

Internet Protocol (IP)

Kameswari Chebrolu

All the figures used as part of the slides are either self created or from the public domain with either 'creative commons' or 'public domain dedication' licensing. The public sites from which some of the figures have been picked include:

<http://commons.wikimedia.org> (Wikipedia, Wikimedia and workbooks); <http://www.sxc.hu> and <http://www.pixabay.com>

Recap

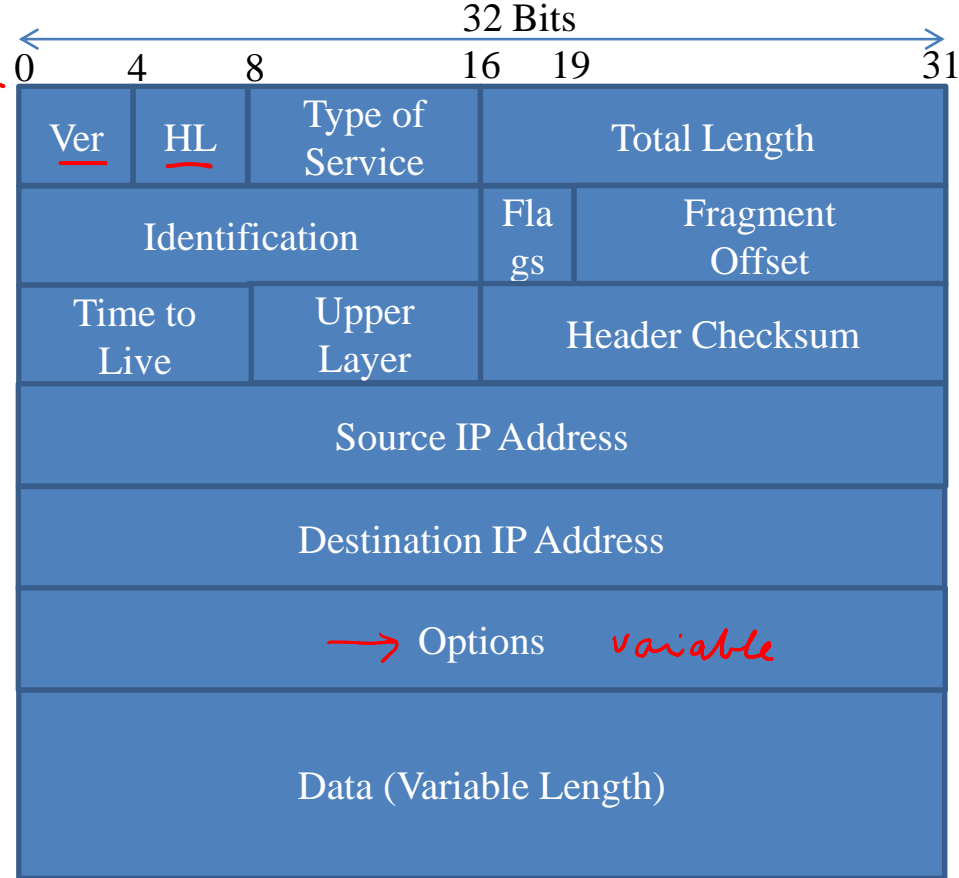
- IP Protocol: Needed functionality
 - IP Protocol: Packet format, addressing
 - Forwarding
 - Routing
 - Error reporting and host signaling
- Focus: Packet Format and Fragmentation and re-assembly

Packet Format

- Version: Specifies the version of the protocol
 - IPv4, IPv6
- Header Length: Specifies the header in 32-bit words
 - 5 words (without options)

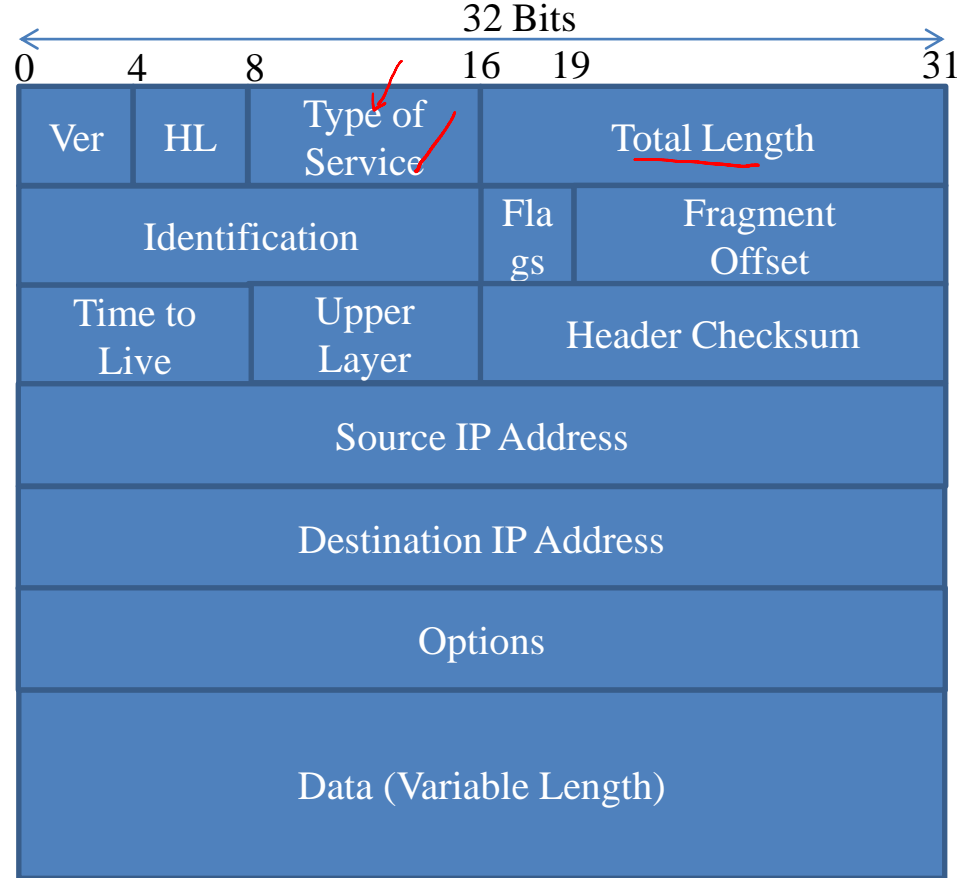
HL = 5

5×32
= 160 bits



Packet Format

- Type of Service:
Permits packets to be treated differently
 - Research Focus
- Total Length: Specifies the length of the datagram (in bytes) including header

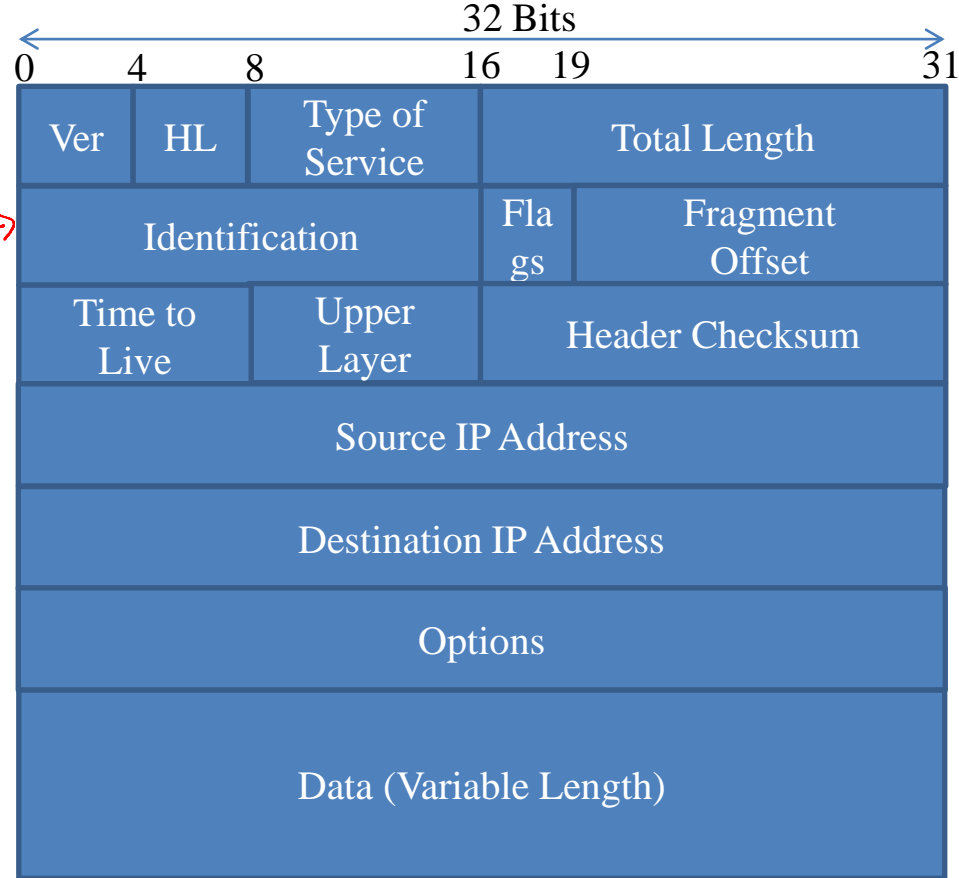


Packet Format

$(2^{14} - 1)$ byte
16

- Identification/Flags/
Fragment Offset:

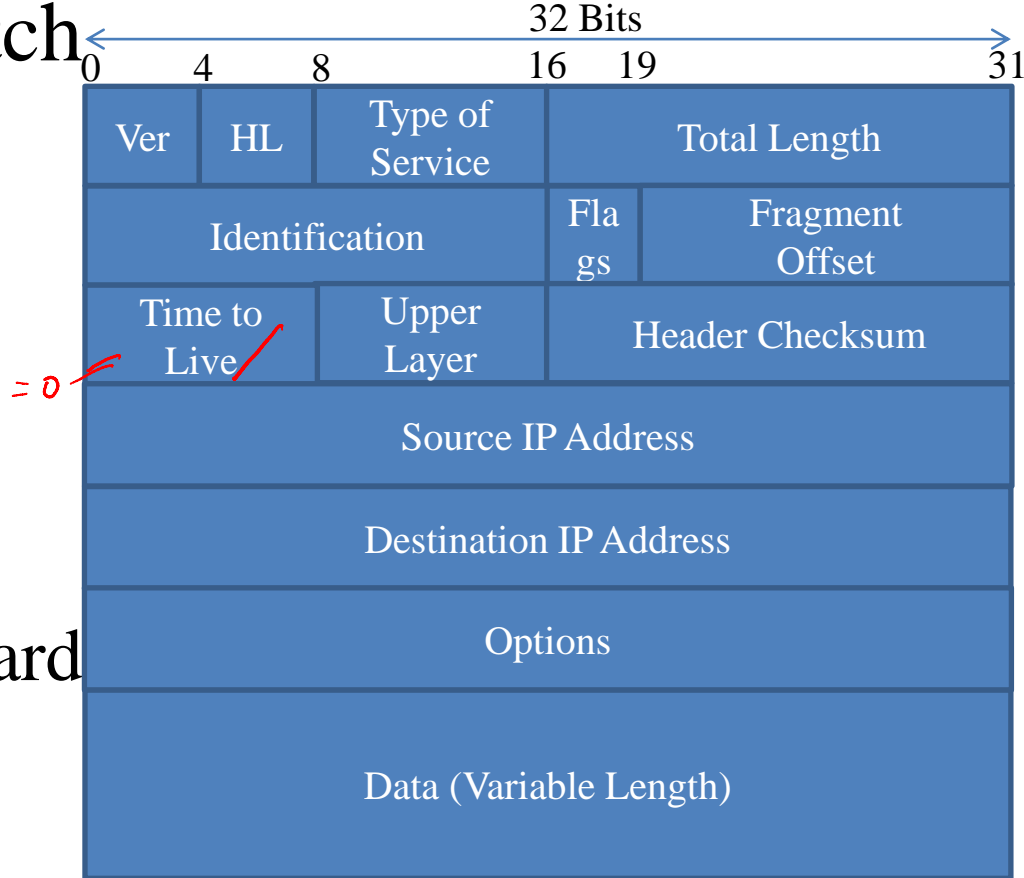
- Max size of IP packet is 65535 Bytes
- Physical Networks may not support large packets
- Need Fragmentation and reassembly (more on it soon)



Packet Format

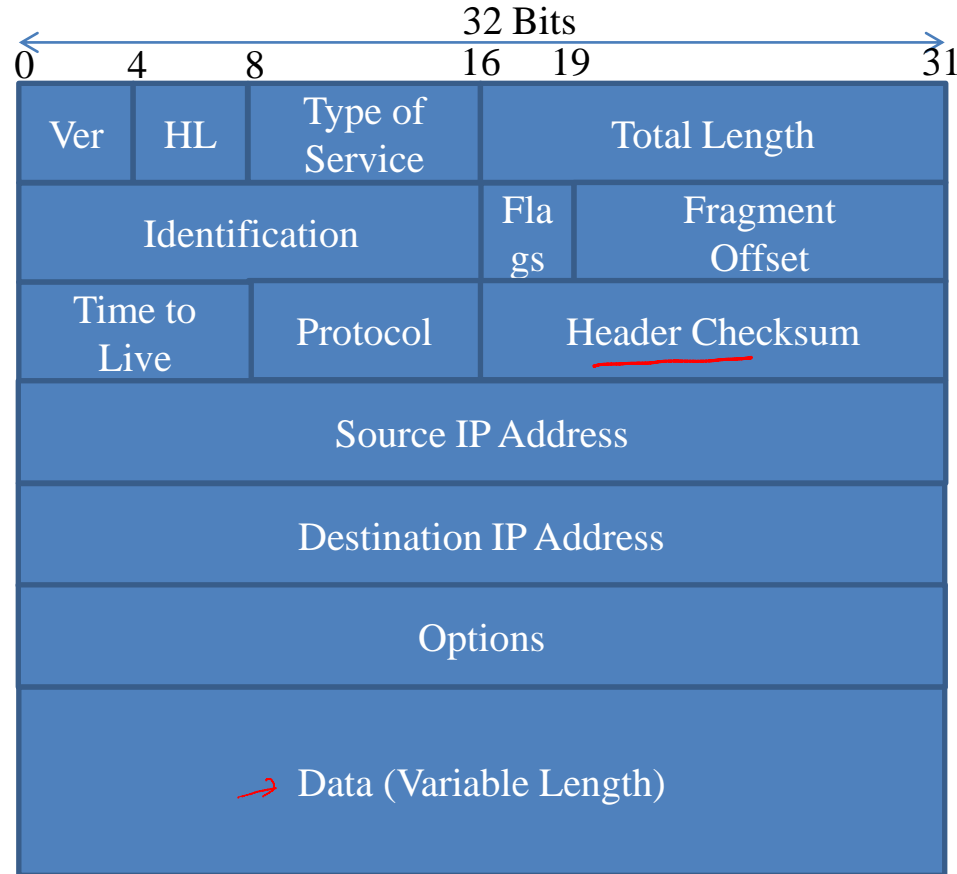
- Time to Live: Helps catch packets doing rounds

- Not really time but ^{expiry date} ~~hop~~ ~~sec~~ count
- Routers decrement the field by one before forwarding; if zero discard
- Default value = 64



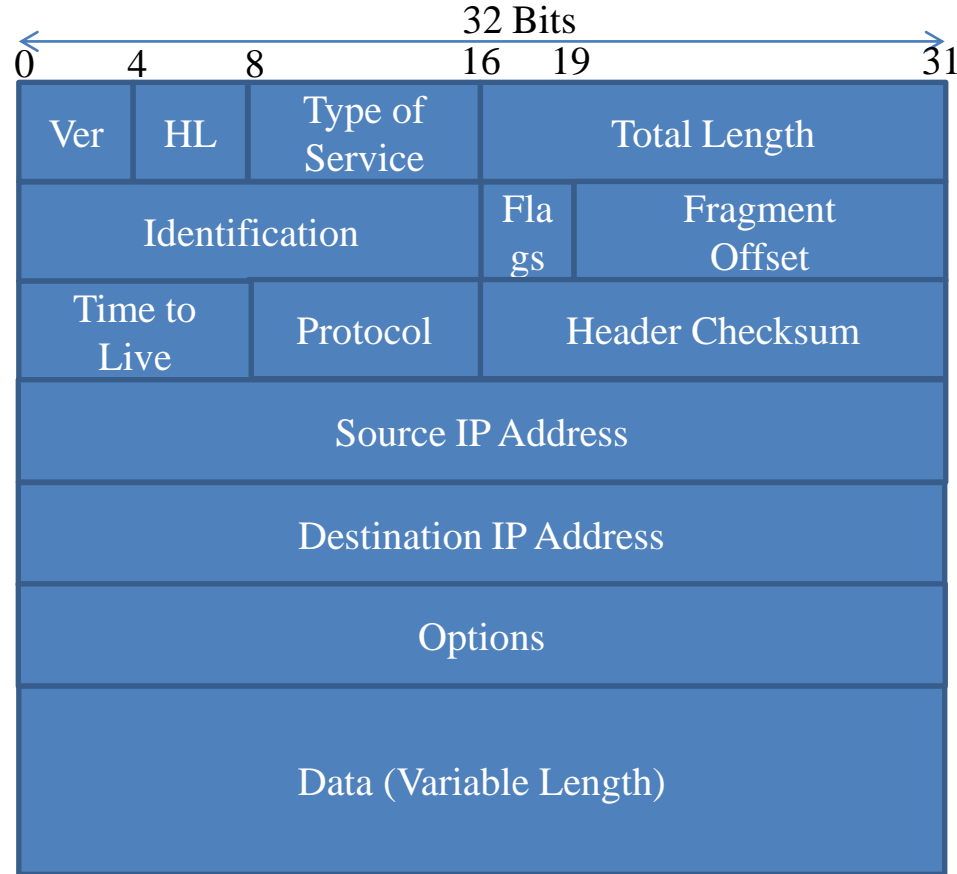
Packet Format

- Protocol: Demux key that identifies higher layer protocol
 - TCP: 6, UDP: 17
- Checksum (Internet): Detects errors in header



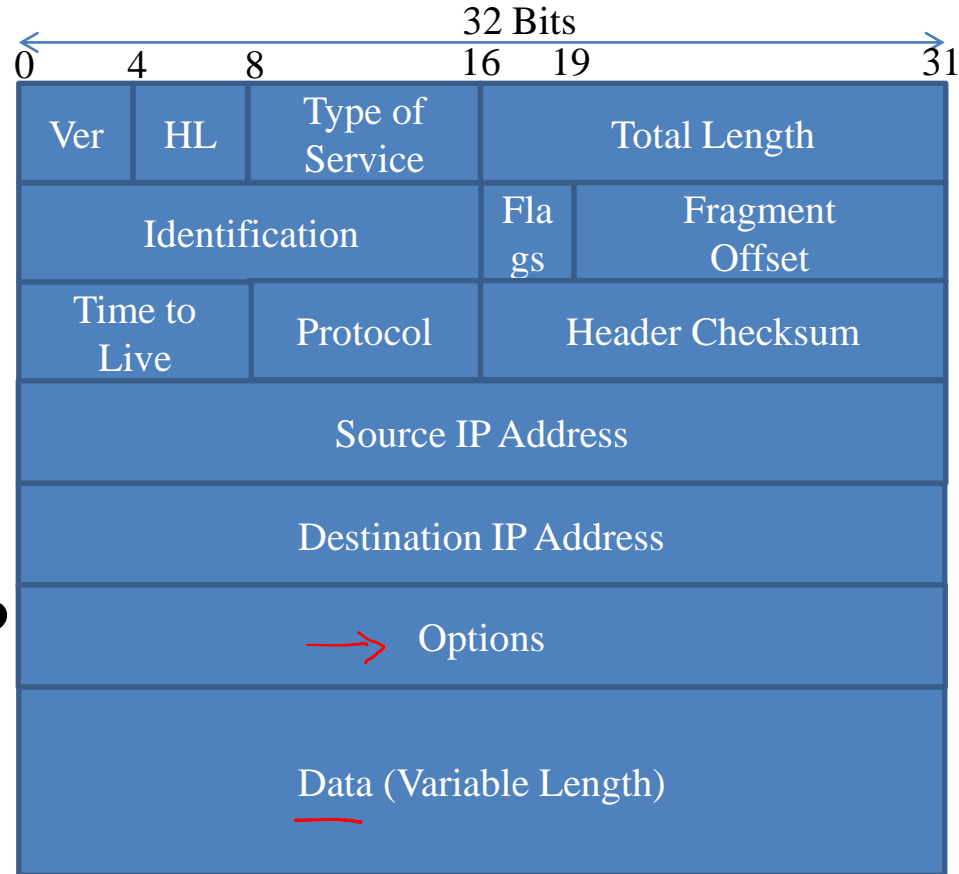
Packet Format

- Source/Destination IP address: 32-bit
 - Destination key to forwarding
 - Source for replying back
 - Global address space, independent of physical network address (MAC)



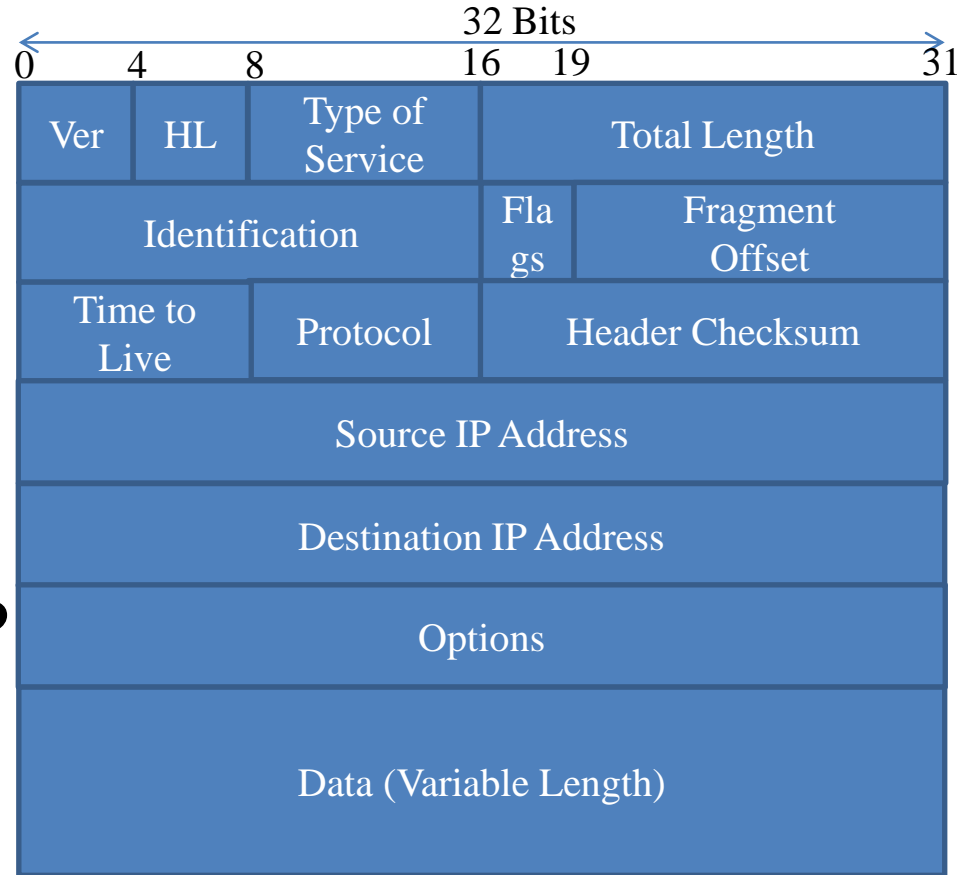
Packet Format

- Options: Rarely used
 - Record Time stamp
 - Record route taken
 - Specify source route
- Data/Payload: Higher Layer Data (TCP or UDP segment)



Packet Format

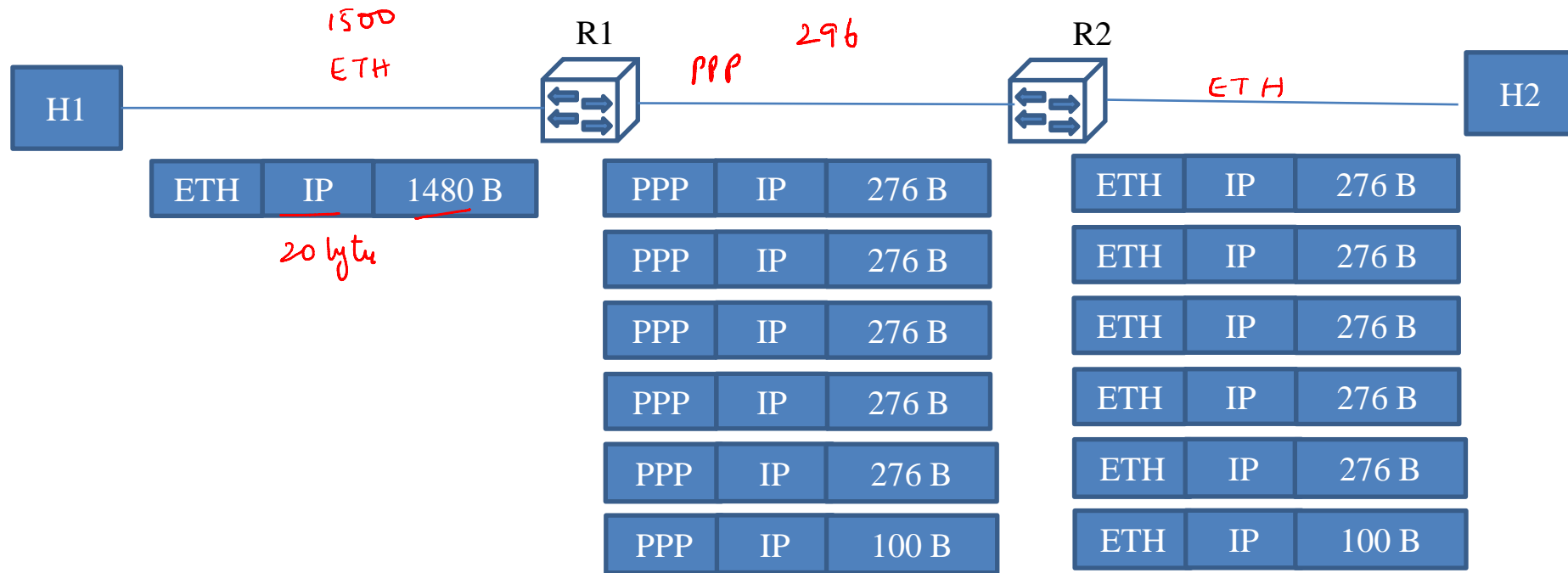
- Options: Rarely used
 - Record Time stamp
 - Record route taken
 - Specify source route
- Data/Payload: Higher Layer Data (TCP or UDP segment)



Fragmentation and Reassembly

- Goal: Interconnect heterogeneous networks
- Problem: Each technology has different Maximum Transmission Unit (MTU) size
 - MTU: Largest IP datagram that can be carried in a frame
 - E.g. Ethernet: 1500, FDDI: 4352, PPP: 296 (Negotiable), WiFi: 7981

- At host: Select MTU of link it is connected to
- At intermediate router: Forward datagram on a network with smaller MTU
 - Need to fragment the datagram
- Where to reassemble?
 - Next hop router? Increases overhead and datagram may again be fragmented
 - Destination is the best place



Note: Above values not true in practice due to Offset field having to be a multiple of 8

Fragmentation Fields

- Identification: Helps identify a datagram
 - All fragments carry same identification

- Flags: 3 bits



- bit 0: Reserved, set to zero
- bit 1: Don't Fragment (DF); Useful for path MTU discovery
- bit 2: More Fragments (MF); Set to one to indicate more fragments to follow

Fields

- Fragmentation Offset: 13 bits long
 - Measures data/payload in units of eight-byte blocks
 - For a particular fragment, offset specifies start of data relative to the beginning of the original unfragmented IP datagram.
 - E.g first fragment would have an offset of zero



Example

- Original datagram:
1500B

→ TCP/UDP

20B

- Data within is 1480B

- MTU: 296B

276

- Max Data within is
276B
- Offset has to be multiple
of 8 → Data within
272B

Original Datagram

Length= <u>1500</u>	ID= <u>x</u>	Fragflag= <u>0</u>	Offset= <u>0</u>
---------------------	--------------	--------------------	------------------

8 bytes
↓

$$272 \times 5 = 1360$$

$$1480 - 1360 = 120$$

Fragmented Datagrams

20

Length= <u>292</u>	ID= <u>x</u>	Fragflag= <u>1</u>	Offset= <u>0</u>
--------------------	--------------	--------------------	------------------

<u>272</u> + 20 Length=292	ID= <u>x</u>	Fragflag= <u>1</u>	Offset= <u>34</u>
-------------------------------	--------------	--------------------	-------------------

$$272 / 8$$

Length=292	ID= <u>x</u>	Fragflag= <u>1</u>	Offset= <u>68</u>
------------	--------------	--------------------	-------------------

Length=292	ID= <u>x</u>	Fragflag= <u>1</u>	Offset= <u>102</u>
------------	--------------	--------------------	--------------------

<u>Length=292</u>	ID= <u>x</u>	Fragflag= <u>1</u>	Offset= <u>136</u>
-------------------	--------------	--------------------	--------------------

7

<u>Length=140</u>	ID= <u>x</u>	Fragflag= <u>0</u>	Offset= <u>170</u>
-------------------	--------------	--------------------	--------------------

7

Summary

- Looked at IP packet format
- Looked at fragmentation and reassembly
- Next: Addressing and Forwarding