DNS and BIND

2000 Rock Eagle Computing Conference

October 27, 2000

CL 10/25/00

1 October 30, 2000

The ARPANET

- ARPA: Advanced Research Projects Agency
  - Part of the Department of Defense
  - Funds defense-related projects
- In the late 1960s, ARPA-funded researchers used ARPA-funded mainframes
  - If in different locations, needed a leased line and a terminal—maybe multiple lines and terminals
  - An ARPA official found this wasteful
- Solution: Connect the mainframes and terminals to a network and give everybody one terminal
- The ARPANET, the first packet-switched network, was born

2 October 30, 2000

In the Beginning…

- **There was the ARPANET's *HOSTS.TXT* file**
  - ***HOSTS.TXT* mapped every ARPANET host's name to its IP address**
  - **Format of an entry looked like:**
    - HOST:<address>:<name,aliases>:<hardware>:<os>:<list of services>
    - e.g.,: HOST : 10.2.0.52 : USC-ISIF,ISIF : DEC-1090T : TOPS20 :TCP/TELNET,TCP/SMTP,TCP/FTP,TCP/FINGER,UDP/TFTP :
  - **With this simple format, mapping from name to address ("forward mapping") and from address to name ("reverse mapping") is easy**
    - On Unix systems, the *HOSTS.TXT* file was converted to */etc/hosts* format

3 October 30, 2000

Life with *HOSTS.TXT*

- **Easily implemented and understood**
- **Everybody (in theory) had the same version of the file**
- **The file was maintained by the SRI Network Information Center (the "NIC")**
  - All file edits were done by hand
- **Network administrators sent updates via the net**
  - Initially via electronic mail
  - Later via FTP
- **The NIC released updated versions of the file twice a week**

4 October 30, 2000

The Network Explodes

- Around 1980, the ARPANET consisted of hundreds of hosts
- The ARPANET changed networking protocols from NCP to TCP/IP
  - NCP required hardware (IMPs)
  - TCP/IP was implemented in software
    - And thanks to the U.S. government, the software was essentially free
- LANs became popular
  - And engineers figured out how to use ARPANET hosts as “routers” so that any host on the same LAN could use the ARPANET

5 October 30, 2000

The Problems with *HOSTS.TXT*

- Consistency
  - The network changed more quickly than the file was updated
- Name collisions
  - No two hosts could have the same name
    - “Good” names were quickly exhausted
  - There was no good method to prevent duplicate names
    - Human intervention was required
- Traffic and load
  - The traffic generated by downloading the file became significant
    - Download time was sometimes longer than update period
- The model didn't scale well

6 October 30, 2000

Solving the Problem

- ARPANET powers-that-were launched an investigation into replacement for ***HOSTS.TXT***
- Goals:
  - Solve the problems inherent in a monolithic host table system
  - Use a consistent naming structure
  - Create a generic solution that can be used for multiple purposes
- Requirements:
  - Decentralized administration
    - With data updated locally, but available globally
  - A hierarchical name space
    - To guarantee unique names
  - Massive scalability

7 October 30, 2000

The Advent of DNS

- Paul Mockapetris, then of USC's Information Sciences Institute, designed the architecture of the new system, called the *Domain Name System,* or *DNS*
- The initial DNS RFCs were released in 1984:
  - RFC 882, “Domain Names - Concepts and Facilities”
  - RFC 883, “Domain Names - Implementation and Specification”
- The transition plan from ***HOSTS.TXT*** to DNS was initially released in November, 1983, transition to be completed by May, 1984

8 October 30, 2000

The DNS RFCs

- RFCs 882 and 883 were superseded by:
  - RFC 1032, "Domain Administrators Guide"
  - RFC 1033, "Domain Administrators Operations Guide"
  - RFC 1034, "Domain Names -- Concepts and Facilities"
  - RFC 1035, "Domain Names -- Implementation and Specification"
- Additional RFCs specified
  - New "resource record" types
  - DNS operational considerations
  - DNS policies
- DNS continues to evolve to meet the changing demands of the Internet

9 October 30, 2000

Newer DNS RFCs

- Work on DNS continues under the auspices of Internet Engineering Task Force Working Groups
  - DNSIND (Incremental Zone Transfer, Notify, Dynamic Update)
    - RFC 1995, "Incremental Zone Transfer"
    - RFC 1996, "A Mechanism for Prompt Notification of Zone Changes"
    - RFC 2136, "Dynamic Updates in the Domain Name System"
  - DNSSEC (Security Extensions)
    - RFC 2535, "Domain Name System Security Extensions"
  - These two working groups are in the process of merging into a working group called DNSEXT, for "DNS Extensions"

10 October 30, 2000

BIND, the Berkeley Internet Name Domain Server

- BIND is the most popular DNS implementation
  - Written at Berkeley as the DNS implementation for 4.3BSD UNIX
  - Originally UNIX-based, but has been ported to numerous operating systems
- BIND development is now funded by the Internet Software Consortium
  - A company called Nominum does the actual development
- The BIND distribution includes:
  - A DNS name server (*named*)
  - A stub resolver (library of C function calls)
  - Several contributed tools (*nslookup*, *dig,* etc.)
  - Documentation (the BOG) and manual pages

11 October 30, 2000

BIND History

- First version was 4.8, released 1985
  - Last version in the BIND 4 release stream was 4.9.7
- BIND 8.1 released 1997
  - Support for dynamic update and NOTIFY
  - Last version in the BIND 8 release stream probably will be 8.2.3
- BIND 9.0.0 released October 6, 2000
  - Support for incremental zone transfer and more
  - 9.0.1 expected in November

12 October 30, 2000

DNS in a Nutshell

- DNS is a distributed database system
  - Data is maintained locally, but available globally
- DNS uses replication to achieve robustness and caching to achieve adequate performance
- *Name servers* are the server half of DNS
  - They store data from specific partitions of the database and
  - Answer questions from...

13 October 30, 2000

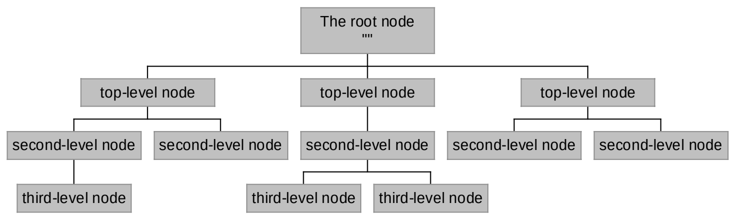
DNS in a Nutshell (continued)

- *Resolvers*, the client half of DNS
  - Which translate applications' requests for data into DNS queries and
  - Interpret name servers' responses to those queries

14 October 30, 2000

The Name Space

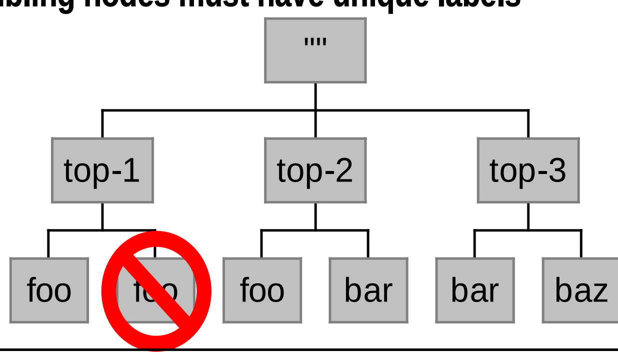
- The *name space* is the structure of the DNS database
- It's an inverted tree with the root node at the top
- Each node has a label
- The root node has a null label, written as " "



15 October 30, 2000

Restrictions on Labels

- The null label is reserved for the root node
- Legal characters for Internet hosts are alphanumerics and dash
- Sibling nodes must have unique labels



16 October 30, 2000

VeriSign® Global Registry Services

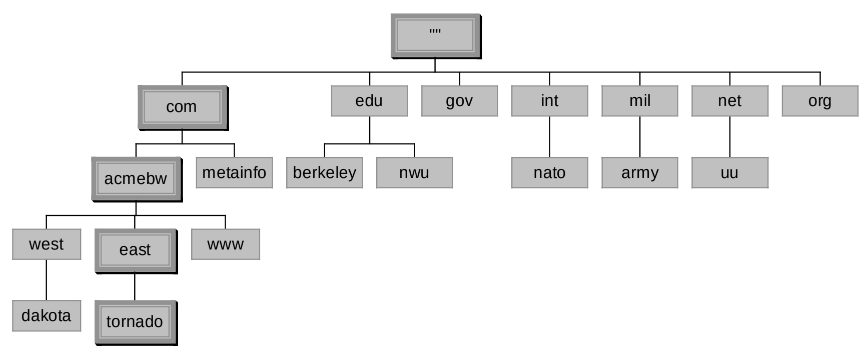
Domain Names

- A *domain name* is the sequence of labels from a node to the root, separated by dots (".”s)
- A node’s domain name identifies its position in the name space
  - Much as a pathname uniquely identifies a file or directory in a filesystem

17 October 30, 2000

VeriSign® Global Registry Services

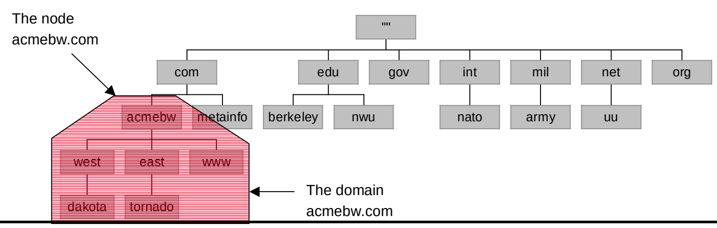
The Domain Name *tornado.east.acmebw.com*



18 October 30, 2000

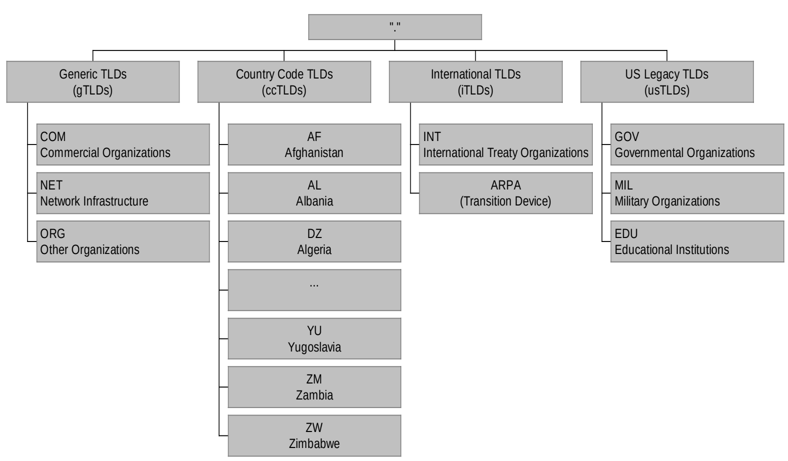
Domains

- A domain is a node in the name space and all its descendants
  - That is, a subtree of the name space
- A domain's domain name is the same as the domain name of the node at the apex (top) of the domain



19 October 30, 2000

The Current Top-Level Domains



20 October 30, 2000

Country Code Top-Level Domains

- With RFC 920, the concept of domains delegated on the basis of nations was recognized
- Conveniently, ISO has a list of “official” country code abbreviations
- The ISO 3166 list is officially available from:

  `http://www.din.de/gremien/nas/nabd/iso3166ma/codlstp1.html`

21 October 30, 2000

ccTLD Organization

- How each country top-level domain is organized is up to the country
  - Some, like Australia’s au, follow the functional definitions
    - com.au, edu.au, etc.
  - Others, like Great Britain’s uk and Japan’s jp, divide the domain functionally but use their own abbreviations
    - ac.uk, co.uk, ne.jp, ad.jp, etc.
  - A few, like the United States’ us, are largely geographical
    - co.us, md.us, etc.

22 October 30, 2000

ccTLD Organization

- Canada uses organizational scope
  - bnr.ca has national scope, risq.qc.ca has Quebec scope
- Some are flat, that is, no hierarchy
  - nlnet.nl, univ-st-etienne.fr
- Considered a question of national sovereignty

23 October 30, 2000

Fully Qualified Domain Names

- A *fully qualified domain name* (abbreviated “FQDN”) ends in a dot
  - A trailing dot (“.”) is actually the final separator between the top-level domain and the root’s null label
  - This is like absolute pathnames, which start with "/”
- Domain names without a trailing "." are not necessarily interpreted relative to the root domain
  - Just as pathnames without a leading "/" are usually interpreted relative to the current directory

24 October 30, 2000

Name Space Limitations

- The name space has a maximum depth of 127 levels
- Individual labels have a maximum length of 63 characters
- Domain names are limited to 255 characters in length

25 October 30, 2000

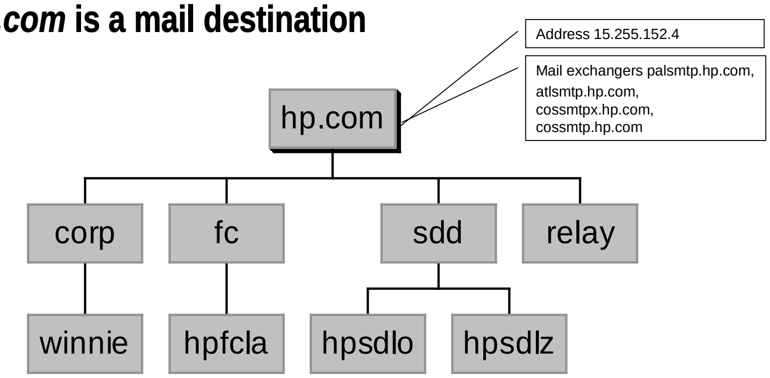
Where Do the Hosts Go?

- Anywhere!
- The nodes in the name space act as keys to the distributed database
  - Some nodes represent hosts
    - These are keys to addresses
  - Some nodes represent mail destinations
    - These are keys to mail routing information
  - Some nodes represent an entire domain
    - These are keys to lists of name servers
  - Some nodes are aliases for other nodes
  - A single node can represent a combination of hosts, mail destinations and domains

26 October 30, 2000

hp.com: Domain, Host and Mail Destination

- ***hp.com*** **is a domain (there are nodes below it)**
- ***hp.com*** **is a host (HP's web server)**
- ***hp.com*** **is a mail destination**



27 October 30, 2000

Delegation

- **Administrators can create subdomains to group hosts**
  - **According to geography, organizational affiliation or any other criterion**
- **An administrator of a domain can delegate responsibility for managing a subdomain to someone else**
- **The parent domain retains links to the delegated subdomain**
  - **The parent domain "remembers" who it delegated the subdomain to**

28 October 30, 2000

Delegation Creates Zones

- Each time an administrator delegates a subdomain to someone else, a new unit of administration is created
  - The subdomain and its parent domain can now be administered independently
  - These units are called *zones*
  - The boundary between zones is a point of delegation in the name space

29 October 30, 2000

What's in a Zone?

- Like a domain, a zone is named after its apex node
- Unlike a domain, a zone contains only descendants of the zone's apex node that aren't in a delegated subdomain
  - Nodes below a delegation point are in another zone

30 October 30, 2000

VeriSign Global Registry Services

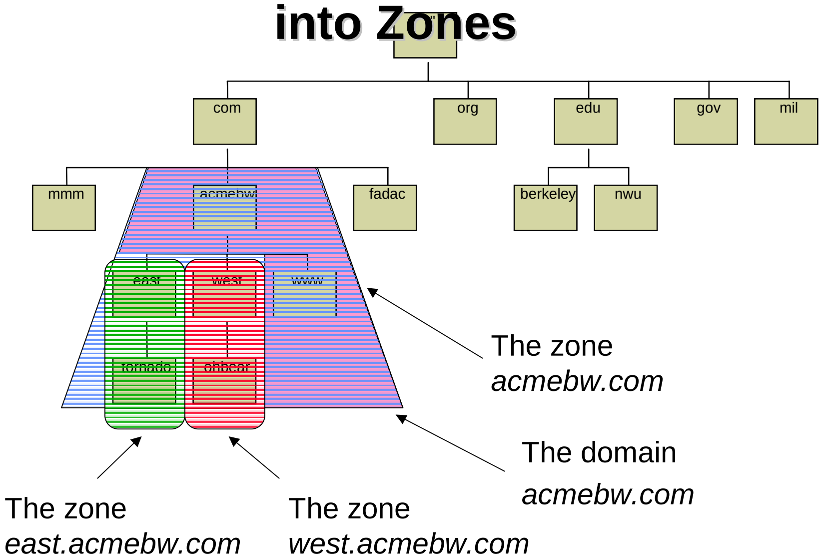
Zones vs. Domains

- A zone (of a given name) contains the same nodes as a domain (of the same name), minus those nodes that are delegated away to other zones
- For example, the zone *acmebw.com* contains the same nodes as the domain *acmebw.com*, minus those nodes in zones beneath *acmebw.com*
- When do a zone and a domain (with the same name) contain the same nodes?

31 October 30, 2000

VeriSign Global Registry Services

The Domain *acmebw.com* Divided into Zones



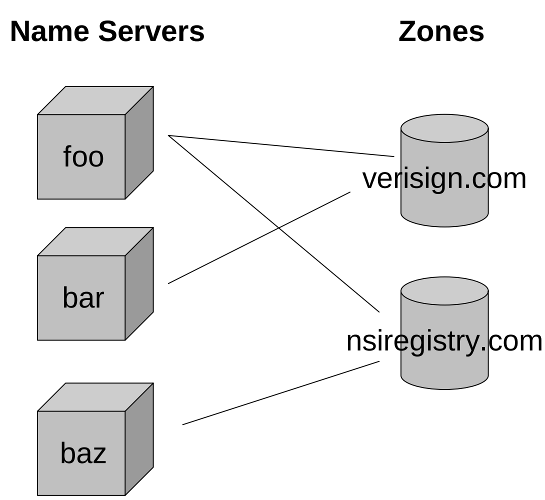
32 October 30, 2000

Name Servers

- Name servers store information about the name space in units of zones
  - The name servers that load a complete zone are said to “have authority for” or “be authoritative for” the zone
- Usually, more than one name server are authoritative for the same zone
  - This ensures redundancy and spreads the load
- Also, a single name server may be authoritative for many zones

33 October 30, 2000

Name Servers and Zones



34 October 30, 2000

Types of Name Servers

- The *primary master* name server for a zone loads the zone's data from a file on disk
- A *slave* name server for a zone loads the zone's data from another authoritative name server (often the primary master)
  - An older term for slave was "secondary master"
  - The server the slave gets its zone data from is called its *master* server

35 October 30, 2000

Types of Name Servers (continued)

- A single name server can be the primary master for some zones and a slave for other zones
  - The relationship is defined zone-by-zone
  - So, strictly speaking, you shouldn't refer to a computer as "the primary name server" unless you also specify which zone you're talking about

36 October 30, 2000

VeriSign Global Registry Services

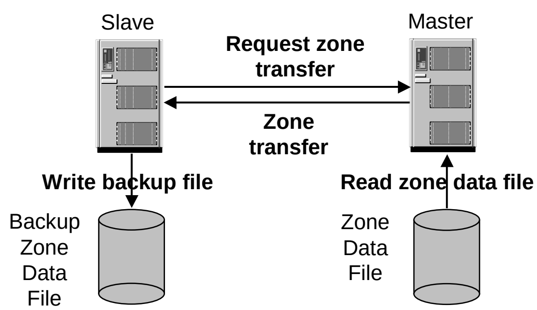
Zone Transfers

- Slave servers retrieve zone data from other authoritative name servers using a zone transfer
- The zone transfer is initiated by the slave
  - By initiating a TCP connection to the master name server

37 October 30, 2000

VeriSign Global Registry Services

Zone Transfers (continued)



38 October 30, 2000

VeriSign Global Registry Services

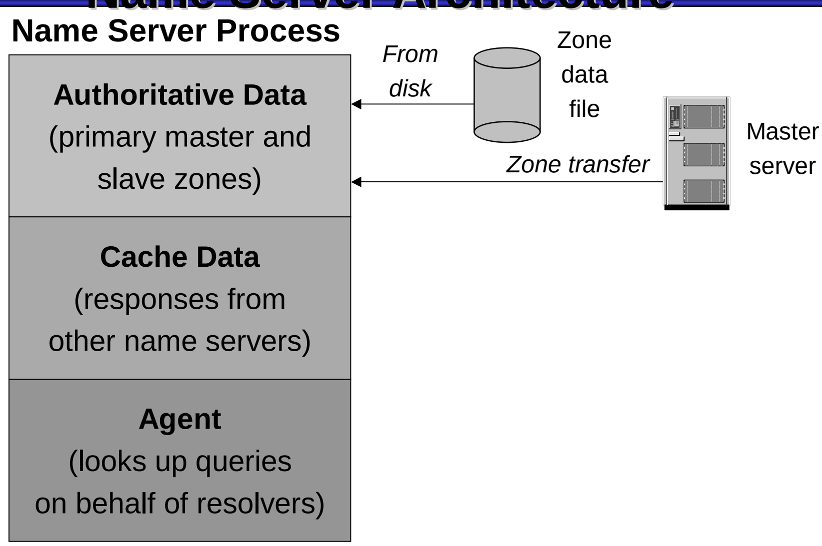
Name Server Architecture

- You can think of a name server as part:
  - *database server,* answering queries about the parts of the name space it knows about (i.e., is authoritative for),
  - *cache,* temporarily storing data it learns from other name servers, and
  - *agent,* helping resolvers and other name servers find data that other name servers know about

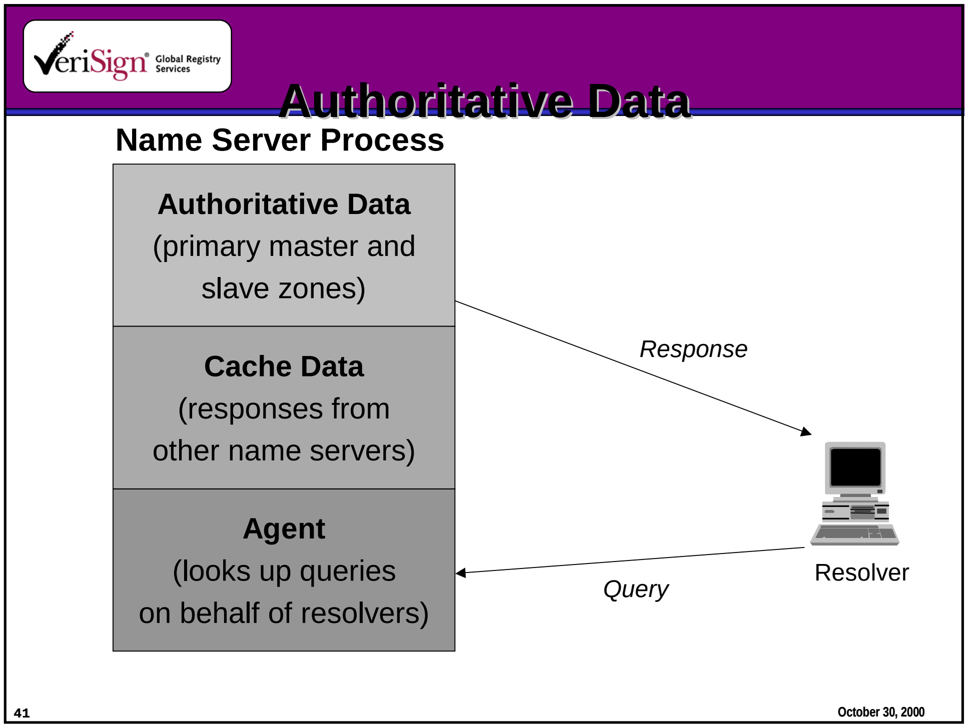
39 October 30, 2000

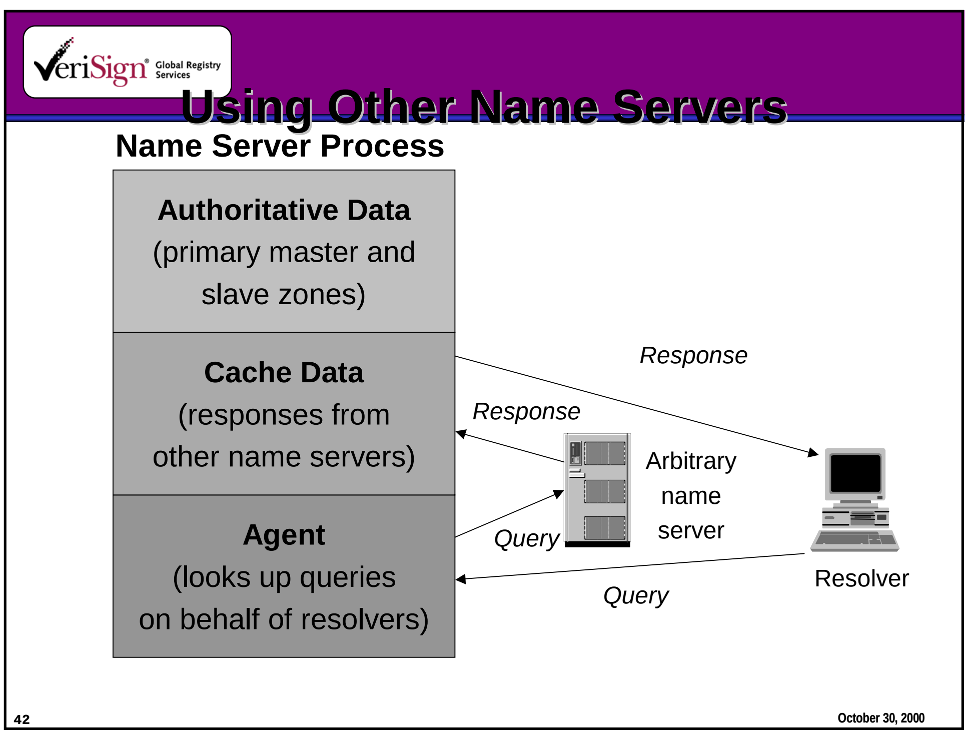
VeriSign Global Registry Services

Name Server Architecture

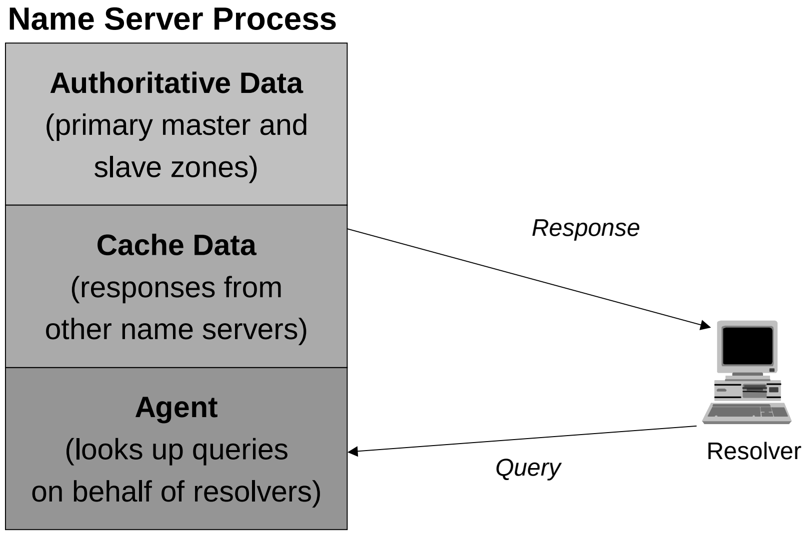


40 October 30, 2000


VeriSign Global Registry Services
Authoritative Data
Name Server Process
Authoritative Data
(primary master and
slave zones)
Cache Data
(responses from
other name servers)
Agent
(looks up queries
on behalf of resolvers)
Response
Query
Resolver
41
October 30, 2000


VeriSign Global Registry Services
Using Other Name Servers
Name Server Process
Authoritative Data
(primary master and
slave zones)
Cache Data
(responses from
other name servers)
Agent
(looks up queries
on behalf of resolvers)
Response
Response
Arbitrary
name
server
Query
Query
Resolver
42
October 30, 2000

Cached Data



43 October 30, 2000

Resource Records

- **Each domain name in the name space points to one or more *resource records***
- **Resource records have as many as five fields, some of which are optional:**
  - ***Owner:* the domain name of the node to which the record is attached**
  - ***Time to live (TTL):* more on this later**
  - ***Class:* the kind of network this record describes**
  - ***Type:* the function of this record**
  - ***RDATA:* record-specific data**
    - **The RDATA can be further subdivided into type-specific fields**

44 October 30, 2000

Record Classes

- By far the most common class of data is the Internet class, abbreviated *IN*
  - This specifies, for example, that the addresses stored are IP addresses
- Other classes include
  - *Hesiod,* for MIT's Hesiod network protocols,
  - *CHAOSNET,* for the (largely experimental) CHAOSNET protocols

45 October 30, 2000

Address Record

- An address (type "A") record specifies an IP address of a host
  - More generally, it specifies the IP address of a domain name
- The owner is the domain name of the host
- The RDATA is the dotted-octet format of a single IP address
- Multihomed hosts and routers can have multiple A records, one for each network interface

```
www.acmebw.com.    IN    A    207.69.209.143
```

46 October 30, 2000

Name Server Record

- A name server (type “NS”) record lists an authoritative name server for a zone
- The owner is the domain name of the zone
- The RDATA is a single domain name (*not* an IP address) of a name server authoritative for the zone

```
acmebw.com.    IN    NS    ns1.sanjose.acmebw.net.
acmebw.com.    IN    NS    ns1.vienna.acmebw.net.
```

47 October 30, 2000

Start of Authority Record

- A start of authority (type SOA) record specifies zone-specific values
- Each zone has one SOA record
- The owner is the domain name of the zone
- The RDATA is, er, complicated

```
acmebw.com.  IN  SOA  raven.acmebw.net. hostmaster.acmebw.com. (
                      2000071200  ; serial
                      3h          ; refresh
                      1h          ; retry
                      1w          ; expire
                      10m )       ; negative caching TTL
```

48 October 30, 2000

SOA Fields

- The fields in the SOA record are, in order:
  - The MNAME field, by convention the domain name of the primary master name server,
  - The email address of the technical contact for the zone, with a dot replacing the “@”,
  - The zone’s *serial number*,
  - The zone’s *refresh interval*, by default in seconds,
  - The zone’s *retry interval*, by default in seconds,

49 October 30, 2000

SOA Fields (continued)

  - The zone’s *expiration interval*, by default in seconds,
  - The *negative caching time to live* (*TTL*) for records in the zone, by default in seconds

50 October 30, 2000

SOA Fields and Zone Transfers

- Most of the SOA fields are related to zone transfers
- A slave server for a zone checks with its master server once during each refresh interval to see if the zone's serial number has gone *up* (not just changed)
  - If the master has a higher serial number than the slave for the zone, the slave transfers the zone
  - If the serial number is the same, the slave resets its refresh timer
  - If the serial number on the master is *lower,* the slave complains

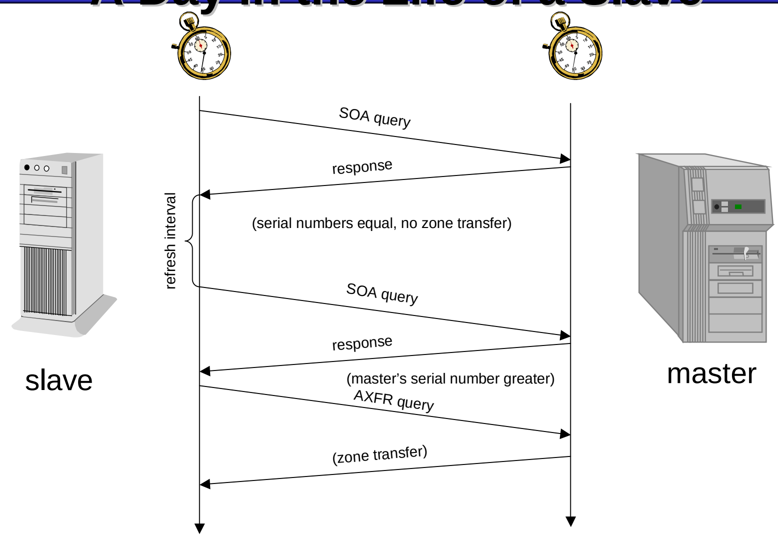
51 October 30, 2000

SOA Fields and Zone Transfers

- If the slave's check of the master's serial number fails, it tries again within the retry interval until it gets the serial number
- If the slave still can't get the serial number within the expire interval, it stops giving out answers about the zone
  - Queries for data in the zone return an error (SERVFAIL)

52 October 30, 2000

A Day in the Life of a Slave



53 October 30, 2000

Name Resolution

- ***Name resolution*** **is the process by which resolvers and name servers cooperate to find data in the name space**
- **To find information anywhere in the name space, a name server only needs the names and IP addresses of the name servers for the root zone (the "root name servers")**
  - **The name space is an inverted tree**
  - **The root name servers know about the top-level zones and can tell name servers whom to contact for all TLDs**

54 October 30, 2000

Name Resolution

- A DNS query has three parameters:
  - A domain name (e.g., *www.acmebw.com*),
    - Remember, every node has a domain name!
  - A class (e.g., *IN*), and
  - A type (e.g., *A*)
- A name server receiving a query from a resolver looks for the answer in its authoritative data and its cache

55 October 30, 2000

VeriSign Global Registry Services

Name Resolution

- If the name server can't find the answer in its authoritative data or cache, it looks for the "closest enclosing" NS record(s) it has
- For example, say the query is the one we described in the last slide
  - The closest enclosing NS records would be NS records for *www.acmebw.com*
  - The next closest enclosing NS records would be NS records for *acmebw.com*, then *com*
  - The default enclosing NS records are the NS records for the root zone, which (nearly) all name servers have

56 October 30, 2000

Name Resolution

- **Once the name server has found the closest enclosing NS records, it chooses one of the name servers from the RDATA and sends it a query**
  - **It sends the same query it received from the resolver (i.e., for *www.acmebw.com/IN/A*), rather than explicitly asking a root name server for the *com* NS records, for example**
- **The name server may get a referral (in the form of NS records) or the answer in response**
- **If the name server receives a referral, it follows it**

57 October 30, 2000

The List of Internet Root Name Servers

```
.                        3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
.                        3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000      A     128.9.0.107
.                        3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.      3600000      A     192.33.4.12
.                        3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.      3600000      A     128.8.10.90
.                        3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.      3600000      A     192.203.230.10
.                        3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.      3600000      A     192.5.5.241
.                        3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.      3600000      A     192.112.36.4
.                        3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.      3600000      A     128.63.2.53
.                        3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.      3600000      A     192.36.148.17
.                        3600000      NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.      3600000      A     198.41.0.10
.                        3600000      NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.      3600000      A     198.41.0.11
.                        3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.      3600000      A     198.32.64.12
.                        3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.      3600000      A     198.32.65.12
```

58 October 30, 2000

The Resolution Process

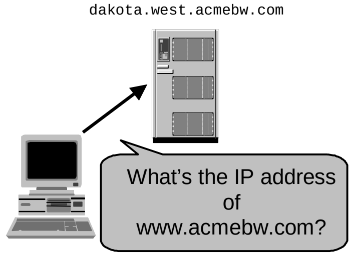
- **Let's look at the resolution process step-by-step:**


```
ping www.acmebw.com.
```

59 October 30, 2000

The Resolution Process

- **The workstation annie asks its configured name server, dakota, for www.acmebw.com's address**

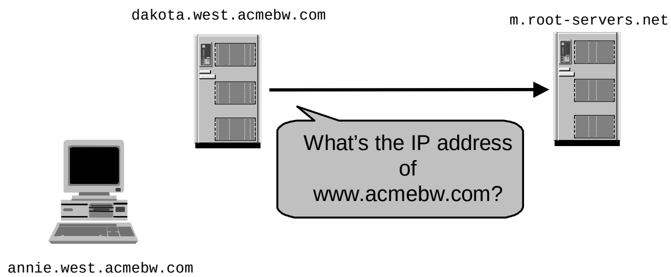



```
ping www.acmebw.com.
```

60 October 30, 2000

The Resolution Process

- The name server *dakota* asks a root name server, *m*, for *www.acmebw.com's* address

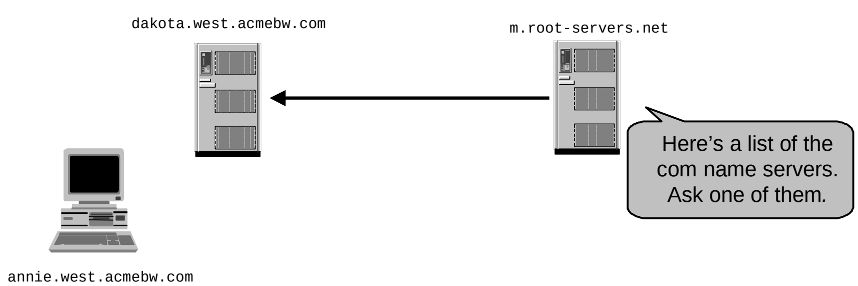



```
ping www.acmebw.com.
```

61 October 30, 2000

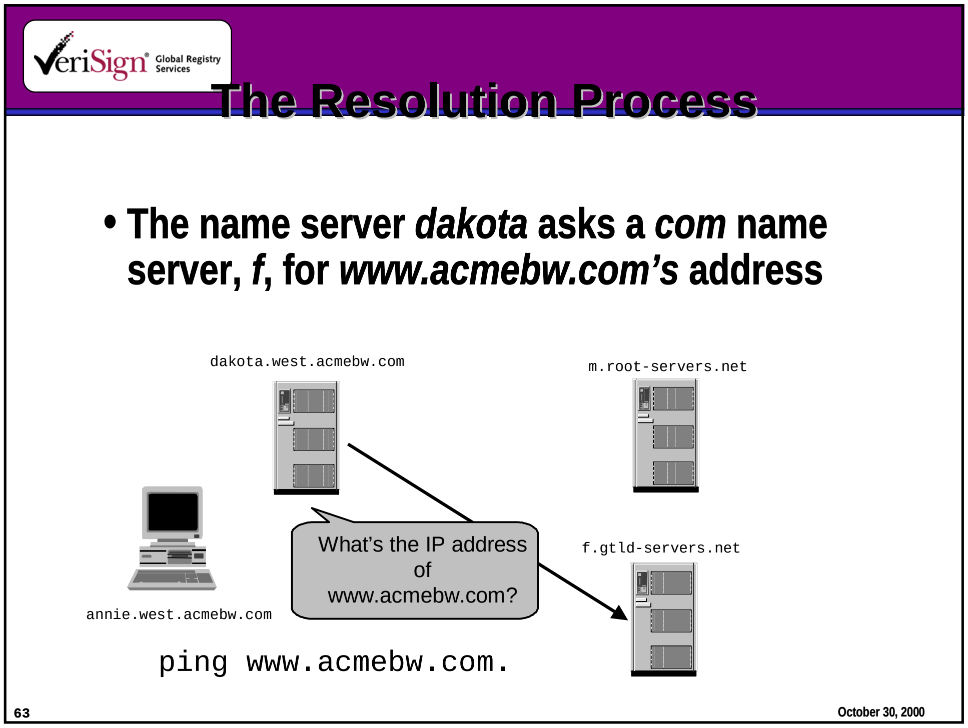
The Resolution Process

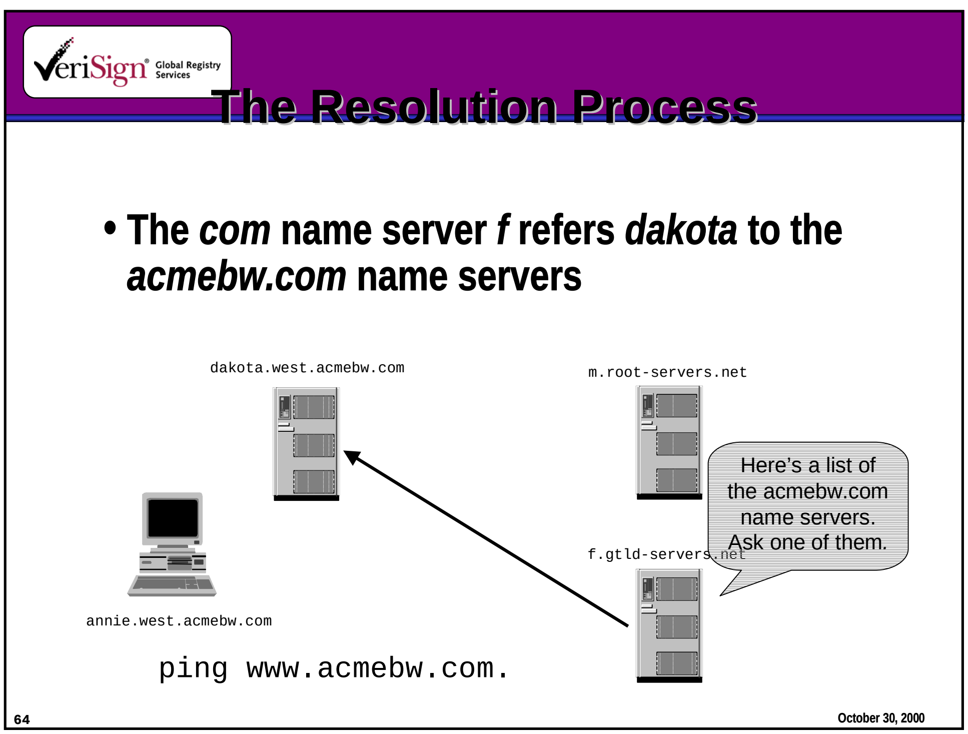
- The root server *m* refers *dakota* to the *com* name servers
- This type of response is called a "referral"




```
ping www.acmebw.com.
```

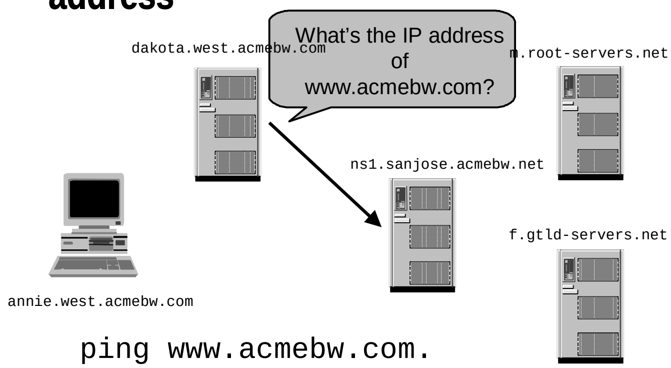
62 October 30, 2000


VeriSign Global Registry Services
The Resolution Process
• The name server *dakota* asks a *com* name server, *f*, for *www.acmebw.com's* address
dakota.west.acmebw.com
m.root-servers.net
What's the IP address of www.acmebw.com?
f.gtld-servers.net
annie.west.acmebw.com
ping www.acmebw.com.
63
October 30, 2000


VeriSign Global Registry Services
The Resolution Process
• The *com* name server *f* refers *dakota* to the *acmebw.com* name servers
dakota.west.acmebw.com
m.root-servers.net
Here's a list of the acmebw.com name servers. Ask one of them.
f.gtld-servers.net
annie.west.acmebw.com
ping www.acmebw.com.
64
October 30, 2000

The Resolution Process

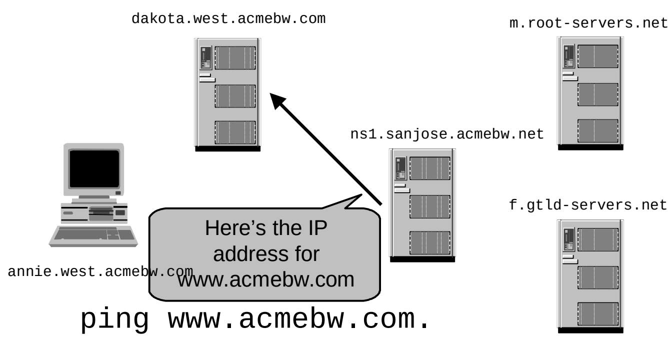
- The name server *dakota* asks an *acmebw.com* name server, *ns1*, for *www.acmebw.com's* address



65 October 30, 2000

The Resolution Process

- The *acmebw.com* name server *ns1* responds with *www.acmebw.com's* address

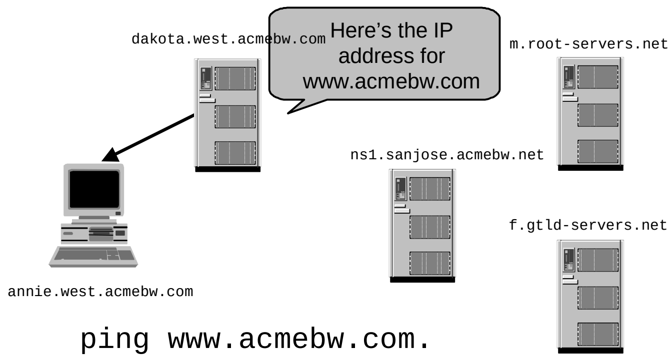


66 October 30, 2000

VeriSign Global Registry Services

The Resolution Process

- **The name server *dakota* responds to *annie* with *www.acmebw.com's* address**



67 October 30, 2000

VeriSign Global Registry Services

The James Brown of Name Servers

- **Notice that one of the name servers (*dakota*) did most of the work**
  - ***dakota* sent three queries on *annie's* behalf**
  - **The other name servers sent no queries**
  - **Why didn't (for example) the root name server query one of the *com* name servers for *dakota*?**
  - **Likewise, why didn't *dakota* refer *annie* to the root name servers, rather than sending the query itself?**

68 October 30, 2000

Types of Queries

- Resolvers send *recursive* queries
  - This type of query requires the recipient to provide an answer, not a referral
  - Resolvers need answers, not referrals, because most resolvers are too dumb to follow referrals
- Name servers send *non-recursive* or *iterative* queries to each other
  - Referrals are allowed
  - Of course, so is the answer to the query
  - Name servers are smart(er) and can follow referrals

69 October 30, 2000

Another Reason for Recursive Queries

- Imagine if all the name servers in the world sent recursive queries to the Internet's root name servers
  - A name server usually sends other name servers non-recursive queries so as not to burden them
- A resolver, on the other hand, is usually configured to query a name server run by the same organization, so that name server should bear most of the burden of resolution

70 October 30, 2000

Caching

- Caching speeds up the resolution process
- Name servers cache all records they receive from other name servers
- Remember the time to live (TTL) field in resource records?
  - It determines how long a name server can cache that particular record

71 October 30, 2000

The Resolution Process (Caching)

- After the previous query, the name server *dakota* now knows:
  - The names and IP addresses of the *com* name servers
  - The names and IP addresses of the *acmebw.com* name servers
  - The IP address of *www.acmebw.com*

72 October 30, 2000

The Resolution Process (Caching)

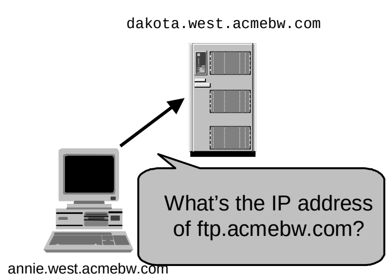
- **Let's look at the resolution process again**


```
ping ftp.acmebw.com.
```

73 October 30, 2000

The Resolution Process (Caching)

- **The workstation *annie* asks its configured name server, *dakota,* for *ftp.acmebw.com's* address**

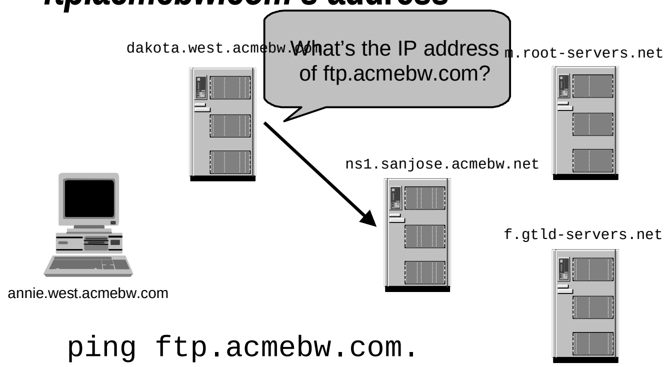



```
ping ftp.acmebw.com.
```

74 October 30, 2000

The Resolution Process (Caching)

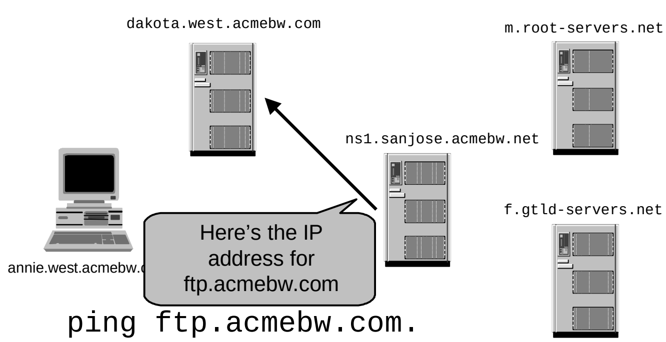
- ***dakota*** **has cached an NS record indicating** ***ns1*** **is an** ***acmebw.com*** **name server, so it asks it for** ***ftp.acmebw.com's*** **address**



75 October 30, 2000

The Resolution Process (Caching)

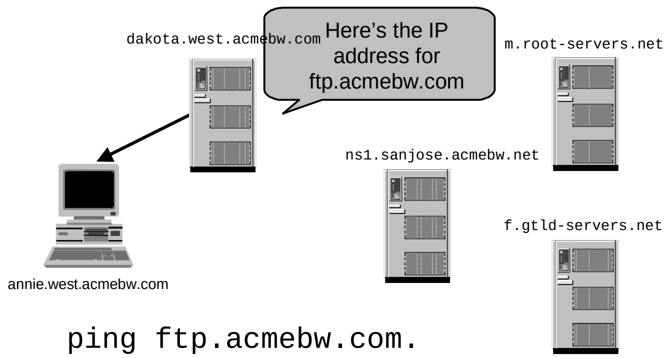
- **The** ***acmebw.com*** **name server** ***ns1*** **responds with** ***ftp.acmebw.com's*** **address**



76 October 30, 2000

The Resolution Process (Caching)

- The name server *dakota* responds to *annie* with *ftp.acmebw.com's* address



77 October 30, 2000

"Dynamic DNS"

- Lots of talk these days about "Dynamic DNS"
- Usually shorthand for three recent DNS protocol enhancements working together:
  - Dynamic Update
  - NOTIFY
  - Incremental Zone Transfer (IXFR)

78 October 30, 2000

BIND Implementations

- BIND now (almost) implements all three protocols:
  - Dynamic Update
    - Since BIND 8.1
  - NOTIFY
    - Since BIND 8.1
  - Incremental Zone Transfer (IXFR)
    - Since BIND 8.2
      - But not functional in BIND 8.2.1 despite documentation to the contrary
      - Server-side IXFR in 8.2.2
      - Client-side IXFR in probably 8.2.3
      - Other enhancements still forthcoming

79 October 30, 2000

Dynamic Update

- Allows authorized agents (e.g., DHCP servers) to update zone data by sending specially formatted messages to a zone's authoritative name server
  - Really a DNS message with fields redesignated
- Updaters can add or delete records
  - The update can be contingent upon certain conditions
    - Existence or non-existence of a domain name
    - Existence or non-existence of a record type for a domain name
- About the only thing you can't do with Dynamic Update is create or delete zones

80 October 30, 2000

Dynamic Update Security

- There is none in the protocol itself
  - Up to the server to decide whether or not to process updates
- BIND 8.2+ currently authorizes updaters by:
  - Source IP address
  - Transaction signature (TSIG)
    - But TSIG is very new and few updaters use it
- The default is to deny all updates
  - Dynamic update is enabled by creating an access list for allowed updaters

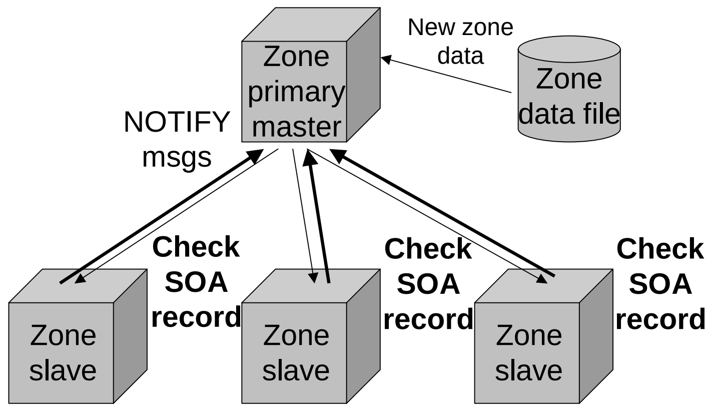
81 October 30, 2000

NOTIFY

- Allows the primary master name server for a zone to notify the slave servers of changes to the zone by sending them specially formatted messages
- Upon receipt of these messages, NOTIFY-capable slaves
  - Verify that the message came from one of their master servers
  - Immediately check the zone's SOA record on their configured master server(s)

82 October 30, 2000

NOTIFY Example



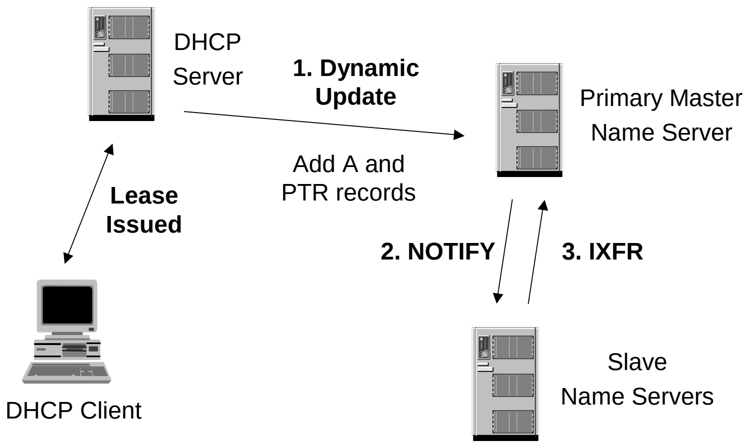
83 October 30, 2000

Incremental Zone Transfer (IXFR)

- **Until now, all transfers of zone data have transferred the entire zone**
  - Even if just a single record in the zone has changed
  - This form of zone transfer is called *AXFR,* after the query type that initiates the transfer
- **Incremental zone transfer allows slave name servers to transfer just the changes in a zone**
  - Which records have been added
  - Which records have been deleted
  - This form of zone transfer is called *IXFR*

84 October 30, 2000

Dynamic DNS: The Big Picture



85 October 30, 2000

Next Up for DNS

- Support for IPv6
- IDN
  - Support for Unicode characters in domain names
- DNSSEC
  - Cryptographic extensions to DNS to provide source authentication and integrity checking
- New top-level domains

86 October 30, 2000