Outline

- Hints on how to read systems papers
- Network Applications
  - Naming
  - Applications
    - Central Server
    - Hierarchical
    - Peer-to-peer

How to Read a Research Paper

- Typical paper
  - Abstract
  - Introduction
  - Motivation, problem description
  - Research questions that are being addressed by this paper
  - Experiment Setup
  - Results
  - Conclusions and Future work

Why do you read a paper?

- Understand and learn new contributions
- However
  - Not all papers are “good”
  - Not all papers are “interesting”
  - Not all papers are “worthwhile” for you
- You have to learn to identify a good paper and spend your time wisely
  1. Breadth
  2. Depth
  3. React

How to Read a Research Paper

- Ask yourself, what is this paper about? (breadth)
  - Read the title and the abstract

    If you still don’t know what this paper is about, then this is a bad paper.
  - Read the conclusion

    Are you now sure you know what this paper is about? If not it is a BAD paper. We will try not to read such papers in this course
- Read the introduction
- Read the section headings
- Read tables and graphs and captions. See what they say

How to read a paper (cont)

- See who wrote it, where it was published, when was it written (credibility)
- Skim bibliography to see if the authors are aware of relevant related work. See if you know the relevant work. See if you know a relevant work that they didn’t refer

How to read a paper - depth

- Approach with scientific skepticism
- Examine the assumptions
  - Do their results rely on any assumptions about trends in environments?
  - Are these assumptions reasonable?
    - E.g. “Lets assume that there are billions of powerful computers, connected by a high speed network, spread across the world, our system will …”
    - E.g. “Our system can enable you to run Windows 98 on a 33Mhz Intel 386 with 640K main memory”

How to read a paper - depth

- Examine the methods:
  - Did they measure what they claim?
  - Can they explain what they observed?
    - It is easy to dump your experimental results on the paper. As a reader you want an analysis of why the system behaves a certain way, not the raw data.
  - Did they have adequate controls
  - Were tests carried out in a standard way? Were the performance metrics standard? If not, do they explain their metrics clearly?

How to read a paper - depth

- Examine the statistics: (there is truth, lies and then there is statistics!!)
  - Were appropriate statistical tests applied properly?
  - Did they do proper error analysis?
  - Are the results statistically significant?
    - Common mistake: "We performed our experiment once at 4 am and noticed a ten fold improvement. Thus we conclude that our system is better"
  - Be very careful with percentages
    - Method A: 0.01 seconds, our Method: 0.005 seconds
    - Our method shows 100% improvement over method A!!

How to read a paper - depth

- Examine the conclusions:
  - Do the conclusions follow logically from the conclusions
    - We performed our experiments with 8 palm pilots and saw a 10 fold improvement. Hence we conclude that our system will scale to millions of palm pilots
  - What other explanations are there for the observed effects
  - What other conclusions or correlations are there in the data that they did not point out
    - Earlier work performed experiments using a 2 Mbit wireless network. Our system (incidentally) used a 11 Mbit network and saw a 5 fold improvement. So our technique works!!

How to read a paper - react

- Take notes
- Highlight major points
- React to the points in the paper
  - Place this work with your own experience
  - If you doubt a statement, note your objection
  - If you find a pleasing quotation, write it down
- Construct your own example
- Summarize what you read
- Maintain your own bibliography of all papers that you ever read

Sample bibliography - bibtex

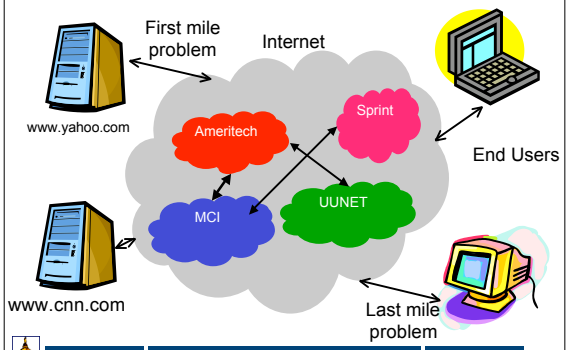
```
@Book{stevens98,
        author = {W. Richard Stevens},
        title = {UNIX Network Programming:
                Networking APIs: Sockets and XTI},
      publisher = {Prentice Hall},
      year = 1998,
      volume = 1,
      series = {ISBN 0-13-490012-X},
      note = {Sample code from this book is available at
  \url{http://www.kohala.com/start/unpv12e.html}},
      edition = 2,
}
```

…. You can refer to the Computer Network books by W. Richard Stevens \cite{stevens98} for sample ….

How to Write a Research Paper

- Write it such that anyone who reads it using the method we discussed understands your ideas.
- Clearly explain what problem you are solving, why it is interesting and how your solution solves this interesting problem
- Be crisp. Explain what your contributions are, what your ideas are and what are others' ideas

Internet



Performance bottlenecks

- First mile problem: Server to the Internet
  - Everyone wants to access one popular service (slashdot effect)
- Last mile problem: End user to the Internet
  - Broadband (cable, DSL), T1, T3, dialup, 2G cellular (slow)
- Peering problem:
  - Data goes through multiple networks and service providers at peering points
- Backbone problem:
  - The information highway for data traffic
- Internet traffic is growing at 260%!! Reportedly carries 2,000 TBytes of data per day (2001)

Applications

- Colloborative: Email, IM, USENET
- Sharing: Gnutella, Kazaa, ….
- Multimedia: MBONE, Movielink.com, launch.yahoo.com etc.
- Voice over IP (VOIP) to integrate voice and IP data networks

Networked distributed system architectures

Central Server based

Web servers

Hierarchical Services

Domain Name System – DNS

Peer-to-Peer Systems

Napster, gnutella

Central Server based

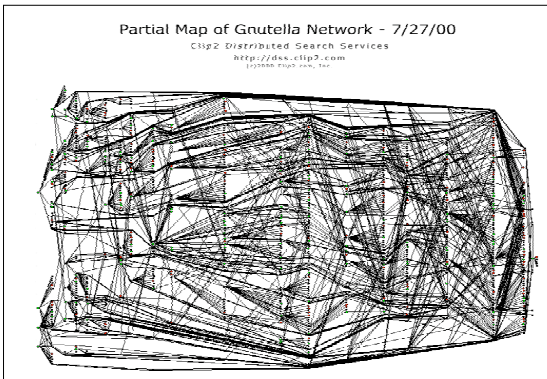
- A central server provides service
  - Reliability and fault tolerance
    - If server shuts down, then no service
  - Scalability
    - Performance bottle neck
    - E.g. if everyone accesses Microsoft.com from the east coast (new release of web browser), accesses to Yahoo.com in California might be slow because we share the same link from east coast till Utah (say)
  - Easy to deploy, administer

Peer-to-peer systems

- Decentralized, no "server"
- Robust – no single point of failure
- "Will perform work for others since they will work for us" computing
- Can scale up
- Locating resources harder
- E.g. napster (has a central directory server)
  gnutella

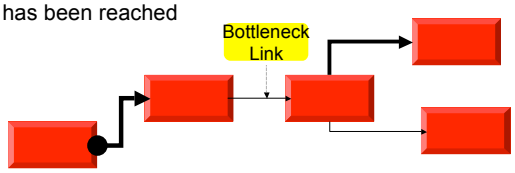
Gnutella

- Queries issued by a servant at a given node propagate out to neighbor nodes
- The neighbors propage the query to their neighbors, and so on, for a given number of hops.
- Depending on where a user's query is first issued, it may or may not reach a node that has the file sought by the user.



Scalability

- The scalability of a Gnutella network to accommodate more users performing more searches is limited by the lowest bandwidth links prevalent within the network
- For dial-up users it is 10 requests per second and has been reached



Applications - Naming

- What do names do?
  - Help identify objects
  - help locate objects
  - define membership in a group
  - specify a role
  - convey knowledge of a secret
- Name space
  - defines set of possible names
  - consists of a set of name to value bindings

Name properties

- Names versus addresses
- Location transparent versus location-dependent
- Flat versus hierarchical
  - /afs/nd.edu/user37/surendar/file.txt
- Global versus local
  - Wizard.cse.nd.edu vs darwin
- Absolute versus relative
  - /afs/nd.edu/user37/surendar/file.txt vs ../../file.txt

Examples

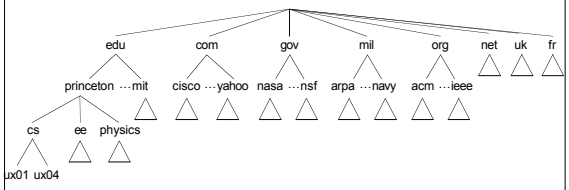
- Hosts
  - Wizard.cse.nd.edu or 129.74.25.101 (IP address) or 00:03:ba:16:c5:7a (ethernet address)
- Peers for your HWP1
  - Name → host:port
- Files
  - /usr/llp/tmp/foo → (server, fileid)
- Users
  - Larry Peterson llp@cs.princeton.edu

Domain Name Service (DNS)

- Provides Internet domain name to IP address translation
  - Domain name translation (nd.edu)
  - Hostname translation (wizard.cse.nd.edu)
  - Service location (MX records, mail service for ND)

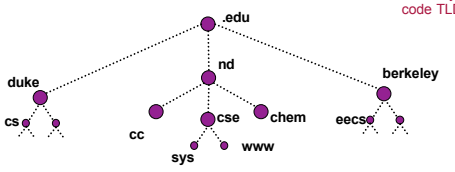
```
$ nslookup –query=mx aol.com
aol.com preference = 15, mail exchanger = mailin-01.mx.aol.com
aol.com preference = 15, mail exchanger = mailin-02.mx.aol.com
aol.com preference = 15, mail exchanger = mailin-03.mx.aol.com
aol.com preference = 15, mail exchanger = mailin-04.mx.aol.com

Authoritative answers can be found from:
aol.com nameserver = dns-07.ns.aol.com
aol.com nameserver = dns-01.ns.aol.com
aol.com nameserver = dns-02.ns.aol.com
aol.com nameserver = dns-06.ns.aol.com
mailin-01.mx.aol.com    internet address = 64.12.136.57
mailin-01.mx.aol.com    internet address = 64.12.137.89
mailin-01.mx.aol.com    internet address = 64.12.137.184
……
```


Domain Naming System Hierarchy



DNS hierarchy

DNS name space is *hierarchical*:
- fully qualified names are "little endian"
- scalability
- decentralized administration
- domains are naming *contexts*



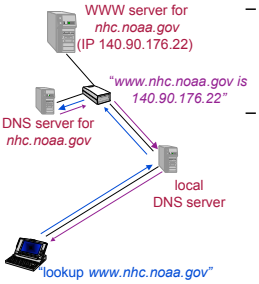
Source: Jeff Chase

Resource Records

- Each name server maintains a collection of resource records
  - (Name, Value, Type, Class, TTL)
- Name/Value: not necessarily host names to IP addresses
- Type
  - NS: Value gives domain name for host running name server that knows how to resolve names within specified domain.
  - CNAME: Value gives canonical name for particle host; used to define aliases.
  - MX: Value gives domain name for host running mail server that accepts messages for specified domain.
- Class: allow other entities to define types
- TTL: how long the resource record is valid

DNS Protocol

- UDP-based client/server
  - client-side *resolvers*
    - typically in a library
    - *gethostbyname*, *gethostbyaddr*
  - cooperating servers
    - query-answer-referral model
    - forward queries among servers
    - server-to-server may use TCP ("zone transfers")



Source: Jeff Chase

DNS Name Server Hierarchy

DNS servers are organized into a hierarchy that mirrors the name space.

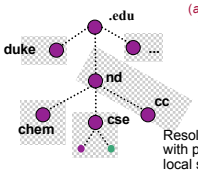
Specific servers are designated as *authoritative* for portions of the name space.

Servers may delegate management of *subdomains* to child name servers.

Parents refer subdomain queries to their children.

Root servers list servers for every TLD.

Subdomains correspond to organizational (*admininstrative*) boundaries, which are not necessarily geographical.

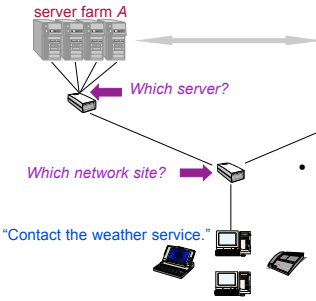


Servers are bootstrapped with pointers to selected peer and parent servers.

Resolvers are bootstrapped with pointers to one or more local servers; they issue *recursive* queries.

Source: Jeff Chase

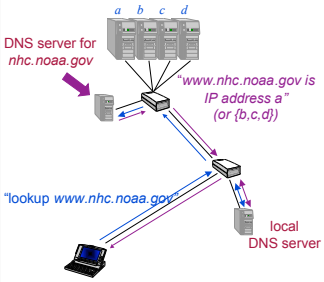
Server selection problem



- Avoid the scaleability problems of central servers by "distributing" load

Source: Jeff Chase

DNS round robin



DNS server returns one of multiple addresses based on load
e.g. www1.aol.com
www2.aol.com

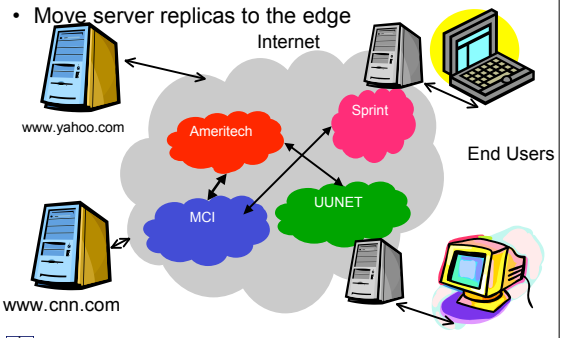
Source: Jeff Chase

DNS record for www.yahoo.com

```
www.yahoo.com.          1002  IN   CNAME  www.yahoo.akadns.net.
www.yahoo.akadns.net.   292   IN   A      64.58.76.223
www.yahoo.akadns.net.   292   IN   A      64.58.76.224
www.yahoo.akadns.net.   292   IN   A      64.58.76.225
www.yahoo.akadns.net.   292   IN   A      64.58.76.227
www.yahoo.akadns.net.   292   IN   A      64.58.76.228
www.yahoo.akadns.net.   292   IN   A      64.58.76.229
www.yahoo.akadns.net.   292   IN   A      64.58.76.176
www.yahoo.akadns.net.   292   IN   A      64.58.76.177
www.yahoo.akadns.net.   292   IN   A      64.58.76.178
www.yahoo.akadns.net.   292   IN   A      64.58.76.179
www.yahoo.akadns.net.   292   IN   A      64.58.76.222

;; AUTHORITY SECTION:
akadns.net.             984   IN   NS     ZF.akadns.net.
akadns.net.             984   IN   NS     ZG.akadns.net.
akadns.net.             984   IN   NS     ZH.akadns.net.
akadns.net.             984   IN   NS     ZA.akadns.net.
akadns.net.             984   IN   NS     ZB.akadns.net.
akadns.net.             984   IN   NS     ZC.akadns.net.
akadns.net.             984   IN   NS     ZD.akadns.net.
akadns.net.             984   IN   NS     ZE.akadns.net.
```


Content delivery network

- Move server replicas to the edge

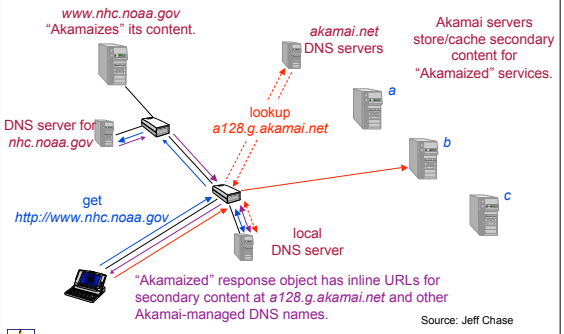


Content Delivery Network

- CDN (e.g., Akamai) creates new domain names for each client content provider.
  - e.g., *a128.g.akamai.net*
- The CDN's DNS servers are authoritative for the new domains.
- The client content provider modifies its content so that embedded URLs reference the new domains.
  - "Akamaize" content
  - e.g.: http://www.cnn.com/image-of-the-day.gif becomes
    http://a128.g.akamai.net/image-of-the-day.gif
- Using multiple domain names for each client allows the CDN to further subdivide the content into groups.
  - DNS sees only the requested domain name, but it can route requests for different domains independently.

Source: Jeff Chase

Akamai with DNS hooks



Source: Jeff Chase