Internet Foundations

Internet Foundations

- brief review of history
- Internet standardization
- Internet architecture
- basic Internet protocols: IP, UDP, TCP, . . .

See http://www.cs.columbia.edu/~hgs/internet for resources.

Internet Standards

Who makes the rules?

ITU.T (itu.int): telecom standards by 16 study groups:

E.	Overall network operation, telephone service (E.164)
G.	transmission systems and media, digital systems and networks (G.711)
H.	Audiovisual and multimedia systems (H.323)
V.	Data communication over the telephone network (V.24)
X.	Data networks and open system communications (X.25)

IETF and IESG (ietf.org): (Internet Engineering Task Force, . . . Steering Group)
develop/bless protocols ("RFCs")
open admission, but not quite egalitarian

W3C: HTML, XML, . . .

Internet Operational Bodies

ISOC: membership organization; legal “home” of IETF

IANA: (Internet Assigned Numbers Authority) assigns numbers, top-level domains

NANOG: North American Network Operators Group

ICANN: administers IANA, registrars

RIPE, ARIN, APNIC: hands out blocks of addresses, regionally

IETF: WG + IESG + IAB

Internet Architecture Board: IAB

- architectural oversight
- process appeals
- elected by ISOC through nominations committee

Internet Engineering Steering Group (IESG): approves standards, composed of area directors

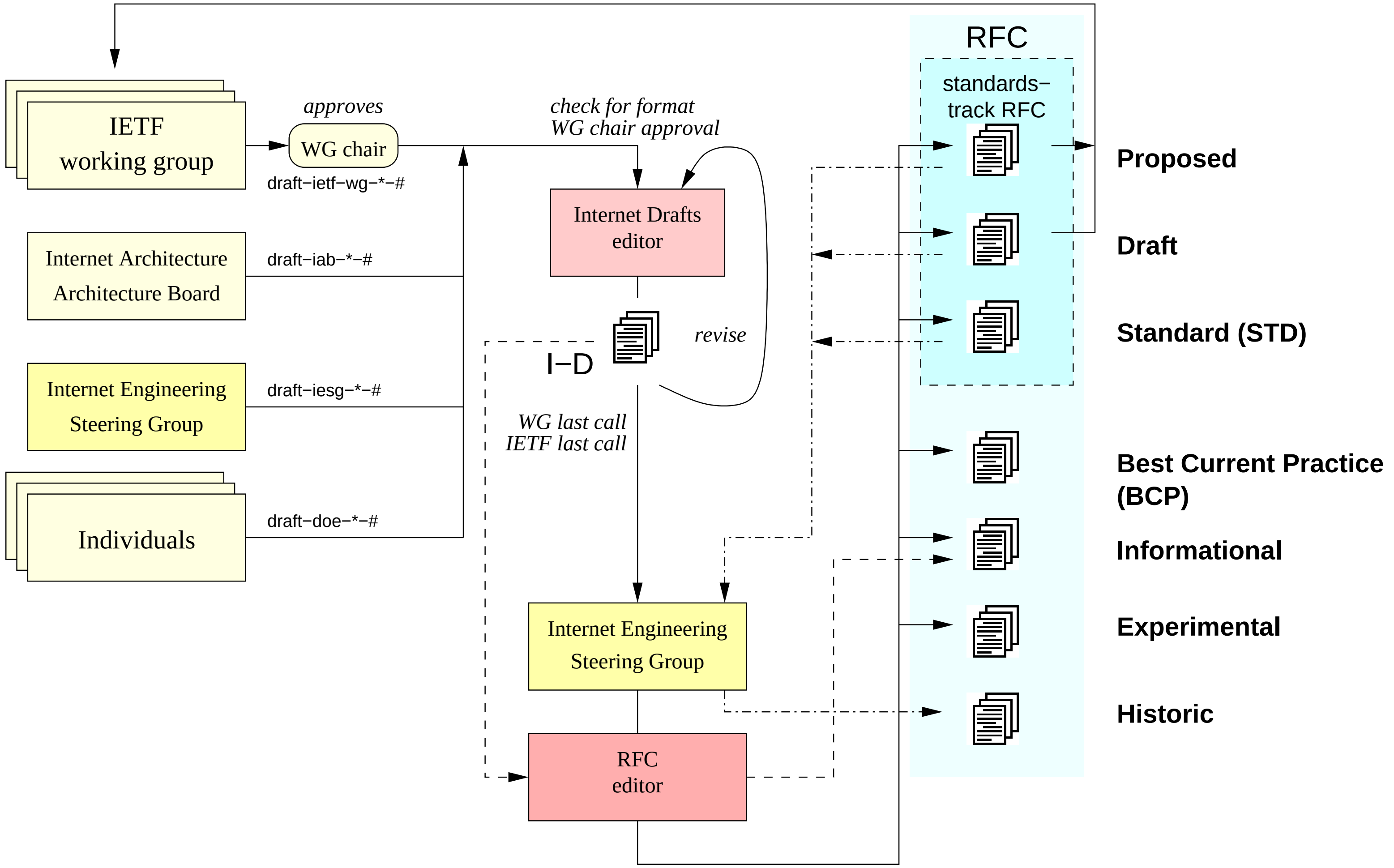
IETF Areas

- applications (28 WGs): calendar, HTTP, LDAP, MIME, NNTP, ...
- general (1): POISSON
- internet (15): IPv6, IP over x, interface MIBs
- operations and management (23): MIBs, routing policy, benchmarking
- routing (17): multicast, RIP, BGP
- security (14): S/MIME, TLS, PGP, XML security
- transport (24): RTP, SIP, RTSP, RSVP admission, TCP
- user services (4): handbooks, guides, standard policies

IETF Working Groups

- headed by chair(s) designated by AD
- should be single, well-defined topic
- discussions on public mailing list
- small groups of authors do detail work
- meet at IETF (three times a year)
- possibly interim meetings
- done ➠ dissolve

IETF standards process



Standardization process

RFC 2026:

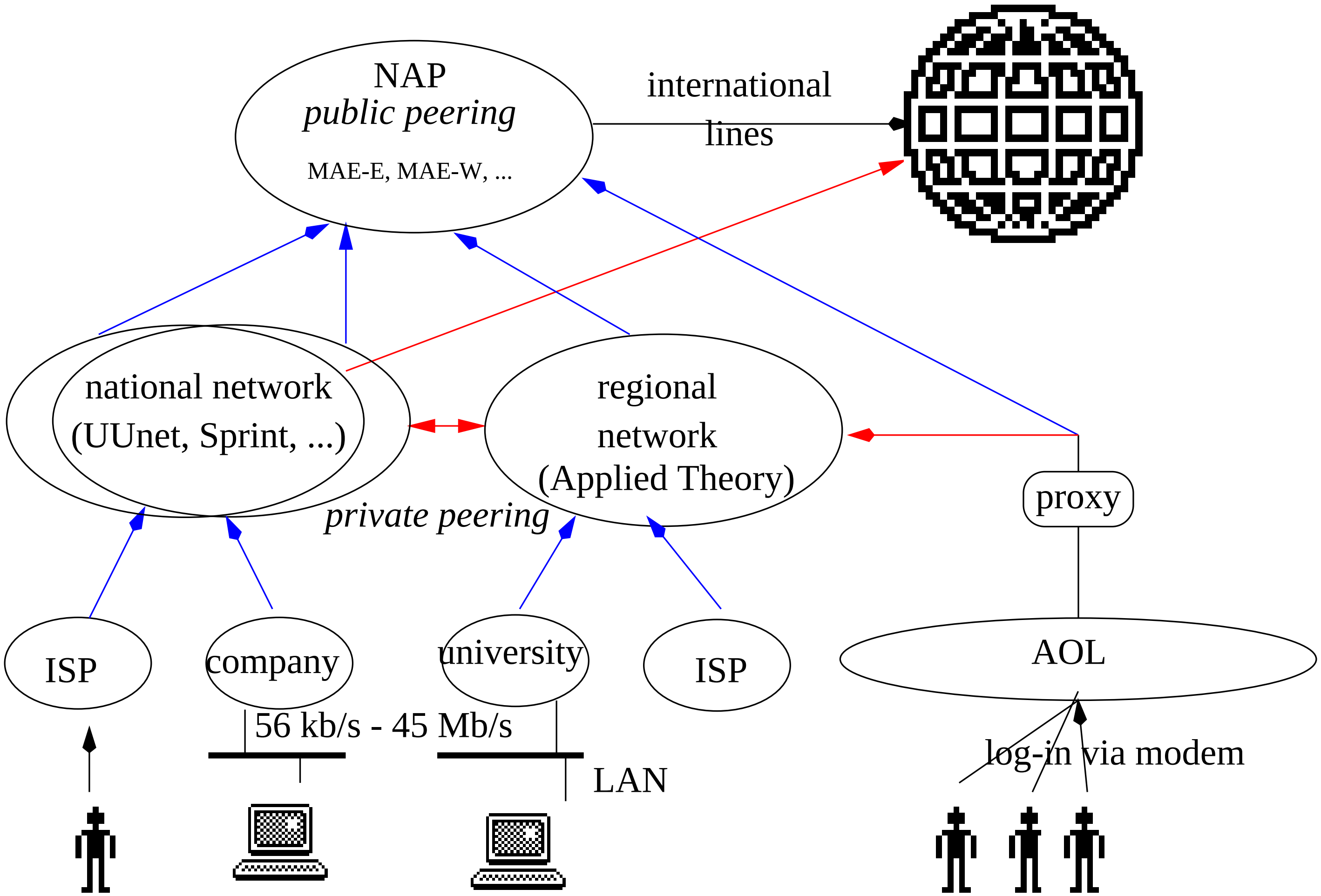
1. new topic ➠ BOF at IETF meeting
2. if response, create working group with charter
3. create Internet drafts = temporary ($\leq$ 6 months) working drafts
4. status and discussion presentations at IETF meetings
5. working group last call
6. IETF last call
7. IESG “votes” (by consensus)
8. published as RFC: *proposed standard*
9. 2 implementations + $\geq$ 6 months ➠ *draft standard*
10. operational experience + 4 months ➠ *Internet standard* (STD)

RFCs

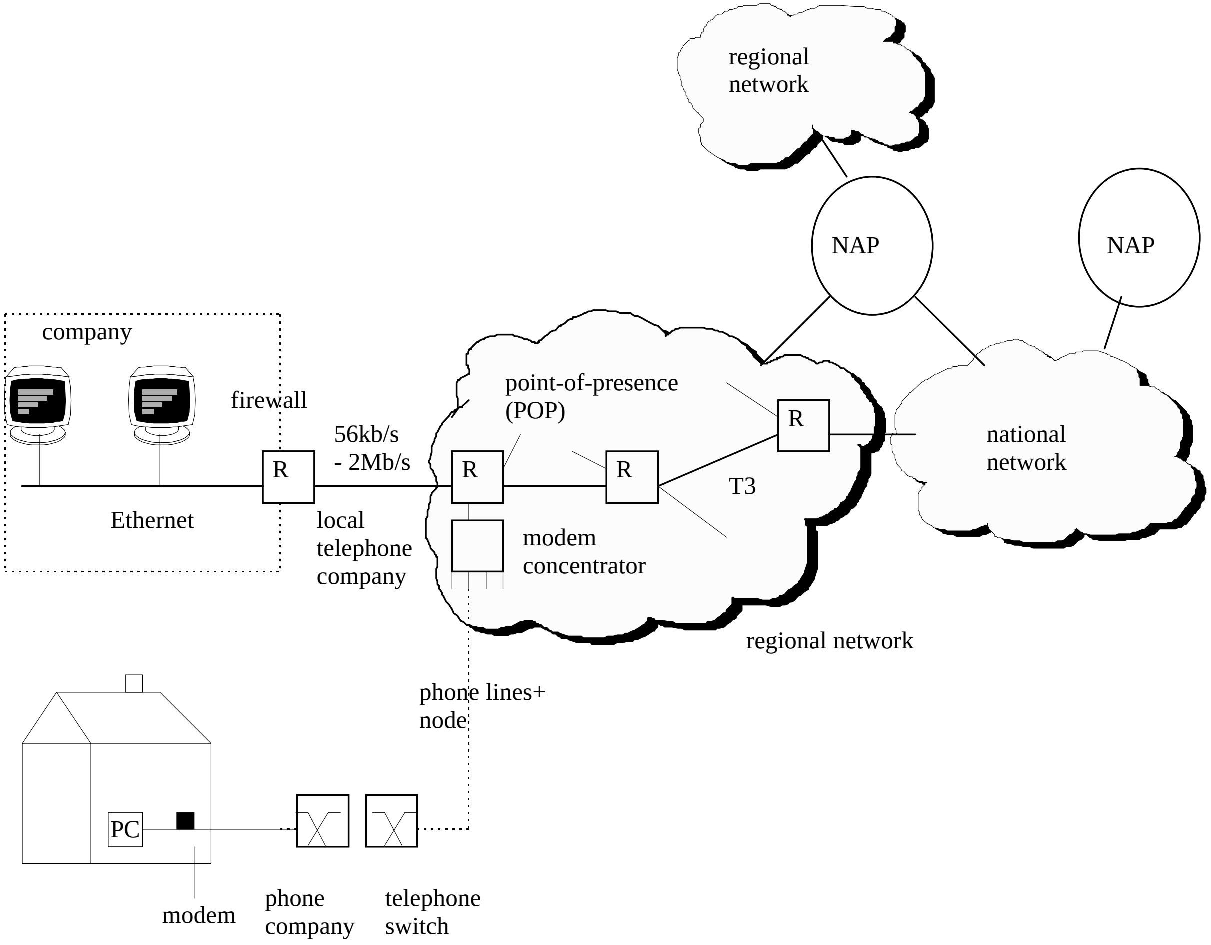
- ASCII + PostScript, no charge (www.normos.org)
- published RFCs never change (no IP-1994)
- also:
  - experimental
  - informational (possibly “FYI”)
  - historic(al)
- anybody can submit RFC, but editor can filter for content, conflict with existing work
- check the April 1 ones. . . (RFC 1149)

Internet Access and Infrastructure

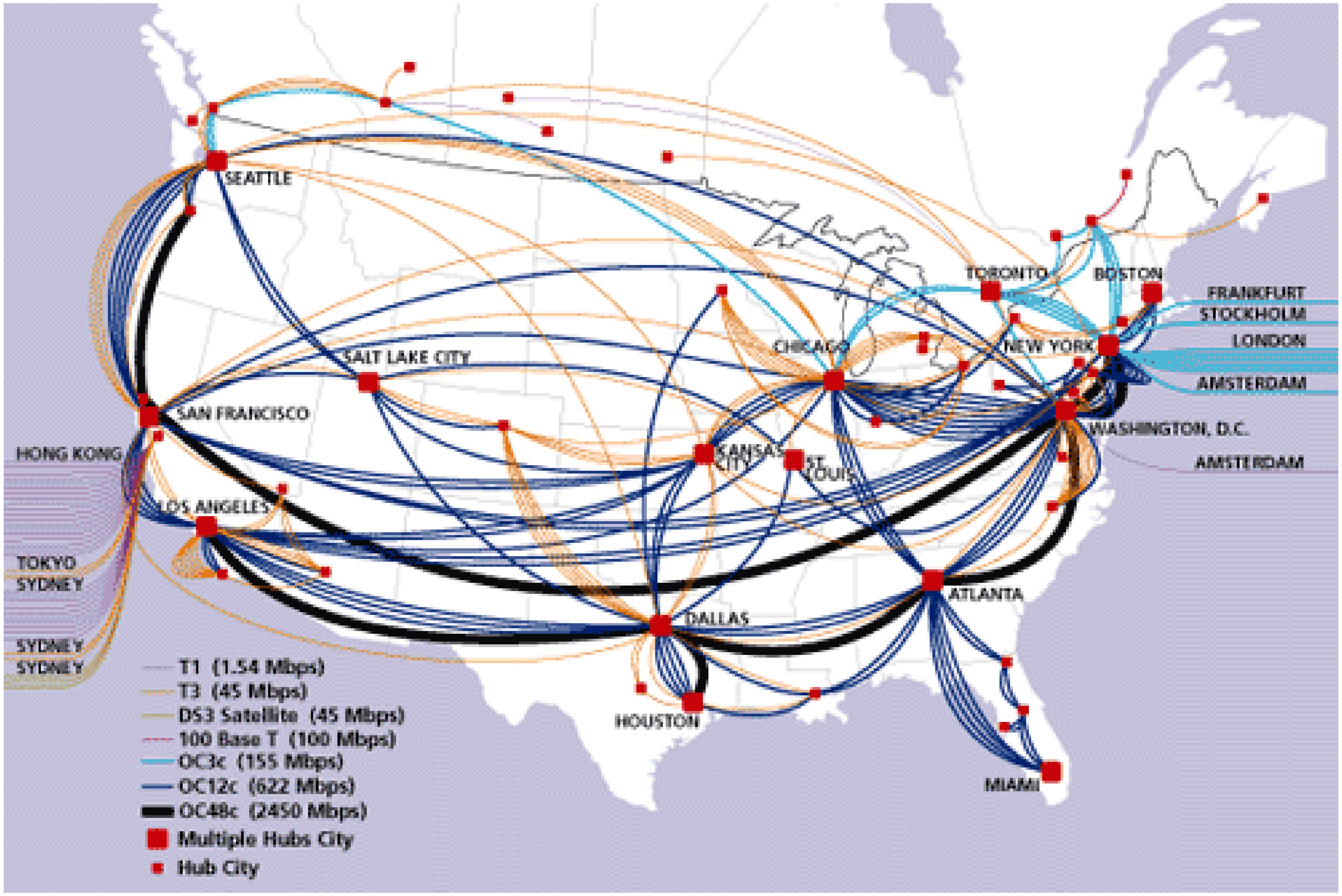
Who pays for the Internet?



Network Access and Interconnection



Example: UUnet Backbone



Large Consumer ISPs

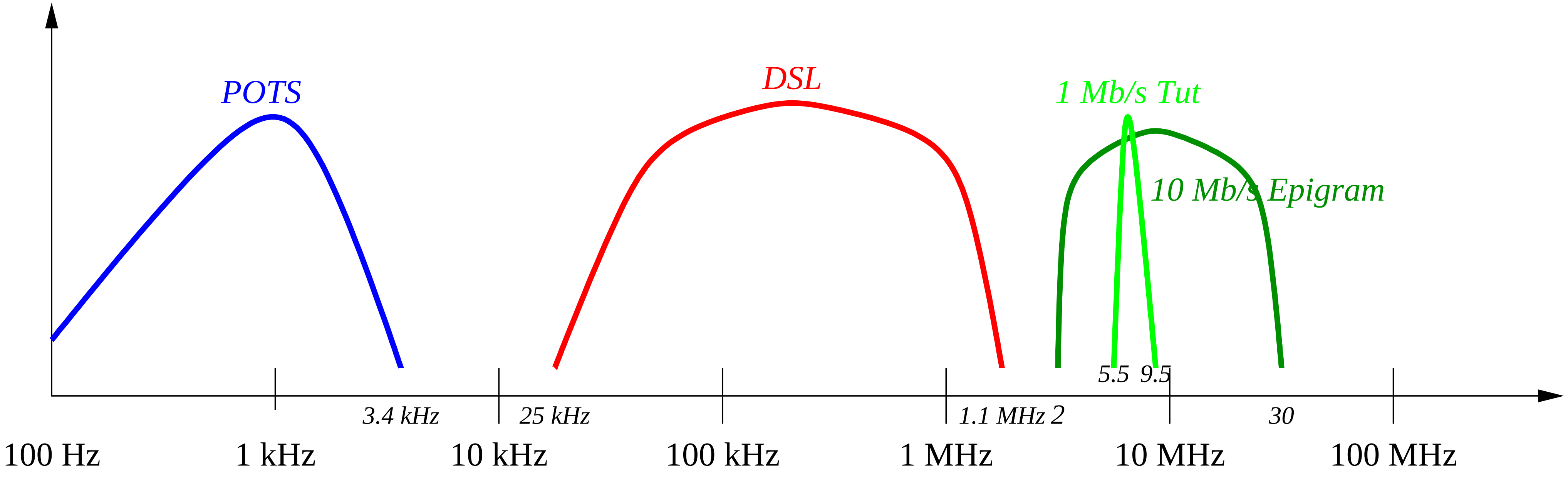
- 201 million Internet users in the world, 112.4 million in U.S. and Canada (1 subscriber = 2.5 users!).
- many lease facilities (e.g., AOL)

company	subscribers (10^6)
AOL	20
EarthLink+Mindspring	3
NetZero	3
Prodigy	2
CompuServe	2
AT&T Worldnet	1.8
Microsoft Network	1.8

Mostly modem; 300,000 DSL and 1.1 million cable modem (end 1999).

Home Networking

phone lines	1-10 Mb/s, higher frequency than DSL
power lines	< 10 Mb/s
wireless	1-11 Mb/s (IEEE 802.11) in 2.4 GHz band



Carriers

About 40 *backbones*, use various *right-of-ways*

UUnet (MCI)		
Sprint		
GlobalCrossing	mostly cross-oceanic	20,000 (U.S.)
AT&T		
Level3	railroad?	11,000 (goal: 16,000)
PSINet	leased	
Qwest	railroad	
Williams	pipelines	25,000
Enron	pipelines, HV	

Peering: Equinix

Internet Access

method	media	downstream	upstream
modem	POTS	$\leq$ 53 kb/s	33.6 kb/s
Intercast	VBI	150 kb/s	modem
ISDN	POTS	128 kb/s	128 kb/s
DSL	POTS	160 kb/s	160 kb/s
ADSL	POTS	0.6…9 Mb/s	16…640 kb/s
cable modem	CATV	10 Mb/s	1 Mb/s
T1	copper	1.5 Mb/s	1.5 Mb/s
T3	fiber, copper	45 Mb/s	45 Mb/s

Network utilization

Averaged over one week:

local phone line	4%
U.S. long distance switched voice	33%
Internet backbones	10-15%
private line networks	3-5%
LANs	1%

- peak personal-use hours: 5-11 pm
- “world wide wait”: web servers? DNS? NAPs? access?
- average speed: 40 kb/s

ISP Service

- average connect time: 310.3 min/month home, 417.4 min/month work ➠ \$3.85 hour
- 66 MB average transfer/month ➠ 33 c/MB
- 10:1 modem concentration ratio
- T1: 500 GB/month each direction ➠ 0.3c/MB (\$1500/month)
- but ISP T1 utilization ≈ 40-45%
- ISP costs: \$2.50/month for phone line, \$2/month for equipment depreciation, \$0.20/month for network
- fiber: \$30,000-\$50,000/mile

ADSL Limits

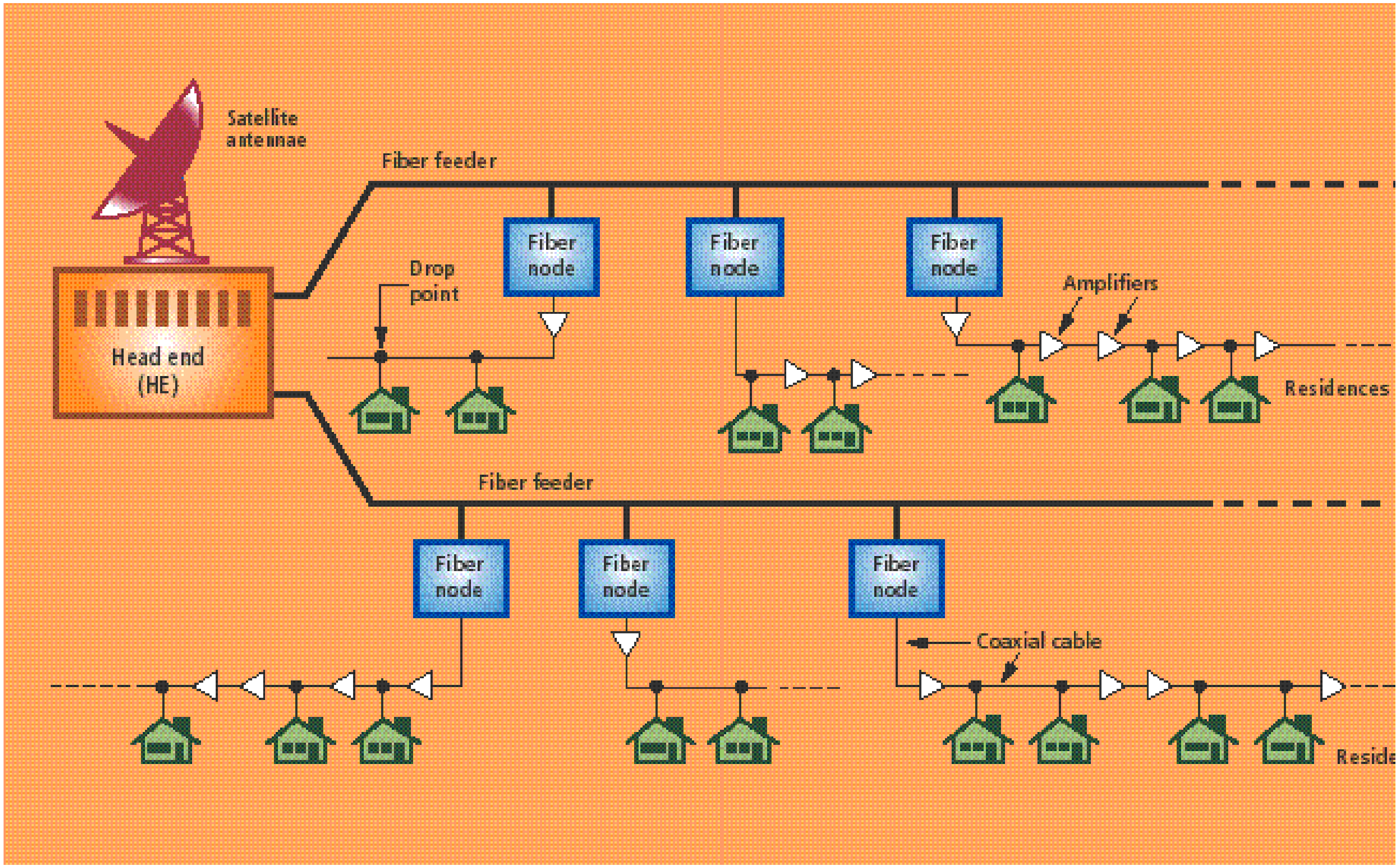
name	Mb/s	distance (ft)	km	
DS1 (T1)	1.544	18,000	4.5	<80%
E1	2.048	16,000	4.1	
DS2	6.312	12,000	3.0	
E2	8.448	9,000	2.3	
1/4 STS-1	12.960	4,500	1.1	
1/2 STS-1	25.920	3,000	0.8	
STS-1	51.840	1,000	0.3	
OC-3	155.000	100	0.03	

ADSL Pricing Example

Bell Atlantic, May 1999:

downstream	upstream	rate	ISP
640 kb/s	90 kb/s	$ 39.95	$10
1.6 Mb/s	90 kb/s	$ 59.95	$40
7.1 Mb/s	680 kb/s	$109.95	$80

Cable plant architecture



A. Dutta-Roy, “Cable – it’s not just for TV”, *IEEE Spectrum*, May 1999; © 1999 IEEE

Cable plant architecture

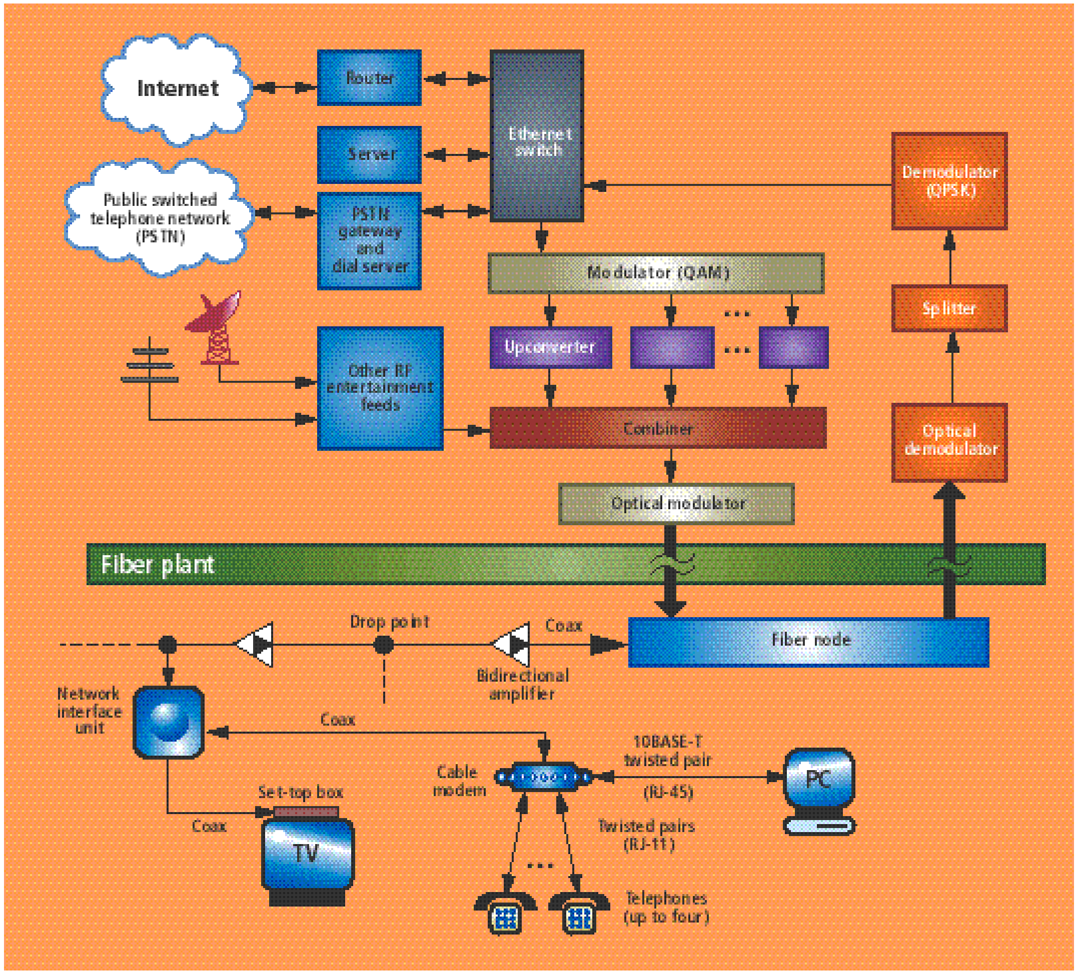
- coax cable: < 1 GHz bandwidth, typically 500 MHz
- 35 TV channels in typical older CATV systems
- 500–2000 homes for single headend or fiber node
- head-end to residence < 80 km
- fiber node to residence < 350 m
- US: 67% of households have, 95 mio. residence “passed”

Cable modems

- always-on, but maybe temporary IP addresses
- hybrid fiber coax
- CMTS (cable modem termination system) = “headend”
- Ethernet interface to user’s PC
- but: conversion to bidirectional amplifiers, power
- DHCP + network address translation (NAT)
- conversion cost: $200–$800
- standards:
  - Data-Over-Cable Service Interface Specification (DOCSIS)
  - IEEE 802.14: ATM MAC

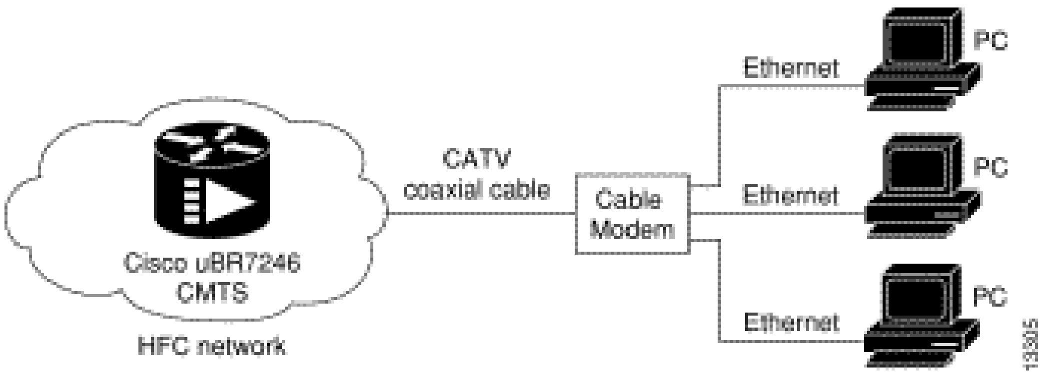
- Multimedia Cable Network System Partners (MCNS): contention
- Davic (Europe)

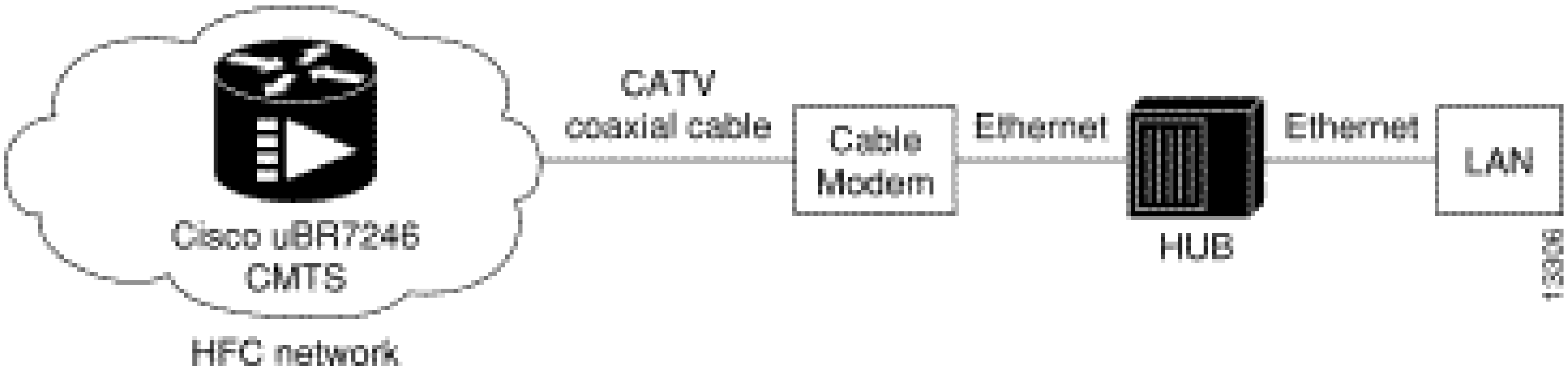
Cable plant architecture



A. Dutta-Roy, "Cable – it's not just for TV", *IEEE Spectrum*, May 1999; © 1999 IEEE

Cable modem network modes





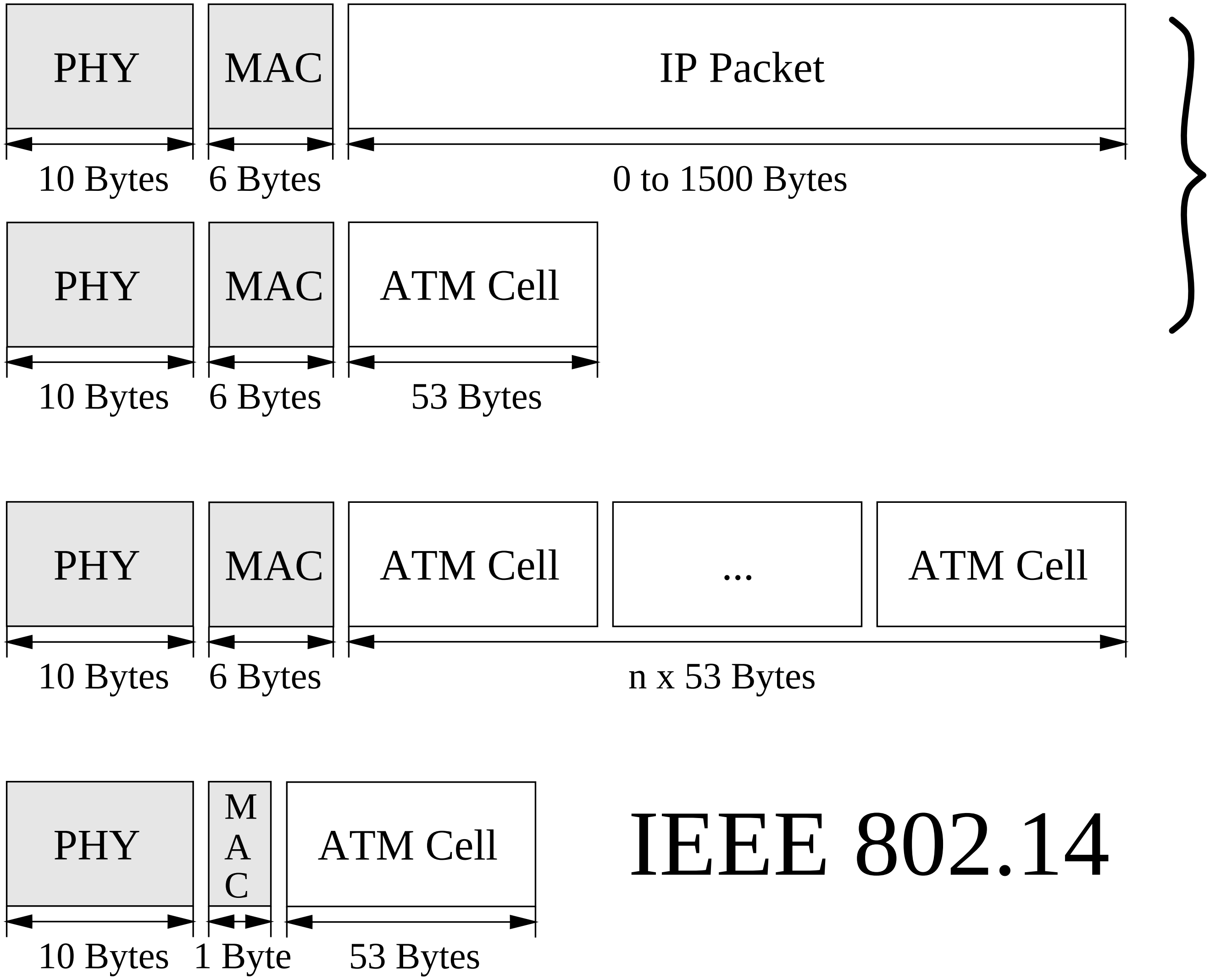
Cable modem: downstream

- one or more 6 MHz channels in 54–550 MHz range
- typical bit consumption (no A/V): 40 kb/s, 4 kb/s upstream
- 30-50% active ➠ 420 customers per channel
- 64 QAM (6 bits/symbol) ➠ $\leq$ 30 Mb/s
- newer equipment: 256 QAM ➠ 40 Mb/s

Cable modem: upstream

- 5–42 MHz (usually band $<$ 3 MHz, typically 200 kHz)
- noise aggregation ➠ QPSK with 2 bits/symbol (5 Mb/s)
- actual througput: 768 kb/s
- can't use Ethernet-style CDMA.
- TDMA variation: headend asks for potential senders
- headend returns *grant*: 2^k 6.25 μs mini slots
- send 6-byte request to transmit ➠ delay variation!
- encryption: 40/56 bit DES

Cable modems: IEEE 802.14 vs. MCNS



PHY | MAC | IP Packet

10 Bytes | 6 Bytes | 0 to 1500 Bytes

PHY | MAC | ATM Cell

10 Bytes | 6 Bytes | 53 Bytes

MCNS

PHY | MAC | ATM Cell | ... | ATM Cell

10 Bytes | 6 Bytes | n x 53 Bytes

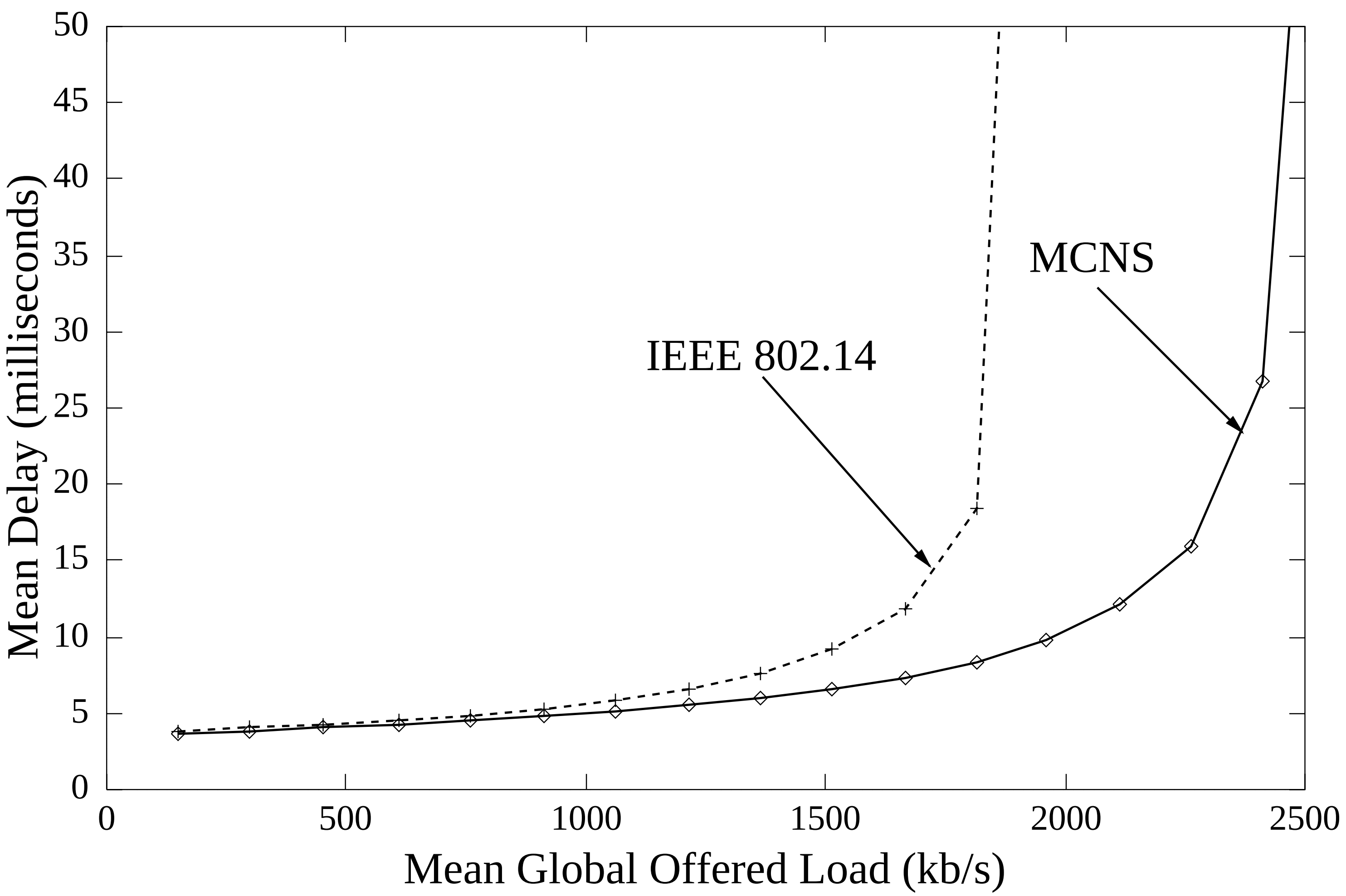
MCNS Concatenation

PHY | MAC | ATM Cell

10 Bytes | 1 Byte | 53 Bytes

IEEE 802.14

Cable modems: access delay



N. Golmie, F. Mouveaux, D. Su, “A comparison of MAC protocols for hybrid fiber/coax networks: IEEE 802.14 vs. MCNS”, ICC, June 99.

Performance comparison

Keynote web retrieval performance (April 1999):

Technology	5-11pm	8am-5pm
DSL (128/384)	3.55	4.30
Cable modem	3.97	3.68
T-1	1.83	2.36

- both shared *somewhere*

RTT delay:

	average	max.
ISDN	≈ 10 ms	
CM	20-45 ms	860
ADSL	≈ 20–30 ms (50 ms for 1.5/224)	63

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.

Internet WAN Physical Layers

	Gb/s	remarks
Giga Ethernet	1.25	fiber
T-3	0.045	fiber, TP or coax
OC-3c	0.155	fiber
OC-12	0.622	fiber
OC-48	2.4	fiber
OC-192	10	fiber

Dense Wavelength Division Multiplexing

- multiple optical λ in single fiber
- 1.6 to 2 Tb/s per fiber
- interfaces typically 622 Mb/s to 10 Gb/s

Link-Layer Mechanisms Used

Roughly in order of popularity:

- ATM
- IP over SONET (synchronous optical network)
- frame relay
- gigabit Ethernet (with range extenders)
- T1, T3

Asynchronous Transfer Mode (ATM)

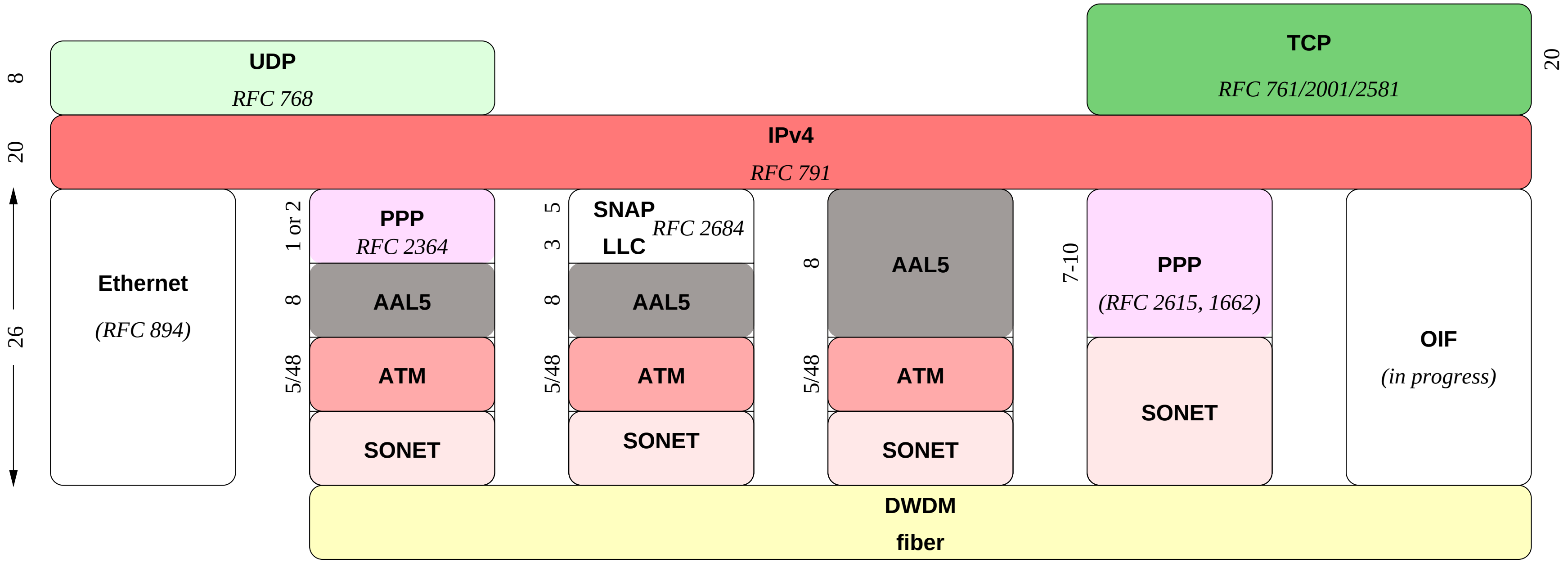
- 48-byte cells plus 5-byte header
- routing by label swapping
- virtual circuits (VCs) and paths (VPs)
- in-order delivery, but cells can be lost
- *adaptation layers*:

AAL1	continuous bit rate (CBR); “circuit emulation”
AAL2	multiplexed low-delay voice
AAL3/4	data (rarely used)
AAL5	IP packet in several cells

Frame Relay

- variable-length packets
- permanent or switched virtual circuits (PVC, SVC)
- typically, lower bandwidth ($\leq$ 45 Mb/s)
- popular as access mechanism, corporate networks

Internet Link Layers



Wireless Access

- Industrial, Scientific, Medical (ISM) bands (unlicensed): 902–928 MHz (US only), 2.4 GHz, 5.8 GHz
- analog cellular: 800 MHz
- PCS: 1.9 GHz

Wireless Ethernet:

- 900 MHz, **2.4 GHz**, or 5 GHz
- 1 or 2 Mb/s, soon 5.5 Mb/s or 11 Mb/s
- collision-based, with reservation (RTS/CTS)
- IEEE 802.11 = FH or DS

Cellular Digital Packet Data (CDPD):

- pauses in AMPS voice traffic

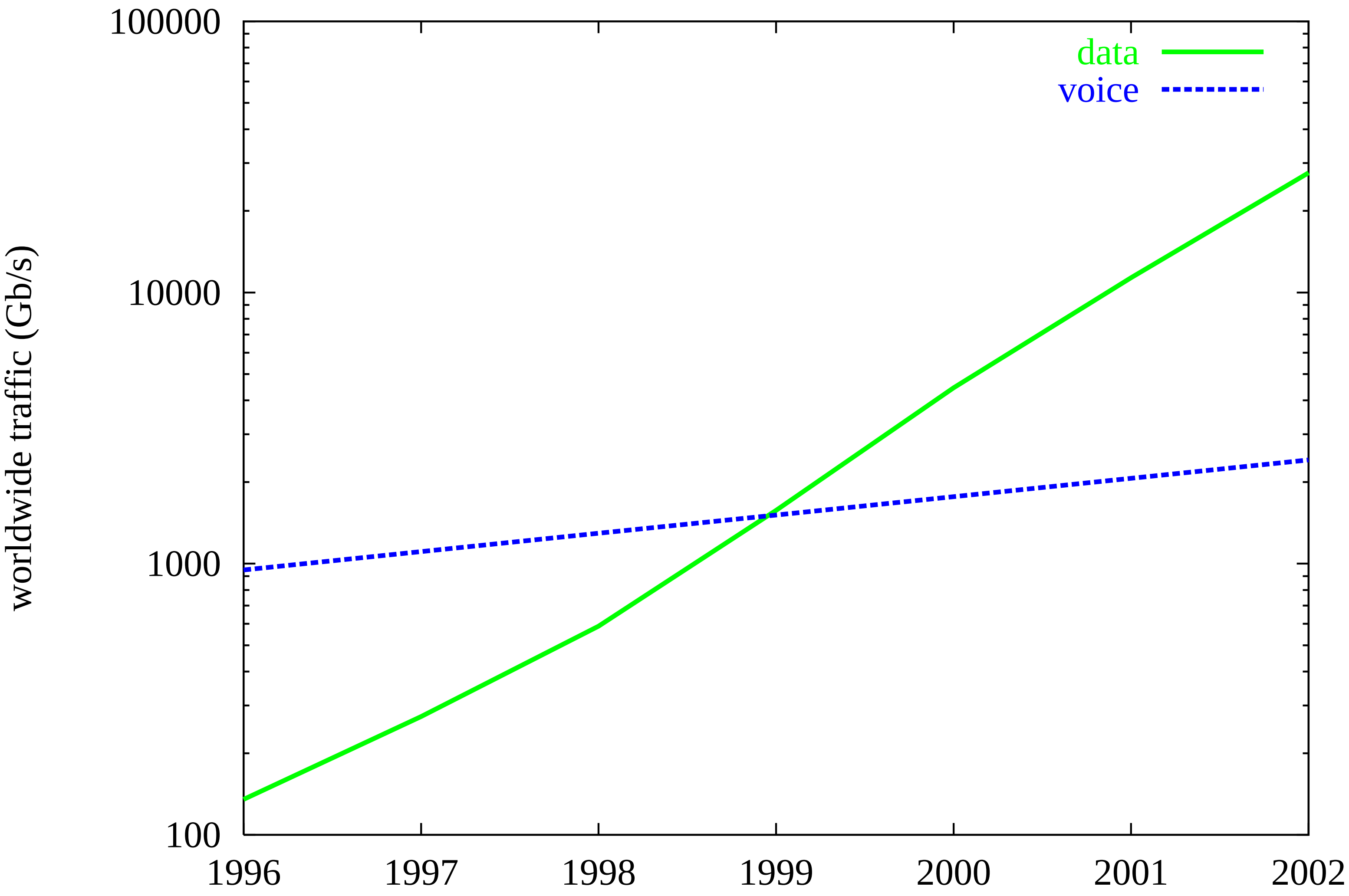
Wireless access

Technology	band	mod.	rate	open range (m)
RAM			8.0 k/bs	
GSM data	1.9 GHz	TDMA	9.6 kb/s	
CDPD			19.2 kb/s	km
Metricom Ricochet	902-928 MHz	FH	28.8 kb/s	300-450
Bluetooth	2.4 GHz	FH	432 kb/s	10
802.11	2.4 GHz	DS	1 Mb/s	540
			2 Mb/s	400
			4 Mb/s	195
			5.5 Mb/s	120

Internet Traffic

- 5,000-8,000 TB/month or 15.4–24.7 Gb/s
- long-distance calls: 525 GDEM or 64 Gb/s
- all the world's telephones: 600 Gb/s
- almost all (90%?) of the traffic is TCP

Voice vs. Data Traffic



data
voice
worldwide traffic (Gb/s)
100000
10000
1000
100
1996
1997
1998
1999
2000
2001
2002

Voice vs. Data Traffic

- local vs. LANs vs. private networks
- capacity vs. traffic
- hop length of data traffic $<$ voice
- link utilization (higher for voice)
- revenue

Protocol Contributions

proto	src	dest	pkts	bytes
TCP	http		35%	66.4%
TCP		http	33%	7%
TCP		nntp	1.8%	3.8%
TCP	ftp		1.4%	3.2%
TCP		smtp	1.8%	1.9%
TCP	nntp		1.3%	1.5%
UDP	dns	dns	3.1%	1.0%

April 1997, NLANR

Internet Names and Addresses

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

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

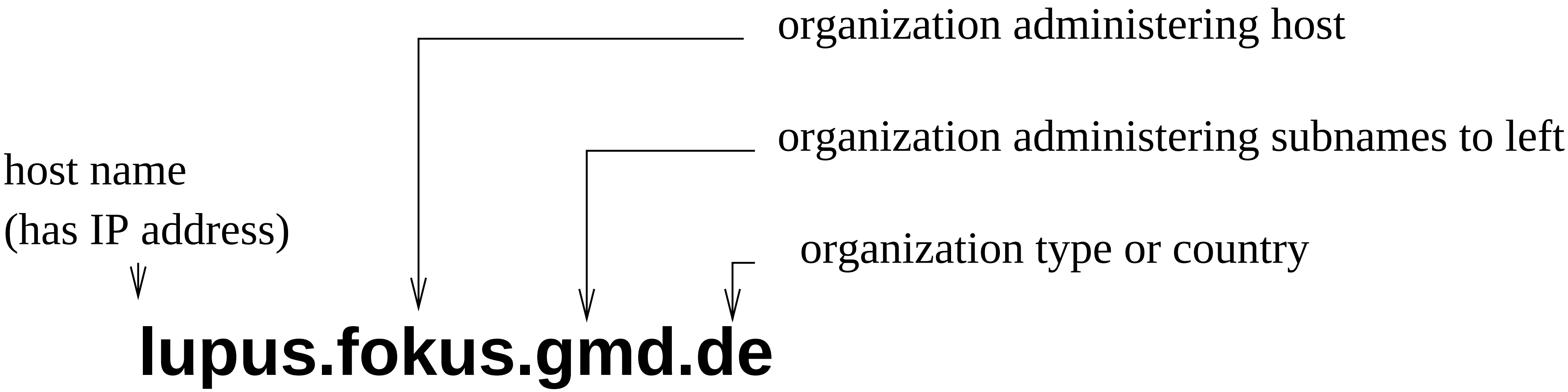
host name $\overset{\mathrm{DNS,many-to-many}}{\rightarrow}$ IP address $\overset{\mathrm{ARP,1-to-1}}{\rightarrow}$ MAC address

Mappings in the Internet

whois	domain name	owner description
LDAP	key (name)	address, other info
YP	name	data item
DNS	host name	IP addresses
	IP address	host name
atmarp	IP address	ATM NSAP
ARP	IP address	Ethernet address
RARP	MAC address	IP address

The Internet Domain Name System

We’ll talk about *name resolution* later…



organization administering host

organization administering subnames to left

host name
(has IP address)

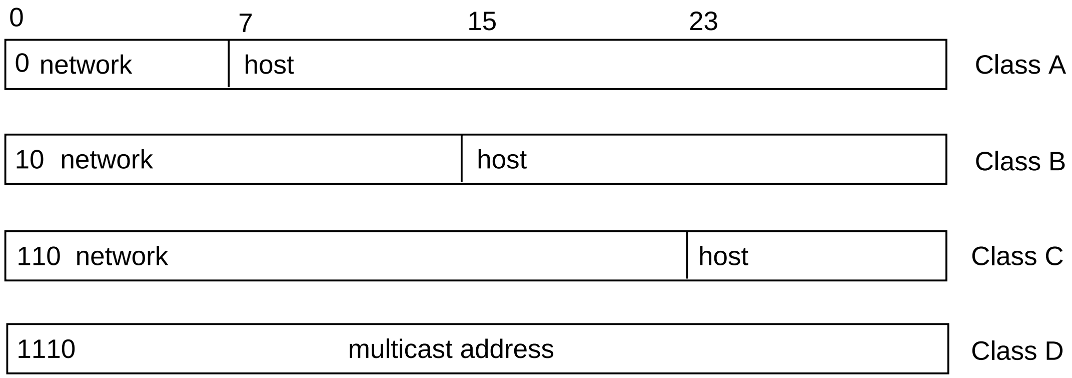
organization type or country

lupus.fokus.gmd.de

Anywhere from two to ∞ parts

Internet (IP) Addresses

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



0 7 15 23

0 network | host — Class A

10 network | host — Class B

110 network | host — Class C

1110 multicast address — Class D

- originally, two-level hierarch → n-level, changing
- an IP address identifies an *interface*, not a host!
- a host may have two or more addresses. Why?

Internet addresses

- (almost) every *interface* has one
- but may
  - change (dial-in)
  - have lots (WWW servers)
  - have none (some routers)
  - not be globally unique
- old: class-{A,B,C} ➠ 2-level addressing: network,host
- new: classless interdomain routing (CIDR) ➠ aggregation, route on prefix and mask

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

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

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
```

IP address exhaustion

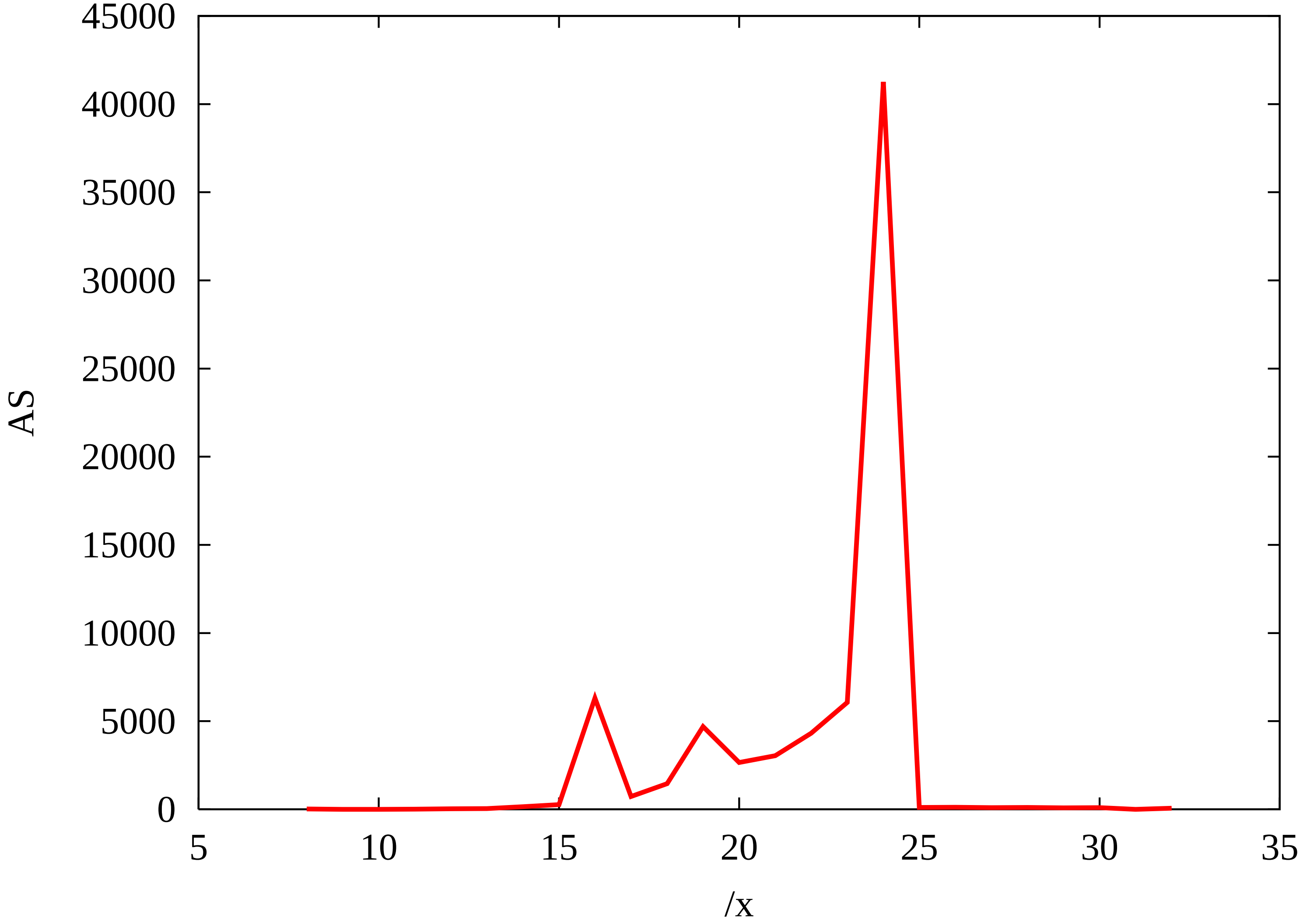
As of February 2000,

- 61.1% of available address space allocated
- 49.4% of allocated address space announced
- 30.2% of available address space announced

Routing table:

- 71,717 “autonomous system” (AS) entries
- 41,256 of which are /24

Routing Table Entries



45000
40000
35000
30000
25000
20000
15000
10000
5000
0
AS
5
10
15
20
25
30
35
/x

Network Address (and Port) Translation (NA(P)T)

- most corporations use private address space, also residential
- 10/8, 172.16/12, 192.168/16
- NAT translates internal ⟷ external as needed
- works for outgoing TCP connections: POP, HTTP, SMTP, Telnet
- need application layer gateway (ALG) for out-of-band protocols (ftp, SIP, RTSP, H.323, . . .)
- problems:
  - controlled connections (ftp, Internet telephony, media-on-demand)
  - UDP services (streaming media)
  - security – rewriting breaks IPsec
- suggestion: Realm-Specific IP (RSIP) makes host aware of mapping

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

Multihoming

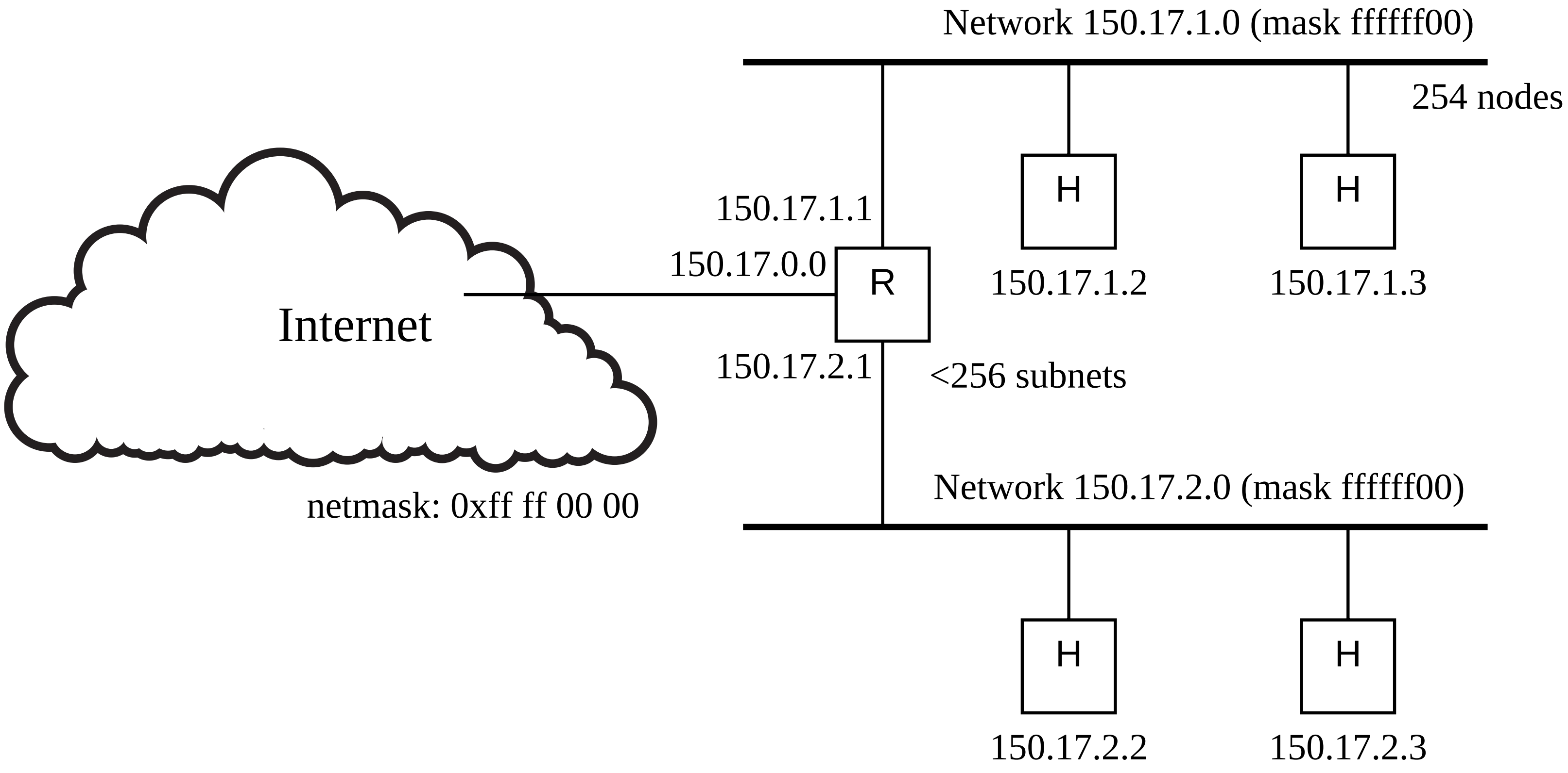
- = one “stub” network, multiple providers
- options:
  1. global prefix ➠ aggregation ↓
  2. divide network ➠ no redundancy
  3. multiple addressess ➠ applications need to try several, address space use ↑

Mobility and Renumbering

- renumber if immediate or up-stream provider changes
- mobility: change network attachment point
- mobility = renumbering: network “location” changes
- IP address as location ➠ keep address, break aggregation
- renumbering is hard: configuration files, transition
- IP address as identifier ➠ break connections

Subnetting

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

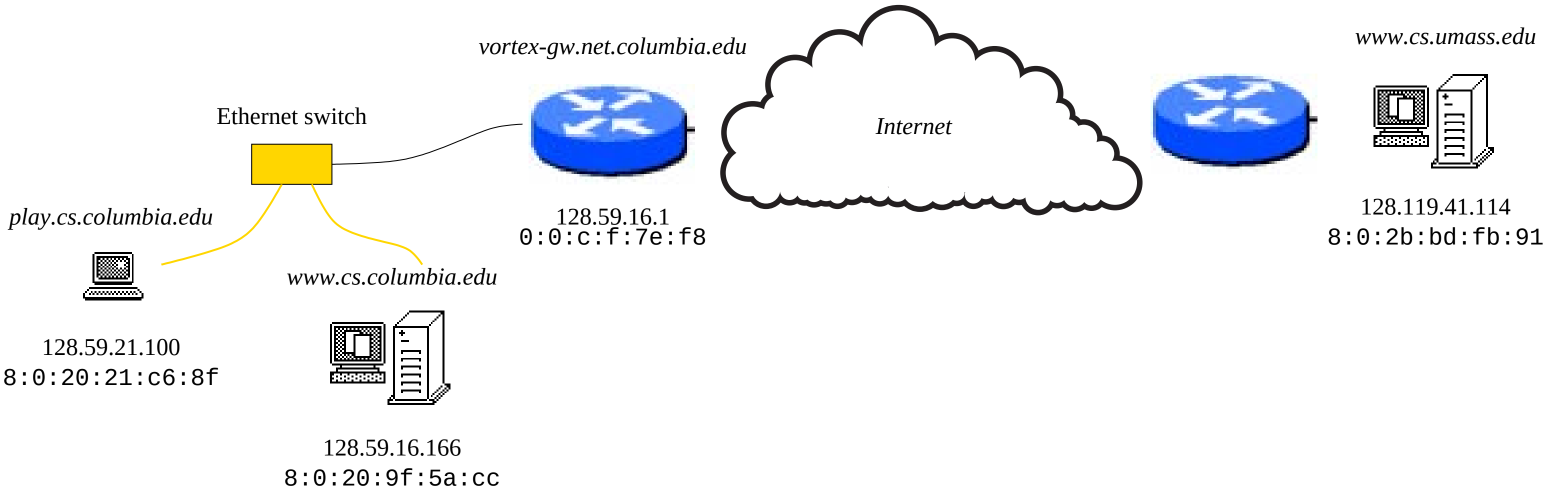


How does a packet get to the server?

E.g., web page from `http://www.cs.umass.edu`:

- get host name `www.columbia.edu` from URL;
- DNS: translate to IP address 128.59.35.60
- is it on local network? no ➠ find local router
- local router sends to Internet
- Internet routes to Columbia network router (128.59.?.?)
- Columbia router routes to web server

Peeking inside a packet



Ethernet	8:0:20:21:c6:8f 8:0:20:9f:5a:cc	8:0:20:21:c6:8f 0:0:c:f:7e:f8
IP (network)	128.59.21.100 128.59.16.166	128.59.21.100 128.119.41.114
TCP (transport)	? 80	? 80
http (application)	http://www.cs.columbia.edu	http://www.cs.umass.edu

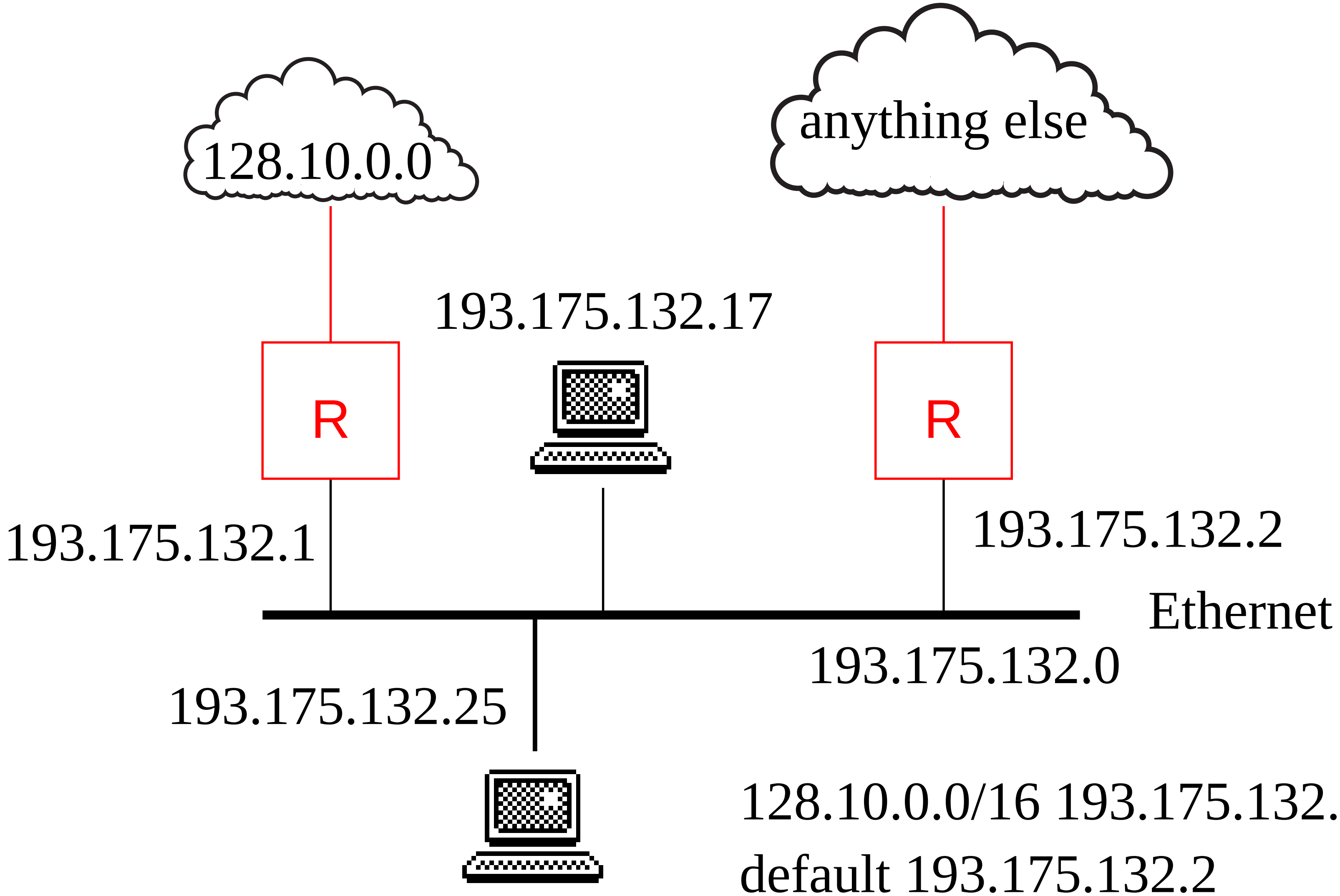
source destination

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



128.10.0.0

anything else

R

R

193.175.132.17

193.175.132.1

193.175.132.2

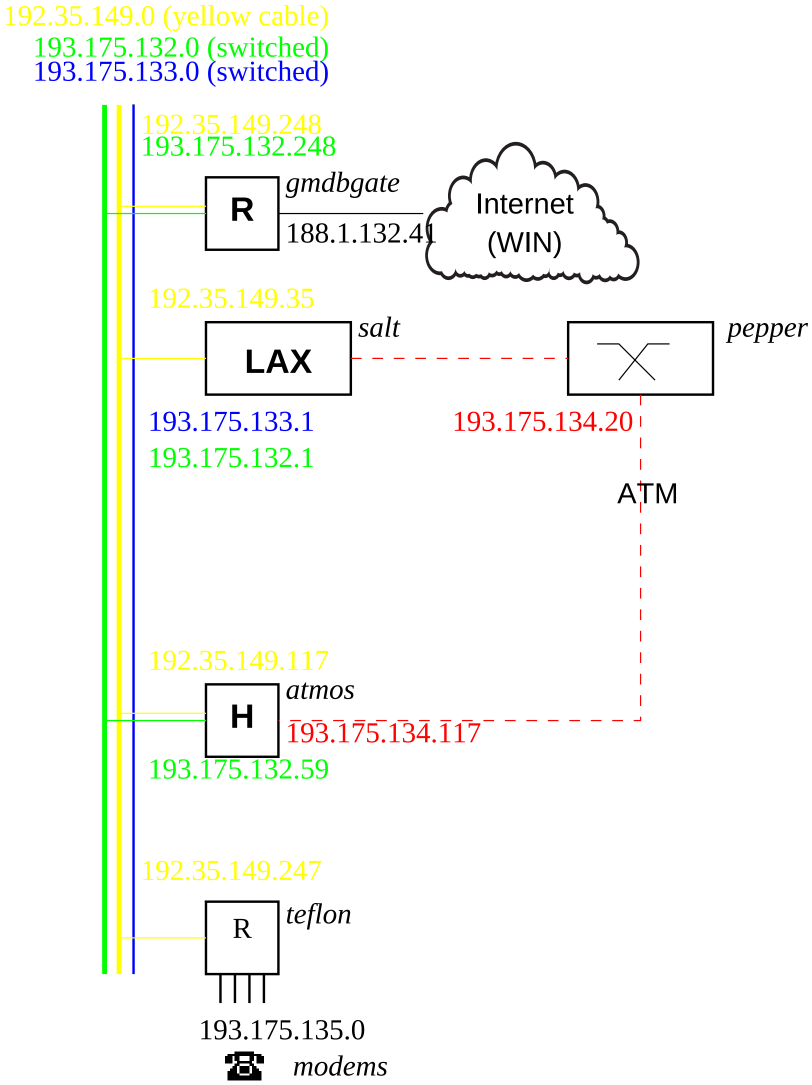
Ethernet

193.175.132.0

193.175.132.25

128.10.0.0/16 193.175.132.1
default 193.175.132.2

GMD Fokus Network


192.35.149.0 (yellow cable)
193.175.132.0 (switched)
193.175.133.0 (switched)
192.35.149.248
193.175.132.248
R
gmdbgate
188.1.132.41
Internet
(WIN)
192.35.149.35
LAX
salt
pepper
193.175.133.1
193.175.132.1
193.175.134.20
ATM
192.35.149.117
H
atmos
193.175.134.117
193.175.132.59
192.35.149.247
R
teflon
193.175.135.0
modems

Network Layer: IPv4 and IPv6

- unreliable datagram ➠ misorder, loose, duplicate
- 32-bit (IPv6: 128 bit) globally unique addresses
- no checksum on payload
- allow *fragmentation* of large packets into MTU-sized frames
- 20 (IPv6: 40) byte header
- IP multicast: receiver group with anonymous membership

IPv4

IPv4 Service Model

datagram: each packet is independent of all others

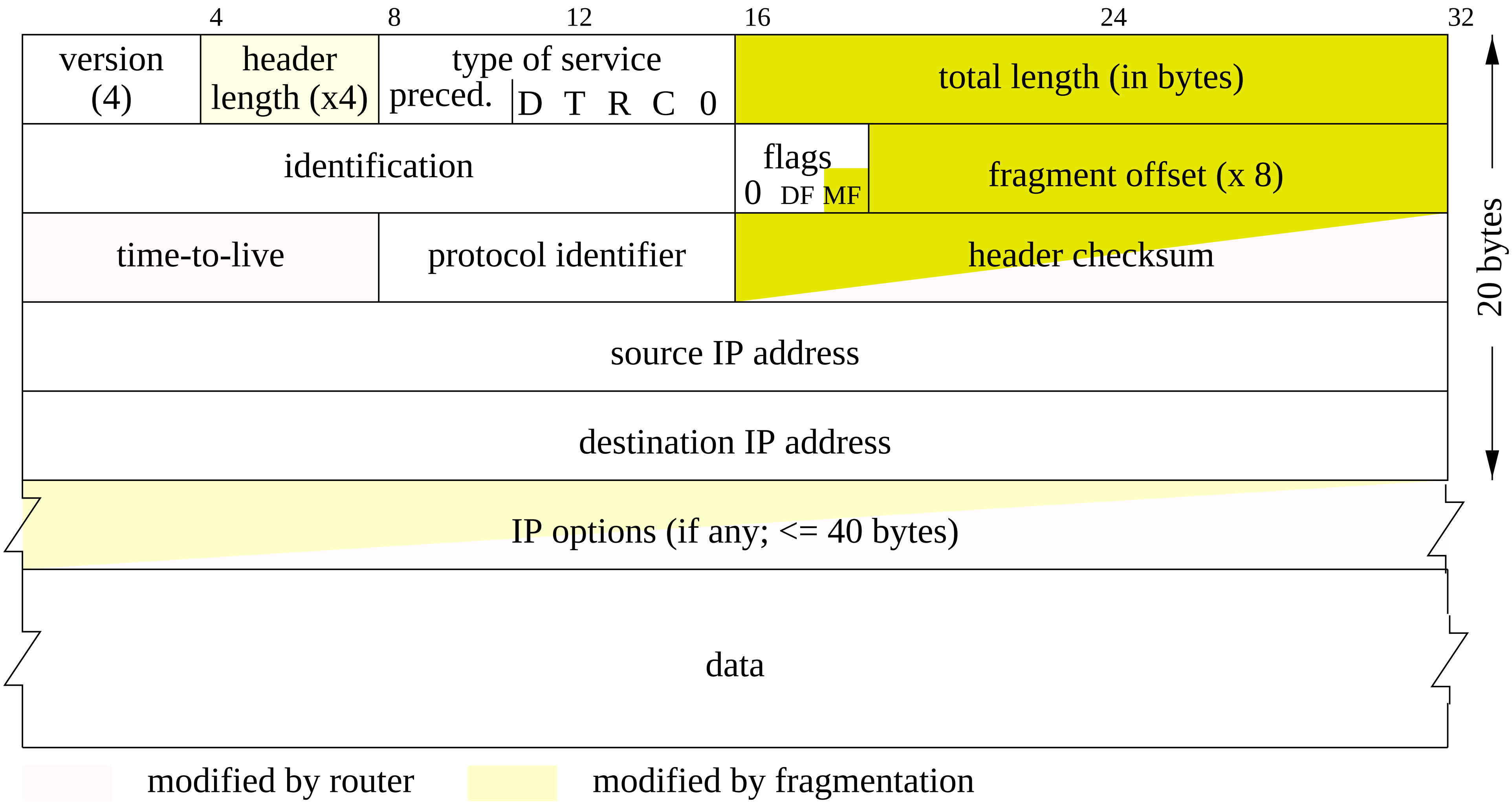
best effort: packet may arrive *or not* after some time

IPv4

- independent packets
- unreliable
- might be reordered (rare), delayed, duplicated, . . .
- but: minimal service on top of *anything* (see RFC 1149)
- only *header* checksum

IPv4 Header

RFC 791


4
8
12
16
24
32
version (4)
header length (x4)
type of service
preced.
D T R C 0
total length (in bytes)
identification
flags
0 DF MF
fragment offset (x 8)
time-to-live
protocol identifier
header checksum
source IP address
destination IP address
IP options (if any; <= 40 bytes)
data
20 bytes
modified by router
modified by fragmentation

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

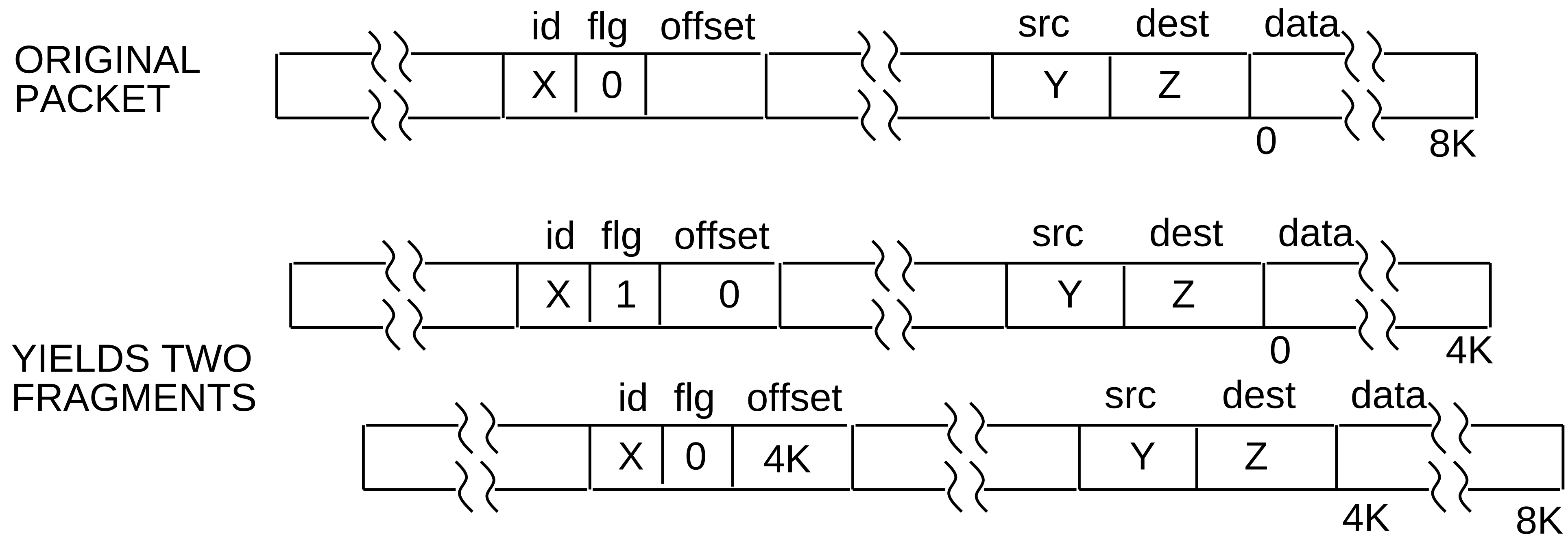
TTL: time to live field; initialized to 64; decremented at each router ➠ drop if TTL = 0 (prevent loops!)

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



ORIGINAL PACKET

YIELDS TWO FRAGMENTS

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 $\downarrow$

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

20−byte standard IP header	8 bit ICMP type	8 bit ICMP code	16−bit checksum	contents of ICMP msg

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

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
ˆC
----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

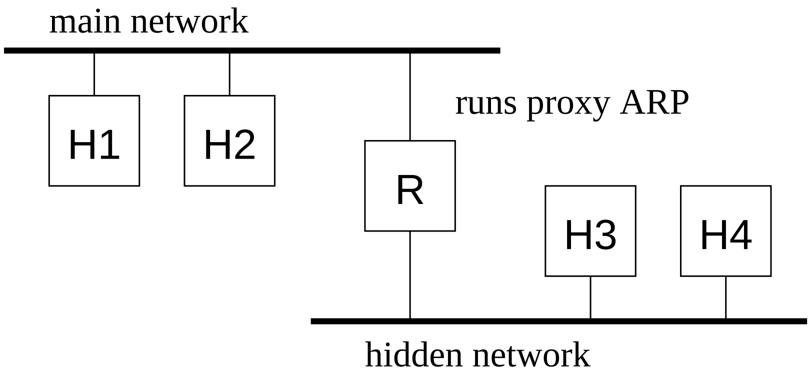
ARP example

```
arp -a
Net to Media Table
Device IP Address             Mask Flags   Phys Addr
------ ----------- --------------- ----- ---------------
le0    hamlet      255.255.255.255       08:00:09:70:7d:16
le0    gaia        255.255.255.255       08:00:20:20:07:03
le0    pern        255.255.255.255       08:00:20:20:75:3c
le0    kite        255.255.255.255       08:00:09:92:0d:d1
le0    condor      255.255.255.255       08:00:20:1c:95:ed
```

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

Transport Layer: UDP and TCP

- UDP service = IP service + checksum + *ports*
- TCP service = UDP service + flow control + congestion control + sequenced, reliable byte stream
- $\not\mathrm{T}$CP for multimedia:
  - loss recovery delay (RTT + ϵ)
  - windowed flow/congestion control ➠ variable bandwidth
  - no multicast

Internet Domain Names

The Internet Domain Name System (DNS)

- hierarchical, dot-separated names
- ➠ multi-level delegation
- by country and by type of organization
- needs to be overhauled (59% of all domains = .com!)

Global top-level domains (gTLDs):

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

domain	usage	example	hosts (7/97)
com	business (global)	research.att.com	4501039
edu	U.S. 4 yr colleges	cs.columbia.edu	2942714
net	network provider	nis.nsf.net	2164815
mil	U.S. military	arpa.mil	542295
gov	U.S. non-military gov’t	whitehouse.gov	418576
org	non-profit orgs (global)	www.ietf.org	327148
us	U.S. geographical	ietf.cnri.reston.va.us	825048
uk	United Kingdom	cs.ucl.ac.uk	878215
de	Germany	fokus.gmd.de	875631

Example

```
server 128.9.0.107
Default Server:  b.root-servers.net
Address:  128.9.0.107

> erlang.cs.columbia.edu
Server:  b.root-servers.net
Address:  128.9.0.107

Name:    erlang.cs.columbia.edu
Served by:
- CUNIXD.CC.COLUMBIA.edu
          128.59.35.142
          COLUMBIA.edu
- DNS2.ITD.UMICH.edu
          141.211.125.17
          COLUMBIA.edu
```

New gTLDs

Proposed for 1998 by CORE, but fate uncertain:

.firm	for businesses, or firms
.shop	for businesses offering goods to purchase
.web	WWW activities
.arts	cultural and entertainment
.rec	recreation/entertainment
.info	information services
.nom	individual or personal nomenclature

DNS Issues

NTIA white paper, June 1998:

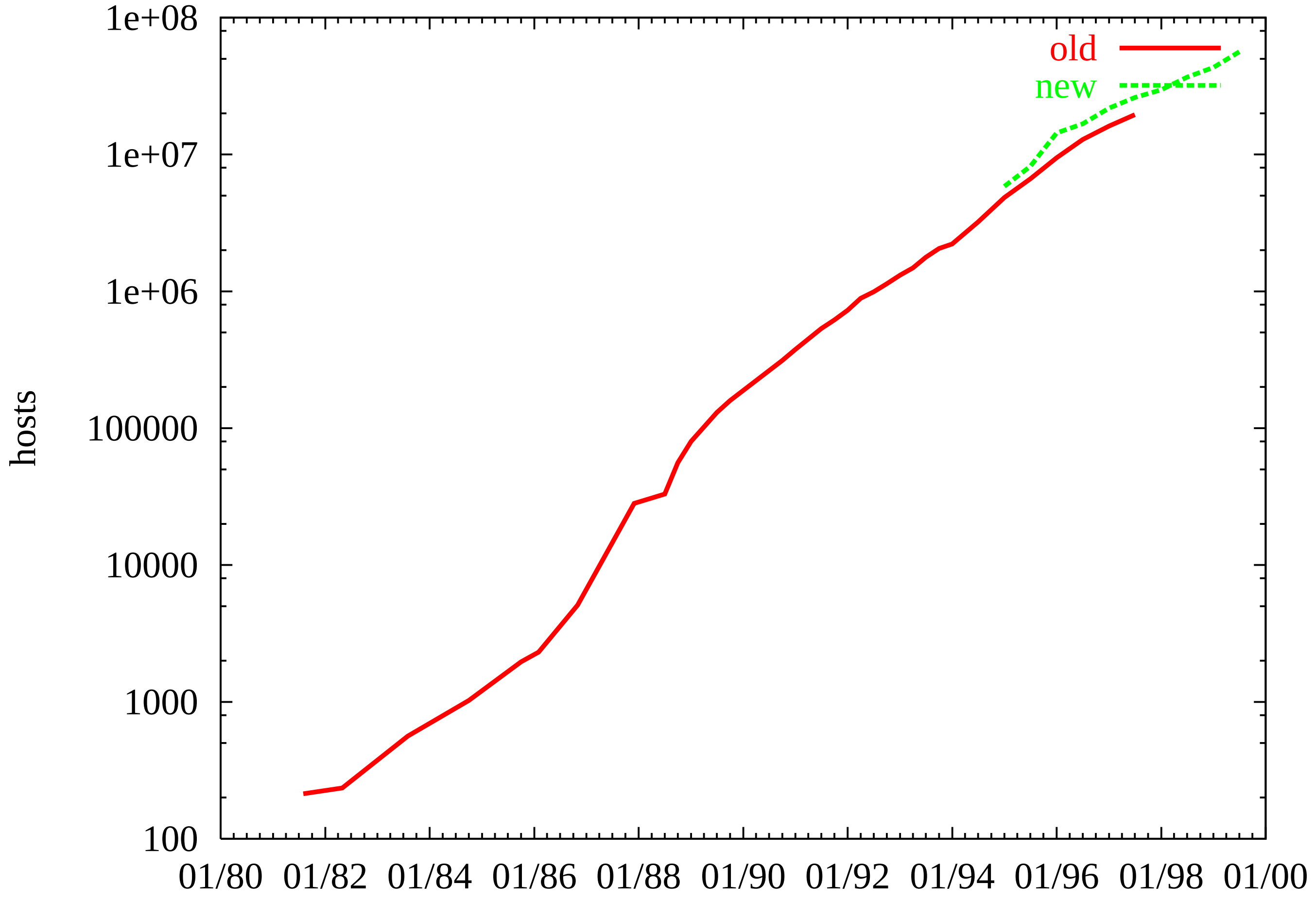
- no competition in registration cost
- registries vs. registrars
- additional TLDs?
- trademark disputes
- access to NSI database
- domain name speculators
- scarcity of good names
- IANA ⟶ non-profit oversight body

http://www.ntia.doc.gov/ntiahome/domainname/dnsdrft.htm

Domain Name Resolution

- hierarchy of redundant servers with time-limited cache
- each server knows the 13 root servers `a.root-servers.net`
- each root server knows gTLDs and refers queries to those
- each domain has ≥ 2 servers, often widely distributed
- also: mailbox translation
- *almost* a distributed database

Internet Growth



Architectural Principles

Architectural principles of the Internet

RFC 1958:

- large existing infrastructure ➠ backward compatibility
- “The goal is connectivity, the tool is the Internet Protocol, and the intelligence is end to end rather than hidden in the network”
- End-to-end functions can best be realised by end-to-end protocols.
- Nobody owns the Internet, there is no centralized control, and nobody can turn it off.

Design principles:

- Heterogeneity is inevitable and must be supported by design.
- If there are several ways of doing the same thing, choose one.

- All designs must scale readily to very many nodes per site and to many millions of sites.
- Performance and cost must be considered as well as functionality.
- Keep it simple.
- In many cases it is better to adopt an almost complete solution now, rather than to wait until a perfect solution can be found.
- Avoid options and parameters whenever possible. Configure them automatically.
- Be strict when sending and tolerant when receiving.
- Be parsimonious with unsolicited packets.
- Circular dependencies must be avoided.
- Objects should be self decribing (include type and size).
- All specifications should use the same terminology and notation, and the same bit- and byte-order convention.

- Nothing gets standardised until there are multiple instances of running code.

Names and addresses:

- No hardcoded addresses.
- Single naming structure.
- Names should be case-insensitive ASCII.
- Addresses must be unambiguous.
- Upper-layer protocols must be able to identify end points unambiguously.

Internet Design Principles

- small, single-function protocols ➠ re-invent similar functionality
- lower-layer protocols: attention to processing efficiency ➠ word alignment, fixed-length fields: CLNP vs. IPv4/IPv6
- incremental: can build small applications easily ➠ embedding of email, HTTP, whois, . . .

Internet Design Principles

upper-layer protocols are text-based ➠

+ “the universal telnet protocol simulator”

+ lots of tools (≈ Unix) such as Tcl, VisualBasic, Perl, . . .

– space overhead

– parsing costs, but usually insignificant

– difficult to represent nested data structures

+/– typically ASCII ➠ internationalization required

but: most upper layer protocols (including NFS, ftp, HTTP, SMTP, RTSP, SIP, . . .) are basically RPC

Soft state vs. hard state

Need to coordinate views of the world in a distributed system ➠

hard state: traditional telecom ➠ state is precious:

- handshake, with retransmission on timeout
- explicit teardown of state

soft state: “optimistic” approach ➠ state is restorable

- send periodic messages to set up or refresh state
- no explicit teardown (except for efficiency) ➠ time-out state
- works even if routers, hosts fail or suffer amnesia
- sometimes allows selective state maintenance
- simpler? higher state overhead? recovery time?