EECS 122, Lecture 10

Kevin Fall

kfall@cs.berkeley.edu

Internetworking

- 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)

Supporting a 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 (64KByte 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
- 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)
  - determined by common prefix
- 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?

IP to MAC Address Mapping

- Could just broadcast everything
  - un-necessary, burdens uninterested stations with others' traffic
- IP to MAC address mapping
  - configured by hand [cumbersome]
  - dynamic [learned by system automatically]

Learning IP-to-MAC Mappings

- 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)

- Base protocol is address independent (at both network & link layer)
- Protocol is specialized for each particular network/link address pairing
- Common example is Ethernet/IPv4

ARP Operation

- 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

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 Timers

- ARP Cache timeout
  - similar issues to bridge station caches
  - could be stale info if MAC address changes
  - RFC recommends 20 minute timeout

ARP Frame Structure

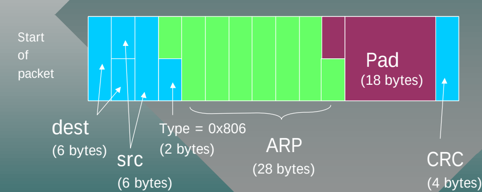
32 bits wide

HW Type	Prot. Type
HW Addr Len, Prot Addr Len	Operation
Sender's HW Addr (1-4)	
SHA (5-6)	SIP (1-2)
SIP (3-4)	THA (1-2)
Target HW Addr (3-6)	
Target IP Addr	

Common Header (HW Type, Prot. Type, HW Addr Len, Prot Addr Len, Operation)

IPv4/Ethernet Specific (Sender's HW Addr through Target IP Addr)

Ethernet ARP Encapsulation



Other ARP Uses

- Proxy ARP
  - one machine responds to ARP requests on behalf of others [can be used to 'hide' routers]
- Gratuitous ARP
  - send an ARP request for your own IP address (during bootstrap)
  - tells if address is already in use; also updates other's tables for own address

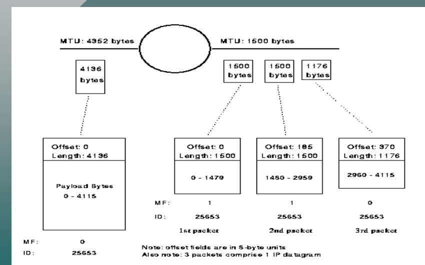
Adapting Datagram Size

- IP datagrams max 64KB, Ethernet frame max 1500 payload bytes…
- Fragmentation & 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

Fragmentation

- Datagrams are identified by the (src, dst, ident) triple
- If fragmented, triple is copied into each
- Also contains (offset, len, more?) triple
  - more? - boolean indicates last frag
  - offset - relative to *original* datagram

Fragmentation Example



Fragmentation Control

- Relating frags to original dgram provides:
  - tolerance to re-ordering and duplication
  - ability to fragment fragments
- When to fragment?
  - Whenever big dgram enters smaller MTU network
  - can happen from originating host!

Reassembly

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

Consequences

- Loss of 1 or more fragments implies loss of datagram at the IP layer
  - IP is best effort, provides no retransmission
  - will time-out if frag(s) appear to be lost
  - [interesting DoS attack perhaps....]
- Would like to avoid fragmentation
  - really want to know the *Path MTU* (later)

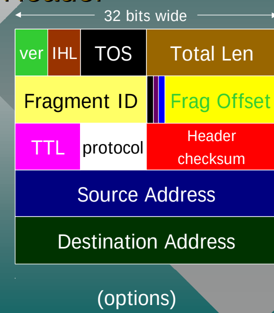
Path MTU Discovery

- The Path MTU is the MIN of MTUs along delivery path
- If dgram 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)

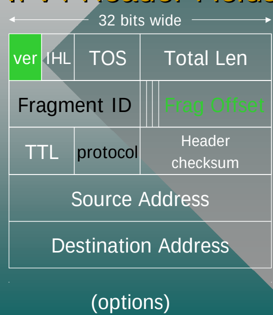
Internet Protocol Details (IP)

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

IPv4 Header

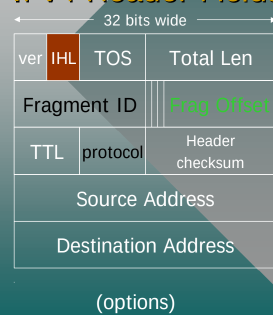


IPv4 Header Fields (ver)



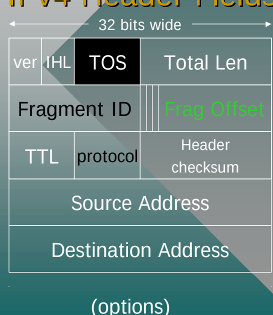
Version: 4 here
5 was for ST II
6 is for IPv6

IPv4 Header Fields (IHL)



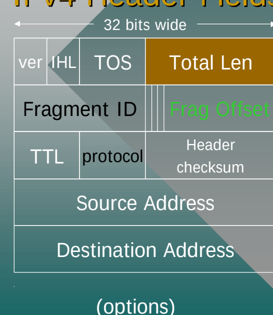
IHL: # 32 bit words in IP header. Min is 5 (20 bytes), max is 15 (60 bytes). Thus, at most 40 bytes of options.

IPv4 Header Fields (TOS)

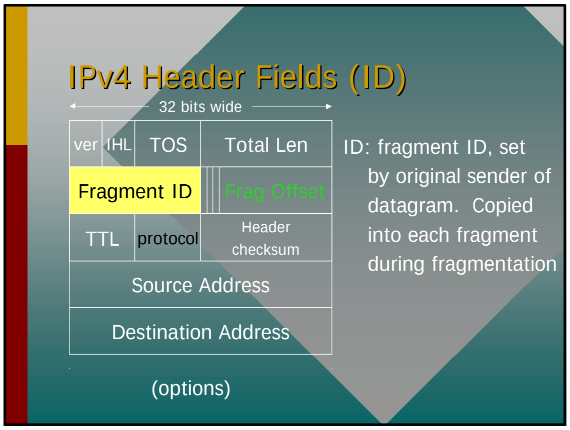


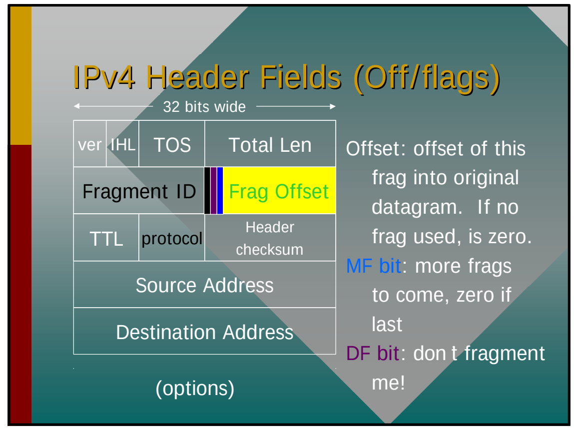
TOS: type of service. Abstract notion of type of service, including precedence. Most routers ignore. Will be used by diff-serv.

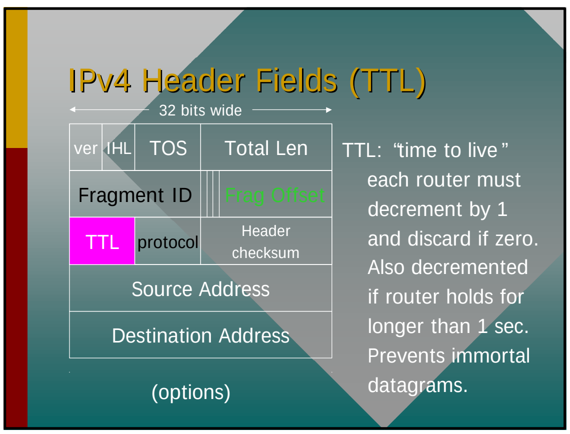
IPv4 Header Fields (Length)

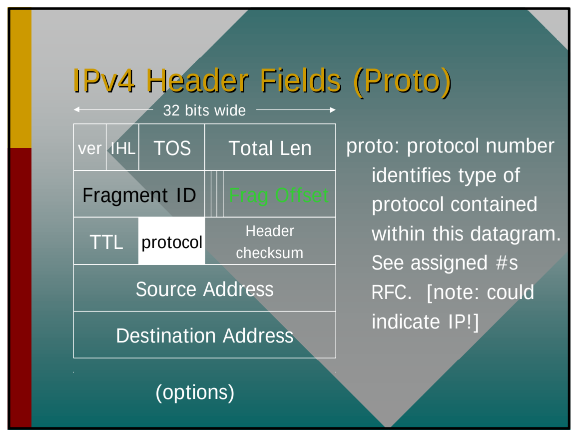


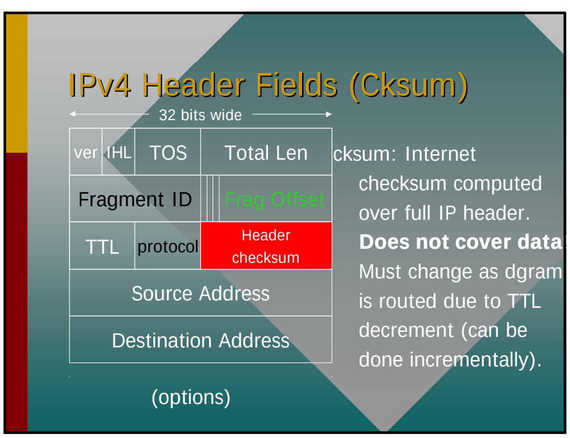
Len: # bytes in this datagram (fragment). Min value is 20. Max value is 65535. Limits max. dgram size.

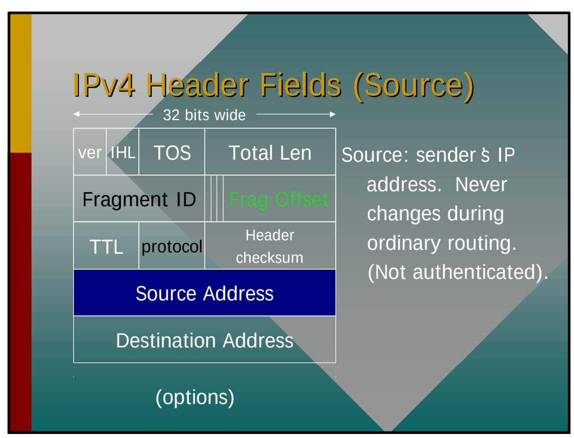

IPv4 Header Fields (ID)
32 bits wide
ver
IHL
TOS
Total Len
Fragment ID
Frag Offset
TTL
protocol
Header checksum
Source Address
Destination Address
(options)
ID: fragment ID, set by original sender of datagram. Copied into each fragment during fragmentation


IPv4 Header Fields (Off/flags)
32 bits wide
ver
IHL
TOS
Total Len
Fragment ID
Frag Offset
TTL
protocol
Header checksum
Source Address
Destination Address
(options)
Offset: offset of this frag into original datagram. If no frag used, is zero.
MF bit: more frags to come, zero if last
DF bit: don t fragment me!


IPv4 Header Fields (TTL)
32 bits wide
ver
IHL
TOS
Total Len
Fragment ID
Frag Offset
TTL
protocol
Header checksum
Source Address
Destination Address
(options)
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 Fields (Proto)
32 bits wide
ver
IHL
TOS
Total Len
Fragment ID
Frag Offset
TTL
protocol
Header checksum
Source Address
Destination Address
(options)
proto: protocol number identifies type of protocol contained within this datagram. See assigned #s RFC. [note: could indicate IP!]


IPv4 Header Fields (Cksum)
32 bits wide
ver
IHL
TOS
Total Len
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 dgram is routed due to TTL decrement (can be done incrementally).


IPv4 Header Fields (Source)
32 bits wide
ver
IHL
TOS
Total Len
Fragment ID
Frag Offset
TTL
protocol
Header checksum
Source Address
Destination Address
(options)
Source: sender s IP address. Never changes during ordinary routing. (Not authenticated).

IPv4 Header Fields (Dest)

32 bits wide

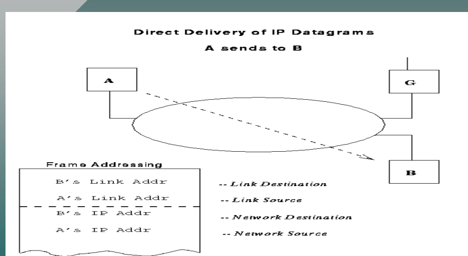
ver	IHL	TOS	Total Len
Fragment ID		Frag Offset	
TTL	protocol	Header checksum	
Source Address			
Destination Address			
(options)			

Destination: receiver's IP address. Never changes during ordinary routing. Changes when *source routing* is used.

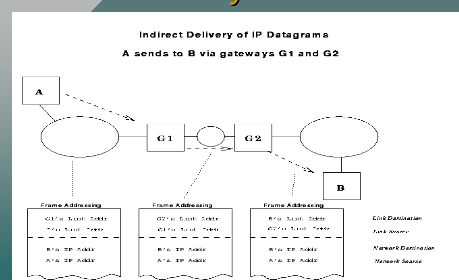
IP Options

- Special handling for particular datagrams, sometimes don't take router's "fast path"
- Rarely used, but the more common are:
  - Loose Source Routing
  - String Source Routing
  - Record Route
  - Timestamp
- Most copied on fragmentation

Direct Delivery (no router)



Indirect Delivery



Direct Delivery (summary)

- Sender acquires receiver's IP address (e.g. through DNS or other mechanism)
- Sender determines receiver is on 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 addr

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

Details

- Note that fragmentation may occur at any place packet is too large for next-hop MTU size (even local delivery!)
- Standards requirements
  - RFC 1812 : Requirements for IPv4 routers
  - RFC 1122,3 : Requirements for Internet hosts