CS615 - Aspects of System Administration

DNS; HTTP

Department of Computer Science
Stevens Institute of Technology
Jan Schaumann
jschauma@stevens-tech.edu
http://www.cs.stevens-tech.edu/~jschauma/615A/

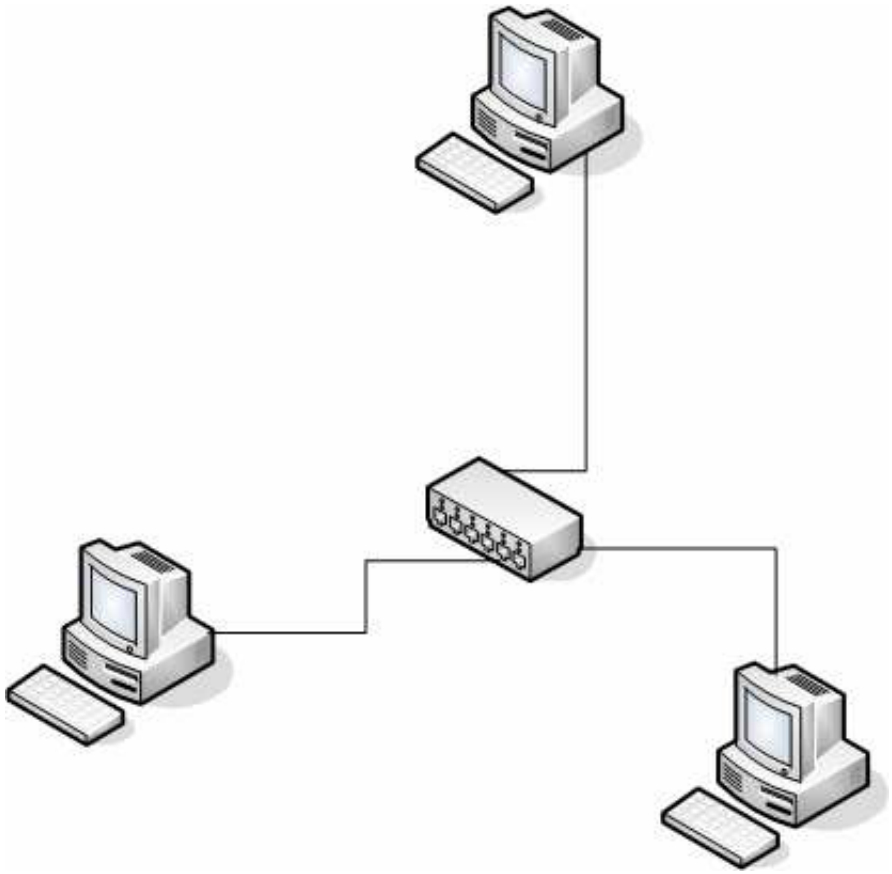
HW3

”Show your work.”

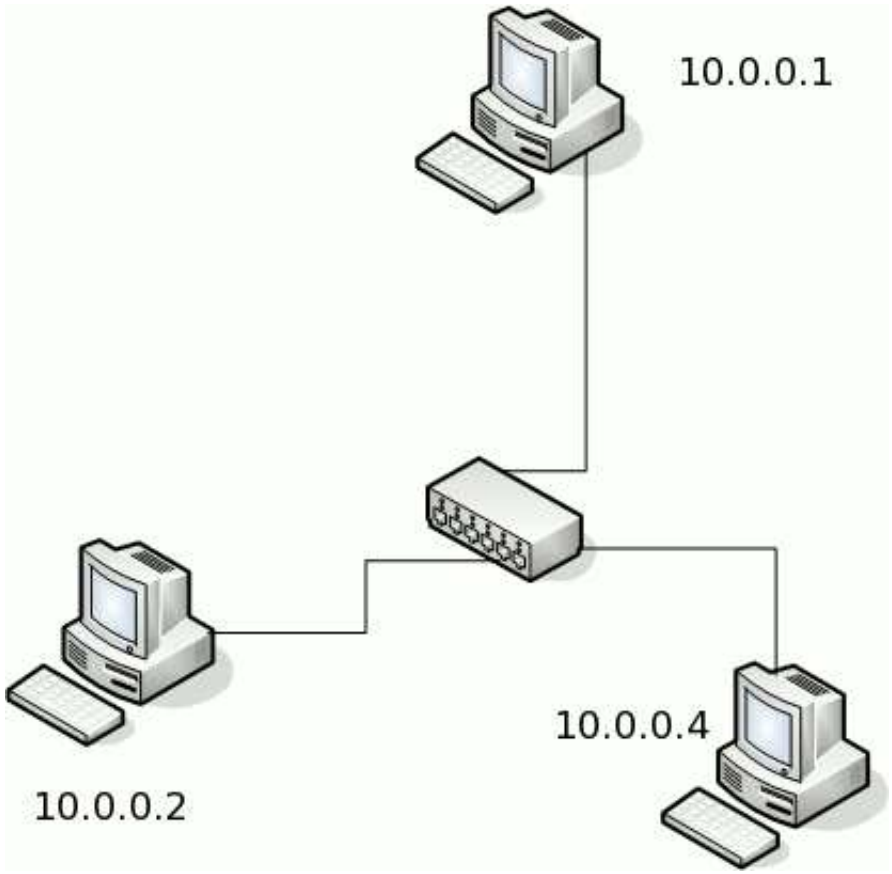
In the beginning...

In the beginning...

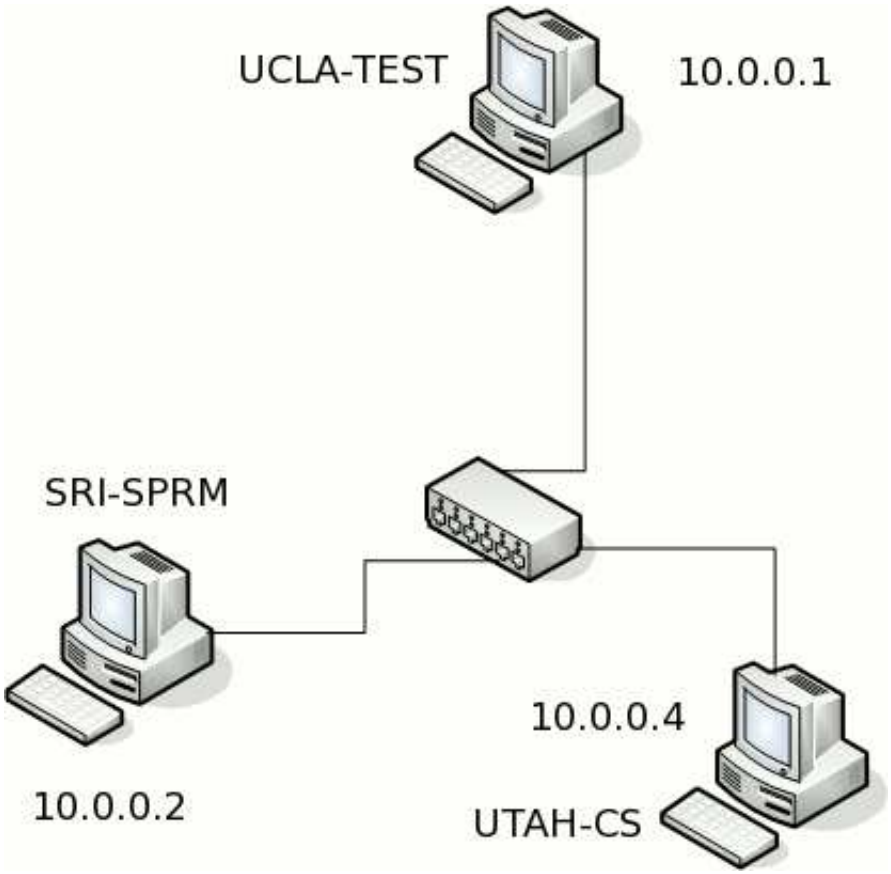
In the beginning...



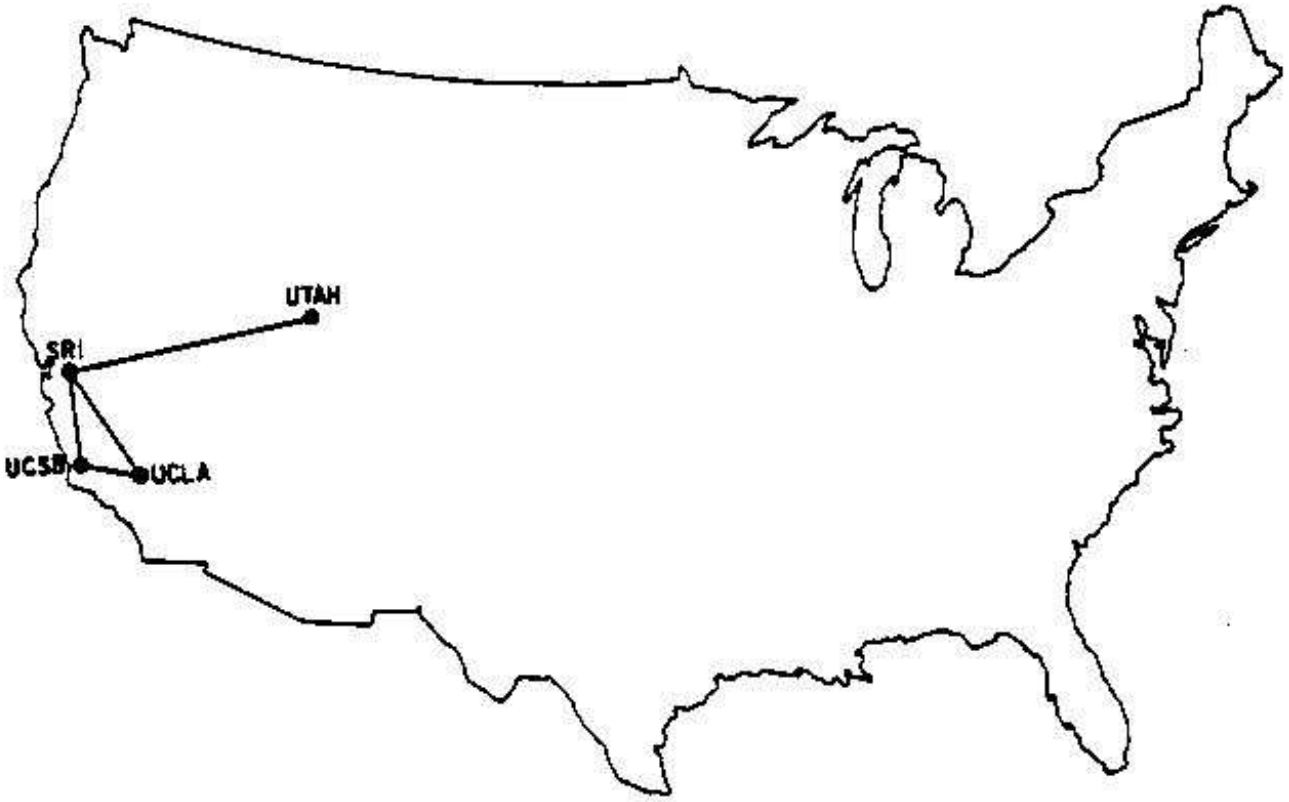
In the beginning...



In the beginning...



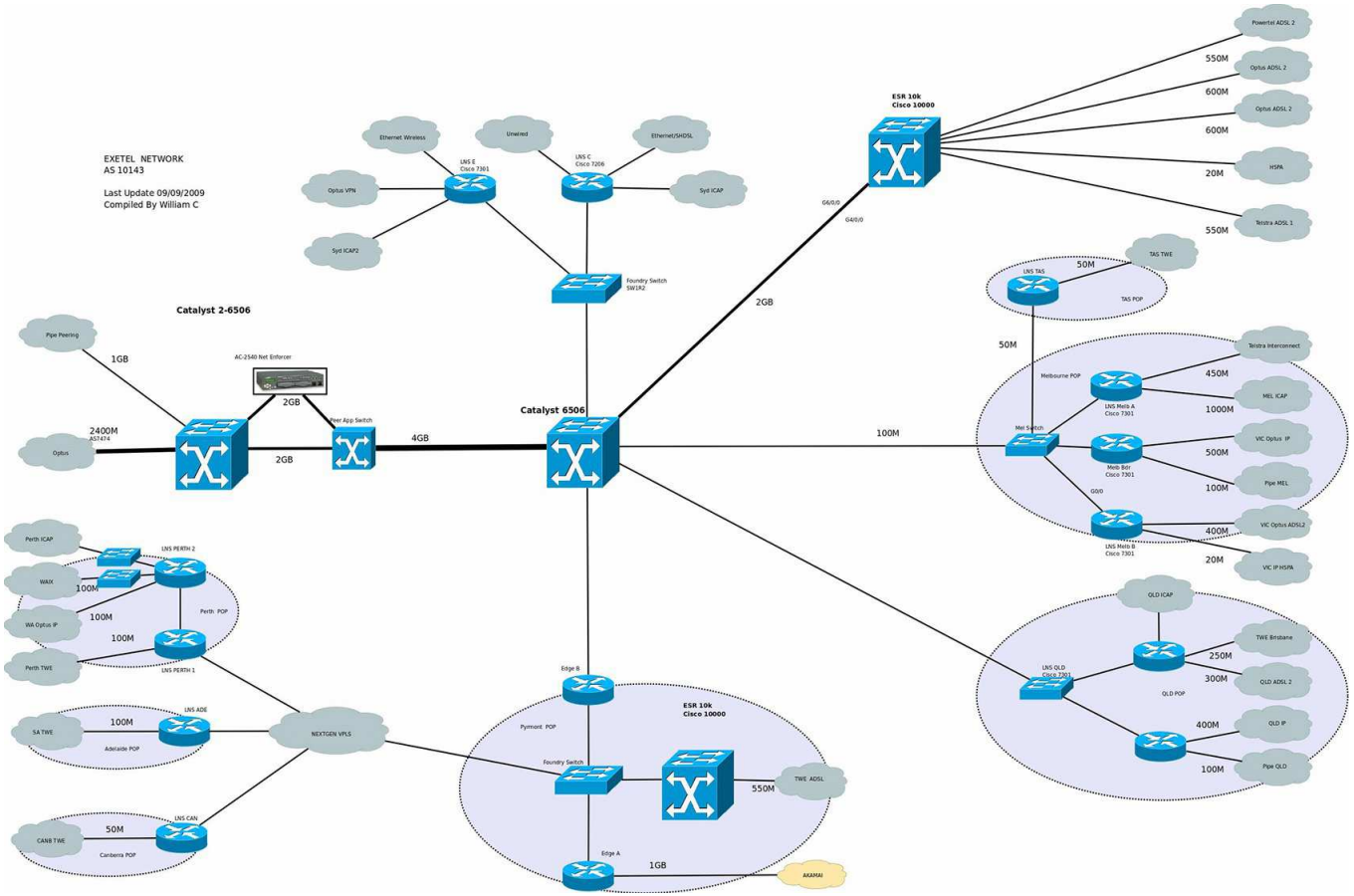
In the beginning...



In the beginning...

```
# Host Database
# This file should contain the addresses and aliases
# for local hosts that share this file.
#
127.0.0.1               localhost localhost.
#
# RFC 1918 specifies that these networks are "internal".
# 10.0.0.0      10.255.255.255
# 172.16.0.0    172.31.255.255
# 192.168.0.0   192.168.255.255
10.0.0.1 UCLA-TEST
10.0.0.2 SRI-SPRM
10.0.0.4 UTAH-CS
```

But then...


EXETEL NETWORK
AS 10143
Last Update 09/09/2009
Compiled By William C
Catalyst 2-6506
Catalyst 6506
ESR 10k
Cisco 10000
Ethernet Wireless
Unwired
Ethernet/SHDSL
Optus VPN
Syd ICAP
Syd ICAP2
Foundry Switch
Pipe Peering
1GB
2400M
Optus
AC-2540 Net Enforcer
2GB
Fiber App Switch
4GB
100M
Powertel ADSL 2
Optus ADSL 2
Optus ADSL 2
HSPA
Telstra ADSL 1
550M
600M
600M
20M
550M
G6/0/0
G4/0/0
50M
TAS TWE
LNS TAS
TAS POP
Telstra Interconnect
Melbourne POP
LNS Melb A
Cisco 7301
MEL ICAP
450M
1000M
Mel Switch
Melb Bdr
Cisco 7301
VIC Optus IP
500M
Pipe MEL
100M
G0/0
LNS Melb B
Cisco 7301
VIC Optus ADSL2
400M
VIC IP HSPA
20M
Perth ICAP
LNS PERTH 2
WAIX
100M
WA Optus IP
Perth POP
100M
Perth TWE
LNS PERTH 1
LNS ADE
100M
SA TWE
Adelaide POP
NEXTGEN VPLS
LNS CAN
50M
CANB TWE
Canberra POP
Edge B
Pymont POP
ESR 10k
Cisco 10000
Foundry Switch
550M
TWE ADSL
Edge A
1GB
AKAMAI
QLD ICAP
LNS QLD
Cisco 7301
TWE Brisbane
250M
300M
QLD ADSL 2
QLD POP
400M
QLD IP
100M
Pipe QLD

The Domain Name System

Computers like numbers.

10011011111101100101100110011111

The Domain Name System

Computers like numbers.

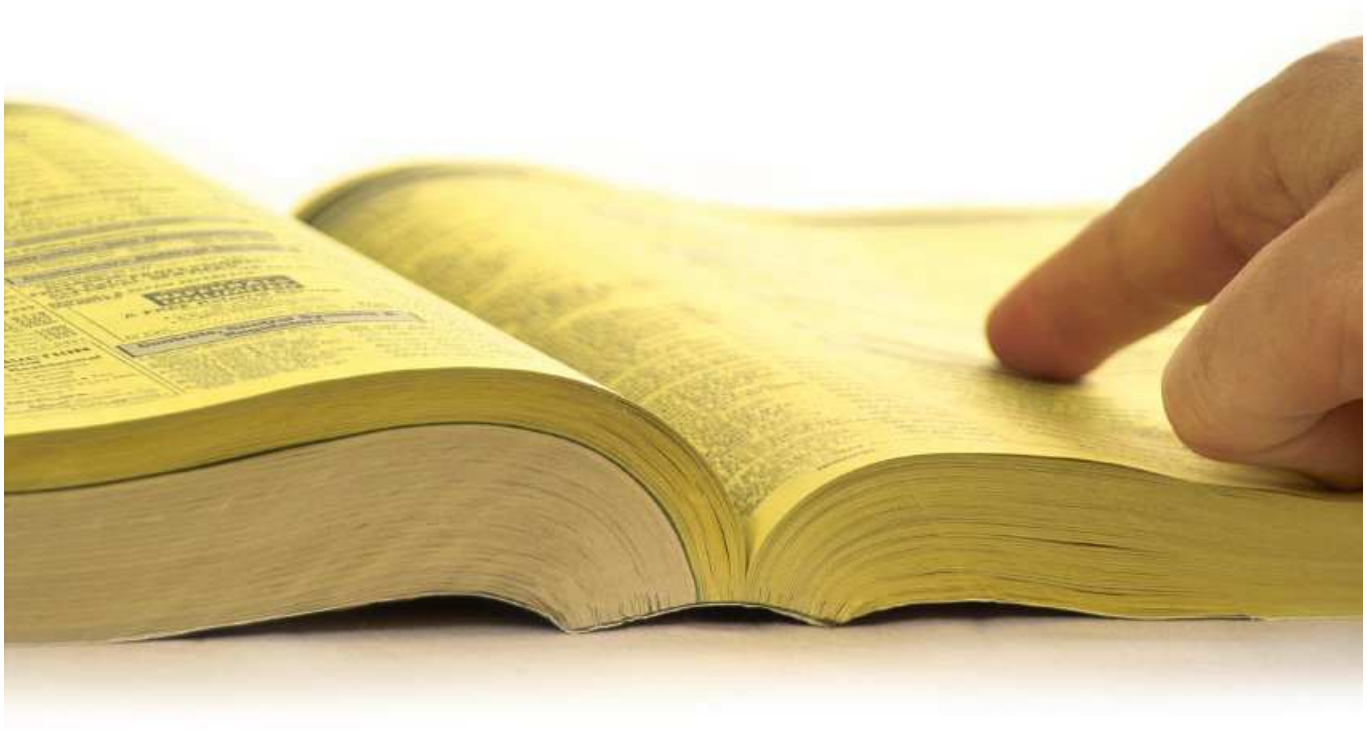
```
10011011   11110110   01011001   10011111

 155    .   246    .    89    .   159
```

The Domain Name System

People like names.

`ash.cs.stevens-tech.edu`

The Domain Name System

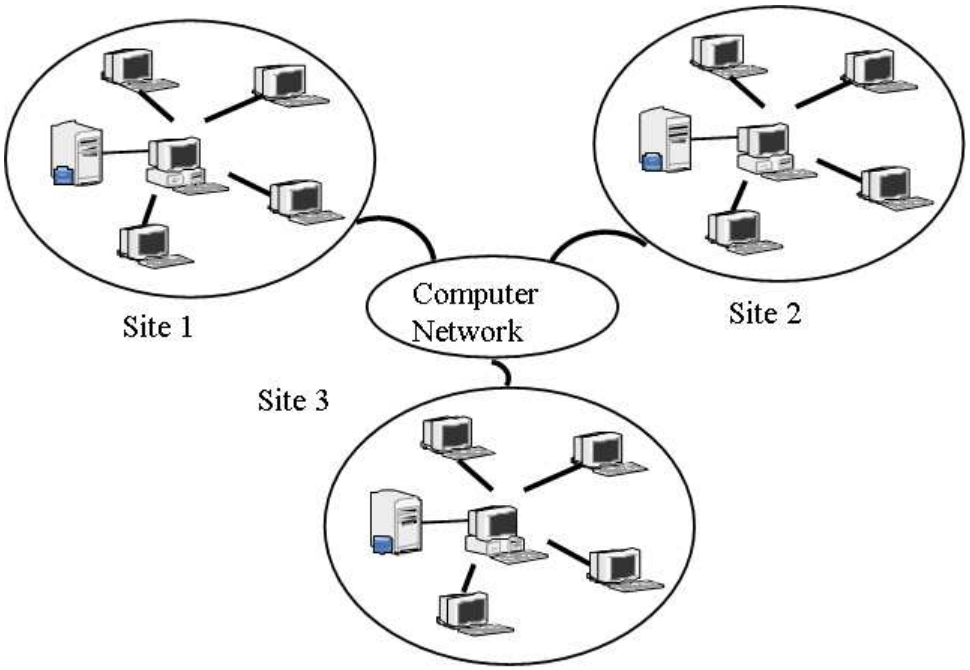


The New Phonebook is here!

```
http://is.gd/XXp2sC
```

```
wget -q -O - http://is.gd/XXp2sC | grep -c "^HOST"
```

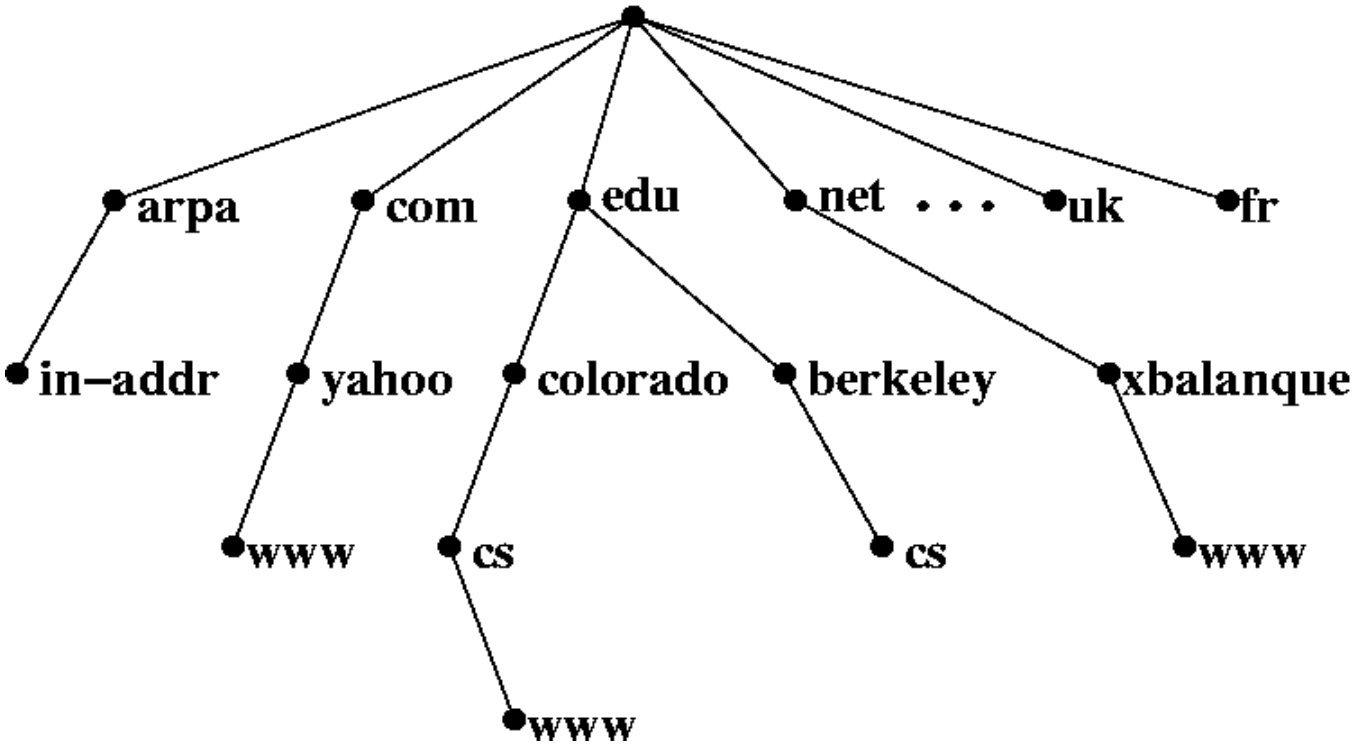
DNS: A distributed database



The Domain Name Space

The domain name space consists of a tree of *domain* names.

DNS: A hierarchical system



The Domain Name Space

The domain name space consists of a tree of *domain* names.

A subtree divides into *zones*.

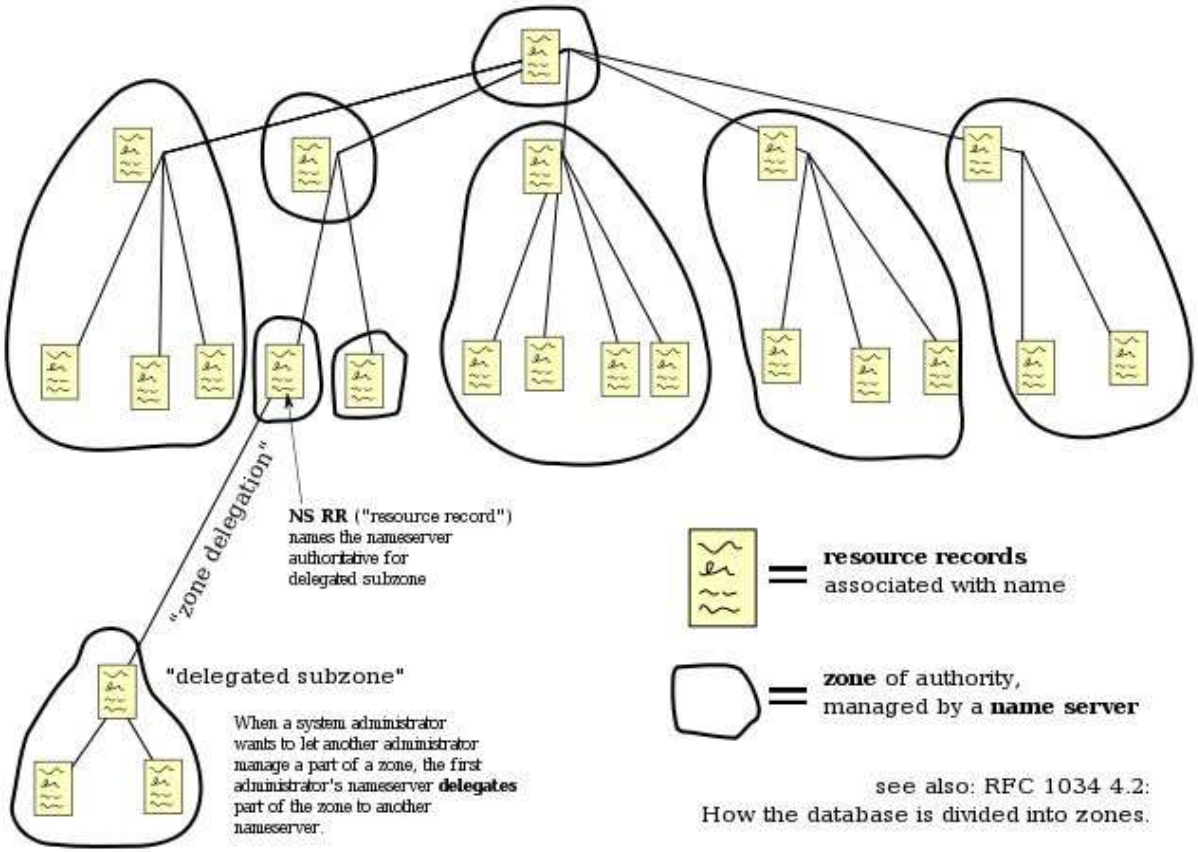
The Domain Name Space

The domain name space consists of a tree of *domain* names.

A subtree divides into *zones*.

Each node may contain *resource records*.

The Domain Name Space



Domain Names

`ash.cs.stevens-tech.edu`

Domain Names are read from right to left and components separated by a ".".

Domain Names

`ash.cs.stevens-tech.edu.`

The *root* is known as “.”, but is usually left out.

Domain Names

`ash.cs.stevens-tech.edu.`

There is a small number of *top level domains*.

Domain Names

ash.cs.stevens-tech.**edu**.

There is a number of *top level domains*.

```
wget -O - ftp://rs.internic.net/domain/root.zone | \
        grep "IN<tab>*NS<tab>" | awk '{print $1}' | sort -u | wc -l
```

http://data.iana.org/TLD/tlds-alpha-by-domain.txt
https://en.wikipedia.org/wiki/List_of_Internet_top-level_domains

Domain Names

`ash.cs.stevens-tech.edu.`

Each *domain* can be divided into any number of *sub domains*.

Domain Names

`ash.cs.stevens-tech.edu.`

Each *domain* can be divided into any number of *sub domains*.

Domain Names

ash.cs.stevens-tech.edu.

The left-most component of a domain name may be a *hostname*.

Fully Qualified Domain Names

`ash.cs.stevens-tech.edu.`

A *hostname* with a domain name is known as a *FQDN*.

DNS servers come in two flavors

Authoritative Nameservers

Recursive Nameservers

Hostname resolution

Resolution on a recursive nameserver (aka *resolver*) involves a number of queries:

```
$ nslookup ash.cs.stevens-tech.edu
Server:         127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
Name:   ash.cs.stevens-tech.edu
Address: 155.246.89.159

$
```

Hostname resolution

Resolution on a *resolver* involves a number of queries:

```
18:39:27.186778 IP panix.netmeister.org.62105 > i.root-servers.net.domain:
        11585 [1au] A? ash.cs.stevens-tech.edu. (52)
18:39:27.446190 IP i.root-servers.net.domain > panix.netmeister.org.62105:
        11585- 0/8/8 (494)
18:39:27.446994 IP panix.netmeister.org.53168 > a.gtld-servers.net.domain:
        46575 [1au] A? ash.cs.stevens-tech.edu. (52)
18:39:27.481565 IP a.gtld-servers.net.domain > panix.netmeister.org.53168:
        46575- 0/6/3 (609)
18:39:27.481998 IP panix.netmeister.org.41071 > nrac.stevens-tech.edu.domain:
        24322 [1au] A? ash.cs.stevens-tech.edu. (52)
18:39:27.486035 IP nrac.stevens-tech.edu.domain > panix.netmeister.org.41071:
        24322*- 1/2/3 A[|domain]
```

Hostname resolution

Resolution on a *resolver* involves a number of queries:

```
$ host -t ns .
. name server I.ROOT-SERVERS.NET.
. name server D.ROOT-SERVERS.NET.
. name server C.ROOT-SERVERS.NET.
. name server M.ROOT-SERVERS.NET.
. name server F.ROOT-SERVERS.NET.
. name server A.ROOT-SERVERS.NET.
. name server E.ROOT-SERVERS.NET.
. name server L.ROOT-SERVERS.NET.
. name server H.ROOT-SERVERS.NET.
. name server J.ROOT-SERVERS.NET.
. name server B.ROOT-SERVERS.NET.
. name server G.ROOT-SERVERS.NET.
. name server K.ROOT-SERVERS.NET.
$
```

Hostname resolution

Resolution on a *resolver* involves a number of queries:

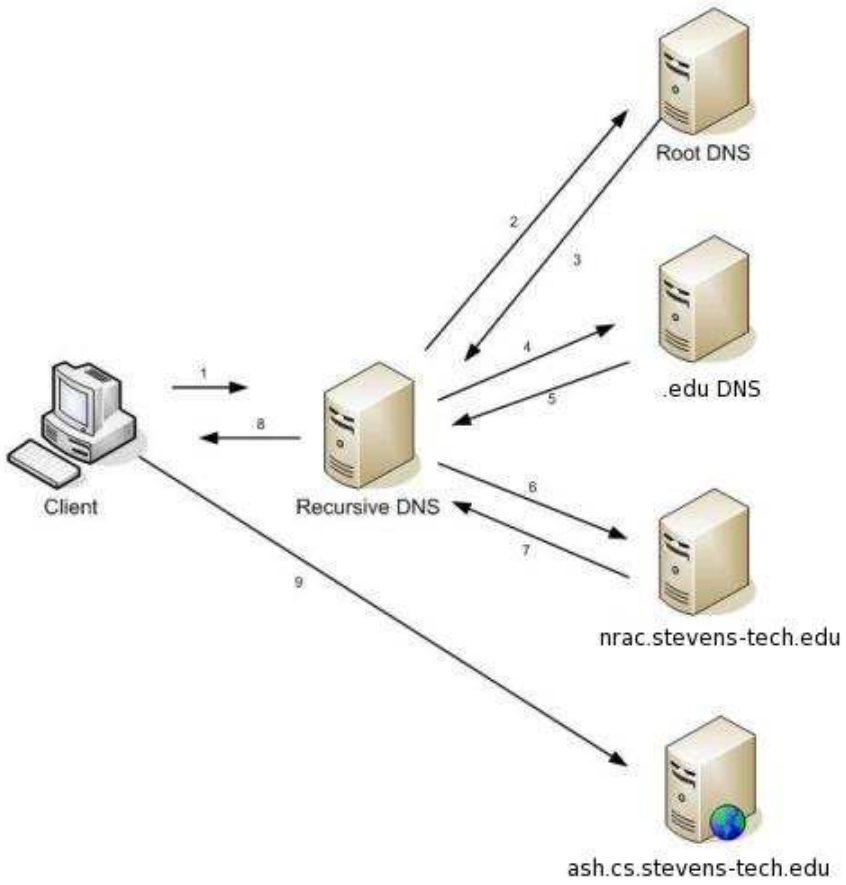
```
$ dig -t ns edu.
[...]
;; ANSWER SECTION:
edu.                    172800  IN      NS      l.edu-servers.net.
edu.                    172800  IN      NS      f.edu-servers.net.
edu.                    172800  IN      NS      c.edu-servers.net.
edu.                    172800  IN      NS      g.edu-servers.net.
edu.                    172800  IN      NS      a.edu-servers.net.
edu.                    172800  IN      NS      d.edu-servers.net.

;; ADDITIONAL SECTION:
c.edu-servers.net.      36626   IN      A       192.26.92.30
d.edu-servers.net.      13274   IN      A       192.31.80.30
l.edu-servers.net.      36626   IN      A       192.41.162.30
[...]
$
```

Hostname resolution

Resolution on a *resolver* involves a number of queries:

```
$ dig @c.edu-servers.net -t ns stevens.edu.
[...]
;; AUTHORITY SECTION:
stevens.edu.            172800  IN      NS      nrac.stevens-tech.edu.
stevens.edu.            172800  IN      NS      sitult.stevens-tech.edu.

;; ADDITIONAL SECTION:
nrac.stevens-tech.edu.  172800  IN      A       155.246.1.21
sitult.stevens-tech.edu. 172800 IN      A       155.246.1.20
[...]
$
```

Hostname resolution



Hostname resolution

Resolution on a *resolver* involves a number of queries:

```
$ nslookup ash.cs.stevens-tech.edu
Server:         127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
Name:   ash.cs.stevens-tech.edu
Address: 155.246.89.159

$
```

Hostname resolution

Hostname resolution

```
$ ftp -o - ftp.internic.net:/domain/db.cache | more
    http://www.internic.net/zones/named.root
```

Operation Global Blackout

http://pastebin.com/XZ3EGsbc

DNS: A distributed system

There are 13 `root` servers.

DNS: A distributed system

There are 13 `root` servers.

Except... there are more.

DNS: A distributed system

There are 13 `root` *authorities*.

DNS: A distributed system

There are 13 `root server` *addresses*.

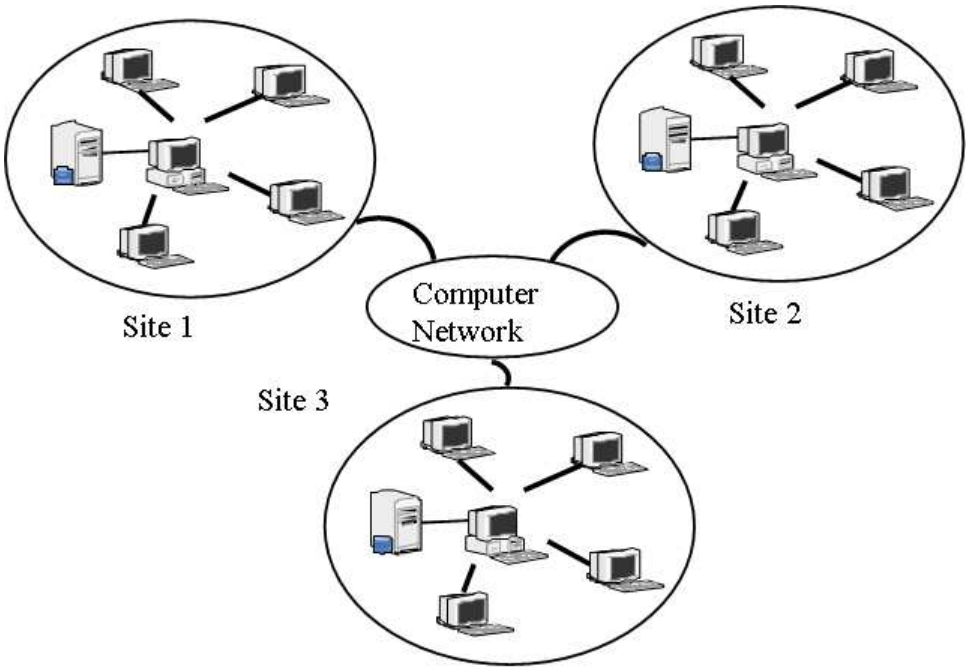
DNS: A distributed system

There are hundreds of `root` servers.

DNS: A distributed system

Operation Global Blackout

DNS: A distributed database



DNS Resource Records

- *NS* – an authoritative name server
- *CNAME* – the canonical name for an alias
- *SOA* – marks the start of a zone of authority
- *PTR* – a domain name pointer
- *HINFO* – host information
- *MX* – mail exchange
- *TXT* text strings
- ...

DNS Resource Records

You've all seen PTR records:

```
$ host ash.cs.stevens-tech.edu
ash.cs.stevens-tech.edu has address 155.246.89.159
ash.cs.stevens-tech.edu mail is handled by 0 guinness.cs.stevens-tech.edu.
$ host 155.246.89.159
159.89.246.155.in-addr.arpa domain name pointer ash.cs.stevens-tech.edu.
$
```

Stevens doesn't have write access to the `in-addr.arpa` domain. How does this work?

Creative uses of DNS Resource Records

- identifying sources of SPAM
- find out if the internet is on fire:
  `dig +short txt istheinternetonfire.com`
- find ASN numbers by IP addresses:
  `dig +short 159.89.246.155.origin.asn.cymru.com TXT`
- check a resolver's source port randomization (to help mitigate DNS Cache Poisoning attacks):
  `dig +short porttest.dns-oarc.net TXT`
- using DNS to publish SSH key fingerprints (RFC4255, ssh_config(5) `VerifyHostKeyDNS`; for best results combine with DNSSEC):
  `dig +short ftp.netbsd.org SSHFP`

```
ssh -o "VerifyHostKeyDNS yes" ftp.netbsd.org
[...]
Matching host key fingerprint found in DNS.
Are you sure you want to continue connecting (yes/no)?
```

Hooray!

5 Minute Break

Hypertext Transfer Protocol

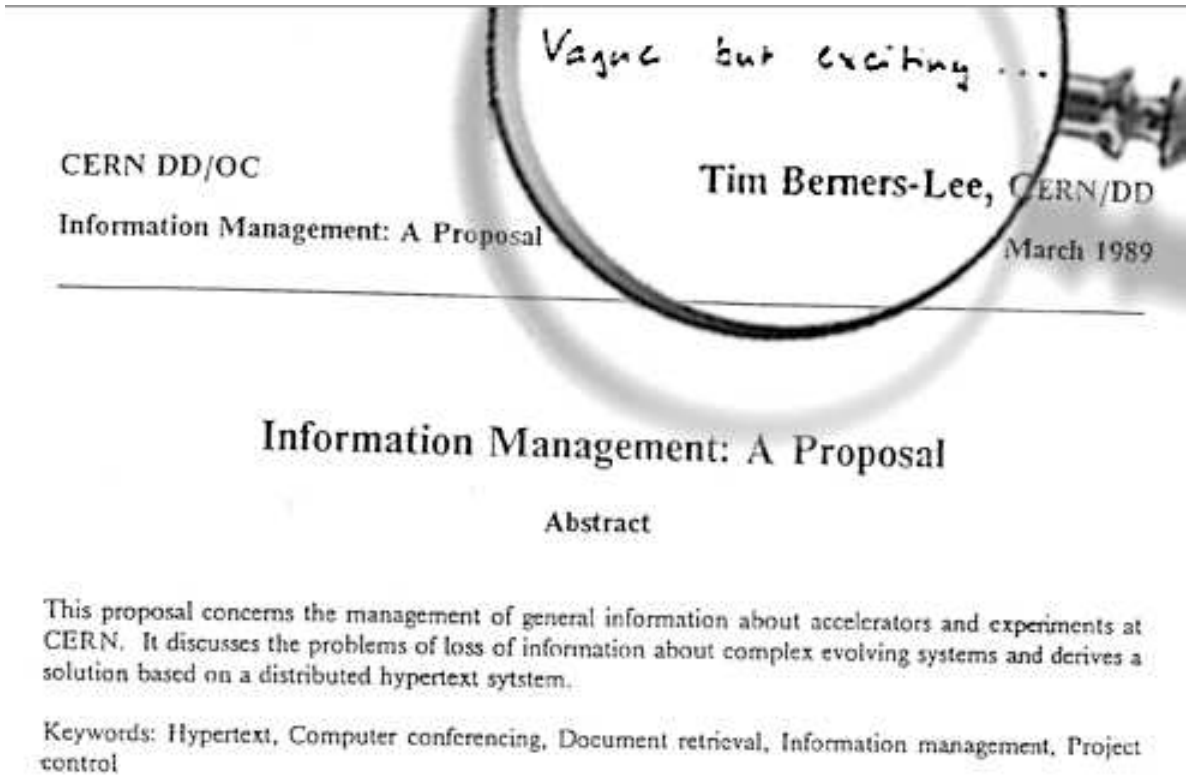
Today's Universal Internet Pipe

HTTP: Hypertext

W W W

“The World Wide Web is the only thing I know of whose shortened form takes three times longer to say than what it’s short for.” – Douglas Adams

HTTP: Hypertext



Vague but exciting ...

CERN DD/OC

Information Management: A Proposal

Tim Berners-Lee, CERN/DD

March 1989

Information Management: A Proposal

Abstract

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext sytstem.

Keywords: Hypertext, Computer conferencing, Document retrieval, Information management, Project control

http://is.gd/JnZaN6

HTTP

Hypertext Transfer Protocol

RFC2616

HTTP

HTTP is a request/response protocol.

The Hypertext Transfer Protocol

HTTP is a request/response protocol:

1. client sends a request to the server
2. server responds

The Hypertext Transfer Protocol

HTTP is a request/response protocol:

1. client sends a request to the server
   - request method
   - URI
   - protocol version
   - request modifiers
   - client information
2. server responds

HTTP: A client request

```
$ telnet www.google.com 80
Trying 173.194.75.147...
Connected to www.google.com.
Escape character is '^]'.
GET / HTTP/1.0
```

The Hypertext Transfer Protocol

HTTP is a request/response protocol:

1. client sends a request to the server
   - request method
   - URI
   - protocol version
   - request modifiers
   - client information
2. server responds
   - status line (including success or error code)
   - server information
   - entity metainformation
   - content

HTTP: a server response

```
HTTP/1.0 200 OK
Date: Sun, 31 Mar 2013 01:54:40 GMT
Set-Cookie: PREF=ID=c5eb56d629b347cc:FF=0:TM=1364694880:LM=1364694880:
S=sIdRFdxV9YvtQOlG; expires=Tue, 31-Mar-2015 01:54:40 GMT; path=/;
domain=.google.com
Set-Cookie: NID=67=hvBnOob2NoZW4haTJVfajbcyn_jips50lKRe-8nawzdCZ6AukNR
_s8CNHD6ZA-Z2721nA3TpLrNXt-2zyIui23j4kdsdF8Gg--PmGsMOJ3Jv5frEzQG1elHJv92HL-w2;
expires=Mon, 30-Sep-2013 01:54:40 GMT; path=/; domain=.google.com; HttpOnly
Server: gws

<!doctype html><html itemscope="itemscope" itemtype="http://schema.org/WebPage">
<head><meta content="Search the...
```

The Hypertext Transfer Protocol

Server status codes:

- **1xx** – Informational; Request received, continuing process
- **2xx** – Success; The action was successfully received, understood, and accepted
- **3xx** – Redirection; Further action must be taken in order to complete the request
- **4xx** – Client Error; The request contains bad syntax or cannot be fulfilled
- **5xx** – Server Error; The server failed to fulfill an apparently valid request

HTTP: A client request

```
$ telnet www.cs.stevens.edu 80
Trying 155.246.89.84...
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 302 Found
Date: Sun, 12 Apr 2015 20:37:23 GMT
Server: Apache/2.2.22 (Debian)
Location: http://www.stevens.edu/ses/cs
Vary: Accept-Encoding
Content-Length: 297
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>302 Found</title>
</head><body>
```

HTTP: A client request

```
$ telnet www.stevens.edu 80
Trying 104.16.126.51...
Connected to www.stevens.edu.cdn.cloudflare.net.
Escape character is '^]'.
GET /ses/cs HTTP/1.1
Host: www.stevens.edu

HTTP/1.1 301 Moved Permanently
Date: Sun, 05 Mar 2017 21:17:24 GMT
Location: https://www.stevens.edu/ses/cs
```

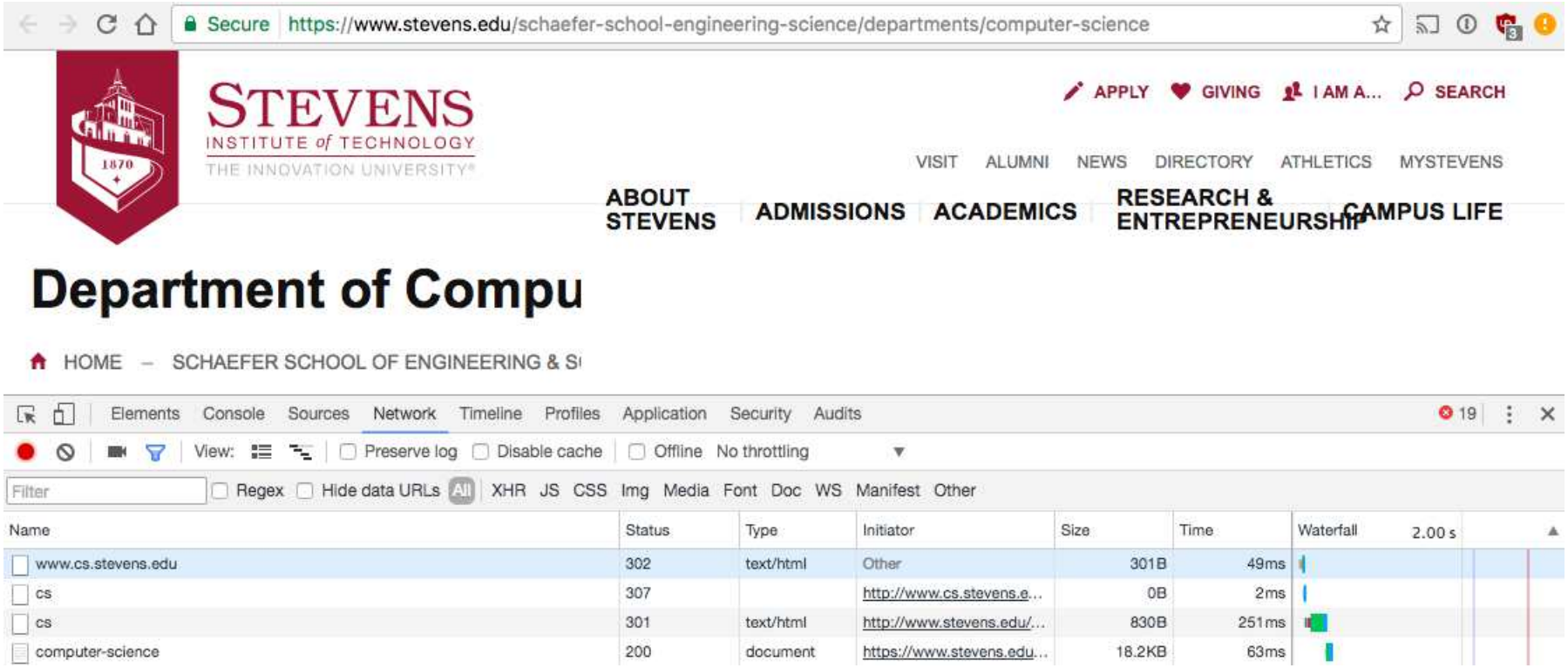
HTTP: A client request

```
$ openssl s_client -connect www.stevens.edu:443
[...]
GET /ses/cs HTTP/1.1
Host: www.stevens.edu

HTTP/1.1 301 Moved Permanently
Location: https://www.stevens.edu/schaefer-school-engineering-science/departments/computer-science
```

HTTP: A client request

```
$ openssl s_client -connect www.stevens.edu:443
[...]
GET /schaefer-school-engineering-science/departments/computer-science HTTP/1.1
Host: www.stevens.edu

HTTP/1.1 200 OK
Date: Sun, 05 Mar 2017 21:26:34 GMT
Last-Modified: Sun, 05 Mar 2017 16:50:25 GMT
Content-Type: text/html; charset=utf-8
X-Drupal-Cache: HIT
X-Generator: Drupal 7 (http://drupal.org)
Server: cloudflare-nginx

7c9f
<!DOCTYPE html>
<html lang="en" class="no-js">
<head>
```

HTTP: A client request



HTTP - more than just text

HTTP is a *Transfer Protocol* – serving *data*, not any specific text format.

- `Accept-Encoding` client header can specify different formats such as *gzip*, *Shared Dictionary Compression over HTTP (SDCH)* etc.
- corresponding server headers: `Content-Type` and `Content-Encoding`

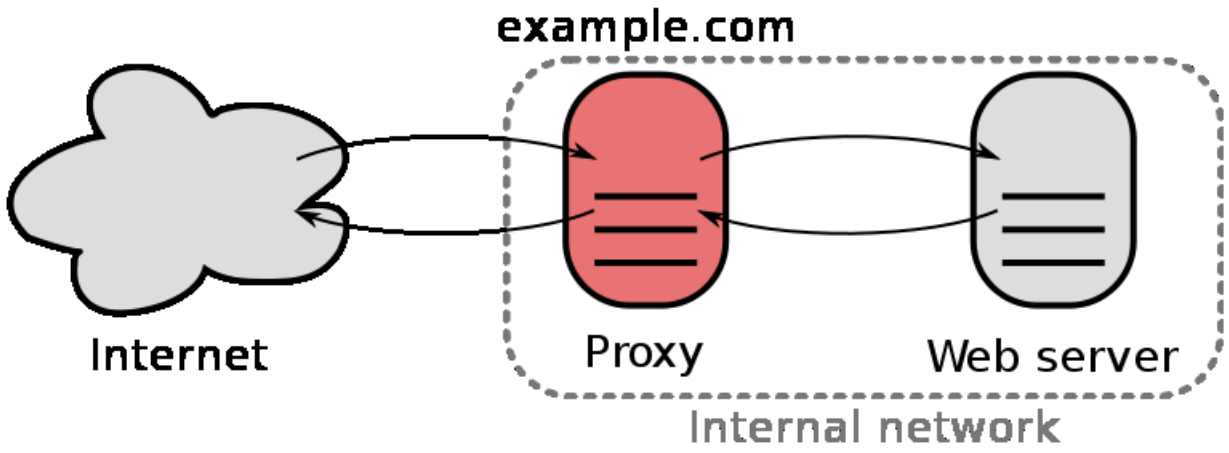
HTTP - more than just static data

HTTP is a *Transfer Protocol* – what is transferred need not be static; resources may generate different data to return based on many variables.

- CGI – resource is *executed*, needs to generate appropriate response headers
- server-side scripting (ASP, PHP, Perl, ...)
- client-side scripting (JavaScript/ECMAScript/JScript,...)
- applications based on HTTP, using:
  - AJAX
  - RESTful services
  - JSON, XML, YAML to represent state and abstract information

HTTP Proxy Servers

- HTTP traffic usually is very asymmetric
- a lot of the content is static
- network ACLs may restrict traffic flow



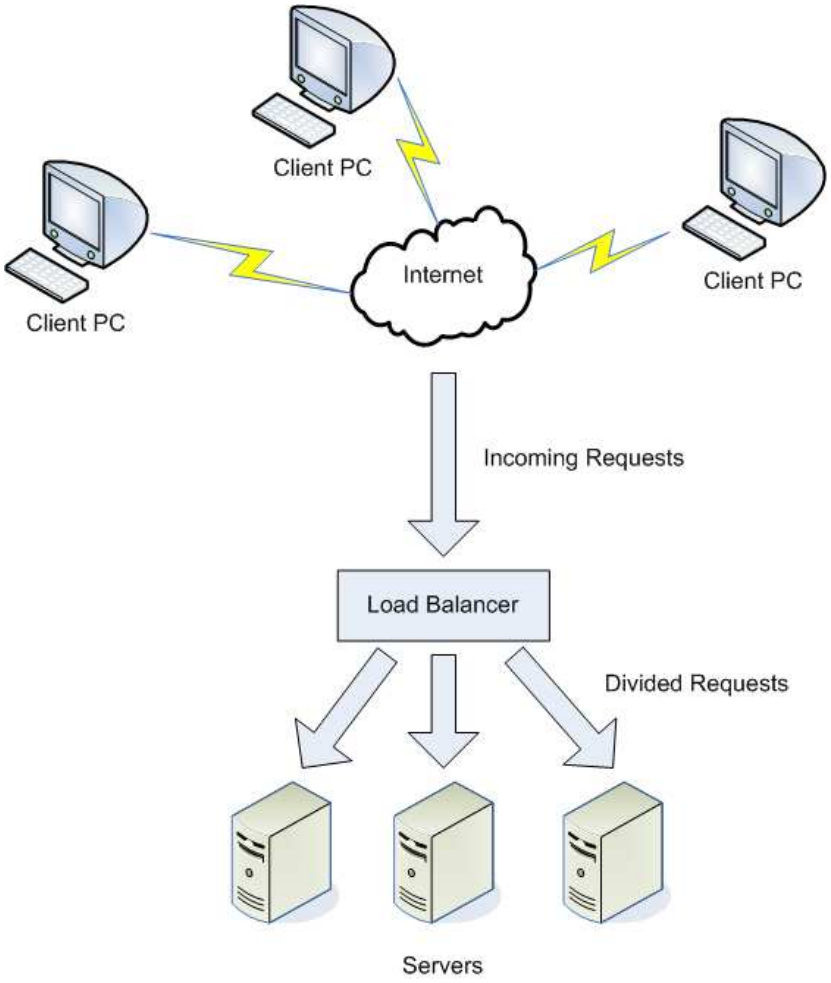
HTTP overload

Ways to mitigate HTTP overload:

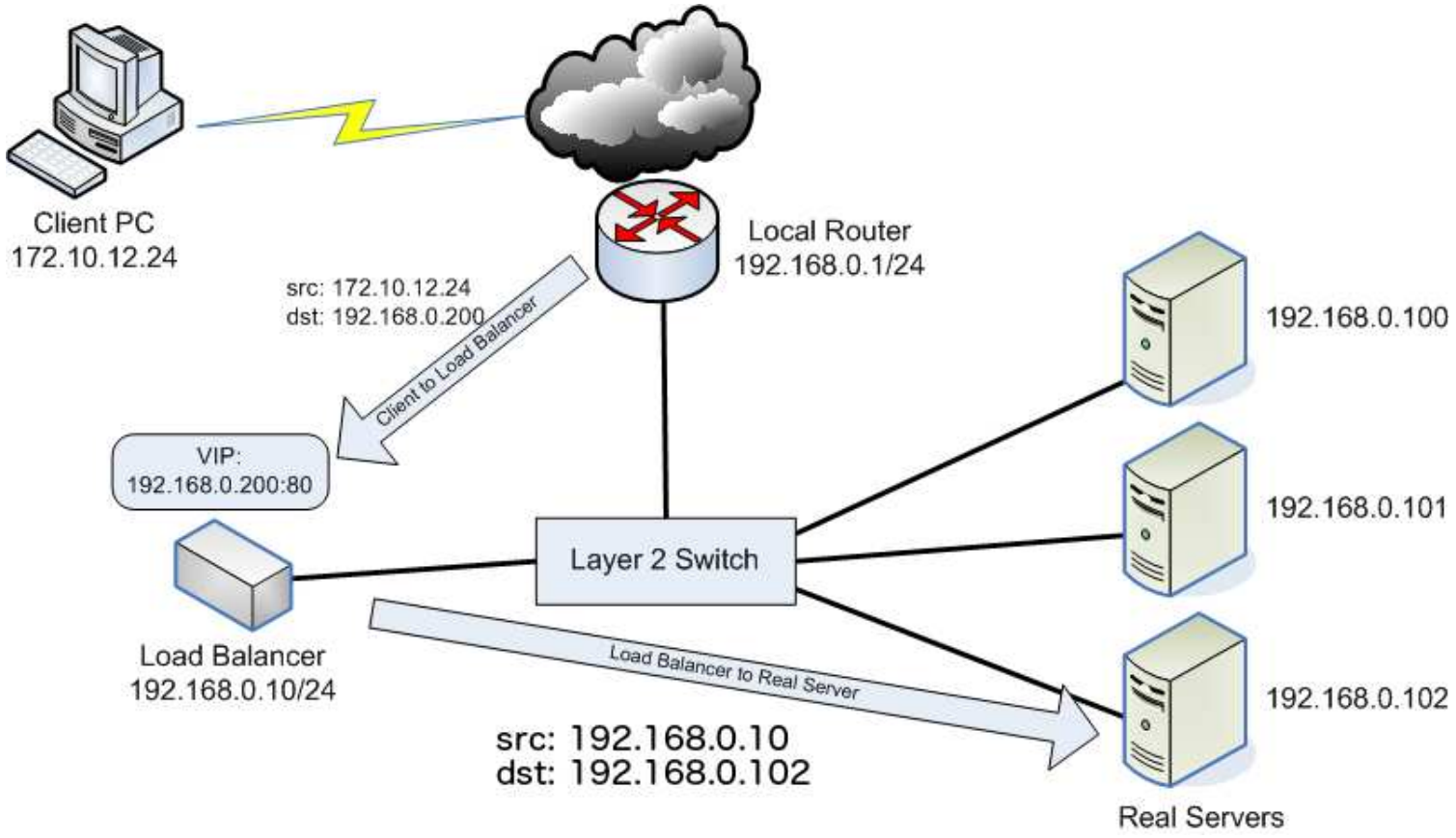
- DNS round-robin to many web servers
- load balancing
- web cache / accelerators (reverse proxies)
- content delivery networks

These solutions depend on the location within the network and the scale of the environment.

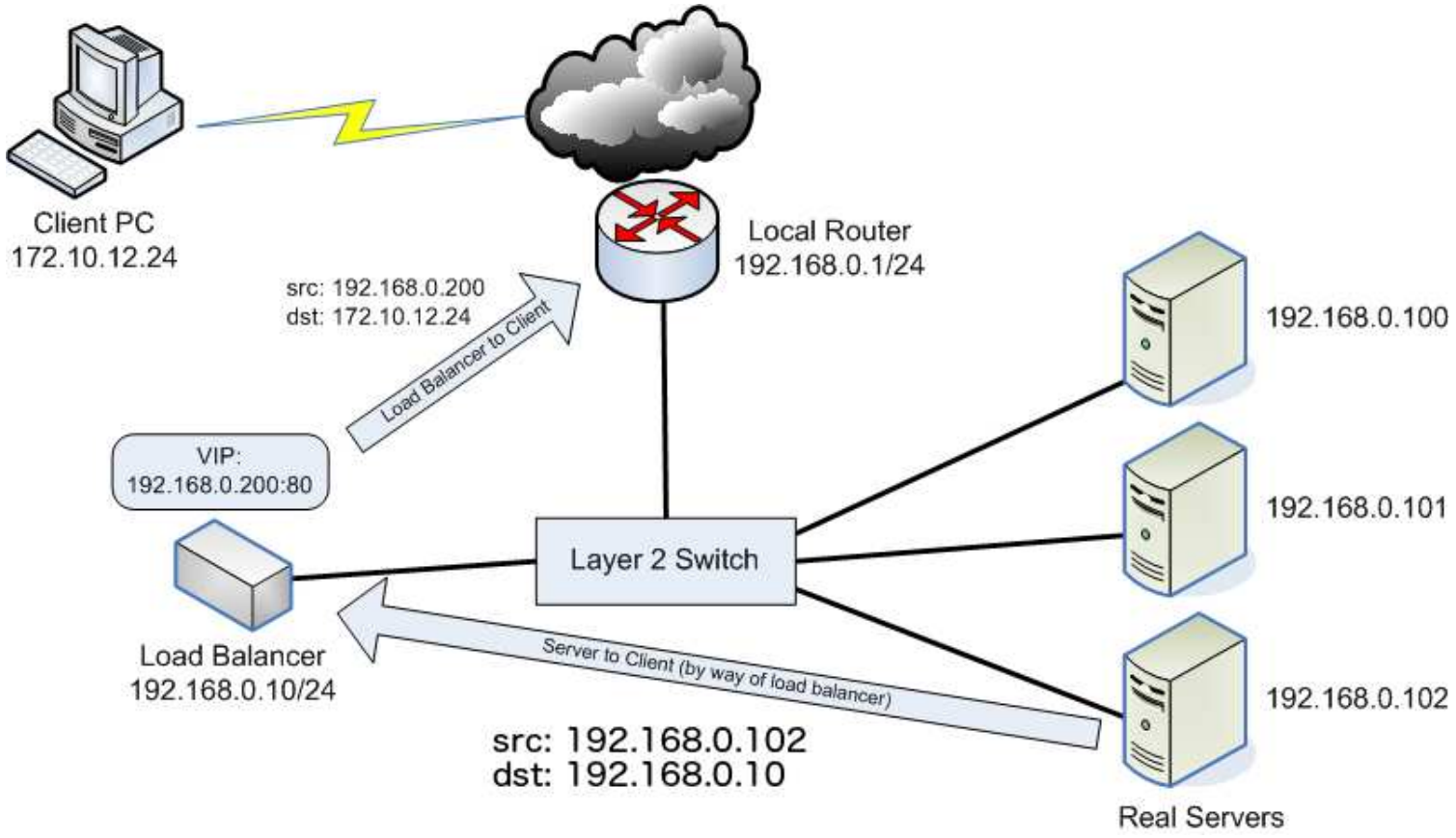
Load Balancing



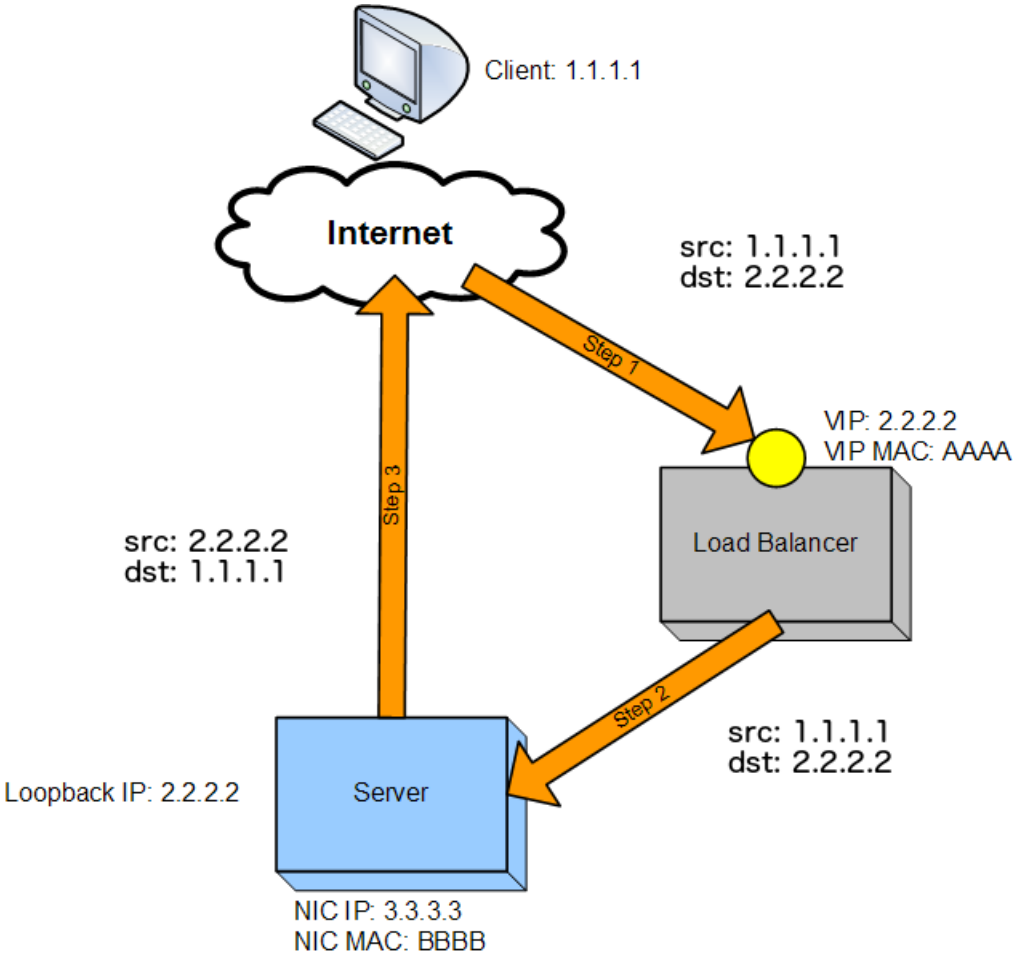
Load Balancing: Inbound



Load Balancing: Outbound



Load Balancing: Direct Server Return



Content Delivery Networks

Content Delivery Networks

- cache content in strategic locations
- determine location to serve from via geomapping of IP addresses (beware IPv6 aggregation!)
- often uses a separate domain to distinguish small objects/large objects or dynamic content/static content
- either out-sourced or in-house (if your organization is a Tier-1 or Tier-2 peering partner)
- request routing happens via Global Server Load Balancing, DNS-based request routing, anycasting etc.
- provides vast amounts of interesting data about your clients (see `http://www.akamai.com/stateoftheinternet/`)

Homework

`https://www.cs.stevens.edu/~jschauma/615/s17-hw4.html`

Reading

HTTP etc.:

- RFC 2616, 2818, 3875
- `http://httpd.apache.org/docs/`
- `http://www.w3.org/Protocols/`
- REST: `http://is.gd/leSvGa`
- CDNs: `http://is.gd/R5DoxA`
  - `http://www.edgecast.com/`
  - `https://aws.amazon.com/cloudfront/`
  - `http://www.akamai.com/`
  - `http://www.limelight.com/`
  - ...
- `http://developer.yahoo.com/performance/rules.html`