Internet Protocol (IP)

Problems

- How can I send a message across the internet that will reach the right destination?
- How can I send messages of larger size than some networks along the route can carry?
- How can I know what protocols are being used to send this message (and so interpret the message correctly)?

Overview

- Fragmentation
- IPv4 header
- IPv6 header
- Path MTU
- ICMP

The Internet Protocol

- Used to send messages across the internet
- To cope with physical network bandwidth, messages may be split into **fragments** and each fragment sent individually

- IP adds a **header** to each fragment to form an IP **datagram**

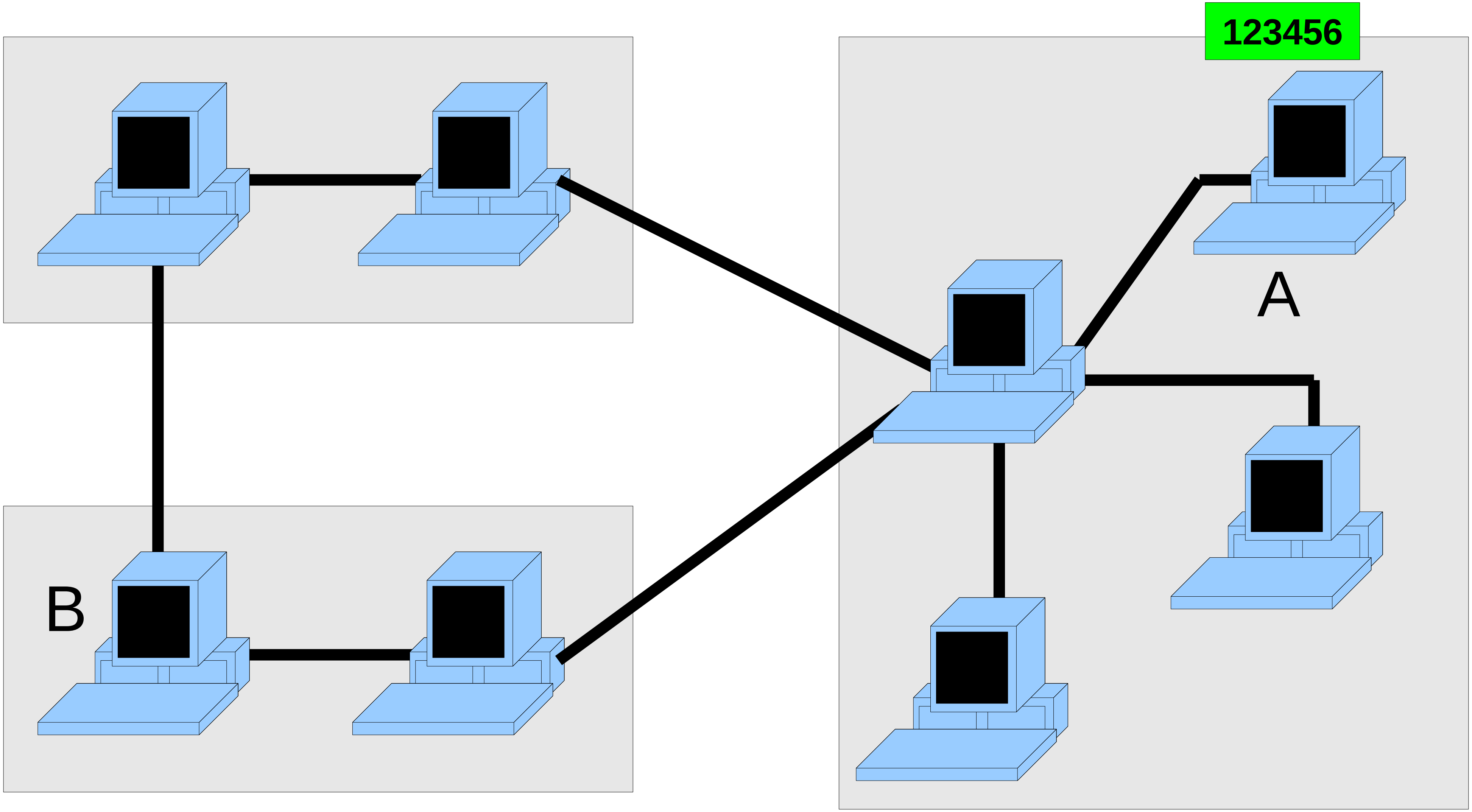
Issues not solved by IP

- It is unreliable, meaning that it provides no guarantees about the message successfully reaching its destination
- The message may:
  - Arrive corrupted
  - Arrive with fragments out of order
  - Arrive multiple times (duplicated)
  - Be lost entirely
- Higher level protocols, such as TCP, add reliability to IP

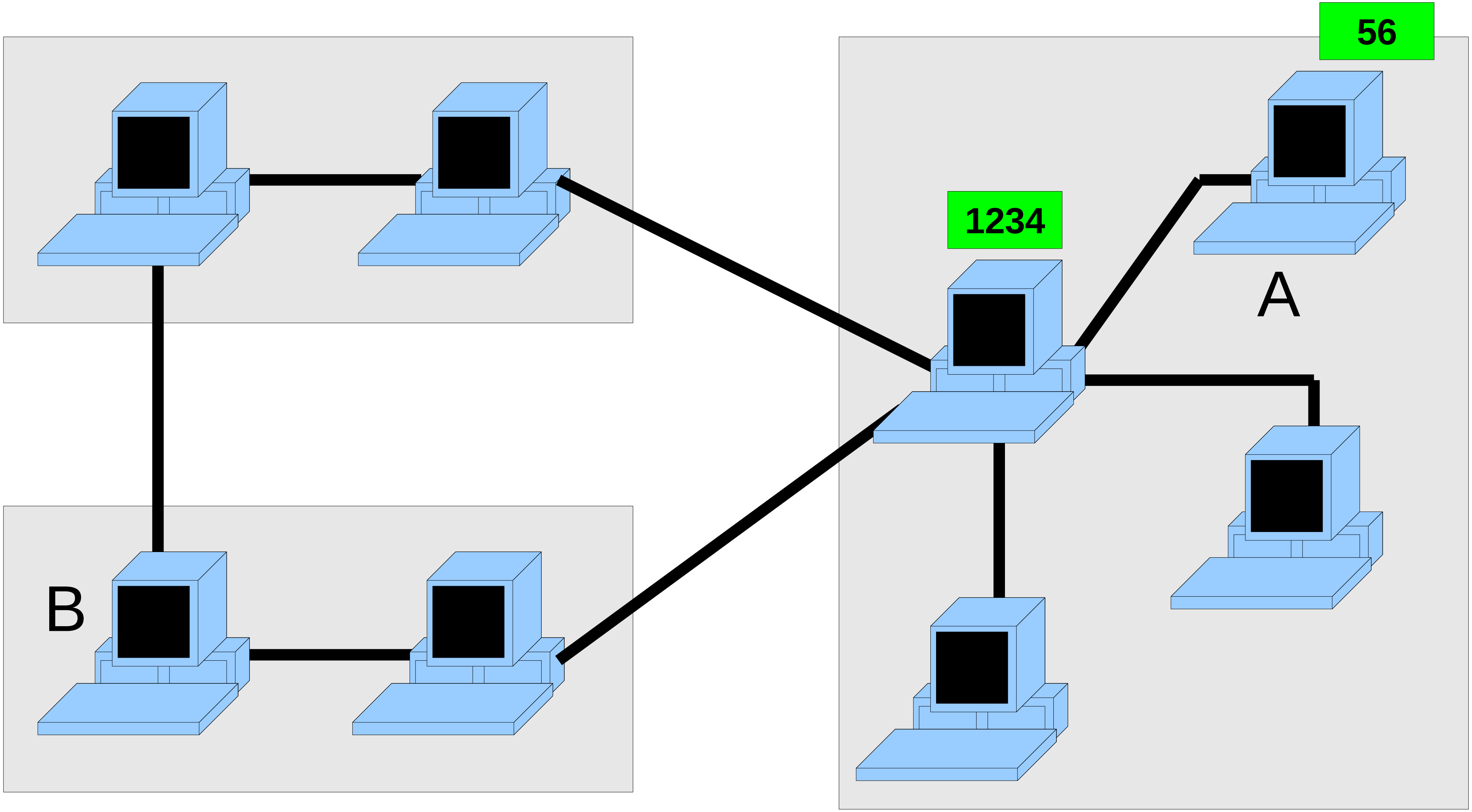
Fragmentation

- Each network has a limit on the size of message it can carry: the **Maximum Transmission Unit (MTU)** of the network
- The sender fragments messages to fit the local MTU
- In IPv4, the fragments may be further fragmented on moving from one network to another with smaller MTU
- Fragments are reassembled at the destination
- In fragmentation, every fragment must be a multiple of 8 bytes (64 bits) except the last one

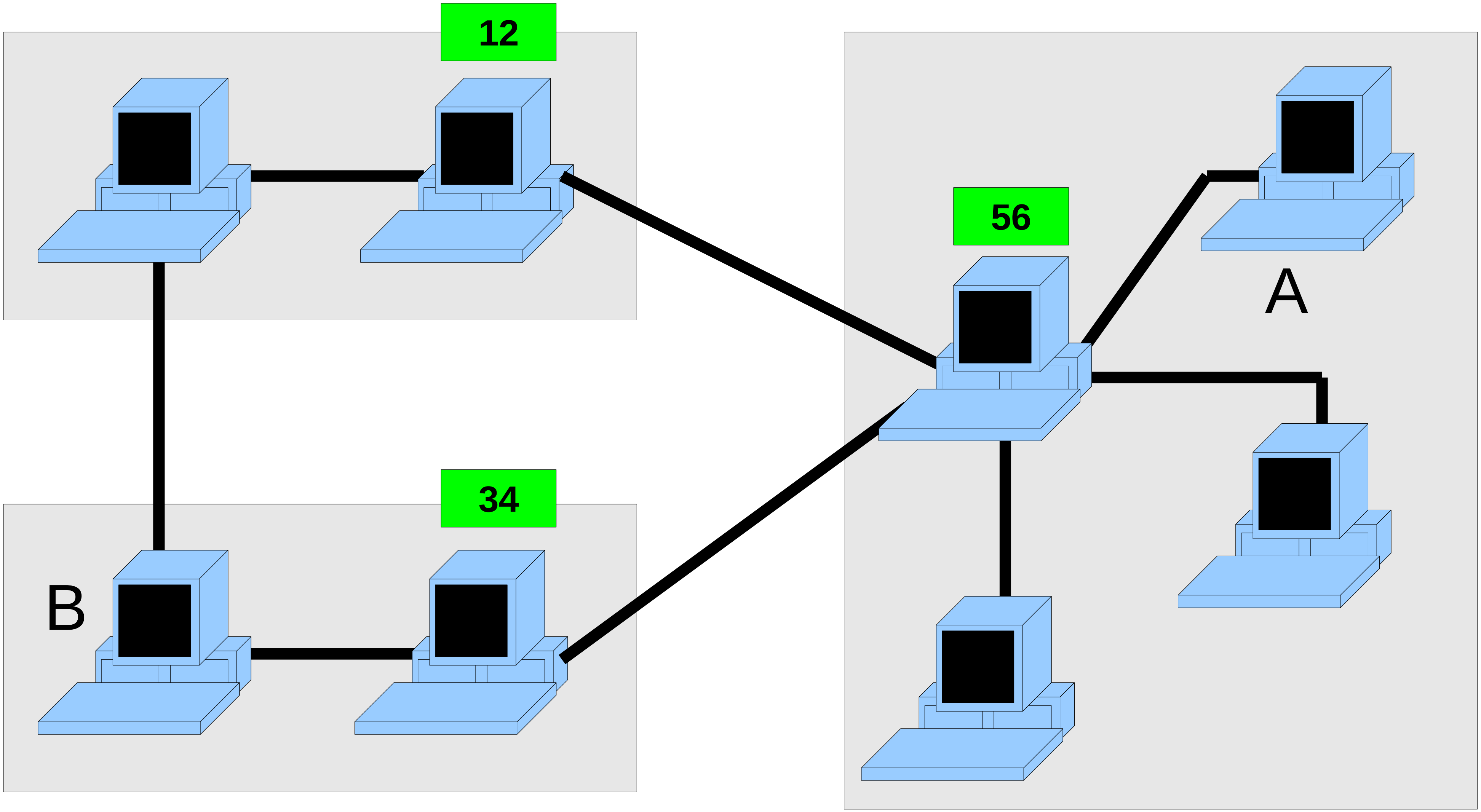
Fragmentation and networks



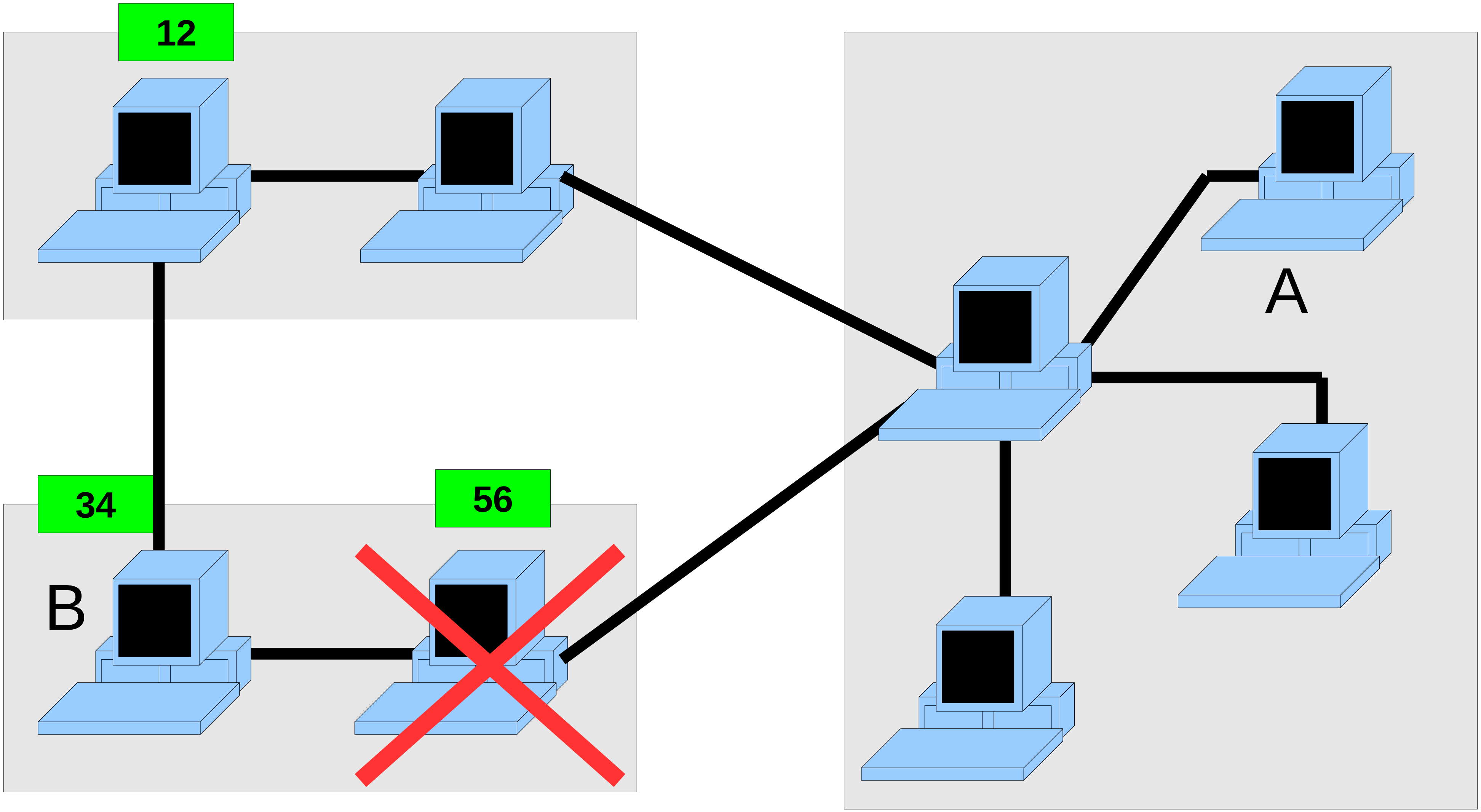
Fragmentation and networks



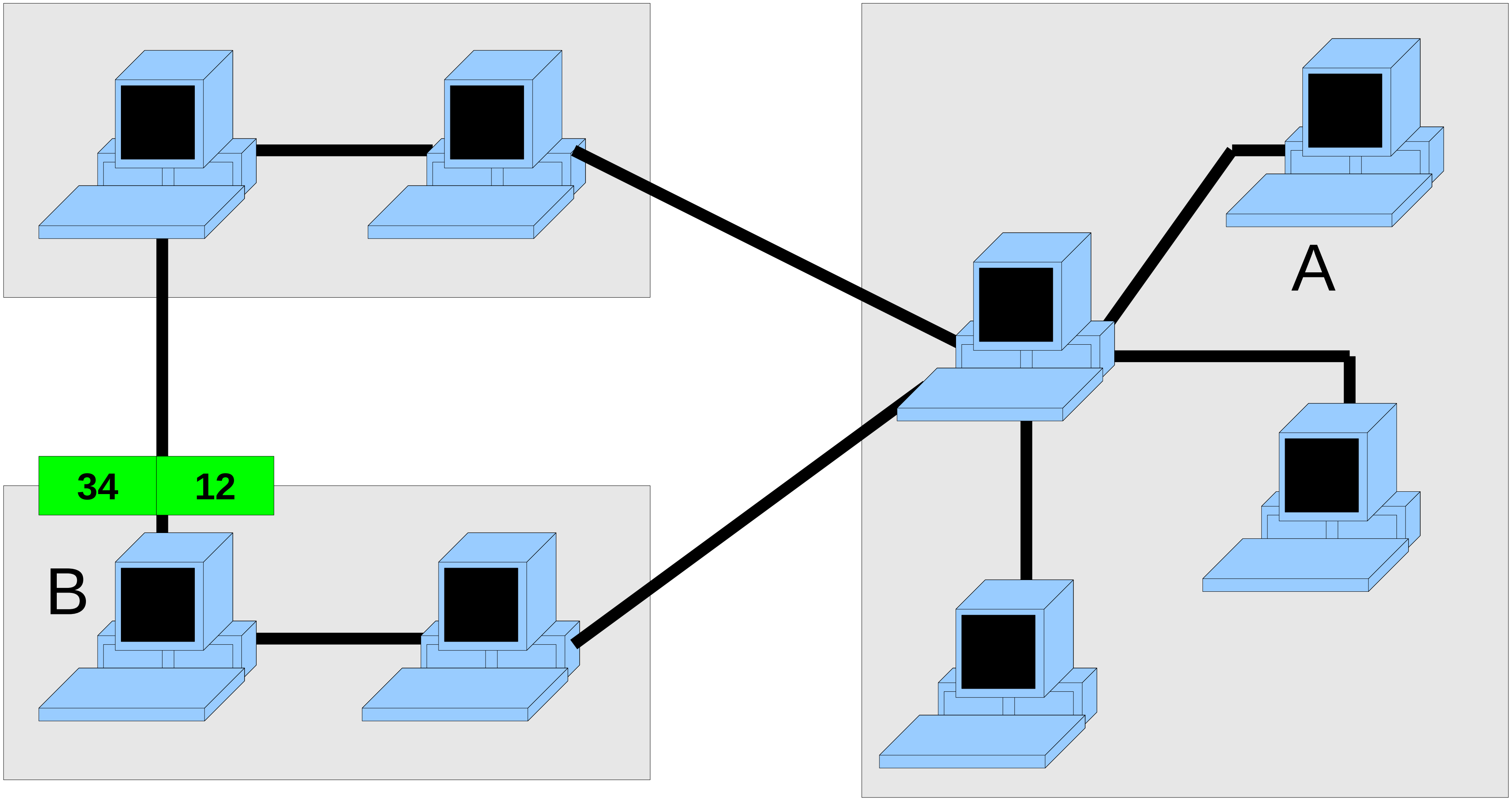
Fragmentation and networks



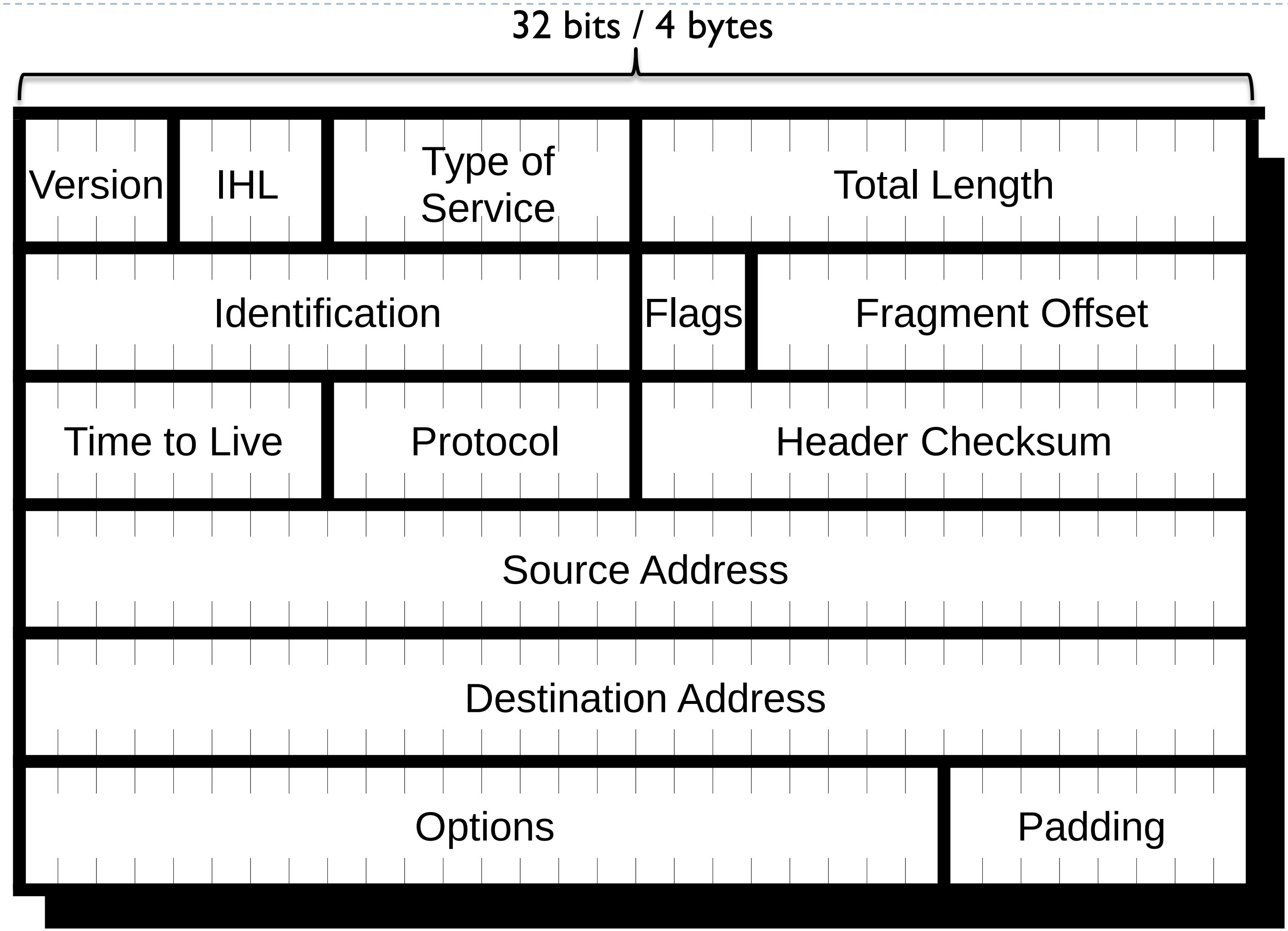
Fragmentation and networks



Fragmentation and networks



IPv4 header



An IP header with no options is 20 bytes long

Version field

- The version of IP being used
- 4 bits: 0 - 15

- 0100 (4) for IPv4
- 0110 (6) for IPv6

Internet header length (IHL) field

- Length of this header in 32-bit words
- 4 bits: 0 - 15
- Minimum value 5, to include all header fields
- Header length must be a multiple of 32 bits
- Padding field makes it up to correct length

IHL	Header in bits
5	160
6	192
10	320

- .....

Type of service field

- Used by message sender to specify how message should be handled in transmission
- 8 bits

- Originally used as set of flags specifying choices of the kind of service required, e.g.
  - Higher throughput vs higher reliability
- Redefined by more recent protocols handling network congestion

Total length field

- Total number of bytes in the whole datagram
- 16 bits : 0 – 65,535

- Header plus fragment data
- Measured in bytes/octets (multiples of 8 bits)
- Minimum: 20

Identification field

- Identifies whole of fragmented message
- 16 bits : 0 – 65,535

- For a receiving host
  - Every fragment with same Identification from same sending host is part of the same message
  - Can then reconstruct message from fragments

Flag fields

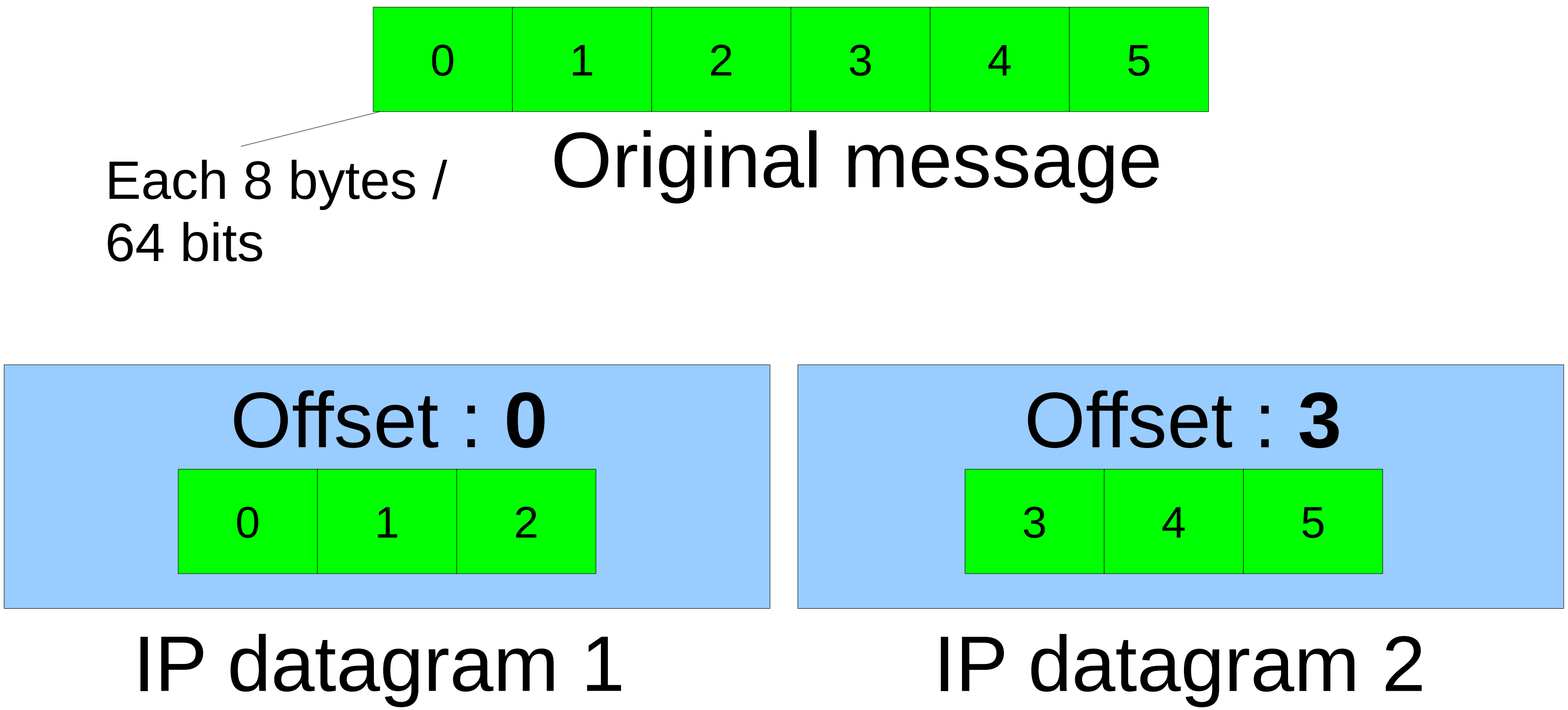
- A flag is an informational bit (0 = no, 1 = yes)
- 3 bits in the IPv4 header = 3 flags

- **Reserved**: First bit must always be 0
- **Don't Fragment (DF)**: This data should not be fragmented during transmission
- **More Fragments (MF):** This fragment is followed by others in the message

Fragment offset field

- The fragment offset (FO) field specifies where in the original message the fragment in this datagram starts
- Used to re-assemble message from fragments in the correct order
- It is a position from the start of the message counted in 8 bytes (64 bits) units, so FO=3 means this fragment starts after 24 bytes in the original message (8 x 3 = 24)
- The FO field is 13 bits long, so allows values 0 - 8191

Fragment Offset illustration



Time to live field

- How long this datagram is allowed to remain in the system before being deleted
- 8 bits : 0 – 255

- Every host handling the message must reduce Time to Live by 1
- When it reaches 0, the datagram should be destroyed
- Prevents data from circling round internet in an endless loop when they cannot reach their intended destination

Protocol field

- The host-to-host transport layer protocol being used by the message sent over IP
- 8 bits

- Protocols include TCP: 6, UDP: 17, ICMP: 1
- Each has a numeric identifier
- Protocol identifiers are assigned and maintained by the Internet Assigned Numbers Authority (IANA), part of ICANN

- http://www.iana.org (Protocol Numbers)

Header checksum field

- Checksum on the header, 16 bits
- Sender, receiver perform same check
- Used to ensure integrity in transmission
  1. One's complement sum of 16 bit words in header
  2. One's complement of final value

<u>One's complement sum</u>

```
      11111111 11111100
      00000000 00000100
    ---------------------
    1 00000000 00000000
                      1
    ---------------------
      00000000 00000001
```

<u>One's complement</u>

```
00110011 10101010
-----------------
11001100 01010101
```

Source and destination fields

- IP address of sending/source host
- IP address of destination host (intended recipient)

- 32 bits each

10001001 01001001 00001001 11101000

137.73.9.232

Options and padding

- Optional arguments usable by IP processing software
- Variable number of bits
- Cover routing options, tracing of route etc.
- Not used much in practice
- See IPv4 specification (RFC 791) for more

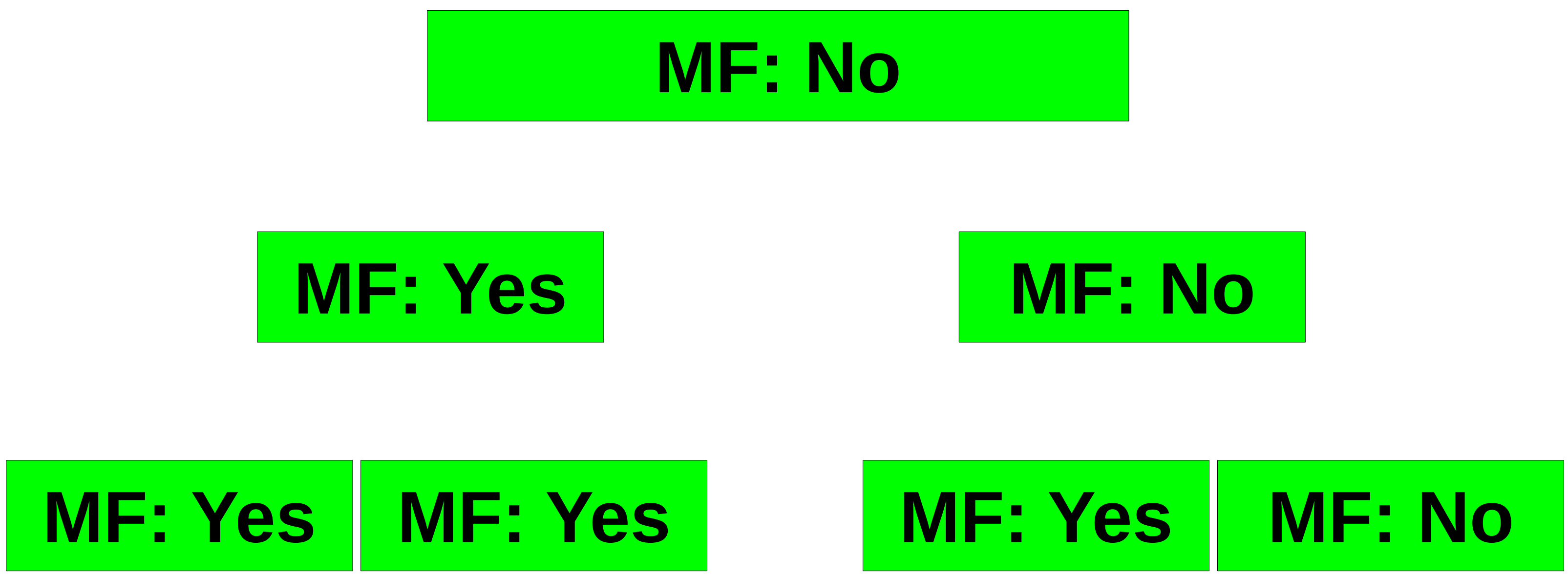
- Followed by padding of 0s to make header up to multiple of 32 bits

Changes in fragmentation

- In IPv4, when moving from one network to another with a smaller MTU, datagrams may be re-fragmented
- Fragments must be a multiple of 8 bytes except for the last one (because the Fragment Offset is counted in 8 byte units)
- Some of the header fields will remain the same in the fragments, others will change
- For example, the Total Length field gives the length for the datagram plus header, so changes in fragmentation

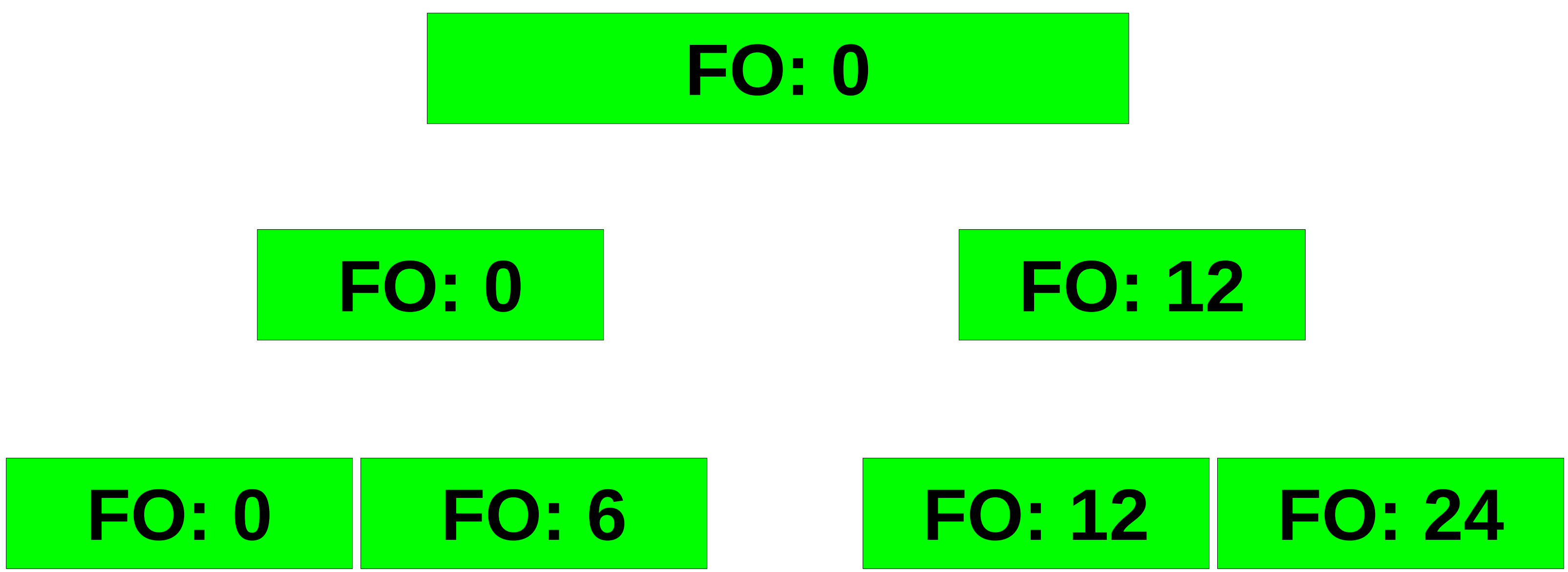
More Fragments flag changes

- More Fragments Flag becomes 1 (Yes) in all but the last fragment, where it is the same as in the original datagram

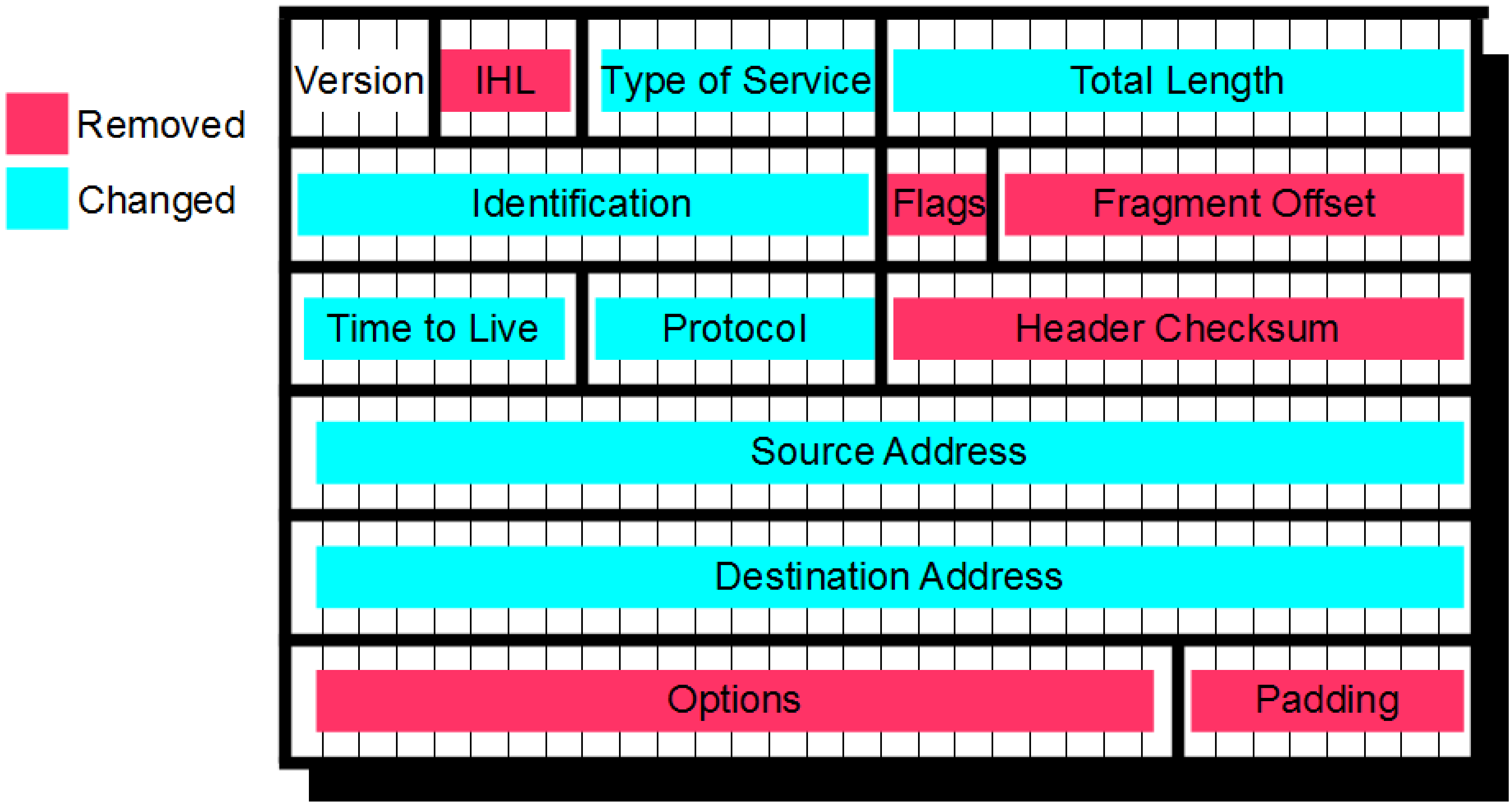


Fragment Offset field changes

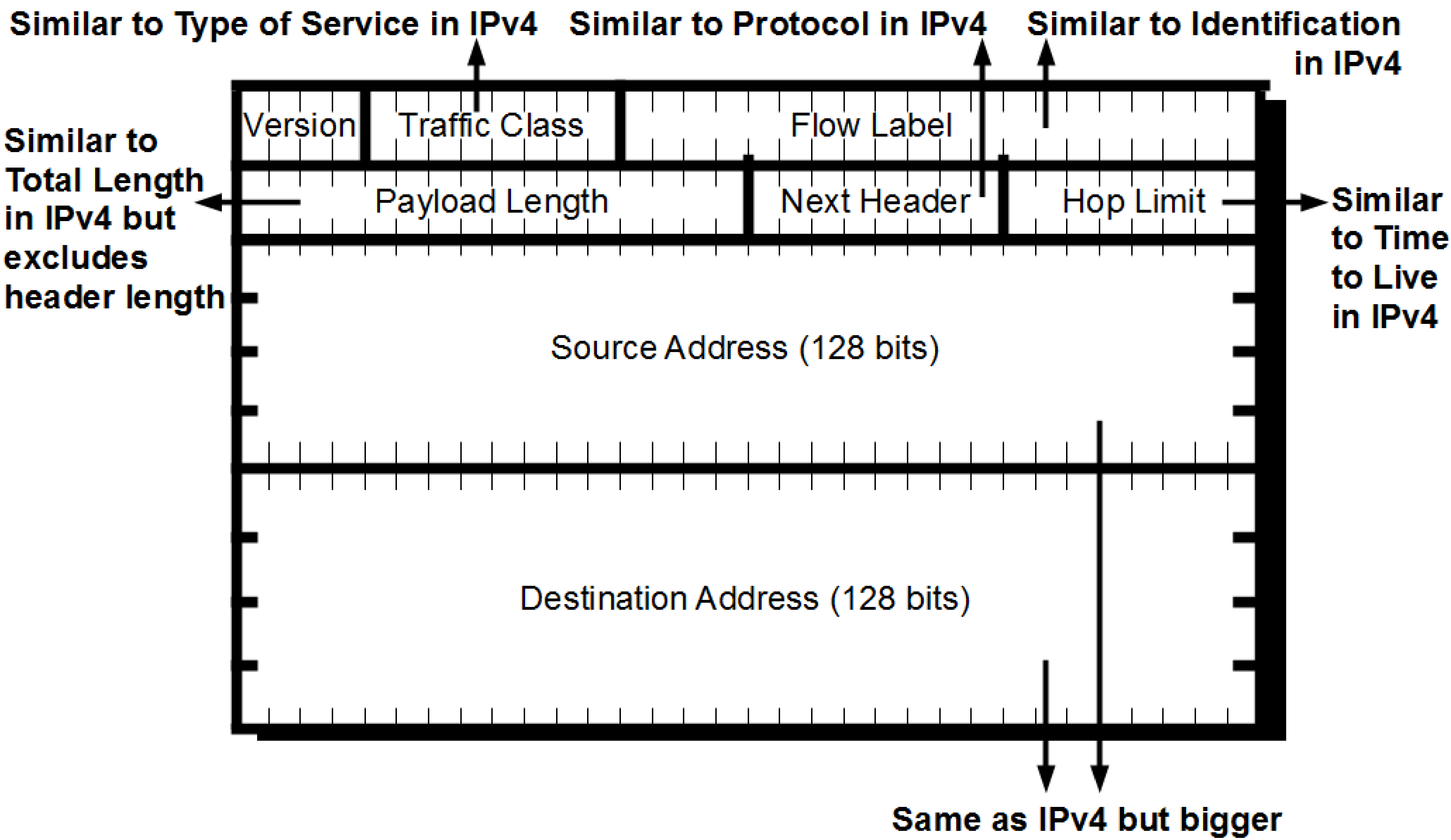
- On fragmentation of message M, the Fragment Offset is the sum of
  - The Fragment Offset in M's header
  - The new fragment's position in M, counted in 8-byte words from the start of the fragment data



Header changes from IPv4 to IPv6



IPv6 header


Similar to Type of Service in IPv4
Similar to Protocol in IPv4
Similar to Identification in IPv4
Similar to Total Length in IPv4 but excludes header length
Version
Traffic Class
Flow Label
Payload Length
Next Header
Hop Limit
Similar to Time to Live in IPv4
Source Address (128 bits)
Destination Address (128 bits)
Same as IPv4 but bigger

IPv6 options

- IPv6 also allows options but, unlike IPv4, they are not all in one block inside the header
- Instead, an IPv6 header can be followed by an **extension header** containing optional data
- An extension header can be followed by another extension header: a chain of options
- The Next Header field is used to specify that the next block of data is an extension header, and which kind.

Extension headers

IPv6 Header Next Header = TCP	TCP Header, Body

IPv6 Header Next Header = Routing	Routing Header Next Header = TCP	TCP Header, Body

IPv6 Header Next Header = Routing	Routing Header Next Header = Fragment	Fragment Header Next Header = TCP	TCP Header, Body

Fragment option

- One example of an IPv6 extension header is the **fragment extension header**
- If a message is fragmented (by the sender), the receiver has to be able to re-assemble it
- In order to do this, a Fragment Offset field is needed, as in IPv4
- Where a IPv6 packet contains a fragment of a message it will have a fragment extension header containing the fragment offset
- Re-assembly occurs just as in IPv4

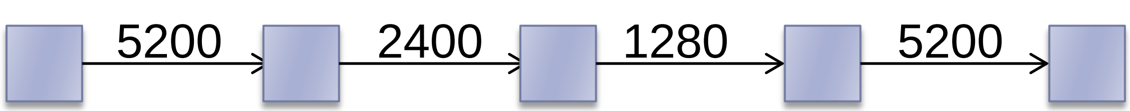
MTU values

- On configuring a router, the MTU for each connection to a local or remote network is set
- It is set by default to the maximum value for the network access protocol used to communicate with the network, e.g. 1500 for Ethernet

- To see the MTUs obtained by IP software:
- netstat -nr to find the interface (network access protocol instantiation) of each route
- netstat -i to get the MTU of an interface

Path MTU

- IPv6 does not use fragmentation in transport
- It requires the sender to ensure messages are small enough to cross the networks **before** sending them
- The Path MTU is the minimum MTU on the path from sender to receiver



Path MTU = 1280

- Sender fragments message to ensure all parts are smaller than the Path MTU
- Path MTU can also be used with IPv4

Path MTU discovery algorithm

1. Initially assume that the Path MTU is the MTU of the first hop in the path (link to first router)
2. Fragments message to current assumed Path MTU, sends first fragment
3. If fragment reaches a link where it is too large to pass through, because the MTU is too small:
   a) An ICMP Packet Too Big messages is return to the sender, stating that smaller MTU
   b) Sender sets its Path MTU to MTU in ICMP message
   c) Try again from step 2 above
4. If the first fragment reaches its destination, rest of message is sent, fragmented to known Path MTU

IPv4 Don’t Fragment

- If the Don't Fragment Flag is set in an IPv4 message, but the message needs to be fragmented to move onto a network with a lower MTU, an error message is sent by ICMP
- ICMP code 4: fragmentation needed and DF set
- Sent back to the IP message sender

Internet Control Message Protocol

- ICMP is part of IP and used to report information on IP processing
- ICMP reports errors and other info regarding IP processing back to the datagram's source
- Examples of ICMP messages include
  - destination unreachable
  - echo request and reply (generated by ping)
  - redirect
  - time exceeded
  - router advertisement and router solicitation

RFCs

Internet Protocol v4:	791
Internet Procotol v6:	2460
Internet Control Message Protocol	792
Fragment reassembly:	815
Path MTU Discovery:	1981

- http://www.tcpipguide.com/free

Overview

- Fragmentation
- IPv4 header
- IPv6 header
- Path MTU
- ICMP