Demystifying DNS

Version 2.1 / July 10th, 2004

Allan P. Hurst

Technical Principal & Partner
Keep It Simple Computer Center / Newark, California USA

Who *is* this guy, anyway?

Allan Hurst / KIS Computer Center

Technical Principal, Enterprise Systems Group

- Has done this for a *very* long time. (Remembers punch cards & paper tape.)
- Specializes in network strategies, planning, re-architecting, and cleanup.
- With Dirk Smith, is one of "The Crash Dummies" … specialists in analyzing server crashes. (Come to our session on Thursday at 4:00 PM … "A Preventative Approach to Resolving Critical Server Issues", Session 3159 !)

Favorite Quote: "**It's never done THAT before!**"

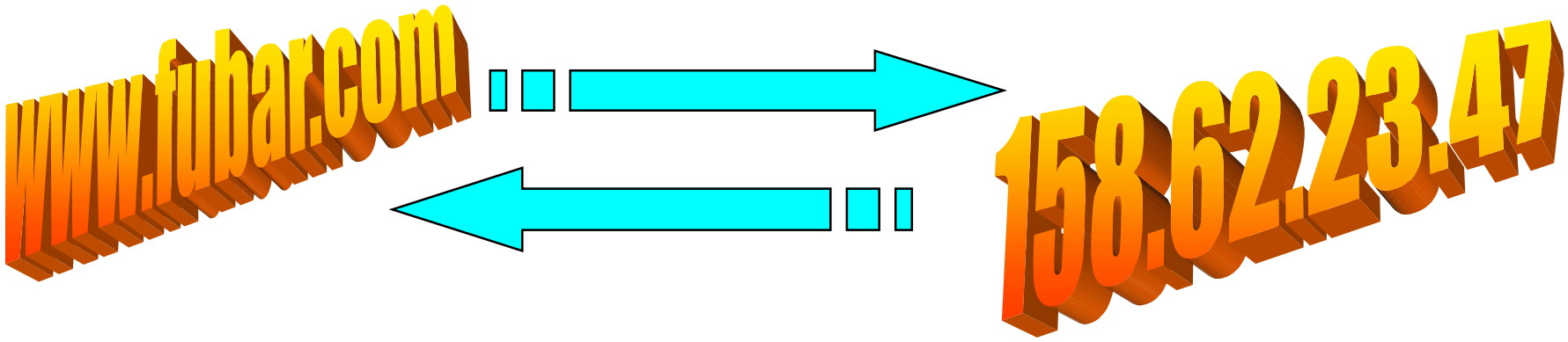
Who are you?

- This session is designed for network admins and/or managers.
- If you're a Novell shop, you probably haven't gone to Pure IP yet -- or you have, you're seeing weird problems.
- "Weird Problems" include sporadic errors when browsing the web, browsing the local network, or intermittent server communication and/or time sync errors.
- You have a manager who thinks "DNS" stands for "Don't Know Stuff".

What *is* “DNS”?

Stands for “Domain Name System”.

It’s just a database to match names with IP addresses …



… and match IP addresses with names.

What is a “domain”?

DNS is divided into Top Level Domains (TLDs).

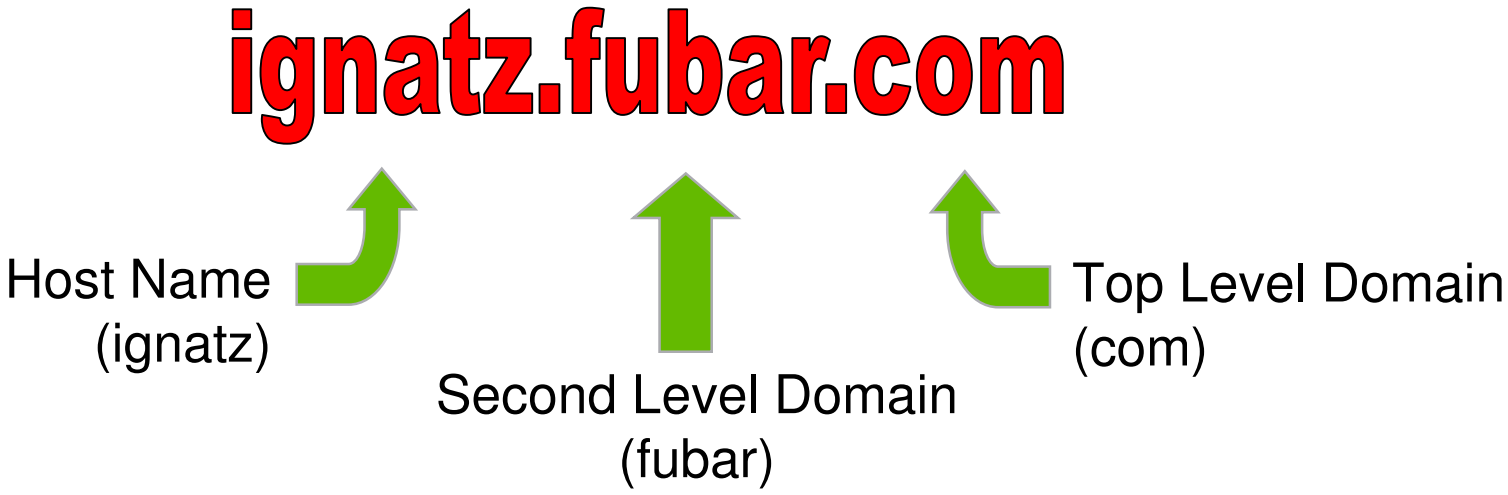
These are also known as "first level" domains.

The original TLDs were:

.com Commercial Enterprises
.org Noncommercial Organizations
.net Private Networks
.edu Educational Institutions
.mil Miltary Installations
.gov Government Installations
country names (.us, .ca, .uk, etc.)

Many new TLDs have been added since then, such as: **.info, .tv, .work, .to,** etcetera.

What is a Domain Name?



Consists of a Top Level Domain (TLD) and at least one second level domain.

The leftmost word is called a “host name”.

Can consist of a 3rd, 4th, 5th, etc. level … all the way to 127 levels. (bryant.wings.ignatz.fubar.com)

Where are domain names stored?

They're stored in a huge distributed database.

Each domain name is stored in something called a "zone file" … this is a text file designed to be processed by a program called "bind".

How are domain names stored?

The DNS database is comprised of many different types of records.

Some of the more common record types are:

- “SOA” – “Start Of Authority” – contains zone info
- “A” – contain IP address/name matching info
- “MX” – “Mail eXchange” – contains mail server info

(Note: There are many more record types than are shown here; these are just the ones you’re most likely to run into on a daily basis.)

DNS Record Types

Let's examine some of the DNS record types…

“Start of Authority” Record (SOA)

```
@    IN    SOA    ns1.fubar.com     hostmaster.fubar.com. (
     2002040101    ; Serial Number (this was created 4/1/02)
     7200          ; Refresh after two hours
     3600          ; Retry after an hour
     604800; Expire after one week
     86400 ; Minimum TTL )
```

SOA is the first record in the zone file.

Contains information about the zone in a series of fields.

Tells the server to be authoritative for the zone.

Defines the record management parameters for the zone.

“Start of Authority” Record (SOA)

```
@    IN    SOA    ns1.fubar.com    hostmaster.fubar.com. (
     2002040101   ; Serial Number (this was created 4/1/02)
     7200         ; Refresh after two hours
     3600         ; Retry after an hour
     604800       ; Expire after one week
     86400        ; Minimum TTL )
```

Serial Number = version of the zone file (typically, the date of creation)

Refresh = How often (in seconds) a secondary name server checks with the primary server responsible for this zone

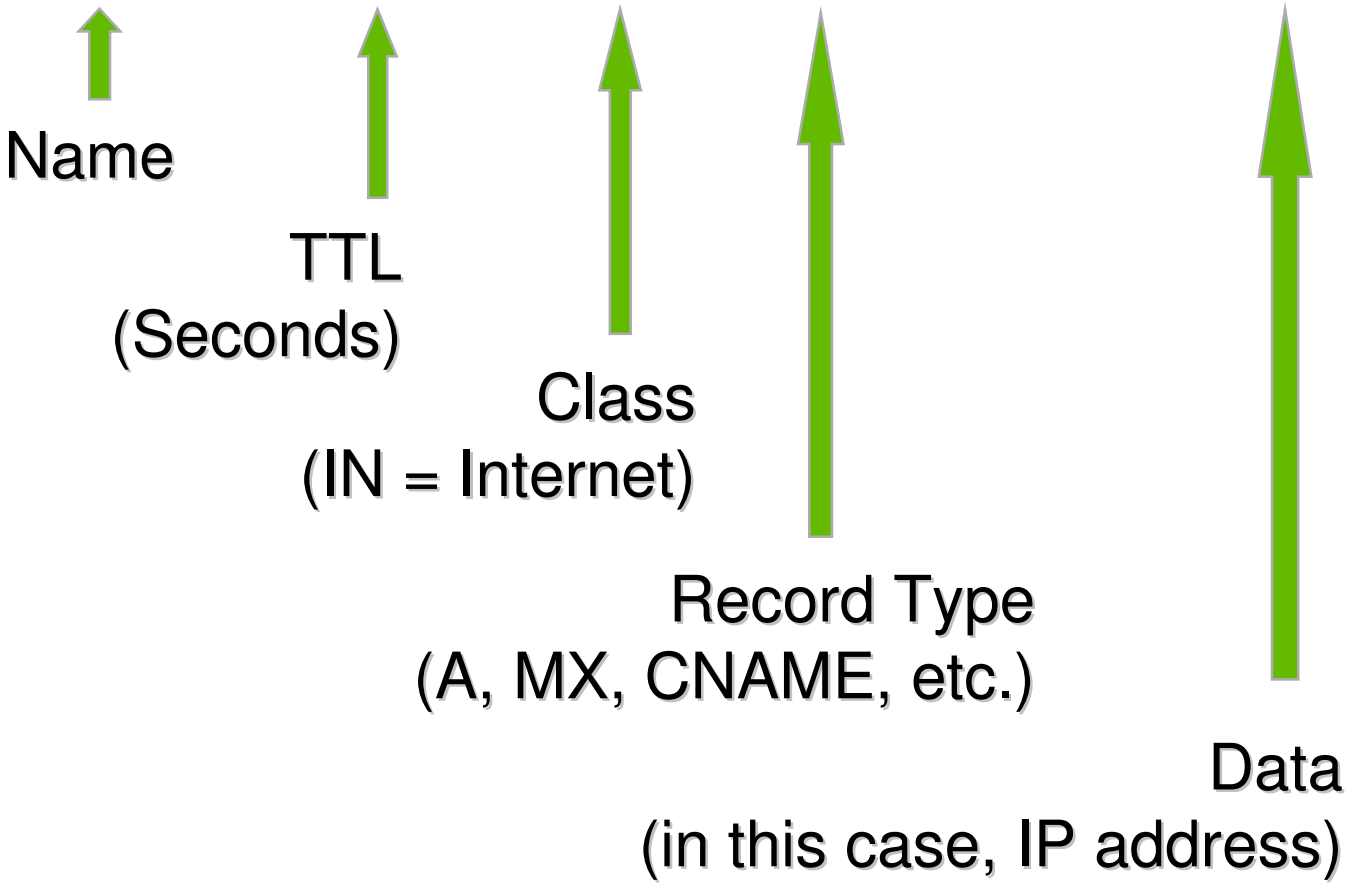
Retry = How long (in seconds) a secondary name server waits to retry a refresh operation if there’s a failure.

Expire = Maximum time (in seconds) that a secondary name server can wait before deleting UN-refreshed information about this zone.

Minimum TTL = Minimum time to live (in seconds) for resource records. This can be overridden during resource record creation.

Resource Record (RR)

```
Ignatz  10    IN   A   183.27.12.2
```



Let's look at a DNS zone file example:

Here's a file for the mythical external domain, "fubar.com":

```
@          NS       ns1.myisp.com
@          NS       ns2.myisp.com
@          MX       10                mail
mail       A        64.183.75.82
ratso      A        64.183.75.83
www        CNAME    ratso
```

Let's look at a DNS zone file example:

Here's a file for the mythical external domain, "fubar.com":

```
@          NS       ns1.myisp.com
@          NS       ns2.myisp.com
@          MX       10                mail
mail       A        64.183.75.82
ratso      A        64.183.75.83
www        CNAME    ratso
```

Each "@" in column 1 is a placeholder for the domain name. In this case, each one represents "fubar.com".

Let's look at a DNS zone file example:

Here's a file for the mythical external domain, "fubar.com":

```
            NS      ns1.myisp.com
            NS      ns2.myisp.com
            MX      10                  mail
mail        A       64.183.75.82
ratso       A       64.183.75.83
www         CNAME   ratso
```

The host names in column 1 are assumed to be the leftmost portions of the fully qualified domain name ("FQDN").

(For example, "mail.fubar.com".)

Let's look at a DNS zone file example:

Here's a file for the mythical external domain, "fubar.com":

```
@          NS       ns1.myisp.com
@          NS       ns2.myisp.com
@          MX       10                mail
mail       A        64.183.75.82
ratso      A        64.183.75.83
www        CNAME    ratso
```

Column 2 is the Record Type (NS / MX / A / CNAME)

Let's look at a DNS zone file example:

Here's a file for the mythical external domain, "fubar.com":

```
@          NS       ns1.myisp.com
@          NS       ns2.myisp.com
@          MX       10                mail
mail       A        64.183.75.82
ratso      A        64.183.75.83
www        CNAME    ratso
```

Column 2 is the Record Type (NS / MX / A / CNAME)

We have two name servers (NS), called "ns1" & "ns2"

Let's look at a DNS zone file example:

Here's a file for the mythical external domain, "fubar.com":

```
@          NS       ns1.myisp.com
@          NS       ns2.myisp.com
@          MX       10                mail
mail       A        64.183.75.82
ratso      A        64.183.75.83
www        CNAME    ratso
```

Column 2 is the Record Type (NS / MX / A / CNAME)

We have two name servers (NS), called "ns1" & "ns2"

We have one mail exchange (MX) server, called "mail"

Let's look at a DNS zone file example:

Here's a file for the mythical external domain, "fubar.com":

```
@          NS       ns1.myisp.com
@          NS       ns2.myisp.com
@          MX       10                mail
mail       A        64.183.75.82
ratso      A        64.183.75.83
www        CNAME    ratso
```

Column 2 is the Record Type (NS / MX / A / CNAME)

We have two name servers (NS), called "ns1" & "ns2"
We have one mail exchange (MX) server, called "mail"
We have two hosts (A) – "mail" & "ratso".

Let's look at a DNS zone file example:

Here's a file for the mythical external domain, "fubar.com":

```
@          NS       ns1.myisp.com
@          NS       ns2.myisp.com
@          MX       10                mail
mail       A        64.183.75.82
ratso      A        64.183.75.83
www        CNAME    ratso
```

Column 2 is the Record Type (NS / MX / A / CNAME)

We have two name servers (NS), called "ns1" & "ns2"
We have one mail exchange (MX) server, called "mail"
We have two hosts (A) – "mail" & "ratso".
We have one canonical name – "www" is an alias for "ratso".

Let's look at a DNS zone file example:

Here's a file for the mythical external domain, "fubar.com":

```
@          NS       ns1.myisp.com
@          NS       ns2.myisp.com
@          MX       10                  mail
mail       A        64.183.75.82
ratso      A        64.183.75.83
www        CNAME    ratso
```

Column 3 is the Data Field.

The NS data is the FQDN for each name server.

Let's look at a DNS zone file example:

Here's a file for the mythical external domain, "fubar.com":

```
@          NS       ns1.myisp.com
@          NS       ns2.myisp.com
@          MX       10                  mail
mail       A        64.183.75.82
ratso      A        64.183.75.83
www        CNAME    ratso
```

Column 3 is the Data Field.

The NS data is the FQDN for each name server.
The MX data is the mail priority (10) and mail host name (mail).

Let's look at a DNS zone file example:

Here's a file for the mythical external domain, "fubar.com":

```
@          NS       ns1.myisp.com
@          NS       ns2.myisp.com
@          MX       10                mail
mail       A        64.183.75.82
ratso      A        64.183.75.83
www        CNAME    ratso
```

Column 3 is the Data Field.

The NS data is the FQDN for each name server.
The MX data is the mail priority (10) and mail host name (mail).
The A data is the IP address for each host name.

Let's look at a DNS zone file example:

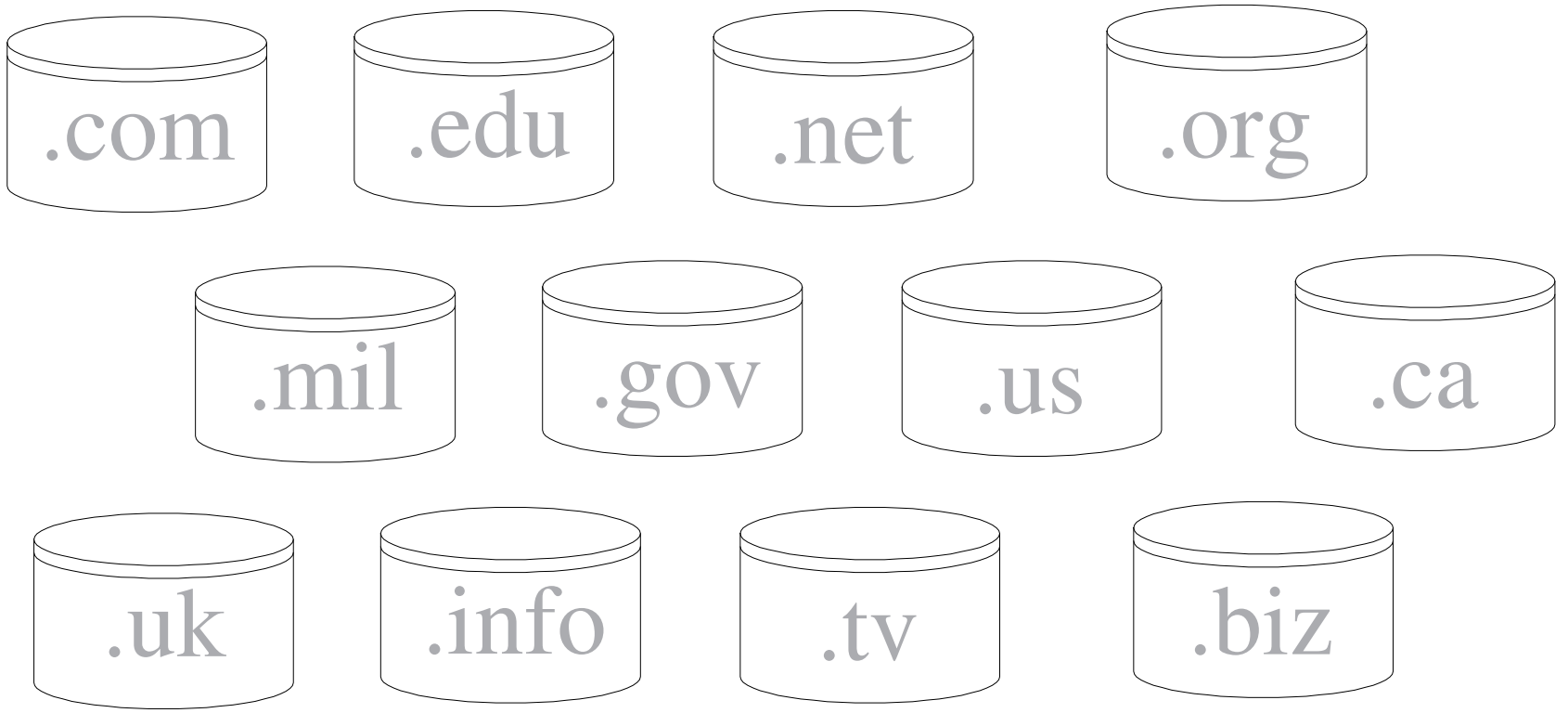
Here's a file for the mythical external domain, "fubar.com":

```
@          NS       ns1.myisp.com
@          NS       ns2.myisp.com
@          MX       10                mail
mail       A        64.183.75.82
ratso      A        64.183.75.83
www        CNAME    ratso
```

Column 3 is the Data Field.

The NS data is the FQDN for each name server.
The MX data is the mail priority (10) and mail host name (mail).
The A data is the IP address for each host name.
The CNAME data points to the aliased host (ratso).

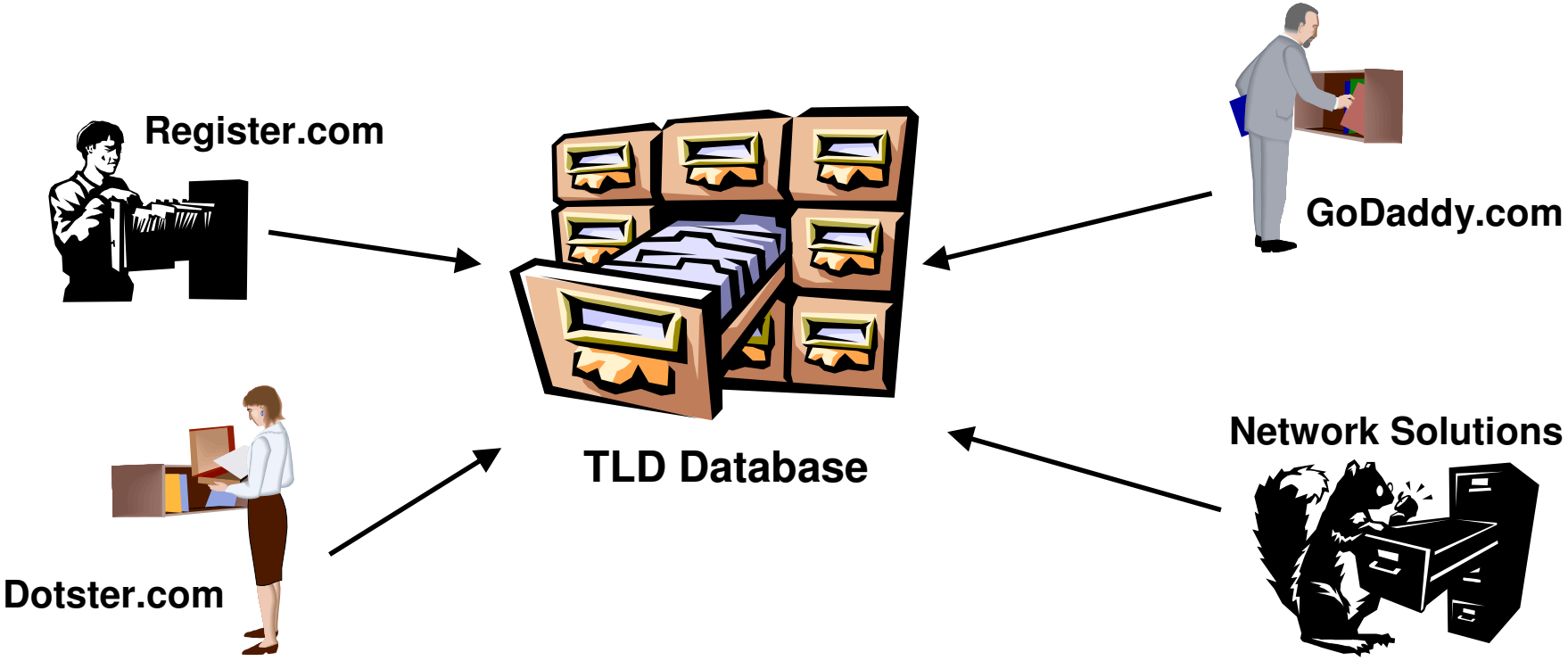
Who controls the DNS databases?



Each TLD is held in a separate database, each of which is controlled by one entity. An entity can be be a private company or a national government. Some entities (like Verisign) control several TLDs.

How is the master DNS maintained?

Domain names are added to the database by entities called registrars.

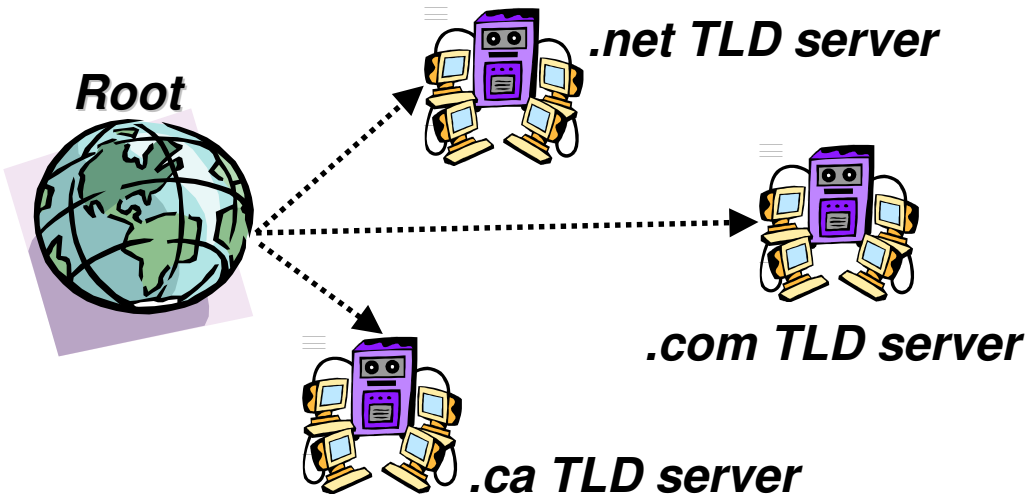


Where are the DNS records kept?

- The root servers have a list of the IP addresses for all of the Top Level Domain (TLD) servers.

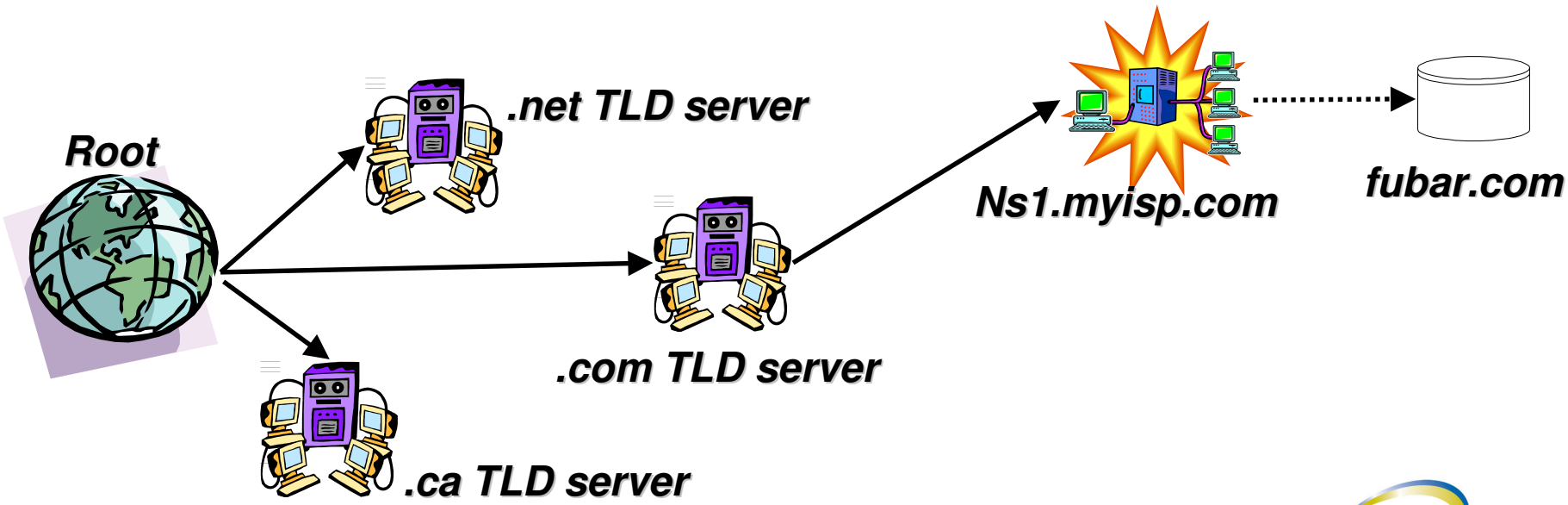
Where are the DNS records kept?

- The root servers have a list of the IP addresses for all of the Top Level Domain (TLD) servers.
- Each TLD server has a list of its second-level domain names and their associated NS records.



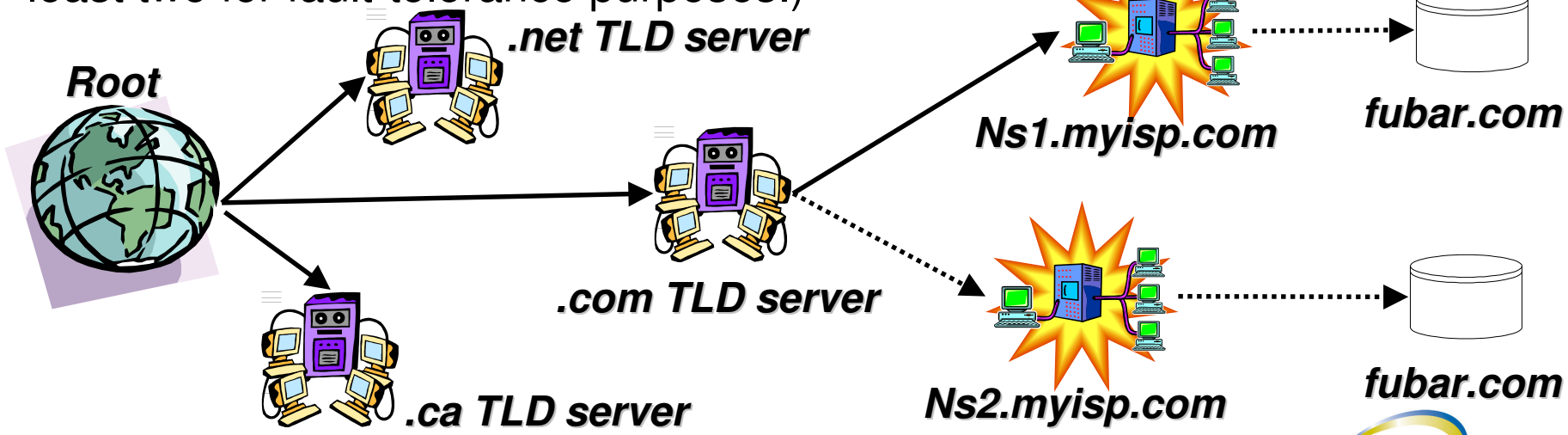
Where are the DNS records kept?

- The root servers have a list of the IP addresses for all of the Top Level Domain (TLD) servers.
- Each TLD server has a list of its second-level domain names and their associated NS records.
- Each Name Server has a listing of all records for one or more domain names (also called “zones”).



Where are the DNS records kept?

- The root servers have a list of the IP addresses for all of the Top Level Domain (TLD) servers.
- Each TLD server has a list of its second-level domain names and their associated NS records.
- Each Name Server has a listing of all records for one or more domain names (also called “zones”).
- A zone can have more than one assigned Name Server. (Most have at least two for fault-tolerance purposes.)



So, DNS is a pretty huge database?

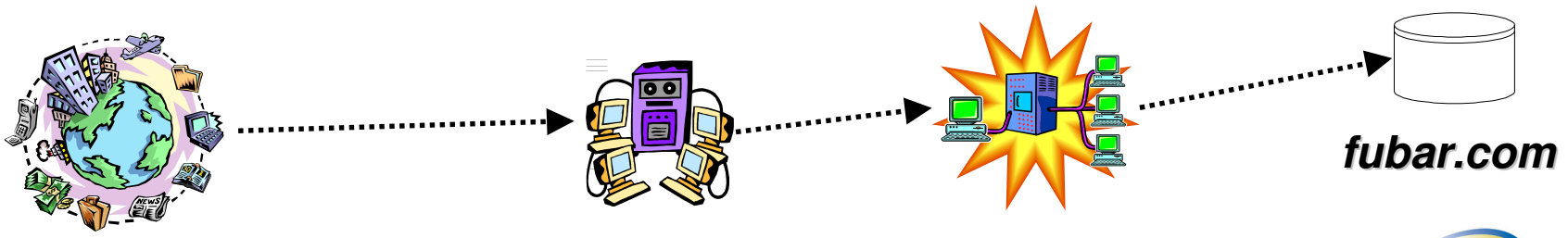
Yes and no. It's a huge data set which is made up out of many small databases.

- The DNS database is maintained on a set of root servers.
- Root servers are scattered around the world, for fault tolerance..

So, DNS is a pretty huge database?

Yes and no. It's a huge data set which is made up out of many small databases.

- The DNS database is maintained on a set of root servers.
- Root servers are scattered around the world, for fault tolerance.
- Each root server has a list of all TLD servers.
- Each TLD server has a list of name servers for all of its domains.
- Each domain's name server has a list of IP addresses for all of its hosts and/or subdomains.



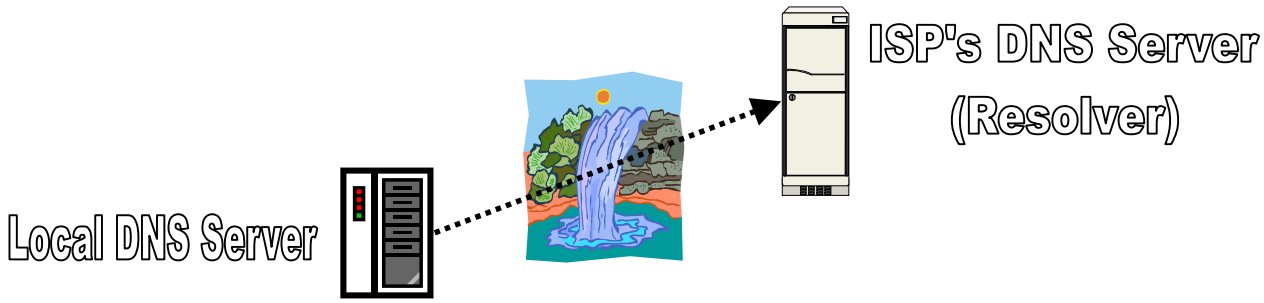
So who knows what about whom?

Each DNS server knows about at least one “upstream” DNS server it can ask for names it doesn’t know about.

One example of an upstream DNS server would be your ISP’s DNS servers.

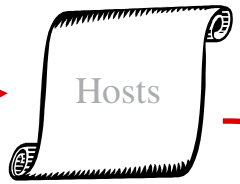
An upstream server is often called a “resolver”.

Each DNS server also has a database of IP addresses for all of the root servers … just in case it can’t decode a given domain name.

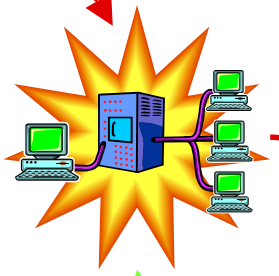


How a PC resolves DNS requests.

"What's the IP address of http://www.novell.ca ?"



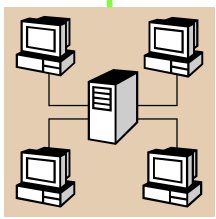
1. PC's local hosts file doesn't contain the entry, so the PC asks the LAN's internal DNS server.



2. Internal DNS Server doesn't know, so it queries the ISP's DNS

7. LAN DNS server tells PC, "www.novell.ca = 130.57.4.70"

6. ISP tells LAN DNS server, "www.novell.ca = 130.57.4.70"



5. TLD tells ISP, "www.novell.ca = 130.57.4.70"

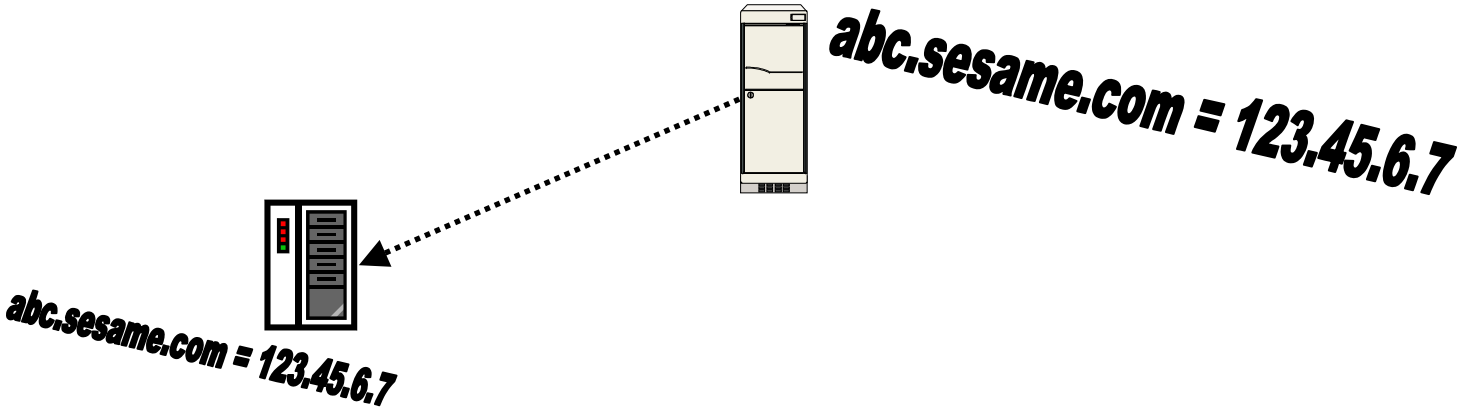
4. TLD Server for ".ca" reports: "www.novell.ca = 130.57.4.70"

3. ISP's DNS Server has no earthly idea, so it queries the .ca TLD server

How are DNS requests resolved so *quickly*?

With all of that talking back and forth, how can DNS requests possibly be resolved so *quickly*?

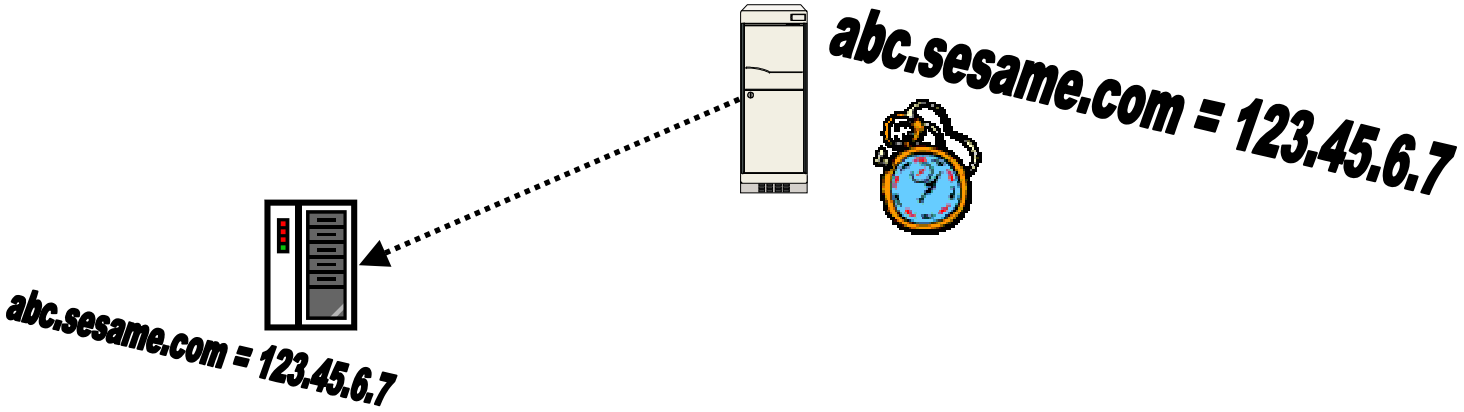
- As each DNS server gets an answer from its upstream "resolver", it caches the information…



How are DNS requests resolved so *quickly*?

With all of that talking back and forth, how can DNS requests possibly be resolved so *quickly*?

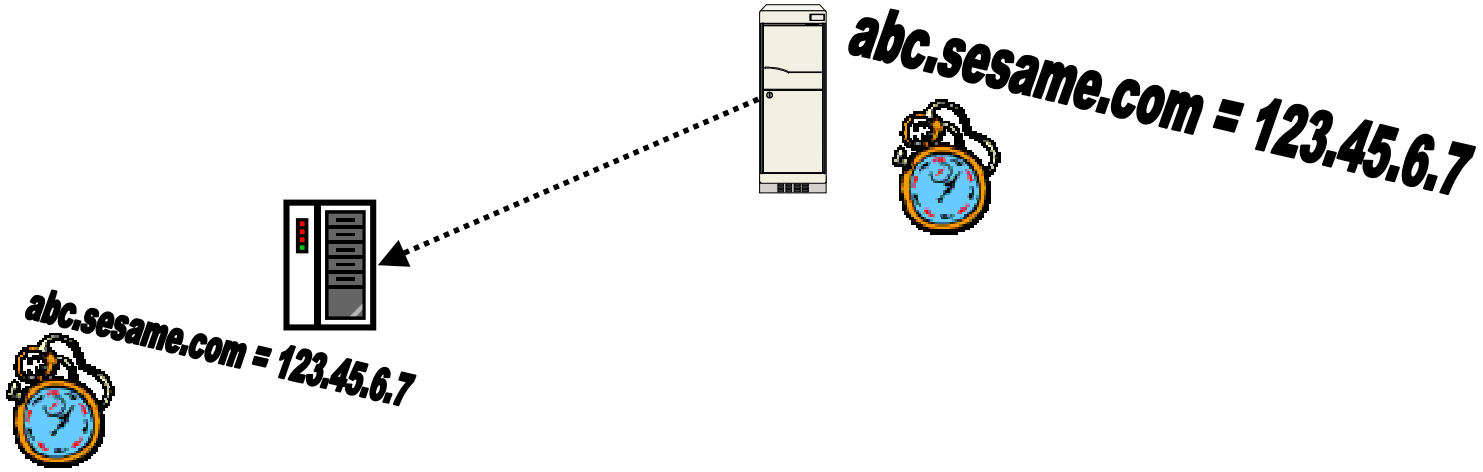
- Cached information doesn't live forever; each DNS record has a "Time To Live" value.



How are DNS requests resolved so *quickly*?

With all of that talking back and forth, how can DNS requests possibly be resolved so *quickly*?

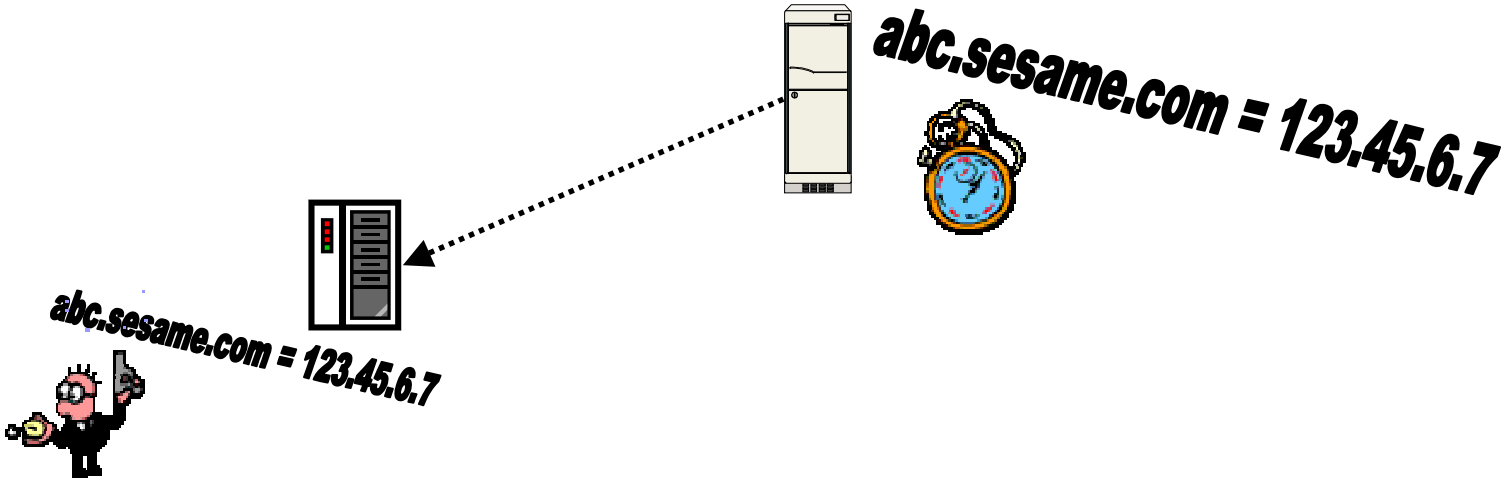
- DNS' TTL allows servers to periodically dump unused (cached) DNS records.



How are DNS requests resolved so *quickly*?

With all of that talking back and forth, how can DNS requests possibly be resolved so *quickly*?

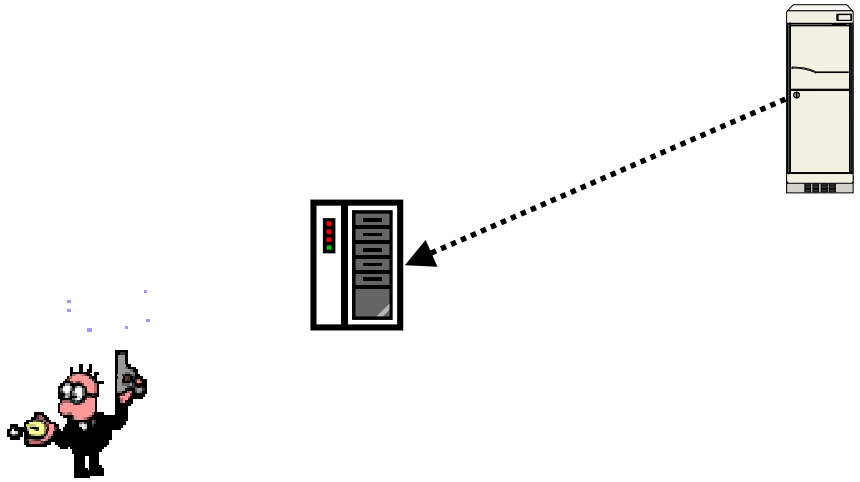
- DNS' TTL allows servers to periodically dump unused (cached) DNS records.



How are DNS requests resolved so *quickly*?

With all of that talking back and forth, how can DNS requests possibly be resolved so *quickly*?

- DNS' TTL allows servers to periodically dump unused (cached) DNS records.



How are DNS requests resolved so *quickly*?

With all of that talking back and forth, how can DNS requests possibly be resolved so *quickly*?

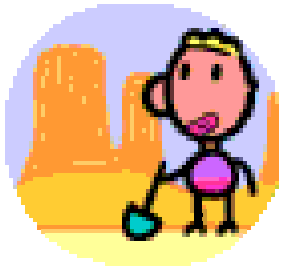
- DNS servers don't *have* to honor TTLs. That's why it can sometimes take days or weeks before everyone on the internet has a copy of a modified DNS record.

Three steps to your very own domain…

1. Find a registrar.

2. Find a DNS hosting provider. This could be:

   - Your ISP
   - Your Web Hosting Company
   - Your Registrar
   - A Third-Party DNS Hosting Company
   - A server (or servers) within your company
   - A server (or servers) within another company



3. Set up the DNS zone on your provider.

How do I set up a domain once I own it?

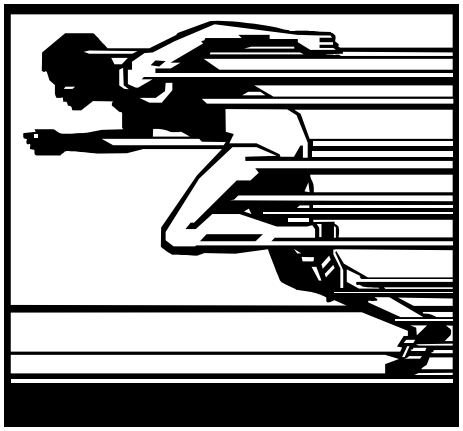
Using the name server software, define the zone:

Zone Name / FQDN	Fubar.com
Start of Authority Record	(Done automatically by NetWare DNS.)
Resource Records - NS	Ns1.fubar.com
IN-ADDR-ARPA (Reverse DNS)	(Separate step in NetWare DNS.)
Resource Records – Hosts	Ignatz A 10.0.0.1
Resource Records - MX	MX 10 mail.fubar.com

(Don't freak out; we'll go through this again later.)

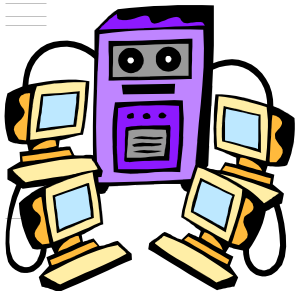
Do I *need* my own DNS Server?!

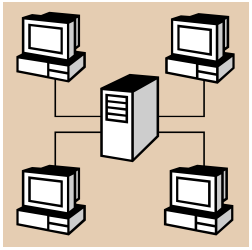
- Many new web services require DNS entries.
- Even smaller networks with only one server will benefit from having an internal DNS server.
- Several Novell services can run more quickly and smoothly if you have an internal DNS server.
- Active Directory requires its own DNS server.
- A caching DNS server (which is what most modern OS'es provide) will improve your network's perceived performance, as outside web addresses will appear to resolve more quickly.

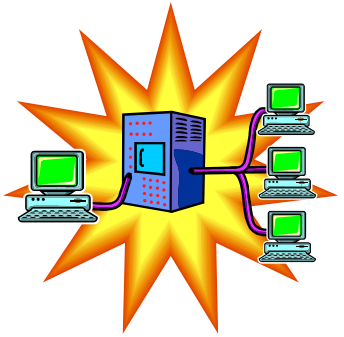


How many DNS Servers do I need?

- At least one internal DNS server AND two external DNS servers.
- If you have two or more servers … it's *not* overkill to have two of them running DNS.
- DNS services don't add a large CPU or traffic load to servers.







DNS & DHCP:
"You got peanut butter in my chocolate!"

DNS & DHCP work even better together.

DNS by itself is called "static" DNS.

DNS linked to DHCP is called "dynamic" DNS.

DNS & DHCP:
"You got peanut butter in my chocolate!"

STATIC DNS

- Manually entered
- Manually changed
- Manually deleted
- Best for servers
- Best for printers
- Best for services

DYNAMIC DNS

- Automatically entered
- Automatically changed
- Automatically deleted
- Best for workstations
- Links W/S name to DHCP-issued address

How does DNS relate to eDirectory?

- Service Location Protocol (SLP) uses DNS to resolve server and directory agent names.
- If SLP isn't working correctly, workstations will try to use DNS to locate their last known default server and/or the eDirectory tree.
- Servers can synchronize time and eDirectory more quickly if your network features working internal DNS.

How does DNS relate to Active Directory?

- Active Directory contains its own, tightly integrated DNS, which must be used for all AD participant machines.
- You must bring up a minimal set of DNS services for an Active Directory tree. This is done during AD installation.
- Early versions of Active Directory required other (non-AD) systems to have full support for BIND version 9. For example, NetWare 4.x & 5.x DNS didn't support that version of BIND. This isn't a problem with current versions of Active Directory and NetWare. (I haven't tested Linux DNS against AD yet.)
- One workaround: create your Microsoft network's DNS using Active Directory, then point AD's DNS to an external DNS (NetWare, Linux, etc.) for resolution of all non-MS hosts.
- Novell Admins: see TID #10061330, entitled "*Does Novell DNS have support for Active Directory?*" – for more information.

Why I like eDirectory & Active Directory based DNS…

- eDirectory & Active Directory store all zone information in the directory.
- Windows NT 4.0/Unix/Linux store DNS info on a local machine.

Why I like eDirectory & Active Directory based DNS…

- If you lose a NetWare DNS server … it's quick and easy to create a new DNS server on a different NetWare box … which reads the entire DNS zone from eDirectory.
- If you lose a Windows 2000/2003 server … Active Directory allows quick downloading of DNS info to a W2K DNS server.
- If you lose an NT/Unix DNS server … you need to manually recreate/re-enter the DNS information on a new box. (Best case is to import your exported bind tables.)

Best Practices for DNS & eDirectory

- Create a separate eDirectory container … I usually call mine “DNSDHCP”, and hang it off of the container where I keep my servers.
- Install both DNS and DHCP objects and services inside the DNSDHCP container.
- In larger and/or very busy networks, split off the DNSDHCP container as a separate partition. This improves performance by limiting DNS/DHCP related directory sync to only those servers with a “need to know”.
- Place replicas of the DNSDHCP partition on each DNS and DHCP server, plus whatever you need to have a minumum of three replicas.

Best Practices for DNS & Active Directory

- **Plan ahead** … DNS is installed as part of Active Directory; figure out what you want DNS to look like before grabbing the installation CD.
- **Use AD Integrated DNS** … After installing DNS, change it to be integrated with Active Directory, to provide automatic replication to multiple servers.

Useful AD DNS Resources

- Setting up Dynamic Name Services for Active Directory:

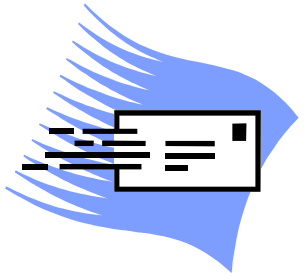
  Microsoft Knowledgebase article 237675
- Dan DiNicolo's "Quick Start Guide to setting up AD":

  http://www.serverwatch.com/tutorials/article.php/1474461

Special internal DNS records needed for Novell environments

GroupWise

- 32-bit clients will find the POA faster if you create an “A” record for “ngwnameserver”, points to the POA’s IP address. (See TID #10063483 for details.)



ZENworks

- Workstations will automatically import faster if you create an “A” record for “zenwsimport”, which points to the ZFD inventory server’s IP address. (See TID #10056752 for details.)

Servers

- You need an “A” record pointing to the IP address of each server running NDS – including NetWare, NT, 2000, Solaris, or Linux. *This is required for proper SLP operation.*

Tree

- *SLP also requires* that the tree name have its own “A” record, which points to the IP address of the server hosting the Master replica of the tree’s [Root] partition.

What other DNS records do I need?

www / ftp

- Points to the address of your public web & ftp server(s), so internal users can get there as easily as external users.

External Resources – for example:

- iFolder
- SharePoint
- GroupWise WebAccess
- Outlook WebAccess
- eCommerce Site

Should I host my external DNS?

- External DNS zones generally need fewer host records than internal zones.
- It's easiest to let an outside entity manage external DNS.
- It's also most secure to let an outside entity manage your external DNS … and the associated risks.
- Only use an external DNS provider that gives **you** full control over your DNS via a web-based, self-service control panel.

Should I host my external DNS?

- Many registrars provide self-service DNS hosting for free (Register.Com, GoDaddy.Com), or for a slight annual up-charge (Network Solutions).
- Avoid DNS providers that require you to email or call in DNS changes; they can't respond quickly enough during emergencies. (This includes most ISPs and web hosting companies.)
- If you do host your own external DNS, you should install it onto a separate set of servers outside your firewall. There should be no connection between your external DNS and your internal DNS or network directory.

Waah! My web hoster won't let me change my own DNS!

Don't want to ditch your stubborn web hosting company for one that *does* allow you to control your own DNS? Try this:

1. Create & register a second domain name, by appending the word "web" to your existing domain (for example … "fubar.com" would turn into "fubarweb.com")
2. Move your website to the new domain, which your web hosting provider will control.
3. Transfer your original domain name (fubar.com) to a registrar which provides self-service DNS. (You may have to extend your registration by one year to do this.)
4. Create a CNAME record for your original domain name (www.fubar.com) which points to your new domain name (www.fubarweb.com)

Setting up your own NetWare DNS

It's not that hard, and it's quite reliable.

If you'd like to try it (and most Novell shops should), the best overall guide I've found is available online from MOREnet consulting:

http://www.more.net/technical/netserv/servers/novell/nw5dnsdhcp.pdf

…or just search Google for "nw5dnsdhcp". This guide takes you through setting up DNS and DHCP, step by step, on NetWare 5.x and 6.

Also, see Novell TID 10013747, entitled "Configuring DNS with NetWare 5 and NetWare 6"

Moving to NetWare 6.5? You'll need to re-install the 6.5 version of the DNS/DHCP Console onto your administrative workstations. The install program can be found on any NetWare 6.5 server at:

\\servername\sys\public\dnsdhcp\setup.exe

Live Demonstration!

(Disclaimer: This will only work if we have real-time access to an internet connection here.)

Let's see what's involved in digging around the internet for DNS information, shall we?

http://www.demon.net/external/

http://www.dnsstuff.com

Thank you!

Allan Hurst
KIS Computer Center
510.494.7111
allanh@kiscc.com
http://www.kiscc.com

Very special thanks to NOBUG - the “Novell Oakland Bayarea User Group” (http://www.nobug.us) - for their invaluable inspiration and assistance in creating, testing, and refining this presentation!

HP WORLD 2004

Solutions and Technology Conference & Expo

Co-produced by:

interex

shared knowledge • shared power

encompass

AN HP USER GROUP