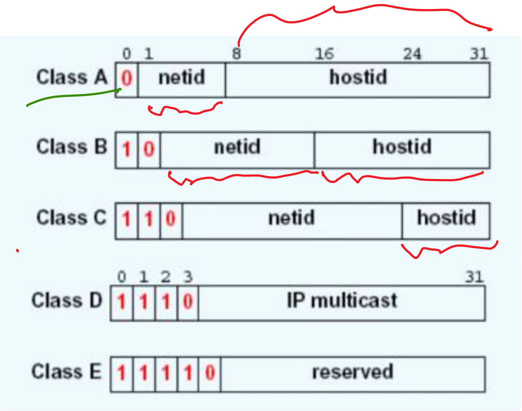
Internet Security

Network Layer
(IP and ICMP Protocols)

IP Address



0 1 8 16 24 31
Class A 0 netid hostid
Class B 1 0 netid hostid
Class C 1 1 0 netid hostid
0 1 2 3 31
Class D 1 1 1 0 IP multicast
Class E 1 1 1 1 0 reserved

reserved for private network

{ 10.0.0.0/8
192.168.0.0/16

2^7

2^{16}

128.230.0.0

netid hostid

149._

ESF

10.0.0.0

10.0.0.0/8

128.230.0.0/16

128.230.0.0/18

128.230.0.5

128.230.128.5

0000 0000

Responsibility of IP Layer

Routing. ✓ ✗

Resending Packet ✗ ✓ ✗

Maintaining the order ✗ ✓ ✗

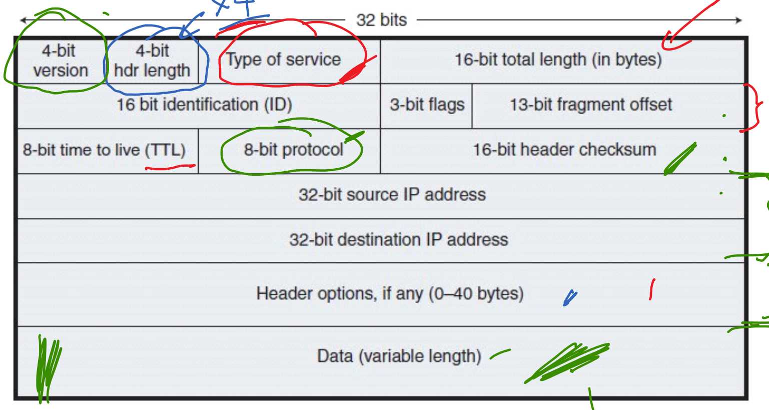
Detection Packet Loss ✗ ✓ ✗

preventing Packet Duplication ✗ ✗

best effort

TCP

IP Header and Protocol



32 bits

4-bit version	4-bit hdr length	Type of service	16-bit total length (in bytes)
16 bit identification (ID)		3-bit flags	13-bit fragment offset
8-bit time to live (TTL)	8-bit protocol	16-bit header checksum	
32-bit source IP address			
32-bit destination IP address			
Header options, if any (0–40 bytes)			
Data (variable length)			

x4

x)

$2^{16} - 1 = 65535$

20 bytes

optim

UDP
TCP
ICMP

45

TTL:
255
50

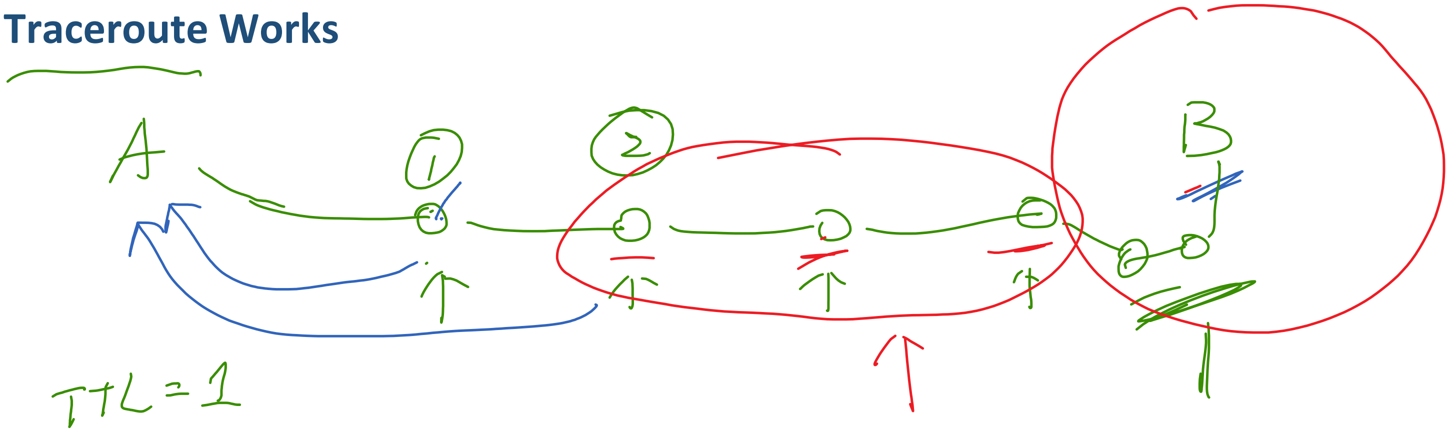
≥20

1111

15

+2

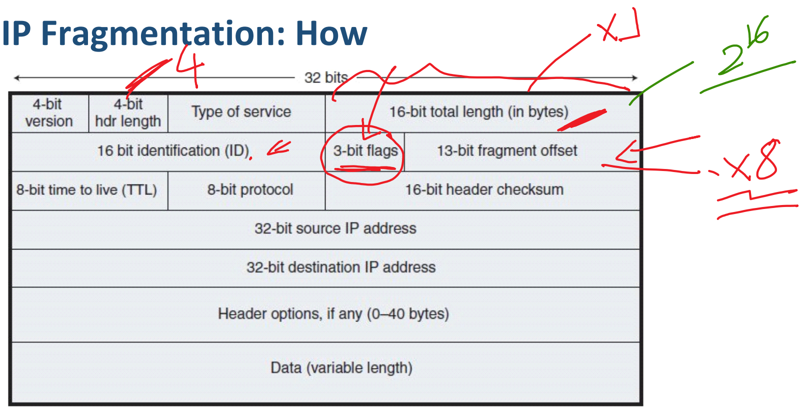
How Traceroute Works

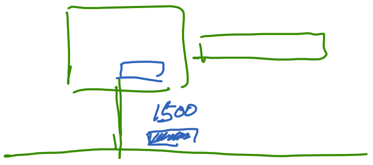


TTL = 1

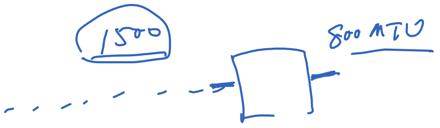
TTL = 2

IP Fragmentation: How

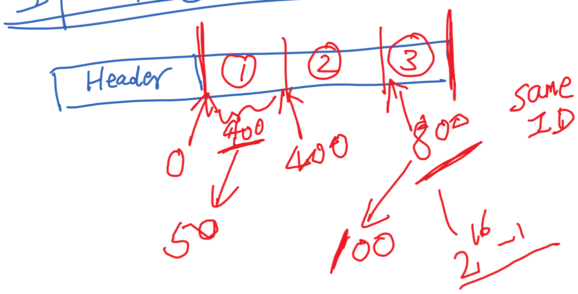


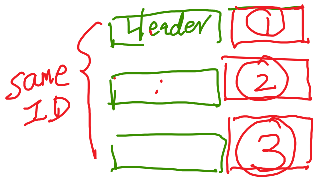


MTU
Maximum
Transmission
Unit



IP Fragmentation

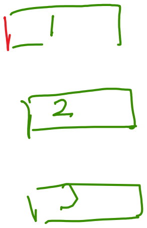


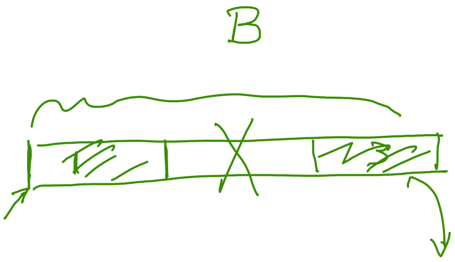


1 sequence (offset)

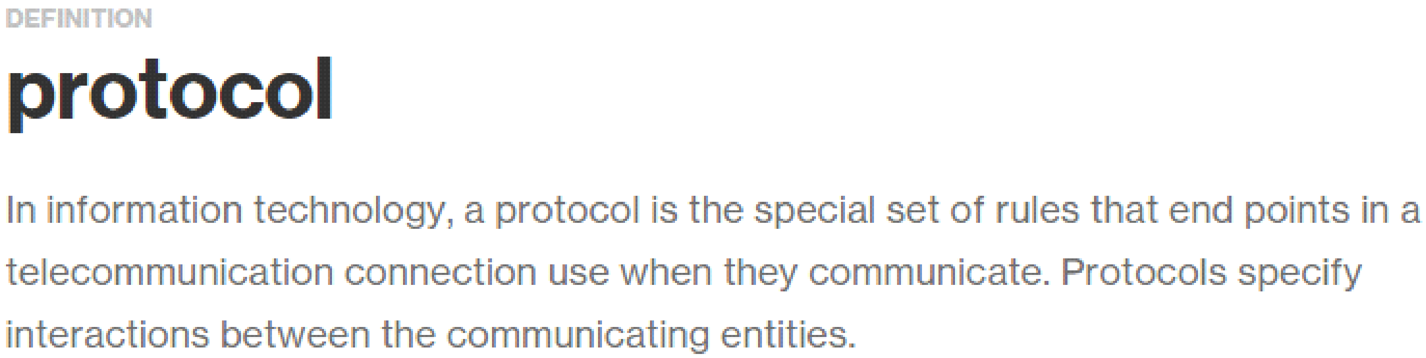
2 last one

3 part of the same IP





Attacks on IP Fragmentation



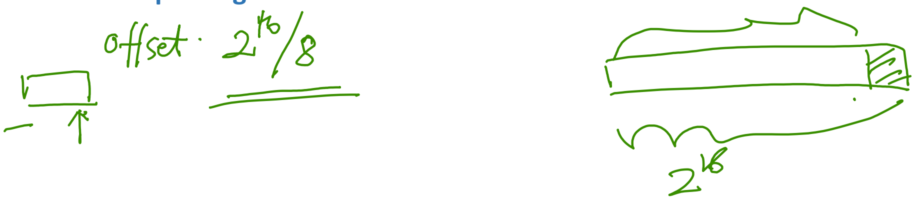
DEFINITION

protocol

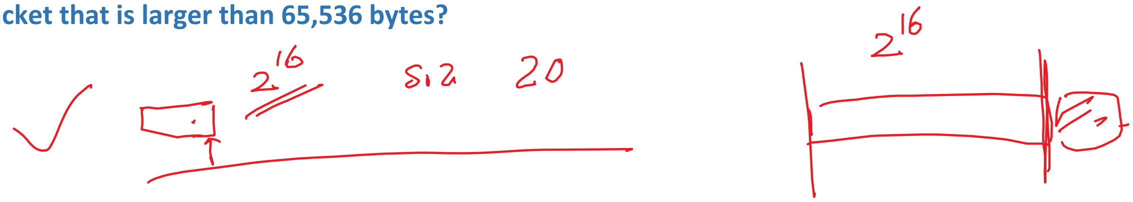
In information technology, a protocol is the special set of rules that end points in a telecommunication connection use when they communicate. Protocols specify interactions between the communicating entities.

Questions: Attacks Using Fragmentation

Q1: Can you use a small amount of bandwidth to tie up a target machine's significant amount of resources?

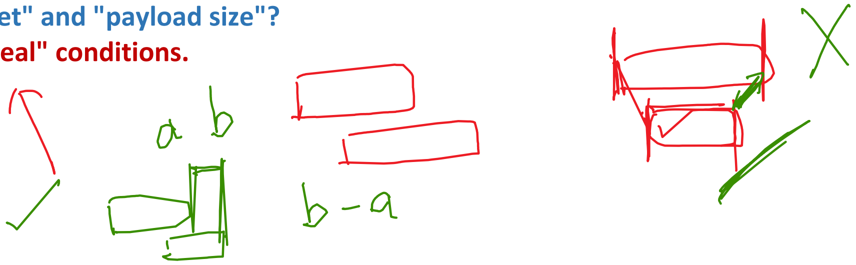


Q2: Can you create an IP packet that is larger than 65,536 bytes?



Q3: Can you create some abnormal conditions using "offset" and "payload size"?

Goal: Test whether a computer can handle these "unreal" conditions.

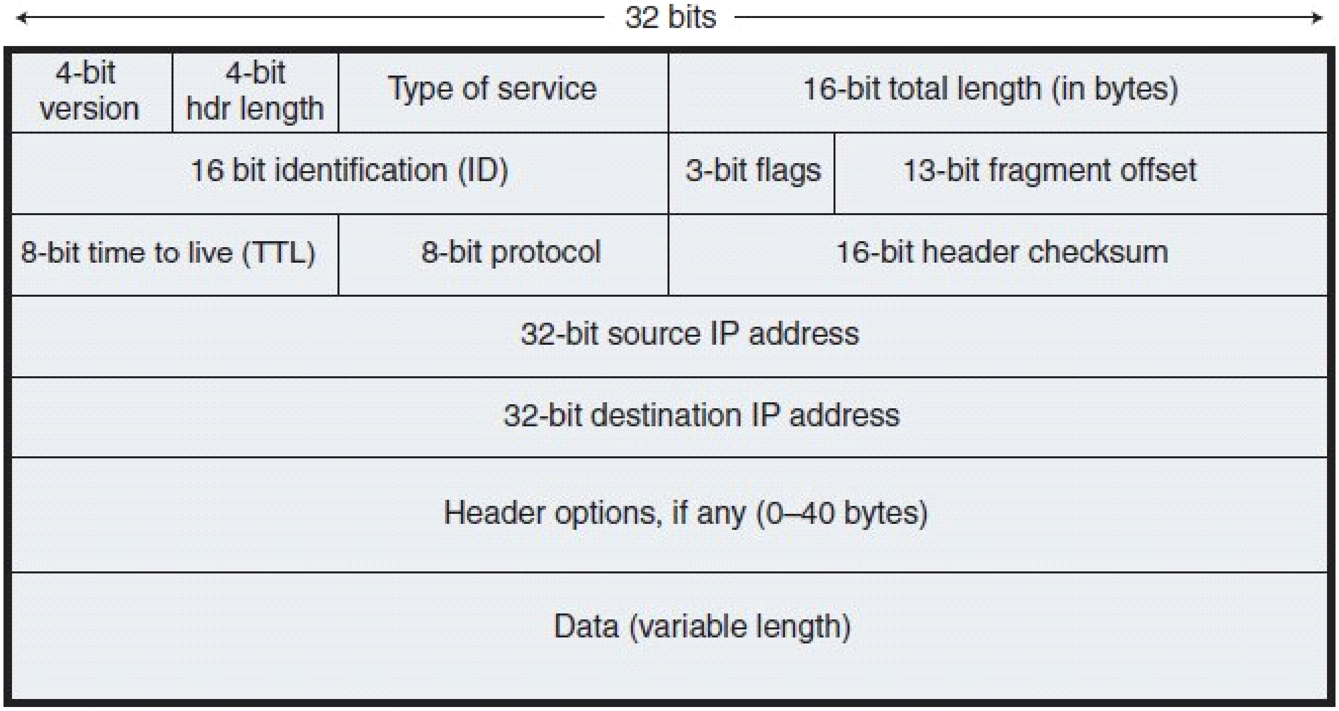


Attack 1: Tie Up Target's Resources

Can you use a small amount of bandwidth to tie up a target machine's significant amount of resources?

Attack 2: Create a Super-Large Packet

Can you create an IP packet that is larger than 65,536 bytes?



32 bits

4-bit version	4-bit hdr length	Type of service	16-bit total length (in bytes)	
16 bit identification (ID)			3-bit flags	13-bit fragment offset
8-bit time to live (TTL)		8-bit protocol	16-bit header checksum	
32-bit source IP address				
32-bit destination IP address				
Header options, if any (0–40 bytes)				
Data (variable length)				

Attack 3: Create Abnormal Situation

Can you create some abnormal conditions using "offset" and "payload size"?

Test whether a computer can handle these "unreal" conditions.

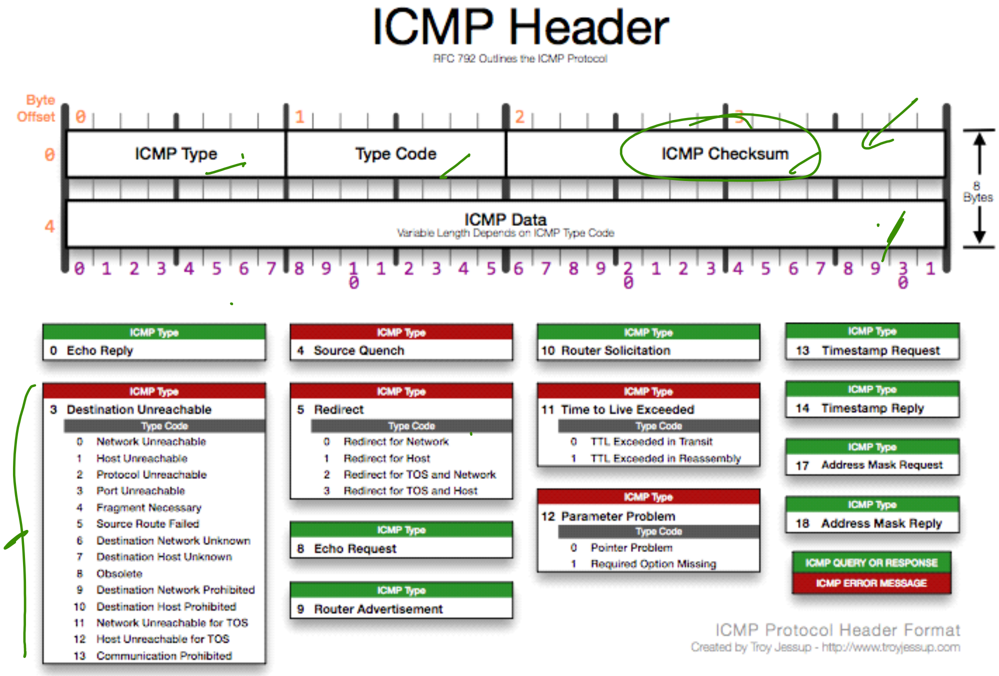
ICMP: Internet Control Message Protocol

control

error

: echo

ICMP Header



ICMP Type
0 Echo Reply

ICMP Type
3 Destination Unreachable

Type Code	
0	Network Unreachable
1	Host Unreachable
2	Protocol Unreachable
3	Port Unreachable
4	Fragment Necessary
5	Source Route Failed
6	Destination Network Unknown
7	Destination Host Unknown
8	Obsolete
9	Destination Network Prohibited
10	Destination Host Prohibited
11	Network Unreachable for TOS
12	Host Unreachable for TOS
13	Communication Prohibited

ICMP Type
4 Source Quench

ICMP Type
5 Redirect

Type Code	
0	Redirect for Network
1	Redirect for Host
2	Redirect for TOS and Network
3	Redirect for TOS and Host

ICMP Type
8 Echo Request

ICMP Type
9 Router Advertisement

ICMP Type
10 Router Solicitation

ICMP Type
11 Time to Live Exceeded

Type Code	
0	TTL Exceeded in Transit
1	TTL Exceeded in Reassembly

ICMP Type
12 Parameter Problem

Type Code	
0	Pointer Problem
1	Required Option Missing

ICMP Type
13 Timestamp Request

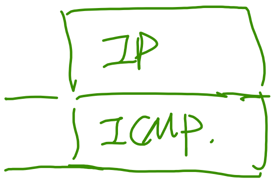
ICMP Type
14 Timestamp Reply

ICMP Type
17 Address Mask Request

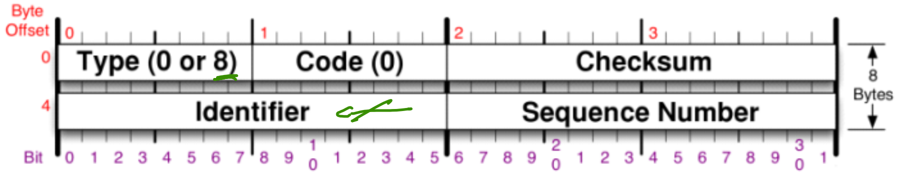
ICMP Type
18 Address Mask Reply

🟩 ICMP QUERY OR RESPONSE
🟥 ICMP ERROR MESSAGE

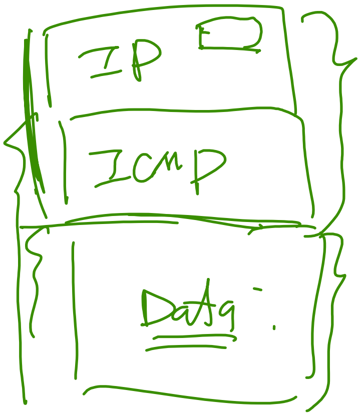
ICMP Protocol Header Format
Created by Troy Jessup - http://www.troyjessup.com



ICMP Echo Request/Reply

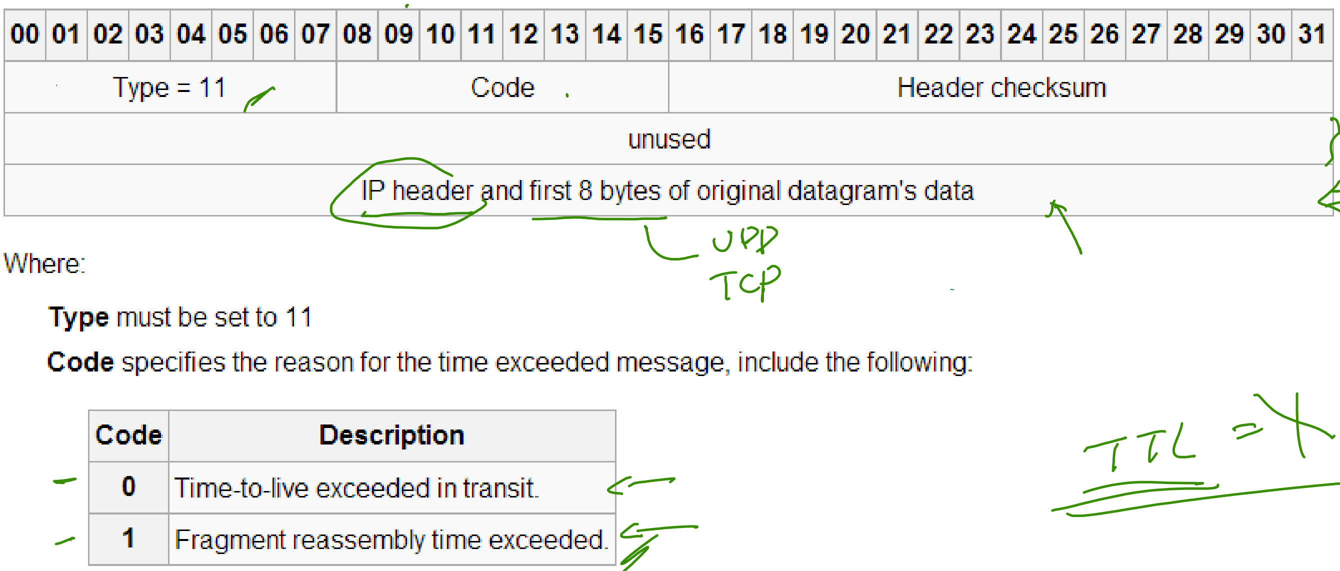


Data: Echo reply (type 0) must return any data sent in echo request



ping

ICMP Time Exceeded



<table>
<tr><th>00</th><th>01</th><th>02</th><th>03</th><th>04</th><th>05</th><th>06</th><th>07</th><th>08</th><th>09</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td colspan="8">Type = 11</td><td colspan="8">Code</td><td colspan="16">Header checksum</td></tr>
<tr><td colspan="32">unused</td></tr>
<tr><td colspan="32">IP header and first 8 bytes of original datagram's data</td></tr>
</table>

UDP
TCP

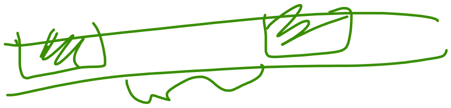
Where:

Type must be set to 11

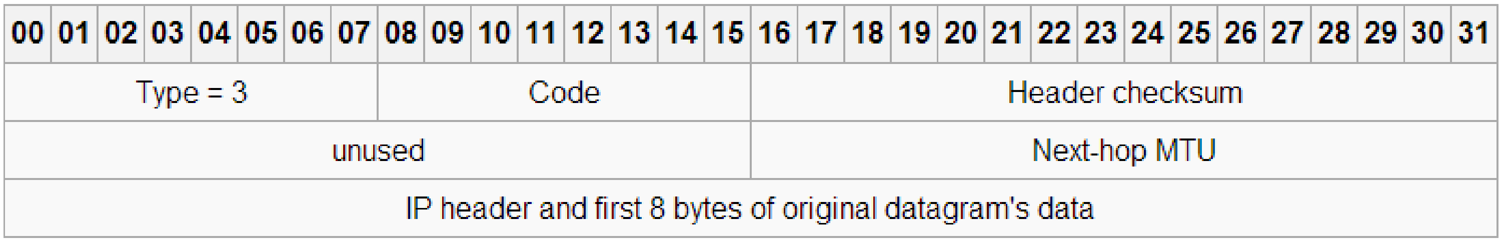
Code specifies the reason for the time exceeded message, include the following:

Code	Description
0	Time-to-live exceeded in transit.
1	Fragment reassembly time exceeded.

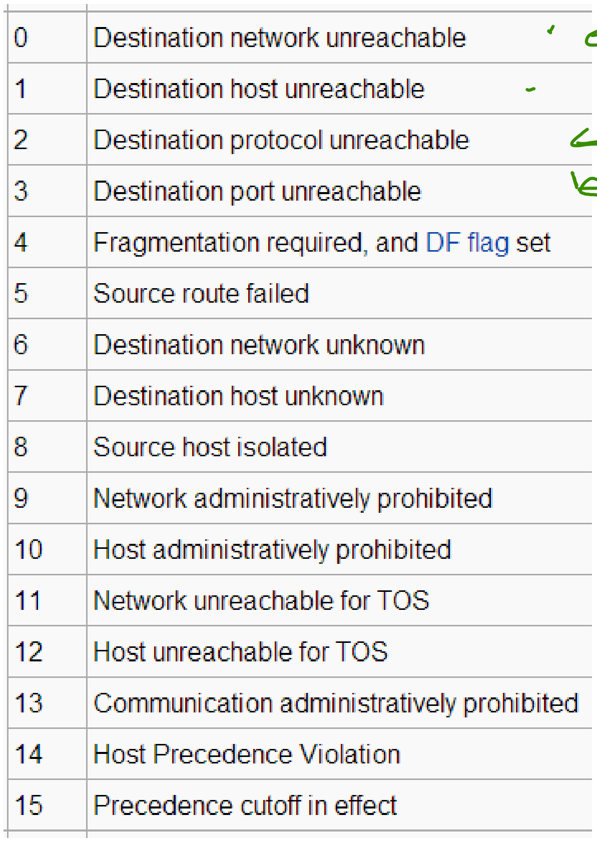
TTL => 2



ICMP Destination Unreachable

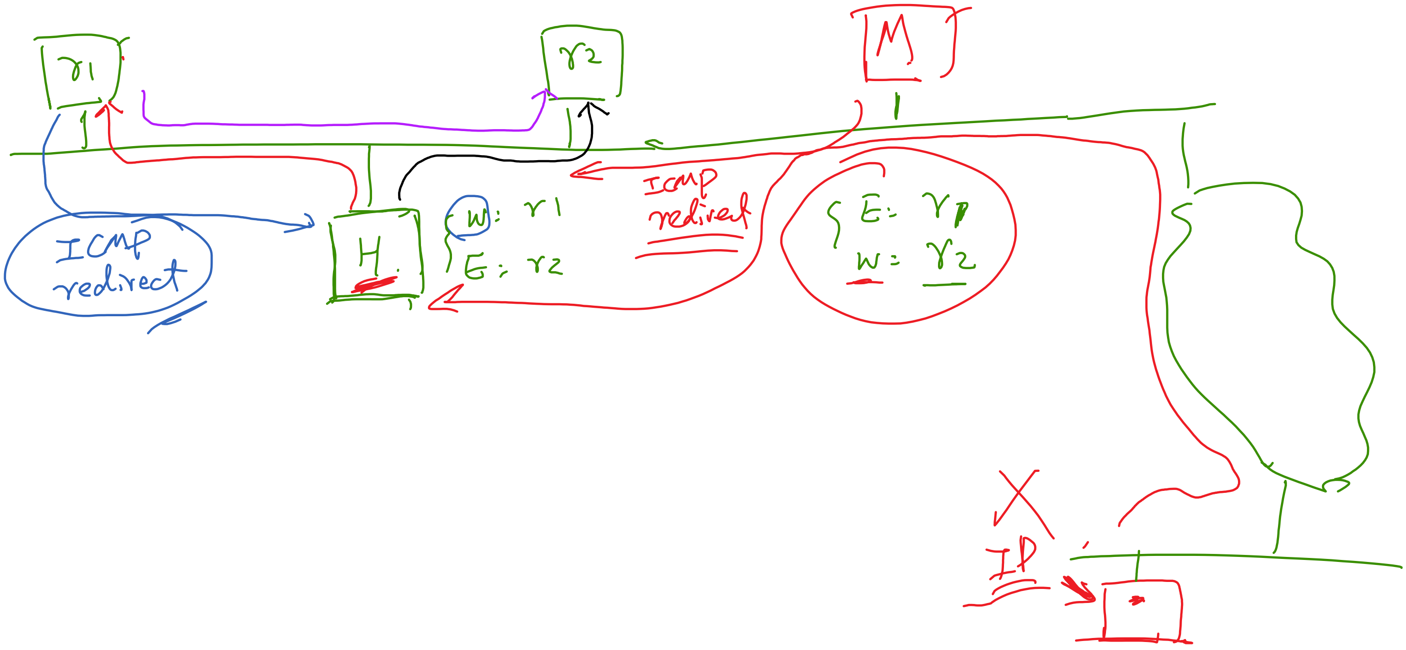


00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type = 3								Code								Header checksum															
unused																Next-hop MTU															
IP header and first 8 bytes of original datagram's data																															

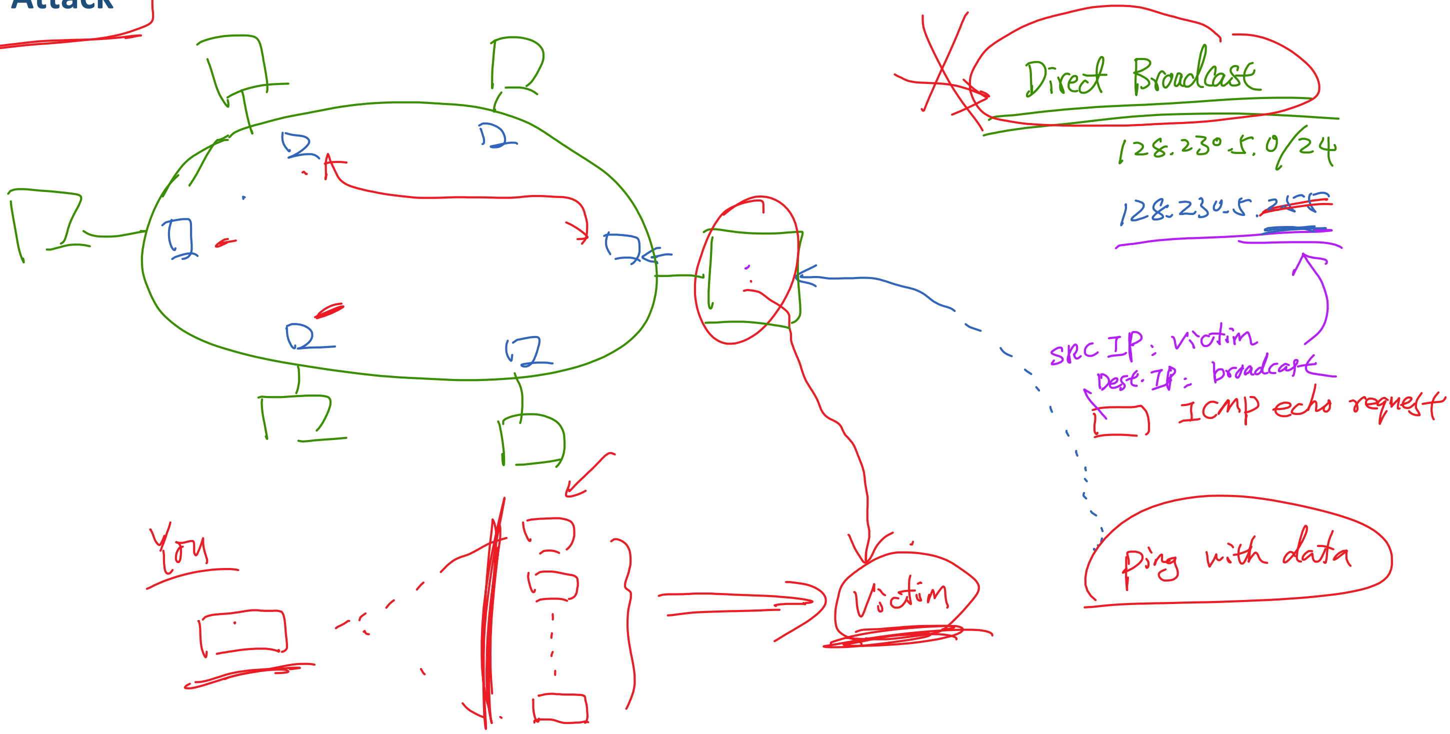


0	Destination network unreachable
1	Destination host unreachable
2	Destination protocol unreachable
3	Destination port unreachable
4	Fragmentation required, and DF flag set
5	Source route failed
6	Destination network unknown
7	Destination host unknown
8	Source host isolated
9	Network administratively prohibited
10	Host administratively prohibited
11	Network unreachable for TOS
12	Host unreachable for TOS
13	Communication administratively prohibited
14	Host Precedence Violation
15	Precedence cutoff in effect

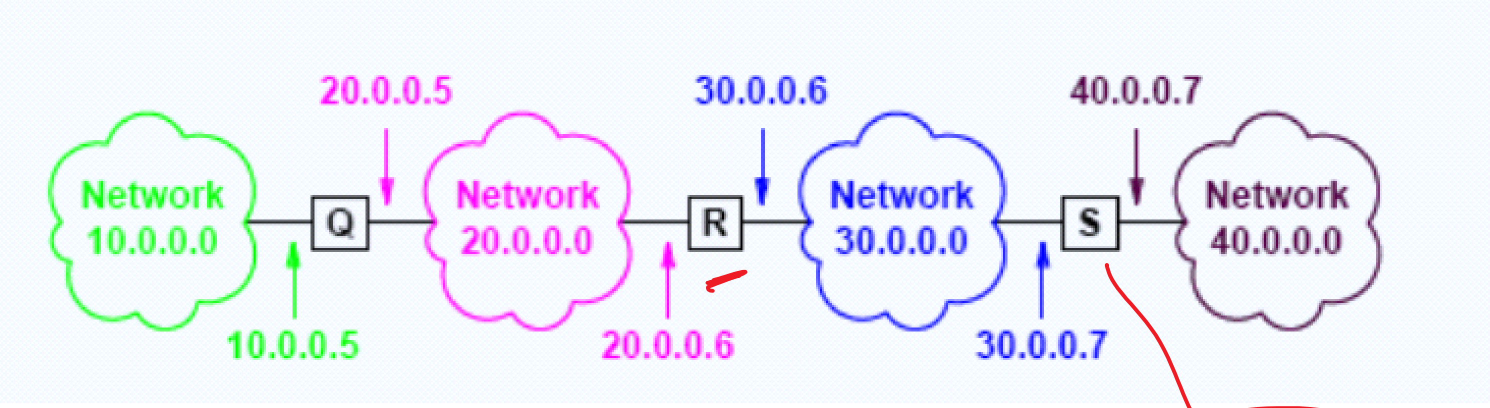
ICMP Redirect and Attacks

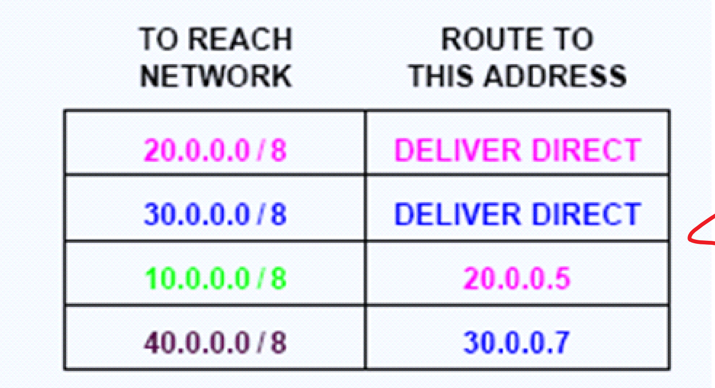


Smurf Attack



Routing

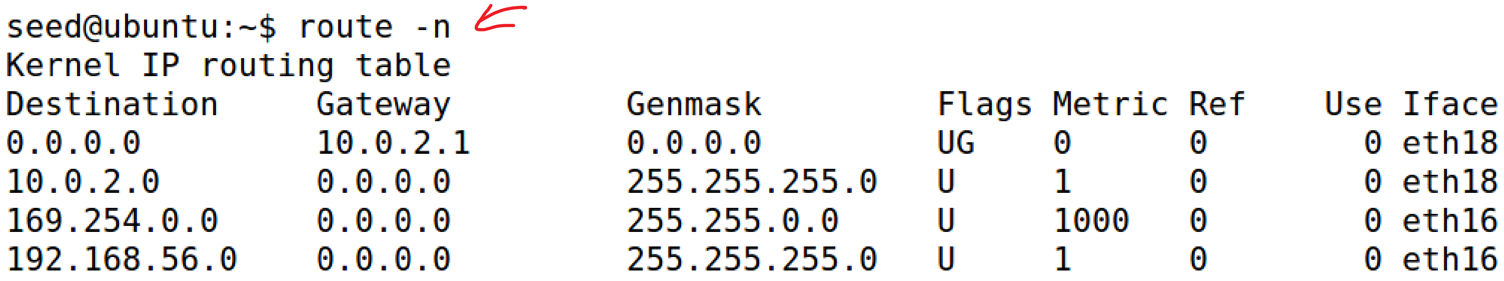




TO REACH NETWORK	ROUTE TO THIS ADDRESS
20.0.0.0 / 8	DELIVER DIRECT
30.0.0.0 / 8	DELIVER DIRECT
10.0.0.0 / 8	20.0.0.5
40.0.0.0 / 8	30.0.0.7

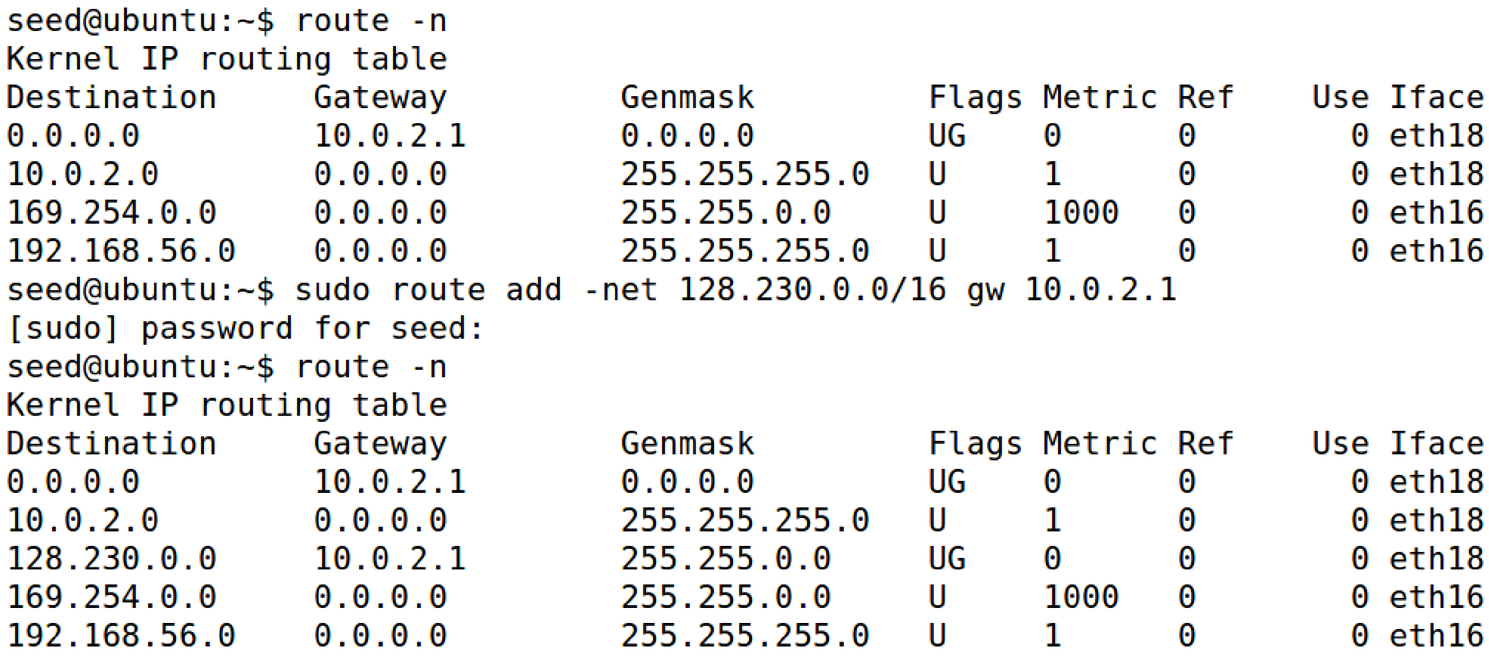
The routing table for router R

Routing Table on a Host



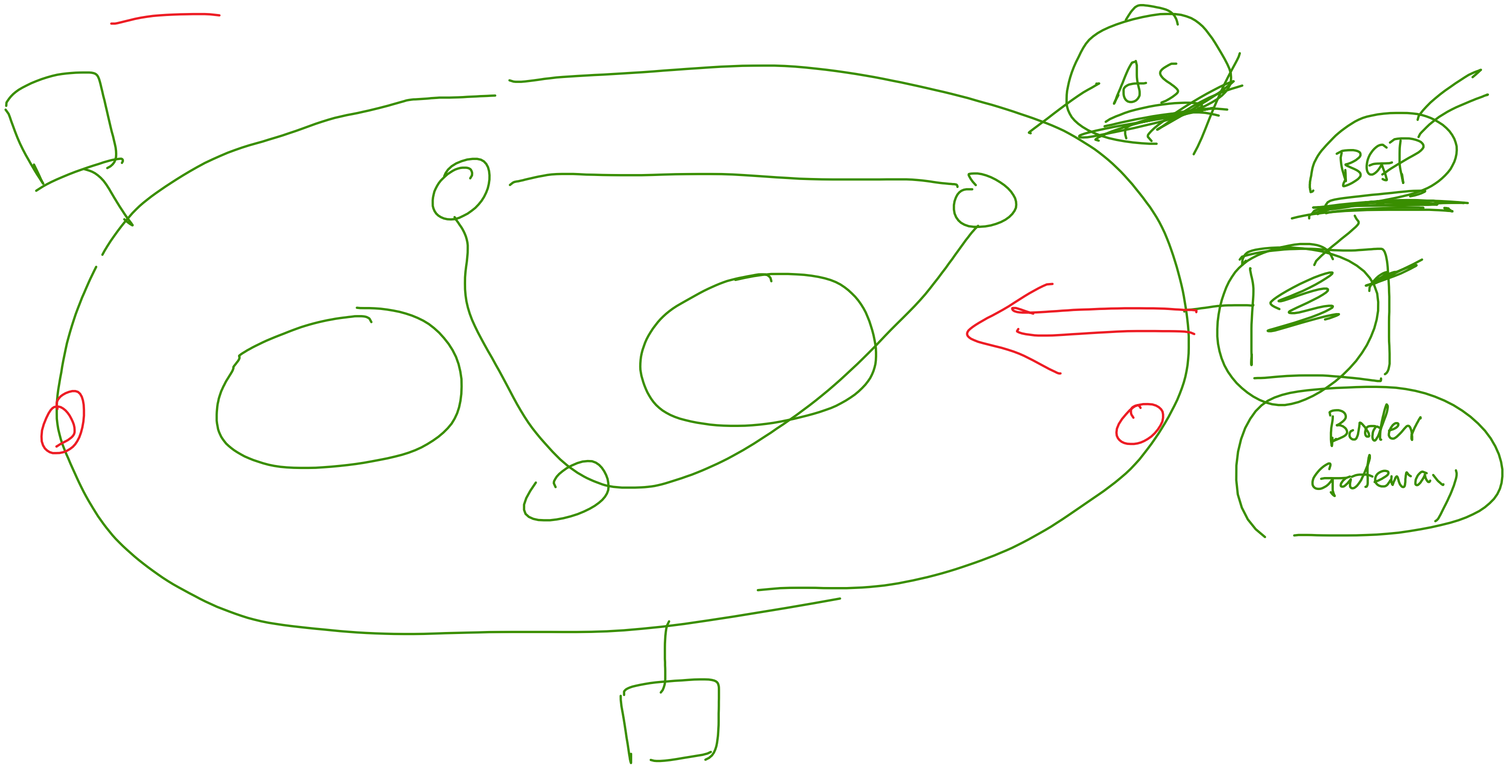
```
seed@ubuntu:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.2.1        0.0.0.0         UG    0      0        0 eth18
10.0.2.0        0.0.0.0         255.255.255.0   U     1      0        0 eth18
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth16
192.168.56.0    0.0.0.0         255.255.255.0   U     1      0        0 eth16
```

Change Routing Table



```
seed@ubuntu:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.2.1        0.0.0.0         UG    0      0        0 eth18
10.0.2.0        0.0.0.0         255.255.255.0   U     1      0        0 eth18
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth16
192.168.56.0    0.0.0.0         255.255.255.0   U     1      0        0 eth16
seed@ubuntu:~$ sudo route add -net 128.230.0.0/16 gw 10.0.2.1
[sudo] password for seed:
seed@ubuntu:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.2.1        0.0.0.0         UG    0      0        0 eth18
10.0.2.0        0.0.0.0         255.255.255.0   U     1      0        0 eth18
128.230.0.0     10.0.2.1        255.255.0.0     UG    0      0        0 eth18
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth16
192.168.56.0    0.0.0.0         255.255.255.0   U     1      0        0 eth16
```

How Do Routers and Host Get Routing Information?


AS
BGP
Border Gateway