IP (INTERNET PROTOCOL)

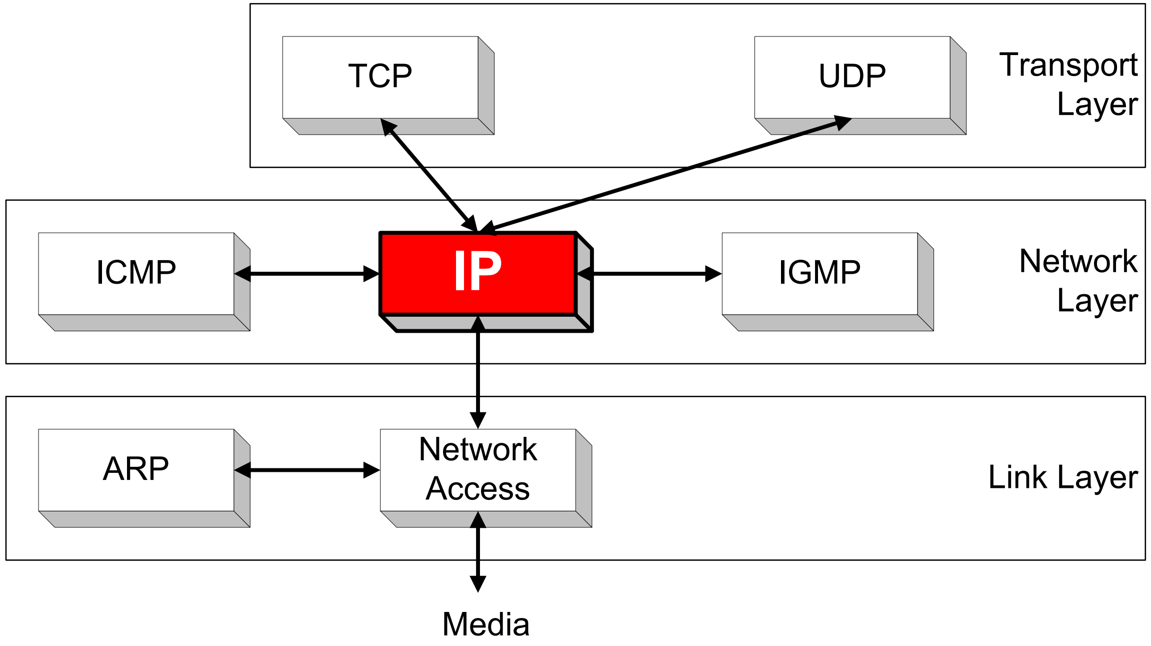
Mobile Computing

Prof. Jongwon Yoon

Intelligent Machines Lab.

Overview

- IP (Internet Protocol) is a Network Layer Protocol.



- IP's current version is Version 4 (IPv4). It is specified in RFC 891.

Five Basic Design Decisions

1. Packet-switching
2. Best-effort service model
3. Layering
4. A single internetworking layer
5. The end-to-end principle (and fate-sharing)

1. What is Packet Switching?

- Divide messages into a sequence of packets
- Network deals with each packet individually
- Means that each packet must contain all relevant network information in its header
  - Design of protocol almost synonymous with header

→ Achieve higher levels of utilization (statistical multiplexing)

→ Avoid per-flow state inside the network

- Plenty of routing state, but no per-flow state
- Follows from notion of fate-sharing
- Enables robust fail-over

2. What is “Best Effort”?

- Network makes no service guarantees
  - Just gives its “best effort
- Service can be imperfect
  - Packets may be lost, corrupted, delivered out of order, delayed

- → BE means never having to say you’re sorry…
  - No need to reserve bandwidth and memory
  - No need to do error detection & correction
  - No need to remember from one packet to next
  - No need to make packets follow same path
- → Easier to survive failures
  - Transient disruptions are okay during failover
- → Simplifies interconnection between networks
  - Minimal service promises

Use Higher Layers to Compensate

- No error detection or correction
  - Higher-level protocol can provide error checking
- Successive packets may not follow the same path
  - Not a problem as long as packets reach the destination
- Packets can be delivered out-of-order
  - Receiver can put packets back in order (if necessary)
- Packets may be lost or arbitrarily delayed
  - Sender can send the packets again (if desired)
  - Receiver can buffer packets for smooth playout
- No network congestion control (beyond "drop")
  - Sender can slow down in response to loss or delay

3. What is and Why Layering?

- Modularity partitions functionality into modules
- Laying is a particularly simple form of modularity
- Modules only deal with layers above and below
  - Simplifies interactions between modules
  - Simplifies introduction of new protocols

4. Why one networking layer protocol?

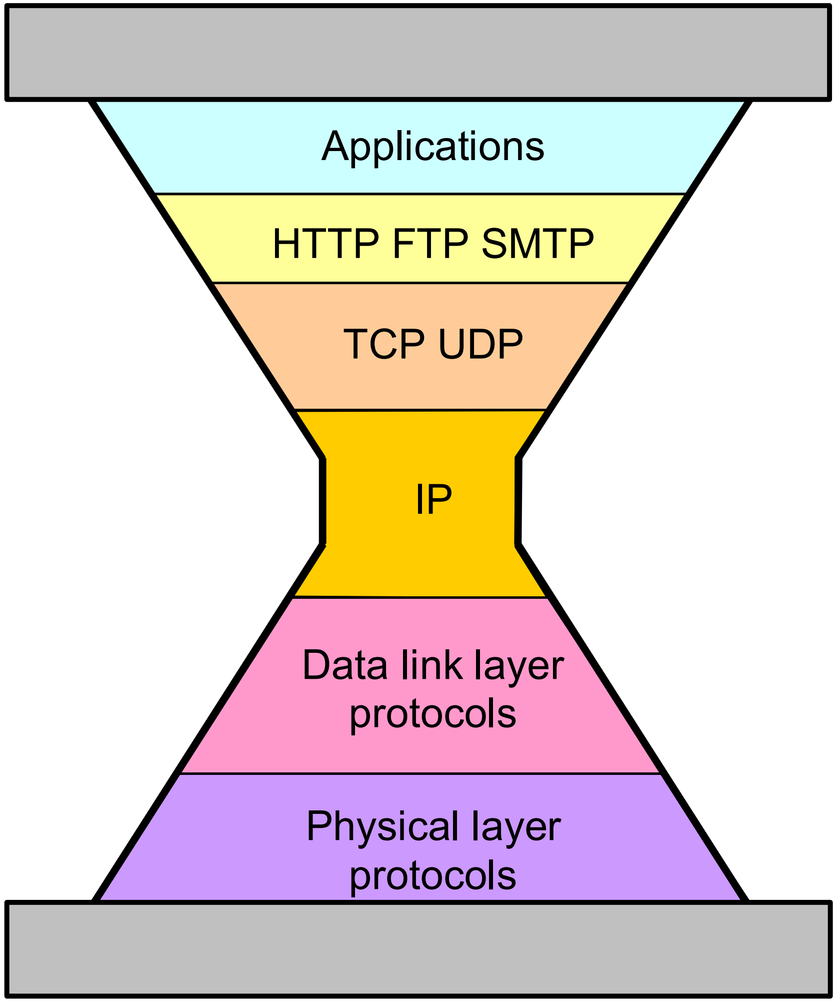
- Unifies the architecture
- As long as applications can run over IP-based protocols, they can run on any network
- As long as networks support IP, they can run any application

IP: The waist of the hourglass

- IP is the waist of the hourglass of the Internet protocol architecture

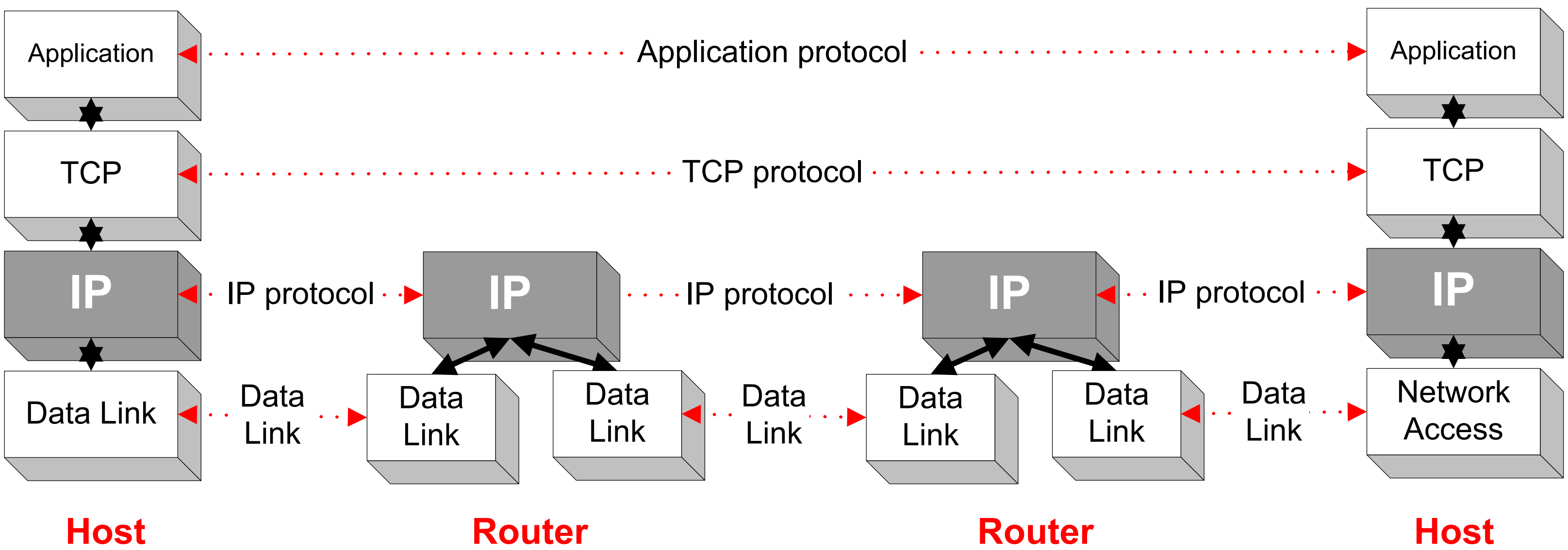
- Multiple higher-layer protocols
- Multiple lower-layer protocols

- Only one protocol at the network layer.



Application protocol

- IP is the highest layer protocol which is implemented at both routers and hosts



5. End-to-End Principle

- E2E Principle guides placement of functionality
  - Application-specific features reside in the communicating end nodes of the network (intermediate routers exist to establish the network)
  - If hosts can implement functionality correctly, implement it in a lower layer only as a performance enhancement
  - But do so only if it does not impose burden on applications that do not require that functionality

IP Service

- Delivery service of IP is minimal

- IP provides an unreliable connectionless best effort service (datagram service).
  - **Unreliable:** IP does not make an attempt to recover lost packets
  - **Connectionless:** Each packet (datagram) is handled independently. IP is not aware that packets between hosts may be sent in a logical sequence
  - **Best effort:** IP does not make guarantees on the service (no throughput guarantee, no delay guarantee,…)

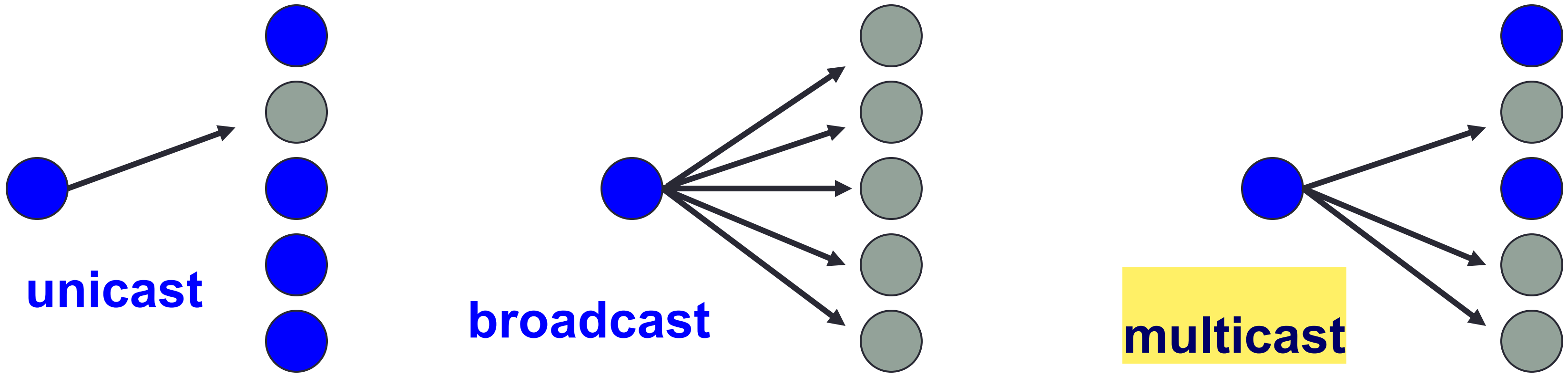
- Consequences:

  - Higher layer protocols have to deal with losses or duplicate packets

  - Packets may be delivered out-of-sequence

IP Service

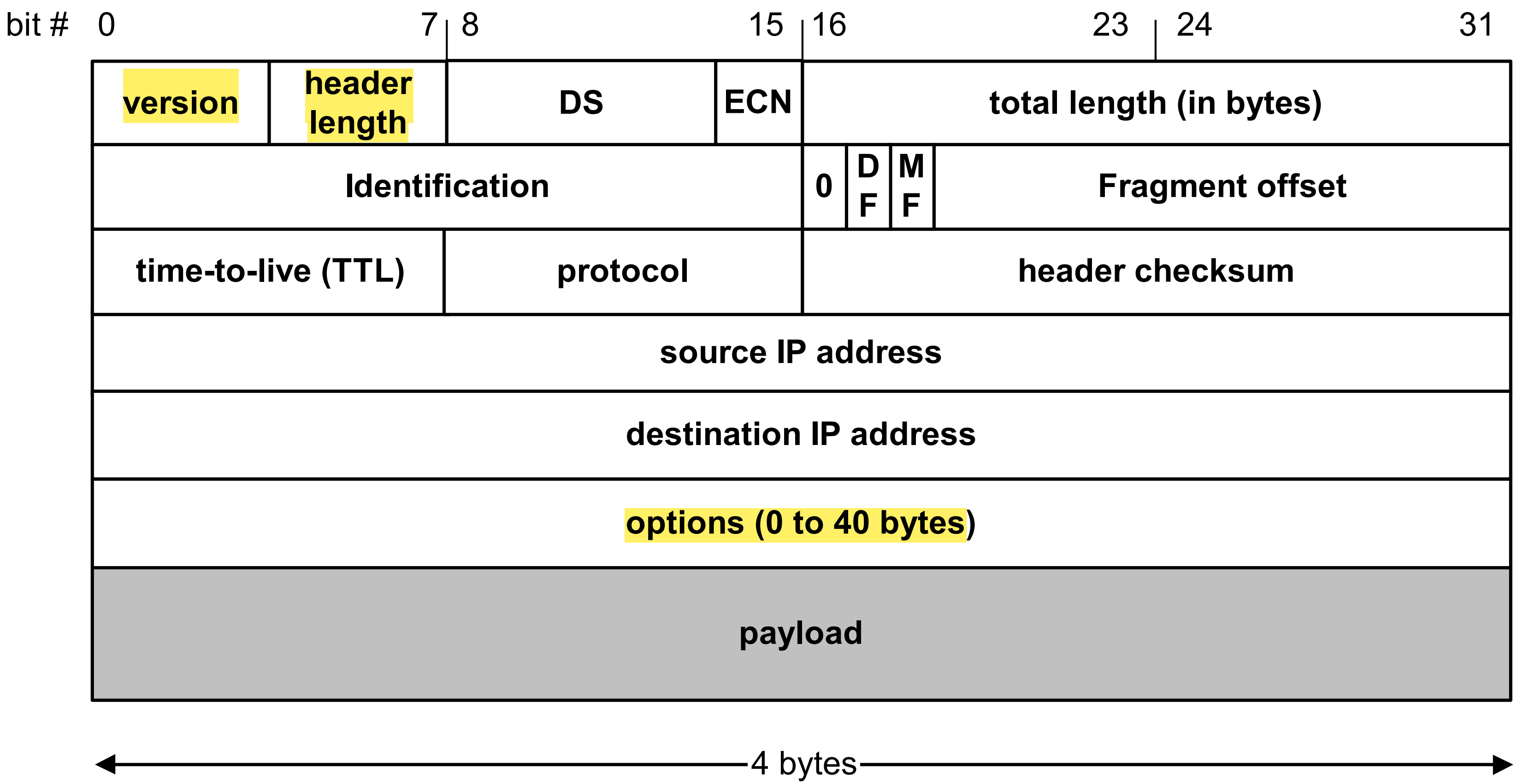
- IP supports the following services:
  - one-to-one (unicast)
  - one-to-all (broadcast)
  - one-to-several (multicast)



- IP multicast requires support of other protocols (IGMP, multicast routing)

IP HEADER

IP Header



- 20 bytes ≤ Header Size < 2^4 x 4 bytes = 64 bytes
- 20 bytes ≤ Total Length < 2^{16} bytes = 65536 bytes

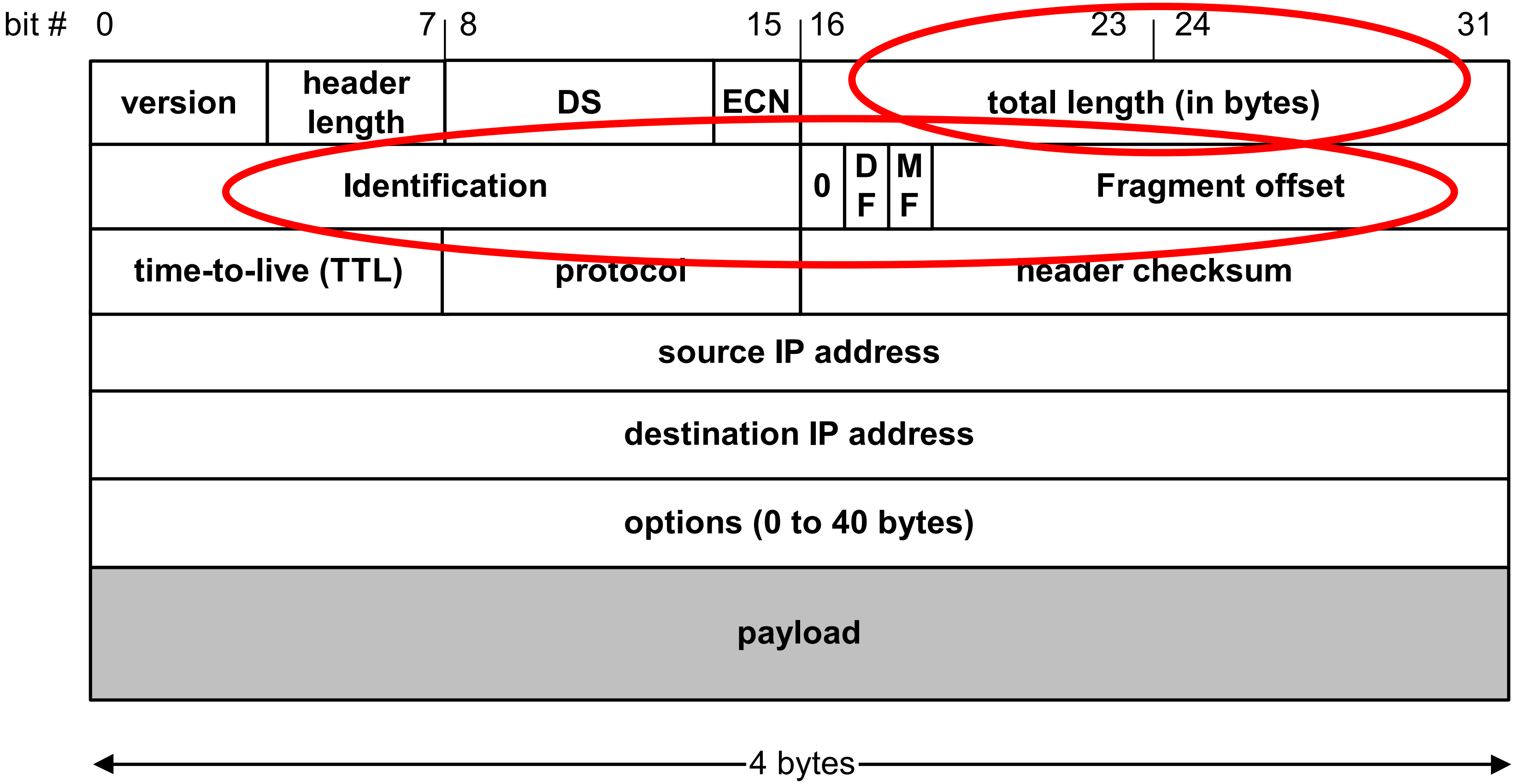
Fields of the IP Header

- Version (4 bits): Typically “4” (for IPv4), and sometimes “6” (for IPv6)

- Header length (4 bits): length of IP header, in multiples of 4 bytes, Typically “5” (for a 20-byte IPv4 header)

- DS/ECN field (1 byte)
  - This field was previously called as Type-of-Service (TOS) field. The role of this field has been re-defined, but is “backwards compatible” to TOS interpretation
  - Differentiated Service (DS) (6 bits):
    - Used to specify service level (currently not supported in the Internet)
  - Explicit Congestion Notification (ECN) (2 bits):
    - New feedback mechanism used by TCP

IP Datagram Format



Fields of the IP Header

- Total length (16 bits)
  - Number of bytes in the packet
  - Maximum size is 65,535 bytes (2^{16} -1)
  - … though underlying links may impose smaller limits
- Identification (16 bits)
  - Unique identification of a datagram from a host.
  - Incremented whenever a datagram is transmitted
- Flags (3 bits)
  - First bit always set to 0
  - DF bit (Do not fragment): Instead, they drop the packet and send back a “Too Large” ICMP control message
  - MF bit (More fragments)

Maximum Transmission Unit (MTU)

- Maximum size of IP datagram is 65535, but the data link layer protocol generally imposes a limit that is much smaller
  - Ethernet frames have a maximum payload of 1500 bytes

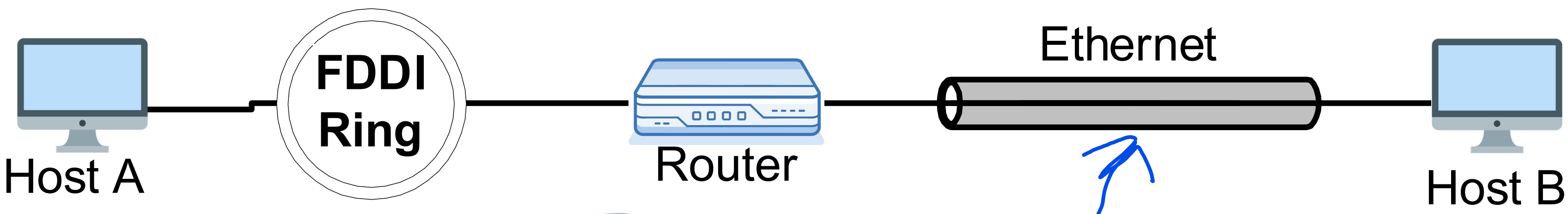
- The limit on the maximum IP datagram size, imposed by the data link protocol is called **maximum transmission unit (MTU)**

- MTUs for various data link protocols:

Ethernet:	1500	FDDI:	4352
802.3:	1492	ATM AAL5:	9180
802.5:	4464	PPP:	negotiated

Maximum Transmission Unit (MTU)

- What if the route contains networks with different MTUs?

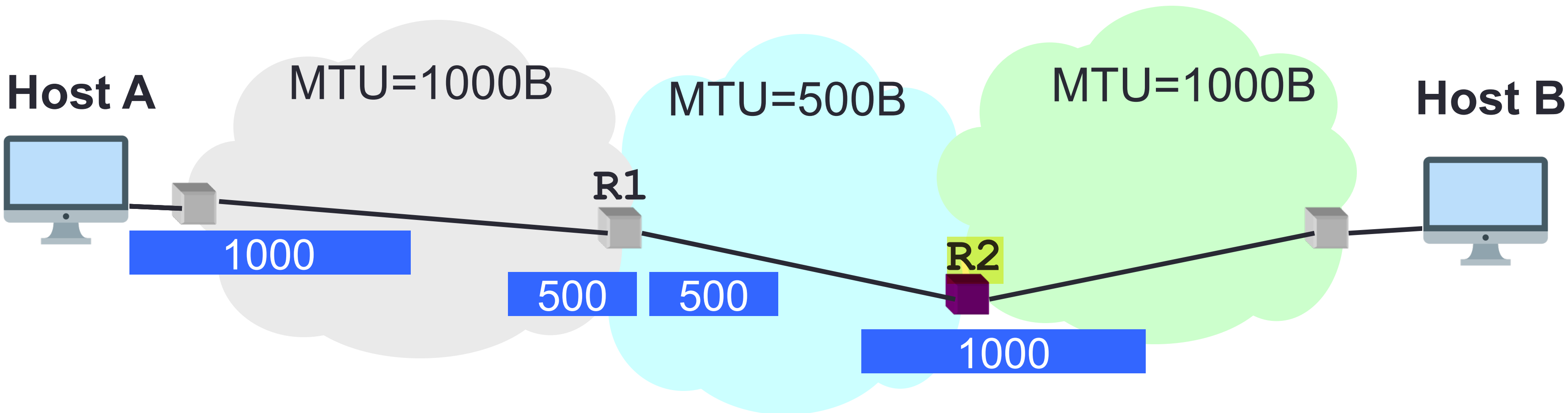


MTUs: FDDI: 4352 Ethernet: 1500

- **Fragmentation**:
  When forwarding, IP router splits the datagram into several datagram.

- Where is fragmentation done?
  - Fragmentation can be done at the sender or at intermediate routers
  - The same datagram can be fragmented several times.

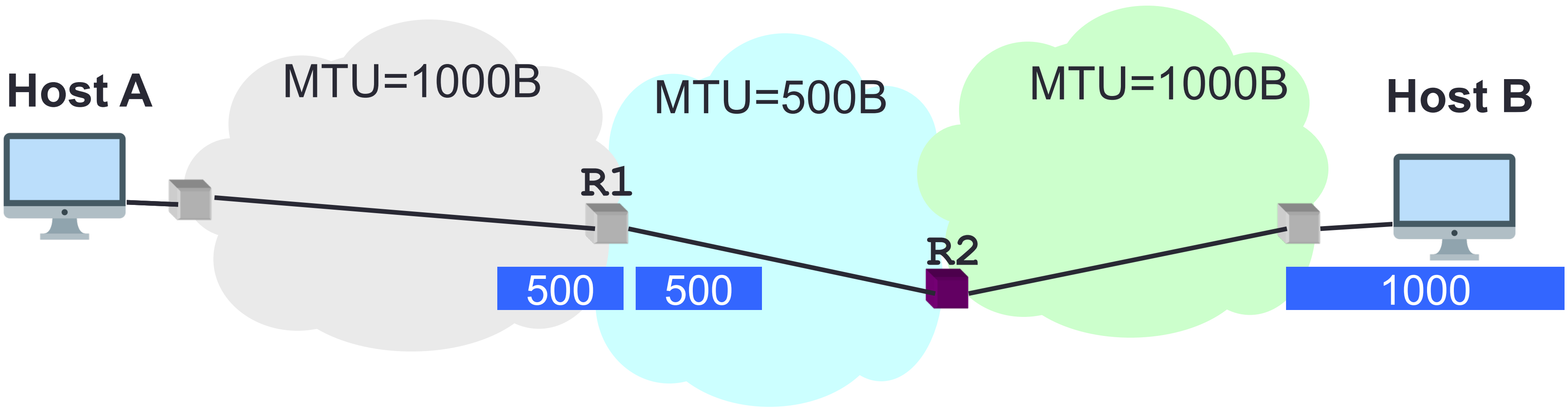
Where Should Reassembly Happen?



MTU (Maximum Transfer Unit) = Maximum packet size handled by network

- A1: router R2

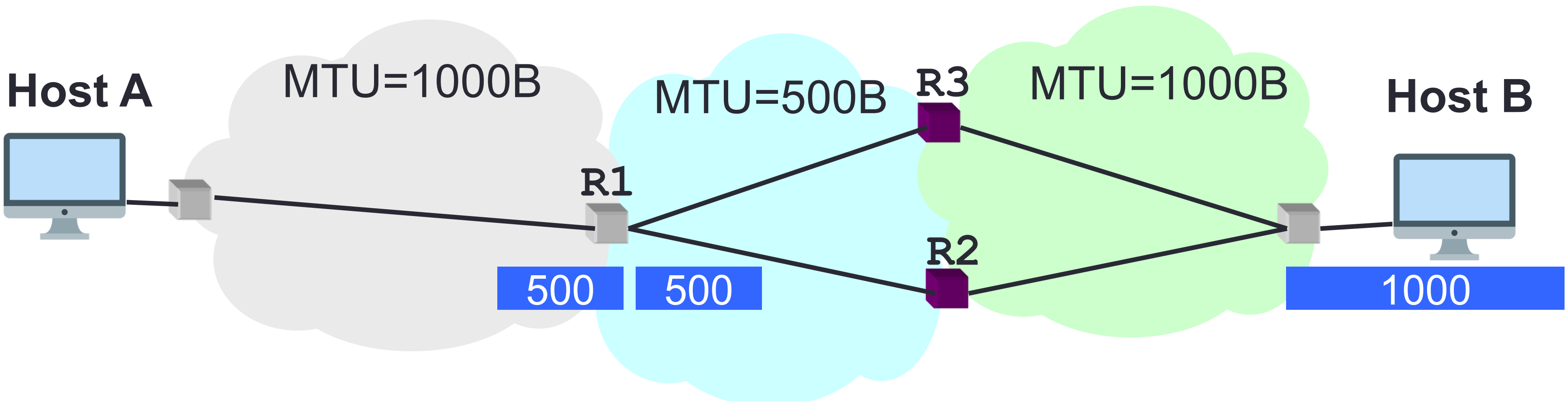
Where does Reassemble Happen?



MTU (Maximum Transfer Unit) = Maximum packet size handled by network

- A2: end-host B (receiver)

Where does Reassemble Happen?



MTU (Maximum Transfer Unit) = Maximum packet size handled by network

- A2: correct answer
  - Fragments can travel across different paths!

What's involved in Fragmentation?

- The following fields in the IP header are involved:

<table>
<tr><td>version</td><td>header length</td><td>DS</td><td>ECN</td><td colspan="4">total length (in bytes)</td></tr>
<tr><td colspan="4">Identification</td><td>0</td><td>DF</td><td>MF</td><td>Fragment offset</td></tr>
<tr><td colspan="2">time-to-live (TTL)</td><td colspan="2">protocol</td><td colspan="4">header checksum</td></tr>
</table>

Identification

When a datagram is fragmented, the identification is the same in all fragments

Flags DF bit is set: Datagram cannot be fragmented and must be discarded if MTU is too small

MF bit set: This datagram is part of a fragment and an additional fragment follows this one

Fragment offset Offset of the payload of the current fragment in the original datagram

Total length Total length of the current fragment

Example of Fragmentation

- Suppose we have a 4,000 byte datagram sent from host 1.2.3.4 to host 3.4.5.6 …

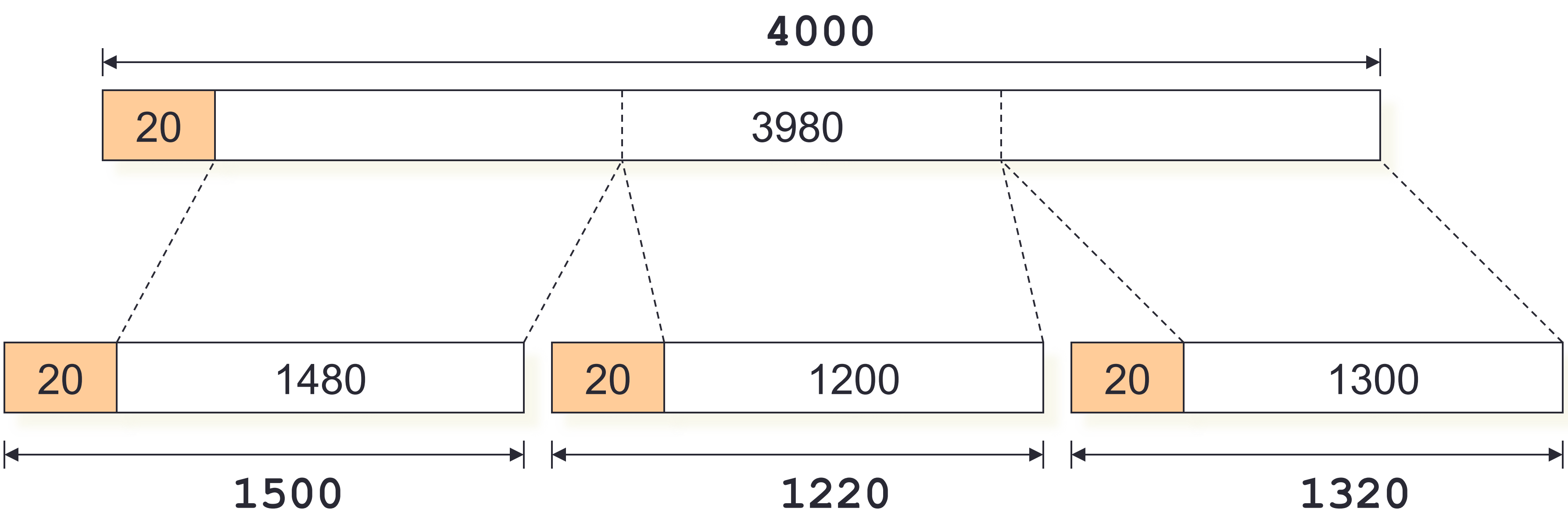
<table>
<tr><td>Version
4</td><td>Header Length
5</td><td>Type of Service
0</td><td colspan="2">Total Length: 4000</td></tr>
<tr><td colspan="3">Identification: 56273</td><td>R/D/M
0/0/0</td><td>Fragment Offset: 0</td></tr>
<tr><td colspan="2">TTL
127</td><td>Protocol
6</td><td colspan="2">Checksum: 44019</td></tr>
<tr><td colspan="5">Source Address: 1.2.3.4</td></tr>
<tr><td colspan="5">Destination Address: 3.4.5.6</td></tr>
</table>

(3980 more bytes here)

- … and it traverses a link that limits datagrams to 1,500 bytes

Example of Fragmentation

- Datagram split into 3 pieces



Example of Fragmentation

- Possible first piece:

<table>
<tr><td>Version
4</td><td>Header Length
5</td><td>Type of Service
0</td><td colspan="2">Total Length: 1500</td></tr>
<tr><td colspan="3">Identification: 56273</td><td>R/D/M
0/0/1</td><td>Fragment Offset: 0</td></tr>
<tr><td colspan="2">TTL
127</td><td>Protocol
6</td><td colspan="2">Checksum: xxx</td></tr>
<tr><td colspan="5">Source Address: 1.2.3.4</td></tr>
<tr><td colspan="5">Destination Address: 3.4.5.6</td></tr>
</table>

Example of Fragmentation

- Possible second piece:

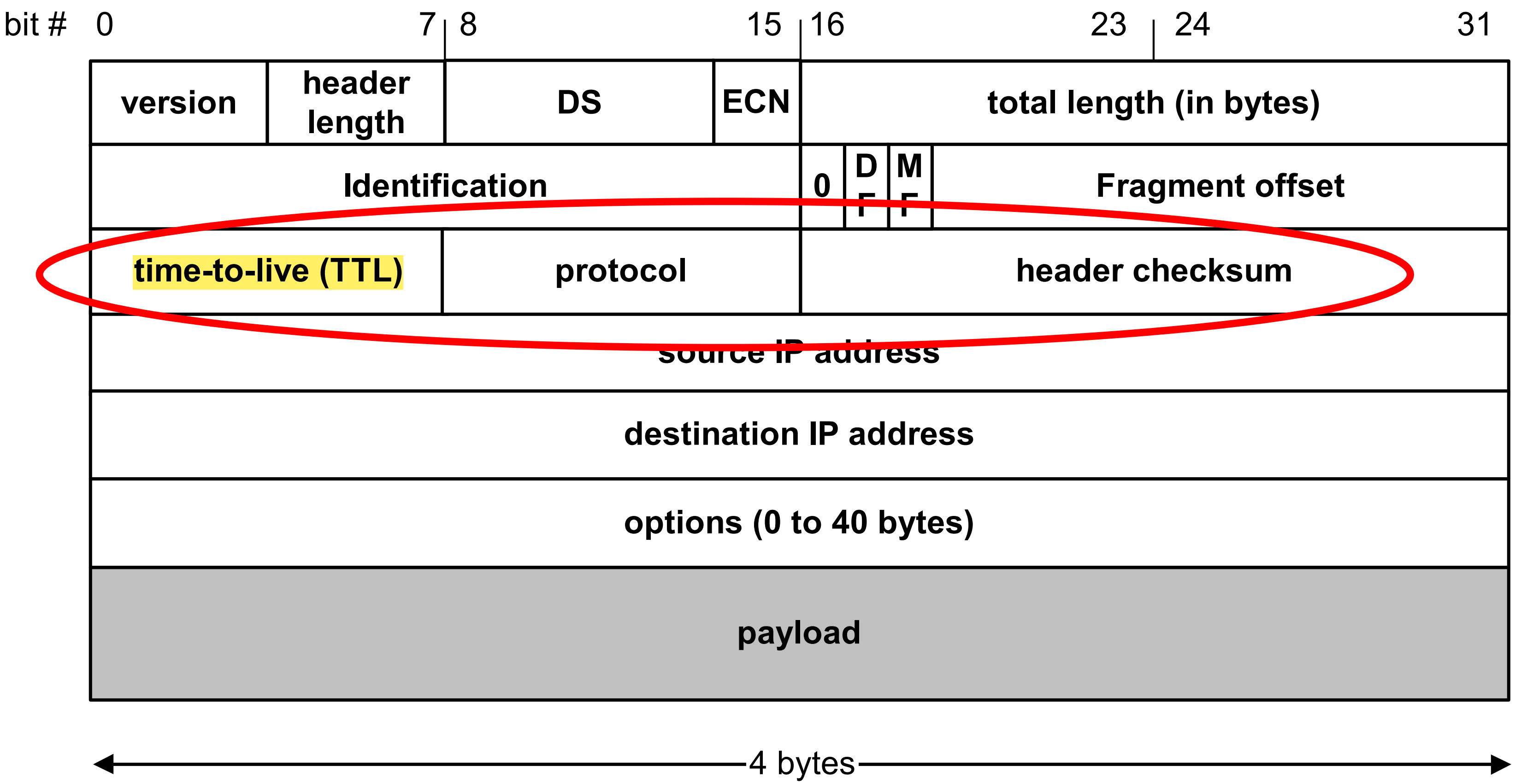
<table>
<tr><td>Version
4</td><td>Header Length
5</td><td colspan="2">Type of Service
0</td><td colspan="2">Total Length: 1220</td></tr>
<tr><td colspan="3">Identification: 56273</td><td colspan="2">R/D/M
0/0/1</td><td>Fragment Offset: 185
(185 * 8 = 1480)</td></tr>
<tr><td colspan="2">TTL
127</td><td colspan="2">Protocol
6</td><td colspan="2">Checksum: yyy</td></tr>
<tr><td colspan="6">Source Address: 1.2.3.4</td></tr>
<tr><td colspan="6">Destination Address: 3.4.5.6</td></tr>
</table>

Example of Fragmentation

- Possible third piece:

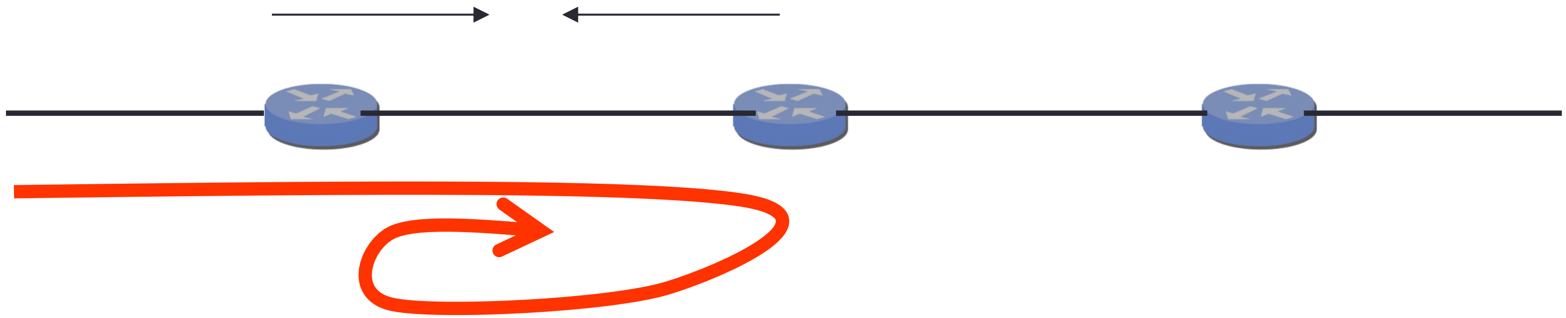
Version 4	Header Length 5	Type of Service 0	Total Length: 1320	
Identification: 56273		R/D/M 0/0/0	Fragment Offset: 335 (335 * 8 = 2680)	
TTL 127	Protocol 6	Checksum: zzz		
Source Address: 1.2.3.4				
Destination Address: 3.4.5.6				

IP Datagram Format



Time-to-Live (TTL) Field (8 bits)

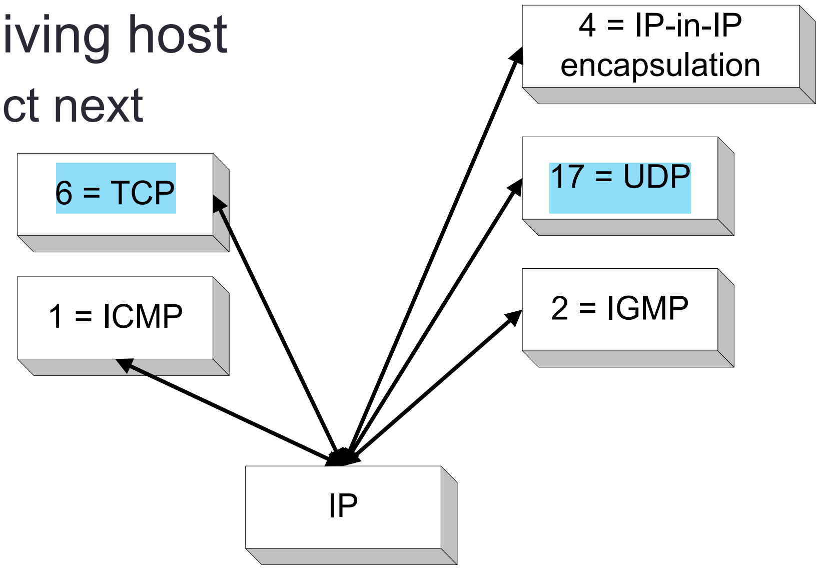
- Potentially lethal problem
  - Forwarding loops can cause packets to cycle forever
  - As these accumulate, eventually consume **all** capacity



- Time-to-live field in packet header
  - Sender sets the value (e.g., 64)
  - Each router decrements the value by 1
  - When the value reaches 0, the datagram is dropped
  - …and "time exceeded" message is sent to the source
    - Using "ICMP" control message; basis for **traceroute**

IP Packet Header Fields

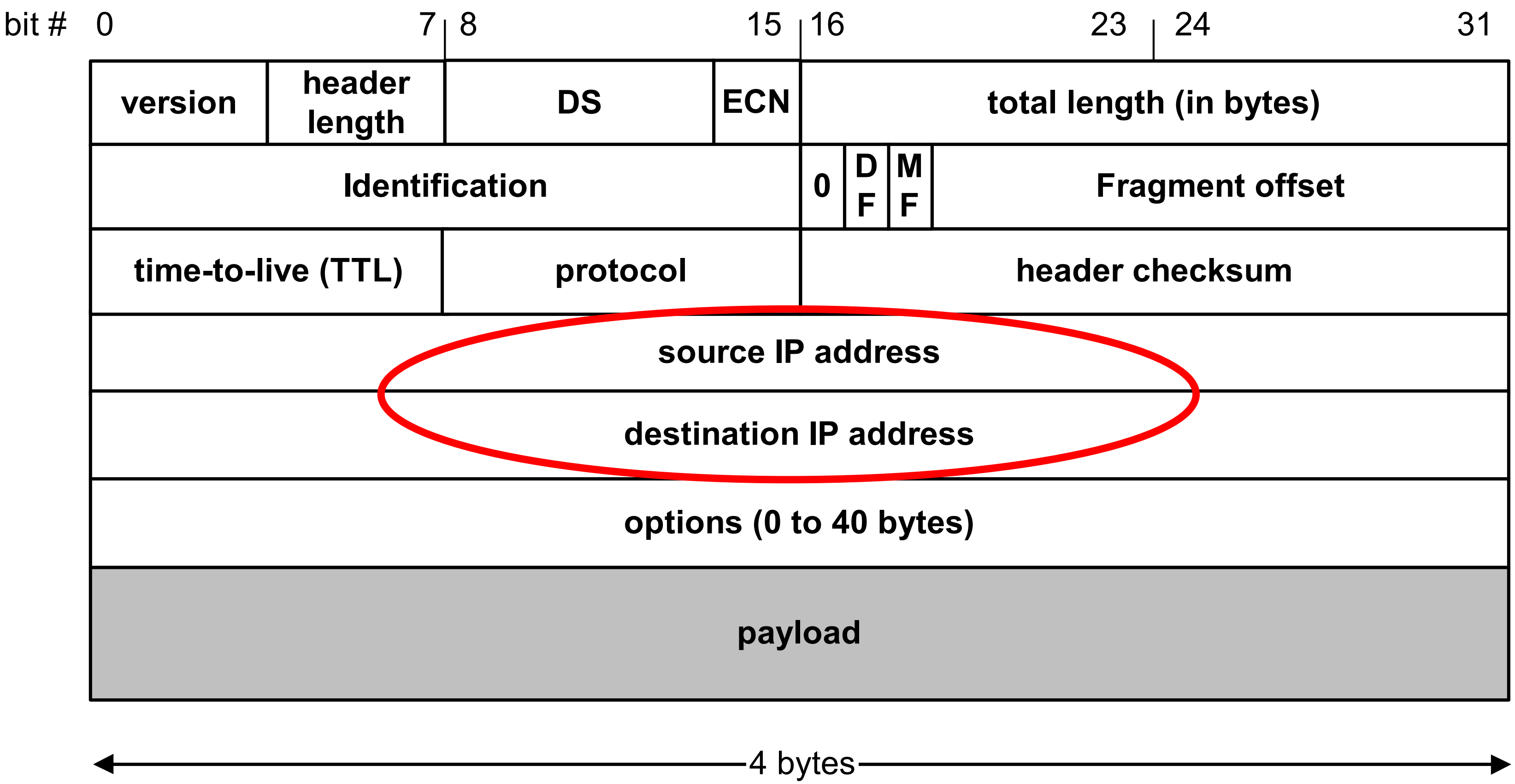
- Protocol (8 bits)
  - Specifies the higher-layer protocol.
  - Important for de-multiplexing at receiving host
    - Indicates what kind of header to expect next



- Checksum (16 bits)
  - Complement of the *ones-complement* sum of all 16-bit words in the IP **packet header**
  - Checksum recalculated at every router

  -> If not correct, router discards packet

IP Datagram Format



Fields of the IP Header

- Two IP addresses
  - Source IP address (32 bits)
  - Destination IP address (32 bits)

- Destination address
  - Unique identifier/locator for the receiving host
  - Allows each node to make forwarding decisions

- Source address
  - Unique identifier/locator for the sending host
  - Recipient can decide whether to accept packet
  - Enables recipient to send a reply back to source

Fields of the IP Header

- **Options:**
  - Security restrictions
  - Record Route: each router that processes the packet adds its IP address to the header.
  - Timestamp: each router that processes the packet adds its IP address and time to the header.
  - (loose) Source Routing: specifies a list of routers that must be traversed.
  - (strict) Source Routing: specifies a list of the only routers that can be traversed.
- **Padding:** Padding bytes are added to ensure that header ends on a 4-byte boundary

IPV6

IPv6 - IP Version 6

- **IP Version 6**
  - Is the successor to the currently used IPv4
  - Specification completed in 1994
  - Makes improvements to IPv4 (no revolutionary changes)

- One (not the only!) feature of IPv6 is a significant increase in of the IP address to **128 bits (16 bytes)**
  - IPv6 will solve – for the foreseeable future – the problems with IP addressing
  - 10^{24} addresses per square inch on the surface of the Earth.

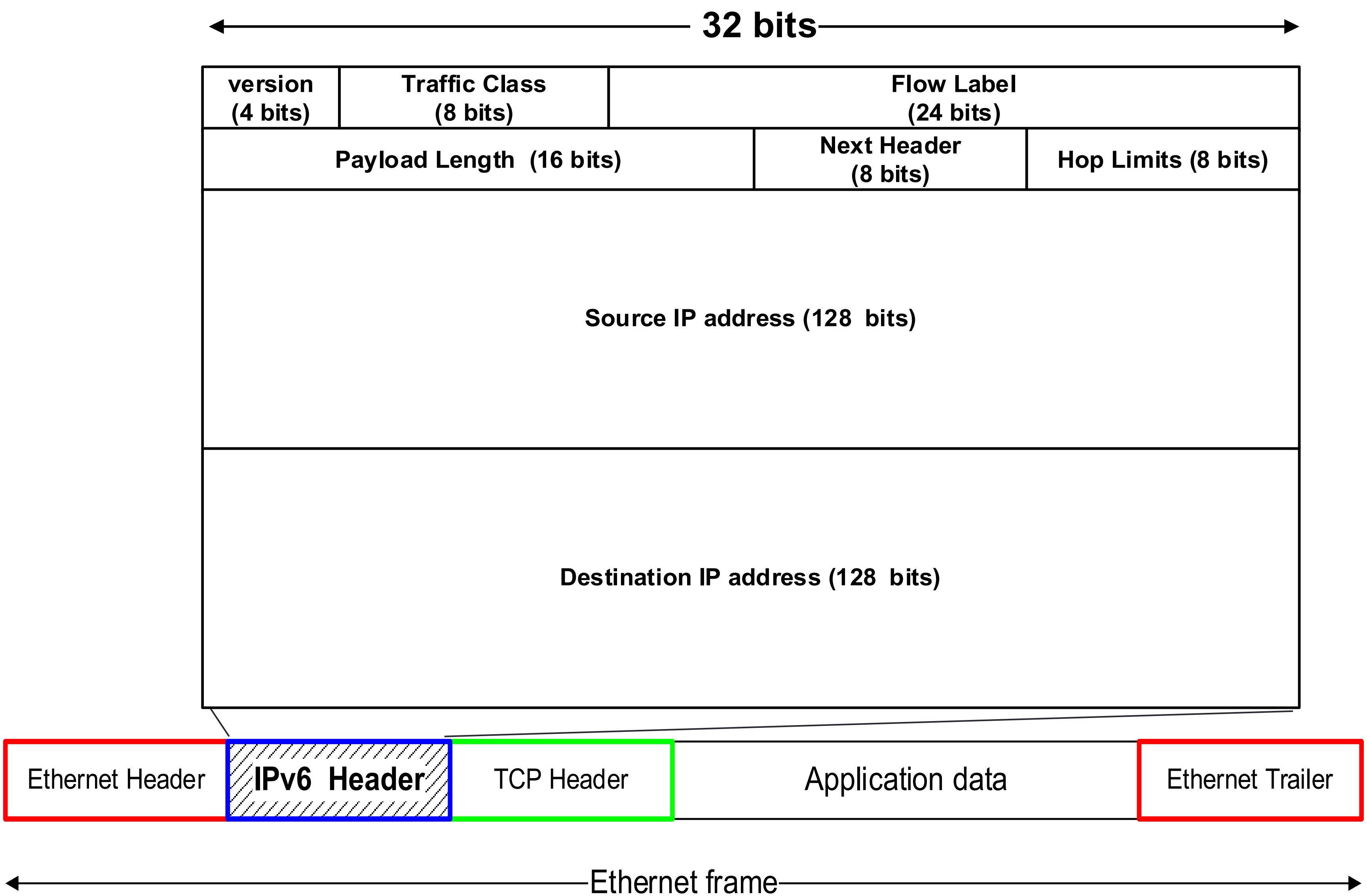
- **IPv4** has a maximum of

  $2^{32} \approx 4$ billion addresses

- **IPv6** has a maximum of

  $2^{128} = (2^{32})^4 \approx 4$ billion x 4 billion x 4 billion x 4 billion addresses

IPv6 Header



32 bits
version
(4 bits)
Traffic Class
(8 bits)
Flow Label
(24 bits)
Payload Length (16 bits)
Next Header
(8 bits)
Hop Limits (8 bits)
Source IP address (128 bits)
Destination IP address (128 bits)
Ethernet Header
IPv6 Header
TCP Header
Application data
Ethernet Trailer
Ethernet frame

Notation of IPv6 addresses

- **Convention**: The 128-bit IPv6 address is written as **eight 16-bit integers** (using hexadecimal)

CEDF:BP76:3245:4464:FACE:2E50:3025:DF12

- **Short notation:**
- Abbreviations of leading zeroes:

CEDF:BP76:0000:0000:009E:0000:3025:DF12
→ CEDF:BP76:0:0:9E :0:3025:DF12

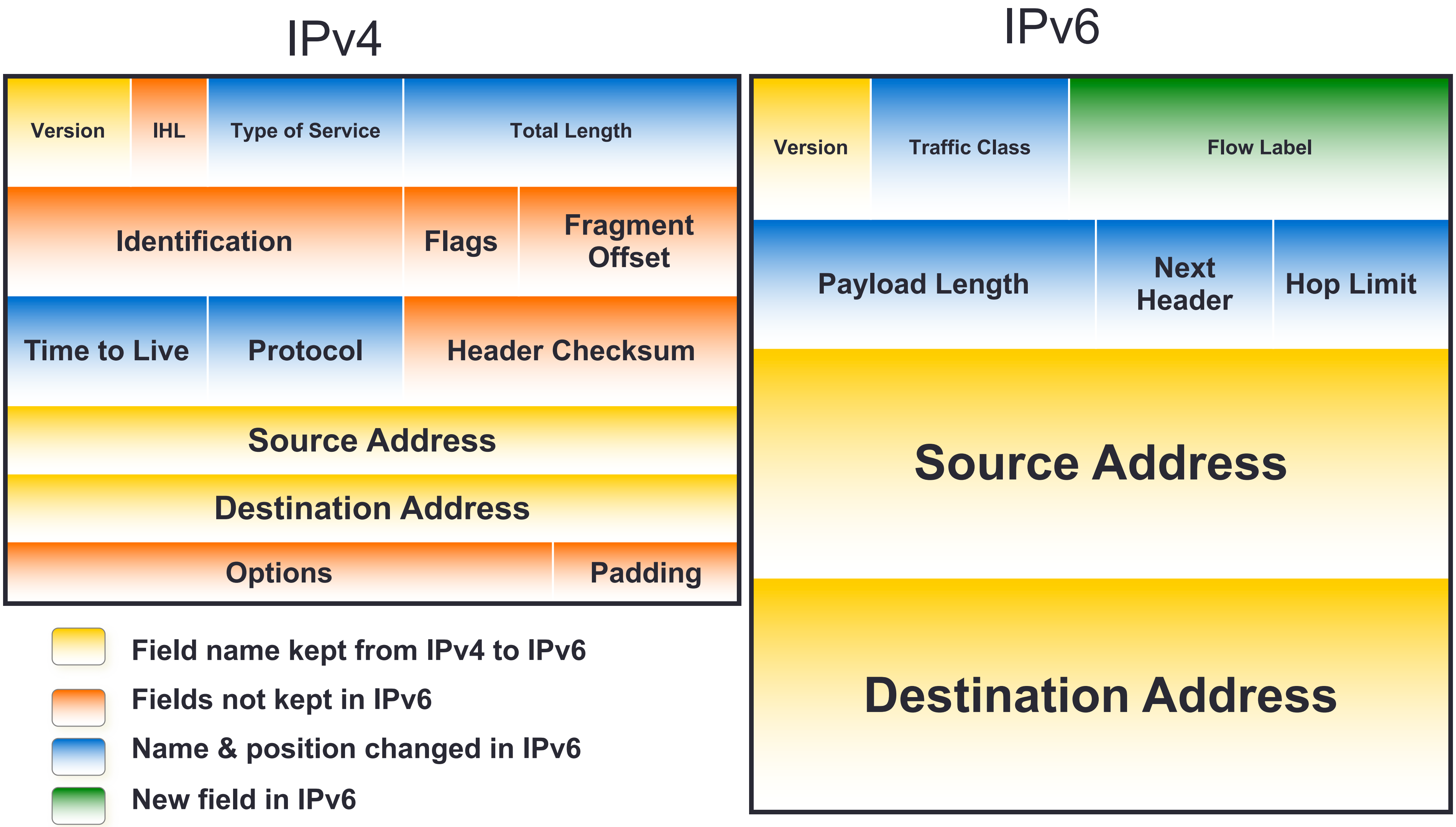
- “:0000:0000:0000” can be written as “::”

CEDF:BP76:0:0:FACE:0:3025:DF12 → CEDF:BP76::FACE:0:3025:DF12

- IPv6 addresses derived from IPv4 addresses have 96 leading zero bits. Convention allows to use IPv4 notation for the last 32 bits.

::80:8F:89:90 → ::128.143.137.144

IPv4 vs IPv6


IPv4
Version
IHL
Type of Service
Total Length
Identification
Flags
Fragment Offset
Time to Live
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding
IPv6
Version
Traffic Class
Flow Label
Payload Length
Next Header
Hop Limit
Source Address
Destination Address
Field name kept from IPv4 to IPv6
Fields not kept in IPv6
Name & position changed in IPv6
New field in IPv6

Attacks involving IP

- Primarily on the operation of options, or by exploiting bugs in specialized code (fragment reassembly).
- Trying to get a router to crash or perform poorly by modifying one or more of the IP header fields (e.g., bad header length or version number).
- IP spoofing -> ingress filtering (check IP prefix)