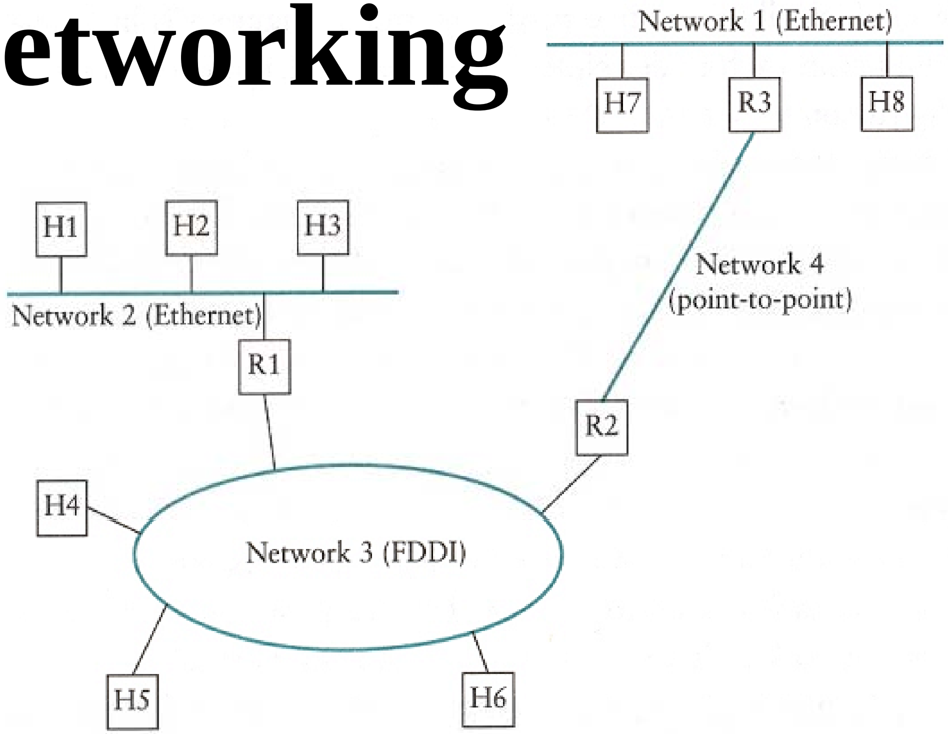
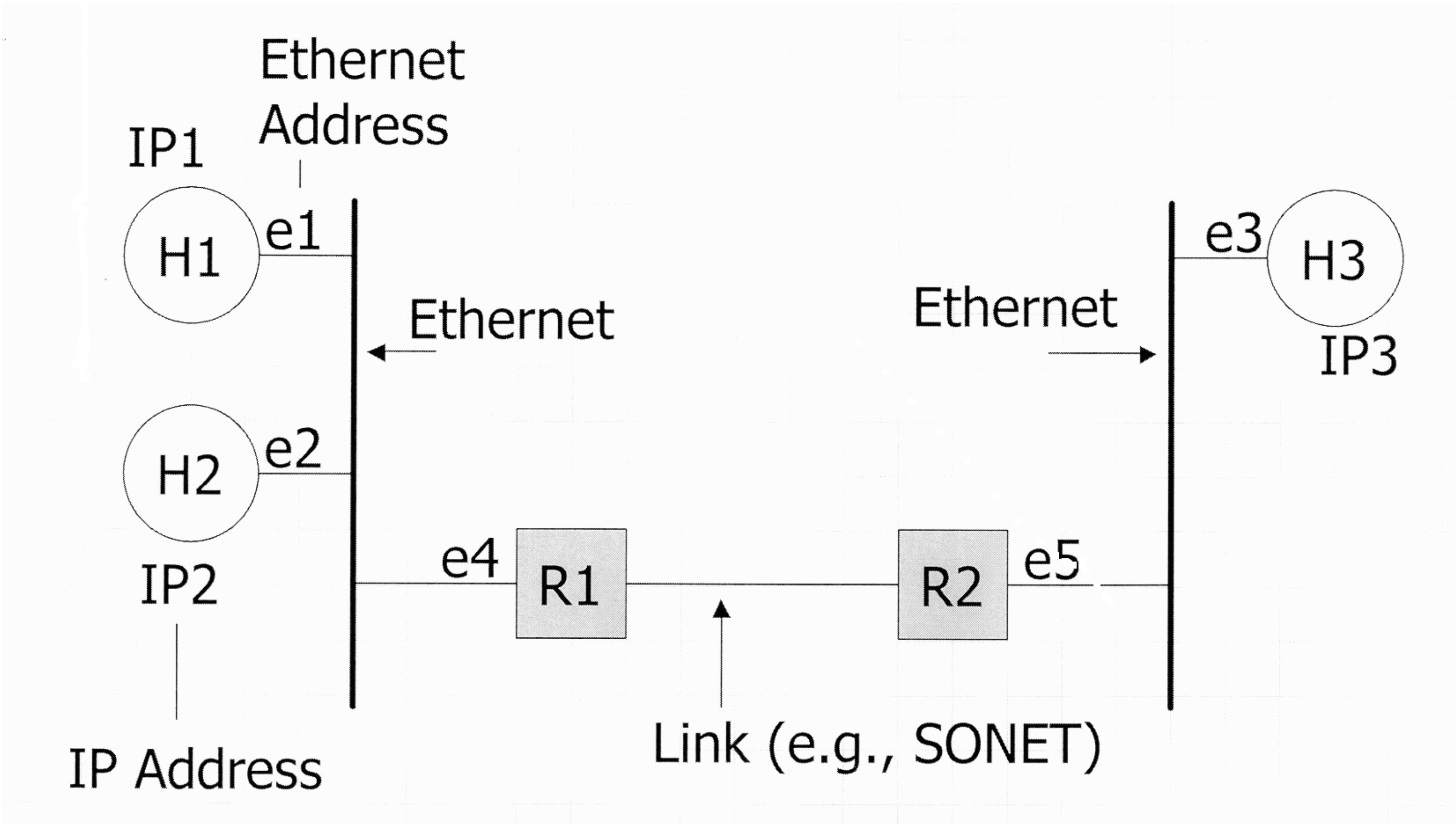
Internetworking

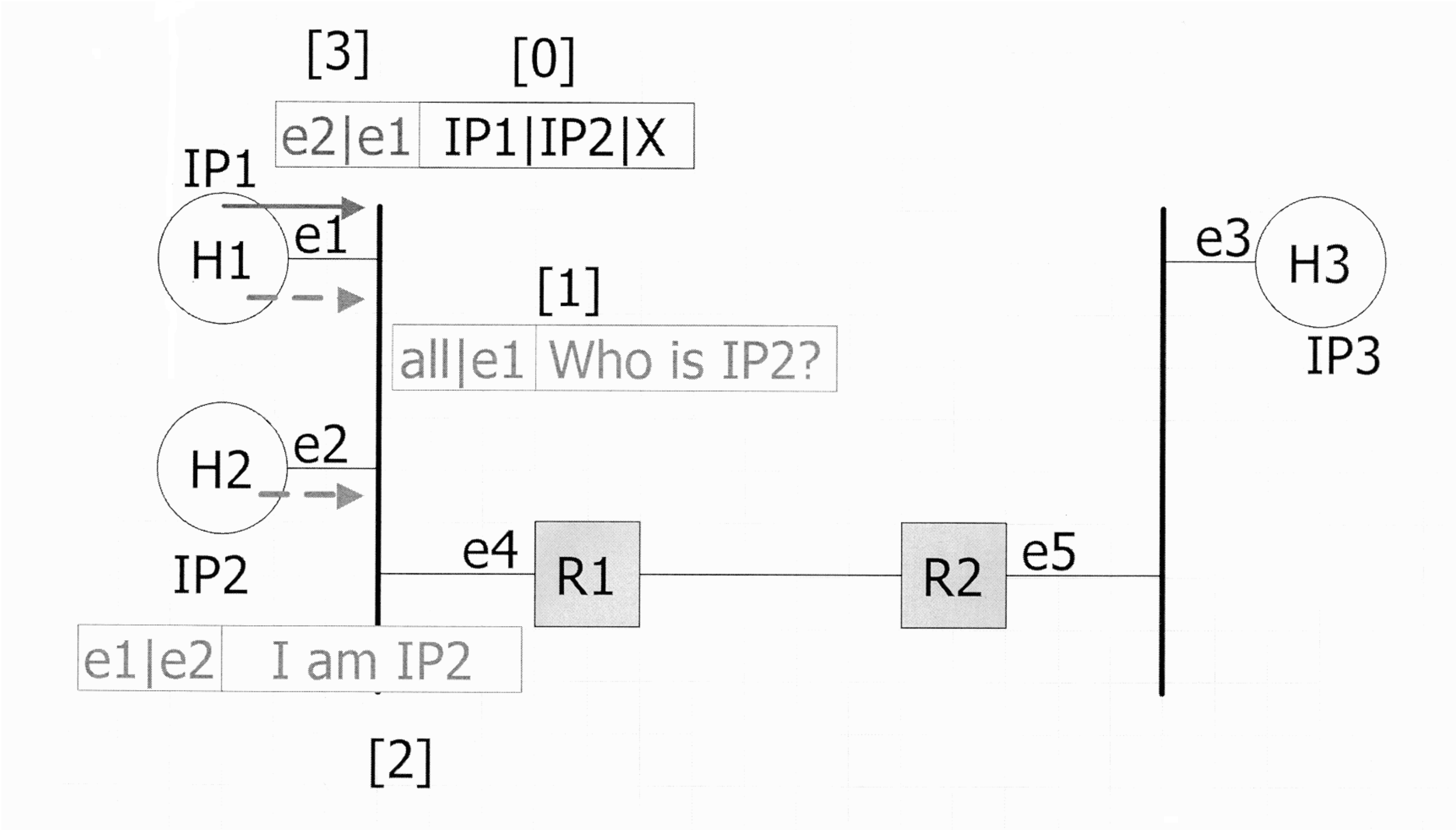


Contents

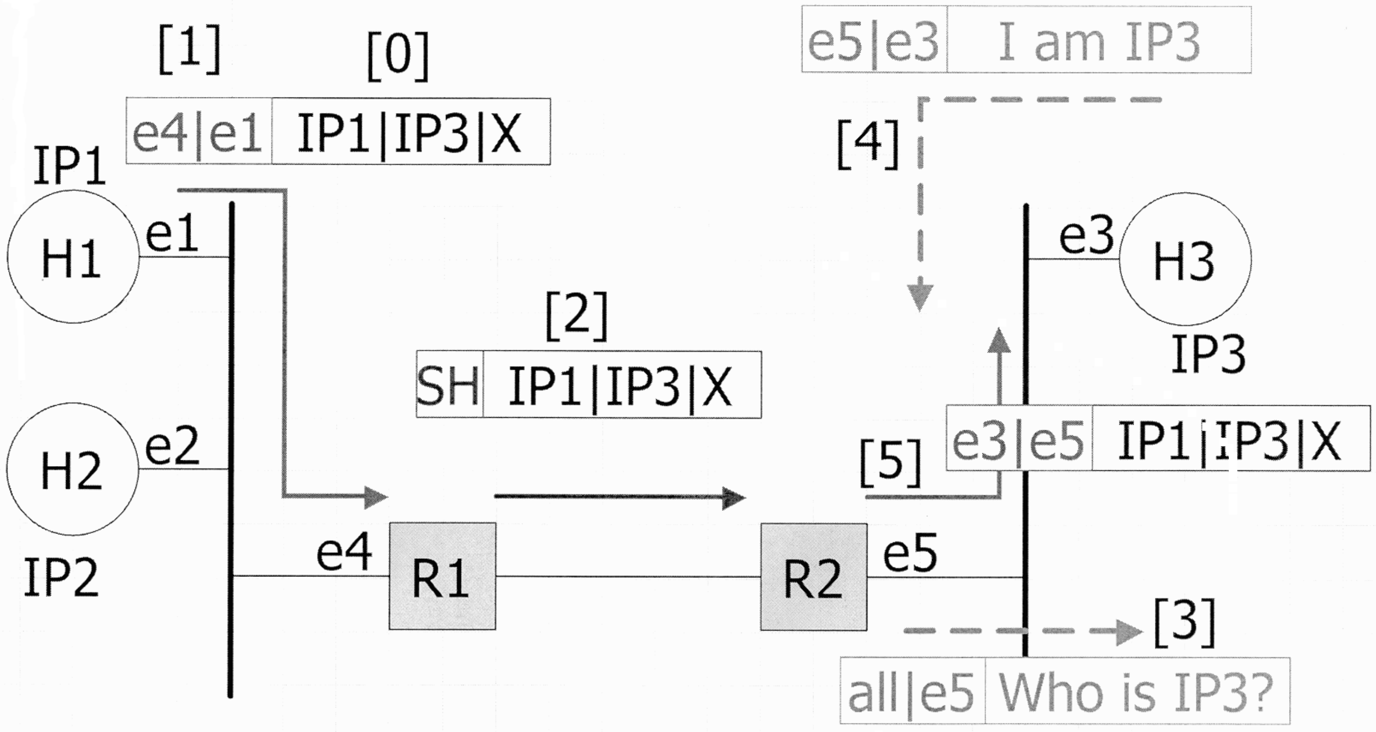
Overview - Example



Direct Delivery



Indirect Delivery



Note: Fragmentation may be required at R1

Internetworking: Functions

Datagram delivery *between* networks

Routers touch two or more networks, forward network-layer datagrams between them (routers use layer 3)

Routers execute *routing protocols* to learn how to reach destinations

Internetworking: Issues

Network layer provides end-to-end delivery (routing)

Provides consistent datagram abstraction:

- Best-effort delivery
- No error detection on data
- Consistent max. datagram size
- Consistent global addressing scheme

Internetworking: Issues

Link layer networks provide delivery within the same network

Typically includes its own addressing format (e.g. Ethernet), and maximum frame size (MTU)

Internetworking requires a consistent view of the basic delivery unit (datagram)

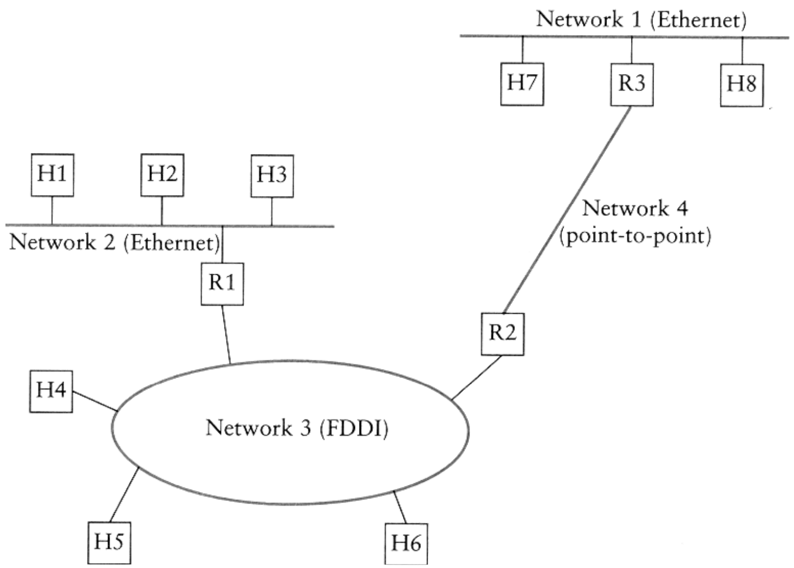


Figure 4.1 A simple internetwork. H*n* = host; R*n* = router.

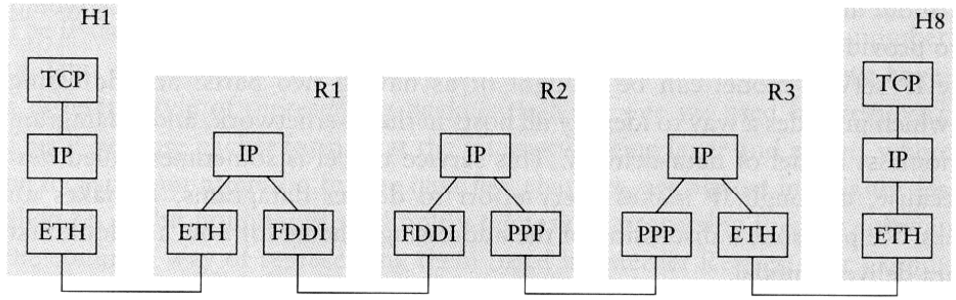


Figure 4.2 A simple internetwork, showing the protocol layers used to connect H1 to H8 in Figure 4.1. ETH is the protocol that runs over Ethernet.

Basic Delivery Unit

- Address adaptation
  - Mapping from Internet standard addresses (IP addresses) to link-specific addresses
- Datagram size adaptation
  - Internet datagram has universal common size (64K Byte for IP)
  - Mapping from common size to link-specific MTU requires fragmentation

Addressing

- IP addresses are topologically sensitive
  - Interfaces on same network share prefix
  - Prefix is assigned via ISP/net admin
  - 32-bit globally unique
- IEEE 802.X addresses are vendor-specific
  - Interfaces made by same vendor share prefix
  - 48-bit globally unique

Datagram Delivery

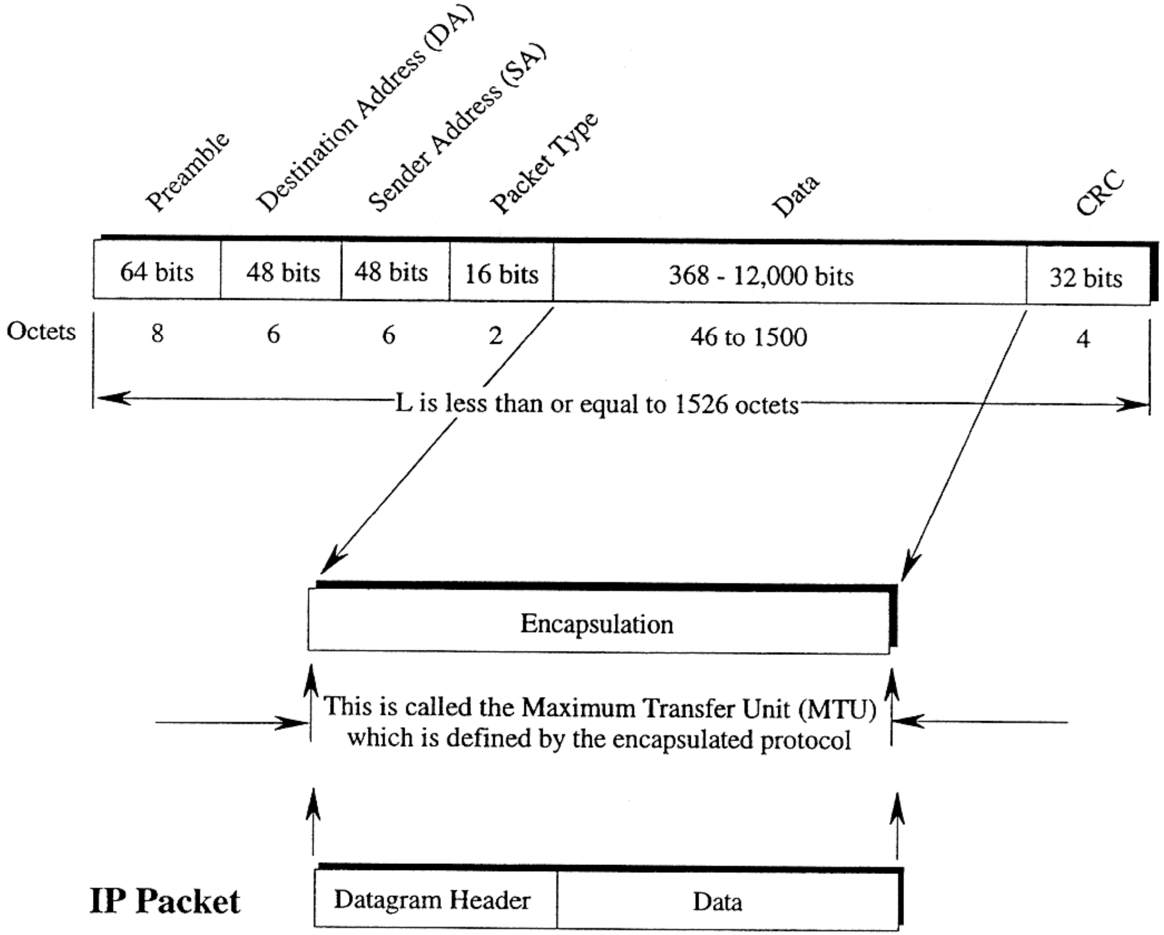
- Two types of delivery:
  - Local delivery (no router involved)
  - Non-local delivery (router needed)
- Local delivery
  - On multi-access LAN, requires MAC address!

Address Mapping

For local delivery, need to map network-layer address to link-layer address:

- Consider 128.32.15.6/24 and 128.32.15.18/24? [on same network]
- Encapsulate IP datagram within link-layer frame
- What destination MAC address to use?

DATA ENCAPSULATION IN AN ETHERNET FRAME



IP Packet

Address Mapping: IP to MAC

- Could just broadcast everything
  - Burdens uninterested stations with others' traffic
- IP to MAC address mapping
  - Configured by hand [cumbersome]
  - Dynamic [learned by system automatically]

Address Mapping IP to MAC: Learning

Dynamic approach

- Each station runs Address Resolution Protocol (ARP)
- Client/server architecture, each station is both client and server [routers too]
- Cache lookups with timeouts on each resolution

Address Resolution Protocol (ARP)

Basic protocol is address independent (at both network and link layer)

Protocol is specialized for each particular network/link address pairing

Common example is IPv4/Ethernet

ARP Operations

Requesting station A has IP address I, wants the associated MAC address M

A **broadcasts** query: who has I? tell A

Machine assigned address I responds directly to A with its MAC address M

A adds the (I,M) entry to its ARP cache

ARP Operations: Observations

A cannot communicate with station using IP address I until it knows M

ARP enables direct local delivery

For indirect delivery, will need MAC address of router (also uses ARP)

Isolates Internet layer from link layer

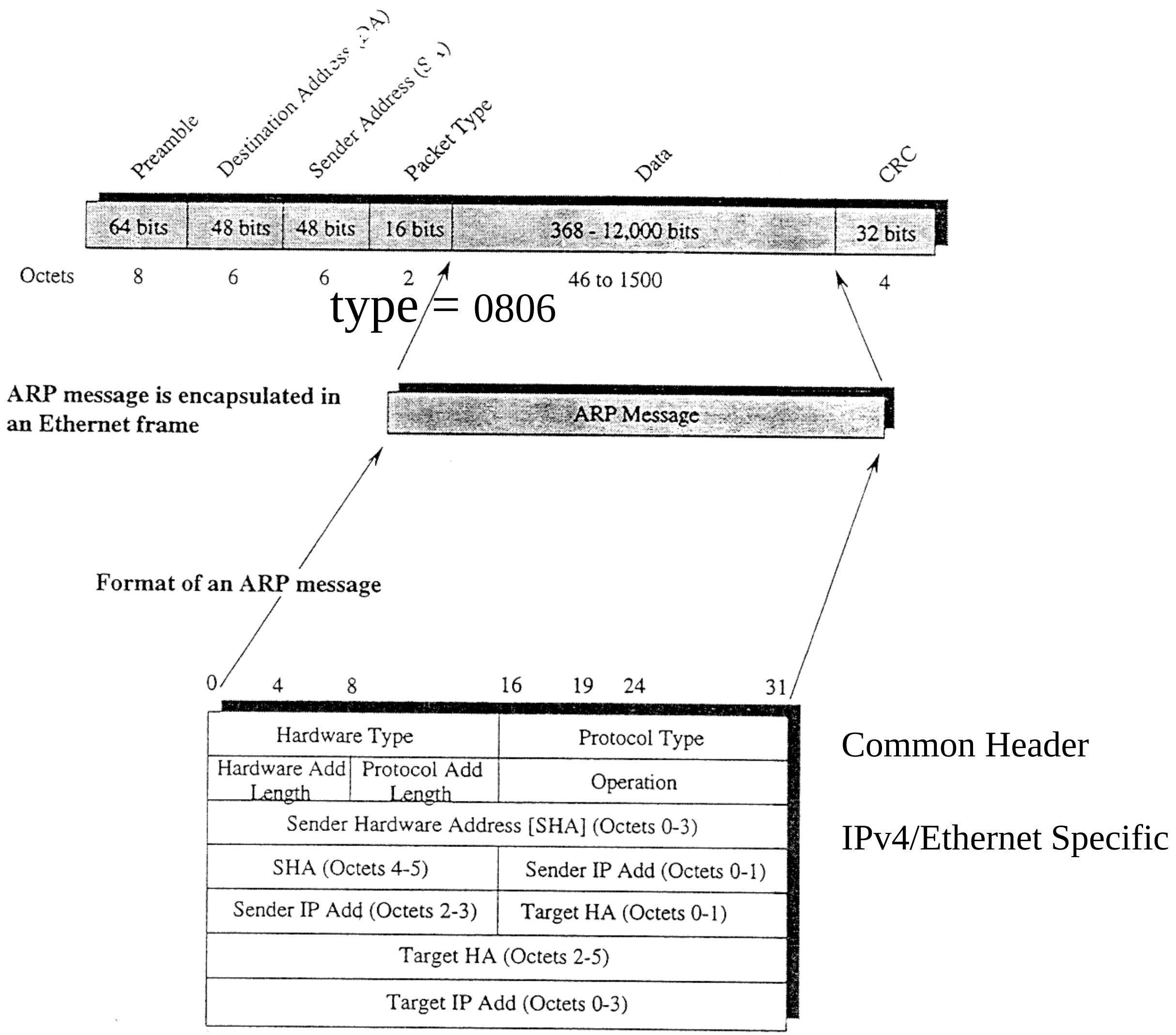
ARP requires broadcast delivery

ARP Operations: Timers

ARP Cache timeout

- Similar issues to bridge station caches
- Could be stale info if MAC address changes
- RFC recommends 20 minute timeout

ARP ENCAPSULATION



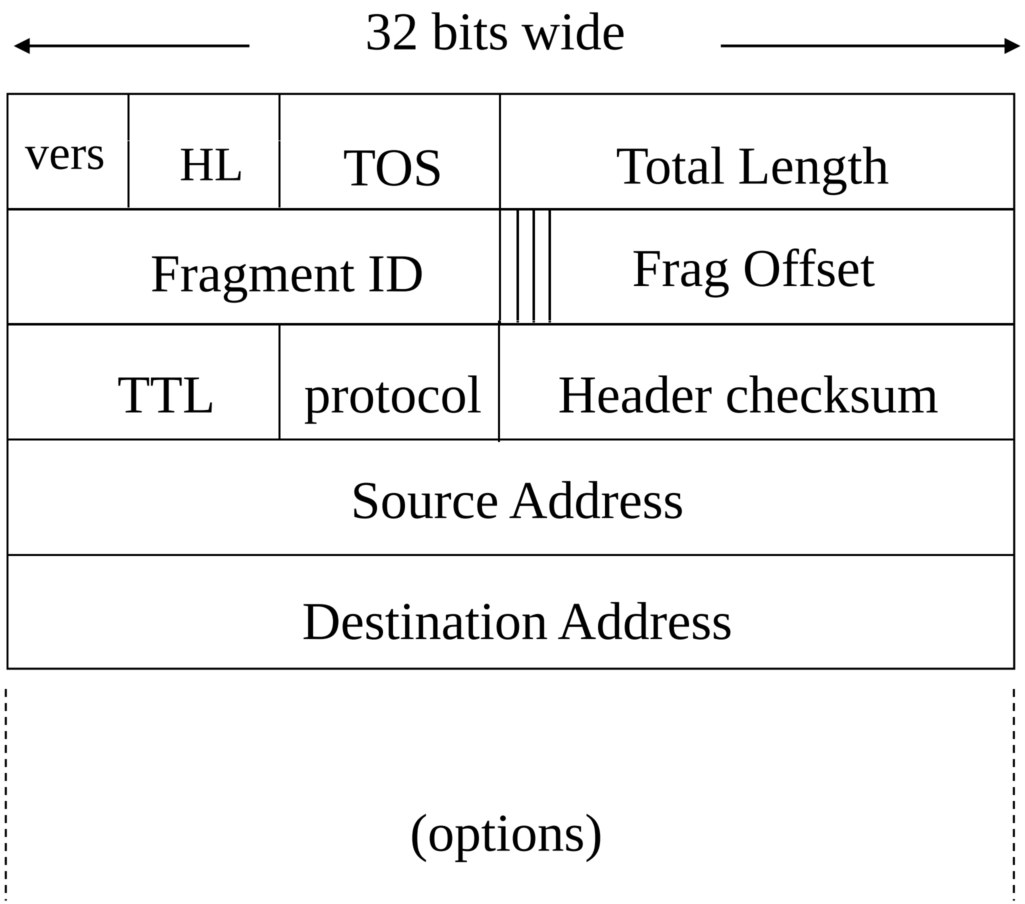
ARP: Other ARP Uses

- Proxy ARP
  - One machine responds to ARP requests on behalf of others
  - HA impersonates the MN in MIP
- Gratuitous ARP
  - The proxy ARP message is not a response to a normal ARP request
  - HA broadcasts IP and Ethernet addresses of MN

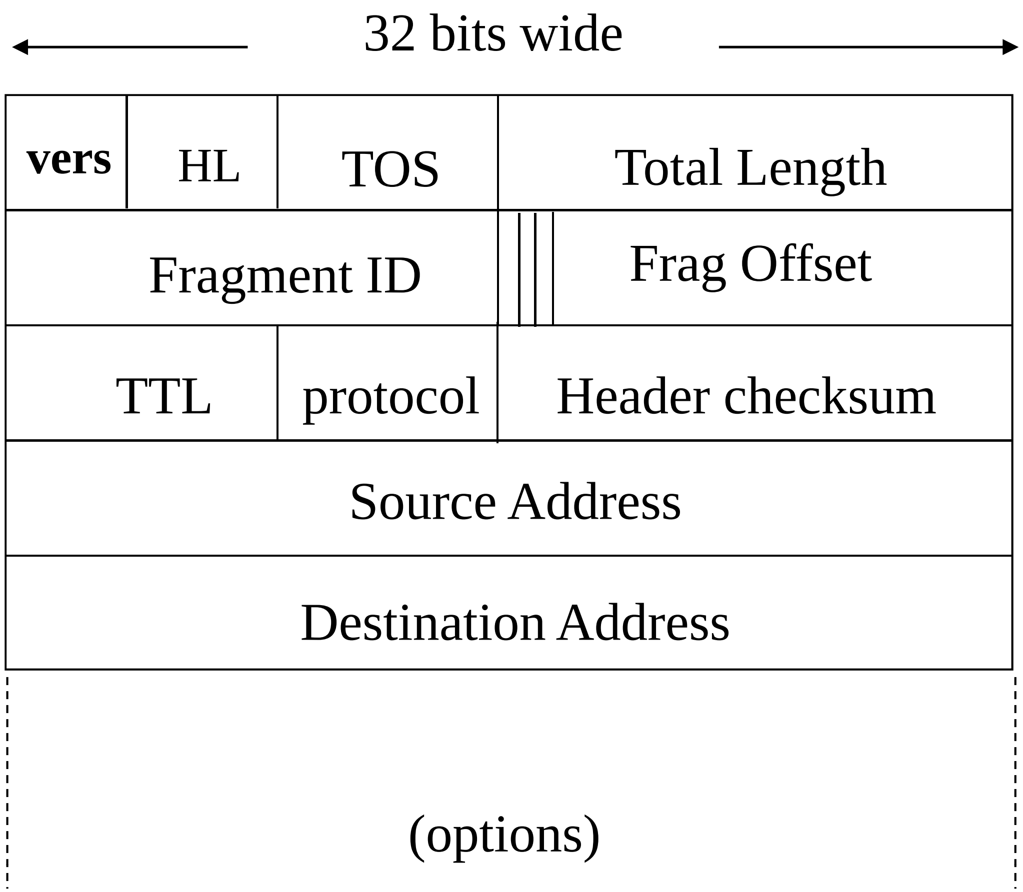
Internet Protocol Details (IP)

- IP version 4 is current, IPv6 forthcoming
- Protocol header includes:
  - version, source and destination addresses, lengths (header, options, data), header checksum, fragmentation control, TTL, and TOS info
  - Today, TOS info often ignored

IPv4 Header

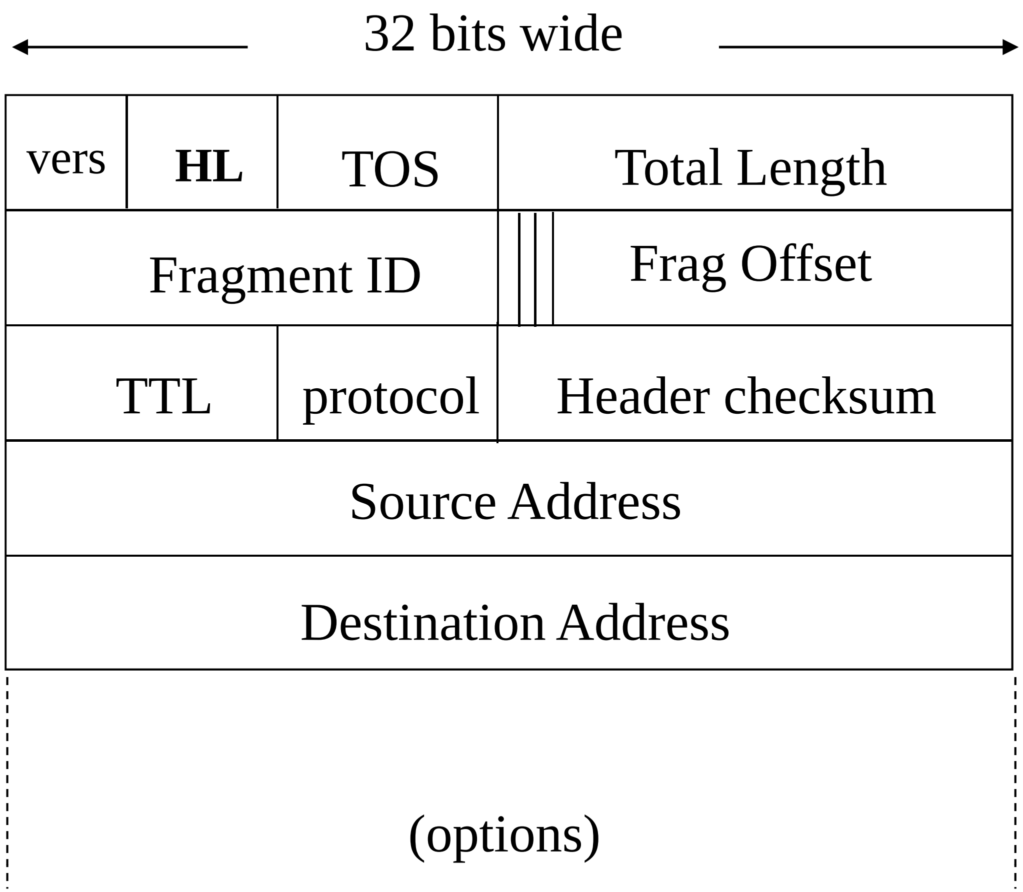

32 bits wide
vers
HL
TOS
Total Length
Fragment ID
Frag Offset
TTL
protocol
Header checksum
Source Address
Destination Address
(options)

IPv4 Header - vers



Version: 4 here
6 is for IPv6

IPv4 Header - HL



HL: # 32 bit words in IP header

- Min is 5 (20 bytes)
- Max is 15 (60 bytes)
- At most 40 bytes of options

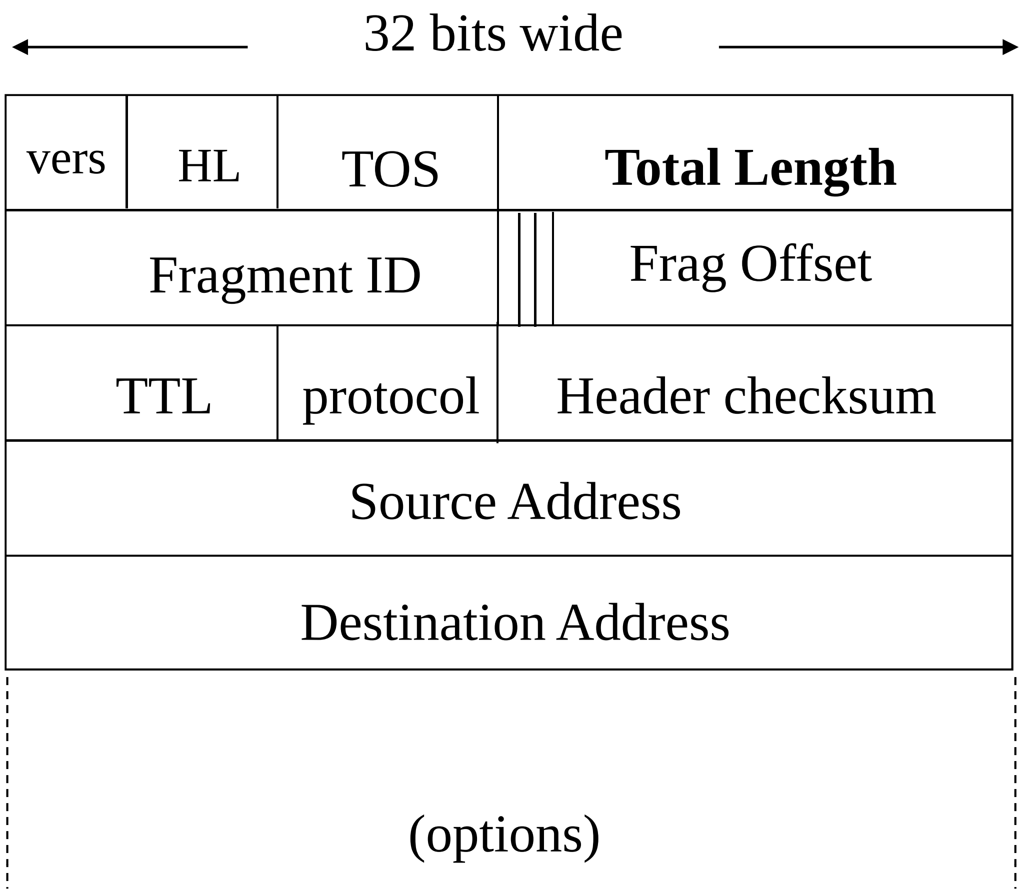
IPv4 Header - TOS

← 32 bits wide →

vers	HL	**TOS**	Total Length
Fragment ID			Frag Offset
TTL		protocol	Header checksum
Source Address			
Destination Address			
(options)			

TOS: type of service
- Abstract notion of type of service, including priority
- Most routers ignore
- Will be used by diff-service

IPv4 Header - Length



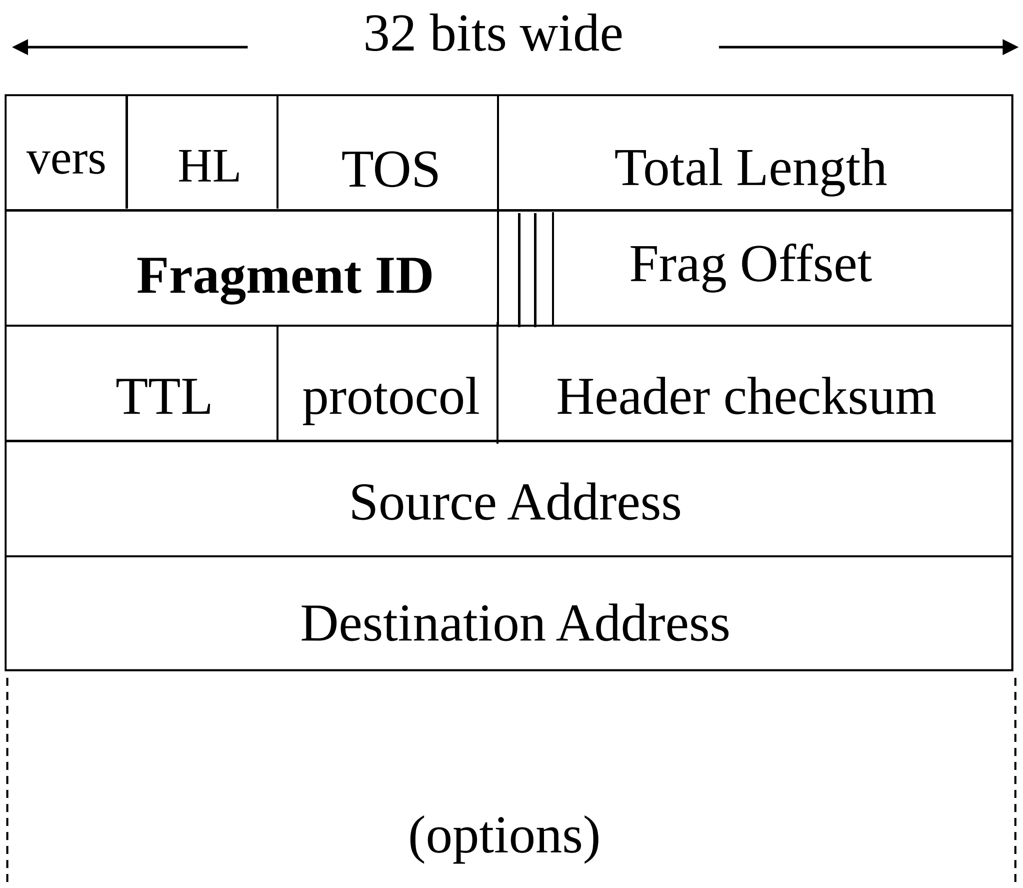
Length: # bytes in the datagram (fragment)

Min value is 20

Max value is 65535

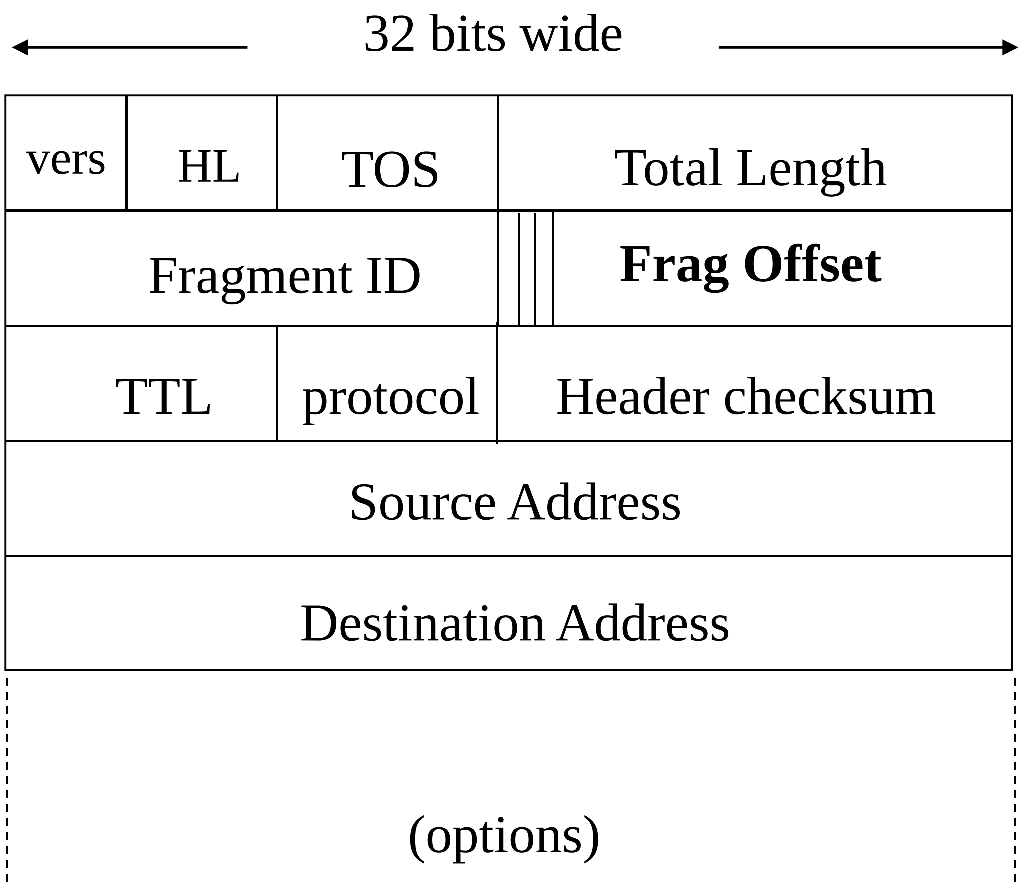
Limits max datagram size

IPv4 Header - Fragment ID



Fragment ID, set by original sender of datagram
Copied into each fragment during fragmentation

IPv4 Header – Offset/Flags

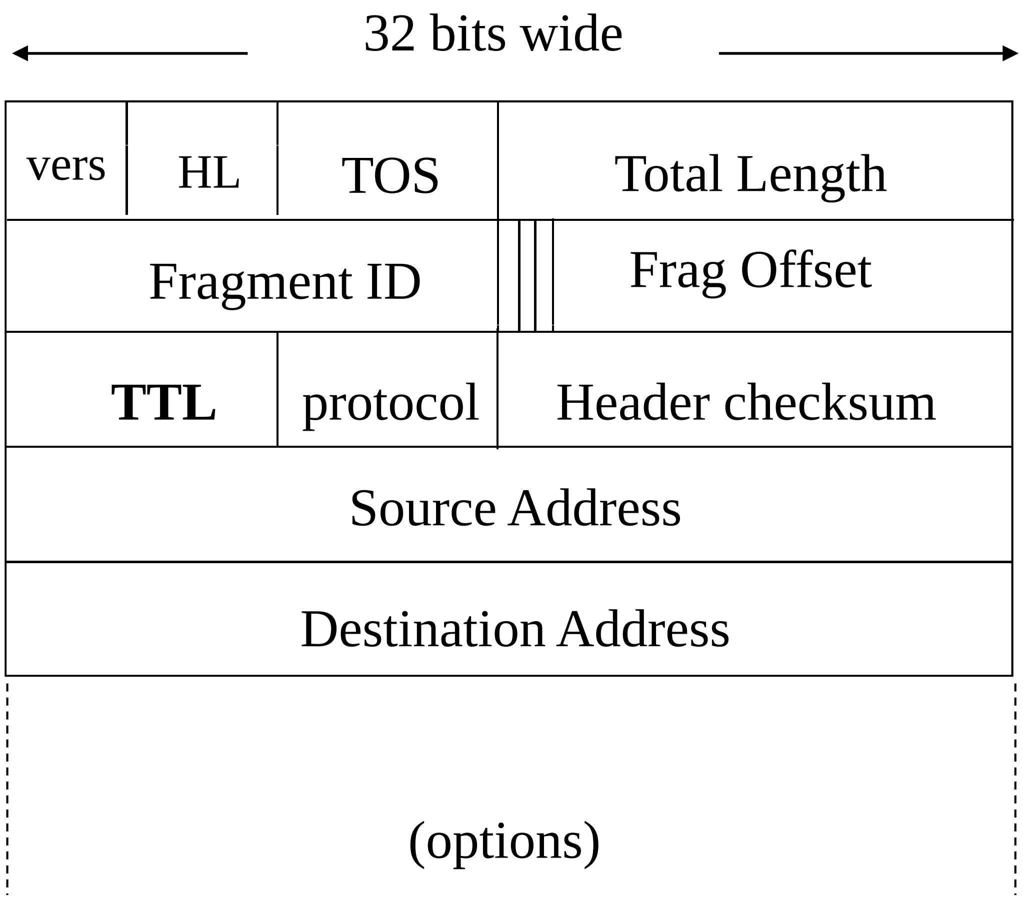


Offset: offset of this fragment into original datagram
If no fragment used, offset is zero

MF bit: more fragments to come, zero if last

DF bit: don’t fragment

IPv4 Header – TTL



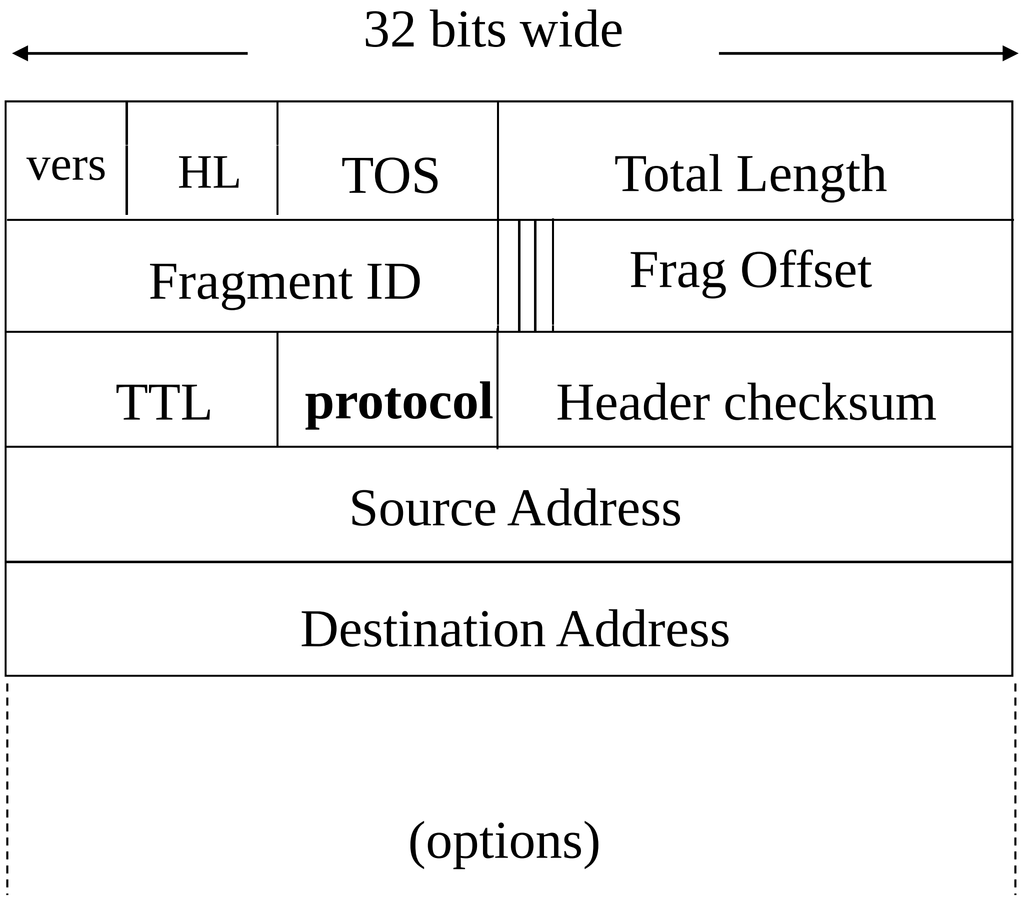
TTL: "Time to live"

each router must decrement by 1 and discard if zero

Also decremented if router holds for longer than 1 sec.

Prevents immortal datagrams

IPv4 Header - protocol



proto: protocol number identifies type of protocol contained within this datagram See assigned #s RFC [note: could indicate IP!]

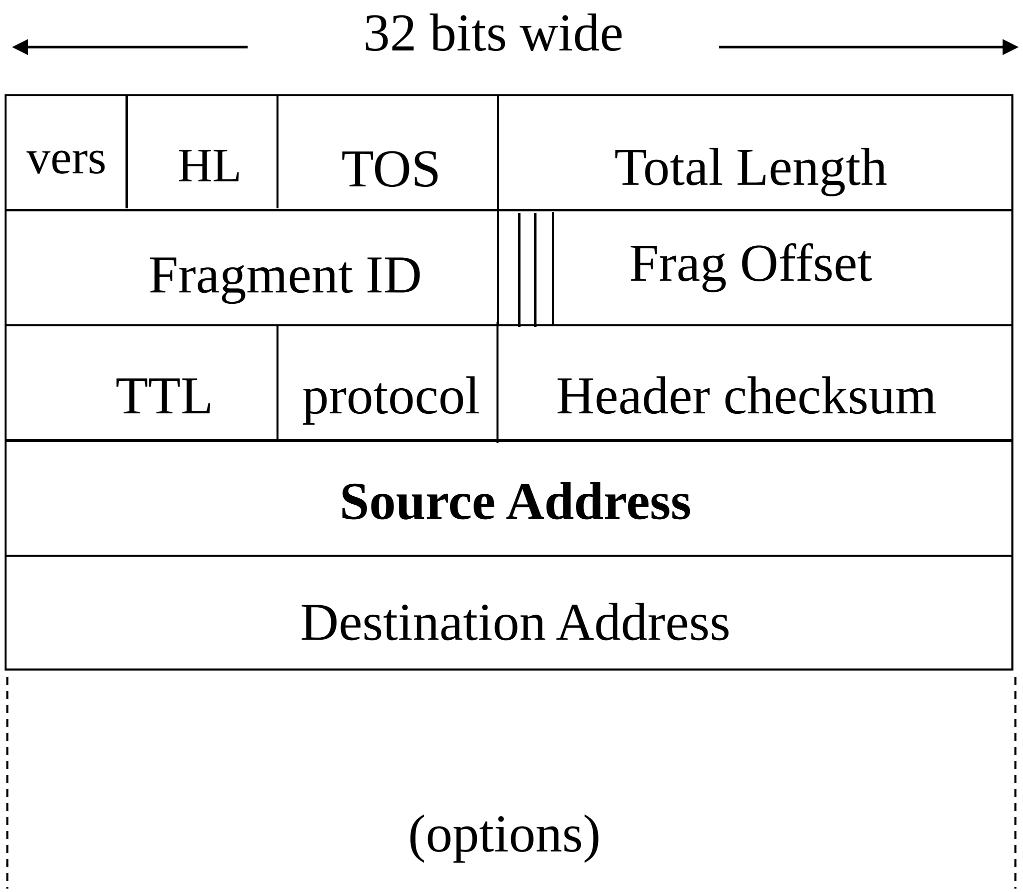
IPv4 Header - cksum

32 bits wide

vers	HL	TOS	Total Length
Fragment ID			Frag Offset
TTL		protocol	**Header checksum**
Source Address			
Destination Address			
(options)			

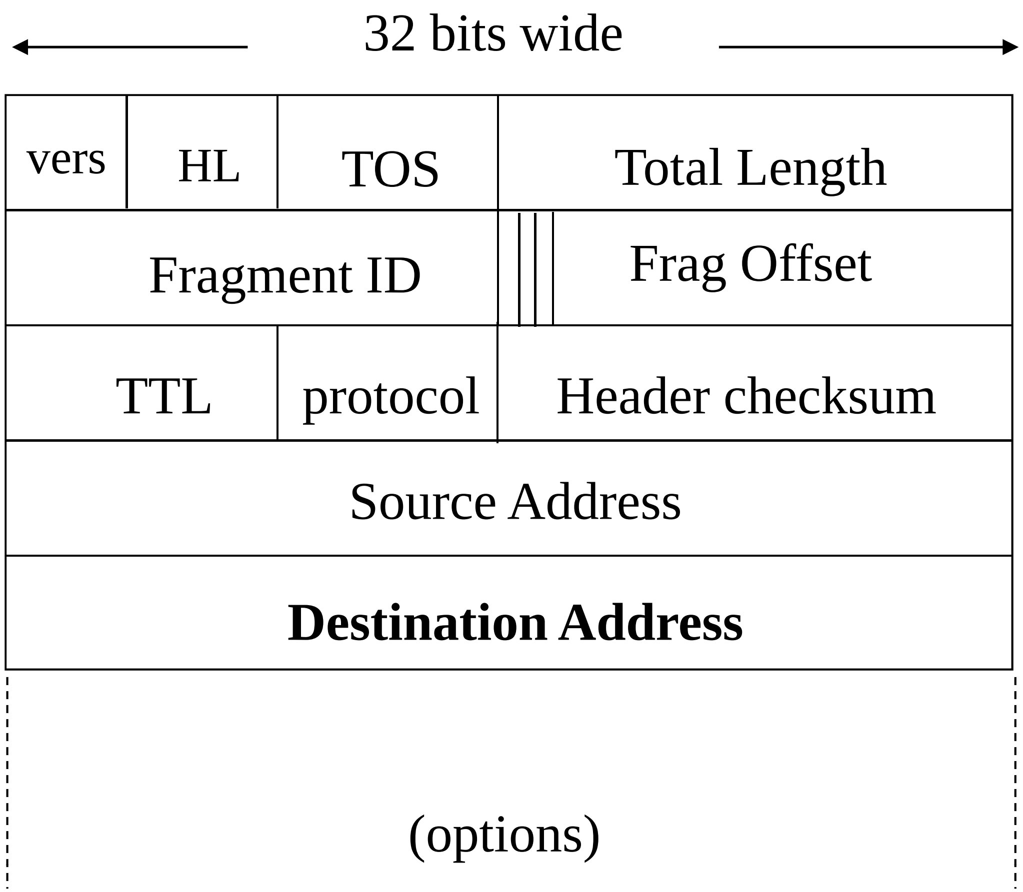
cksum: Internet checksum computed over full IP header
Does not cover data
Must change as datagram is routed due to TTL decrement (can be done incrementally)

IPv4 Header - Source



Source: sender's IP address
Never change during ordinary routing (Not authenticated)

IPv4 Header - Destination



Destination: receiver's IP address

Never changes during ordinary routing

Changes when source routing is used

IPv4 Header - Options

- Special handling for particular datagrams, sometimes don't take router's "last path" (40bytes of Option)
- Rarely used, but the more common are:
  - Loose Source Routing
  - Strict Source Routing
  - Record Route
  - Timestamp
- Most copied on fragmentation

Adapting Datagram Size

- IP datagrams: max 64KB, Ethernet frame: max 1500 payload bytes
- Fragmentation and Reassembly
  - Divide network-layer datagram into multiple link-layer units, all ≤ link MTU size
  - Reconstruct datagram at final station
  - Each fragment otherwise acts as a complete, routable datagram

Adapting Datagram Size: Fragmentation

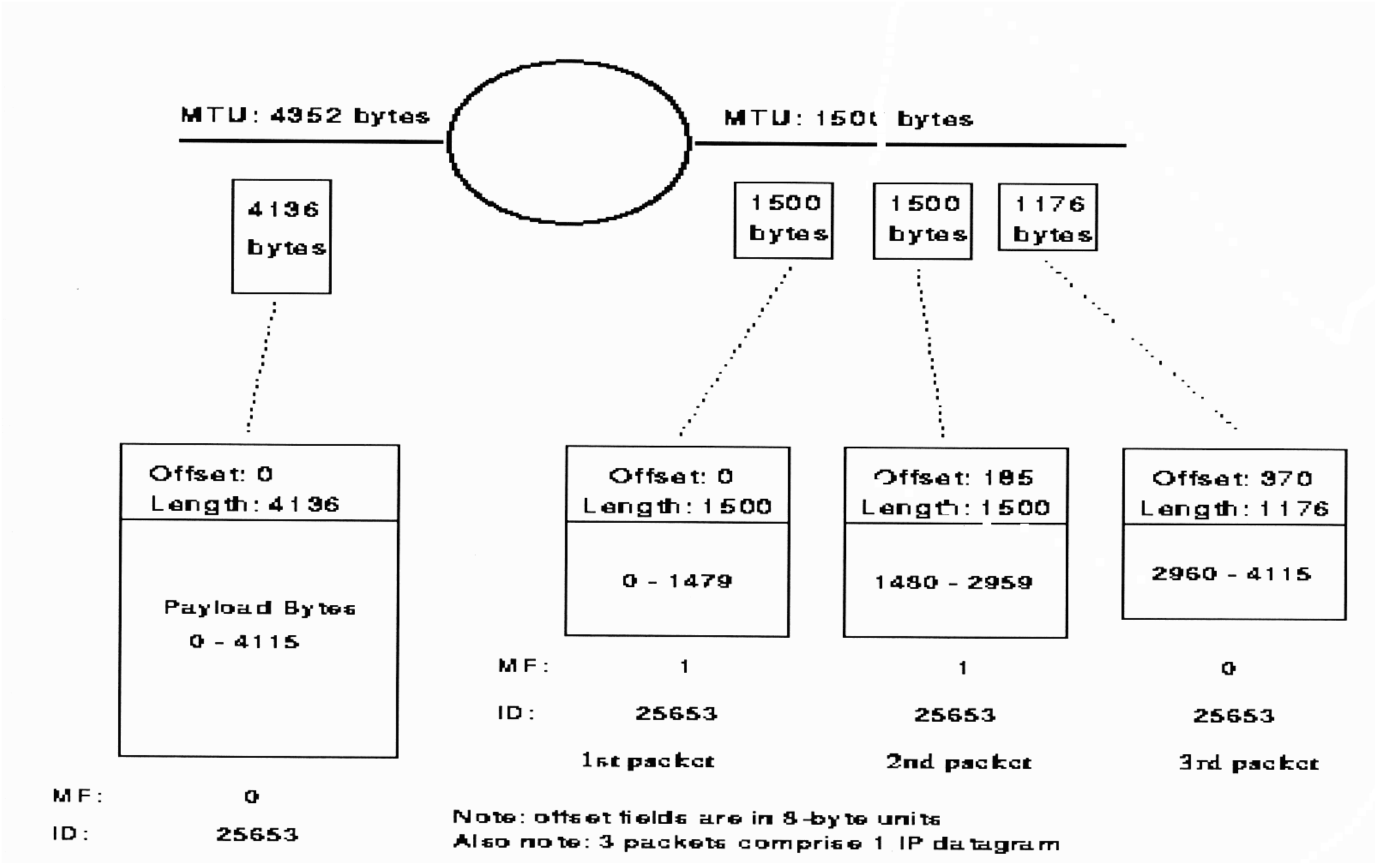
Datagrams are identified by the (source, destination, identification) triple

If fragmented, triple is copied into each

Also contains (offset, length, more?) triple

- More: 1 = more fragment, 0 = last fragment
- Offset: relative to original datagram

Adapting Size: Fragmentation Example



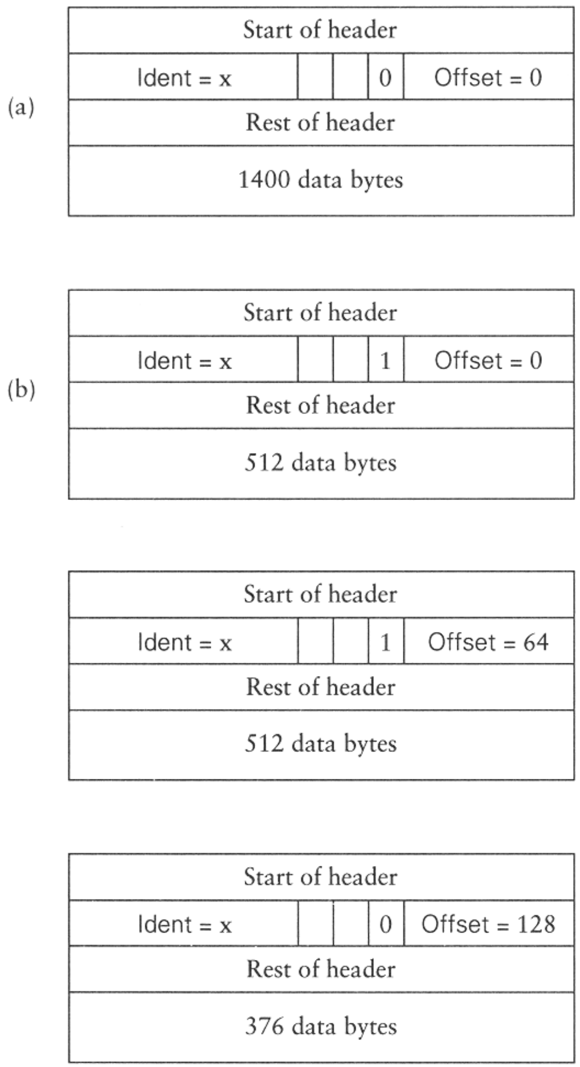


Figure 4.5 Header fields used in IP fragmentation. (a) Unfragmented packet; (b) fragmented packets.

Adapting Size: Fragmentation Control

- Relating fragmentations to original datagram provides:
  - Tolerance to re-ordering and duplication
  - Ability to fragment fragments
- When to fragment?
  - Whenever a big datagram enters smaller MTU network: typically occurs in a router
  - Common values for the MTU: 1460, 536, 512 bytes
  - Can happen from originating host!

Adapting Size: Reassembly

- IP fragments are re-assembled at final destination before a datagram is passed up to transport layer
- Routers do not reassemble fragmented datagrams
  - Allows for independent routing of fragments
  - reduces complexity/memory in router

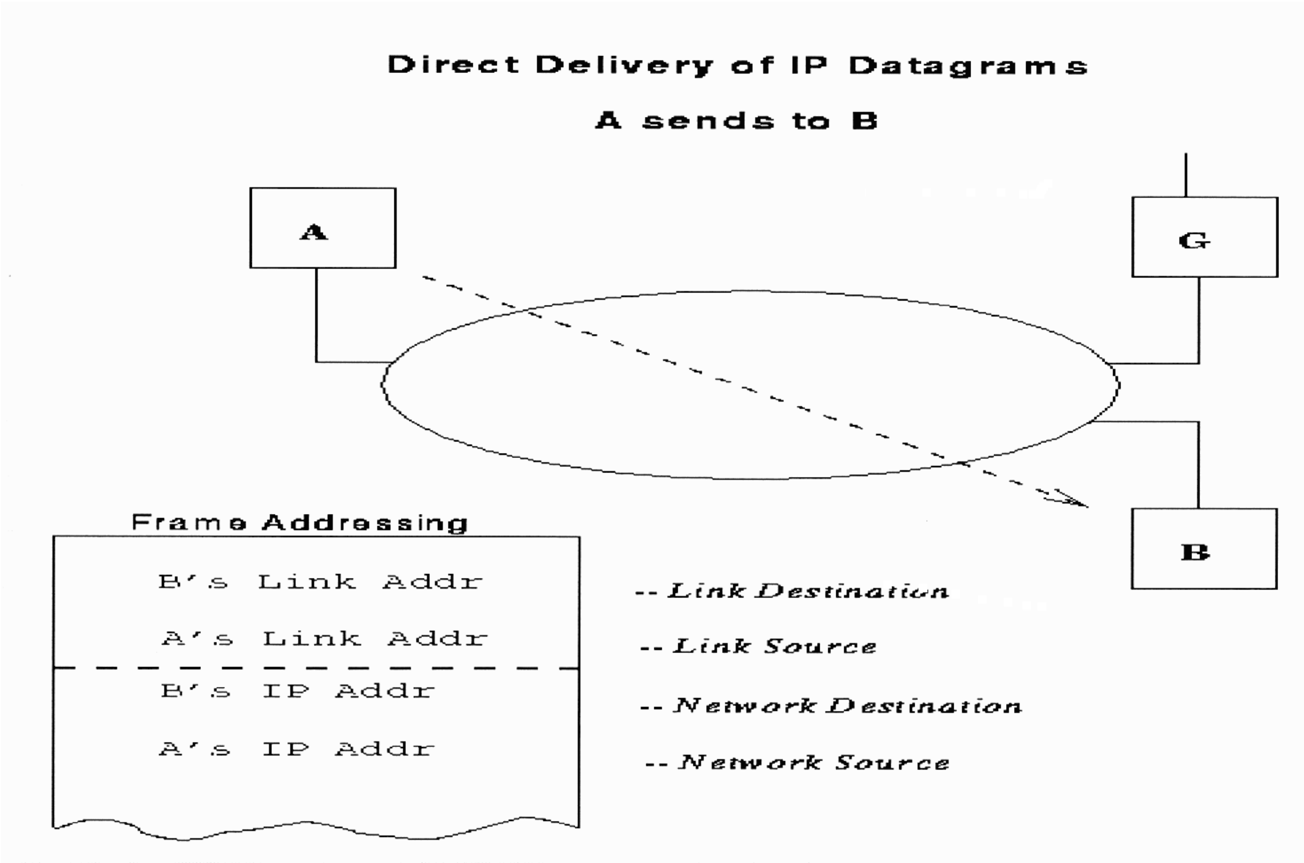
Adapting Size: Consequences

- Loss of one or more fragments implies loss of datagram at the IP layer
  - IP is best effort, provides no retransmission
  - Will time-out if fragments appear to be lost
- Would like to avoid fragmentation
  - Really want to know the Path MTU (later)

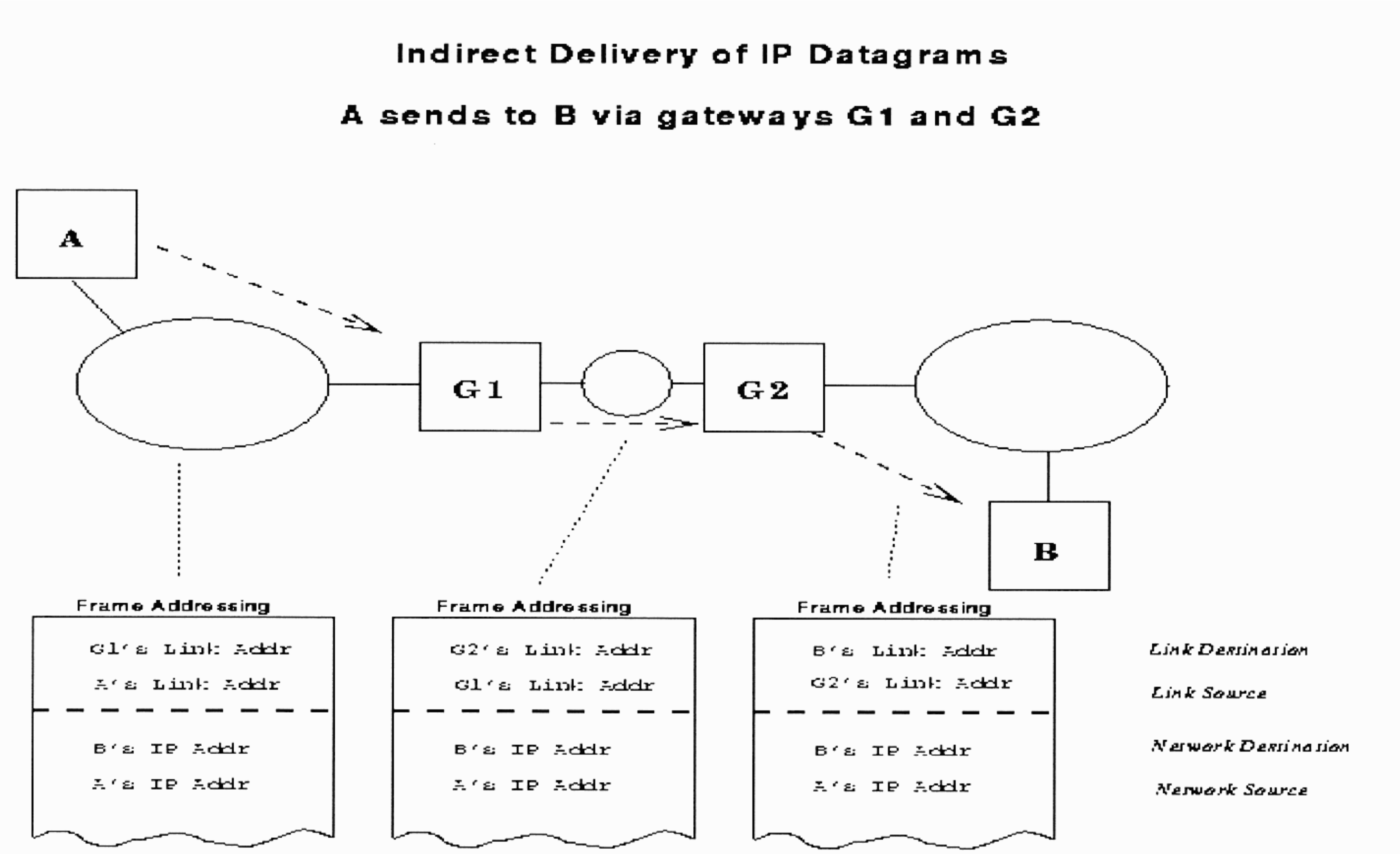
Adapting Size: Path MTU Discovery

- The Path MTU is the MIN of MTUs along delivery path
- If datagram size < MTU, no fragmentation!
- How to do this?
  - Probe network for largest size that will fit
  - If possible, have network tell "use this size"
  - (revisit this once we see ICMP)

Internetworking: Direct Delivery



Internetworking: Indirect Delivery



Direct Delivery - Summary

Sender acquires receiver’s IP address (e.g. through DNS or other mechanism)

Sender determines receiver is on the same network (by comparing network prefixes)

Sender performs ARP query to obtain receiver’s MAC address

Sender encapsulates IP packet in local frame destined for receiver’s MAC address

Indirect Delivery - Summary

Same as direct, except sender determines receiver is on different net

Sender queries routing table to determine correct next hop router

Encapsulates IP packet in local frame destined for router's MAC address

Routers repeat this procedure

Internetworking: Notes

- Note that fragmentation may occur at any router packet is too large for next hop MTU size (even local delivery!)
- Standards requirements
  - RFC 1812: Requirements for IPv4 routers
  - RFC 1122/1123: Requirements for Internet hosts

Summary

- For local delivery, need to map network-layer address to link-layer address
- ARP is specialized for each particular network/link address pairing
- For indirect delivery, will need MAC address of router (also uses ARP)
- Fragmentation and Reassembly
  - Divide network-layer datagram into multiple link-layer units, all ≤ link MTU size
  - Reconstruct datagram at final station