Introduction to DNS and Application Issues related to DNS

Kirk Farquhar

SCO

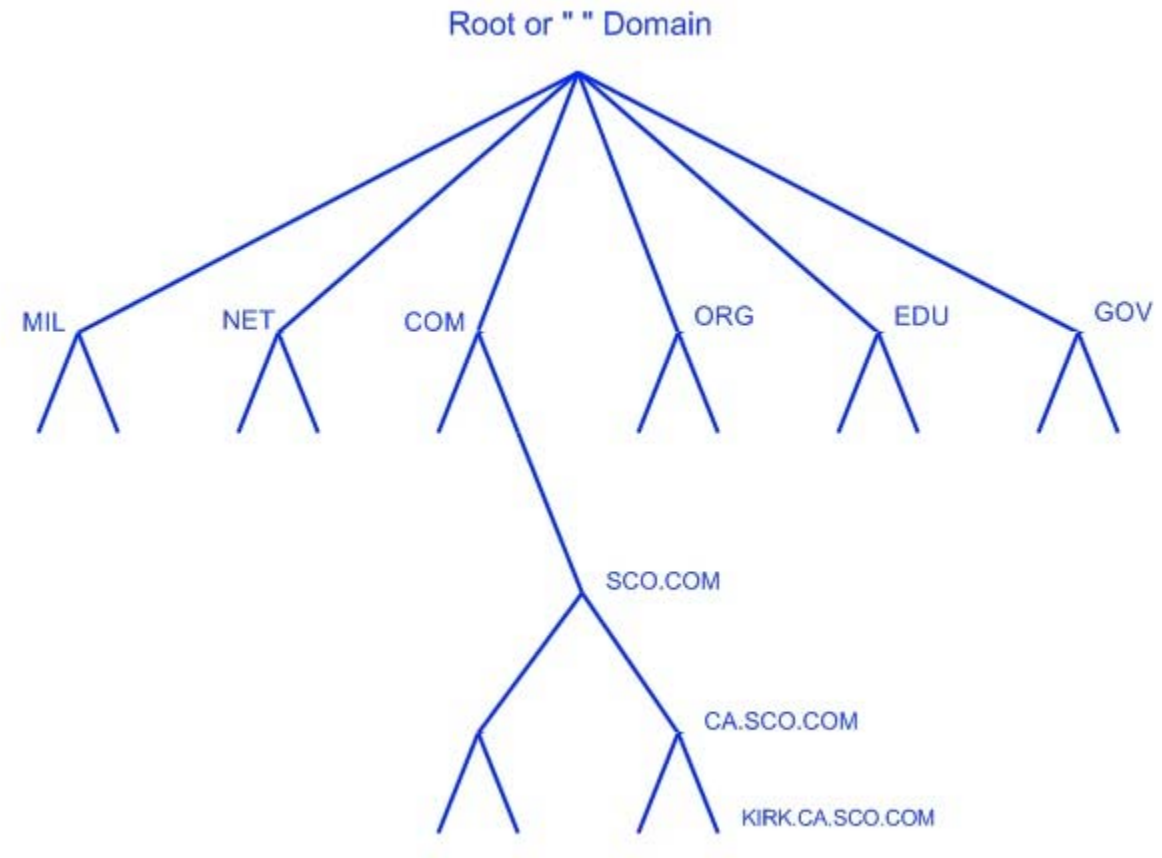
Content

- What is DNS?
- How it all works
- Setting up your domain
- Creating your nameserver files
- The Resolver
- Testing
- Firewall configuration
- ACL’s
- Cache poisoning
- Attacks
- Controlling users
- Controlling & securing zone transfers

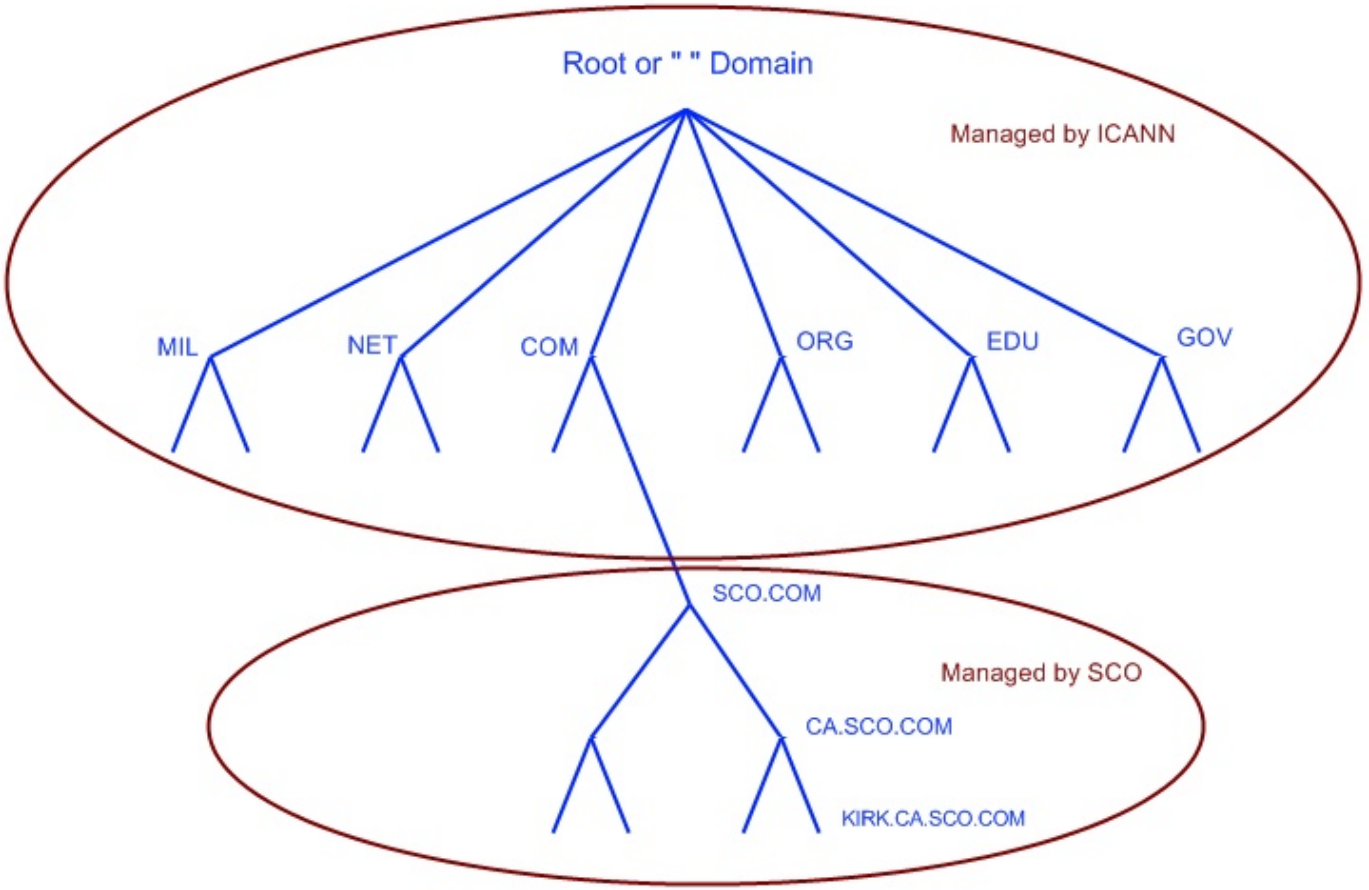
What is DNS?

- DNS – Domain Name System
- A distributed database for mapping readable names to domains and nodes in the Internet, allowing distributed and delegated management
- The DNS database follows the model of an inverted tree, the leaves are individual nodes, and branches represent domains and sub-domains.
- All branches come from a single “root” or null domain

The Domain Tree



Managing the Domains

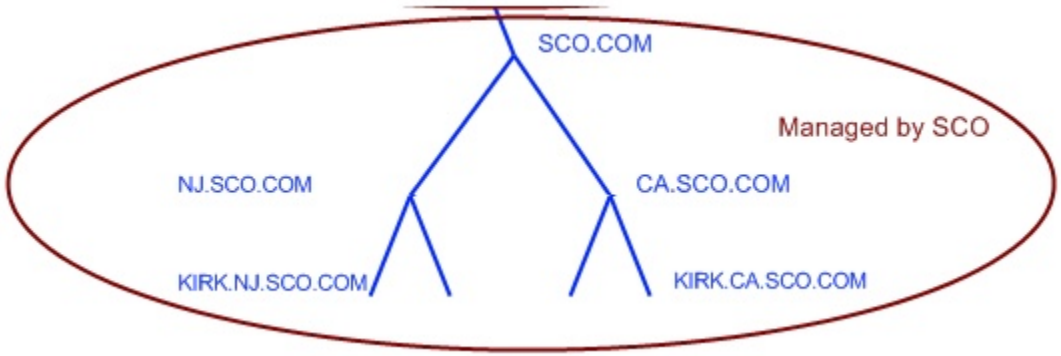


How it all works

- Each entity in the Internet is provided with a domain name
- Entities can be Domains, sub-domains or individual nodes or systems
- Entities are named by following the tree in reverse order
  - i.e. kirk.ca.sco.com.
- The last "." actually refers to the root or null domain name and has significance
  - i.e. this is really kirk.ca.sco.com**." "**

How it all works

- DNS requires that sister nodes, i.e. nodes at that are children of a common parent have unique names
- Nodes with different parents however can have common names



How it all works - Domains

- A Domain is simply a sub-tree of the DNS database
- Any domain or any node within the sub-tree is considered part of the Domain
  - kirk.nj.sco.com and kirk.ca.sco.com are both part of sco.com
  - kirk.nj.sco.com is not however part of ca.sco.com
  - Likewise nj.sco.com and ca.sco.com, although domains are also part of sco.com

How it all works – Domain Hierarchy

- A Domain that is a child of the null or root domain is a “First-level Domain”, also known as a Top-level Domain
- A Domain that is a child of any First-level Domain is a Second-level Domain, and so on
- There are very few naming restrictions on domains and no particular significance is attached to names

How it all works – First-level Domains

- The original First-level domains are:
  - Com - for commercial enterprises
  - Mil - for Military entities
  - Gov - for US Government entities
  - Edu - for educational institutions
  - Org - for non-commercial organizations, although now unrestricted
  - Net - for network infrastructure, although also now unrestricted
  - Int - for international organizations, i.e. nato.int

How it all works – new First-level Domains

- Aero – for aerospace industry
- Biz – a generic addition to com
- Coop – for cooperatives
- Info – for generic information services
- Museum – for museums
- Name – for individuals' domains
- Pro – for professionals

How it all works – Country First-level Domains

- Need to accommodate internationalization and a compromise was struck
- Each country can manage the policies within their country first-level domain
- Each country has a two character code for the domain name, i.e ca for Canada, uk for Great Britain
- Countries may or may not follow consistent conventions below that
  - aci.on.ca, amazon.co.uk, google.com.au

How it all works - Delegation

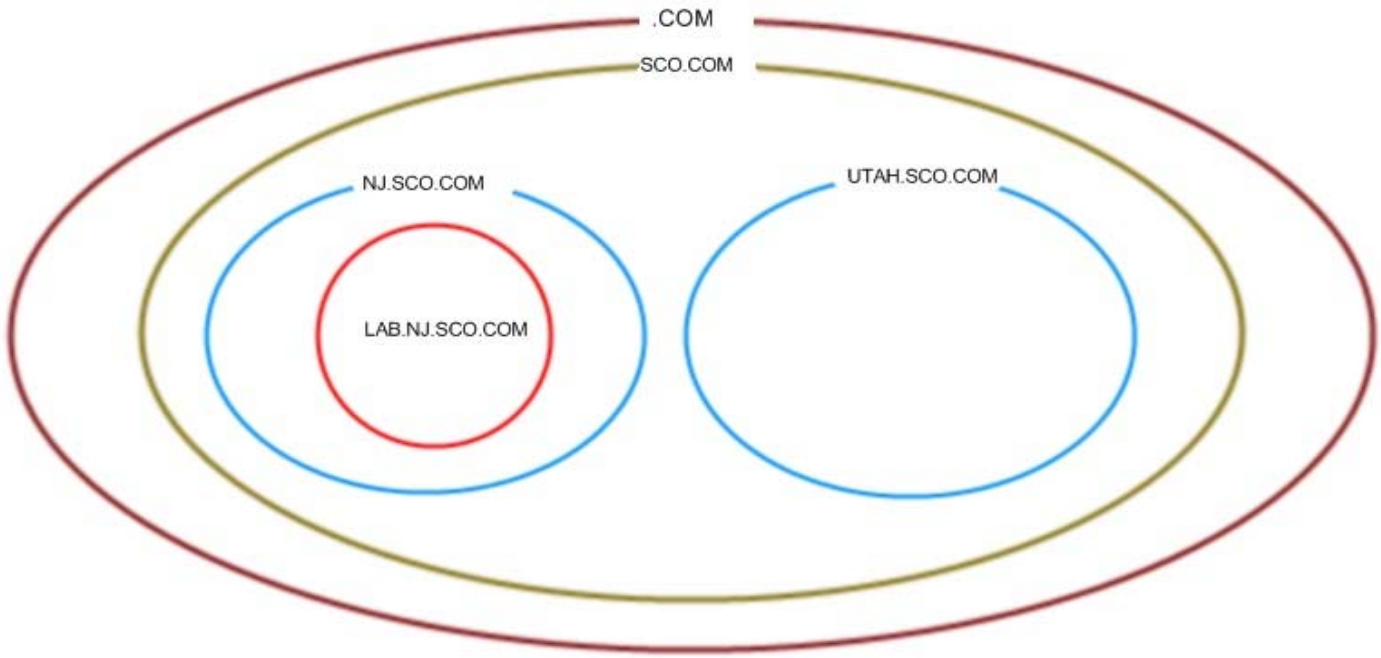
- Without delegation, managing the Internet domain names would be impossible
- Delegation can occur at any level and delegates can re-delegate portions of their space
- ICANN manages the root domain, but delegates .ca to Canadian entities
- on.ca in turn is delegated to the Province of Ontario
- aci.on.ca is delegated to ACI

How it all works – NameServers and Zones

- The applications that store and distribute information about the Domain namespace are NameServers
- NameServers typically have all information and are authoritative for a part of the namespace, referred to as a zone
- NameServers can be authoritative for multiple zones

How it all works – NameServers and Zones

- NameServers can service multiple or single zones
- Authority can be delegated by zone



Types of NameServers

- There are three types of NameServers
  - Primary Master NameServers
    - These contain data records for the entire zone and are authoritative
  - Secondary Master or Slave NameServer
    - Gets its data about zone(s) from its primary master or another secondary master, and is authoritative for the zone
  - Caching NameServer
    - Does not store any permanent data and is not authoritative. Simply looks up data and caches it for the TTL
    - Used to offload processing only

Resolvers

- Resolvers are the clients that connect to NameServers to get DNS information to provide to applications
- The Resolver queries the NameServer, interprets the responses and provides the information to an application in a form it can use
- On Unix system, the primary resolver is a set of libraries that are linked into applications

Root NameServers

- To query the DNS system, a NameServer only needs information about the Root NameServers
- The Root NameServers know the IP addresses of the servers that are authoritative for each first-level zone
- The are currently 13 root NameServers spread around the world

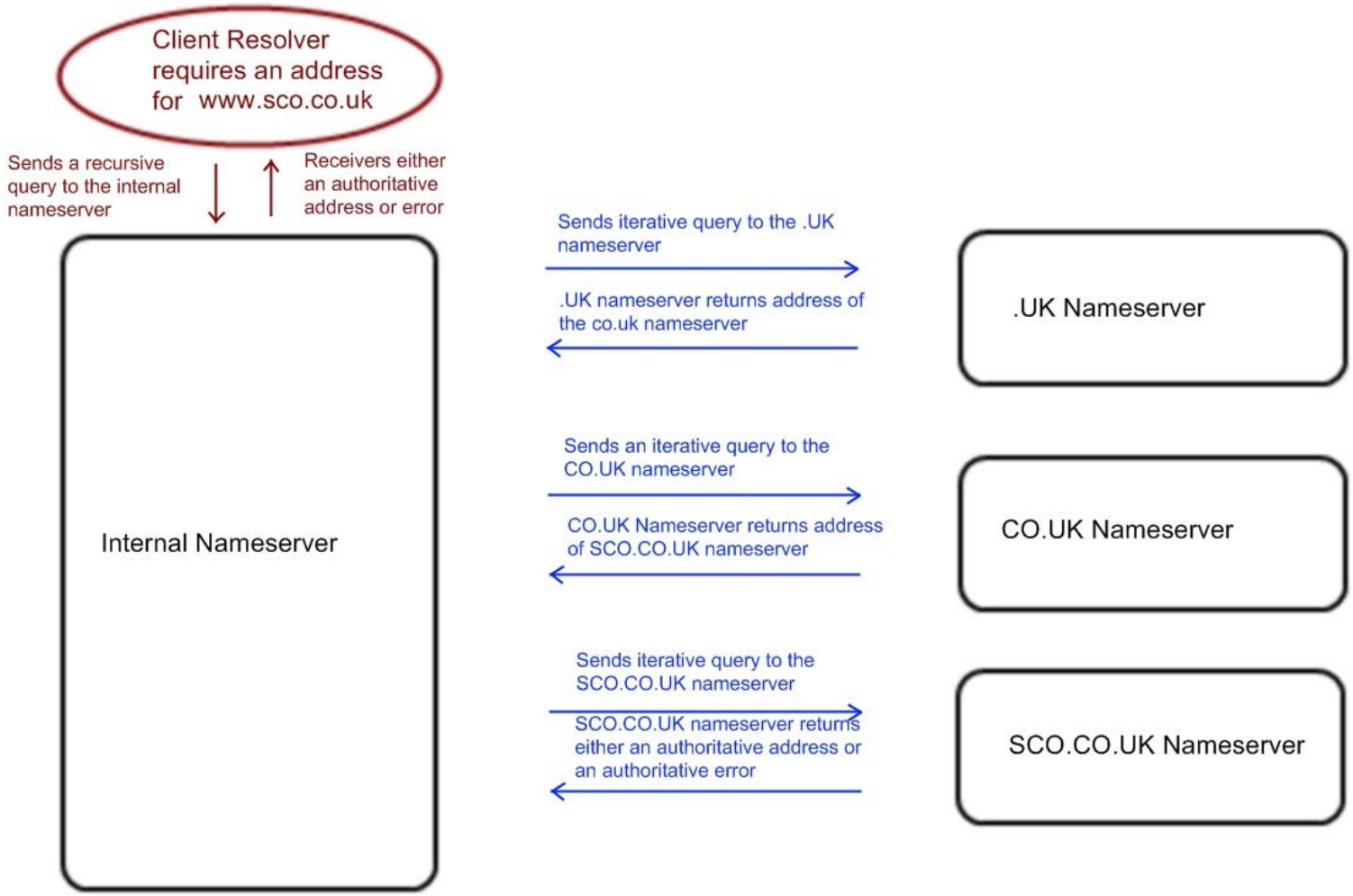
Recursion & Iteration

- A NameServer receiving a recursive query is obligated to respond with the information or an error
  - A server in NJ receives a query for hr.utah.sco.com
  - If it doesn't have the information it will in turn query the "nearest" NameServer to the requested name
    - Try utah.sco.com NameServer
    - Try sco.com to find who is authoritative for utah.sco.com
    - Try .com to find who is authoritative for sco.com
    - Try the root NameServers to find who is authoritative for .com
  - It will return an answer or an error

Recursion & Iteration

- A NameServer receiving an iterative query will simply provide its best answer, or will refer the query to the nearest NameServer.
- If your NameServer doesn't know hr.utah.sco.com but does know the NameServer for sco.com it will simply tell you to go ask there
  - This referral includes all NameServers in the local cache
  - The resolver must "select" a NameServer from the referral to query
- This reduces the load on the NameServer dramatically
- The NameServer tracks the Round Trip Time of queries to NameServers (RTT) and locks onto the one with the lowest RTT

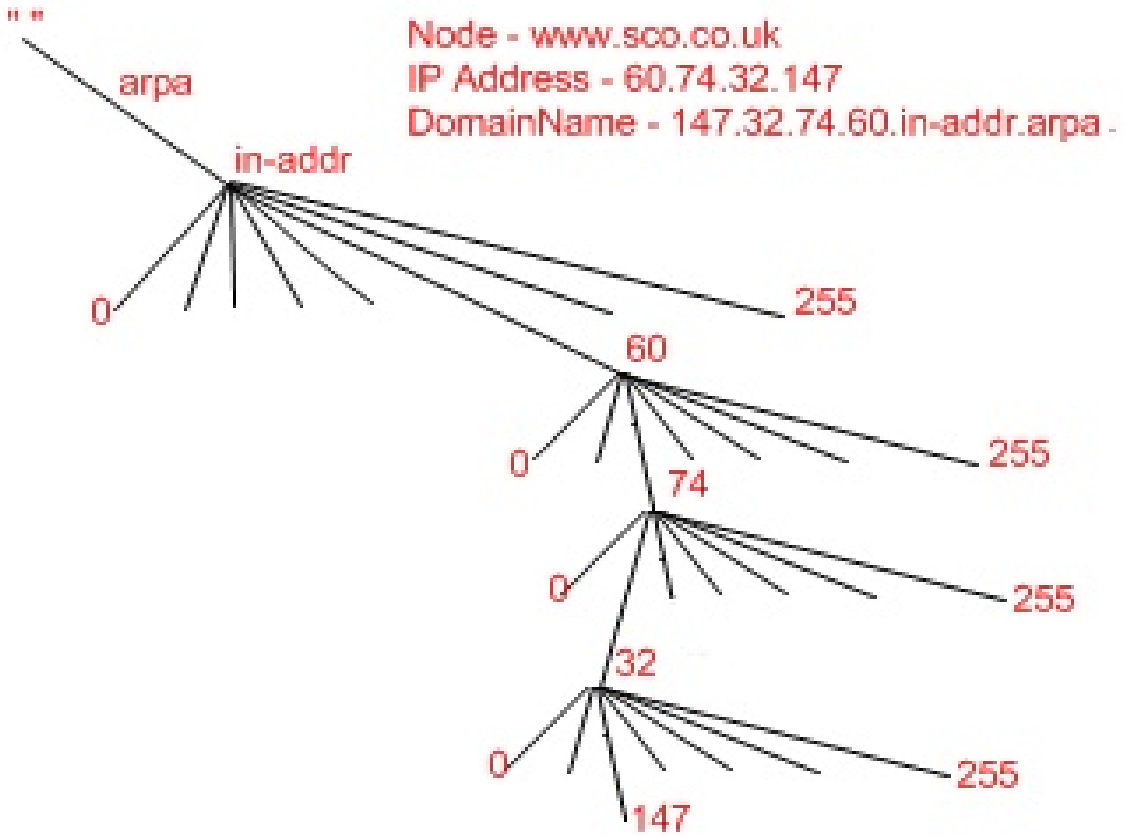
Query Flow


Client Resolver requires an address for www.sco.co.uk
Sends a recursive query to the internal nameserver
Receivers either an authoritative address or error
Internal Nameserver
Sends iterative query to the .UK nameserver
.UK nameserver returns address of the co.uk nameserver
.UK Nameserver
Sends an iterative query to the CO.UK nameserver
CO.UK Nameserver returns address of SCO.CO.UK nameserver
CO.UK Nameserver
Sends iterative query to the SCO.CO.UK nameserver
SCO.CO.UK nameserver returns either an authoritative address or an authoritative error
SCO.CO.UK Nameserver

Reverse Mapping

- The foregoing provides an effective way to find addresses from names
- We also need to be able to match a name to an address
  - This is especially important for security of services
  - Many services “insist” on matching IP addresses in request packets to advertised names
- This is done with a special domain:
  - in-addr.arpa

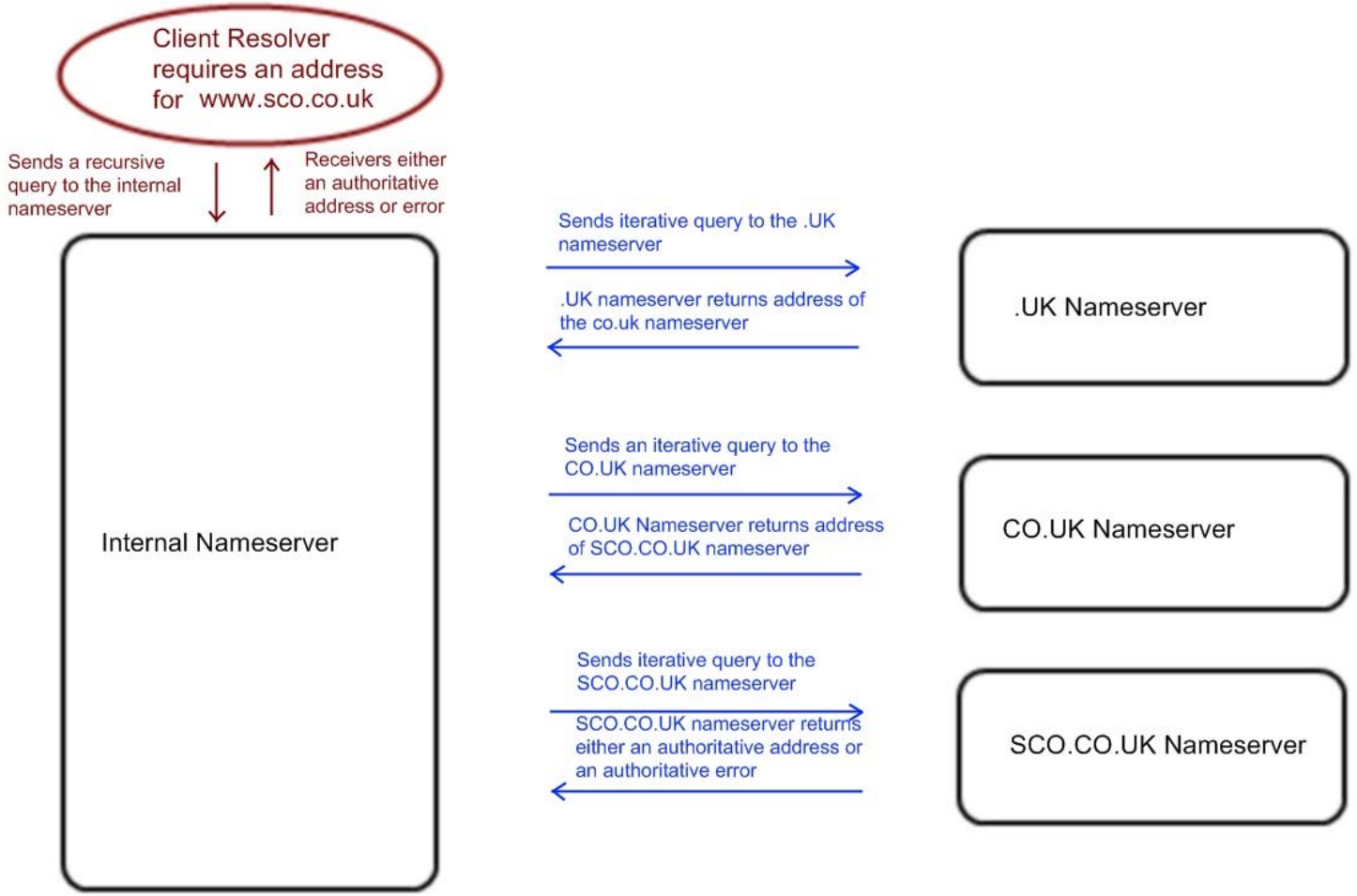
The in-addr.arpa domain



Caching

- To improve performance NameServers do not need to look up the same data over and over
- The Server will cache the information it finds for a lifetime the Domain Administrator has chosen, defined by the record's Time-To-Live (TTL)
- This also applies to negative answers, i.e. a response that the requested domain name doesn't exist

Query Flow - Caching



Setting Up Your Domain

- Choosing a domain Name
  - Select your desired top-level domain
  - Find a registrar
  - Create a unique and descriptive name
- Registering your Zones
- Building your NameServer(s)

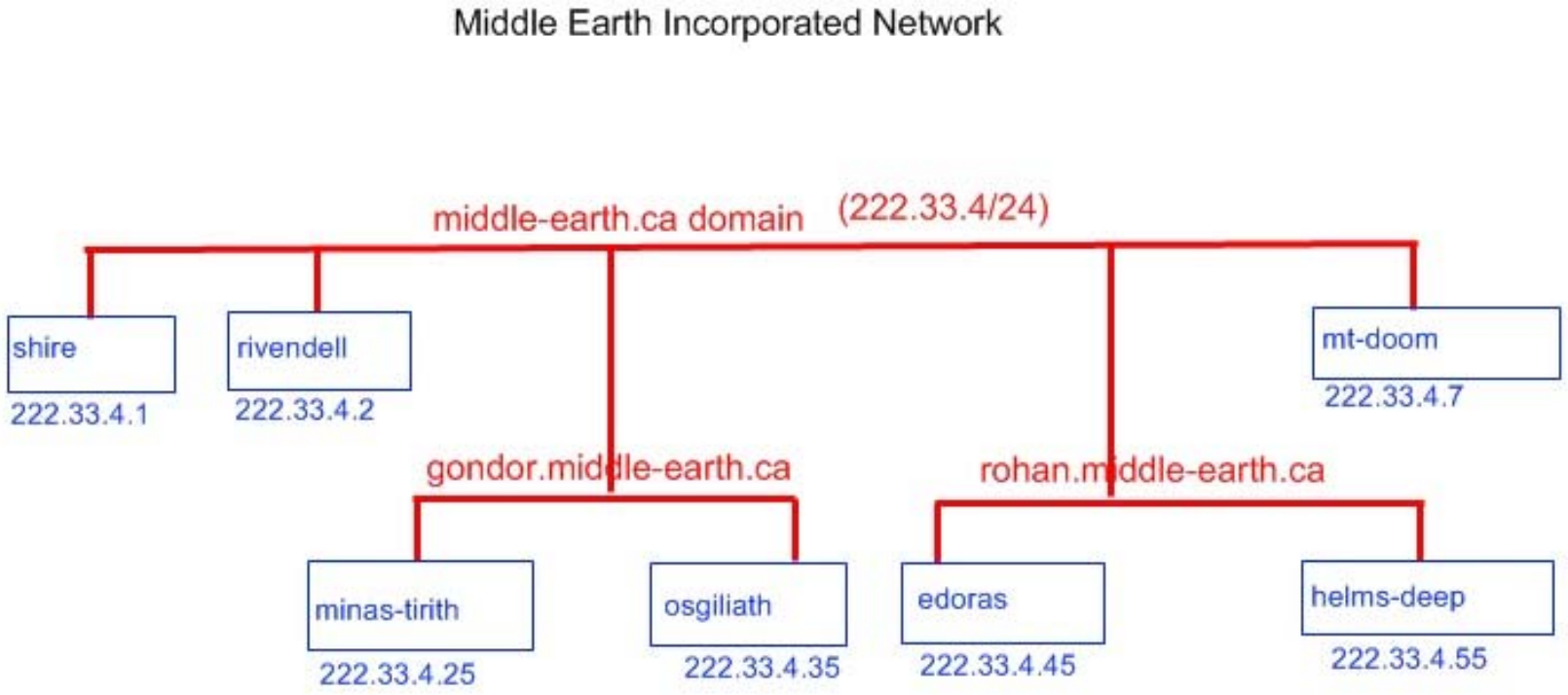
Choosing a Domain Name

- General guidelines
  - There are some restrictions imposed by registrars, but often not thoroughly enforced
    - i.e. .on.ca domains should be in Ontario, Canada
    - .ca domains should only be used by organizations with offices in multiple provinces
    - .com domains “should” be used by commercial organizations
    - .mil domains are restricted to the US Military (enforced)
  - As a good Internet citizen you should comply with these restrictions
  - Names can be any alpha-numeric characters as well as a hyphen in the middle of the name. Names may not include underscores and spaces

Choosing a Registrar

- There are many registrars, with varying levels of service
- Choose one with a good reputation, uptime, customer service
- Cost should be secondary
- Your registrar will probably be an authoritative NameServer for your Domain
- The registrar is going to require basic business information as well as contact information for key staff
- When in doubt – ask friends ☺
- Verify that your network is registered – if not nag your ISP

Creating our NameServer Files



Note – names use hyphens not underscores, apostrophes etc

Case is ignored but can be preserved

Creating our NameServer Files

- Several machines in this network serve multiple functions
  - Rivendell is the company mail Server
  - Rivendell is also an ftp server and is configured to respond to ftp.middle-earth.ca
  - Minas-Tirith is the companies webserver and must also respond to requests to www.middle-earth.ca
  - Minas-Tirith also has a backup mail server to relay our inbound email
  - The domain's primary NameServer is Shire, and the back-up is Mt-Doom

Setting up the DNS Configuration Files

- By default the DNS configuration files are located in /etc/named.d
- At minimum, we need to create
  - named.conf
  - A forward lookup file for your domain
  - A reverse lookup file for your domain
  - A forward lookup file for the loopback
  - A reverse lookup for the loopback
  - A root hints file (This is included by default on SCO servers and can be updated from the Corporate Download Site)

Creating our NameServer Files

- For each Domain we will create seven resource record (RR) types
  - SOA : Start of Authority
  - MX : Mail Exchanger Records
  - NS : NameServer
  - A : Address record
  - PTR : Name to Address Mapping
  - CNAME : Canonical Name or Alias
  - Other : other record types including TXT (text)
- Records are split into 3 files, named.soa, named.hosts and named.rev
- Server configuration data is in named.conf
- By default on SCO systems these files are in /etc/named.d

Creating our NameServer Files

- The controlling file named.conf (or named.boot) that tells the named daemon its configuration
- named.soa is a stub file that contains our start of authority for the domain.
- A separate soa file allows us to increment domain wide data in one file

Creating named.conf


```
Options {
    directory “/etc/named.d”;
};
zone “middle-earth.ca” in {
    type master;
    file “db.hosts”;
};
zone “4.33.222.in-addr.arpa in {
    type master;
    file “db.rev”;
};
zone “0.0.127.in-addr.arpa in {
    type            master;
    file            “db.rev”;
    notify no;
};
zone “.” in {
    type            hint;
    file            “db.cache”;
};
Zone “porn.com” in {
    type master;
    file            “db.block”;
};
```

Analyzing /etc/named.conf

- /etc/rc2.d/S85tcp will test for named.boot or named.conf and start ndc
- By default S85tcp looks in /etc/named.d
- Reads this file to configure which domains ndc will be authoritative for, and will provide location of root servers

Analyzing named.conf

- **Options {**
- **directory “/etc/named.d”;**
- **};**
- The Options section allows for multiple options to be set to apply to all domains for which the server is Authoritative
- The zone sections list information about each zone this server is authoritative
- **zone “middle-earth.ca” in {**
- **type master;**
- **file “db.hosts”;**
- **};**
- The middle-earth.ca zone section says
  - We are the master and authoritative for the domain middle-earth.ca
  - Its forward lookup data is stored in /etc/named.d in file db.hosts

Analyzing named.conf

```
zone “4.33.222.in-addr.arpa” in {
   type master;
   file “db.rev”;
};
```

- The 4.33.222.in-addra.arpa secion says
  - We’re master and authoritative for reverse lookups on this subnet
  - Data is stored in /etc/named.d/db.rev

```
zone “0.0.127.in-addr.arpa” in {
   type   master;
   file “db.127”;
   notify no;
};
```

- The 0.0.127.in-addr.arpa says we’re also authoritative for the loopback address space
- The ‘notify no’ option tells the server not to notify slaves if the serial number of this zone changes

Analyzing named.conf

```
zone “.” in {
        type      hint;
        file      “db.cache”;
};
```

- The . Zone section tells the server to go to the db.cache file for hints on any domain that it is not authoritative for
- This will cause an iterative query to be sent to the closest matched server listed in db.cache

Analyzing named.conf

```
Zone “nasty.com” in {
        type master;
        file        “db.nasty”;
};
```

- Are we really authoritative for nasty.com?
- Why would we do this?

Creating named.soa file

```
$TTL 21600
Middle-Earth.CA.  IN SOA Shire.Middle-Earth.CA. frodo.Middle-Earth.CA.
                  (
                  200804011245    ; Serial Number
                  3h              ; Refresh after 3 hours
                  1h              ; Retry after 1 hour
                  1w              ; Expire after 1 week
                  1h              ; Negative TTL cache time
                  )
                  IN NS    Shire.Middle-Earth.CA.
                  IN NS    Mt-Doom.Middle-Earth.CA.
```

Analyzing our SOA File

- $TTL 21600
  - This is the default TTL for this domain in seconds
  - Raise this if data is likely to be stagnant, lower it for testing

Analyzing our SOA File

- Middle-Earth.CA. IN SOA Shire.Middle-Earth.CA. frodo.Middle-Earth.CA.
- Middle-Earth.CA. is the domain this file is for
  - Note the trailing period
  - Case is insignificant but is preserved in responses
  - This could be replaced with an @ sign representing the $ORIGIN from named.conf (the second field in the zone statement)
- IN SOA
  - This is an Internet Start of Authority Record
- Shire.Middle-Earth.CA.
  - This is the primary master NameServer for the Domain
  - Note –the trailing period
- Frodo.Middle-Earth.CA.
  - Replace the first period with @ and this is the email address of the domain's administrator

Analyzing our SOA File

```
(
200804011245    ; Serial Number
3h              ; Refresh after 3 hours
1h              ; Retry after 1 hour
1w              ; Expire after 1 week
1h              ; Negative TTL cache time
)
```

- These are the variables for the SOA
- 200804011245, a serial number that should be incremented any time a DNS change is made

Analyzing our SOA File

- In order the timeouts are:
  - 3h - the Refresh tells any slave server that caches this data to re-query the master after 3 hours
  - 1h - the interval to wait to retry connecting to an unavailable master
  - 1w - Expire cached data after 1 week if you can't reload from the master
  - 1h - The time to cache negative data

Creating Forward lookup file – db.hosts


```
$INCLUDE                named.soa

Middle-Earth.CA.        1w      IN      MX 10 Rivendell.Middle-Earth.CA.
Middle-Earth.CA.        1w      IN      MX 20 Minas-Tirith.Middle-Earth.CA.
                                IN      TXT     “A wonderful place to visit”
Localhost               IN      A       127.0.0.1
Shire                           IN      A       222.33.4.1
Rivendell               IN      A       222.33.4.2
Mt-Doom.Middle-Earth.CA.  IN    A       222.33.4.7
Minas-Tirith.Gondor     IN      A       222.33.4.25
Osgiliath.Gondor        IN      A       222.33.4.35
Edoras.Rohan                    IN      A       222.33.4.45
Helms-Deep.Rohan        IN      A       222.33.4.55
Mail                            IN      CNAME   Rivendell
www                             IN      CNAME   Minas-Tirith
ftp                     IN      CNAME   Minas-Tirith
```

Analyzing db.hosts

```
$INCLUDE                named.soa

Middle-Earth.CA.        1w      IN      MX 10 Rivendell.Middle-Earth.CA.
Middle-Earth.CA.        1w      IN      MX 20 Minas-Tirith.Middle-Earth.CA.
```

- The $INCLUDE directive says to include the contents of named.soa in this position in the file
- The two MX records identify the mail exchangers for the domain middle-earth.ca
  - The 1w option says the TTL for these records only is 1 week, regardless of the TTL set by default in the SOA
  - The MX preferences 10 and 20 indicate the order in which mail is to be delivered. Servers will always try the lowest preference first
  - Multiple MX records with the same MX value will round-robin
  - Any RR with a domain name as its value must point to an A record, not a CNAME

Analyzing db.hosts

IN TXT “A wonderful place to visit”

- The txt record type can be used to insert any block of text up to 65535 characters describing the record

  Localhost IN A 127.0.0.1

- The localhost forward address is required to resolve connections to localhost or localhost.middle-earth.ca

Analyzing db.hosts

```
        IN    A          222.33.4.45
        Helms-Deep.Rohan                    Shire                                   IN
                          A          222.33.4.1
        Rivendell                           IN          A          222.33.4.2
        Mt-Doom.Middle-Earth.CA.            IN          A          222.33.4.7
        Minas-Tirith.Gondor                 IN          A          222.33.4.25
        Osgiliath.Gondor                    IN          A          222.33.4.35
        Edoras.Rohan                        IN          A          222.33.4.55
```

- Each node in the network requires a single A record
- You can create multiple A records as in

```
        Rivendell                           IN          A          222.33.4.2
                                            IN          A          222.33.4.3
                                            IN          A          222.33.4.4
        Mt-Doom.Middle-Earth.CA.            IN          A          222.33.4.7
```

- This effectively provides a rudimentary load-balancing solution
  - The server will round-robin shuffle responses to a query for Rivendell
  - Resolvers will cache the response for the TTL

Analyzing db.hosts

```
Mail        IN    CNAME   Rivendell
www         IN    CNAME   Minas-Tirith
ftp         IN    CNAME   Minas-Tirith
```

- CNAME records are Canonical Names, or aliases for actual node names
- CNAMES allow you to provide multiple identities for a single node
- There can be multiple CNAME records pointing to any node
- Do not create duplicate CNAMES

```
Mail        IN    CNAME   Rivendell
Mail        IN    CNAME   Lothlorien
ftp         IN    CNAME   Minas-Tirith
```

- This will break BIND 9.1 servers

Controlling Round-Robin

- Sometimes we do not want round-robin to work
  - Example, we may have a back-up WWW server on an old 486, but the primary is on a quad-core XEON
- In this scenario, we can add a an rrset options directive (note- this works in BIND 8.2+ and BIND 9.3+)

```
Options {
        rrset-order {
        class IN type A name “Minas-Tirith.Middle-Earth.CA” order
        fixed;
        };
};
```

- Lower the specific TTL for these records as well so that the back-up doesn’t get cached

Creating The db.rev file

```
@       IN      SOA Shire.Middle-Earth.CA. frodo.Middle-Earth.CA.
                (200804011245 3h 1h 1w 1h)
        IN      NS      Shire.Middle-Earth.CA
        IN      NS      Mt-Doom.Middle-Earth.CA

1.4.33.222.in-addr.arpa.    IN      PTR     Shire.Middle-Earth.CA.
2.4.33.222.in-addr.arpa.    IN      PTR     Rivendell.Middle-Earth.CA
7.4.33.222.in-addr.arpa.    IN      PTR     Mt-Doom.iddle-Earth.CA
25.4.33.222.in-addr.arpa.   IN      PTR     Minas-Tirith.Gondor.Middle-Earth.CA
35.4.33.222.in-addr.arpa.   IN      PTR     Osgiliath.Gondor.Middle-Earth.CA
45.4.33.222.in-addr.arpa.   IN      PTR     Edoras.Rohan.Middle-Earth.CA
55.4.33.222.in-addr.arpa.   IN      PTR     Helms-Deep.Middle-Earth.CA
```

Analyzing db.rev

```
@       IN      SOA Shire.Middle-Earth.CA. frodo.Middle-Earth.CA.
                (200804011245 3h 1h 1w 1h)
        IN      NS      Shire.Middle-Earth.CA
        IN      NS      Mt-Doom.Middle-Earth.CA
```

- The @ says that this SOA record applies ORIGIN defined in named.conf for this file. The ORIGIN is the second field in the named.conf record
- This is a separate domain and requires its own SOA data
  - SOA data can be shared using $INCLUDE directives
- We also need to define the NameServers that are authoritative for this zone. These might be different for forward and reverse lookups

Analyzing db.rev

```
1.4.33.222.in-addr.arpa.    IN    PTR    Shire.Middle-Earth.CA.
7                           IN    PTR    mt-doom.Middle-
Earth.CA.
```

- Each address in the zone requires one and only one PTR record
- DO NOT create multiple PTR records for a single address
- Multple PTR's worked for round-robin in BIND 4
  - In BIND 8+ this will break many apps
- DO NOT place CNAMES on the data side of the record

Managing the loop-back address

```
@                                IN      SOA     shire.middle-earth.ca
        frodo.middle-earth.ca            (200804011245   1w 1w 1w 1w)
                                 IN      NS      shire.middle-eath.ca
                                 IN      NS      mt-doom.middle-earth.ca
1.0.0.127.in-addr.arpa.          IN      PTR     localhost.
```

Creating db.nasty


```
@       IN      SOA     shire.middle-earth.ca       frodo.middle-earth.ca
                (200804011245 1w 1w 1w 1w)
        IN      NS      shire.middle-eath.ca
        IN      NS      mt-doom.middle-earth.ca
*       IN      CNAME   you-bad-person.middle-earth.ca.
```

Making db.nasty manageable

- Create a separate db.blocked file with records in the format:

```
Zone “nasty.com” in {
        type master;
        file        “db.nasty”;
};
Zone “competitor.com” in {
        type master;
        file        “db.nasty”;
};
```

- Replace the zone section for nasty.com in named.conf with the statement

```
$INCLUDE db.blocked
```

- Decide who can administer db.blocked

Creating a Slave Server

- Slave Servers can act as Master Authoritative servers for your domain(s) but don't need all of the configuration files
- To create a slave add entries to named.conf:

```
//this is a slave for Middle-Earth.CA
// it gets its data and updates from 222.33.4.2
zone "middle-earth.ca" {
    type slave;
    masters {222.33.4.2; 222.33.4.7;};
    file            "slave.middle-earth.ca";
};
```

- This will automatically create and maintain the domain file
- Similarly, you can set-up slave in-addr.arpa files
- Note – slaves should include 127.0.0.in-addr.arpa and root hints files as well

Creating a Caching Only Server

- Caching only servers will off-load your master servers
- This requires a named.conf, db.cache (root hints), and one zone file
- Named.conf structure

```
options {
        directory “/etc/named.d”;
};
zone “0.0.127.in-addr.arpa” {
        type master;
        file “127.rev”;
};
zone “.” {
        type hint;
        file “db.cache”;
};
```

Registering you NameServers

- You do not need to register all of your NameServers
- Only register NameServers you want to be accessible to the public
- It might make sense to have internal and external NameServers for your domain
- You can register a maximum of 10 NameServers for a domain

The Resolver

- The Resolver is the client side of DNS
- Typically you can configure
  - Your local client's domain name
  - What domains you want to search when a FQDN isn't used
  - The NameServers to ask for DNS information
- In UNIX you can also configure multiple resolvers, BIND, Hosts, NIS

Configuring Your Resolver - Unix

- The Resolver is configured in /etc/resolv.conf
- The are 6 possible directives
  - domain
  - hostresorder
  - search
  - nameserver
  - sortList
  - options
- If resolv.conf exists and points to at least one NameServer, DNS is used to resolve addresses
  - Otherwise hosts is used (or possibly NIS)

Configuring Your Resolver - Unix

Hostresorder Directive

- Hostresorder controls how the library routines gethostbyname and gethostbyaddr will resolve names and addresses

  hostresorder local bind nis

- This says to check hosts first, then DNS the NIS

  hostresorder bind nis / local

- This says to check DNS the NIS and if a response isn't found stop looking. The / delimiter forces the lookup to stop before the next database is checked

- If hostresorder is not specified the default is to search DNS, NIS then local
- On OpenServer 6 hostresorder is only recognized by OSR5 binaries

Domain Directive

```
domain middle-earth.ca
```

- This is the domain name that is appended to any host request that is not a FQDN
- This is also used by authorization files such as .rhosts, hosts.equiv and hosts.lpd
- If this isn't specified this is determined from everything after the first dot in the hostname
- The resolver will not report errors if this is incorrect
  - In older resolvers, trailing spaces are not allowed

Configuring Your Resolver - Unix

Search Directive

- The Search directive lists all of the domains that should be tried if a matching hostname is not found in the domain specified by the Domain Directive

  Search gondor.middle-earth.ca middle-earth.ca

- You can disable search by specifying a hostname with a trailing period

  ftp ftp.gondor.

- The default search list is only the local domain
- If the name contains a period, the resolver will assume it is a fully qualified domain name and try it before appending other domains

  ftp edoras.gondor

  - Will try ftp edoras.gondor first, then ftp edoras.gondor.gondor.middle-earth.ca then ftp edoras.gondor.middle-earth.ca

Configuring Your Resolver - Unix

The NameServer directive

```
nameserver 222.33.4.2
nameserver 222.33.4.7
```

- Each NameServer should be listed on a separate line
- By default you can specify a maximum of 3 NameServers
  - This is controlled by the kernel tunable MAXNS
- If you only have one NameServer,
  - the resolver will query it with a 5 second timeout
  - If it receives an ICMP port or host unreachable error, it doubles the 5 seconds and tries again
  - It will retry 4 times, 5 sec., 10 sec, 20 sec. for a total of 35 seconds then fall back to hosts
  - If it times out on even one query, it returns a null response and does not fall back. It will never fall back to hosts!!

Configuring your Resolver - Unix

- If you have two or more NameServers specified
  - If the resolver either times out or gets an error from the first NameServer, it queries the second NameServer
  - If it times out on all NameServers it changes the retry to int(10/number of servers)
    - So two NameServers – second retry is 5 seconds, 3 NameServers, 3 seconds etc.
  - Subsequent queries double the retry and continue
  - This may take up to 75 seconds total for BIND versions before 8.2.1
  - In BIND 8.2.1 and later, there is only one retry cycle so this is a maximum of 24 seconds
  - If all servers timeout, a null response is returned

Configuring your Resolver - Unix

- The sortlist directive
  - sortlist 128.32.42.0/255.255.255.0
  - This directive lets you specify a preferred network to use if the resolver gets multiple addresses for a host
  - For example
    - We have a dual homed server www.rabbits.com
    - Its internal address on the gigabit backbone is 128.32.42.155
    - Its external address is 142.117.195.23
    - The DNS server for rabbits.com returns both addresses
    - The above sortlist directive says to prefer the 128 address
    - You can specify up to 10 networks to “prefer”

Configuring your Resolver - Unix

- The Options Directive
  - Allows you to vary the behaviour of the resolver
  - Turn on debug mode

    options debug
  - Boost the default number of retry attempts

    options attempts:4
  - Increase or decrease your timeout

    options timeout:2
  - Force the resolver to use all NameServers in round-robin

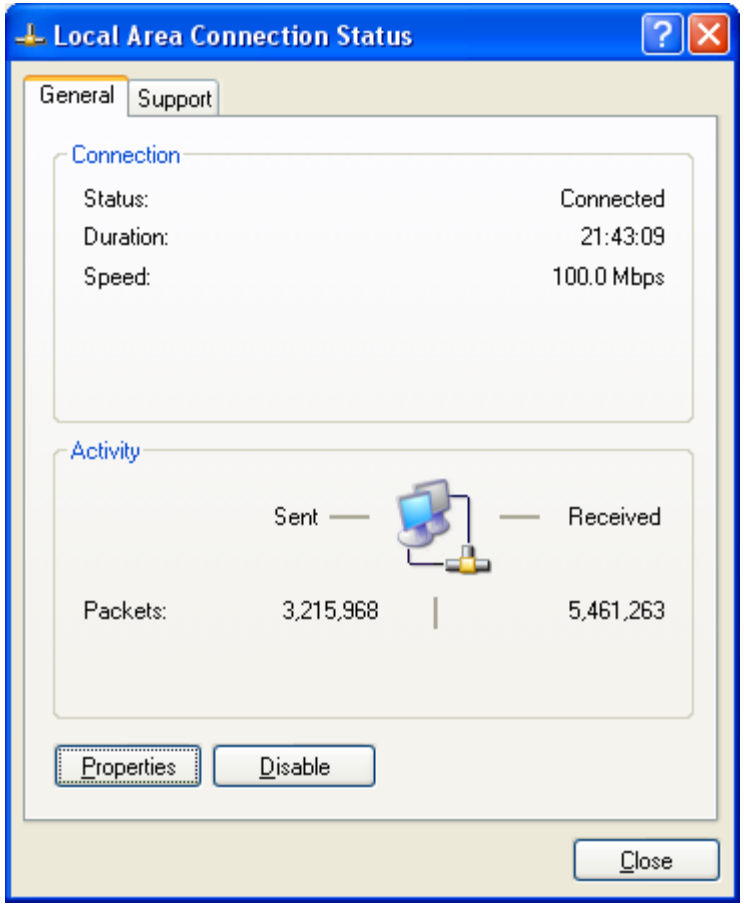
    options rotate
  - Increase the number of dots that trips a root server lookup

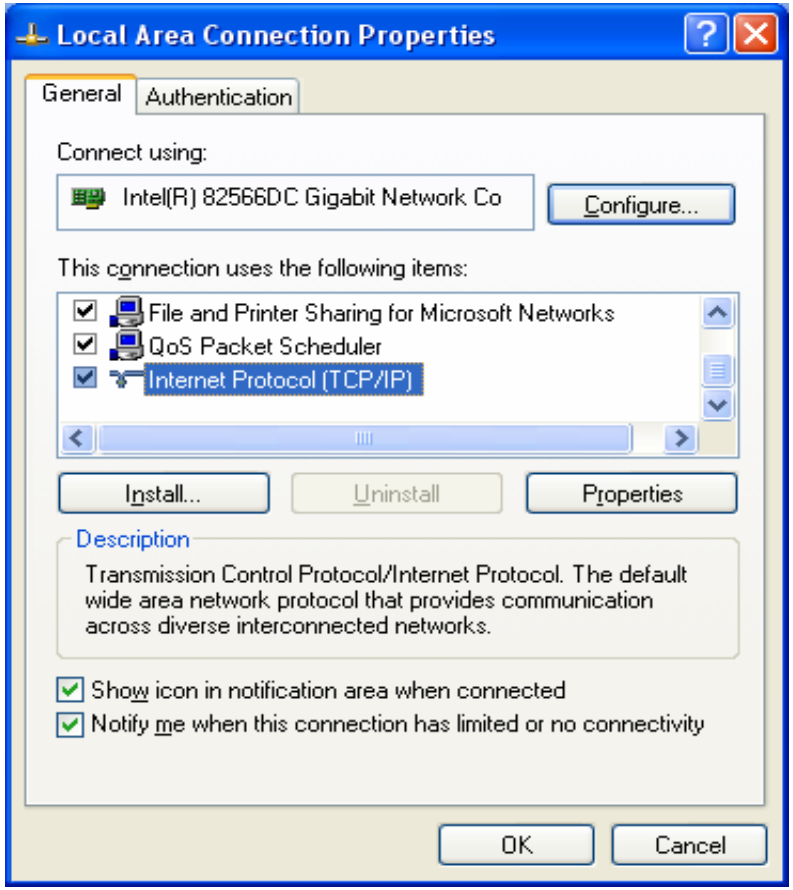
    options ndots:2

Resolver – Unix Service Behaviours

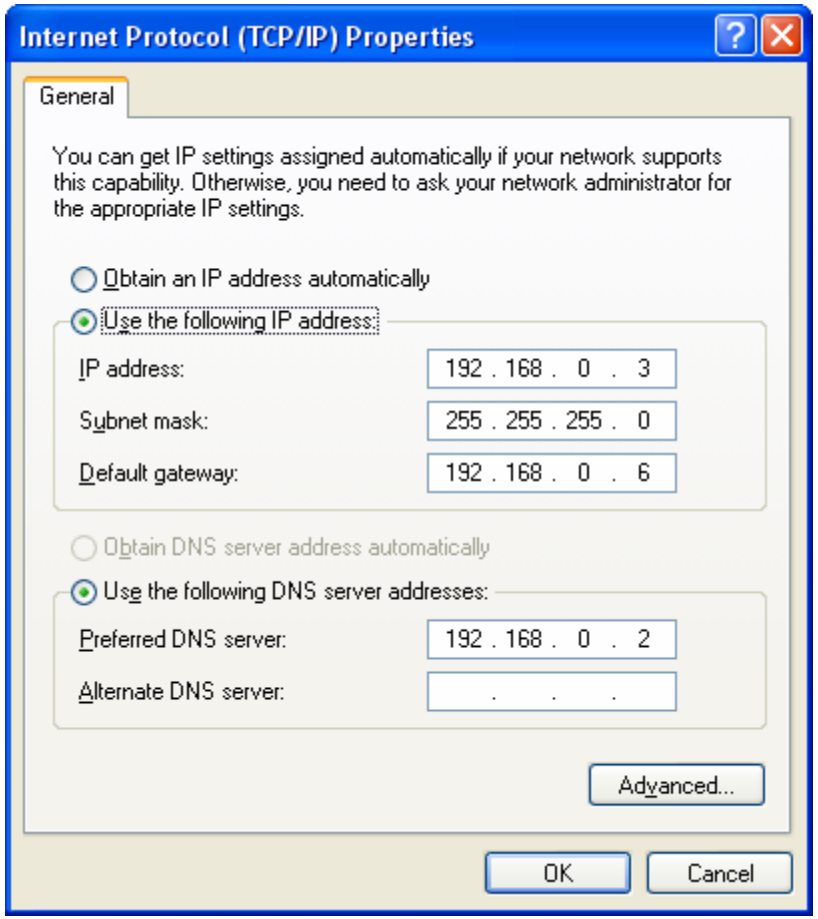
- telnet, ftp, rlogin and rsh will apply the searchlist
- NFS - /etc/exports & netgroup client name must match the client's hostname
- Sendmail
  - Assumes ndots:1 and does an ANY query
  - If it gets a CNAME, it resolves that and then canonicalizes it
  - If it gets a A record, it takes the address's domain name
  - If it finds an MX record it uses the domain name of the MX record
- Always create an MX record!!!

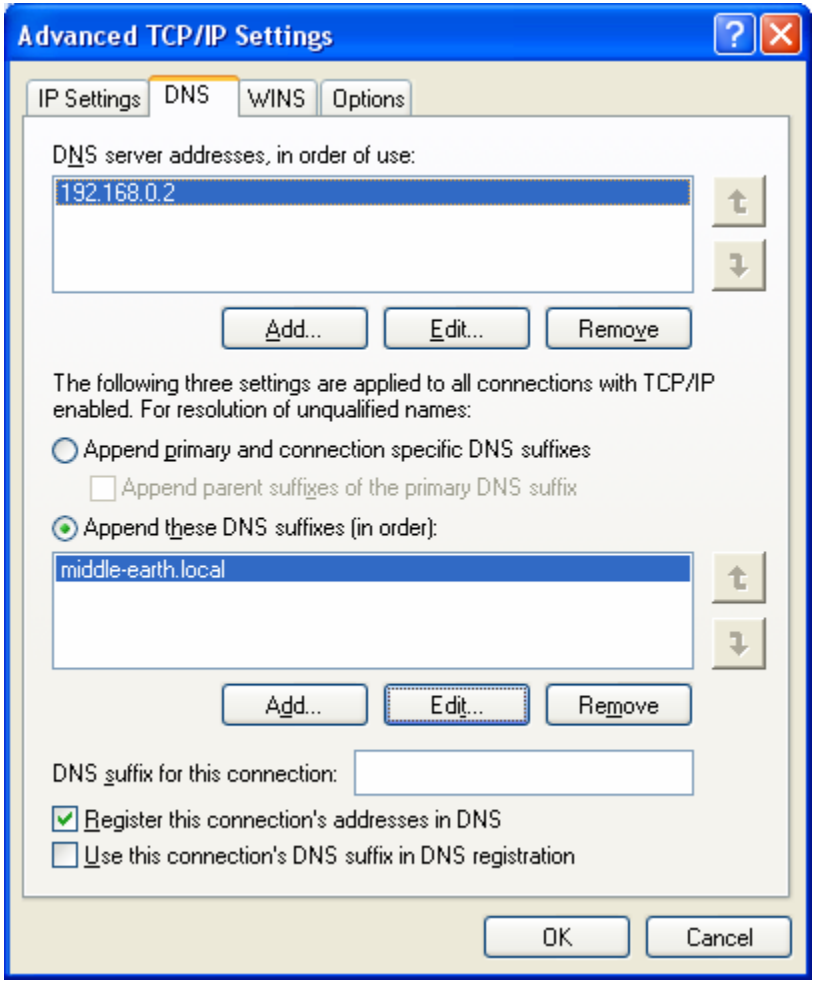
Configuring Your Resolver - Windows





Configuring your Resolver - Windows





Dealing with Microsoft

- If you have a Microsoft Active Directory Domain
- Use the AD server as your primary DNS Server

Testing your Server

- Once the DNS configuration files are created you can start the server by restarting TCP, rebooting the server or manually running /etc/ndc
- We can test with either nslookup or dig
  - Windows does not support dig

Using nslookup

- nslookup will allow you to send iterative queries to NameServers
- It will only connect to one server at a time, it will not fail-over on timeouts or errors
- nslookup actually performs a zone transfer, but it does not check serial numbers

- To use nslookup in non-interactive mode

```
#nslookup www.sco.com
Server Rivendell.Middle-Earth.CA
Address 222.33.4.2

Non-authoritative answer:
Name:    www.sco.com
Address: 132.147.63.12
```

- To use it in interactive mode

```
# nslookup
Default Server:  Rivendell.Middle-Earth.CA
Address:222.33.4.2
```

Using nslookup

- Authoritative responses
  - The first lookup of a name causes the server to go to the master for that domain. The response is flagged as authoritative
  - Subsequent lookups, for the duration of the TTL are flagged non-authoritative

Using nslookup

- Iterative Commands
  - server 222.33.4.7
  - server Mt-Doom
  - set querytype=MX, A, SOA, TXT, PTR
  - set port 953
  - set timeout=35
  - set retry=6
  - Set root=mt-doom.middle-earth.ca
  - set debug/set nodebug
  - set norecurse
  - set novc (set TCP queries rather than UDP)

nslookup errors

nslookup middle-earth.ca

Server: rivendell.middle-earth.ca

Address: 222.33.4.2

- rivendell can't find middle-earth.ca: Non-existent host/domain name
- There is no DNS record for that name of that type
- To find record types you can list the zone

  # nslookup

  Server: rivendell.middle-earth.ca

  Address: 222.33.4.2

  ls middle-earth.ca.
- Can't list middle-earth.ca: Unexpected Error
  - (there is no DNS server running)
- If there is no PTR record for your Server

  # nslookup

  Can't find server name for rivendell.me.local: non-existent host/domain
- Unspecified error
  - typically occurs because there is too much data for a UDP packet (65519 bytes)

Using dig

- dig – Domain Information Groper
- On SCO servers nslookup is now considered obsolete and using dig is recommended
- dig doesn't use searchlists
- You can get a full zone transfer with dig with arguments axfr
- dig @rivendell.middle-earth.ca middle-earth.ca axfr
- This will display all data about a zone
- See the man pages

How DNS can Break Things

- Email
  - Do a reverse DNS lookup on the sending ip – No PTR: mail rejected
  - Does the PTR record match the to the HELO / EHLO hostname? – No: Reject mail
  - Does the ip address's "PTR hostname" have an A record? – No:Reject mail
  - Does the ip address's "PTR hostname" have an MX record?- No: Reject Mail
  - Is the reverse delegation correct?-No: reject mail.
- The bad news – these are all silent ☺
- If your server sends an ehlo
  - Does the hostname resolve to an A record? No: reject mail.
  - Does the hostname resolve to an MX record? No: reject mail.
  - Is the hostname a FQHN: Fully Qualified Host Name? No: Reject Mail
- The good/bad news – these may or may not be silent ☺

Sender Policy Framework - SPF

- SPF originated with Microsoft, now public domain
- This is a protocol to allow receiving mail domains to verify that a sending email server is authorized to send email for a given domain
- Many email servers are adopting SPF as an anti-spam and ant-phishing tool
- SPF data is placed in TXT records
- Formats:

  Middle-Earth.CA IN TXT “v=spf1 +mx –all”
  - This indicates any MX relay for Middle-Earth.CA can send email from user@middle-earth.ca

  Middle-Earth.CA IN TXT “v=spf1 ip4:222.33.4.2 –all”
  - This will reject any inbound mail not from 222.33.4.2

Firewall Issues

- Does your firewall allow traffic on port 53
- Does your upstream provider allow access
- Are you being throttled
- Is load-balancing on any firewall on port 53
- If you use commands that require a zone transfer, are these allowed

Questions