed by all hosts in a network segment)

Internet Control Message Protocol (ICMP)

- Provides feedback about network operation
  - Error reporting
  - Reachability testing
  - Congestion control
- Example message types
  - Destination unreachable
  - Time-to-live exceeded
  - Parameter problem
  - Redirect to better gateway
  - Echo/echo reply - reachability test

Distance-vector routing - RIP

- Each node:
  - Maintains a vector with distances to all of the nodes.
  - Sends periodically its distance-vector to all its neighbors.
  - Updates its distance vector based on the information received from the neighbors (shortest path Bellman-Ford): for each network path, the receiving routers pick the neighbor advertising the lowest cost, then add this entry into its routing table for re-advertisement.
- Example: RIP
  - Prevents routing loops by using a maximum number (15) f hops allowed in a path from the source to a destination.
  - Used UDP

Link-state routing - OSPF

- Each node:
  - Maintains global view of the network.
  - Sends periodically the current state of all links (link-state updates or advertisements) to all nodes (via flooding).
  - Notes the change and recompute its routes (use shortest-path – Dijkstra algorithm) to destination.
- Less bandwidth-intensive than Distance-Vector, but more complex and more computational and memory intensive.
- Example: OSPF
  - Uses link-state routing.
  - Uses IP directly

Path vectors - BGP

- Similar to distance vector protocols, but routing updates contain an ordered list of the path of traversed "nodes"
- Example: BGP
  - Updates contain an ordered list or AS path of traversed autonomous systems and a set of network prefixes belonging to the first AS in the list (UPDATE messages)
  - Each BGP router receives UPDATEs from its neighbors and selects best path for each prefix and reports that path to its neighbors (before that it has to withdraw the "old" path)
  - Selecting "best path": policies, local preference, shortest AS path, other metrics
  - Uses TCP to exchange routing updates

Domain Name System (DNS)

- Distributed, hierarchical database that maps host names with IP addresses
- ICANN oversees the domain name assignments
- Tree structure
  - Divided into zones
  - Delegating responsibilities
- Name servers
  - Authoritative information (hints to whom might be able to answer the request)
  - Cached data updated periodically
- Uses UDP

Relationships between protocols

- It is important to understand
  - how protocols depend on each other
  - what protection is provided for each protocol
- Examples
  - TCP relies on IP, protection at TCP layer will not impact data specific IP
  - BGP relies on TCP; vulnerabilities on TCP will impact BGP
  - Web applications rely on DNS; DNS redirection to malicious sites will affect those web applications

2: Basic attacks against network protocols

Basic attacks

- Basic attacks:
  - They are not protocol specific
  - They can be applied to any protocol
  - They can be composed into more complex and effective attacks
- Basic functionality of network protocols:
  - Data delivery
  - Data generation
  - Manage entities

Basic data delivery attacks

- Dropping packets to prevent them from arriving to destination
- Artificially delaying the delivery: time or number of hops
- Loading nodes with many request
- Filling links with traffic
- Connection-oriented protocols
  - Preventing connection establishment
  - Downgrading the promised quality of the delivery service: congestion control

Basic message content attacks

- Packet modification, an existing packet is intercepted and modified
- Packet injection, a new packet is created and injected
- Both happen because data is not authenticated and does not have integrity protection
  - Errors correction codes are not sufficient for protecting against modifications in adversarial networks – solution needs cryptographic MACs
- Some attacks require proximity and protocol knowledge
- Packet replay, an old packet stored and replayed later
  - Cryptography not sufficient to prevent replay attacks

Basic message content attacks

- Eavesdropping, lack of encryption makes data vulnerable to eavesdropping
- Man in the middle: attacker can intercept and decrypt communication without the two parties being aware
- Some networks more vulnerable than others
- TEMPEST: monitoring of electromagnetic radiation

Cache poisoning

- Open vs closed networks, and ability to check identities is critical as they are used to bootstrap all services
- Caching: store data for future used, usually used for identities to improve performance
- Poisoning of cache is common for protocols that do not use authentication and integrity
- Dangerous because malicious data in cache is used for longer time, not only one time

Challenge-Response Protocols

- Goal: one entity authenticates to other entity proving the knowledge of a secret, 'challenge'
- Time-variant parameters used to prevent replay, interleaving attacks, provide uniqueness and timeliness : nounce (used only once)
- Three types:
  - Random numbers
  - Sequences
  - Timestamp

Challenge-Response Protocols

- Random numbers:
  - pseudo-random numbers that are unpredictable to an adversary;
  - vulnerable to birthday attacks, use larger sample;
  - must maintain state;
  - do not prevent interleaving attacks (parallel sessions)
- Sequences:
  - serial number or counters;
  - long-term state information must be maintained by both parties+ synchronization
- Timestamp:
  - provides timeliness and detects forced delays;
  - requires synchronized clocks.

Challenge-Response Examples

- unilateral authentication with timestamp

  $A \rightarrow B$: $cert_A, t_A, B, S_A(t_A, B)$

- unilateral authentication with random numbers

  $A \leftarrow B$: r_B

  $A \rightarrow B$: $cert_A, r_A, B, S_A(r_A, r_B, B)$

- mutual authentication with random numbers

  $A \leftarrow B$: r_B

  $A \rightarrow B$: $cert_A, r_A, B, S_A(r_A, r_B, B)$

  $A \leftarrow B$: $cert_B, A, S_B(r_B, r_A, A)$

Attacks: Examples

- E1: "Man-in-the-middle" attack on unauthenticated DH
- E2: Reflection attack

Protocol

$A \rightarrow B : r_A$

$B \rightarrow A : E_k(r_A, r_B)$

$A \rightarrow B : r_B$

Attack

(1) $A \rightarrow E : r_A$ A thinks she asks B to authenticate to A

(2) $E \rightarrow A : r_A$: E starts a new session,

(3) $A \rightarrow E : E_k(r_A, r_A')$: Reply of (2),

(4) $E \rightarrow A : E_k(r_A, r_A')$: Reply of (1), E makes A believe she's B

(5) $A \rightarrow E : r_A'$, A finishes the (1) protocol

Attacks: Examples (cont.)

- E3: Interleaving attacks

Protocol

$A \rightarrow B : r_A$

$B \rightarrow A : r_B, S_B(r_B, r_A, A)$

$A \rightarrow B : r_A', S_A(r_A', r_B, B)$

Attack

$E \rightarrow B : r_A$

$B \rightarrow E : r_B, S_B(r_B, r_A, A)$

$E \rightarrow A : r_B$

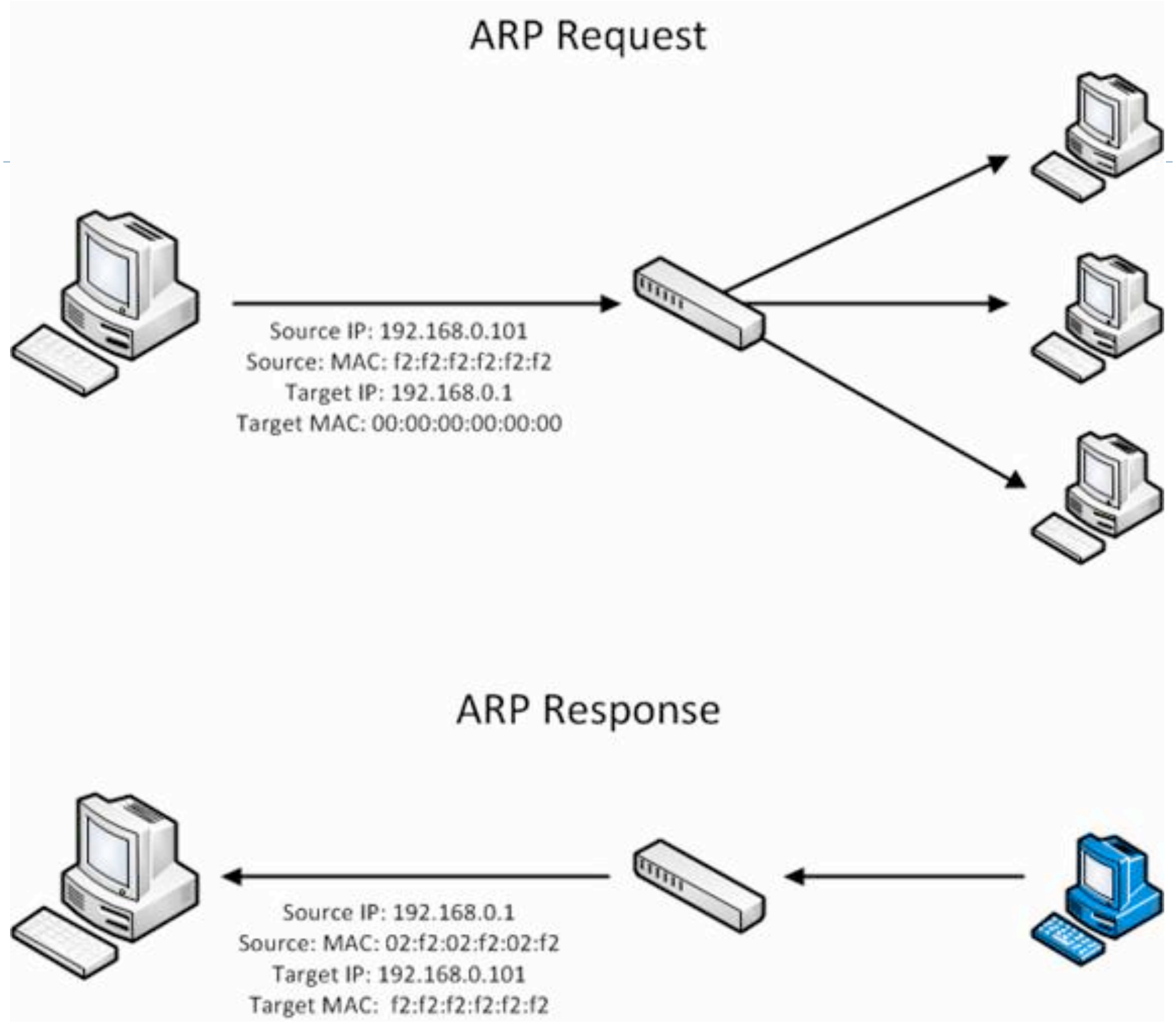
$A \rightarrow E : r_A', S_A(r_A', r_B, B)$

$E \rightarrow B : r_A', S_A(r_A', r_B, B)$

3: Attacks against ARP

Address Resolution Protocol (ARP)

- Primarily used to translate IP addresses to Ethernet MAC addresses
  - The device drive for Ethernet NIC needs ARP to send a packet
- Also used for IP over other LAN technologies, e.g. IEEE 802.11
- Each host maintains a table of IP to MAC addresses
- Message types:
  - ARP request
  - ARP reply
  - ARP announcement



ARP Request

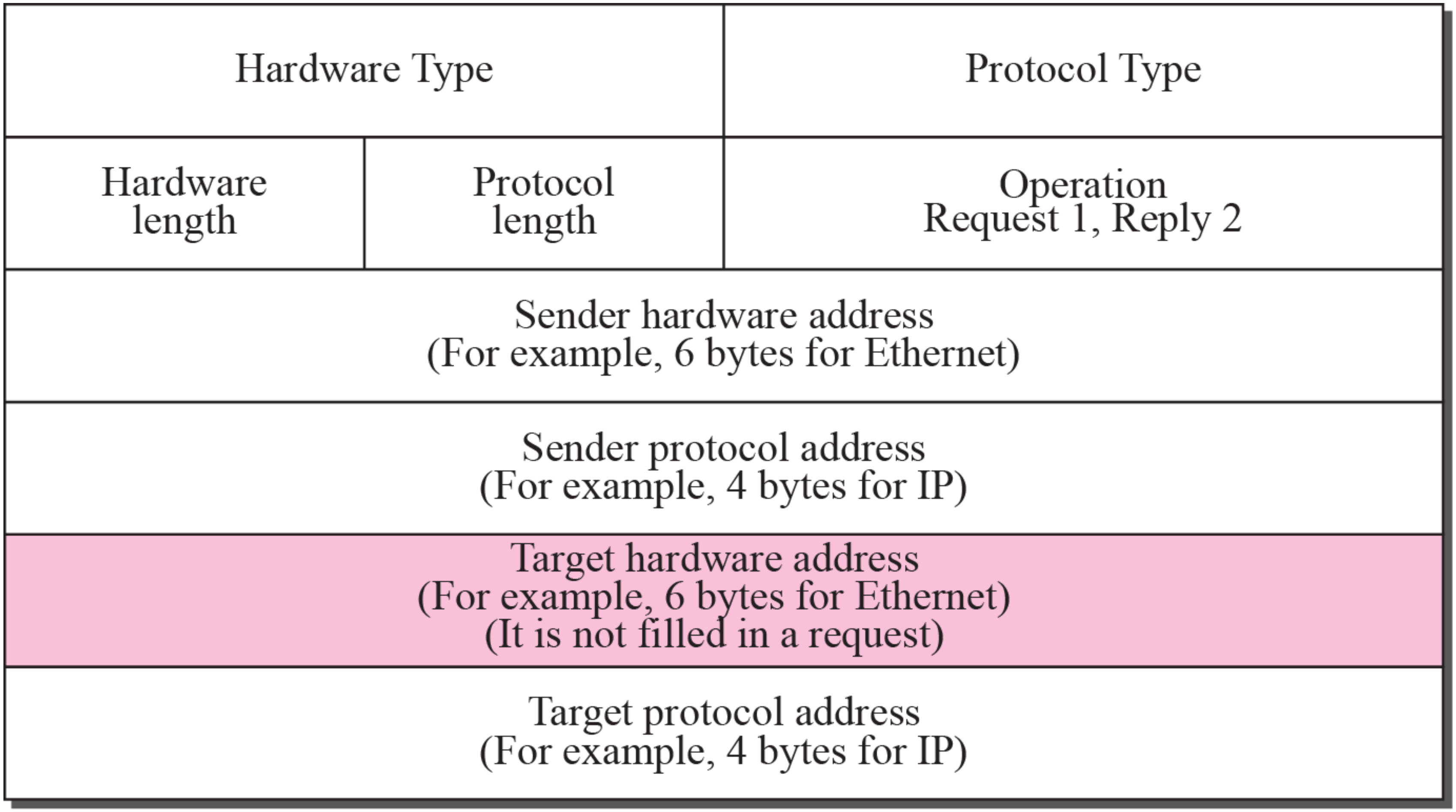
Source IP: 192.168.0.101
Source: MAC: f2:f2:f2:f2:f2:f2
Target IP: 192.168.0.1
Target MAC: 00:00:00:00:00:00

ARP Response

Source IP: 192.168.0.1
Source: MAC: 02:f2:02:f2:02:f2
Target IP: 192.168.0.101
Target MAC: f2:f2:f2:f2:f2:f2

http://www.windowsecurity.com

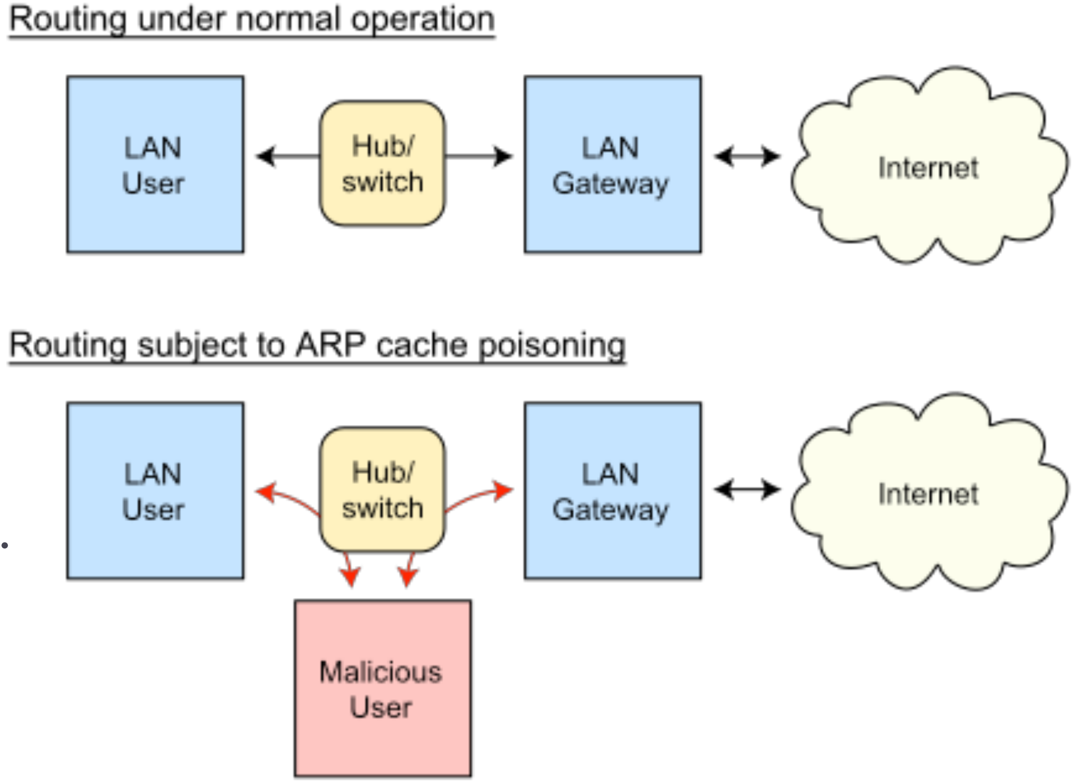
ARP packet



<table>
<tr><td colspan="2">Hardware Type</td><td colspan="2">Protocol Type</td></tr>
<tr><td>Hardware length</td><td>Protocol length</td><td colspan="2">Operation
Request 1, Reply 2</td></tr>
<tr><td colspan="4">Sender hardware address
(For example, 6 bytes for Ethernet)</td></tr>
<tr><td colspan="4">Sender protocol address
(For example, 4 bytes for IP)</td></tr>
<tr><td colspan="4">Target hardware address
(For example, 6 bytes for Ethernet)
(It is not filled in a request)</td></tr>
<tr><td colspan="4">Target protocol address
(For example, 4 bytes for IP)</td></tr>
</table>

ARP spoofing (ARP Poisoning)

- Send fake or 'spoofed' ARP messages to an Ethernet LAN
  - To have other machines associate honest IP addresses with the attacker's MAC
- Legitimate use
  - Redirect a user to a registration page before allow usage of the network.
  - Implementing redundancy and fault tolerance



ARP spoofing defenses

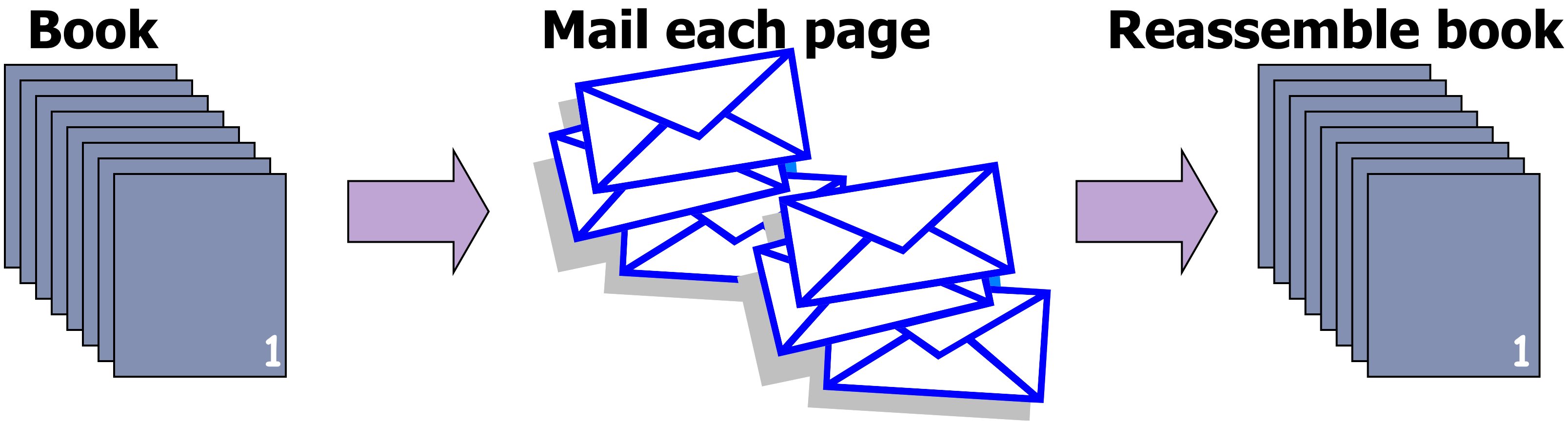
- Static ARP table
  - Effective for small and fixed size networks
- DHCP Certification (use access control to ensure that hosts only use the IP addresses assigned to them, and that only authorized DHCP servers are accessible).
- Monitoring and detection of changes: Arpwatch (can also send email when updates occur)
- Some switches can block gratuitous ARP replies

4: Attacks against TCP

Security problems in the TCP/IP Protocol Suite.
S.M. Bellovin

Transmission Control Protocol - TCP

- Connection oriented protocol for a user process: establishes a connection (channel) between two end-points
  - Reliable, full-duplex channel: acknowledgements, retransmissions, timeouts
  - Messages broken in packets
  - Congestion control mechanisms
  - Packets are delivered in the same order in which they were sent

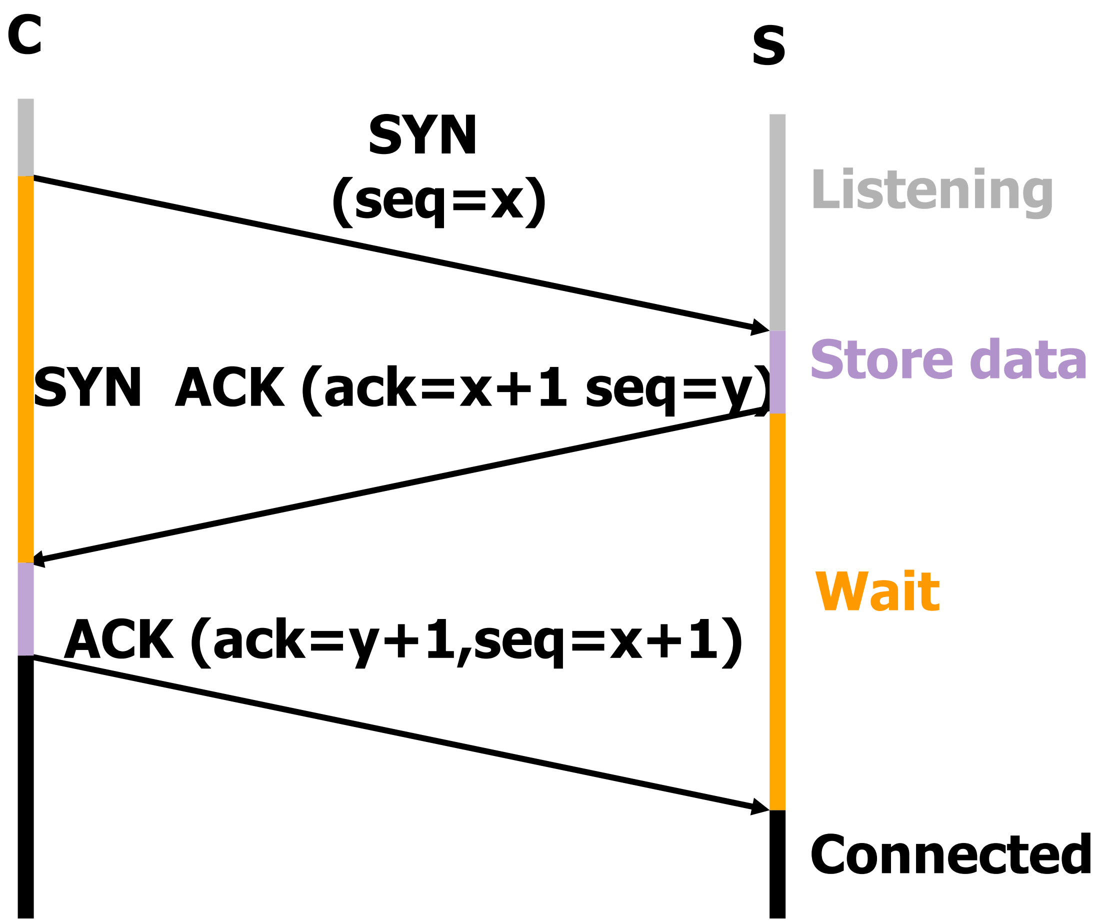


Book

Mail each page

Reassemble book

TCP handshake



- Resources allocated; There is a max. number of connections that can be in this state (SYN_RECVD state)
- Wait for the ACK (75 seconds)
- If timeout expires or RST received, data deallocated
- If ACK received, connection established, can also contain data.

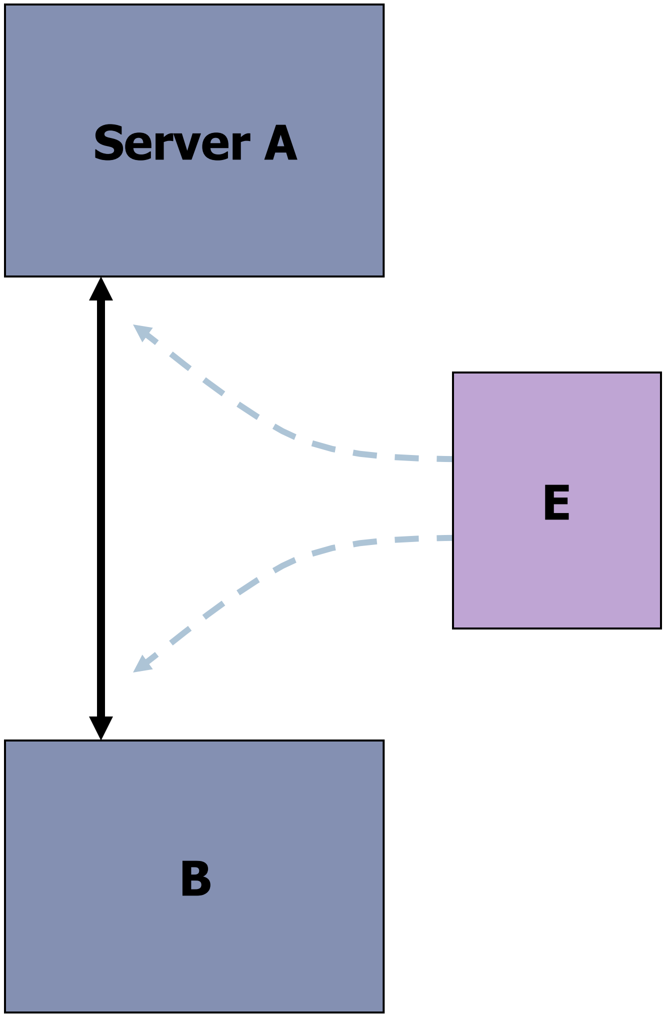
TCP sequence numbers

- Sequence number (32 bits) – has a dual role:
  - If the SYN flag is set, then this is the initial sequence number. The sequence number of the actual first data byte is this sequence number plus 1.
  - If the SYN flag is clear, then this is the accumulated sequence number of the first data byte of this packet for the current session.
- Acknowledgment number (32 bits) –
  - If the ACK flag is set then this the next sequence number that the receiver is expecting.
  - This acknowledges receipt of all prior bytes (if any).

TCP sequence prediction attack

- Predict the sequence number used to identify the packets in a TCP connection, and then counterfeit packets.
- Adversary: do not have full control over the network, but can inject packets with fake source IP addresses
  - E.g., control a computer on the local network
- TCP sequence numbers are used for authenticating packets
- Initial seq# needs high degree of unpredictability
  - If attacker knows initial seq # and amount of traffic sent, can estimate likely current values
  - Some implementations are vulnerable

Blind TCP session hijacking



- A, B trusted connection
  - Send packets with predictable seq numbers
- E impersonates B to A
  - Opens connection to A to get initial seq number
  - DoS B's queue
  - Sends packets to A that resemble B's transmission
  - E cannot receive, but may execute commands on A

Risks from session hijacking

- Inject data into an unencrypted server-to-server traffic, such as an e-mail exchange, DNS zone transfers, etc.
- Inject data into an unencrypted client-to-server traffic, such as ftp file downloads, http responses.
- Spoof IP addresses, which are often used for preliminary checks on firewalls or at the service level.
- Carry out MITM attacks on weak cryptographic protocols.
  - often result in warnings to users that get ignored
- Denial of service attacks, such as resetting the connection.

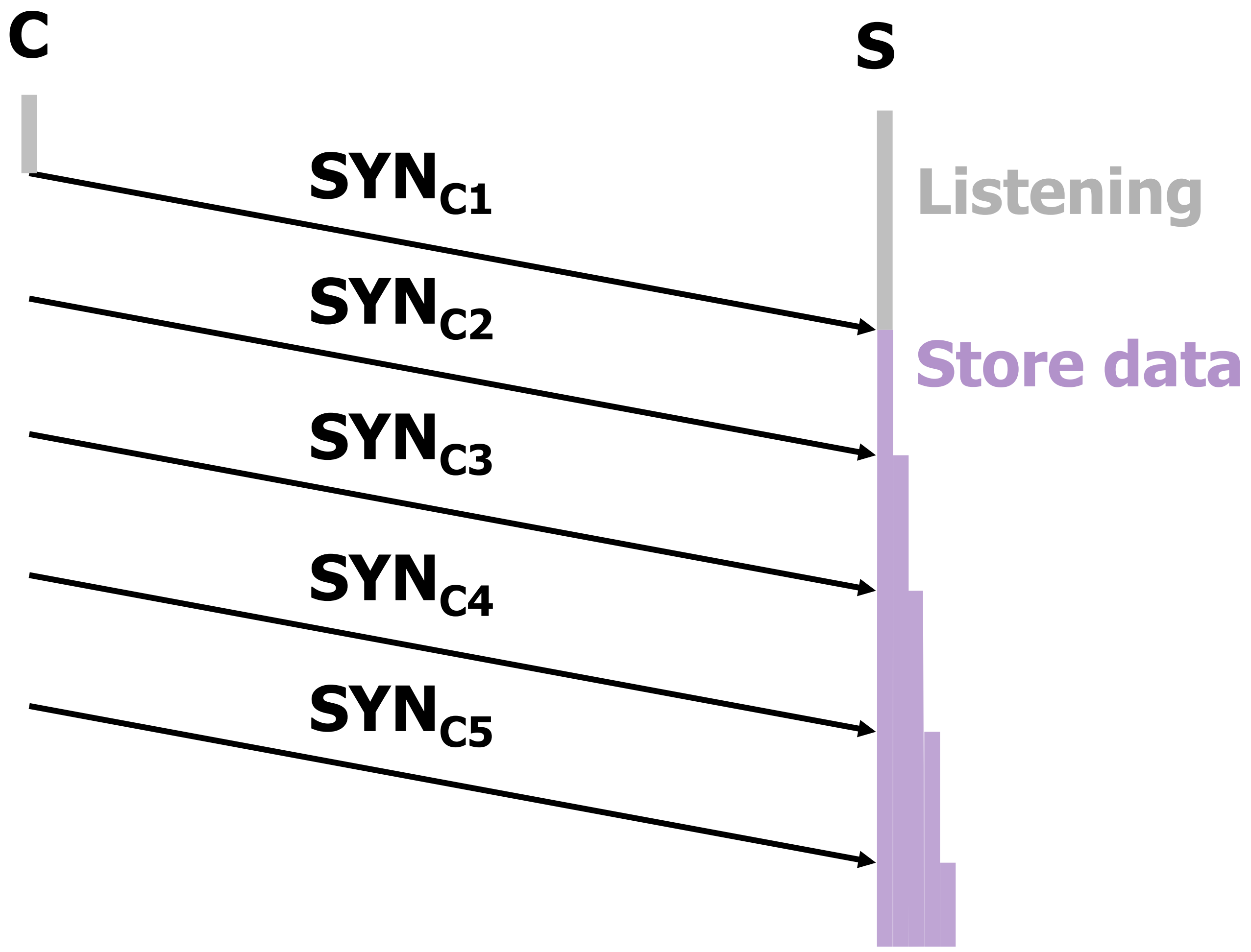
DoS vulnerability caused by session hijacking

- Suppose attacker can guess seq. number for an existing connection:
  - Attacker can send Reset packet to close connection. Results in DoS.
  - Naively, success prob. is 1/232 (32-bit seq. #'s).
  - Most systems allow for a large window of acceptable seq. #'s
    - Much higher success probability.
- Attack is most effective against long lived connections, e.g. BGP.

SYN flooding 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).

SYN flooding



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

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 outside coming packets that have source addresses from the internal network
  - Block packets to the outside that have source addresses from outside the internal network

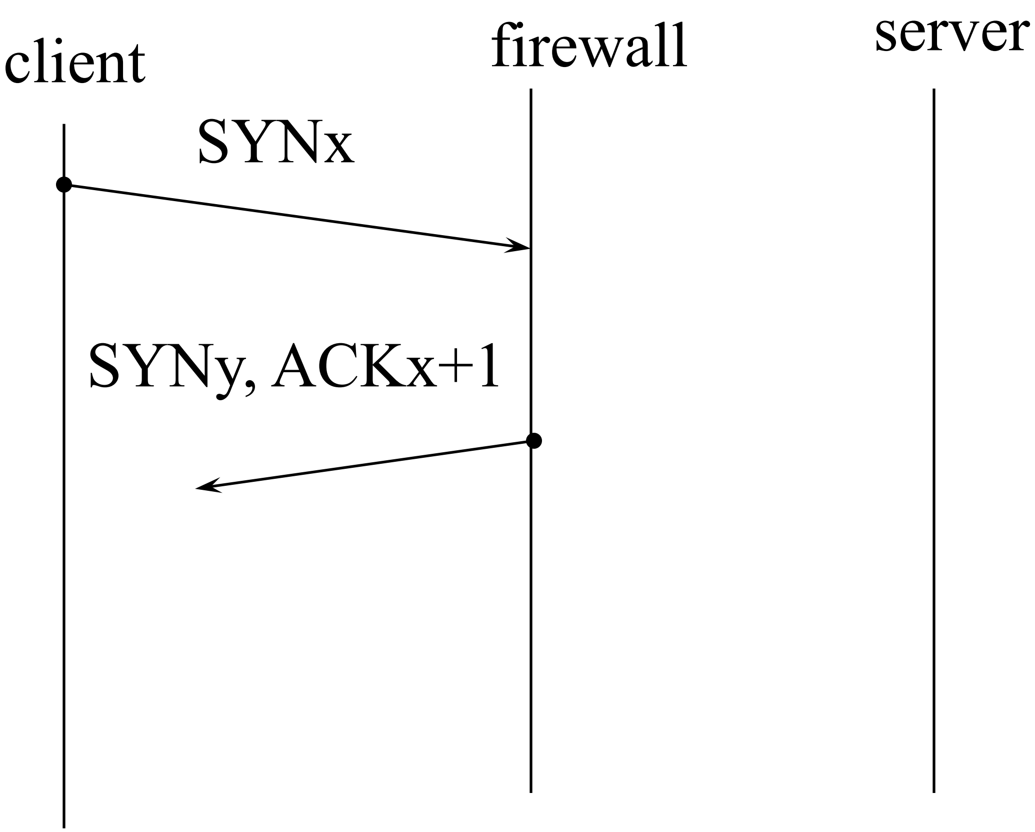
Infrastructure improvements

- 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.

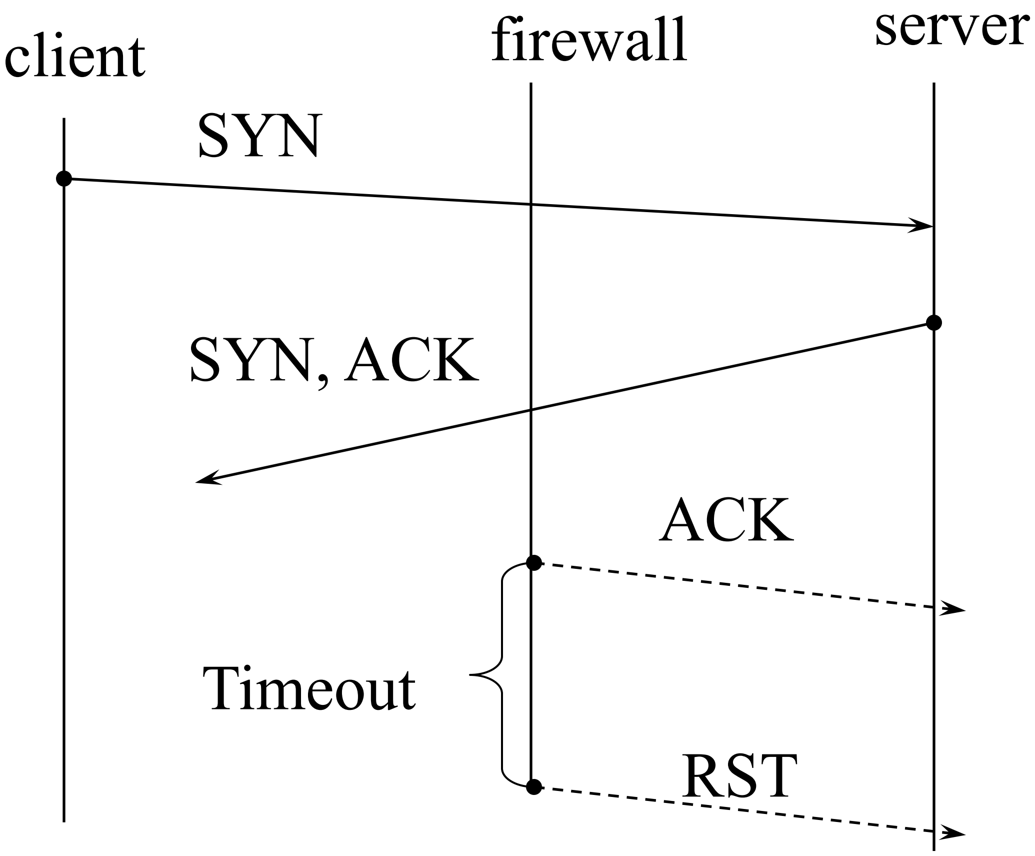
Firewall approach

- Main idea: each packet for inside network if 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 connections.
- 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

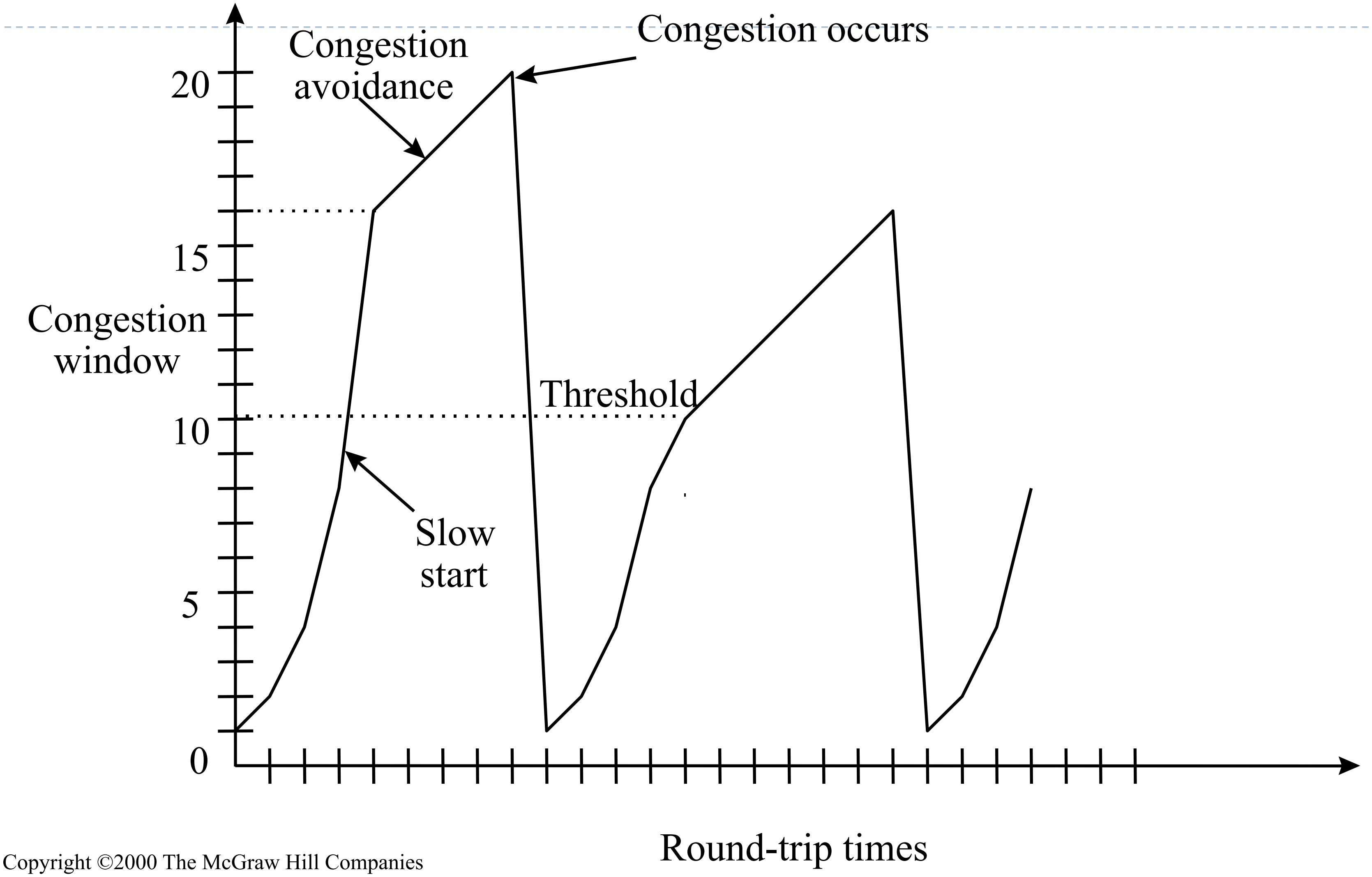
So far…

- Attacks require high-rate transmission (flood of SYN packets), unusual network traffic, attackers are relatively easy to detect and filter.
- However ….TCP can be attacked by using TCP friendly traffic (exploit congestion control mechanism), low rate, therefore it can cause significant 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

TCP congestion control

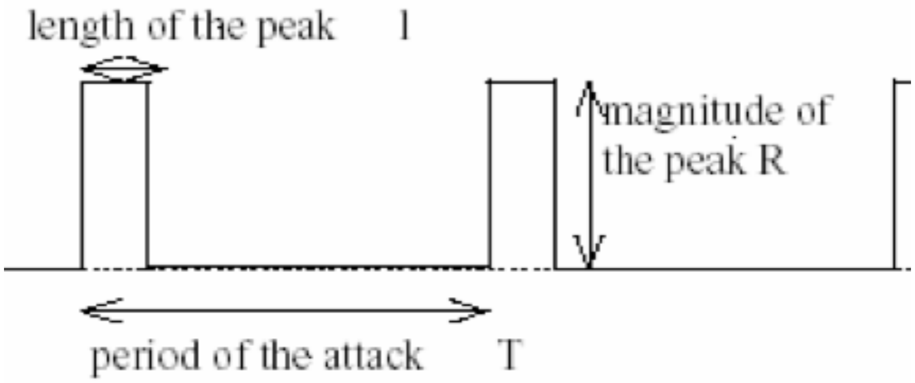


Leon-Garcia & Widjaja: *Communication Networks*

Figure 7.63

Attack against congestion control

- 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

Defenses

- 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

Take home lessons

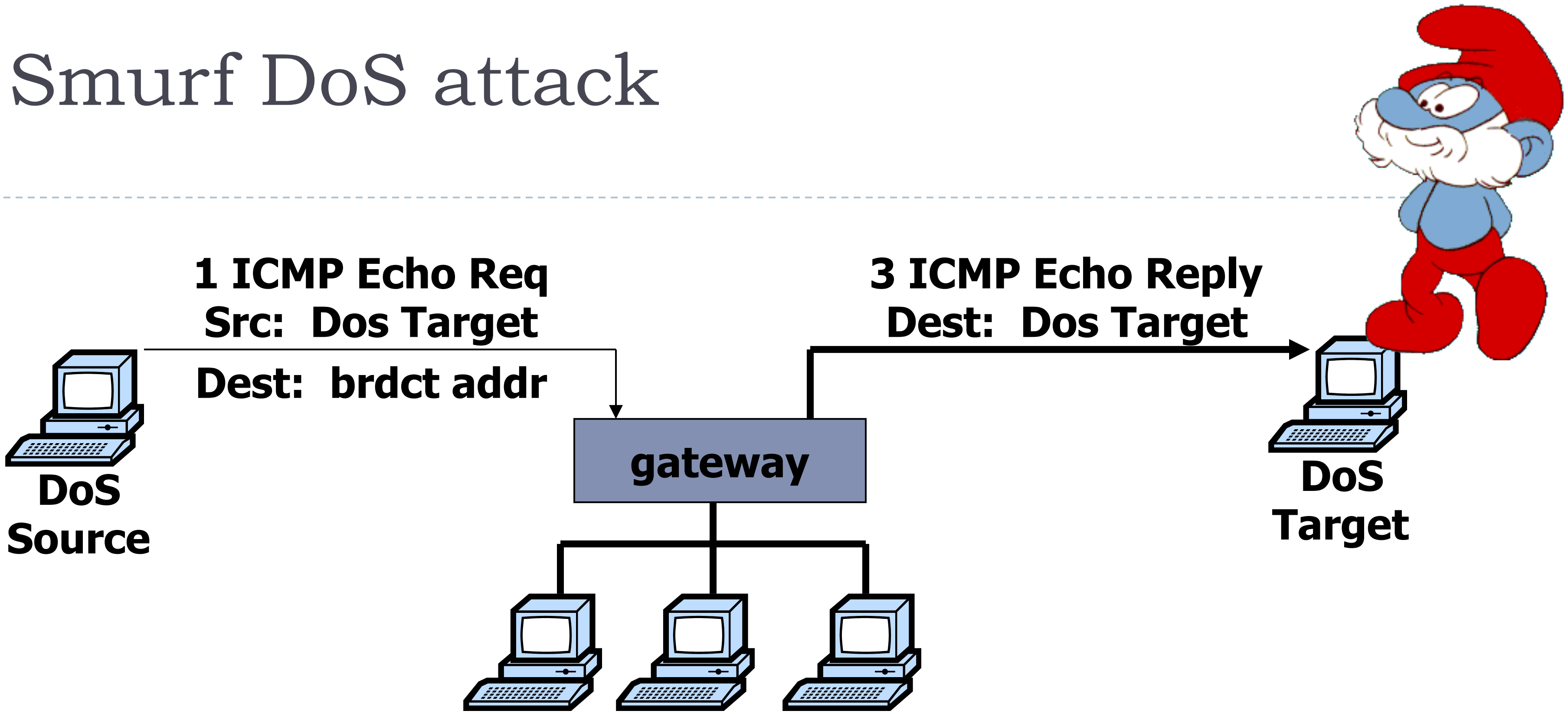
- Connection hijacking
  - Sequence prediction
- Denial of service against server
  - SYN-flooding attacks
- Attacks leveraging the congestion control mechanism

5: Attacks against ICMP and DDOS

Internet Control Message Protocol (ICMP)

- Provides feedback about network operation
  - Error reporting
  - Reachability testing
  - Congestion Control
- Example message types
  - Destination unreachable
  - Time-to-live exceeded
  - Parameter problem
  - Redirect to better gateway
  - Echo/echo reply - reachability test

Smurf DoS attack



1 ICMP Echo Req
Src: Dos Target
Dest: brdct addr

3 ICMP Echo Reply
Dest: Dos Target

gateway

DoS
Source

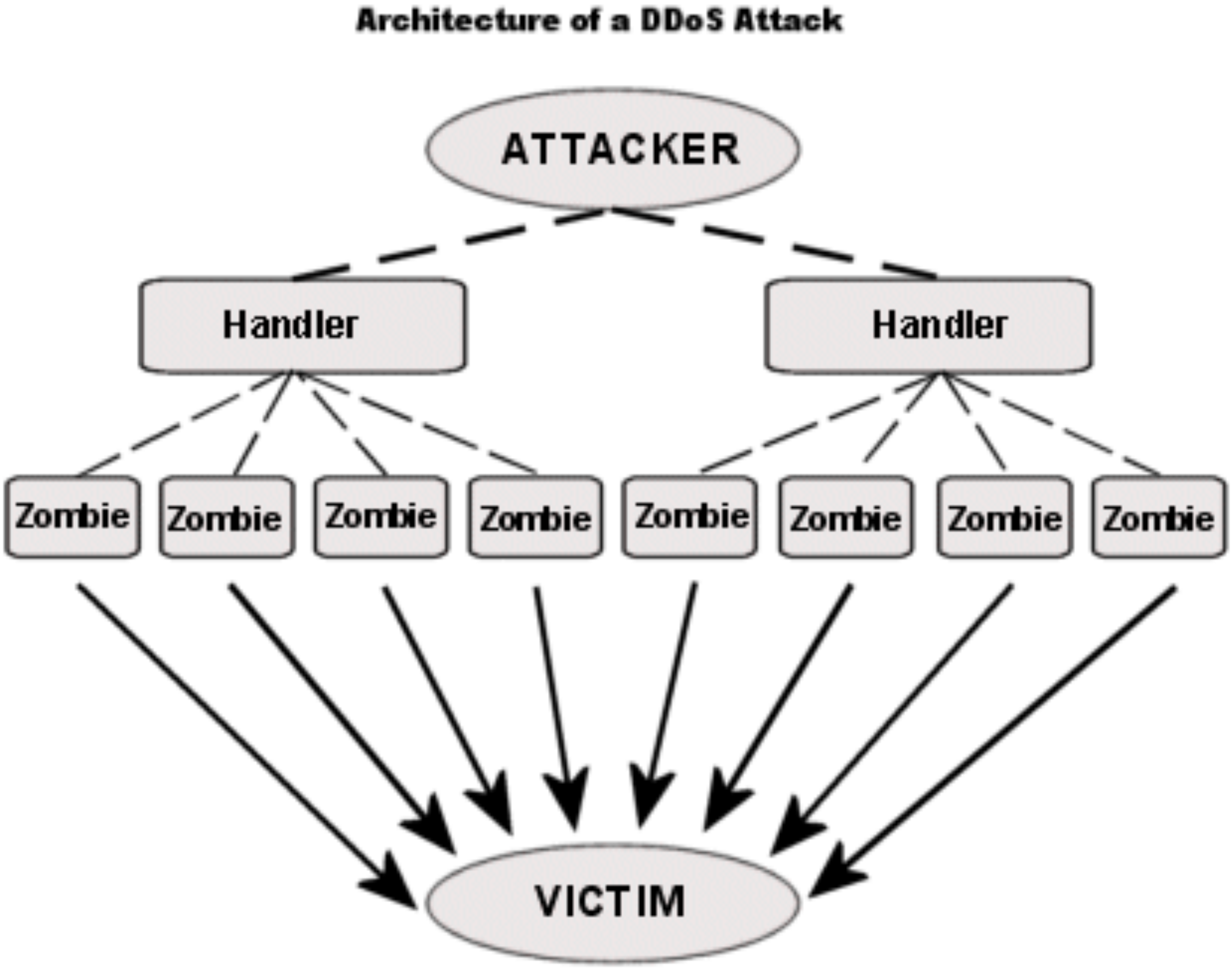
DoS
Target

- Send ping request to broadcast 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

Defense

- Configure individual hosts and routers to not respond to ICMP requests or broadcasts;
- Configure routers to not forward packets directed to broadcast addresses.
- Network ingress filtering, which rejects the attacking packets on the basis of the forged source address.

Distributed DoS (DDoS)



Hiding DDoS attacks

- Reflection
  - Find big sites with lots of resources, send packets with spoofed source address, response to victim
    - PING => PING response
    - SYN => SYN-ACK
- Pulsing zombie floods
  - each zombie active briefly, then goes dormant;
  - zombies taking turns attacking
  - making tracing difficult

Cryptographic network protection

- Solutions above the transport layer
  - Examples: SSL and SSH
  - Protect against session hijacking and injected data
  - Do not protect against denial-of-service attacks caused by spoofed packets
- Solutions at network layer
  - Use cryptographically random ISNs [RFC 1948]
  - More generally: IPsec
  - Can protect against
    - session hijacking and injection of data.
    - denial-of-service attacks using session resets.