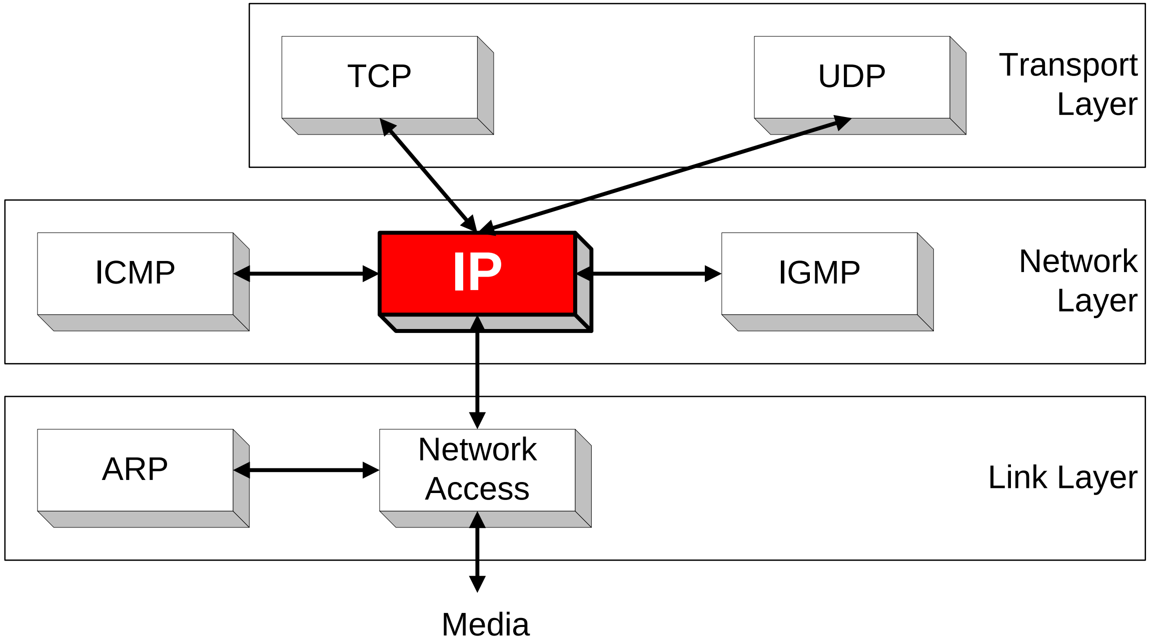
IP - The Internet Protocol

Relates to Lab 2.

A module on the Internet Protocol.

Orientation

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



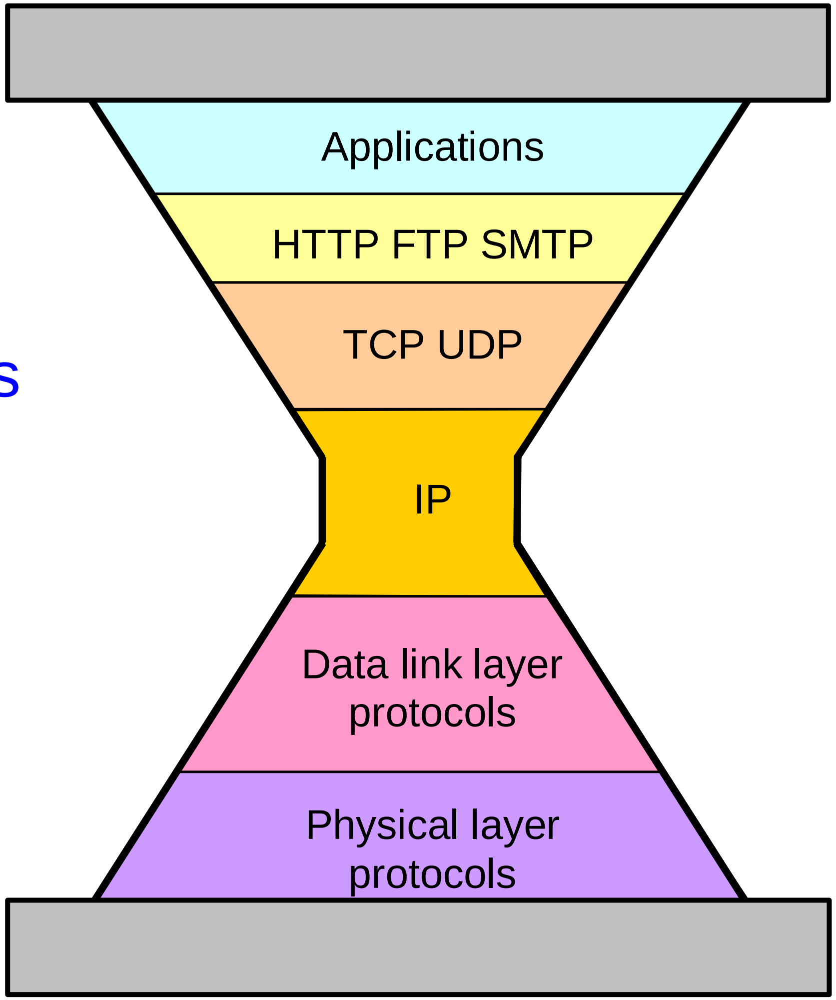
- IP's current version is Version 4 (IPv4). It is specified in RFC 891.

IP: The waist of the hourglass

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

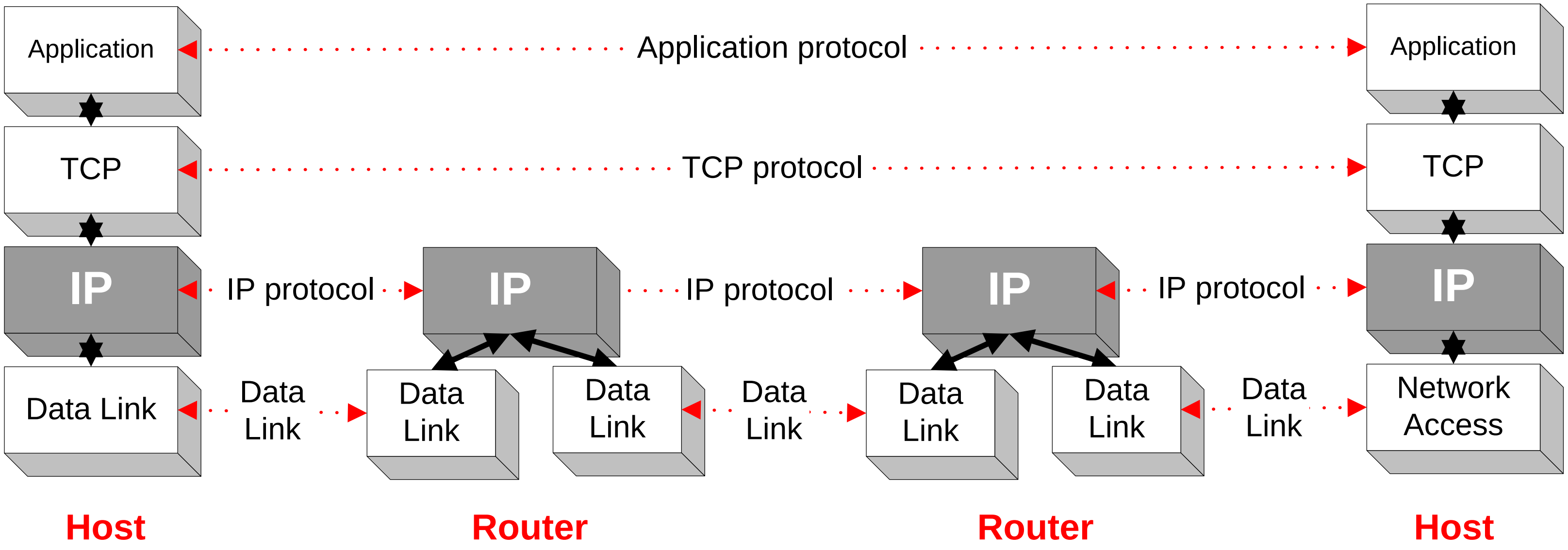
- Multiple higher-layer protocols
- Multiple lower-layer protocols

- Only one protocol at the network layer.



Application protocol

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



IP Service

- Delivery service of IP is minimal

- IP provide provides an unreliable connectionless best effort service (also called: “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,…)

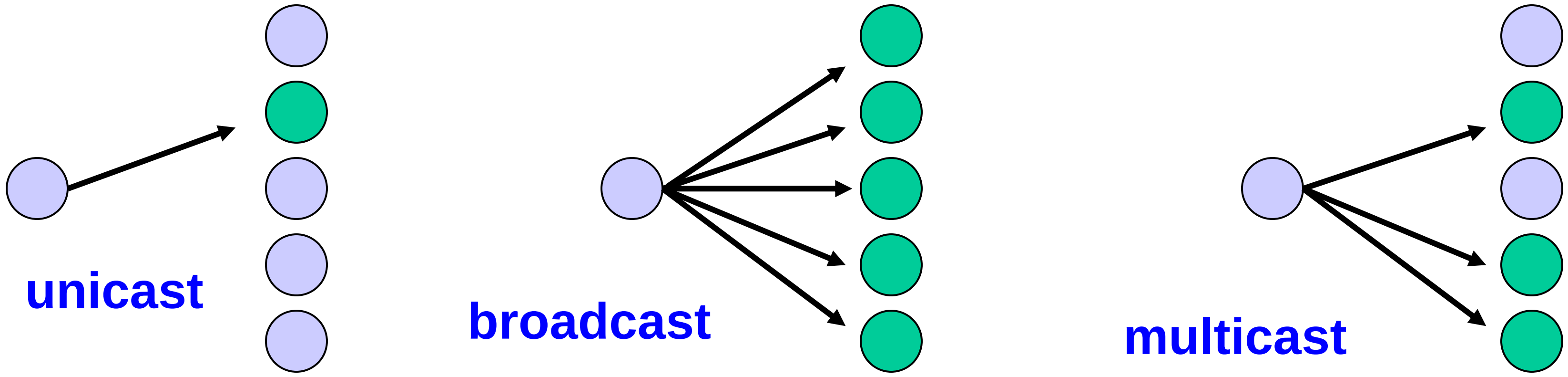
- Consequences:

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

  - Packets may be delivered out-of-sequence

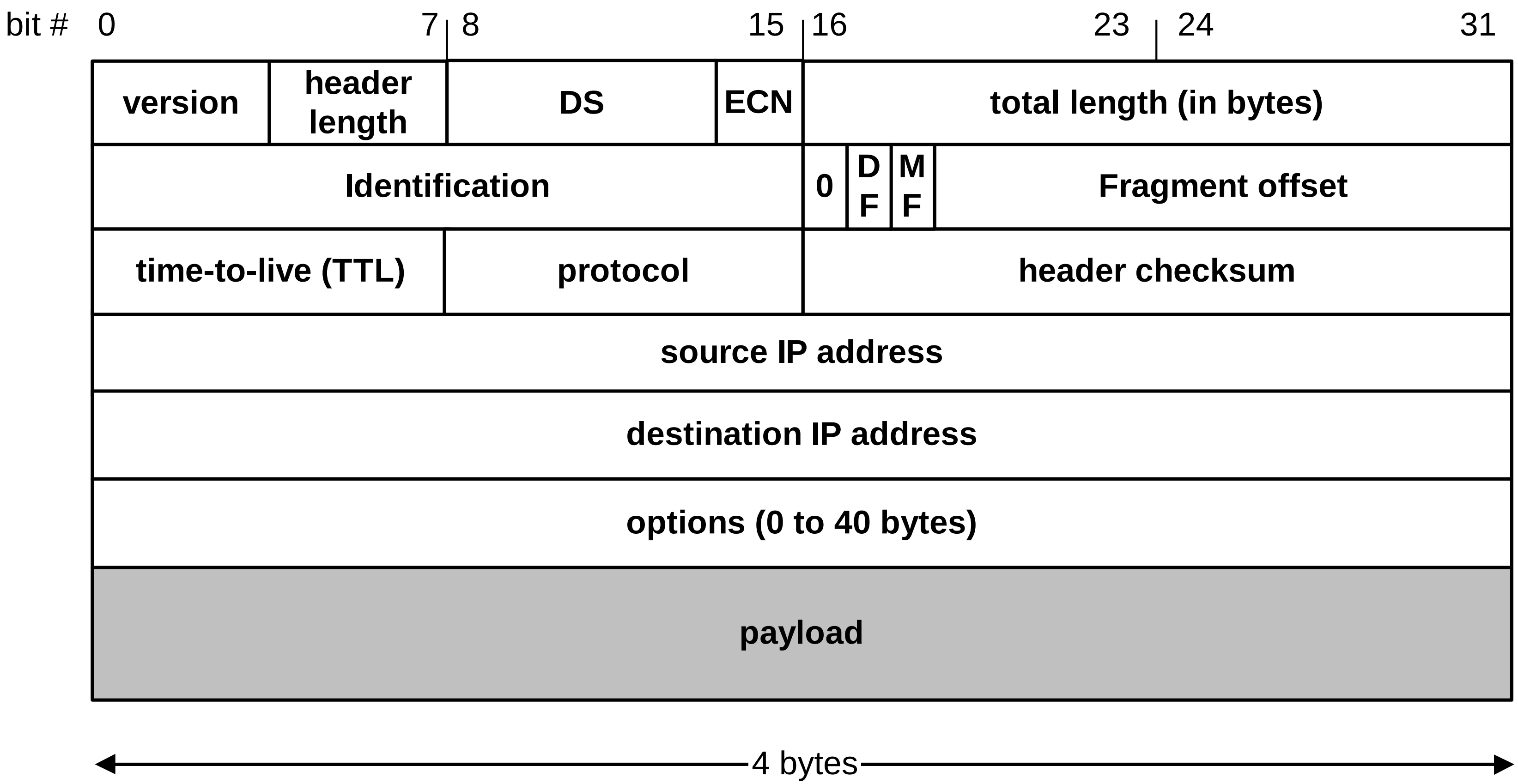
IP Service

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



- IP multicast also supports a many-to-many service.
- IP multicast requires support of other protocols (IGMP, multicast routing)

IP Datagram Format



- 20 bytes ≤ Header Size < 2^4 x 4 bytes = 60 bytes
- 20 bytes ≤ Total Length < 2^{16} bytes = 65536 bytes

IP Datagram Format

- **Question:** In which order are the bytes of an IP datagram transmitted?
- **Answer:**
    - Transmission is row by row
    - For each row:
        1. First transmit bits 0-7
        2. Then transmit bits 8-15
        3. Then transmit bits 16-23
        4. Then transmit bits 24-31
- This is called **network byte** order or **big endian** byte ordering.

- **Note:** Many computers (incl. Intel processors) store 32-bit words in little endian format. Others (incl. Motorola processors) use big endian.

Big endian vs. small endian

- Conventions to store a multibyte work
- Example: a 4 byte Long Integer **`Byte3 Byte2 Byte1 Byte0`**

Little Endian

- Stores the low-order byte at the lowest address and the highest order byte in the highest address.

```
Base Address+0 Byte0
Base Address+1 Byte1
Base Address+2 Byte2
Base Address+3 Byte3
```

- Intel processors use this order

Big Endian

- Stores the high-order byte at the lowest address, and the low-order byte at the highest address.

```
Base Address+0 Byte3
Base Address+1 Byte2
Base Address+2 Byte1
Base Address+3 Byte0
```

Motorola processors use big endian.

Fields of the IP Header

- **Version (4 bits)**: current version is 4, next version will be 6.
- **Header length (4 bits)**: length of IP header, in multiples of 4 bytes
- **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

Fields of the IP Header

- **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)
  - MF bit (More fragments)

  Will be explained later→ Fragmentation

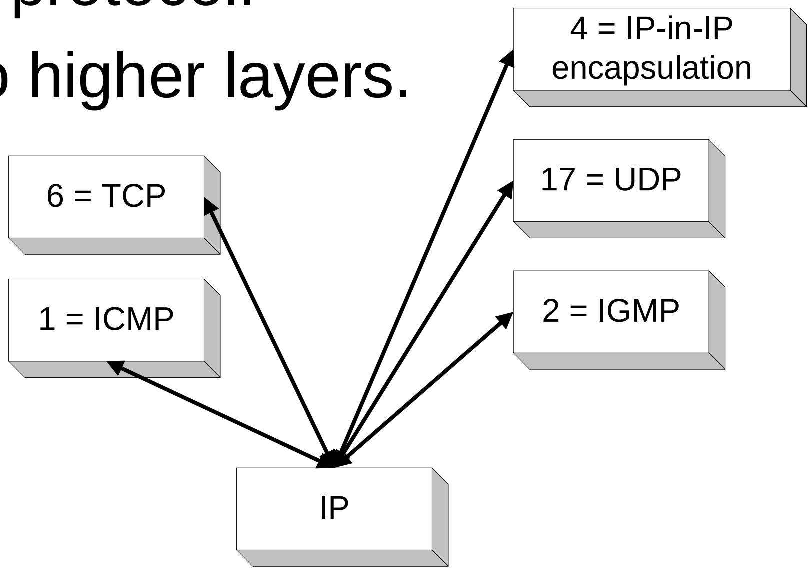
Fields of the IP Header

- **Time To Live (TTL) (1 byte):**
  - Specifies longest paths before datagram is dropped
  - Role of TTL field: Ensure that packet is eventually dropped when a routing loop occurs

  Used as follows:
  - Sender sets the value (e.g., 64)
  - Each router decrements the value by 1
  - When the value reaches 0, the datagram is dropped

Fields of the IP Header

- **Protocol (1 byte):**
  - Specifies the higher-layer protocol.
  - Used for demultiplexing to higher layers.



- **Header checksum (2 bytes):** A simple 16-bit long checksum which is computed for the header of the datagram.

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

Maximum Transmission Unit

- Maximum size of IP datagram is 65535, but the data link layer protocol generally imposes a limit that is much smaller

- Example:
  - Ethernet frames have a maximum payload of 1500 bytes

    → IP datagrams encapsulated in Ethernet frame cannot be longer than 1500 bytes

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

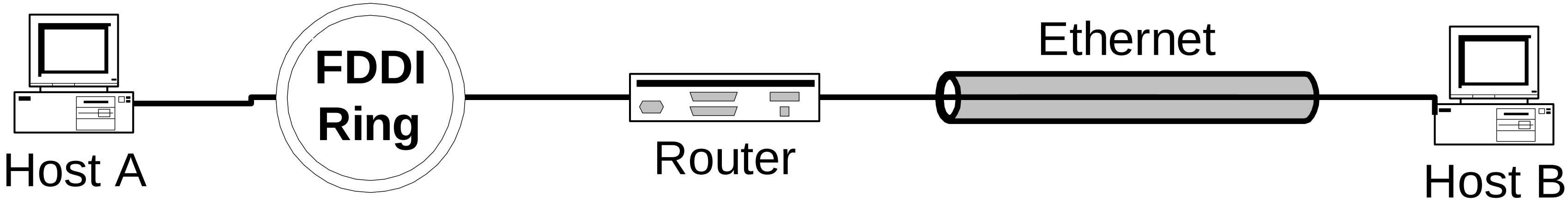
- MTUs for various data link protocols:

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

IP Fragmentation

- What if the size of an IP datagram exceeds the MTU?
  IP datagram is fragmented into smaller units.

- What if the route contains networks with different MTUs?

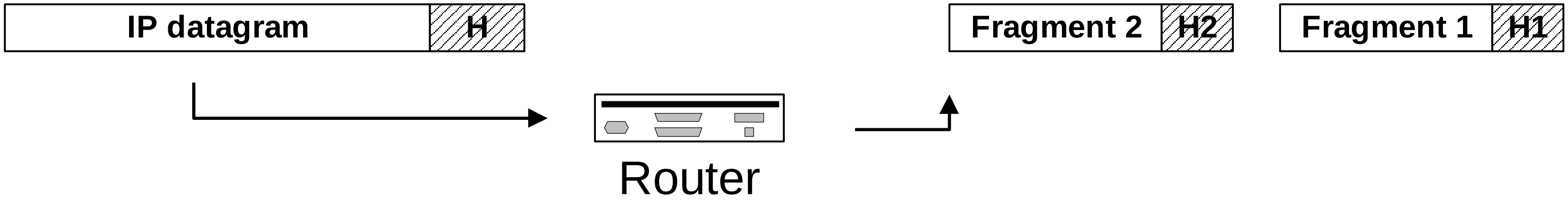


MTUs: FDDI: 4352 Ethernet: 1500

- **Fragmentation**:
  - IP router splits the datagram into several datagram
  - Fragments are reassembled at receiver

Where is Fragmentation done?

- Fragmentation can be done at the sender or at intermediate routers
- The same datagram can be fragmented several times.
- Reassembly of original datagram is only done at destination hosts !!



What's involved in Fragmentation?

- The following fields in the IP header are involved:

version	header length	DS	ECN	total length (in bytes)			
Identification				0	DF	MF	Fragment offset
time-to-live (TTL)		protocol		header checksum			

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

What’s involved in Fragmentation?

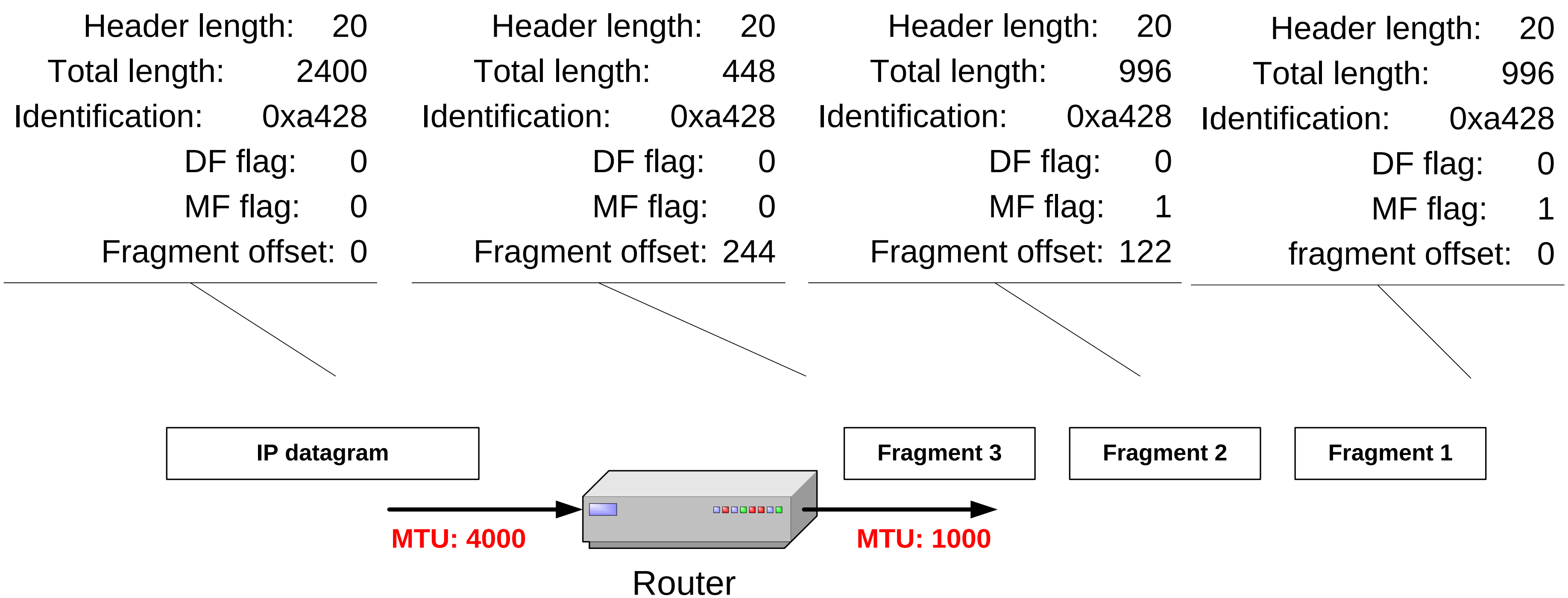
- The following fields in the IP header are involved:

<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>

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

- A datagram with size 2400 bytes must be fragmented according to an MTU limit of 1000 bytes



Determining the length of fragments

- To determine the size of the fragments we recall that, since there are only 13 bits available for the fragment offset, the offset is given as a multiple of eight bytes. As a result, the first and second fragment have a size of 996 bytes (and not 1000 bytes). This number is chosen since 976 is the largest number smaller than 1000–20= 980 that is divisible by eight. The payload for the first and second fragments is 976 bytes long, with bytes 0 through 975 of the original IP payload in the first fragment, and bytes 976 through 1951 in the second fragment. The payload of the third fragment has the remaining 428 bytes, from byte 1952 through 2379. With these considerations, we can determine the values of the fragment offset, which are 0, 976 / 8 = 122, and 1952 / 8 = 244, respectively, for the first, second and third fragment.