BRAC UNIVERSITY

Inspiring Excellence

Network Layer: IPv4 Functions

Lecture 9 | CSE421 – Computer Networks

Department of Computer Science and Engineering
School of Data & Science

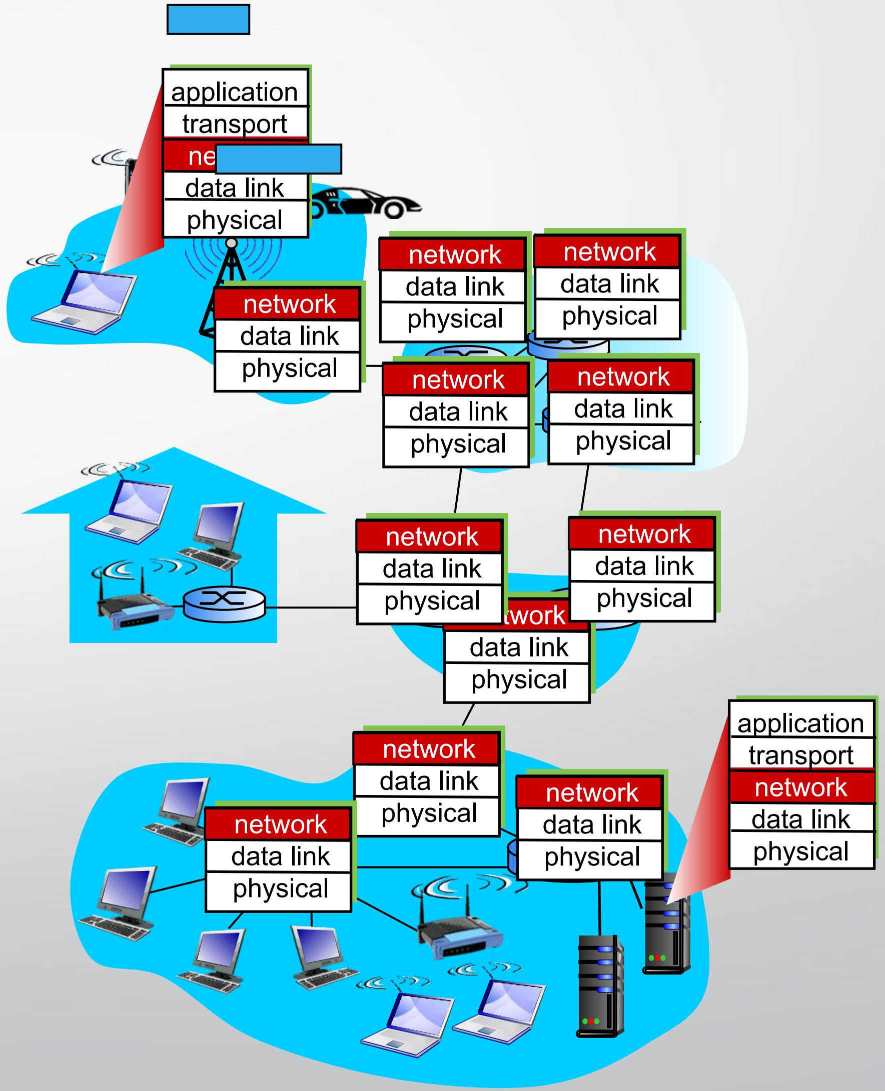
Objectives

- Short overview of the Network Layer
- Packet Switching: Virtual Circuits & Datagram Networks
- IPv4 Packet Format
- IP Fragmentation & Reassembly
- ICMP
  - Ping
  - Traceroute

The Network Layer

- Encapsulates data into **packets** on the sending side.
- Network Layer protocols operates on **hosts** and **routers**.
- **Routers** inspect IP header fields for forwarding.

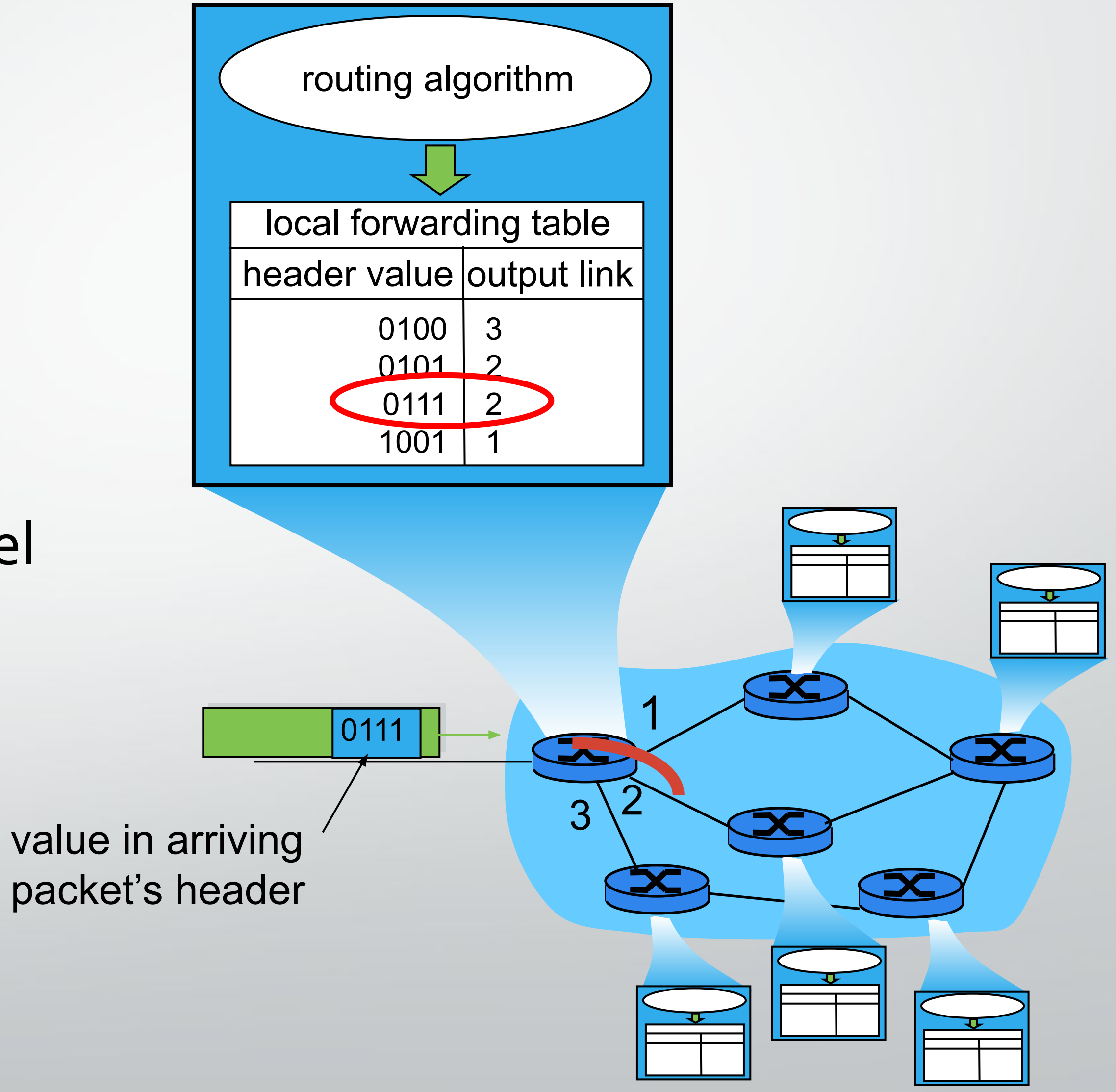
Delivers segments to the **transport layer** on the



Functions of Network Layer

- **Routing:**
  - Finds the best path from source → destination
  - Done by routing algorithms
  - Analogy: planning a trip before you travel

- **Forwarding:**
  - move packets from router's input to appropriate router output
  - Analogy: process of getting through a single interchange



local forwarding table	
header value	output link
0100	3
0101	2
0111	2
1001	1

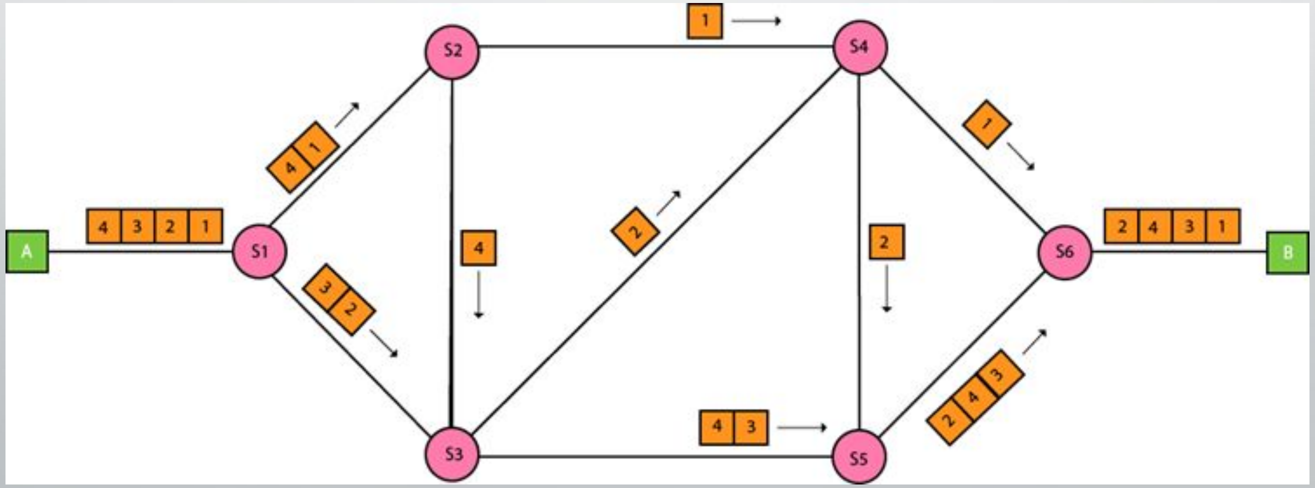
value in arriving packet's header

Packet Switching

Packet Switching

- Packet Switching is a method of transferring data across a network by breaking it into smaller packets.

- Two type of networks based on packet switching
  - **Datagram Networks**
  - **Virtual Circuit Networks**



Datagram networks

- **No Call Setup**:

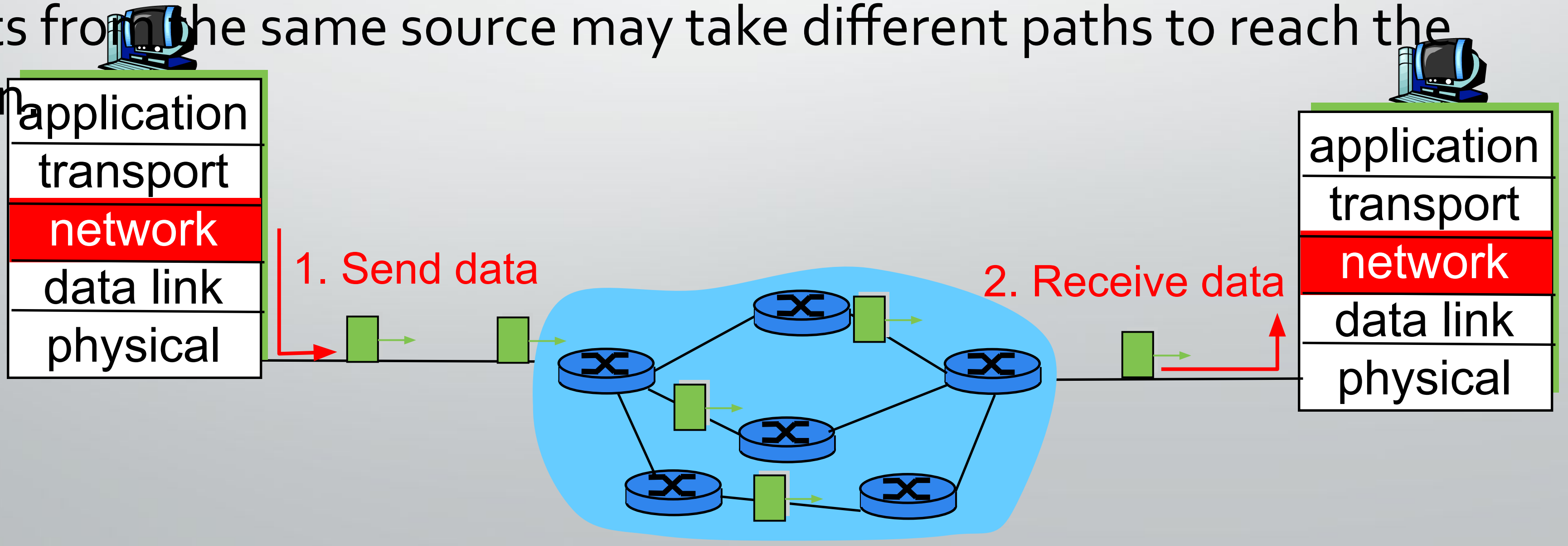
  Devices can send data immediately without establishing a connection.

- **Stateless Routers**:

  Routers forward packets independently based on their destination IP address.

- **Packet Forwarding**:

  Packets from the same source may take different paths to reach the destination.



Virtual Circuits: Signaling Protocols

- **Call Setup**:

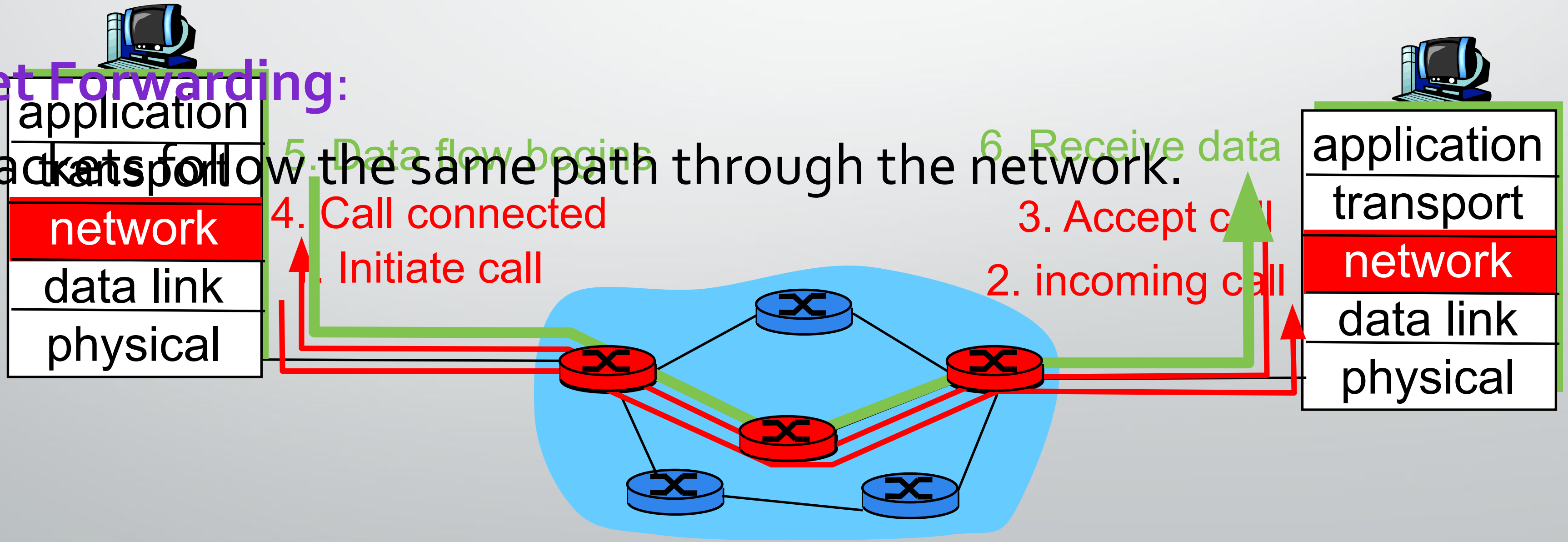
  A connection (virtual circuit) is established between sender and receiver before data transfer.

- **Stateful Routers**:

  Routers maintain information about active connections (virtual circuits).

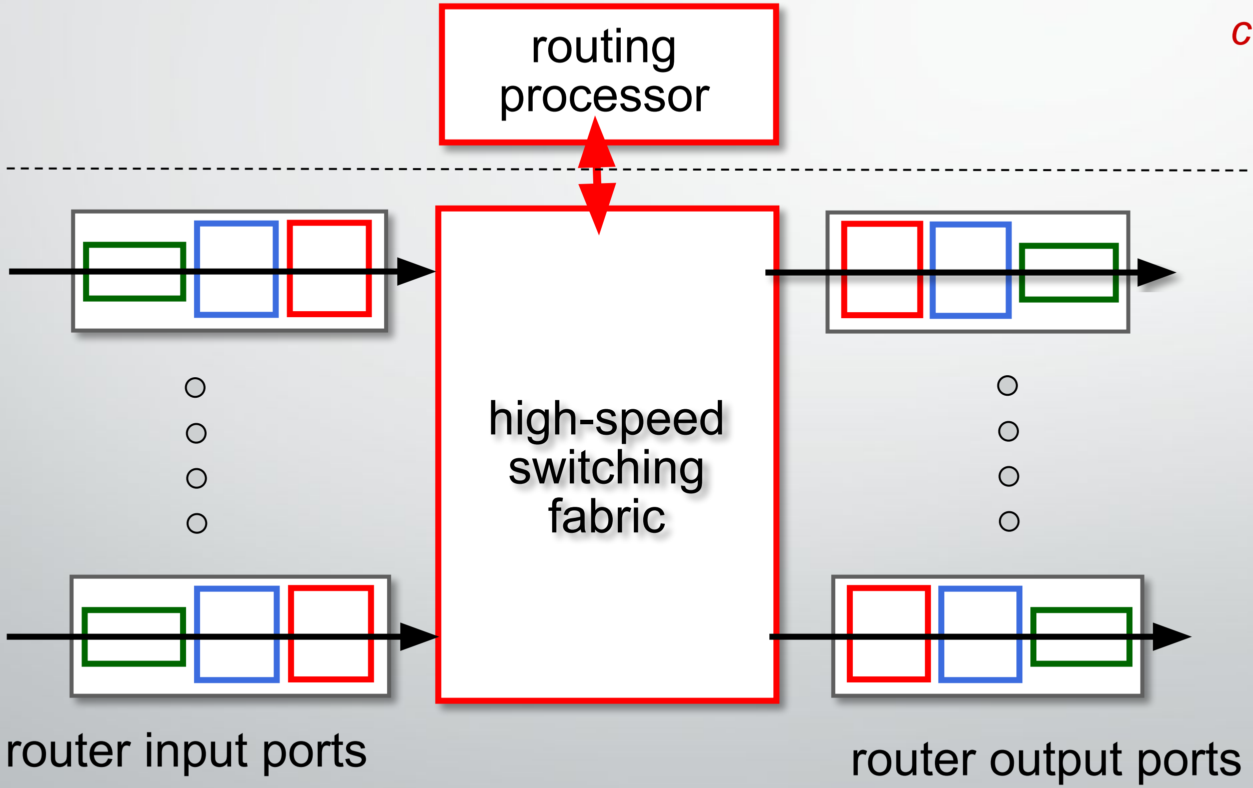
- **Packet Forwarding**:

  All packets follow the same path through the network.



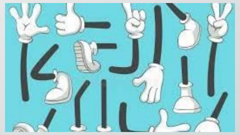
Router architecture overview

high-level view of generic router architecture:



routing, management
control plane **(software)**
operates in millisecond
time frame

forwarding data plane
(hardware) operates
in nanosecond
timeframe



Control Plane – the brain!

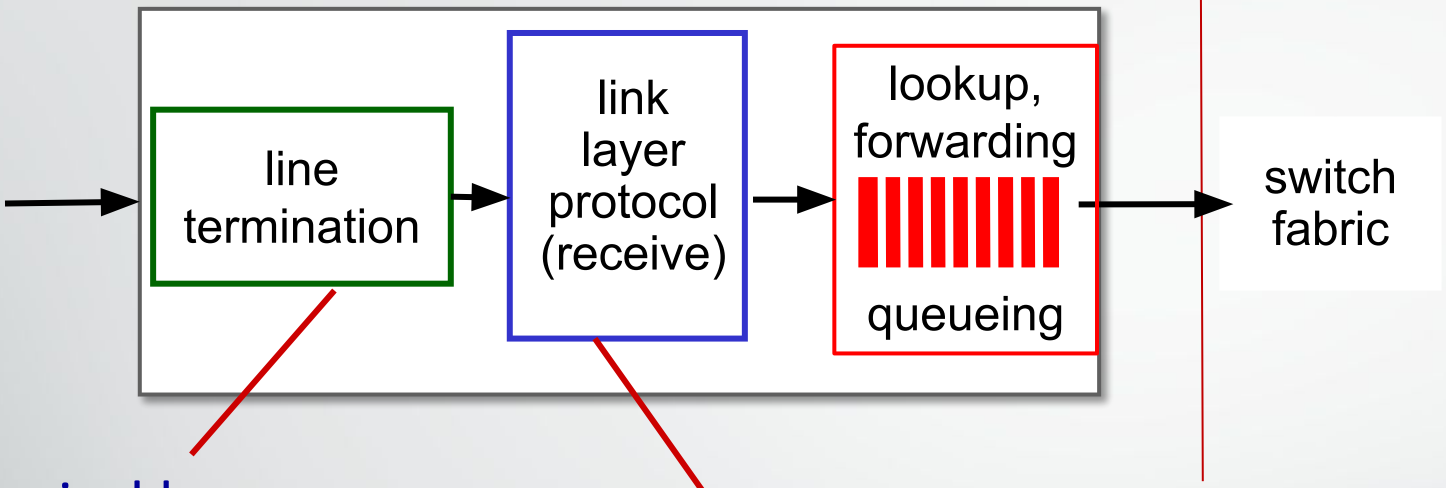
- Control plane builds the **routing table** using routing protocols.
- It creates the **forwarding table FIB** (Forwarding Information Base), a simplified version of routing table.
- The FIB may be created using
  - simple destination-based rules
  - or more advanced generalized rules.
- Control plane installs this FIB into every input port.

Decentralized Switching

Data Plane – the hands and legs!

- How input ports use the FIB?
- Forwarding rules can be:
  - **Destination-based forwarding :**
    - Uses **only destination IP** to choose output port
    - If destination = 200.20.20.0/24 → send to port s0/0/0.
  - **Generalized forwarding :**
    - Admins can add **multiple header fields** (IP, protocol, port, etc.) Or special rules (QoS)
    - If video packet (UDP, port 5000) → high-priority queue.
- These rules are stored in the FIB created by the control

Input port functions



physical layer:

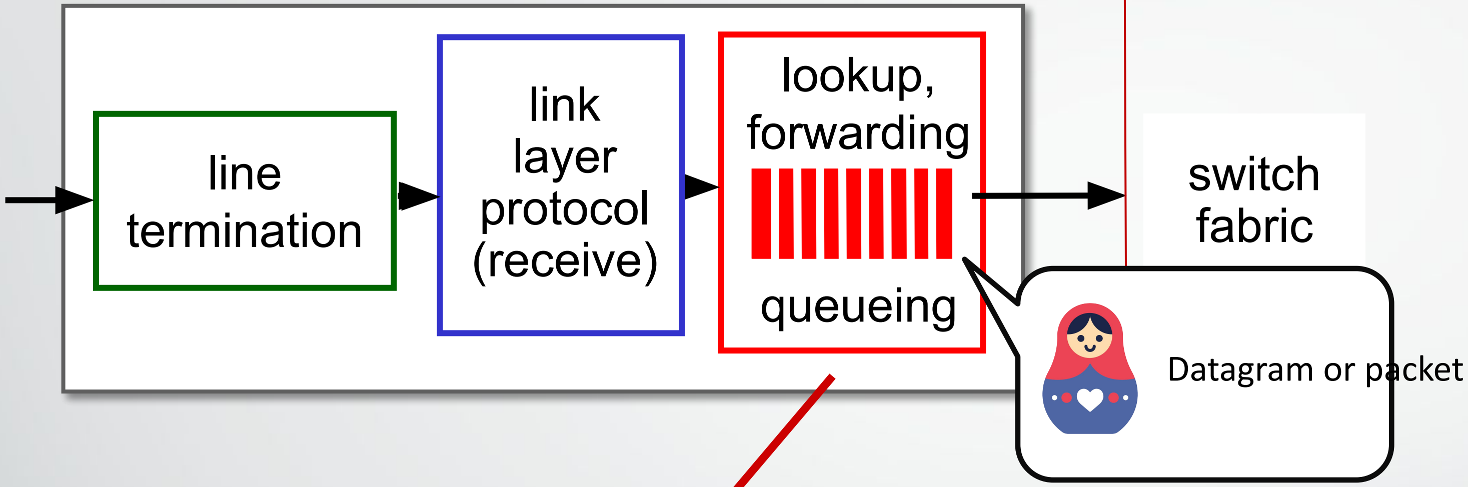
- Converts signal ⟶ digital bits

link layer:

- Removes data link header/trailer
- Checks for errors
- Extracts the IP packet

Frame

Input port functions



Network layer:

- Input port checks packet header
- Matches against local FIB copy
- Decides output port (using destination-based or generalized rule)

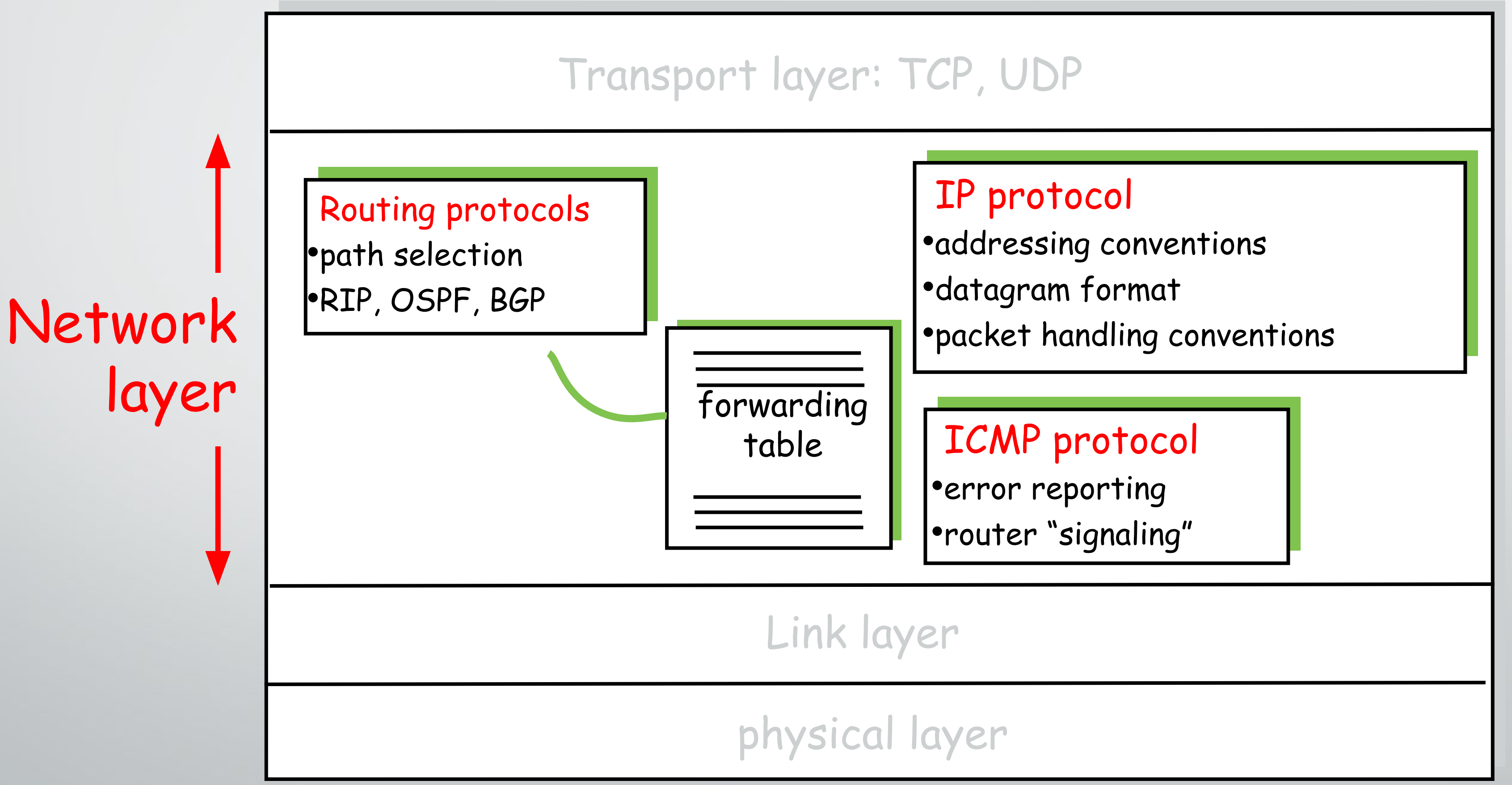
Input Port Queueing

- If packets arrive too fast → they wait here
- If queue is full → packet is dropped

Internet Protocol IPv4

Internet Network Layer

- Host, router network layer functions:



Transport layer: TCP, UDP

Routing protocols
- path selection
- RIP, OSPF, BGP

forwarding table

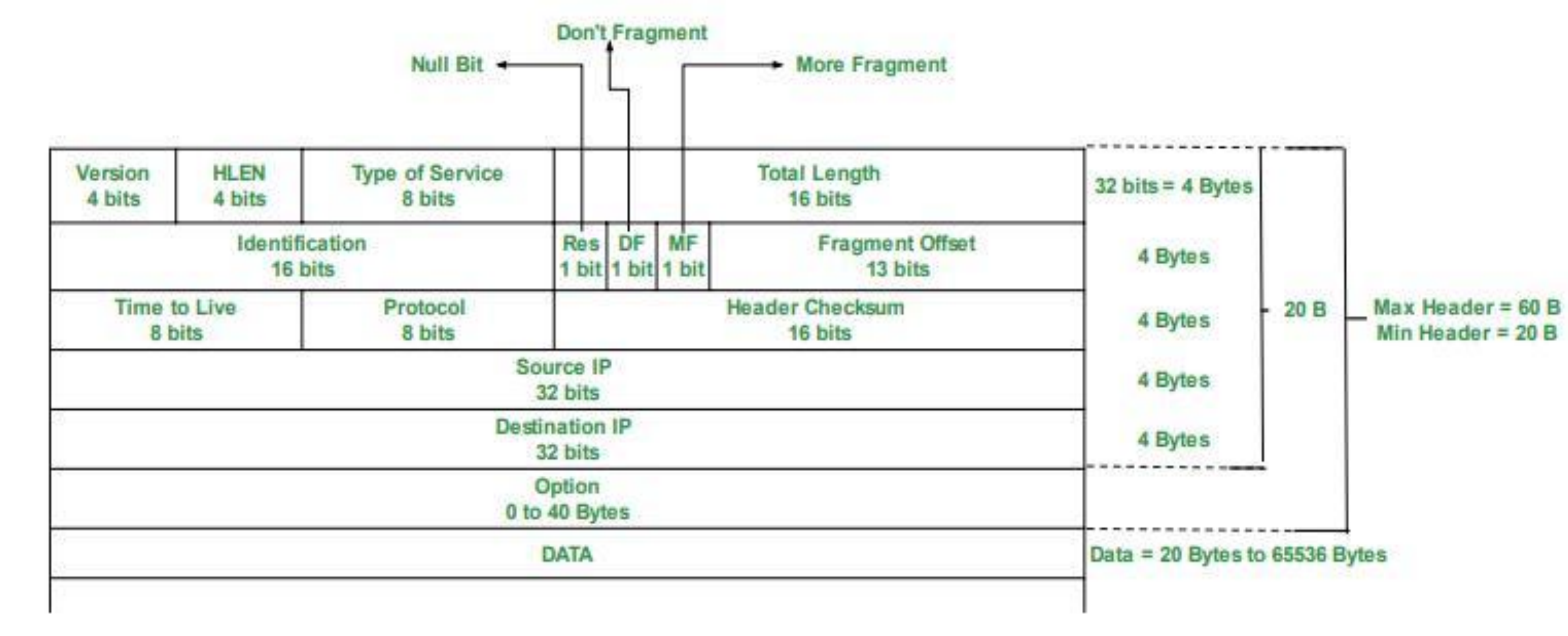
IP protocol
- addressing conventions
- datagram format
- packet handling conventions

ICMP protocol
- error reporting
- router "signaling"

Link layer

physical layer

IPv4 Datagram Format

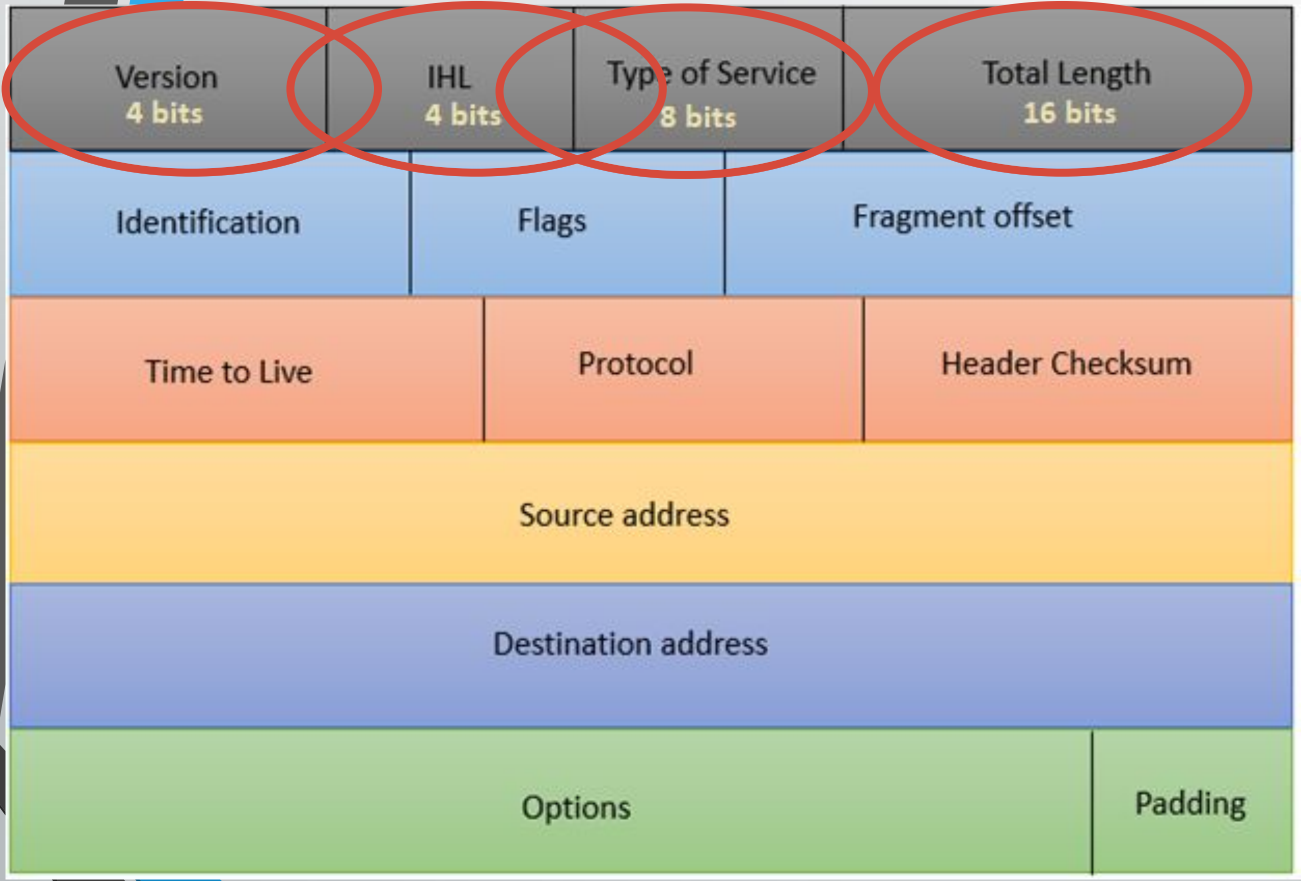


Version 4 bits	HLEN 4 bits	Type of Service 8 bits	Total Length 16 bits			
Identification 16 bits			Res 1 bit	DF 1 bit	MF 1 bit	Fragment Offset 13 bits
Time to Live 8 bits		Protocol 8 bits	Header Checksum 16 bits			
Source IP 32 bits						
Destination IP 32 bits						
Option 0 to 40 Bytes						
DATA						

The size of an IP datagram:

- The **minimum** size is **20 bytes** (if you have no data)
- The **maximum** size is **65,535 bytes**

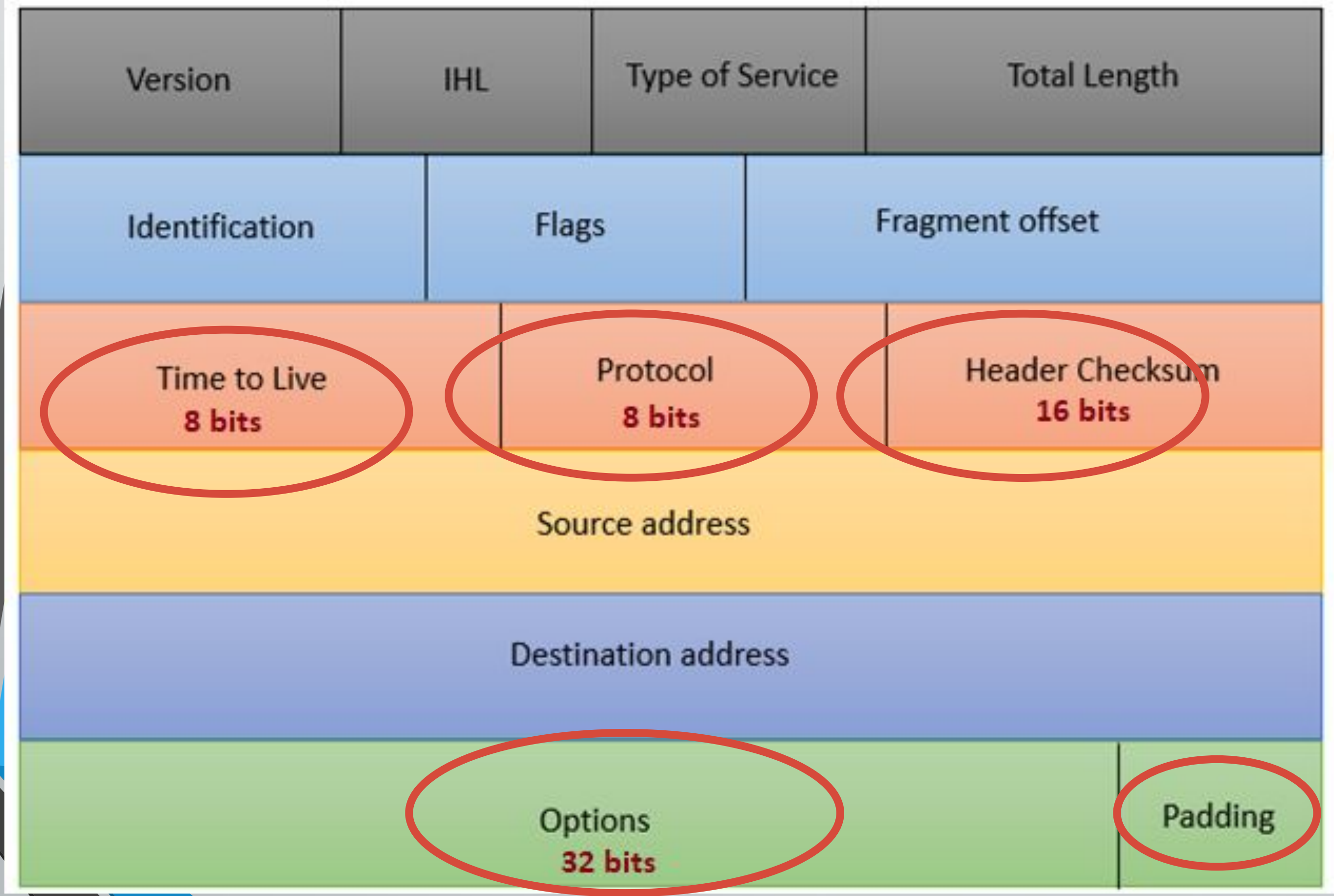
IPv4 Datagram Header Format



Version 4 bits	IHL 4 bits	Type of Service 8 bits	Total Length 16 bits
Identification	Flags	Fragment offset	
Time to Live	Protocol	Header Checksum	
Source address			
Destination address			
Options	Padding		

- **Version:** value of which IP version is being used. For IPv4 the value will be 4 here.
- **Internet Header Length**: value of the header length, min 20 bytes, max 60 bytes. Shown in 4 byte word. **So min value 5, max 15.**
- **Type of Service**: for QoS (Quality of Service). To mark the packet to give special treatment or priority.
- **Total Length:** value of the entire size of the IP packet (header and data) in bytes.

IPv4 Datagram Header Format



Version | IHL | Type of Service | Total Length

Identification | Flags | Fragment offset

Time to Live 8 bits | Protocol 8 bits | Header Checksum 16 bits

Source address

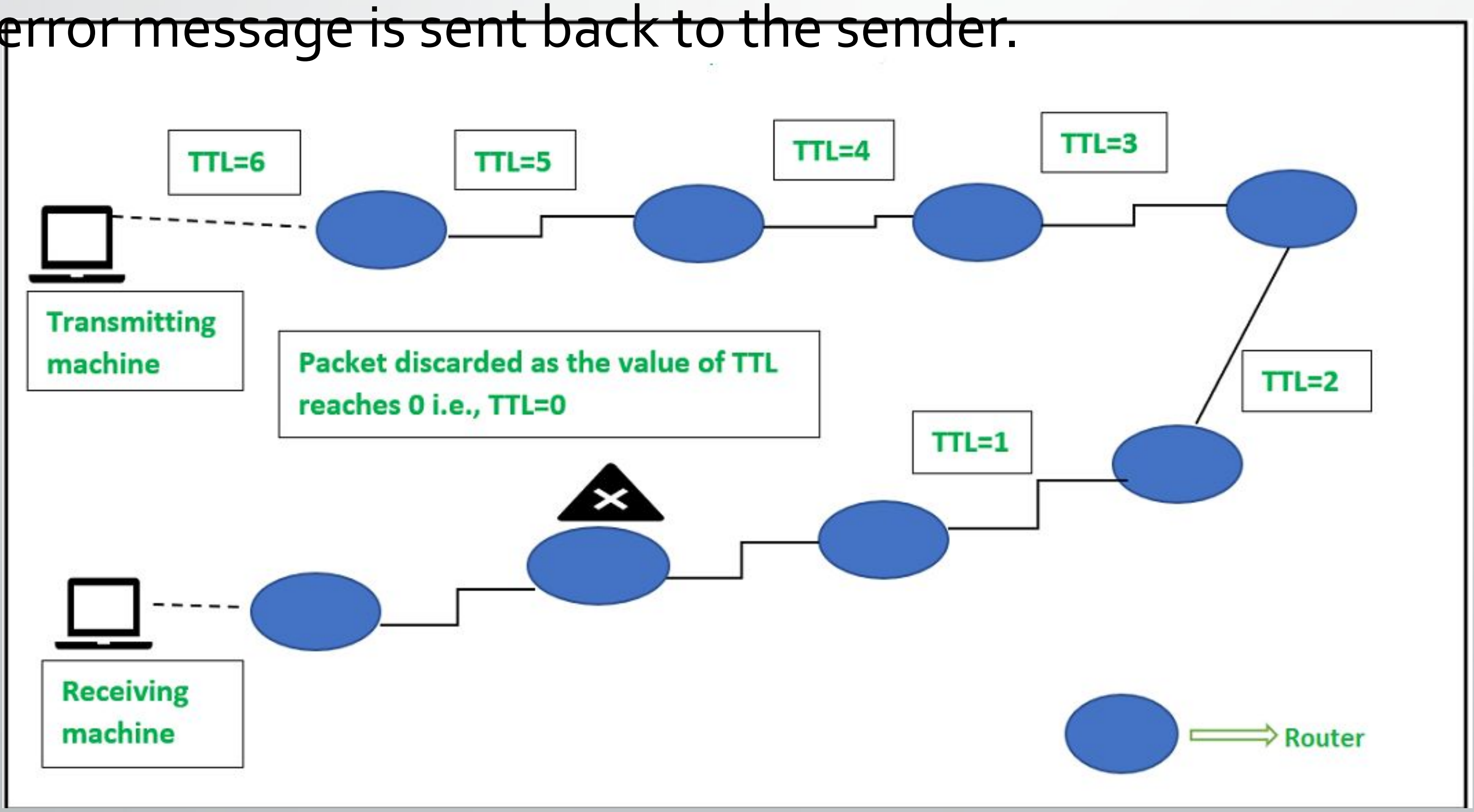
Destination address

Options 32 bits | Padding

- **Time to Live:** maximum number of **hops** (routers) a packet can travel
- **Protocol:** value tells us which upper layer protocol is present, for example **TCP** has value **6** and **UDP** has value **17**.
- **Header Checksum**: to check if there are any errors in the header.
- **Options:** value of any extra information. Options are rarely used now.
- **Padding:** Used only when Options are present. Ensures the header length becomes a multiple of 4 bytes

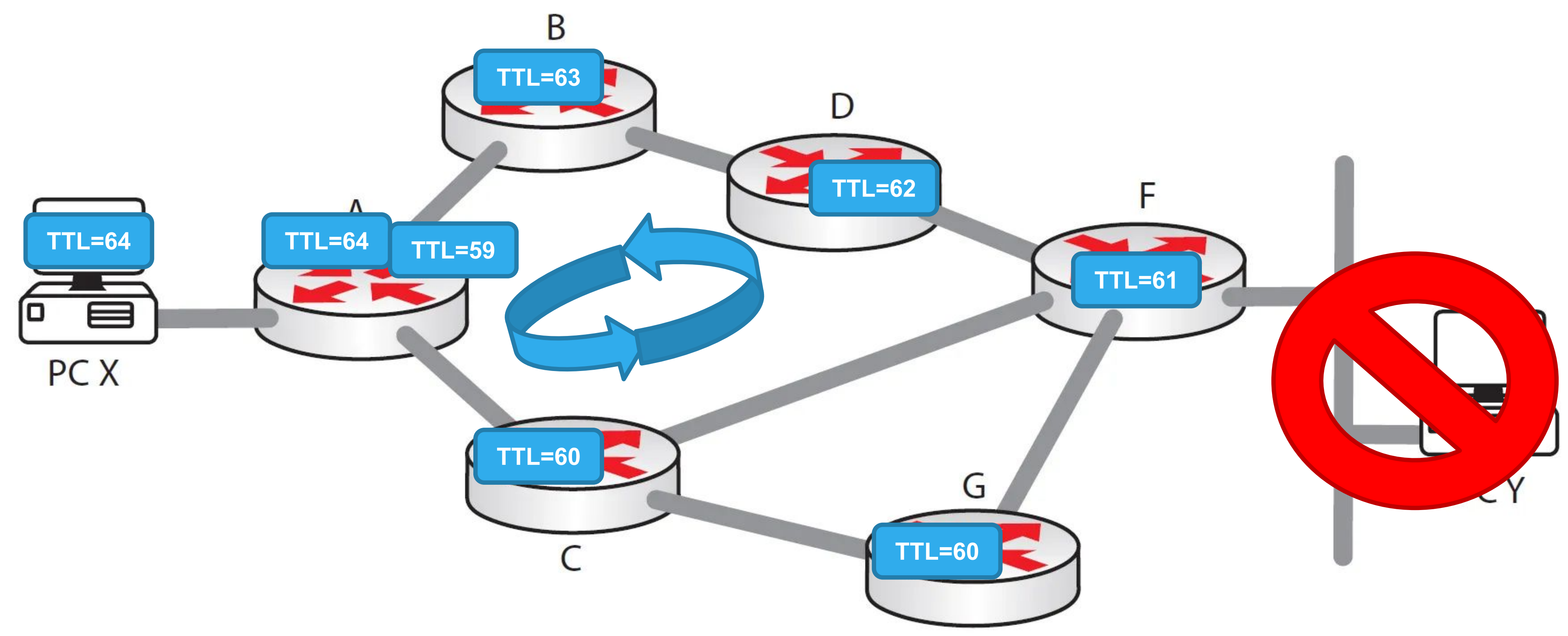
Time to Live - TTL

- Maximum number of **hops** (routers) a packet can traverse before being discarded.
- At each hop, the TTL is decreased by **1**.
- When the TTL reaches **0**, the packet is dropped.
- And an error message is sent back to the sender.

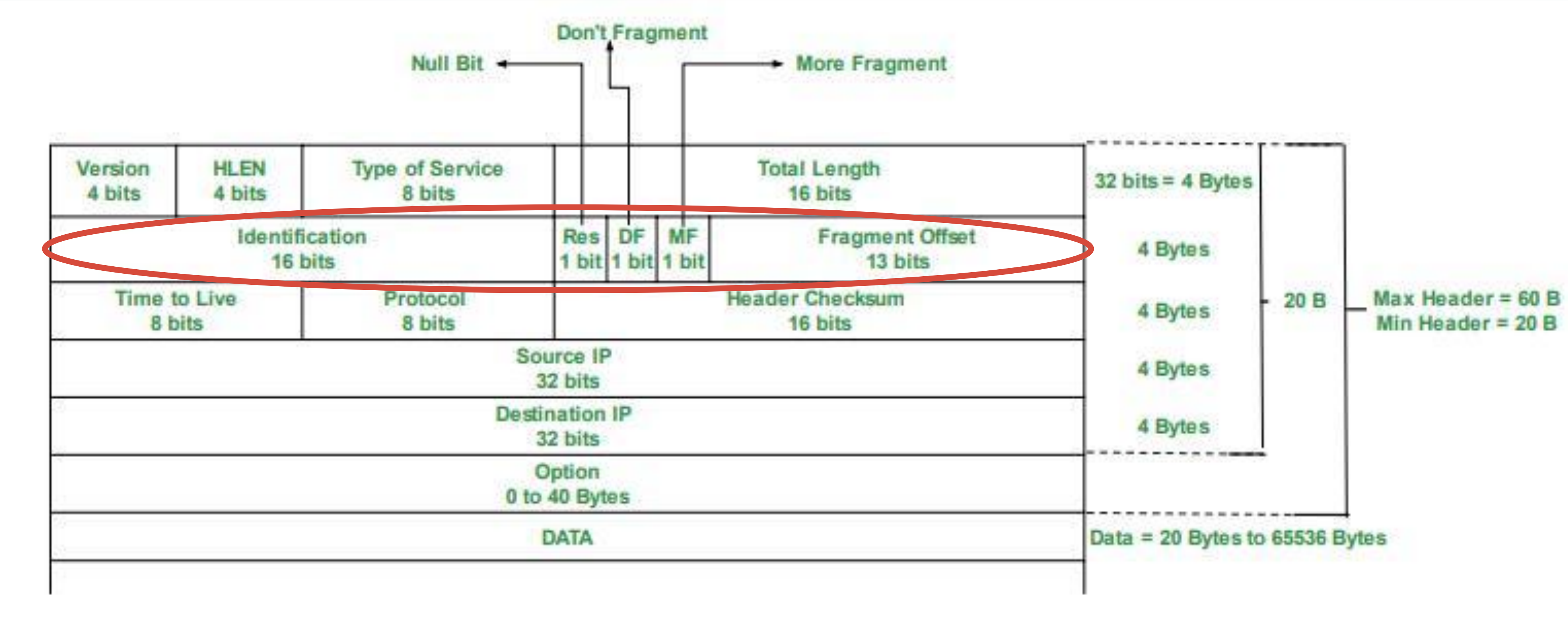


Time to Live - TTL

- **Not** just the "value of hops"
- It's a mechanism to prevent packets from **looping endlessly** in the network.
- Ensure finite packet lifetimes.



IPv4 Datagram Format



Don't Fragment

Null Bit

More Fragment

Version 4 bits	HLEN 4 bits	Type of Service 8 bits	Total Length 16 bits

Identification 16 bits	Res 1 bit	DF 1 bit	MF 1 bit	Fragment Offset 13 bits

Time to Live 8 bits	Protocol 8 bits	Header Checksum 16 bits

Source IP
32 bits

Destination IP
32 bits

Option
0 to 40 Bytes

DATA

32 bits = 4 Bytes

4 Bytes

4 Bytes

4 Bytes

4 Bytes

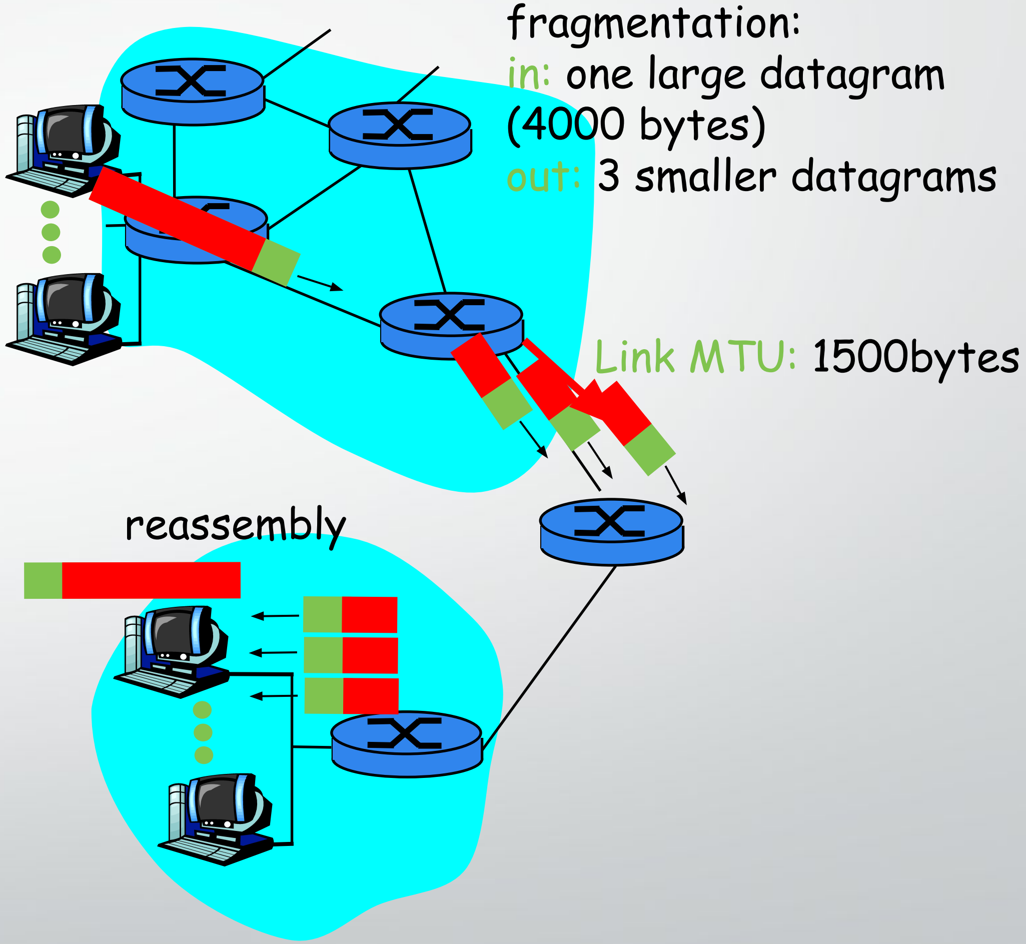
20 B

Max Header = 60 B
Min Header = 20 B

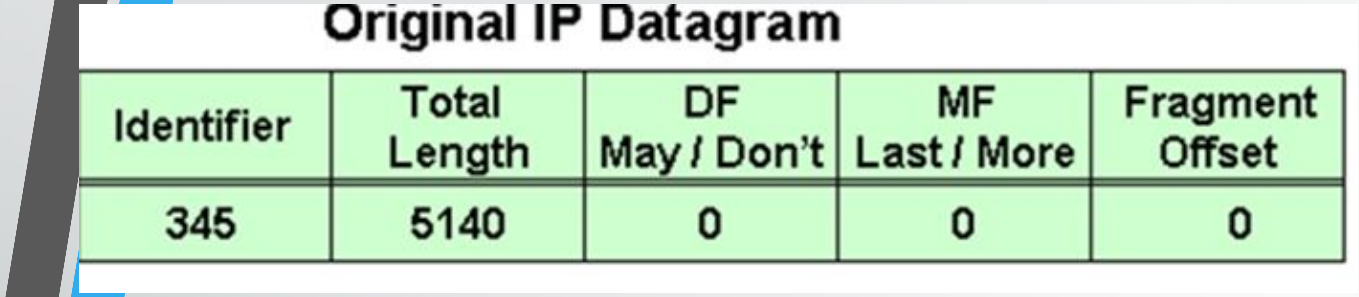
Data = 20 Bytes to 65536 Bytes

IP Fragmentation & Reassembly

- Network links have **MTU**
  - Maximum transmission unit or maximum transfer size
  - Different link types have different MTUs



IP Fragmentation & Reassembly



Original IP Datagram

Identifier	Total Length	DF May / Don't	MF Last / More	Fragment Offset
345	5140	0	0	0

Data 5140=> 20(H)+5120(D)

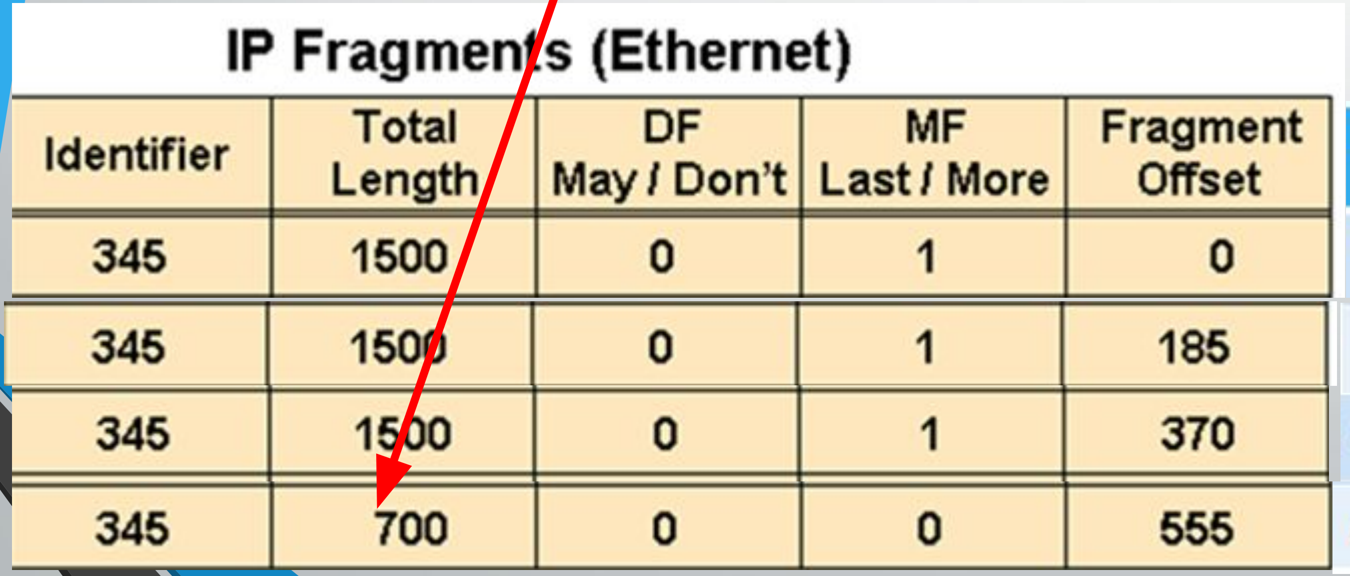
MTU=1500 => 20(H)+1480(D)

1st fragment : 5120-1480=3640

2nd fragment : 3640-1480=2160

3rd fragment : 2160-1480=680

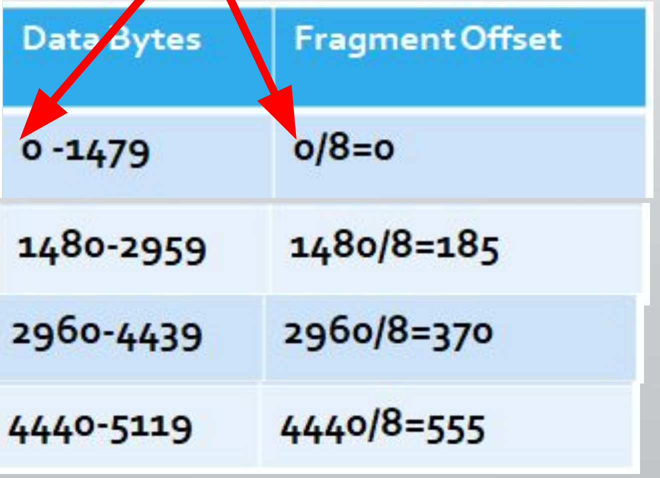
Last/4th Segment : 680+20=700



IP Fragments (Ethernet)

Identifier	Total Length	DF May / Don't	MF Last / More	Fragment Offset
345	1500	0	1	0
345	1500	0	1	185
345	1500	0	1	370
345	700	0	0	555

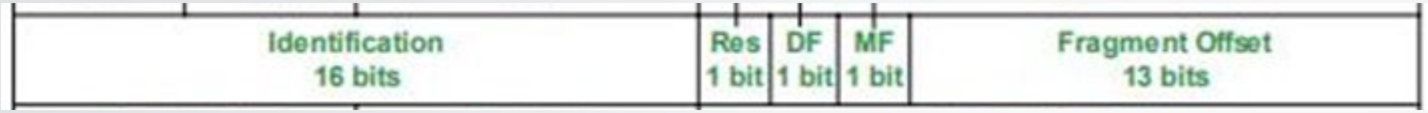
1st / start byte number



Data Bytes	Fragment Offset
0 -1479	0/8=0
1480-2959	1480/8=185
2960-4439	2960/8=370
4440-5119	4440/8=555

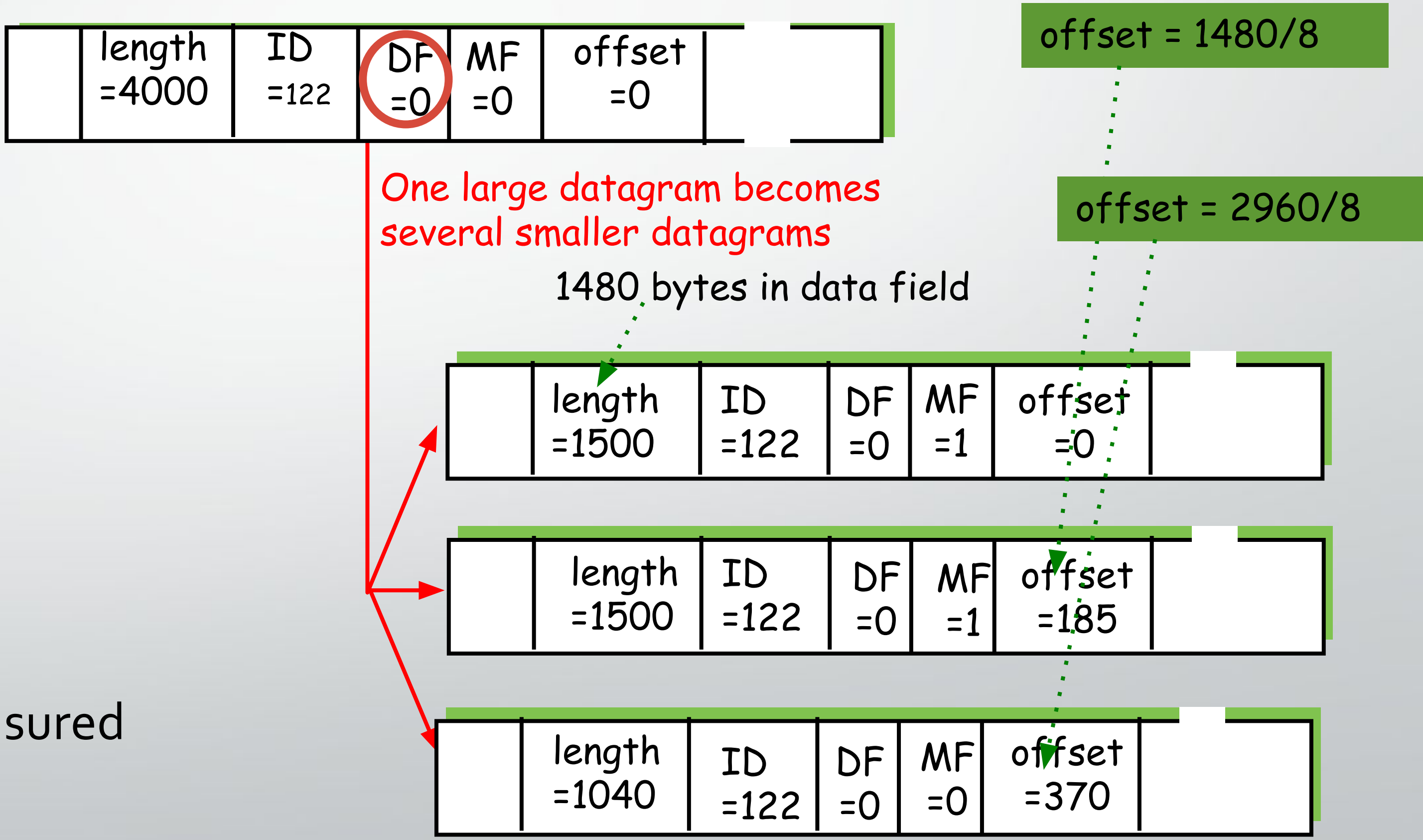
IP Fragmentation & Reassembly

Extra Example



Identification 16 bits	Res 1 bit	DF 1 bit	MF 1 bit	Fragment Offset 13 bits

- Example:
  - 4000 Bytes of datagram
  - MTU = 1500 Bytes
- **DF – Don't Fragment Bit**
  - Value 0 or 1
- **Fragment Offset**
  - The value of the offset is measured in units of 8 bytes.



length	ID	DF	MF	offset
=4000	=122	=0	=0	=0

One large datagram becomes several smaller datagrams

1480 bytes in data field

offset = 1480/8

offset = 2960/8

length	ID	DF	MF	offset
=1500	=122	=0	=1	=0
=1500	=122	=0	=1	=185
=1040	=122	=0	=0	=370

ICMP

Internet Control Message Protocol

ICMP

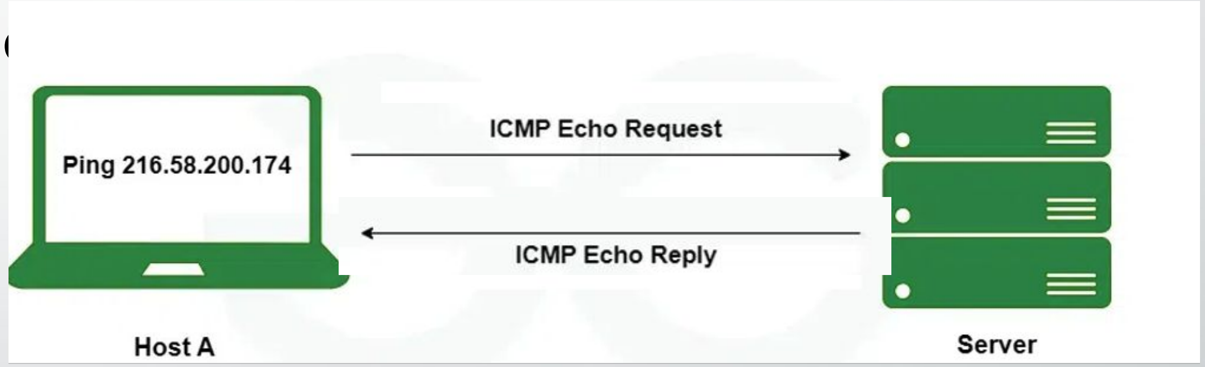
- It helps devices send error messages and status updates.
- **Functions**:
  - Reporting errors in the network
  - Checking reachability (Is the host alive?)
  - Diagnosing delays and congestion

Network Error

- **Key Points**:
  - ICMP does not carry actual user data
  - Mainly used by the **operating systems** for network management
- **Example of ICMP in practice**
  - **Ping**
  - **Traceroute**

Ping

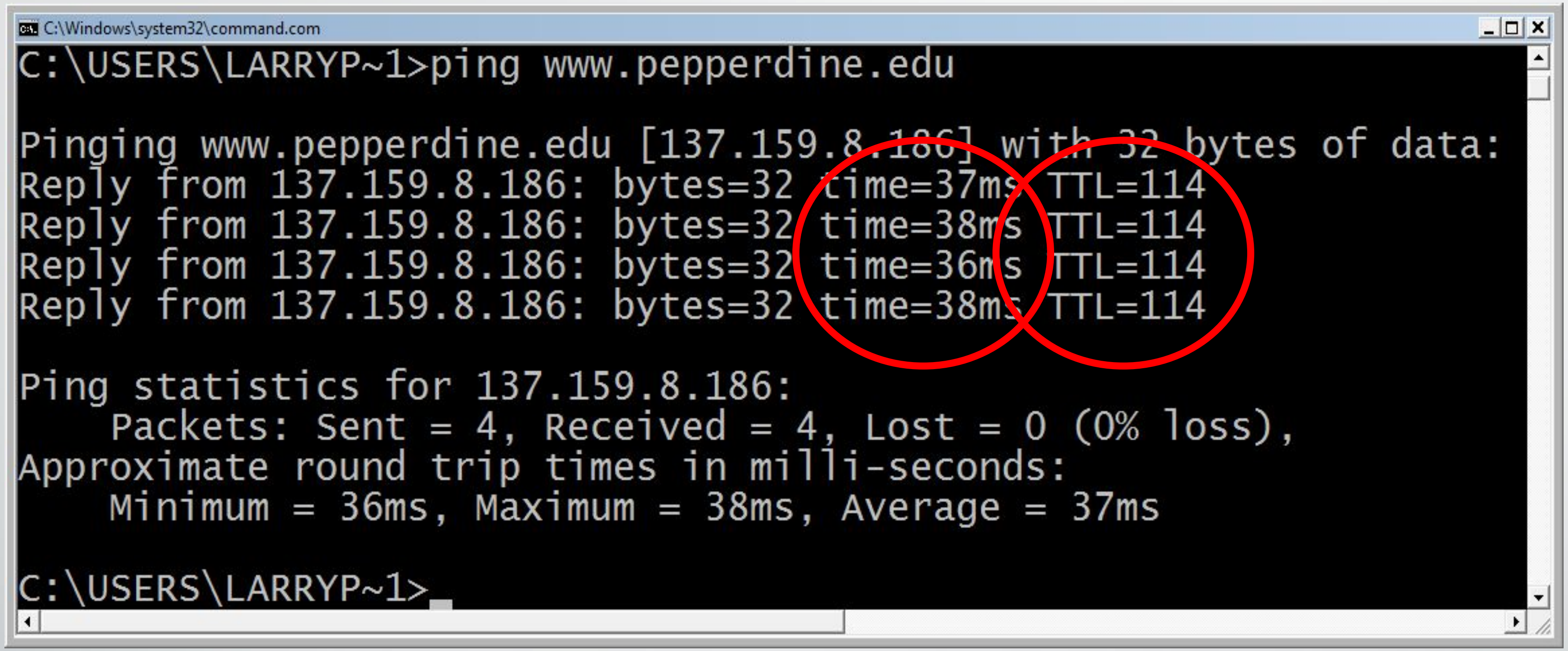
- Packet Internet Groper and is a network utility tool.
- **Purpose**:
- Checks if a device is reachable
- Measures the time it takes for data t
- **Mechanism**:
- Sends ICMP Echo Request packets
- Receives Echo Reply packets.
- **Results**:



- Commands:

ping 216.58.200.174

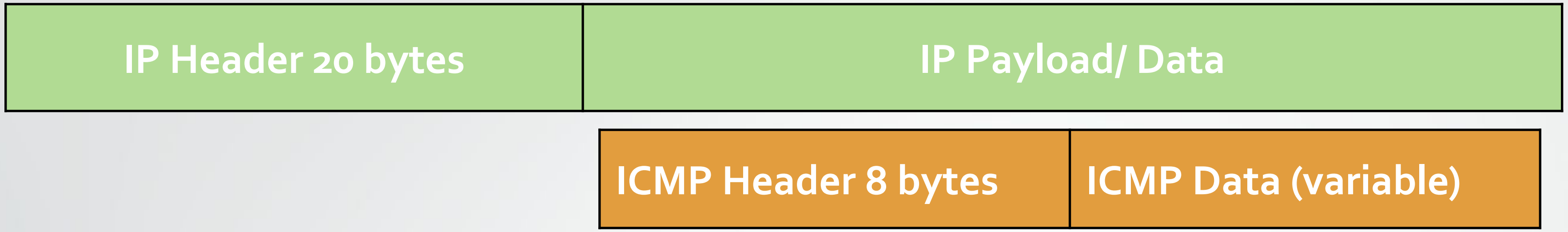
Ping



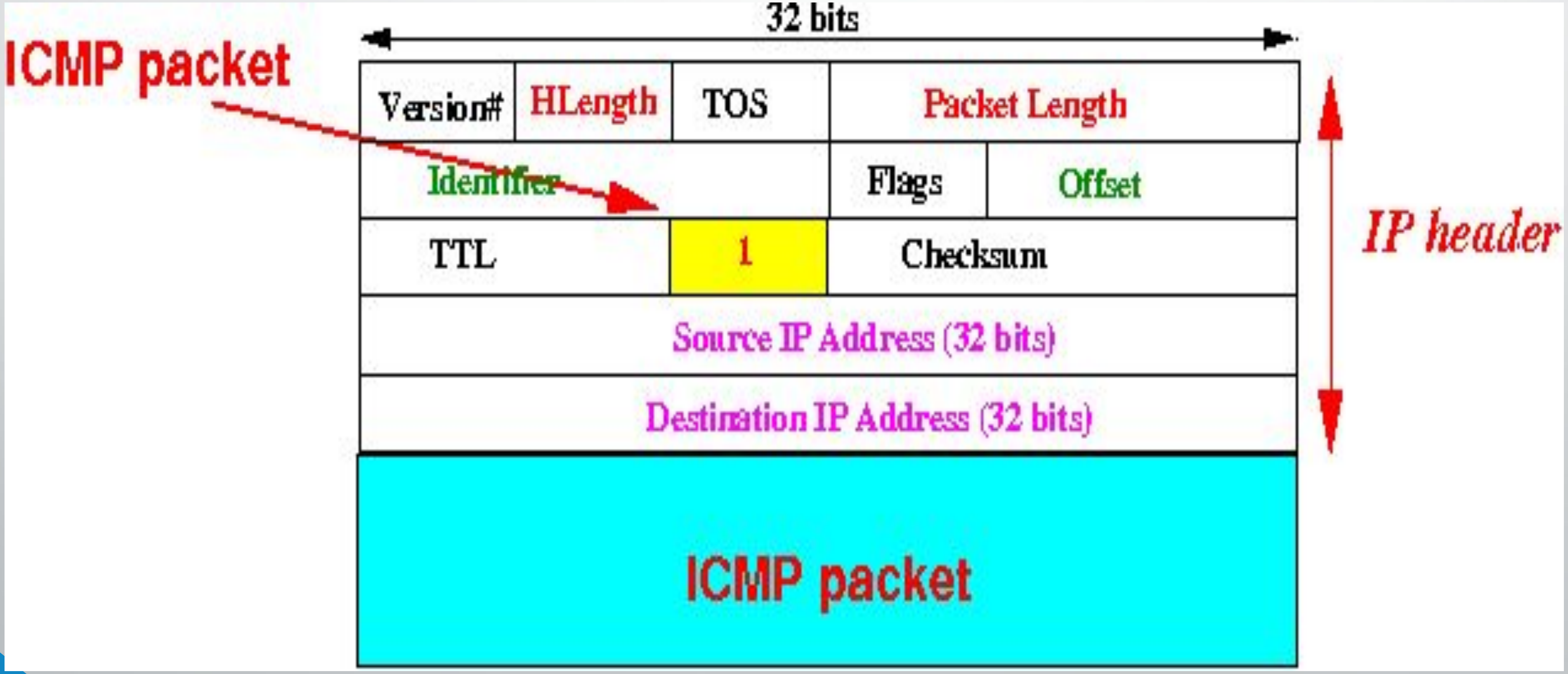
```
C:\USERS\LARRYP~1>ping www.pepperdine.edu

Pinging www.pepperdine.edu [137.159.8.186] with 32 bytes of data:
Reply from 137.159.8.186: bytes=32 time=37ms TTL=114
Reply from 137.159.8.186: bytes=32 time=38ms TTL=114
Reply from 137.159.8.186: bytes=32 time=36ms TTL=114
Reply from 137.159.8.186: bytes=32 time=38ms TTL=114

Ping statistics for 137.159.8.186:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 36ms, Maximum = 38ms, Average = 37ms

C:\USERS\LARRYP~1>_
```

- **Questions :**
- Why 4 replies?
- What the time refer to?

ICMP Packet Format

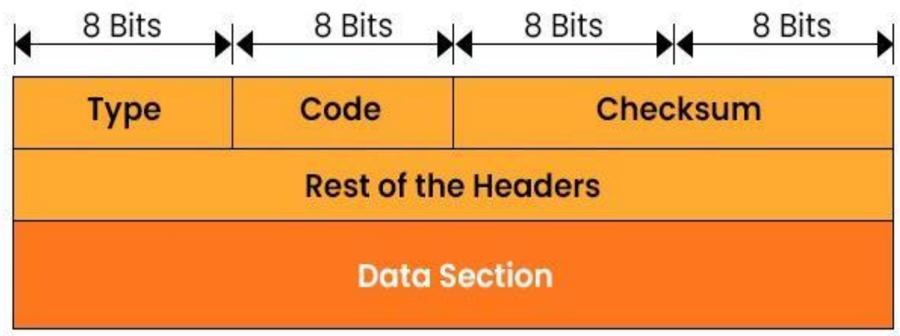


IP Header 20 bytes	IP Payload/ Data
	ICMP Header 8 bytes | ICMP Data (variable)

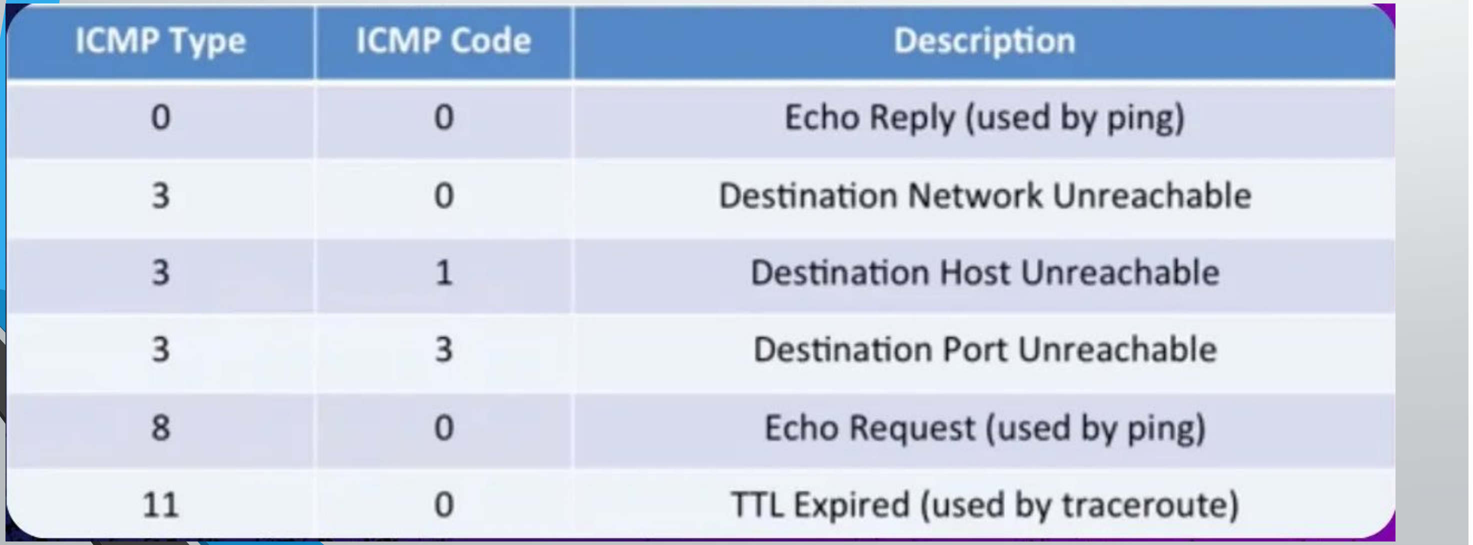


Version#	HLength	TOS	Packet Length
Identifier		Flags	Offset
TTL	1	Checksum	
Source IP Address (32 bits)			
Destination IP Address (32 bits)			
ICMP packet			

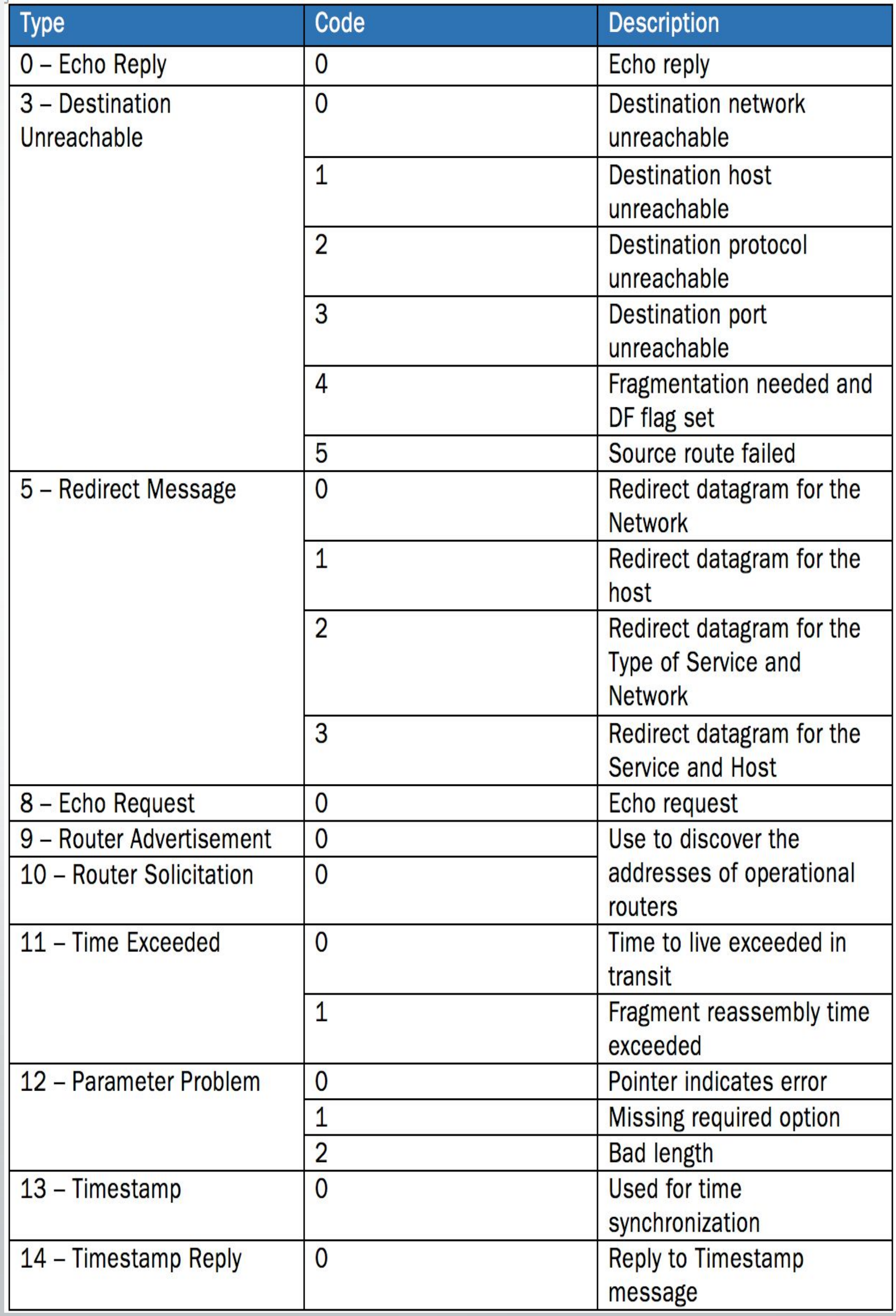
ICMP Packet Format



ICMP Message Format

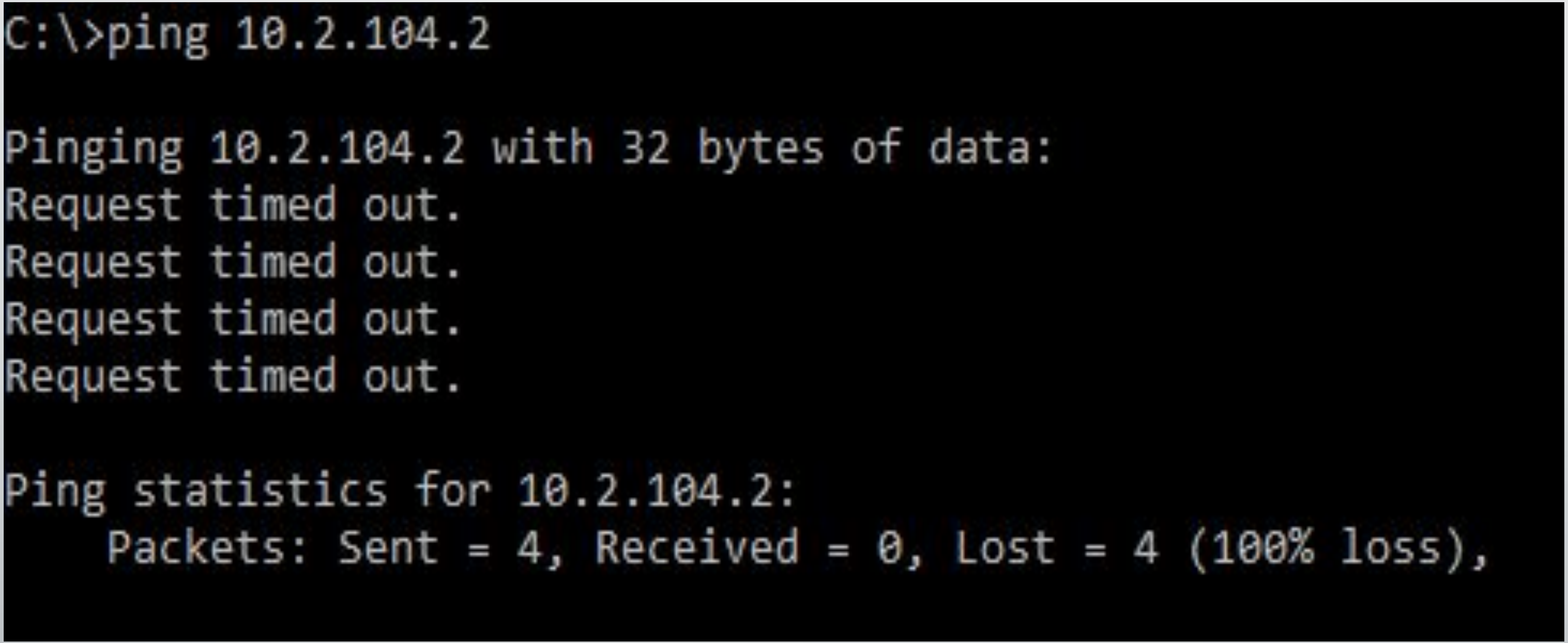


ICMP Type	ICMP Code	Description
0	0	Echo Reply (used by ping)
3	0	Destination Network Unreachable
3	1	Destination Host Unreachable
3	3	Destination Port Unreachable
8	0	Echo Request (used by ping)
11	0	TTL Expired (used by traceroute)



Type	Code	Description
0 – Echo Reply	0	Echo reply
3 – Destination Unreachable	0	Destination network unreachable
	1	Destination host unreachable
	2	Destination protocol unreachable
	3	Destination port unreachable
	4	Fragmentation needed and DF flag set
	5	Source route failed
5 – Redirect Message	0	Redirect datagram for the Network
	1	Redirect datagram for the host
	2	Redirect datagram for the Type of Service and Network
	3	Redirect datagram for the Service and Host
8 – Echo Request	0	Echo request
9 – Router Advertisement	0	Use to discover the addresses of operational routers
10 – Router Solicitation	0	
11 – Time Exceeded	0	Time to live exceeded in transit
	1	Fragment reassembly time exceeded
12 – Parameter Problem	0	Pointer indicates error
	1	Missing required option
	2	Bad length
13 – Timestamp	0	Used for time synchronization
14 – Timestamp Reply	0	Reply to Timestamp message

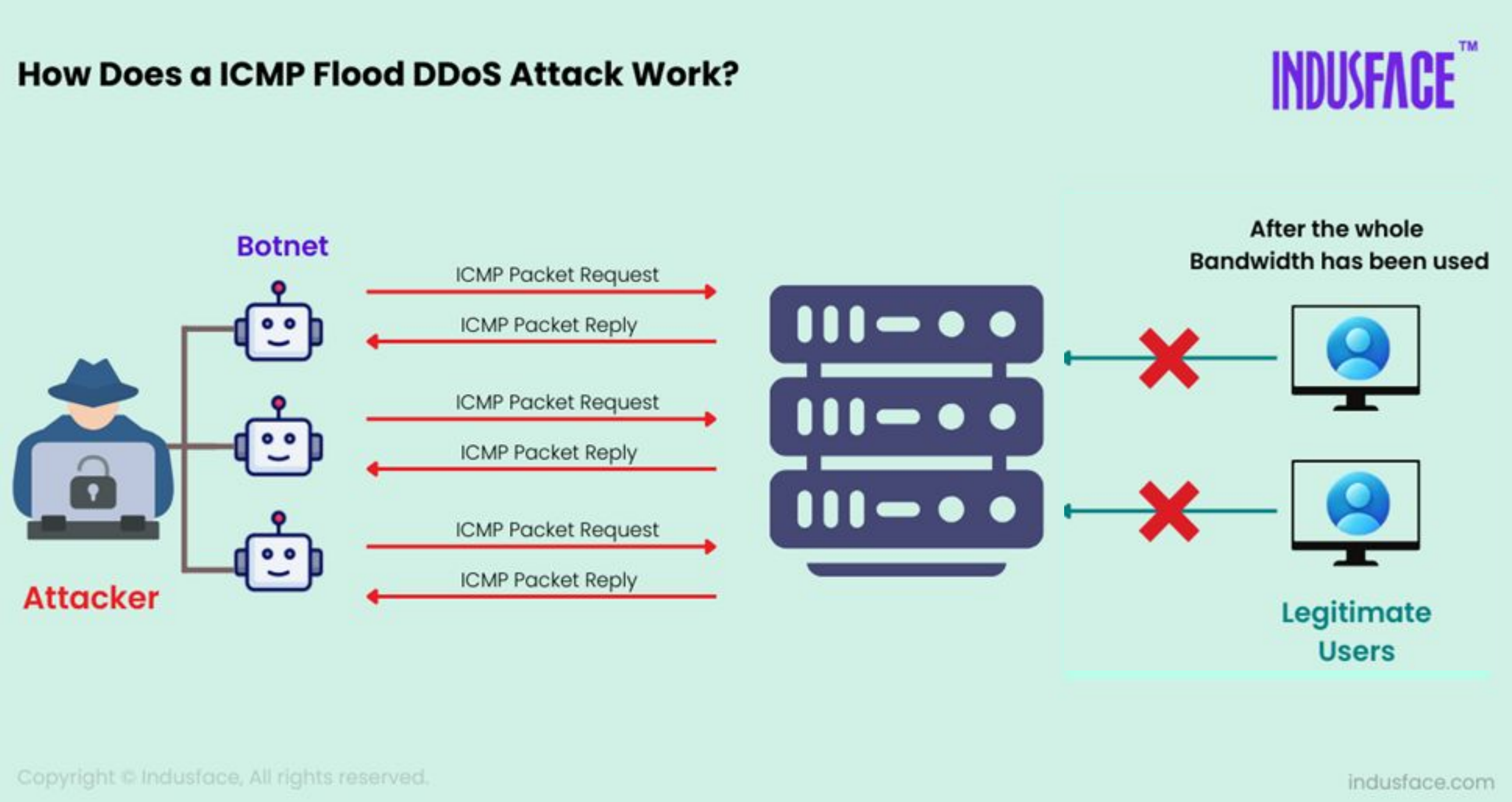
Unsuccessful Ping



```
C:\>ping 10.2.104.2

Pinging 10.2.104.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.2.104.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

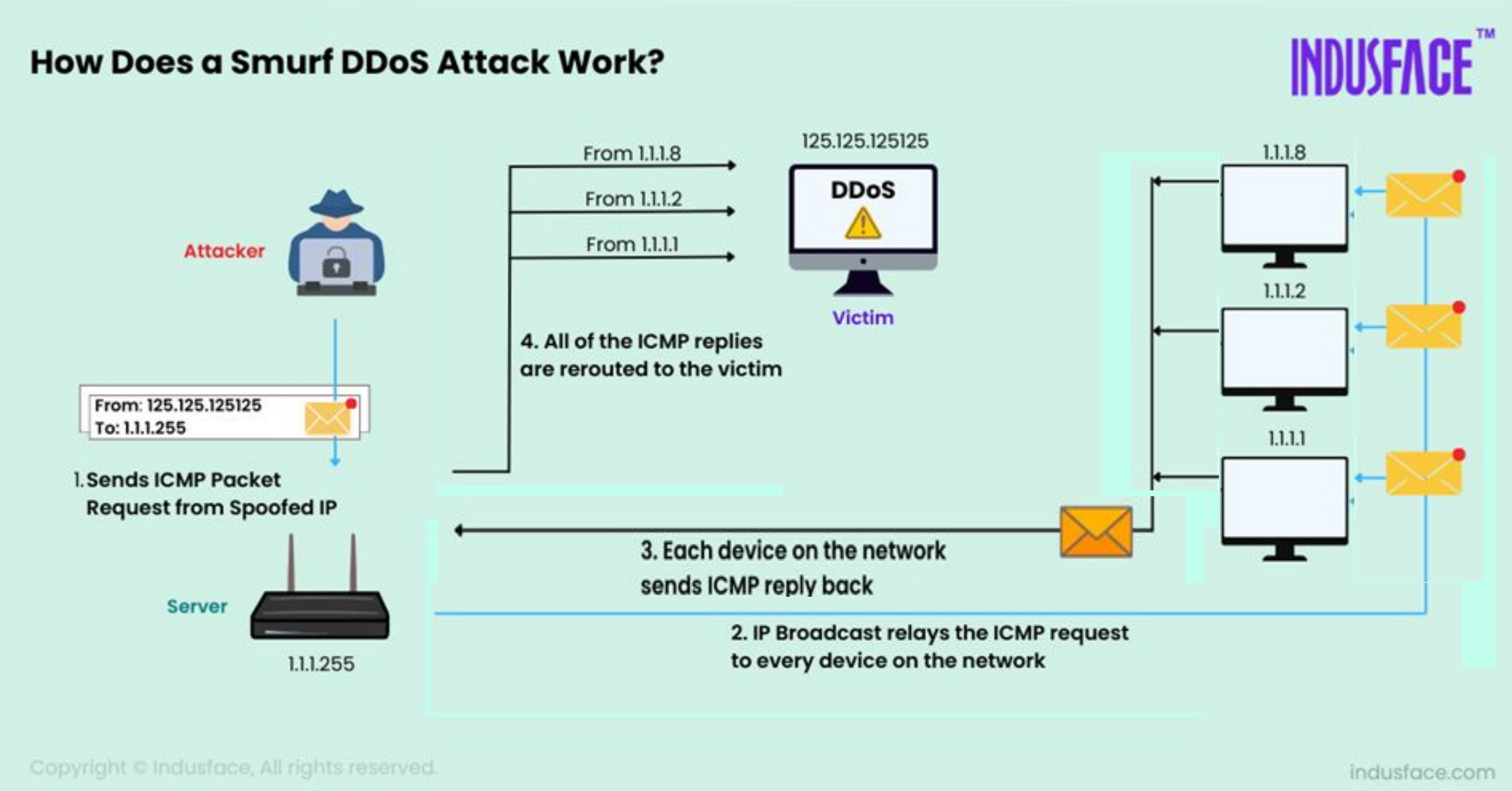
Ping Attacks

- **ICMP DDOS attack – Zombie Attack:**



Ping Attacks

- **ICMP DDOS attack – Packet magnification (or ICMP Smurf):**



How Does a Smurf DDoS Attack Work?

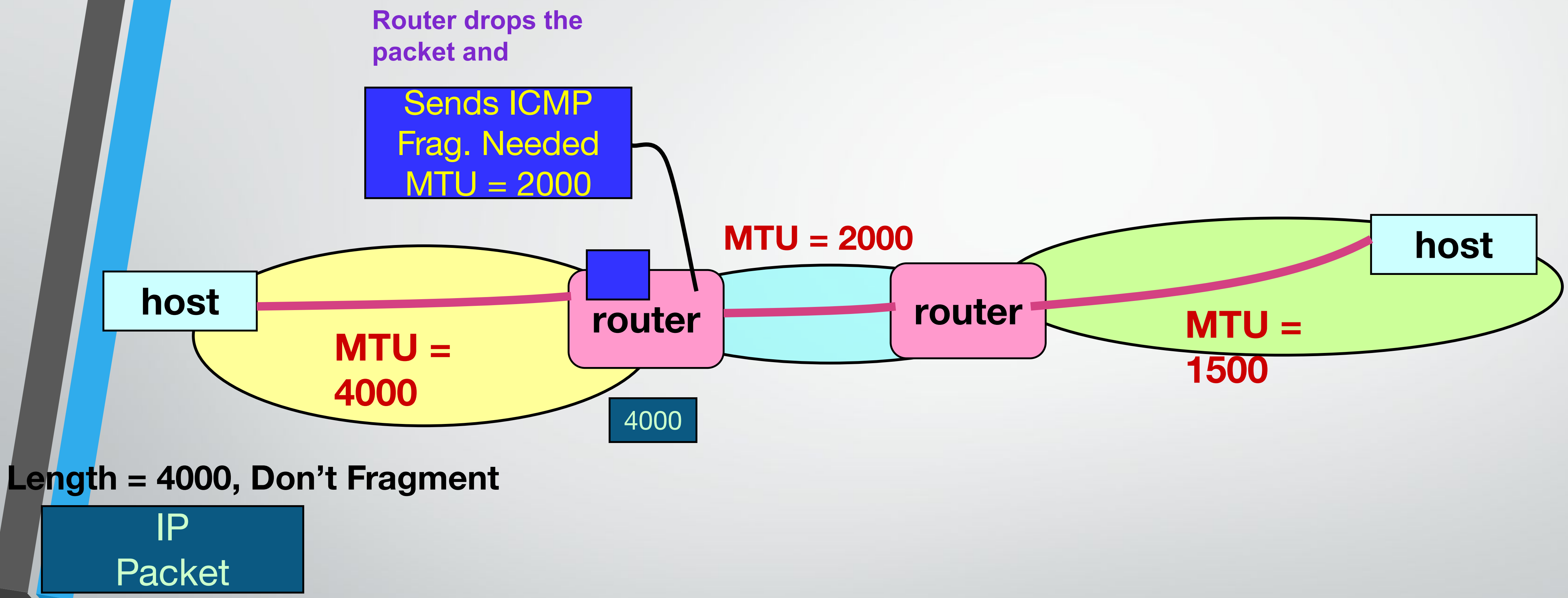
1. Sends ICMP Packet Request from Spoofed IP

2. IP Broadcast relays the ICMP request to every device on the network

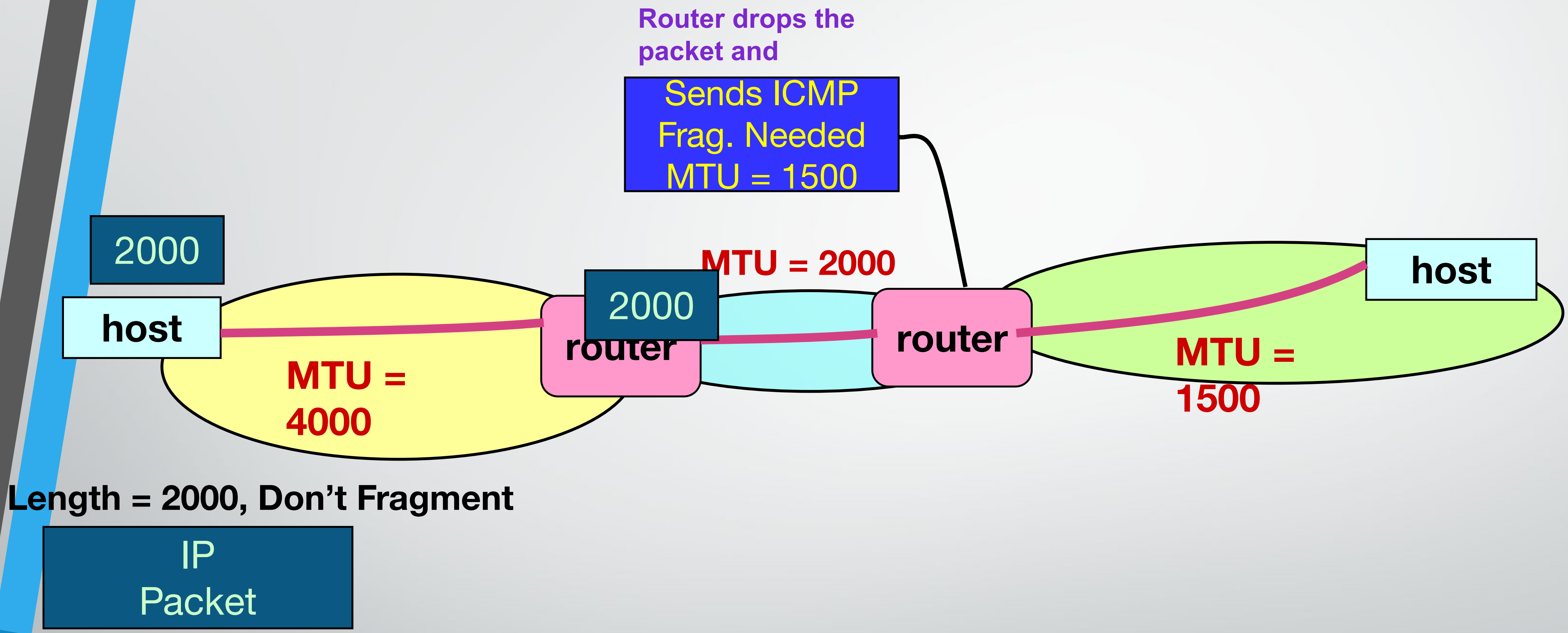
3. Each device on the network sends ICMP reply back

4. All of the ICMP replies are rerouted to the victim

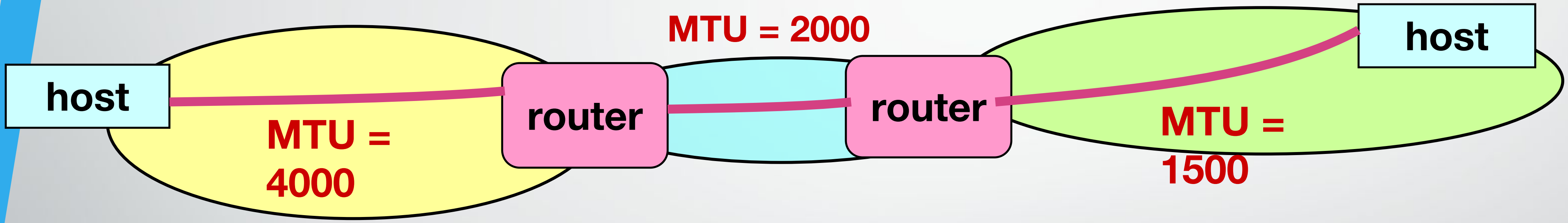
IP MTU Discovery with ICMP


Router drops the packet and
Sends ICMP Frag. Needed MTU = 2000
MTU = 2000
host
router
router
host
MTU = 4000
4000
MTU = 1500
Length = 4000, Don't Fragment
IP Packet

IP MTU Discovery with ICMP


Router drops the packet and
Sends ICMP Frag. Needed MTU = 1500
2000
host
MTU = 4000
2000
router
MTU = 2000
router
host
MTU = 1500
Length = 2000, Don’t Fragment
IP Packet

IP MTU Discovery with ICMP



Length = 1500, Don’t Fragment

When successful, no reply at IP level

“No news is good news”

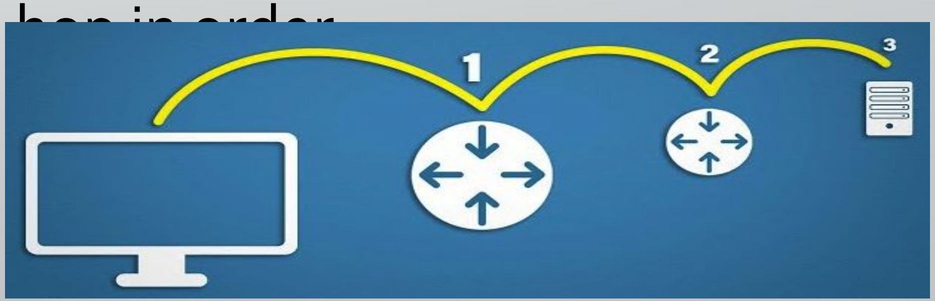
Higher level protocol might have some form of acknowledgement

Traceroute

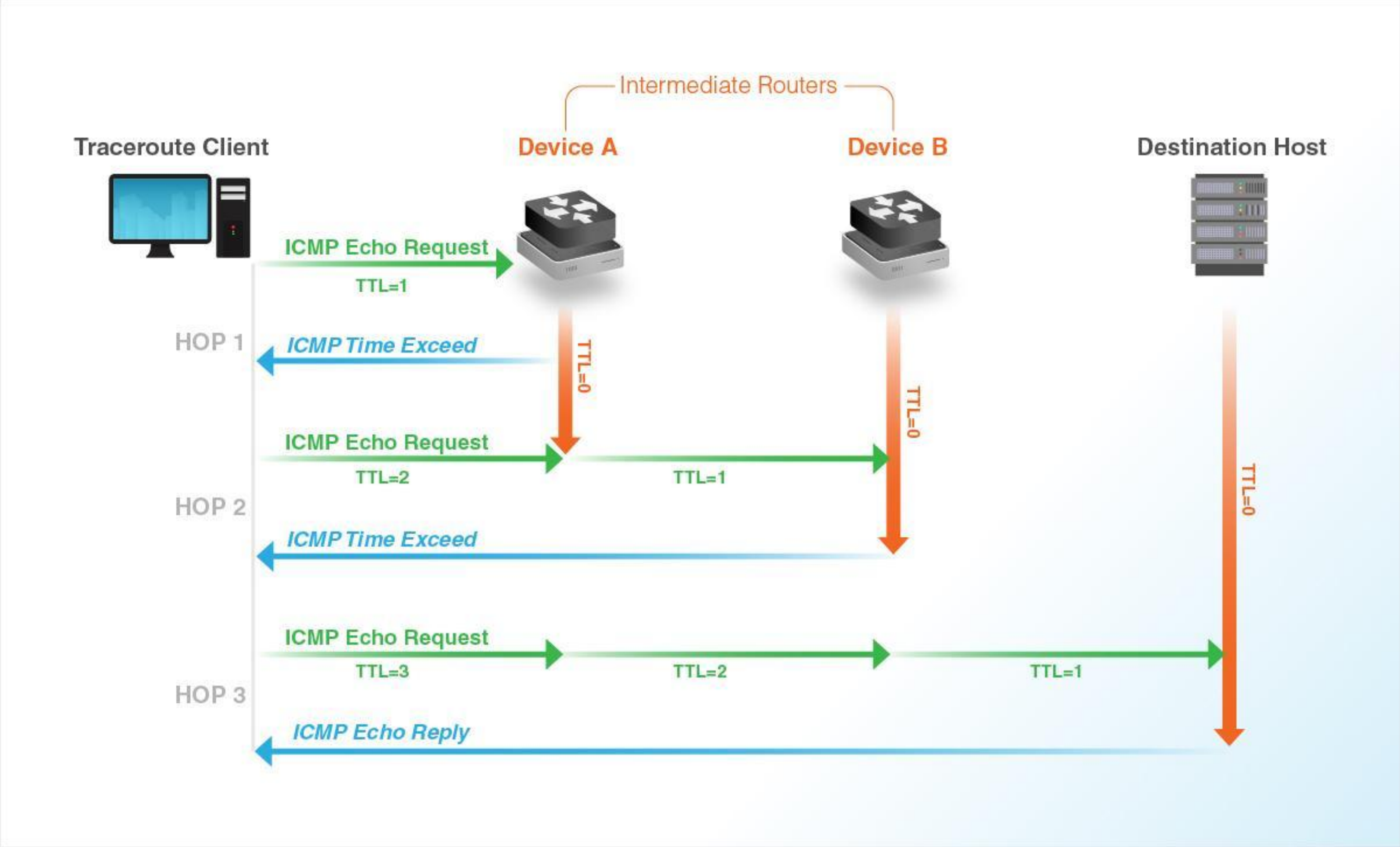
- Shows the path your packets take through different routers to reach a destination.

- **Purpose**:
  - Identifies the routers or hops data passes through to reach its destination mostly for troubleshooting

- **Mechanism**:
  - Sends ICMP packets with TTL = 1, 2, 3…
  - Each router where TTL becomes 0 sends ICMP Time Exceeded
  - Traceroute uses these replies to list each hop in order.



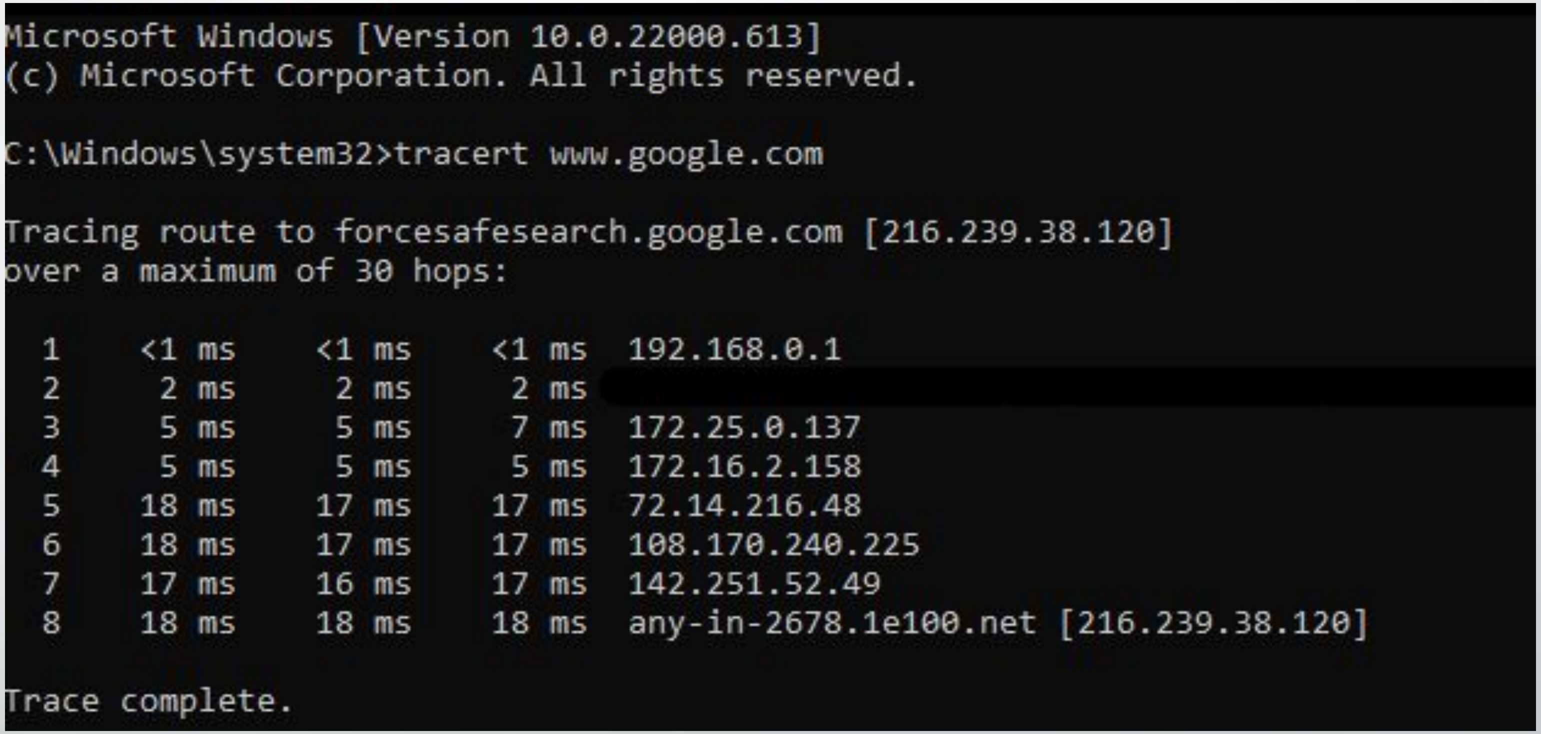
Traceroute



Intermediate Routers
Traceroute Client
Device A
Device B
Destination Host
ICMP Echo Request
TTL=1
HOP 1
ICMP Time Exceed
TTL=0
TTL=0
ICMP Echo Request
TTL=2
TTL=1
HOP 2
TTL=0
ICMP Time Exceed
ICMP Echo Request
TTL=3
TTL=2
TTL=1
HOP 3
ICMP Echo Reply

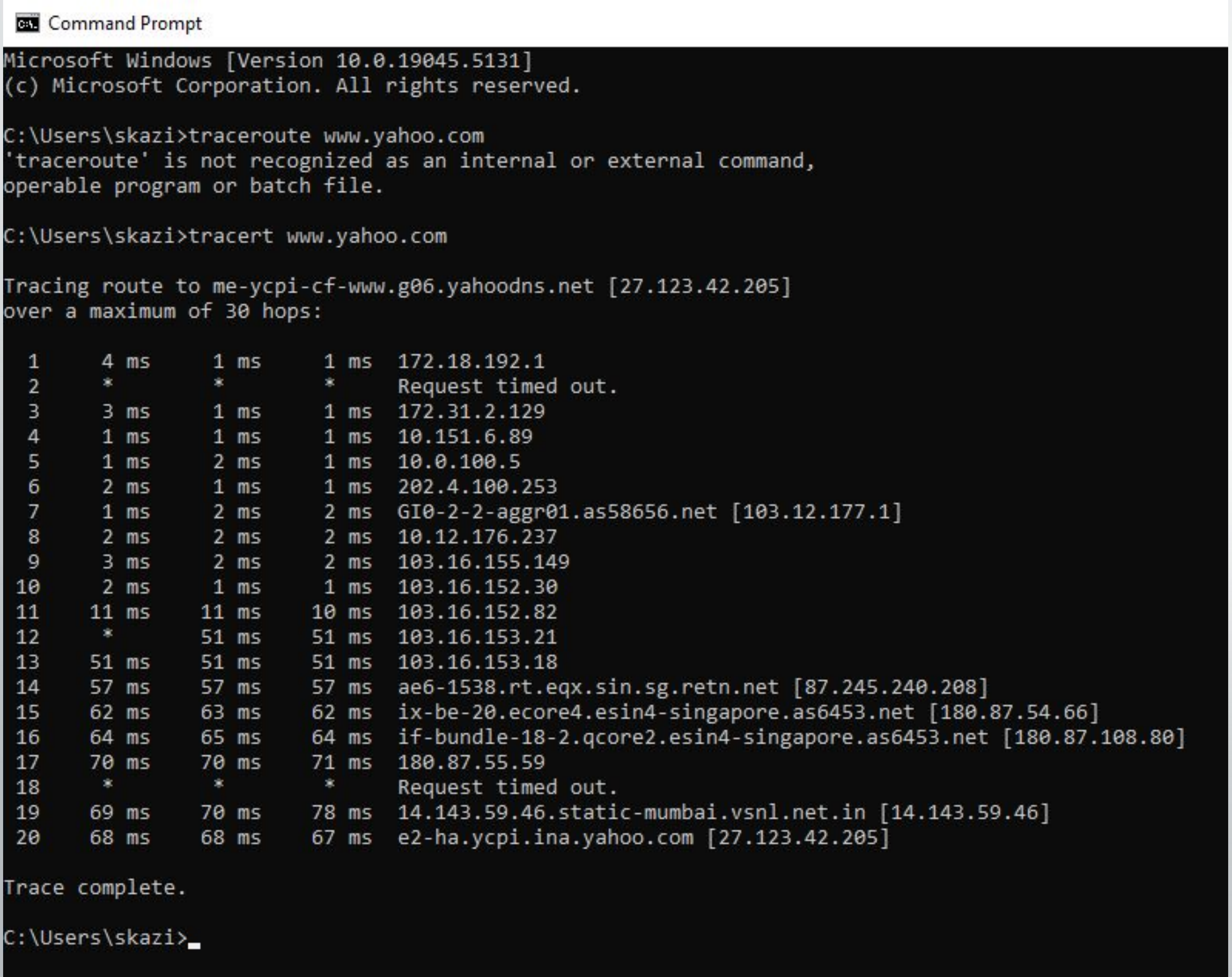
Traceroute

- **Results** The IP or hostname of every hop and the time each hop takes (latency)



```
Microsoft Windows [Version 10.0.22000.613]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>tracert www.google.com

Tracing route to forcesafesearch.google.com [216.239.38.120]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.0.1
  2     2 ms     2 ms     2 ms  [redacted]
  3     5 ms     5 ms     7 ms  172.25.0.137
  4     5 ms     5 ms     5 ms  172.16.2.158
  5    18 ms    17 ms    17 ms  72.14.216.48
  6    18 ms    17 ms    17 ms  108.170.240.225
  7    17 ms    16 ms    17 ms  142.251.52.49
  8    18 ms    18 ms    18 ms  any-in-2678.1e100.net [216.239.38.120]

Trace complete.
```

- Commands:
  - Unix: **traceroute**
  - Cisco IOS: **traceroute (trace)**
  - DOS: **tracert**



```
Microsoft Windows [Version 10.0.19045.5131]
(c) Microsoft Corporation. All rights reserved.

C:\Users\skazi>traceroute www.yahoo.com
'traceroute' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\skazi>tracert www.yahoo.com

Tracing route to me-ycpi-cf-www.g06.yahoodns.net [27.123.42.205]
over a maximum of 30 hops:

  1     4 ms     1 ms     1 ms  172.18.192.1
  2     *        *        *     Request timed out.
  3     3 ms     1 ms     1 ms  172.31.2.129
  4     1 ms     1 ms     1 ms  10.151.6.89
  5     1 ms     2 ms     1 ms  10.0.100.5
  6     2 ms     1 ms     1 ms  202.4.100.253
  7     1 ms     2 ms     2 ms  GI0-2-2-aggr01.as58656.net [103.12.177.1]
  8     2 ms     2 ms     2 ms  10.12.176.237
  9     3 ms     2 ms     2 ms  103.16.155.149
 10     2 ms     1 ms     1 ms  103.16.152.30
 11    11 ms    11 ms    10 ms  103.16.152.82
 12     *       51 ms    51 ms  103.16.153.21
 13    51 ms    51 ms    51 ms  103.16.153.18
 14    57 ms    57 ms    57 ms  ae6-1538.rt.eqx.sin.sg.retn.net [87.245.240.208]
 15    62 ms    63 ms    62 ms  ix-be-20.ecore4.esin4-singapore.as6453.net [180.87.54.66]
 16    64 ms    65 ms    64 ms  if-bundle-18-2.qcore2.esin4-singapore.as6453.net [180.87.108.80]
 17    70 ms    70 ms    71 ms  180.87.55.59
 18     *        *        *     Request timed out.
 19    69 ms    70 ms    78 ms  14.143.59.46.static-mumbai.vsnl.net.in [14.143.59.46]
 20    68 ms    68 ms    67 ms  e2-ha.ycpi.ina.yahoo.com [27.123.42.205]

Trace complete.

C:\Users\skazi>_
```

Using Tracert

Hop 1: Our local router or gateway (private IP address).
Hops 2–5: Internal routing within Bracu ISP's private network (non-public IPs).
Hop 6: First public IP, ISP's gateway to the internet.
Hops 7–9: Routing through regional and backbone ISPs.
Hops 10–13: Routing through Singapore (a major internet hub).
Hops 14–19: Routing through Indian networks, ending in Mumbai.
Hop 20: Final destination—Yahoo's server, located in India, near Mumbai.

Traceroute: Another example

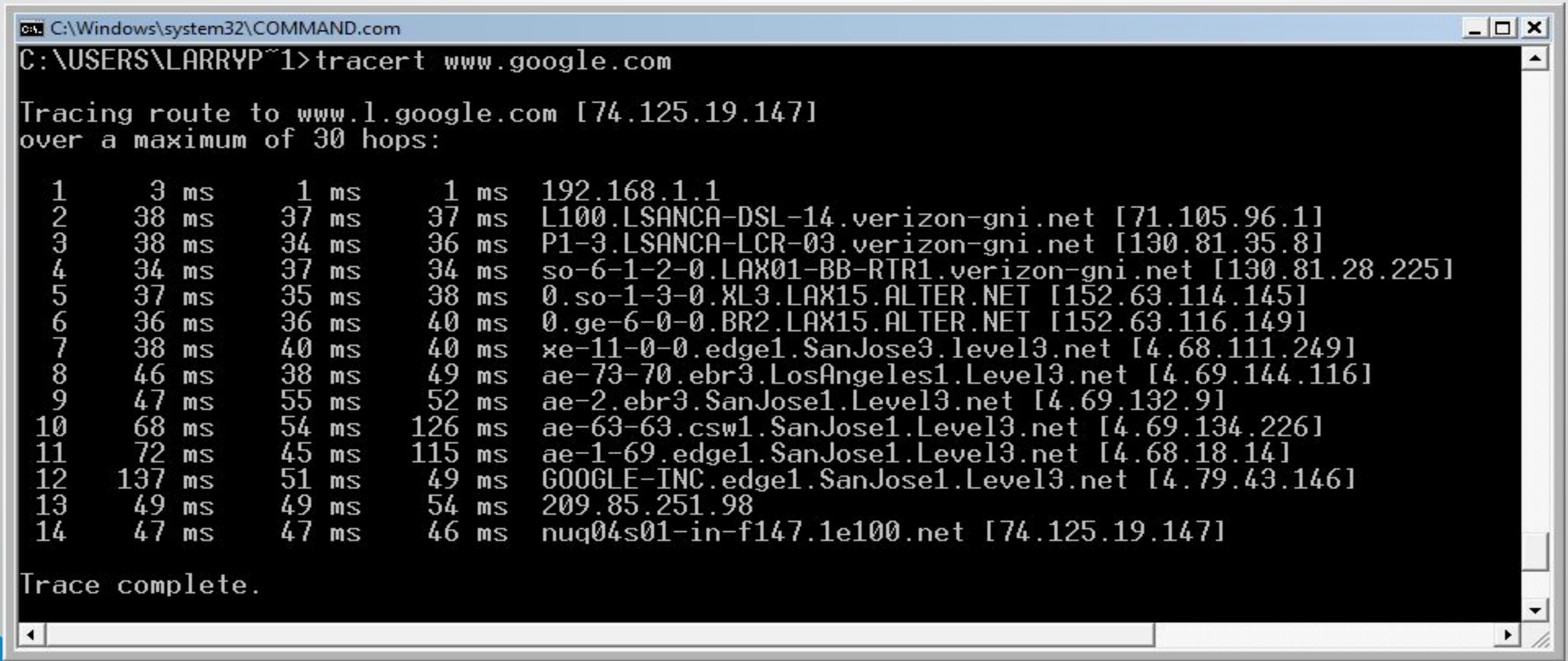
Hop 1: User LAN router

Hops 2-4: Verizon network (a backbone ISP)

Hops 5-6: Alternet (a backbone ISP)

Hops 7-11: Level 3 (a backbone ISP)

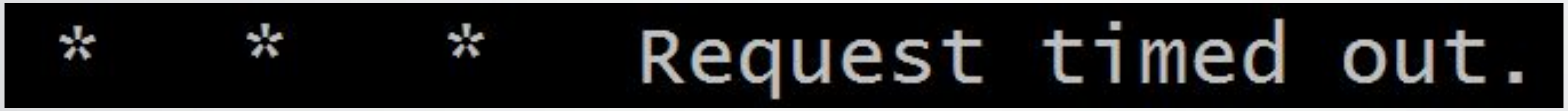
Hops 12-14: the Google LAN



```
C:\USERS\LARRYP~1>tracert www.google.com

Tracing route to www.l.google.com [74.125.19.147]
over a maximum of 30 hops:

  1     3 ms     1 ms     1 ms  192.168.1.1
  2    38 ms    37 ms    37 ms  L100.LSANCA-DSL-14.verizon-gni.net [71.105.96.1]
  3    38 ms    34 ms    36 ms  P1-3.LSANCA-LCR-03.verizon-gni.net [130.81.35.8]
  4    34 ms    37 ms    34 ms  so-6-1-2-0.LAX01-BB-RTR1.verizon-gni.net [130.81.28.225]
  5    37 ms    35 ms    38 ms  0.so-1-3-0.XL3.LAX15.ALTER.NET [152.63.114.145]
  6    36 ms    36 ms    40 ms  0.ge-6-0-0.BR2.LAX15.ALTER.NET [152.63.116.149]
  7    38 ms    40 ms    40 ms  xe-11-0-0.edge1.SanJose3.level3.net [4.68.111.249]
  8    46 ms    38 ms    49 ms  ae-73-70.ebr3.LosAngeles1.Level3.net [4.69.144.116]
  9    47 ms    55 ms    52 ms  ae-2.ebr3.SanJose1.Level3.net [4.69.132.9]
 10    68 ms    54 ms   126 ms  ae-63-63.csw1.SanJose1.Level3.net [4.69.134.226]
 11    72 ms    45 ms   115 ms  ae-1-69.edge1.SanJose1.Level3.net [4.68.18.14]
 12   137 ms    51 ms    49 ms  GOOGLE-INC.edge1.SanJose1.Level3.net [4.79.43.146]
 13    49 ms    49 ms    54 ms  209.85.251.98
 14    47 ms    47 ms    46 ms  nuq04s01-in-f147.1e100.net [74.125.19.147]

Trace complete.
```

Traceroute: Request Timed Out

This message indicates that the router security settings keep it from revealing its identity or the router and connection are slow.



```
  *        *        *     Request timed out.
```

The End