DRONACHARYA
College of Engineering

Computer Science & Engineering

Data Communication and Computer Networks

(MTCSE-101-A)

TCP/IP Protocol Suite

TCP/IP provides end-to-end connectivity specifying how data should be formatted, addressed, transmitted, routed and received at the destination. Protocols exist for a variety of different types of communication services between computers.

TCP/IP Protocol Suite

History

The Internet Protocol Suite (commonly known as TCP/IP Suite or the TCP/IP Model) is the set of computer network communications protocols and a description framework used for the Internet and other similar networks. The TCP/IP Model was created in the 1970s by DARPA, an agency of the United States Department of Defense (DOD). It evolved from ARPANET, which was the world's first wide area network and a predecessor of the Internet.

TCP/IP Protocol Suite

TCP/IP has evolved. The protocols within the TCP/IP Suite have been tested, modified, and improved over time. The original TCP/IP protocol suite targeted the management of large, evolving internetwork. Some TCP/IP goals included:

- ✓ ***Hardware independence*** **- A protocol suite that could be used on a Mac, PC, mainframe, or any other computer.**
- ✓ ***Software independence*** **- A protocol suite that could be used by different software vendors and applications. This would enable a host on one site to communicate with a host on another site, without having the same software configuration: heterogeneous networks.**
- ✓ ***Failure recovery and the ability to handle high error rates*** **- A protocol suite that featured automatic recovery from any dropped or lost data. This protocol must be able to recover from an outage of any host on any part of the network and at any point in a data transfer.**

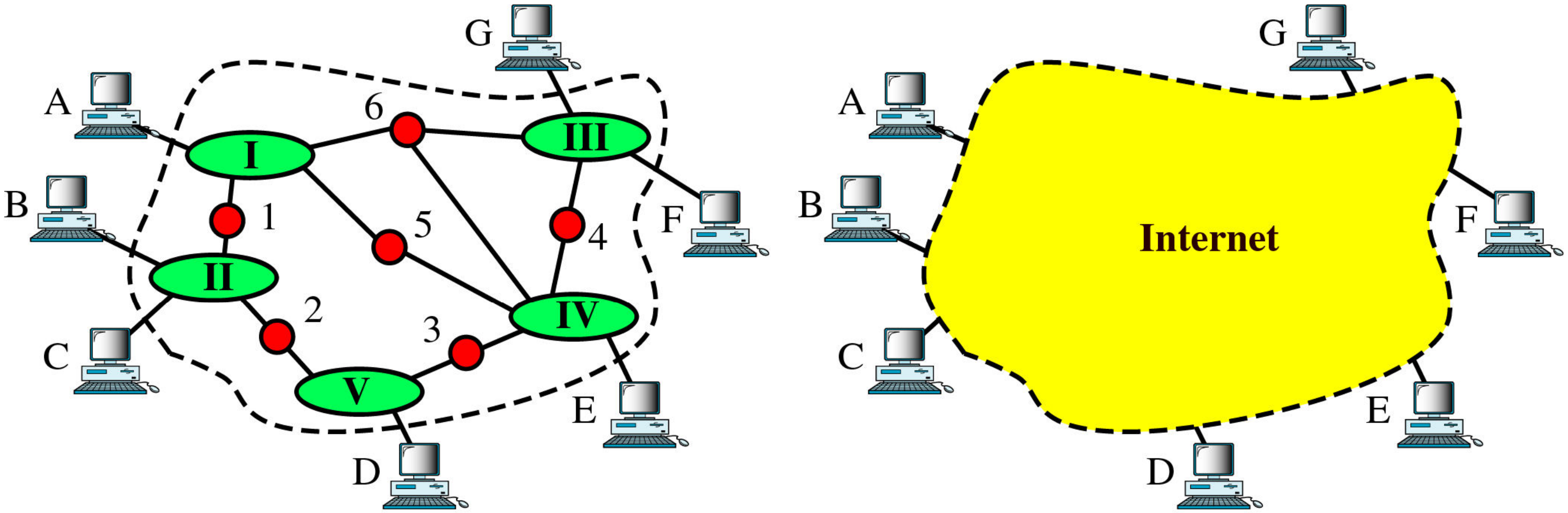
TCP/IP Protocol Suite

- ✓ ***Efficient protocol with low overhead*** **- A protocol suite that had a minimal amount of "extra" data moving with the data being transferred. This extra data called overhead, functions as packaging for the data being transferred and enables the data transmission. Overhead is similar to an envelope used to send a letter, or a box used to send a bigger item—having too much overhead is as efficient as using a large crate to send someone a necklace.**
- ✓ ***Ability to add new networks to the internetwork without service disruption*** **- A protocol suite that enabled new, independent networks to join this network of networks without bringing down the larger internetwork.**
- ✓ ***Routable Data*** **- A protocol suite on which data could make its way through an internetwork of computers to any possible destination. For this to be possible, a single and meaningful addressing scheme must be used so that every computer that is moving the data can compute the best path for every piece of data as it moves through the network.**

What is TCP/IP?

- TCP/IP is a set of protocols developed to allow cooperating computers to share resources across a network
- TCP stands for “Transmission Control Protocol”
- IP stands for “Internet Protocol”
- They are Transport layer and Network layer protocols respectively of the protocol suite
- The most well known network that adopted TCP/IP is Internet – the biggest WAN in the world

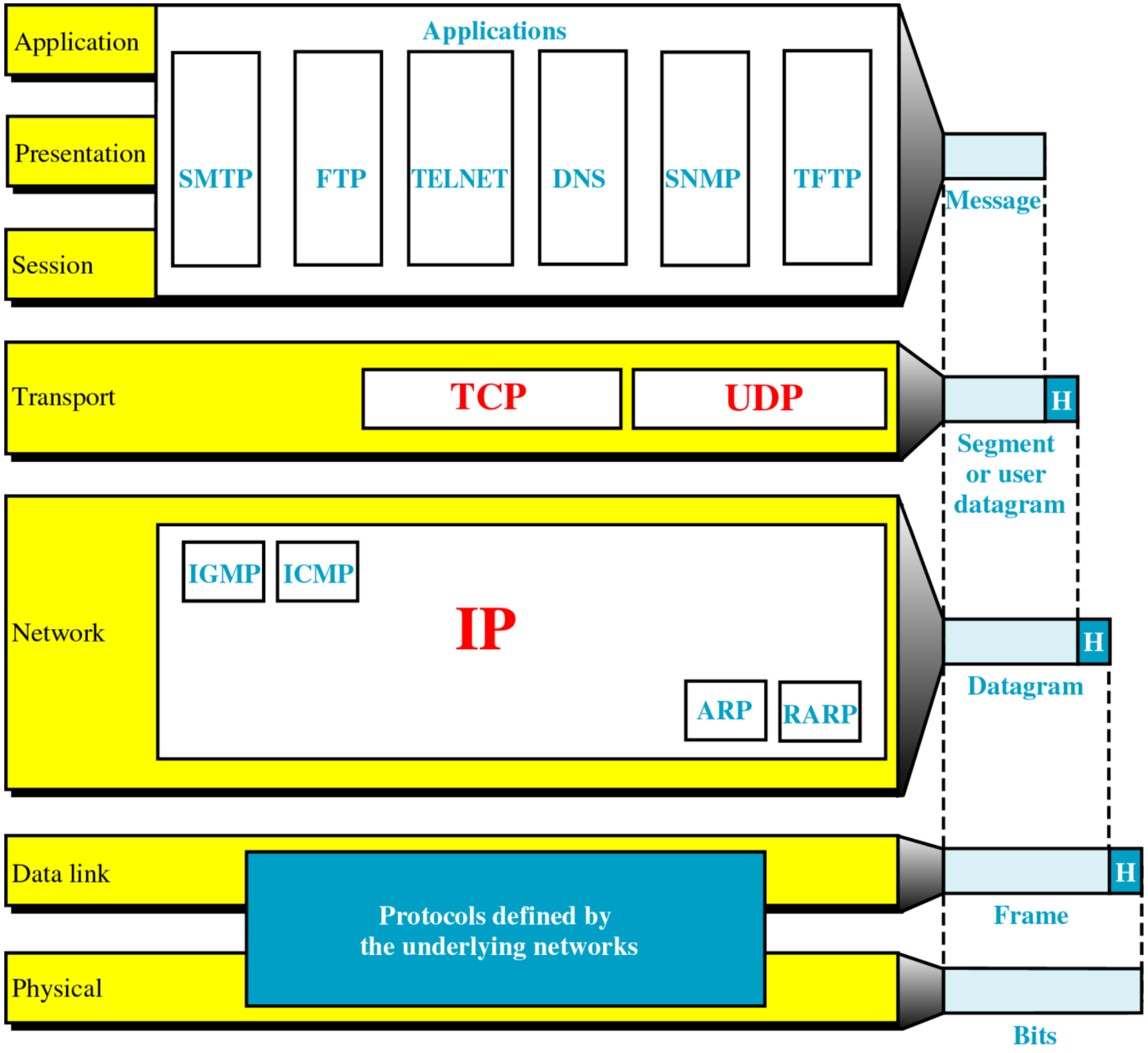
An Internet According to TCP/IP



a. An actual internet

b. An internet seen by TCP/IP

TCP/IP and the OSI Model

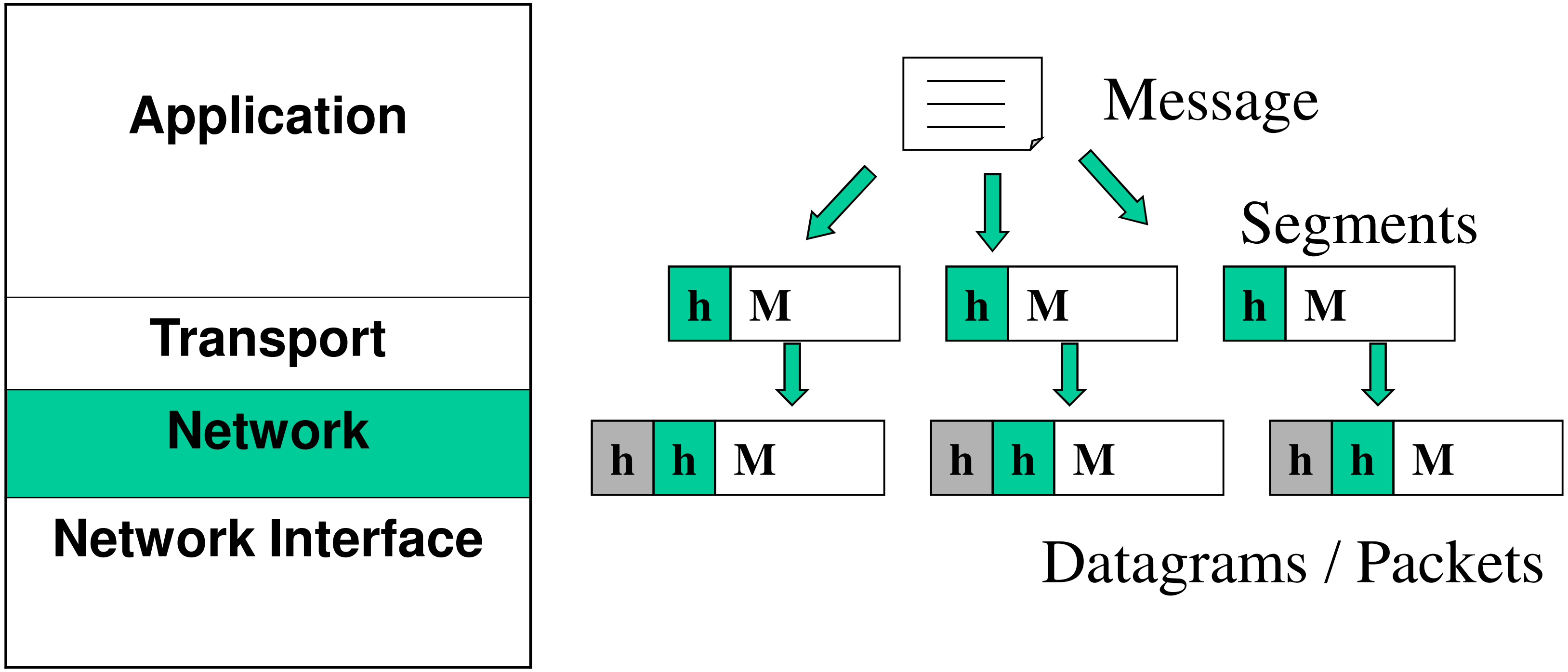


- Application layer protocols define the rules when implementing specific network applications
- Rely on the underlying layers to provide accurate and efficient data delivery
- Typical protocols:
  - FTP – File Transfer Protocol
    - For file transfer
  - Telnet – Remote terminal protocol
    - For remote login on any other computer on the network
  - SMTP – Simple Mail Transfer Protocol
    - For mail transfer
  - HTTP – Hypertext Transfer Protocol
    - For Web browsing

- **TCP/IP is built on "connectionless" technology, each datagram finds its own way to its destination**
- **Transport Layer protocols define the rules of**
  - **Dividing a chunk of data into segments**
  - **Reassemble segments into the original chunk**
- **Typical protocols:**
  - **TCP – Transmission Control Protocol**
    - **Provide further the functions such as reordering and data resend**
  - **UDP – User Datagram Service**
    - **Use when the message to be sent fit exactly into a datagram**
    - **Use also when a more simplified data format is required**

- **Network layer protocols define the rules of how to find the routes for a packet to the destination**
- **It only gives best effort delivery. Packets can be delayed, corrupted, lost, duplicated, out-of-order**
- **Typical protocols:**
  - **IP – Internet Protocol**
    - **Provide packet delivery**
  - **ARP – Address Resolution Protocol**
    - **Define the procedures of network address / MAC address translation**
  - **ICMP – Internet Control Message Protocol**
    - **Define the procedures of error message transfer**

Network Layer



Application

Transport

Network

Network Interface

Message

Segments

Datagrams / Packets

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,…)

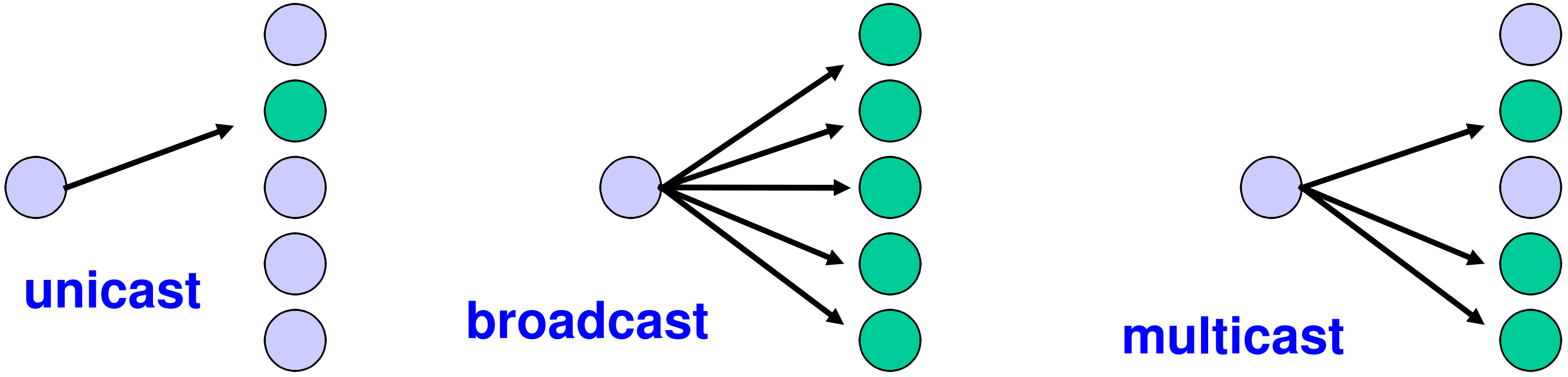
- Consequences:

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

  - Packets may be delivered out-of-sequence

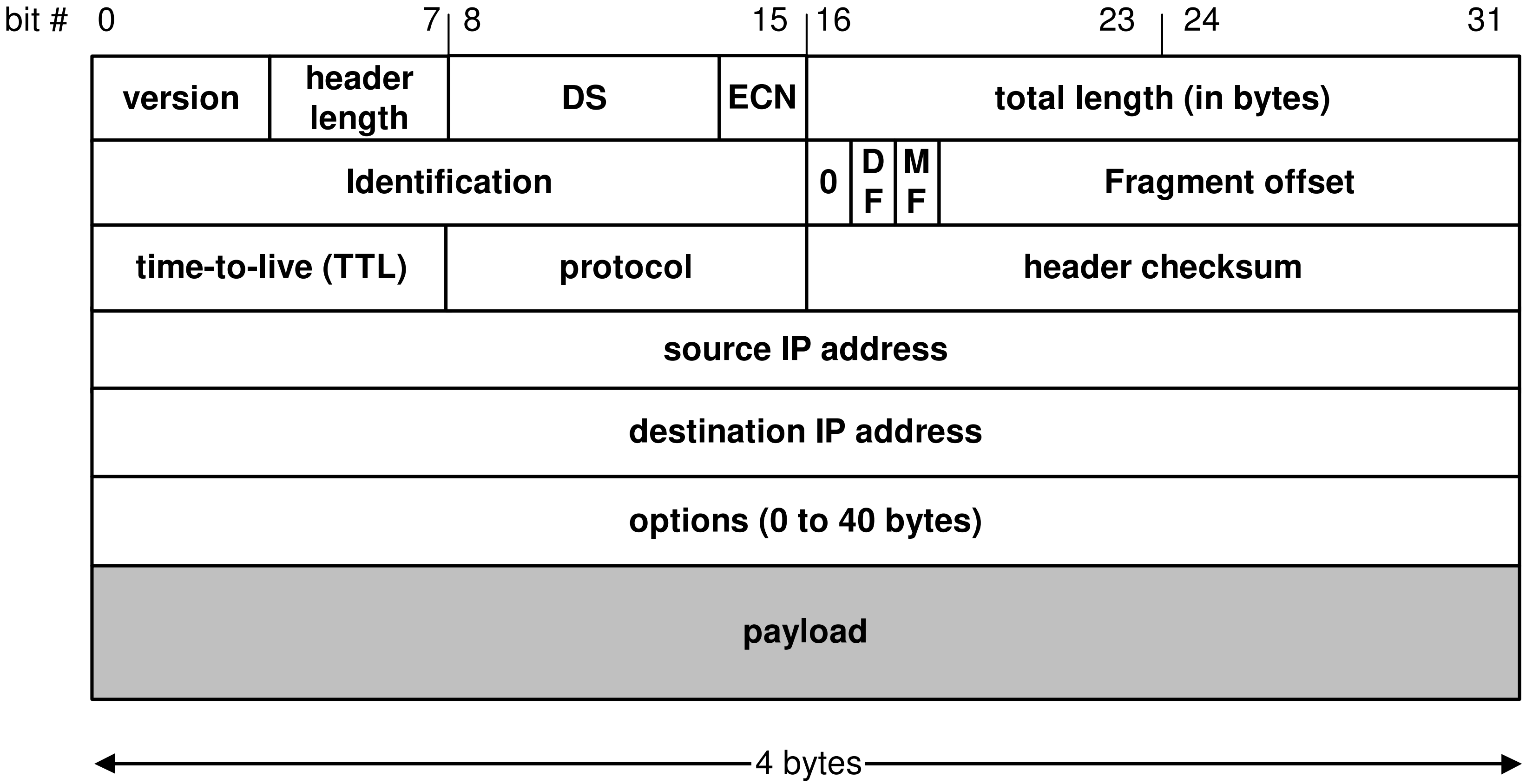
IP Service

- 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



bit # 0 7 8 15 16 23 24 31

version | header length | DS | ECN | total length (in bytes)

Identification | 0 | DF | MF | Fragment offset

time-to-live (TTL) | protocol | header checksum

source IP address

destination IP address

options (0 to 40 bytes)

payload

4 bytes

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

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

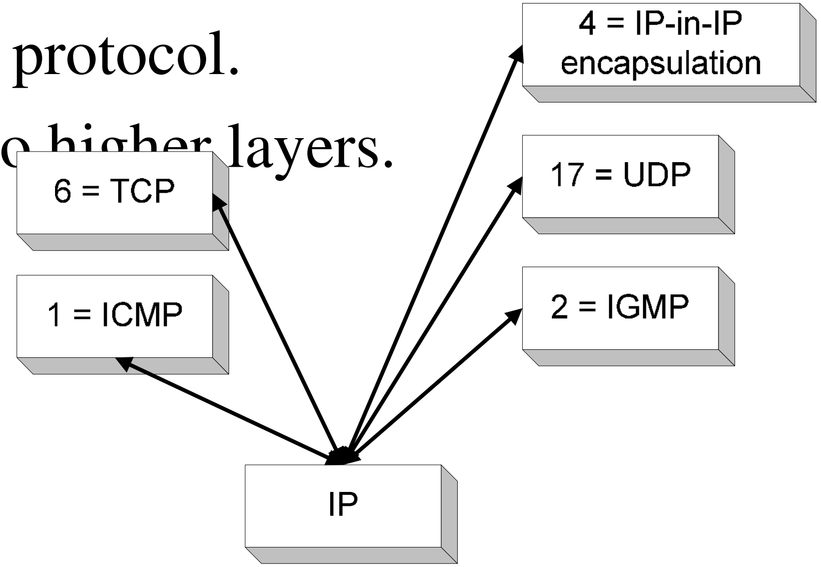
Fields of the IP Header

- **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)**

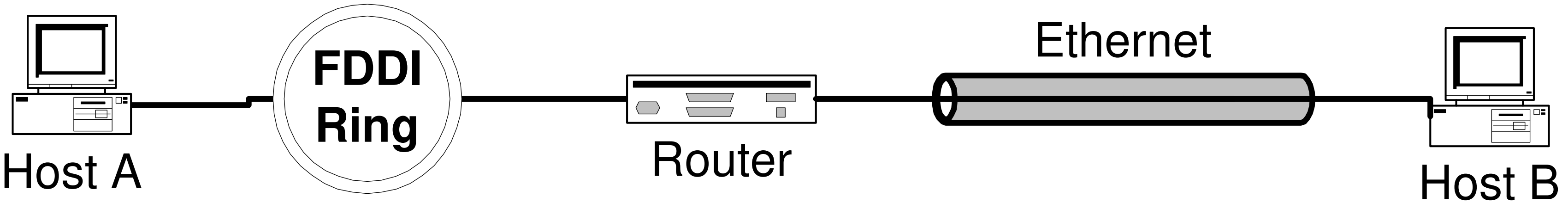
- MTUs for various data link protocols:

Ethernet:	1500	FDDI:	4352
		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?

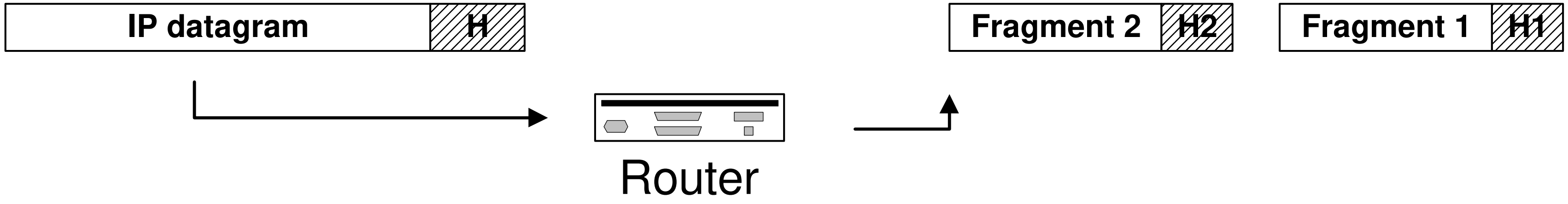


MTUs: FDDI: 4352 Ethernet: 1500

- **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:

<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

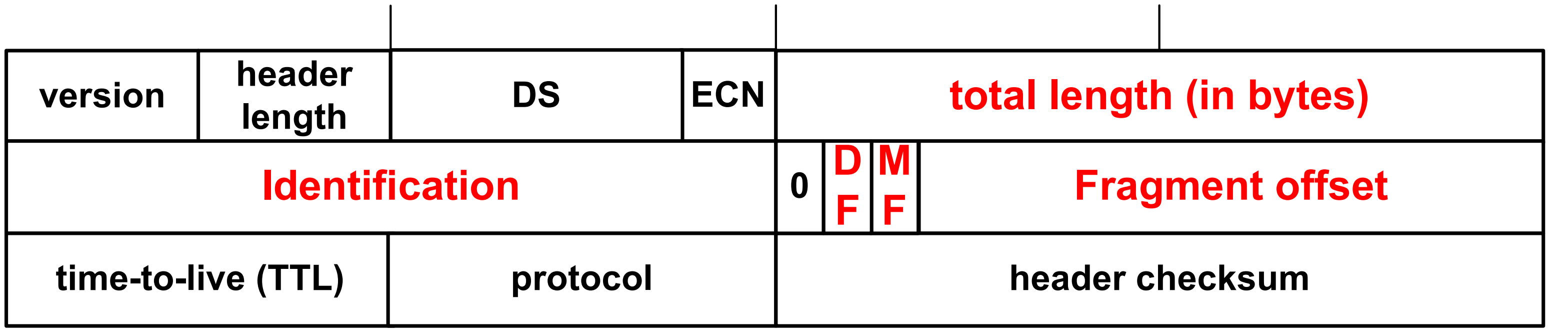
Flags

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

What’s involved in Fragmentation?

- The following fields in the IP header are involved:

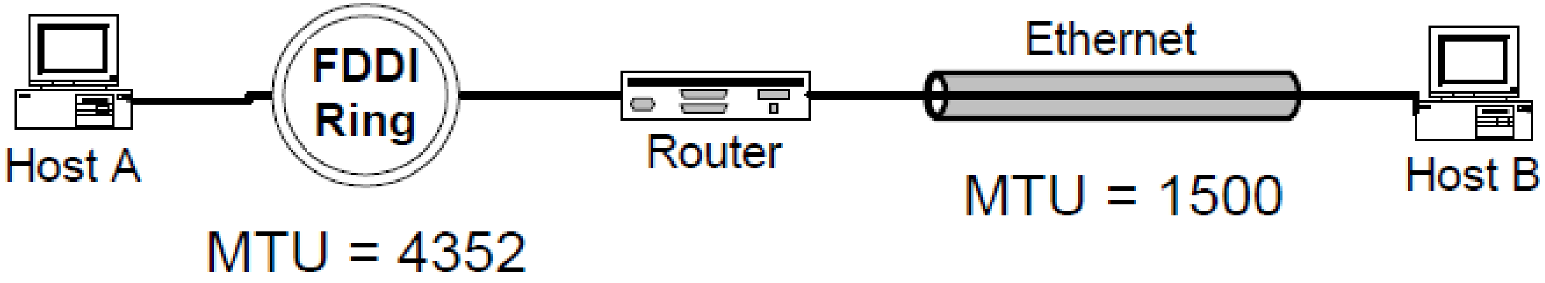


Fragment offset Offset of the payload of the current fragment in the original datagram

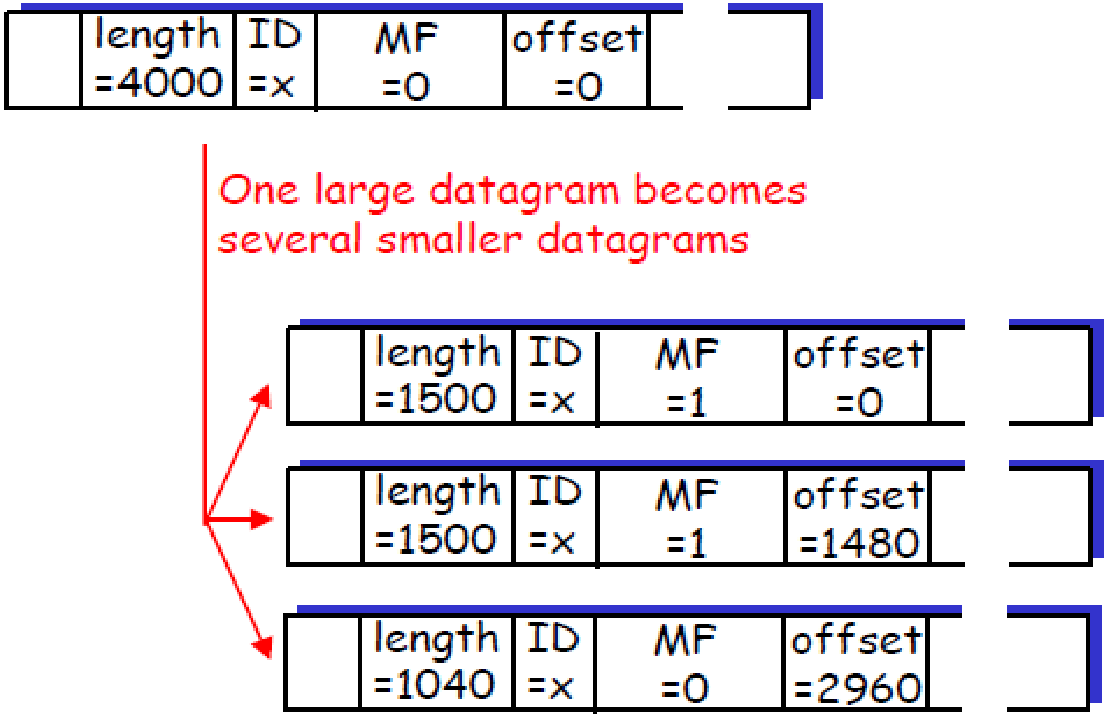
Total length Total length of the current fragment

IP Fragmentation Example

- Host A wants to send to Host B an IP datagram of size = 4000 Bytes



IP Fragmentation Example



One large datagram becomes several smaller datagrams

Multiple Fragmenting Points

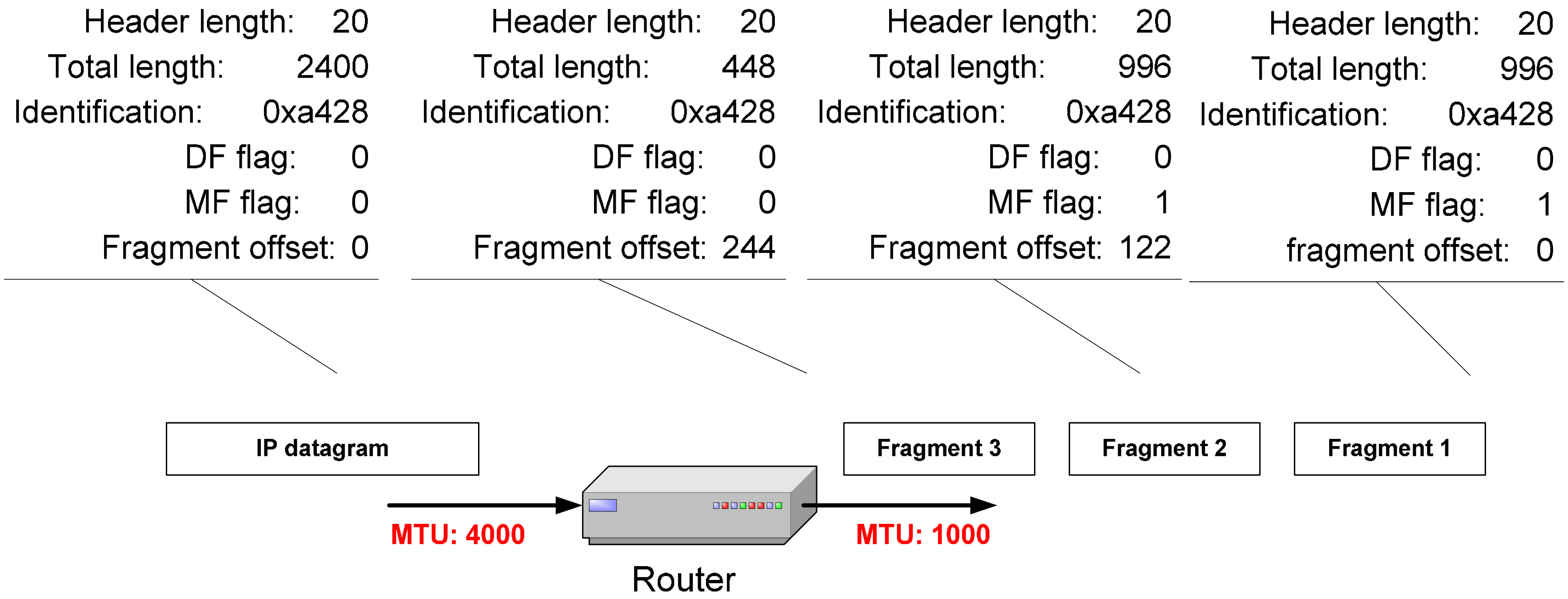
- Let MTUs along internet path be
  - 1500
  - 1500
  - 1000
  - 1500
  - 576
  - 1500
- Result: fragmentation can occur twice

Fragment Loss

- Receiver
  - Collects incoming fragments
  - Reassembles when all fragments arrive
  - Does not know identity of router that did fragmentation
  - Cannot request missing pieces
- Consequence: Loss of one fragment means entire datagram lost

Example of Fragmentation

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



Header length:	20
Total length:	2400
Identification:	0xa428
DF flag:	0
MF flag:	0
Fragment offset:	0

Header length:	20
Total length:	448
Identification:	0xa428
DF flag:	0
MF flag:	0
Fragment offset:	244

Header length:	20
Total length:	996
Identification:	0xa428
DF flag:	0
MF flag:	1
Fragment offset:	122

Header length:	20
Total length:	996
Identification:	0xa428
DF flag:	0
MF flag:	1
fragment offset:	0

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.