

DNS

- IP Addresses
- Domain Names
- Domain Name System
- The Domain Name Hierarchy
- Components of a Domain Name
- How DNS Works
- DNS Name Resolution
- Configuring DNS on it20

IP Addresses

- For a computer to talk to the Internet it must have an IP address
- The IP address is a unique number assigned to an individual machine
- There are two protocols used in forming IP addresses
 - Internet Protocol version 4 (IPv4)
 - Internet Protocol version 6 (IPv6)
- IPv6 was created as the world started running out of unique IPv4 addresses

IP Addresses

- An IPv4 address looks like this

192.168.106.239

- An IPv4 address consists of 4 numbers each of which runs from 0 to 255 and are separated from the other numbers by a dot, .
- Each one of these numbers is called an **octet** because each one can be represented by 8 bits
- An IPv6 address consists of eight **hexadecimal** numbers each of which has 4 digits and is separated from the other

IP Addresses

numbers by a colon, :

- An IPv6 address looks like this

```
fe80::250:56ff:fe9a:1862
```

- An IP address may be either public or private
- A public IP address is unique on the entire Internet
- A private IP address is only unique in a local network

Domain Names

- In order for one computer to talk to another over the Internet it must know the IP address of the other machine
- Humans are not very good at remembering numbers but they are much better at remembering names
- A domain name is a unique string of characters which corresponds to a unique IP address
- For example, the domain name www.cs.umb.edu corresponds to the public

Domain Names

IP address 158.121.106.222

- In order to obtain a service from another machine like get a web page from a web server you need to know that machine's IP address but it is easier to remember the machines domain name
- Wherever you use a domain name you can also use the corresponding IP address

Domain Name System

- The work of turning a domain name into an IP address is performed by the **Domain Name System** usually abbreviated **DNS**
- DNS is like a phone book for the Internet
- Just like a phone book translates a name into a telephone number the Domain Name System translates a domain name into an IP address
- The mechanics of this system are governed by the DNS protocol

Domain Name System

- Since there are many millions of IP addresses it would be impractical to have the list of all domain names and their corresponding IP addresses on a single machine
- So the Domain Name System is hierarchical

The Domain Name Hierarchy

- What makes the Domain Name System work is its hierarchical nature
- Instead of having one place that lists all IP addresses and the corresponding domain names the domain name system splits that responsibility among different machines
- Each machine is responsible for a different part of the hierarchy
- At the top of this hierarchy is the **DNS Root Domain**

The Domain Name Hierarchy

- The DNS Root Domain has no name
- Beneath the DNS Root Domain are the **Top Level Domain Names**
- There are a limited number of Top Level Domain Names though new ones are being added
- Some examples are
 - **com**
 - **edu**
 - **mil**
 - **net**

The Domain Name Hierarchy

- The next level in this hierarchy are domain names under a given Top Level Domain Name
- For example
 - apple.com
 - ibm.com
 - google.com
- Once you have a domain name like this you can set up as many subdomains as you wish

The Domain Name Hierarchy

- For example the CS Department has set up the **cs** subdomain under the registered domain name umb.edu
- cs.umb.edu has its own subdomains, such as:
 - users
 - it20

Components of a Domain Name

- A domain name has two components
 - The name of the machine
 - The name of the network
- The machine name is often the hostname though it does not have to be
- The **www** that is used to get to a web site designates the web server for that site but it is probably **not** the hostname of the web server

Components of a Domain Name

- A domain name can either be local or a **fully qualified domain name**
- A local domain name can be used on a local network but not outside that network
- A fully qualified domain name can be used anywhere in the world and the last part is always a **Top Level Domain Name**
- So a fully qualified domain name is like a public IP address and a local domain is like a private IP address

Components of a Domain Name

- Each component of a domain name is separated from what follows by a dot, .
- As you read a domain name from left to right you go from specific to more general
- Consider the machine `users1` on our CS department network
- Its domain name is `users1.cs.umb.edu`
- This domain name has four components:
 - `users1`
 - `cs`
 - `umb`
 - `edu`

Components of a Domain Name

- The first part of this domain name, **users1** specifies a unique machine
- The next component, **cs** specifies CS Department subdomain
- The next component, **umb** is a name registered under **edu**
- The last component, **edu** is the Top Level Domain Name

How DNS Works

- There are two parts to the DNS system
 - Domain name registration
 - Domain name resolution
- To obtain a domain name visible on the Internet you have to go to a **domain name registrar**
- Registrars only deal with certain Top Level Domain Names
- The domain name registrar is also responsible for providing a machine that

How DNS Works

serves as the source of all information about the domain

- This machine is called an **authoritative name server**
- So when our University acquired the domain name **umb.edu** it had to contact a domain registrar for the top level domain **edu**
- to register the domain name **umb**
- This is the registration part of the DNS system

How DNS Works

- This registrar provides an authoritative name server that points all requests for domain names ending in [umb.edu](#) to a machine on our network
- This machine serves as the authoritative name server for any domain that ends in [umb.edu](#)
- Every subdomain of a fully qualified domain name must have an authoritative name server for that subdomain

How DNS Works

- Each authoritative name server is responsible for providing IP addresses for domain names in its **zone**
- The zone consists of the network and subnetworks for which that proper authoritative name server is the ultimate authority
- Though one machine can be an authoritative name server many other machines are given copies of the information to spread out the

How DNS Works

- The process of turning a domain name into an IP address is called **DNS name resolution**
- This is the second part of the DNS system

DNS Name Resolution

- DNS name resolution is something that happens every time you put a domain name into your browser
- Every Unix machine has a DNS configuration file `/etc/resolv.conf`
- This file contains the address of the first machine to ask for the IP address that corresponds to a given domain name
- In Ubuntu this file is generated automatically by the *resolvconf* utility

DNS Name Resolution

- This DNS server will consult its cache of domain names to see if it already knows the IP address you are looking for
- If the name is not in the cache what happens next depends on the configuration of this DNS Server
- If has been configured with a forwarder it contacts that machine for the IP address
- A forwarder is a machine that that one DNS Server contacts when it cannot resolve a domain name

DNS Name Resolution

- If no forwarder is configured the DNS Server contacts the root name server
- The root name server has the IP address of every top level domain name such as **com**, **edu** and **org**
- With this information the DNS server contacts the top level domain server and asks for the IP address of the authoritative name server for the first part of the domain name

DNS Name Resolution

- With this information, the DNS server contacts the authoritative name server and asks for the IP address it is looking for
- If the domain name has several components the process of contacting an authoritative domain server may happen more than once
- All of the above is spelled out on page 269 of Beginning Ubuntu LTS Server Administration

Configuring DNS on [it20](#)

- The name of the DNS server software Linux uses is BIND which stands for **Berkeley Internet Name Domain**
- The name of the BIND name server daemon is *named*
- The software package Ubuntu uses for DNS is bind9
- The package was installed using *apt-get*
`sudo apt-get install bind9`
- bind9 can be restarted, after configuration

Configuring DNS on it20

changes, using

```
sudo service bind9 restart
```

- The configuration files for BIND are located in */etc/bind/*
- The configuration file for the *named* daemon is */etc/bind/named.conf*
- Our DNS is set up to provide to serve as a DNS server for three different zones
- Each of these zones provides the IP address for the virtual machines on our Lab network

Configuring DNS on [it20](#)

but they provide these addresses to machines on three different subnets

- internals - provides the IP addresses to machines on the Lab network, [it.cs.umb.edu](#)
- cslan - provides the IP addresses to machines on the CS LAN [cs.umb.edu](#)
- external - provides the IP address to any machine on the internet
- The primary configuration file for the *named*

Configuring DNS on it20

daemon is `/etc/bind/named.conf`

- On `it20`, this file does little but reference `/etc/bind/named.conf.local`
- This file contains the first information about the zones but the actual addresses associated with each machine's hostname is contained in separate zone files
- Inside `/etc/bind` you will see directories for each zone
- Inside each of these directories is a `db.it` file

Configuring DNS on it20

with hostnames of the virtual machines and the IP addresses for each one

- Once these files are all properly configured the hosts line in `/etc/nsswitch.conf` must be changed to

```
hosts: dns files
```

- This line instructs `it20` to use its own DNS service for name resolution before using its version of `/etc/hosts`

Configuring DNS on [it20](#)

- After this configuration was done, the service had to be restarted

```
/etc/init.d/bind9 restart
```