Internet Review

- Architecture
- Addressing and Naming
- basic protocols: IPv4, IPv6, UDP, TCP
- Internet Philosophy

Basic Internet Components

- rules and regulations
- infrastructure
- addresses: MAC (Ethernet), ATM $\leftrightarrow$ IP addresses $\leftrightarrow$ domain names, URLs, ...
- protocols: network, transport,
- applications

Who makes the rules?

ITU.T: telecom standards: G.??, X.??, V.??

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

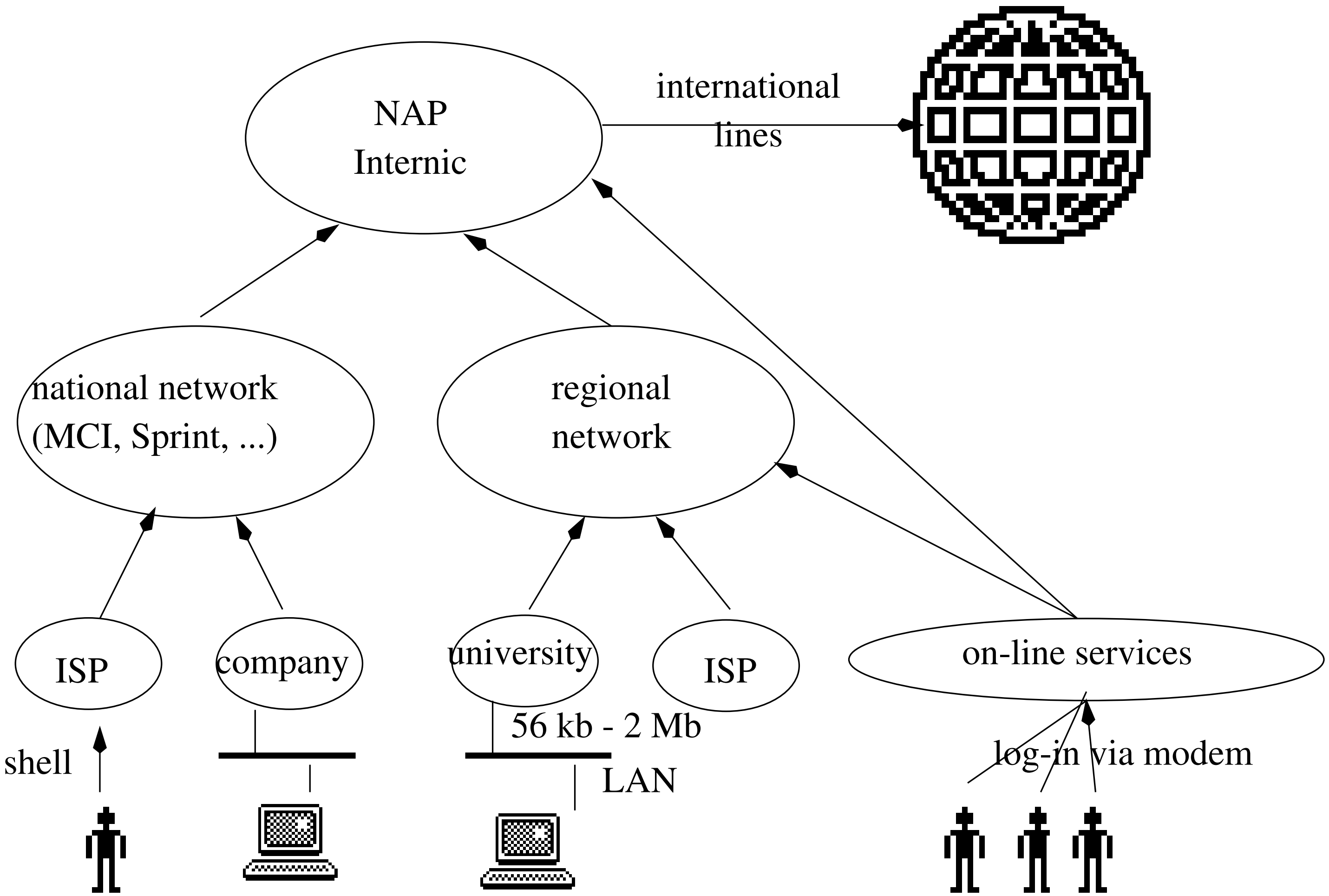
IETF and IESG: (Internet Engineering Task Force, . . . Steering Group)
develop/bless protocols ("RFCs")
open admission, but not quite egalitarian
most RFCs are not standards!

NANOG: North American Network Operators Group

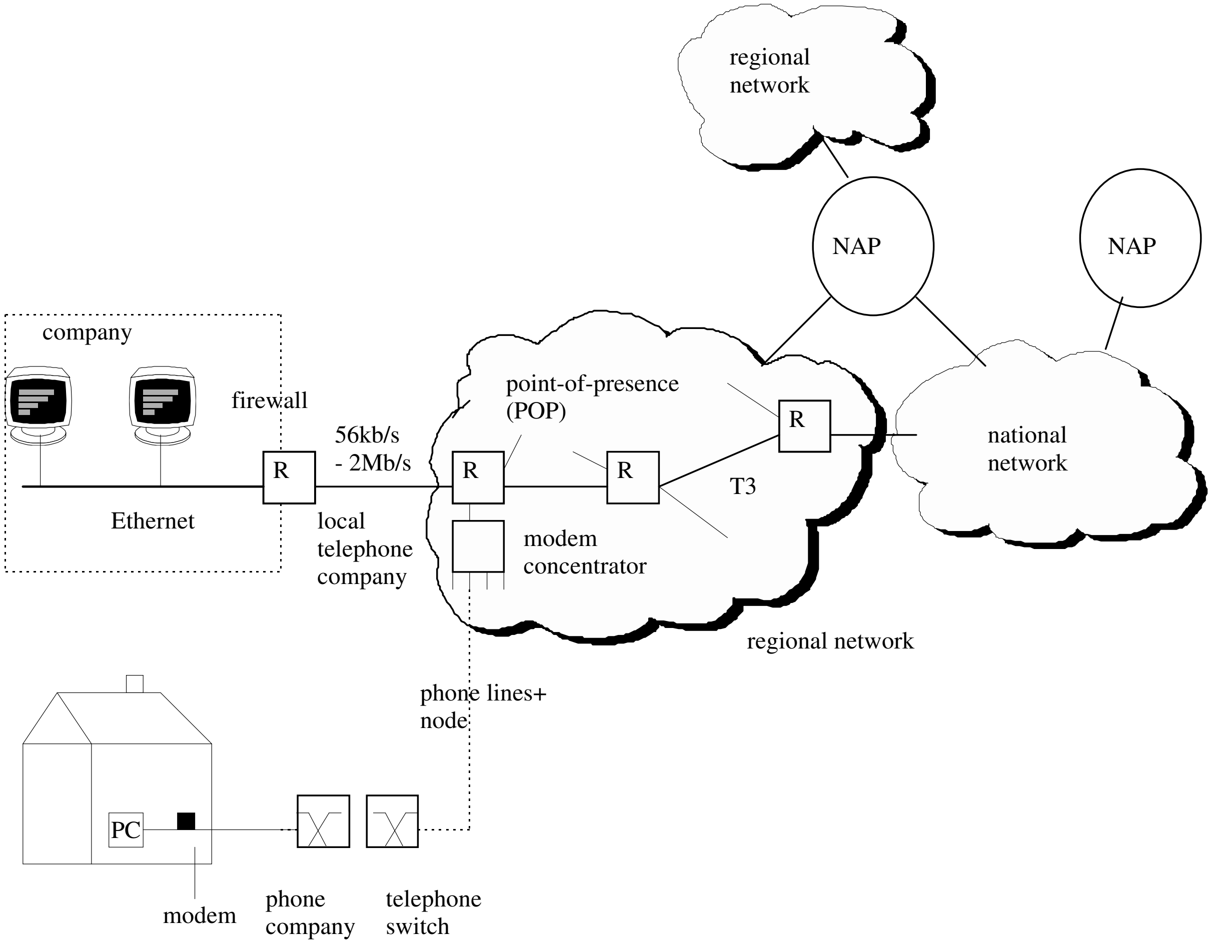
Internic, . . . : hands out names, blocks of addresses

Lots of others *want* to get involved. . .

Who pays for the Internet?



Network Access and Interconnection



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

Internet Addresses

- 32-bit addresses
- (almost) every *interface* has one
- but may
  - change (dial-in)
  - have lots (WWW servers)
  - have none (some routers)
- old: class-{A,B,C} ➠ 2-level addressing: network,host
- new: classless interdomain routing (CIDR) ➠ aggregation, route on prefix and mask

IP Address Classes

class	first octet	hosts per network	nets	used (7.96)
Class A	< 128	16 mio.	128	95
Class B	$128 \ldots 191$	65534	16384	5892
Class C	$192 \ldots 223$	254	2 mio.	128378
Class D	$224 \ldots 239$		268 mio.	dynamic
Class E	$240 \ldots 255$		134 mio.	reserved

- distinguished by prefix
- similar to ISBN, European telephone numbers

The Internet Domain Name System

- hierarchical, dot-separated names
- ➠ multi-level delegation
- by country and by type of organization
- needs to be overhauled (63% of all domains = .com!)

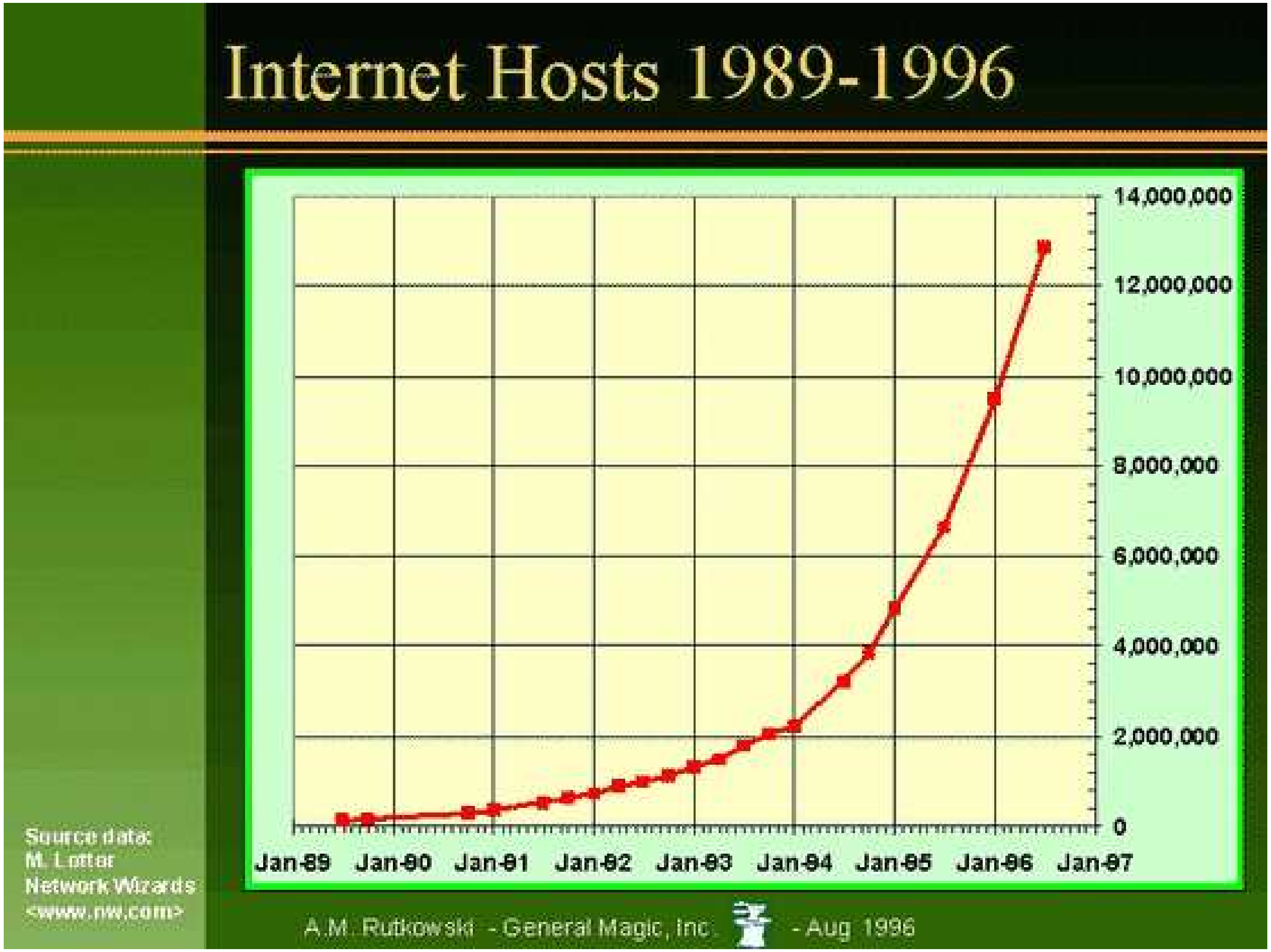
Top-level domains:

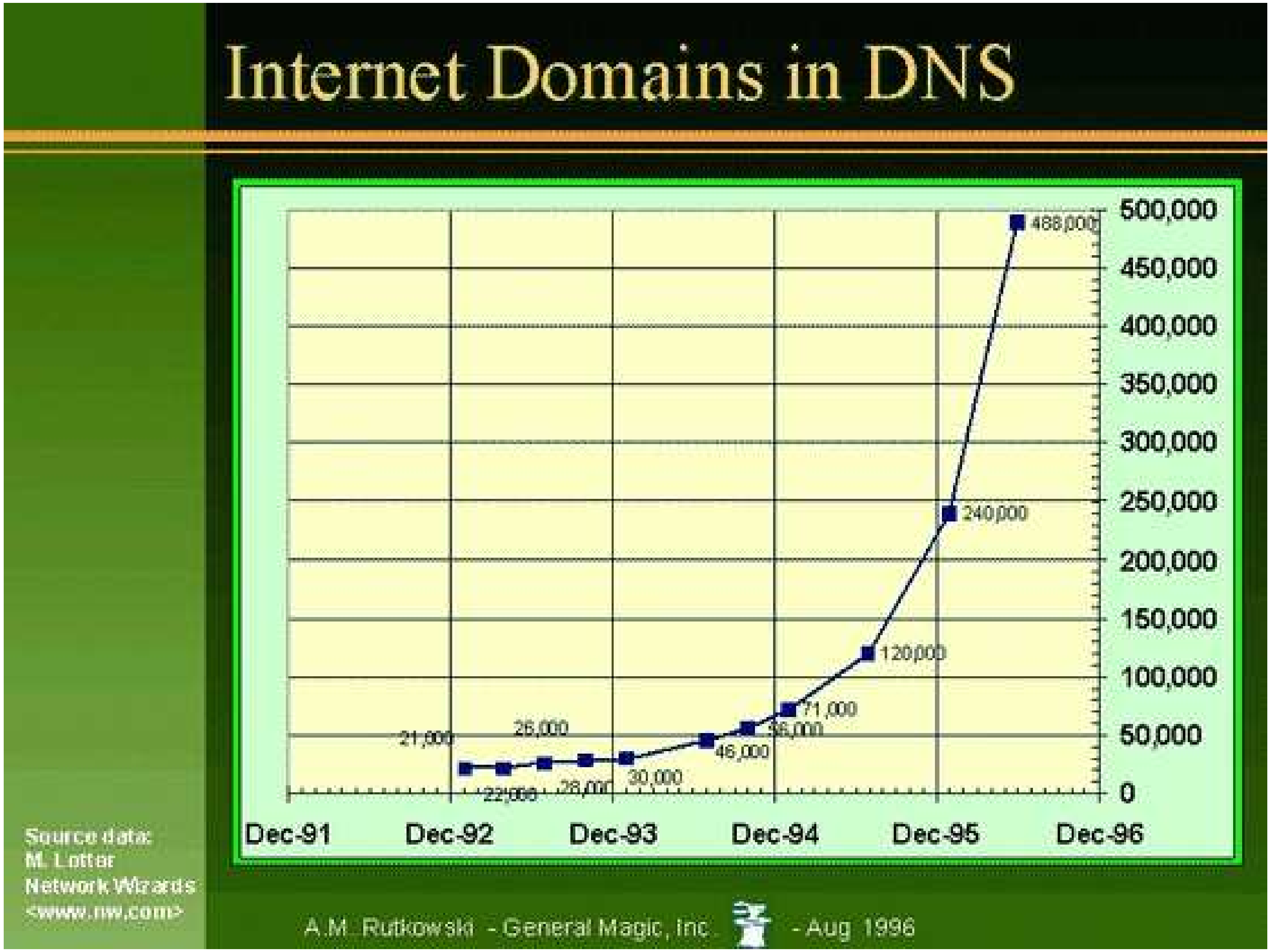
2 letters:	countries
3 letters:	independent of geography (except edu, gov, mil)

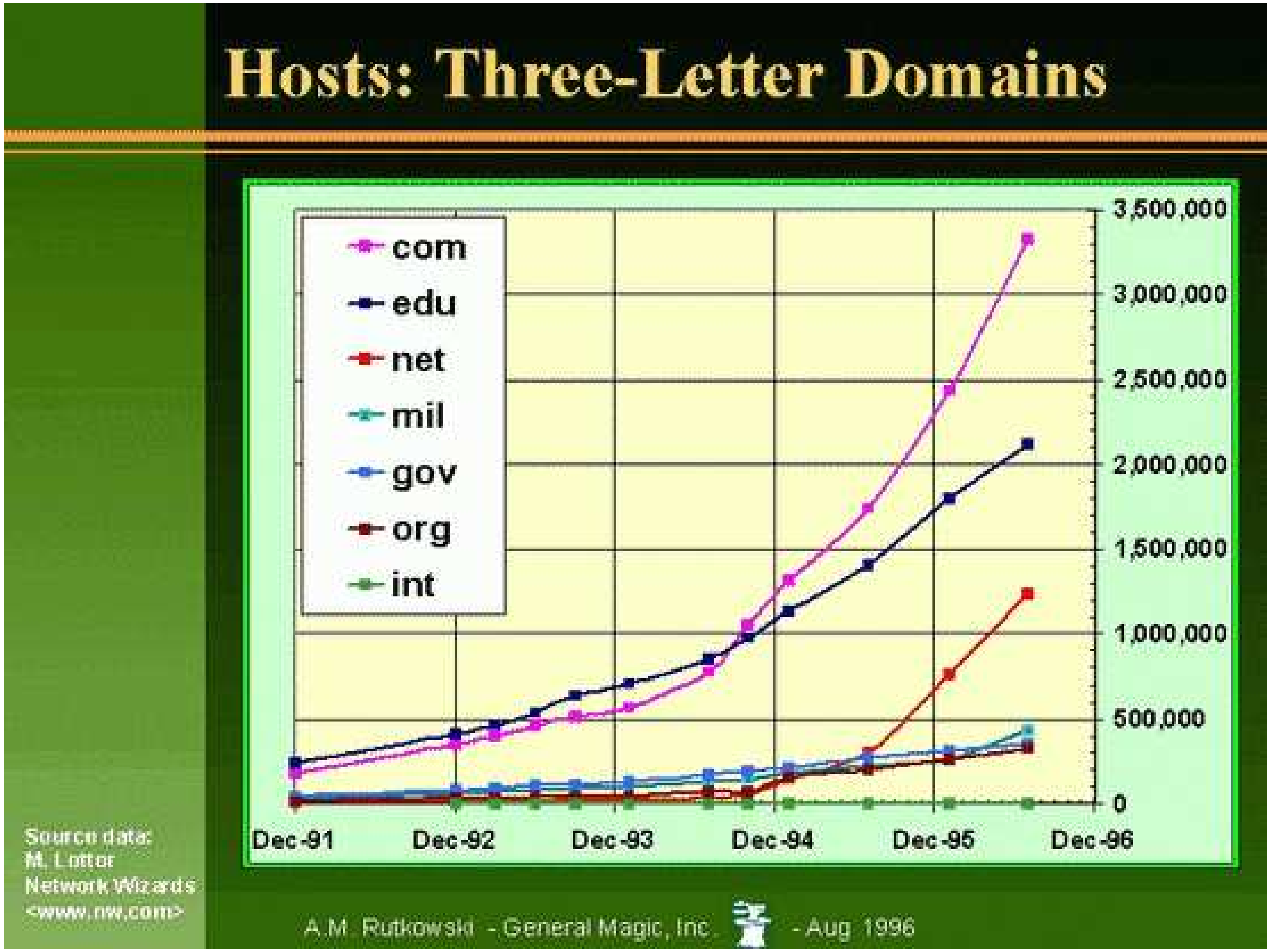
domain	usage	example	hosts (7.96)
com	business (global)	research.att.com	3323647
edu	U.S. 4 yr colleges	cs.columbia.edu	2114851
gov	U.S. non-military gov't	whitehouse.gov	361065
mil	U.S. military	arpa.mil	431939
org	non-profit orgs (global)	www.ietf.org	327148
net	network provider	nis.nsf.net	1232902
us	U.S. geographical	ietf.cnri.reston.va.us	432727
uk	United Kingdom	cs.ucl.ac.uk	579492
de	Germany	fokus.gmd.de	548168

Domain Name Resolution

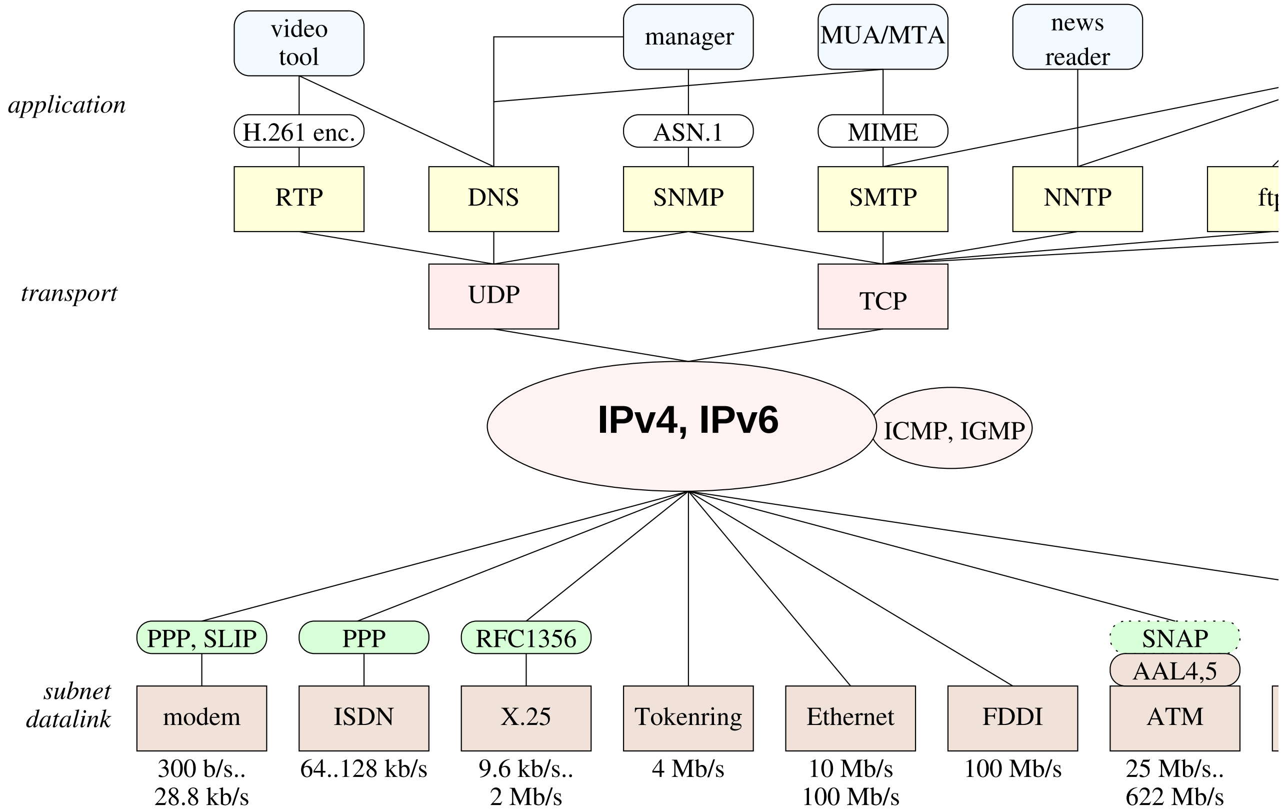
- hierarchy of redundant servers with time-limited cache
- each server knows the (9) root servers
- each root server knows
- each domain has ≥ 2 servers, often widely distributed
- also: mailbox translation
- *almost* a distributed database


Internet Hosts 1989-1996
14,000,000
12,000,000
10,000,000
8,000,000
6,000,000
4,000,000
2,000,000
0
Jan-89 Jan-90 Jan-91 Jan-92 Jan-93 Jan-94 Jan-95 Jan-96 Jan-97
Source data:
M. Lottor
Network Wizards
<www.nw.com>
A.M. Rutkowski - General Magic, Inc. - Aug 1996


Internet Domains in DNS
488,000
240,000
120,000
71,000
56,000
46,000
30,000
28,000
26,000
22,000
21,000
500,000
450,000
400,000
350,000
300,000
250,000
200,000
150,000
100,000
50,000
0
Dec-91
Dec-92
Dec-93
Dec-94
Dec-95
Dec-96
Source data:
M. Lottor
Network Wizards
<www.nw.com>
A.M. Rutkowski - General Magic, Inc. - Aug 1996


Hosts: Three-Letter Domains
com
edu
net
mil
gov
org
int
3,500,000
3,000,000
2,500,000
2,000,000
1,500,000
1,000,000
500,000
0
Dec-91
Dec-92
Dec-93
Dec-94
Dec-95
Dec-96
Source data:
M. Lottor
Network Wizards
<www.nw.com>
A.M. Rutkowski - General Magic, Inc. - Aug 1996

The Internet Protocol Zoo



Protocol Taxonomy

Data link layer shim: SLIP, PPP

network layer + support: IPv4, IPv6; ARP, DHCP, BOOTP

transport layer: UDP, TCP

directory: LDAP

information access: HTTP, ftp, telnet

file sharing: NFS, WebNFS

electronic mail: SMTP, POP, IMAP

routing: OSPF, RIP, BGP, EGP

multicast routing: PIM, DVMRP, CBT

Many upper-layer protocols are basically RPC…

IPv4

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

IPv6

- 32 bits ➠ $4 \cdot 10^9$ hosts, but can't number sequentially
- aggregation for routing $\leftrightarrow$ efficiency
- every {light bulb, toaster, fire alarm, ...} with IP address ➠ address shortage
- 128 bits instead of 32 bits ➠ 340,282,366,920,938,463,463,374,607,431,768,211,456 ($3 \cdot 10^{38}$) hosts or 665,570,793,348,866,943,898,599 hosts/m^2 of earth surface (more than a mole!)
- hierarchical assignment ➠ inefficiency ➠ only $8 \cdot 10^{17}$ to $2 \cdot 10^{33}$ addresses if similar to telephones, etc.

IPv6: While we're at it…

- *flows*: short handle for group of packets (but $\neq$ connection)
- destination address names *intermediate* destination
- well-defined priority field
- no header checksum
- unlimited option space
- jumbograms, TTL changed, e-e fragmentation, …
- ➠ 40-byte header (instead of 20)

UDP

- same service as IP
- adds *ports*: differentiation of services within host
- adds payload checksum
- overhead: 8 bytes

TCP

- reliable, sequenced byte stream on top of datagram
- connections with graceful close or hard reset
- windowed congestion control: not part of the protocol spec…
- windows:
  - flow control: avoid overrunning slow receiver ➠ advertised by receiver
  - congestion control: avoid network overload ➠ maintained by sender – *cwnd*
- probe for available bandwidth:
  - *slow start*: add one segment size for each ACK ➠ exponential
  - *congestion avoidance: cwnd* += 1/*cwnd* for each ACK ➠ linear growth in window

- loss ➠ timeout or duplicate acks, retransmit and reduce *cwnd*
- Different flavors: Tahoe, Reno, Vegas, vary in response to loss

- bandwidth-delay product: capacity of "pipe" ➠ continental T3 = 45 Mb/s $\times$ 5000 km $\times$ 5 μs / km $\approx$ 140 kB
- extensions: large ($>$ 64 kB) windows for fat/long pipes, SACK
- overhead: 20 bytes

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
- 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, . . .
- upper-layer protocols are text-based ➠
  - "the universal telnet protocol simulator"
  - lots of tools ($\approx$ Unix)
  - space overhead
  - parsing costs
  - difficult to represent nested data structures
  - typically ASCII ➠ internationalization required

- most upper layer protocols (including NFS, ftp, HTTP, SMTP, . . .) 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?