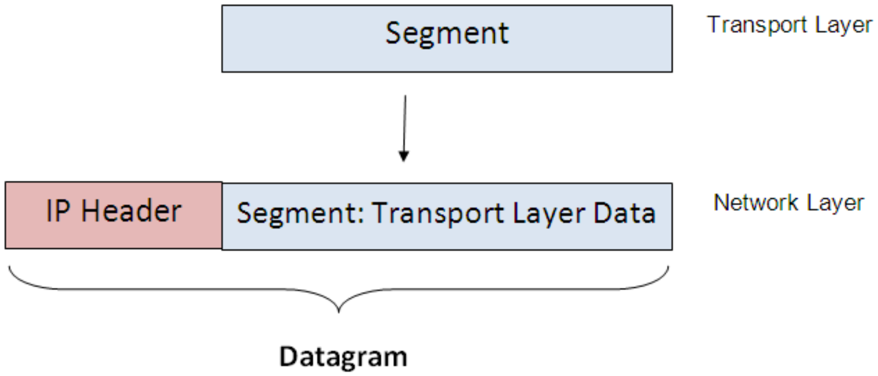
Internet Protocol (IP)

The Internet Protocol (IP) is the cornerstone of the TCP/IP protocol suite. TCP/IP refers to a combination of two protocols, IP at the network layer and the Transmission Control Protocol (TCP) at the transport layer, which together provide one of the most common network transport services used today.

On an internetwork, IP is the protocol responsible for transmitting data from its source to its final destination. A transport layer protocol like TCP or the User Datagram Protocol (UDP) passes data down to the network layer, and IP encapsulates it by adding a header, creating what's known as a datagram. The datagram is addressed to the computer that will ultimately make use of the data, whether that computer is on the local network or on another network far away. Once it has created the datagram, IP passes it down to a data-link layer protocol for transmission over the network.



During the transportation process, various systems might encapsulate the datagram in different data-link layer protocol headers, but the datagram itself remains intact.

Note: Although IP is the most commonly used Network Layer protocol, there are other examples on Network Layer protocols such as Internetwork Packet Exchange (IPX), NetBIOS Enhanced User Interface (NetBEUI), and AppleTalk protocol.

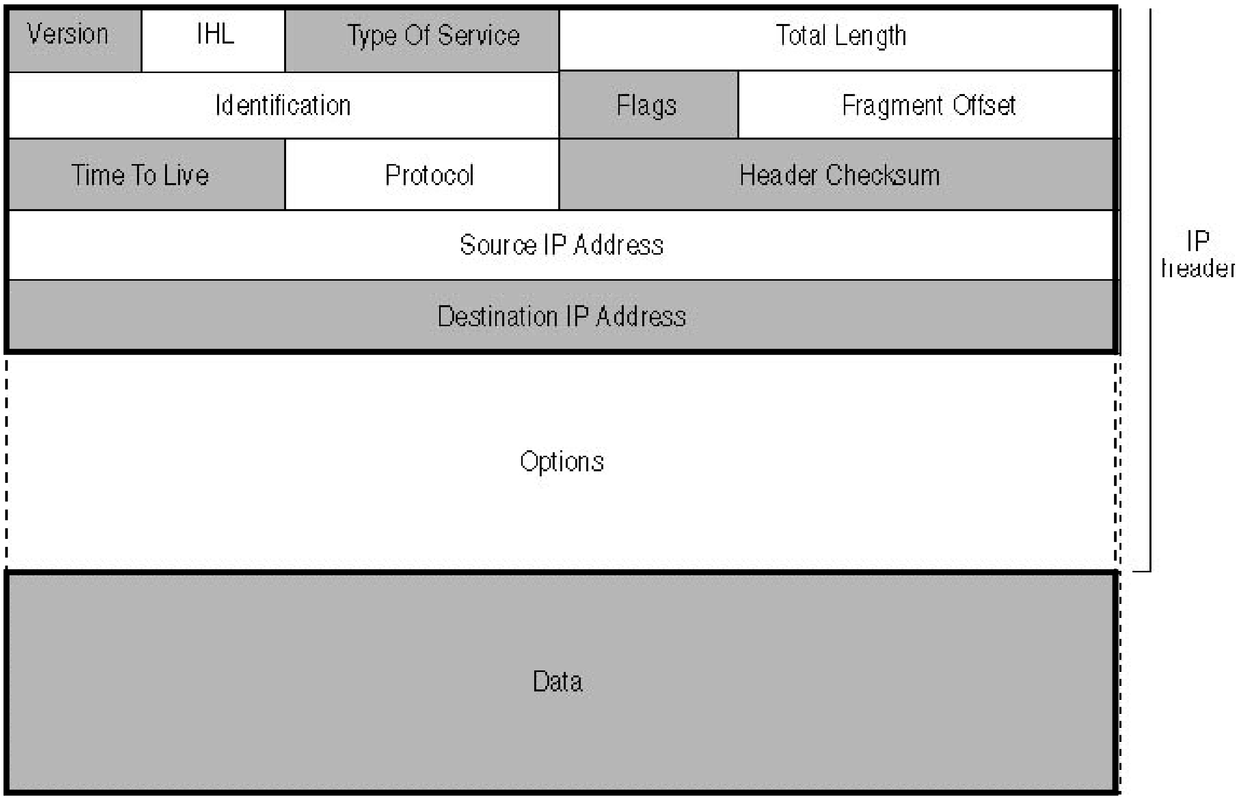
IP Functions

IP performs several functions that are essential to the internetworking process, including the following:

- **Encapsulation:** The packaging of the transport layer data into a datagram
- **Fragmentation:** The division of data into fragments of an appropriate size for transmission over the network
- **Protocol identification:** The specification of the transport layer protocol that generated the data in the datagram
- **Addressing:** The identification of systems in the network using IP addresses
- **Routing:** The identification of the most efficient path to the destination system through the internetwork

Datagram Format:

Datagram format is illustrated in the figure below



- **Version (4 bits):** This field specifies the version of the IP protocol used to create the datagram. The version currently in use is 4. IP version 6 is another example that may play a big role in the next few years.
- **Internet Header Length (IHL, 4 bits):** This field specifies the length of the datagram's header, in 32-bit (4-byte) words. The minimum length of a datagram header is five words (20 bytes), but if the datagram includes additional options, it can be longer, which is the reason for having this field. However the maximum length of the IPv4 datagram header is 60 Bytes.
- **Type Of Service (1 byte):** This field contains a code that specifies the priority for the datagram. This is a rarely used feature that enables a system to assign a priority to a datagram that routers observe while forwarding it through an internetwork.
- **Total Length (2 bytes):** This field specifies the length of the datagram, including that of the Data field and all of the header fields, in bytes.
- **Identification (2 bytes):** This field in combination with source IP address field uniquely identifies the datagram. IP guarantees that the value in this field is not duplicated for datagrams transmitted from a source host to a specific destination host. The value in this filed is initiated by a number that is incremented by one for each new datagram. Identification value helps to reassemble datagrams that have been fragmented during transmission as will be discussed later.
- **Flags (3 bits):** This field contains bits used to regulate the datagram fragmentation process.
- **Fragment Offset (13 bits):** When a datagram is fragmented, the system inserts a value in this field that identifies this fragment's place in the datagram. This will be discussed in more details in fragmentation section of this lecture.

- **Time To Live (TTL, 1 byte):** This field specifies the number of networks (hops) that the datagram is permitted to travel through on the way to its destination. Each router that processes the datagram reduces the value of this field by one. If the value reaches zero, the datagram is discarded.
- **Protocol (1 byte):** This field contains a code that identifies the protocol that generated the information found in the Data field.
- **Header Checksum (2 bytes):** This field contains a checksum value computed on the IP header fields only (and not the contents of the Data field) for the purpose of error detection.
- **Source IP Address (4 bytes):** This field specifies the IP address of the system that generated the datagram.
- **Destination IP Address (4 bytes):** This field specifies the IP address of the system for which the datagram is destined.
- **Options (variable):** This field is present only when the datagram contains one or more of the 16 available IP options. The size and content of the field depends on the number and the nature of the options and it should never exceed 40 Bytes.
- **Data:** This field contains the information generated by the protocol specified in the Protocol field. The size of the field depends on the data-link layer protocol used by the network over which the system will transmit the datagram.

Fragmentation

In an internetwork, routers are intermediate systems that are used to connect different networks. Routers can connect networks that use different media types and different data-link layer protocols, but to forward packets from one network to another, routers must often repackage the datagrams into different data-link layer frames. In some cases, this is simply a matter of stripping off the old frame and adding a new one, but at other times the data-link layer protocols are different enough to require more extensive repackaging.

To overcome this problem, the router splits the datagram arriving from the Token Ring network into multiple fragments, Each fragment has its own IP header and is transmitted in a separate data-link layer frame. The size of each fragment is based on the Maximum Transmission Unit (MTU) size for the outgoing network. If they encounter a network with an even smaller MTU, fragments can themselves be split into smaller fragments. Once fragmented, the individual parts of a datagram are not reassembled until they reach the end system, which is their final destination.

IP header fields that concerns fragmentation process are Identification, Flags and Fragment Offset. The role of these fields and is discussed below.

- **Identification:-** this field in combination with source IP field uniquely identifies a datagram. When this datagram is fragmented, all fragments carry the same value in the identification field. This is very important when the datagram need to be reassembled at the final destination to distinguish the fragments of a datagram that may be arrive out of order with respect to other datagrams or fragments.
- **Flags:** this field consists of three bits. The first bit is reserved and it is (0). The other two are the "Don't fragment" (D) and "More fragment" (M) bits.

1-bit	1-bit	1-bit
Res.	D	M

  - **Don't Fragment bit (D):** When this flag is set (D=1) , the datagram is not allowed to be fragmented. If a router needs to fragment this datagram due MTU limitation of the next data-link network, the router is not allowed to do so and this datagram is discarded.

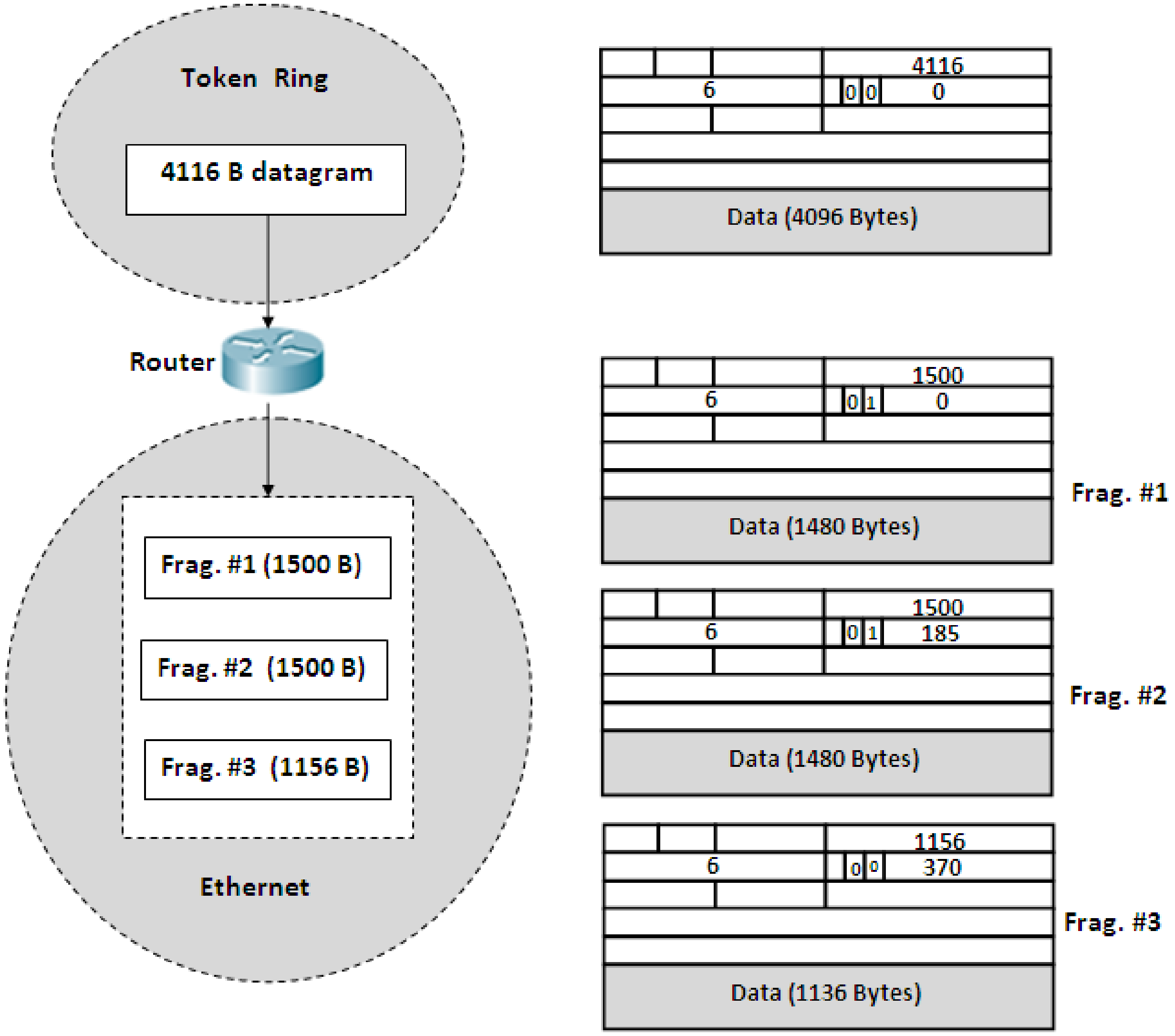
- **More Fragment bit (M):** in a fragmented datagram, this flag is set in all fragments (M=1) in all fragments except the last one. It also should be noted that the value of M equals zero in a non fragmented datagram.

- **Fragment Offset:** this field contains a value that specifies each fragment's place in the datagram with respect to other fragments. The value of this field gives total size of data fields in the previous fragments measured in 8-bytes. The first fragment has a value of 0 in this field, and the value in the second fragment is the size of data field in the first fragment. The third fragment's offset value is the size of data fields in the first two fragments, and so forth. The destination system uses these values to reassemble the fragments in the proper order. It should be noticed that in non fragmented datagram the value of fragment offset filed must be (0).

Note: the values of Total length, TTL and Checksum fields must also change when a datagram is fragmented.

Example:

This example shows the fragmentation process of a 4116 Bytes datagram that has no option fields. The datgram is forwarded by the router from a Token Ring network to an Ethernet network. The datagram need to be fragmented so that it can encapsulated in Ethernet frames that has MTU equals 1500. The example shows the fragmentation process and focuses on Total length, Identification, Flags, and Fragment Offset fields.



Protocol Identification

For the destination system to process the incoming datagram properly, it must know which protocol generated the information carried in the Data field. The most commonly used values for the Protocol field are as follows:

- 6 TCP
- 17 UDP

Checksum:

Checksum is the method used by IP for error detection. In IP, checksum covers only the header of the datagram. This may be for two reasons; first upper layer protocols that delivers data for IP usually have checksums and the second reason is that only the header of the datagram may be altered with each router visited by the datagram. Checksum is calculated at the original source of the datagram, its final destination and intermediate systems (such as routers).

Each system sends the datagram calculates the checksum value as follows:

- First, the checksum value is set to (0).
- The header of the datagram is divided into 16-bit sections.
- The 16-bit sections are added together.
- The summation result is complemented and transferred to the checksum field.

While the system receives datagram checks it as follows

- The header of the datagram is divided into 16-bit sections.
- The 16-bit sections are added together.
- The summation result is complemented, the result must be (0) for a correct datagram; else the datagram is rejected.

Example:

For the following datagram that has no option fields; calculate the checksum value.

<table>
<tr><td>4</td><td>5</td><td>0</td><td colspan="2">28</td></tr>
<tr><td colspan="3">1</td><td>0</td><td>0</td></tr>
<tr><td colspan="2">4</td><td>17</td><td colspan="2">Checksum ?</td></tr>
<tr><td colspan="5">10.12.14.5</td></tr>
<tr><td colspan="5">12.6.7.9</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>

Sol:

4, 5, 0	→	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0
28	→	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
1	→	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0 , 0	→	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4, 17	→	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1
Checksum (0)	→	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10, 12	→	0	0	0	0	1	0	1	0	0	0	0	0	1	1	0	0
14, 5	→	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0	1
12, 6	→	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0
7, 9	→	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	1
Sum		0	1	1	1	0	1	0	0	0	1	0	0	1	1	1	0
Checksum		1	0	0	0	1	0	1	1	1	0	1	1	0	0	0	1

IP Options

IP options are additional header fields that enable datagrams to carry extra information and, in some cases, accumulate information as they travel through an internetwork on the way to their destinations. Some of the options defined in the IP standard are as follows:

○ **Loose Source Route:** This option contains a list of router addresses that the datagram can use as it travels through the internetwork. The datagram also can use other routers in addition to those listed.

○ **Strict Source Route:** This option contains a complete list of the router addresses that the datagram must use as it travels through the internetwork. The datagram cannot use any routers other than those listed.

○ **Record Route:** This option provides an area in which routers can add their IP addresses when they process the datagram.

○ **Timestamp:** This option provides an area in which routers can add timestamps indicating when they processed the datagram.

Laith Wajeeh