The World According to Internet

Protocols

- rules by which active network elements communicate with each other is a *protocol*
- protocols = "algorithms + data structures"
  - formats of messages exchanged
  - actions taken on receipt of messages
  - how to handle errors
- hardware/operating-system independent
- real-life examples:
  - Robert's rules for meetings
  - conversational rules (interrupts, request for retransmission, …)

Some Terminology

internet: collection of packet switching networks interconnected by routers

(the) Internet: "public" interconnection of networks

end system = host: computer that is attached to the network ↔ router; usually *one* network interface

router = gateway = intermediate system: routes packets, several interfaces

subnetwork: part of an internet (e.g., single Ethernet)

firewall: router placed between an organization's internal internet and a connection to the external Internet, restricting packet flows to provide security.

What Do Protocols Do for a Living?

error control: make channel more reliable ➠ retransmission

resequencing: reorder out-of-sequence messages

flow control: avoid flooding slower receiver

congestion control: avoid flooding slower network

fragmentation: divide large message into smaller chunks to fit lower layer

multiplexing: combine several higher-layer sessions into one "channel"

addressing/naming: manage identifiers

compression: reduce data rate

privacy, authentication: even if somebody else is listening

resource allocation: bandwidth, buffers among contenders

Protocol Layering

send side layer N takes protocol data (PDU) from layer $N+1$, adds header, and passed to $N-1$

receive side layer N takes PDU from $N-1$, strips N headers, processes, and passes rest to $N+1$

Routers and Bridges

host: all layers

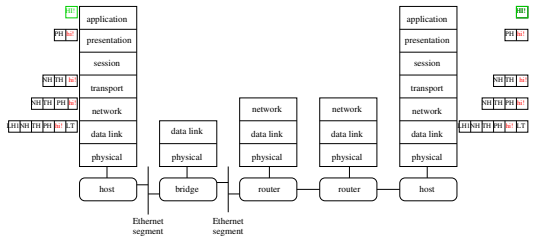
router: modifies data link headers/address, may touch network headers (IP options!)

bridge: may modify data link header

repeater: physical layer

➠ IP packet maintains same source and destination addresses end-to-end, but gets many different link headers/trailers and addresses

Protocol Layering



TH (transport header): sequence numbers, error detection, timestamp information ➠ end-to-end

NH (network header): source and destination address, hop counts

LHn (link header): error detection, hop-by-hop error control

Layering Considered Harmful?

- need layers to manage complexity ➠ don't want to reinvent Ethernet-specific protocol for each application
- common functionality → "ideal" network

but:

- layer N may duplicate lower layer functionality (error recovery)
- different layers may need same information
- layer N may need to peek into layer $N-2$ (e.g., fragmentation)
- implementation issues: avoid copying

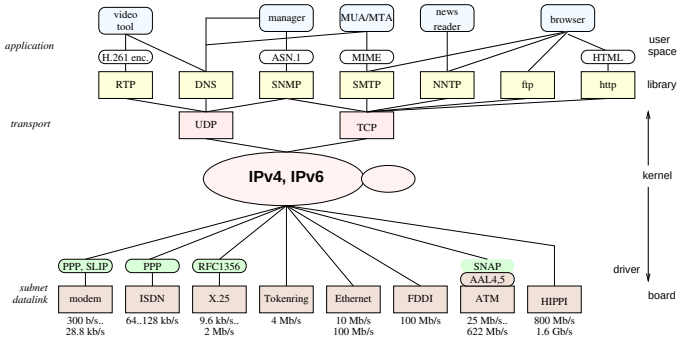
Internet View of the World

"anything over IP, IP over anything"

- subnetwork: ATM, Ethernet, ISDN (with PPP, SLIP)
- network layer: IP, IPng, (CLNP?)
- transport: UDP, TCP, …
- application: http, ftp, telnet, RTP, …
- control: RSVP
- management: SNMP
- directory: DNS

➠ no session, presentation; also: OSI, AppleTalk over IP

The Internet Protocol Zoo



Subnetwork Technologies

Some examples:

technology	bandwidth	WAN, LAN
ATM	25 Mb/s …2.4 Gb/s	WAN
leased line	56 kb/s, 1.5 Mb/s (T1), 2.0 Mb/s (E1)	WAN
satellite	2.4 kb/s …Mb/s	WAN
Ethernet	10 Mb/s, 100 Mb/s	LAN
Tokenring	4 Mb/s, 10 Mb/s	LAN
ISDN	64 kb/s	LAN
POTS modem	2.4 …28.8 kb/s	LAN

Refresher: Ethernet

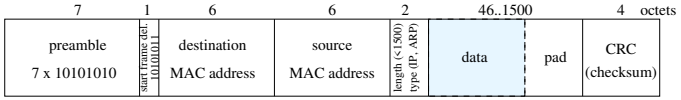
- multiple access network
- 10 Mb/s "raw" speed (new: 100 Mb/s *Fast Ethernet*)
- media: coaxial cable, fiber, UTP-5 (unshielded twisted pair)
- cocktail party protocol: listen, transmit, back off

```
L = 1
start:
  if (nobody else transmitting)
    transmit
    if (collision detected)
      stop transmission immediately
      L *= 2
      wait random period of time (0, L) minislots; goto start
  else
    wait random period of time; goto start
```

Ethernet Packet

preamble: 7 bytes for clock synchronization

length/frame type: 2 bytes, < 1500; or IPv4: 0x0800; ARP: 0x0806



Names, Addresses, Routes

Shoch (1979):

Name identifies what you want,

Address identifies where it is,

Route identifies a way to get there.

Saltzer (1982):

Service and users: time of day, routing, …

Nodes: end systems and routers

Network attachment point: ≥ 1 per node ➠ multihomed host vs. router

Paths: traversal of nodes and links

binding = (temporary) equivalence of two names

Names and Addresses

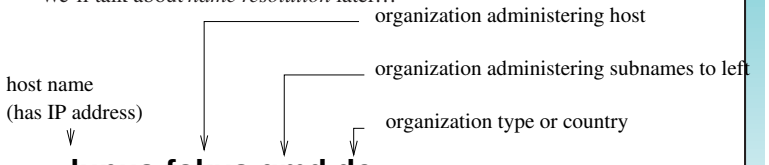
Internet Names and Addresses

	example	organization
MAC address	8:0:20:72:93:18	flat, permanent
IP address	132.151.1.35	topological (mostly)
Host name	www.ietf.org	hierarchical

host name $\overset{\mathrm{DNS,many-to-many}}{\longrightarrow}$ IP address $\overset{\mathrm{ARP,1-to-1}}{\longrightarrow}$ MAC address

The Internet Domain Name System

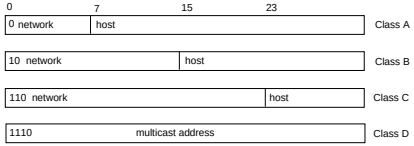
We'll talk about *name resolution* later…



Anywhere from two to ∞ parts

IP Addresses

Each Internet host has one or more globally unique 32-bit IP addresses, consisting of a network number and a host number:



- two-level hierarch → three-level (later)
- an IP address identifies an *interface*, not a host!
- a host may have two or more addresses. Why?
- net 10: reserved for internal use

The Internet Domain Name System

2 letters: countries
3 letters: independent of geography (except edu, gov, mil)

domain	usage	example
com	business (global)	research.att.com
edu	U.S. higher education	cs.umass.edu
gov	U.S. non-military gov't	whitehouse.gov
mil	U.S. military	arpa.mil
org	non-profit organization (global)	www.ietf.org
net	network provider	nis.nsf.net
us	U.S. geographical	ietf.cnri.reston.va.us
de	Germany	fokus.gmd.de
uk	United Kingdom	cs.ucl.ac.uk

IP Addresses

- dotted decimal notation: 4 decimal integers, each specifying one byte of IP address:
  host name lupus.fokus.gmd.de
  32-bit address 1100 0000 0010 0011 1001 0101 0011 0100
  dotted decimal 192.35.149.52
- loopback: 127.0.0.1 (packets never appear on network)
- own network (broadcast): hostid = 0; own host: netid = 0
- directed broadcast: hostid = all ones
- local broadcast: 255.255.255.255

IP Address Classes

class	first	hosts per network	nets	used (1.96)
Class A	< 128	16 mio.	128	92
Class B	$128 \ldots 191$	65534	16384	5655
Class C	$192 \ldots 223$	254	2 mio.	87924
Class D	$224 \ldots 239$		268 mio.	dynamic
Class E	$240 \ldots 255$		134 mio.	reserved

Problems with IP Addresses

- if a host moves from one network to another, its IP address changes
- currently, mostly assigned without regards to topology → too many networks ➠ CIDR
- limited space ➠ IPv6
- class thresholds: class C net grows beyond 254 hosts
- hard to change: hidden in lots of places
- multihomed host: path taken to host depends on destination address

Example: ifconfig

```
ifconfig -a
le0:  flags=863<UP,BROADCAST,NOTRAILERS,RUNNING>
        inet 192.35.149.117 netmask ffffff00 broadcast 192.35.149.0
fa0:  flags=863<UP,BROADCAST,NOTRAILERS,RUNNING>
        inet 194.94.246.72 netmask ffffff00 broadcast 194.94.246.0
qaa0: flags=61<UP,NOTRAILERS,RUNNING>
        inet 193.175.134.117 netmask ffffff00
qaa1: flags=61<UP,NOTRAILERS,RUNNING>
        inet 129.26.216.231 netmask ffff0000
qaa2: flags=60<NOTRAILERS,RUNNING>
qaa3: flags=60<NOTRAILERS,RUNNING>
lo0:  flags=849<UP,LOOPBACK,RUNNING>
        inet 127.0.0.1 netmask ff000000
```


CIDR: Classless Interdomain Routing

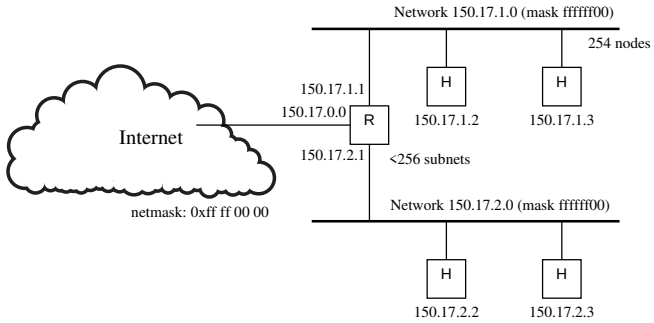
- problem: too many networks ➠ routing table explosion
- problem: class C too small, class B too big (and scarce)
- discard class boundaries → supernetting
- ISP assigns a contiguous group of 2^n class C blocks
- "longest match routing" on masked address; e.g. 192.175.132.0/22

address/mask	next hop
192.175.132.0/22	1
192.175.133.0/23	2
192.175.128.0/17	3

- e.g.,: all sites in Europe common prefix ➠ only single entry in most U.S. routers

Subnetting

- large organizations: multiple LANs with single IP network address
- subdivide "host" part of network address ➠ subnetting

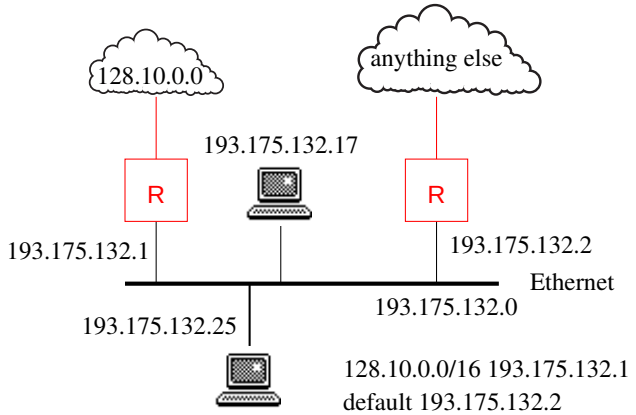


IP Forwarding

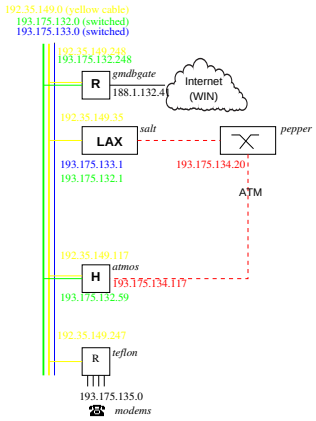
```
get destination IP address D
if network(D) == directly attached network {
  ARP: D -> MAC address
  put in link layer frame
  forward
else
  foreach entry in routing table {
    if (D & subnet mask) == network(entry) {
      get next hop address N
      ARP: N -> MAC address
      put in link layer frame
      forward
    }
  }
}
```

➠ IP source/destination remains same, MAC changes

IP Forwarding



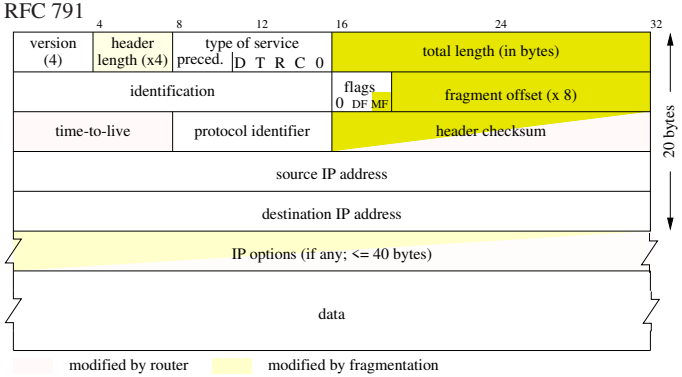
GMD Fokus Network



The Internet Protocol

IPv4 Header

RFC 791



IPv4 Service Model

datagram: each packet is independent of all others

best effort: packet may arrive *or not* after some time

IPv4

version: always 4

TOS (type of service): precedence (3 bits) and “minimize delay”, “maximize throughput”, “maximize reliability”, “minimize cost” bits ➠ rarely used

identifier: identifier, different for each packet from host

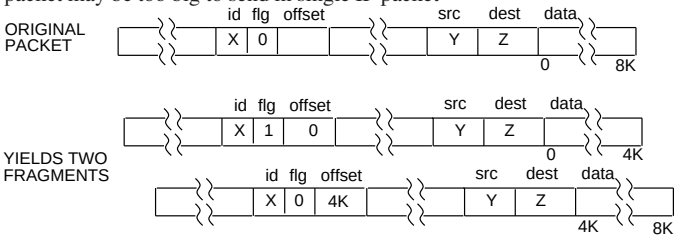
TTL: time to live field; initialized to 64; decremented at each router ➠ drop if TTL = 0 (prevent loops!)

protocol: next higher protocol (TCP: 6, UDP: 17)

header checksum: add together 16-bit words using one’s complement ➠ optimized for software

IP Fragmentation and Reassembly

data link protocol may limit packets $< 65,536$ bytes ➠ transport layer packet may be too big to send in single IP packet



IP Fragmentation and Reassembly

➠ split TPDU into *fragments*

- each fragment becomes its own IP packet (routers don't care)
- each fragment has same identifier, source, destination address
- fragment offset field gives offset of data from start of original packet
- *more fragments* (MF) flag of 0 if last (or only) fragment of packet
- fragments reassembled only at final destination
- routers must handle at least 576 bytes
- *do not fragment* bit prevents fragmentation ➠ drop + error message
- avoid multiple fragmentation (1500 → 620) ➠ MTU discovery

IP Options

Extend functionality of IP without carrying useless information:

- security and handling restrictions for military
- determine route (source route)
- record route
- record route and timestamps

(rarely used ↔ not all routers support them)

IP Record Route Option

- source creates empty list of ≤ 9 IP addresses
- option: length, pointer, list of IP addresses
- routers note outgoing interface in list
- …and bump pointer

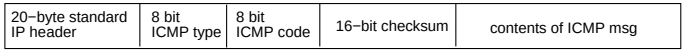
IP Source Route Option

- source determines path taken by packet (≤ 9 hops)
- *loose*: any number of hops in between
- *strict*: every hop; if not directly connected, discard
- same format as record route option
- router overwrites with address of outgoing interface
- must be copied to fragments
- destination should reverse route for return packets
- not too popular ➠ router performance ↓

type	code	description
0	0	echo reply (to a ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	4	fragmentation needed and DF set
3	6	destination network unknown
3	7	destination host unknown
3	…	other reasons
4	0	source quench (slow down)
5	1	redirect message to host
8	0	echo request (ping)
9	0	IS-ES router advertisement (new)
10	0	ES-IS router discovery (new)
11	0	time exceeded = TTL zero
12	0	IP header bad
17	0	address (subnet) mask request
18	0	address (subnet) mask reply

ICMP

- used to communicate network-level error conditions and info to IP/TCP/UDP entities or user processes
- often considered part of the IP layer, but
  - IP demultiplexes up to ICMP using IP protocol field
  - ICMP messages sent within IP datagram
- ICMP contents always contain IP header and first 8 bytes of IP contents that caused ICMP error message to be generated



ping

- checks if host is reachable, alive
- uses ICMP echo request/reply
- copy packet data request → reply

```
ping -s gaia.cs.umass.edu
PING gaia.cs.umass.edu: 56 data bytes
64 bytes from gaia.cs.umass.edu (128.119.40.186): icmp_seq=0 time=276 ms
64 bytes from gaia.cs.umass.edu (128.119.40.186): icmp_seq=1 time=281 ms
64 bytes from gaia.cs.umass.edu (128.119.40.186): icmp_seq=2 time=276 ms
^C
----gaia.cs.umass.edu PING Statistics----
4 packets transmitted, 3 packets received, 25% packet loss
round-trip (ms)  min/avg/max = 276/277/281
```

traceroute

- allows to follow path taken by packet
- send UDP to unlikely port; 'time exceeded' and 'port unreachable' ICMP replies
- can use source route (-g), but often doesn't work

```
$ traceroute gaia.cs.umass.edu
 1  gmdbgate (192.35.149.248)  6 ms  2 ms  2 ms
 2  188.1.132.142 (188.1.132.142)  263 ms  178 ms  188 ms
 3  gmdisgate.gmd.de (192.54.35.68)  153 ms  187 ms  151 ms
 4  icm-bonn-1.gmd.de (192.76.246.17)  226 ms  207 ms  242 ms
 5  icm-dc-1-S2/6-512k.icp.net (192.157.65.209)  320 ms  315 ms  393 ms
 6  icm-mae-e-H1/0-T3.icp.net (198.67.131.9)  372 ms  297 ms  354 ms
 7  mae-east (192.41.177.180)  456 ms  537 ms  401 ms
 8  borderx2-hssi2-0.Washington.mci.net (204.70.74.117)  529 ms  385 ms  340 ms
 9  core-fddi-1.Washington.mci.net (204.70.3.1)  437 ms  554 ms  581 ms
10  core-hssi-3.NewYork.mci.net (204.70.1.6)  418 ms  547 ms  492 ms
11  core-hssi-3.Boston.mci.net (204.70.1.2)  453 ms  595 ms  724 ms
12  border1-fddi-0.Boston.mci.net (204.70.2.34)  789 ms  404 ms  354 ms
13  nearnet.Boston.mci.net (204.70.20.6)  393 ms  323 ms  346 ms
14  mit3-gw.near.net (192.233.33.10)  340 ms  465 ms  399 ms
15  umass1-gw.near.net (199.94.201.66)  557 ms  316 ms  369 ms
16  lgrc-gw.gw.umass.edu (192.80.83.1)  396 ms  309 ms  389 ms
17  cs-gw.cs.umass.edu (128.119.44.1)  276 ms  490 ms  307 ms
18  gaia.cs.umass.edu (128.119.40.186)  335 ms  317 ms  350 ms
```


ARP: IP address → MAC address

- for broadcast networks like Ethernet, token ring, …
- if MAC address unknown, send ARP request and hold on to packet
- ARP request → broadcast: sender IP, MAC; target IP, MAC
- *all* machines update their cache ➠ efficiency, allow change of interface
- ARP reply → requestor: reverse source/target; fill in source MAC
- directly on Ethernet, *not* IP!
- cache ARP replies; drop after 20 minutes

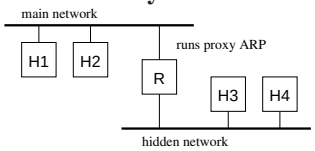
ARP example

```
rp -a
Net to Media Table
Device   IP Address               Mask      Flags   Phys Addr
------ -------------------- --------------- ----- ---------------
le0    hamlet               255.255.255.255       08:00:09:70:7d:
le0    gaia                 255.255.255.255       08:00:20:20:07:
le0    pern                 255.255.255.255       08:00:20:20:75:
le0    kite                 255.255.255.255       08:00:09:92:0d:
le0    condor               255.255.255.255       08:00:20:1c:95:
```


RARP: MAC → IP address

- determine IP address at boot for diskless workstations
- remember: MAC address is unique and permanent
- host broadcasts RARP request (with its own MAC address)
- RARP server responds with reply
- allows third-party queries
- want several servers for reliability

Proxy ARP



- extend network: router fronts for H3, H4
- router answers ARP requests for H3, H4 from H1, H2 with its *own* hardware address
- assumes trusting relationship
- only needs to be added to single router
- only works for broadcast networks