TCP/IP intro

Jim Binkley
http://www.cs.pdx.edu/~jrb
jrb@cs.pdx.edu

Very Brief Internet History

- 1957 Sputnik/USSR. US creates ARPA
- 62 - Paul Baran, packet-switches (missiles)
- 69 -ARPA/DOD starts ARPANET
- 71 - 15 nodes
- 73 - Ethernet/Bob Metcalfe Harvard Ph.D
- 79 - USENET/UUCP over modems
- 82/83 Darpa starts using TCP/IP on Arpanet
- 83 - BSD UNIX with TCP/IP, enet

Inet history, cont

- 84 - DNS and 10k hosts
- 88 - 6k/of 60k hosts visited by Morris worm
- 89 - IETF and IRTF under IAB
- 92 - 1st MBONE audio/video over Inet
- 93 - Hillary is root@whitehouse.gov
- 93 - WWW begins to take over
- 94 - businesses and biz begin to take over
- 94 - gov. decides OSI not best idea...

citations:

- 95 - NSFNET replaced by commercial backbones
- 93-now Internet does not fail ...
- 2002 - term “switch” no longer refers to circuits ...
- See **Hobbes Internet Timeline**: RFC 2235
- http:info.isoc.org/guest/zakon/Internet/History/HIT.html for most of these

Internet Growth - DNS surveys

Date	*Hosts*	*Nets*	*Domains*
1969	4		
1984	1024		
1987	28174		
1989	130000	650	3900
1990	313000	2063	9300
1992	727000	4526	
1993	1313000	7505	21000
7/94	3212000	25210	46000
7/95	6.6 M	?	120000
7/96	12.8M	?	488000
97	20-30M	45/55k	>1m

By 2000, even dogs will be on the Inet?

scalability issues

- # ip addresses, # ip nets
  - IPv6 may address this (pun in there …?)
- # dns names (variation, too many .com)
  - politics as well as engineering
- # of routes in routers
  - CIDR - classless internet domain routing
  - IPv6 doesn’t help

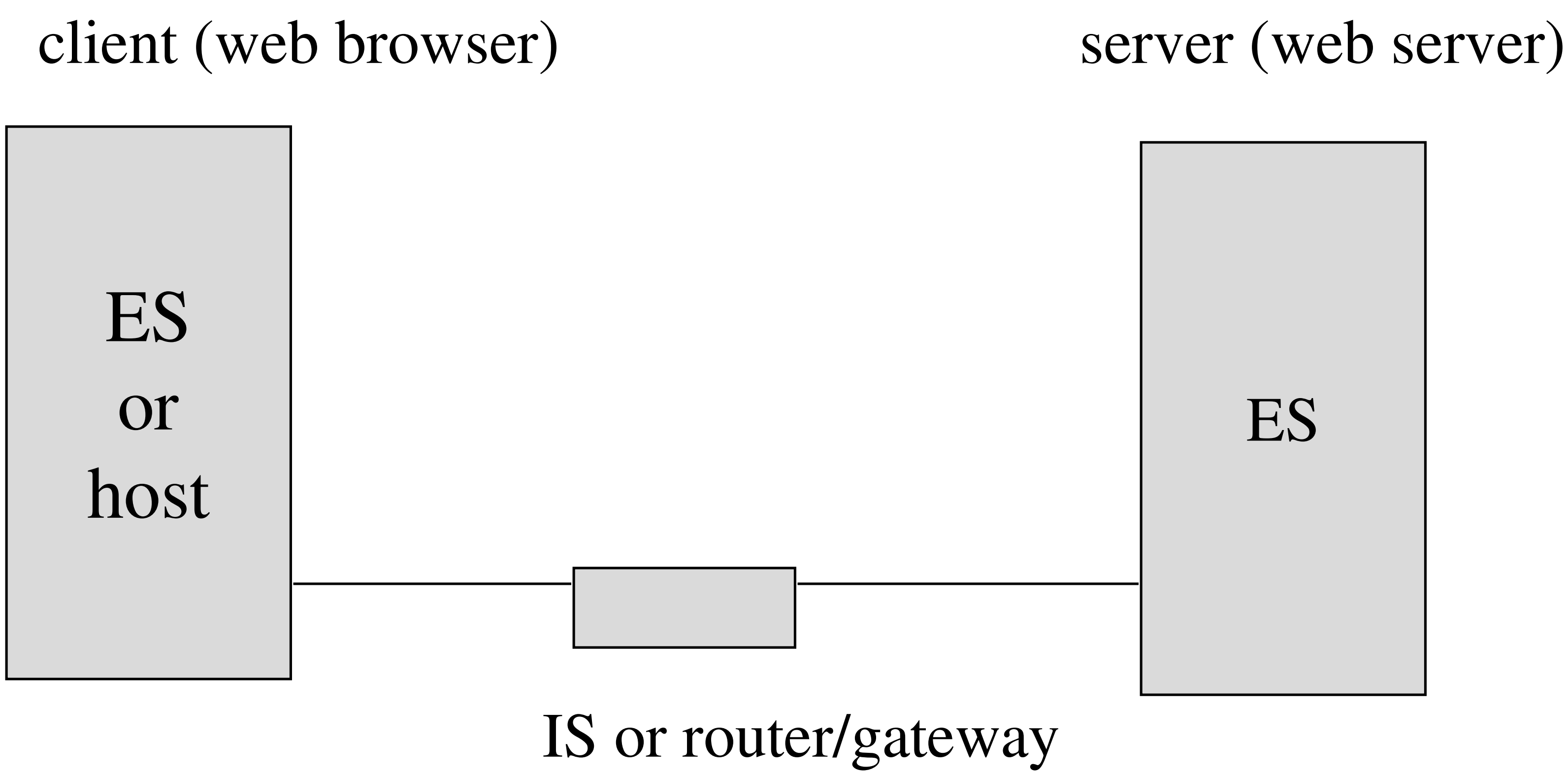
Tcp Intro

- TCP/IP - Internet protocol suite, TCP and IP are *protocols* in the suite, there are more
- open system, not proprietary, stacks from different vendors **INTEROPERATE**
  - Novell ipx, Apple appletalk - closed systems
- **Internet** - uses TCP/IP protocols
- amazingly: THERE CAN ONLY BE ONE INTERNET ...

Protocol layers

- protocol layers - each layer has its own focus, associated *encapsulation* and *addressing*
- 4 layers in TCP/IP (older)
- 7 in Open Systems Interconnect (newer)
- layer is logical idea and may be in fact be ignored in implementation

end systems and intermediate systems

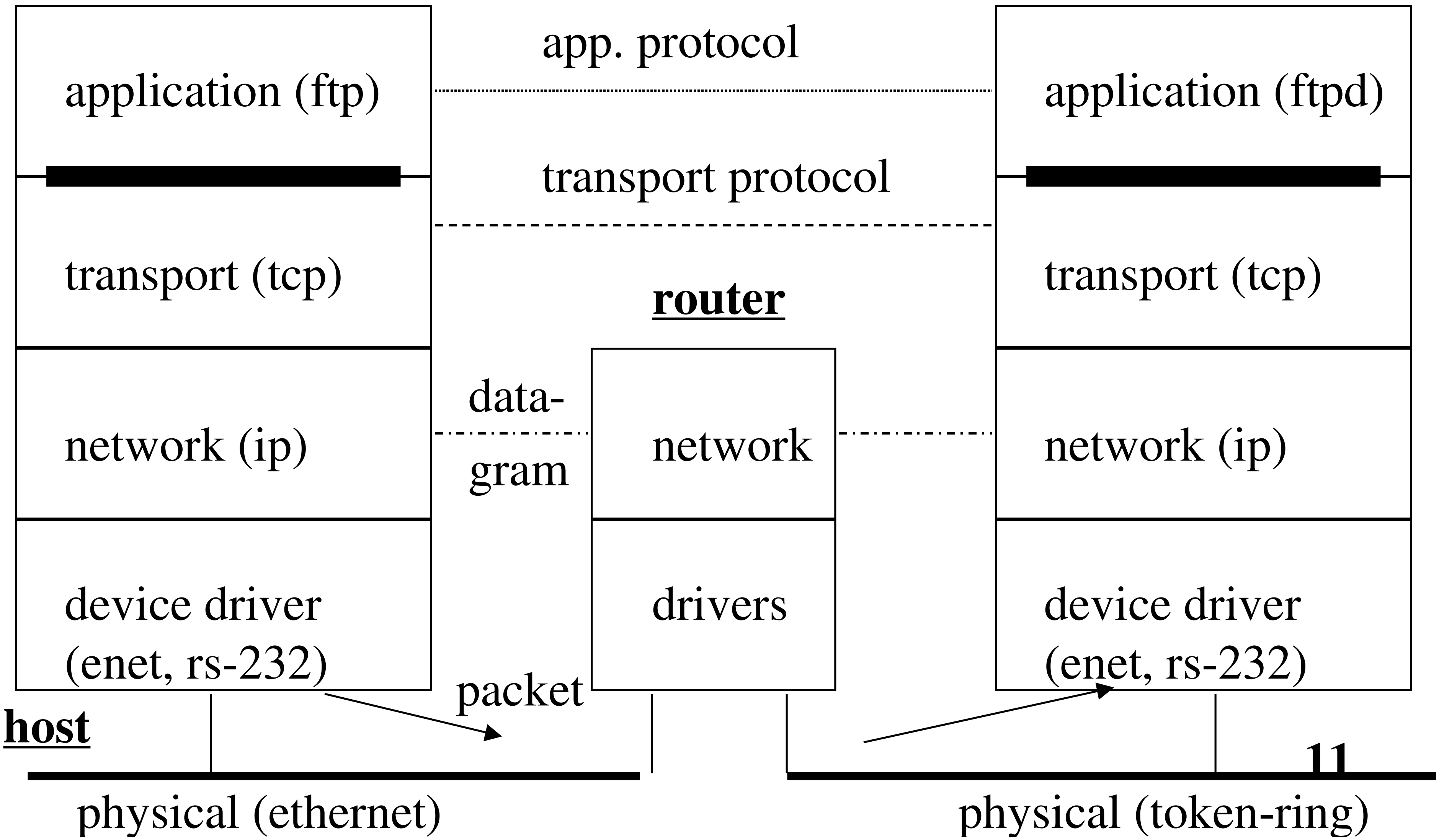


ISO/OSI Reference Model

layer 7	application	app. protocol
layer 6	presentation	
layer 5	session	
		etc ....
layer 4	transport	**end/end**
layer 3	network	**routing**
layer 2	data link	**node addr**
layer 1	physical	**“wires”**

learn the numbers 1..7

TCP Layering



Internet Protocols

apps	email (smtp) telnet/rlogin ftp/rcp http(www)/gopher	dns nfs snmp rip bootp	ping traceroute ospf
transports	**tcp**	udp	“raw”/ip
network	**ip** + icmp + igmp		
device	arp/rarp ethernet II (or 802.3)	slip/ppp/hdlc phone line, ISDN	

TCP layers/architecture

- data flows up/down stack
  - each layer on write adds header/addr. info. This process is called **encapsulation**
  - on read, data is **demultiplexed** - decide which protocol upstairs to feed it to, and **decapsulated**
- demux example: from link layer, packet
  - could go to IP, ARP, RARP

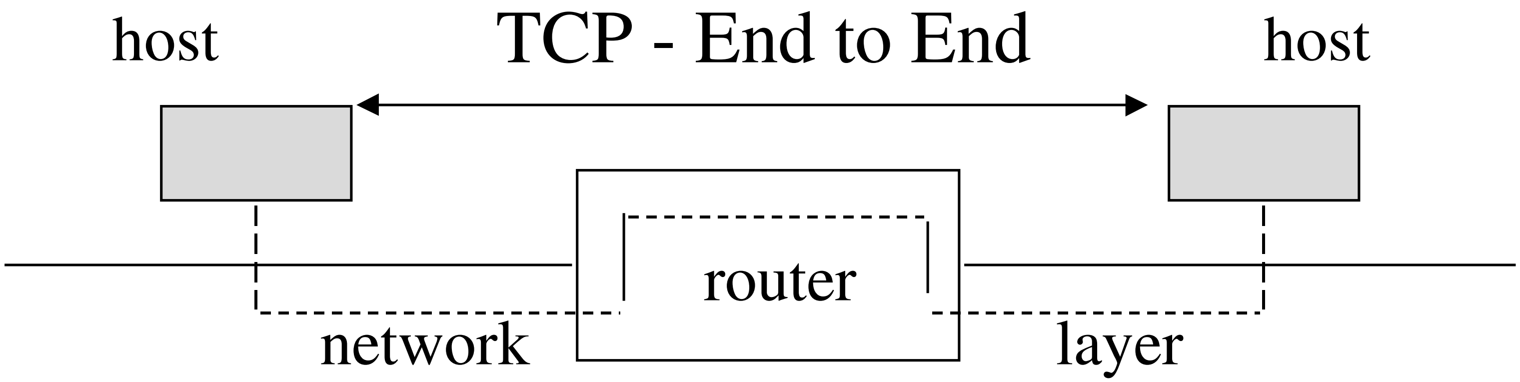
transport/network layer

network layer - hides physical layer

ip is hop by hop

transport layer - end to end, error correction

tcp is end to end



Two Big Ideas

- **peer layers in stack virtually talk to each other -- this is a "protocol"**
  - tcp talks to remote endpoint tcp
  - ftp clients talks to ftp server
  - ip src talks to ip dest and may talk to routers too
- **network layer hides transport/apps from exact details of physical layer**
  - routers glue together networks

addressing/encapsulation

- application -
  - *Domain Name System* (sirius.cs.pdx.edu)
  - sockets
- tcp/udp, use *ports*, 16 bit unsigned ints
- ip - uses *IP address*, 32 bit int
  - (net, subnet, host)
- link layer, ethernet uses IEEE 48 bit *MAC address*

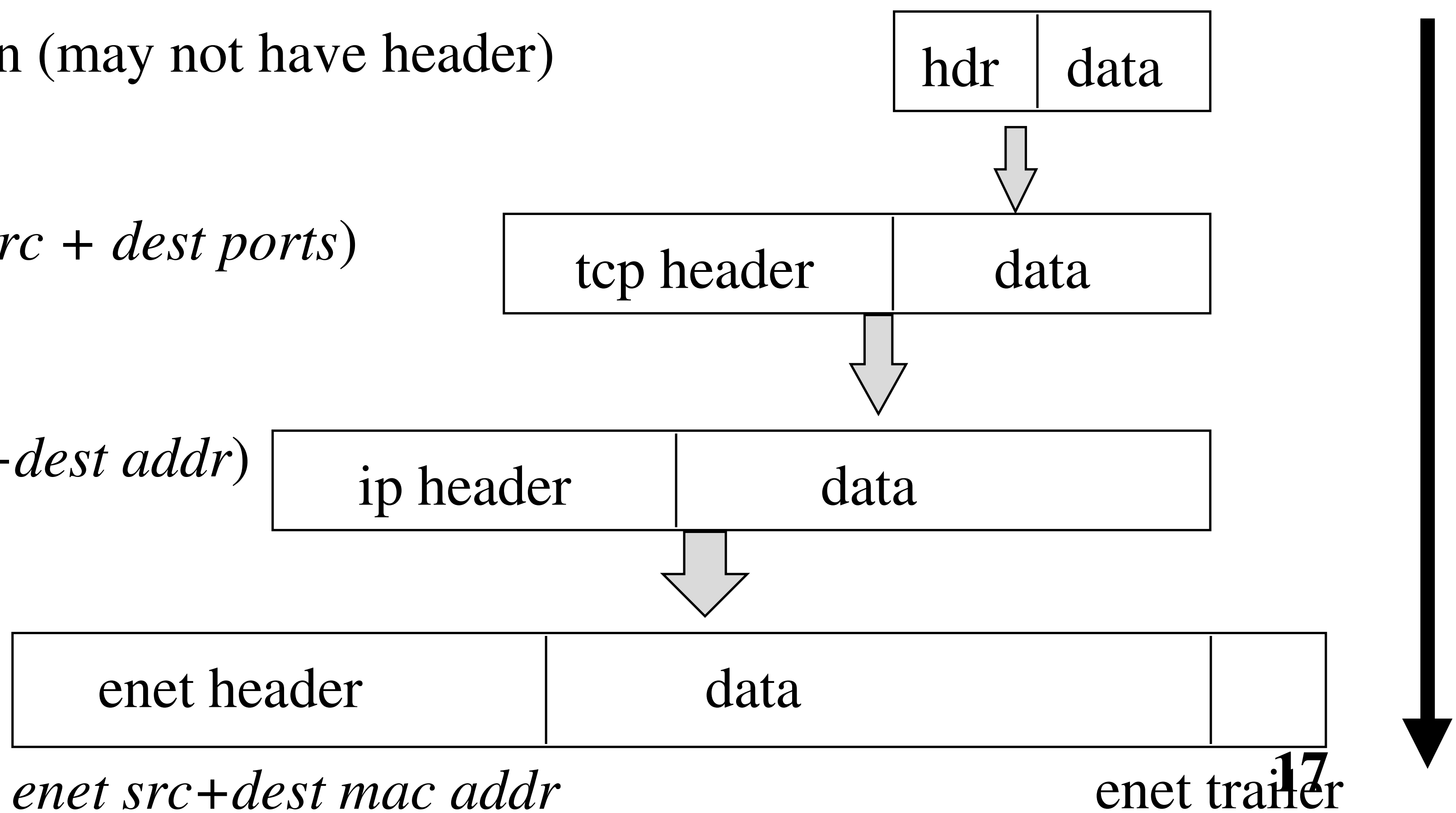
encapsulation (packet goes out)

application (may not have header)

tcp/udp (*src* + *dest ports*)

ip (*ip src+dest addr*)

enet



IP addresses

- per interface. each i/f has
  - (ip address, broadcast address, subnet mask)
- (network, subnet, host)
- written in *dotted decimal* in *network byte order* (big-endian)
  200.12.0.14 (0..255)
- 5 classes, A to E, each takes a bit at the hi-order end

IP class address table

class	bits	net	host	range	
class A	0	7 bits	24	0.0.0.0	127.255.255.255
class B	10	14	16	128.0	191.255.255.255
class C	110	21	8	192.0	223.255.255.255
class D	1110	28	-	224.0	239.255.255.255
class E	11110	27	-	240.0	247.255.255.255

ip addresses, cont

- 3 types of IP address (topographical)
  - unicast
    - 127.0.0.1, 201.3.4.5
  - broadcast
    - 255.255.255.255, 129.14.255.255,
    - 0.0.0.0
  - multicast
    - 225.1.2.3

ip address, cont

- uniqueness must be handled by humans
- various IP authorities at this point
- U.S. authority is ARIN, www.arin.net
- ISP feeding chain in U.S., ends up at ARIN
- IP addresses + A.S. numbers (later)
- DNS from Internic: rs.internic.net, Network Solutions (www.networksolutions.com), ICANN (www.icann.org)
- on unix: try the whois utility, network registry DB
  - % whois pdx.edu
  - % whois -h www.arin.net 129.95

obtaining an IP address

- you used to get it from the Internic, but now usually from IP/pipe "ISP"
- we need to worry about making sure that addresses can be hierarchical
  - CIDR blocks, allocated top-down from your "provider" to you
  - if you change providers, you get to renumber

transport/port numbers

- TCP/UDP unsigned 16-bits shorts
  - 0..64k-1
- servers are known by "well-known" ports
  - e.g., telnet 23, http 80, ftp 20, mail 25
- IAssigned Numbers Authority (IANA) assigns them
- on UNIX stored imperfectly in
  - */etc/services*
- UNIX reserves ports 0..1023 for "root"/su-only
- dynamically viewed with % *netstat -a*

Domain Name Systems

- primary function - map human readable names to IP numbers
  - sirius.cs.pdx.edu -> 131.252.220.13
- done entirely as application on top of UDP
- client-server model, with DNS servers in relatively flat hierarchy
- **o.s. deals in ip addresses, not DNS names**

client - server paradigm

- applications (and sometimes o.s.) organized in application architecture paradigm called *client-server*
- usually but not always message oriented
- client app talks app. protocol to remote server that processes each message
- servers might be
  - **iterative** (process message to conclusion) / UDP
  - or **concurrent** (**master/slave**) / TCP

client-server, server forms:

- iterative:

```
do forever
        wait/read client message
        process message
        write ACK to client
```

- concurrent

```
do forever
        wait for connection
        fork (spawn task)
                child does i/o and exits
```

Internet - what is it?

- elephant and blind men ... many Points of View
- a suite of many app protocols on top of TCP/UDP/IP - open system, etc., etc.
  - packet switched net on top of circuit/telco
- on MANY physical networks, WAN/LAN
- the World Wide Web (http/TCP)
  - or chat rooms?
- a computer network that can survive atomic attack?
  - but where network security is an oxymoron?

Internet - what is it?

- *Internet* - the world-wide set of nets combined with TCP/IP
- *internet* - a bunch of nets tied together
- The Internet is built on TOP of the phone co's net and views the TELCO network as a link layer black box (**subnet model** as opposed to **peer model**)

physically?

- 10+ Network Access Points or NAPs/MAEs
  - where backbones meet
- N backbones that cross the U.S.
  - UUNET/PSI/GTE(BBN Planet)/Sprint/C&W
  - T3, or faster OC3/OC12 ATM
- regionals (being purchased by the above)
- local (and national) ISPs
  - AOL/teleport/raindrop labs
- Jane User with her pc/28.8k modem

telco WAN technologies

- ATM (maybe) OC3 (155), OC12 (655)
- T3 (<45Mbps) - STM - $25k/month
- SMDS
- T1 (1.54Mbps) - $500 - $2k/month
- frame relay (shared load)
- ADSL - new, cable modem, 256-T1 or so
- ISDN 64/128k
- analog modems (POTS) 56k/28.8k/14.4k

ISPs - Internet Service Provider

- provides you with a connection + X services
- services might include:
  - a wire, however big/small
  - ip address space (or an ip network for N lan machines) + DNS name/server, ppp (routing)
  - SMTP email (POP accounts)
  - UNIX login account
  - NNTP Usenix news
  - web pages or ... servers or “e-commerce”

who controls it?

- Internet is world-wide - question of govt. control is very interesting
- IAB/IETF determine standards
- but industry may preemptively determine standards (early bird ...)
  - Netscape/Microsoft/Sun/Intel/Cisco

Internet Organization (well...)

- ISOC - Internet Society. professional society to faciliate, support, promote Inet
- IAB - technical oversight and coordination, falls under ISOC
- IESG - Inet Eng. Steering Group oversees:
- IETF - meets 3 times a year, develops, argues over, and standardizes protocols for Inet. 70-80 wgs. Organized in areas, e.g., routing area.
- IRTF - Internet Research Task Force - long term research, just a few people compared to IETF

Standards Process

- standards called RFCs - Requests For Comment
- numbers > 2000 now
- IETF wg members write "drafts", eventually hopefully may become standards
- not all protocols have RFCs. not all RFCS are actually used
- *% ftp ftp.isi.edu (cd in-notes) or go to IETF web site*

RFCS, continued

- some important RFCs:
  rfc 1700 - Assigned Numbers RFC
  - (now IANA web site)
  rfc 1500 - Official Protocol standards
  rfc 1122, 1123 host and protocol
  requirements - numerous
  corrections for basic protocols
- see rfc index for latest info

TCP/IP free "stack" implementations

- "stack" == o.s. part, not the apps
- de facto source standard is BSD, now 4.4
  - 4.2 BSD 83 - first widely spread tcp/ip
  - 4.3 BSD 86 - perf. improvements
  - 4.3 BSD Tahoe 88 - slow start, congestion avoidance
  - 4.3 BSD Reno 90 - tcp header prediction, slip header compression, new router algorithm
  - 4.4 BSD 93, multicasting
- others: KA9Q for dos; linux (unix)
- 4.4 BSD book, Steven's volume 2 (freebsd)
- reference implementations: bsd tcp/ip, apache, bind, mrouted, gated, etc ...