Introduction to Network Security

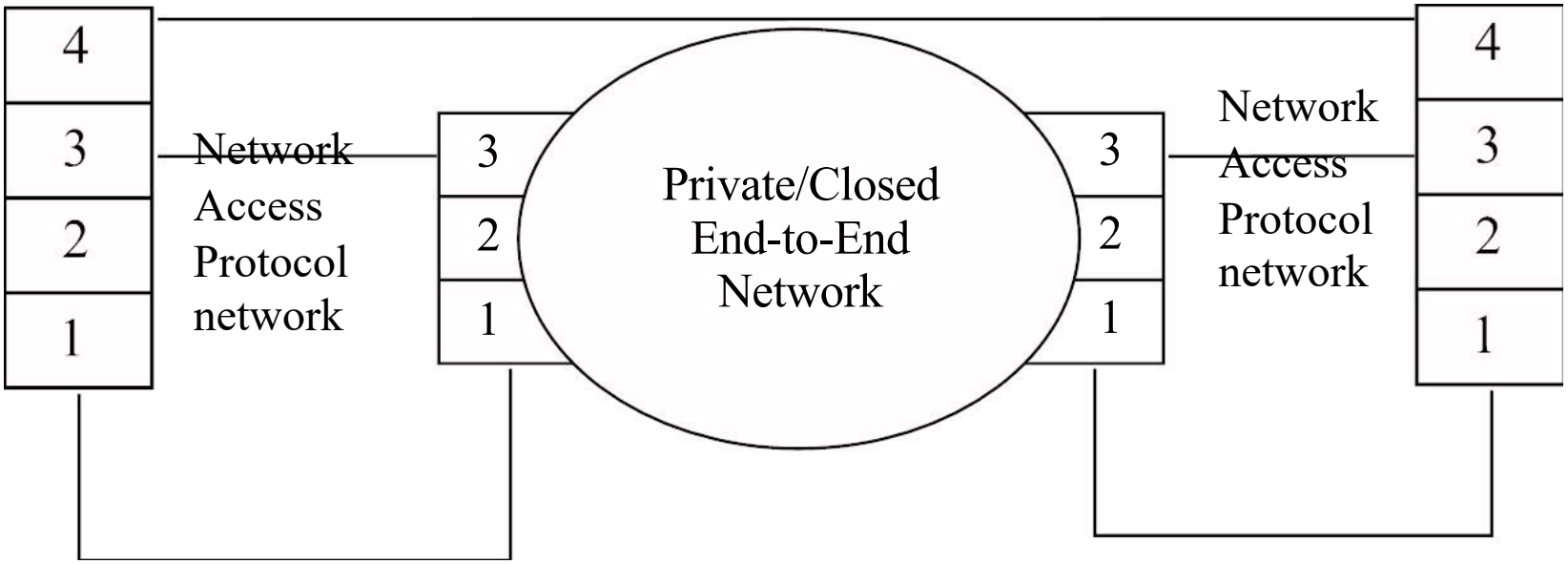
Chapter 6

Network Layer Protocols

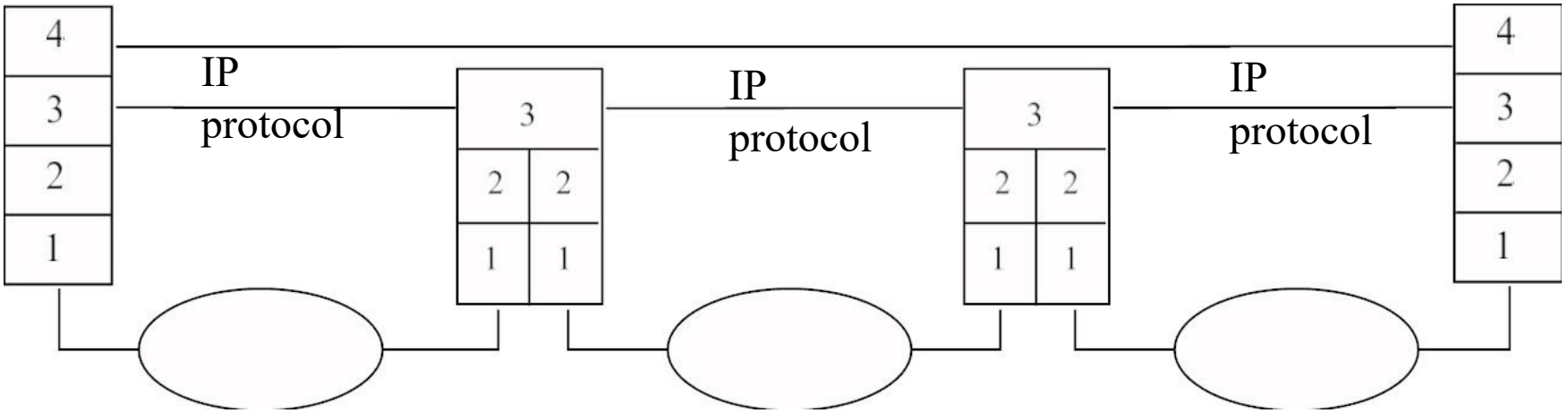
Network Layer

- Two Types:
  - Network access layer
    - Connection to a private end-to-end network
    - Used by ISPs to interconnect
  - Internetwork Layer
    - Distributed set of network layers working together
    - Used throughout the Internet

Network Access



Internetwork



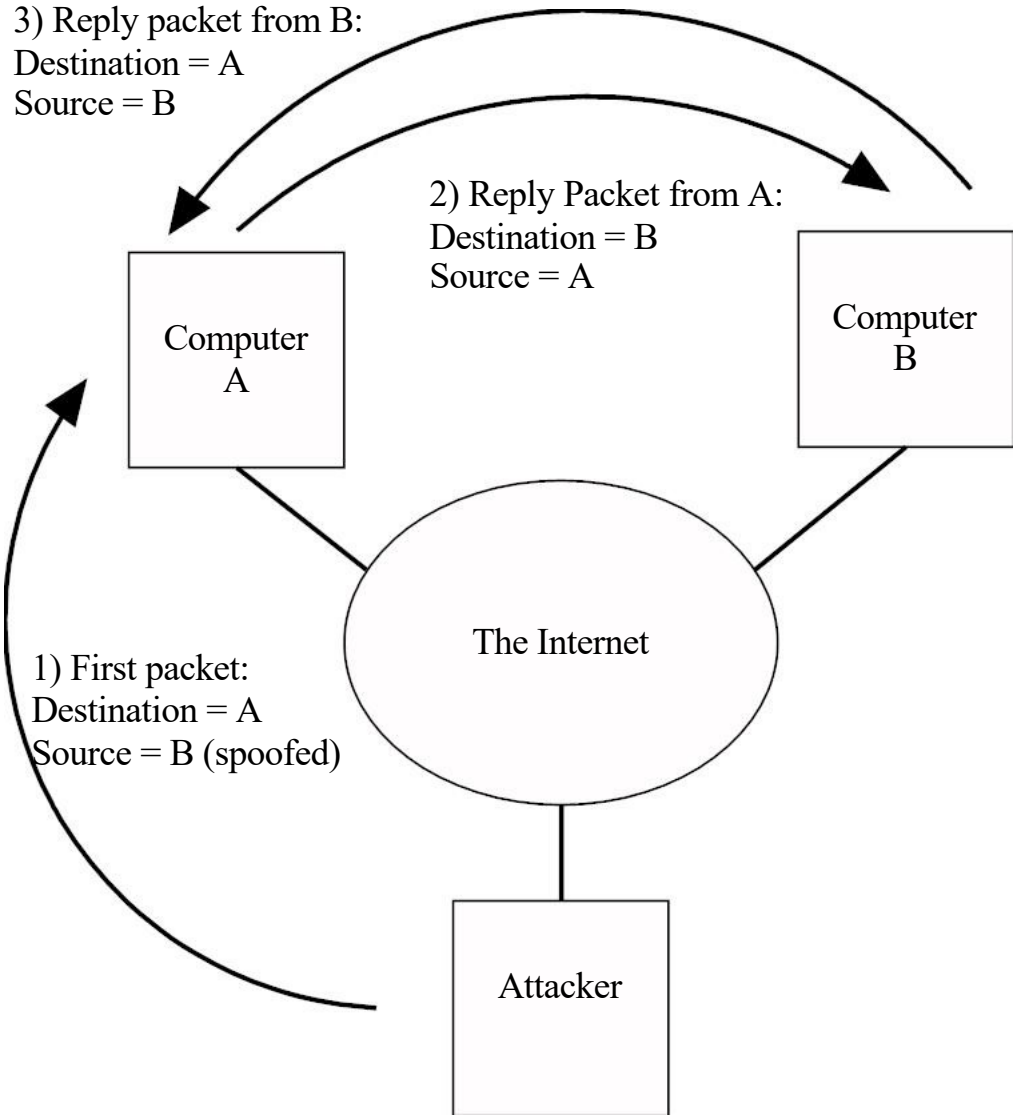
Differences between networks

Differences	Remediation
Physical network layer addressing schemes	The network will need to adapt to the different address types which is more complex in devices like routers
Maximum and minimum packet sizes	The network layer will need to implement segmentation and reassembly
Network access methods	The network layer will need to provide buffering which handle different access methods, especially in a router
Error and flow control	The network layer will need to handle lost and delayed packets
Machine and user authentication	The network layer will need to provide authentication to the physical network if required

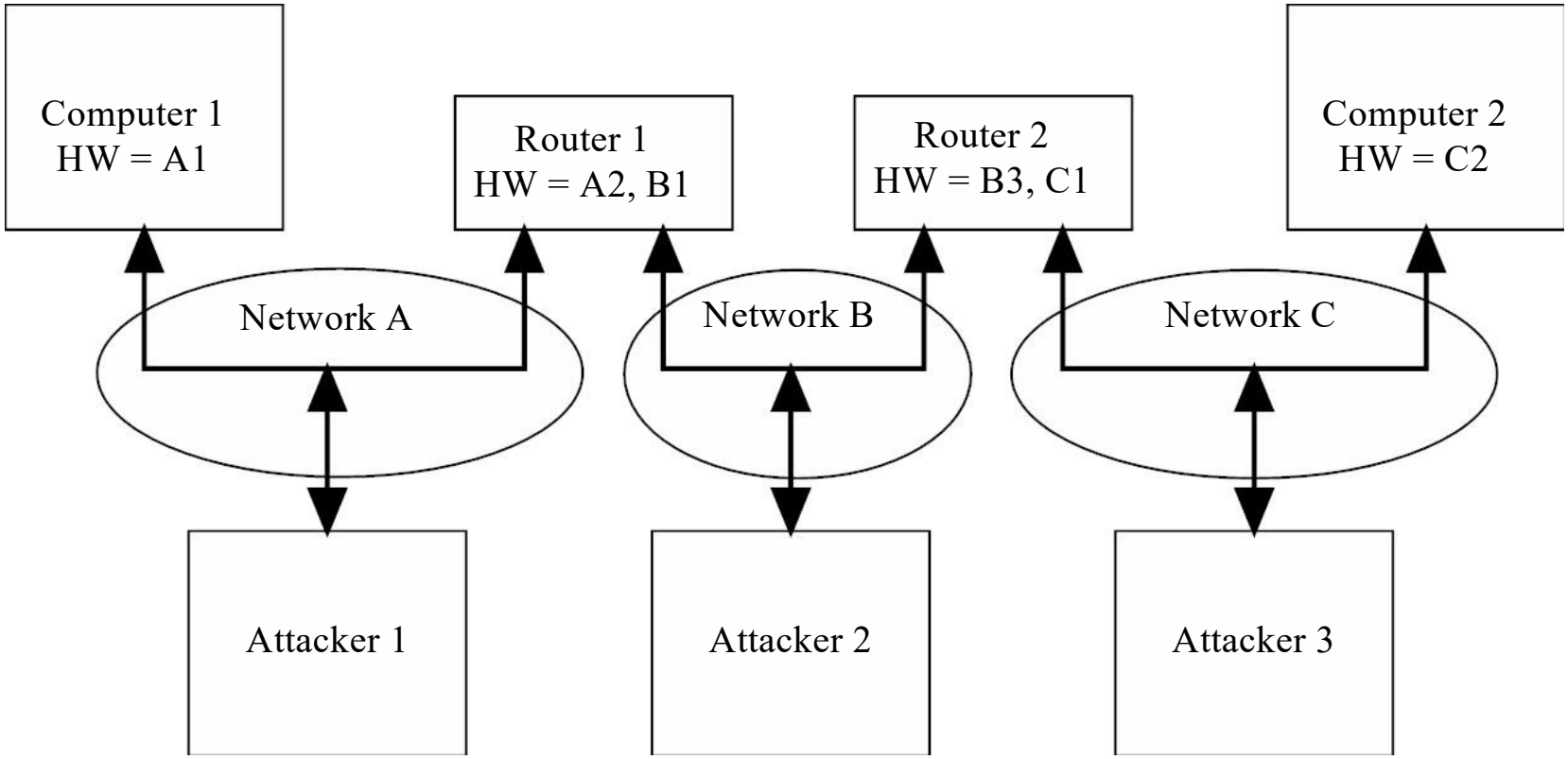
Common Attack Methods

- Address Spoofing
- Network Sniffing
- Network Scanning

Address Spoofing



Network Sniffing



Network Scanning

- Network layer is a global address space
- You can use the network layer protocols to locate targets

IP Layer Topics

- 1. Addressing
- 2. Routing
- 3. Packet Formats
- 4. ICMP Internet Control Message Protocol

Addressing

- 1. IP addresses
- 2. Name to IP addresses translation
- 3. IP address to station datalink address

IP Addresses

- Globally unique
- Two parts
  - Network address
  - Host address

Example IP addresses

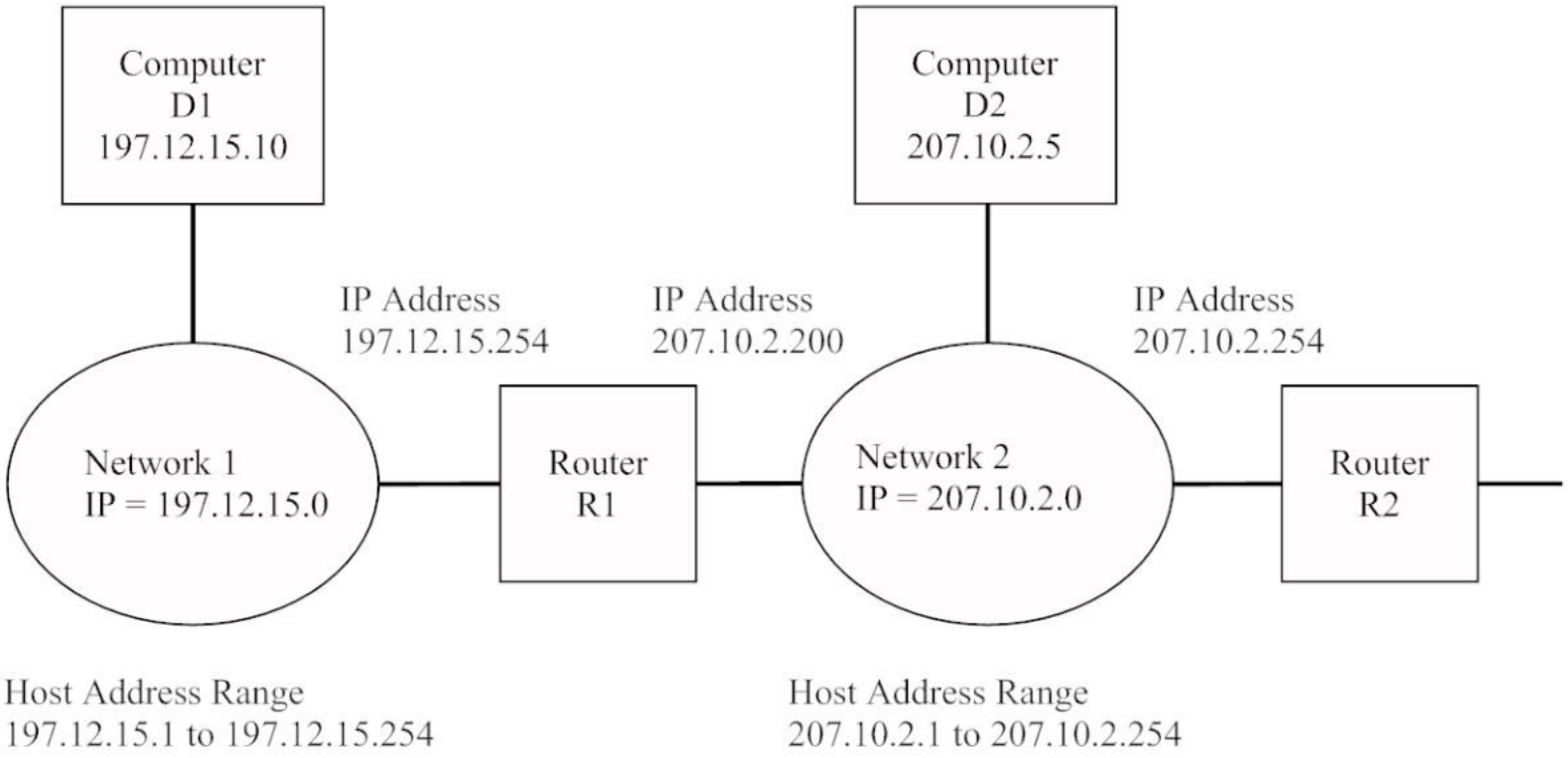


Figure 6.3 Example IP Addresses

IP addresses

- The IP address is written as a four-tuple where each tuple is in decimal and are separated by a "." (called a dot). When talking about an address you pronounce the word dot. So 129.186.5.102 is pronounced 129 dot 186 dot 5 dot 102

IP Addresses

A	0 + Netid (7 bits)	Host ID (24 bits)	
B	10 + NetID (14 bits)	Host ID (16 bits)	
C	110 + Net ID (21 Bits)		Host ID (8 bits)
D	1110 + Multicast address		
E	11110 Reserved		

IP Address Allocation

Class	# of Addresses	%
A	2^31 = 2,147,483,648	50%
B	2^30 = 1,073,741,824	25%
C	2^29 = 536,870,912	12.5%
D	2^28 = 268,435,456	6.25%
E	2^28 = 268,435,456	6.25%

IP Address Distribution

Class	First network	Last network	# of Networks	# of hosts per network
A	1.0.0.0	126.0.0.0	126	16,777,214
B	128.0.0.0	191.255.0.0	16,384	65,534
C	192.0.0.0	223.255.255.0	2,097,152	254

IP Address Space

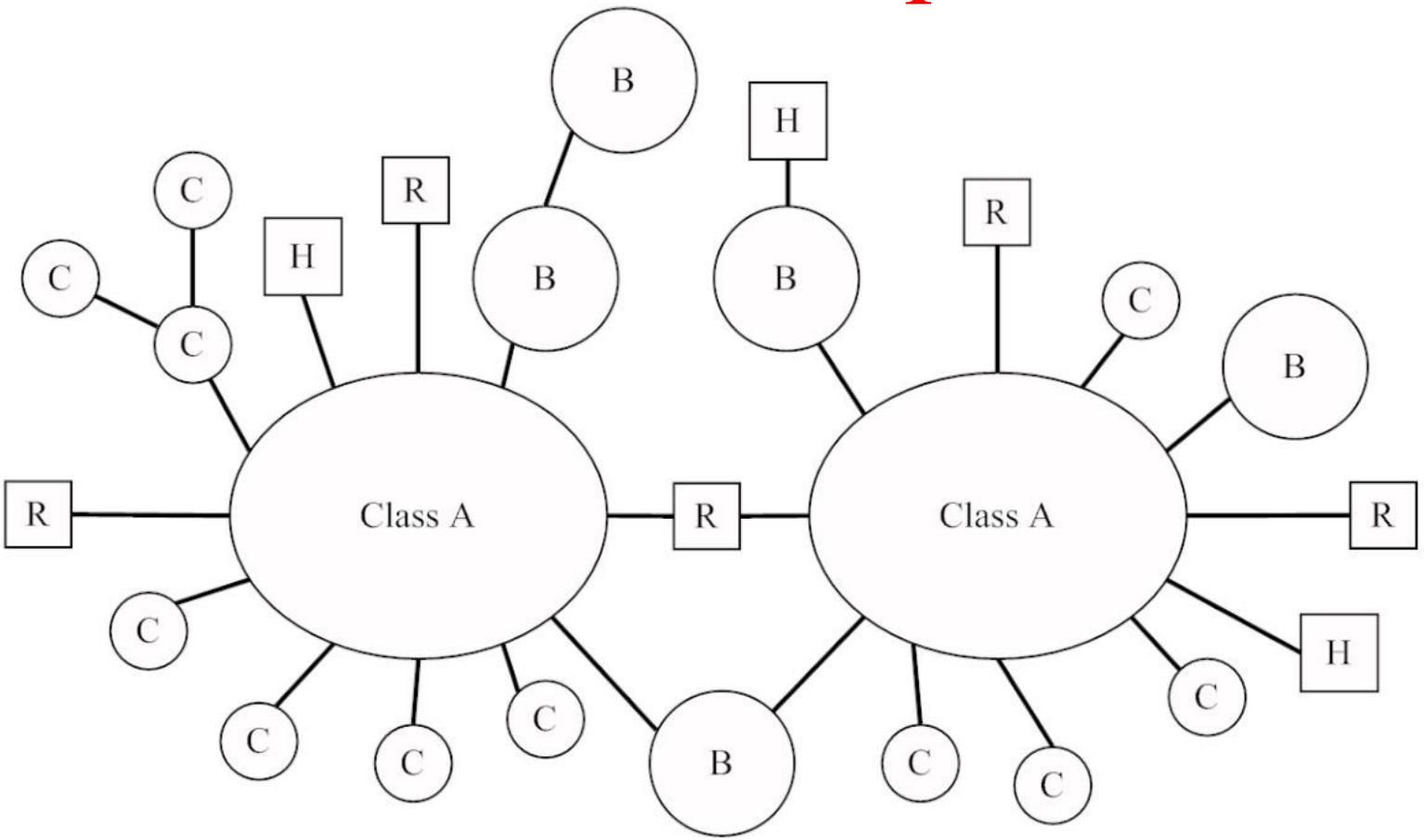
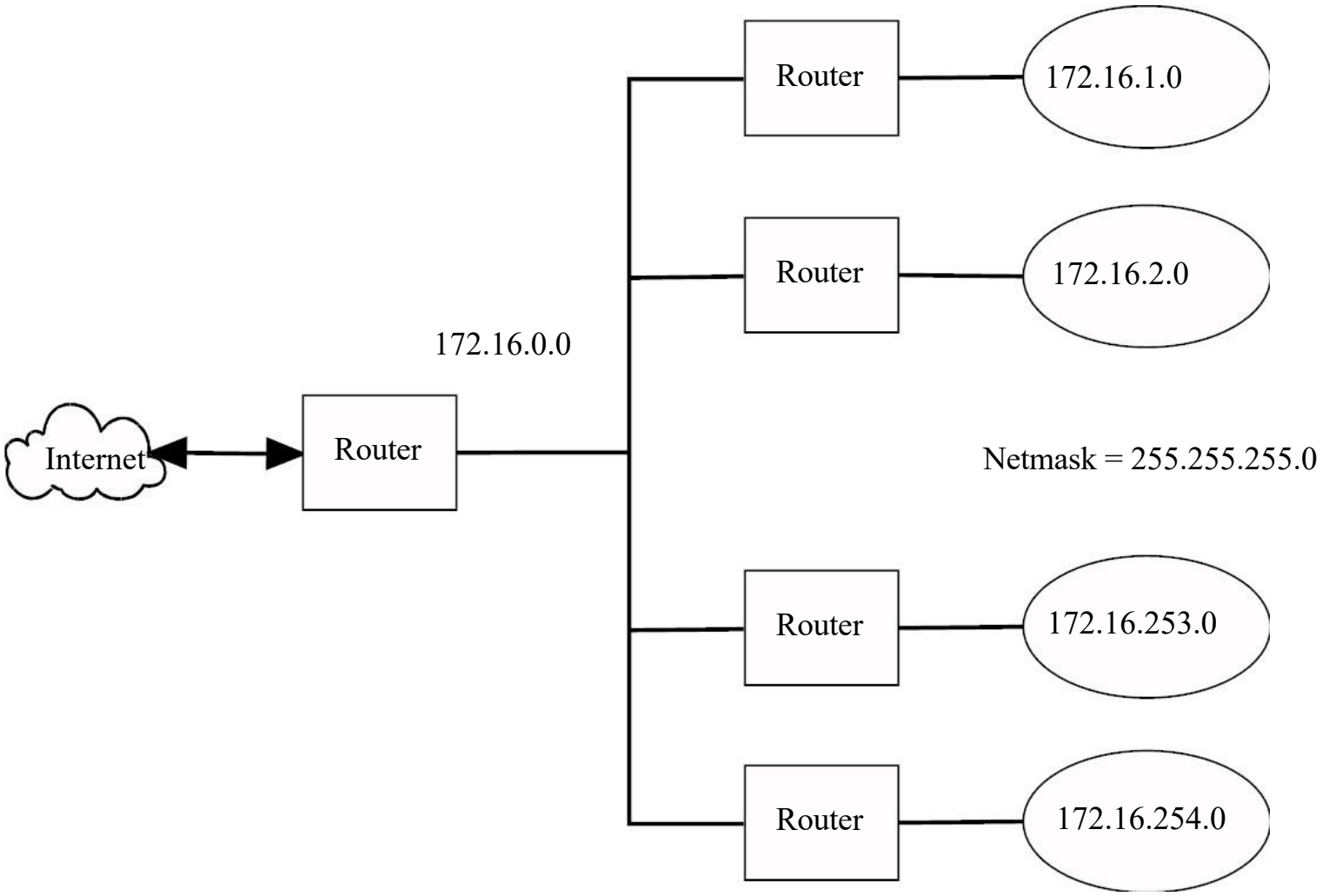


Figure 6.4 IP Address Space

Special Addresses

Network Part	Host Part	Purpose
Network	All 0s	Network address—Not used in the packet.
Network	All 1s	Directed broadcast—Destination address only.
All 1s	All 1s	Broadcast address—Destination address only.
All 0s	All 0s	This host on this network—Source address only.
All 0s	Host	A specific host on this network—Destination address only.
127	Any	Loopback address

Subnets


Router
172.16.1.0
Router
172.16.2.0
172.16.0.0
Internet
Router
Netmask = 255.255.255.0
Router
172.16.253.0
Router
172.16.254.0

Classless Addresses CIDR

Class	Netmask	Example CIDR address
A	255.0.0.0	15.35.26.234/8
B	255.255.0.0	129.186.34.54/16
C	255.255.255.0	192.168.1.30/24

Routing

- All hosts and gateways store routing tables
- Each row in the route table contains:
  - Destination address or address range
  - Next hop for that destination address range
  - The physical interface to use for that address range. (i.e.: which Ethernet card to use)

Example:	Destination	Next	Interface
	129.186.4.0	129.186.5.254	en0

Routing

In order to route a packet:

1. IP layer finds the route table entry where the destination address matches the range given in the table.
2. If the next hop falls within the local network, the packet is sent directly to the destination. Otherwise the packet is sent to the next hop.

Next Hop Routing

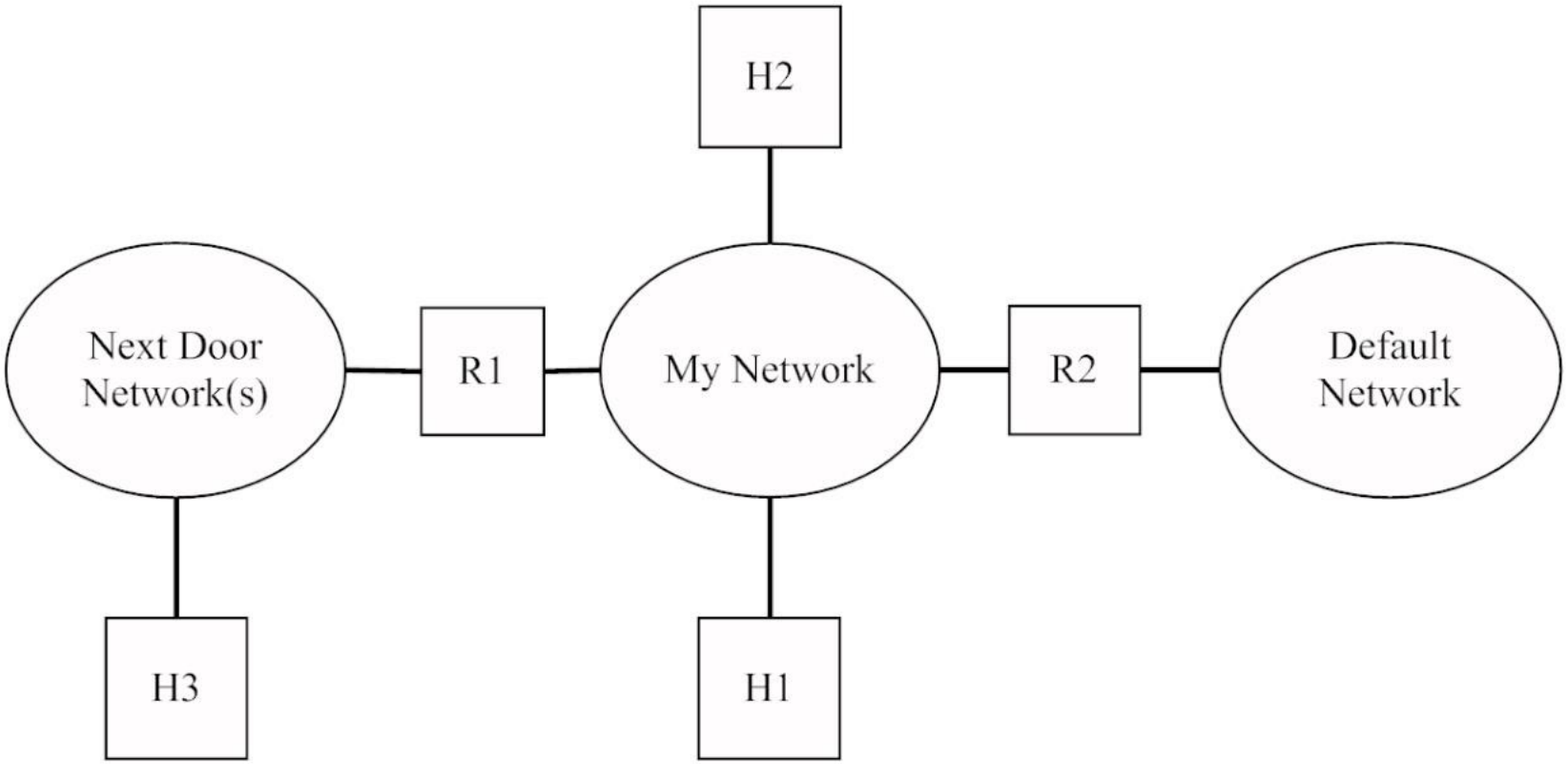


Figure 6.6 IP Next Hop Routing

Routing

Netmask

- Determines which part of the IP address is network and which part is host
- Allows for the ability to create subnetworks
- Example: a netmask of 255.255.255.0 indicates that the first 3 bytes of the IP address is the network, and the last 8 bytes is the host.
- The above netmask allows for 254 subnetworks each with up to 254 attached hosts.
- The following are examples of subnetworks:
  - 129.186.5.0 129.186.15.0 129.186.55.0

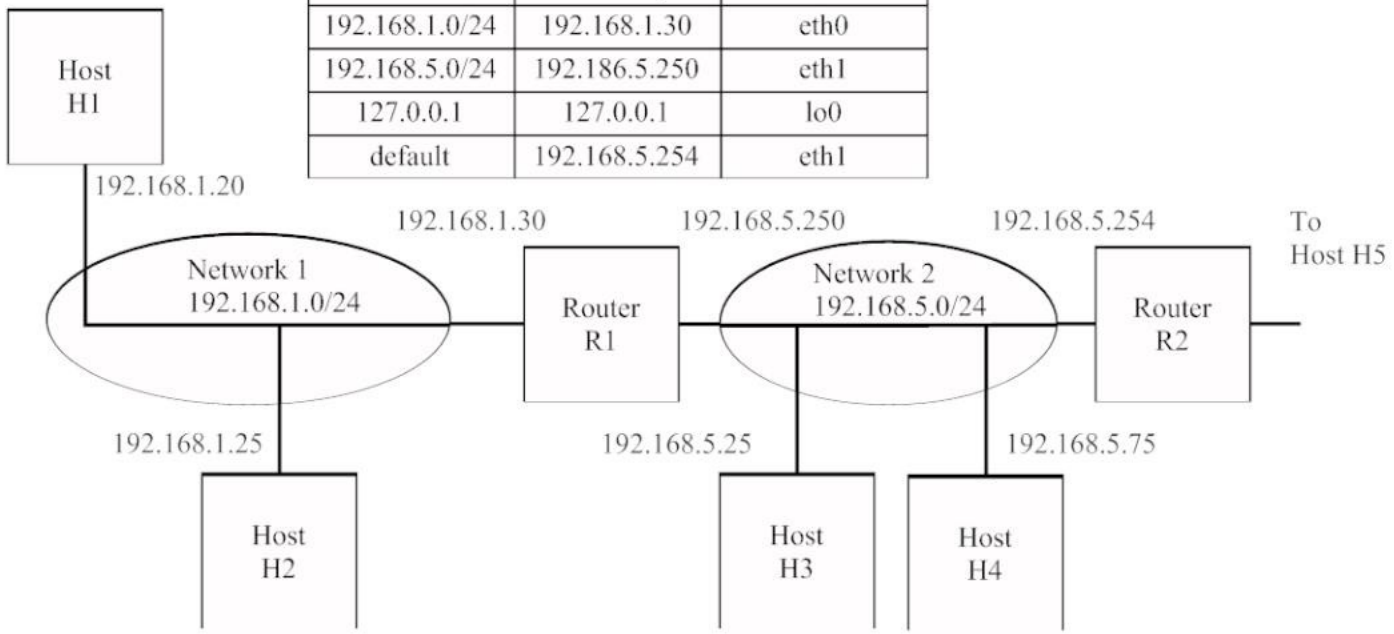
Routing

We will study routing using three scenarios:

1. A simple network with only one router
2. A network with multiple routers

Route Table Router R1

Destination	Next Hop	Interface
192.168.1.0/24	192.168.1.30	eth0
192.168.5.0/24	192.186.5.250	eth1
127.0.0.1	127.0.0.1	lo0
default	192.168.5.254	eth1



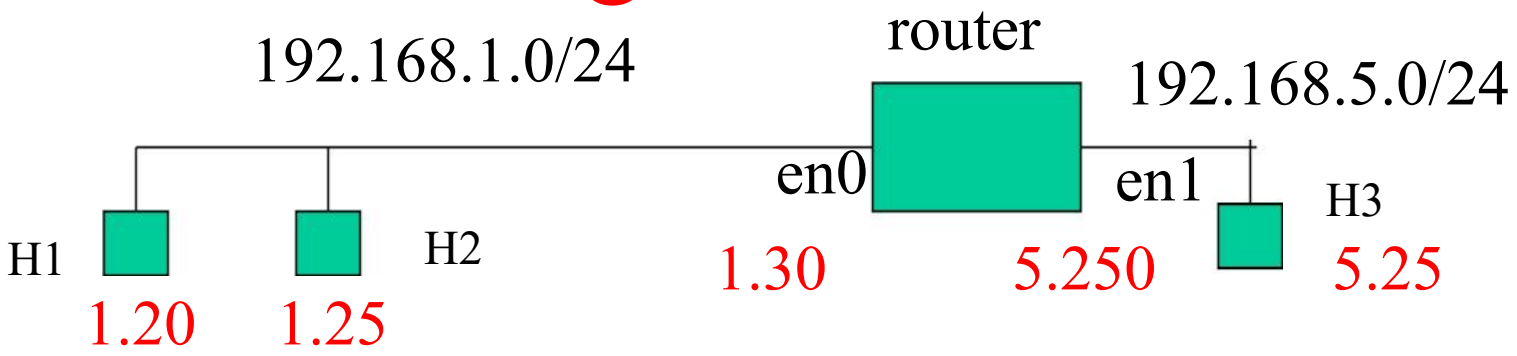
Route Table Host H1

Destination	Next Hop	Interface
192.168.1.0/24	192.168.1.20	eth0
127.0.0.1	127.0.0.1	lo0
default	192.168.1.30	eth0

Route Table Host H3

Destination	Next Hop	Interface
192.168.5.0/24	192.168.5.25	eth0
192.168.1.0/24	192.186.5.250	eth0
127.0.0.1	127.0.0.1	lo0
default	192.168.5.254	eth0

Routing Scenario

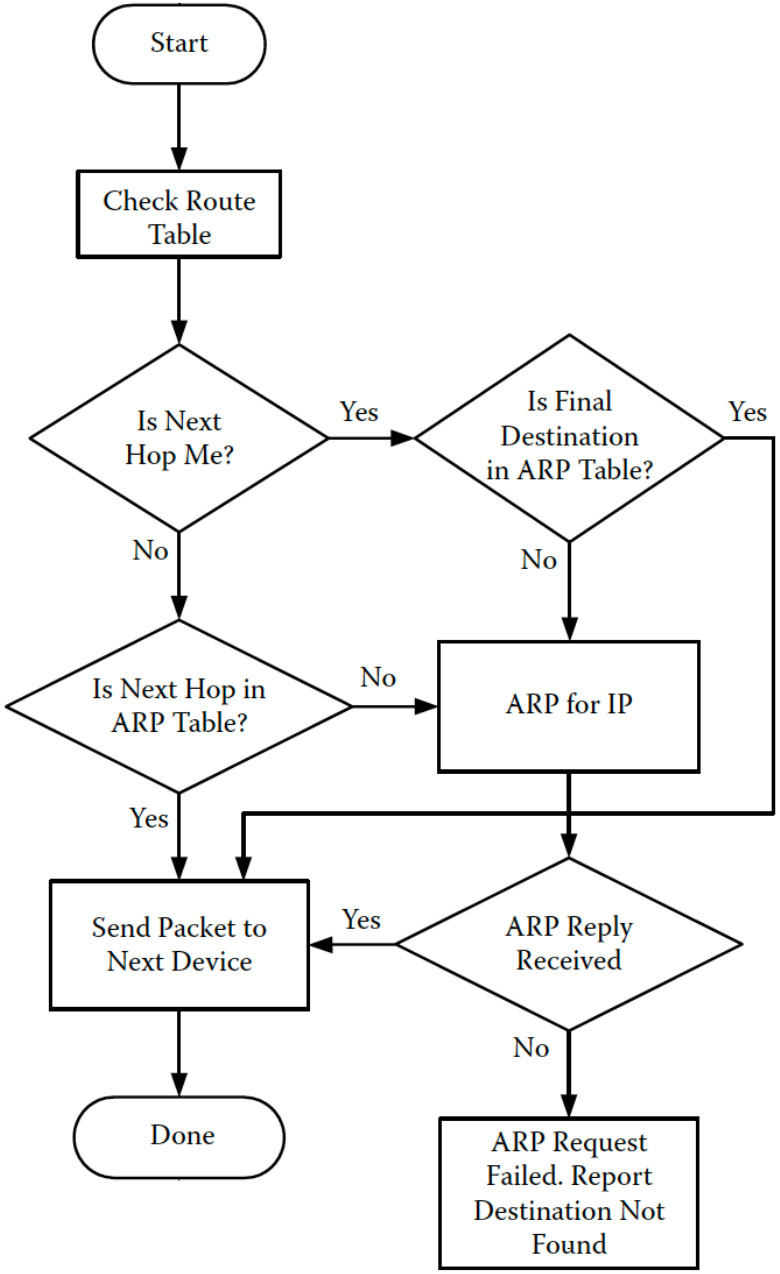


Packet from H1 to H2 (same network)

IP Address		Hardware Address	
SRC	DEST	SRC	DEST
H1	H2	H1	H2

Packet from H1 to H3 (Next door network)

IP Address		Hardware Address	
SRC	DEST	SRC	DEST
H1	H3	H1	R1 (EN0)
H1	H3	R1 (EN1)	H3


Start
Check Route Table
Is Next Hop Me?
Yes
Is Final Destination in ARP Table?
Yes
No
No
Is Next Hop in ARP Table?
No
ARP for IP
Yes
Send Packet to Next Device
Yes
ARP Reply Received
No
Done
ARP Request Failed. Report Destination Not Found

Routing Scenario 1

Steps involved in sending a packet from H1 to H2:

Destination	Next Hop
192.168.1.0/24	192.168.1.20
Default	192.168.1.30

1. Route table is checked.
   192.168.1.25/24 matches the 192.168.1.0 entry
2. The next hop is the host itself (192.168.1.20). This means the destination is on the local network.
3. H1 then sends an ARP packet to find the data link address of the destination
4. Once the data link address is found, the packet is sent

Routing Scenario 2

Steps involved in sending a packet from H1 to an address that is on another network:

1. Route table is checked. The destination address matches the default entry in the table.
2. The next hop is 192.168.1.30. This means the destination is on the other side of a router.
3. H1 sends an ARP packet to determine the data link address of the gateway.
4. The packet is sent to the router
5. The router's route table is checked and the packet is sent to the next hop
6. This continues until the packet reaches the final destination

IP Packet Format

VER=4	IHL	TYPE	TOTAL LENGTH (bytes)	
ID			FLAG	OFFSET
TTL		PROTOCOL	CHECKSUM	
SOURCE IP				
DESTINATION IP				
OPTION				
DATA ….				

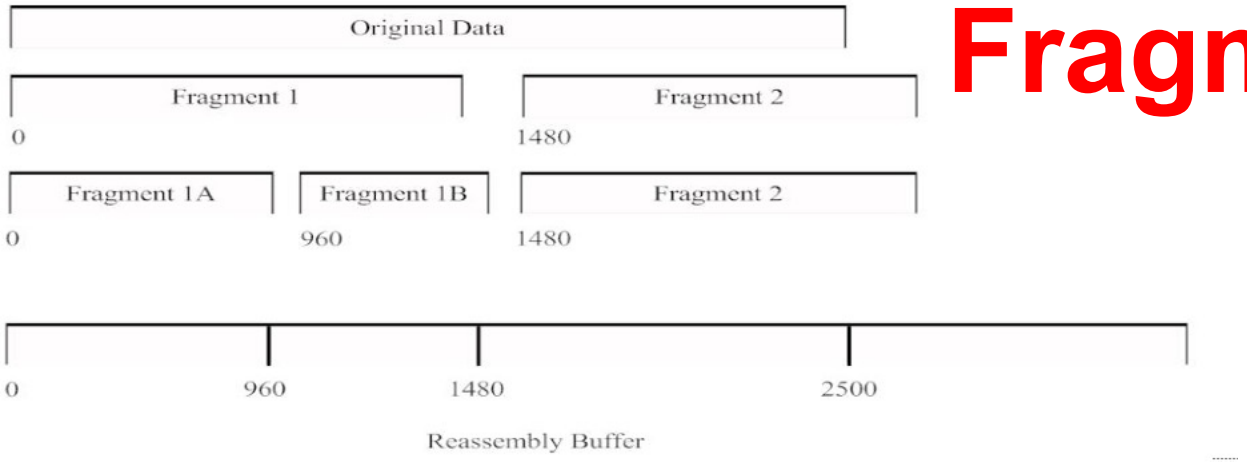
IP Packet Format

- IHL: header length in words
- Type of service: almost always 0
- Total length (bytes) includes header length.
  Max packet size = 2^{11} bytes
- ID: used in fragmentation
- Flag: 0: not used
  D=1: don't fragment
  M=:1 more data. M=0: last packet of fragment
- Offset: #8 bytes
- TTL (time to live): starts at 255 then decrements after each hop
- Checksum: worthless because it must be recalculated after every router due to the TTL decrement

IP Protocol Field

1	Internet Control Message Protocol (ICMP)
3	Gateway-to-Gateway protocol
5	Stream
6	Transport Control Protocol (TCP)
8	Exterior Gateway Protocol
9	Any private interior gateway protocol
11	Network voice protocol
17	User datagram protocol (UDP)
20	Host Monitoring Protocol
22	Xerox Network System Internet Datagram Protocol
27	Reliable Datagram Protocol
28	Internet Reliable Transaction Protocol
30	Bulk Data Transfer Protocol
61	Any Host Internet Protocol

Fields	Original Packet	Fragment 1	Fragment 2	Fragment 1a	Fragment 1b
Ver/HLEN	4/5	4/5	4/5	4/5	4/5
Type	0	0	0	0	0
Length	2520	1500	1040	980	540
ID	2356	2356	2356	2356	2356
Flags	0	0 0 1	0 0 0	0 0 1	0 0 1
Offset	0	0	185	0	120
TTL	150	Computed	Computed	Computed	Computed
Protocol	TCP	TCP	TCP	TCP	TCP
Checksum	Computed	Computed	Computed	Computed	Computed
Source IP	IP1	IP1	IP1	IP1	IP1
Dest IP	IP2	IP2	IP2	IP2	IP2
Data Len	2500	1480	1020	960	520



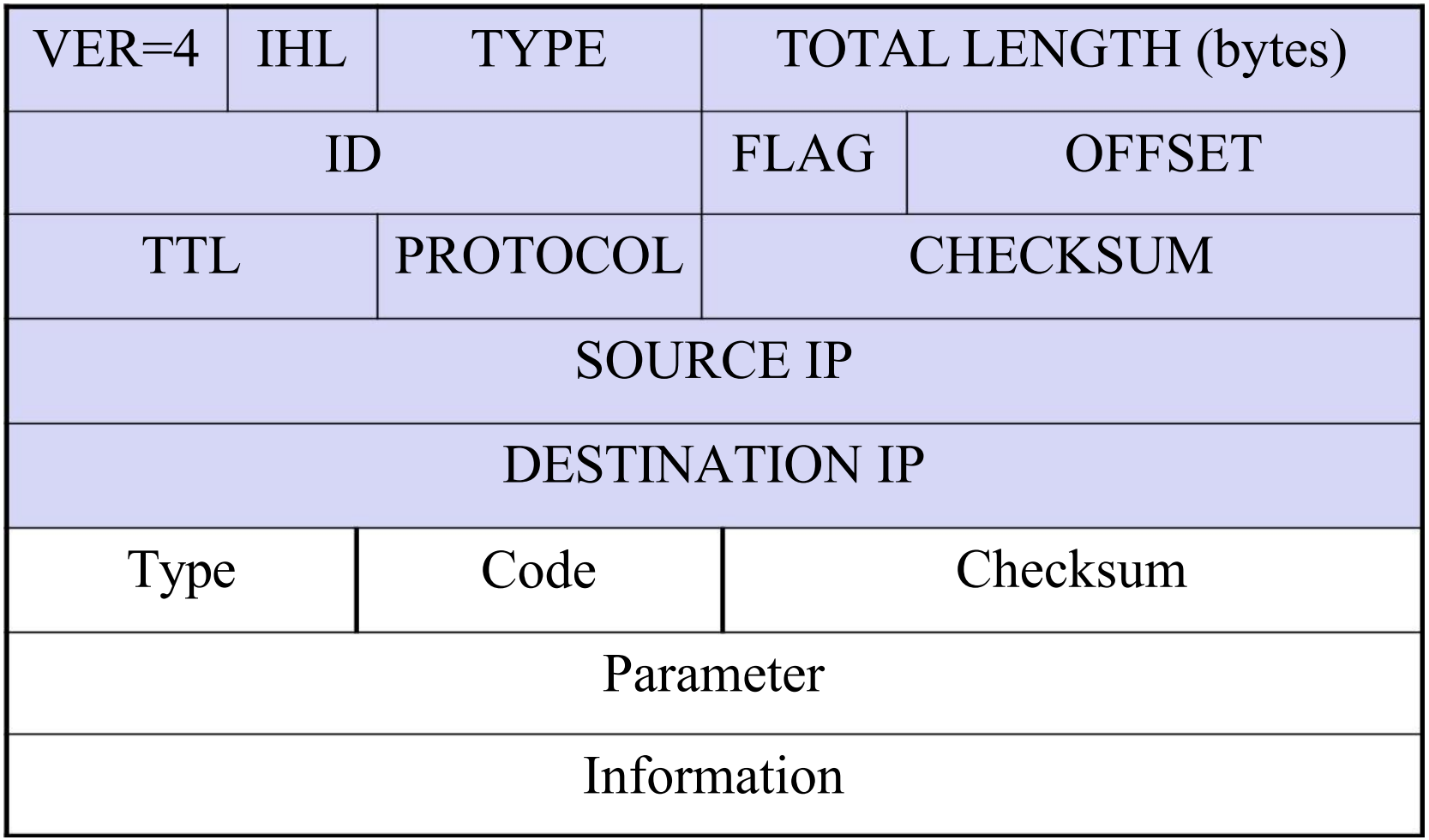
Fragmentation

ICMP

Internet Control Message Protocol

- Designed as error control
- Examples for use:
  - When a datagram cannot reach its destination
  - When a gateway can direct the host to send traffic on a shorter route
  - Ping

ICMP Packet Format



ICMP Packet Format

- ICMP packets are carried within the data of an IP packet
- Fields:
  - Type (8 bits): message type
  - Code (8 bits): message sub-type
  - Checksum (16 bits)
  - Parameter (32 bits)
  - Information (variable)

ICMP Message Types

0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply

ICMP Echo (Ping)

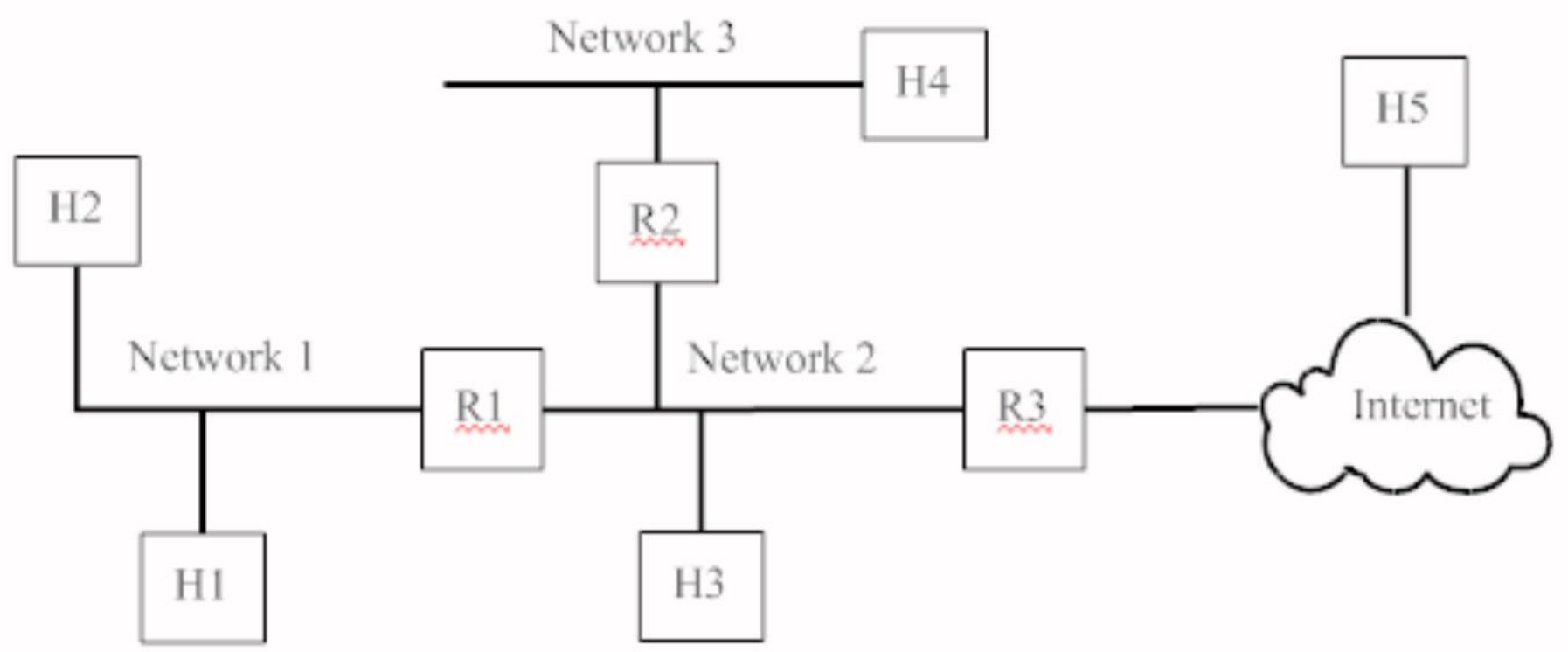
- Type = 8 (echo)
  Type = 0 (reply)
- Code = 0
- Parameter
  - ID number (2 bytes)
  - Sequence number (2 bytes)
- Optional Data

Note: the optional data field of ping has been used in the past for tunneling information through a firewall

ICMP Destination Unreachable

- Type = 3
- Code:
  - 0 Network Unreachable
  - 1 Host Unreachable
  - 2 Protocol Unreachable
  - 3 Port Unreachable
  - 4 Fragmentation needed and DF set
  - 5 Source Route Failed
- Parameter = 0
- Data = IP header + first 8 bytes of datagram

Putting it all together



Route tables

Route Table H1 & H2

Destination	Next Hop
Network 1	Me
default	R1

Route Table H4

Destination	Next Hop
Network 3	Me
default	R2

Route Table H3

Destination	Next Hop
Network 1	R1
Network 2	H3
Network 3	R2
default	R3

Route Table Router R1

Destination	Next Hop	Interface
Network 1	R1	Int 1
Network 2	R1	Int 2
Network 3	R2	Int 2
default	R3	Int 2

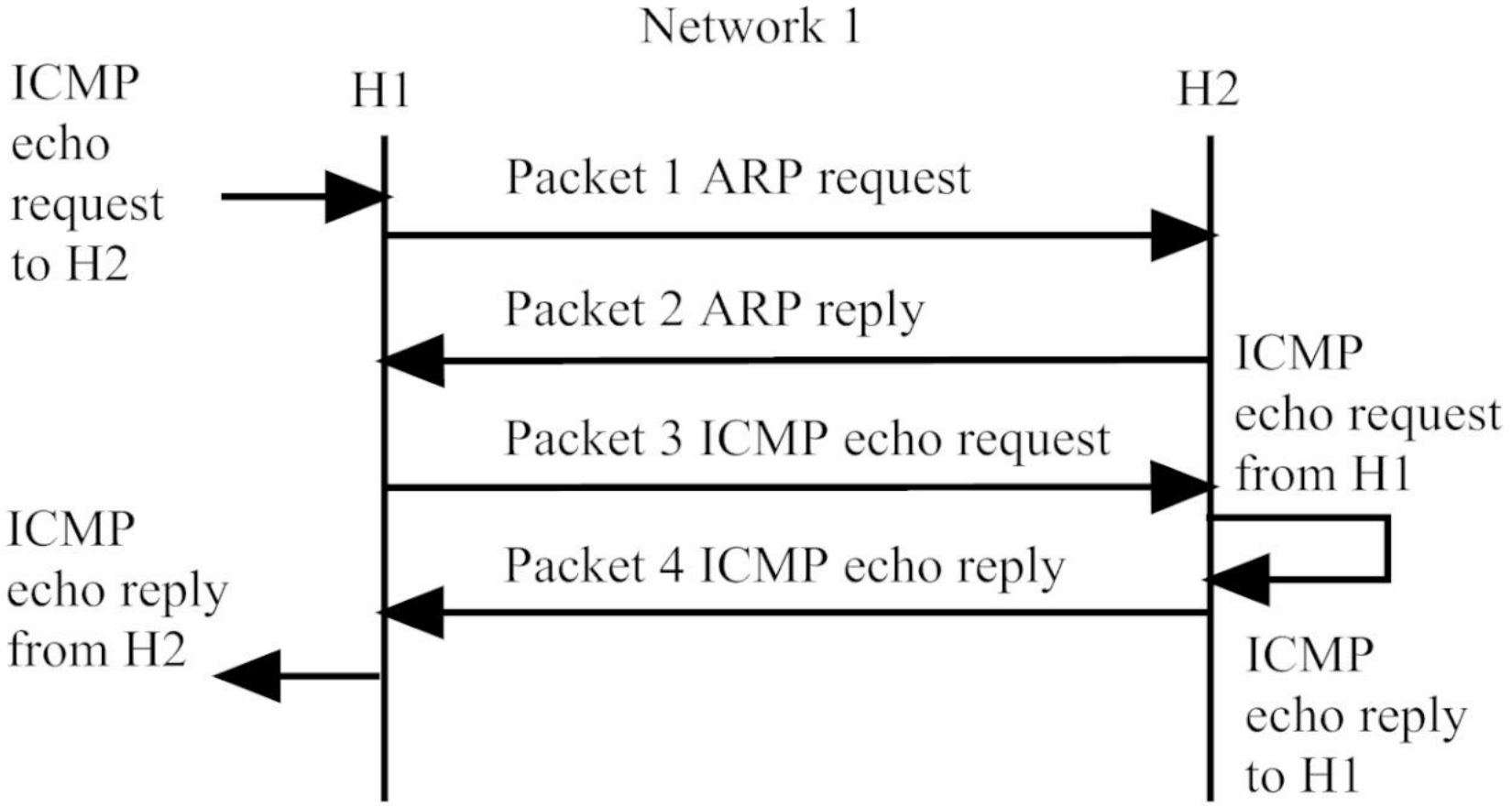
Route Table Router R2

Destination	Next Hop	Interface
Network 1	R1	Int 1
Network 2	R2	Int 1
Network 3	R2	Int 2
default	R3	Int 1

Route Table Router R3

Destination	Next Hop	Interface
Network 1	R1	Int 1
Network 2	R3	Int 1
Network 3	R2	Int 1
default	Next Hop	Int 2

Scenario 1 (H1 to H2)



Scenario 1 (H1 to H2)

Packet	Hardware Addresses		IP Addresses		Payload
	DST	SRC	DST	SRC	
1	Broadcast	H1	N/A	N/A	ARP
2	H1	H2	N/A	N/A	ARP
3	H2	H1	H2	H1	ICMP
4	H1	H2	H1	H2	ICMP

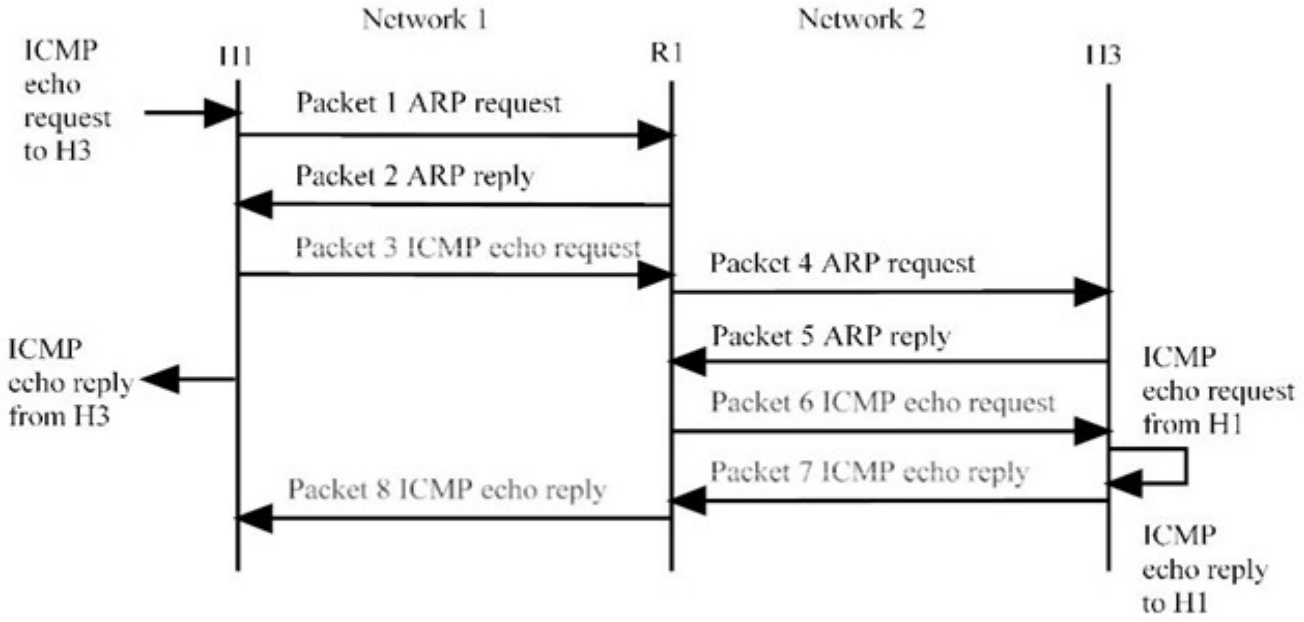
ARP table for H1

Time	Destination	HW Address
Start	Empty	Empty
After P2	H2	H2

ARP table for H2

Time	Destination	HW Address
Start	Empty	Empty
After P1	H1	H1

Scenario 2 (H1 to H3)



Scenario 2 H1 to H3)

Packet	Hardware Addresses		IP Addresses		Payload
	DST	SRC	DST	SRC	
1	Broadcast	H1	N/A	N/A	ARP
2	H1	R1 (Int 1)	N/A	N/A	ARP
3	R1 (Int 1)	H1	H3	H1	ICMP
4	Broadcast	R1 (Int 2)	N/A	N/A	ARP
5	R1 (Int 2)	H3	N/A	N/A	ARP
6	H3	R1 (Int 2)	H3	H1	ICMP
7	R1 (Int 2)	H3	H1	H3	ICMP
8	H1	R1 (Int 1)	H1	H3	ICMP

ARP table for H1

Time	Destination	HW Address
Start	H2	H2
After P2	R1	R1 (Int 1)

ARP table for H3

Time	Destination	HW Address
Start	Empty	Empty
After P4	R1	R1 (Int 2)

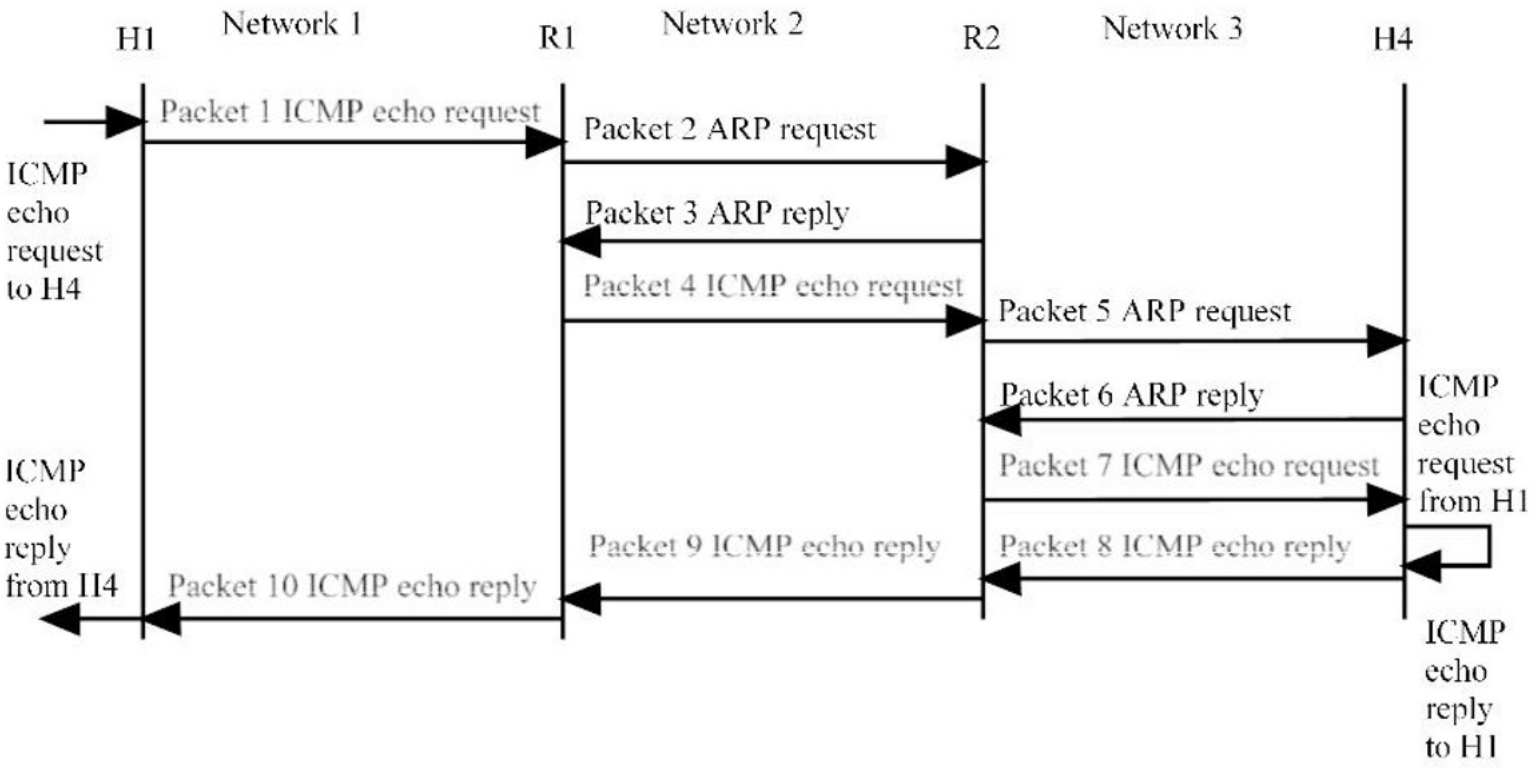
ARP table for R1 (int 1)

Time	Destination	HW Address
Start	Empty	Empty
After P1	H1	H1

ARP table for R1 (int 2)

Time	Destination	HW Address
Start	Empty	Empty
After P5	H3	H3

Scenario 3 (H1 to H4)



Scenario 3 H1 to H4)

Packet	Hardware Addresses		IP Addresses		Payload
	DST	SRC	DST	SRC	
1	R1 (Int 1)	H1	H4	H1	ICMP
2	Broadcast	R1 (Int 2)	N/A	N/A	ARP
3	R1 (Int 2)	R2 (Int 1)	N/A	N/A	ARP
4	R2 (Int 1)	R1 (Int 2)	H4	H1	ICMP
5	Broadcast	R2 (Int 2)	N/A	N/A	ARP
6	R2 (Int 2)	H4	N/A	N/A	ARP
7	H4	R2 (Int 2)	H4	H1	ICMP
8	R2 (Int 2)	H4	H1	H4	ICMP
9	R1 (Int 2)	R2 (Int 1)	H1	H4	ICMP
10	H1	R1 (Int 1)	H1	H4	ICMP

ARP table for H1

Time	Destination	HW Address
Start	H2	H2
	R1	R1 (Int 1)

ARP table for H4

Time	Destination	HW Address
Start	Empty	Empty
After P5	R2	R2 (Int 2)

ARP table for R1 (int 1)

Time	Destination	HW Address
Start	H1	H1

ARP table for R1 (int 2)

Time	Destination	HW Address
Start	H3	H3
After P3	R2	R2 (Int 1)

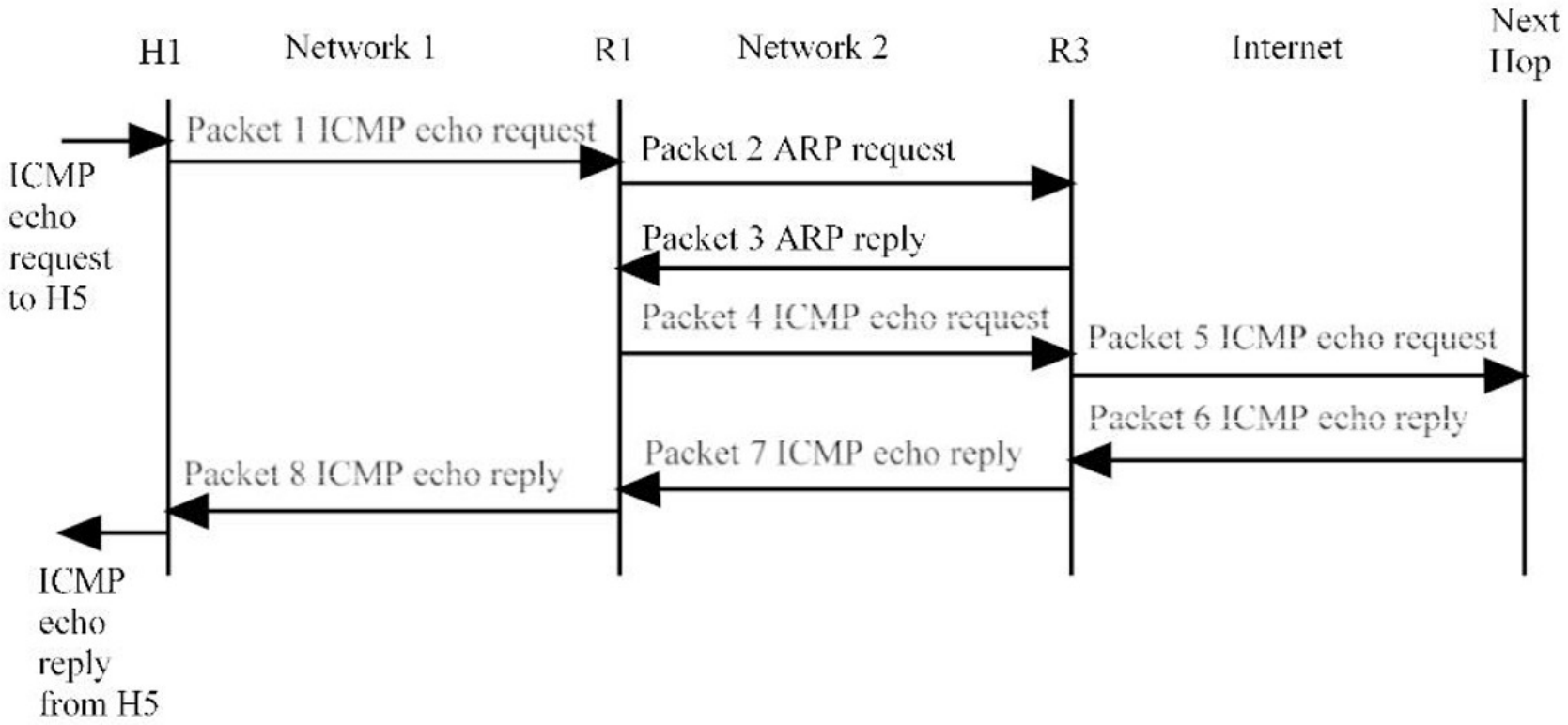
ARP table for R2 (int 1)

Time	Destination	HW Address
Start	Empty	Empty
After P2	R1	R1 (Int 2)

ARP table for R2 (int 2)

Time	Destination	HW Address
Start	Empty	Empty
After P6	H4	H4

Scenario 4 (H1 to H5)



Scenario 4 (H1 to H5)

Packet	Hardware Addresses		IP Addresses		Payload
	DST	SRC	DST	SRC	
1	R1 (Int 1)	H1	H5	H1	ICMP
2	Broadcast	R1 (Int 2)	N/A	N/A	ARP
3	R1 (Int 2)	R3 (Int 1)	N/A	N/A	ARP
4	R3 (Int 1)	R1 (Int 2)	H5	H1	ICMP
5	Next hop	R3 (Int 2)	H5	H1	ICMP
6	R3 (Int 2)	Next hop	H1	H5	ICMP
7	R1 (Int 2)	R3 (Int 1)	H1	H5	ICMP
8	H1	R1 (Int 1)	H1	H5	ICMP

ARP table for H1

Time	Destination	HW Address
Start	H2	H2
	R1	R1 (Int 1)

ARP table for H4

Time	Destination	HW Address
Start	Empty	Empty
	R2	R2 (Int 2)

ARP table for R1 (int 1)

Time	Destination	HW Address
Start	H1	H1

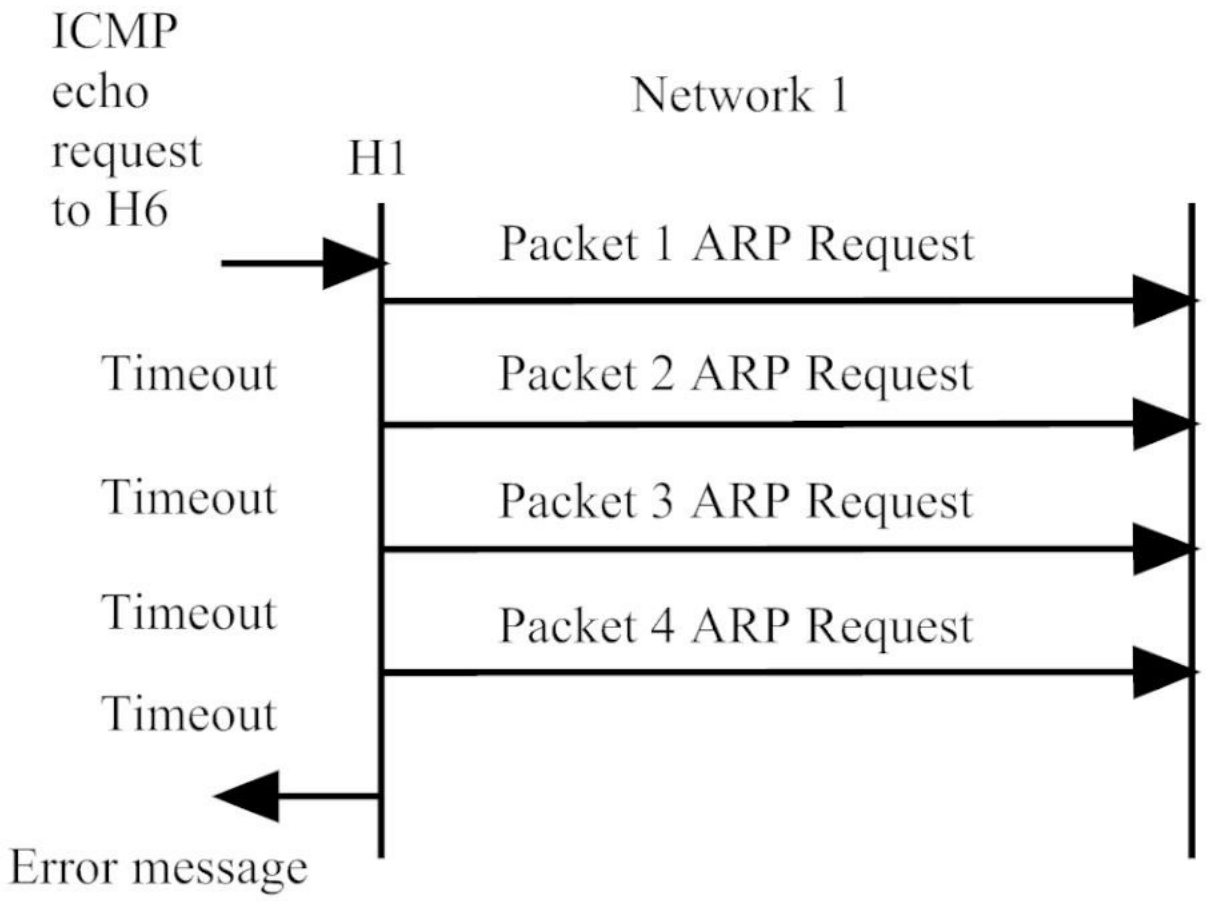
ARP table for R1 (int 2)

Time	Destination	HW Address
Start	H3	H3
	R2	R2 (Int 1)
After P3	R3	R3 (Int 1)

ARP table for R3 (int 1)

Time	Destination	HW Address
Start	Empty	Empty
After P2	R1	R1 (Int 2)

Scenario 5 (H1 to no host on net 1)



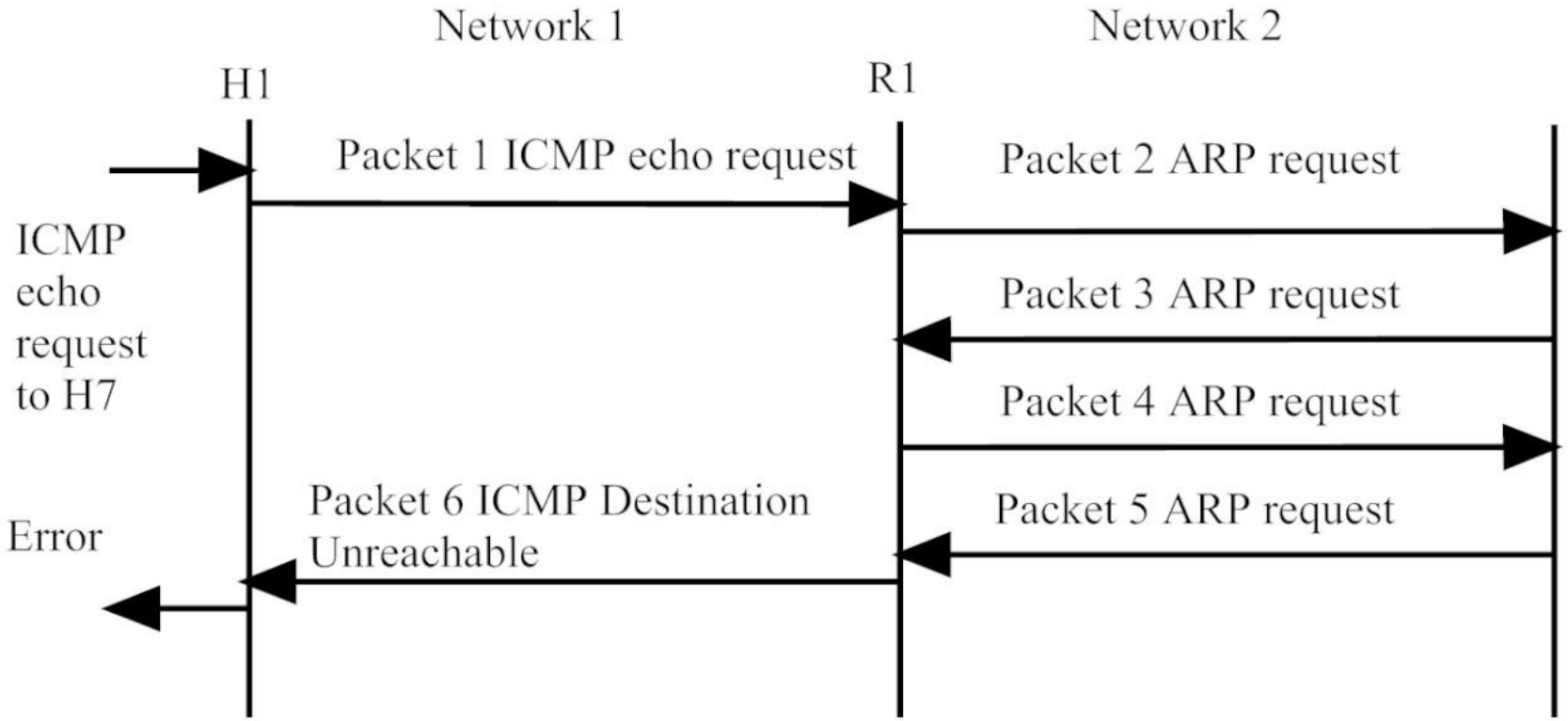
Scenario 5 (H1 to no host on net 1)

Packet	Hardware Addresses		IP Addresses		Payload
	DST	SRC	DST	SRC	
1	Broadcast	H1	N/A	N/A	ARP
2	Broadcast	H1	N/A	N/A	ARP
3	Broadcast	H1	N/A	N/A	ARP
4	Broadcast	H1	N/A	N/A	ARP

ARP table for H1

Time	Destination	HW Address
Start	H2	H2
	R1	R1 (Int 1)

Scenario 6 (H1 to no host on net 2)



Scenario 6 (H1 to no host on net 2)

Packet	Hardware Addresses		IP Addresses		Payload
	DST	SRC	DST	SRC	
1	R1 (Int 1)	H1	H7	H1	ICMP
2	Broadcast	R1 (Int 2)	N/A	N/A	ARP
3	Broadcast	R1 (Int 2)	N/A	N/A	ARP
4	Broadcast	R1 (Int 2)	N/A	N/A	ARP
5	Broadcast	R1 (Int 2)	N/A	N/A	ARP
6	H1	R1 (Int 1)	H1	R1	ICMP

ARP table for H1

Time	Destination	HW Address
Start	H2	H2
	R1	R1 (Int 1)

ARP table for R1 (int 2)

Time	Destination	HW Address
Start	H3	H3
	R2	R2 (Int 1)
	R3	R3 (Int 1)

ARP table for R1 (int 1)

Time	Destination	HW Address
Start	H1	H1

Header Based

- There have been some IP header attacks.
- Most famous is the ping of death
- Fewer ARP and ICMP header attacks

Fragmentation

- If the datagram can be fragmented:
  - The header is copied in each fragment
    - In particular, the “datagram id” is copied in each fragment
  - The “more fragments” flag is set with the exception of the last fragment
  - The “fragmentation offset” field contains the position of the fragment with respect to the original datagram expressed in 8 byte units
  - The “total length field” is changed to match the size of the fragment
- Each fragment is then delivered as a separate datagram
- If one fragment is lost the entire datagram is discarded after a timeout

Ping of Death

```
23:01:06.266646 < 128.111.48.69 > 128.111.48.70: icmp: echo request (frag
    4321:1480@0+)
    23:01:06.421261 < 128.111.48.69 > 128.111.48.70: (frag 4321:1480@1480+)
    23:01:06.575953 < 128.111.48.69 > 128.111.48.70: (frag 4321:1480@2960+)
    23:01:06.730065 < 128.111.48.69 > 128.111.48.70: (frag 4321:1480@4440+)
    23:01:06.884625 < 128.111.48.69 > 128.111.48.70: (frag 4321:1480@5920+)
    23:01:07.038801 < 128.111.48.69 > 128.111.48.70: (frag 4321:1480@7400+)
    23:01:07.193403 < 128.111.48.69 > 128.111.48.70: (frag 4321:1480@8880+)
    23:01:07.348185 < 128.111.48.69 > 128.111.48.70: (frag 4321:1480@10360+)
    23:01:07.502326 < 128.111.48.69 > 128.111.48.70: (frag 4321:1480@11840+)
    [...]
    23:01:12.451121 < 128.111.48.69 > 128.111.48.70: (frag 4321:1480@59200+)
    23:01:12.605235 < 128.111.48.69 > 128.111.48.70: (frag 4321:1480@60680+)
    23:01:12.759927 < 128.111.48.69 > 128.111.48.70: (frag 4321:1480@62160+)
    23:01:12.917811 < 128.111.48.69 > 128.111.48.70: (frag 4321:1480@63640+)
    23:01:13.090936 < 128.111.48.69 > 128.111.48.70: (frag 4321:398@65120)
```

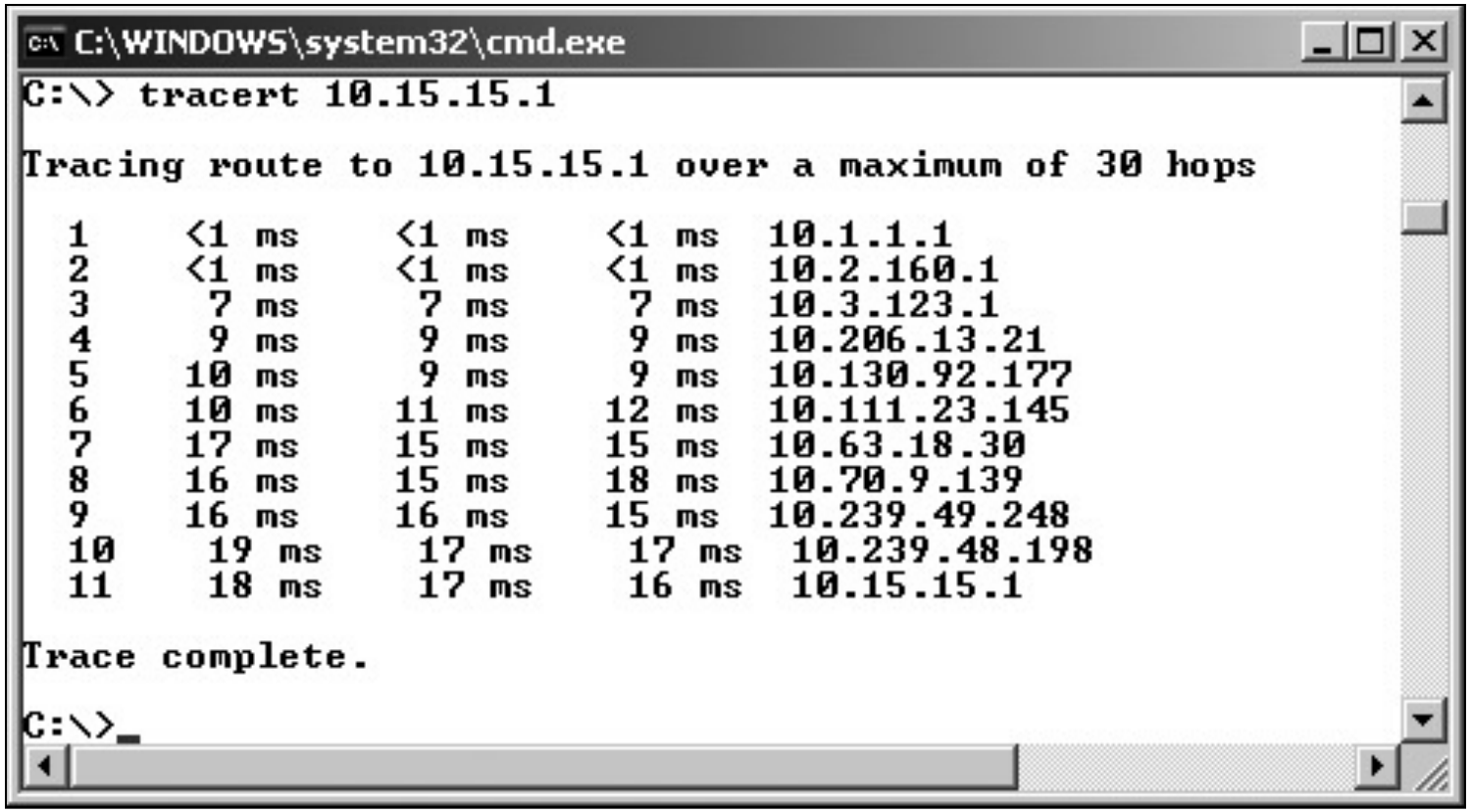
Total 65120 + 398 = 65518 + 20 bytes of header = 65538 > 65535!

Traceroute

- TTL field in IP header
  - Usually decremented by each router
- When TTL reaches 0...
  - Router kills packet
  - Sends ICMP time exceeded msg to source
- Traceroute
  - UNIX/Linux: traceroute uses UDP packets
  - Windows: tracert uses ICMP packets

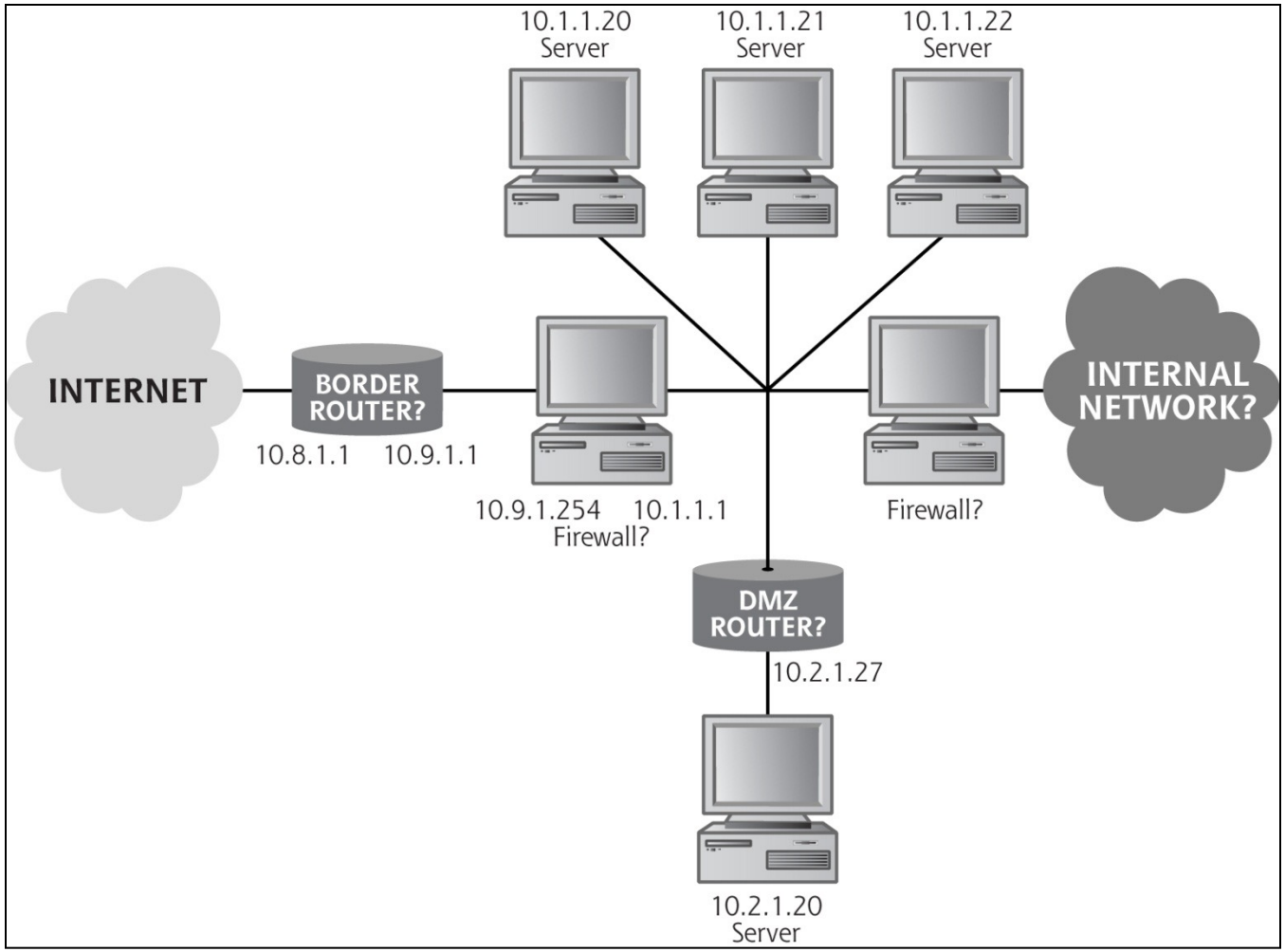
Traceroute

- In Windows



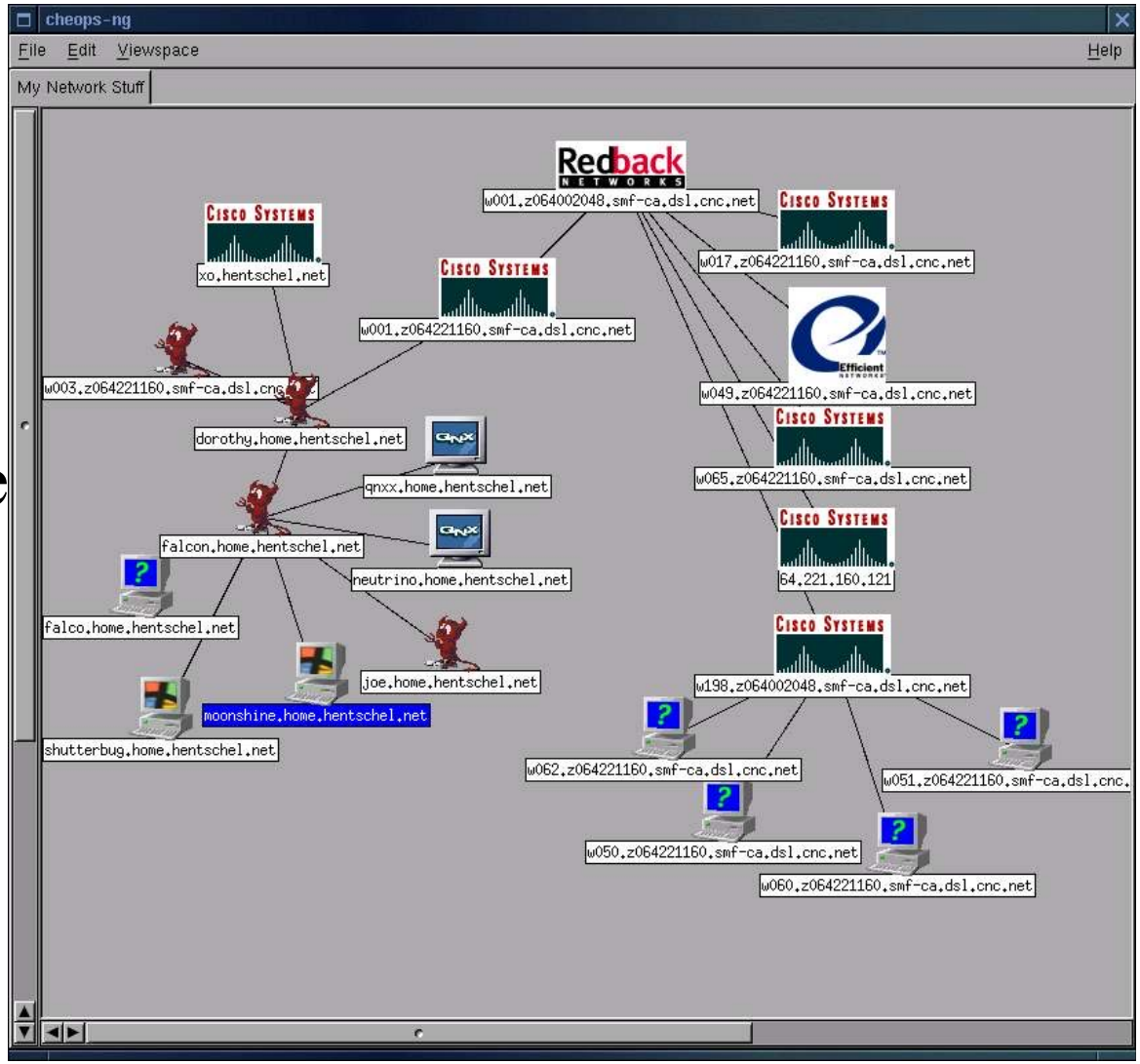
Traceroute

- Might find, for example:



Traceroute

- Cheops-ng
  - Free
  - Pretty pictures
  - Lots of info (type of OS ...)
  - Useful for admins too



Protocol Based

- Even though the IP protocol is simple, the routing is complex.
- There are a large number of protocol based attacks involving sending packets the confuse the receiver or interjects packets into the receiver.
- They work because there is no authentication of the sender and receiver.

Protocol Based

- ICMP:
  - Use icmp error messages to cause service denial or redirect traffic to the wrong place
- ARP
  - ARP cache poisoning (better classified as an authentication attack)

Authentication Based

- This is a big problem, since we often use the IP address as authentication.
- IP Address spoofing

IP Spoofing

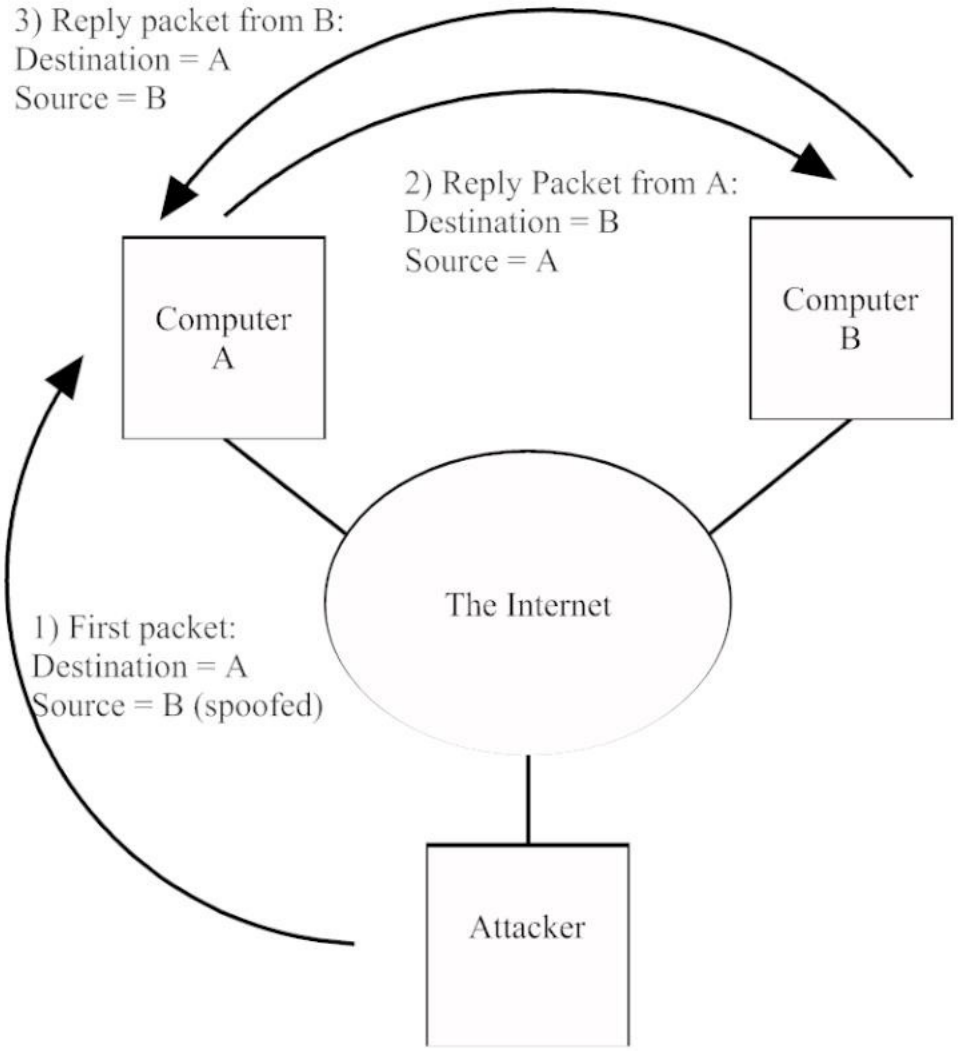


Figure 6.20 IP Address Spoofing

IP Spoofing

- DOS
  - Send a request from one address or multiple addresses
  - Send a direct IP broadcast packet
- Check source IP address before allowing packet into the Internet

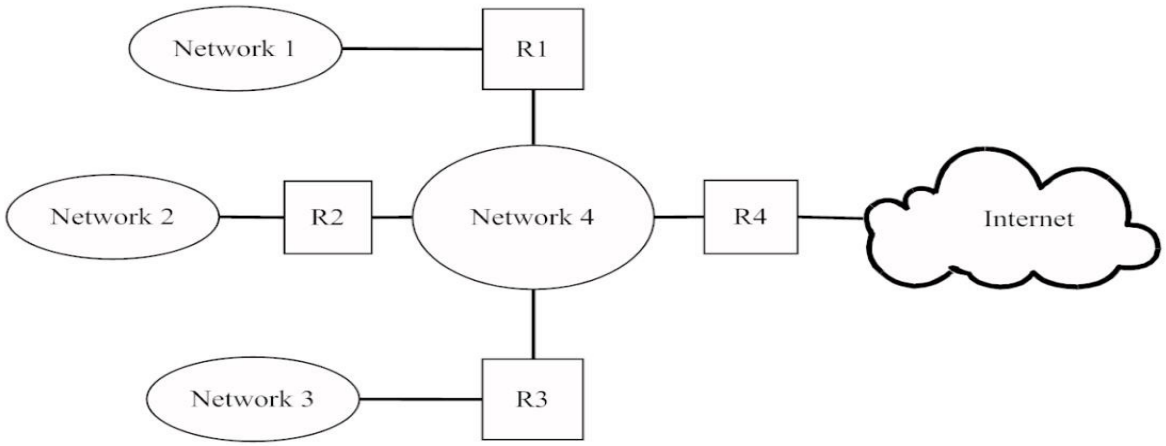
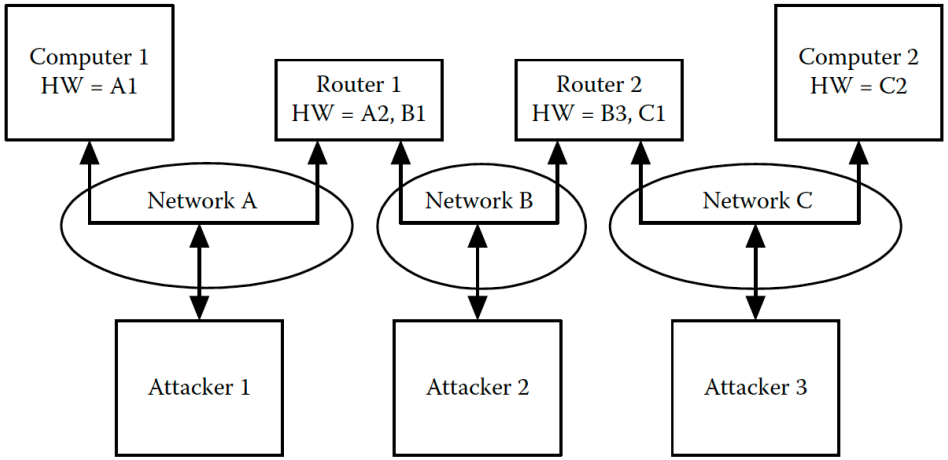


Figure 6.21 IP Address Spoofing Mitigation

Traffic Based

- Sniffing is a problem



- There are some flooding-based attacks using the IP broadcast address. The most common was discussed earlier and is when the attacker sends an IP broadcast packet into a remote network and gets all of the hosts to reply

ARP Broadcast Flood

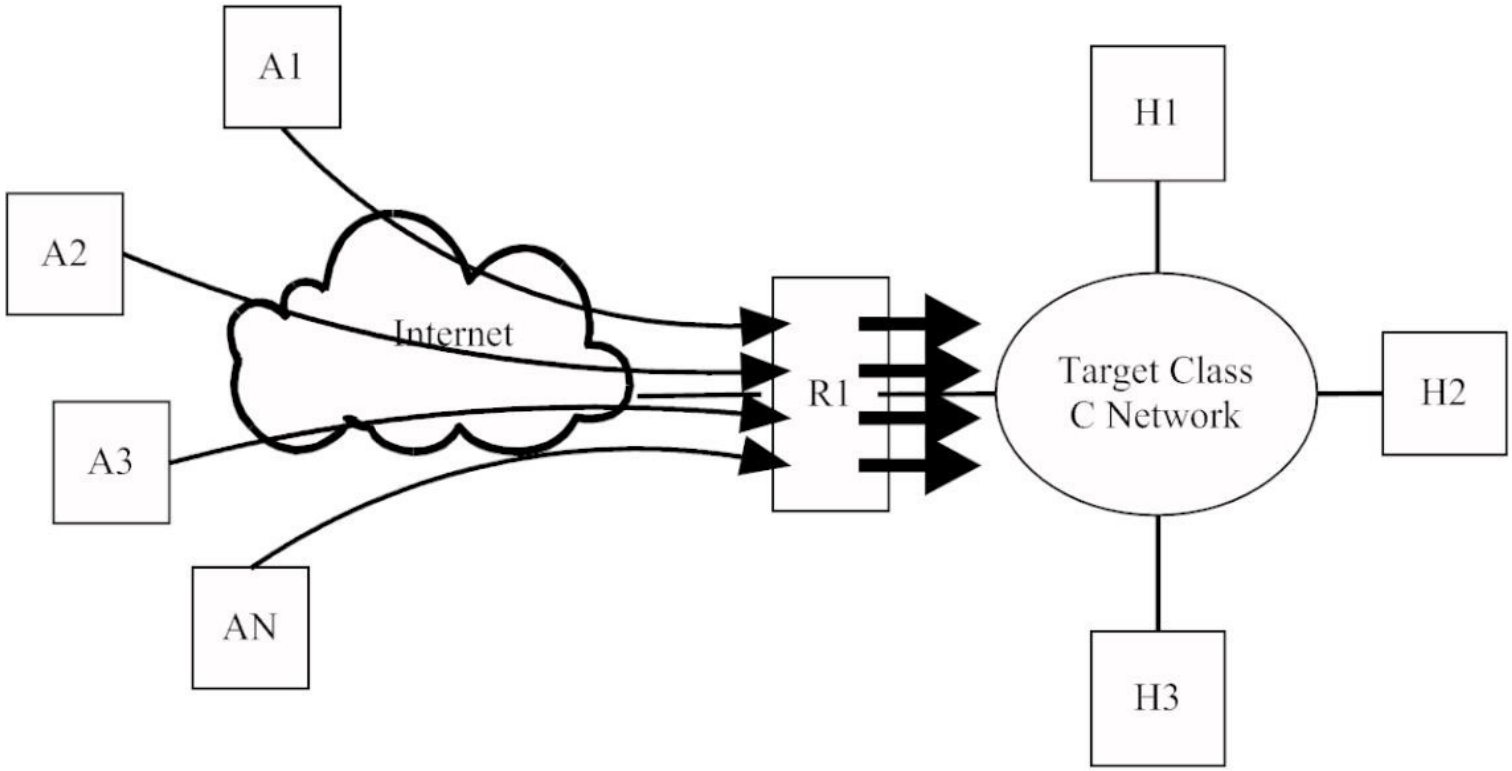


Figure 6.23 ARP Broadcast Flood Attack

BOOTP

- Bootstrap Protocol

  -Allows a networked machine to automatically acquire an IP address

- Client-server program
- Server has configuration file which contains a one-to-one mapping between the hardware address of the client and an IP address
- Used for networked laser printers and other diskless machines

BOOTP

- BOOTP server provides client with:
  - IP address
  - Subnet mask
  - IP address of a router
  - IP address of a nameserver

BOOTP

Sample configuration for a printer

```
hp255:\
  :hn:ht=ether:vm=rfc1048:\
  :ha=0800094ce9f5:\
  :ip=129.186.5.7:\
  :sm=255.255.255.0:\
  :gw=129.186.5.254:\
  :lg=129.186.5.2:\
  :T144="hp.printer":
```


Hardware address

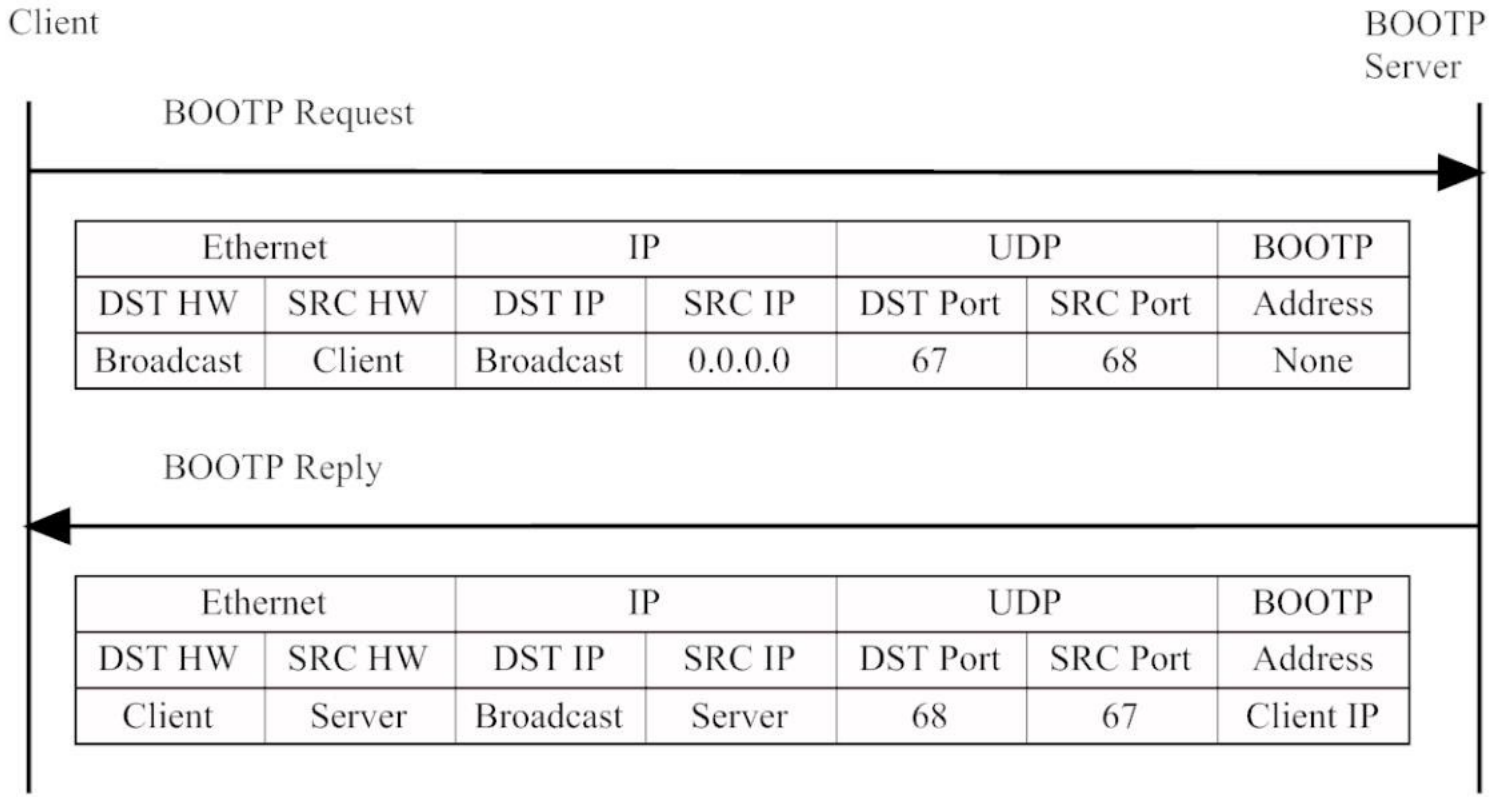

IP


Netmask


Gateway


Logging device

BOOTP Protocol



Ethernet		IP		UDP		BOOTP
DST HW	SRC HW	DST IP	SRC IP	DST Port	SRC Port	Address
Broadcast	Client	Broadcast	0.0.0.0	67	68	None

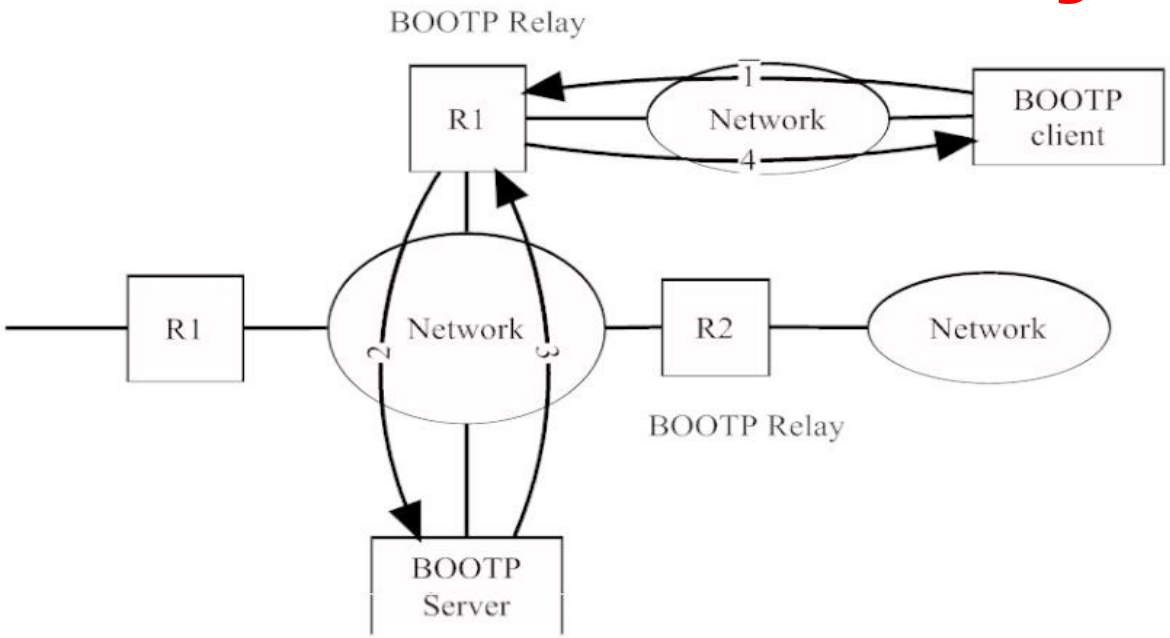
Ethernet		IP		UDP		BOOTP
DST HW	SRC HW	DST IP	SRC IP	DST Port	SRC Port	Address
Client	Server	Broadcast	Server	68	67	Client IP

Figure 6.24 BOOTP Protocol

BOOTP

- Note that the client must broadcast it's request, since it does not know who the local router is
- BOOTP relay
  - Used when client and server are on different subnets
  - Relay receives requests, appends its address, sends requests to server
  - Server replies to relay who then replies to client

BOOTP Relay



	Ethernet		IP		UDP		BOOTP
Packet	DST HW	SRC HW	DST IP	SRC IP	DST Port	SRC Port	Address
1	Broadcast	Client	Broadcast	0.0.0.0	67	68	none
2	Server	Relay	Server	Relay	67	68	none
3	Relay	Server	Relay	Server	68	67	Client IP
4	Client	Relay	Broadcast	Relay	68	67	Client IP

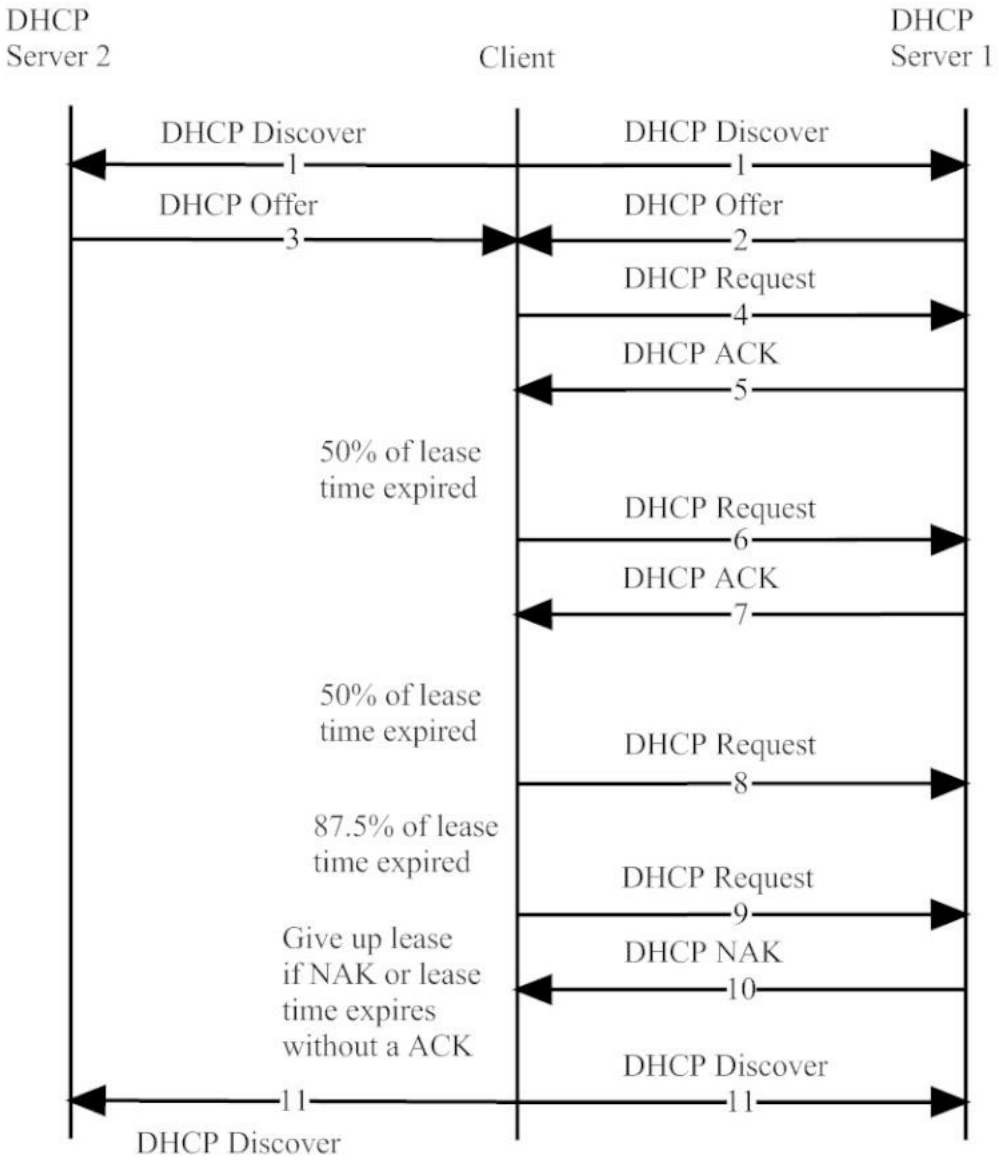
DHCP

- Dynamic Host Configuration Protocol
- An enhancement to BOOTP
- Leases IP addresses to hosts requesting an address
- Dynamic leases (not a one-to-one mapping)

DHCP

- Two databases for each DHCP server:
  - Static IP pool (like bootp)
  - Dynamic pool
- Server checks static pool for match before dynamic pool
- Dynamic pool
  - Addresses are temporary (default lease = 1 hr)
  - After lease expires, client must ask for a renewal
  - If renewal is rejected, client must give up the IP address

DHCP Operation



DHCP Operation

	Ethernet		IP		UDP		DHCP
Packet	DST HW	SRC HW	DST IP	SRC IP	DST Port	SRC Port	
1	Broadcast	Client	Broadcast	0.0.0.0	67	68	Discover
2	Client	Server 1	Broadcast	Server 1	68	67	Offer
3	Client	Server 2	Broadcast	Server 2	68	67	Offer
4	Server 1	Client	Server 1	0.0.0.0	67	68	Request
5	Client	Server 1	Broadcast	Server 1	68	67	ACK
6	Server 1	Client	Server 1	Client	67	68	Request
7	Client	Server 1	Broadcast	Server 1	68	67	ACK
8	Server 1	Client	Server 1	Client	67	68	Request
9	Server 1	Client	Server 1	Client	67	68	Request
10	Client	Server 1	Broadcast	Server 1	68	67	NAK
11	Broadcast	Client	Broadcast	0.0.0.0	67	68	Discover

DHCP Packet Format

Op Code	Hardware type	Hardware Len	Hop Count
ID			
Number of Seconds		Flag + Unused	
Client IP Address			
Client IP Address (used in reply packet)			
Server IP Address			
Gateway IP Address			
Client Hardware Address (16 bytes)			
Server Name (64 bytes)			
Boot File Name (128 bytes)			
Options (contains DHCP message types)			

Figure 6.27 DHCP/BOOTP Header Format

Header based attacks

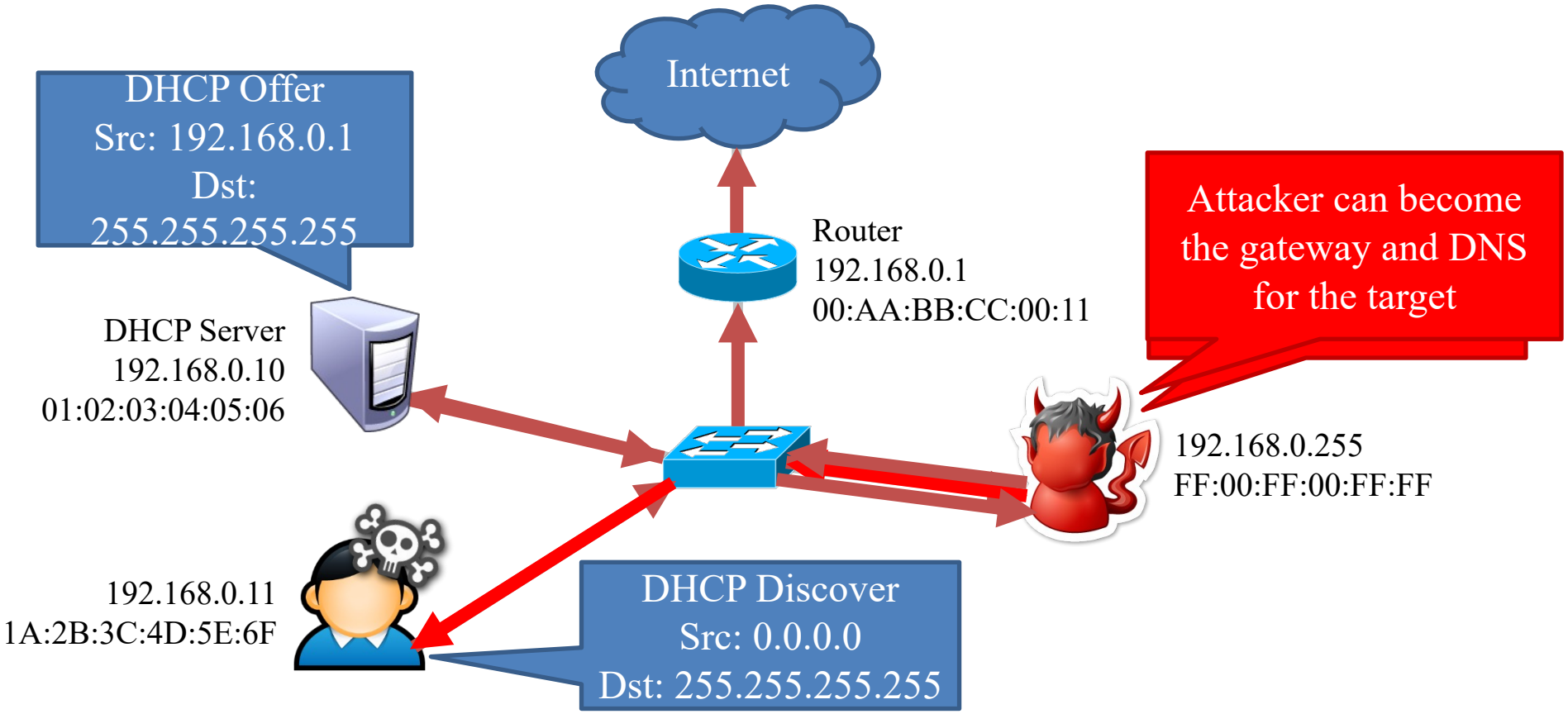
- Very simple header, no attacks

Protocol / Auth based attacks

- BOOTP is a simple protocol
  - An attacker could try and give false information causing a host to get the wrong IP address. (really an authentication attack)
- DHCP is more complex
  - An attacker could reserve all of the addresses
  - An attacker could send fake release packets

DHCP Starvation attack

DHCP Poisoning



Traffic Based

- Sniffing is not an issue since the information is not a secret
- Not any real good flooding based attacks due to the slow nature of the protocol

General countermeasures

- Since IP is so ingrained in the Internet it is hard to provide security. There are a few general countermeasures.
  - IP Filtering
  - Network Address Translation (NAT)
  - Virtual Private Network (VPN)
  - Encrypted IPV4 & IPV6 (IPSec)

IP Filtering

- Routers can be configured to filter out packets based on:
  - IP Address (black listing)
    - Hard to keep list current
    - Hard to get off the list (DOS)
  - Port numbers
    - Rogue protocols use multiple ports
  - Protocol types (TCP, UDP, ICMP)
    - Coarse grain filtering

Network Address Translation

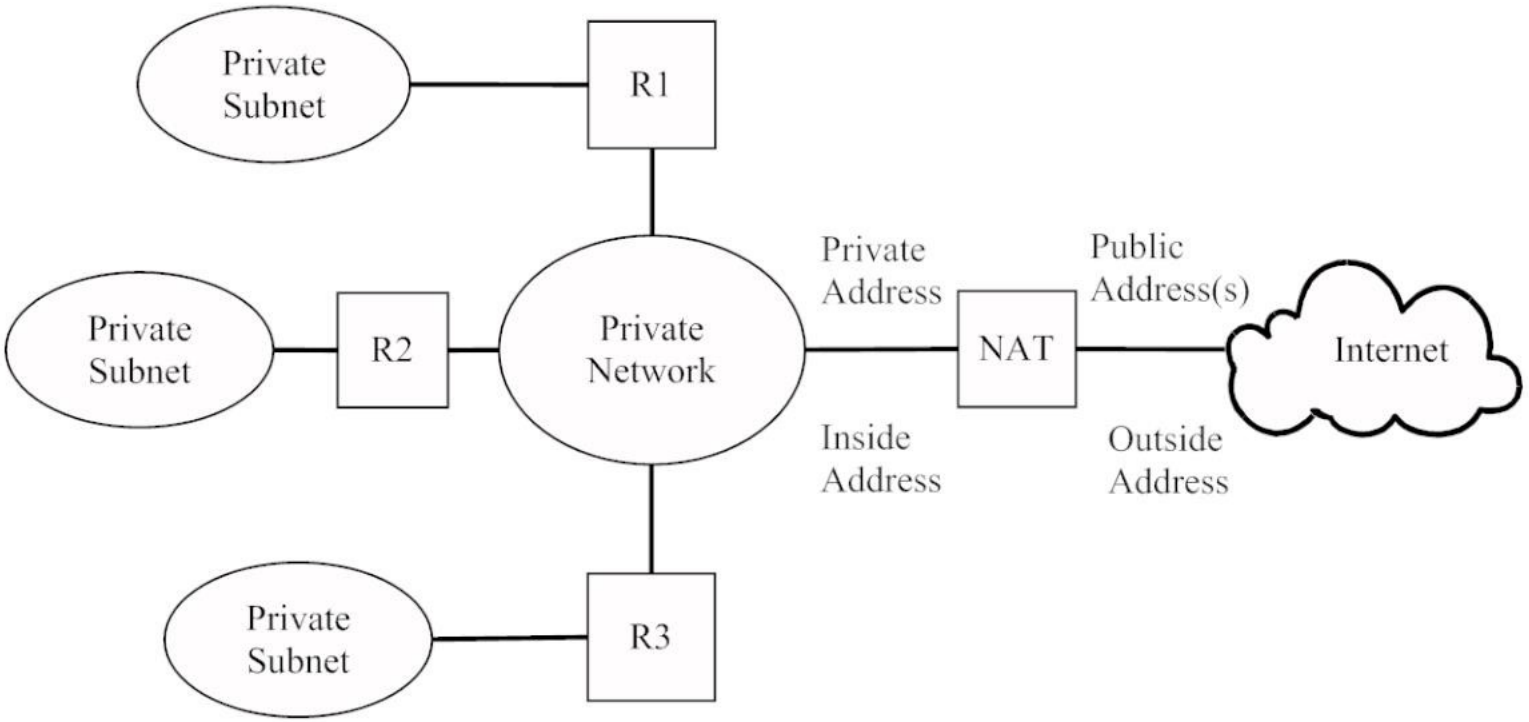


Figure 6.30 Private Network

Network Address Translation

- Used to extend the address space
  - Internal address ranges
    - 10/8 10.0.0.0
    - 172.16/12 172.16.0.0 (16 class B networks)
    - 192.168/16 192.168.0.0 (class B network)
- Static NAT
- Dynamic NAT

Static NAT

- One to one mapping of external addresses to internal addresses
- Used when a small number of machines need Internet access.
- NAT looks like a router to the inside machines and the destination to outside machines

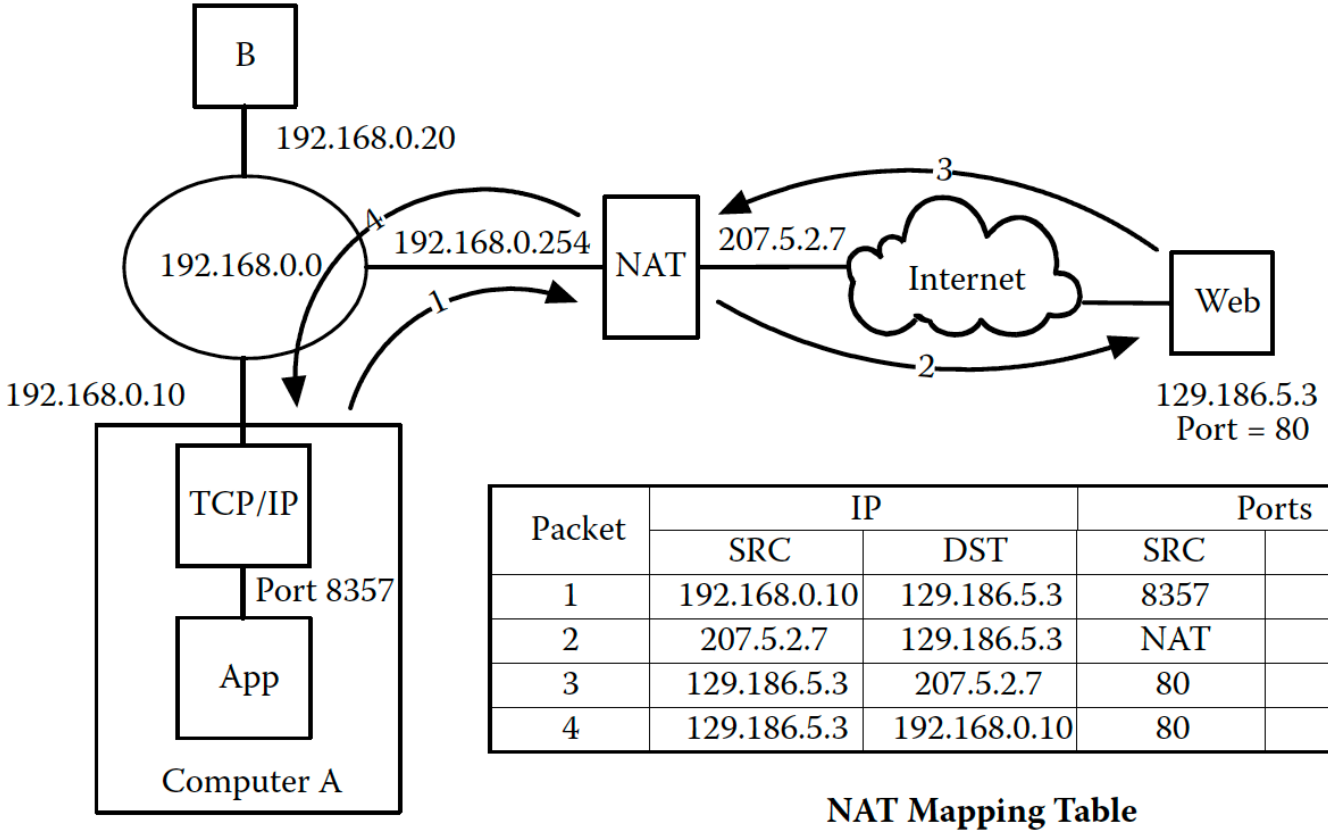
Static NAT

Public	Port	Private	Port
129.186.5.100	80	192.168.20.30	80
129.186.5.150	25	192.168.20.50	80

Dynamic NAT

- More machines on the inside than IP addresses on the outside.
- Used for outgoing access
- Can use tunnels for servers or combine with static NAT
- Inside can have same address range as a valid outside network (overlapping)

Dynamic NAT (Port mapping)



Packet	IP		Ports	
	SRC	DST	SRC	DST
1	192.168.0.10	129.186.5.3	8357	80
2	207.5.2.7	129.186.5.3	NAT	80
3	129.186.5.3	207.5.2.7	80	NAT
4	129.186.5.3	192.168.0.10	80	8357

NAT Mapping Table

Public IP	Ports		Private IP	Ports	
	SRC	DST		SRC	DST
129.186.5.3	NAT	80	192.168.0.10	8357	80

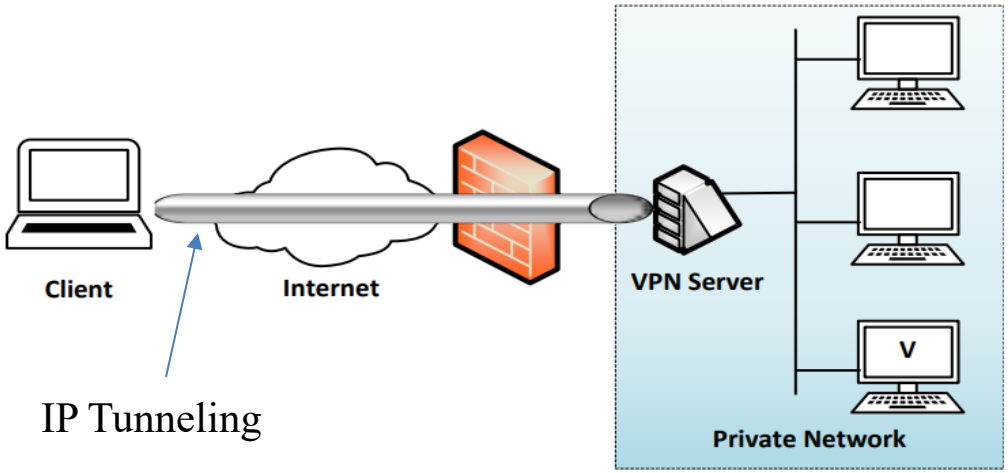
Virtual Private Network

Virtual Private Network

- VPN allows users to create a secure, private network over a public network such as the Internet. This is achieved by:
  - Having a designated host (VPN server) on the network
  - Outside computers have to go through the VPN server to reach the hosts inside a private network via authentication.
  - VPN server is exposed to the outside and the internal computers are still protected, via firewalls or reserved IP addresses.

A Typical Setup

- This is a typical VPN setup where the “Client” machine wants to connect with machine “V” on a private network. “Client” uses the “VPN Server” to get authenticated to the private network



IPsec

- Provides Layer 3 security (RFC 2401)
  - Transparent to applications (no need for integrated IPSec support)
- A set of protocols and algorithms used to secure IP data at the network layer
- Combines different components:
  - Security associations (SA)
  - Authentication headers (AH)
  - Encapsulating security payload (ESP)
  - Internet Key Exchange (IKE)
- A security context for the VPN tunnel is established via the ISAKMP

Why IPsec?

- Internet Protocol (IP) is not secure
  - IP protocol was designed in the early stages of the Internet where security was not an issue
  - All hosts in the network are known
- Possible security issues
  - Source spoofing
  - Replay packets
  - No data integrity or confidentiality

IPsec Standards

- RFC 4301 “The IP Security Architecture”
  - Defines the original IPsec architecture and elements common to both AH and ESP
- RFC 4302
  - Defines authentication headers (AH)
- RFC 4303
  - Defines the Encapsulating Security Payload (ESP)
- RFC 2408
  - ISAKMP
- RFC 5996
  - IKE v2 (Sept 2010)
- RFC 4835
  - Cryptographic algorithm implementation for ESP and AH

Benefits of IPsec

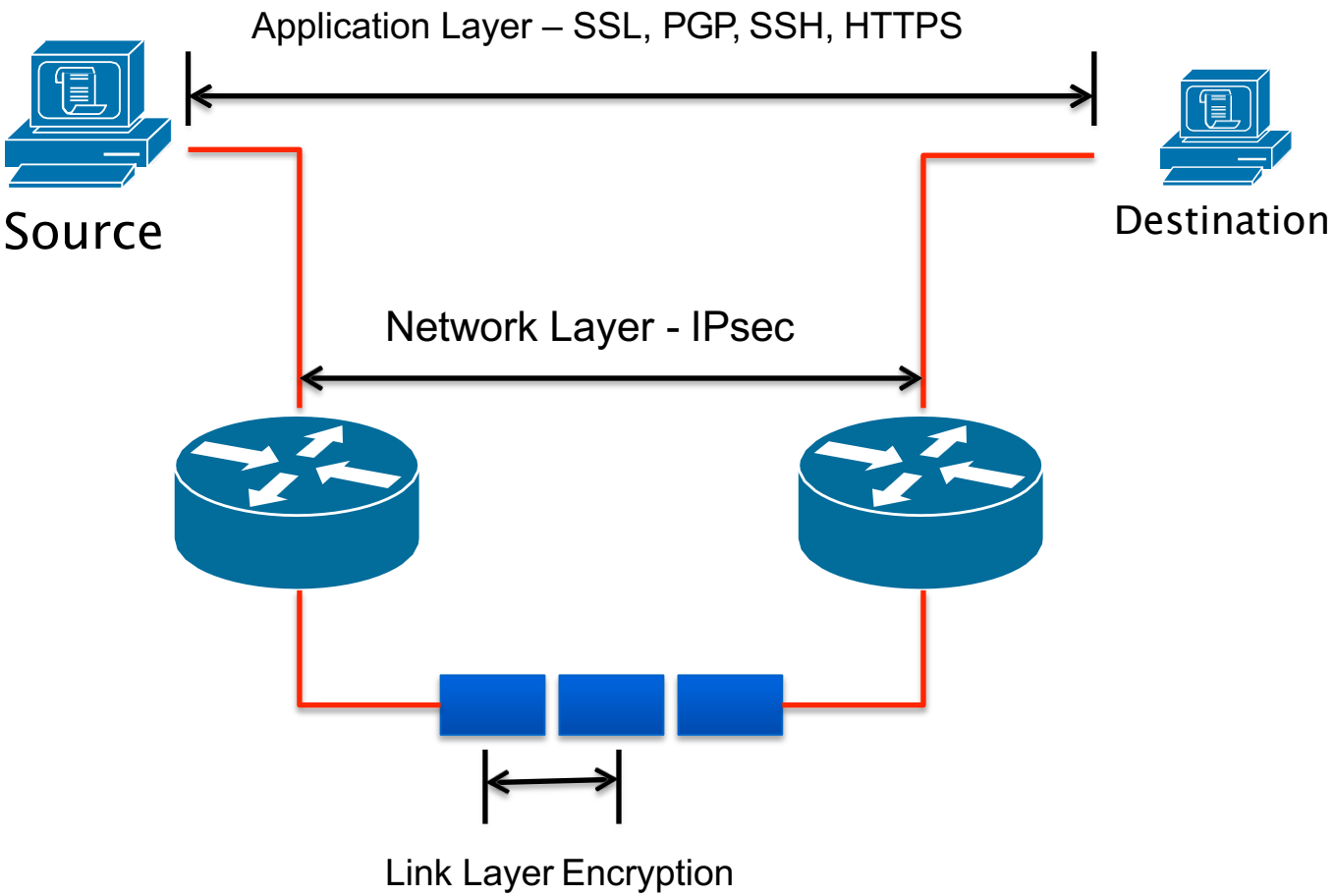
- Confidentiality
  - By encrypting data
- Integrity
  - Routers at each end of a tunnel calculates the checksum or hash value of the data
- Authentication
  - Signatures and certificates
  - All these while still maintaining the ability to route through existing IP networks

"IPsec is designed to provide interoperable, high quality, cryptographically-based security for IPv4 and IPv6" - (RFC 2401)

Benefits of IPsec

- Offers Confidentiality (encrypting data), Integrity , and Authentication
- Data integrity and source authentication
  - Data “signed” by sender and “signature” is verified by the recipient
  - Modification of data can be detected by signature “verification”
  - Because “signature” is based on a shared secret, it gives source authentication
- Anti-replay protection
  - Optional; the sender must provide it but the recipient may ignore
- Key management
  - IKE – session negotiation and establishment
  - Sessions are rekeyed or deleted automatically
  - Secret keys are securely established and authenticated
  - Remote peer is authenticated through varying options

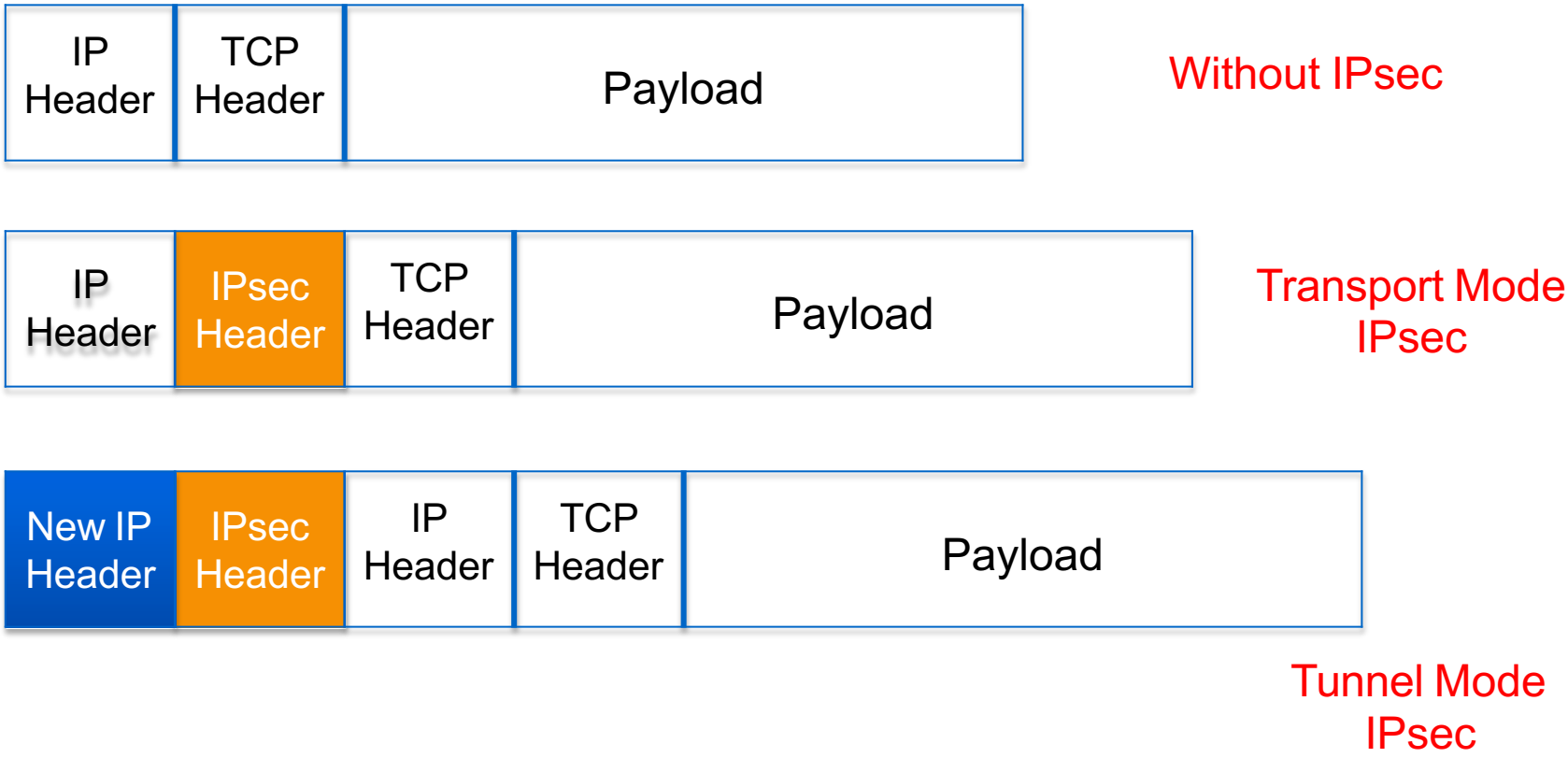
Different Layers of Encryption



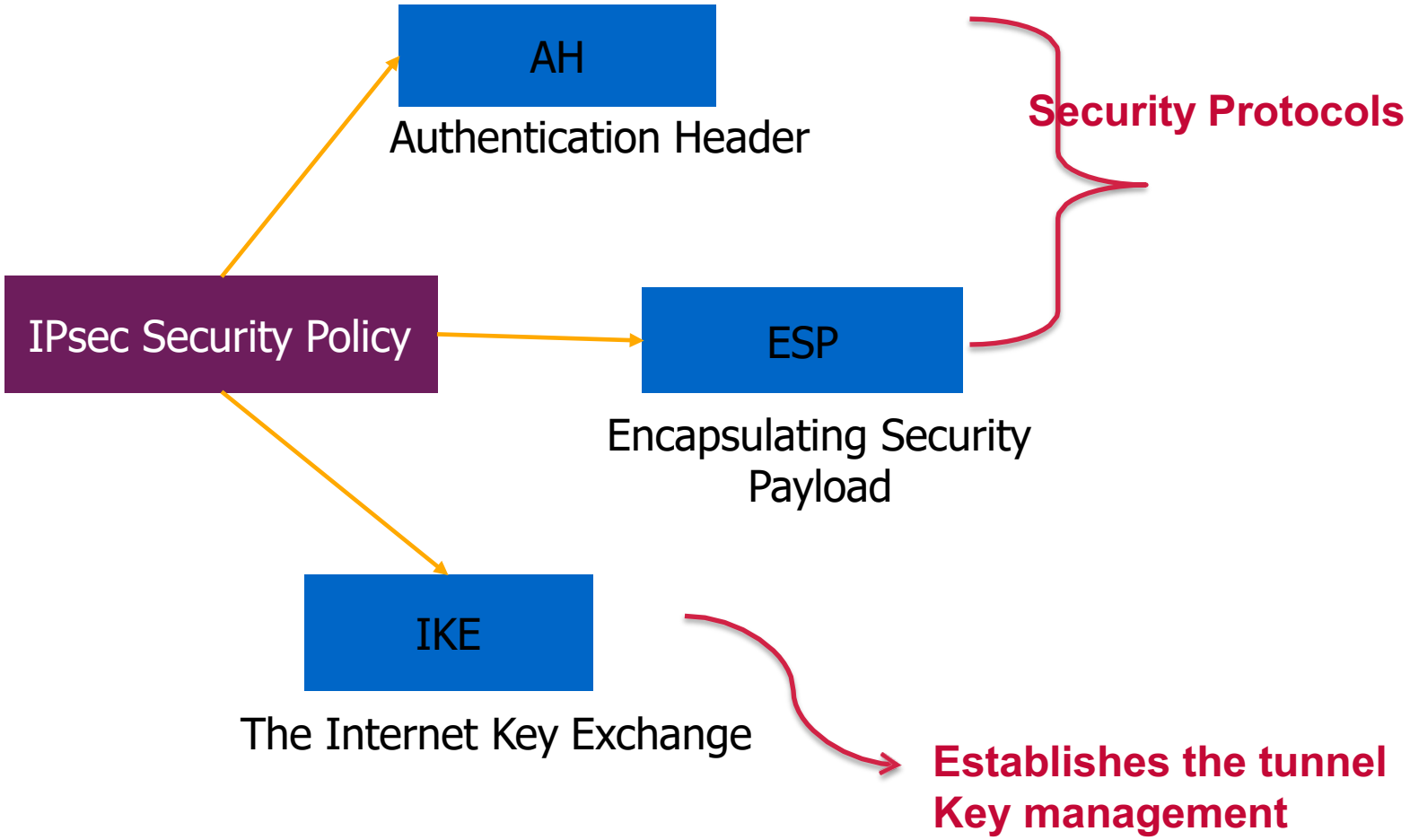
IPsec Modes

- Transport Mode
  - IPsec header is inserted into the IP packet
  - No new packet is created
  - Works well in networks where increasing a packet's size could cause an issue
  - Frequently used for remote-access VPNs
- Tunnel Mode
  - Entire IP packet is encrypted and becomes the data component of a new (and larger) IP packet.
  - Frequently used in an IPsec site-to-site VPN

Tunnel vs. Transport Mode IPsec



IPsec Architecture



Security Associations (SA)

- A collection of parameters required to establish a secure session
- An SA is unidirectional
  - Two SAs required for a bidirectional communication
- A single SA can be used for AH or ESP, but not both
  - must create two (or more) SAs for each direction if using both AH and ESP
- Uniquely identified by three parameters consisting of
  - Security Parameter Index (SPI)
  - IP destination address
  - Security protocol (AH or ESP) identifier

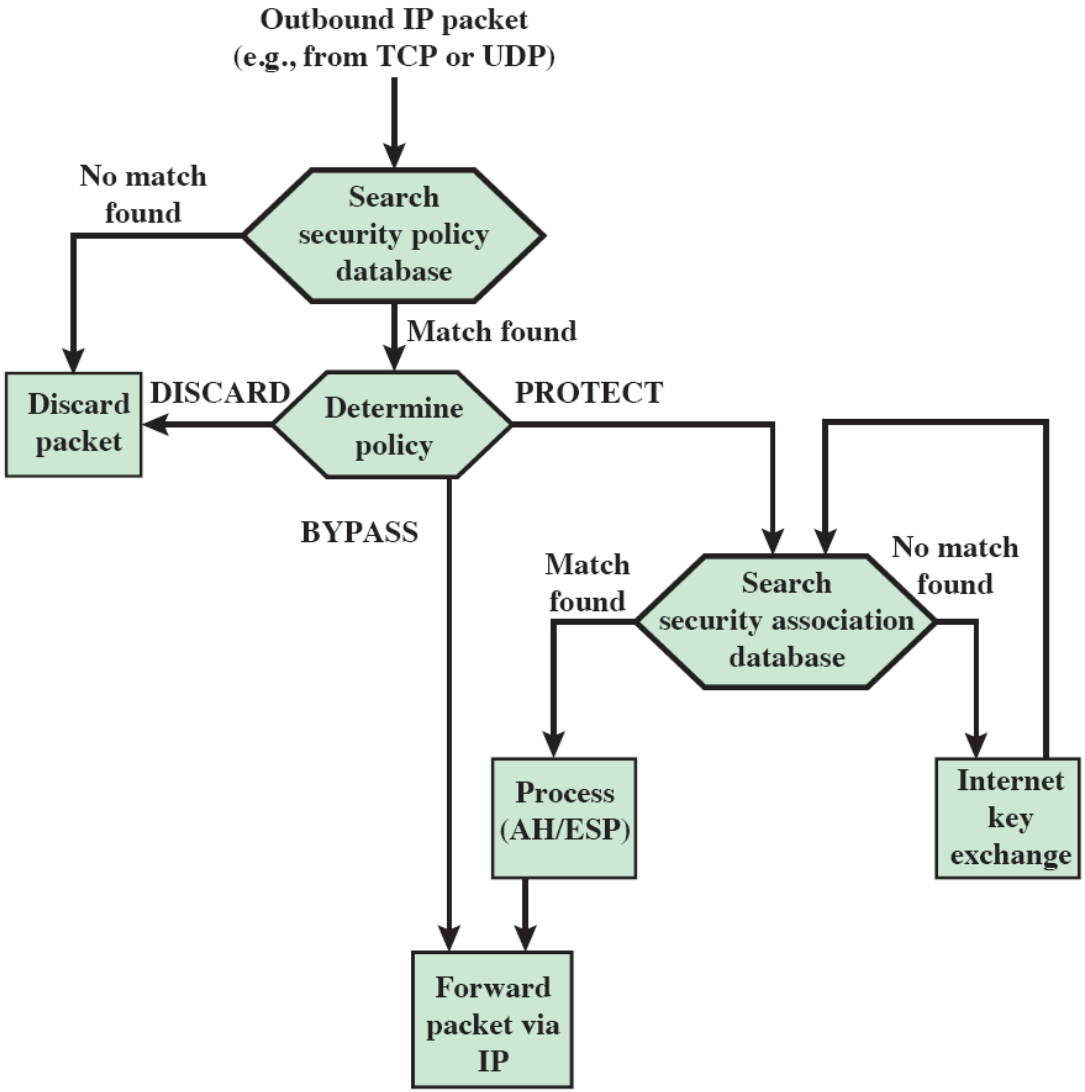
Security Associations (SA)

- There are several other parameters associated with an SA stored locally in Security Association Databases (SAD)
  - Anti-replay related
    - Sequence Number Counter
      - to generate sequence numbers
    - Anti-replay window
      - something like sliding-window; will be discussed later.
  - AH info
    - authentication algorithms, keys, key lifetimes, etc.
  - ESP info
    - encryption (and authentication) algorithms, keys, key lifetimes, etc.
  - Lifetime of SA
  - IPSec Mode: Transport or Tunnel

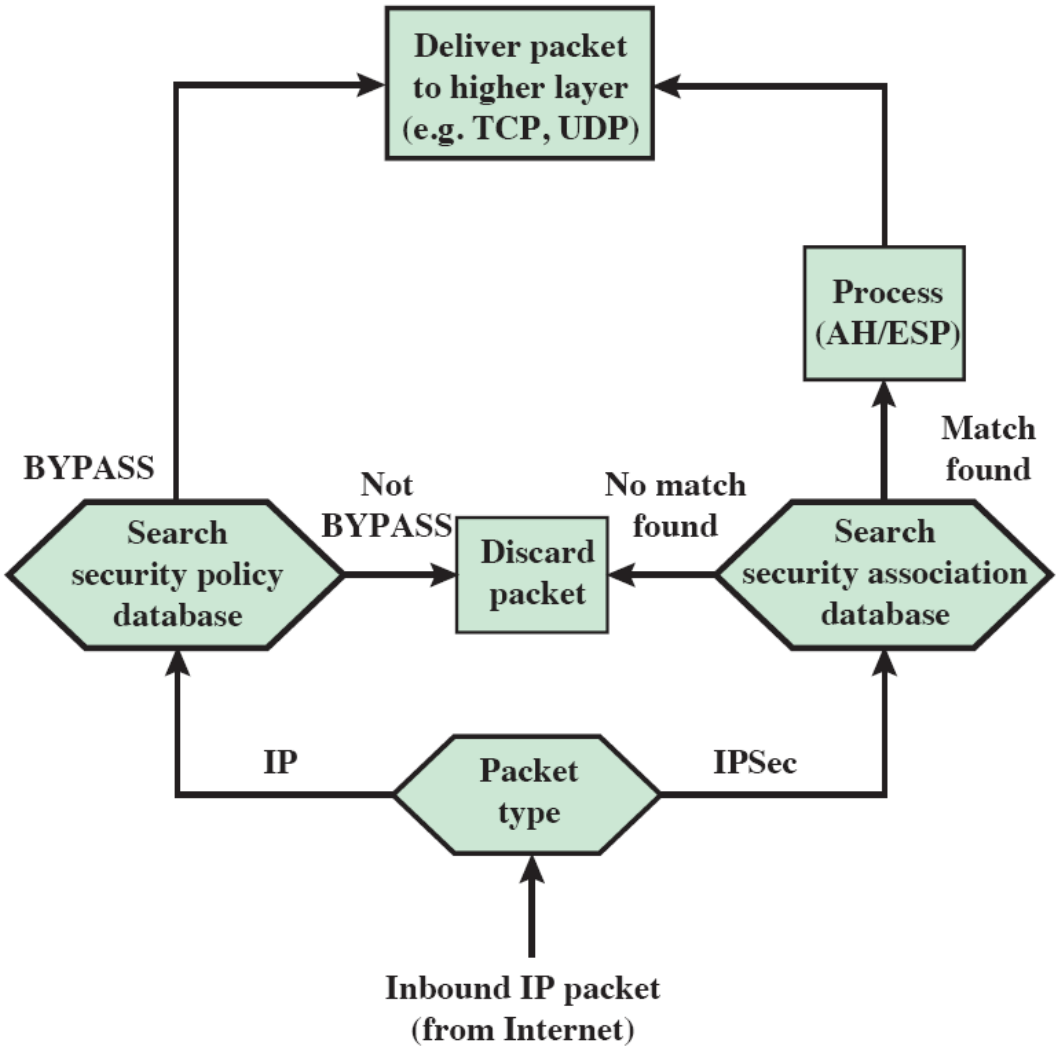
SA Selectors

- IPSec is a flexible protocol
  - traffic from IP address X to IP address Y may use several SAs or no SA if that particular traffic will not be secured
- Security Policy Database (SPD) is used to assign a particular IP traffic to an SA
  - fields of an SPD entry are called selectors
- Outbound processing
  - compare the selector fields of SPD with the one in the IP traffic
  - Determine the SA, if any
  - If there exists an SA, do the AH or ESP processing
- Inbound processing
  - Check the incoming IPSec packet and process with AH or ESP
  - Discard in case of an anomaly

Outbound Processing Model



Inbound Processing Model



How to Set Up an SA

- Manually
  - Sometimes referred to as “manual keying”
  - You configure on each node:
    - Participating nodes (I.e. traffic selectors)
    - AH and/or ESP [tunnel or transport]
    - Cryptographic algorithm and key
- Automatically
  - Using IKE (Internet Key Exchange)

ISAKMP

- Internet Security Association and Key Management Protocol
- Defined by RFC 2408
- Used for establishing Security Associations (SA) and cryptographic keys
- Only provides the framework for authentication and key exchange, but key exchange independent
- Key exchange protocols
  - Internet Key Exchange (IKE) and Kerberized Internet Negotiation of Keys (KINK)

Authentication Header (AH)

- Provides support for data integrity and authentication of IP packets
  - malicious modifications are detected
  - address spoofing is prevented
  - replays are detected via sequence numbers
- Authentication is based on use of a MAC
  - parties must share a secret key
    - in SA

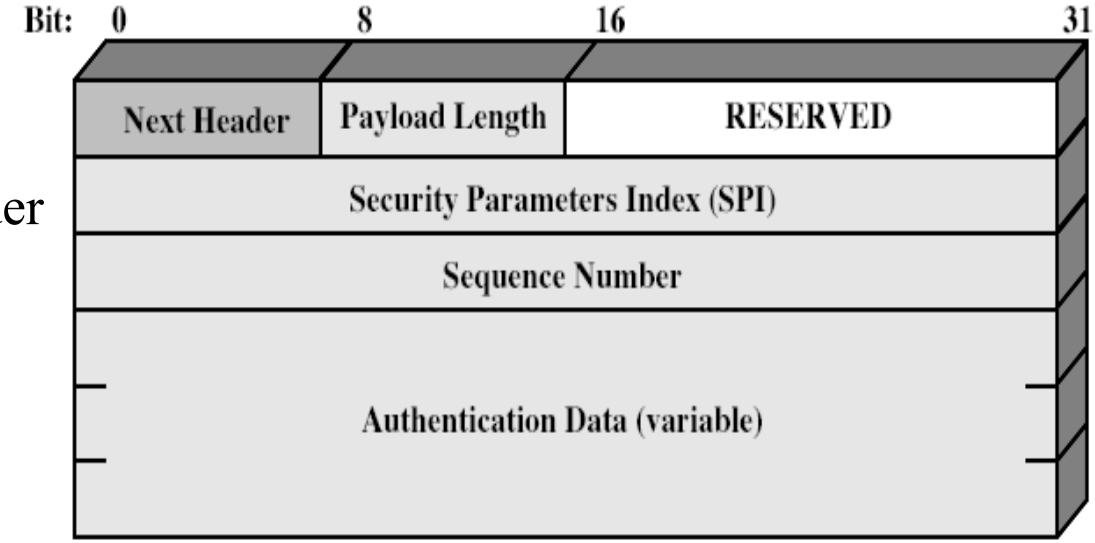
Authentication Header

Next Header: specifies next header or upper layer protocol

Payload length: to specify header length

SPI: to identify SA

Sequence number: used for replay control

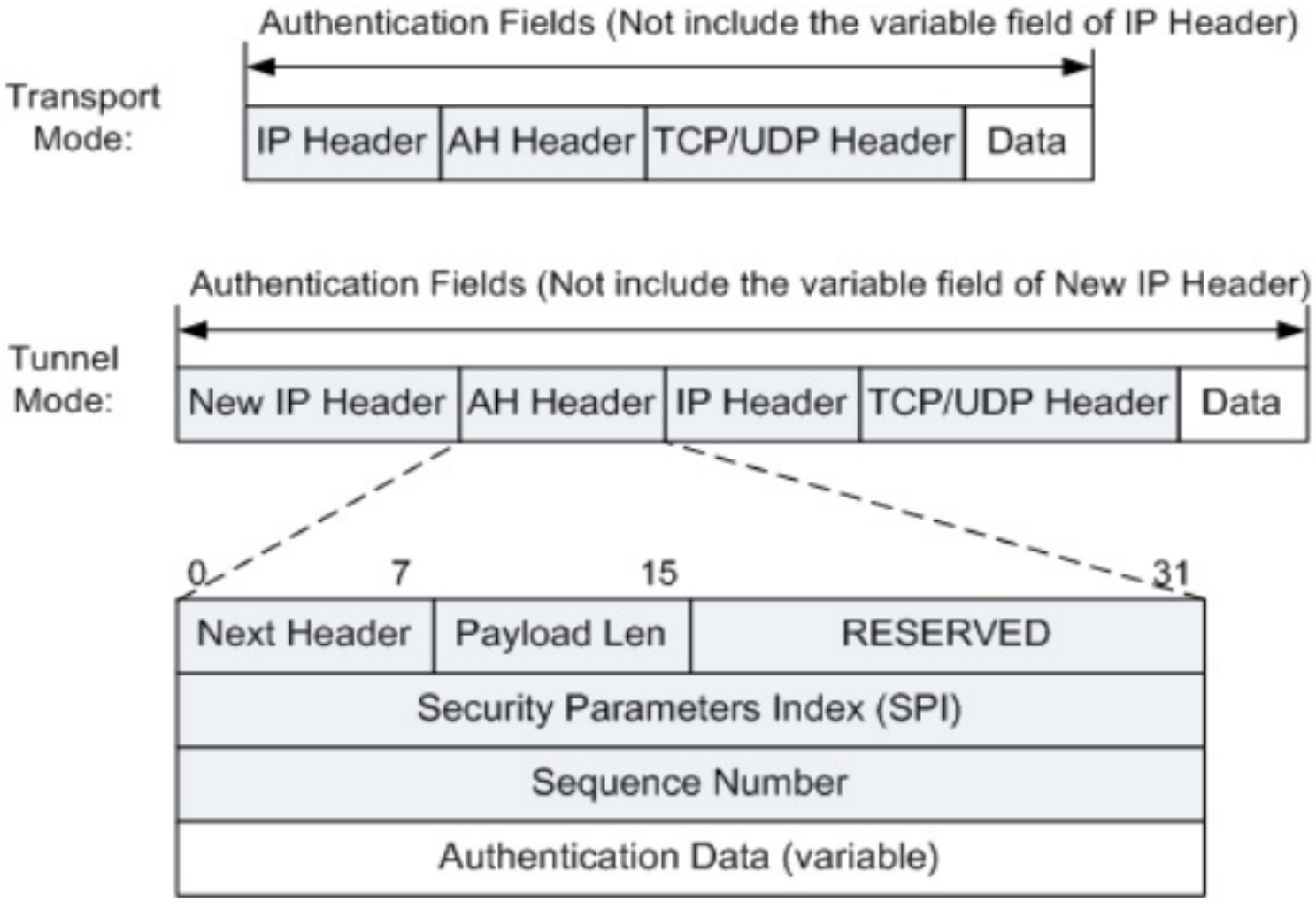


Authentication data: MAC value (variable length)

AH - Integrity Check Value (ICV)

- Actually it is a MAC
- HMAC is used
  - with a secure hash algorithm
  - default length of authentication data field is 96
    - so HMAC output is truncated
- MAC is calculated over
  - IP payload (upper layer protocol data)
  - IP Headers that are “immutable” or “mutable but predictable” at destination
    - e.g. source address (immutable), destination address (mutable but predictable)
    - Time to live field is mutable. Such mutable fields are zeroed for MAC calculation

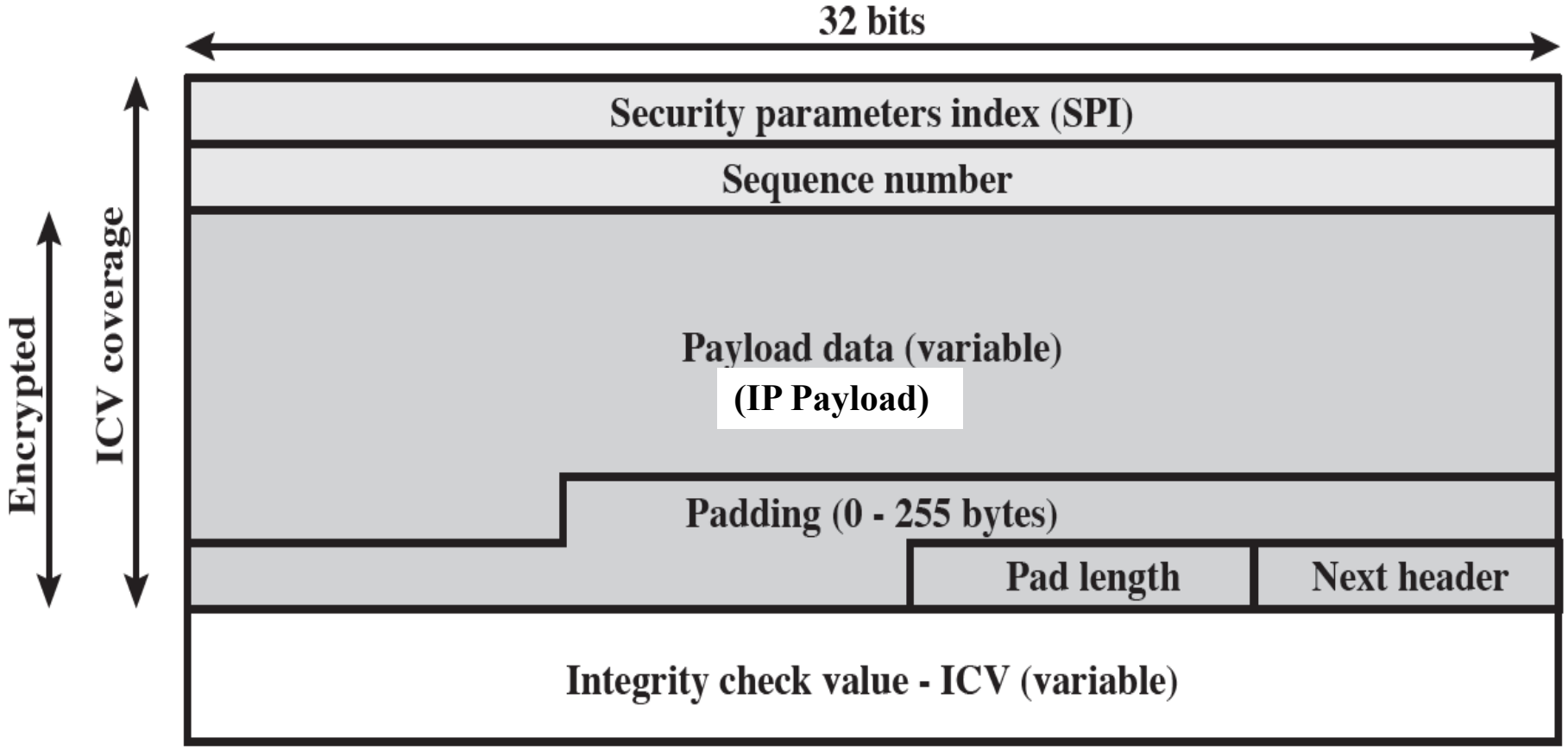
AH – Transport Mode



Encapsulating Security Payload (ESP)

- provides
  - message content confidentiality
    - via encryption
  - limited traffic flow confidentiality and measures for traffic analysis
    - by padding (may arbitrarily increase the data)
    - by encrypting the source and destination addresses in tunnel mode
  - optionally authentication services as in AH
    - via MAC (HMAC), sequence numbers
- supports range of ciphers, modes
  - DES, Triple-DES, RC5, IDEA, Blowfish, etc.
  - CBC is the most common mode

Encapsulating Security Payload

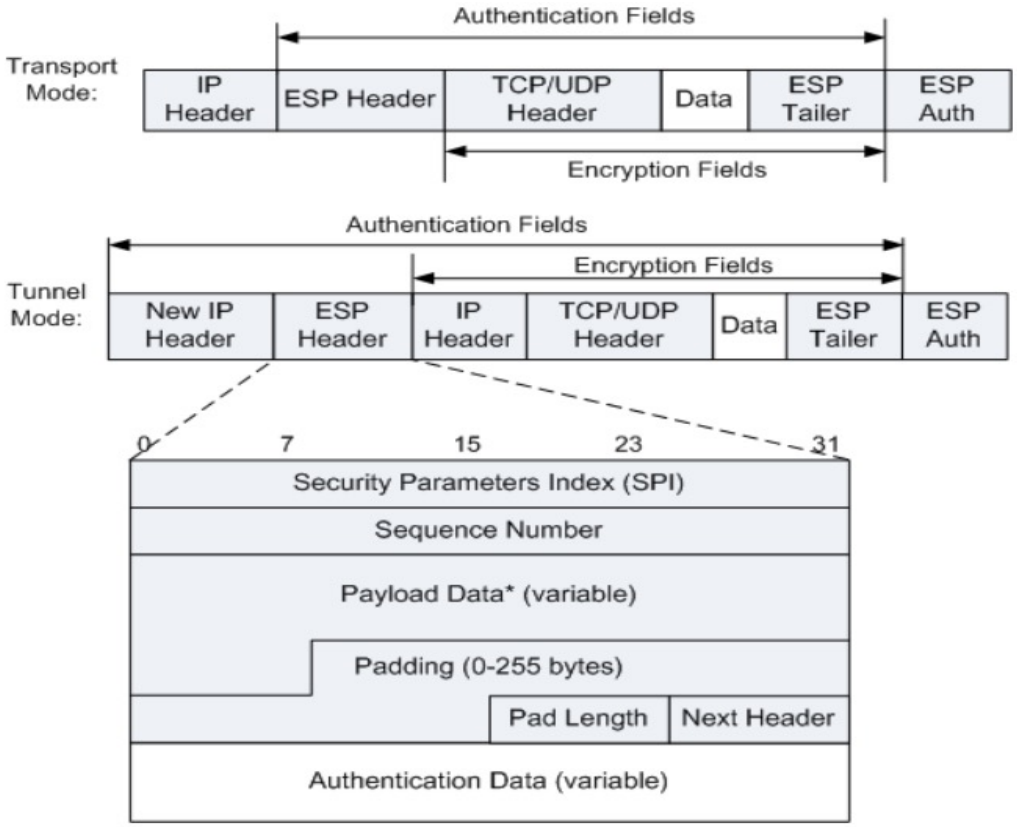


Padding in ESP

- several purposes and reasons
  - encryption algorithm may require the plaintext to be multiple of some integer *n*
  - ESP format requires 32-bit words
  - additional padding may help to provide partial traffic flow confidentiality by concealing the actual length of data
    - Other than the existing padding field, extra padding can be added to the end of the payload to improve traffic flow confidentiality

Transport Mode ESP

- transport mode is used to encrypt & optionally authenticate IP payload (e.g. TCP segment)
  - data protected but IP header left in clear
  - so source and destination addresses are not encrypted
  - Mostly for host to host (end-to-end) traffic



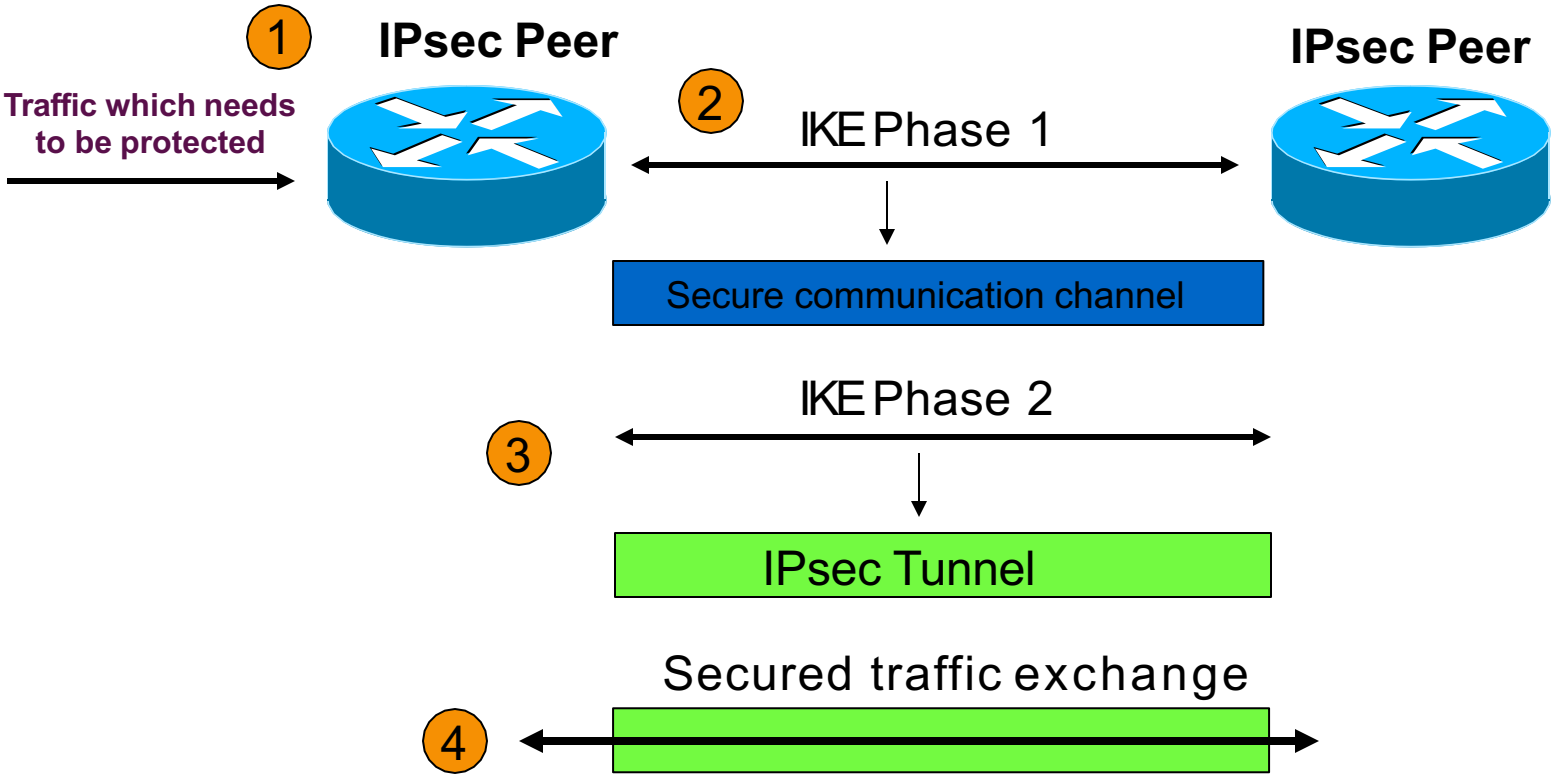
Internet Key Exchange (IKE)

- “An IPsec component used for performing mutual authentication and establishing and maintaining Security Associations.” (RFC 5996)
- Typically used for establishing IPsec sessions
- A key exchange mechanism
- Five variations of an IKE negotiation:
  - Two modes (aggressive and main modes)
  - Three authentication methods (pre-shared, public key encryption, and public key signature)
- Uses UDP port 500

IKE Modes

Mode	Description
Main mode	Three exchanges of information between IPsec peers. Initiator sends one or more proposals to the other peer (responder) Responder selects a proposal
Aggressive Mode	Achieves same result as main mode using only 3 packets First packet sent by initiator containing all info to establish SA Second packet by responder with all security parameters selected Third packet finalizes authentication of the ISAKMP session
Quick Mode	Negotiates the parameters for the IPsec session. Entire negotiation occurs within the protection of ISAKMP session

Overview of IKE



Phases of IKE

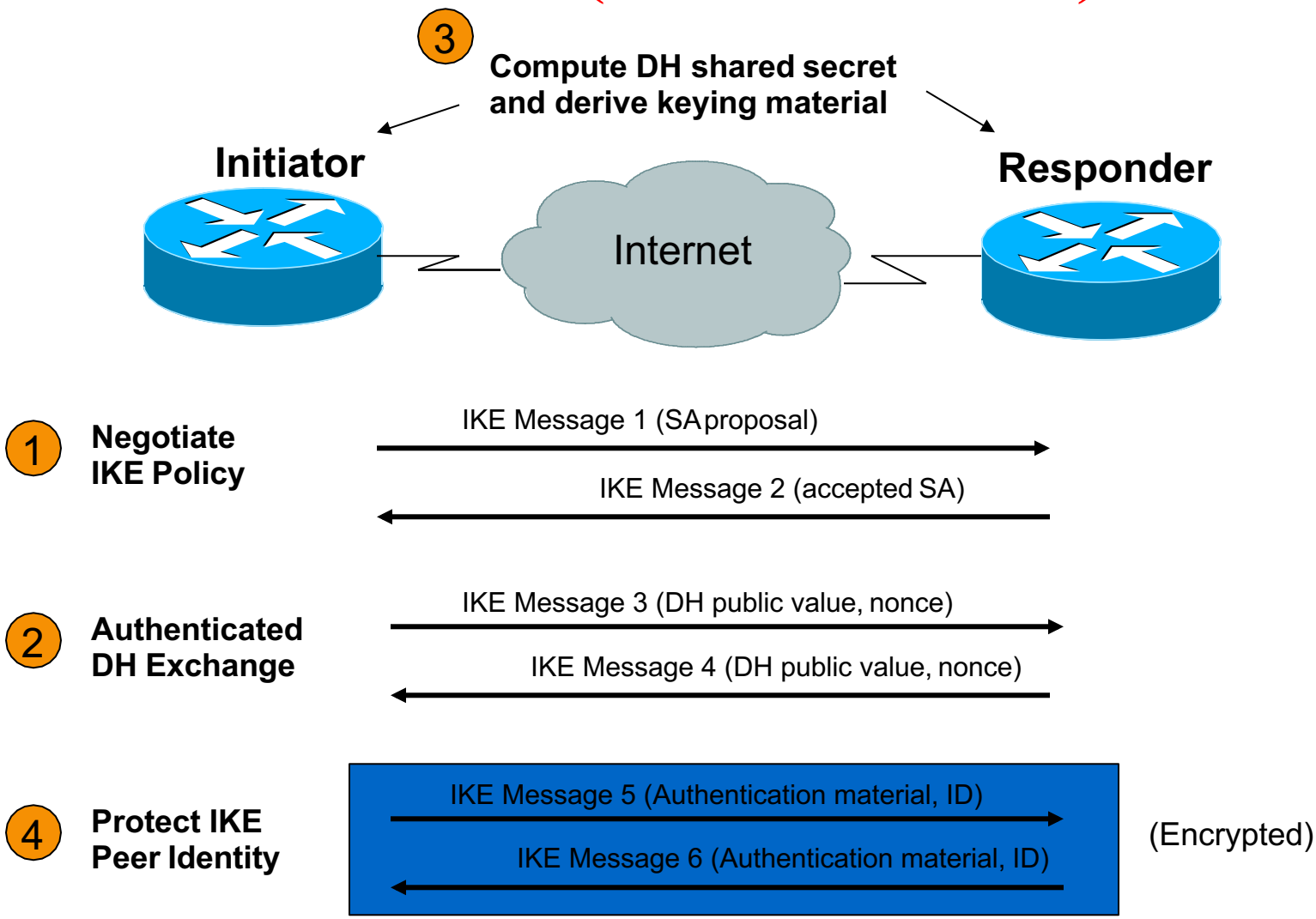
- Phase 1: establish IKE SA
  - Main mode (DH with identity protection)
    - ISAKMP identity protection exchange
  - Aggressive mode (DH without identity protection)
    - ISAKMP aggressive mode

- Phase 2: establishes SA for target protocol (AH or ESP)
  - CREATE_CHILD_SA exchange (only 2 messages)
  - IKE SA is used to protect this exchange
  - Several SAs can be established in this way

IKE Phase 1 (Main Mode)

- Main mode negotiates an ISAKMP SA which will be used to create IPsec SAs
- Three steps
  - SA negotiation (encryption algorithm, hash algorithm, authentication method, which DF group to use)
  - Do a Diffie-Hellman exchange
  - Provide authentication information
  - Authenticate the peer

IKE Phase 1 (Main Mode)



IKE Phase 1 (Aggressive Mode)

- Uses 3 (vs 6) messages to establish IKE SA
- No denial of service protection
- Does not have identity protection
- Optional exchange and not widely implemented

IKE Phase 2 (Quick Mode)

- All traffic is encrypted using the ISAKMP Security Association
- Each quick mode negotiation results in two IPsec Security Associations (one inbound, one outbound)
- Creates/refreshes keys

IKE Phase 2 (Quick Mode)