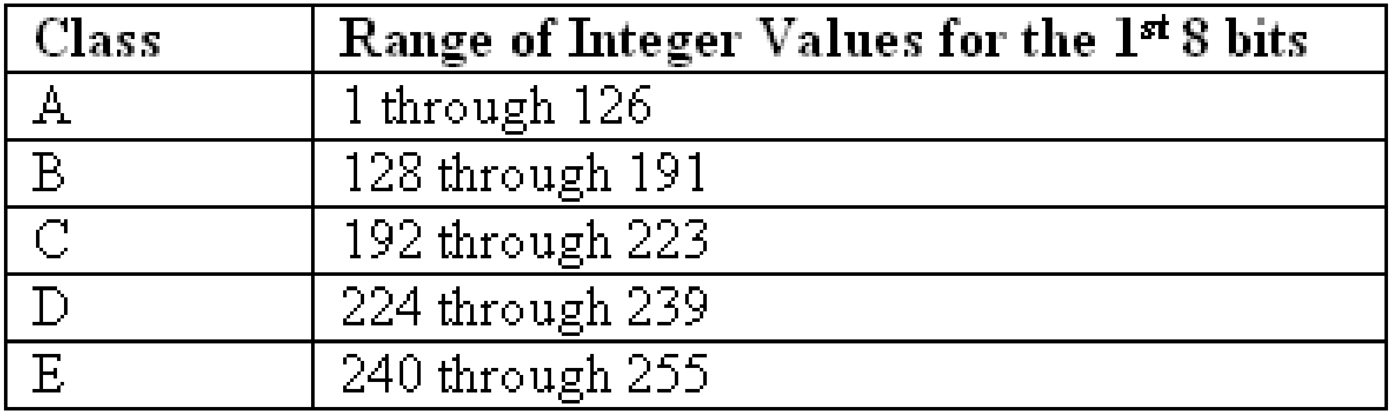
Denial of Service Attacks

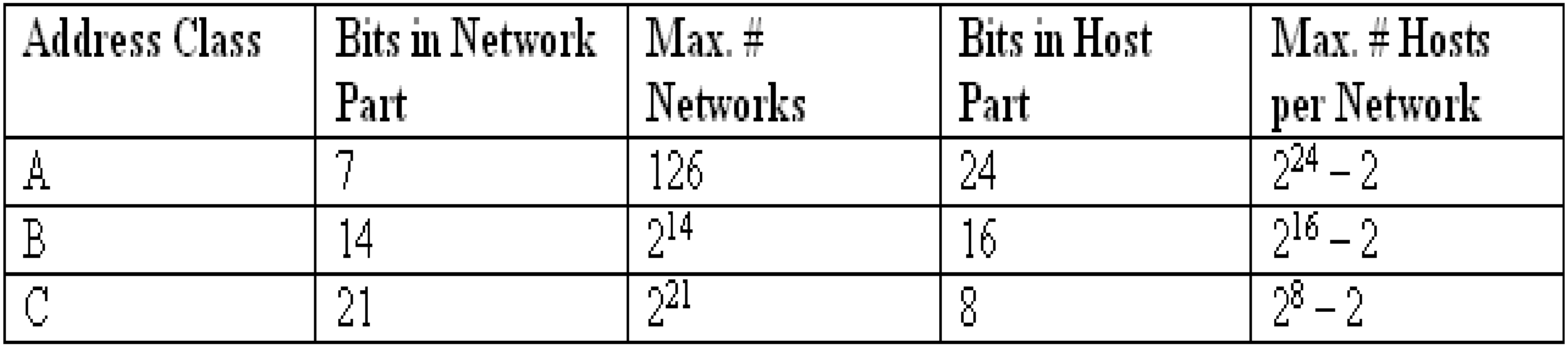
Dr. Natarajan Meghanathan
Associate Professor of Computer Science
Jackson State University
E-mail: natarajan.meghanathan@jsums.edu

Networks Primer

Range of IP Addresses and Network Prefixes for Different Classes



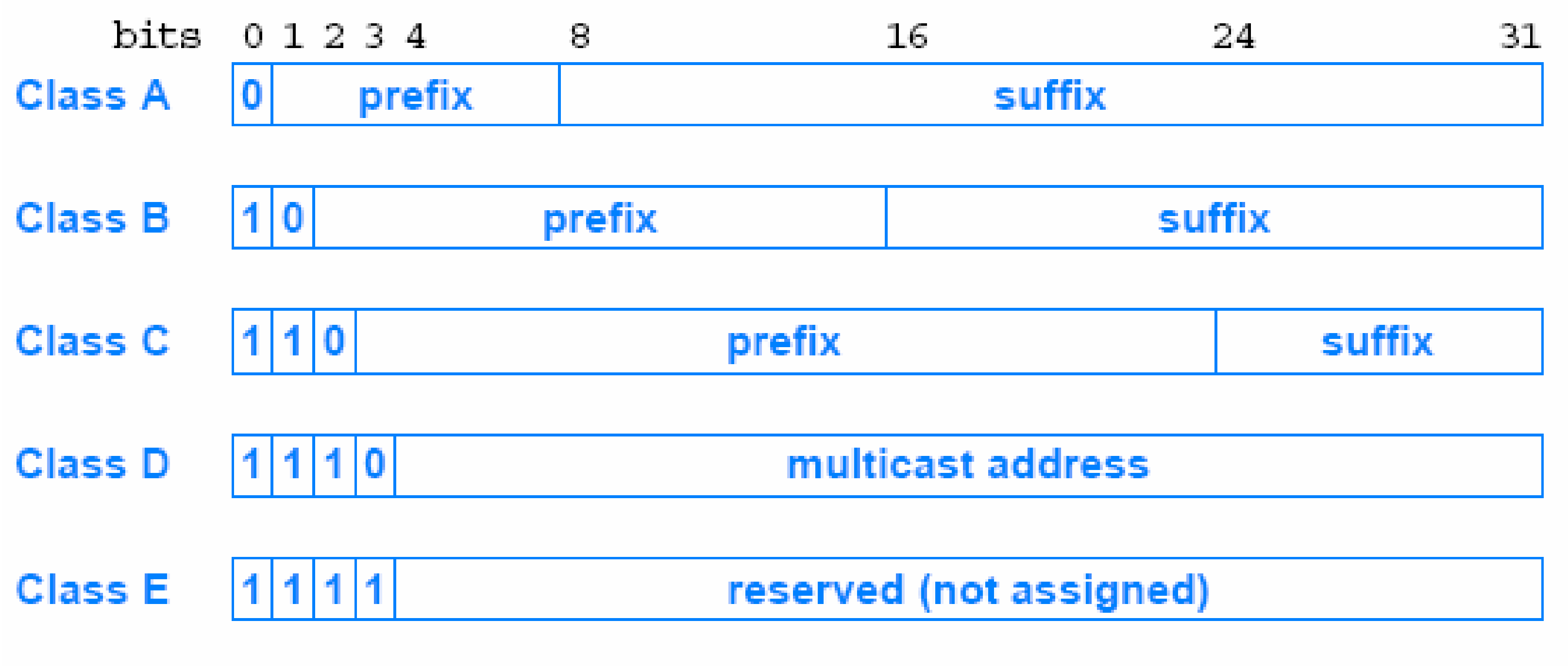
Class	Range of Integer Values for the 1st 8 bits
A	1 through 126
B	128 through 191
C	192 through 223
D	224 through 239
E	240 through 255



Address Class	Bits in Network Part	Max. # Networks	Bits in Host Part	Max. # Hosts per Network
A	7	126	24	$2^{24} - 2$
B	14	2^{14}	16	$2^{16} - 2$
C	21	2^{21}	8	$2^{8} - 2$

Classes of IP Address

- Internet contains a few large physical networks and many small networks
- The original classful IP addressing divided the IP address space into three (3) primary classes
  - each class has a different size prefix and suffix
- The first four bits of an IP address determined the class to which the address belonged



Private IP Addresses

- IANA reserves certain blocks of IP addresses (called private IP address) for use by the private internets. The **private ip address blocks are:**

  10.0.0.0 - 10.255.255.255
  172.16.0.0 - 172.31.255.255
  192.168.0.0 - 192.168.255.255

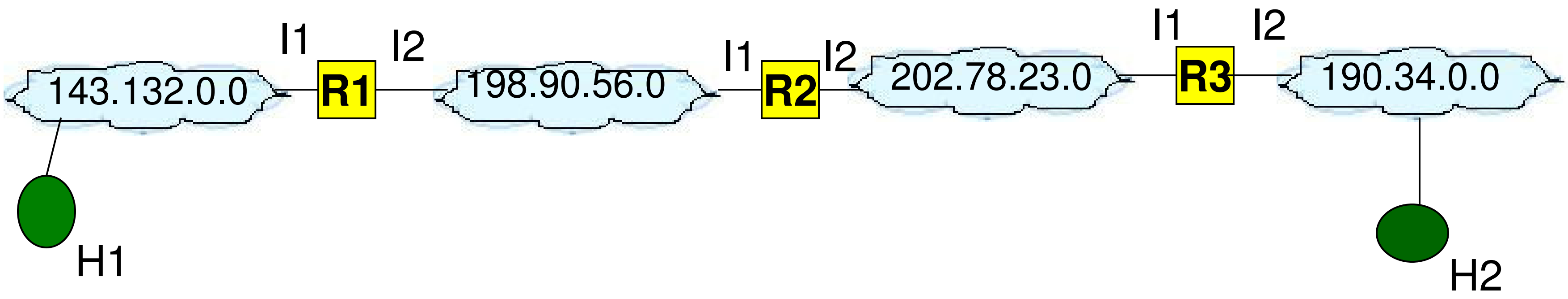
- The same set of private IP addresses can be used at different organizations (i.e., a private IP address has to be only locally unique); where as a public IP address (all IP addresses other than the above blocks of private IP addresses) has to be globally unique.
- Private IP addressing is one of the solutions to reduce the exhaustion of IP address space.
- **The private ip addresses are not routable in the public internet** (i.e., packets bearing private ip addresses are not forwarded by routers in the Internet). We need to through a public gateway and use its IP address.

- For networks connected to the public internet, the service provider makes the class of IP address to be assigned to an organization's network; where as in a private internet, the local administrator selects the class.

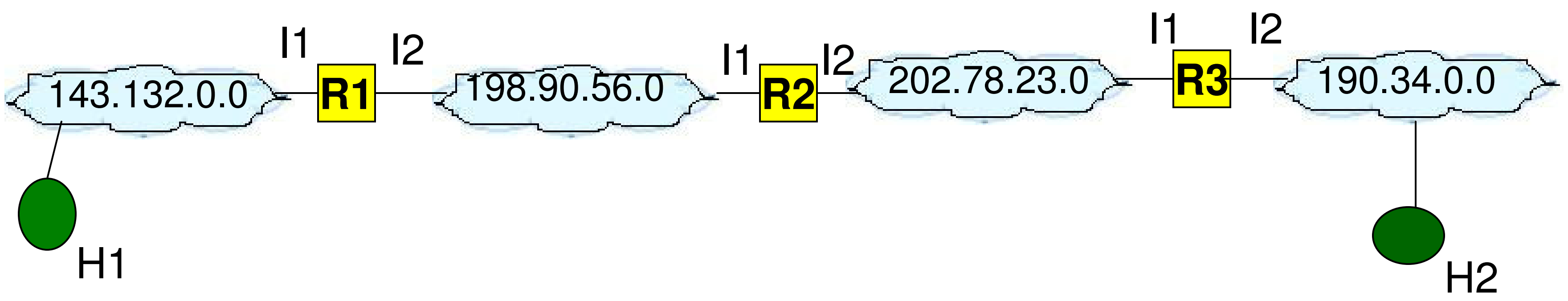
Examples: IP Addresses

- Identify whether the following is a network address, broadcast IP address, unicast IP address, multicast IP address or a private IP address:
  - ❖ a) 143.132.10.1 – unicast IP address for a class B network
  - ❖ b) 229.0.1.2 – multicast IP address
  - ❖ c) 16.1.255.255 – unicast IP address for a class A network
  - ❖ d) 10.1.1.1 – private IP address
  - ❖ e) 172.18.12.34 – private IP address
  - ❖ f) 202.14.12.255 – broadcast IP address for a class C network
  - ❖ g) 156.25.32.0 – unicast IP address for a class B network
  - ❖ h) 202.45.69.0 – network address for a class C network

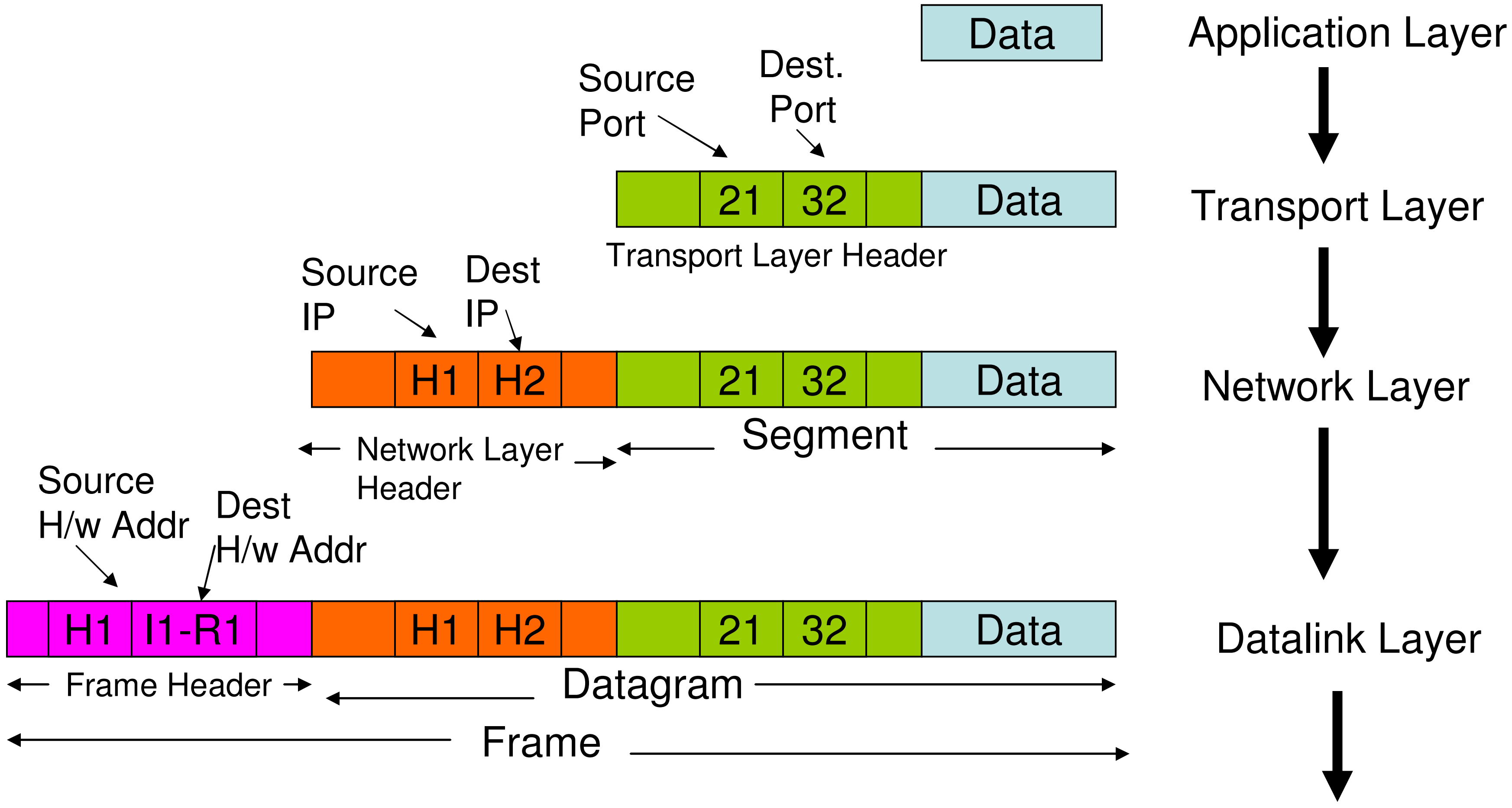
Example for End-to-End Packet Transmission across the Internet



Host/ Router	IP address	Hardware address
H1	143.132.0.1	34:12:45:AB:CD:EF
Interface 1 of R1	143.132.90.2	38:45:A9:E2:B5:C3
Interface 2 of R1	198.90.56.1	4C:9A:3B:54:DF:12
Interface 1 of R2	198.90.56.2	24:3B:1C:4A:52:CD
Interface 2 of R2	202.78.23.1	9C:12:AB:89:CF:33
Interface 1 of R3	202.78.23.2	BC:32:11:A2:45:23
Interface 2 of R3	190.34.0.1	28:12:AB:45:69:12
H2	190.34.0.2	30:90:CD:EF:AB:43



At H1



Data

Application Layer

Source Port

Dest. Port

	21	32		Data

Transport Layer Header

Transport Layer

Source IP

Dest IP

	H1	H2			21	32		Data

Network Layer Header

Segment

Network Layer

Source H/w Addr

Dest H/w Addr

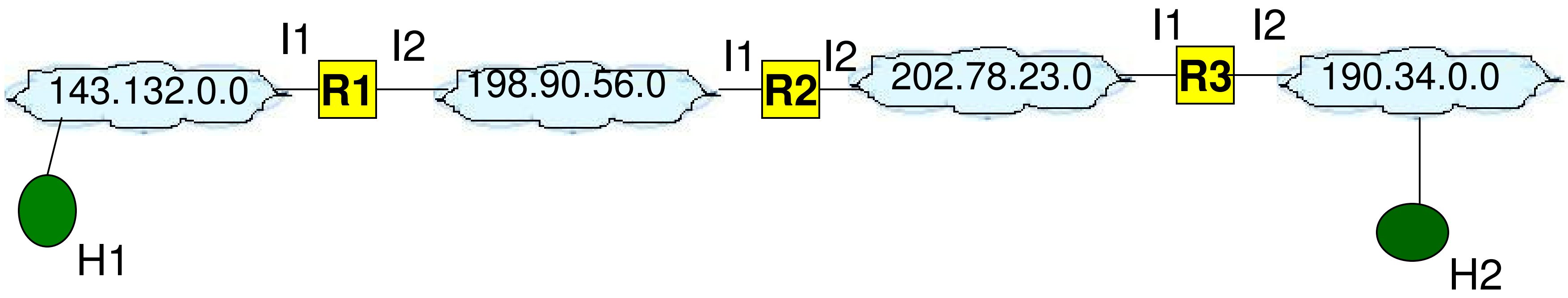
	H1	I1-R1			H1	H2			21	32		Data

Frame Header

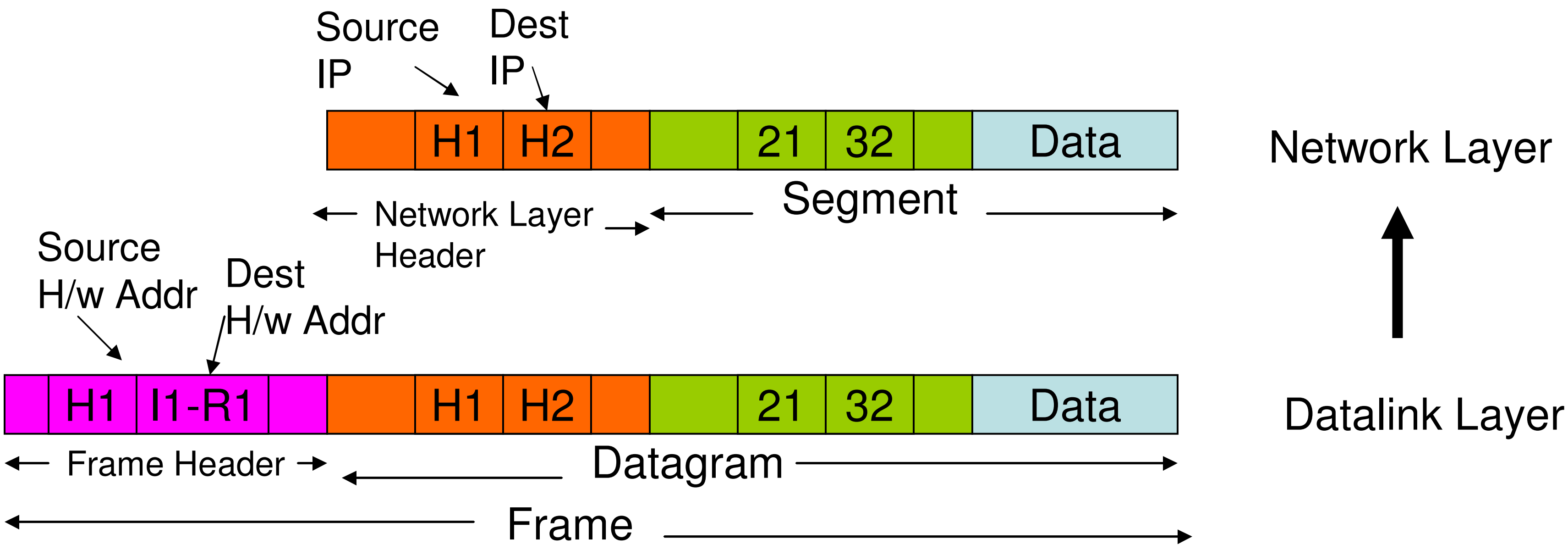
Datagram

Frame

Datalink Layer



At Interface 1 of R1



Source IP

Dest IP

H1 | H2 | 21 | 32 | Data

Network Layer Header

Segment

Network Layer

Source H/w Addr

Dest H/w Addr

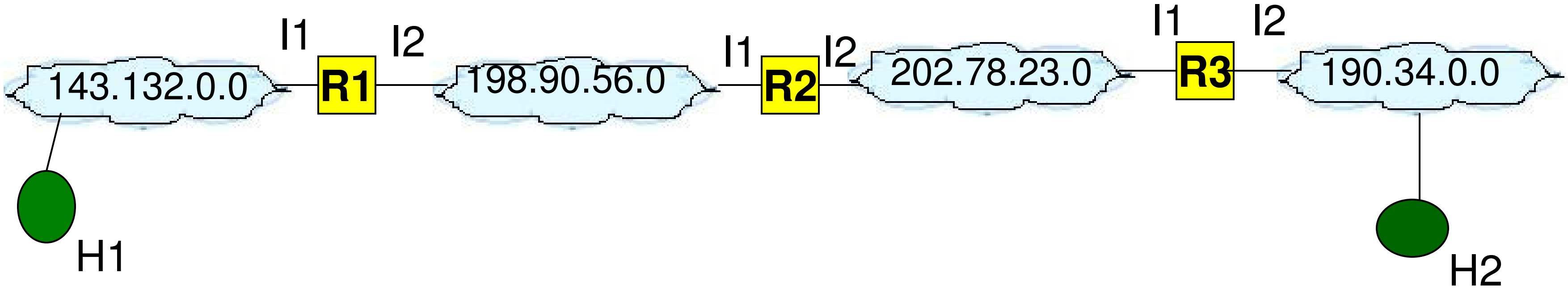
H1 | I1-R1 | H1 | H2 | 21 | 32 | Data

Frame Header

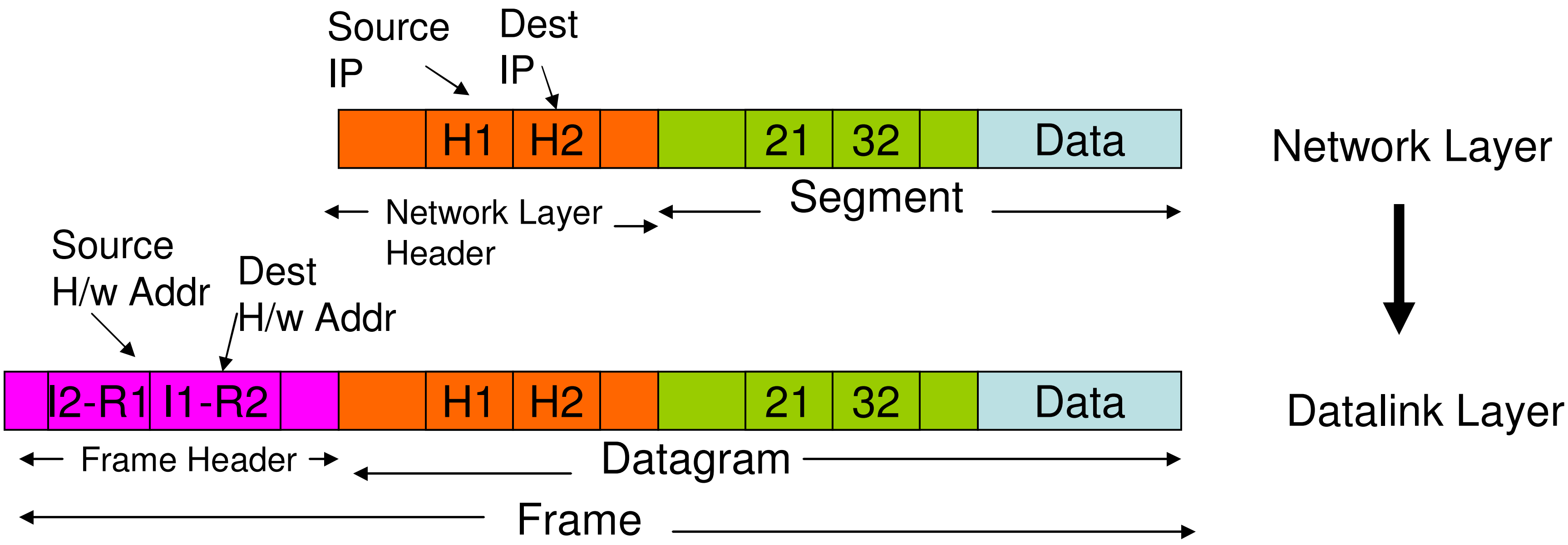
Datagram

Frame

Datalink Layer

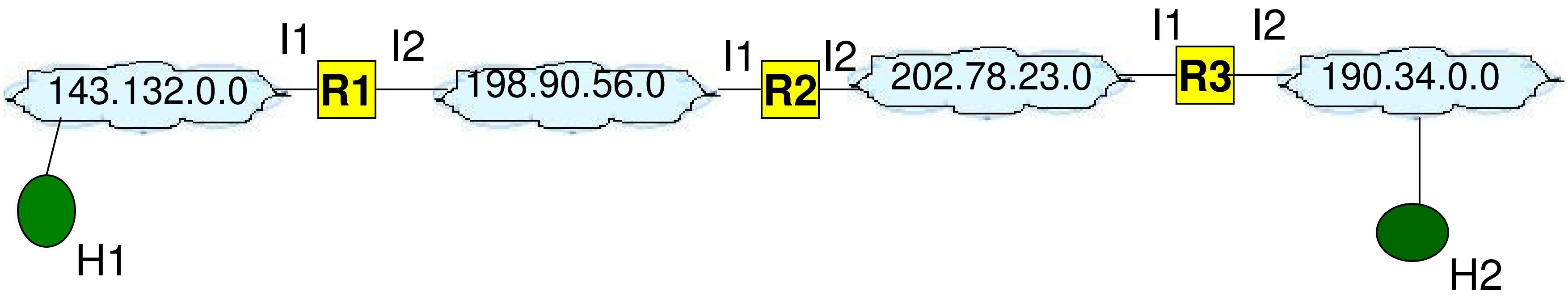


At Interface 2 of R1

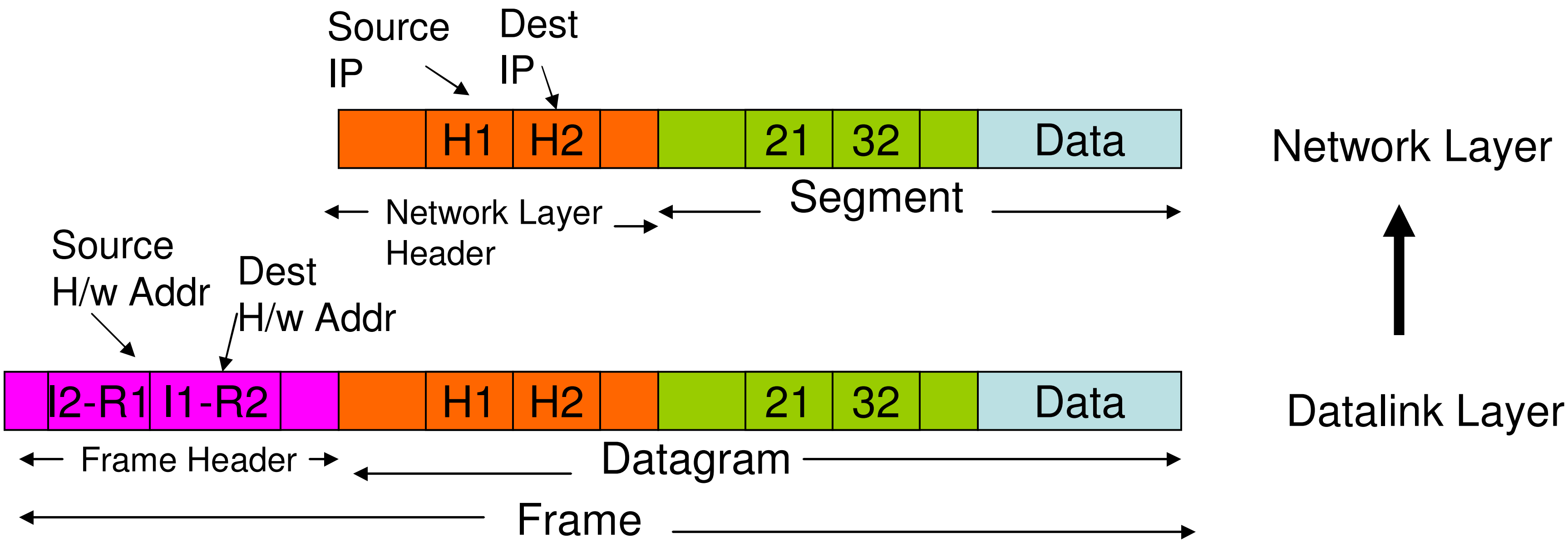


	H1	H2			21	32		Data

	I2-R1	I1-R2			H1	H2			21	32		Data



At Interface 1 of R2



Source IP

Dest IP

	H1	H2			21	32		Data

Network Layer Header

Segment

Network Layer

Source H/w Addr

Dest H/w Addr

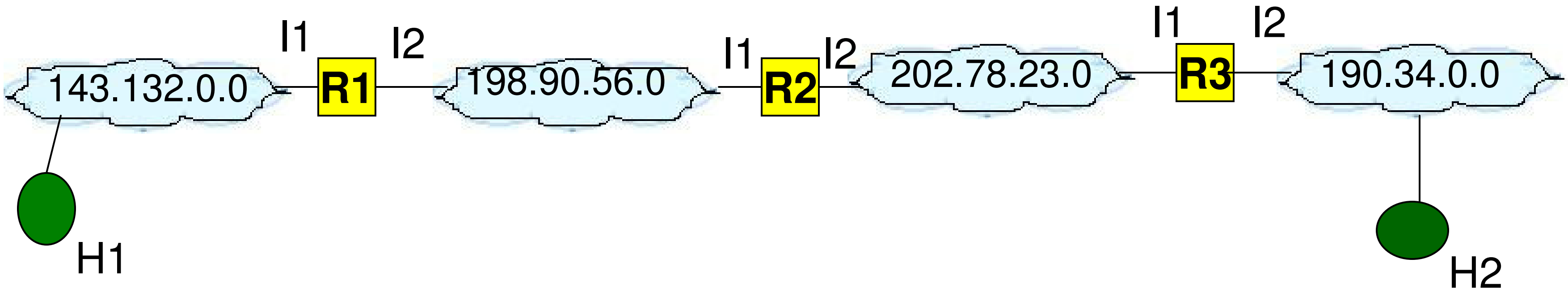
	I2-R1	I1-R2			H1	H2			21	32		Data

Frame Header

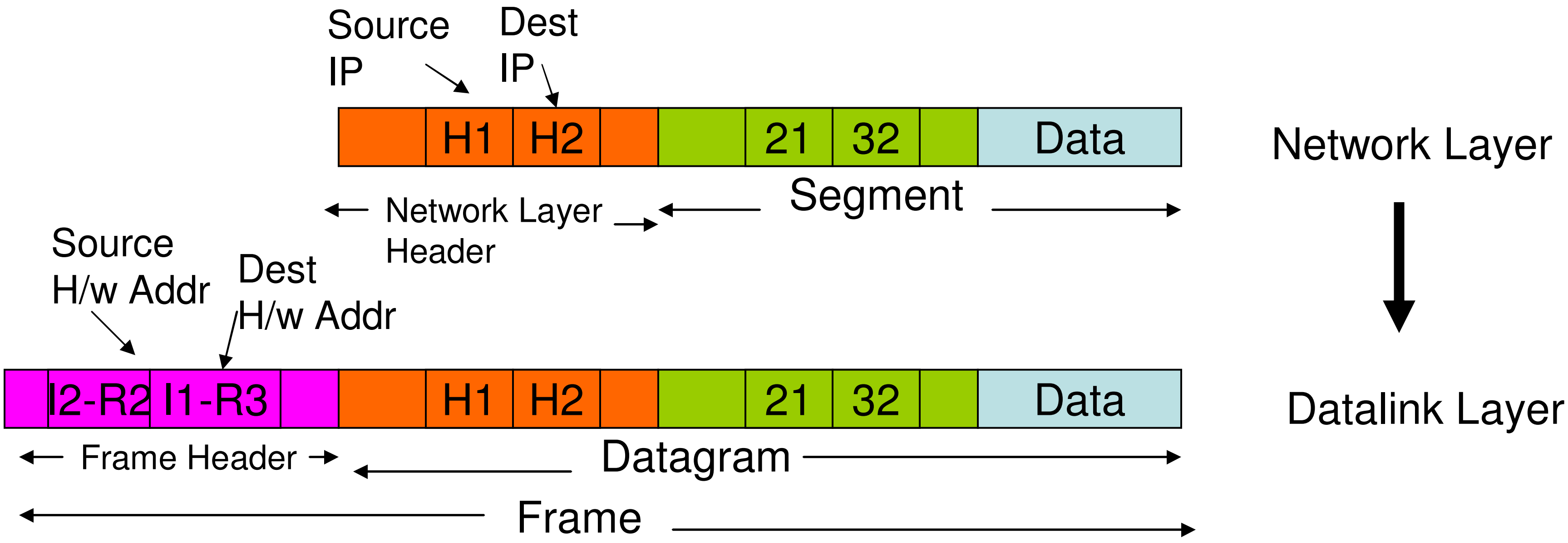
Datagram

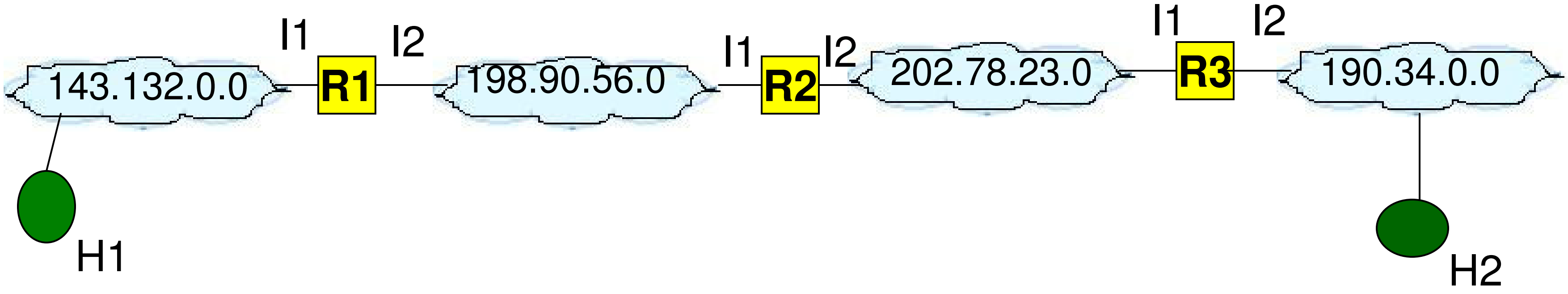
Frame

Datalink Layer

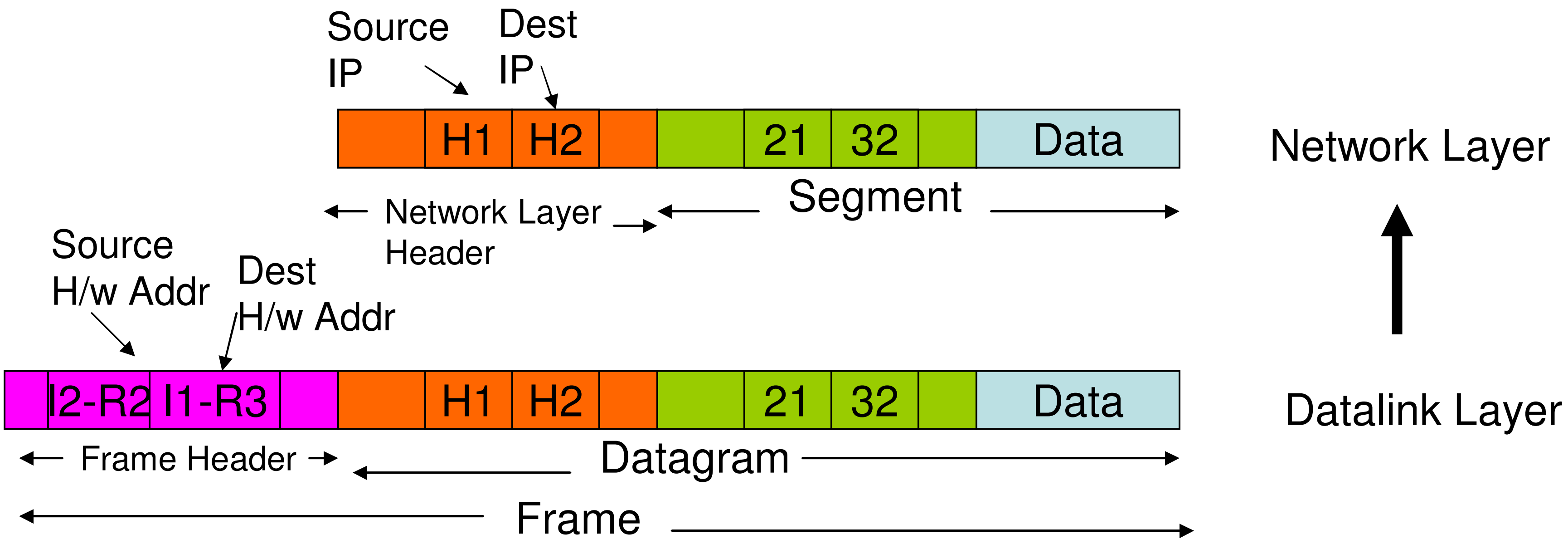


At Interface 2 of R2





At Interface 1 of R3


Source IP

Dest IP

| | H1 | H2 | | | 21 | 32 | | Data |

← Network Layer Header →

← Segment →

Network Layer

Source H/w Addr

Dest H/w Addr

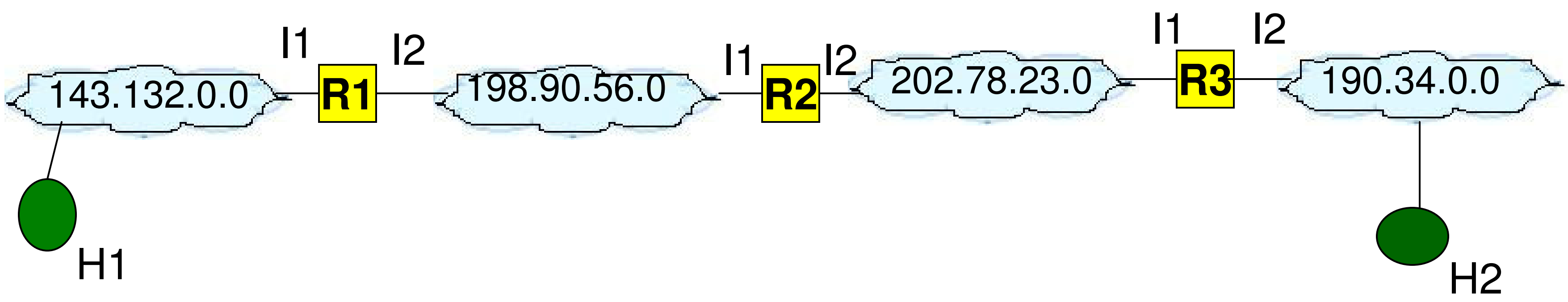
| | I2-R2 | I1-R3 | | | H1 | H2 | | | 21 | 32 | | Data |

← Frame Header →

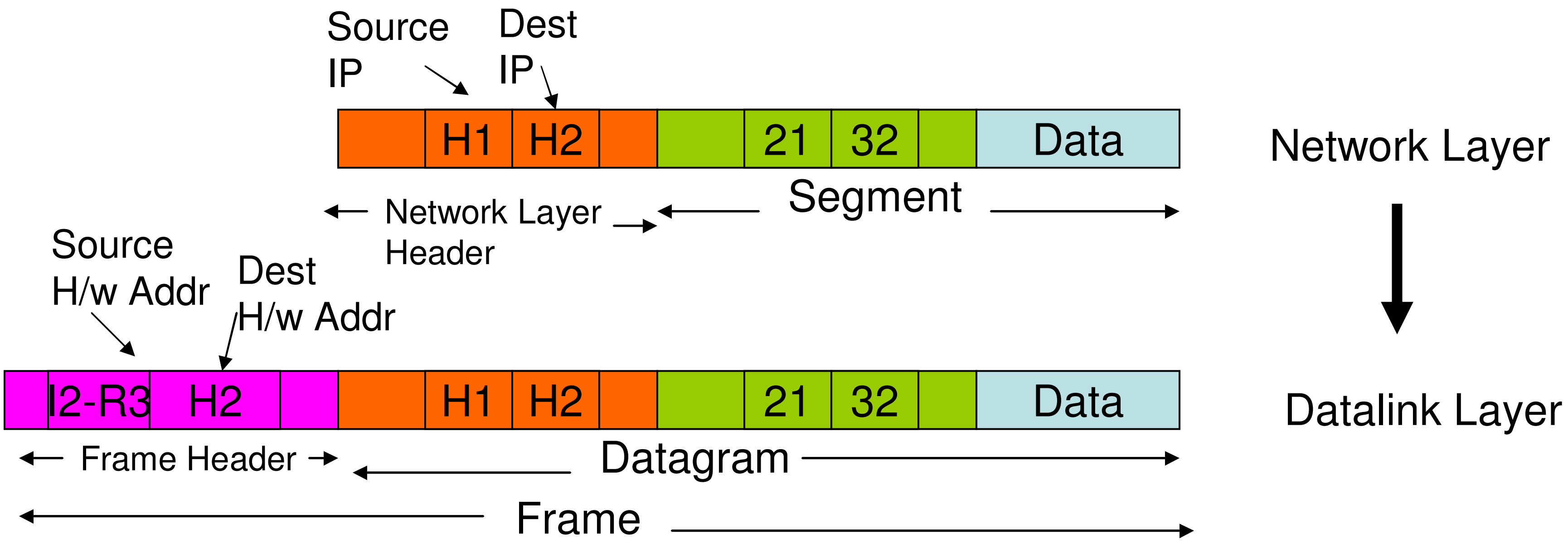
← Datagram →

← Frame →

Datalink Layer



At Interface 2 of R3



Source IP

Dest IP

	H1	H2			21	32		Data

← Network Layer Header → ← Segment →

Network Layer

Source H/w Addr

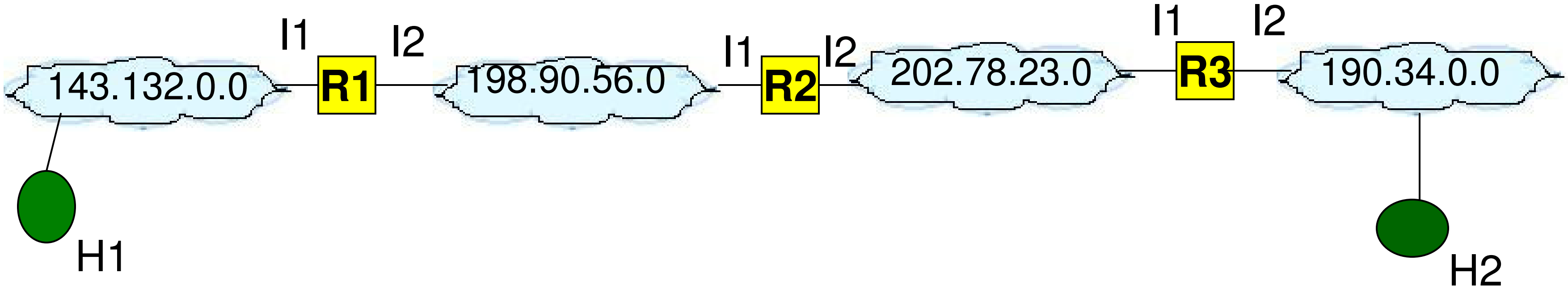
Dest H/w Addr

	I2-R3	H2			H1	H2			21	32		Data

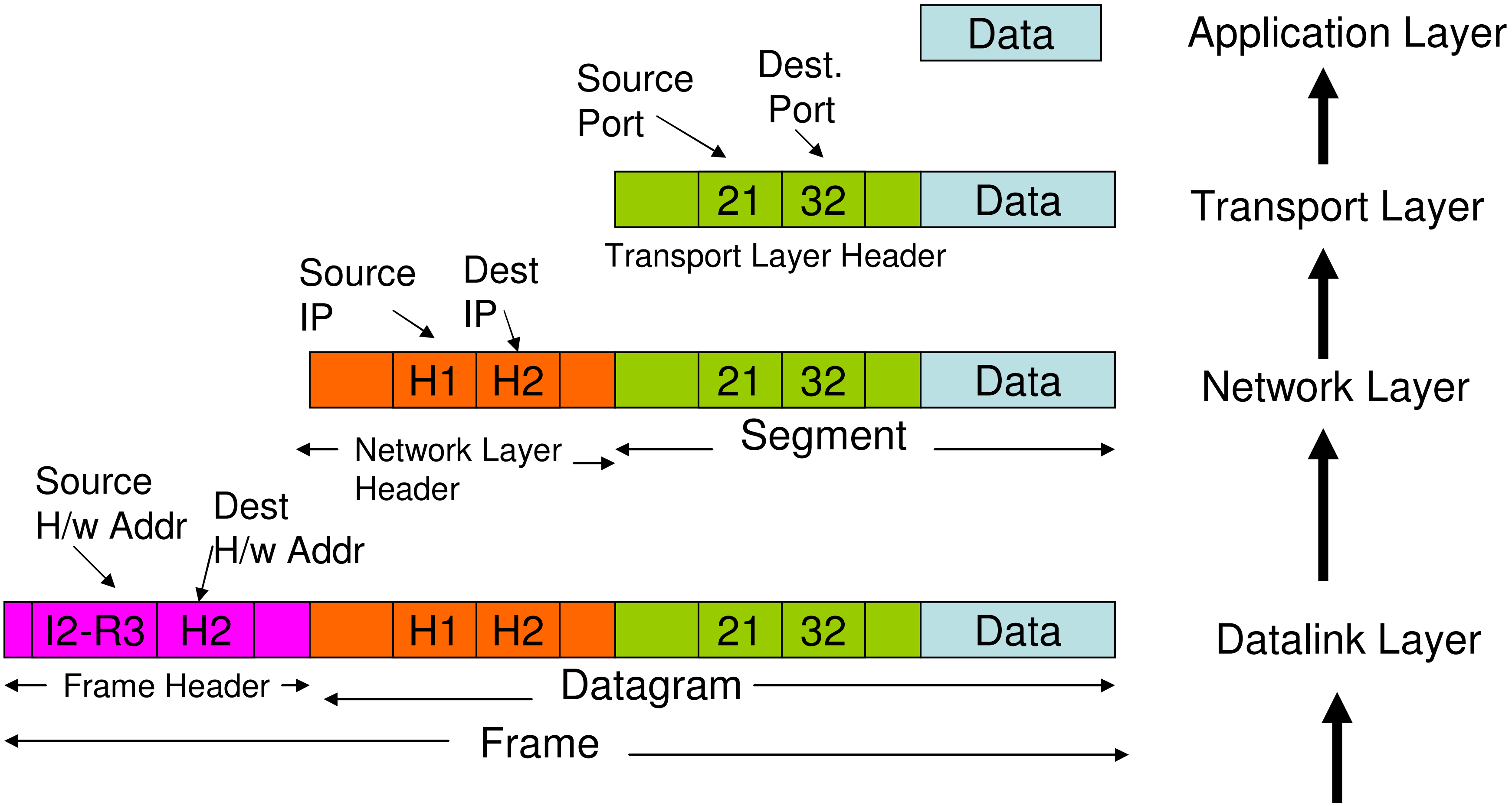
← Frame Header → ← Datagram →

← Frame →

Datalink Layer


I1
I2
143.132.0.0
R1
198.90.56.0
I1
I2
R2
202.78.23.0
I1
I2
R3
190.34.0.0
H1
H2

At H2


Data
Application Layer

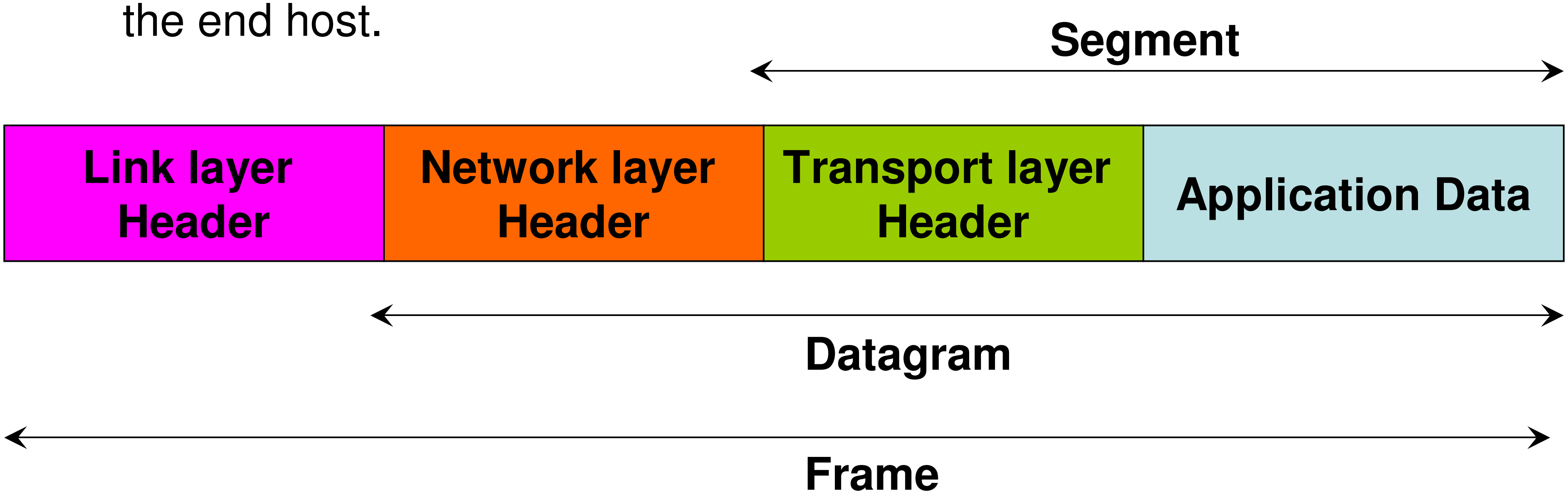
Source Port
Dest. Port
21 | 32 | Data
Transport Layer Header
Transport Layer

Source IP
Dest IP
H1 | H2 | 21 | 32 | Data
Network Layer Header
Segment
Network Layer

Source H/w Addr
Dest H/w Addr
I2-R3 | H2 | H1 | H2 | 21 | 32 | Data
Frame Header
Datagram
Frame
Datalink Layer

Segment, Datagram and Frame

- Segment – Transport layer (TCP or UDP) header + Application Data
- Datagram – Network layer (IP) header + Segment
- Frame – Link layer (frame) header + Datagram
- The physical layer, network interface and Internet layers are called the host-to-host layers as the headers corresponding to these layers are exposed at each intermediate; whereas, the transport and application layers are called as the end-to-end layers as the header and application data corresponding to these layers are seen only at the end host.



TCP/IP Protocol Stack

1. Physical Layer: Corresponds to the ISO 7-layer model's physical layer.

2. Network Interface Layer: Specifies organization of data into frames and transmission of frames over a network; similar to the ISO 7-layer model's data link layer.

3. Internet Layer: Specifies the format of packets sent across an internet and mechanisms to forward packets from a source computer through one or more routers to a final destination.

4. Transport Layer: Similar to the ISO model's transport layer, specifying reliable transfer.

5. Application Layer: Corresponds to the presentation and application layer of the ISO model; TCP/IP's application layer protocols specify how an application uses an internet.

Denial of Service Attacks

Introduction

- A Denial of Service (DoS) attack is an action that prevents or impairs the authorized use of networks, systems, or applications by exhausting resources such as CPUs, memory, bandwidth and disk space.
  - Example: Flooding a web server with several spurious requests that it is unable to respond to valid requests from users in a timely manner.

- Attack categories:
  - Bandwidth (TCP ACK Storm)
  - System buffer (SYN Spoofing)
  - Poison packet (Packets whose structure triggers a bug in the system's network handling software, causing it to crash; Example: Teardrop attack)
  - Distributed DoS attacks (attacks from distributed systems – multiple sources)

- Often attackers (to complicate detection) try to hide the identity of the source systems and launch the DoS attacks using a falsified, or spoofed, address [Need administrative access to the network interface].

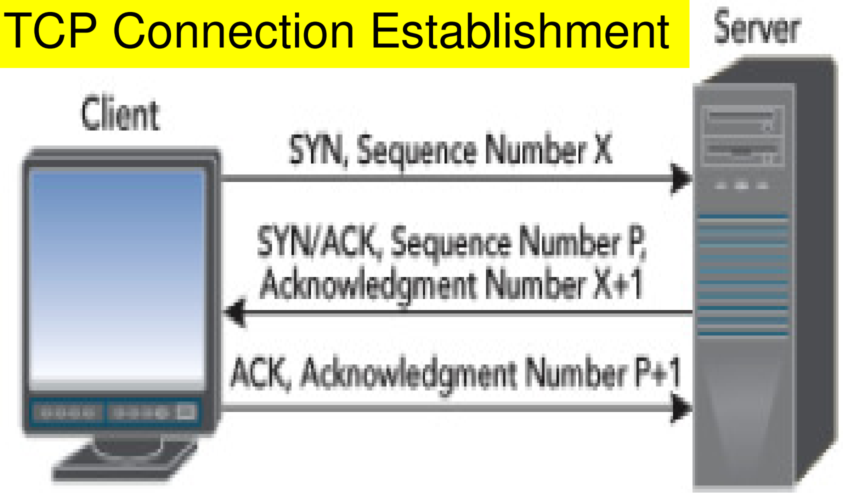
Port Scanning

- Port scan: is a program that when run for a particular IP address, reports the following:
  - Which standard ports or services are running and responding on the target system.
  - What operating system is installed on the target system
  - What applications (and their versions) are present at the target system
- All of the above information can be collected quietly, anonymously, without identification or authentication, drawing little attention to the scan.
- After knowing details like the OS, the application programs and versions, an attacker can explore the weaknesses of these software and potential loopholes to get into the target system using these services.
  - Sometimes sending messages from an application to another application running at the target host may help us to obtain the version details of that application at the target.
- Port scanning tools: nmap, netcat – free tools, several commercial tools are also available.
  - nmap: Given an IP address, nmap reports all open ports, the service they support, and the user ID of the daemon providing the service.
  - The user ID is important because it helps us to further explore the gains that could be obtained if his/ her service running on the target host is compromised.

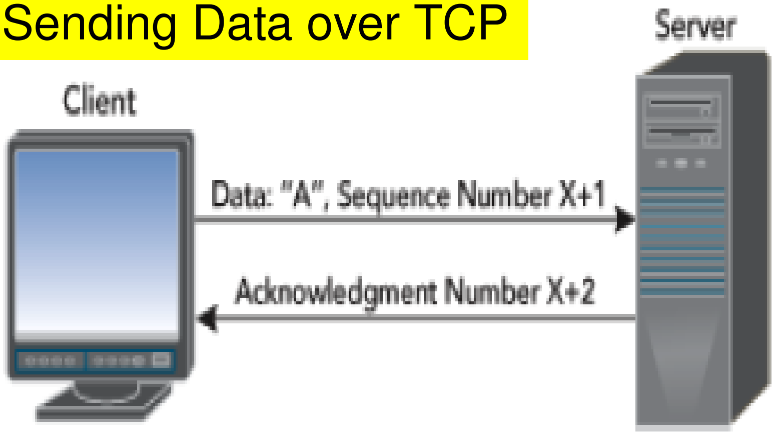
TCP Connection Establishment

- The client first initiates a session with the server by sending a synchronization (SYN) packet to the server with initial sequence number *x*.
- The server responds with a SYN/ACK packet that contains the server's own sequence number *p* and the ACK number for the client's original SYN packet, *x*+1. (the ACK number indicates the next sequence number the server expects from the client).
- The client acknowledges receipt of the SYN/ACK packet by sending back to the server an ACK packet with the next sequence number, *p*+1, it expects from the server.
- The client and server are now ready to start exchanging data

TCP Connection Establishment


Server
Client
SYN, Sequence Number X
SYN/ACK, Sequence Number P,
Acknowledgment Number X+1
ACK, Acknowledgment Number P+1

Sending Data over TCP



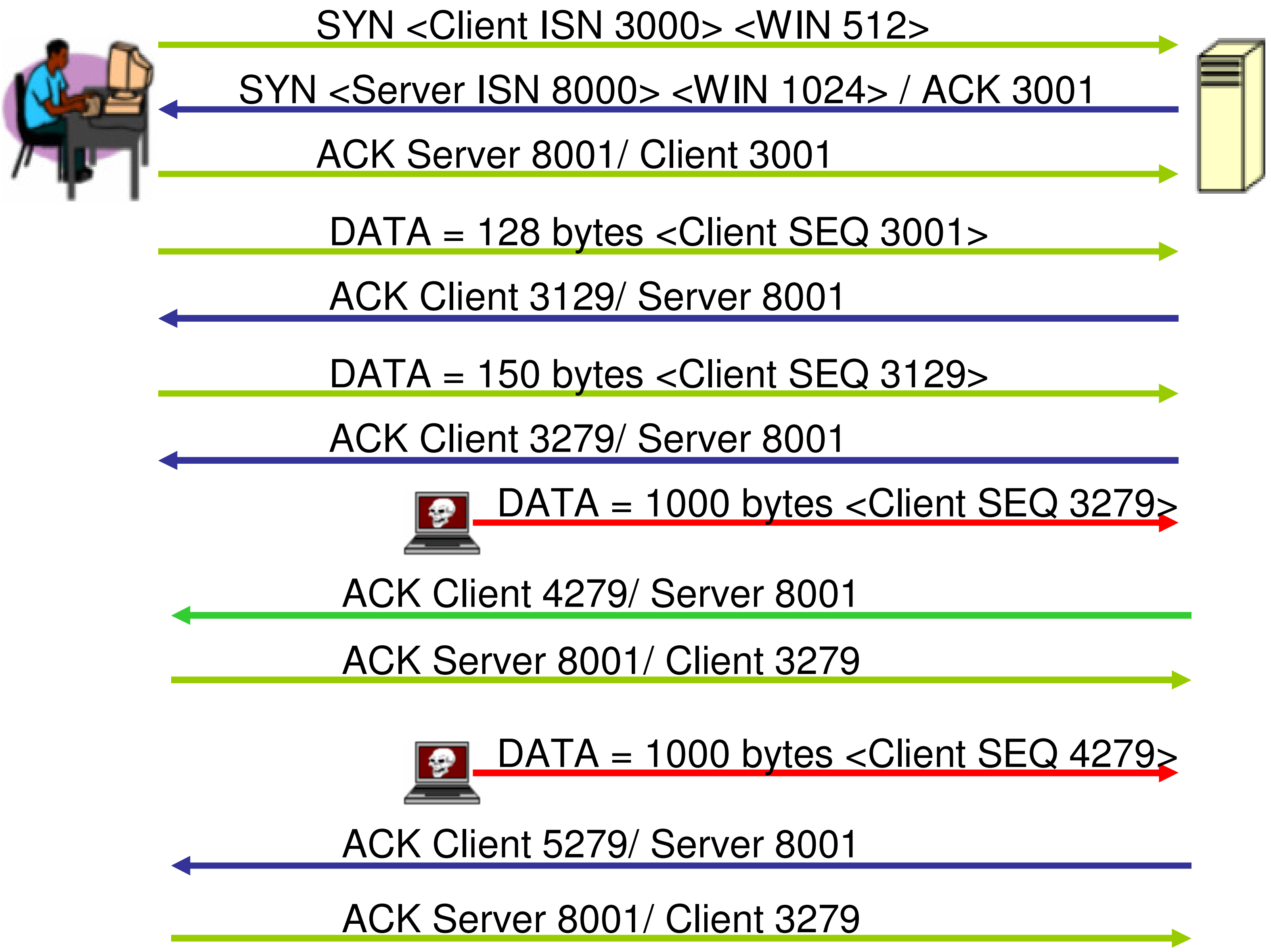
SYN Spoofing / Flooding Attacks

- When a client attempts to start a TCP connection to a server, the client and serve exchange a series of messages as follows:
  - The client requests a connection by sending a SYN message to the server.
  - The server acknowledges the request by sending SYN-ACK back to the client
  - The client responds with an ACK and the connection is established.
- Occasionally, packets get lost or damaged in transmission. The destination maintains a queue called the SYN_RECV connections, tracking those connection requests for which a SYN-ACK has been sent but the corresponding ACK has not yet been received.
- If the SYN-ACK or the ACK packet is lost, eventually the destination host will timeout the incomplete connection and discard it from its queue.
- The attacker sends many SYN requests from spoofed non-existing IP addresses and never responds back with ACKs, thereby filling up the SYN-RECV queue at the server.
- The server waits with the connection requests in the SYN-RECV queue and denies permitting any legitimate client connection request arriving in the mean time.
- SYN Spoofing attacks target exploiting the system resources (like buffer space); while SYN flooding attacks target the network bandwidth.
- The volume of spoofed traffic needed for flooding attacks is much larger than that needed for spoofing attacks.

TCP Session Hijacking

- Session hijacking is the act of taking over an already established TCP session and injecting your own packets into that stream so that your commands are processed as the authentic owner of the session.
- A TCP session is merely identified by the quadruple: Client IP address, Client Port number, Server IP address and Server Port number.
- Any packets that reach either machine with the above identifiers are assumed to be part of the existing session.
- Hence, if an attacker can spoof these items, they can pass TCP packets to the client or server and have those packets processed as coming from the other machine.
- Two steps: Desynchronize the session and Inject own commands
- Desynchronizing the session: Predict the sequence number to be used by a client (or server) and use that sequence number before the client (or server) gets a chance to.
  - How to do it? Use Local Session Hijacking or Blind Session Hijacking
  - <u>Local Session Hijacking:</u> If we have access to the network and can sniff the TCP session, we can tell the next expected sequence number from the ACK packets exchanged
  - <u>Blind Session Hijacking:</u> If we do not have the ability to sniff the TCP session between the client and server, then we have try all options and guess the expected sequence number.

Desynchronizing a Session



SYN <Client ISN 3000> <WIN 512>

SYN <Server ISN 8000> <WIN 1024> / ACK 3001

ACK Server 8001/ Client 3001

DATA = 128 bytes <Client SEQ 3001>

ACK Client 3129/ Server 8001

DATA = 150 bytes <Client SEQ 3129>

ACK Client 3279/ Server 8001

DATA = 1000 bytes <Client SEQ 3279>

ACK Client 4279/ Server 8001

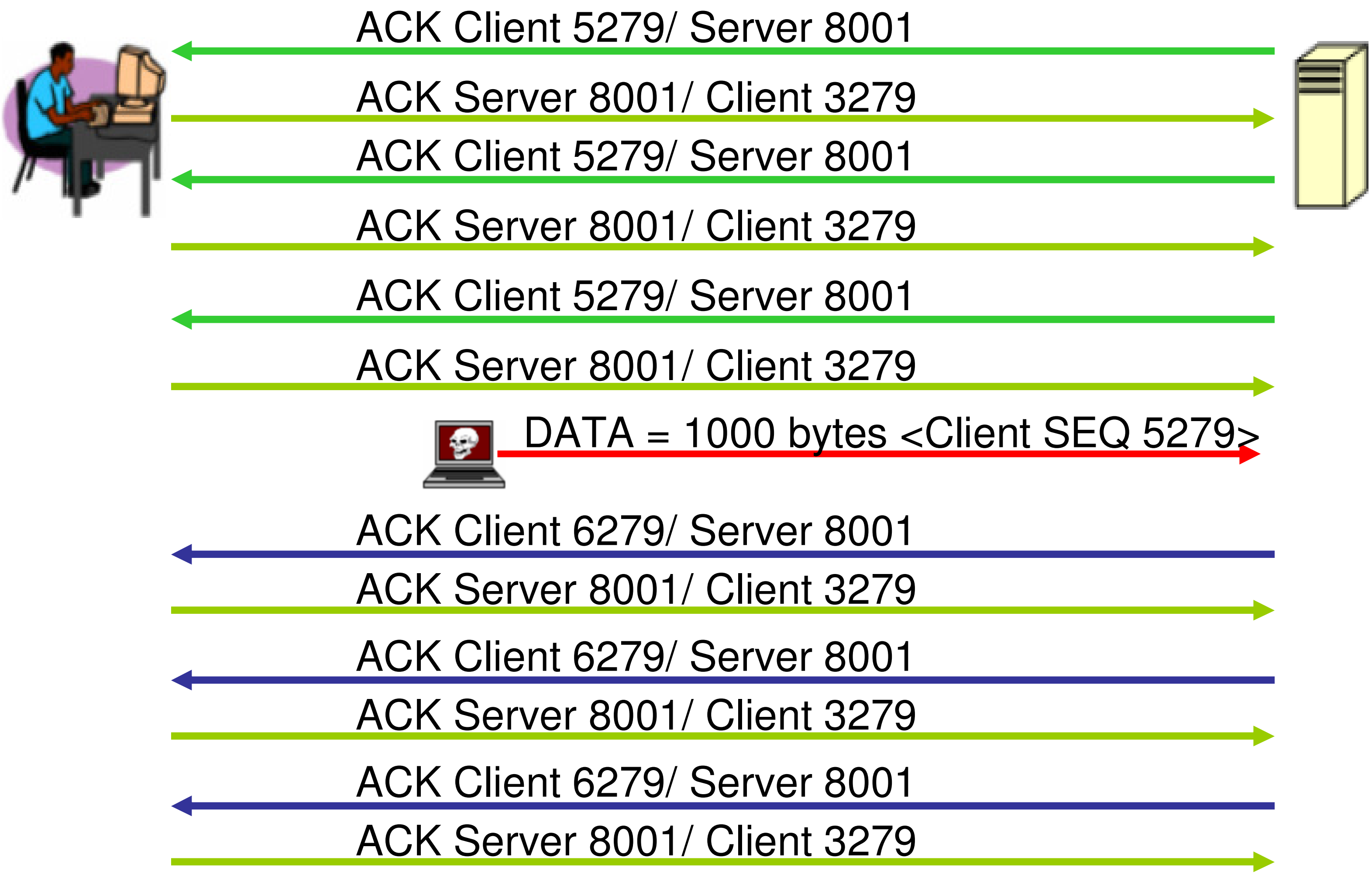
ACK Server 8001/ Client 3279

DATA = 1000 bytes <Client SEQ 4279>

ACK Client 5279/ Server 8001

ACK Server 8001/ Client 3279

TCP ACK Storm



ACK Client 5279/ Server 8001

ACK Server 8001/ Client 3279

ACK Client 5279/ Server 8001

ACK Server 8001/ Client 3279

ACK Client 5279/ Server 8001

ACK Server 8001/ Client 3279

DATA = 1000 bytes <Client SEQ 5279>

ACK Client 6279/ Server 8001

ACK Server 8001/ Client 3279

ACK Client 6279/ Server 8001

ACK Server 8001/ Client 3279

ACK Client 6279/ Server 8001

ACK Server 8001/ Client 3279

TCP Session Hijacking

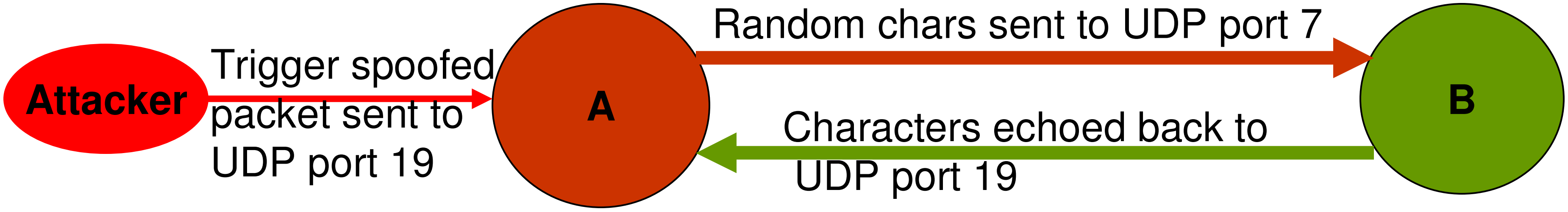
- When the attacker successfully hijacks the TCP session and injects its own data (and spoofs the server as if the data is coming from the original client), the server will acknowledge the receipt of the data by sending to the original client an ACK packet.
- This ACK packet will most likely contain a sequence number that the client is not expecting, so when the client receives this packet, it will try to resynchronize the TCP Session with the server by sending it an ACK packet with the sequence number that it is expecting.
- This ACK packet will in turn contain a sequence number that the server is not expecting and so the server will resend its last ACK packet.
- This cycle will go on and on, and this rapid passing back and forth of the ACK packets creates the TCP ACK Storm.
- The attacker could keep injecting more and more data, the size of the ACK storm increases and can quickly degrade the network performance.
- The original client will have to eventually get exhausted after a certain number of resynchronization attempts and close the connection with the server.

Internet Control Message Protocol

- The ICMP protocol suite is normally used for system and network diagnostics. Several ICMP messages are used:
  - Echo Request: Is an ICMP message whose data is expected to be received back in an echo reply. The host must respond to all Echo Requests with an Echo Reply containing the exact data received in the request message.
  - Echo Reply: Is an ICMP message generated in response to an Echo request and is mandatory for all hosts and routers.
    - The ICMP Echo service is run in UNIX systems at UDP port 7
  - Destination Unreachable: Is an ICMP message generated by a router/ host to inform the source that the destination is unreachable for certain reasons like:
    - No next hop router is available to forward the data packet toward the destination.
    - The port number at the destination is not active
    - The data must be fragmented but the ‘Don’t fragment’ flag is on.
  - Source Quench: Is an ICMP message that request the sender to reduce the traffic rate of messages to a router or a host. This message may be generated if the router or host does not have sufficient buffer space to process the request, or may occur if the router or host’s buffer is approaching its limit.

Echo-Chargen Attack (Reflection Attack)

- The Chargen (Character Generator) service is an internet-layer protocol and is intended for testing and measurement purposes.
- A host may connect to a server that supports the Chargen protocol, on either TCP or UDP port 19.
- Upon opening a TCP connection, the server starts sending arbitrary characters to the connecting host and continues until the host closes the connection.
- In the UDP version of the protocol, the server sends an UDP packet containing a random number (between 0 and 512) of characters every time it receives an UDP packet from the connecting host.
- Attack Scenario: The attacker sends a packet to machine A's UDP port 19 with machine B as the forged source address and UDP port 7 as the source port.
  - The attacker spoofs a conversation between the two services and redirects the output of each service to the other, creating a rapidly expanding spiral of traffic.
  - Eventually, the attack begins to consume memory and processor power at the targeted devices A and B, causing them to become non-responsive to user commands.



Attacker

Trigger spoofed packet sent to UDP port 19

A

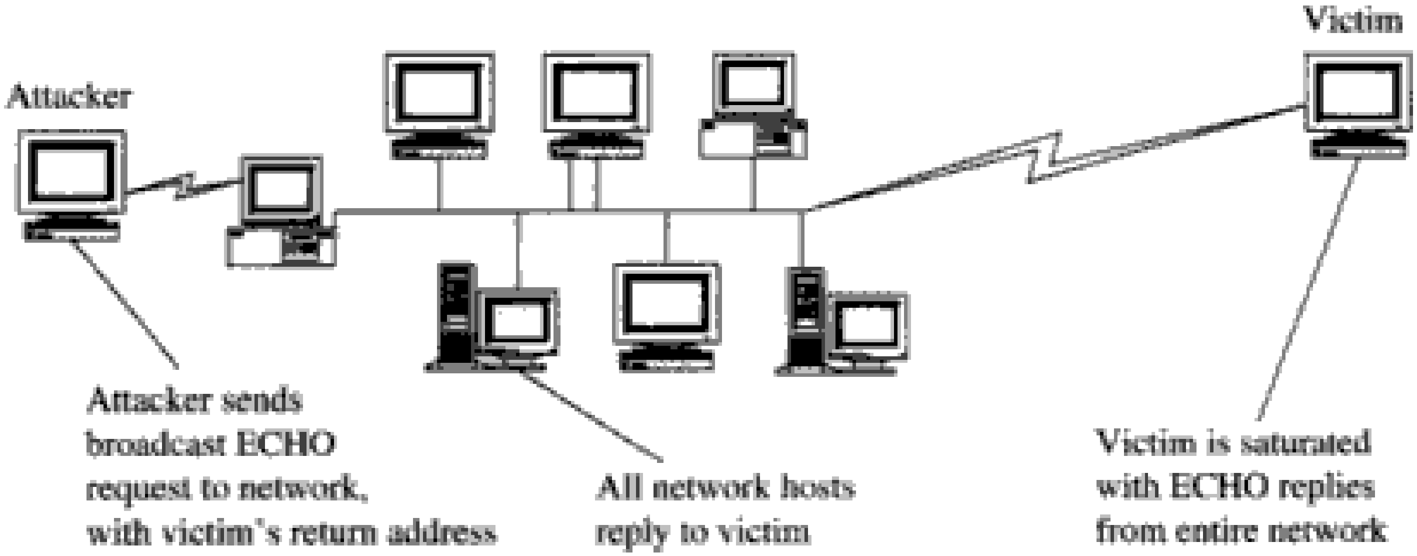
Random chars sent to UDP port 7

B

Characters echoed back to UDP port 19

Smurf Attack (Amplification Attacks)

- The smurf attack, a kind of denial-of-service attack, floods a target system via spoofed broadcast ping Echo-Reply messages.
- A perpetrator sends a ping Echo-Request message (having a spoofed address of the intended victim) to the broadcast IP address of a network.
- All the hosts in that network on receiving the ping message, send a reply to the source of the ping, which is the victim machine.


Victim
Attacker
Attacker sends broadcast ECHO request to network, with victim's return address
All network hosts reply to victim
Victim is saturated with ECHO replies from entire network

- Solution: After the incidence of several Smurf attacks, routers in the Internet were configured not to forward packets having a broadcast IP address as the destination address. Hosts are also configured not to respond to ping requests that were sent to them as a broadcast message.

Teardrop Attack and Traffic Redirection

- The Teardrop attack involves sending IP fragments with overlapping, over-sized, payloads to the target machine.
- The attacker sends a series of datagrams to the target machine, such that the fragments cannot fit together properly.
- Example:
  - One datagram might say it is position 0 for length 60 bytes, another position 30 for length 90 bytes and another position 41 for length 173 bytes.
  - As the above three pieces overlap, they cannot be reassembled properly.
  - The OS locks up with these partial data units it cannot reassemble, thus leading to denial-of-service attacks.
- Modern operating systems are configured to discard reassembly when overlapping fragments arrive and a simple reboot could bring the system back to normality.

- Traffic Redirection: If a router is compromised, then it could advertise to all its neighboring routers that it lies on the shortest path to every other destination network in the Internet. Soon, all the traffic will be redirected to this router, the router is flooded, drops all the packets and they do not make it to their intended destination.

DNS Attacks

- A Domain name server (DNS) is a machine that holds a table mapping domain names to IP addresses.
- A DNS server queries other DNS servers to resolve domain names it does not know and updates its table with the mapping learnt.
- DNS Cache Poisoning: Is a technique that tricks a DNS server into believing it has received authentic information when, in reality, it has not.
- Once the DNS server has been poisoned, the information is generally cached for a while, spreading effect of the attack to users of the server.
- Example:
  - An attacker poisons the IP address DNS entries for a target website on a given DNS server, replacing them with the IP address of the server the attacker controls.
  - The attacker then creates fake entries on the server he/she controls with names matching those on the target server.
  - These files could contain malicious content, such as a worm or a virus.
  - A user whose computer has referenced the poisoned DNS server would be tricked into thinking that the content comes from the target server and unknowingly download malicious content.

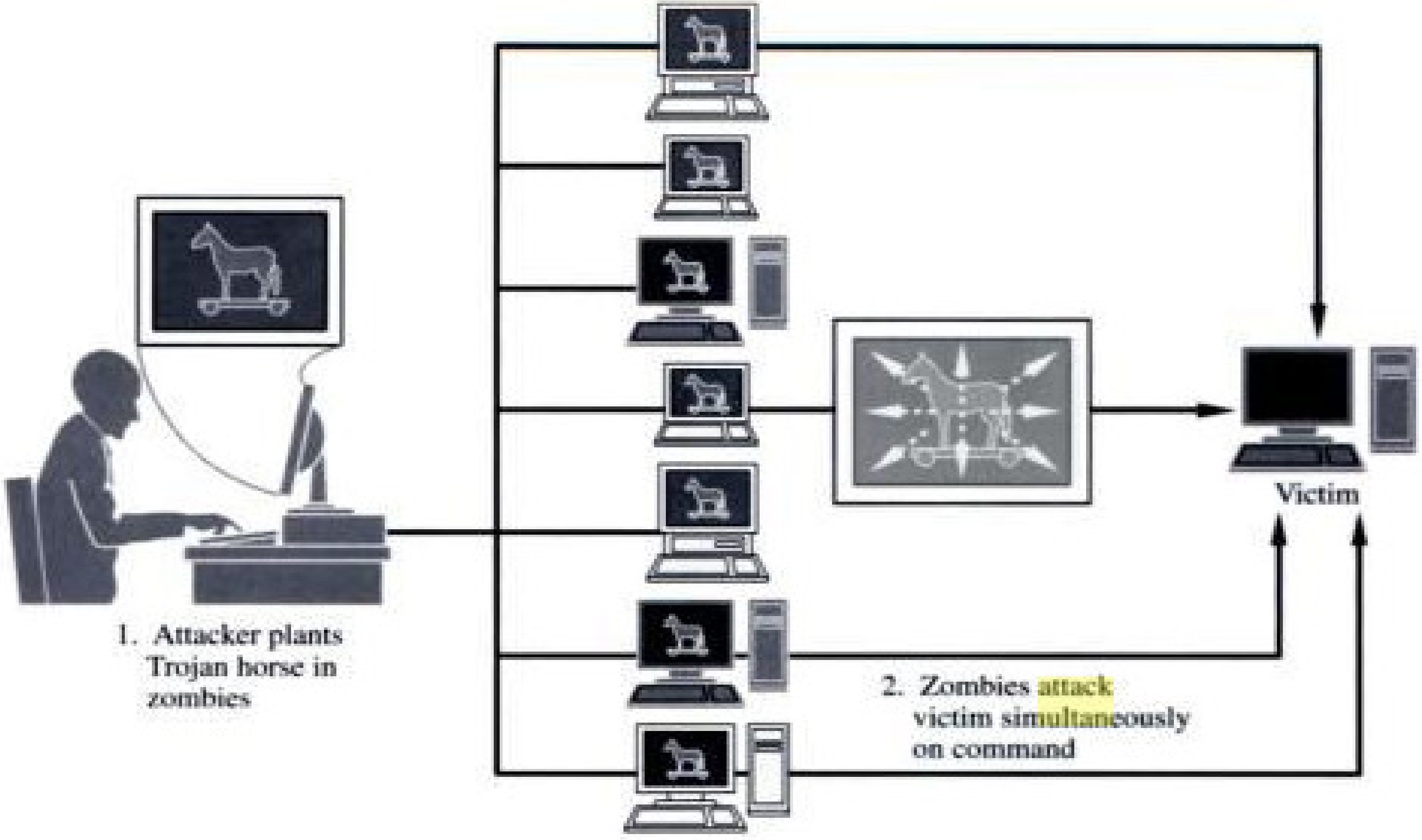
Distributed Denial-of-Service Attacks

- DDoS attacks involve breaking into hundreds or thousands of machines all over the Internet. Then, the attacker installs DDoS software on these burgled machines and controls them to launch coordinated attacks on victim sites.
- DDoS attacks typically exhaust bandwidth, router processing capacity and break network connectivity to the victims.

- First stage: Forming Zombies
  - The attacker uses any convenient attack (such as exploiting buffer overflow or tricking the victim to open and install unknown code from an email attachment) to plant a Trojan Horse on a target machine.
  - The attacker also installs software such as “rootkit” on these compromised machines. The rootkit helps to conceal the fact of the break-in, hide traces of subsequent malicious activities and also replaces the standard commands for displaying running processes with versions that fail to replace the attacker’s processes. The compromised victim machine is referred to as a “zombie”
  - A zombie could also be used to break into some more machines, install the Trojan Horse and rootkits and convert them to be a zombie.

Distributed Denial-of-Service Attacks

- Second stage: Launch the DDoS attack through the zombies.
  - The attacker choose a victim and sends a signal to all the zombies to launch the attack. Each zombie could launch a different type of attack on the victim.
  - A victim of the DDoS attack will thus have to counter multiple zombies, launching the same or different types of attacks.


Victim
1. Attacker plants Trojan horse in zombies
2. Zombies attack victim simultaneously on command

Solutions to Control DoS Attacks

- Filter packets with spoofed source addresses at the originating network gateway router itself. Firewall solution
- Trace the path of certain control packets back to the source. Internet layer soln
  - With IP being stateless, this would require coordination among the ISPs and their routers in the Internet.
- The volume of ICMP packets (like Echo Reply/Request, Destination not reachable, etc) that are transmitted in and out of the networks should be within a threshold. (Firewall solution)
  - Only a small volume of the network traffic
- Modify TCP code to generate an encrypted cookie with the incremented sequence number and the resources requested, etc; send it to the requesting client along with the SYN-ACK packet. The legitimate client should respond back with the cookie, and the server can then commit the resources. (Transport layer solution)
  - Tradeoff: Time/resources spent at the server for cookie encryption/decryption. The resources are not committed right away. It is possible that certain connections do not get the requested resources at the end of the handshake.
- When the available resources are all tied up to existing open connections, just drop some randomly chosen open connection. Transport layer solution
- Limit the number of connection requests coming to a particular port or coming from a particular IP address (firewall solutions)
- No directed broadcasts over the Internet Internet layer solution
- Secure coding of applications (e.g., no buffer overflow attacks) that are run on the systems so that they are not exploited to compromise a system (limit the zombies).

Application layer solution