# Automatic Naming
# CS 118
# Computer Network Fundamentals
# Peter Reiher

BOOTP
ARP
DHCP

IPv4
IPv6

page 67 - naming pros and cons

# Outline

- What is automatic naming?
- Why automatic?
- Designed-in
- Asking someone else
- Figuring it out for yourself
- Issues

# What is automatic naming?

- Assigning a name to a network entity without human intervention
- Usually very dynamically
- Usually at the moment when it is first needed
- Often using different names for the same thing at different times

# Why automatic?

- "Because it must be!"
- Ease of configuration
- Adapting to changes

# Because it must be!

- Without a name, what can you do?
  - Anonymous reporting (N:1)
  - Broadcast announcements (1:N)

Not all that useful, but…
we can use these to get a name!

# Ease of configuration

- Convenience matters
  - Plug-and-play, Zero-touch, etc.
- Complexity is painful
  - How many devices do you own?
  - Are they all configured the same way?
  - What if you had to configure them explicitly?

# Adapting to changes

moving around changes network names, because the network location has changed

- Mobility
- Renaming

# Mobility

change of physical location

- Change of physical location:
  - Changes network location
    - Topological or geographic names change
    - E.g., USC IP on campus, TimeWarner at home
  - Changes network
    - Name space changes
    - E.g., phone number on 4G,. IP address on WiFi

# Renaming

area code split

change by network operator

renaming - change from wifi to VPN

- Change by the network operator
  - E.g., area code "split"
- Change by the user
  - E.g., off-campus WiFi then VPN to campus

translation - renaming --> moving from one place to another

# How can you get a name?

What are the options?

# Alternatives

- Design-in (preconfigure)
- Pick at random
- Ask someone else

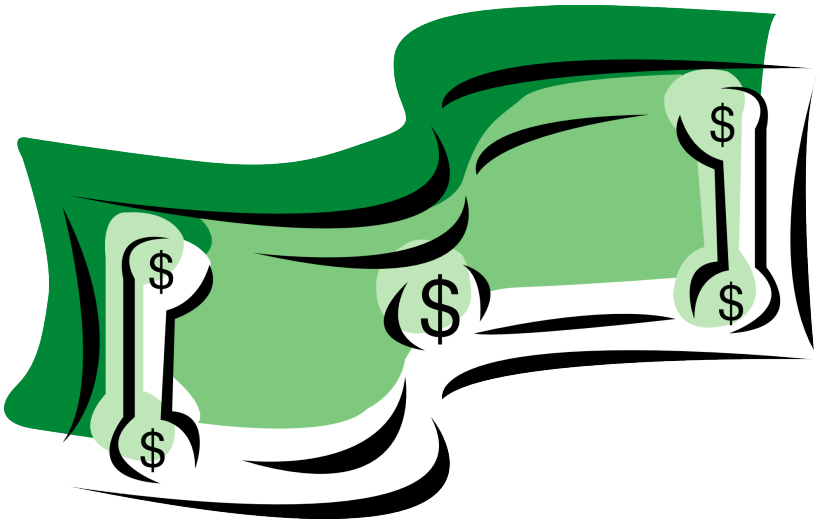
# Designed-in sub-options

- The $1 solution
- Dude, where's my card?
- Getting the boot

# The $1 solution

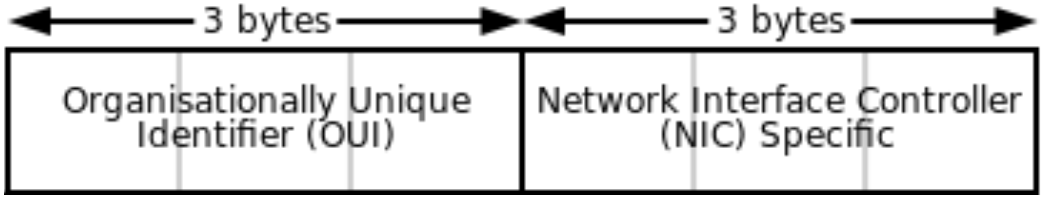
everything is a globally unique name

- Maximum cost of globally unique names
  - Use a USD $1 serial number as your name
  - Put the $1 in the device (or whatever)



# Ethernet

- Two part solution:
  - IEEE assigns OUI
    - Organizationally-Unique Identifier
    - $2,575 per block of 16M addresses ($2^{24}$)
    - $0.0001535 per address (6,515 per $1)
  - OUI assignee manages the block



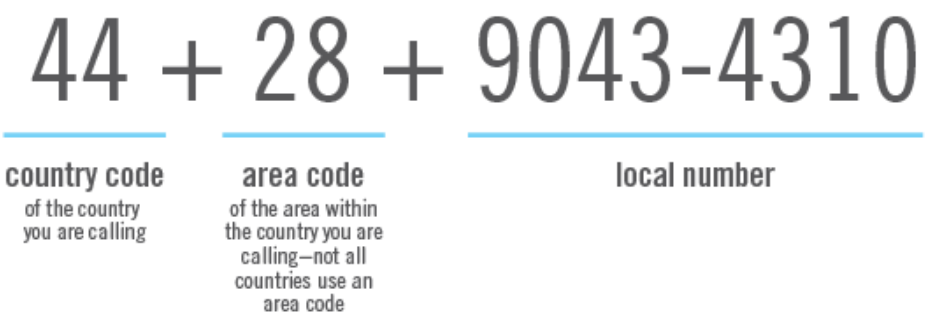
# Ethernet addresses

ethernet address is important to know

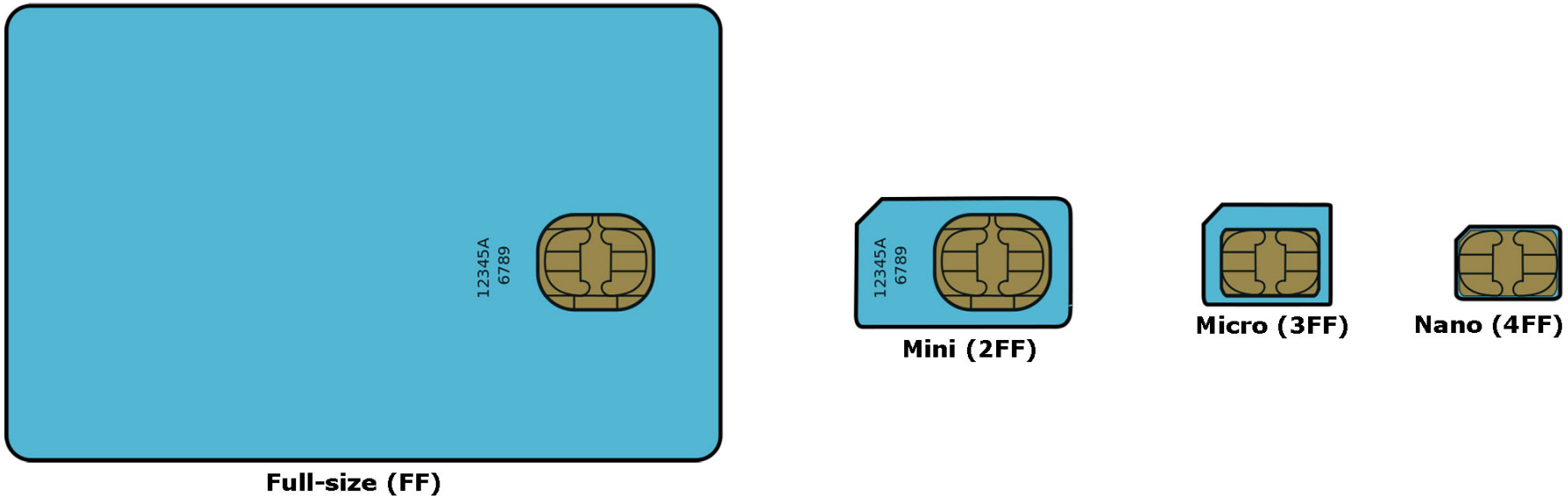
- All Ethernet devices have:
  - Fixed
    - Wired-in or write-only by manufacturer
    - Unique Burned-in (BIA) / hardware (EHA) address
    - Broadcast (all 1's)
  - Writeable
    - To change your BIA (to replace systems)
    - To add multicast addresses

# POTS, non-SIM cellphones

- Assigned by a hierarchy of authorities
  - ITU country codes, country area codes, …
  - POTS – paired to the "tail circuit" (house wire)
  - Non-SIM cell – paired to 7-byte MEID
    (Mobile Equipment ID; 32-bit ESNs ran out in 2008)



# Dude, where's my card?



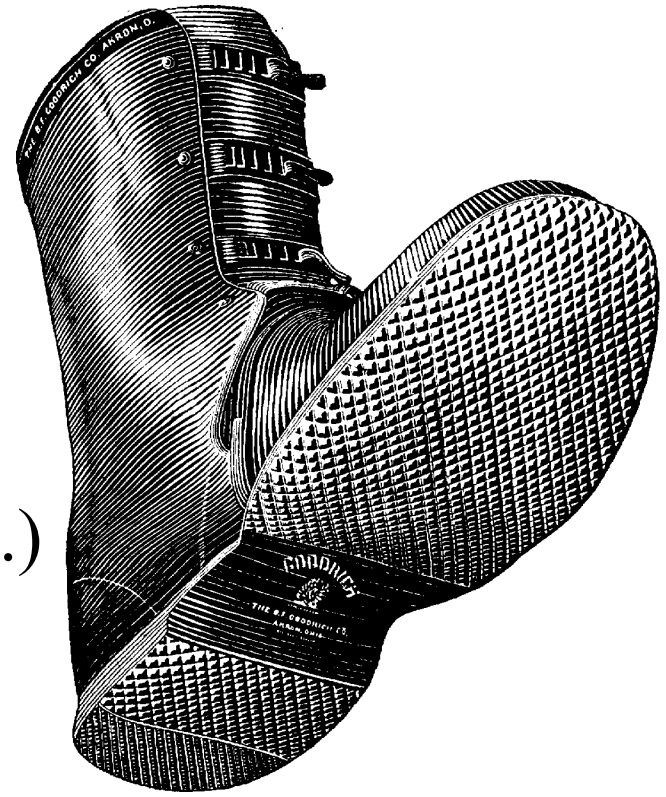
# SIM-based cellphones

- GSM phones have two names
  - The phone (IMEI)
    (International Mobile Equipment ID 14 digits, 6.228 bytes)
  - The SIM card (Subscriber Identity Module)
    - Includes a 20 digit ICCID (IC circuit ID)

- Telco links ICCID to your phone number
  - Also checks your IMEI isn't blacklisted (stolen)

# Getting the boot

- Power-on configuration
  - Files on disk, USB, floppy
  - Flash memory
  - *PROM (EEPROM)
  - Ask the user (let's hope not . . .)



# Figuring it out for yourself

- Pick me a winner!
- Parental support

# Rolling the dice…

- If the number space is large enough
  - Why not just pick one?
  - What could go wrong?

# People names

- Hierarchical in spirit
  - Given name(s) are "random"
  - But are they?
  - What if your last name is common?

| Rank | Male name | Female name |
|---|---|---|
| 1 | Michael | Lisa |
| 2 | John | Mary |
| 3 | David | Susan |
| 4 | James | Karen |
| 5 | Robert | Linda |
| 6 | Mark | Donna |
| 7 | William | Patricia |
| 8 | Richard | Lori |
| 9 | Thomas | Sandra |
| 10 | Jeffrey | Cynthia |
| 11 | Kevin | Kimberly |
| 12 | Scott | Tammy |
| 13 | Joseph | Deborah |
| 14 | Steven | Pamela |
| 15 | Timothy | Brenda |

# IPv4 link local

- 169.254.x.x
  - EXCEPT first 256, last 256 (RFC 3927)
  - Based on MS Automatic Private IP Addressing (APIPA)
  - Pick randomly, do a test to confirm
  - Works only on the local link
    - Where the test works (ARP)
    - NEVER relayed
    - E.g., on your Ethernet

# Pseudo-what?

- Random
  - Having no predictability
  - A sequence with maximum disorder
- Is a single number ever random?
  - No such thing!
  - Random applies to a *sequence*

# Random number generation

- Cannot be generated by a TM in finite time
  - A TM would read only a finite tape
  - TM + finite tape = predictable output

So what do we do?

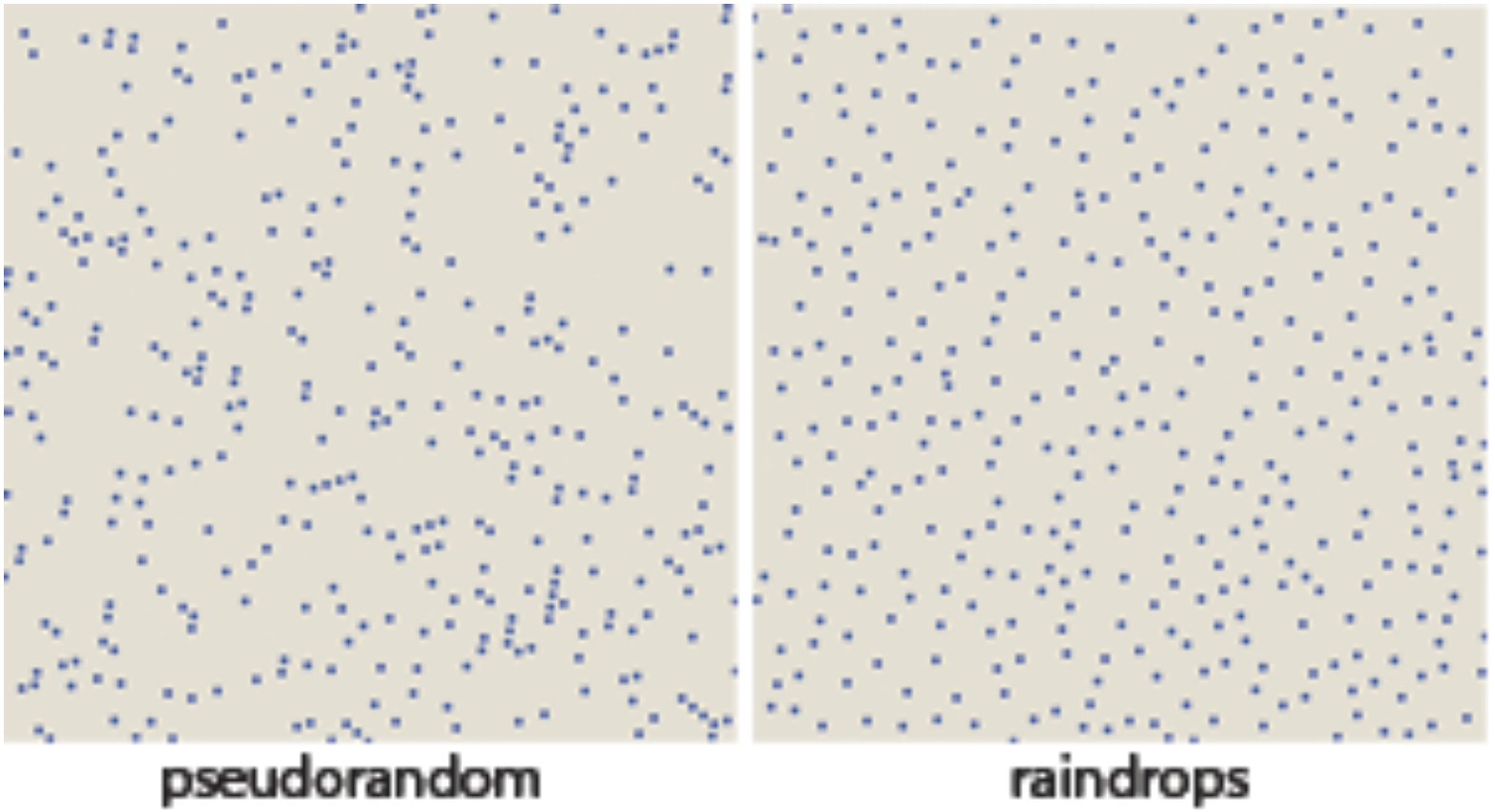
# True random

- Need an external source of infinite entropy
  - A random physical event
  - E.g., radioactive decay, thermal noise, Brownian motion

# Pseudorandom

- Deterministic, but appearing random
  - Unix rand()
  - Sometimes includes arbitrary "seed" (input)
    - Ethernet BIA
    - Disk access times
    - Keystroke delays
    - Mouse movements
  - Repeatable
    - Useful to replay simulations

# “Spot” the difference

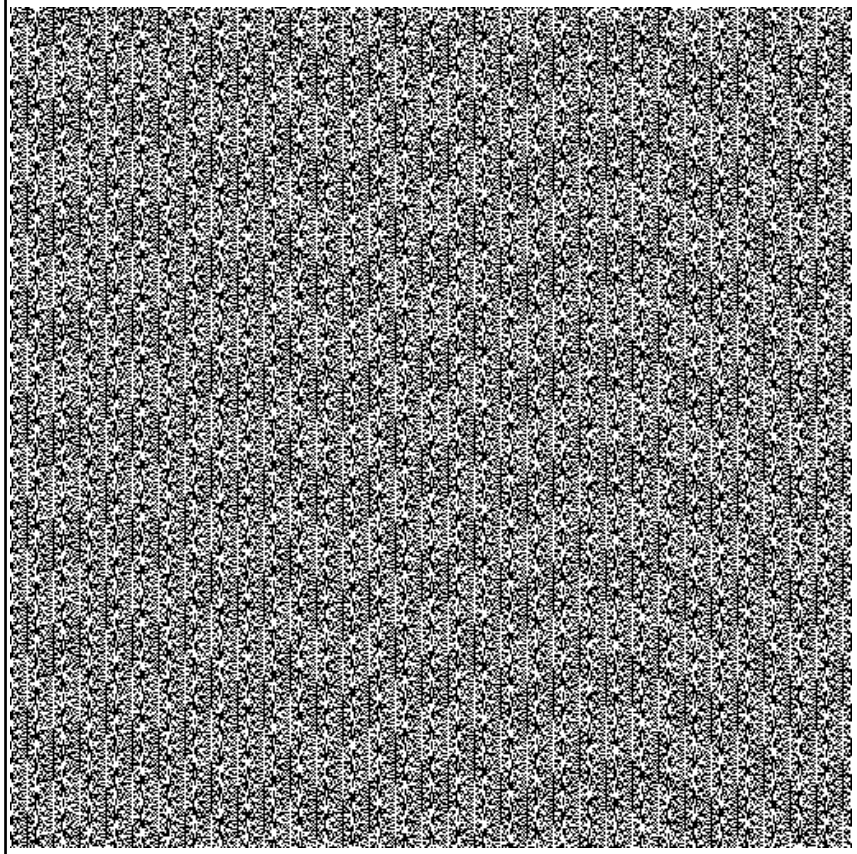


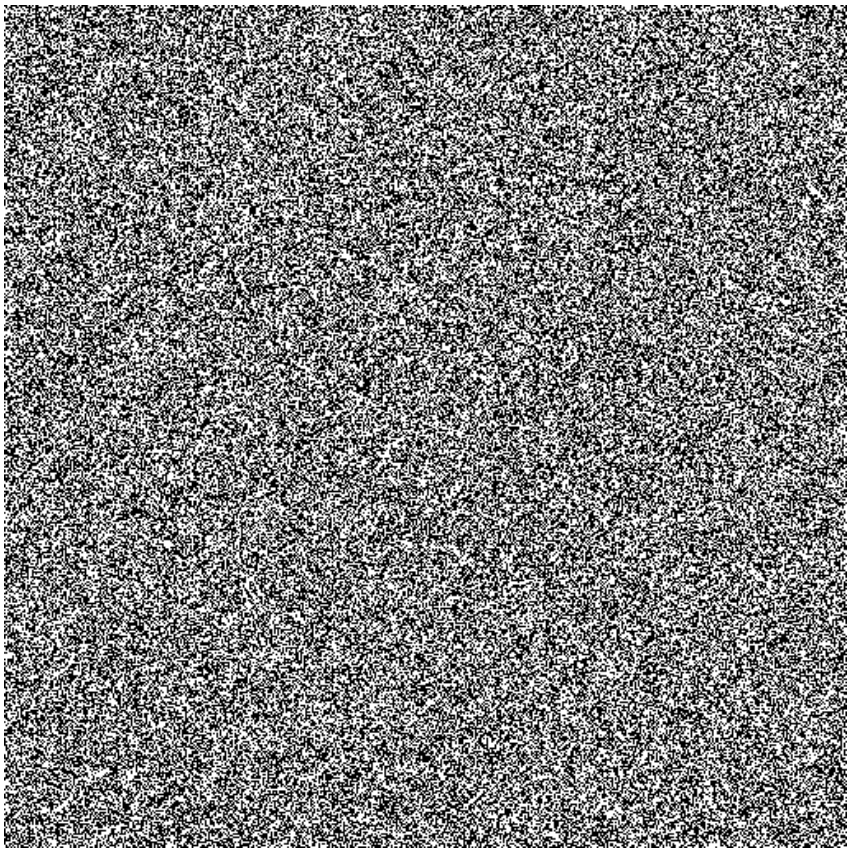

random (not evenly distributed)

# Eyeballs aren't always useful

```
20899862803482534 21
17067982148086513 28
23066470938446095 50
58223172535940812 84
81117450284102701 93
85211055596446229 48
```

# Compute the difference





more random here

# IPv6 link local

- FE80::/10
  - Assign based on MAC address
    or
    Pick randomly (RFC 4193)

    can't change the MAC address of a device
  - Do a test to confirm
  - Works only on the local link
    - Where the test works (ND)
    - NEVER relayed

# iOS Ethernet anonymity

- When configured
  - Every time device wakes from “sleep” (almost never, FWIW)
  - Pick a new random MAC
  - Hope it doesn’t collide (!)
    - There is no test!
  - Avoids “fingerprinting” SSID requests
    - Some stores monitor these

# Asking DAD for help

- Duplicate Address Detection
  - Any general mechanism
  - "DAD" is specific to IPv6
- Works where?
  - IPv4: yes
  - IPv6: yes
  - Ethernet: NO

# IPv4 duplicate detection

- Use ARP

see if something is occupied

  - Send an ARP probe for yourself
    - Source IP = none
    - Destination IP = broadcast
    - Owner MAC = yours (*presumed unique*)
    - Query for = the tested address

could have overwrite existing stuff

if you are testing address, don't send a query from that address

  - Do NOT send a query *from* the tested address
    - It will overwrite the cache of others!
    - Possibly even the existing owner!

# Crossing the streams?

- ARP vs. IP
  - Different layers
  - IP nodes sit on *both*
    - Nodes on shared links
- Are these gateways?
  - Not quite
  - We never translate, only encapsulate (stack)

ARP - protocol that used for checking if address is available
IP - internet protocol

# Implications for IPv4

because it broadcasts --> send to everyone

- IPv4 addressing
  - Ask one network layer for help with another
  - Exchange ARP so IP can autonumber
  - Exchange ARP so IP can discover
  - IP on shared links doesn't exist alone!

- What about non-shared links?
  - Addresses are assigned statically

# IPv6 DAD

IPv6 can't broadcast (send to everyone)

- Use IPv6 Neighbor Solicitation
  - Same basic principle as IPv4
  - Ask to see if anyone has the desired address
  - If nobody asks, we get it

# IPv6 Neighbor Solicitation

another way of ARP probing - broadcast
IPv6 can only do broadcast --> neighbor solicitation

- IP-level replacement for ARP
  - But IPv6 has no broadcast
  - Use multicast instead

find the MAC of the party that would likely contain it

it knows who you are targeting, so it is like 'targeted ARP'

- How?

only send to MAC based on IPv6 addr

  - Could multicast to “all nodes” (like ARP does)
  - Instead multicast to MAC based on IPv6 addr
  - Only the node we want joins that group
  - NOBODY ELSE IS BOTHERED!

# More parental support – IPv6

- Global IPv6 address
  - Listen for a Router Advertisement (or ask routers via Router Solicitation)
- Create an address you know is unique
  - Combine RA information with Ethernet MAC
- Do a test to confirm
  - The test is only on the local link
    - Avoids MAC collisions
  - But the address is good globally
    - RA part is assumed unique

router and ethernet >> unique

# IPv6 example

- Listen for router advertisements
  - Collect them as they come in
- For each RA received on an interface
  - Combine the router prefix with the MAC BIA
  - Also join an IPv6 multicast based on the BIA

# Asking someone else

- A horse with no name
- Name servers for self-namers

# A horse with no name

- Asking a question without an ID

broadcasting >> sending to everyone

- Getting an answer without an ID?

# Asking a question…

- How do you start?
  - If you don't know who to ask, broadcast the question
  - If you do know who to ask, send directly
- What's your address?
  - At the layer you need to know, NONE (typically "0")

# What layer do you ask?

- IPv4
  - Another layer (generally)
- IPv6
  - Your layer (always)

# IPv4

IPv4 go to different layer
network layer no name
go to ethernet layer to get your name
and do it there

- Mixing the layers
  - On a *different* layer that already has an address
    - E.g., broadcast Ethernet ARP with your MAC address
    - E.g., ATMARP request to LANE server on known circuit
- Same layer

another option - go to DHCP server

  - IP (with UDP inside) to DHCP server
    - On the same layer to a server
    - Using source address 0

# IPv6

- IP directly
  - Neighbor Discovery neighbor solicitation
  - Source address = 0

# Getting an answer…

- Broadcast
  - When you didn't know who was asking
- Unicast
  - When you do send to one person
    (e.g., when the request is over a different layer)

# What can someone else tell you?

What are the options now?

# What can someone else tell you?

- Just the facts
  - An address based on a table
- The facts and stuff
  - An address based on a table
  - A file that could have anything
- A loan
  - More specific information
  - Organized by type
  - Loaned out, then recovered for reuse

# Reverse ARP

also over ethernet like ARP

- ARP
  - Broadcasts request providing IP address
  - *Owner* replies with corresponding Ethernet MAC
- RARP

  opposite of ARP
  - Broadcasts request providing Ethernet MAC
  - *Server* replies with corresponding IP address

# RARP limitations

- Only provides an IP address
  - Systems often need more, e.g., default router, DNS server (to avoid bugging the roots), etc.
- Requires preconfigured server
  - Each expected request must match an entry
- Runs on its own protocol
  - Like ARP, this isn't over IP; it's over Ethernet

# BOOTP

runs UDP over IP

- Bootstrap Protocol
  - Still needs a static, preconfigured table
- Replacement for RARP
  - Runs over UDP over IP
    (rather than Ethernet directly)
  - Also provides a file to retrieve
    - That file can be a script, a program, or a table

# DHCP

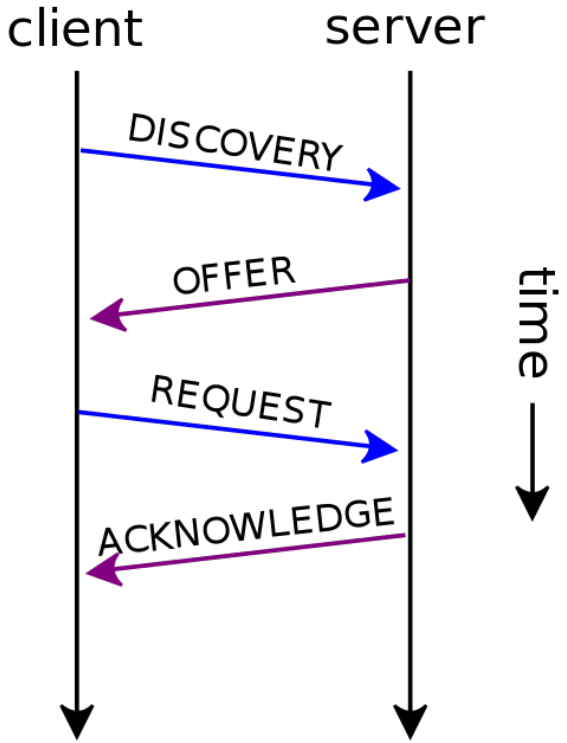
replace BOOTP

- Dynamic Host Configuration Protocol
- Replacement for BOOTP
  - Runs over UDP over IP
  - Explicit way to manage specific configuration parameters
  - Managed via *leases*
    - Assignment has an expiration; can be renewed, released
    - Allows easy reassignment

DHCP loans you a name

# Steps in DHCP

- ARP-like two-phase address assignment
  - Client broadcasts (IPv4) or multicasts (IPv6) a UDP DISCOVER request
  - DHCP servers *all* broadcast/multicast a UDP lease OFFER
  - Client picks one offer and *unicasts* a REQUEST
  - DHCP server *unicasts* a UDP ACK



# Why two phases?

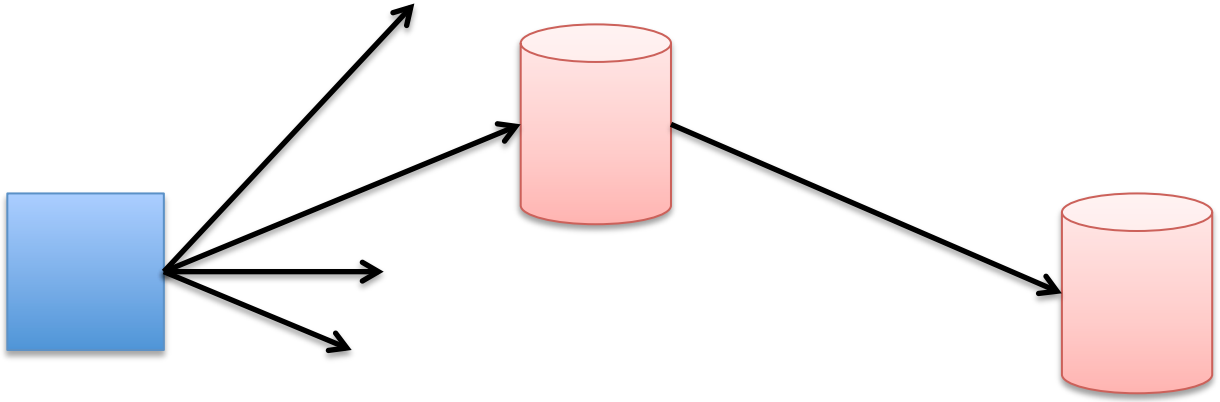
- Multiple servers can make an offer
  - Client picks only one
  - Second phase confirms selection
  - Offers are released after a time if not selected

# B/Mcast vs unicast

- Unicast where possible
  - If you know which DHCP server you want
  - If you've already leased some info
    - E.g., and you go back to get more…

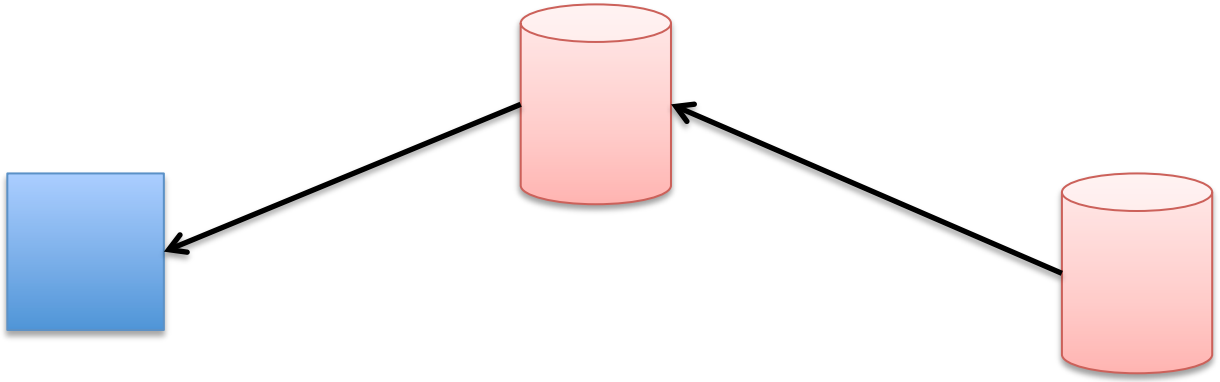
# DHCP relay

- A little like proxy ARP
  - But in both directions



# DHCP relay

- A little like proxy ARP
  - But in both directions



# What can DHCP configure?

- DHCP offer
  - Information critical to configuring the channel
  - IP address
    - dynamic from a range or static based on a table
  - Default router
    - And “netmask” (indicates shared link addresses)
  - Lease time
  - DNS server
    - To avoid root overload
- DHCP inform
  - Other additional context
  - Time server
    - Network Time Protocol
  - Web proxy
    - Address, parameters, etc. for shared caching
  - Just about anything else

# DHCP events

this is important - what DHCP means

- Request
  - Client searching for initial offers
- Offer
  - Servers making initial offers
- Request
  - Client picking one offer
- ACK
  - Server confirming offer
- Renew
  - Client asking for lease extension
- Release
  - Client asking for lease cancellation

# USB

- Master (host), assigns to slaves
  - Assigned each time a device is plugged-in
  - 127 addresses (7 bits, 0=not set yet)

The single master controls “the world”

# Name service for self-namers

- Recall: bind
  - Maps a process to a TCP/UDP port
  - How does another party find that port?
    - It knows the number (IANA list, pre-agreement)
    - It knows the name, but not the number
- Register your name
  - Contact the DNS that has your name:IP map
    - Add the portname:portnum entry too

# Issues

- Telling everyone else
- Configuring DHCP
- Impact to communication in progress

# Telling everyone else…

- How do others know your new name?
  - Esp. if you make one up
- Remember the DNS?
  - Can also map persistent names to changing ones
  - lever.cs.ucla.edu -> IPv4 address that isn't 131.179.192.136
  - IMAP@lever.cs.ucla.edu -> port that isn't 110

# Using the net to find names

- Remember the need for glue?
  - DHCP's "glue" to the client:
    - Router address
      - Even better when it's a "default" router
    - Channel subnet mask
      - What's reachable without contacting the router
    - DNS server
      - A way to get names without needing a default router
      - It needs to be reachable either on the shared channel or via the router indicated

# Configuring DHCP server

- DHCP makes leases
  - Where does it get its land (resources)?
- Currently:
  - Manual configuration
- Experimentally:
  - Another server ("Dynamic DHCP Configuration")

# Pros and cons

- Design-in (preconfigure)
  - Pro: easiest, known to work
  - Con: won't deal with mobility, changes
- Pick at random
  - Pro: second easiest, might work
  - Con: might not (verify?), finding others is hard
- Ask someone else
  - Pro: easy for the client, allows coordination
  - Con: right back where you started for the server!

# Impact on in-progress comm.

- What happens to connections or relays using addresses that change?
  - Continue using the old name
    - How do you know if this is even possible?
  - Shift to the new name
    - What if there isn't one?
    - What if there's more than one?

# Summary

- Giving a name to yourself can be easy
  - Verification is needed
  - Using that name beyond the shared link is harder
- Most naming involves
  - Assumed uniqueness
  - Asking someone else
- Getting started is still manual
  - True “zero configuration” is very rare