Host Naming

15-441, Computer Networks
March 20, 2006

Dave Eckhardt
some slides from Dave Maltz

Overview

- Three names for your PC
  - Why?
- Two “resolution” protocols
  - DNS, ARP
- Turning on
  - DHCP

Three names for ~~your~~ my PC

- PIPER.NECTAR.CS.CMU.EDU
  - What's a “nectar”?
  - What's a “piper”?
- 128.2.194.80
- 00-20-AF-D9-FD-CA

All are globally unique

- Isn't *one* globally unique name enough?

Questions about names

- Who uses the name?
  - For what?
- Who owns/defines the namespace?
- How long is the name valid?

PIPER.NECTAR.CS.CMU.EDU

- Who?
  - Human beings
- What?
  - Remembering a name for each box
  - Crude service-location mechanism
    - ***www***.<organization>
  - Crude ***device***-location registry
    - dsl093-172-001.pit1.dsl.speakeasy.net
    - 110.ge-0-0-0.cr1.wdc1.speakeasy.net

Fun break – hostname schemes

- Animals, birds, dinosaurs
- Cars, wines
- CMU SCS Facilities
  - Desktop machines: astronomical entities
  - Servers: fruits, nuts, vegetables
- NECTAR Project: self-destructive celebrities
- Wean cluster: medication
- MIT AI Lab: Breakfast cereals

PIPER.NECTAR.CS.CMU.EDU

- Who owns the namespace?
  - Broadly, CMU School of Computer Science (see below)
- How long is it valid?
  - Lifetime of “the machine”
- What does it “mean” (bind to?)
  - How long is the ***binding*** valid?
    - See below

128.2.194.80

- Who/what ***pairs***
  - Who=IP router, usage=...
  - Who=end-system hosts, usage=

IP Routers

- Usage="Which link does the packet leave on?"
  - "Definition" of IP router:
    - Box computing "IP address ⇒ departure link"
  - Used as table lookup key
    - Addresses should be short, mostly fixed length
    - String would ***not*** do
  - IP address structure
    - "Network ID": top bits
    - "Host ID": bottom bits
    - Network/host division depends on frame of reference

IP routers

- Usage=Link parameters (some optional)
  - IP address $\Rightarrow$ link/station address (ARP, see below)
  - IP address $\Rightarrow$ link-level encryption state (802.11)
  - IP address $\Rightarrow$ link-level scheduling policy (rare today)
  - Again, IP address is table-lookup key

End-system hosts

- Usage=connection management
  - TCP connection ***defined by*** (IP1, port1, IP2, port2)
    - "only" 65536 TCP connections per host pair
  - Client: my _____ server is x.y.z.w
  - IPsec security layer: IP address ⇒ security state
    - For end-to-end security, independent of link-level security
- Usage=access control
  - Trust certain IP addresses more than others
    - *Very* weak "security"; you *must* add something more

128.2.194.80

- Who owns the namespace?
  - Roughly, CMU School of Computer Science
- How long is it valid?
  - Historically: “a long time”
    - 128.2 = CMU.EDU
    - 194 = some chunk of CS
    - 80 = random selection
    - No need to change for “lifetime of machine”
  - But...

128.2.194.80

- Nothing fails like success
  - Internet popularity ⇒ IP router table size explosion
- CIDR compresses via hierarchy
  - 12.0.0.0/8 (12.*) belongs to ATT.net
  - 216.218.128.0/17 belongs to he.net (Hurricane Electric)
  - 216.218.132.24/29 belongs to Panasas.com
- Change ISPs, your netblock changes
  - ... “ISP” can be Starbucks 802.11

128.2.194.80

- Who owns the namespace?
  - Your ISP, probably
- How long is the name good for?
  - At least a couple of minutes

00-20-AF-D9-FD-CA

- Who assigns?
  - IEEE http://standards.ieee.org/regauth/oui/
  - 00-20-AF assigned to 3Com
  - D9-FD-CA assigned by factory

IEEE 802 MAC address

- Globally unique address
- For every “Ethernet” “card”
- “Ethernet”
  - Or 802.11, or ATM, or Frame Relay, or ...
- “card”
  - Semi-permanent expansion card
  - PCMCIA/CompactFlash card
  - Chip on motherboard

IEEE 802 MAC address - Usage

- “Station” identification on “a network”
- Cooperating set of bridges agree on location
  - Which bridge owns which stations
  - Dynamic “spanning tree” algorithm
- Not “routable” outside that network
  - If somebody steals my laptop, knowing the Ethernet address does not generally help me find the laptop.
  - Then why is it ***globally*** unique?

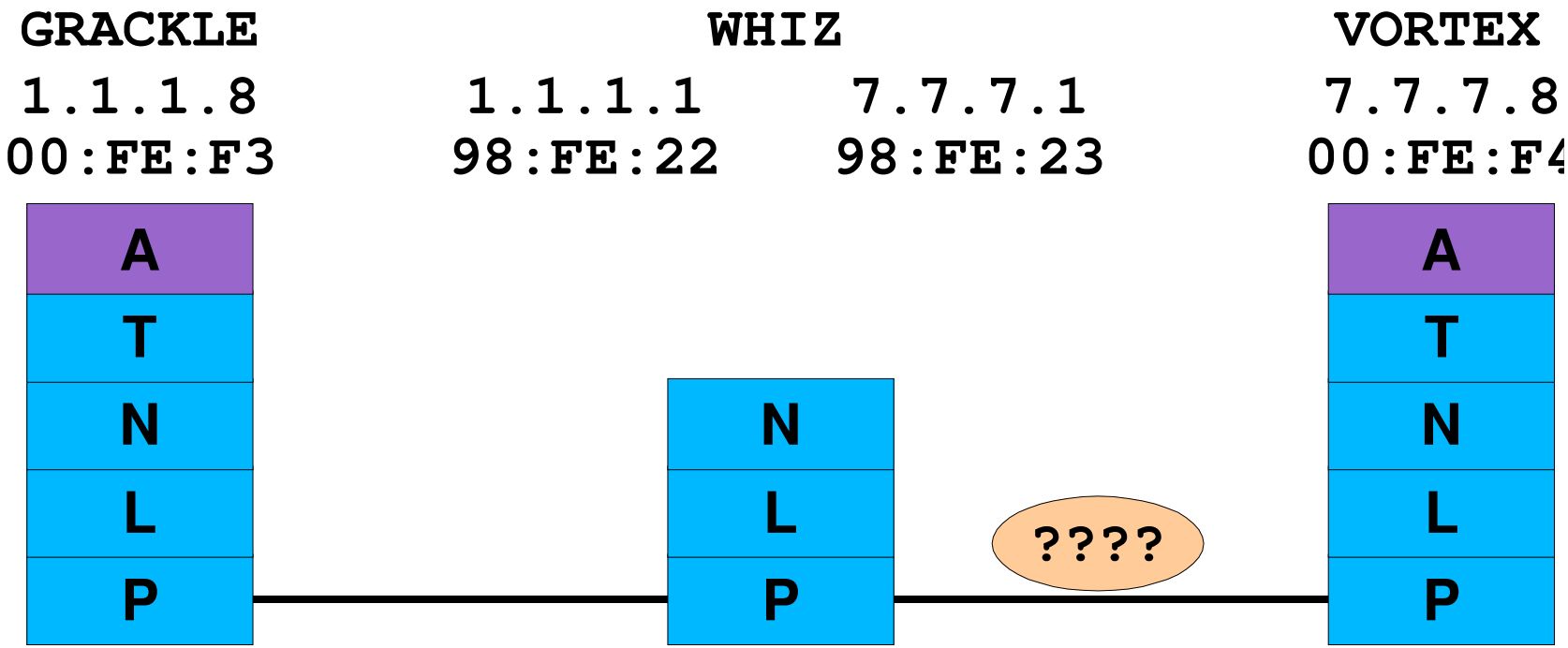
Must IP routers know MAC addresses?

- Why do we need a MAC address?
  - Can't IP-layer entity ignore link-level addresses?
- IP was designed to be ***subnet-independent***
  - ARPAnet, SATnet, ARPA mobile radio network
  - DIX Ethernet, IBM Token Ring, Corvus Omninet, PPP
  - Each link has its own kind of address
    - Differ in size, meaning
- “In theory” IP forwarding is “about” IP addresses, doesn't involve link addresses

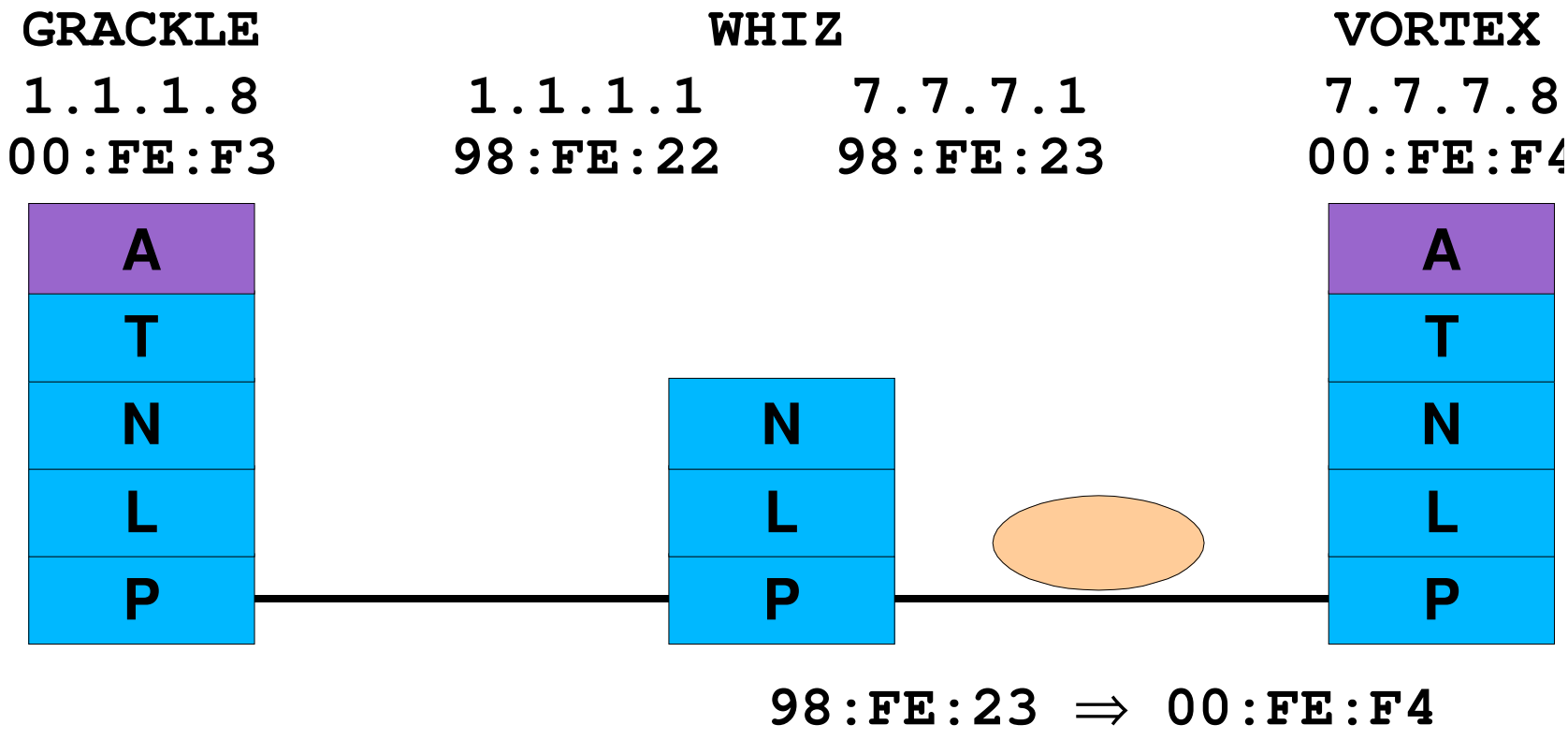
Must IP routers know MAC addresses?

- Link layers are designed to be *network-independent*
- It is a feature that Ethernet can carry
  - PUP, IP, XNS, Banyan Vines, DECnet, SNA
  - Each network layer has its own kind of address
    - Differ in size, meaning
- Link layers use MAC addresses for efficiency
  - Each station can ignore not-for-it traffic in hardware
- "In theory" MAC frames can contain any IP address–or none!

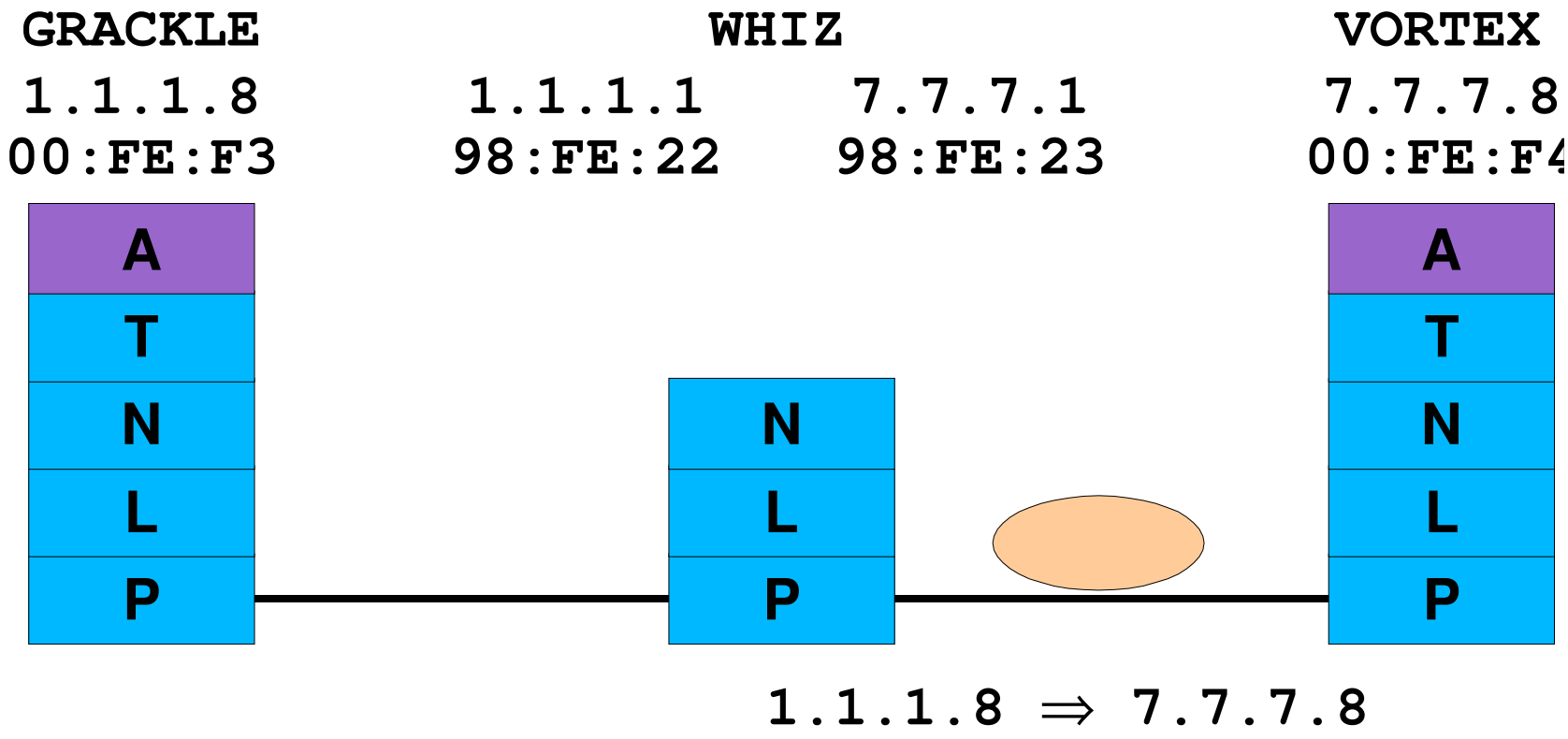
Which Addresses Are In The Packet?



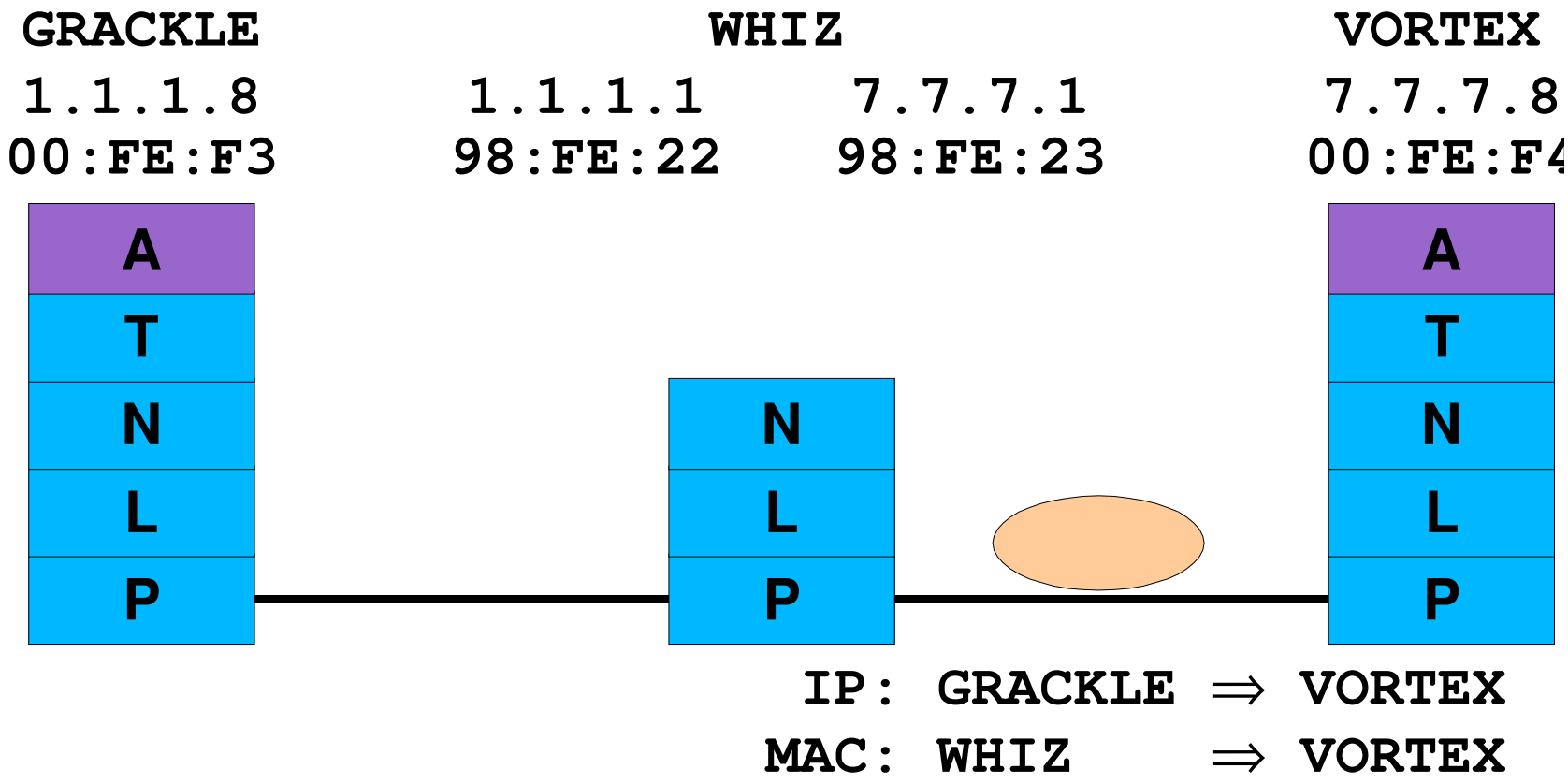
Frame MAC Addresses



Packet IP Addresses



Address “Mismatch”



Must IP routers know MAC addresses?

- Result: router-level entities must know MAC addresses
- To forward toward destination
  - Know MAC address of next-hop router
- To deliver to final destination
  - Know MAC address of end-system host

Three names for my PC

- PIPER.NECTAR.CS.CMU.EDU
  - For human use
  - Good for “a long time”
  - Maps to IP address for IP routers efficiency
- 128.2.194.80
  - For use by IP routers and IP protocols
  - Good while attached via a given ISP
  - Mapped to link-level address for link-level efficiency
    - (not point-to-point links)

Three names for my PC

- 00-20-AF-D9-FD-CA
  - Address used by Ethernet link hardware
  - Good for lifetime of interface card
  - Binding to *machine* is variable
    - Motherboard: pretty permanent
    - PCI card: rarely moved from one machine to another
    - PCMCIA/CF card: resides in a machine at least 1 minute
  - Binding to *IP address* is variable too
    - Change ISPs...

Three names implies two lookups

- User specifies host name
- Data packet sent to IP address
- Last-hop router must know MAC address
- Two lookup problems
  - Name $\Rightarrow$ IP address: global, pretty stable
    - "Host name lookup": HOSTS.TXT, DNS
  - IP address $\Rightarrow$ MAC address: local, somewhat variable
    - ARP

Host name lookup

- In the beginning...
  - RFC 606: HOSTS.TXT!
  - One line per host
  - HOST : 128.2.194.80 : PIPER.NECTAR.CS.CMU.EDU : INTEL-GATEWAY : NetBSD ::
  - Available by FTP from SRI-NIC.ARPA
    - 10.0.0.51, the first time
  - Good for ~10 years, 1973 – 1983

Problems with HOSTS.TXT

- http://www.textfiles.com/internet/hosts.txt
- Size
  - July 23, 1992
  - Version 1160
  - 22,000 hosts, 1 megabyte
  - Scale ***that*** up!
- Update-frequency problem
  - Annoying to update too frequently
  - Annoying to update too rarely

Domain Name System

- RFC 882 (1983)
- Goals
  - Distributed database
  - Frequent updates
  - Cacheing
  - High availability
    - Map host's name to its address even while it is down
      - “No such host” is very different from “host down”
        - Consider what should happen to e-mail

DNS concepts

- Resource Record (RR) =
  - (Name, class, type, value)
  - PIPER.NECTAR.CS.CMU.EDU IN A 128.2.194.80
  - cs.cmu.edu IN NS CABBAGE.srv.cs.cmu.edu
  - cs.cmu.edu IN NS LETTUCE.srv.cs.cmu.edu
  - cs.cmu.edu IN NS SPINACH.srv.cs.cmu.edu

DNS concepts

- TTL = Time-To-Live
  - How many seconds a record will remain valid
    - (How long you can cache it)
  - Promise about stability of mapping
  - *.CS.CMU.EDU default: 2 days, by local convention
- Query =
  - (Question, flags, query id #)

DNS Concepts

- Response =
  - (Question, flags, query id #) - echoed from query
  - Result (Ok vs. “No such domain”, vs. “I am broken”)
  - Answer record(s)
    - Answer(s) to your question
    - Helpful answers to questions you ***meant*** to ask
      - Q: “Who are the nameservers for CS.CMU.EDU?”
      - A: “CABBAGE.SRV.CS.CMU.EDU”
      - [Q: “What is the IP address of CABBAGE so I can talk to it?”]
      - A: “CABBAGE.SRV IN A 128.2.194.121”

DNS Concepts

- DNS server
  - Knows “all the answers” for a sub-tree
    - Except for sub-sub-trees it ***delegates***
    - Like Unix file system mounts
    - EDU servers delegate CMU.EDU
    - CMU.EDU servers delegate CS.CMU.EDU
- Resolver (library)
  - Gethostbyname(“PIPER.NECTAR.CS.CMU.EDU”)
  - Consults one or more DNS servers
  - Contains retry logic, “marshalling”

DNS Flow

- `gethostbyname("PIPER.NECTAR.CS.CMU.EDU")`
- Resolver contacts D.ROOT-SERVERS.NET
  - EDU IN NS L3.NSTLD.COM (and others)
  - ***By the way, L3.NSTLD.COM IN A 192.41.162.32***
- Resolver contacts L3.NSTLD.COM
  - CMU.EDU IN NS T-NS1.NET.cmu.edu (...)
  - ***By the way, T-NS1.NET.CMU.EDU IN A 128.2.4.14***

DNS Flow

- Resolver contacts T-NS1.NET.cmu.edu
  - CS.CMU.EDU IN NS PEACH.SRV.cs.cmu.edu
  - ***CABBAGE.SRV.CS.CMU.EDU IN A 128.2.194.121***
- Resolver contacts CABBAGE.SRV.CS.CMU.EDU
  - PIPER.NECTAR.CS.CMU.EDU IN A 128.2.194.80
    - TTL = 180,000 (50 hours)

Advanced topics

- Flow for LAPIS.PRT.CS.CMU.EDU?
- How do we handle gethostbyaddr()?
  - Map ***IP address onto name***

Advanced topics

- How do we handle gethostbyaddr()?
  - Map ***IP address onto name***
  - Q: 80.194.2.128.IN-ADDR.ARPA IN PTR
  - A: 80.194.2.128.IN-ADDR.ARPA IN PTR PIPER.NECTAR.CS.CMU.EDU
- IP over DNS

Three names implies two lookups

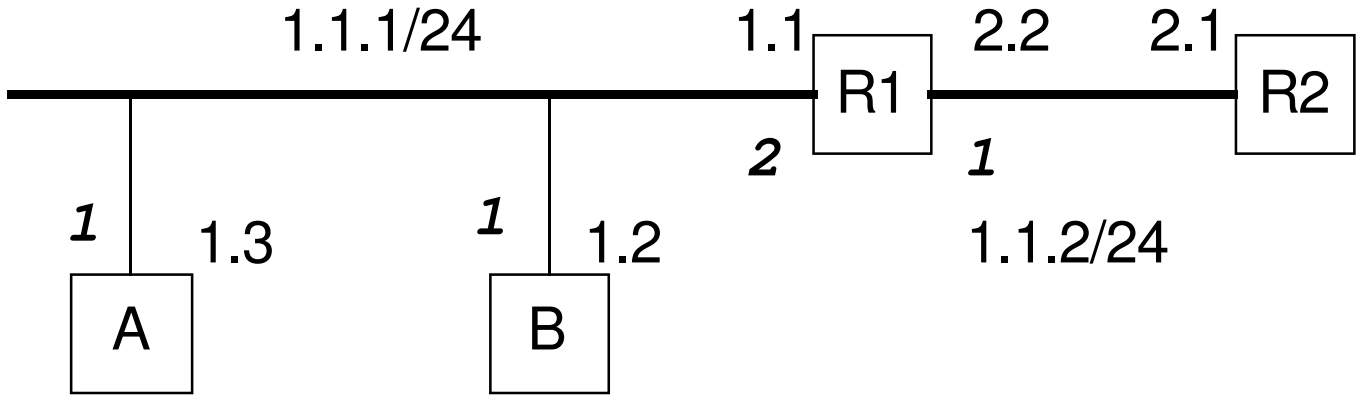
- User specifies host name
- Data packet sent to IP address
- Last-hop router must know MAC address
- Two lookup problems
  - Name $\Rightarrow$ IP address: global, pretty stable
    - "Host name lookup": HOSTS.TXT, DNS
  - IP address $\Rightarrow$ MAC address: local, somewhat variable
    - ARP

ARP design

- Map IP address onto MAC address
- Within a single “network”
  - Broadcast domain, e.g., departmental bridged Ethernet
- MAC addresses have internal structure
  - But it's wrong: manufacturer, serial-number
  - Doesn't help you find IP $\Rightarrow$ MAC mapping

An Example IP Network

dest	gw	link
default	1.1.2.1	1
1.1.1/24	direct	2
1.1.2/24	direct	1



dest	gw	link
default	1.1.1.1	1
1.1.1/24	direct	1

How does A learn B or R1's link layer (MAC) address?

ARP design

- Two solutions
  - Ask a server
    - Why not?
  - ...?

ARP protocol

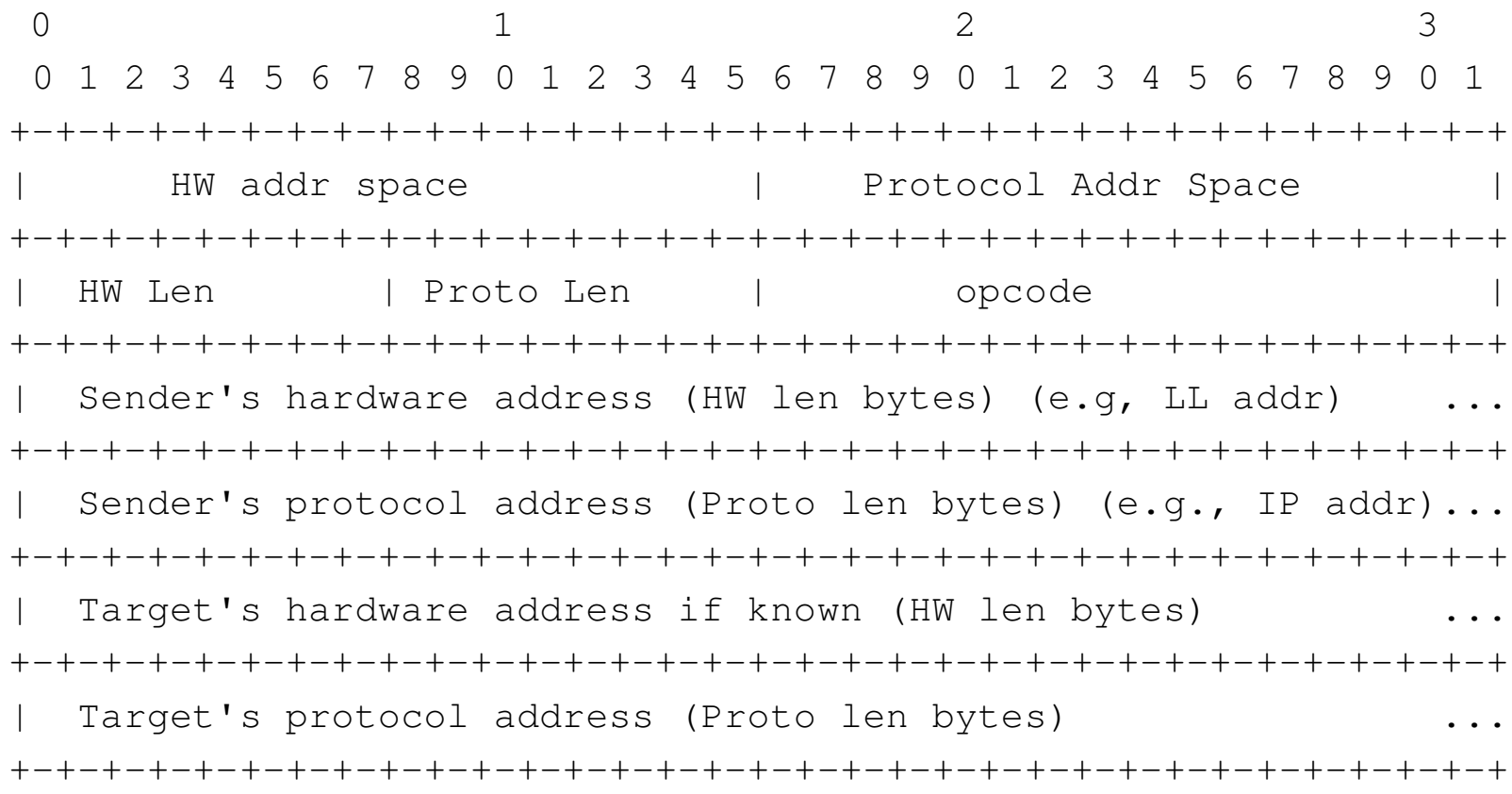
- Ask ***everybody!***
  - That should include asking the right person
- Ethernet supports broadcast
  - Send packet to all stations on “network”
- WHO-HAS 128.2.194.80 TELL 128.2.254.36
  - Broadcast to everybody
- REPLY 128.2.194.80 IS-AT 00-20-AF-D9-FD-CA

Address Resolution Protocol (ARP)

Each node keeps a cache of IP to LL address mappings

Cache is filled by exchanging ***ARP Requests*** and ***ARP Replies***

Defined by RFC 826



Address Resolution Protocol Rules

IP forwarding rules

If gateway field is `direct` next-hop ⇐ IP dest

Otherwise, next-hop ⇐ gateway field

Foreach packet sent

- If next-hop is in ARP cache, send packet using cached MAC addr
- Otherwise, queue packet and send ARP Request on link
- Retransmit ARP Request up to 5 times
- Dump queued packet if no ARP Reply received

Upon receiving any ARP packet

- If sender's IP address is in cache, update cached MAC addr
- If I am not the target IP address, DONE
- If a Request, cache sender's info and send Reply
- If a Reply, cache sender's info and transmit any queued packets

Turning On

- Problems
  - I have a machine with no disk
  - I have a machine with a blank disk
- ...and I want to boot it from a server
- “Easy” answer
  - Download OS (or installer) from some server
- Hard questions
  - Which server? Which file?

Turning On

- Questions, questions
  - Which server? (an IP address)
  - Which file? (maybe server can decide for us)
  - What is my IP address? (so I can send packets)
  - What is the IP address of the next hop to the server?
  - (What is the MAC address of the next hop to the server?)

Turning On

- First approach (Sun Microsystems, 1980's)
  - What is my IP address?
    - RARP (reverse ARP): MAC address $\Rightarrow$ IP address
  - Which server?
    - Whoever answered your RARP request
  - Which file?
    - Filename = my MAC address, download via TFTP
  - What is the next hop to the server?
    - Server must be located on “my network”

Turning On

- Limits to RARP/TFTP approach
  - Server must be located on “my network”
  - TFTP server = RARP server
  - Filename = my MAC address
  - No way to learn “parameters” (netmask, ...)
    - Insertion: SunRPC “bootparam” service
    - Zoo: RARP, TFTP, SunRPC, bootparam – where's the bug?

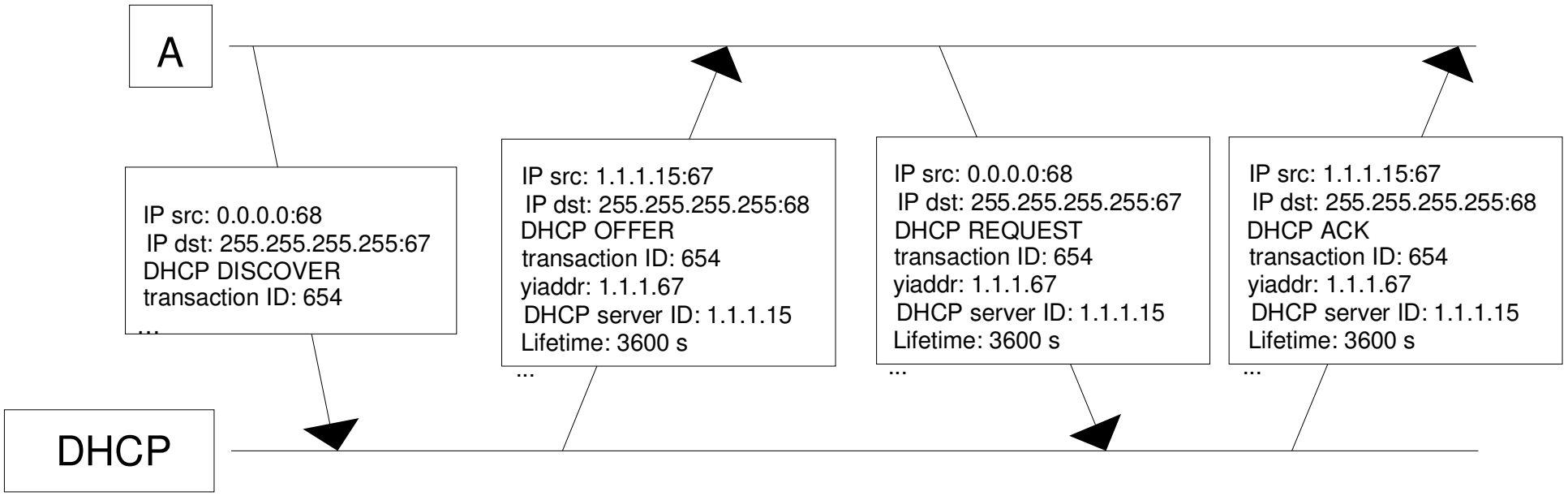
Turning On

- Replacement: DHCP (RFC 2131 and 2132)
  - Use one protocol to determine everything
    - IP address, boot server, boot router, boot filename
  - Useful for hosts who need only some information
    - If you already have an OS installed, don't need boot info
  - Allows temporary allocation of IP addresses
    - Useful for, e.g., wireless hot-spots, temporary visitors
    - “lease time” like DNS TTL

DHCP Transaction Flow

Tricky issue

- How to send IP packets w/o owning an IP address!
  - (After all, we want replies...)



Summary

- Three names for three purposes
- Two mapping protocols
  - Totally different according to function
  - Both “distributed databases”
    - Internet-wide redundant server-trees vs. local broadcast
- Turning on
  - Plan: somebody “nearby” will help
    - Broadcast will reach that “somebody”