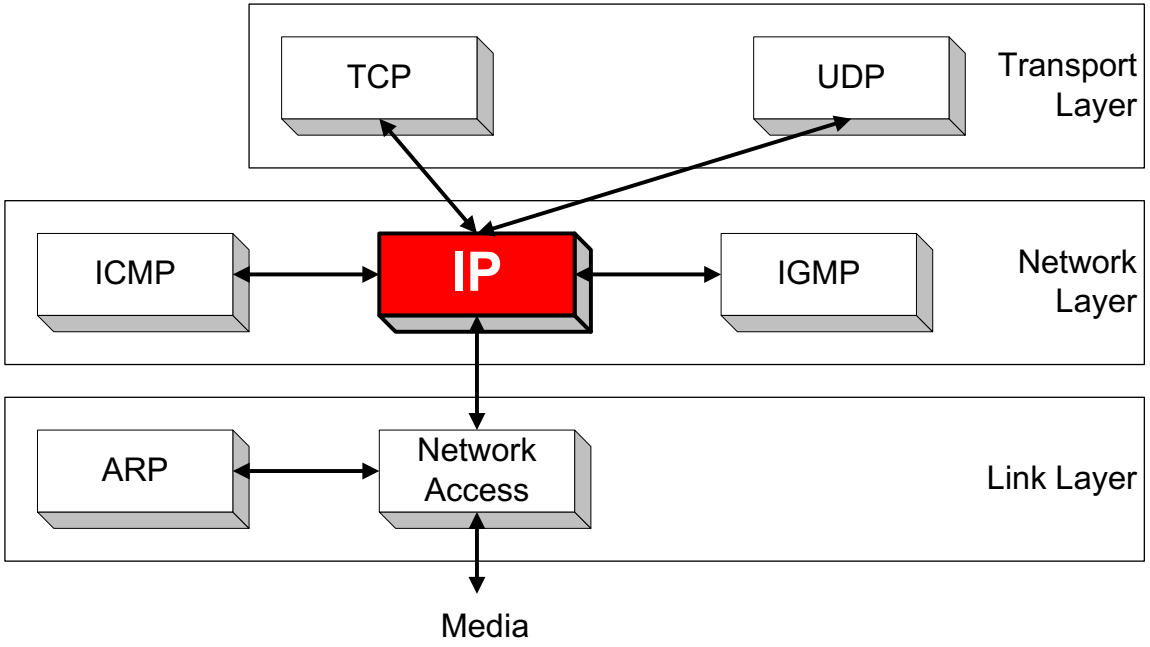
IP - The Internet Protocol

A module on the Internet Protocol.

IP Orientation

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



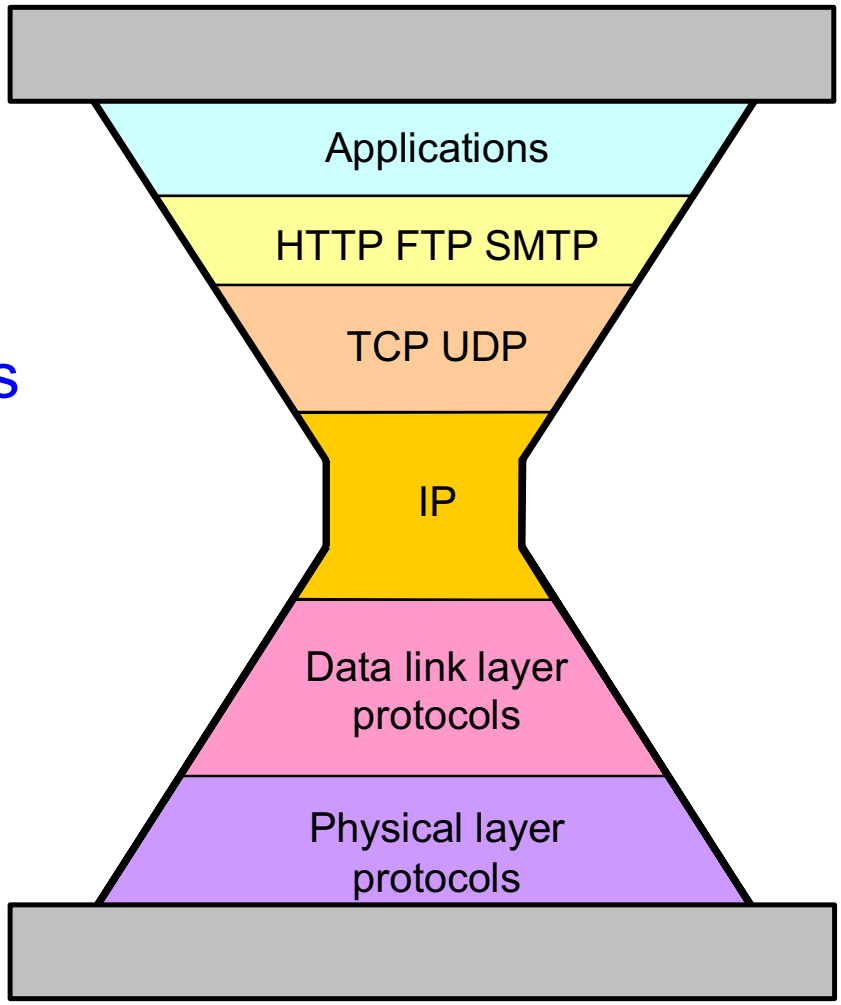
- IP's current version is Version 4 (IPv4). It is specified in RFC 891.

IP: The waist of the hourglass

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

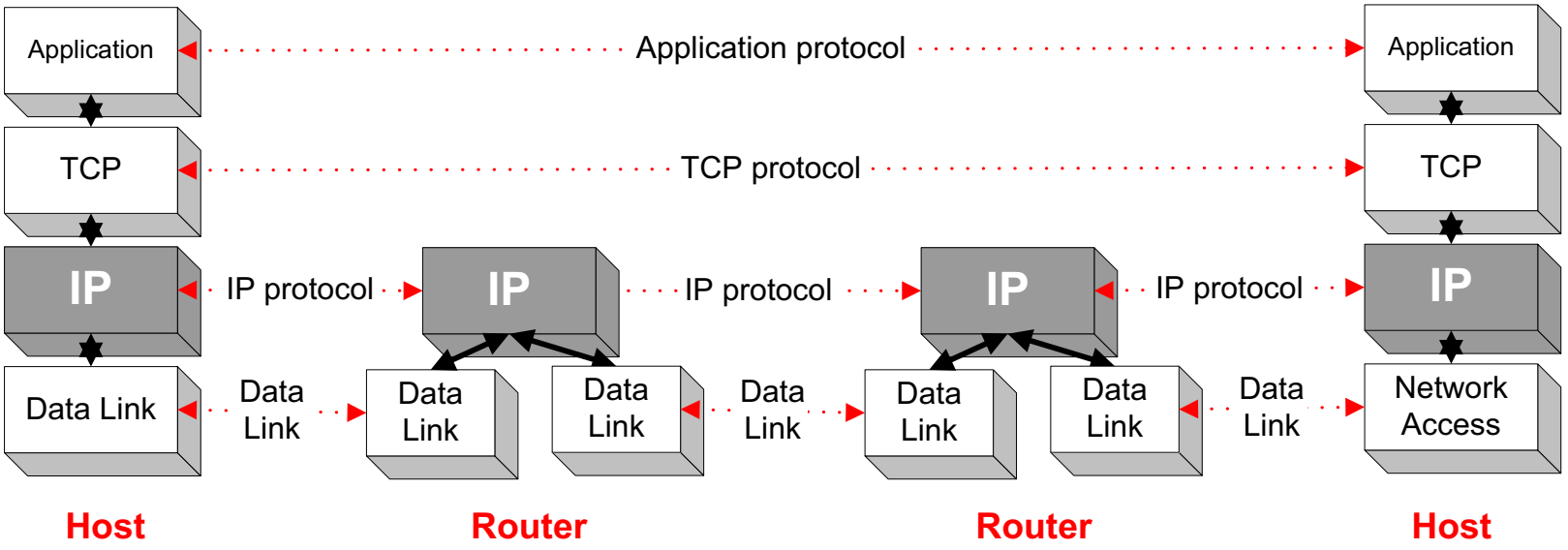
- Multiple higher-layer protocols
- Multiple lower-layer protocols

- Only one protocol at the network layer.



IP Implementation

- 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,…)

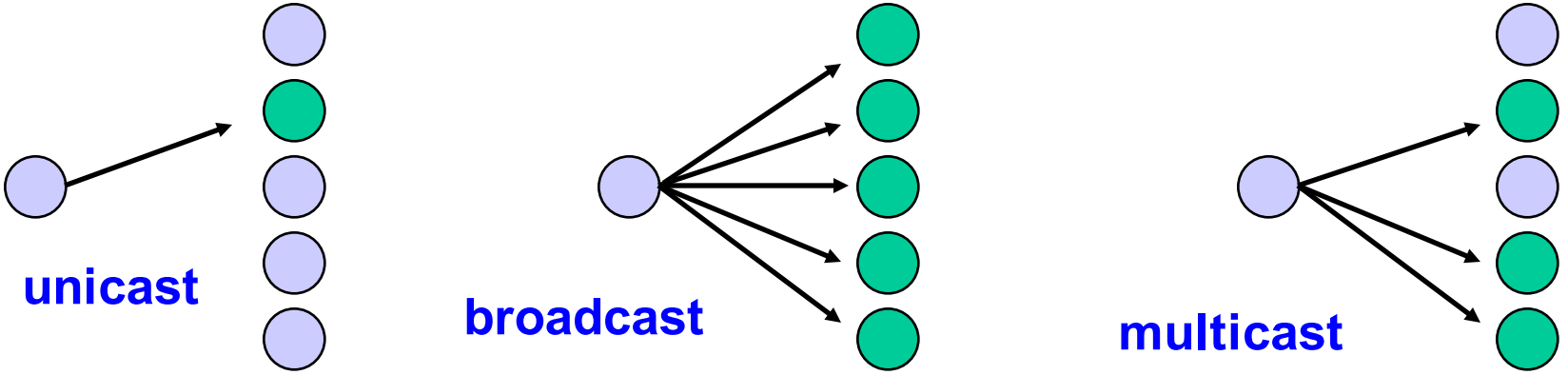
- Consequences:

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

  - Packets may be delivered out-of-sequence

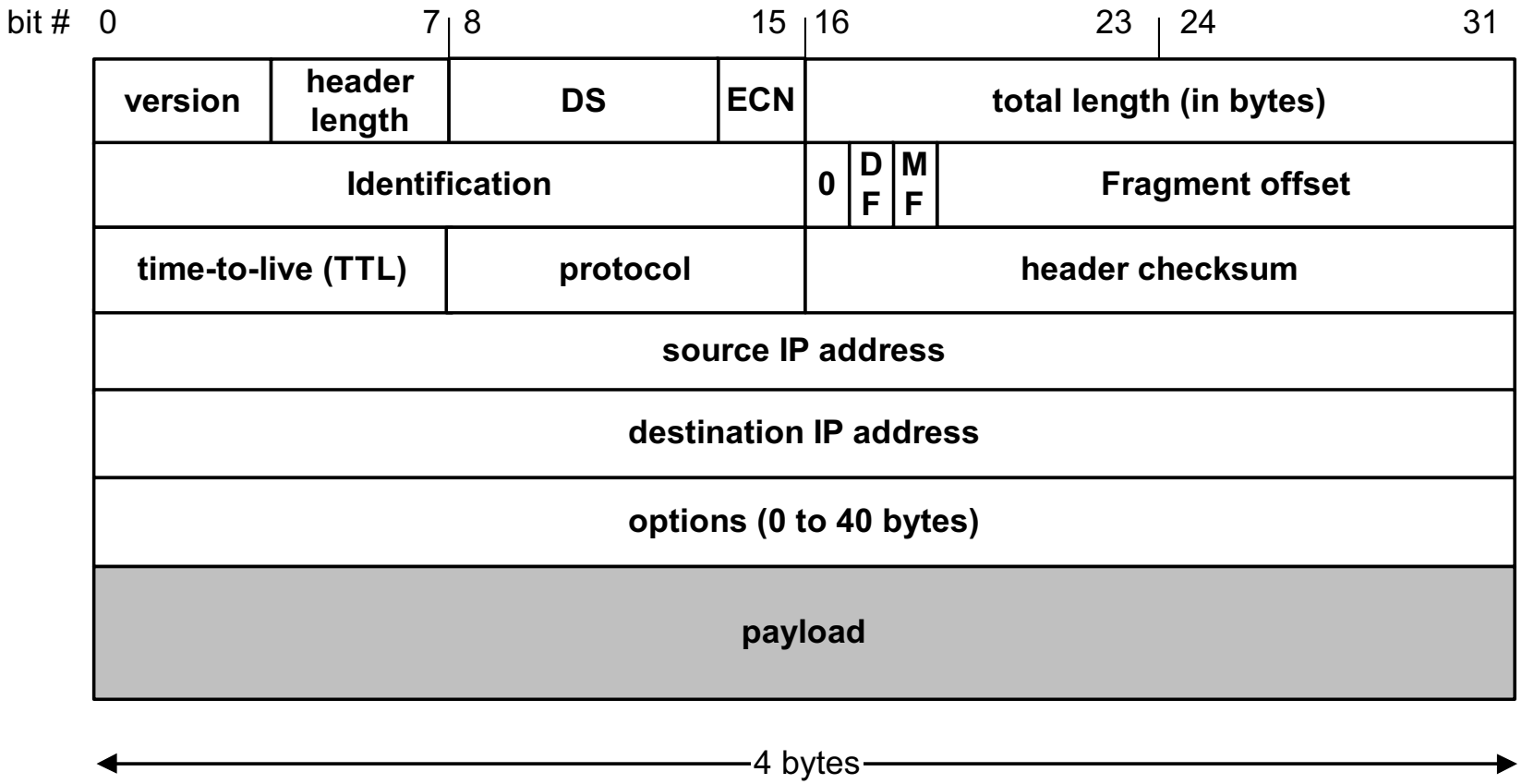
IP Service

- 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:** some computers store 32-bit words in little endian format.

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

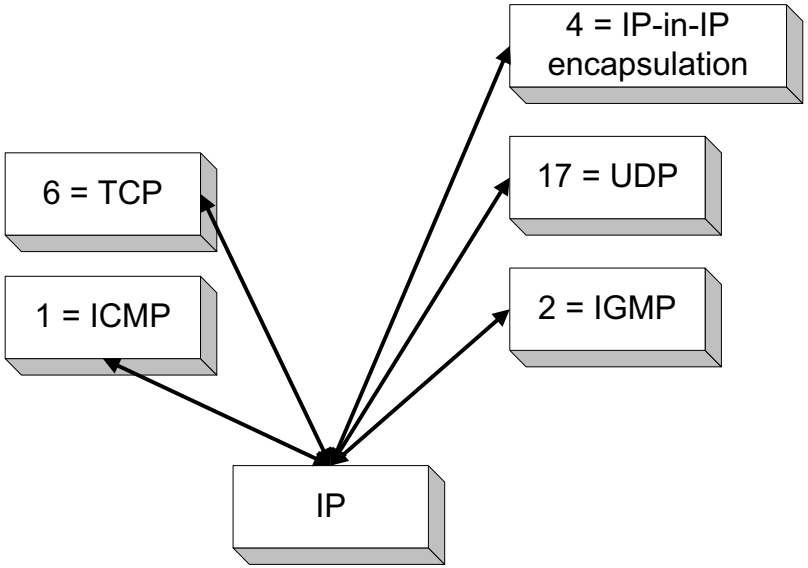
Fields of the IP Header

- **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.

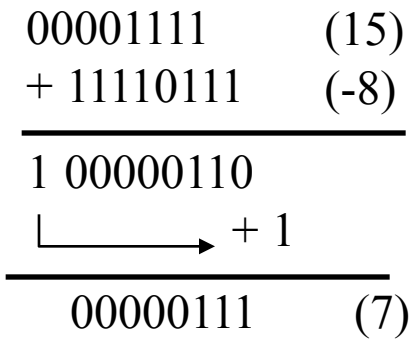


Fields of the IP Header

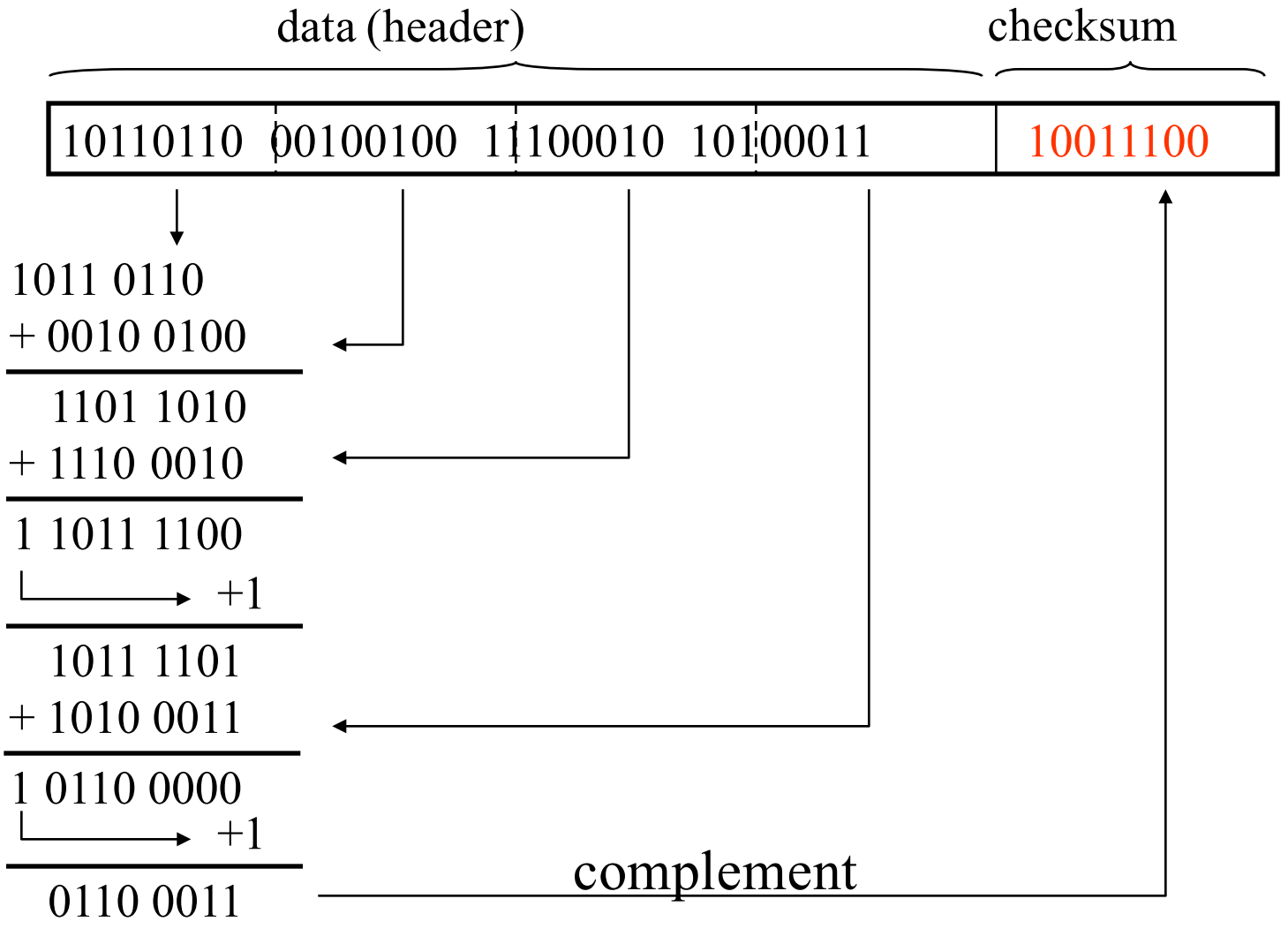
- **Header checksum (2 bytes):** A simple 16-bit long checksum which is computed for the header of the datagram.
- Checksum (16 bits)
  - Complement of the *one's-complement* sum of all 16-bit words in the IP **packet header**
- Each router computes ones-complement sum of entire header *including checksum* …
  - … should get 0 (or 0xffff)
  - If not, router discards packet as corrupted
    - So it doesn't act on bogus information

Fields of the IP Header

- Assume 8-bit numbers
  - Numbers starting with “0” are positive
    - E.g., 00001111 → 15;
  - Numbers starting with “1” are negative; negative number is obtained by inverting all bits of the positive number
    - E.g., 11110000 → -15
  - 00000000 and 11111111 both represent 0
- Addition: add carry-on to result



Fields of the IP Header - Checksum Example



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 (Bytes), 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 (Bytes), imposed by the data link protocol is called **maximum transmission unit (MTU)**

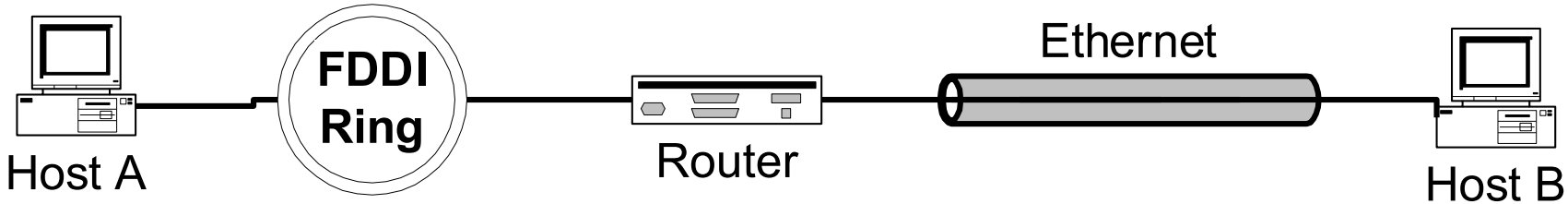
- MTUs for various data link protocols:

Ethernet:	1500	FDDI:	4352
802.3:	1492	ATM AAL5:	9180
802.5:	4464	PPP:	negotiated < 1492
X.25:	576		

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?

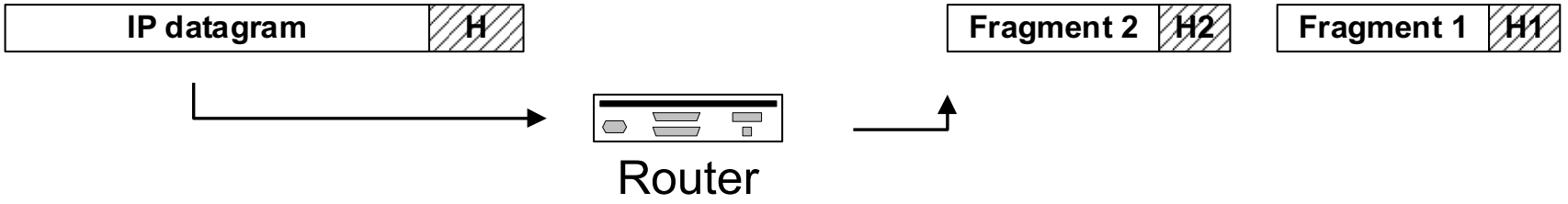


MTUs: FDDI: 4352 Ethernet: 1500

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

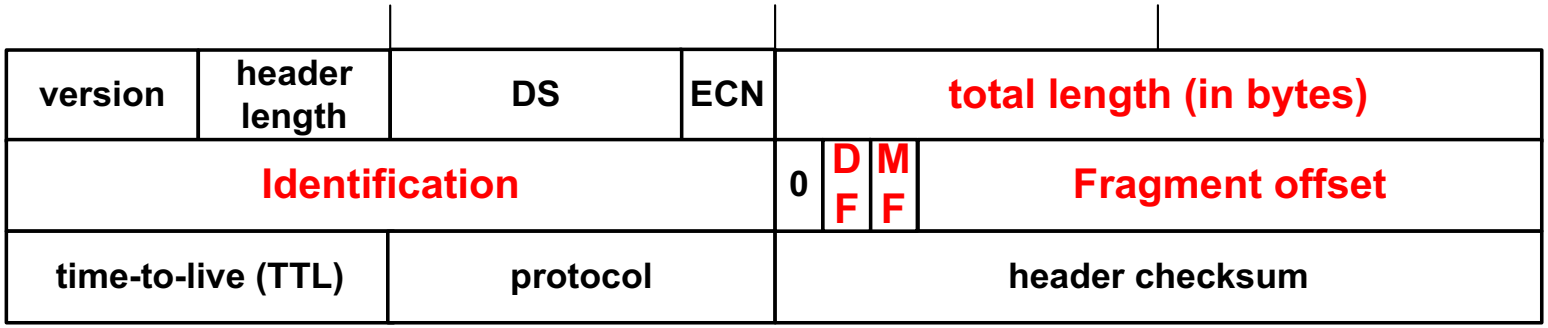
Where is Fragmentation done?

- 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:



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

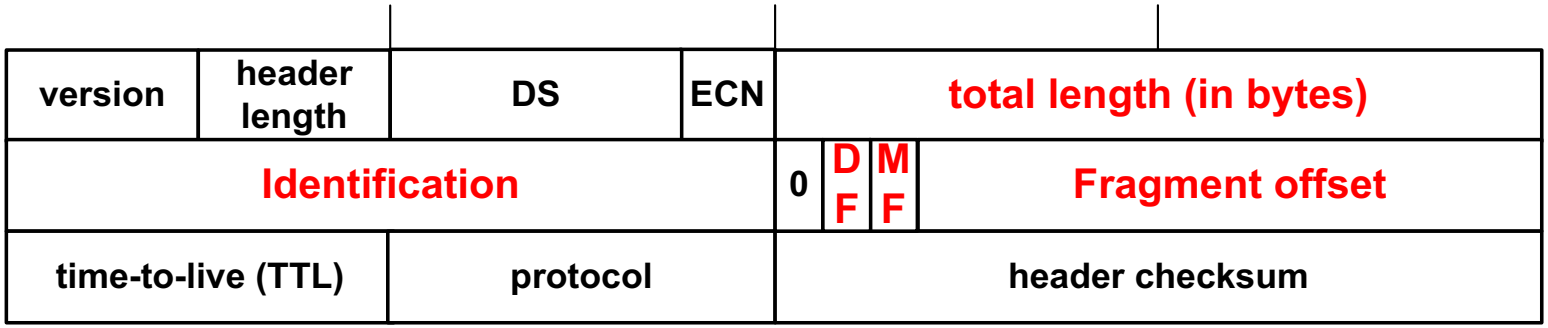
Flags

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:

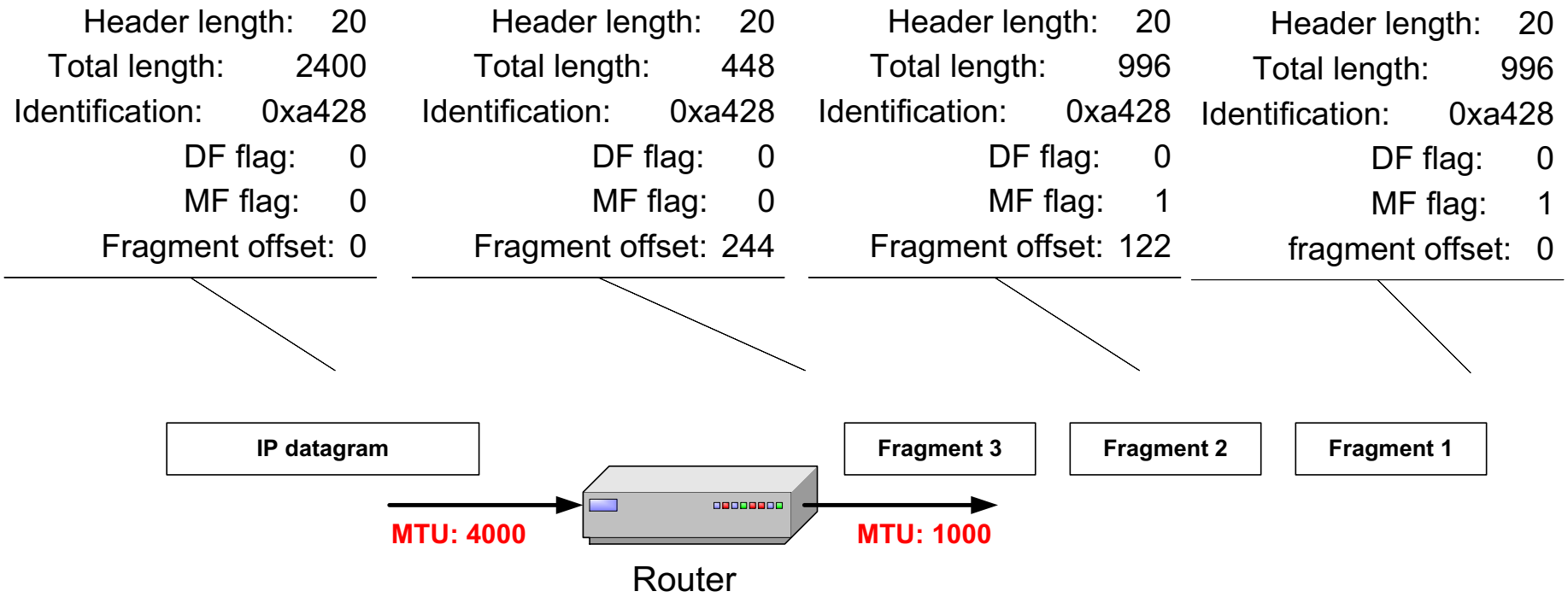


Fragment offset — Offset of the payload of the current fragment in the original datagram

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



Определение оптимального MTU

MTU - определяет размер фрейма при передаче блока данных на канальном уровне сети.
Для Ethernet по умолчанию это 1500 байт.
Когда IP хочет отослать блок данных большего размера происходит его фрагментация (разбиение).
Для разных сетей этот параметр различен, например, Ethernet 1500, FDDI 4352, X.25 576 байт.

Соответственно когда IP пакет попадает допустим из Ethernet с MTU 1500 на РОУТЕР и далее должен пойти по X.25 сети, РОУТЕР произведет фрагментацию пакета - разбивая его на несколько размером не больше параметра MTU для сети X.25

Однако в передаваемом IP пакете может стоять специальный бит DontFragment - запрещающий фрагментацию пакета - в связи с этим в вышеописанном случае РОУТЕР просто отбросит пакет например в 1400 байт, так как он не может (без разбивки) быть переданным по X.25. и выходом из ситуации может быть ручное уменьшение через конфигурацию операционной системы значение MTU на конечном узле.

Поскольку в Windows начиная с XP автоматически выбирает наилучшее MTU, то необходимо просто удостовериться, что для этого соединения не установлено какое-то, отличное от оптимального (фиксированное) значение.

Но, например, в WiFi роутере или на ADSL-модеме может потребоваться установить оптимальный для VPN размер MTU.

Определение оптимального MTU

Простой и точный способ определения оптимального MTU – MTU Ping Test.

Вы просто отправляете запросы и постепенно снижаете размер пакета до тех пор, пока пакет перестанет фрагментироваться.

Figure 1

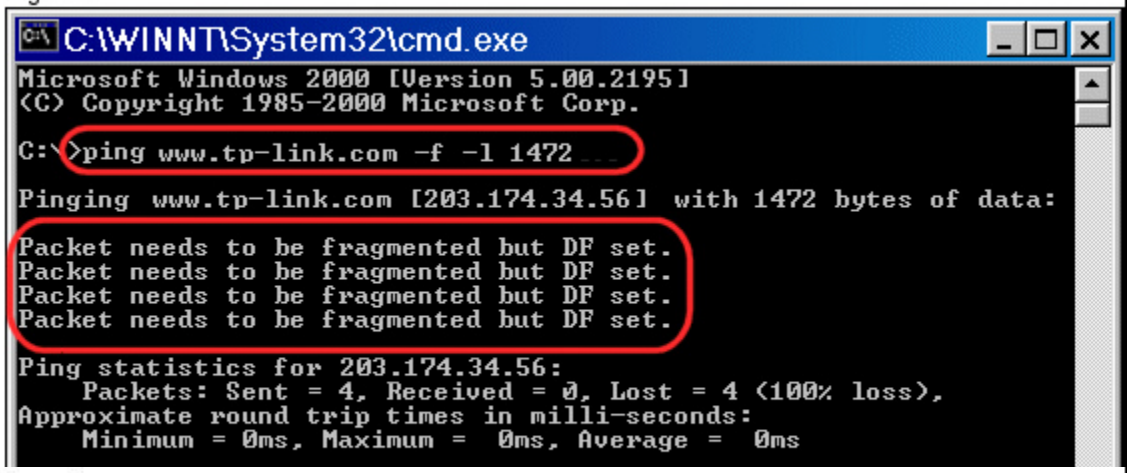
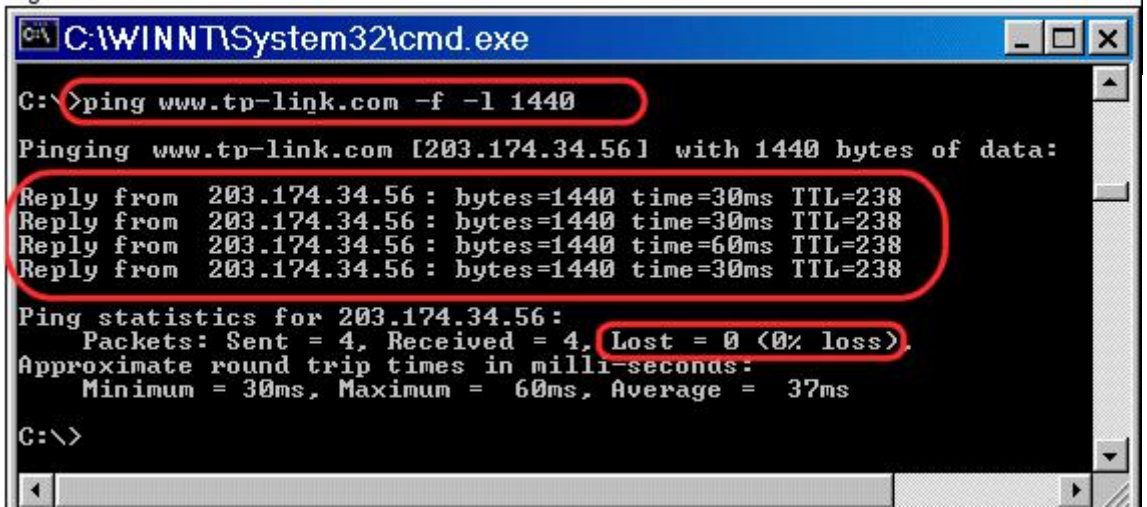


Figure 3



1. Start>Run>“cmd”.
2. ping www.google.com - f - l 1472

Пакет должен фрагментироваться (Figure 1)

3. Уменьшайте размер тестового пакета до тех пор, пока размер отправляемого пакета не будет фрагментироваться (Figure 3)
4. К полученному размеру пакета, прибавьте **28** (20 байт для заголовка IP и 8 байт для заголовка ICMP (ICMP Echo Request).

Пример:

- 1440 – максимальный размер пакета в пинг-тесте
- 1468 – оптимальный размер вашего MTU
- См. Дополнение к лекции