[DNS](http://shabbathster.blogspot.com/2013/09/dns.html)

                         ====================  
                         A.  DNS Architecture  
   Domain Name Service  
   
                         ====================         
  
  The Domain Name System  
  using ISC BIND's named\*  
  ======================  
  
  
      I INTRODUCTION  
      ===============  
  
\*  What is DNS ?  
  
   Domain Name System (DNS) converts the name of a Web site (www.vadapav.com)  
   to an IP address (65.115.71.34).  
   This step is important, because the IP address of a Web site's server,  
   NOT THE WEB SITE's name, is used in routing traffic over the Internet.  
   And they may or may not be the same !!  
  
   Here we shall see how to configure your own DNS server to help guide  
   Web surfers to your site.  
  
\*  What is a DNS domain ?  
  
   Everyone in the world has a first name and a last, or family name. The same  
   thing is true in the DNS world: A family of Web sites can be loosely  
   described a domain.  
  
   For example, the domain vadapav.com has a number of children, such as  
   www.vadapav.com and mail.vadapav.com for the Web and mail servers,  
   respectively.  
  
\*  What is BIND ?  
  
   BIND is an acronym for the Berkeley Internet Name Domain project, which is a  
   group that maintains the DNS-related software suite that runs under Linux.  
   The most well known program in BIND is named\*, the daemon that responds to  
   DNS queries from remote machines.  
  
     II CLIENTS-SIDE DNS  
        ================  
  
\*  What is a DNS client ?  
  
   The file /etc/resolv.conf is the DNS client.  
  
   A DNS client doesn't store DNS information; it must always refer to a DNS  
   server to get it. The only DNS configuration file for a DNS client is the  
   /etc/resolv.conf file, which defines the IP address of the DNS server it  
   should use. You shouldn't need to configure any other files.  
  
\*  How do I configure this resolver ?  
  
      The /etc/resolv.conf File - The DNS resolver  
      =========================  
  
   DNS clients (servers not running BIND) use the /etc/resolv.conf file to  
   determine both the location of their DNS server and the domains to which  
   they be long.  
  
   The file generally has two columns; the first contains a keyword, and the  
   second contains the desired values separated by commas.  
  
       keyword       value  
       =======       =====  
  
     nameserver        IP address of your DNS nameserver.  
  
     domain        The local domain name to be used by default.  
        If the server is ganesh.bom.labs.net, then the entry  
        would just be "bom.labs.net"  
  
     search        If you refer to another server just by its name without  
        the domain added on, DNS on your client will append the  
        server name to each domain in this list and do a DNS  
        lookup on each, to get the remote servers' IP address.  
   This is a handy time-saving feature to have so that  
        you can refer to servers in the same domain by only their  
        servername without having to specify the domain.  
        The domains in this list must separated by spaces.  
  
IMP:        The first domain listed after the search directive must  
        be the home domain of your network, in this case  
        vadapav.com.  
Example :  
  
    /etc/resolv.conf  
    ================  
    search vadapav.com vadapav.net vadapav.org  
        or  
    domain vadapav.com vadapav.net vadapav.org  
  
    options timeout:1  
  
    options attempts:1  
  
    nameserver 192.168.1.100  
    nameserver 192.168.1.102  
  
  
Note : Placing a domain AND search entry in /etc/resolv.conf is, therefore,  
       redundant  
  
\*  How do I test a DNS server and query it for an IP or hostname and other  
   info ?  
  
     III CLIENTS-SIDE DNS TS TOOLS  
        ==========================  
  
   There are 3 imp commands for clients to use : host, nslookup and dig  
  
\*  The host command  
   ================  
  
   To perform a forward lookup :  
  
   # host google.com  
  
   > google.com has address 64.233.187.99  
   > google.com has address 64.233.167.99  
   > google.com has address 72.14.207.99  
  
   To do a reverse lookup :  
  
   # host 64.233.187.99  
  
   > 99.187.233.64.in-addr.arpa domain name pointer google.com.  
  
   As can be seen, the forward and reverse entries don't match.  
   The reverse entry matches the entry of the ISP.  
  
\*  The nslookup command [ -sil ---> silent ]  
   ===================  
   The nslookup command provides the same results on Windows PCs.  
  
\*  # nslookup -sil  
     > set all  
     or  
     > .     <------------- def="" ns="" p="" your="">  
     Default server: 192.168.1.1  
     Address: 192.168.1.1#53  
  
     Set options:  
     novc    nodebug    nod2  
     search    recurse  
     timeout = 0    retry = 2    port = 53  
     querytype = A    class = IN  
     srchlist =  
  
     > 192.168.1.1  <------------- def="" ns="" p="" your="">     Server:      192.168.1.1  
     Address:      192.168.1.1#53  
  
     Non-authoritative answer:  
     1.1.168.192.in-addr.arpa      name = mygateway1.ar7.  
  
     > set timeout=5   Resend dns querie after 5 secs, 10 secs, 20, secs etc  
  
     > set [no]debug   Shows timeouts and displays response msgs  
  
     > set [no]defname   Add local domainname to names w/o dots first  
   and ignore search list of /etc/resolv.conf  
   Only if search is turned off  
  
     > set [no]search   Search list supersedes local domain name [defname]  
   By def nslookup appends domain names in the search  
   list to names that don't end in a dot  
  
     > set [no]d2   level 2 debugging. Turns on debug too  
  
     > set [no]vc   send dns queries using TCP not UDP [Virtual Circuit]  
  
     > root=a.root-servers.net Switch def NS to root name server  
       aka  
     > server=a.root-servers.net  
  
     > set retry=5   Retry 5 times. After each retry, the timeout value  
   is doubled  
  
\*  What is my name server's IP ?  
  
   # nslookup -sil .  
     Server:      192.168.1.1  
     Address:      192.168.1.1#53  
  
     Non-authoritative answer:  
     \*\*\* Can't find .: No answer  
  
\*  To do a forward lookup :  
  
   # nslookup -sil google.com  
  
     Server:      192.168.1.1  
     Address:      192.168.1.1#53  
  
     Non-authoritative answer:  
     Name:   google.com  
     Address: 64.233.187.99  
  
\*  To do a reverse lookup :  
  
   # nslookup -sil 64.233.187.99  
  
     Server:      192.168.1.1  
     Address:      192.168.1.1#53  
  
     Non-authoritative answer:  
     99.187.233.64.in-addr.arpa      name = google.com.  
     Authoritative answers can be found from:  
  
\*  Looking up different records :  
  
   # nslookup -sil  
  
     > 192.168.1.1  
  
     Server:      192.168.1.1  
     Address:      192.168.1.1#53  
  
     Non-authoritative answer:  
     1.1.168.192.in-addr.arpa      name = mygateway1.ar7.  
  
  
     > mygateway1.ar7.  
       Server:        192.168.1.1  
       Address:        192.168.1.1#53  
  
     Non-authoritative answer:  
     Name:   mygateway1.ar7  
     Address: 192.168.1.1  
  
     > set q=soa  
     or  
     > set type=soa  
  
     > set q=A  
     > set q=CNAME  
     > set q=NS  
     > set q=PTR  
  
\*  The dig command - The Domain Information Groper  
   ===============  
  
   \* dig never applies the search list as nslookup does  
   \* dig has no interactive mode  
  
\*  Looking up different records :  
  
   # dig yahoo.com  
  
   # dig yahoo.com mx  
  
   # dig yahoo.com soa  
  
   # dig @a.root-servers.net ns .  
  
     ; <<>> DiG 9.2.1 <<>> @a.root-servers.net ns .  
     ;; global options:  printcmd  
     ;; Got answer:  
     ;; ->>HEADER<<- 10875="" id:="" noerror="" opcode:="" p="" query="" status:="">     ;; flags: qr aa rd; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13  
  
     ;; QUESTION SECTION:  
     ;. IN NS  
  
     ;; ANSWER SECTION:  
     . 518400 IN NS G.ROOT-SERVERS.NET.  
     . 518400 IN NS F.ROOT-SERVERS.NET.  
     . 518400 IN NS B.ROOT-SERVERS.NET.  
     . 518400 IN NS J.ROOT-SERVERS.NET.  
     . 518400 IN NS K.ROOT-SERVERS.NET.  
     . 518400 IN NS L.ROOT-SERVERS.NET.  
     . 518400 IN NS M.ROOT-SERVERS.NET.  
     . 518400 IN NS I.ROOT-SERVERS.NET.  
     . 518400 IN NS E.ROOT-SERVERS.NET.  
     . 518400 IN NS D.ROOT-SERVERS.NET.  
     . 518400 IN NS A.ROOT-SERVERS.NET.  
     . 518400 IN NS H.ROOT-SERVERS.NET.  
     . 518400 IN NS C.ROOT-SERVERS.NET.  
  
     ;; ADDITIONAL SECTION:  
     G.ROOT-SERVERS.NET. 3600000 IN A 192.112.36.4  
     F.ROOT-SERVERS.NET. 3600000 IN A 192.5.5.241  
     B.ROOT-SERVERS.NET. 3600000 IN A 192.228.79.201  
     J.ROOT-SERVERS.NET. 3600000 IN A 192.58.128.30  
     K.ROOT-SERVERS.NET. 3600000 IN A 193.0.14.129  
     L.ROOT-SERVERS.NET. 3600000 IN A 198.32.64.12  
     M.ROOT-SERVERS.NET. 3600000 IN A 202.12.27.33  
     I.ROOT-SERVERS.NET. 3600000 IN A 192.36.148.17  
     E.ROOT-SERVERS.NET. 3600000 IN A 192.203.230.10  
     D.ROOT-SERVERS.NET. 3600000 IN A 128.8.10.90  
     A.ROOT-SERVERS.NET. 3600000 IN A 198.41.0.4  
     H.ROOT-SERVERS.NET. 3600000 IN A 128.63.2.53  
     C.ROOT-SERVERS.NET. 3600000 IN A 192.33.4.12  
  
     ;; Query time: 319 msec  
     ;; SERVER: 198.41.0.4#53(a.root-servers.net)  
     ;; WHEN: Mon Sep 11 23:08:35 2006  
     ;; MSG SIZE  rcvd: 436  
  
   # dig -x 198.41.0.4  
  
     ; <<>> DiG 9.2.1 <<>> -x 198.41.0.4  
     ;; global options:  printcmd  
     ;; Got answer:  
     ;; ->>HEADER<<- 49618="" id:="" noerror="" opcode:="" p="" query="" status:="">     ;; flags: qr rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0  
  
     ;; QUESTION SECTION:  
     ;4.0.41.198.in-addr.arpa.      IN      PTR  
  
     ;; ANSWER SECTION:  
     4.0.41.198.in-addr.arpa. 10000  IN      PTR     a.root.  
  
     ;; Query time: 3 msec  
     ;; SERVER: 192.168.1.1#53(192.168.1.1)  
     ;; WHEN: Mon Sep 11 23:48:02 2006  
     ;; MSG SIZE  rcvd: 61  
  
   # dig -x 198.41.0.4  
  
  
####################################################################  
[i] /etc/hosts [HOSTS.TXT]  
    ----------------------  
Users like names , NWs like numbers  
  
Also called Local resolver or local DNS file  
  
Use of Aliases - historic, generic [www,mail], alternate names [spellings etc]  
  
[ii] Uses of /etc/hosts  
-----------------------  
  
   \* Primary names are used by system to do reverse lookups [ IP - names ]  
   \* Reverse lookups are reqd to create more readable displays  
  
             # netstat --inet [-net]  
  
Active Internet connections (w/o servers)  
Proto Recv-Q Send-Q Local Address           Foreign Address         State       
tcp        0      0 ganesh.bom.labs.n:32822 ganesh.bom.labs.net:ftp ESTABLISHED  
tcp        0      0 ganesh.bom.labs.net:ftp ganesh.bom.labs.n:32822 ESTABLISHED  
tcp        0      0 localhost.localdom:smtp localhost.localdo:32823 TIME\_WAIT    
tcp        0      0 localhost.localdom:smtp localhost.localdo:32824 TIME\_WAIT    
  
Displays open TCP/IP connections and hosts/Ports involved in the connection  
  
Why use /etc/hosts when we have DNS ?  
  
A: DNS may not be available at boot time, immediately  
  
[iii] Limitations of /etc/hosts  
-------------------------------  
  
 \* Flat file, easy to read/edit, hard to search  
 \* not indexed or encrypted  
 \* Central maint. reqd for new entries [NIC - HOSTS.TXT]  
    \* Govt agency  
    \* Enter a whale of new m/c entries everyday  
    \* Download daily for latest version from NIC so traffic-problems  
                       
                         ================  
                         B. DNS Hierarchy  
                         ================  
[i] Structure of DNS  
--------------------  
  
         A  Unix FS v/s DNS Database Comparision study  
         =============================================  
  
    \* Distributed, hierarchical DB  
    \* Localized, not centralized maint. reqd  
  
    eg     / etc / httpd / conf / httpd.conf  <--------- filename="" p="">           /etc  / httpd / remote / httpd.conf  
          |  
          root of FS        
          ----------------------->  
      <--- ---="" file="" generic="" nbsp="" specific="" to="">  
  
    A hostname on the Internet is exactly the opposite  
     
    eg     willy.dolphin.mammals.org. <-------- denotes="" dns="" dot="" of="" p="" root="" tree="">            |      |       |      |  
      hostname sub-domain domain TLD      
          <----------------------- p="">      <--- ---="" generic="" host="" nbsp="" specific="" to="">  
  
willy    -  Name of the computer [hostname -s]  
dolphin  -  sub-domain under 'mammals'  
mammals  -  Domain we have purchased under 'org'  
org      - A TLD  
.        - Root node  
  
    . or root node is like /, the root of the FS  
  
    Dirs are like domains or more precisely sub-domains  
     
    Files are like hosts - or computers with IPs and hostnames  
     
    Each domain can be further divided or partitioned in to subdomains, just  
    like a dir can be further subdivided into subdirs.  
  
    Directores use '/' as the seperator. Domains are seperated by '.'.  
   
Note:  
    Subdomains are like dirs unders a parent dir but this dir is not any  
    normal dir but appears as a dir bcos it is like a NFS mounted share.  
    A dir which is a FS system on some other host. Can be detached but once  
    mounted on a particular FS, becomes part of that.  
  
    Like every dir, a domain name has a unique name and identifies its  
    position in the db; Much as a dir's Abs-PN  identifies its place in the  
    FS.  
    ---------------------------------------------------------------------  
    A domain is a sequence of labels from the node at the root of the domain  
    to the root of the whole tree, with the labels separated by dots.  
  
    In Unix, a dir's abs PN is a list of relative names read from root '/' to  
    leaf [opp dir to DNS], using a slash to seperate the names.  
  
    Just as one can have two files with the same name in seperate dirs,  
    so can one have two similar hosts but in different domains [nodes].  
  
    ---------------------------------------------------------------------  
    DNS requires that sibling nodes - nodes that are children of the same  
    parent - have different labels - This is to ensure uniqueness.  
     
      eg  willy.dolphin.mammals.org  
          willy.whale.mammals.org  
      
       Here the parent is 'mammals' and has 2 siblings - 'dolphin' and  
   'whale'. These have to have different labels.  
    
          The above 2 are names of 2 machines i.e. they are 2 completely  
   different hosts.  
  
Now examine the similarity with the Unix FS :  
  
    Similarly, the Unix FS, requires that sibling dirs or files in the same  
    dir have different names. This is to ensure uniqueness.  
  
           Like /usr/local/funny and /usr/bin/funny  
  
       Here the parent is '/usr' and has 2 siblings - 'local' and  
   'bin'. These have to have different labels.  
    
          The above 2 are names of 2 subdirs i.e. they are 2 completely  
   different directories and subsequently 2 different files although  
   they have the same name.  
  
    ---------------------------------------------------------------------  
     
                    Domains and Domain names  
                    ------------------------  
  
    Domains : A subtree of the domain name space  
              The domain name of a domain is the same as the domain name of  
              the node at the very top of the domain.  
  
                             "." [root node]    
                              |  
                 -----------------------------  
                 |            |         |    
                edu          org       com  
                |  
                |  
          ------\*<---------- mumbai.edu.="" nbsp="" node="" p="">          |   mumbai    |    or the domain name of the mumbai.edu domain  
          |             |  
          |       <------------------- domain="" mumbai.edu="" name="" nbsp="" p="" space="">          |             |  
          |             |  
           -----\*-------  
  
Check out the comparision with the Unix FS :  
  
                             "/" [root node]    
                              |  
                 -----------------------------  
                 |            |         |    
                usr          bin       usr  
                |  
                |  
          ------\*<---------- bin="" nbsp="" node="" p="" usr="">          |    bin      |    or the dir name of the /usr/bin dir  
          |             |  
          |       <------------------- bin="" dir="" p="" space="" usr="">          |             |  
          |             |  
           -----\*-------  
  
      -------------------------------------------------------------------  
  
      A domain name can also be in many domains.  
  
      Eg the domain name  "maths.mumbai.edu" is a part of the "mumbai.edu"  
         domain as well as of the "edu." domain which is once again a part of  
  the . domain.  
  
  All DNS servers are in the . domain  
  
                             "." [root node]    
                              |  
                 -----------------------------  
      +-------------------+    |         |    
      |         edu       |  org       com  
      |         |         |  
      |         |         |<----------- domain="" edu="" nbsp="" p="">      |  ---------------  |                 
      |  |   mumbai    |<--------------- domain="" mumbai.edu="" p="">      |  |  ---------  |  |  
      |  |  | maths <----- domain="" maths.mumbai.edu="" p="">      |  |  | bio   <----- bio.mumbai.edu="" domain="" nbsp="" p="">      |  |  ---------  |  |  
      |  |-----\*-------|  |  
      |                   |  
       -------------------  
  
                        SubDomain Delegation  
                        --------------------  
  
  One of the main goals of the DNS was to decentralize administration.  
  How is this done ?  
  
  Consider the CEO of a large Corp. How does she delegate responsibility?  
  
                               CEO  [BOSS]     [ "." root node or / ]  
                                |  
                ----------------------------------  
               |           |            |         |      
              MKTG       SALES         HRD      ACCTS [Depts or gTLDs or TLDirs]  
               |  
               | Mr M  
         -----------------------  
        |           |          |  
        LOCAL     INTL      PLANNING  
        |                                  [sub-Depts or sub-domains delegated  
        |                                   by Mr M to Mr L, Mr I, Mr P]  
        Mr L      Mr I       Mr P    
         |  
   -----------------------        
   |           |         |  
   RJ         TN         WB  
   Mr R       Mr T      Mr W  
                      ---------  
       |       |  
       KOL     DARJEELING  
                       |  
                   Mr WW [WW.KOL.WB.LOCAL.MKTG.CEO]  
  
   She breaks up the org into Depts. Each with its own head.  
    
   The Head has total responsibility for his Dept.  
    
   The Dept is created by the CEO and hence cannot be made without the CEO  
   knowing about it.  
    
   That is, its made by the CEO and total responsibility is delegated to the  
   Dept Head to handle his Dept.  
    
   The Dept Head CAN create more sub-Depts under his Dept, without consulting  
   the CEO. He has total authority over his Dept [domain]  
   He is said to be authorative over his Dept [domain]  
  
   CEO only knows about Depts but nothing about the sub-Depts [sub-domains].  
    
   Why ?  
    
   She does not have to.  
    
   That would mean redundancy of information [extra work].  
   All she has to know is the Dept Head and she can procure whatever info she  
   needs about sub-Depts etc from the respective Dept Head concerned.  
    
   After all, what is he being paid for if she has to keep all his sub-Dept  
   information !! And all other sub-Depts too.  
  
   This is called "Department Delegation".  
  
   The parent Dept [domain] retains only pointers to sources of the  
   sub-Depts [sub-domains] data, so that it can refer queries there.  
  
  
   Now what would happen if I asked the CEO his Question ?  
  
   Where is Mr W or  WB.LOCAL.MKTG.CEO ?  
  
   The CEO would not directly answer this query.  
    
   It would refer me to MKTG.CEO. i.e. to Mr M  
    
   I would then have to ask [Mr M] MKTG.CEO which would refer me to  
   LOCAL.MKTG.CEO. [Mr L]  
    
   I would then have to ask Mr L who would finally direct me to Mr W at  
   WB.LOCAL.MKTG.CEO  
  
   I have found my man !  
  
   This referral business is called 'recursive querying' and most DNS  
   servers are non-recursive , in that, they just put you on the path to  
   another server.  
  
   ----------------------------------------------------------------------  
  
   NOW LET'S SEE HOW DNS DOES DOMAIN DELEGATION   
  
                             "." [root node]    
                              |  
                 -----------------------------  
                 |            |         |    
                edu          org       com  
                |  
   -------------------------------  
  |           |           -------|---------       
  kolkatta chennai        |    mumbai     |  
                          |      |        |  
                          | ------------  |     
                          | |          |  |  
                          | kalina    fort|  
                          |               |  
                          -----------------  
  
  The "." is controlled by ICANN.  [Int'l Corp for Assigned Names/Nos]  
  "edu" domain is controlled by Network Solutions.  
  Network Solutions has sub-domains kolkatta, chennai and mumbai.  
  
  Network Solutions can handle all the data there, but why should it ?  
  
  It therefore delegates the subdomain "mumbai.edu' to the folks at Mumbai  
  to manage.  
  
  The folks at mumbai have total authority over this domain now and can  
  create more domains [subdomains et al] and they do.  
  
  They create "kalina.mumbai.edu" and "fort.mumbai.edu" and let the folks  
  at Kalina and Fort handle those domains.  
  
  At Kalina Office : shiva.kalina.mumbai.edu  
                     ganesh.kalina.mumbai.edu  
  
                     Some two hosts at the Kalina office.  
  
  At Fort Office :   shiva.fort.mumbai.edu  
                     ganesh.fort.mumbai.edu  
  
                     Some two hosts at the Fort office.  
  
   Now I do "ping shiva.fort.mumbai.edu".  
  
   The query would go to "." which would direct me to a server which handles  
   the "edu" domain. The "edu" domain knows about the "mumbai" subdomain, so  
   it directs me another DNS server which knows all about the "mumbai.edu"  
   domain. Once there, this server again directs me to another DNS server  
   which know all about "fort.mumbai.edu" domains.  
  
   At the DNS server which handles the "fort.mumbai.edu" domain, I finally  
   find a RR [Resource Record] which is something like this :  
  
   shiva.fort.mumbai.edu.    IN    A    192.168.0.10  
  
   And at last, I have the DNS-resolved IP of my shiva.fort.mumbai.edu !  
  
   The programs which store info about the domain name space are called  
   "name servers" and, yes, one of these prgs was running on this last  
   machine which solved by problem.  
  
   Let's do some light stuff now !  
  
[ii] gTLDS : Global Top-Level Domains  
---------------------------------------  
  
    \*  com, .edu, .gov, .int, .mil, .net, .org  [in US]  
  
DNS Database  :  
-------------                ICANN -   [Int'l Corp for Assigned Names/Nos]  
                                       Responsible for managing domain name  
                                       space  
  
                                "." [root node]   Managed by ICANN  
                                 |  
     -------------------------------------------------------------------  
     |           |            |         |           |          |       |  
    org         int          gov       edu         mil        net     com  
---------    [Int' orgs]                       
|       |  
fish   mammals  
|  |     |  
        ---------------  
        |       |     |  
      dolphin whale  bat  
        |       |     |  
      willy  hector  blindy        
  
      Hence, willy.dolphin.mammals.org is a host on the DNS DB system  
  
  
Consider the similarity to the Unix FS :  
  
                                 /  
                                 |  
     -------------------------------------------------------------------  
     |           |            |         |           |          |       |  
    etc         usr          lib       int         mil        net     org  
 --------    
|       |  
       httpd  
        |  
        conf  
        |  
      --------  
     |       |  
httpd.conf  src.conf  
  
    Geopolitical domains : ccTLDs or ISO country codes - eg .uk, .de, .in  
  
    2nd Level registration under ccTLDS : eg amazon.de, amazon.co.uk  
  
    3nd level domains as states of the US : .ca.us, .ny.us  
  
    Eg: 'mammals' is the 2nd level domain purchased and is registered under  
         the gTLD - 'net'  
  
[iii] Reverse Domains :  
-----------------------  
  
\* Addr to Name conversion  
\* Reverse lookups are reqd to create more readable o/p displays  / logs etc  
\* They are called RDs, since they are written in the reverse order  
  
Consider :    /etc/hosts  
  
              192.168.0.10  crow.birds.org   
        -----------------------------------------------  
        Consider the hostname :                   --------------- [A]  
         
                crow      .birds.    org  
                 
         <-------- ----------="" host="" specific=""> generic [.]  
                 host      domain    gTLD  
         
        -----------------------------------------------  
         
        Now consider the IP :                       --------------- [B]  
                
                  192.168.0.       10  
               <--nw portion--="">   
  
                
                 generic ---------> specific  
  
        -----------------------------------------------  
  
   Bcos the structure of an IP addr [B] is the opposite of the domainname [A],  
   to create a 'Reverse Domain Name' we reverse the IP address  
  
                   10.0.168.192.in-addr.arpa.  
  
     'in-addr.arpa.' is a special TLD domain in which all reverse domains are  
     located.  
                
                                 . [Reverse Domain root DNS servers]  
                                 |  
                                arpa  
                                 |  
                               in-addr  
       -------------------------------------------------------------------  
       |           |            |         |           |          |       |  
     192.168  
       |  
     -------  
     |->|->|  
     0  1 255  
     |  
     -------  
     |  |  |       
     1->40 254  
#################################################################################DNS summary  
  
                  Setting up a DNS SERVER/CLIENT using BIND9  
                       Berkeley InterNet Name Domain  
  
        =================  
                        DNS ON THE CLIENT   
        =================  
  
â€¢ Configuring Local Client resolvers - /etc/hosts and /etc/resolv.conf  
â€¢ The hosts.txt file and /etc/hosts local DNS resolver on the Client  
â€¢ Internet DNS Hierarchy  
â€¢ Client resolver -  
             /etc/resolv.conf [nameserver, domain, search, options, sortlist]  
  
              domain bom.labs.net  
              nameserver 192.168.0.20  
              options timeout:1  
              options attempts:1  
       
          The Host Table â€“ /etc/host.conf,/etc/nsswitch.conf and DNS precedence  
  
         =================  
                             DNS ON THE SERVER  
         =================  
  
â€¢ Types of DNS Servers  
      A Trivial Caching-only Nameserver  
      Master [Primary] DNS Server  
      Slave [Secondary] DNS Server  
  
â€¢ Installing / Checking the DNS Server Software â€“ BIND9  
  
â€¢ DNS Server Configuration Files after installing BIND9  
  
           /etc/named.conf  - DNS Boot File or Config File  
  
           /var/named/named.ca          Root DNS Hints File  
           /var/named/localhost.zone    Local LoopBack Fwd Zone DB/File  
           /var/named/named.local       Local LoopBack Rev Zone DB/File  
           We never touch these 3 files!  
  
           We have to create the Fwd/Rev zone files for our domain- bom.labs.net  
  
           /var/named/named.bom.labs.net   My domain's Fwd Zone DB/File  
           /var/named/named.192.168.0      My domain's Rev Zone DB/File  
  
    The named daemon  - service named restart  
  
â€¢ Overview of the Files required for DNS  
  
â€¢ The in-addr.arpa domain  
  
  
\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
         Creating a Authoritative Master/Primary DNS Server  
\* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
Format :  
   1          2       3                4                          5  
name/owner [TTL...] Class            record-type                 data  
         @          IN/CH/HS -         SOA   
                    Internet        NS,A,PTR,MX,CNAME,RP,  
                    ChaosNet         TXT,HINFO,NULL...  
                    Hesiod  
  
 []  []  SOA      (  
 of zone                     
      
      
      
    )  
  
                  DNS Record-Types  
   ================  
                      1. SOA    - Start of Authority  
       2. NS     - Name Server  
       3. A      - Internet Address  
       4. PTR    - Pointer  
       5. MX     - Mail Exchanger  
       6. CNAME  - Canonical Name [nickname pointer]  
       7. RP     - Responsible Person  
       8. TXT    -  
       9. HINFO  - Host Information - OS/CPU etc  
      10. NULL  
      11. RDATA  
  
 SOA Rec :  7 components  [Start of Authority]  
 =============================================  
1. Primary or Master Nameserver  
2. DNS Admin's Mail-Id  
   The next 5 are the characterisics :  
    3. Serial Number  
    4. Refresh rate  
    5. Retry  
    6  Expiry  
    7. Default TTL for the domain  
        
  
         name/owner, Zone File directives :  
             [TTL, $ORIGIN,  $INCLUDE, $GENERATE],  
                   class [INternet, CHaosnet, HeSiod],  
  
    Testing :  
  
    named-checkconf : checks the syntax, but not the semantics, of a named  
                      configuration file.  
       If not  specified, it defaults to /etc/named.conf.  
       Returns an exit status of 1 if errors were detected and  
       0 otherwise.  
       Check with echo $?  
  
    named-checkzone : checks the syntax and integrity of a zone file.  
                      It performs the same checks as named does when loading  
       a zone. This makes named-checkzone useful for checking  
       zone files before configuring them into a name server.  
   
                      configuration file.  
       If not  specified, it defaults to /etc/named.conf.  
  
 eg   named-checkzone  bom.labs.net /var/named/named.bom.labs.net  
 eg   named-checkzone  0.168.192.in-addr.arpa /var/named/named.192.168.0  
  
                               \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*