TA: Xifan Zheng
Email: zhengxifan0403@gmail.com

Welcome to CPSC 441!

Today's Tutorial

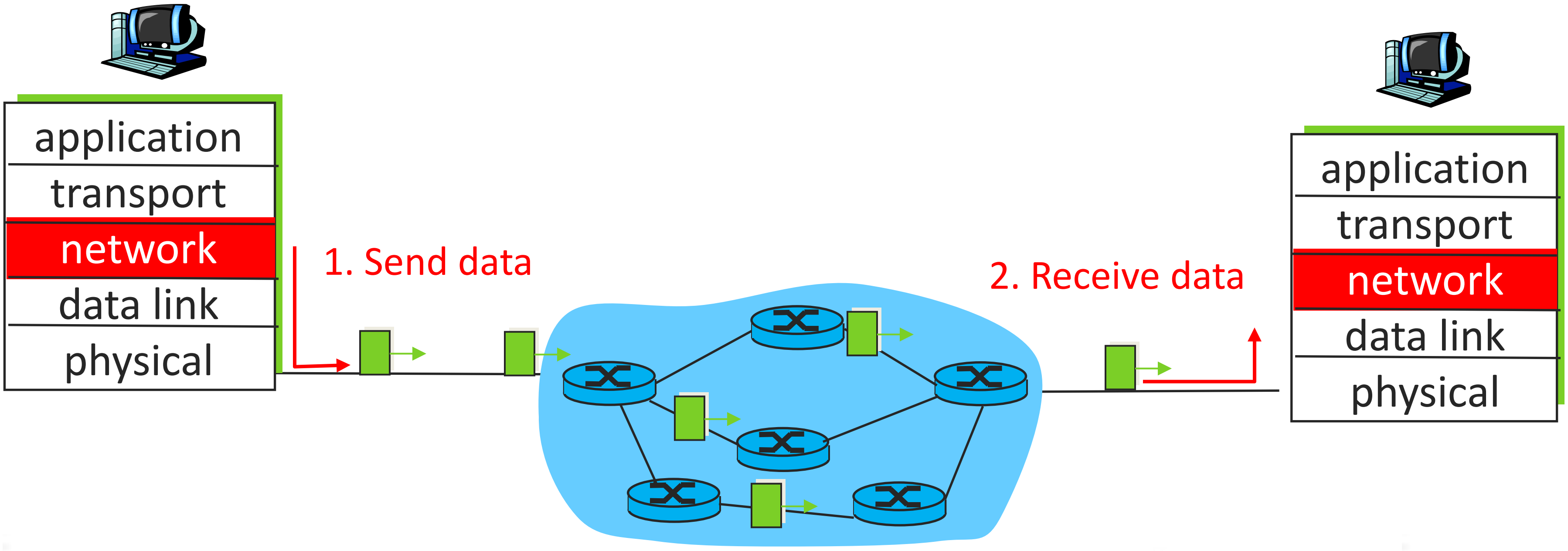
- **Introduction to IP protocol**
- **Big endian vs. small endian**
- **IP header**
- **Fragmentation**

Welcome to CPSC 441

This slide is taken from Ruiting Zhou and modified by Xifan Zheng

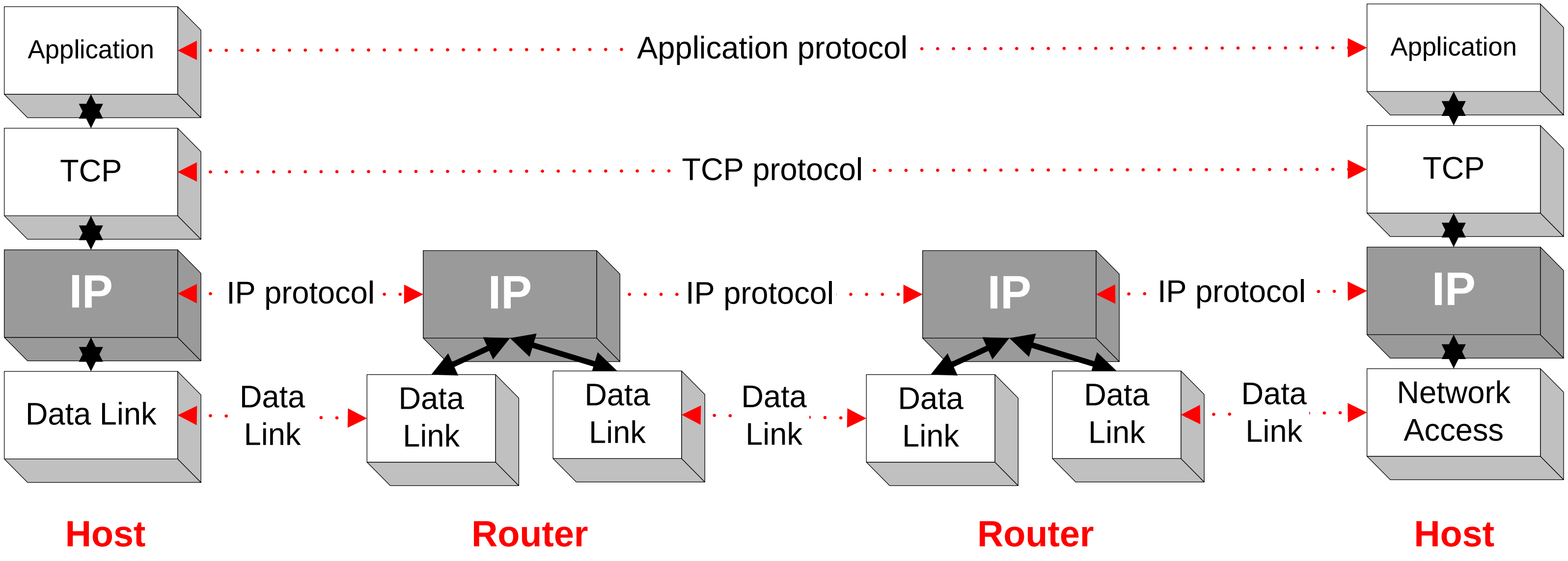
The Network Layer

- IP (Internet Protocol) is a Network Layer Protocol.
- RFC 791 provides the specification for IP.



Highest layer in routers

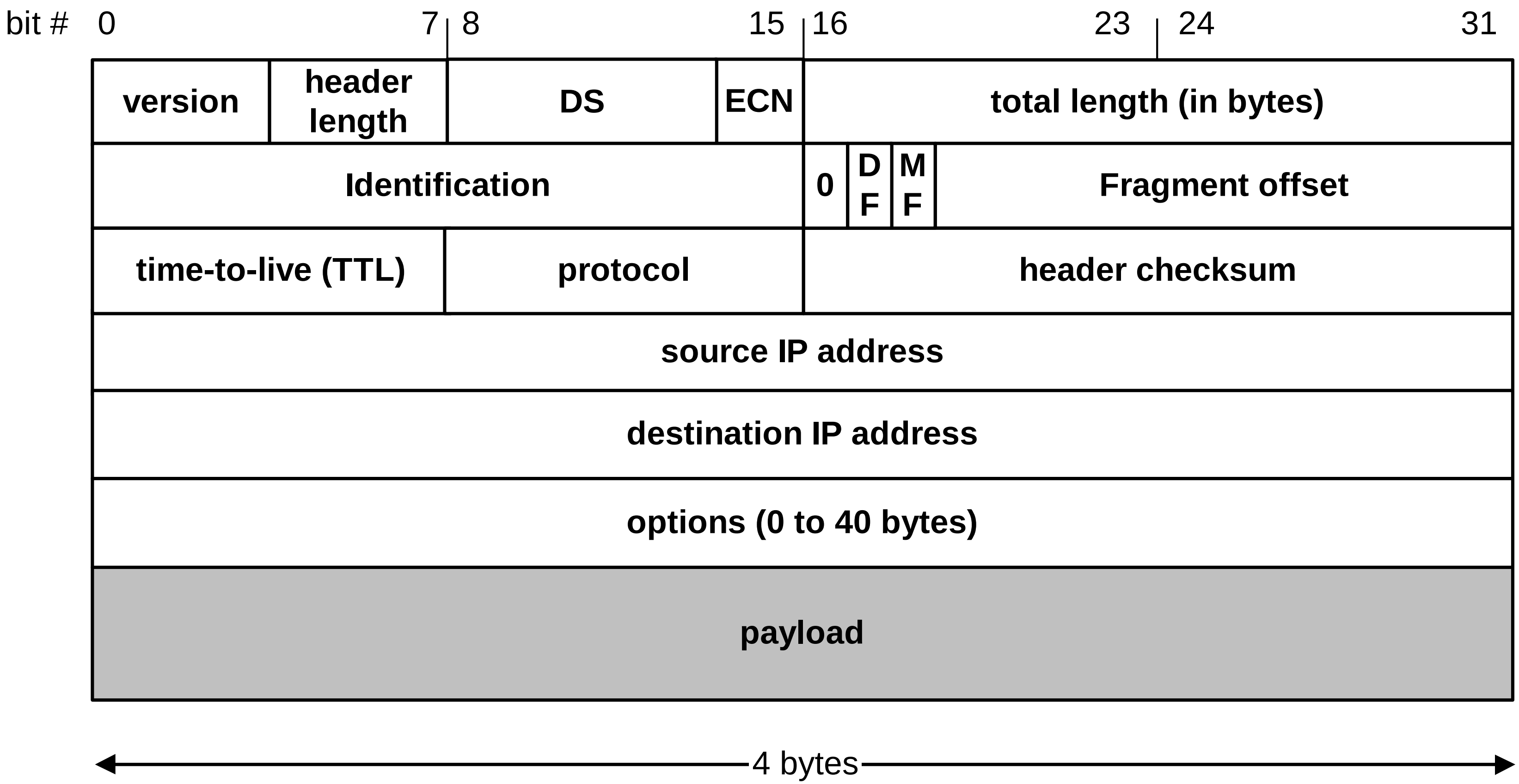
- IP is the highest layer protocol which is implemented at both routers and hosts



Best effort protocol

- IP provides an unreliable connectionless best effort service (also called: “datagram service”).
  - **Unreliable:** no guarantee for delivery of 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, etc.)
- Consequences: Higher layer protocols have to take care of delivery guarantees.

IP Datagram



- Header Size: at least 20 bytes and at most 60 bytes (with options)
- Total Length: at most 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.

Big endian vs. small endian

- Conventions to store a multibyte work
- Example: a 4 byte Long Integer **`0x1234abcd`**

Little Endian (Host Byte Order)

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

```
0x00 0xcd
0x01 0xab
0x02 0x34
0x03 0x12
```

- Intel processors use this order

Big Endian (Network Byte Order)

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

```
0x00 0x12
0x01 0x34
0x02 0xab
0x03 0xcd
```

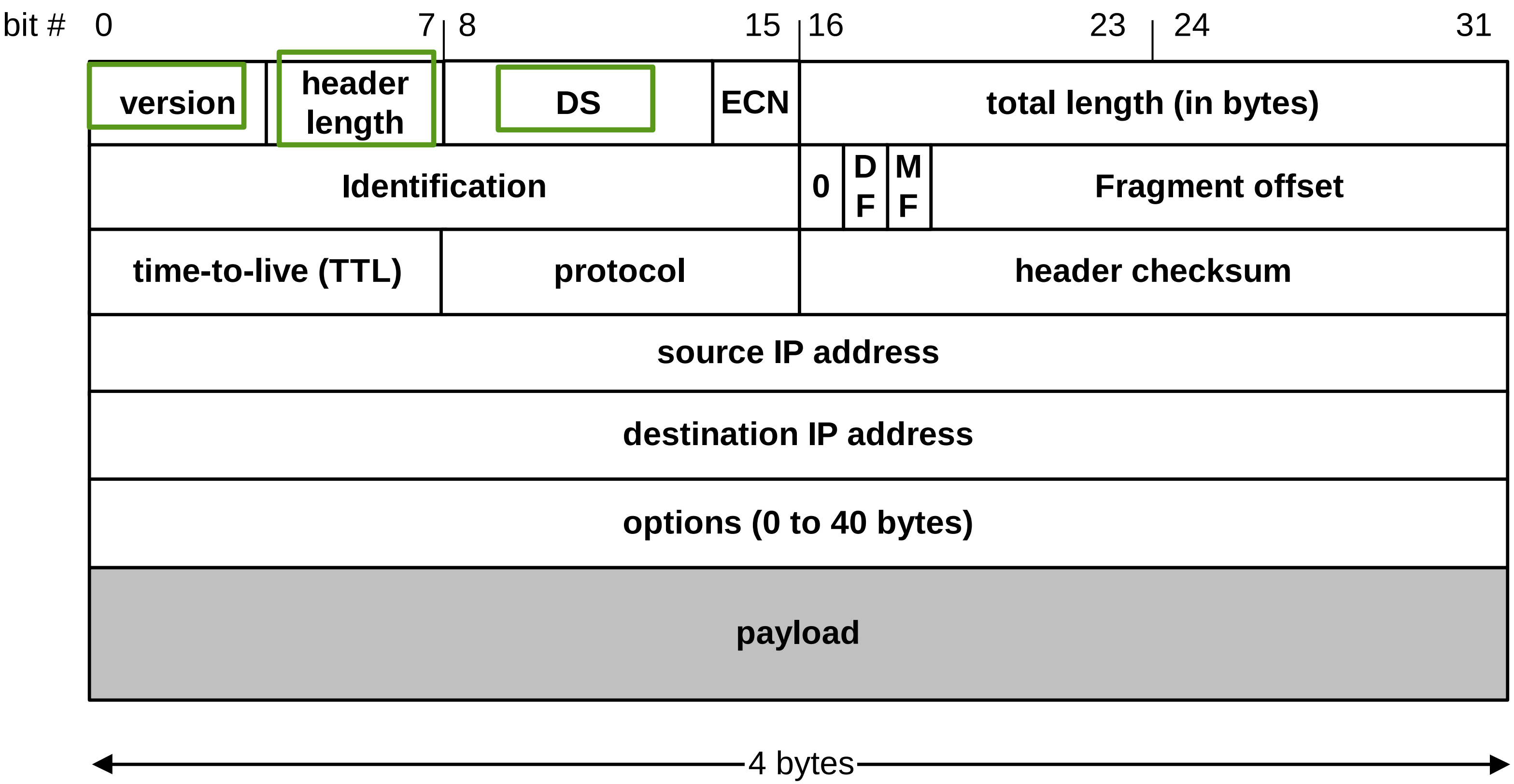
Htons and htonl

- When you're building packets or filling out data structures, you'll need to make sure your two- and four-byte numbers are in Network Byte Order.
- Through a function to set *Host Byte Order* to *Network Byte Order*.
- There are two types of numbers that you can convert: short (two bytes) and long (four bytes)

- **htons()**
- **htonl()**

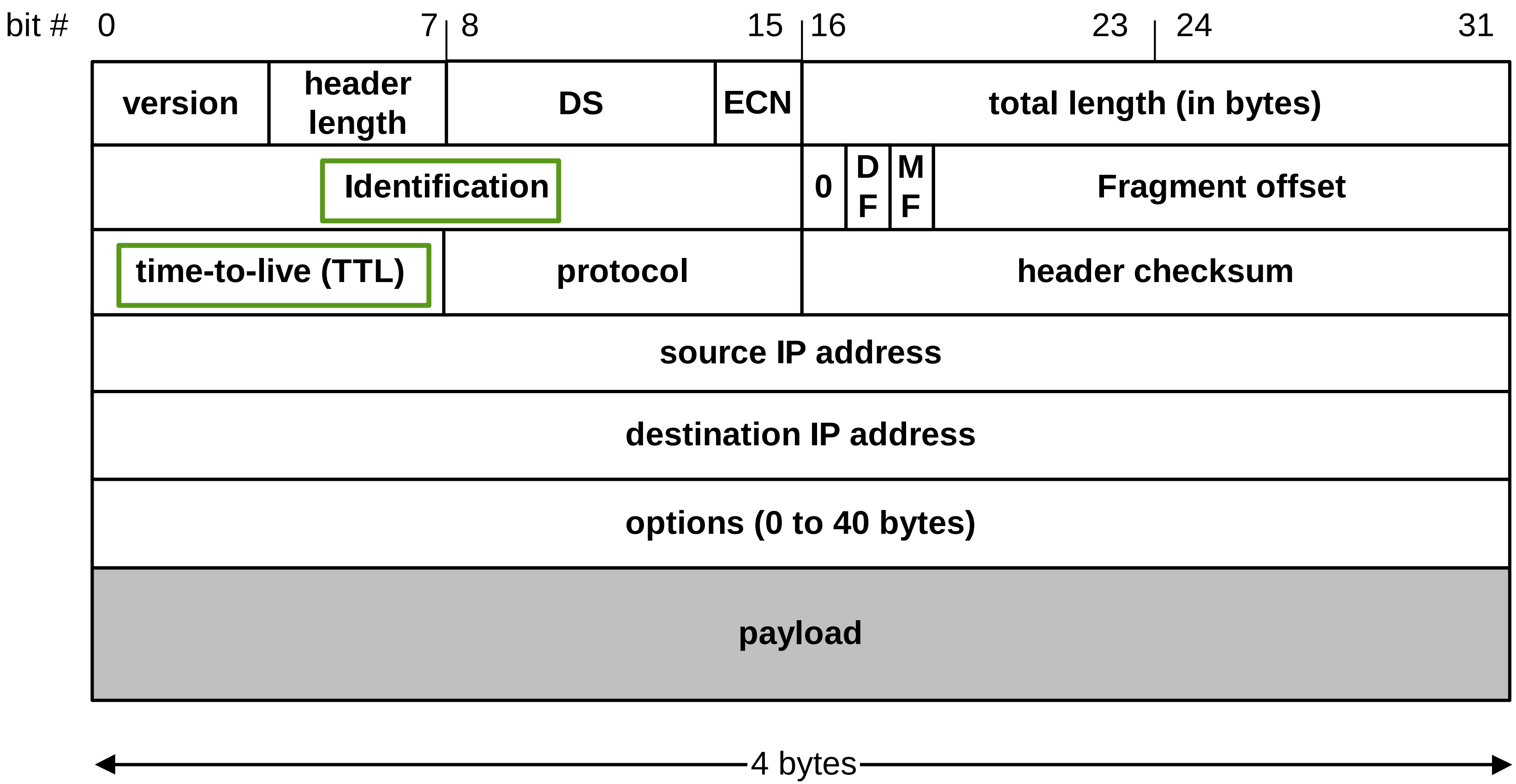
- **h**ost **to** **n**etwork **s**hort
- **h**ost **to** **n**etwork **l**ong

IP Datagram Fields



- **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, typical 20 bytes
- **DS:** Differentiated Services Code Point. Type of service, or type of data (used to specify priority or request low-delay routes)

IP Datagram Fields



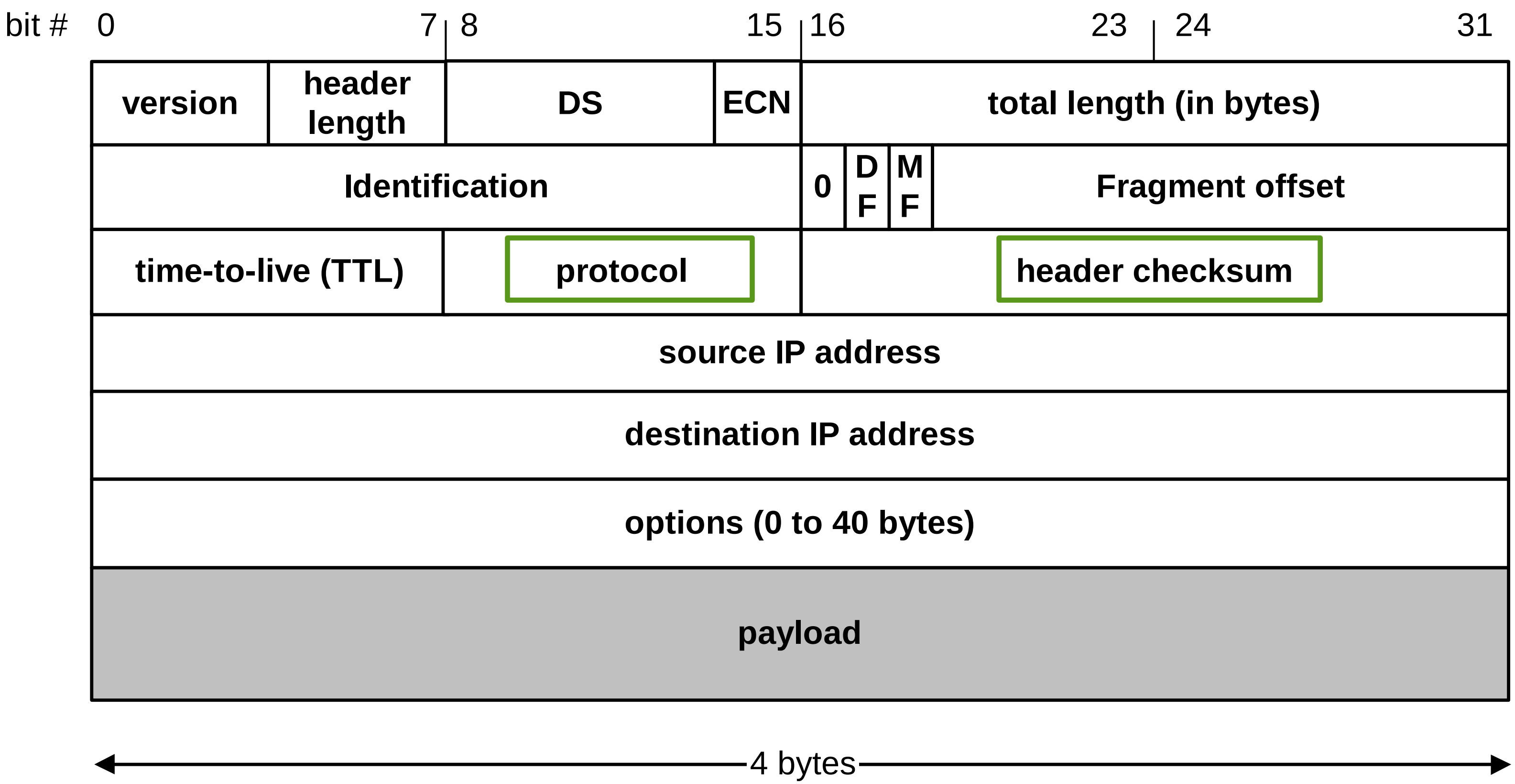
- **Identification (16 bits):** Unique identification of a datagram from a host. Incremented whenever a datagram is transmitted

Time to live

- **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

IP Datagram Fields

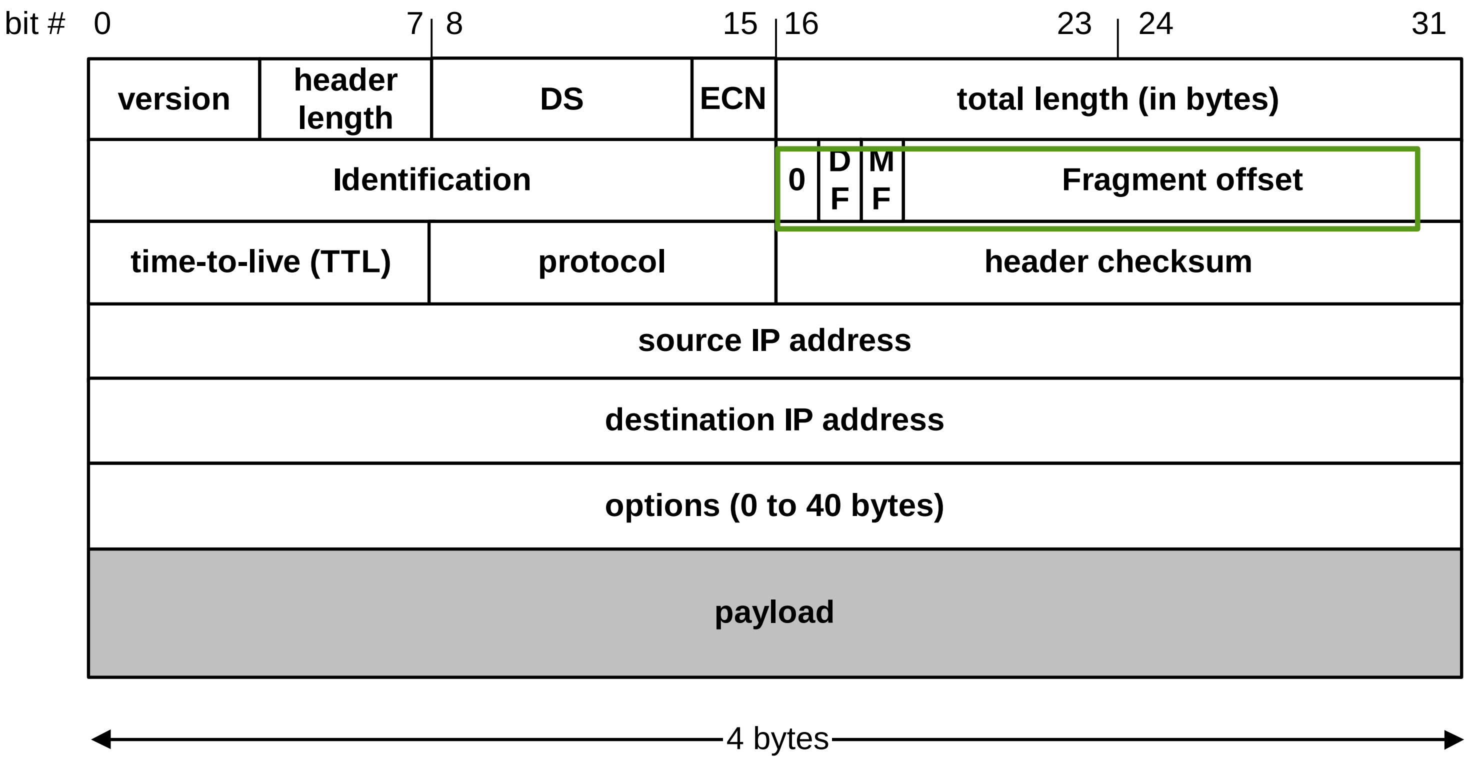


- **Protocol (1 byte):** Specifies the higher-layer protocol. Used only when an IP datagram reaches its final destination
- **Header checksum (2 bytes):** A simple 16-bit long checksum of the header

The rest

- **Source and Destination IPs**
- **Options:**
    - Security restrictions: Specifies the levels of security.
    - 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

Fragment flags and offset



- **Flags (3 bits):** First bit always set to 0, DF bit (Do not fragment), MF bit (More fragments)
- **Fragment offset:** For fragmentation/reassembly

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)**

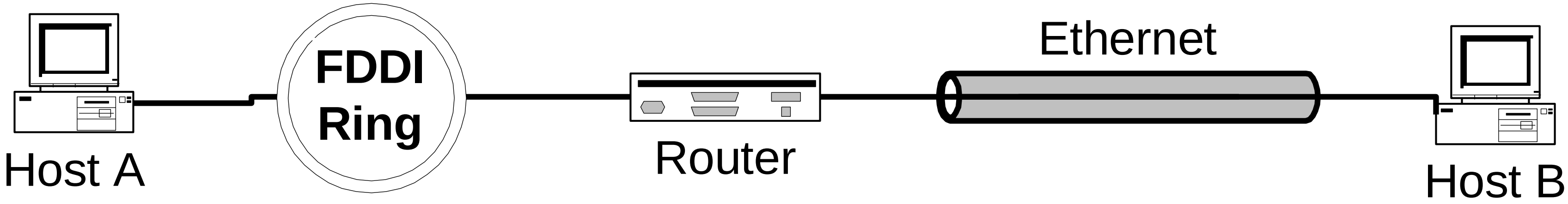
- MTUs for various data link protocols:

-- Ethernet:	1500	-- FDDI:	4352
-- 802.3:	1492	-- ATM AAL5:	9180
-- 802.5:	4464	-- 802.11(WLAN):	2272

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?
  FDDI: Fiber Distributed Data Interface

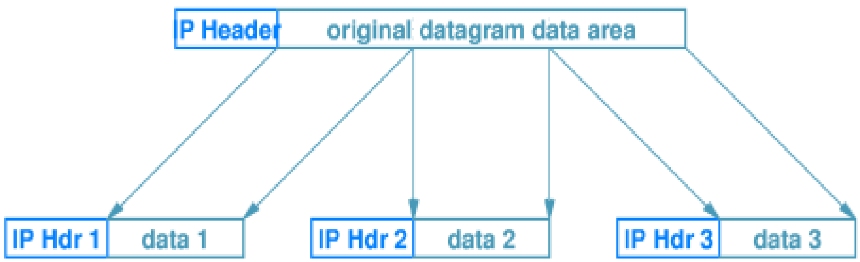


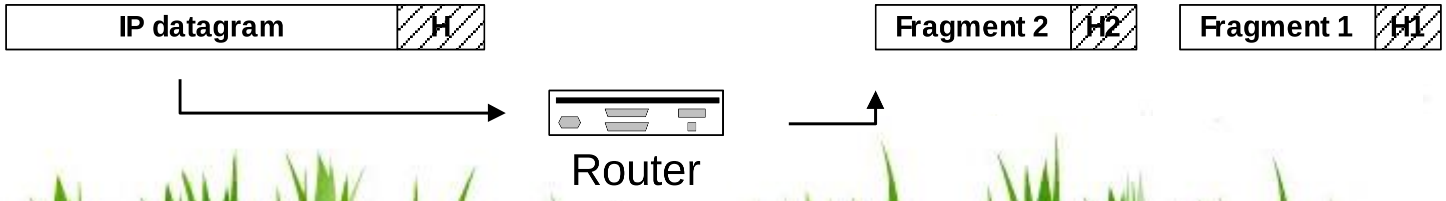
MTUs: FDDI: 4352 Ethernet: 1500

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

Fragmentation / reassembly

- 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 !!





Fields used for fragmentation

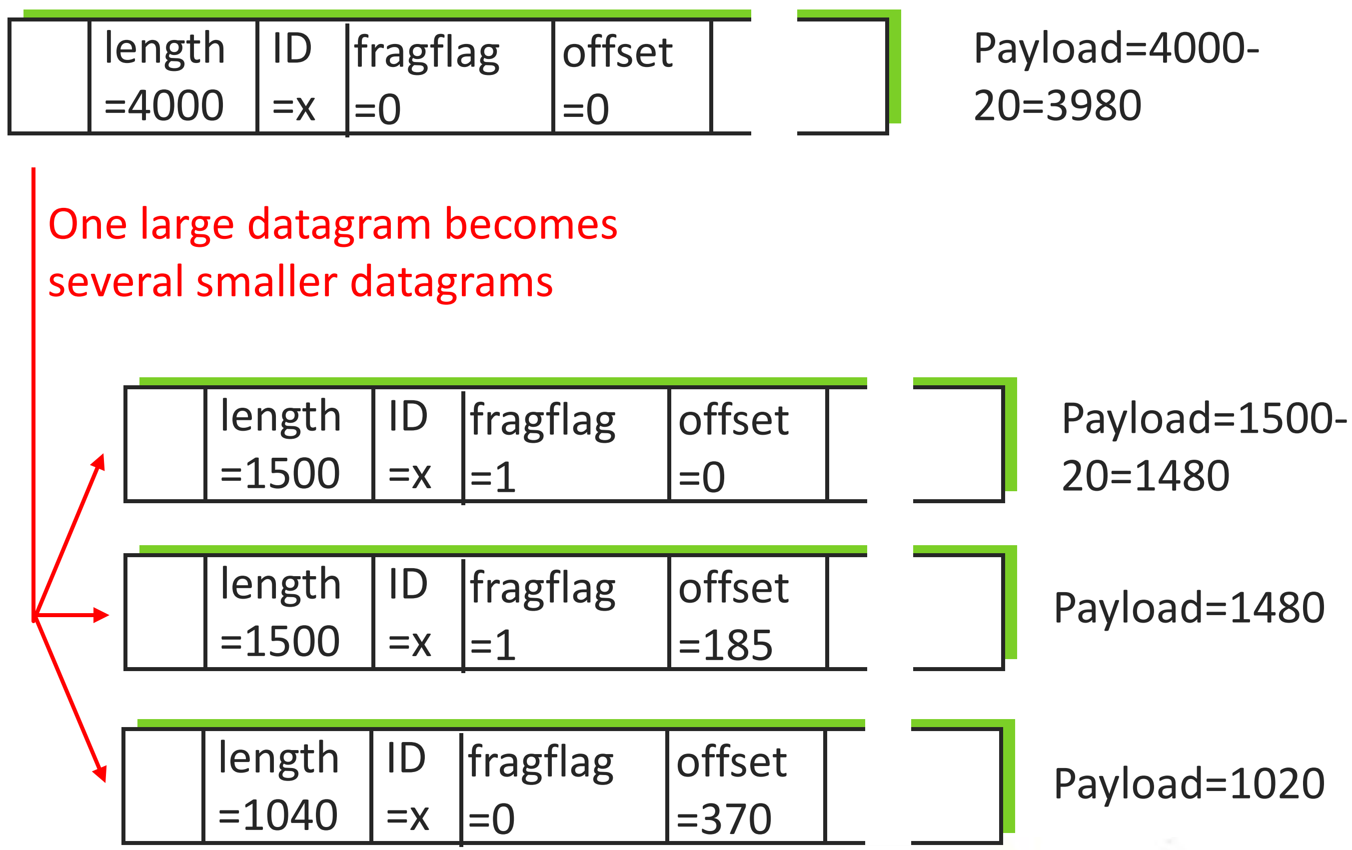
- The following fields in the IP header are involved:

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: Should not fragment this Datagram, should 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: specifies where the fragment fits within the original IP datagram (specified in 8-byte chunks)
- Total length: Total length of the current fragment

Example of Fragmentation

- A datagram of 4000B from a network of 4000 MTU to 1500 MTU



Resources

- Slides from the book: “Mastering Computer Networks: An Internet Lab Manual”, J. Liebeherr, M. El Zarki, Addison-Wesley, 2003.
- Slides from the book: “Computer Networking: A Top Down Approach”, 5th edition. Jim Kurose, Keith Ross Addison-Wesley, 2009.
- RFC 791
  - http://tools.ietf.org/pdf/rfc791.pdf

Thanks for attending!