

Relating the “logical” with the “physical”

# Domain Name System

*Lawrence Snyder*  
*University of Washington, Seattle*

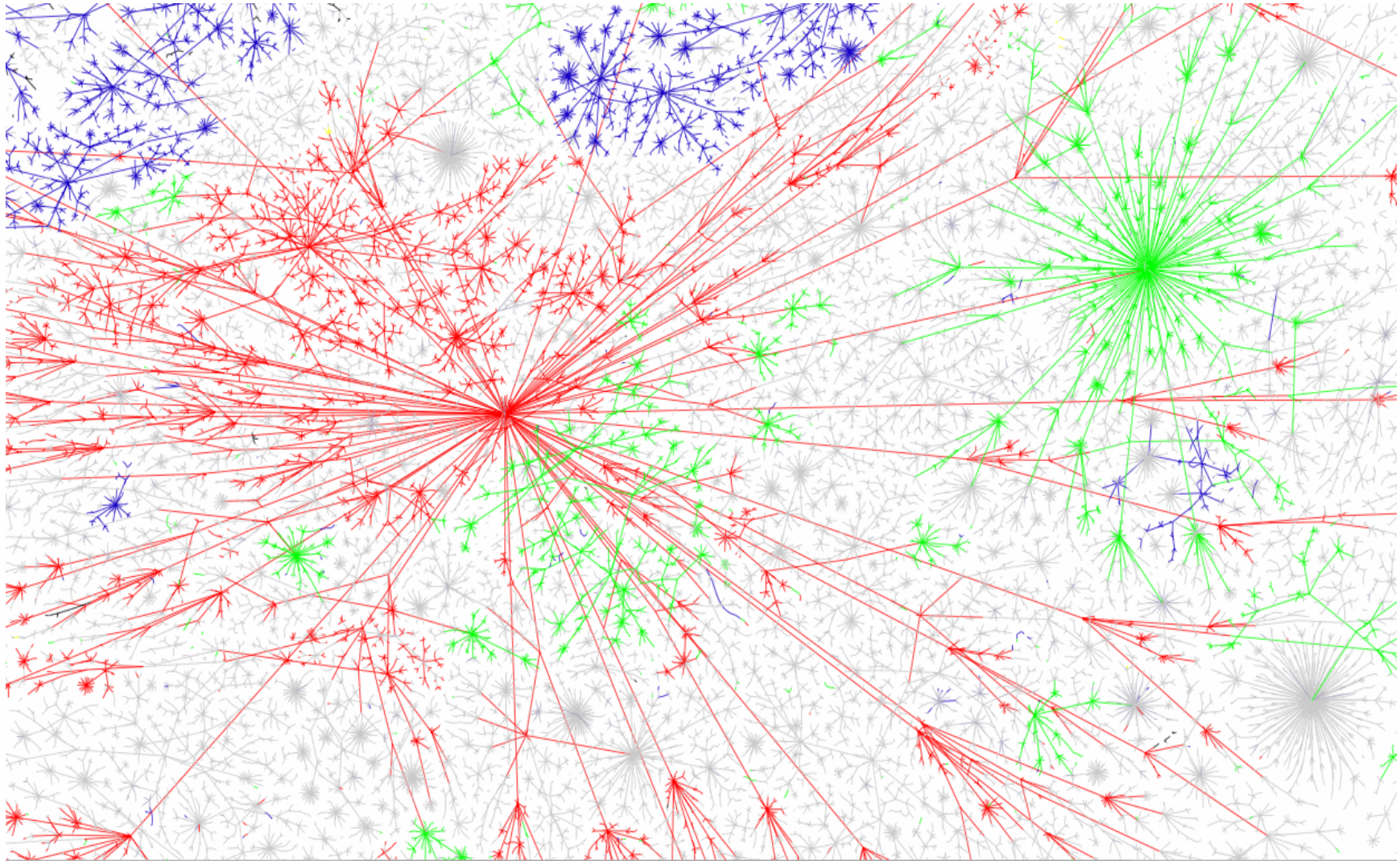
# Recall 2 Ways To Name Computers

- **Logical:** Humans use domain names
  - `spiff.cs.washington.edu`
- **Physical:** Computers use number-quads
  - `128.208.3.136`
- This is different than the phone system:
  - The people use numbers: 1 800 555 1212
  - The equipment uses the same numbers
- A key property of computers: they can separate the logical form (preferred by people) from the physical form they must use

# Today ...

- Today, we explain how the logical/physical separation is implemented for domain names
- But, this is also a chance to illustrate the structure of LARGE systems
  - Study the basic components
  - Study design ideas that make the system work well
  - This matters to you because you' ll probably have “big ideas” about using computers

# Portion of Physical I'net



# What's the Problem?

- The Internet is completely decentralized
  - No one is in charge – ICANN
  - A few companies get permission to give users or organizations IP-addresses – not much logic to it
  - When a person or organization gets an IP-address, it picks a domain name – few rules except to tell the company that gave it out, what the domain is
- Once connected to I'net, users start using domain name ... but when someone refers to it, how does their computer get its number??

Internet Corporation for  
Assigned Names and Numbers



# Recall mail to "friend@cise.ufl.edu"

68.87.205.1	-	Mt Laurel, usa
68.85.240.101	<a href="http://be-70-ar01.burien.wa.seattle.comcast.net">be-70-ar01.burien.wa.seattle.comcast.net</a>	Mt Laurel, usa
68.85.240.69	<a href="http://be-30-ar01.seattle.wa.seattle.comcast.net">be-30-ar01.seattle.wa.seattle.comcast.net</a>	Seattle, WA, USA
68.86.90.213	<a href="http://pos-0-5-0-0-cr01.seattle.wa.ibone.comcast.net">pos-0-5-0-0-cr01.seattle.wa.ibone.comcast.net</a>	Seattle, WA, USA
68.86.85.206	<a href="http://pos-0-8-0-0-cr01.portland.or.ibone.comcast.net">pos-0-8-0-0-cr01.portland.or.ibone.comcast.net</a>	Portland, OR, USA
68.86.85.197	<a href="http://pos-1-15-0-0-cr01.sacramento.ca.ibone.comcast.net">pos-1-15-0-0-cr01.sacramento.ca.ibone.comcast.net</a>	Sacramento, CA, USA
68.86.85.181	<a href="http://pos-0-9-0-0-cr01.sanjose.ca.ibone.comcast.net">pos-0-9-0-0-cr01.sanjose.ca.ibone.comcast.net</a>	San Jose, CA, USA
154.54.11.105	<a href="http://te3-3.mpd01.sjc04.atlas.cogentco.com">te3-3.mpd01.sjc04.atlas.cogentco.com</a>	San Jose, CA, USA
154.54.0.177	<a href="http://te9-1.ccr02.sfo01.atlas.cogentco.com">te9-1.ccr02.sfo01.atlas.cogentco.com</a>	San Francisco, CA, USA
154.54.3.137	<a href="http://te3-8.ccr01.lax01.atlas.cogentco.com">te3-8.ccr01.lax01.atlas.cogentco.com</a>	Los Angeles, CA, USA
154.54.0.226	<a href="http://te3-8.ccr01.iah01.atlas.cogentco.com">te3-8.ccr01.iah01.atlas.cogentco.com</a>	Houston, TX, USA
154.54.24.194	<a href="http://te3-2.ccr01.mia01.atlas.cogentco.com">te3-2.ccr01.mia01.atlas.cogentco.com</a>	Miami, FL, USA
154.54.1.186	<a href="http://te3-3.ccr01.mia03.atlas.cogentco.com">te3-3.ccr01.mia03.atlas.cogentco.com</a>	Miami, FL, USA
38.112.31.66	<a href="http://florida_lambda_rail_llc.demarc.cogentco.com">florida_lambda_rail_llc.demarc.cogentco.com</a>	Washington, DC, USA
198.32.155.10	<a href="http://tpa-flrcore-7609-1-te21-1.net.flrnet.org">tpa-flrcore-7609-1-te21-1.net.flrnet.org</a>	Marina del Rey, usa
198.32.173.161	<a href="http://tlh-flrcore-7609-1-te41-1907.net.flrnet.org">tlh-flrcore-7609-1-te41-1907.net.flrnet.org</a>	Marina del Rey, usa
198.32.173.162	<a href="http://ctx36-ewan-msfc-1-v1907-1.ns.ufl.edu">ctx36-ewan-msfc-1-v1907-1.ns.ufl.edu</a>	Marina del Rey, usa
128.227.236.85	<a href="http://ctx36-nexus-msfc-1-v50-1.ns.ufl.edu">ctx36-nexus-msfc-1-v50-1.ns.ufl.edu</a>	Gainesville, FL, USA
128.227.236.14	<a href="http://csev1-core-msfc-1-v41-1.ns.ufl.edu">csev1-core-msfc-1-v41-1.ns.ufl.edu</a>	Gainesville, FL, USA
128.227.254.74	-	Gainesville, FL, USA
128.227.205.2	<a href="http://cise.ufl.edu">cise.ufl.edu</a>	Gainesville, FL, USA

- A packet sent to 128.227.205.2 finds its way

# But, how do we get 128.227.205.2?

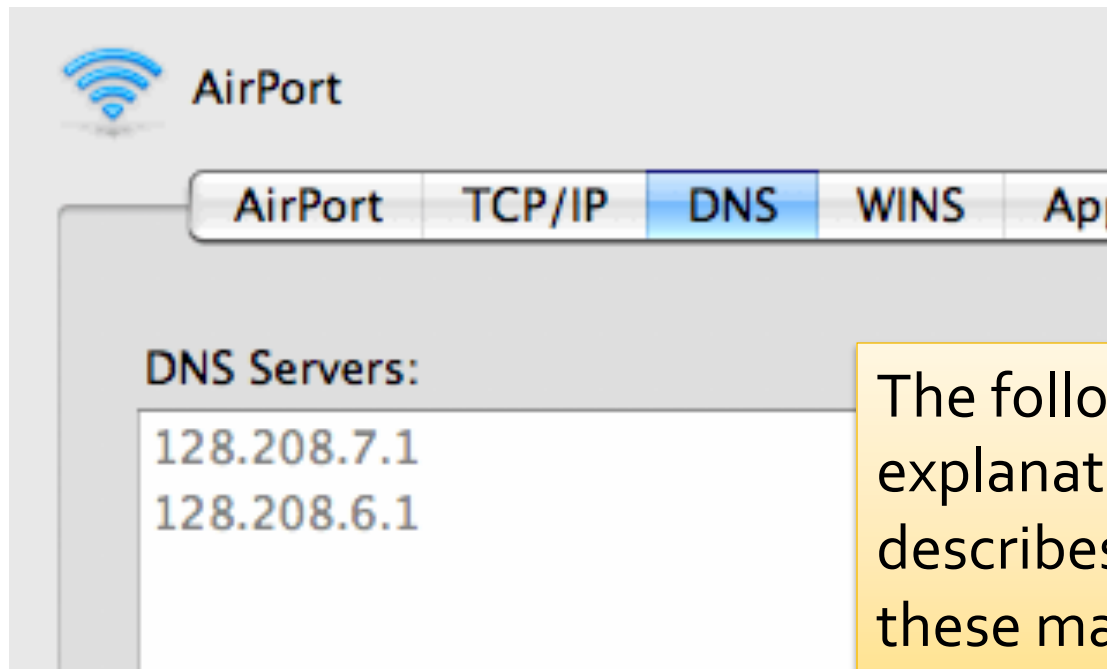
- When we send mail to a friend at the U of FL, we type friend@cise.ufl.edu and the computer that sends mail for us on campus needs to find out this fact:

cise.ufl.edu == 128.227.205.2

- We said it asks the Domain Name System, or DNS ... so what happens

# But Wait!

- How does it know the address of the DNS?
- You (or someone or something who set up your computer) told it when connecting it to the network ... look in net control panel



The following explanation describes what these machines do



# First Step

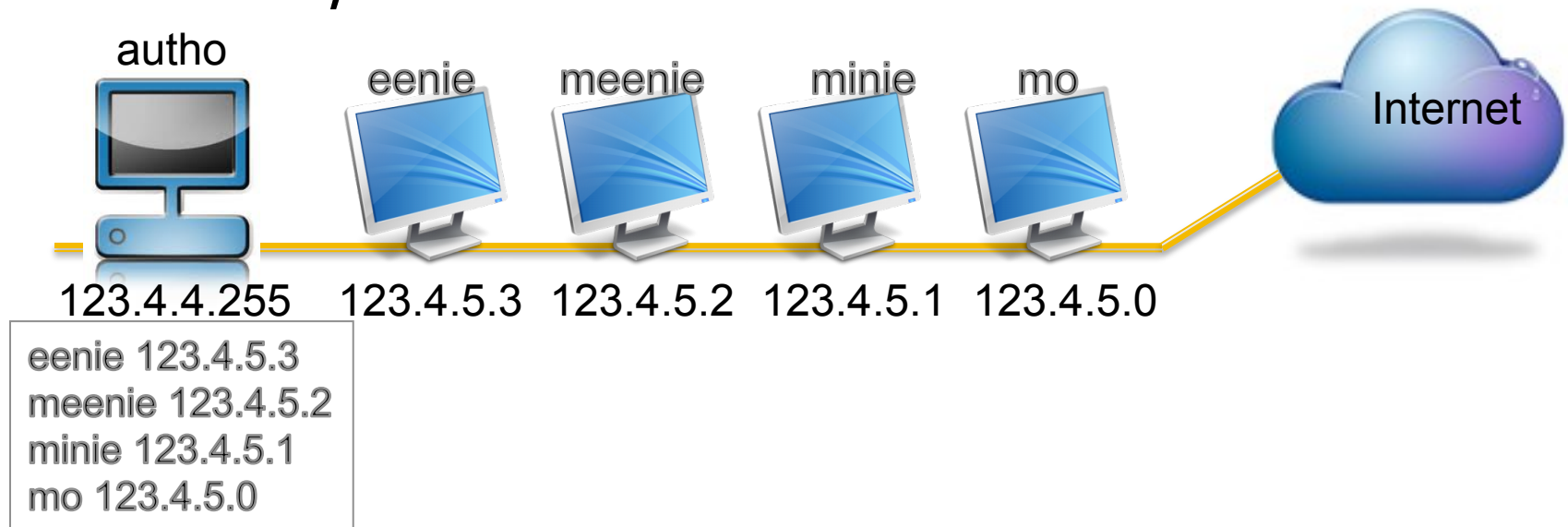
- The DNS server answers the question “what number is **cise.ufl.edu**?” by this method
- First Step: Look it up in its “address book”
  - The DNS server does that
  - It keeps its own address book, a list of all of the domain names like **cise.ufl.edu** that it has been asked about and found
  - We say it *caches* the addresses it’s found
    - *caching* – keeping a copy around in case its needed again
  - It checks the cache first

# If It Has Never Been Asked ...

- The address will not be in the cache if this is the first request
- Second Step: The DNS server begins a process of finding the address on behalf of your computer ...
  - That process uses 2 Facts of I'net

# The DNS Design: Fact 1

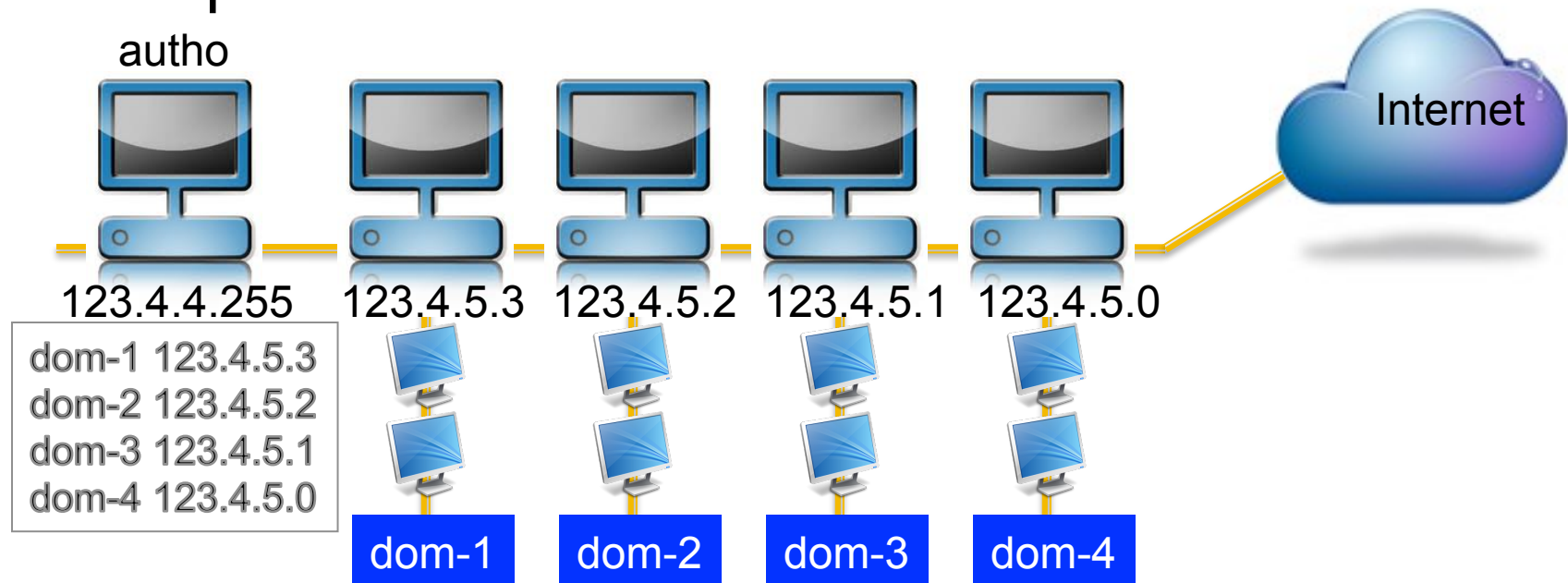
- Every domain has an authoritative name server, which I'll call autho



- Two Cases: Autho knows the number of every computer in its domain

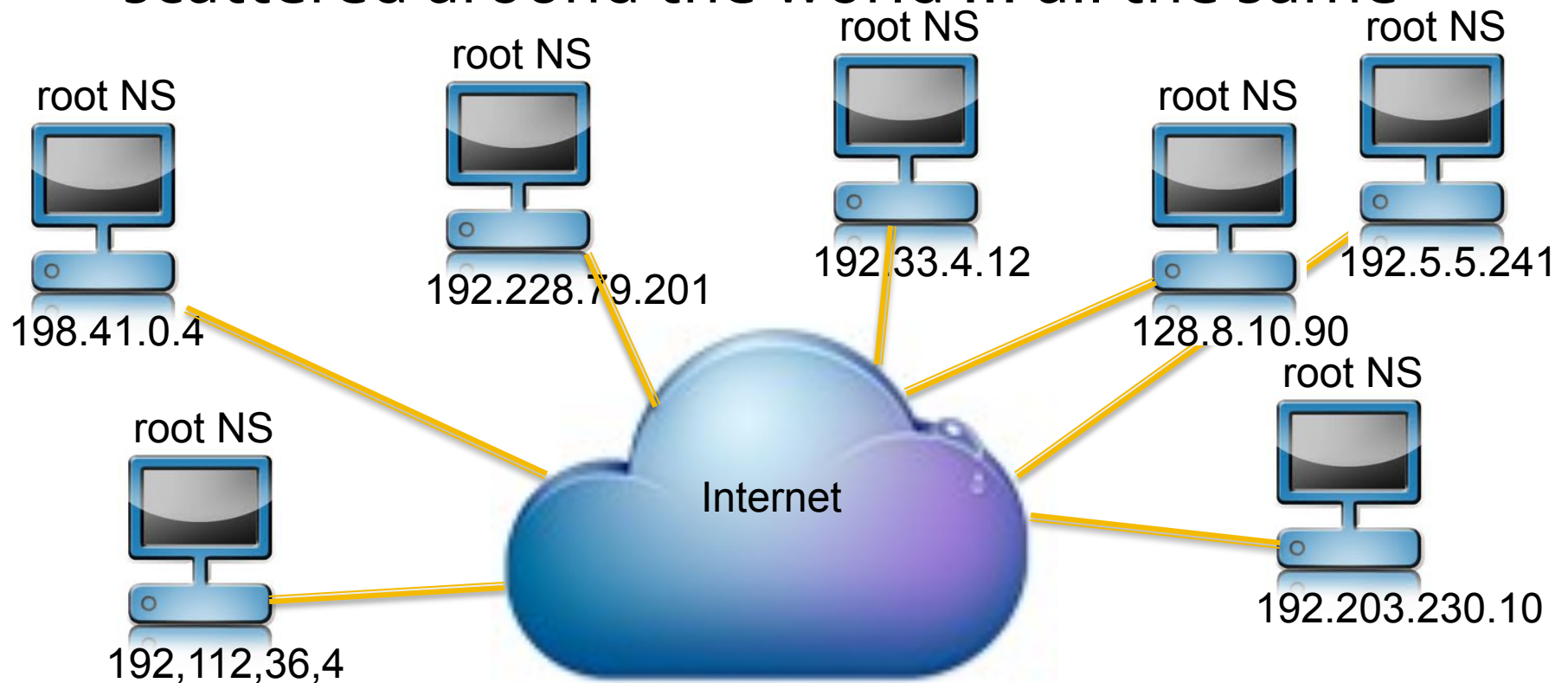
# The DNS Design: Fact 1 (Continued)

- OR Autho knows the number of every autho computer in its domain



# The DNS Design: Fact 2

- There are 13 Internet “root name servers” scattered around the world ... all the same



- All DNS servers have their numbers

# www.root-servers.org





# "J" Root Name Server

J	VeriSign, Inc.	<b>Sites: 70</b> Global: 63 Local: 5  <b>Dulles, VA, US (2 sites); Dulles, VA, US (1 sites); Ashburn, VA, US *; Miami, FL, US; Atlanta, GA, US; Seattle, WA, US; Chicago, IL, US; New York, NY, US *; Honolulu, HI, US; Mountain View, CA, US (1 sites); Mountain View, CA, US (1 sites); San Francisco, CA, US (2 sites) *; Dallas, TX, US; Amsterdam, NL; London, UK; Stockholm, SE (2 sites); Tokyo, JP; Seoul, KR; Beijing, CN; Singapore, SG; Dublin, IE; Kaunas, LT; Nairobi, KE; Montreal, CA; Perth, AU; Sydney, AU; Cairo, EG; Cairo, EG; Warsaw, PL (2 sites); Brasilia, BR; Sao Paulo, BR; Sofia, BG; Prague, CZ; Johannesburg, ZA; Toronto, CA; Buenos Aires, AR; Madrid, ES; Fribourg, CH; Hong Kong, HK (2 sites); Turin, IT; Mumbai, IN; Oslo, NO; Brussels, BE; Paris, FR (2 sites); Helsinki, FI; Frankfurt, DE *; Riga, LV; Milan, IT; Rome, IT; Lisbon, PT; San Juan, PR; Edinburgh, UK; Tallin, EE; Taipei, TW; New York, NY, US *; Palo Alto, CA, US *; Anchorage, US; Moscow, RU; Manila, PH; Kuala Lumpur, MY; Luxembourg City, LU; Guam, GU, US; Vancouver, CA; Wellington, NZ</b>	IPv4: 192.58.128.30 IPv6: 2001:503:C27::2:30
---	----------------	--	--

# So, Here's How It Goes ...

- Your computer's DNS server never heard of `cise.ufl.edu.root` ... so it pulls the domain name apart:
  - `cise`, a computer in the `.ufl` domain
  - `ufl`, a domain in the `.edu` domain
  - `edu`, a domain in the `.root` domain
- So, the DNS begins at the end and starts asking for the numbers of the autho computers ... who's the autho for the `.root` domain?

Implied – on all addresses,  
so don't bother with it

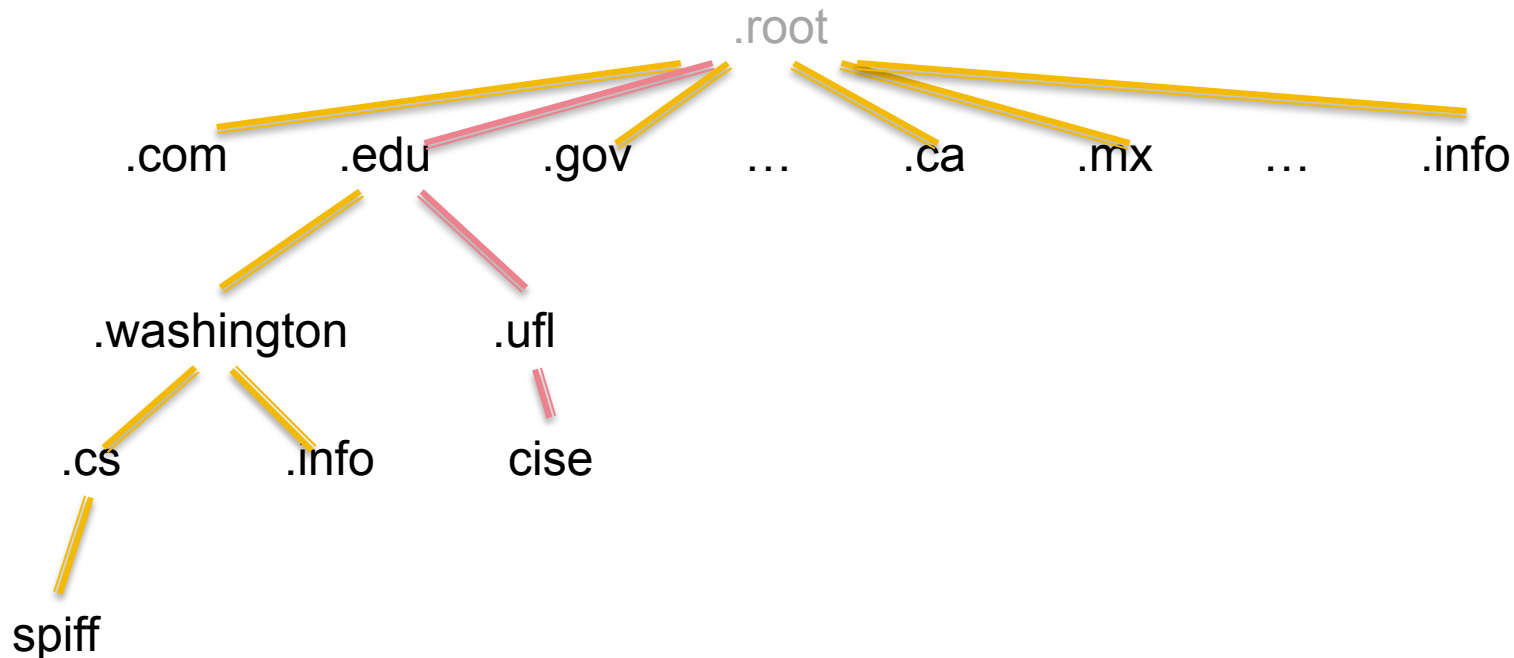


# Your DNS Asks the .root NS

- Please give me the number of **.edu** autho
  - Getting that it asks it, ...
- Please give me the number of **.ufl** autho
  - Getting that it asks it, ...
- Please give me the number of the **cise** machine
  - Getting **128.227.205.2**, it addresses your email and sends it on
- Simplification: it might have cached **.edu** autho and **.ufl** autho, which saves those requests

# Logical Names Form Hierarchy

- As a hierarchy, it can be shown as a tree:

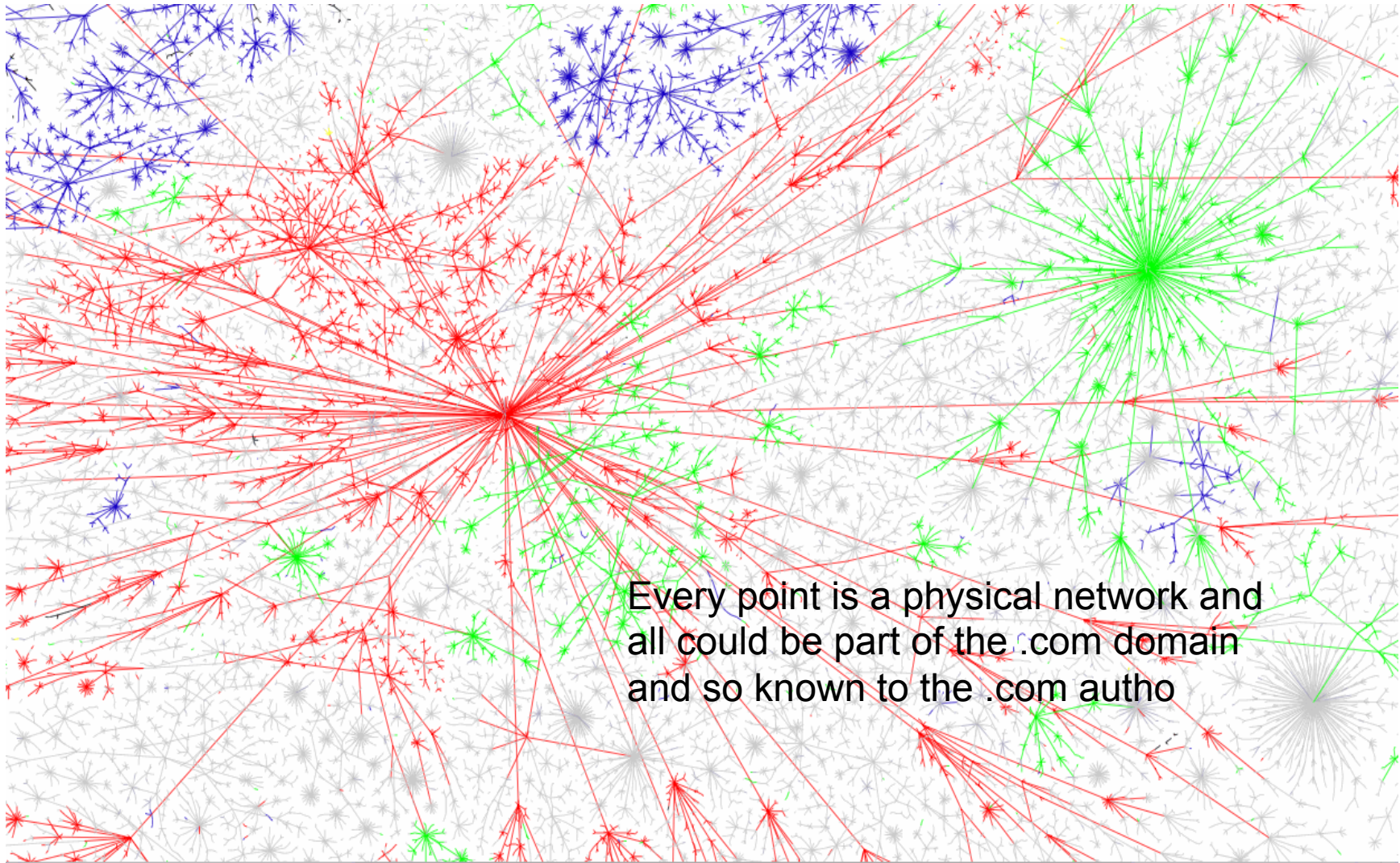


- The DNS is simply “walking” down the tree asking each autho for the number of next item

# Exercise:

- I was in NYC last week working at a hotel and went to log into my computer at UW
  - [spiff.cs.washington.edu](http://spiff.cs.washington.edu)
- How did the hotel's ISP find **128.208.3.136**?

# Think About This Scheme: Huge



Every point is a physical network and all could be part of the .com domain and so known to the .com autho



# Suppose A Domain Adds Computer

- When a domain, say `.ufl`, adds a new computer it gets a name and an IP-address
- They add its name and number to the list in ufl autho's memory and its up and running, "known to the world"
- This is a completely decentralized solution – no one needs to be in charge except to make sure that the domain autho is up & correct

# Properties ...

- **Fault tolerant:** when a hurricane takes out Miami's power, only the domains without power are affected ...
- **Robust:** when a fire burns down the building of a .root name server, 12 others can carry the load
- **Enormous capacity:** most lookups are independent and do not collide (b/c higher level domain authos are cached), but more capacity is possible by replicating authos

# Compare DNS Structure To ..

- Master List Solution ...
  - Suppose the design was for the root NS computers to have a master list of all  
domain\_name: IP-address  
pairs connected to the Internet
- How would it be different, better or worse?