

Domain Name System Technology Overview

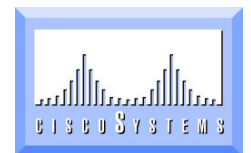
(DNS and Bind)





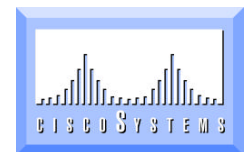
DNS Roadmap

- **DNS Introduction and Overview**
Function of DNS Client, DNS Server
- **DNS Terminology**
DNS Resource Records
- **Types of Nameservers**
- **DNS Hints**

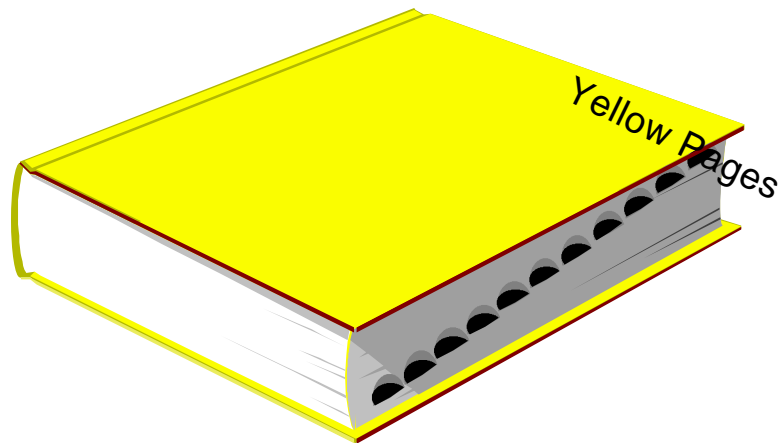
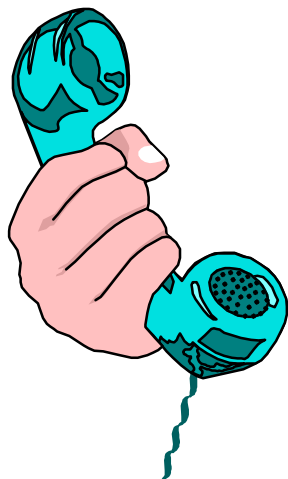




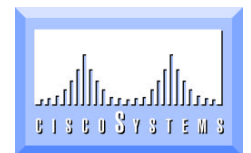
Introduction



The situation...



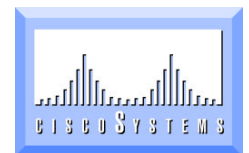
What is the number for ???





In the beginning...

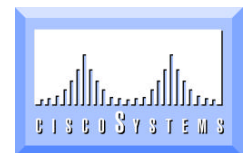
- **Systems used host tables for name to address translation**
- **When size of the Internet grew to about 1000 hosts in 1984, it became obvious that host tables would not scale well as the Internet continued to grow**
- **All hosts need to have their host tables updated when hosts are added or removed**





Implementations...

- **BIND**
Berkeley (Unix)
- **WINS**
Microsoft
- **NIS (“Yellow Pages”)**
Sun
- **DECdns**
Digital





Domain Name System

- **Also called BIND**

Berkeley Internet Name Domain

- **Distributed database**

Not all information is in one place

Entire database is not centrally managed

Both a feature and a potential weakness

- **DNS Resource Records**

A, PTR, MX, HINFO, TXT, NS, SOA, CNAME



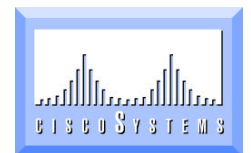


DNS Non-Functions

- **DNS does not control routing**
- **DNS does not affect IP connectivity**

However,

- **When hostnames can not be translated due to DNS failure, the user often assumes that the network is down**



DNS host naming

- Fully-Qualified Domain Names (FQDN)

most specific

least specific

host.subdomain.domain



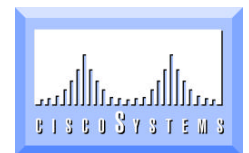
cone.tgv.com

www.tgv.com

hq.tgv.cisco.com

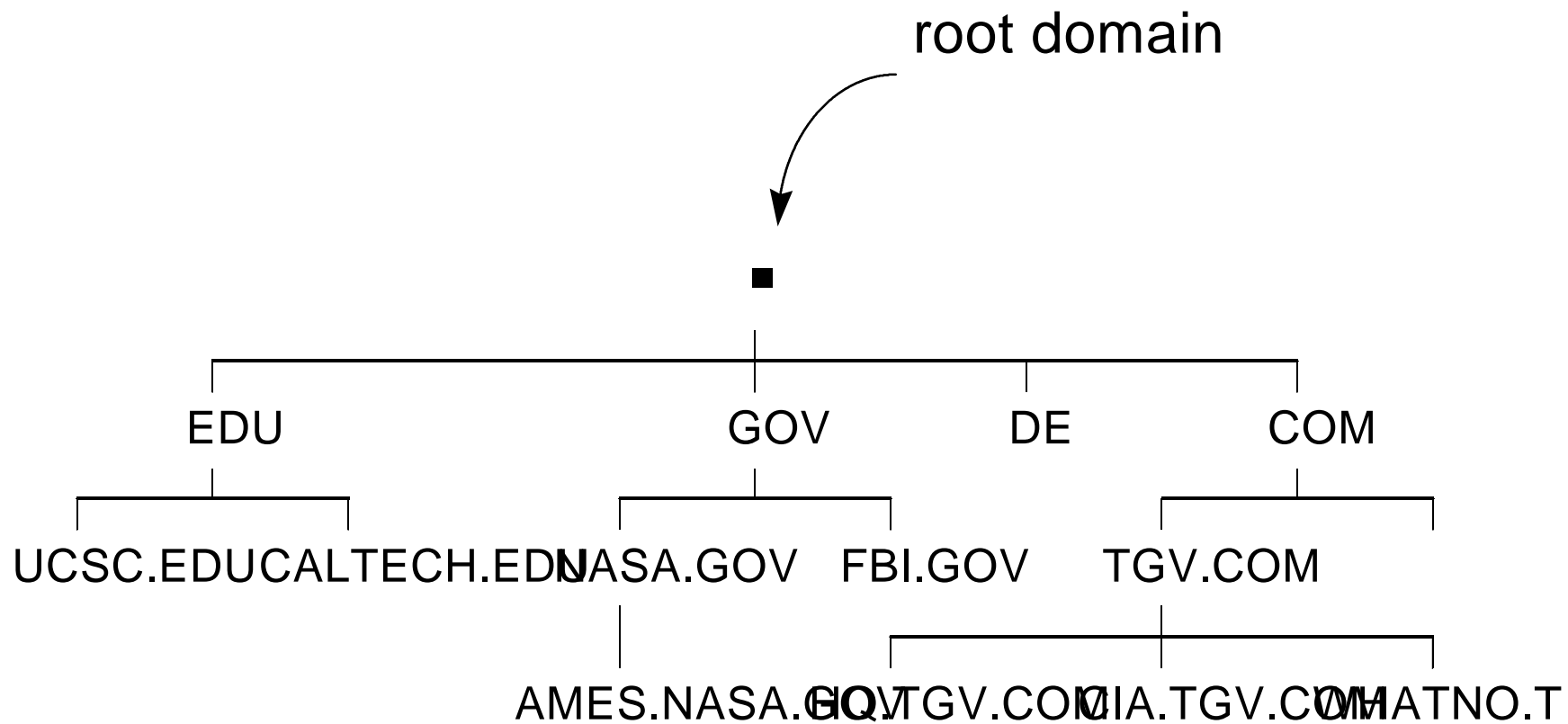
fog.isdn.cisco.com

eql.caltech.edu





DNS Structure





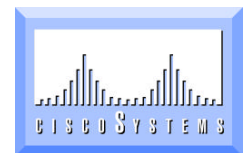
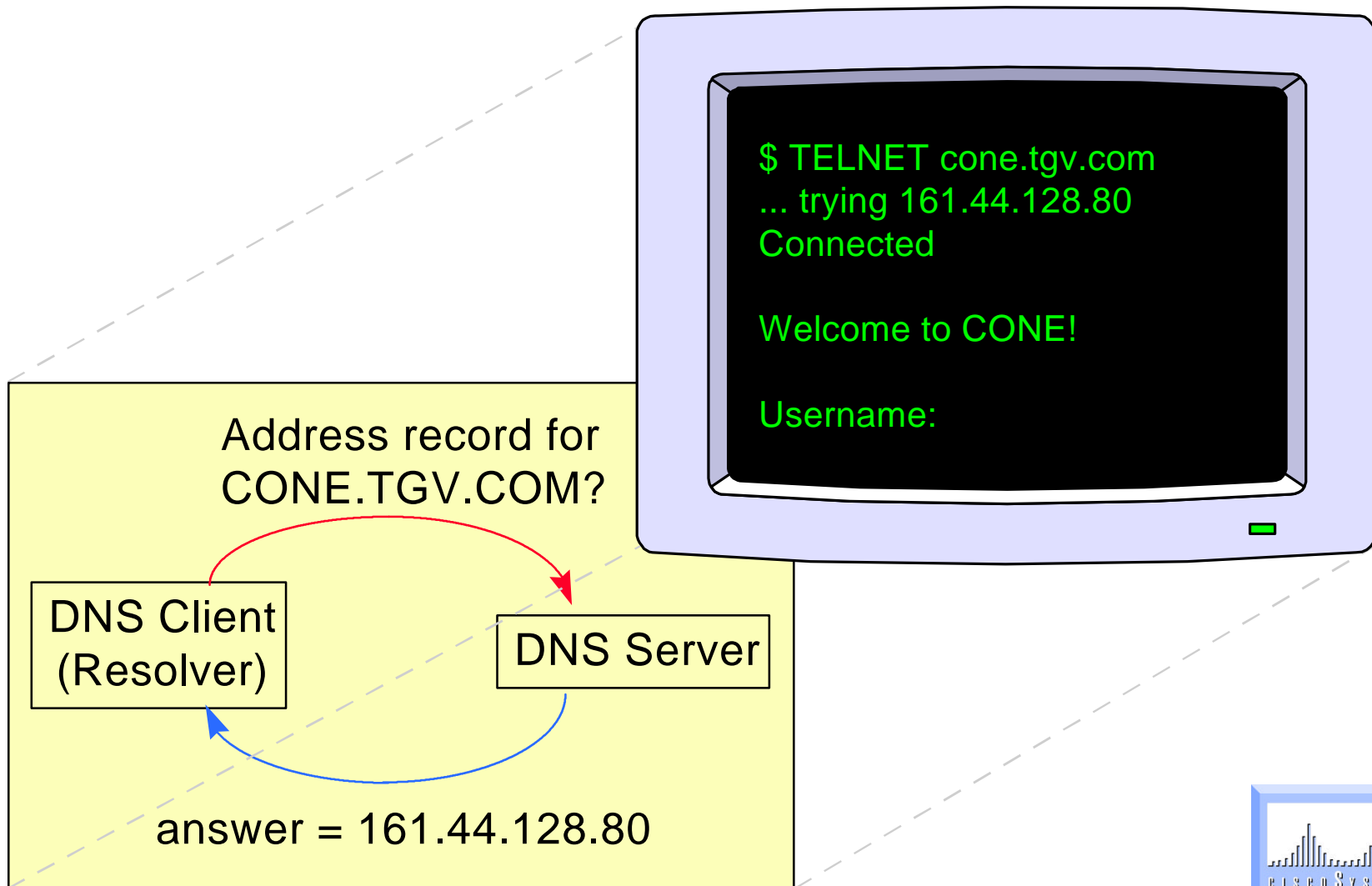
Nameservers

host.department.organization.domain

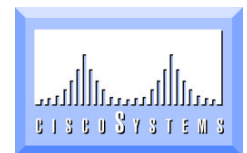
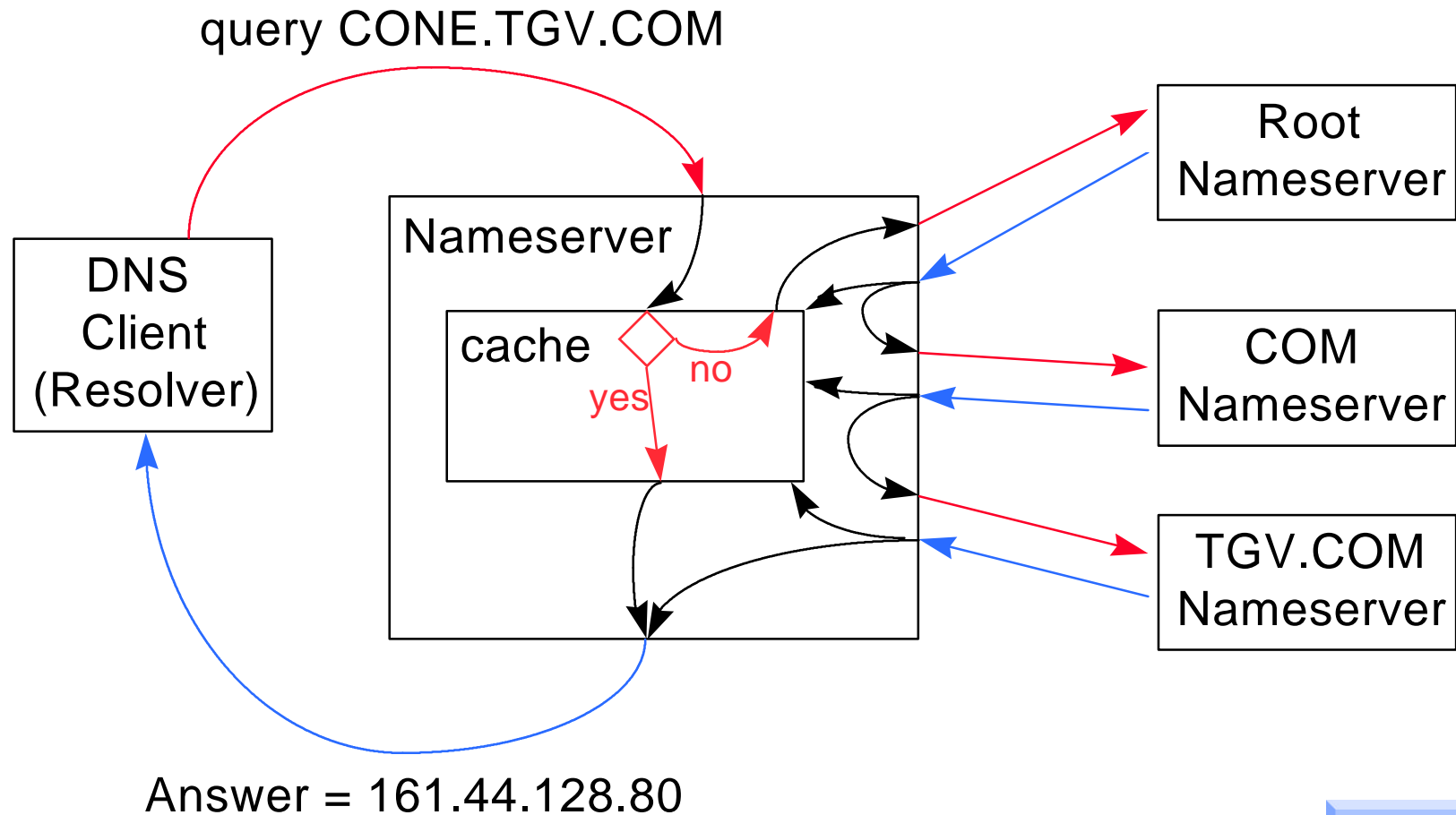
- **Nameservers hold the DNS data or know how to find the answer**
- **Each “dot” separates a subdomain**
- **Each subdomain may have a nameserver associated with it that has the DNS data**



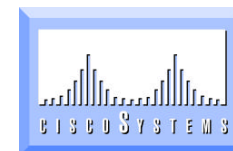
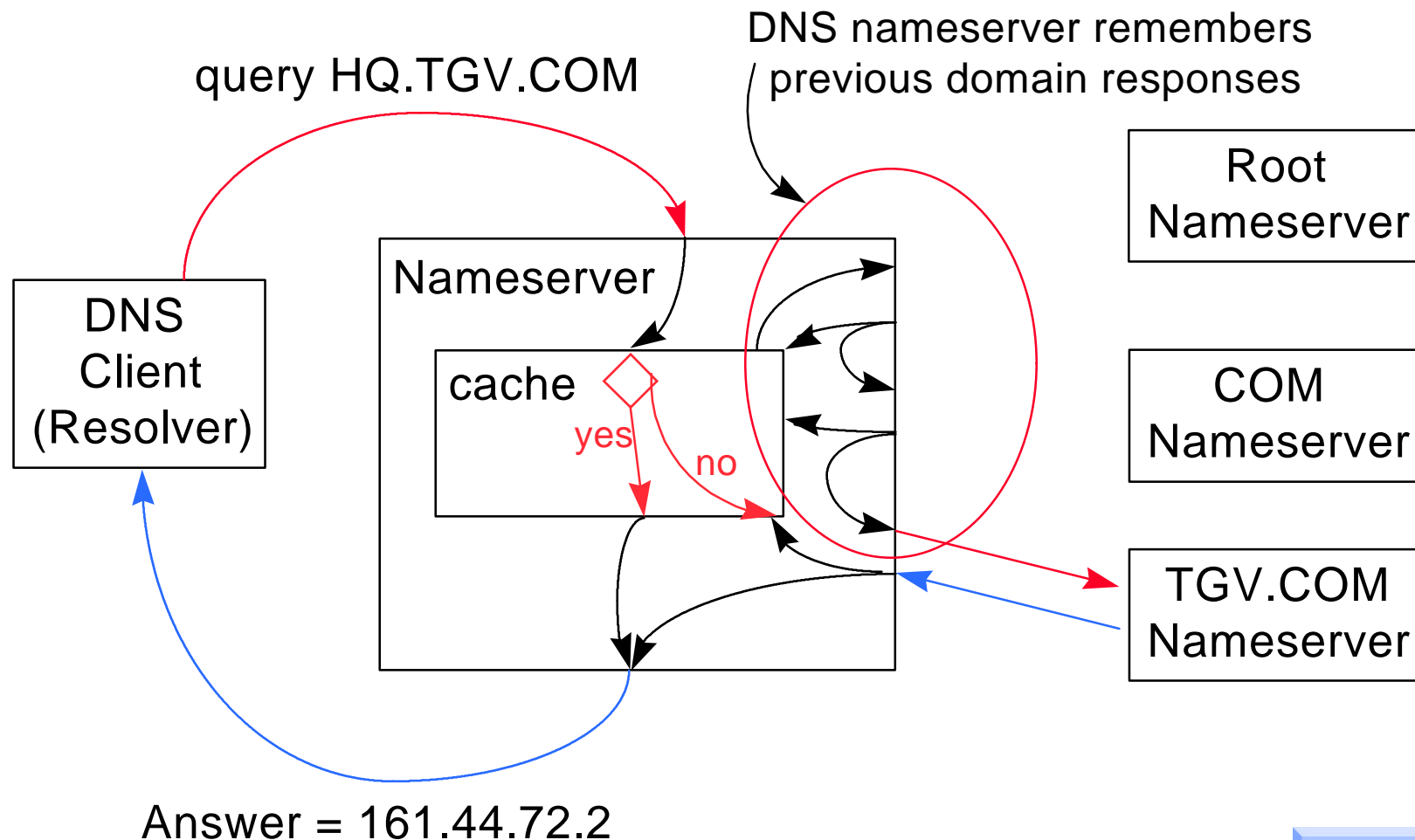
► Tasks of DNS Client (Resolver)



► Tasks of the DNS server...



► If partial answer is known...

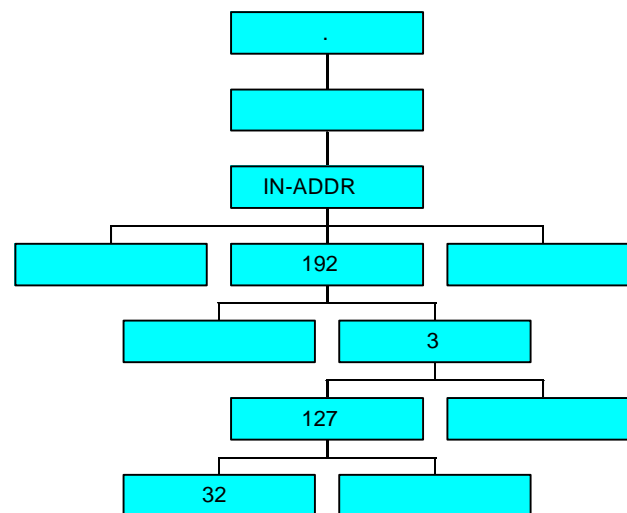


► “Reverse” lookups

- **Maintain right-to-left parsing**

Most generic to the right

Most specific to the left



- **Reverse IP Addresses**

PTR records

least specific most specific

192.3.127.32 → 32.127.3.192.in-addr.arpa

161.44.128.70 → 70.128.44.161.in-addr.arpa

most specific

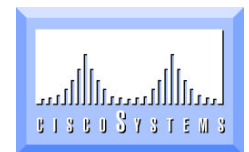
least specific

cone.tgv.com



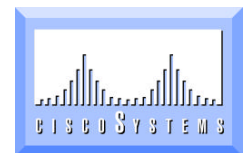


DNS Nameservers



Types of nameservers

- **Root nameserver**
- **Primary nameserver**
- **Secondary nameserver**
- **Caching-only nameserver**
- **Forwarder**
- **Slave**

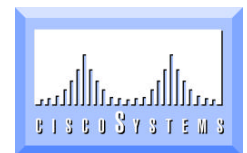




Root Nameservers

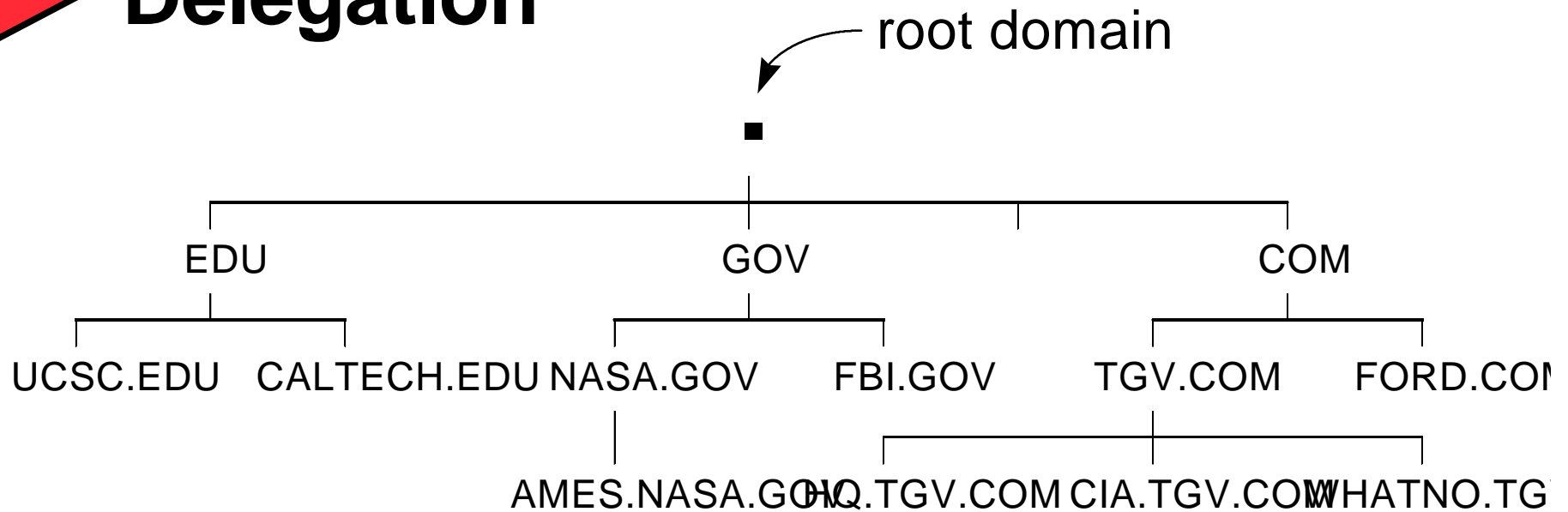
- **Authoritative for root (“.”) domain**
- **Responsible for COM, EDU, GOV, ARPA, IE, US, DE, and other top-level domains, including IN-ADDR.ARPA**
- **Not one of your nameservers**

Unless you are not connected to the Internet

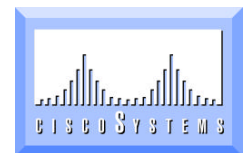




Delegation



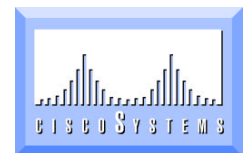
- **Delegation is giving part of a zone to another nameserver**
- **Permits decentralized administration**
Decentralization is DNS's power
- **Delegation can be done at any “.”**





Primary Nameserver

- **Authoritative for a zone**
- **Configuration file (bootfile) identifies the database files with the resource records**





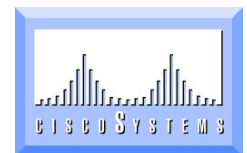
Primary Nameserver

Configuration file

```
cache .  
primary 0.0.127.in-addr.arpa  
primary tgv.com  
primary 44.161.in-addr.arpa
```

```
domain-name-service.cache  
domain-name-service.local  
domain-name-service.tgv  
domain-name-service.tgv-net
```

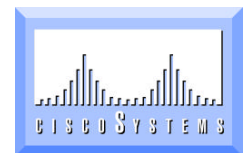
zones



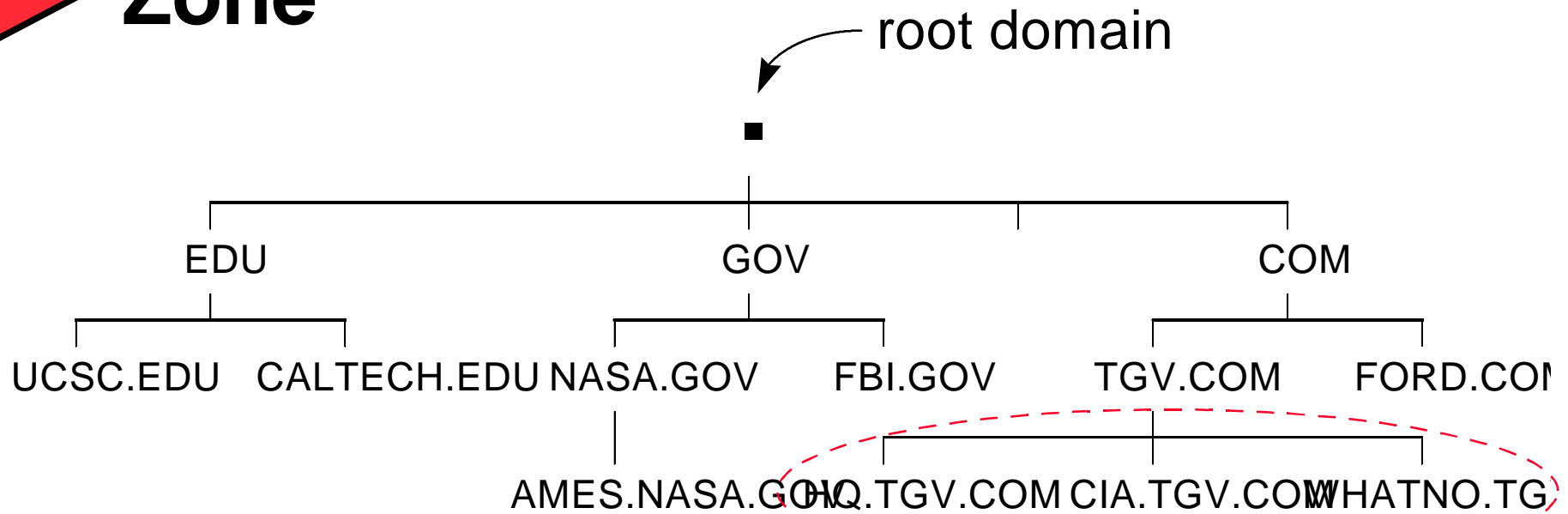


Secondary Nameserver

- **Authoritative for a zone**
- **Automatically loads data from Primary**
- **Data is NOT maintained on the secondary nameserver**
- **A backup datafile may be created and used for occasions when the primary nameserver is unavailable**



Zone



- **A 'piece' of a domain**

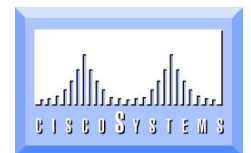
Such as tgv.com, sales.tgv.com, enet.dec.com

- **Zone file**

Datafile that describes a zone

Contains the resources records

Maintained on the primary nameserver





Sample Zone file

```
tgvl.com.  in  soa  vaxa.tgv.com. wing.tgv.com. (
                                199501091 ; serial number
                                10800      ; refresh 3 hr
                                3600       ; retry 1 hr
                                604800    ; expire 1 wk
                                86400)    ; min. TTL 1 day
```

```
tgvl.com.  in  ns  ns1.tgv.com.
```

```
ns1.tgv.com.  in  a      161.44.128.70
```

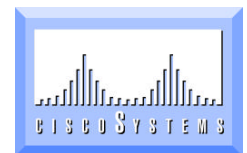
```
hq.tgv.com.   in  a      161.44.128.70
```

```
              in  hinfo  VAXSTATION-4000-90 VMS
```

```
tgvl.com.     in  mx 10  hq.tgv.com.
```

```
fang.tgv.com. in  a      161.44.128.87
```

```
              in  mx 10  hq.tgv.com.
```





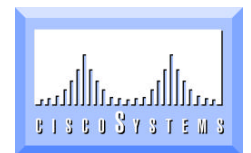
Zone Transfer

- **The act of transferring a zone**
- **Typically from a Primary to Secondary**

**Secondary checks SOA on Primary every
REFRESH seconds**

Automatically

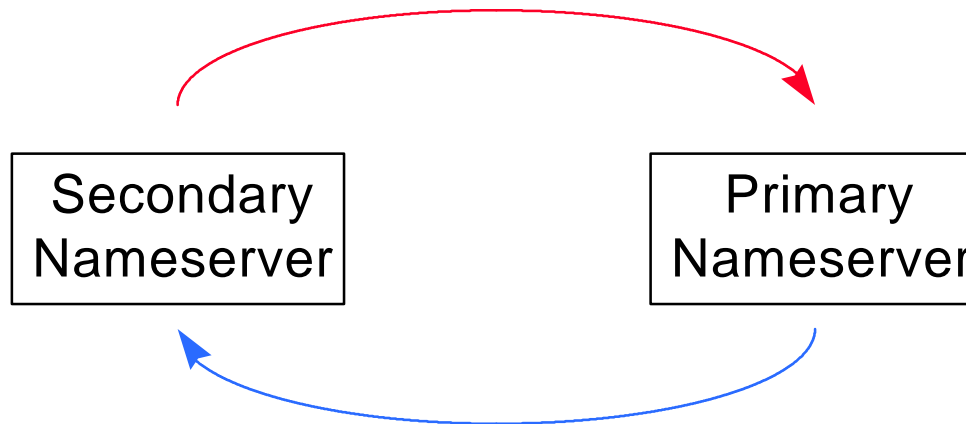
**If serial number on Primary is higher,
secondary gets copy of zone file**



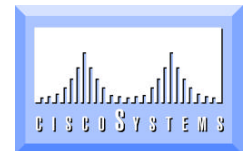


Secondary Asks for SOA

SOA for zone TGV.COM?

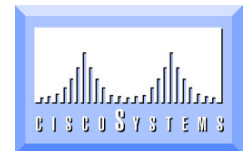
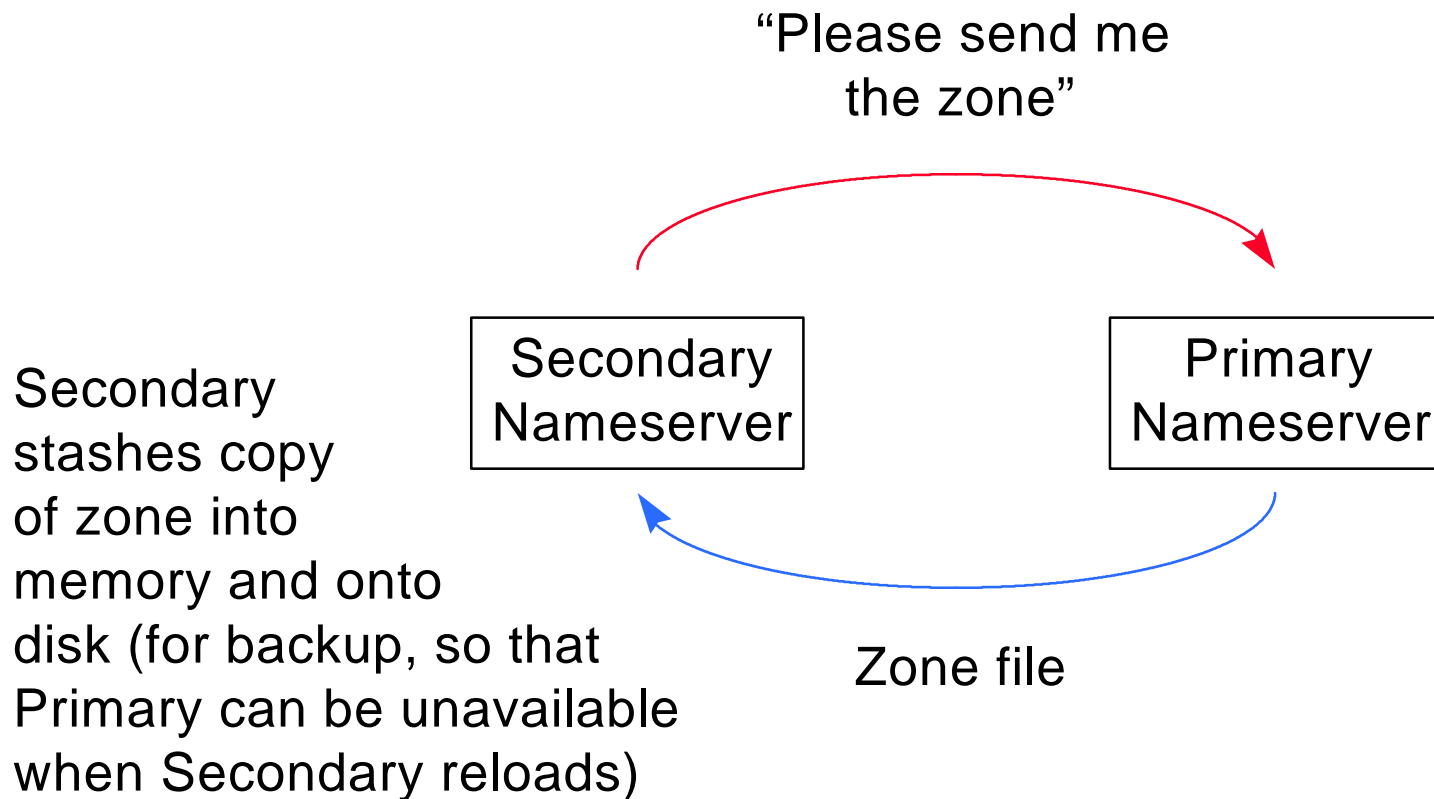


SOA = vaxa.tgv.com system.vaxa.tgv.com
95090101 10800 3600 604800 86400





Secondary Performs Zone Transfer





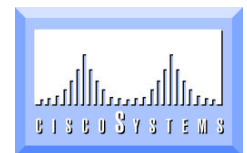
Secondary Nameserver Configuration file

```
cache      .                domain-name-service.cache
primary    0.0.127.in-addr.arpa domain-name-service.local
secondary  tgv.com          161.44.128.70  dns.tgv-bkp
secondary  44.161.in-addr.arpa 161.44.128.70  dns.tgv-net-bkp
```

← zones

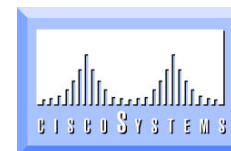
← backup files

161.44.128.70 = IP Address of primary nameserver for this zone



Caching-only Nameserver

- **Responds to DNS resolver queries**
- **Caches answers**
- **Improves performance**
- **Recommended default DNS configuration**
- **Does not contain local DNS information**
(except for localhost, net#.in-addr.arpa)

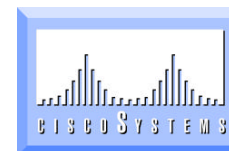




Caching-only Nameserver

Configuration file

```
cache . domain-name-service.cache  
primary 0.0.127.in-addr.arpa domain-name-service.local
```





Forwarder

- If answer is not in cache, send query to Forwarder

Not necessary for DNS to function

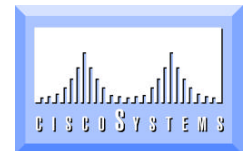
Improves performance

- If Forwarder doesn't respond, act normally

Send query to root nameservers

```
cache      .                domain-name-service.cache
primary    0.0.127.in-addr.arpa domain-name-service.local

forwarder  161.44.128.70
```

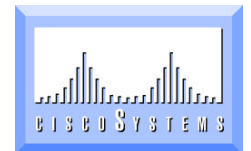


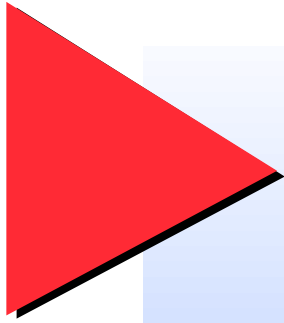
Slave

- Prevents communication to root nameservers
- Useful when behind firewall
- Forwarder is required

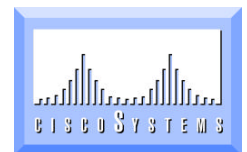
```
cache      .                domain-name-service.cache
primary    0.0.127.in-addr.arpa domain-name-service.local
```

```
forwarder  161.44.128.70
slave
```





DNS Hints





Location of Nameservers

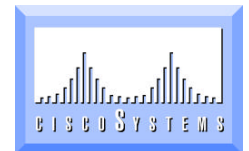
- **Your IP network relies on nameserving**
Nameservers must be accessible and running

- **2-3 nameservers best**

two on-site, one off-site

**Configure clients to know about a local
nameserver and a remote nameserver**

**Many IP implementations make it awkward to
configure clients to use more than one
nameserver**





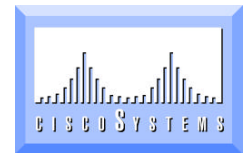
No Internet Connection

- **Pseudo-root nameserver necessary for non-internet connected sites**

Because DNS needs root nameservers to function

- **CACHE file must be modified on all nameservers**

Or possible nameserver corruption can occur





NSLOOKUP

- **NSLOOKUP requires lowercase commands**

Verify DNS information

```
$ multinet nslookup
```

```
Default Server:  LOCALHOST
```

```
Address:  127.0.0.1
```

```
> set query=any
```

```
> cone.tgv.com
```

```
Server:  LOCALHOST
```

```
Address:  127.0.0.1
```

```
cone.tgv.com      canonical name = Cone-Of-Silence.TGV.COM
```

```
TGV.COM nameserver = NS1.TGV.COM
```

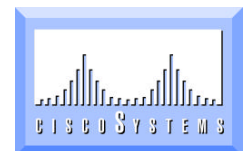
```
TGV.COM nameserver = NS2.TGV.COM
```

```
TGV.COM nameserver = EQL.Caltech.Edu
```

```
NS1.TGV.COM      internet address = 161.44.72.2
```

```
NS2.TGV.COM      internet address = 161.44.224.2
```

```
EQL.Caltech.Edu internet address = 131.215.29.1
```





NSLOOKUP, Cont.

- **Verify reverse name mapping**

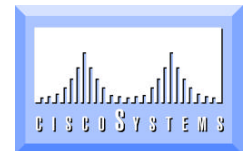
```
> set query=ptr
```

```
> 161.44.192.51
```

```
Server:  LOCALHOST
```

```
Address: 127.0.0.1
```

```
51.192.44.161.in-addr.arpa      name = Cone-Of-Silence.TGV.COM
44.161.IN-ADDR.ARPA            nameserver = NS1.TGV.COM
44.161.IN-ADDR.ARPA            nameserver = NS2.TGV.COM
44.161.IN-ADDR.ARPA            nameserver = EQL.Caltech.Edu
NS1.TGV.COM                     internet address = 161.44.72.2
NS2.TGV.COM                     internet address = 161.44.224.2
EQL.Caltech.Edu                internet address = 131.215.29.1
```





DNS Myths

- **1. Configuration file needs FORWARDER**

False: not required

DNS works by going down DNS tree

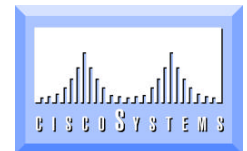
- **2. CACHE file contains “my” nameservers**

False: It contains the root nameservers

Your nameservers are found by going down DNS tree

- **3. Using 127.0.0.1 for resolver won't work**

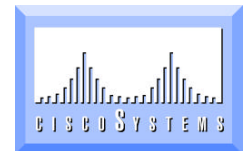
True: It is the best configuration if your system is a nameserver itself (and is the only way to get caching)





Common configuration errors

- **Syntax errors**
 - No trailing “.” when needed
 - Trailing “.” inserted when it shouldn’t be
- **Pointer records (reverse lookups) are often forgotten**
- **Serial number is not increased when changes are made**
- **Records pointing to configuration files are not accurate**





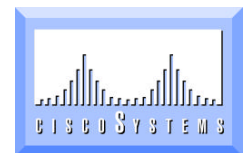
“The” book on DNS

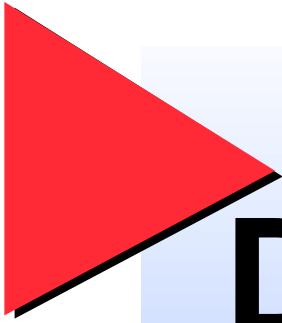
- **DNS and BIND in a Nutshell**

By Paul Albitz and Cricket Liu

Published by O'Reilly & Associates

300+ pages. Excellent reference.



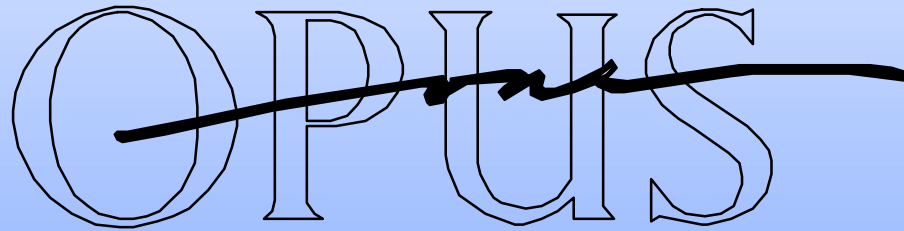


Domain Name System Technology Overview

(DNS and Bind)



Domain Name Service (DNS) Troubleshooting



Jan Trumbo

trumbo@Opus1.COM

DNS Reference book

- ❖ DNS and BIND in a Nutshell
 - ❖ Written by Paul Albitz and Cricket Liu
 - ❖ Published by O'Reilly & Associates
 - ❖ Copyright 1992
- ❖ New edition due December 1996
 - ❖ www.ora.com for ordering info

DNS Terminology

Terminology Roadmap

- ❖ Zone

 - ❖ Zone file

 - ❖ Zone transfer

- ❖ Authoritative

- ❖ Root nameserver

- ❖ Delegation (Nameserver Delegation)

- ❖ Resource Records

- ❖ A 'piece' of a domain
 - ❖ Such as tgv.com, sales.tgv.com
- ❖ Zone file
 - ❖ Datafile that describes a zone
- ❖ Zone transfer
 - ❖ sending zone file from primary to secondary

DECUS

Fall 1996

Anaheim

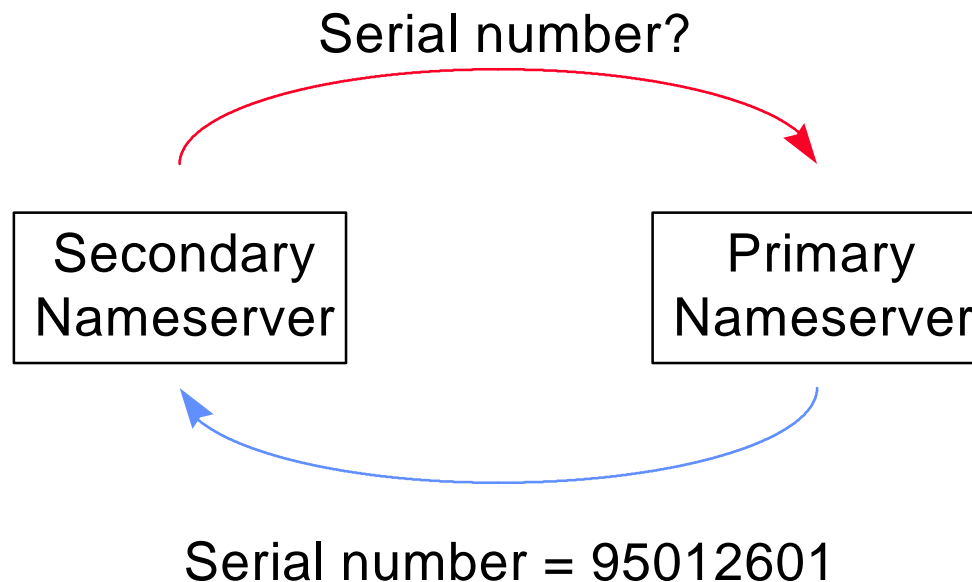
Example Zone file

```
@ in soa vaxa.tgv.com. wing.tgv.com. (  
    9501091 ; serial number  
    10800   ; refresh 3 hr  
    3600    ; retry 1 hr  
    604800  ; expire 1 wk  
    86400)  ; min. TTL 1 day  
  
@ in ns ns1.tgv.com.  
ns1.tgv.com. in a 161.44.128.70  
hq.tgv.com. in a 161.44.128.70  
tgv.com. in mx 10 hq.tgv.com.  
fang.tgv.com. in a 161.44.128.87
```

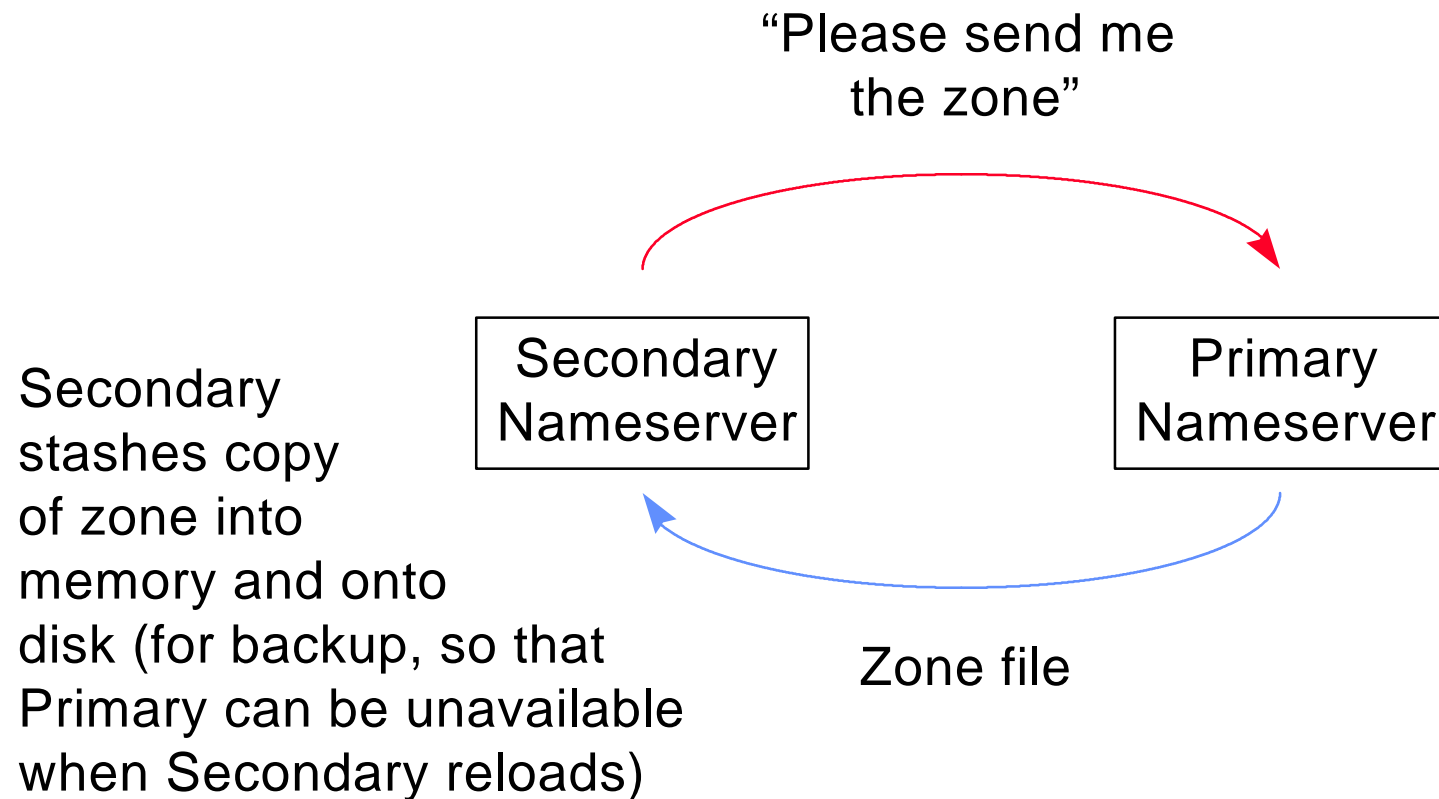
Zone Transfer

- ❖ The act of transferring a zone
- ❖ Typically from a Primary to Secondary
 - ❖ Secondary checks SOA on Primary
 - ❖ The Secondary “pulls” the file
- ❖ NSLOOKUP (debugging tool) also does zone transfers (using `ls -d`)
- ❖ Performed over TCP port 53

Secondary Asking for Serial Number



Secondary Performs Zone Transfer



Authoritative Answer

- ❖ Authoritative Answer bit is set on answer
 - ❖ Typically only seen with NSLOOKUP
 - ❖ Useful for debugging
- ❖ Only set by Primary or Secondary
- ❖ Indicates the nameserver thinks it is authoritative for the zone

Root Nameservers

- ❖ Authoritative for root (".") domain
- ❖ Responsible for COM, EDU, GOV, and other top-level domains
- ❖ Typically, not one of your nameservers
 - ❖ Unless you've set up pseudo- (fake-) root nameserver

Root Nameservers

- ❖ Don't perform recursive queries
 - ❖ They cannot get answers for you
 - ❖ Only point you to other nameservers
 - ❖ This reduces their load
- ❖ Initial list of roots is in CACHE entry
 - ❖ Up-to-date list of root nameservers is automatically obtained and used
 - ◆ However, on-disk CACHE file isn't changed

Delegation

- ❖ Delegation is giving part of a zone to another nameserver
- ❖ Permits decentralized administration
- ❖ Delegation can be done at any subdomain, and can be done to any arbitrary depth

Resource Records

❖ Data in zone file that describes the zone

❖ A

❖ PTR

❖ MX

❖ NS

❖ SOA

❖ HINFO

❖ WKS

❖ TXT

❖ CNAME

Resource Record Overview

❖ Zone

- ❖ Required: SOA, NS, A

❖ Host (forward lookup)

- ❖ Required: A

- ❖ Optional: MX, WKS, HINFO, (CNAME)

❖ Host (inverse lookups)

- ❖ Required: PTR

Multiple Resource Records

❖ Multiple Resource Records are legal

optional →

tgvr-router.tgv.com.	in a	161.44.128.1
tgvr-router.tgv.com.	in a	161.44.129.1
tgvr-router.tgv.com.	in a	161.44.130.1
tgvr.com.	in mx 10	hq.tgv.com.
tgvr.com.	in mx 10	cone.tgv.com.
tgvr.com.	in mx 20	fang.tgv.com.

❖ Useful for multi-homed hosts, or for hosts that have duplicate features

A Record

- ❖ Address record
- ❖ Hostname to IP address mapping

```
cone.tgv.com.    in a    161.44.128.98
hq.tgv.com.      in a    161.44.128.70
```

PTR Record

- ❖ Pointer record - also called 'inverse'
- ❖ IP address to hostname mapping
- ❖ Often incorrectly configured, or forgotten
- ❖ Required for some services to function
 - ❖ "r" Services, some FTP Servers

- ❖ PTRs are what causes hostname in
SHOW USERS/FULL display

- ❖ Uses 'inverted' IP addresses:

71.128.44.161.in-addr.arpa.	in ptr	hq.tgv.com.
72.128.44.161.in-addr.arpa.	in ptr	fang.tgv.com.
73.128.44.161.in-addr.arpa.	in ptr	tide.tgv.com.
74.128.44.161.in-addr.arpa.	in ptr	wash.tgv.com.

MX Record

- ❖ Mail exchanger record
- ❖ Directs mail to a host
- ❖ Can help provide simpler Email address
- ❖ Preference value
 - ❖ Crude load balancing
 - ❖ Can cause mail to spool at nearby system

tgvl.com. in mx 10 cad.tgv.com.

tgvl.com. in mx 10 hq.tgv.com.

tgvl.com. in mx 20 fang.tgv.com.

NS Record

- ❖ Nameserver record
- ❖ Lists nameservers for the zone
- ❖ Should agree with parent domain
- ❖ “Glue” records needed for names in same domain
 - ❖ Nameservers need “A” records

```
tgv.com.    in ns    ns1.tgv.com.
```

SOA Record

- ❖ Start of Authority record
- ❖ Indicates zone-wide information:
 - ❖ originating system for zone information
 - ◆ (typically the primary nameserver)
 - ❖ Email address of DNS administrator
 - ❖ Various numbers and times ...

Example SOA Record

```
@   in   soa   vaxa.tgv.com. wing.tgv.com. (  
                                9501091   ; serial number  
                                10800      ; refresh 3 hr  
                                3600       ; retry 1 hr  
                                604800     ; expire 1 wk  
                                86400)    ; min. TTL 1 day
```

SOA Record Fields

❖ Serial number

- ❖ Identifies the 'version' of the zone file
- ❖ Higher number means newer version
 - ◆ And causes zone transfers with Secondaries

❖ Refresh time (seconds)

- ❖ How often Secondary checks the Primary's serial number
- ❖ If serial number is higher, transfer zone

❖ Retry time (seconds)

- ❖ If unable to check serial number at Refresh time, keep retrying every Retry seconds

More SOA Record Fields

- ❖ Expire time (seconds)
 - ❖ How long Secondary remembers data if unable to do zone transfer with Primary
 - ❖ If exceeded, Secondary forgets everything about this zone
- ❖ Minimum Time-To-Live (seconds)
 - ❖ Also called “Default TTL”
 - ❖ Default time a caching nameserver can cache a Resource Record
 - ❖ Can be overridden on specific Resource Record

HINFO Record

- ❖ Host Information record
- ❖ Human-readable information
 - ❖ Usually Hardware type, Operating system
- ```
cone.tgv.com. in hinfo vax vms
```

```
whirr.tgv.com. in hinfo pc windows
```
- ❖ Don't record OS version - it will always be out of date

# WKS Record

---

- ❖ Well Known Service record
- ❖ Originally intended to indicate which services run on the host (FTP, TELNET, SMTP, etc.)
- ❖ Not consulted by any client applications
- ❖ Not very useful

# TXT Record

---

- ❖ Text record
- ❖ Human-readable free-form information
- ❖ Location, owner, or humor

`wade.tgv.com. in txt building-3`

`cad.tgv.com. in txt "Contrived Accident"`

`cad.tgv.com. in txt "Division"`

# CNAME Record

---

- ❖ Canonical name record
- ❖ Host alias name
- ❖ Useful when renaming host, or host has several functions

|                              |                       |                              |
|------------------------------|-----------------------|------------------------------|
| <code>www.tgv.com.</code>    | <code>in cname</code> | <code>zaphod.tgv.com.</code> |
| <code>gopher.tgv.com.</code> | <code>in cname</code> | <code>zaphod.tgv.com.</code> |
| <code>ftp.tgv.com.</code>    | <code>in cname</code> | <code>hq.tgv.com.</code>     |

- ❖ Not recommended for mail aliases

# Terminology

## Key Concepts

---

### ❖ Resource Records

#### ❖ Zone

- ◆ Required: SOA, NS, A

#### ❖ Host (forward lookup)

- ◆ Required: A
- ◆ Optional: MX, WKS, HINFO, (CNAME)

#### ❖ Host (inverse lookups)

- ◆ Required: PTR

# DNS Troubleshooting

# Query types

---

- ❖ Any of the Resource Records can be queried
  - ❖ A, PTR, MX, SOA, TXT, etc.
  - ❖ Can also send an “any” query
    - ◆ Returns contents of cache
- ❖ Non-recursive query
  - ❖ Useful for debugging - causes nameserver to only return information from its cache



# Answers

---

- ❖ Servers may return additional records
  - ❖ MX query returns MX answers and A records
  - ❖ Non-authoritative servers return NS records

# Debugging

---

- ❖ DNS & BIND book is very useful
  - ❖ Detailed troubleshooting in chapter 12
  - ❖ Information on various DNS configurations
  - ❖ Surviving outages to DNS server (p 175)
- ❖ Firewalls can cause interesting DNS behaviors
  - ❖ Check from 'both sides' of a firewall

# DNS Troubleshooting Using NSLOOKUP

---

## ❖ Use NSLOOKUP

- ❖ DNS & BIND, Chapter 10
- ❖ Can tell if nameserver is authoritative
- ❖ Can't tell Primary from a Secondary
- ❖ Available on Unix, VMS, many PCs

# Using NSLOOKUP

---

- ❖ Use lowercase with NSLOOKUP
- ❖ Only single-line command recall

```
$ multinet nslookup
```

```
Server: LOCALHOST
```

```
Address: 127.0.0.1
```

```
>
```

DECUS

Fall 1996

Anaheim

# NSLOOKUP Appends the Default Domain

```
$ show log *domain*
"MULTINET_SEARCHDOMAINS" = "Opus1.COM"
```

```
$ mu nsl
Default Server: LOCALHOST
Address: 127.0.0.1
```

```
> tennis
Server: LOCALHOST
Address: 127.0.0.1
```

```
Name: Tennis.Opus1.COM
Address: 192.245.12.2
```

```
> tennis.
Server: LOCALHOST
Address: 127.0.0.1
```

**Use final period to  
disable domain  
appends**

```
*** LOCALHOST can't find tennis.: Non-existent host/domain
```

DECUS

Fall 1996

Anaheim

# NSLOOKUP Can Query Other Nameservers

```
$ mu nslookup
Default Server: LOCALHOST
Address: 127.0.0.1
```

Queries the local  
resolver by default

>

```
> server ns.opus1.com
Default Server: ns.Opus1.COM
Address: 192.245.12.50
```

Use the 'server'  
command to send  
queries elsewhere

>

DECUS

Fall 1996

Anaheim

# Find the Right Server From Whois Database

```
$ whois dom opusone.com
Opus One (OPUSONE-DOM)
 1404 East Lind Road
 Tucson, AZ 85719
```

Domain Name: OPUSONE.COM

Administrative Contact:

Julietta, Romeo (RJ9) Romeo\_Julietta@LOGIN.COM  
(602) 324-0494

Technical Contact, Zone Contact:

Snyder, Joel M. (JMS56) Joel\_M\_Snyder@OPUS1.COM  
+1 520 324 0494 (FAX) +1 520 324 0495 (FAX) +1 520 324 0495

Record last updated on 25-Oct-96.

Record created on 21-Jan-95.

Domain servers in listed order:

|              |                 |
|--------------|-----------------|
| NS.OPUS1.COM | 192.245.12.50   |
| ARIZONA.EDU  | 128.196.128.233 |

DECUS

Fall 1996

Anaheim

# Or, Find Servers From the Root Servers

```
$ mu nsl
Default Server: LOCALHOST
Address: 127.0.0.1

> set type=ns
> server d.root-servers.net
Default Server: D.ROOT-SERVERS.NET
Address: 128.8.10.90

> opusone.com
Server: D.ROOT-SERVERS.NET
Address: 128.8.10.90
```

Authoritative answers can be found from:

|              |                                    |
|--------------|------------------------------------|
| opusone.com  | nameserver = NS.OPUS1.COM          |
| opusone.com  | nameserver = ARIZONA.EDU           |
| NS.OPUS1.COM | internet address = 192.245.12.50   |
| ARIZONA.EDU  | internet address = 128.196.128.233 |



DECUS

Fall 1996

Anaheim

# Primary and Secondary Look Alike

```
> set type=soa
> server ns.opus1.com
Default Server: ns.Opus1.COM
Address: 192.245.12.50
```

```
> opusone.com.
Server: ns.Opus1.COM
Address: 192.245.12.50
```

```
OpusOne.COM
 origin = NS.Opus1.COM
 mail addr = hostmaster.Opus1.COM
 serial = 1996110800
 refresh = 86400 (1 days)
 retry = 7200 (2 hours)
 expire = 2592000 (30 days)
 minimum ttl = 604800 (7 days)
```

**Notice we disable  
domain appending to  
prevent unnecessary  
thrashing**

DECUS

Fall 1996

Anaheim

# ... Or Is This One The Primary?

```
> server arizona.edu
```

```
Default Server: ARIZONA.EDU
```

```
Addresses: 128.196.128.234, 128.196.128.233
```

```
> opusone.com.
```

```
Server: ARIZONA.EDU
```

```
Addresses: 128.196.128.234, 128.196.128.233
```

OpusOne.COM

origin = NS.Opus1.COM

mail addr = hostmaster.Opus1.COM

serial = 1995072804

refresh = 86400 (1 days)

retry = 7200 (2 hours)

expire = 2592000 (30 days)

minimum ttl = 604800 (7 days)

Only the DNS  
administrators know for  
sure ... furthermore, it  
doesn't matter to you!

# Lame Delegations

---

- ❖ A 'Lame Delegation' occurs when a zone has been delegated to a nameserver, and that nameserver is not authoritative for the zone - i.e. no SOA record
- ❖ The most common DNS problem
- ❖ Results from lack of communication between DNS managers

# Tracing Lame Delegations

---

```
> server d.root-servers.net
Default Server: d.root-servers.net
Address: 128.8.10.90
```

```
> set type=ns
> aspect-ts.com.
Server: d.root-servers.net
Address: 128.8.10.90
```

Non-authoritative answer:

```
aspect-ts.com nameserver = NS1.ACES.COM
aspect-ts.com nameserver = NS.OPUS1.COM
```

Authoritative answers can be found from:

```
NS1.ACES.COM internet address = 192.195.240.1
NS.OPUS1.COM internet address = 192.245.12.50
```

DECUS

Fall 1996

Anaheim

# Now Ask That Server

---

```
> server ns1.aces.com
```

```
Default Server: ns1.ACES.COM
```

```
Address: 192.195.240.1
```

```
> set type=soa
```

```
> aspect-ts.com.
```

```
Server: ns1.ACES.COM
```

```
Address: 192.195.240.1
```

```
*** ns1.ACES.COM can't find aspect-ts.com: Non-existent
host/domain
```

Whoops!

DECUS

Fall 1996

Anaheim

# A and PTR Mismatches

```
> set type=a
> compurad.com.
Server: ns.opus1.com
Address: 192.245.12.50
```

```
Name: compurad.COM
Address: 204.153.44.5
```

```
> set type=ptr
> 204.153.44.5
Server: ns.opus1.com
Address: 192.245.12.50
```

```
5.44.153.204.IN-ADDR.ARPA name = s5.204-153-44-NET.AccessOne.NET
44.153.204.IN-ADDR.ARPA nameserver = NS.Opus1.COM
44.153.204.IN-ADDR.ARPA nameserver = NS1.ACES.COM
NS.Opus1.COM internet address = 192.245.12.50
NS1.ACES.COM internet address = 192.195.240.1
```

Some applications care, some don't. This is NOT necessarily a problem!

NSLOOKUP does the work of reversing the IP number for us with type=ptr

DECUS

Fall 1996

Anaheim

# Mail Looks at MX Records

---

```
> server ns.opus1.com
```

```
Default Server: ns.Opus1.COM
```

```
Address: 192.245.12.50
```

```
> set type=any
```

```
> mail.opusone.com
```

```
Server: ns.Opus1.COM
```

```
Address: 192.245.12.50
```

```
Mail.OpusOne.COM
```

```
preference = 10, mail exchanger =
```

```
mail.opus1.COM
```

DECUS

Fall 1996

Anaheim

# But an MX to an MX is Not What You Think It Is

```
Mail.OpusOne.COM preference = 10, mail exchanger =
mail.opus1.COM
> mail.opus1.com
```

```
Server: ns.Opus1.COM
Address: 192.245.12.50
```

```
Mail.Opus1.COM text = "Where Opus One gets mail"
Mail.Opus1.COM preference = 10, mail exchanger = Cello.Opus1.COM
Mail.Opus1.COM preference = 15, mail exchanger = Tennis.Opus1.COM
Mail.Opus1.COM preference = 20, mail exchanger = Piano.Opus1.COM
Mail.Opus1.COM preference = 30, mail exchanger = Arizona.EDU
Mail.Opus1.COM internet address = 192.245.12.7
Opus1.COM nameserver = ns.Opus1.COM
Opus1.COM nameserver = Arizona.EDU
Cello.Opus1.COM internet address = 192.245.12.7
Tennis.Opus1.COM internet address = 192.245.12.2
Piano.Opus1.COM internet address = 192.245.12.69
Arizona.EDU internet address = 128.196.128.233
ns.Opus1.COM internet address = 192.245.12.50
```

This is all you're  
pointing to



DECUS

Fall 1996

Anaheim

# You Can't Mail to a CNAME

```
$ mu ns1
Default Server: LOCALHOST
Address: 127.0.0.1
```

```
> set type=any
> smtp.opusone.com.
Server: LOCALHOST
Address: 127.0.0.1
```

**Bad, bad DNS  
Administrator!**

|                  |                                    |
|------------------|------------------------------------|
| smtp.OpusOne.COM | canonical name = mail.opus1.COM    |
| OpusOne.COM      | nameserver = ns.opus1.COM          |
| OpusOne.COM      | nameserver = NS1.ACES.COM          |
| OpusOne.COM      | nameserver = Arizona.EDU           |
| ns.opus1.COM     | internet address = 192.245.12.50   |
| NS1.ACES.COM     | internet address = 192.195.240.1   |
| Arizona.EDU      | internet address = 128.196.128.233 |

DECUS

Fall 1996

Anaheim

# TXT Records Are Worth Checking

```
> set type=any
```

```
> tgv.com.
```

```
Server: NS1.CISCO.COM
```

```
Address: 161.44.72.2
```

```
TGV.COM text = "Cisco Systems "
```

```
TGV.COM text = "Internet Business Unit"
```

```
TGV.COM text = "101 Cooper Street"
```

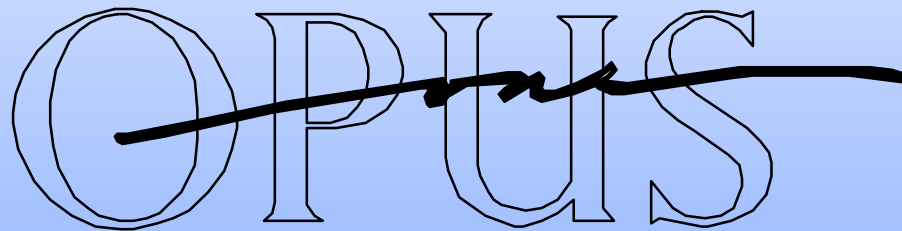
```
TGV.COM text = "Santa Cruz, CA 95060"
```

```
TGV.COM text = "(408) 457-5200 for main operator or sales
assistance"
```

```
TGV.COM text = "(408) 457-5201 or SERVICE@TGV.COM for technical
assistance"
```

```
TGV.COM text = "This zone is being maintained by the UBERserver"
```

# DNS Troubleshooting



Jan Trumbo

Trumbo@Opus1.COM

<ftp://ftp.opus1.com/decus/dns-trouble.powerpoint>