

## - Domain Name System -

### Name Resolution

Name resolution systems provide the translation between alphanumeric *names* and numerical *addresses*, alleviating the need for users and administrators to memorize long strings of numbers.

There are two common methods for implementing name resolution:

- A **static file** on each host on the network, containing all the name-to-address translations (examples include the HOSTS and LMHOSTS files).
- A **centralized server** that all hosts on the network connect to for name resolution.

The two most common name resolution systems are **Domain Name System (DNS)** and **Windows Internet Name Service (WINS)**. WINS was used in Microsoft networks to translate IP addresses to NetBIOS names, and is mostly deprecated.

DNS is heavily utilized on the Internet and on systems such as Active Directory.

### Domain Name System (DNS)

**Domain Name System (DNS)** translates between *domain names* and *IP addresses*, and is supported by nearly every operating system. All Internet-based name resolution utilizes DNS.

DNS is organized as a hierarchy. Consider the following translation:

[www.google.com](http://www.google.com) = 209.85.225.104

The above domain name represents a **Fully Qualified Domain Name (FQDN)**:

- **.com** represents a top level domain.
- **.google** represents a secondary level domain
- **www** represents a host computer in the .google.com domain.

Other top level domains include **.org**, **.net**, and **.gov**. Top level domains can also include country codes, such as **.ca**, **.nl**, and **.de**

\* \* \*

All original material copyright © 2010 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## Methods of configuring DNS

Recall that DNS name resolution can be implemented in the form of local HOSTS files, or a centralized name server(s). When employing **HOSTS** files, each translation must be statically configured on each device. In Windows 2000/XP operating systems, this file is located:

c:\windows\system32\drivers\etc\hosts

In UNIX/Linux operating systems, this file is generally located: /etc/hosts

There are many disadvantages to using HOSTS files. The HOSTS file must be configured on every device. If a change occurs, every device's HOSTS file must be updated.

Using one or more **DNS servers** provides several advantages over HOSTS files. All devices point to this centralized DNS server for name resolution, ensuring that changes only need to occur in one place.

If a particular DNS server does not contain the required DNS information, the request will can be *forwarded* to servers up the DNS hierarchy.

**BIND (Berkeley Internet Name Domain)** is the standard implementation of DNS. Microsoft, UNIX/Linux, and Novell all employ some version of BIND.

DNS servers assume one of three roles:

- **Primary (or master) DNS Server** - maintains the **SOA (Start of Authority)**, and contains the master **zone file** containing the DNS records for the domain. This server is often referred to as the **Authoritative Name Server** for a specific domain.
- **Secondary (or slave) DNS Server** - maintains a current *copy* of the master zone file, obtained from the primary server. The secondary server cannot make changes to the zone file, but instead forwards changes to the primary server.
- **Caching DNS Server** - does not maintain a zone file, and is not authoritative for any domain. This server will merely cache the results of DNS queries.

Both hosts and DNS servers will cache the result of DNS queries for a period of time.

\* \* \*

All original material copyright © 2010 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## DNS Zone File Example

There are two types of **zones** in DNS:

- **Forward Lookup Zones** - translates a host name to an IP address.
- **Reverse Lookup Zones** - translates an IP address to a hostname (otherwise known as the IN-ADDR.ARPA zone).

The following is an example zone file for the fictional *example.com* domain:

```

$ORIGIN example.com
$TTL 86400
@      IN      SOA      dns1.example.com.
hostmaster.example.com. (
                                2001062501 ; serial
                                21600       ; refresh after 6 hours
                                3600        ; retry after 1 hour
                                604800     ; expire after 1 week
                                86400      ) ; minimum TTL of 1 day

      IN      NS       dns1.example.com.
      IN      NS       dns2.example.com.
      IN      MX       10      mail.example.com.
      IN      MX       20      mail2.example.com.

server1      IN      A       10.0.1.5
server2      IN      A       10.0.1.5
server2      IN      A       10.0.1.7
dns1         IN      A       10.0.1.2
mail         IN      CNAME   server1
mail2        IN      CNAME   server2
www          IN      CNAME   server2

```

Entries within a zone file are referred to as DNS **records**. There are a variety of DNS record types, including:

- **NS (Name Server)** – identifies a DNS server for the domain.
- **SOA (Start of Authority)** – identifies the primary (authoritative) DNS server for the domain.
- **A (Address)** – identifies an individual host in the domain.
- **CNAME (Canonical Name)** – assigns an alias for another host name.
- **MX (Mail Exchanger)** - identifies a mail server in the domain.
- **PTR (Pointer)** - used for *reverse* DNS lookups.

The number defined in the MX record is a *priority*. A *lower* priority is more preferred.

\* \* \*

All original material copyright © 2010 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## DNS Process

DNS follows a strict process when performing a query. The process is as follows:

1. The *local DNS cache* on the host is queried first.
2. If there is no entry in the local cache, the *local HOSTS file* is queried next.
3. If there is no entry in the local HOSTS, the query is forwarded to any *configured DNS servers* on the host. If no DNS servers are configured, the query will fail.
4. If the configured DNS server is not authoritative for that domain, and does not have that DNS entry locally cached, the query will be *forwarded up* the DNS hierarchy. DNS servers can be configured with one or more **forwarders**. Organizations often point to their ISP's DNS servers for DNS forwarding purposes.
5. If no forwarders are available, the query is forwarded to the *Root DNS server(s)*, which will likely have the entry cached.
6. In the rare circumstance that the Root servers do not have a cached entry, the query will be forwarded back down the hierarchy to the *authoritative DNS server* for that domain.

**Dynamic DNS** allows DNS to be integrated with Dynamic Host Configuration Protocol (DHCP). When DHCP hands out an IP address lease, it will automatically update the DNS entry for that host on the DNS server.

\* \* \*

All original material copyright © 2010 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.

## Resolving Hostnames on Cisco IOS Devices

There are two methods of name resolution on Cisco IOS devices:

- A static **host table** on each device (similar to a HOSTS file).
- A **centralized DNS server(s)** configured on each device.

To manually build a local host table on an IOS device:

```
Router(config)# ip host Router1 172.16.1.1
Router(config)# ip host Router2 172.17.1.2
```

To view the local host table:

```
Router# show hosts
```

To point an IOS device to a centralized DNS server:

```
Router(config)# ip name-server 10.0.1.2
```

To disable DNS lookups on an IOS device:

```
Router(config)# no ip domain-lookup
```

To configure the local domain on an IOS device:

```
Router(config)# ip domain-name CISCO.COM
```

\* \* \*

All original material copyright © 2010 by Aaron Balchunas ([aaron@routeralley.com](mailto:aaron@routeralley.com)), unless otherwise noted. All other material copyright © of their respective owners.

This material may be copied and used freely, but may not be altered or sold without the expressed written consent of the owner of the above copyright. Updated material may be found at <http://www.routeralley.com>.