Internet Protocol

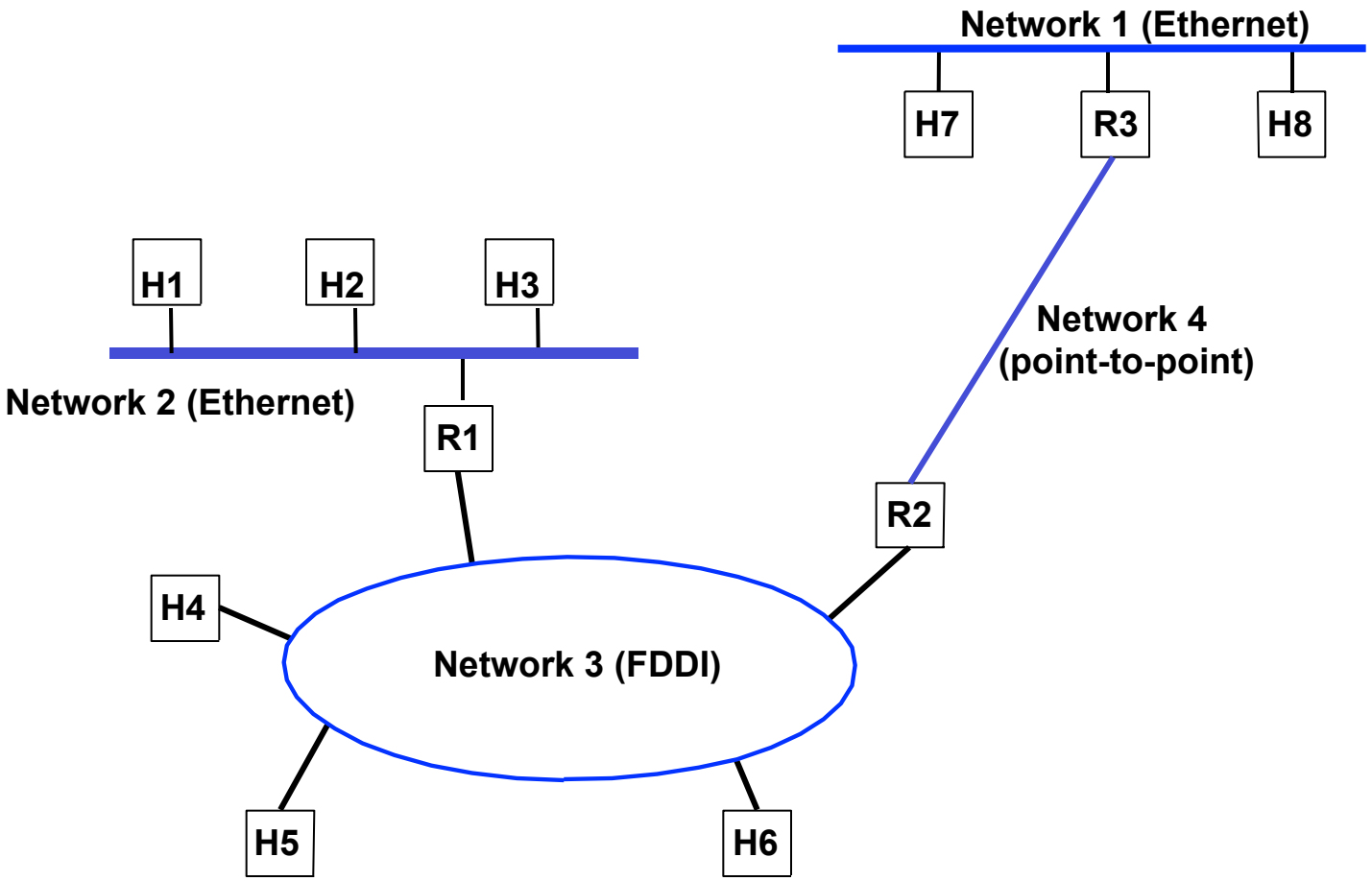
Outline

- Introduction to Internet Protocol
- Header and address formats
- ICMP
- Tools

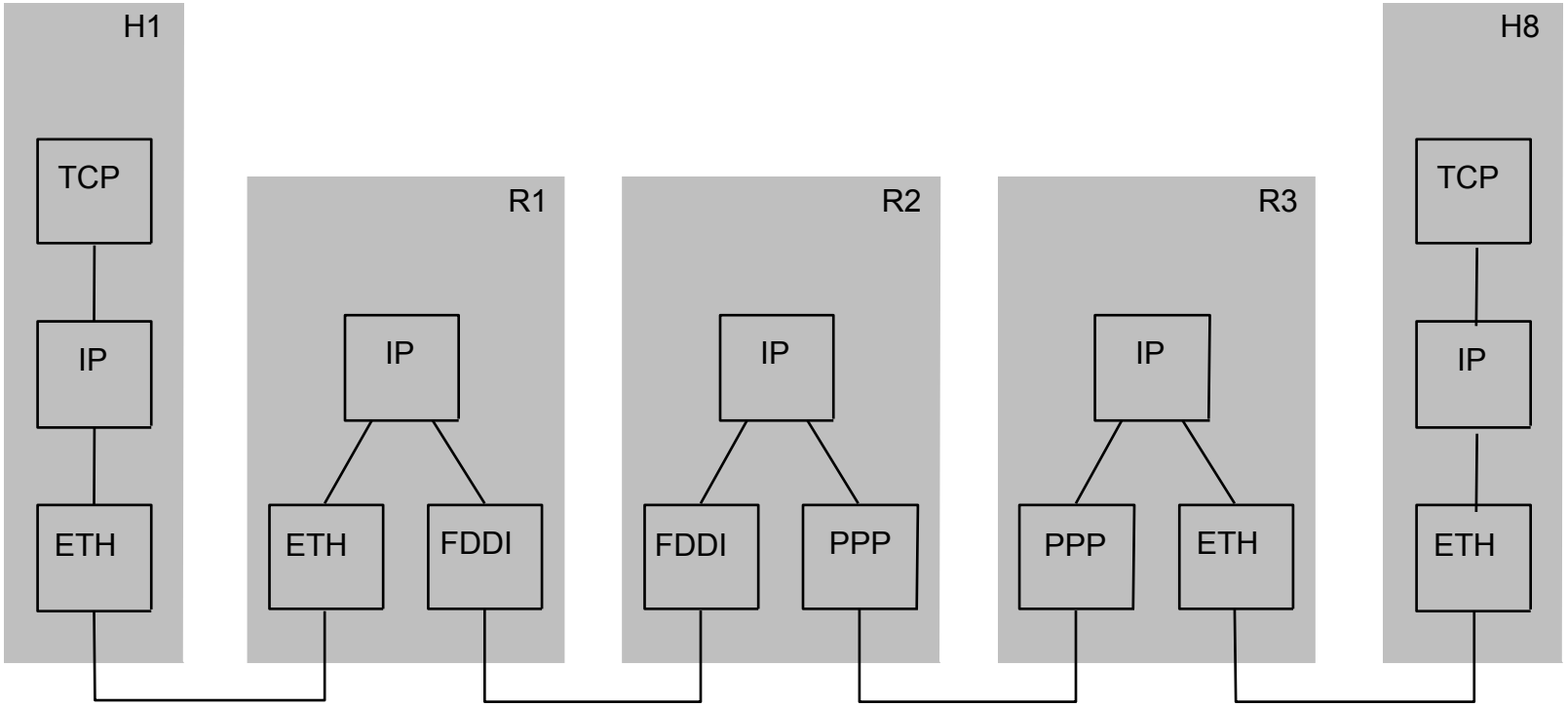
Internet Protocol

- Runs on all hosts in the Internet and enables packets to be routed between systems
  - Key protocol for building networks
  - Kahn-Cerf
- Datagram delivery of packets
  - Connectionless and based on routing protocols
- Well defined packet format
- Global addressing
  - Means for identifying Internet hosts
- Fragmentation and reassembly
  - Since packets can be of varying size
- Error reporting

An IP Internet – Network of Networks

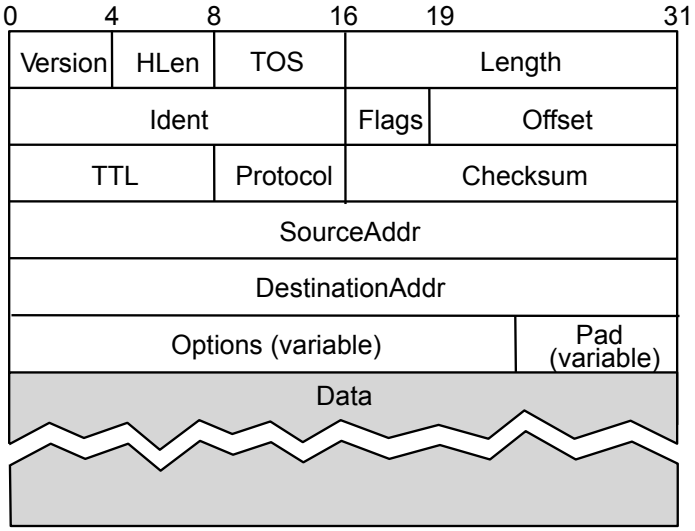


Protocol Stack – IP is Common to All

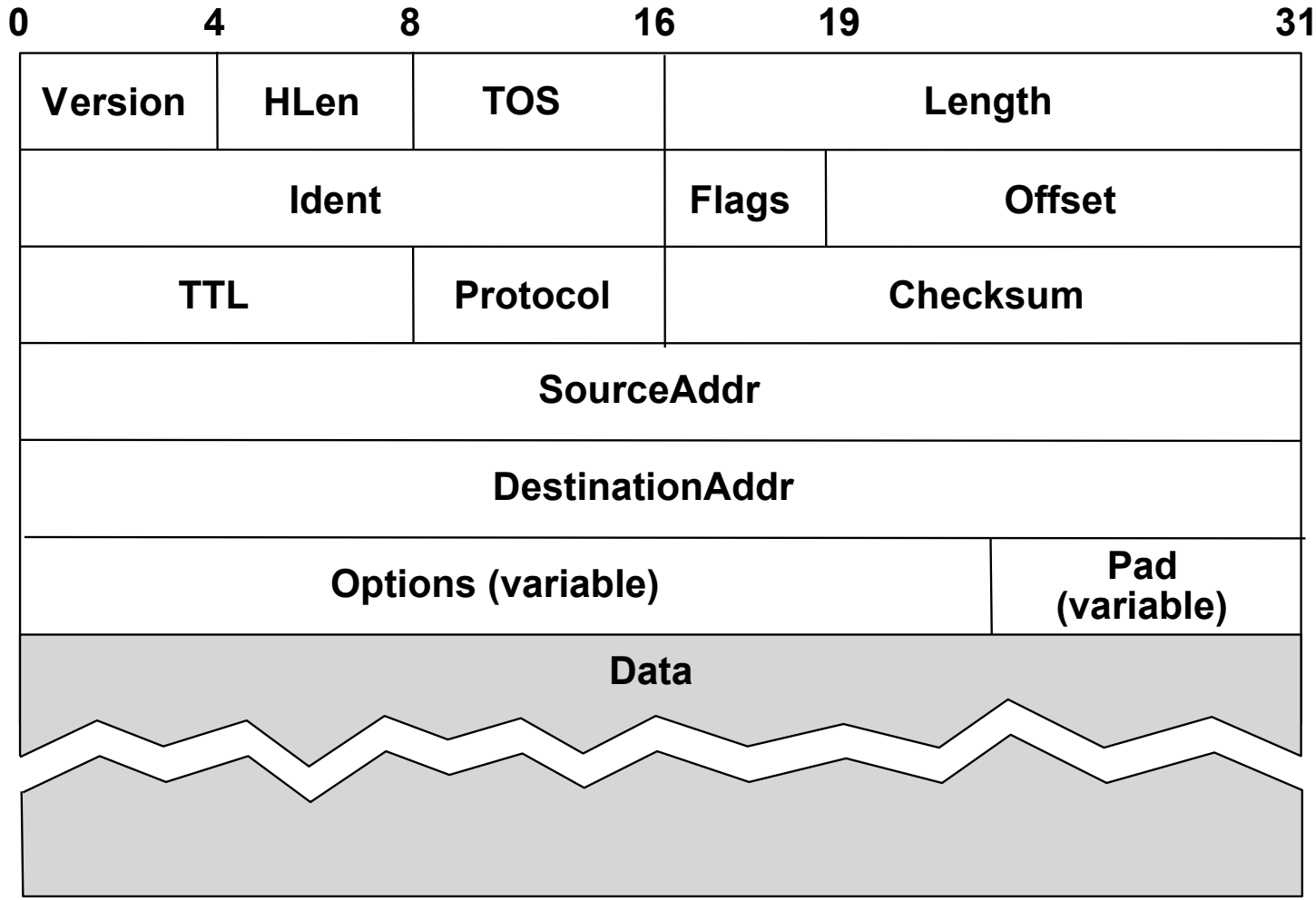


Service Model

- Connectionless (datagram-based)
- Best-effort delivery (unreliable service)
  - packets are lost
  - packets are delivered out of order
  - duplicate copies of a packet are delivered
  - packets can be delayed for a long time
- Datagram format



IPv4 Header Format



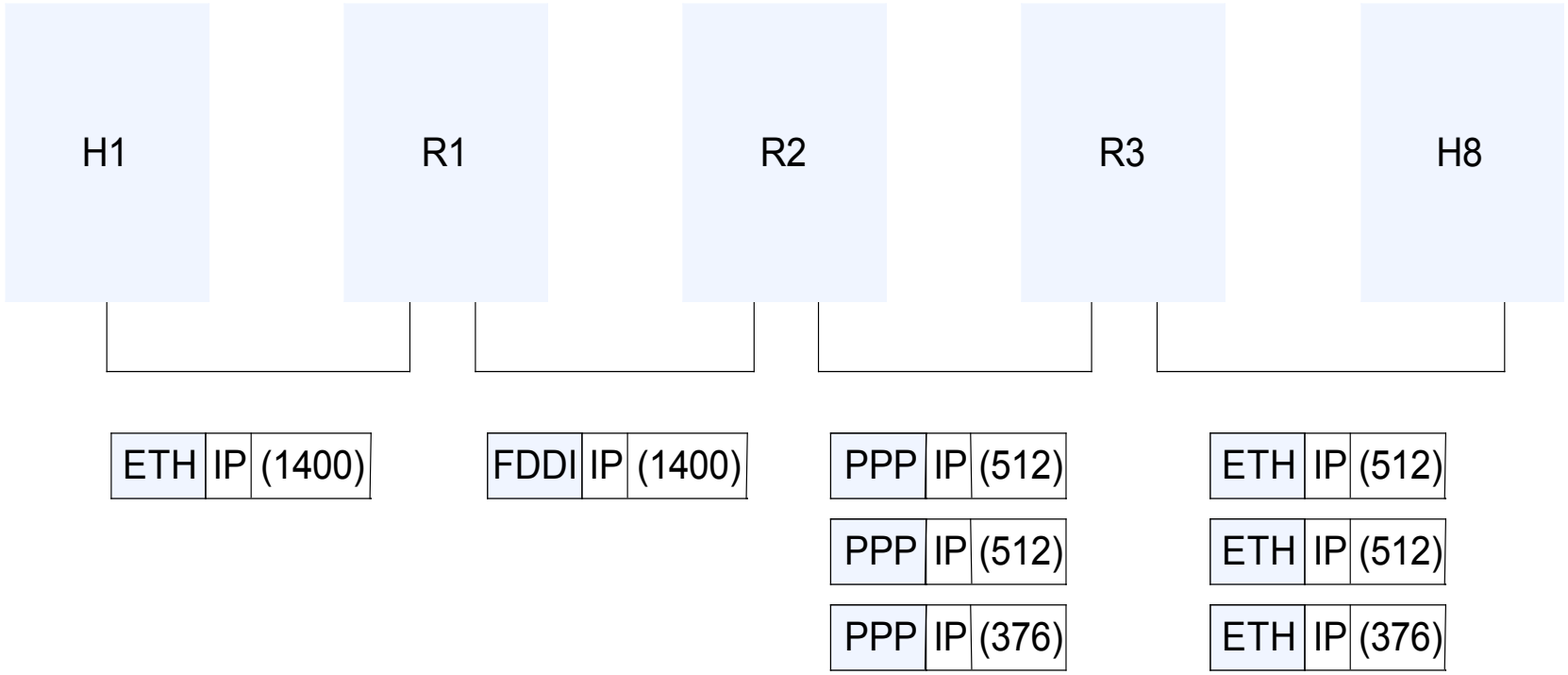
Various fields

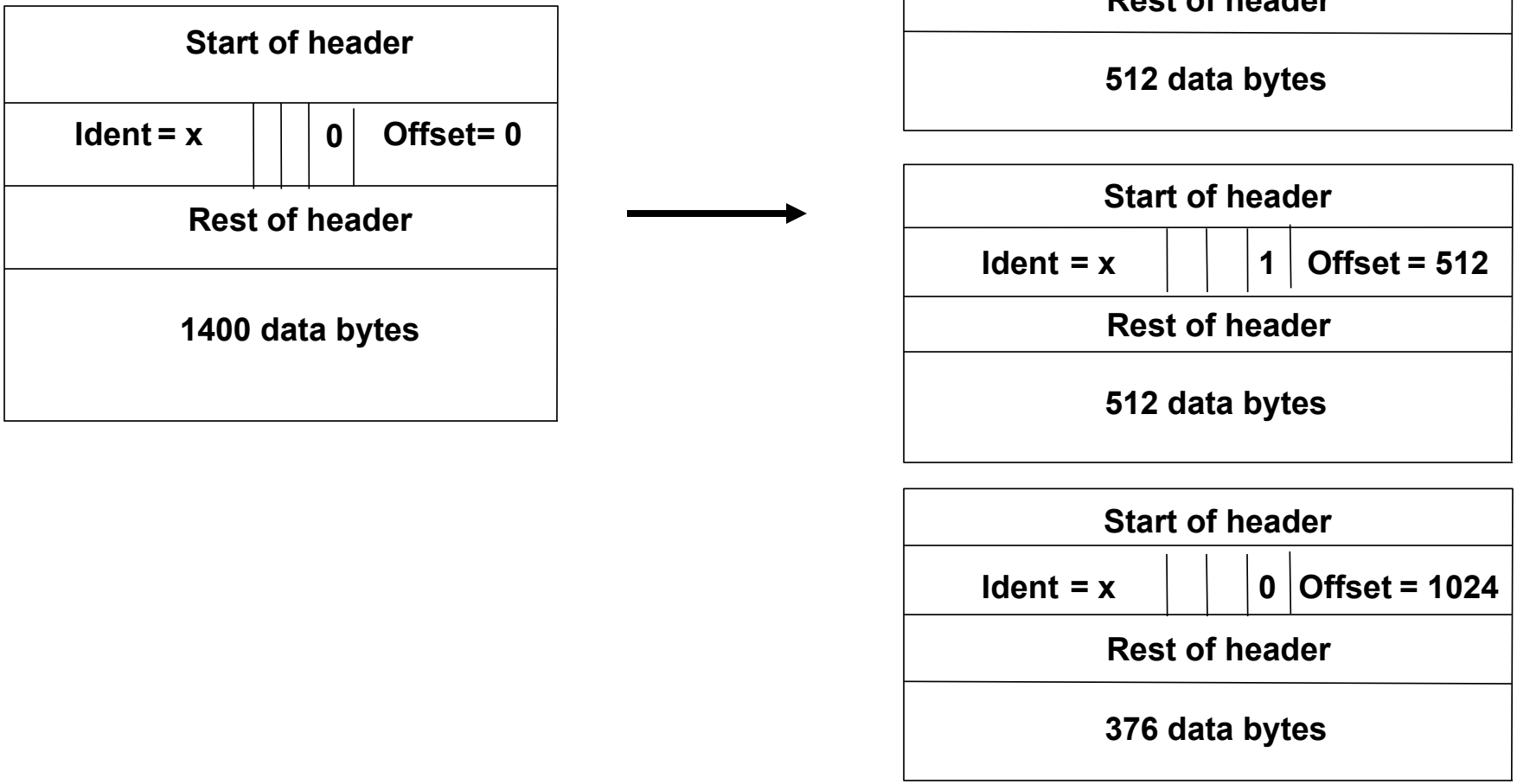
- Hlen --- length of the header as multiple of 32-bits
- Length --- header+data as number of octets (bytes)
- Ident --- unique ID field for each packet
- Time To Live (TTL) --- an integer value that is decremented in each hop, and packets are discarded when TTL is 0
- Offset --- if a packet gets fragmented, it is the position of this fragment relative to other fragments (note it is 13 bits only, why?)

Fragmentation and Reassembly

- Each network has some Maximum Transmission Unit (MTU)
  - Largest datagram that a network can carry in a frame
- Strategy
  - fragment when necessary (MTU < Datagram)
  - try to avoid fragmentation at source host
    - Due to overhead of reassembly
  - re-fragmentation is possible
  - fragments are self-contained datagrams
  - delay reassembly until destination host
    - Keep this functionality out of the network
  - do not recover from lost fragments
    - End hosts try to reassemble fragmented packets –if a fragment is lost…
- End hosts are encouraged to do MTU discovery

Example


H1
R1
R2
R3
H8
ETH IP (1400)
FDDI IP (1400)
PPP IP (512)
PPP IP (512)
PPP IP (376)
ETH IP (512)
ETH IP (512)
ETH IP (376)

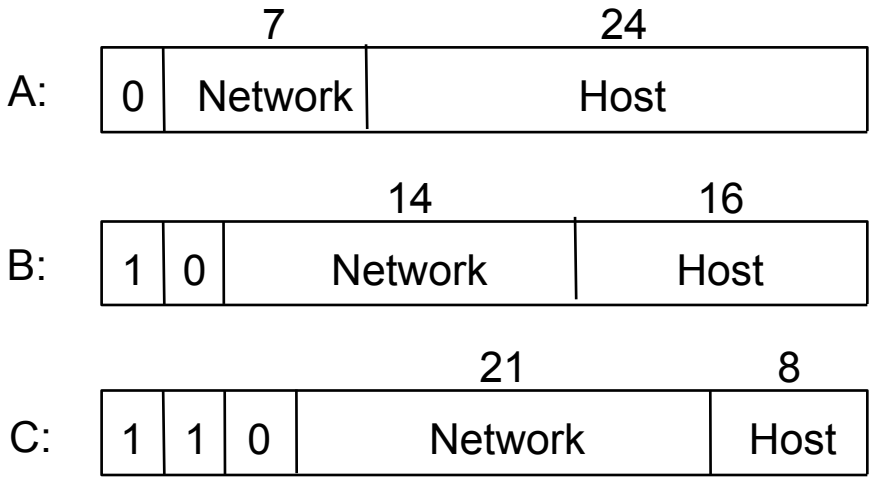

Start of header
Ident = x
0
Offset= 0
Rest of header
1400 data bytes
Start of header
Ident = x
1
Offset = 0
Rest of header
512 data bytes
Start of header
Ident = x
1
Offset = 512
Rest of header
512 data bytes
Start of header
Ident = x
0
Offset = 1024
Rest of header
376 data bytes

IPv4 Global Addresses

- Properties
  - globally unique
  - hierarchical: network + host

- Dot Notation
  - 10.3.2.4
  - 128.96.33.81
  - 192.12.69.77

- Classes A, B, C
- Class D: Multicast
- Class E: reserved



Originally, address blocks had classes --- A, B, C, D, E

Datagram Forwarding

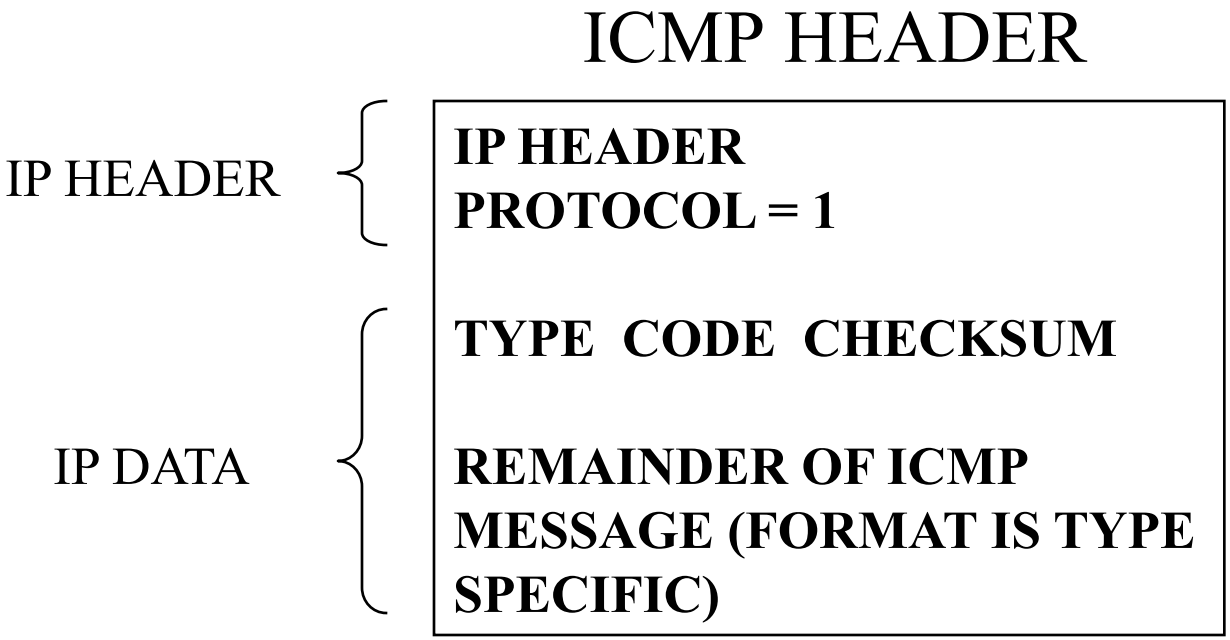
- Every datagram contains destination's address
- The "network part" of an IP address uniquely identifies a single physical network
- If directly connected to destination network, then forward to host
- If not directly connected to destination network, then forward to some router
- Forwarding table maps network number into next hop
  - Mapping is based on routing algorithm
- Each host has a default router
- Each router maintains a forwarding table

Internet Control Message Protocol (ICMP)

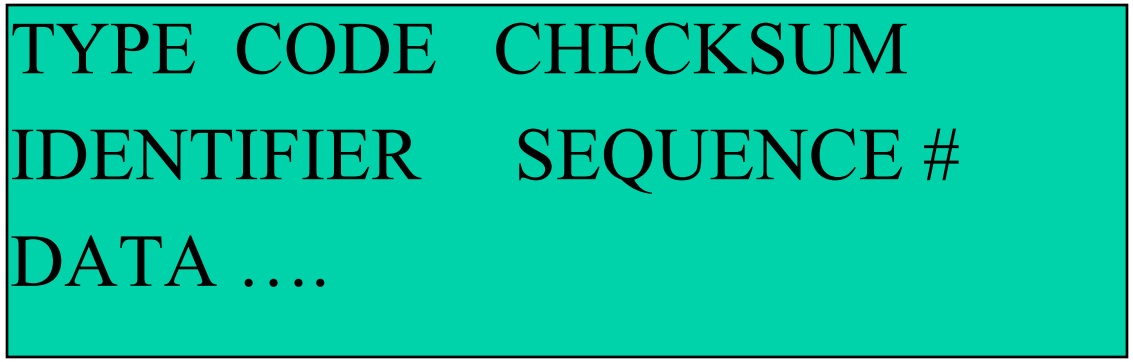
- Echo (ping)
- Redirect (from router to source host)
- Destination unreachable (protocol, port, or host)
- TTL exceeded (so datagrams don't cycle forever)
- Checksum failed
- Reassembly failed
- Cannot fragment

ICMP

- Uses IP but is a separate protocol in the network layer



Echo and Echo Reply



TYPE: 8 = ECHO, 0 = ECHO REPLY CODE; CODE = 0

IDENTIFIER

An identifier to aid in matching echoes and replies

SEQUENCE #

Same use as for IDENTIFIER

UNIX “ping” uses echo/echo reply

Ping Example

C:\WINDOWS\Desktop>ping www.soi.wide.ad.jp

Pinging asari.soi.wide.ad.jp [203.178.137.88] with 32 bytes of data:
Reply from 203.178.137.88: bytes=32 time=253ms TTL=240
Reply from 203.178.137.88: bytes=32 time=231ms TTL=240
Reply from 203.178.137.88: bytes=32 time=225ms TTL=240
Reply from 203.178.137.88: bytes=32 time=214ms TTL=240

Ping statistics for 203.178.137.88:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 214ms, Maximum = 253ms, Average = 230ms

Redirect when no route to Destination

TYPE CODE CHECKSUM
NEW ROUTER ADDRESS
IP HEADER + 64 bits data
from original DG

TYPE = 5
CODE =

- 0 = Network redirect
- 1 = Host redirect
- 2 = Network redirect for specific TOS
- 3 = Host redirect for specific TOS

Destination Unreachable

TYPE CODE CHECKSUM
UNUSED
IP HEADER + 64 bits data from original DG

TYPE = 3
CODE 0 = Net unreachable
1 = Host unreachable
2= Protocol unreachable
3 = Port unreachable
4 = Fragmentation needed but DF set
5 = Source route failed

Time Exceeded

TYPE CODE CHECKSUM
UNUSED
IP HEADER + 64 bits data from original DG

TYPE = 11
CODE
0 = Time to live exceeded in transit
1 = Fragment reassembly time exceeded

Source Quench

TYPE CODE CHECKSUM
UNUSED
IP HEADER + 64 bits data from original DG

TYPE = 4; CODE = 0

Indicates that a router has dropped the original DG or may indicate that a router is approaching its capacity limit.

Correct behavior for source host is not defined.

Traceroute

- UNIX utility - displays router used to get to a specified Internet Host
- Operation
  - router sends ICMP Time Exceeded message to source if TTL is decremented to 0
  - if TTL starts at 5, source host will receive Time Exceeded message from router that is 5 hopes away
- Traceroute sends a series of probes with different TTL values… and records the source address of the ICMP Time Exceeded message for each
- Probes are formatted to that the destination host will send an ICMP Port Unreachable message

TraceRoute Example

```
C:\windows\desktop> tracert www.soi.wide.ad.jp
Tracing route to asari.soi.wide.ad.jp [203.178.137.88]
over a maximum of 30 hops:
  1    19 ms    27 ms    23 ms  208.166.201.1
  2    17 ms    13 ms    14 ms  204.189.71.9
  3    25 ms    29 ms    29 ms  aar1-serial4-1-0-0.Minneapolismpn.cw.net [208.174.7.5]
  4    24 ms    27 ms    24 ms  acr1.Minneapolismpn.cw.net [208.174.2.61]
  5    26 ms    22 ms    23 ms  acr2-loopback.Chicagochd.cw.net [208.172.2.62]
  6    29 ms    29 ms    27 ms  cand-w-private-peering.Chicagochd.cw.net [208.172.1.222]
  7    28 ms    24 ms    28 ms  0.so-5-2-0.XL2.CHI2.ALTER.NET [152.63.68.6]
  8    26 ms    27 ms    28 ms  0.so-7-0-0.XR2.CHI2.ALTER.NET [152.63.67.134]
  9    25 ms    24 ms    26 ms  292.at-2-0-0.TR2.CHI4.ALTER.NET [152.63.64.234]
 10    73 ms    74 ms    73 ms  106.ATM7-0.TR2.LAX2.ALTER.NET [146.188.136.142]
 11    74 ms    76 ms    76 ms  198.ATM7-0.XR2.LAX4.ALTER.NET [146.188.249.5]
 12    73 ms    75 ms    77 ms  192.ATM5-0.GW9.LAX4.ALTER.NET [152.63.115.77]
 13    80 ms    73 ms    76 ms  kdd-gw.customer.ALTER.NET [157.130.226.14]
 14    84 ms    84 ms    91 ms  202.239.170.236
 15    97 ms    81 ms    86 ms  cisco1-eth-2-0.LosAngeles.wide.ad.jp [209.137.144.98]
 16   174 ms   174 ms   178 ms  cisco5.otemachi.wide.ad.jp [203.178.136.238]
 17   201 ms   196 ms   194 ms  cisco2.otemachi.wide.ad.jp [203.178.137.34]
 18   183 ms   182 ms   196 ms  foundry2.otemachi.wide.ad.jp [203.178.140.216]
 19   183 ms   185 ms   178 ms  gsr1.fujisawa.wide.ad.jp [203.178.138.252]
 20   213 ms   205 ms   201 ms  asari.soi.wide.ad.jp [203.178.137.88]
Trace complete.
```

Other interesting tools

- ping: estimate RTT between src-dest pairs
- ifconfig: check status of and configure different interfaces
- route (but you have to have route access)