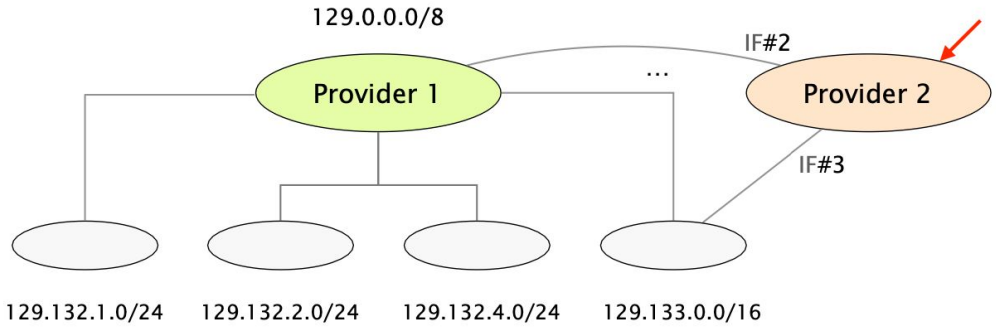
Does CIDR Make This Easier or Harder?

Multiple Matches

Provider 2's Forwarding table

IP prefix	Output
129.0.0.0/8	IF#2
129.132.1.0/24	IF#2
129.132.2.0/24	IF#2
129.133.0.0/16	IF#3

Packet arrives for 129.133.0.1



What Should We Do?

What Should We Do?

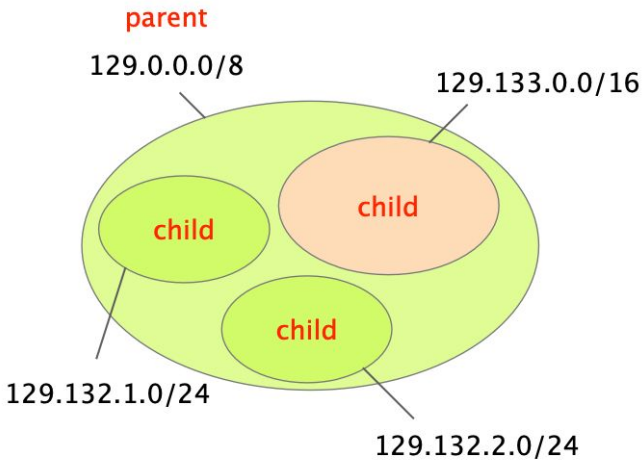
To resolve ambiguity, forwarding is done along the most specific prefix (i.e., the longer one)

Route Aggregation

Child prefixes can be removed from the table if they share the same output interface as the parent

Routing Table

IP prefix	Output Interface
…	
129.0.0.0/8	IF#2
129.132.1.0/24	IF#2
129.132.2.0/24	IF#2
129.133.0.0/16	IF#3
…	

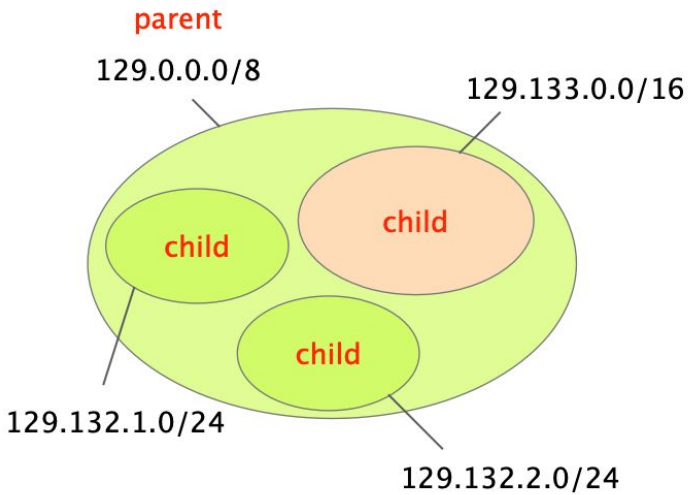


Route Aggregation

Child prefixes can be removed from the table if they share the same output interface as the parent

Routing Table

IP prefix	Output Interface
…	
129.0.0.0/8	IF#2
~~129.132.1.0/24~~	~~IF#2~~
~~129.132.2.0/24~~	~~IF#2~~
129.133.0.0/16	IF#3
…	

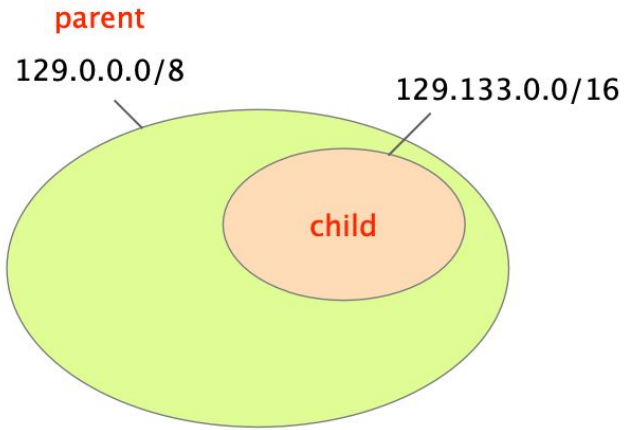


Route Aggregation

Child prefixes can be removed from the table if they share the same output interface as the parent

Routing Table

IP prefix	Output Interface
…	
129.0.0.0/8	IF#2
129.133.0.0/16	IF#3
…	



Exactly the same forwarding as before

Route Tables

Consider the following routing table:

Network Destination	Next Hop
142.150.64.0/20	A
142.150.71.128/28	B
142.150.71.128/30	D
142.150.0.0/16	C

- Assume that a router receives an IP datagram with destination 142.150.71.132. What is the next hop? Why?

- Add a routing table entry to the table above which enforces that all IP datagrams with destination 142.150.71.132 have “A” as Next Hop. For all other IP destination addresses, the Next Hop should not change.

- Add a routing table entry to the table above which enforces that all IP datagrams whose destination address does not match any of the entries in the table, are forwarded to next hop “C”.

Route Tables

Consider the following routing table:

Network Destination	Next Hop
142.150.64.0/20	A
142.150.71.128/28	B
142.150.71.128/30	D
142.150.0.0/16	C

- Assume that a router receives an IP datagram with destination 142.150.71.132. What is the next hop? Why?
  The second entry has the longest matching prefix, so the next hop is B.

- Add a routing table entry to the table above which enforces that all IP datagrams with destination 142.150.71.132 have “A” as Next Hop. For all other IP destination addresses, the Next Hop should not change.
  Network Destination Next Hop
  142.150.71.132/32 A

- Add a routing table entry to the table above which enforces that all IP datagrams whose destination address does not match any of the entries in the table, are forwarded to next hop “C”.
  Network Destination Next Hop
  0.0.0.0/0 C
  Known as a “default route” - when you don’t have something specified, go there

Route Aggregation

Aggregate the following set of four /24 IP network addresses to the highest degree possible.

212.56.132.0/24

212.56.133.0/24

212.56.134.0/24

212.56.135.0/24

Route Aggregation

Aggregate the following set of four /24 IP network addresses to the highest degree possible.

212.56.132.0/24
212.56.133.0/24
212.56.134.0/24
212.56.135.0/24

One way to do this: List each address in binary format and determine the common prefix for all of the addresses:

212.56.132.0/24 11010100.00111000.10000100.00000000
212.56.133.0/24 11010100.00111000.10000101.00000000
212.56.134.0/24 11010100.00111000.10000110.00000000
212.56.135.0/24 11010100.00111000.10000111.00000000

Common Prefix: 11010100.00111000.10000100.00000000

The CIDR aggregation is: 212.56.132.0/22

Route Aggregation

Aggregate the following set of four /24 IP network addresses to the highest degree possible.

212.56.146.0/24
212.56.147.0/24
212.56.148.0/24
212.56.149.0/24

Route Aggregation

Aggregate the following set of four /24 IP network addresses to the highest degree possible.

212.56.146.0/24
212.56.147.0/24
212.56.148.0/24
212.56.149.0/24

One way to do this: List each address in binary format and determine the common prefix for all of the addresses:

212.56.146.0/24 11010100.00111000.10010**010**.00000000
212.56.147.0/24 11010100.00111000.10010**011**.00000000
212.56.148.0/24 11010100.00111000.10010**100**.00000000
212.56.148.0/24 11010100.00111000.10010**101**.00000000

This set of four /24s cannot be summarized as a single block:
212.56.146.0/23
212.56.148.0/23

Note that if two /23s are to be aggregated into a /22, then both /23s must fall within a single /22 block. Since each of the two /23s is a member of a different /22 block, they cannot be aggregated into a single /22 (even though they are consecutive).

IP Subnetting

Assume that you have been assigned the 198.42.180.0/22 block of IP addresses.

You can carve that block up into smaller blocks. Specify an extended network prefix that allows the creation of 200 hosts on each subnet.

How many blocks can you create? How large are each?

IP Subnetting

Assume that you have been assigned the 198.42.180.0/22 block of IP addresses.

You can carve that block up into smaller blocks. Specify an extended network prefix that allows the creation of 200 hosts on each subnet.

How many blocks can you create? How large are each?

8 bits are needed (2^8 >200 > 2^7) Extended network prefix is /24 or 255.255.255.0

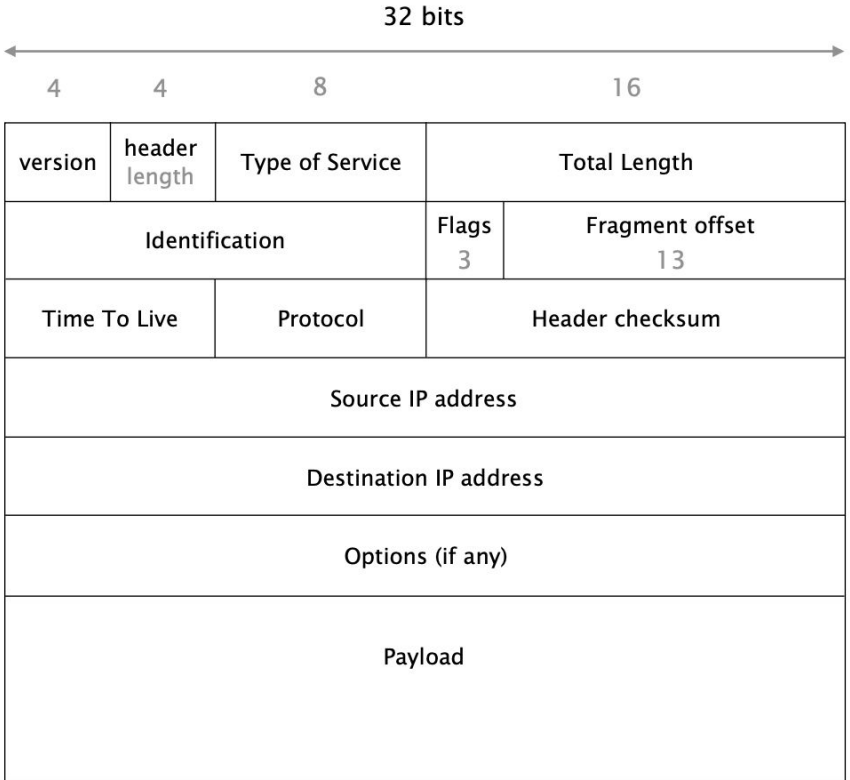
/24 == 254 hosts (broadcast address and network address need to be subtracted)

Four /24s are available:
198.42.180.0/24
198.42.181.0/24
198.42.182.0/24
198.42.183.0/24

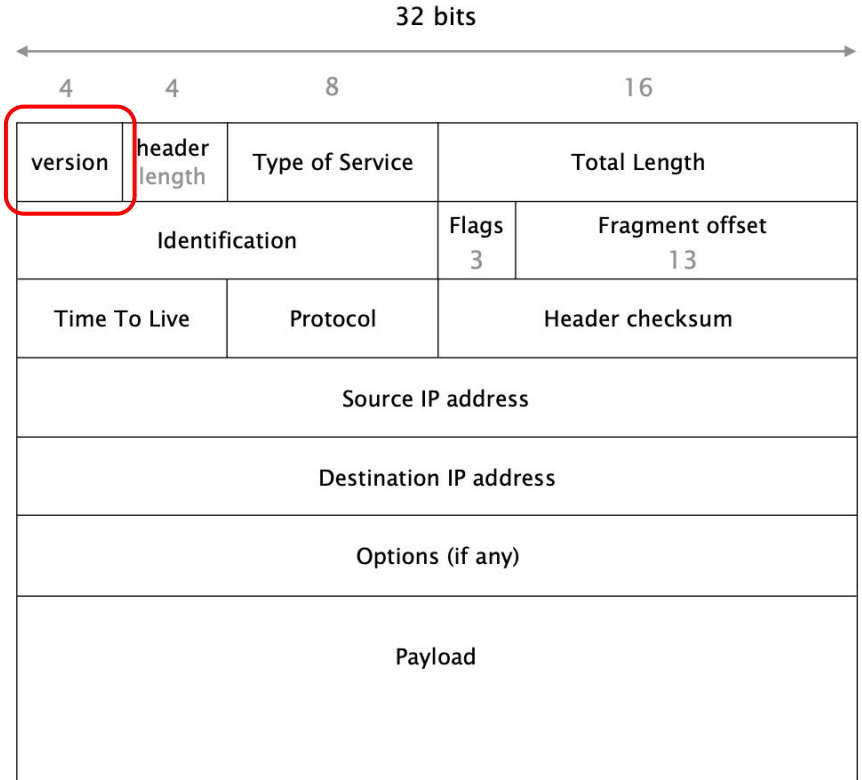
Network (IP) Layer

1. IP addresses
    - use, structure, allocation
2. IP forwarding
    - longest prefix match rule
3. IP header
    - IPv4 and IPv6, wire format

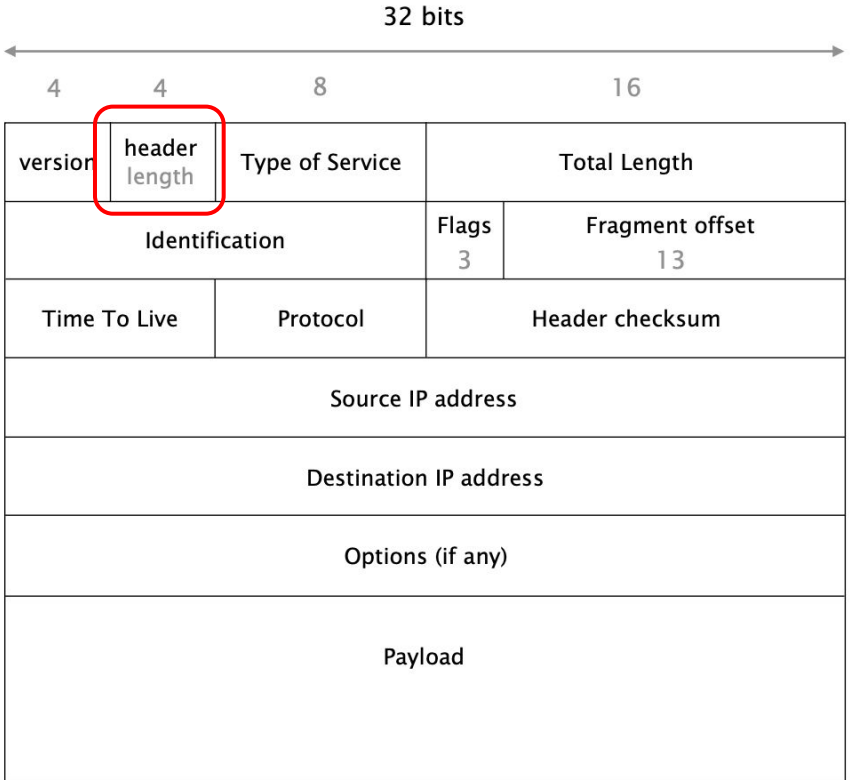
IPv4 Packet Header Looks Like This



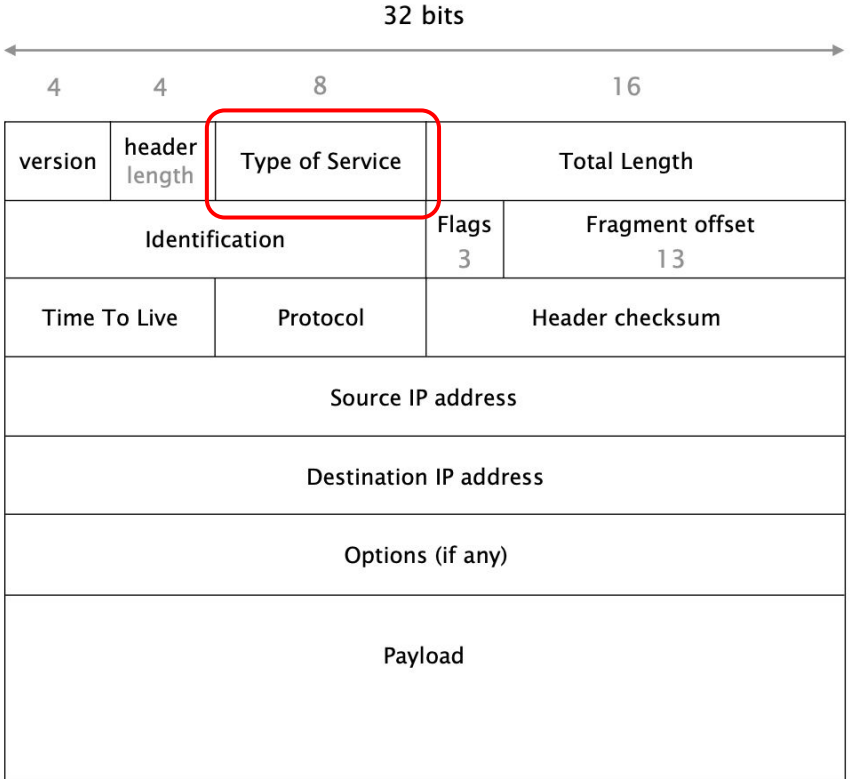
Version Lets Us Know What Fields to Expect (either 4 or 6)



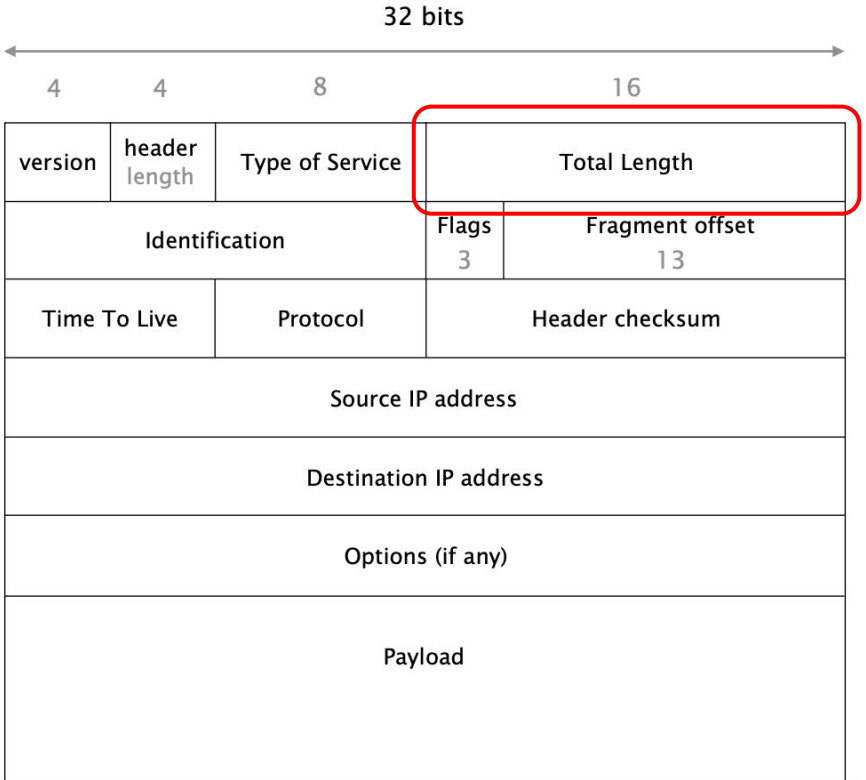
Header Length is the Number of 32-bit Words in the Header (typically 5 for 20 bytes)



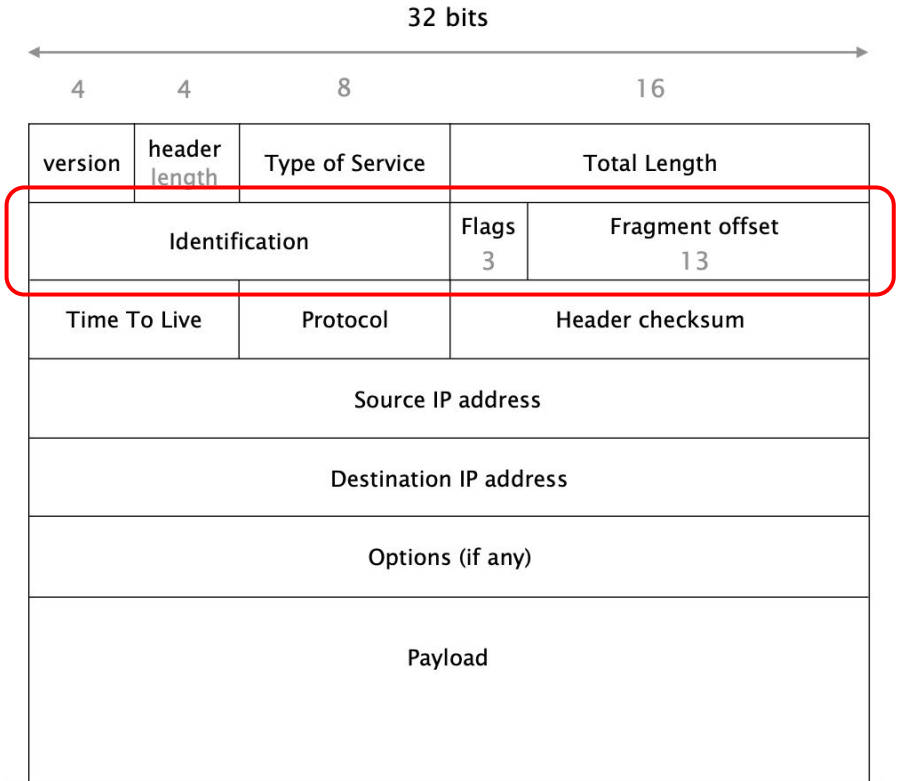
ToS Allows for Different Types of Packets to be Treated Differently (low delay for voice, high bandwidth for video, etc.)



Total Length is Entire Size of Packet (max 65535 bytes)



Next Three are About Fragmentation



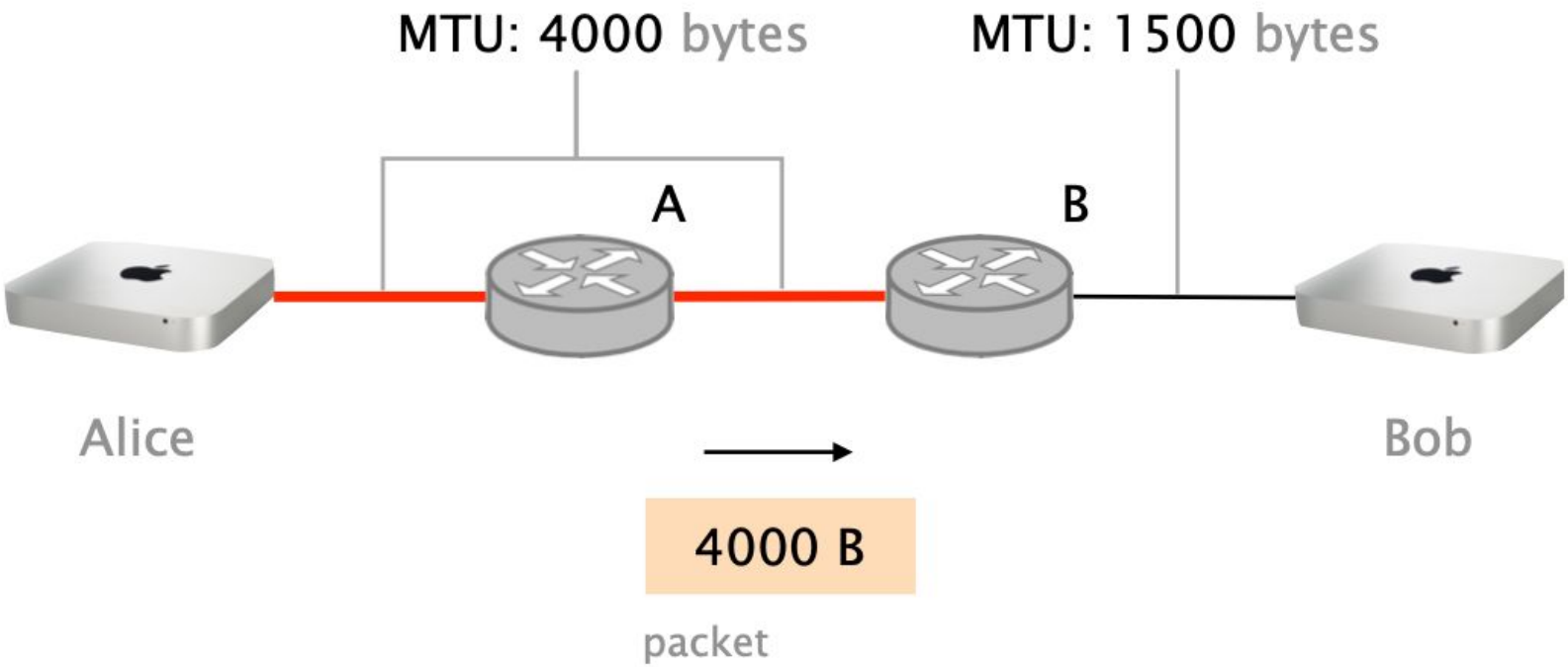
Every Link has a Maximum Transmission Unit (MTU)

- MTU is the max. # of bytes a link can carry as one unit
  - e.g., 1500 bytes for normal Ethernet

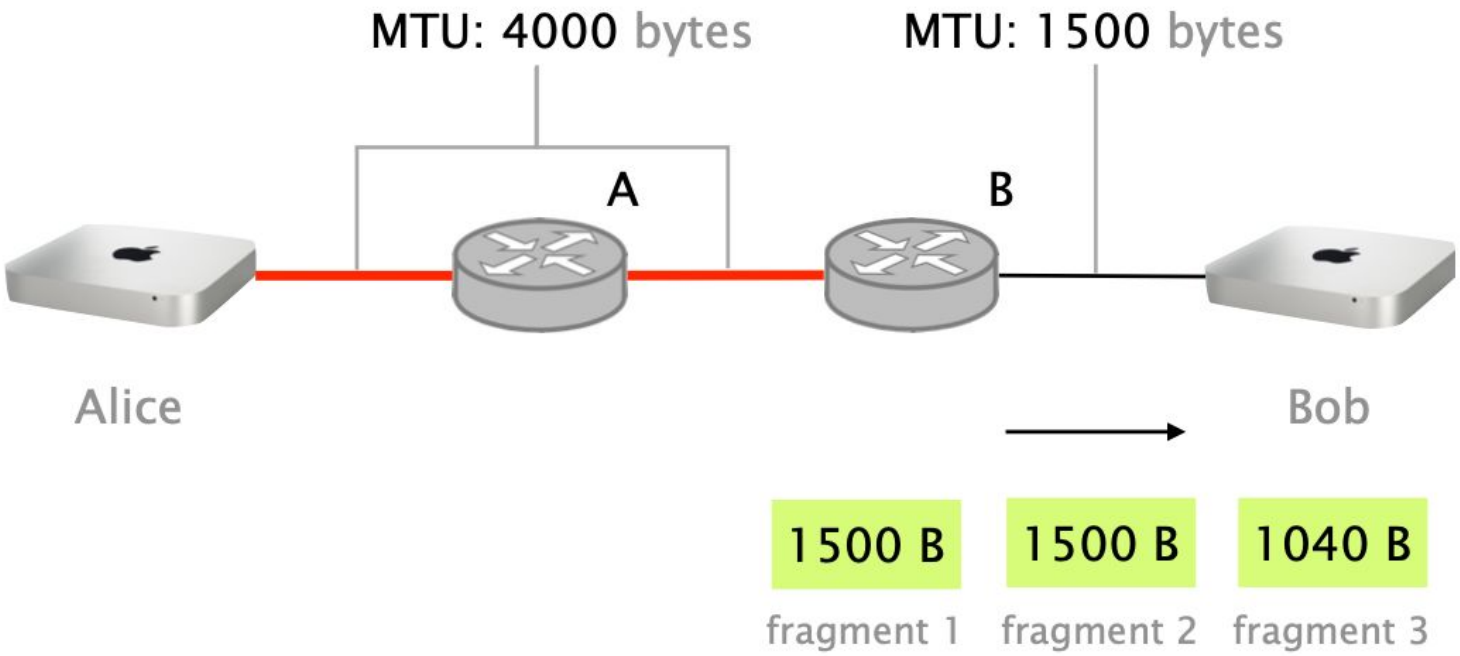
- A router can fragment a packet if the outgoing link MTU is smaller than the total packet size

- Fragmented packets are recomposed at the destination

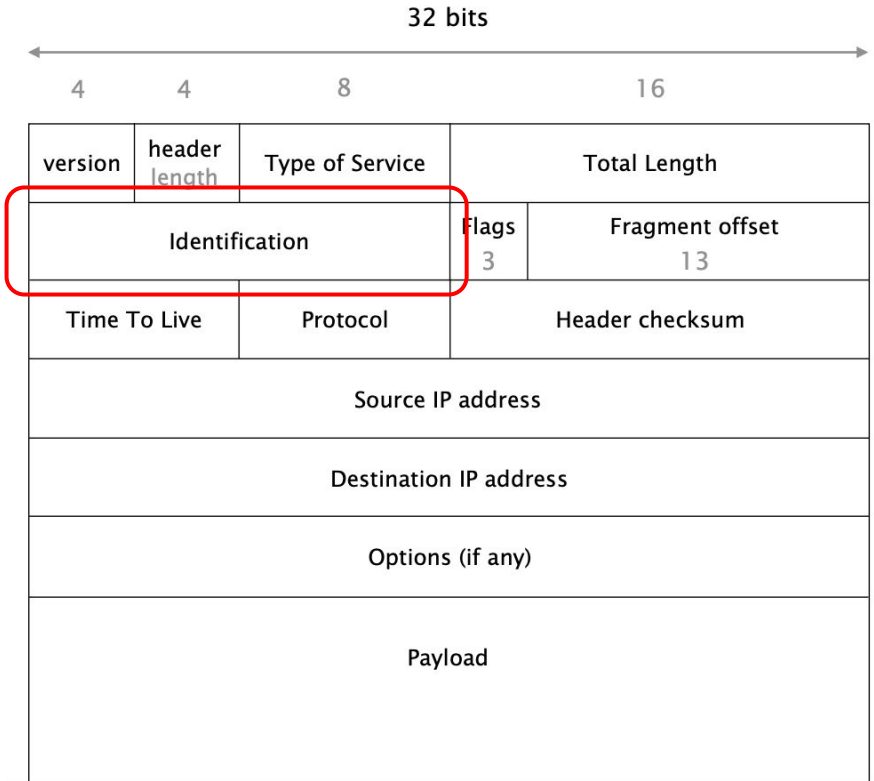
Fragmentation



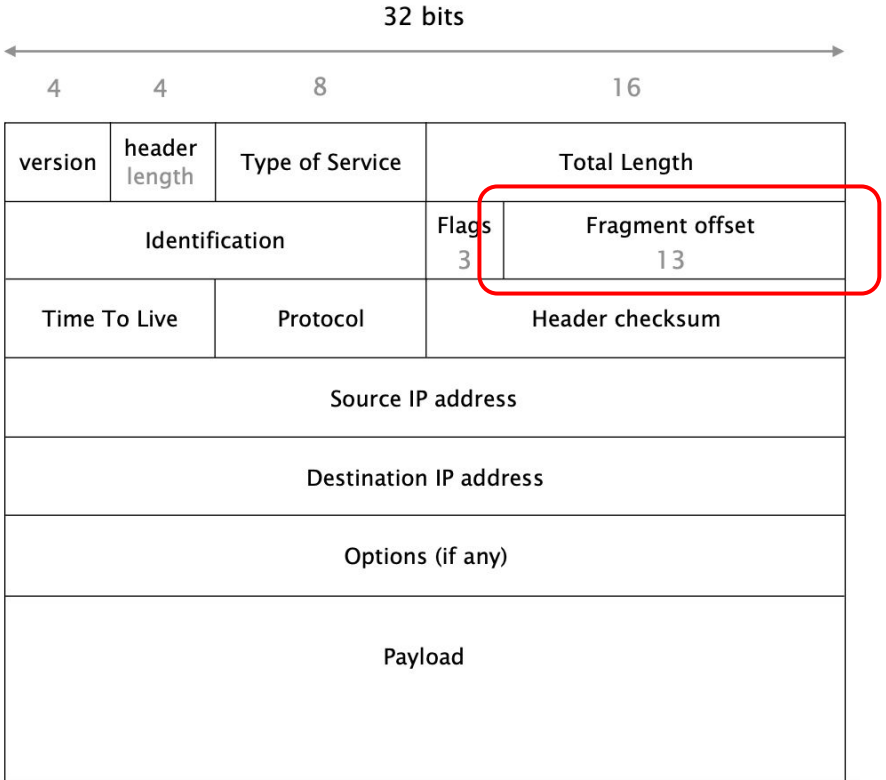
Fragmentation - Packet Larger than the MTU, Router B Fragments Packet



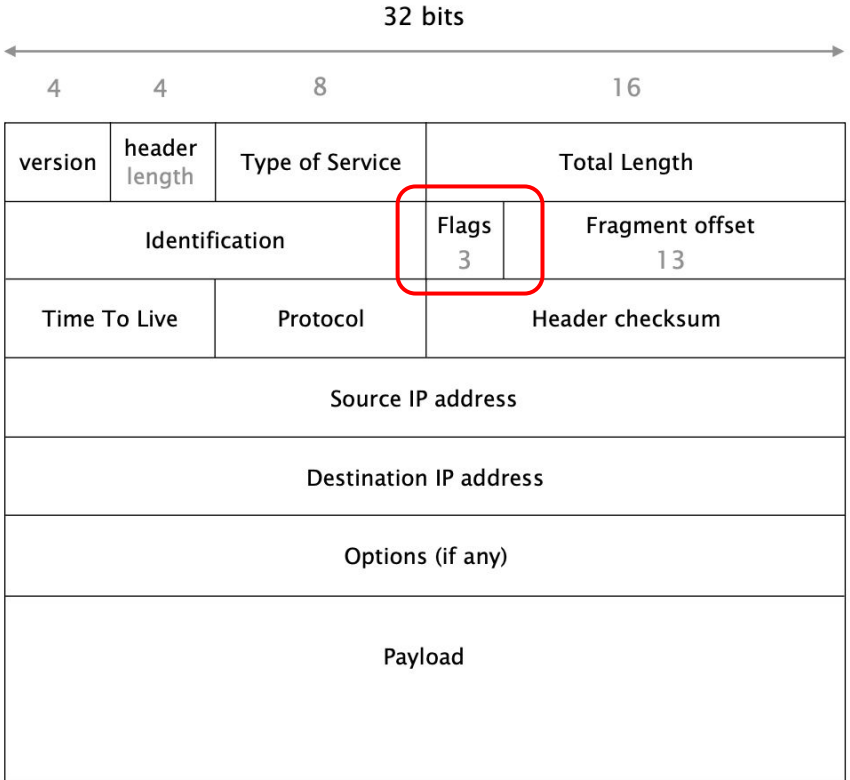
Identification Uniquely Identifies Fragments of a Given Packet



Fragment Offset Used to Put Fragments Back Together in Order



Flags Say Whether More Fragments are Coming

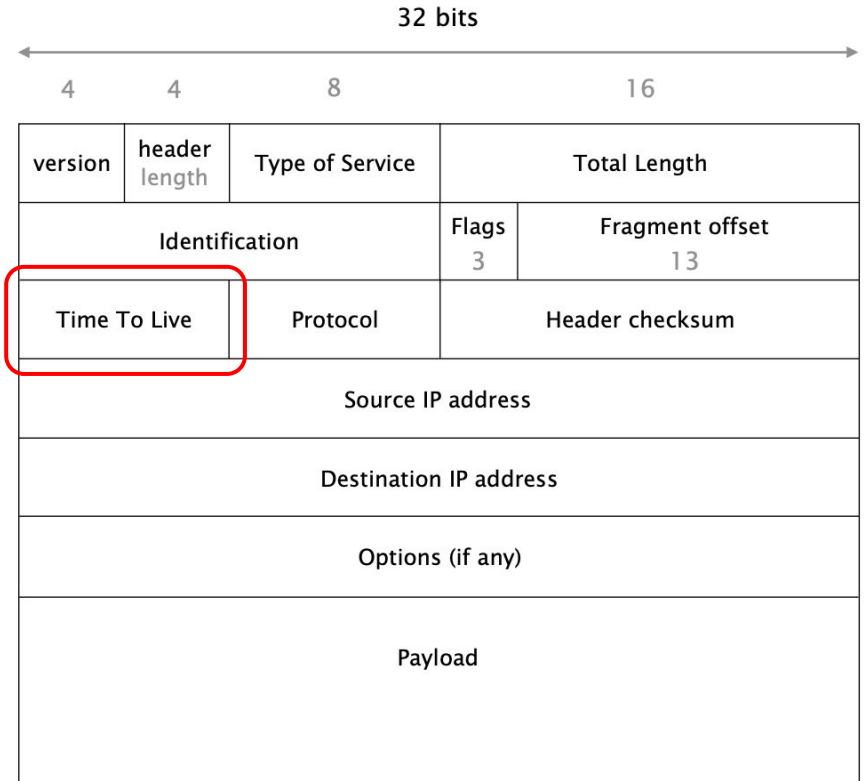


TTL Identifies Looping Packets, Dropped if They Reach Zero

Default TTL values:

Linux/MAC: 64
Windows: 128

Why would this be useful?

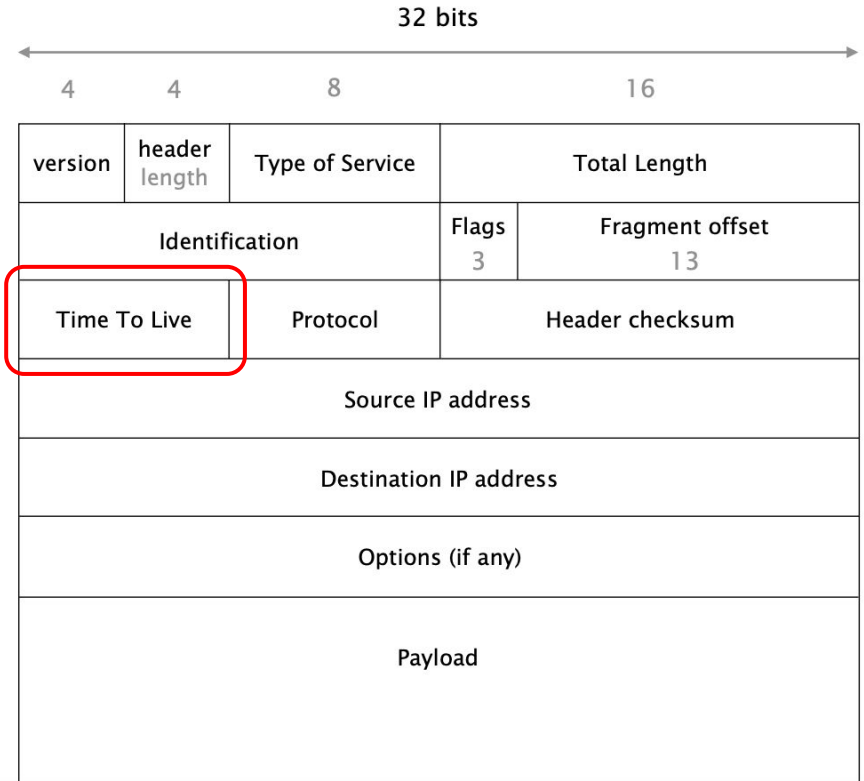


TTL Identifies Looping Packets, Dropped if They Reach Zero

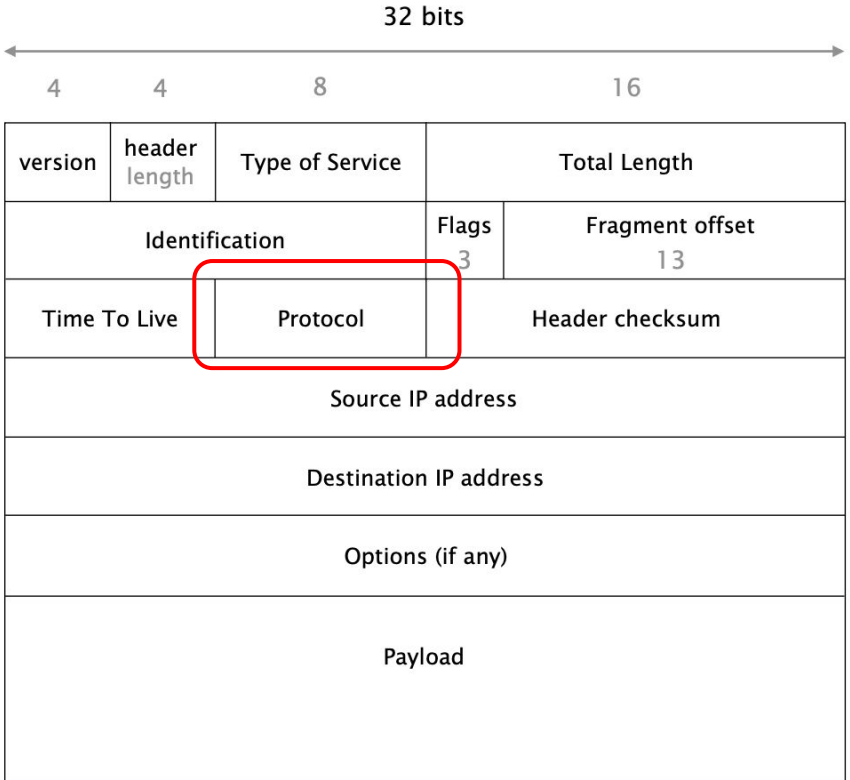
Default TTL values:

Linux/MAC: 64
Windows: 128

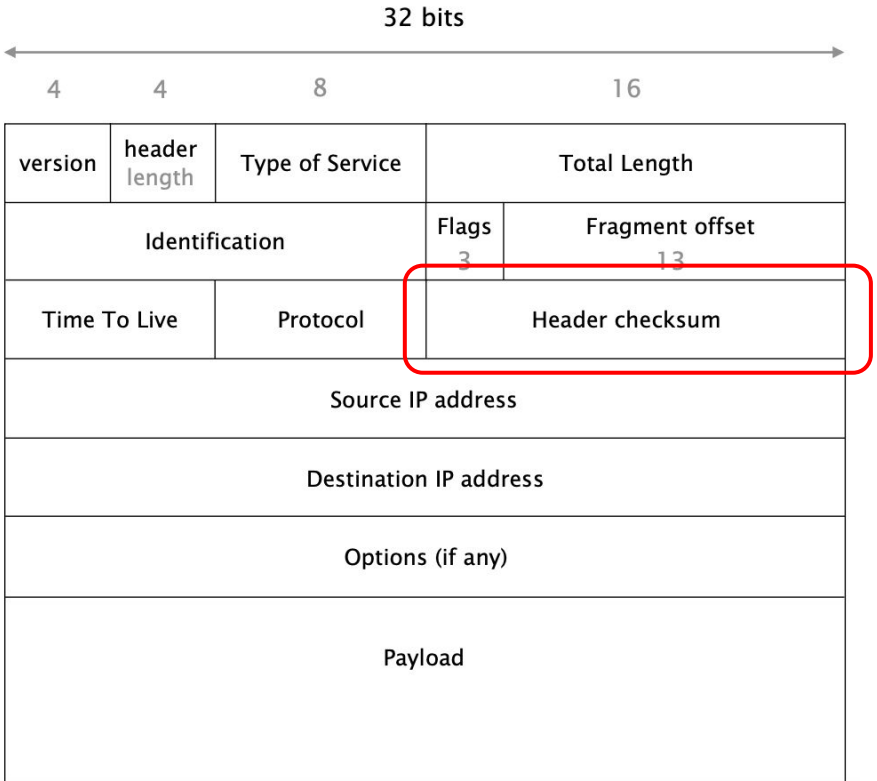
This can be used to fingerprint OSes



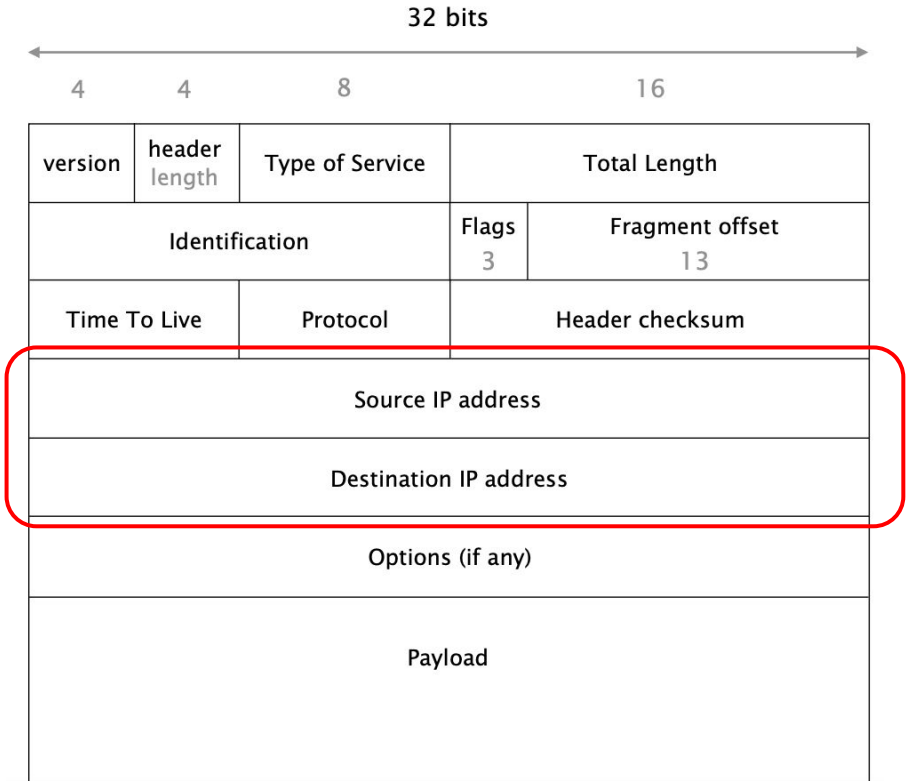
Protocol Identifies Higher Layer Protocol: 6 = TCP, 17 = UDP



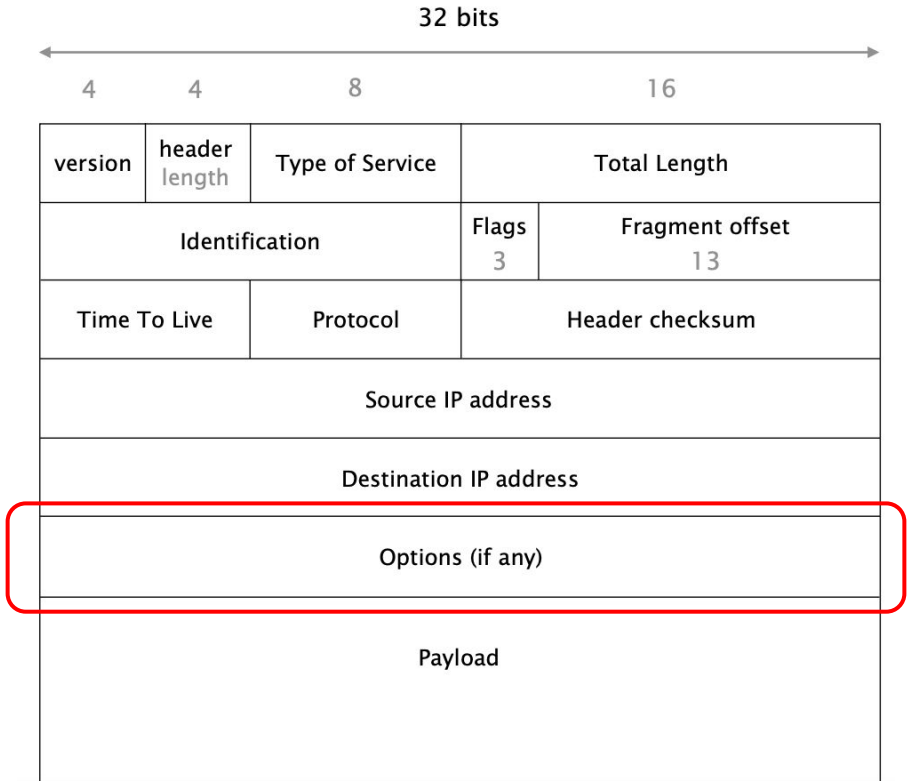
Checksum is the Sum of all 16 bit Words in the Header (NOT the payload)



Source and Destination IPs are the Host Endpoints



Options Provide Flexibility, They are Often Disabled



IPv4 vs IPv6

v4 still dominates the Internet (for now), even though we ran out of addresses

IPv6

Simpler than IPv4

- IPv6 was motivated by address exhaustion
  - IPv6 addresses are 128 bits long, huge address space
- IPv6 got rid of anything that wasn’t necessary
  - Fragmentation - leave problems to the endpoints
  - Checksum - leave problems to the endpoints
  - Header length - simplify handling
- Result is an elegant, boring, protocol
  - Allows for arbitrary options (IPv4 options have to be processed by every router - SLOW)

IPv6

Why hasn’t IPv6 taken over?

- Every device needs to support it (routers, middleboxes, end hosts, applications, ...)
- Most features we backported to IPv4 - no obvious advantage
- Network Address Translation (NAT) works
  - Most people have no idea we ran out of IPv4 addresses

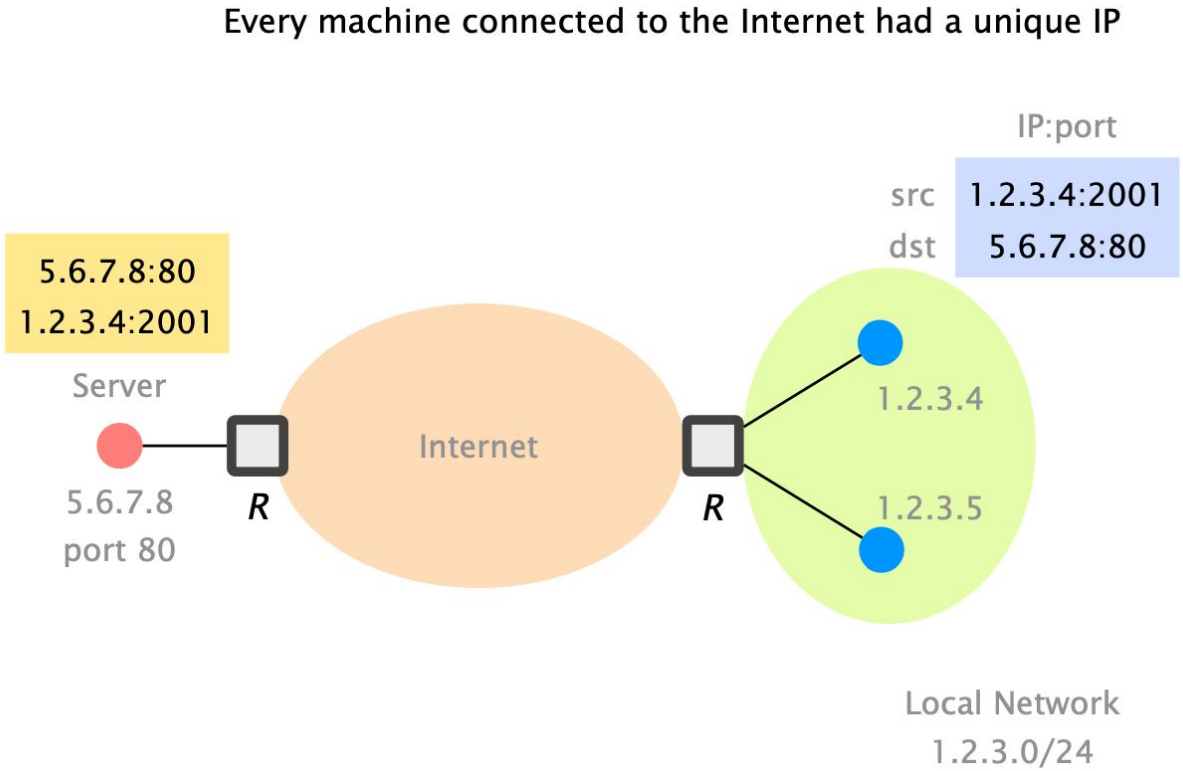
NAT

- Share a single (public) address or a pool (CG NATs) between hosts
  - Port numbers (transport layer) are used to distinguish

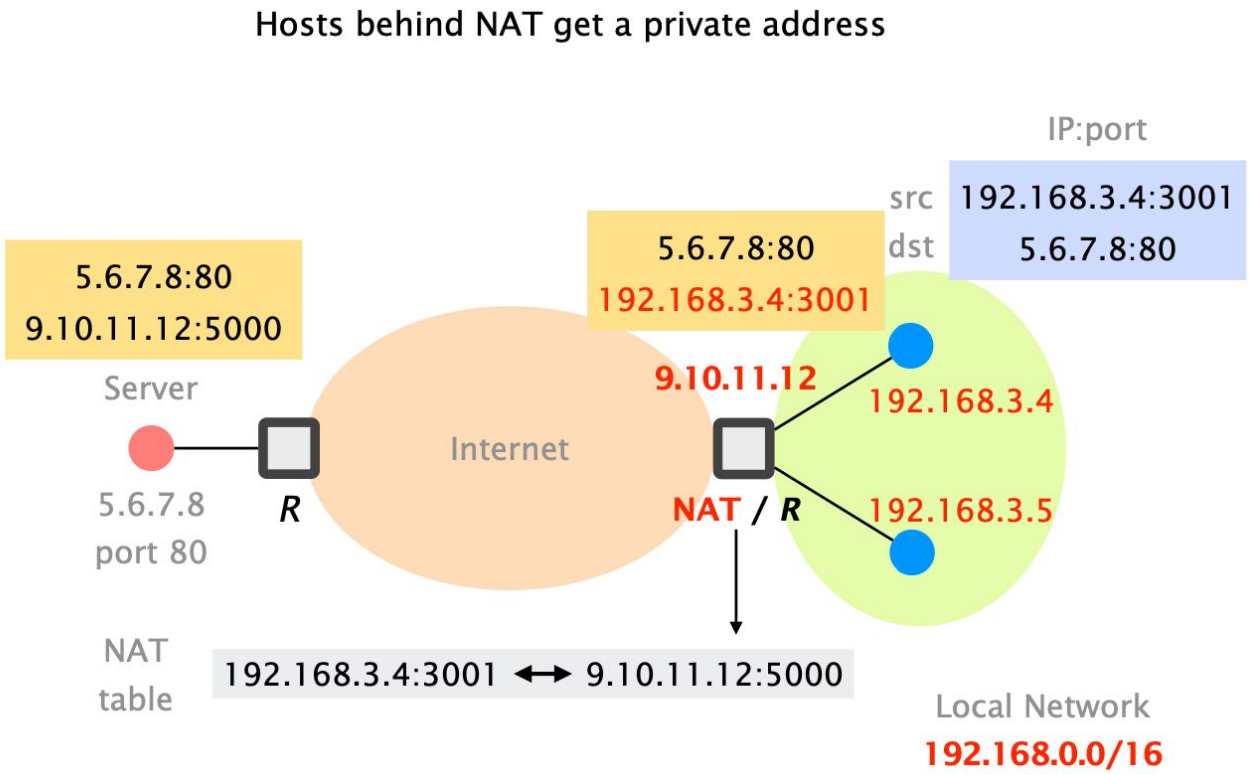
- One of the main reasons why we can still use IPv4
  - Saved us from address depletion

- Violates the general end-to-end principle of the Internet
  - A NAT box adds a layer of indirection

Before NAT

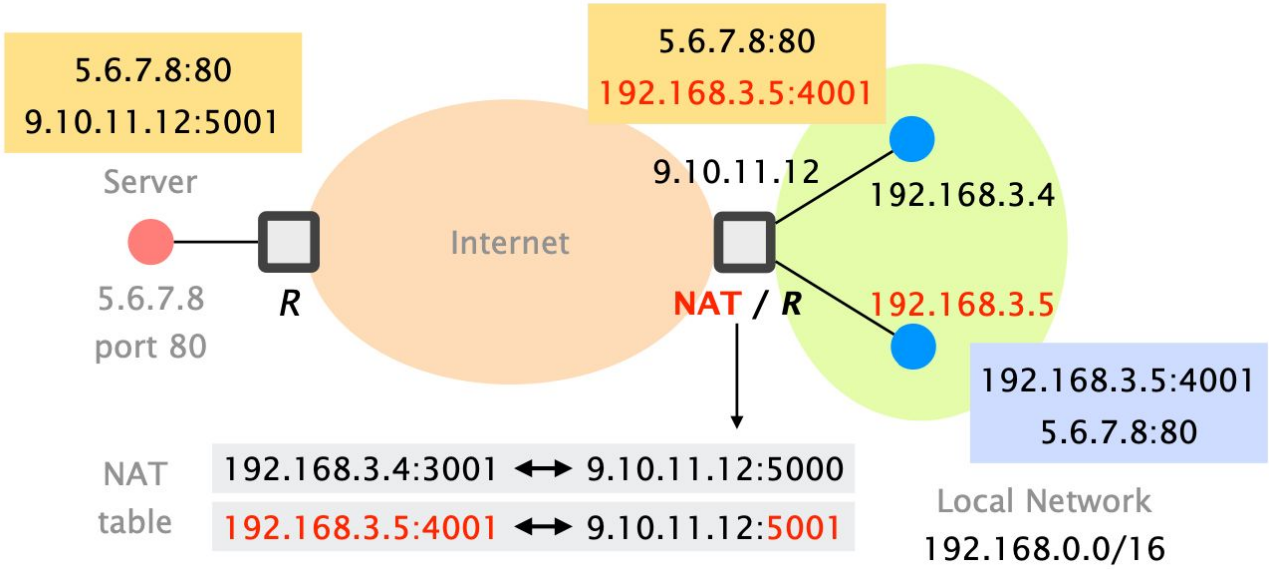

Every machine connected to the Internet had a unique IP
IP:port
src 1.2.3.4:2001
dst 5.6.7.8:80
5.6.7.8:80
1.2.3.4:2001
Server
5.6.7.8
port 80
R
Internet
R
1.2.3.4
1.2.3.5
Local Network
1.2.3.0/24

After NAT


Hosts behind NAT get a private address
IP:port
src
dst
192.168.3.4:3001
5.6.7.8:80
5.6.7.8:80
9.10.11.12:5000
5.6.7.8:80
192.168.3.4:3001
Server
9.10.11.12
192.168.3.4
Internet
5.6.7.8
port 80
R
NAT / R
192.168.3.5
NAT
table
192.168.3.4:3001 ↔ 9.10.11.12:5000
Local Network
192.168.0.0/16

After NAT

The port numbers are used to multiplex single addresses



NAT Pros and Cons

- Better privacy/anonymization
  - All hosts in one network get the same public IP
  - But, cookies, browser version, ... still identify hosts
- Better security
  - From the outside you cannot directly reach the hosts
  - Problematic e.g., for online gaming
- Limited scalability (size of the mapping table)
  - Example: Wi-Fi access problems in public places (e.g., lecture hall) often due to a full NAT table