Announcements

- Tuesday’s Lab is not in the lab ... go to CSE403
  - Richard Ladner speaks on accessibility ... this is course content
- Due Date for Pairs Programming ... Wednesday at lecture time
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, 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
- Once connected to I’net, users start using domain name ... but when someone refers to it, how does their computer get its number??
Recall mail to “friend@cise.ufl.edu”

- 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
First Step

- The DNS server answers the question “what number is cise.ufl.edu?” by this method
- First Step: Look it up in the “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 need it 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
Every domain has an authoritative name server, which I’ll call autho.

Two Cases: Autho knows the number of every computer in its domain.
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
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?
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
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 Miami last week working at a hotel and went to log into my computer at UW
  - spiff.cs.washington.edu
- How did the hotel’s ISP find \texttt{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
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?