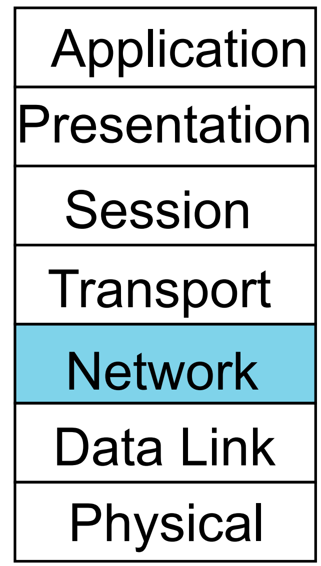
CSE 461: IP Addressing and Forwarding

Next Topic

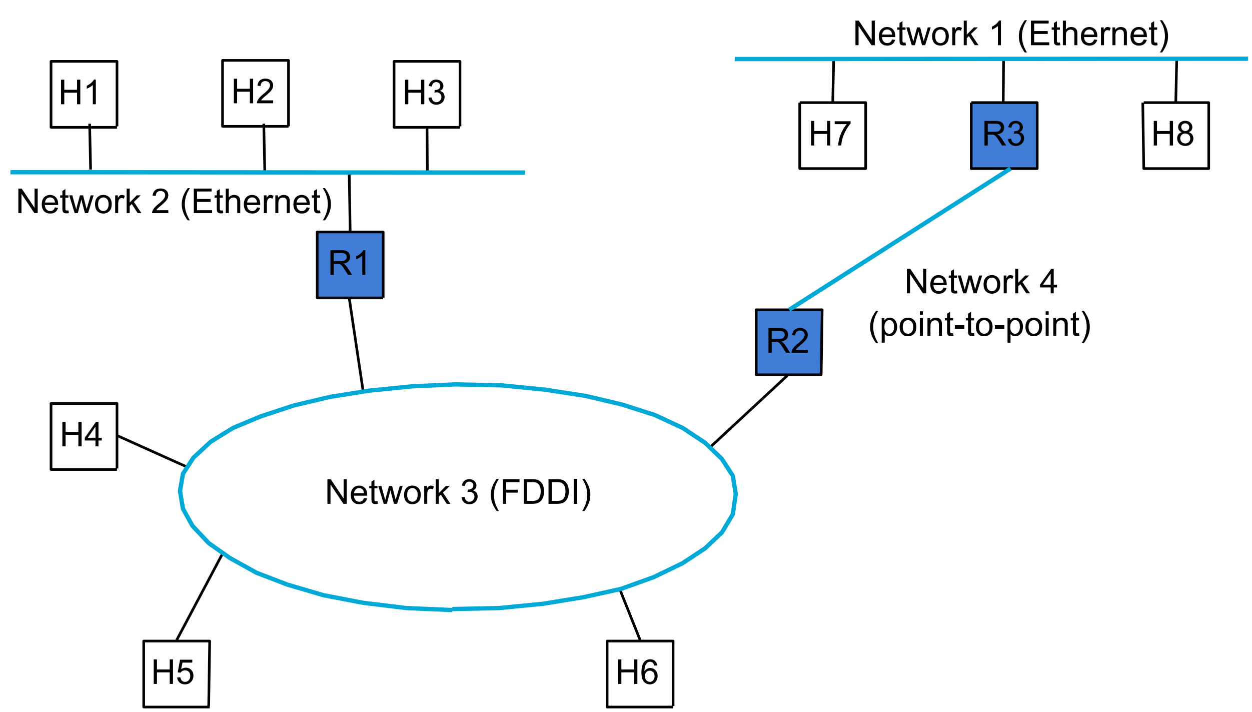
- Focus:
  - How do we build large networks?

- Introduction to the Network layer
  - Internetworks
  - Service models
  - IP, ICMP



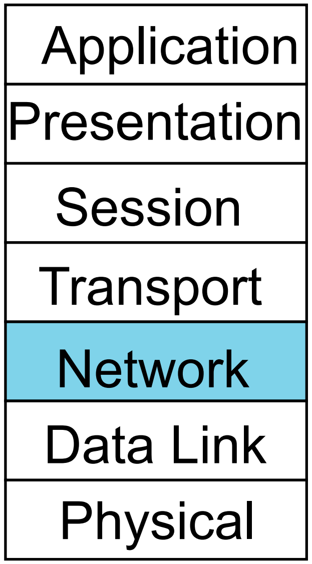
Internetworks

- Set of interconnected networks, e.g., the Internet
  - Scale and heterogeneity



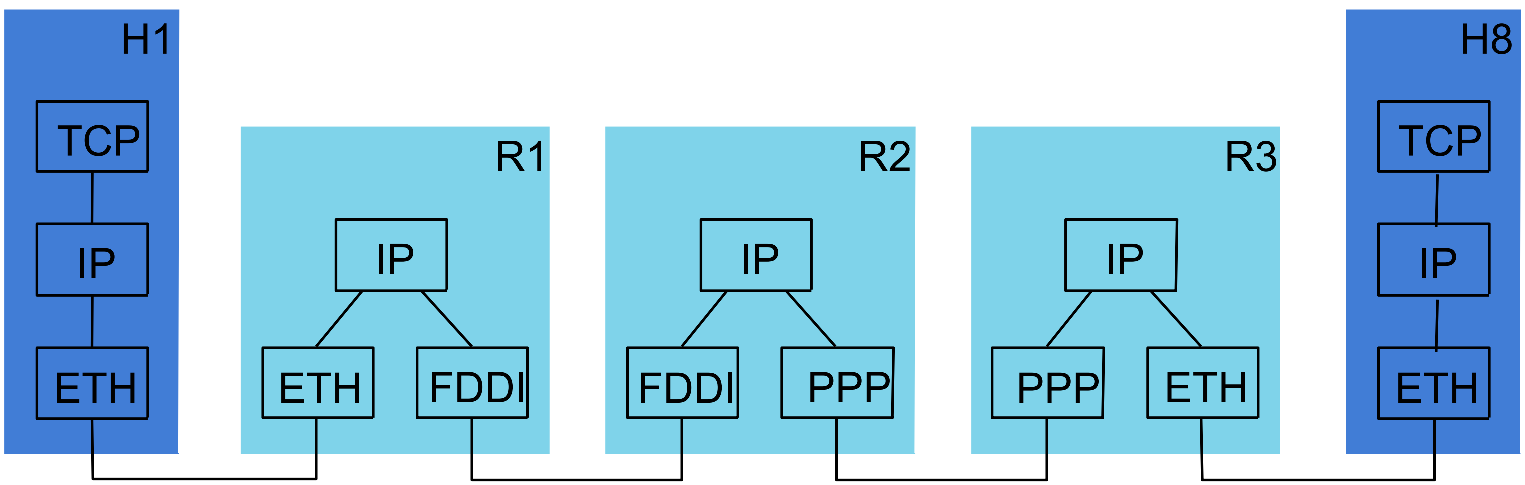
The Network Layer

- Job is to provide end-to-end data delivery between hosts on an internetwork
- Provides a higher layer of addressing



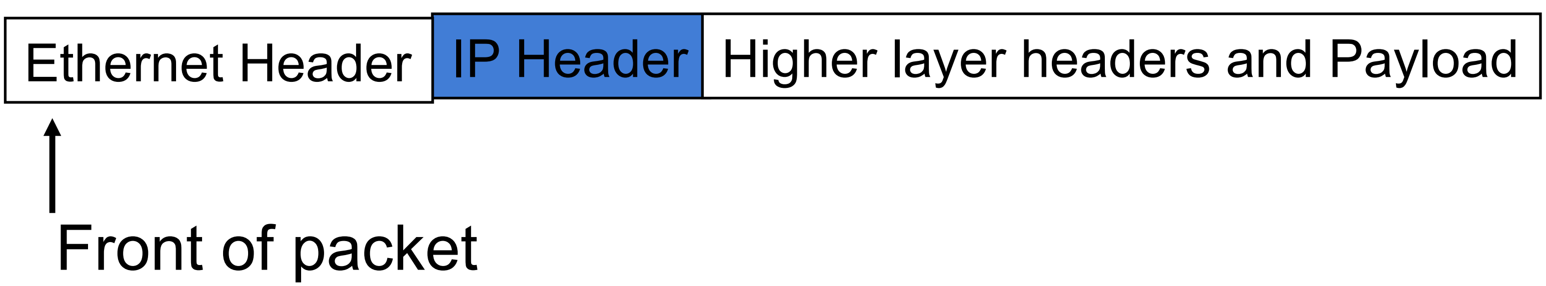
In terms of protocol stacks

- IP is the network layer protocol used in the Internet
- Routers are network level gateways
- Packet is the term for network layer Protocol Data Unit (PDU)



In terms of packet formats

- View of a packet on the wire
- Routers work with IP header, not higher
  - Higher would be a "layer violation"
- Routers strip and add link layer headers



Network Service Models

- Datagram delivery: postal service
  - connectionless, best-effort or unreliable service
  - Network can't guarantee delivery of the packet
  - Each packet from a host is routed independently
  - Example: IP
- Virtual circuit models: telephone
  - connection-oriented service
  - Connection establishment, data transfer, teardown
  - All packets from a host are routed the same way (router state)
  - Example: ATM, Frame Relay, X.25

Internet Protocol (IP)

- IP (RFC791) defines a datagram “best effort” service
- Works on top of a wide variety of networks
- Undemanding enough to work with underlying link technologies
  - Packet carries enough info for network to forward to destination
  - May be loss, reordering, duplication, and errors
    - No effort to recover from failure
    - Keep routers as simple as possible
- Scales to billions of hosts
- Currently IPv4 (IP version 4), IPv6 on the way (apparently!)

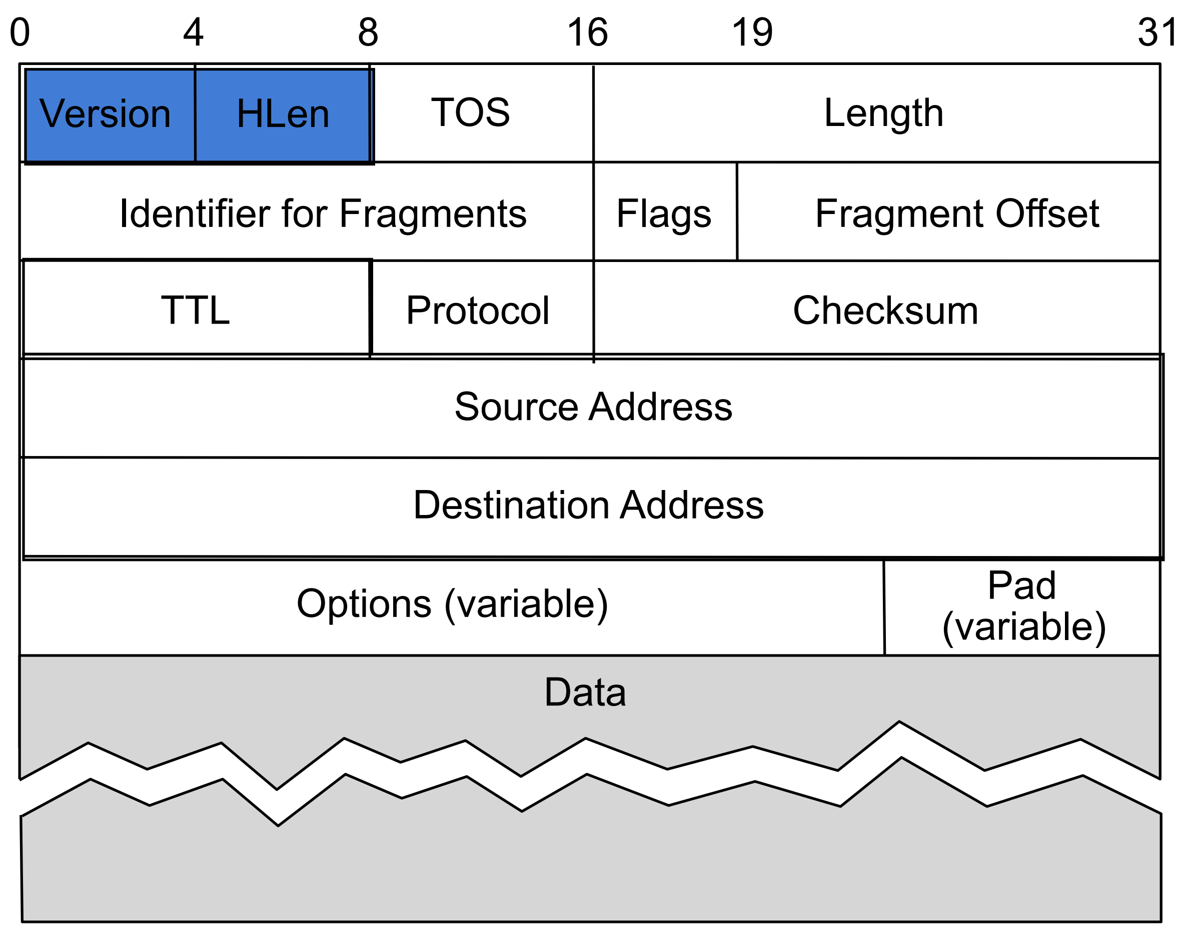
Internet Protocol (IP) continued

- Routers forward packets using predetermined routes
  - Routing protocols (RIP, OSPF, BGP) run between routers to maintain routes (routing table)

- Global, hierarchical addresses, not flat addresses
  - 32 bits in IPv4 address; 128 bits in IPv6 address
  - ARP (Address Resolution Protocol) maps IP to MAC addresses

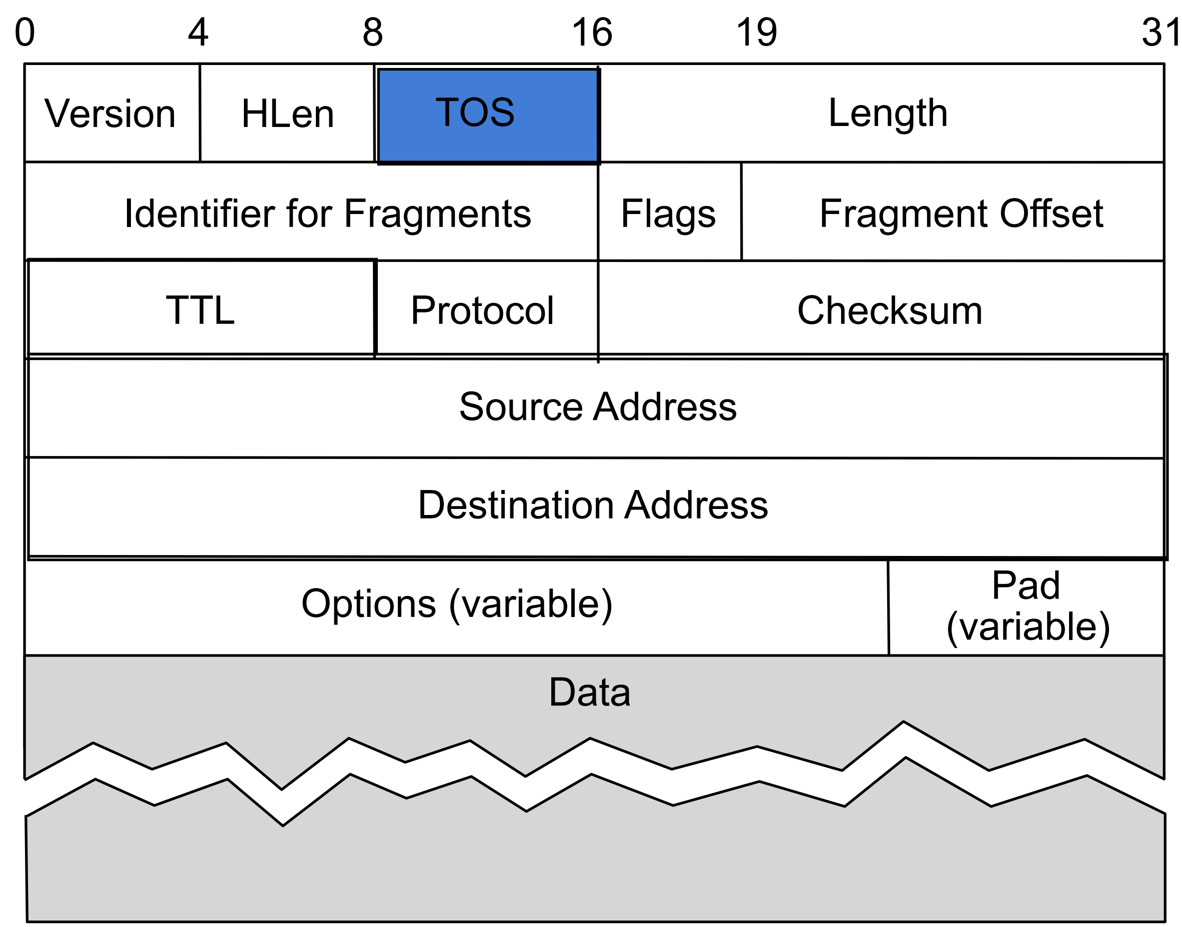
IPv4 Packet Format

- Version is 4
- Header length is number of 32 bit words
- Limits size of options



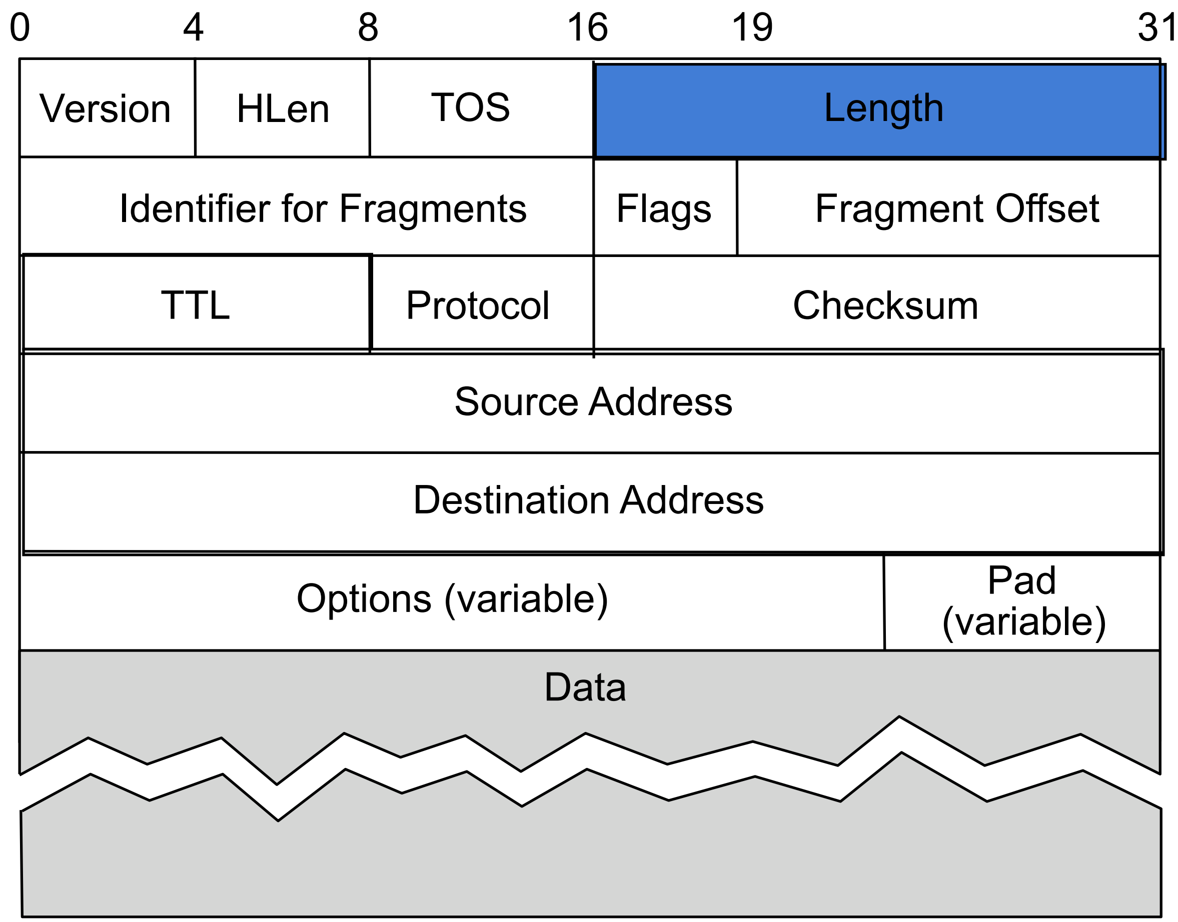
IPv4 Header Fields …

- Type of Service
- Abstract notion, never really worked out
  - Routers ignored



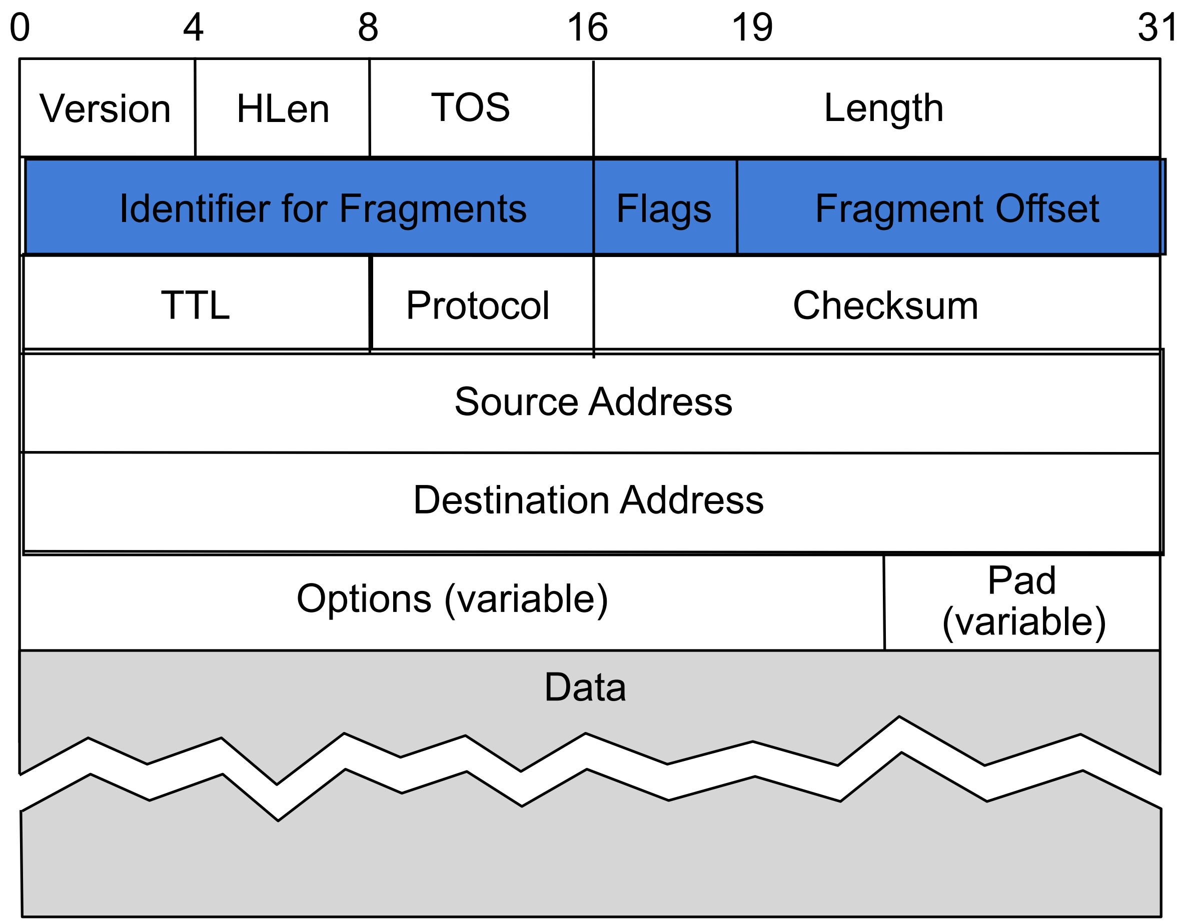
IPv4 Header Fields …

- Length of packet
  - In bytes
  - Includes header
- Min 20 bytes, max 64K bytes (limit to packet size)



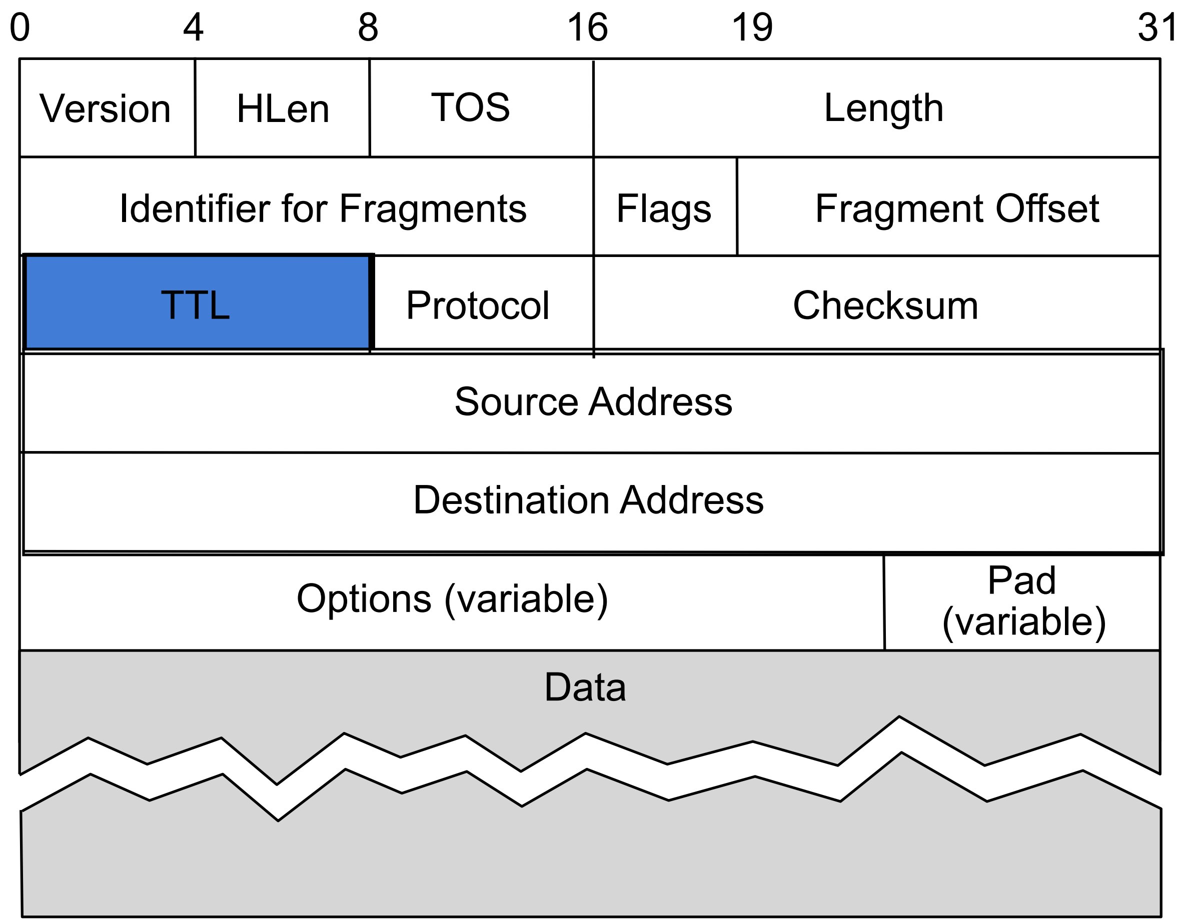
IPv4 Header Fields …

- Fragment fields
- Different LANs have different frame size limits
- May need to break large packet into smaller fragments



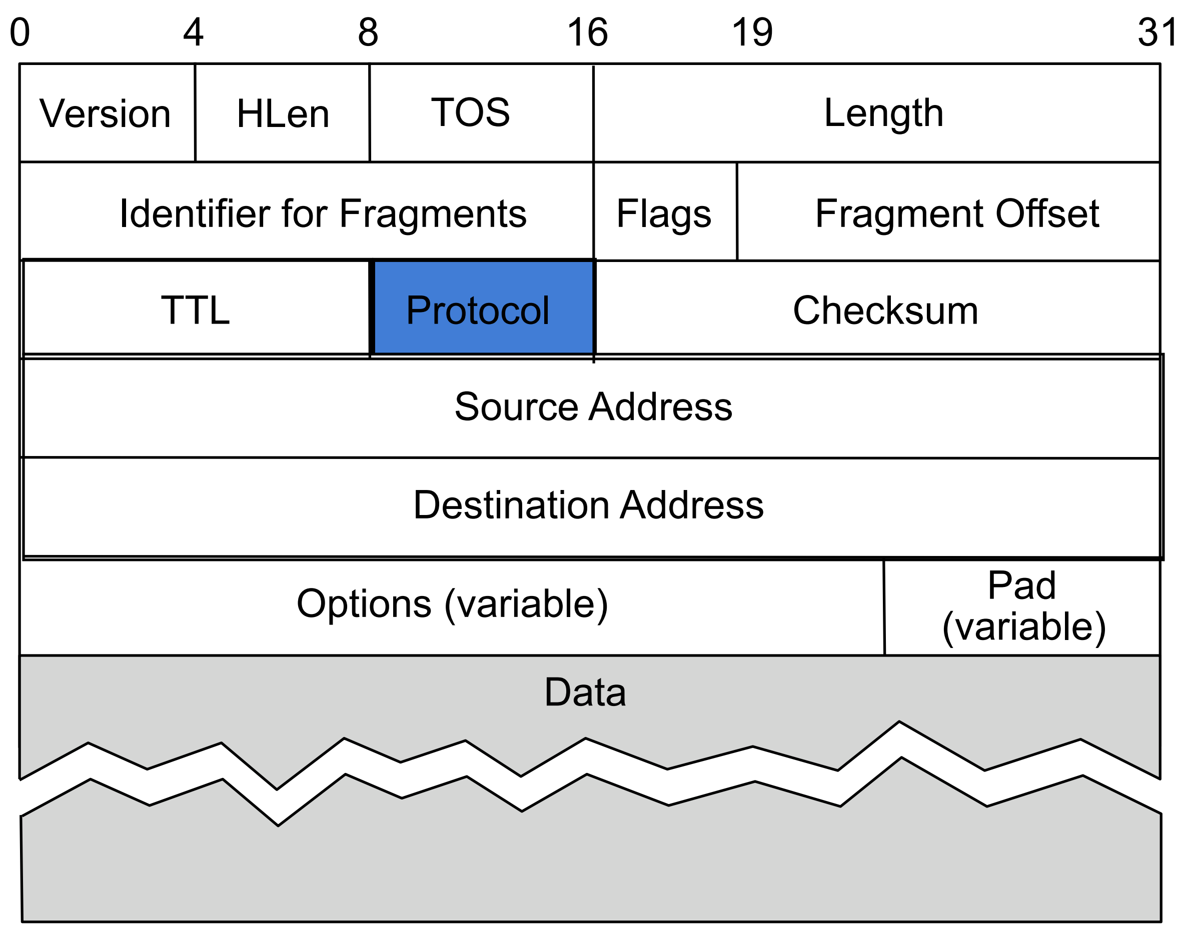
IPv4 Header Fields …

- Time To Live
- Decremented by router and packet discarded if = 0
- Prevents immortal packets



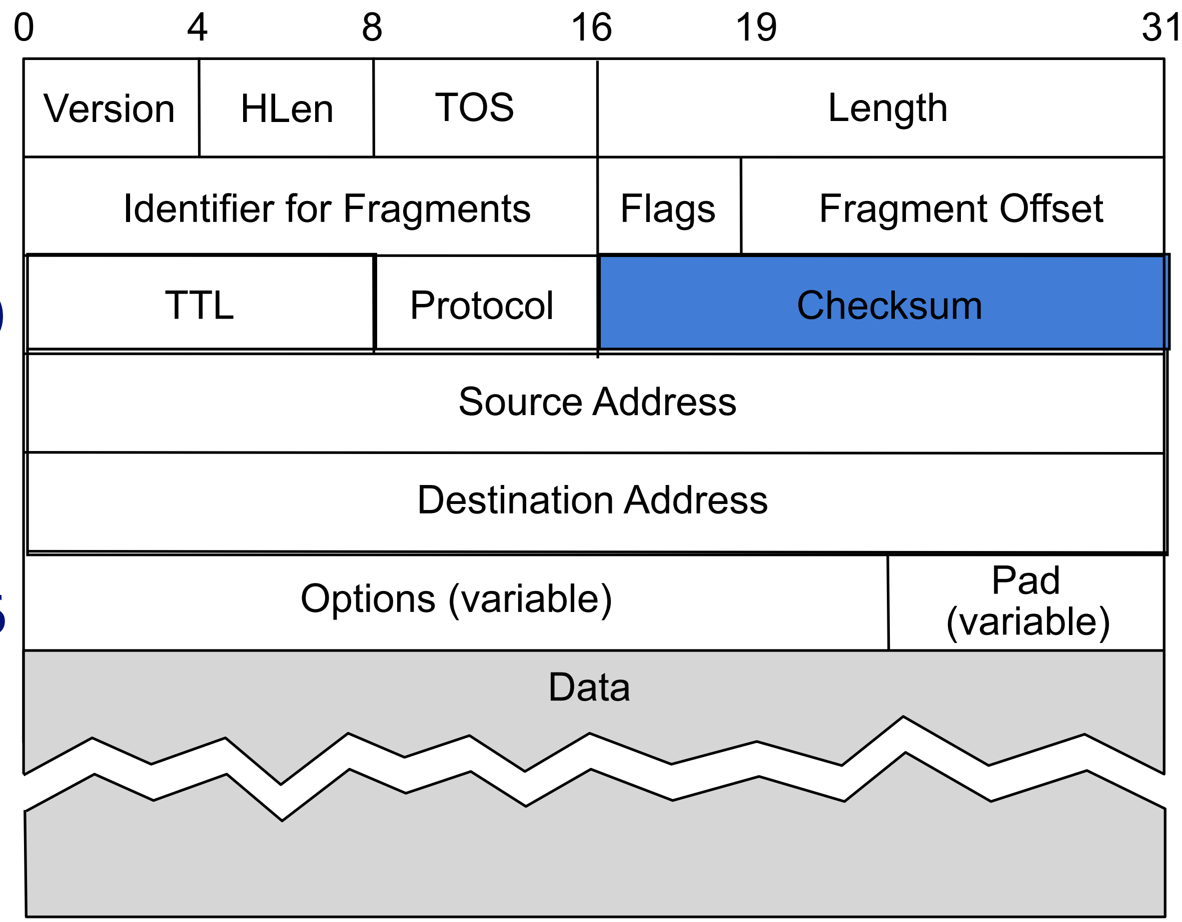
IPv4 Header Fields …

- Identifies higher layer protocol
  - E.g., TCP, UDP



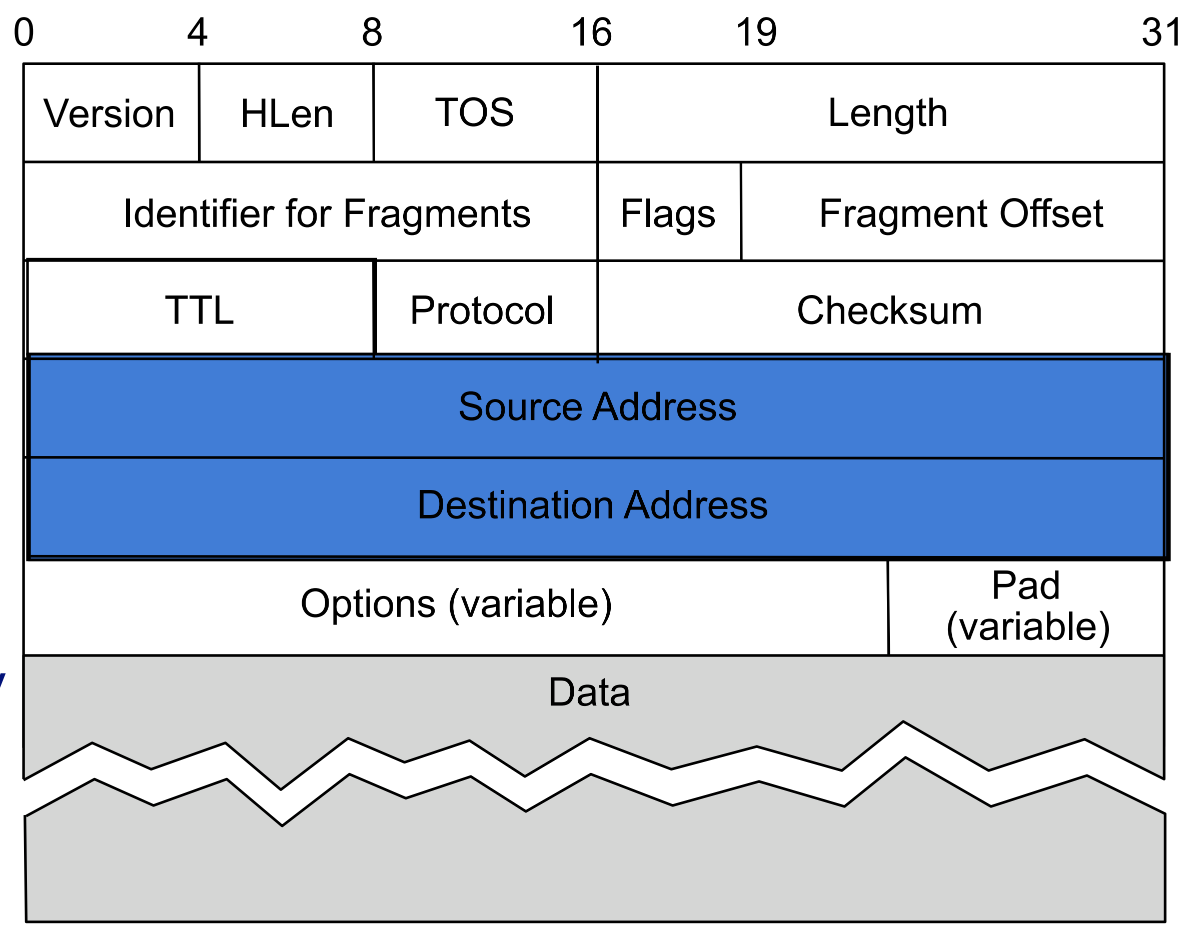
IPv4 Header Fields …

- Header checksum
- Recalculated by routers (TTL drops)
- Doesn't cover data
- Disappears for IPv6



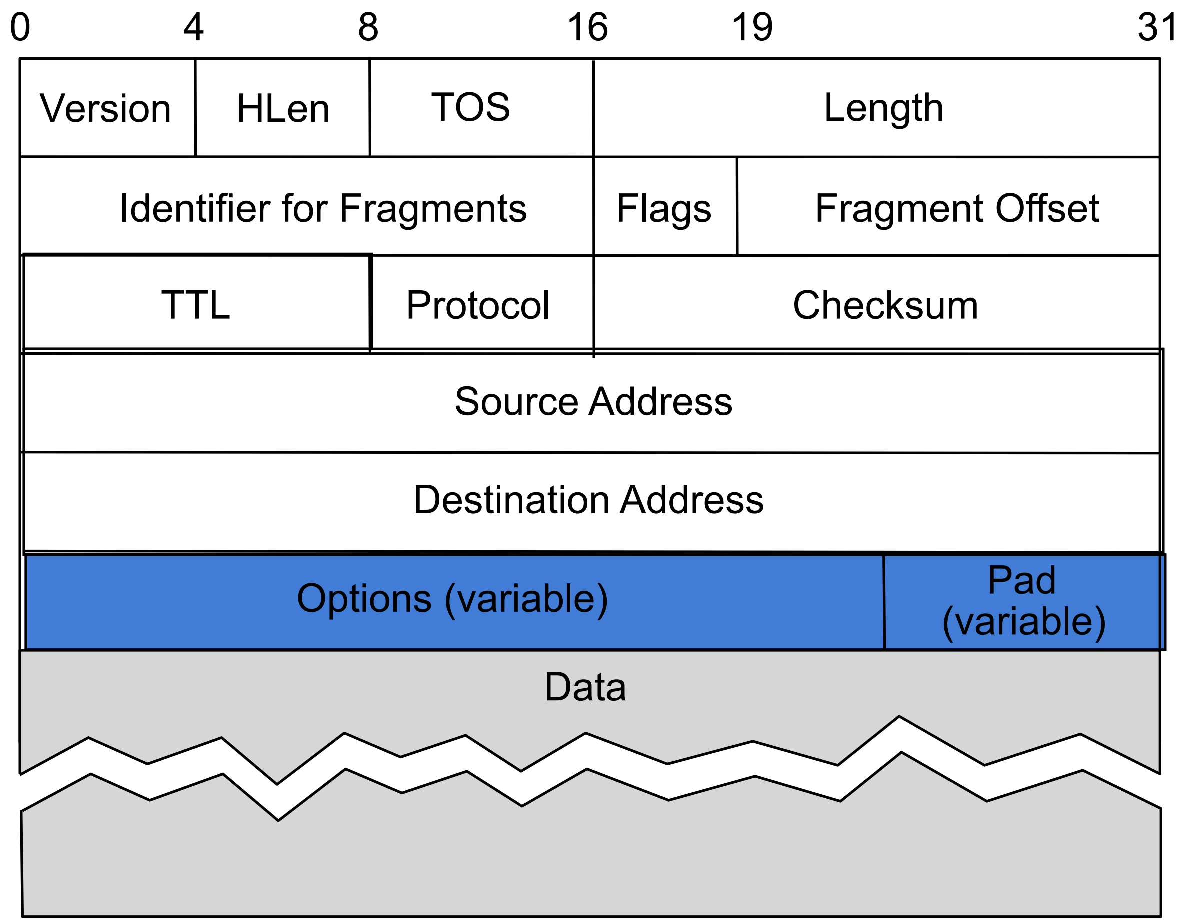
IPv4 Header Fields …

- Source/destination IP addresses
  - Not Ethernet
- Unchanged by routers
  - Except NAT
- Not authenticated by default



IPv4 Header Fields …

- IP options indicate special handling
  - Timestamps
  - "Source" routes
- Rarely used …
- Btw, where are the ports?



IP Addresses

- E.g., 192.168.1.1
- 32 bits, hierarchical, conceptually split into 2 parts:

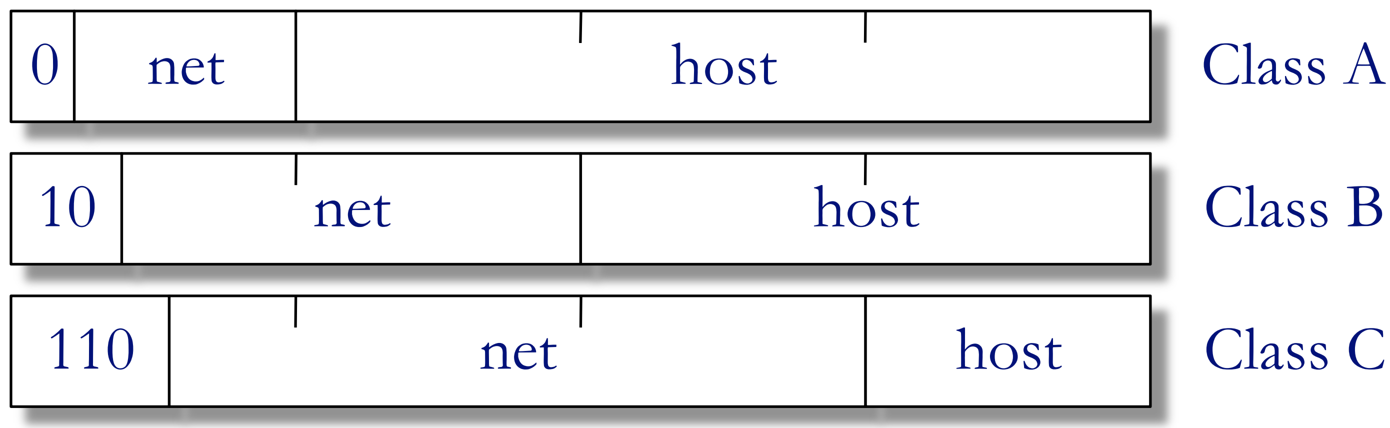
Network	Host

  - Flexible boundary

11111111	11111111	11000000	00000000	mask
11101000	01010101	01000000	00000000	address

  - Routers don't support noncontiguous subnet masks
- Host must learn its address, usually via dhcp
  - Unlike Ethernet addresses, which typically are burned into ROM

IP Address Classes and Subnetting



- Defined by a mask
  - Networks with lots of hosts would get class A, etc.
- Notion of subnetting…
  - Net number (e.g., 1st octet in Class A)
  - Subnet number (as far as the 1s extend in the mask)
  - Host number
- Classless Interdomain Routing (CIDR) adds flexibility

Data Forwarding

Destination	Gateway
default	192.168.1.1
192.168.1	Link #4

My laptop's routing table (netstat –r)

- Send a packet to my printer (192.168.1.254)
  - Note: netmask is FFFFFF00
- Send a packet to cnn (157.166.224.25)

Modern IP Address Lookup

- routing tables contain (*prefix, next hop*) pairs
- address in packet compared to stored prefixes, starting at left
- prefix that matches largest number of address bits is desired match
- packet forwarded to specified next hop

routing table

prefix	next hop
10*	7
01*	5
110*	3
1011*	5
0001*	0
0101 1*	7
0001 0*	1
0011 00*	2
1011 001*	3
1011 010*	5
0100 110*	6
0100 1100*	4
~~1011 0011*~~	8
1001 1000*	10
0101 1001*	9

address: <u>1011 0010</u> 1000

Problem - large router may have 100,000 prefixes in its list

Feamster, GA Tech, 2008

Fragmentation Issue

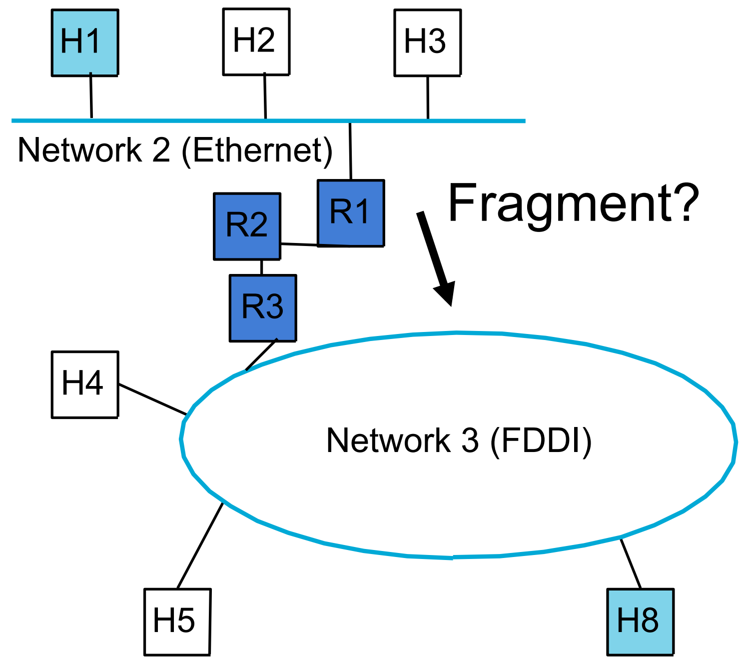
- Different networks may have different frame limits (MTUs)
  - Ethernet 1.5K, FDDI 4.5K

- Don't know if packet will be too big for path beforehand

Options:

1. Fragment and reassemble at each link
2. Fragment and reassemble at destination

Which is better?

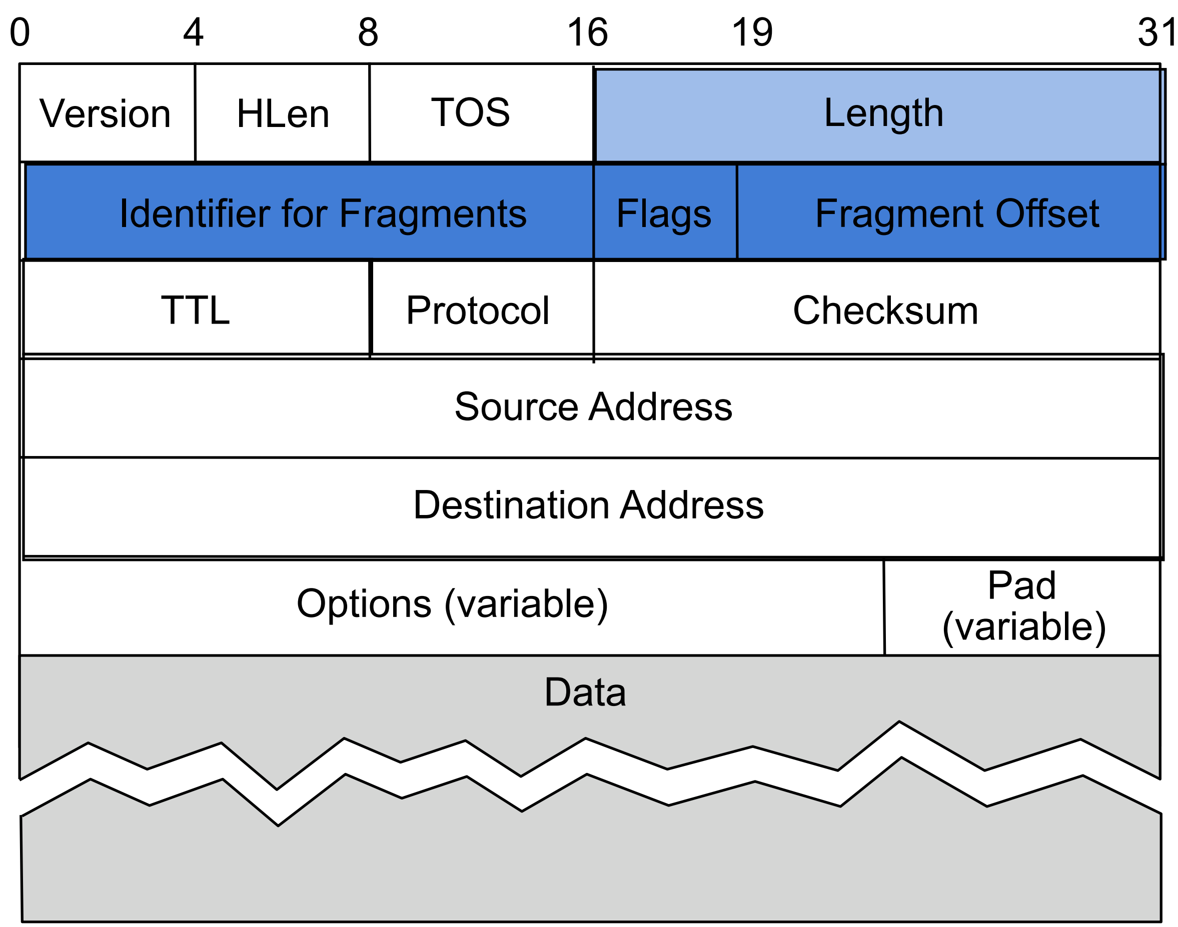


Fragmentation and Reassembly

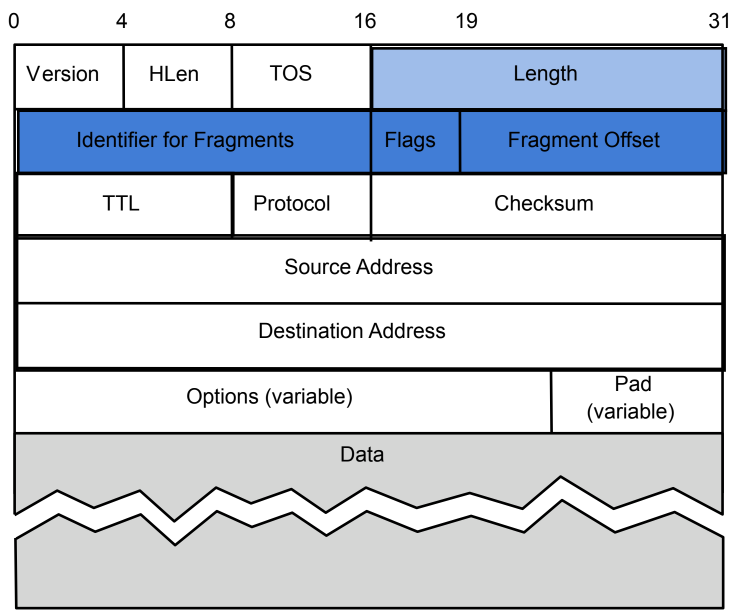
- Strategy
  - fragment when necessary (MTU < Datagram size)
  - refragmentation is possible
  - fragments are self-contained IP datagrams
  - delay reassembly until destination host
  - do not recover from lost fragments

Fragment Fields

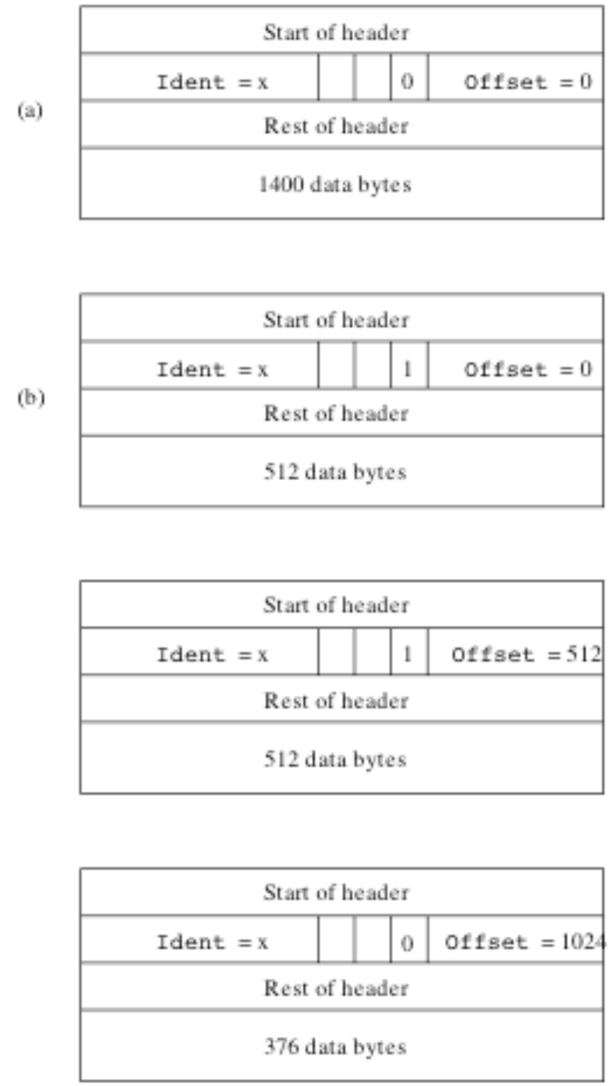
- Fragments of one packet identified by (source, dest, frag id) triple
  - Make unique
- Offset gives start, length changed
- Flags are More Fragments (MF) Don't Fragment (DF)



Fragmenting a Packet



Packet Format



Fragment Considerations

- Making fragments be datagrams provides:
  - Tolerance of reordering and duplication
  - Ability to fragment fragments
- Reassembly done at the endpoint
  - Puts pressure on the receiver, not network interior
- Consequences of fragmentation:
  - Loss of any fragments causes loss of entire packet
  - Need to time-out reassembly when any fragments lost

Fragmentation Issues Summary

- Causes inefficient use of resources within the network
  - BW, CPU
- Higher level protocols must re-xmit entire datagram
  - on lossy network links, hard for packet to survive
- Efficient reassembly is hard
  - Lots of special cases
  - (think linked lists)

Avoid Fragmentation with Path MTU Discovery

- Path MTU is the smallest MTU along path
  - Packets less than this size don't get fragmented
  - Idea: Avoid fragmentation too by having hosts learn path MTUs

- Non-option: send very small datagrams
  - Overly conservative, lots of header overhead

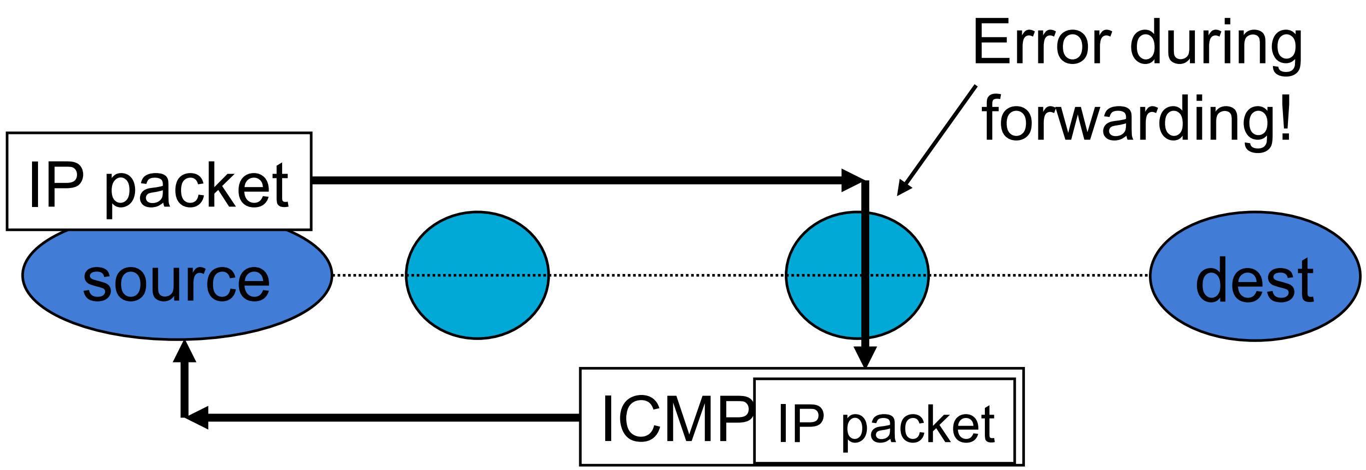
- Hosts send packets, routers return error if too large
  - Use DF flag
  - Hosts discover limits, can fragment at source
  - Reassembly at destination as before

- Learned lesson from IPv4, streamlined in IPv6

ICMP

- What happens when things go wrong?
  - Need a way to test/debug a large, widely distributed system
- ICMP = Internet Control Message Protocol (RFC792)
  - Companion to IP – required functionality
- Used for error and information reporting:
  - Errors that occur during IP forwarding
  - Queries about the status of the network

ICMP Generation



Common ICMP Messages

- Destination unreachable
  - "Destination" can be network, host, port or protocol
- Packet needs fragmenting but DF is set
- Redirect
  - To shortcut circuitous routing
- TTL Expired
  - Used by the "traceroute" program
- Echo request/reply
  - Used by the "ping" program
- Cannot Fragment
- Busted Checksum

- ICMP messages include portion of IP packet that triggered the error (if applicable) in their payload

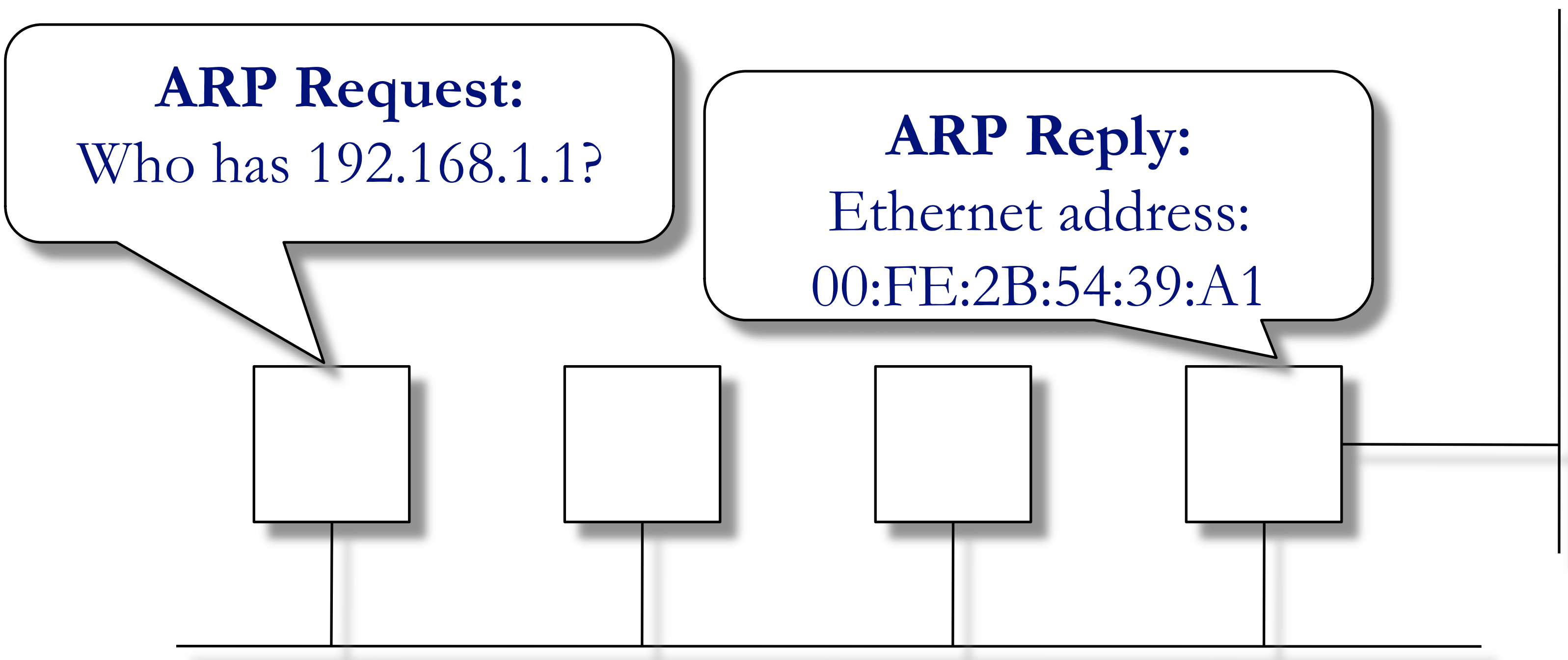
ICMP Restrictions

- The generation of error messages is limited to avoid cascades … error causes error that causes error!
- Don’t generate ICMP error in response to:
  - An ICMP error
  - Broadcast/multicast messages (link or IP level)
  - IP header that is corrupt or has bogus source address
  - Fragments, except the first
- ICMP messages are often rate-limited too.

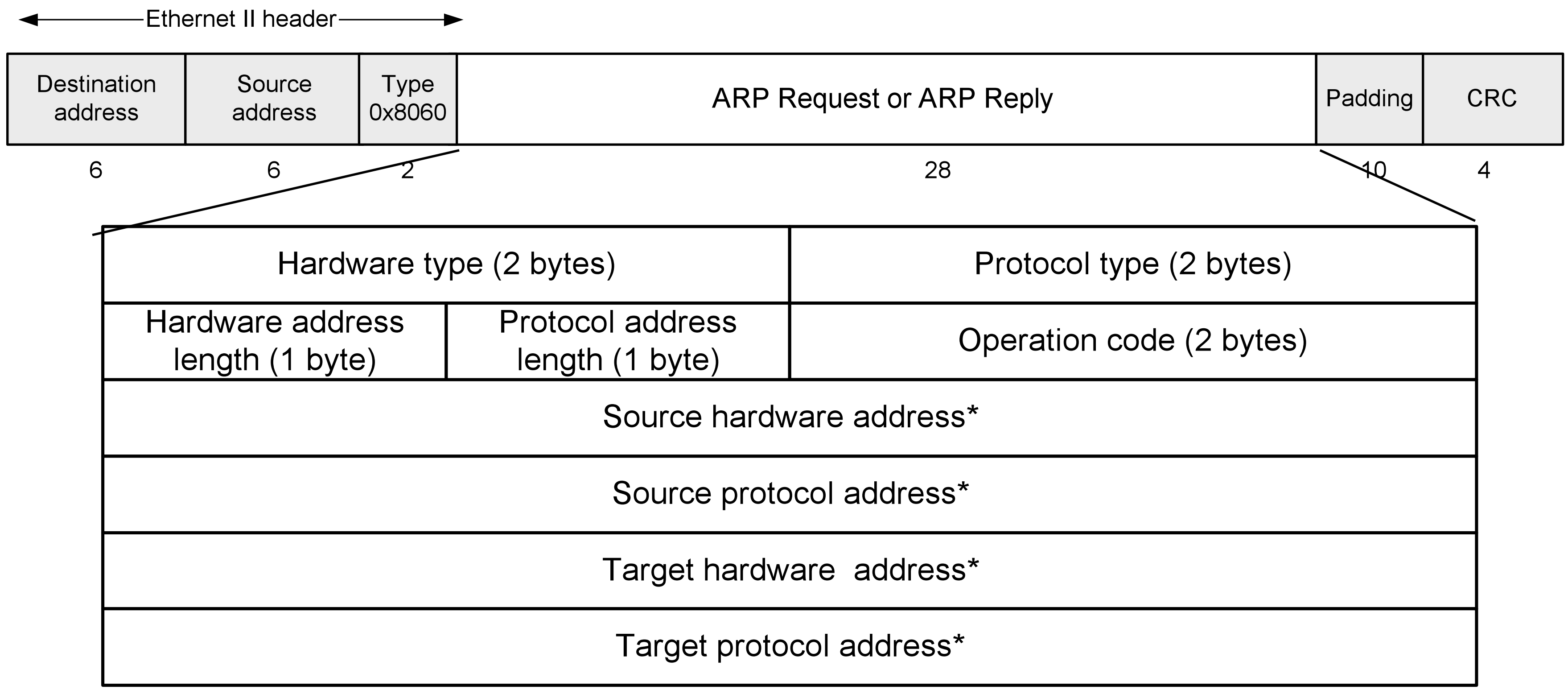
Address Resolution Protocol (ARP)

- Problem: We know a destination IP address, but how do we find the actual device on the LAN with that address?

- Solution: ARP



ARP Packet Format



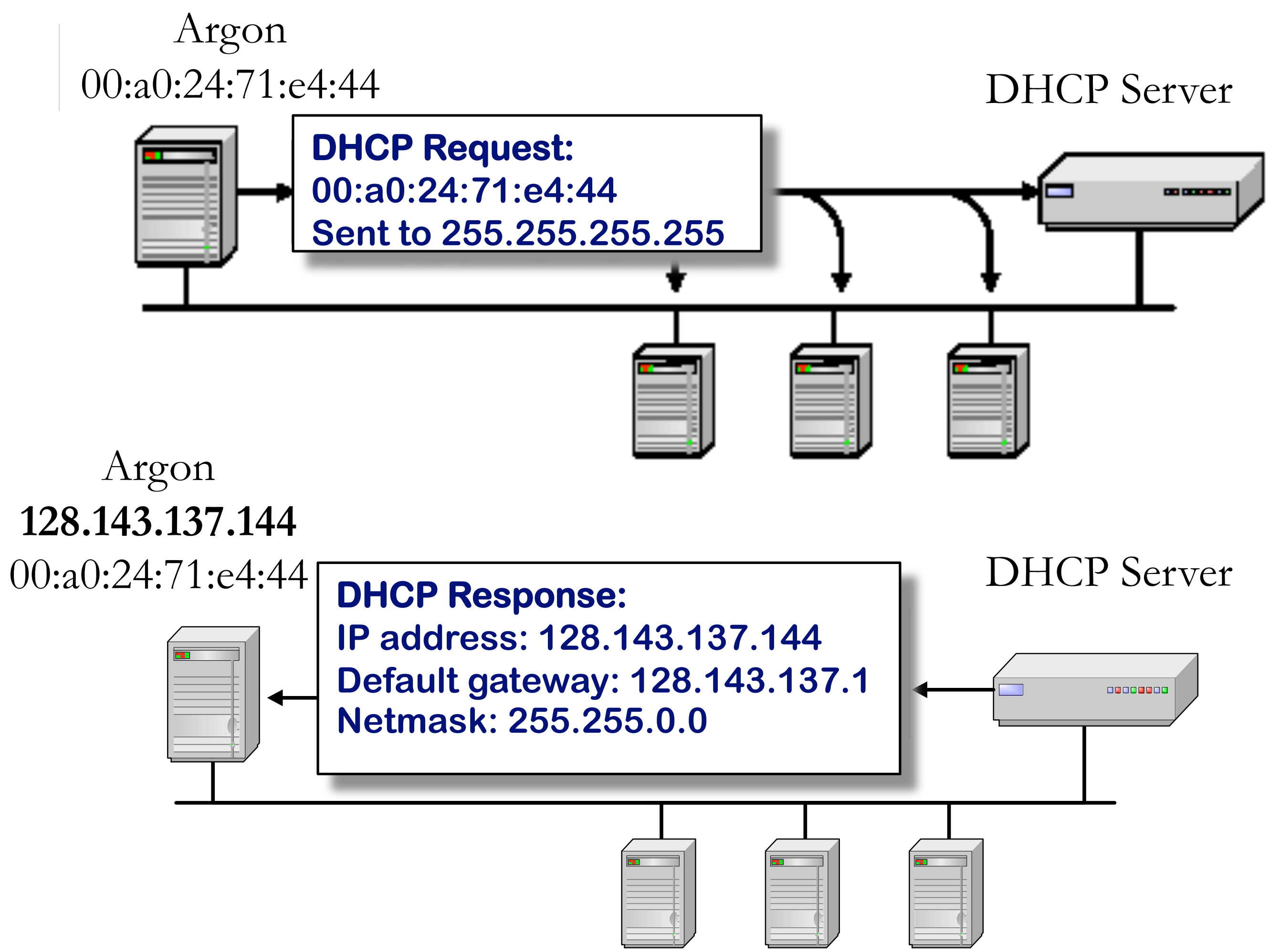
- Host that requests caches destination
- Host that replies caches source address
- Other devices ignore
- Values typically stay in ARP cache for 20 minutes

Dynamic Host Configuration Protocol (DHCP)

- How does a host get an IP address?
- DHCP designed in 1993
- An extension of BOOTP (Many similarities to BOOTP)
- Runs over UDP, which in turn runs over IP
  - Same port numbers as BOOTP (67, 68)
- Extensions:
  - Supports temporary allocation (“leases”) of IP addresses
  - DHCP client can acquire all IP configuration parameters

- DHCP is the preferred mechanism for dynamic assignment of IP addresses
- DHCP can interoperate with BOOTP clients

DHCP Interaction (simplified)



DHCP Message Format

1 (Request), 2 (Reply)

1, 6 (Ethernet)

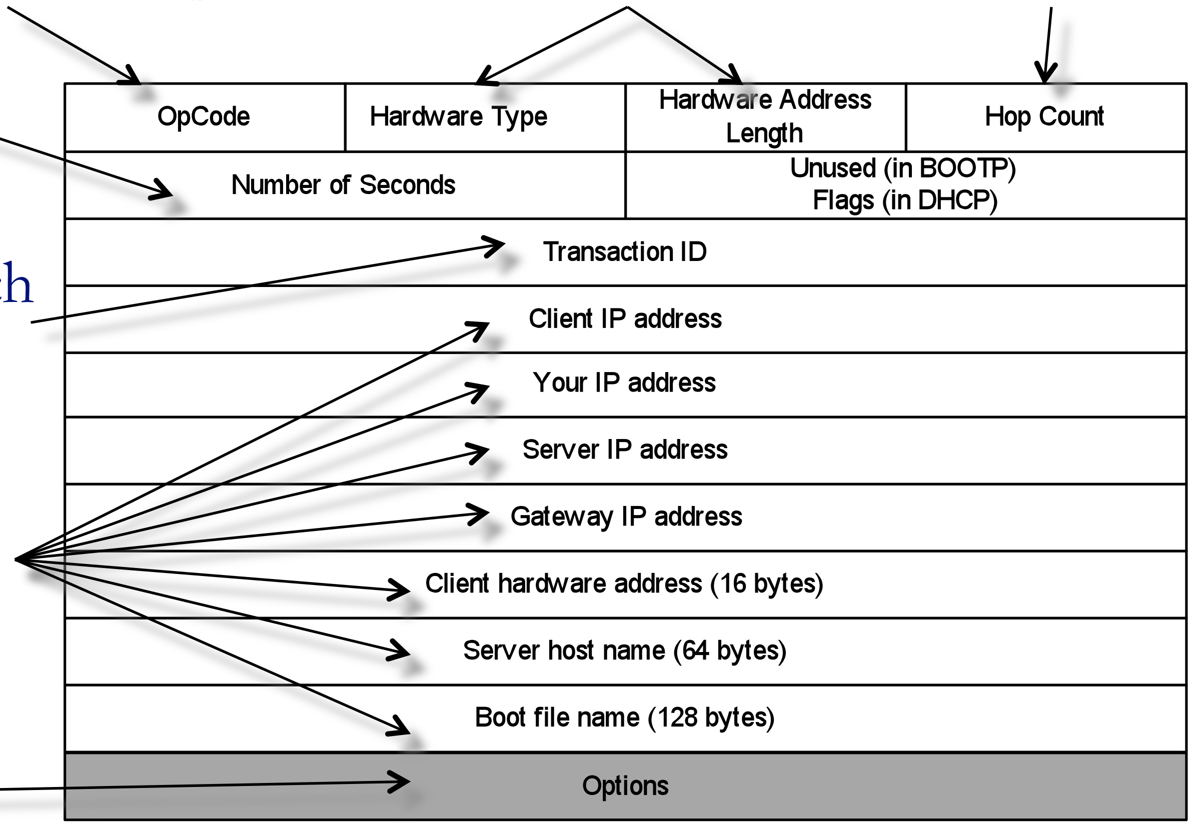
0, usually

seconds since client started to boot

Random. Used to match request to response

Client fills in the info that it has, leaves rest blank

There are over 100 different options



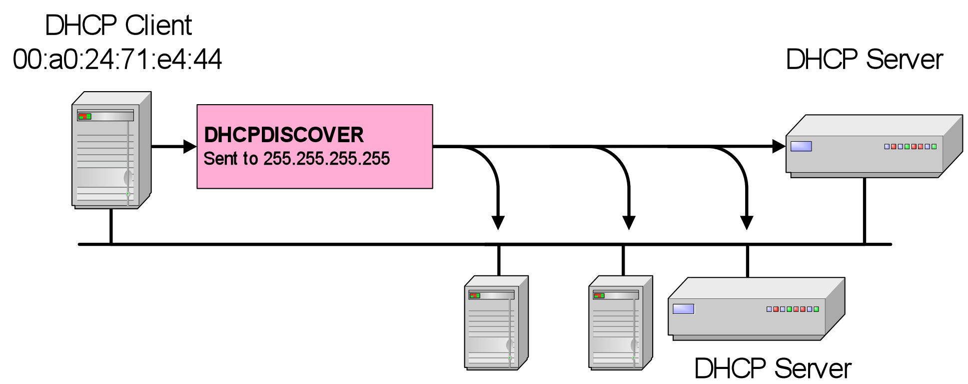
DHCP Message Type

- Message type is sent as an option.
- Other info in options:
  Subnet Mask, Name Server, Hostname, Domain Name, Forward On/Off, Default IP TTL, Broadcast Address, Static Route, Ethernet Encapsulation, X Window Manager, X Window Font, DHCP Msg Type, DHCP Renewal Time, DHCP Rebinding, Time SMTP-Server, SMTP-Server, Client FQDN, Printer Name, …

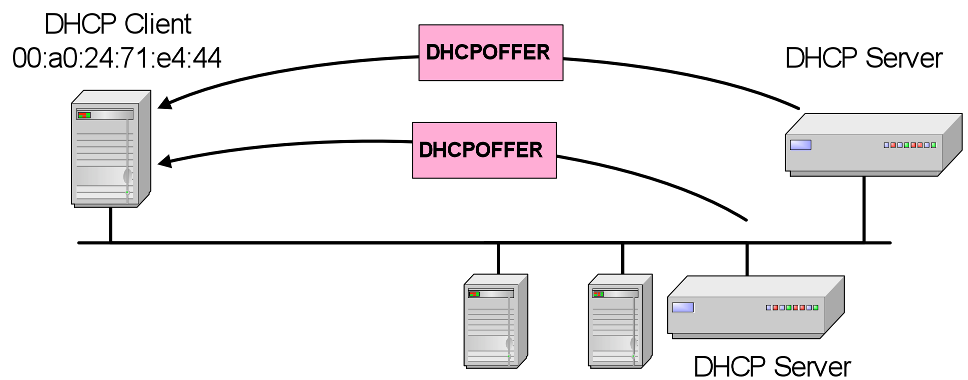
Value	Message Type
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNAK
7	DHCPRELEASE
8	DHCPINFORM

DHCP Operation

- DHCP DISCOVER
  - Can be relayed



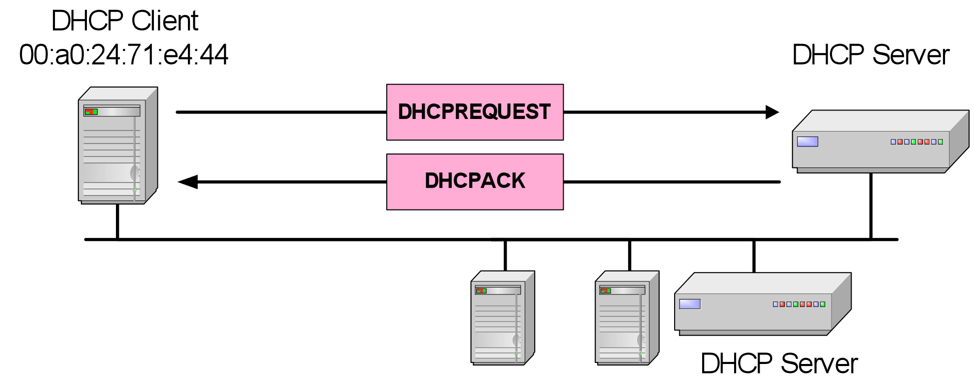
- DHCP OFFER



DHCP Operation

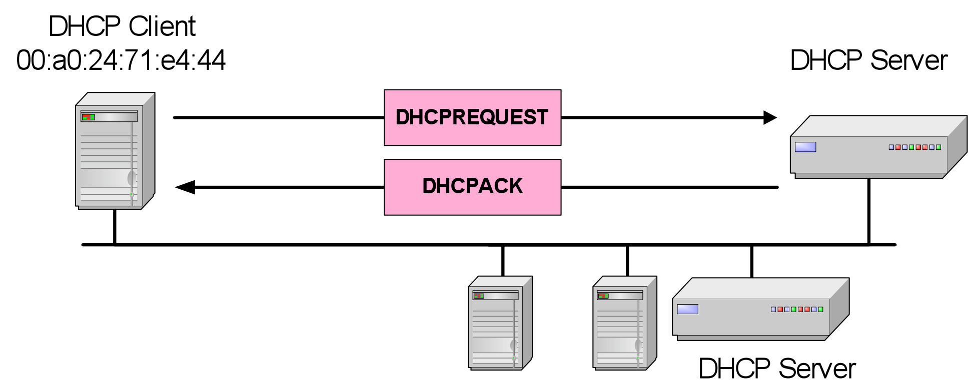
DHCP REQUEST
DHCP ACK

At this time, the DHCP client can start to use the IP address



Renewing a Lease
(sent when 50% of lease has expired)

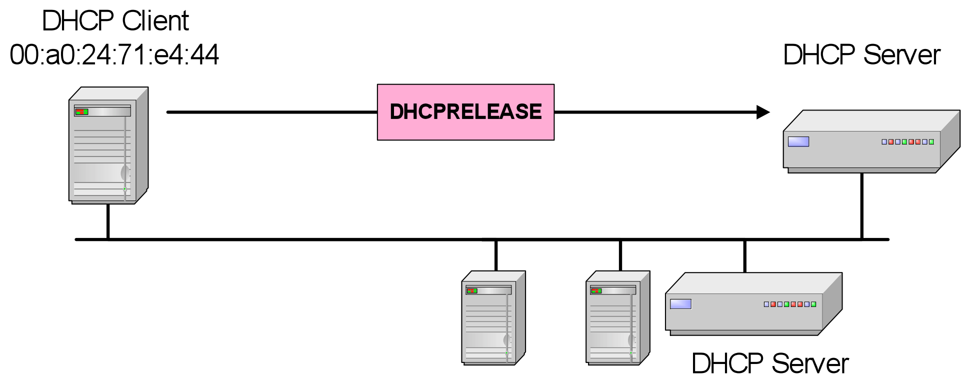
If DHCP server sends DHCPNACK, then address is released.



DHCP Operation

DCHP RELEASE

At this time, the DHCP client has released the IP address



Network Address Translation (NAT)

- NATs originally invented as a way to help migrate to a hybrid IPv4 IPv6 world
  - Took on a life of their own
  - May have substantially delayed IPv6 deployment by reducing address pressure!
  - You probably encounter them every day

NAT and Private Addresses

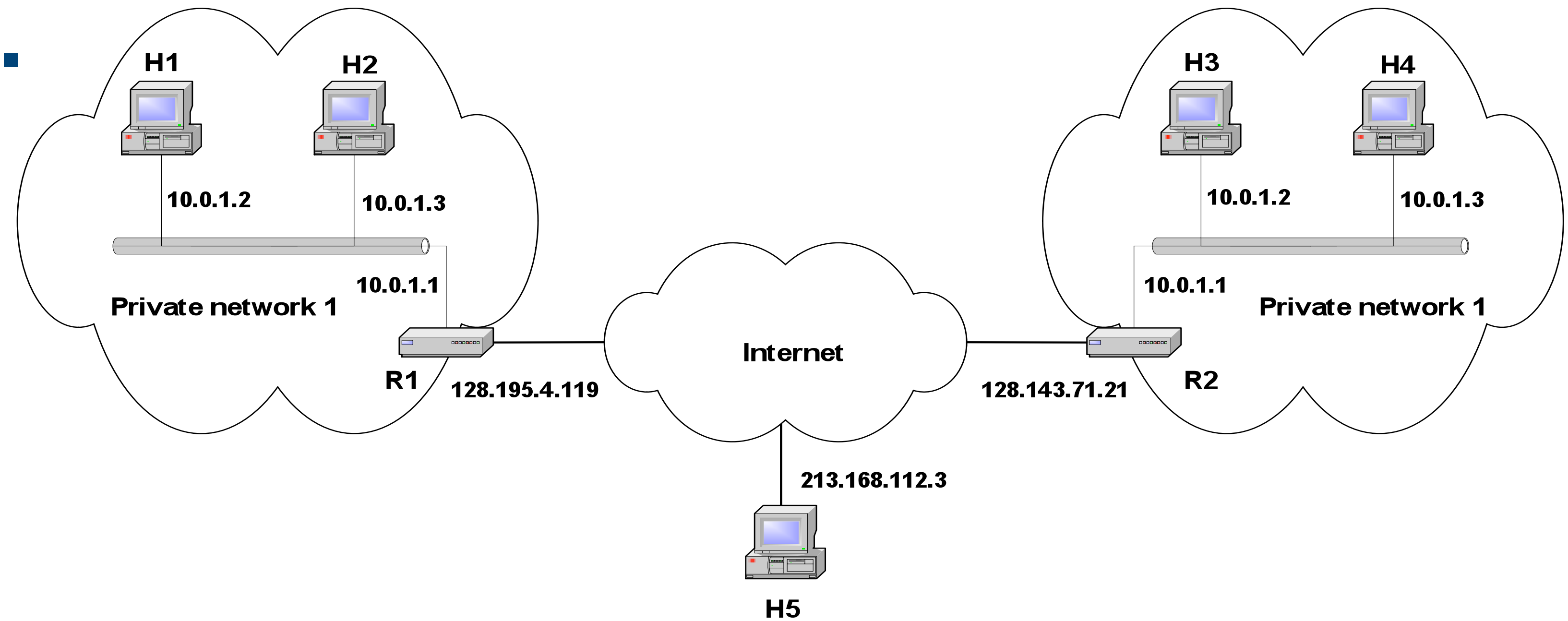
- *Private IP* network is an IP network that is not directly connected to the Internet

- IP addresses in a private network can be assigned arbitrarily.
  - Not registered and not guaranteed to be globally unique

- Generally, private networks use addresses from the following experimental address ranges (*non-routable addresses*):
  - 10.0.0.0 – 10.255.255.255
  - 172.16.0.0 – 172.31.255.255
  - 192.168.0.0 – 192.168.255.255

Network Address Translation

- Router function where IP addresses and port numbers of datagrams are replaced
  - NAT device has a translation table
- Enables hosts on private networks to communicate with hosts on the Internet
-



Main uses of NAT

- Pooling of IP addresses
  - Some corporate networks use pool of IP addresses to communicate with hosts on Internet
- Supporting migration between network service providers
  - Update of NAT to change provider, instead of changing all addresses on network
- IP masquerading
  - Single public IP address is mapped to multiple hosts in private network
- Load balancing of servers
  - Balance the load on a set of identical servers which are accessible from a single IP address

Concerns about NAT

- **Performance**
  - Changing the IP address requires NAT recalculates header checksum
  - Modifying port number requires NAT recalculates TCP checksum
- **Fragmentation**
  - Fragments should not be assigned different IP addresses or ports
- **End-to-end connectivity**
  - NAT destroys universal end-to-end reachability of hosts on the Internet.
  - A host in the public Internet often cannot initiate communication to a host in a private network.
- **IP address in application data**
  - Applications that carry IP addresses in IP payload generally do not work across a private-public network boundary
  - Some NAT devices inspect the payload of widely used application layer protocols and translate

NAT with FTP

- Client: USER anonymous
- Server: 331 Guest login ok, send your e-mail address as password.
- Client: PASS NcFTP@
- Server: 230 Logged in anonymously.
- Client: PORT 192,168,1,2,7,138
- Server: 200 PORT command successful.
- Client: LIST
- Server: 150 Opening ASCII mode data connection for /bin/ls.
- Server: 226 Listing completed.
- Client: QUIT
- Server: 221 Goodbye.

The client wants the server to send to port number 1930 on IP address 192.168.1.2

The server would connect out from port 21 to port 1930 on 192.168.1.2

Key Concepts

- Network layer provides end-to-end data delivery across an internetwork, not just a LAN

  - Datagram and virtual circuit service models
  - IP/ICMP is the network layer protocol of the Internet

  - Important support protocols and techniques: ARP, DHCP, NAT

- Next topic: More detailed look at routing and addressing