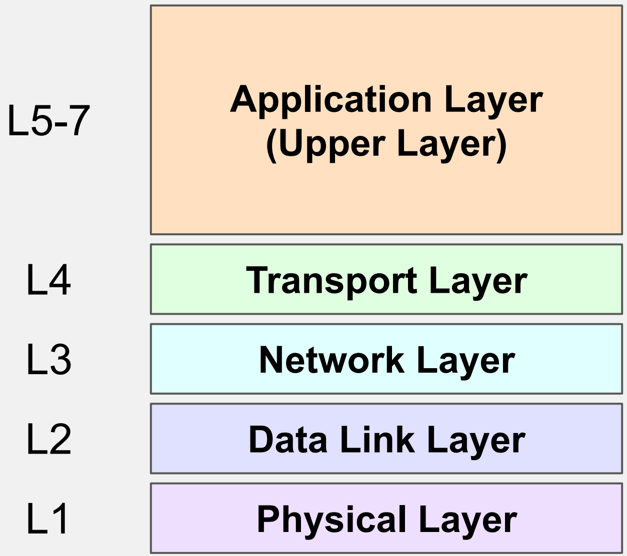
Network Security

From Bits to Broadcast: IP Unmasked

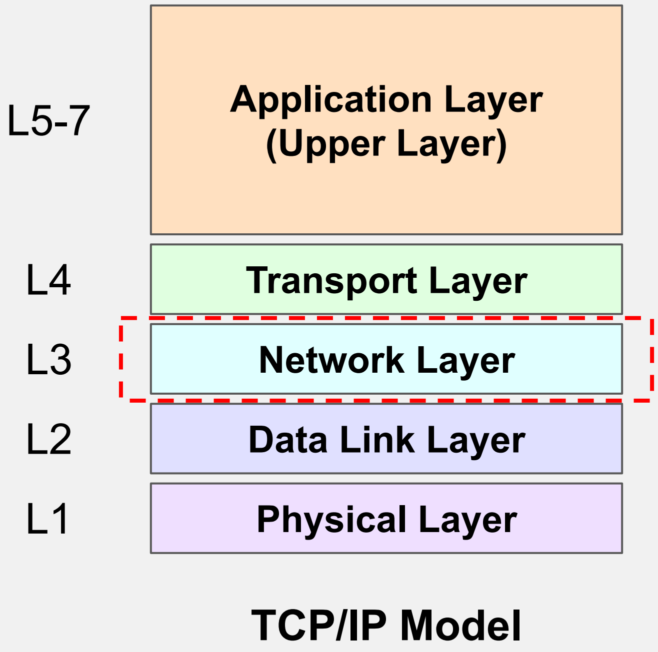
Gwendal Patat
Univ Rennes, CNRS, IRISA
2025/2026

Recall TCP/IP Model



TCP/IP Model

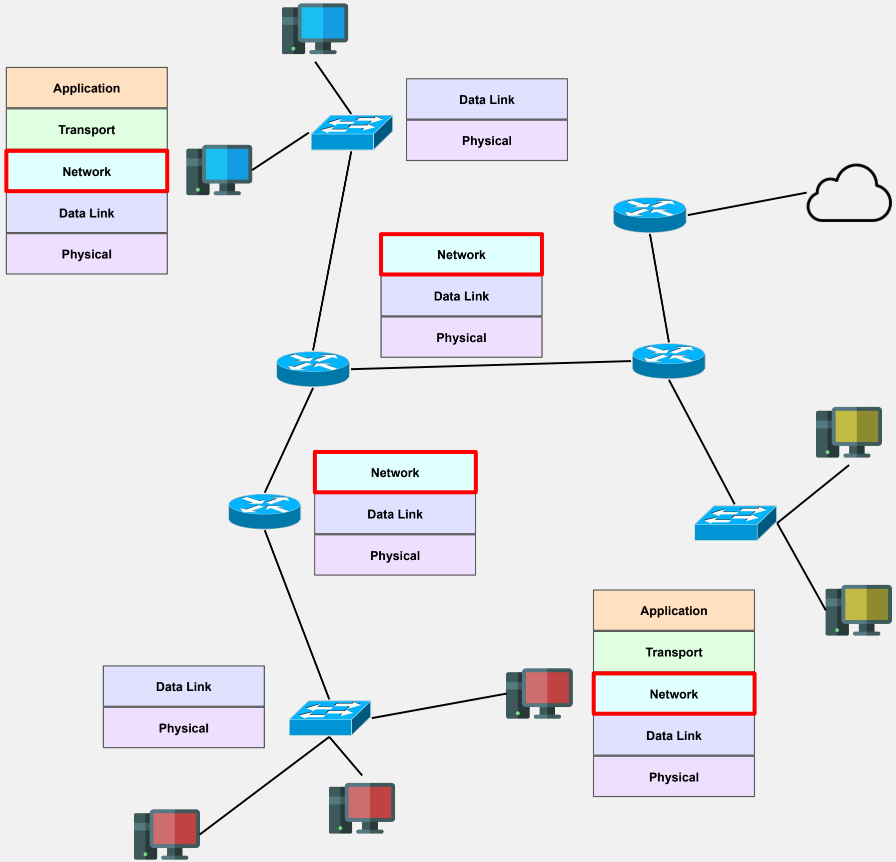
Today's Topic: Network Layer



Network Layer functions 1/2

Network Layer:

- Transport segments from sender to receiver hosts.
- Encapsulate/Decapsulate the segments for the next layer.
- Layer 3 is found in every Internet devices: e.g., routers
  - Not in switches (layer 2 devices)
- Routers only have to look at layer 3 headers containing the IP addresses to move along the datagram.



Network Layer functions 2/2

Network Layer functions:

- **Forwarding:** Move packets from the router’s input link to the output one.
- **Routing:** Determine which path to take to reach the destination, using routing algorithms.

Network Layer: Data and Control planes

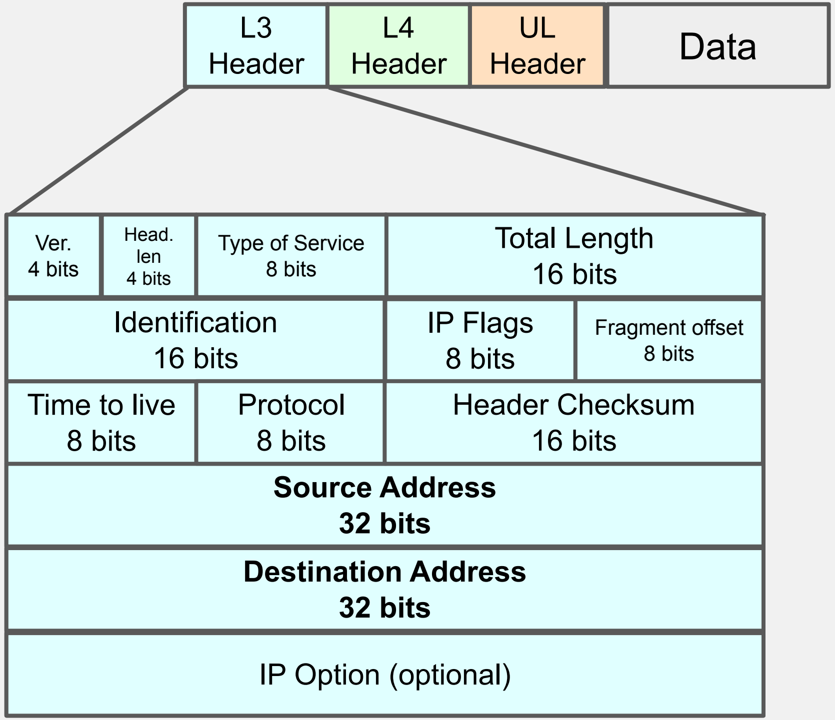
Data plane:

- Local per router function.
- Basic forwarding of ports
  - Input -> output

Control plane:

- Network-wide logic
- Determines how packets are routed among routers from source to host

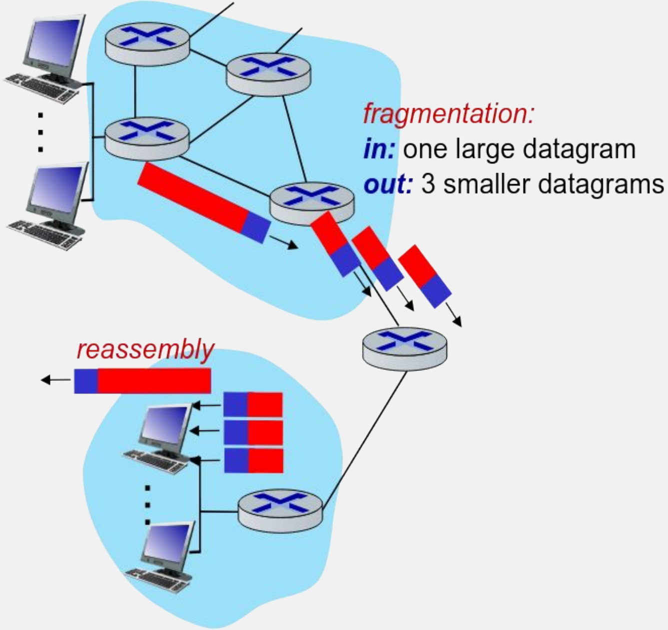
IPv4 Packet



L3 Header
L4 Header
UL Header
Data
Ver. 4 bits
Head. len 4 bits
Type of Service 8 bits
Total Length 16 bits
Identification 16 bits
IP Flags 8 bits
Fragment offset 8 bits
Time to live 8 bits
Protocol 8 bits
Header Checksum 16 bits
Source Address 32 bits
Destination Address 32 bits
IP Option (optional)

Fragmentation/Reassembly

- Network links have an **MTU** (Maximum Transmission Unit), which is the largest possible link-level frame.
- Different link types have different MTUs.
- A large IP datagram can be divided (“fragmented”) within the network.
  - One datagram becomes several smaller datagrams.
  - Reassembly happens only at the final destination.
  - IP header bits are used to identify and order related fragments.



Subnets & DHCP

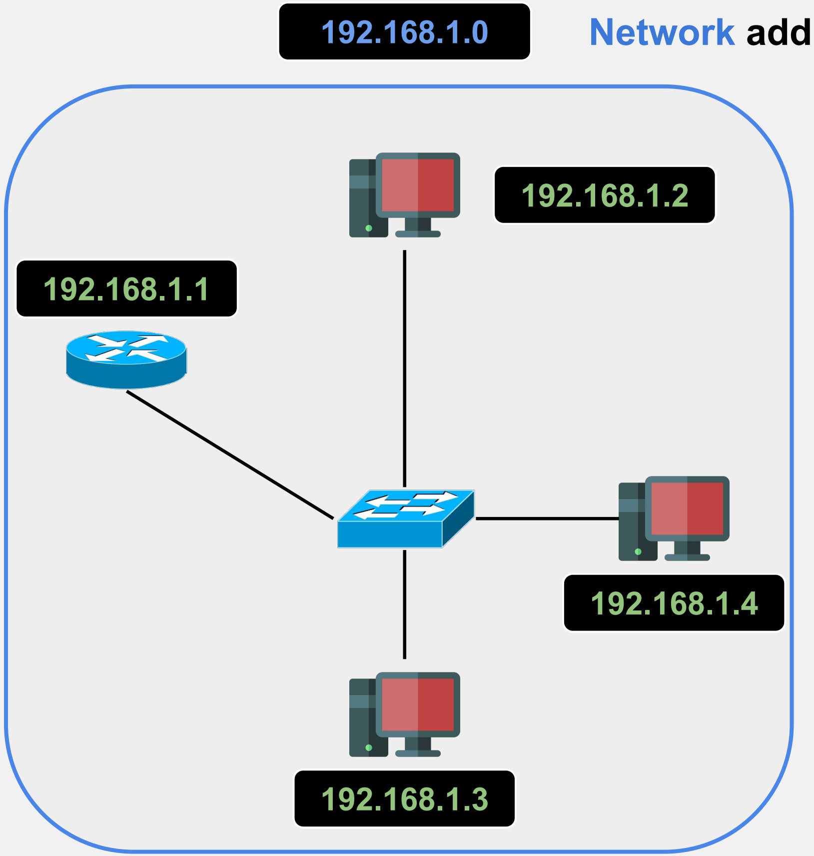
IPv4 Addresses

Layer 3 Address: 32 bits device network address

for instance: **192.168.1.0**
With two distinct parts.

Network address | **Host Address**

IP Addressing



Network address

Host Address

IP Addressing: Subnet Mask 1/4

192 . 168 . 1 . 0

Network address **Host Address**

255 . 255 . 255 . 0

The **subnet mask** defines the bits in the IP address used for the network by **masking** the network part.

IP Addressing: Subnet Mask 2/4

Addresses are in binary formats.

192 . 168 . 1 . 0 ⟹ **11000000 . 10101000 . 00000001 . 00000000**

255 . 255 . 255 . 0 ⟹ **11111111 . 11111111 . 11111111 . 00000000**

$$\underbrace{\mathbf{192\ .\ 168\ .\ 1}}_{\textbf{Network}}\ .\ \underbrace{\mathbf{0}}_{\textbf{Host}}$$

The **subnet mask** defines the bits in the IP address used for the network by **masking** the network part.

IP Addressing: Subnet Mask 3/4

Another example:

192 . 168 . 1 . 0 $\Longrightarrow$ **11000000 . 10101000 . 00000001 . 00000000**

255 . 255 . 0 . 0 $\Longrightarrow$ **11111111 . 11111111 . 00000000 . 00000000**

$$\underbrace{192 . 168}_{\textbf{Network}} . \underbrace{1 . 0}_{\textbf{Host}}$$

The **subnet mask** defines the bits in the IP address used for the network by **masking** the network part.

IP Addressing: Subnet Mask 4/4

Another example:

192 . 168 . 1 . 0 ⟹ **11000000 . 10101000 . 00000001 . 00000000**

255 . 255 . 224 . 0 ⟹ **11111111 . 11111111 . 11100000 . 00000000**

192 . 168 . 1 . 0 $\underbrace{\textbf{11000000 . 10101000 . 000}}_{\textbf{Network}}\underbrace{\textbf{00001 . 00000000}}_{\textbf{Host}}$

The **subnet mask** defines the bits in the IP address used for the network by **masking** the network part.

IPv4 Classes

IPv4 address blocks:

- Allocated depending on needs by:
  - The **IANA** (Internet Assigned Numbers Authority) - Global Pool
  - The **RIRs** (Regional Internet Registries) - World Region (Africa, Europe, etc…)
  - The **ISPs** (Internet Service Providers) - Provider (Orange, Vodafone, etc…)
- Classes were defined with fixed masks and ranges.
  - Since 1993, we use **CIDR**.

Class	**Range**	**Default subnet mask**
A	0.0.0.0 - 127.255.255.255	255 . 0 . 0 . 0
B	128.0.0.0 - 191.255.255.255	255 . 255 . 0 . 0
C	192.0.0.0 - 223.255.255.255	255 . 255 . 255 . 0

CIDR 1/3

CIDR: Classless Inter-Domain Routing

255 . 0 . 0 . 0	⟹	**11111111 . 00000000 . 00000000 . 00000000**	⟹	**/8**
255 . 255 . 0 . 0	⟹	**11111111 . 11111111 . 00000000 . 00000000**	⟹	**/16**
255 . 255 . 224 . 0	⟹	**11111111 . 11111111 . 11100000 . 00000000**	⟹	**/19**
255 . 255 . 255 . 0	⟹	**11111111 . 11111111 . 11111111 . 00000000**	⟹	**/24**

192.168.1.0/24

$\underbrace{\textbf{192.168.1}}_{\textbf{Network}}.\underbrace{\textbf{0}}_{\textbf{Host}}$

CIDR 2/3

192.168.1.0/19

192 . 168 . 1 . 0 ⟹ **11000000 . 10101000 . 00000001 . 00000000**

255 . 255 . 224 . 0 ⟹ $\underbrace{\mathbf{11111111\ .\ 11111111\ .\ 111}}_{\text{19 bits}}\underbrace{\mathbf{00000\ .\ 00000000}}_{\text{13 bits}}$

CIDR 3/3

192.168.1.0/19

192 . 168 . 1 . 0 ⟹ **11000000 . 10101000 . 00000001 . 00000000**

255 . 255 . 224 . 0 ⟹ **11111111 . 11111111 . 11100000 . 00000000**

19 bits 13 bits

11000000 . 10101000 . 00000000 . 00000000 ↓ **Network Address:** 192.168.0.0

11000000 . 10101000 . 00011111 . 11111111 ↓ **Broadcast Address:** 192.168.31.255

Total number of available hosts: $2^{32-19} - 2 = 8190$

Subnetting Example 1/5

Situation:

- We are given the network 192.172.8.0/24.
- We have **3** departments in our company that we want to divide.

/24 ⟹ **11111111 . 11111111 . 11111111 . 00000000**

Subnetting Example 2/5

Situation:

- We are given the network 192.172.8.0/24.
- We have **3** departments in our company that we want to divide.

/24 ⟹ 11111111 . 11111111 . 11111111 . 00000000

/26 ⟹ 11111111 . 11111111 . 11111111 . $\underbrace{11}$000000

2 bits long -> **4** subnets:
00, 01, 10, 11

Subnetting Example 3/5

Situation:

- We are given the network 192.172.8.0/24.
- We have 3 departments in our company that we want to divide.

/26 ⟹ $11111111 . 11111111 . 11111111 . 11\underbrace{000000}$

2^6 = 64 addresses per block

Network	Subnet Mask	# of Hosts	Host Range	Broadcast
192.172.8.0	/26			
192.172.8.64	/26			
192.172.8.128	/26			
192.172.8.192	/26			

Subnetting Example 4/5

Situation:

- We are given the network 192.172.8.0/24.
- We have 3 departments in our company that we want to divide.

/26 ⟹ 11111111 . 11111111 . 11111111 . 11000000

2^6 = 64 addresses per block

Network	Subnet Mask	# of Hosts	Host Range	Broadcast
192.172.8.0	/26			192.172.8.63
192.172.8.64	/26			192.172.8.127
192.172.8.128	/26			192.172.8.191
192.172.8.192	/26			192.172.8.255

Subnetting Example 5/5

Situation:

- We are given the network 192.172.8.0/24.
- We have 3 departments in our company that we want to divide.

/26 $\Longrightarrow$ 11111111 . 11111111 . 11111111 . 11$\underbrace{000000}$

2^6 = 64 addresses per block

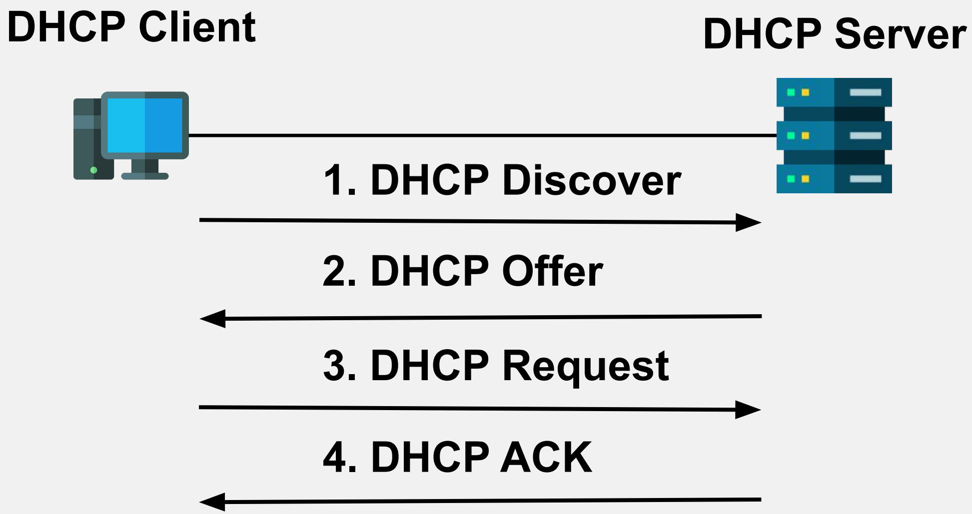
Network	Subnet Mask	# of Hosts	Host Range	Broadcast
192.172.8.0	/26	62	192.172.8.1 - 192.172.8.62	192.172.8.63
192.172.8.64	/26	62	192.172.8.65 - 192.172.8.126	192.172.8.127
192.172.8.128	/26	62	192.172.8.129 - 192.172.8.190	192.172.8.191
192.172.8.192	/26	62	192.172.8.193 - 192.172.8.254	192.172.8.255

DHCP

DHCP: Dynamic Host Configuration Protocol

- Application Layer protocol.
- Here to give IP addresses **from the network IP pool** to devices connecting to the network.
  - **IPs are leased NOT given.**
- Can be on a dedicated server (full host like Linux or Windows).
- Can also be **integrated as a service** in routers.
- Before DHCP, IP addresses, default gateway, DNS IP, etc…, add to be configured by hand for all hosts.

DHCP

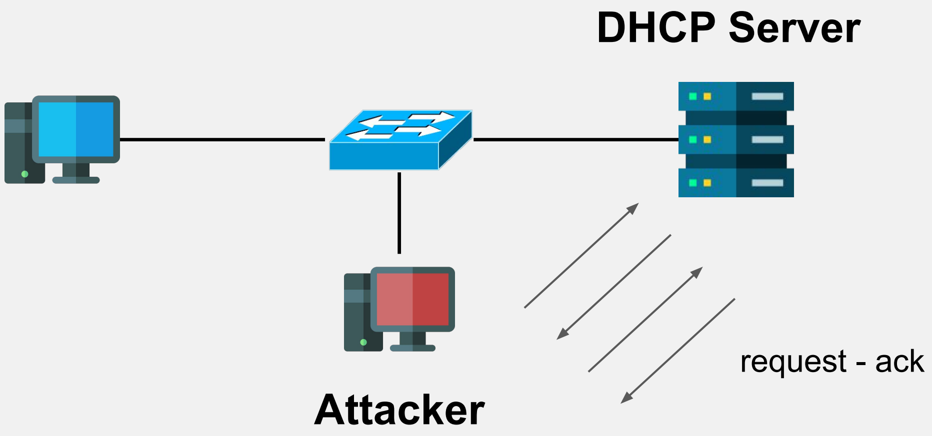


DHCP Client
DHCP Server
1. DHCP Discover
2. DHCP Offer
3. DHCP Request
4. DHCP ACK

DHCP Attacks

DHCP Starvation

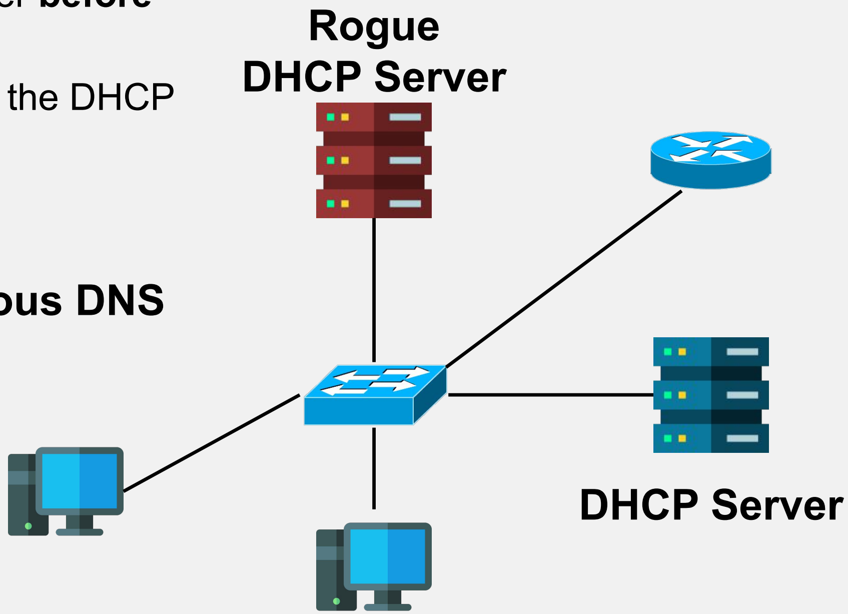
- The IP address pool is **limited**.
- An attacker could forge DHCP Request messages with **fake MAC addresses** to grab all available IP addresses.
- This result in a **Denial-of-Service (DoS)** of the network for newcomers.



DHCP Spoofing: Malicious Intruder in the Middle

- Steps:
  - A legitimate client sends a **DHCP Discover** message.
  - The Rogue DHCP Server sends a DHCP Offer **before the legitimate one**.
  - The client performs the request and receives the DHCP package **from the rogue server**.

- The Rogue Server can put itself as a **malicious DNS** or even the **default gateway** of the victim.



Mitigation

- Switch features:
  - **Port Security** with limited number of MAC addresses per port.
  - **DHCP Snooping** to trust DHCP Offer coming only from trusted ports.

- Other good practices:
  - VLAN segmentation.
  - Monitoring.

Routing & ICMP

Routing/Forwarding table

Routers and hosts maintain a routing/forwarding table:

When a router receive a packet to forward, it checks the IP **destination** address against its **routing table**, to look for the **longest prefix match** or **more specific route**.

In this example:

- A packet to 192.168.1.10 will match the first entry.
- A packet to 192.168.4.4 match on the first but also on the second one. The more specific is preferred.

Q: What about a packet to 192.168.4.254?

Q: /32?

Network	Next Hop
192.168.0.0/22	Fa0/0
192.168.4.0/24	Fa0/1
192.168.4.254/32	10.42.42.42
0.0.0.0/0 (default)	192.0.2.1

ICMP

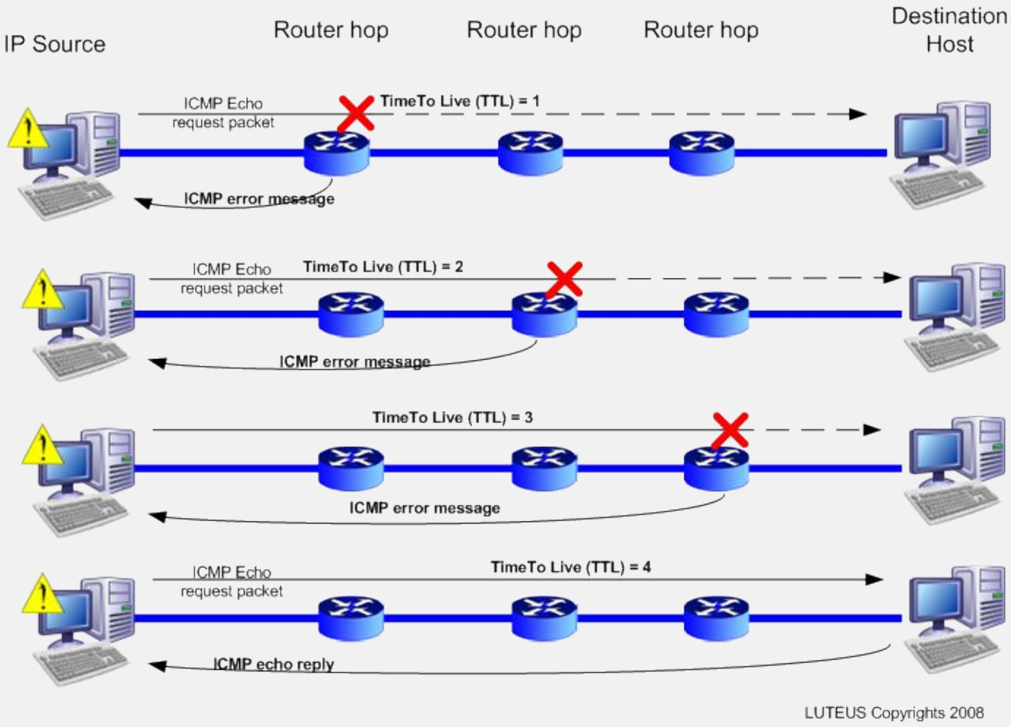
ICMP: Internet Control Message Protocol

- Define in RFC 792 in 1981.
- Used by network devices to communicate error messages and operational info.
- Layer “3” protocol:
  - Addon build on top of the IP packet.
- ICMP message: type | code | checksum | data

Type	Code	Descrip.
0	0	Echo reply (ping)
3	0	Dest. network unreachable
3	1	Dest. host unreachable
3	2	Dest. protocol unreachable
3	3	Dest. port unreachable
3	6	Dest. network unknown
3	7	Dest. host unknown
5	0-3	redirect
8	0	Echo request (ping)
9	0	Router advertisement
10	0	Router discovery
11	0	TTL expired
12	0	Bad IP header

ICMP example: **Traceroute**

- Traceroute relies on **ICMP type 11** messages (TTL expired).
- By sending packets with **increasing TTL** starting from 1.
- Can be used to debug routing, but also to discover the network (for instance during a pentest).



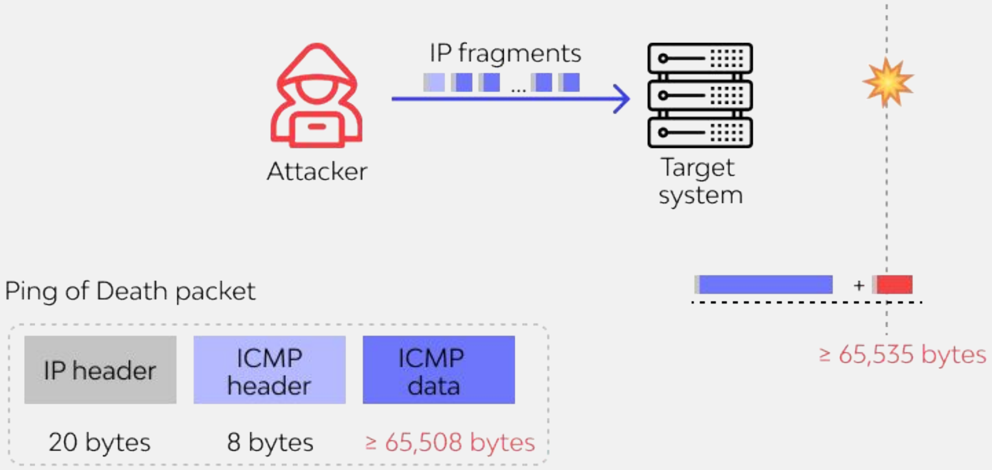
ICMP Attacks

Ping of Death (1996)

Idea: Send a malformed ICMP Echo Request (ping) packet (> 65,535 bytes).

- By splitting the message into multiple packets, the receiver will try to reconstruct an oversized message and crash, leading to a DoS.

Mitigation: Now OSes have protection against this, but in 2013 the same problem was discovered, then patched, with ICMPv6.

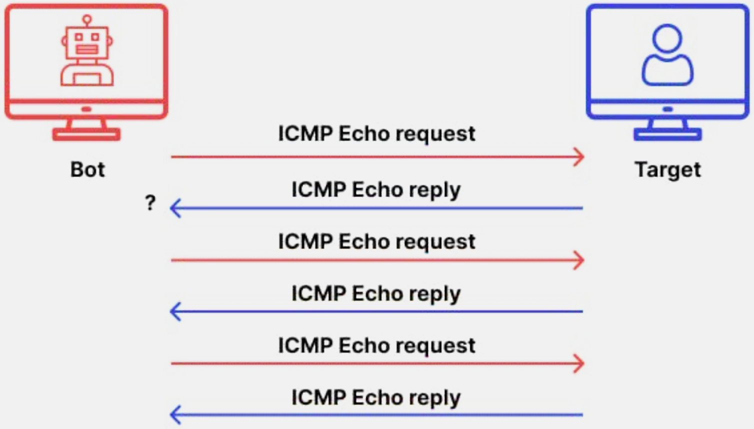


Ping Flood/ICMP Flood

Idea: Flooding a target with ICMP messages.

- The victim will try to keep up by responding to Echo request, mask request, timestamp, etc…, wasting CPU and bandwidth, leading to DoS.

Mitigation: network monitoring and Rate-limiting ICMP within hosts, firewall or routers.

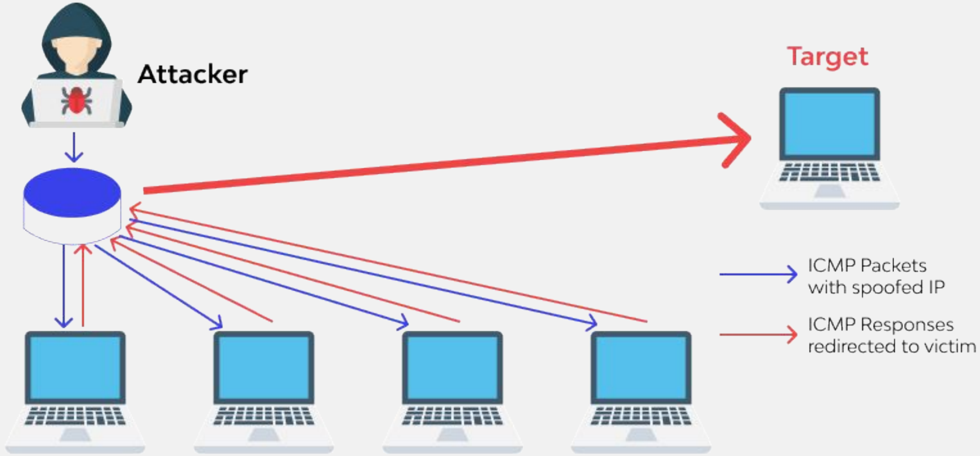


Smurf Attack

Idea: The attack sends an ICMP request with ***the victim IP as the source address*** and the ***broadcast address as the destination***.

- All devices in the network will send a response to the victim address at the same time to hopefully perform a DoS.

Mitigation: Block directed broadcasts (Router can drop messages sent to the broadcast address from outside the LAN)



Other Mitigations

Blocking ICMP?

- Highly discouraged for operational purposes.
- Not possible with ICMPv6 for IPv6 (no ARP anymore, everything is with ICMP).

Resources and Acknowledgements

- Cisco Documentation
- Computer Networking: A Top-down Approach by James F. Kurose, Keith W. Ross
- External materials from Mathieu Goessens.