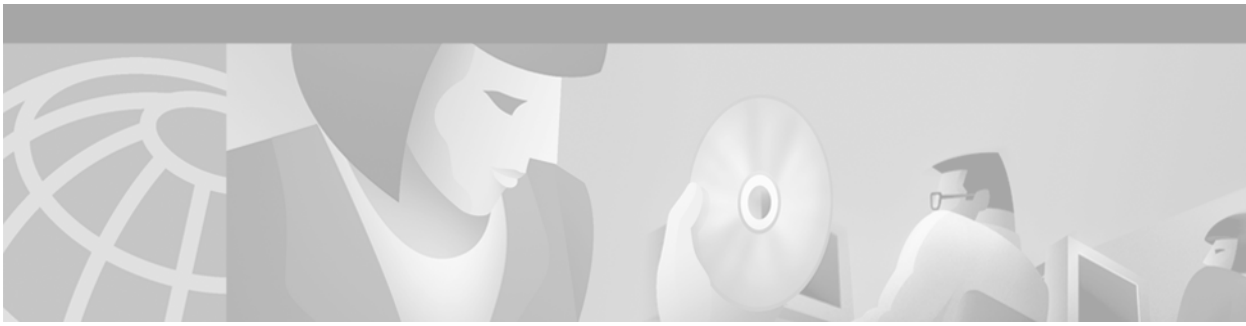


Task 6—Managing IP Addresses by Using DNS

About Managing IP Addresses

Managing IP addresses is a primary network administration function. Assigning and removing IP addresses can be tedious and error prone. Regardless—you must manage IP addresses to avoid duplicate IP subnets and addresses.

Domain Name System (DNS) servers provide two kinds of fundamental lookup services:

- **Forward lookups**—Used for looking up the IP address of a provided device name. This is the most common kind of lookup performed.
- **Reverse lookups**—Used for looking up a device name of a provided IP address. Administratively, reverse-lookup zones are important tools used for tracking IP address assignments.

In this case study, the dial engineers at THEnet:

- Have received a block of IP addresses from the NOC with DNS administrative rights and instructions for setting up IP address space.
- Track IP address assignments by using DNS reverse lookup zones within the existing DNS service.
- Use the application Cisco Network Registrar (CNR) and its CLI to manage the IP address database. CNR is a full-featured IP address management solution for both enterprise and service provider networks. It includes advanced DNS and Dynamic Host Configuration Protocol (DHCP) servers.

Note This section assumes you are familiar with the basics of DNS. For more information about DNS, see *DNS and Bind*, Third Edition, by Paul Albitz and Cricket Liu. The ISBN number is 1565925122.

Table 30 Related References and Documents

Reference	URL
Internet Software Consortium for BIND (Berkeley Internet Name Daemon)—Describes the DNS protocols.	http://www.isc.org/products/BIND/
Cisco Network Registrar—A collection of DNS/DHCP user guides and reference manuals.	http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/ciscoasu/nr/index.htm

Using Cisco Network Registrar CLI Commands

Database locking prevents multiple users from writing to the same database records concurrently. However, an administrator may occasionally not exit a session properly, and the database may be left locked. To release the lock on the database, use the **force-lock** network registrar command.

Network registrar commands sent from the Unix shell lock the database only while commands are running.

The name for a reverse zone is the inverse of your Internet network number, added to the special domain in-addr.arpa. For example if the network number is 1.2.3.0, the reverse zone name is 3.2.1.in-addr.arpa. A second example is the network number 1.2.0.0 with the reverse zone of 2.1.in-addr.arpa.

For a description of the network registrar CLI commands, go to http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/ciscoasu/nr/nr30t/cliref/cli01.htm#68483

To quickly perform administrative tasks by using CNR CLI commands, follow these steps:

Step 1 Log in to the Cisco Network Registrar application by entering the following directory path:

```
/opt/nwreg2/usrbin/nrcmd
nrcmd>
```

After logging in, the command mode is accessed and the prompt “nrcmd>” appears.

Step 2 To create an account for an administrator, enter the **admin** command and an associated password:

```
nrcmd> admin bob create password=xyz
```

In this example, the administrator name is **bob**. The password is **xyz**.

Step 3 To see a list of existing administrators, enter the **admin list** command:

```
nrcmd> admin list
bob: password=********;
omar: password=********;
padma: password=********;
```

Note The **admin list** command is a read-only command.

Step 4 Inspect a reverse zone by entering the **zone** command and **listRR** option:

```
nrcmd> zone 101.21.172.in-addr.arpa. listRR

100 Ok
Static Resource Records
@                       IN      SOA     onionring.the.net. netadmin.the.net 1997121601
3600 1800 86400 86400
@                       IN      NS      onionring.the.net.com.
205                     IN      PTR     unused-205.the.net.
203                     IN      PTR     unused-203.the.net.
210                     IN      PTR     unused-210.the.net.
204                     IN      PTR     unused-204.the.net.
1                       IN      PTR     unused-1.the.net.
10                      IN      PTR     unused-10.the.net.
101                     IN      PTR     unused-101.the.net.
102                     IN      PTR     unused-102.the.net.
103                     IN      PTR     unused-103.the.net.
104                     IN      PTR     unused-104.the.net.
(truncated for brevity)
```

Step 5 When working with a reverse zone, you can map an IP address to a router by entering the **zone** command and the **addRR** resource record (RR) option:

```
nrcmd> zone 101.21.172.in-addr.arpa. addRR 7 PTR bobslake-nas-01.the.net
```

Step 6 Remove a resource record by entering the **zone** command and **removeRR** option:

```
nrcmd> zone 101.21.172.in-addr.arpa. removeRR 7 PTR unused-07.the.net
```

Step 7 To minimize the lock-time on the database, enter the following CNR command from the Unix command line. Use quotations (“ ”) to contain the command and pass it to the shell.

```
/opt/nwreg2/usrbin/nrcmd "zone 101.21.172.in-addr.arpa. listRR"
```

Note The NRCMD command mode is not used.

Step 8 Sort the records and parse the output by entering the following CNR command from the Unix command line:

```
/opt/nwreg2/usrbin/nrcmd "zone 101.21.172.in-addr.arpa. listRR" | sort -n | more
username: password:
0                       IN      PTR     broadcast-0.the.net.
@                       IN      NS      onionring.the.net.
@                       IN      SOA     onionring.the.net. netadmin.the.net.101.
21.172.in-addr.arpa. 1997121606 3600 1800 86400 86400
Dynamic Resource Records
Static Resource Records
1                       IN      PTR     unused-1.the.net.
2                       IN      PTR     unused-2.the.net.
3                       IN      PTR     unused-3.the.net.
4                       IN      PTR     unused-4.the.net.
5                       IN      PTR     unused-5.the.net.
6                       IN      PTR     unused-6.the.net.
7                       IN      PTR     unused-7.the.net.
8                       IN      PTR     unused-8.the.net.
9                       IN      PTR     unused-9.the.net.
10                      IN      PTR     unused-10.the.net.
(truncated for brevity)
```

Step 9 To add an "A" Resource Record (RR) to a forward zone (domain) and map a name to an IP address, enter the **zone** command:

```
nrcmd> zone the.net. addRR bobslake-nas-02 A 172.21.10.18

@                       IN      NS      onionring.the.net.
@                       IN      SOA     onionring.the.net. netadmin.the.net. 56 10800
 3600 604800 86400
Dynamic Resource Records
Static Resource Records
aurora                  IN      A       172.21.100.100
bobslake-nas-01         IN      A       172.21.10.10
bobslake-nas-02         IN      A       172.21.10.18
doc-2610-01             IN      A       172.21.10.13
doc-3810a-01            IN      A       172.21.10.14
doc-3810d-01            IN      A       172.21.10.15
doc-AS5800-01           IN      A       172.21.10.11
doc-core-01             IN      A       172.21.10.5
doc-core-02             IN      A       172.21.10.6
doc-core-03             IN      A       172.21.10.7
(truncated for brevity)
```

In the previous example, the **zone** command:

- Creates an A record for the.net
- Assigns the IP address 172.21.10.18 to the router bobslake-nas-02

Step 10 To reload the server to make all IP assignments or changes take effect, enter the following command:

```
nrcmd> server dns reload
```

Note Reload all changes into the DNS database, so that the changes can be resolved upon lookup.

Using a Batch File to Make Changes to a DNS Configuration

CNR can use batch files to make large and small-scale changes to the DNS configuration within your network.

To use the batch-file facility to add and remove entries, follow these steps:

Step 1 Define the batch file by entering **zone** commands:

```
zone the.net. addRR doc-core-02 A 172.21.10.6
zone the.net. addRR doc-core-03 A 172.21.10.7
zone 10.21.172.in-addr.arpa. removeRR 6 PTR unused-6.the.net.
zone 10.21.172.in-addr.arpa. removeRR 7 PTR unused-7.the.net.
zone 10.21.172.in-addr.arpa. addRR 6 PTR doc-core-02.the.net.
zone 10.21.172.in-addr.arpa. addRR 7 PTR doc-core-03.the.net.
server dns reload
```

The previous batch-file example shows how to add two new device/IP addresses. In addition to adding two "A" records (lines 1 and 2), remove the "unused" PTR records from the reverse zone (lines 3 and 4) before adding the new "PTR" records, in place of the unused records, to the reverse zone (lines 5 and 6). See line 7 to reload the DNS server.

Step 2 Run the script by using the **-b** option:

```
nrcmd> -b < 172.21.10.batch
```

The following output appears:

```
nrcmd>
zone the.net. addRR doc-core-02 A 172.21.10.6
100 Ok
doc-core-02             IN      A       172.21.10.6

nrcmd>
zone the.net. addRR doc-core-03 A 172.21.10.7
100 Ok
doc-core-03             IN      A       172.21.10.7

nrcmd>
zone 10.21.172.in-addr.arpa. removeRR 6 PTR unused-6.the.net.
100 Ok
removing 6                      IN      PTR     unused-6.the.net.

nrcmd>
zone 10.21.172.in-addr.arpa. removeRR 7 PTR unused-7.the.net.
100 Ok
removing 7                      IN      PTR     unused-7.the.net.

nrcmd>
zone 10.21.172.in-addr.arpa. addRR 6 PTR doc-core-02.the.net.
100 Ok
6                       IN      PTR     doc-core-02.the.net.

nrcmd>
zone 10.21.172.in-addr.arpa. addRR 7 PTR doc-core-03.the.net.
100 Ok
7                       IN      PTR     doc-core-03.the.net.

nrcmd>
server dns reload
100 Ok
```

Creating a Primary Forward Zone

To create a domain (or forward zone) and include all forward mapping (the "A" records) for the domain, follow these steps:

Step 1 Create a domain and include all forward mapping (the "A" records) by entering the **zone** command with the **create** option:

```
nrcmd> zone the.net create primary file=the.net.zone.txt
```

To create new subnets by using the CLI, import a BIND zone definition file, which can be edited by using an ASCI text editor. The following example shows an edited BIND file.

```
@                       IN      SOA     onionring.the.net. netadmin.the.net. (
                                2000071600      ; serial number
                                3600            ; Refresh 1 hours
                                1800            ; Retry 30 minutes
                                86400           ; Expire 24 hours
                                86400           ; TTL 24 hours
                                )
                IN      NS      onionring.the.net.
doc-rtr58-01            IN      A       172.21.101.20
doc-rtr54-01            IN      A       172.21.101.21
doc-rtr53-01            IN      A       172.21.101.22
doc-rtr53-05            IN      A       172.21.101.23
doc-3810a-01            IN      A       172.21.10.14
doc-3810d-01            IN      A       172.21.10.15
doc-ubr7246-01          IN      A       172.21.10.16
doc-switch-02           IN      A       172.21.10.17
```

Step 2 Verify that the primary zone was created by entering the **zone** command with the **listRR** option:

```
nrcmd> zone the.net listRR
100 Ok
Static Resource Records
@                       IN      SOA     onionring.the.net.
netadmin.the.net.0
@                       IN      NS      onionring.the.net.
doc-rtr58-01            IN      A       172.21.101.20
doc-rtr54-01            IN      A       172.21.101.21
doc-rtr53-01            IN      A       172.21.101.22
doc-rtr53-05            IN      A       172.21.101.23
(Truncated for brevity)
Dynamic Resource Records
```

Creating an IP Tracker Web Page

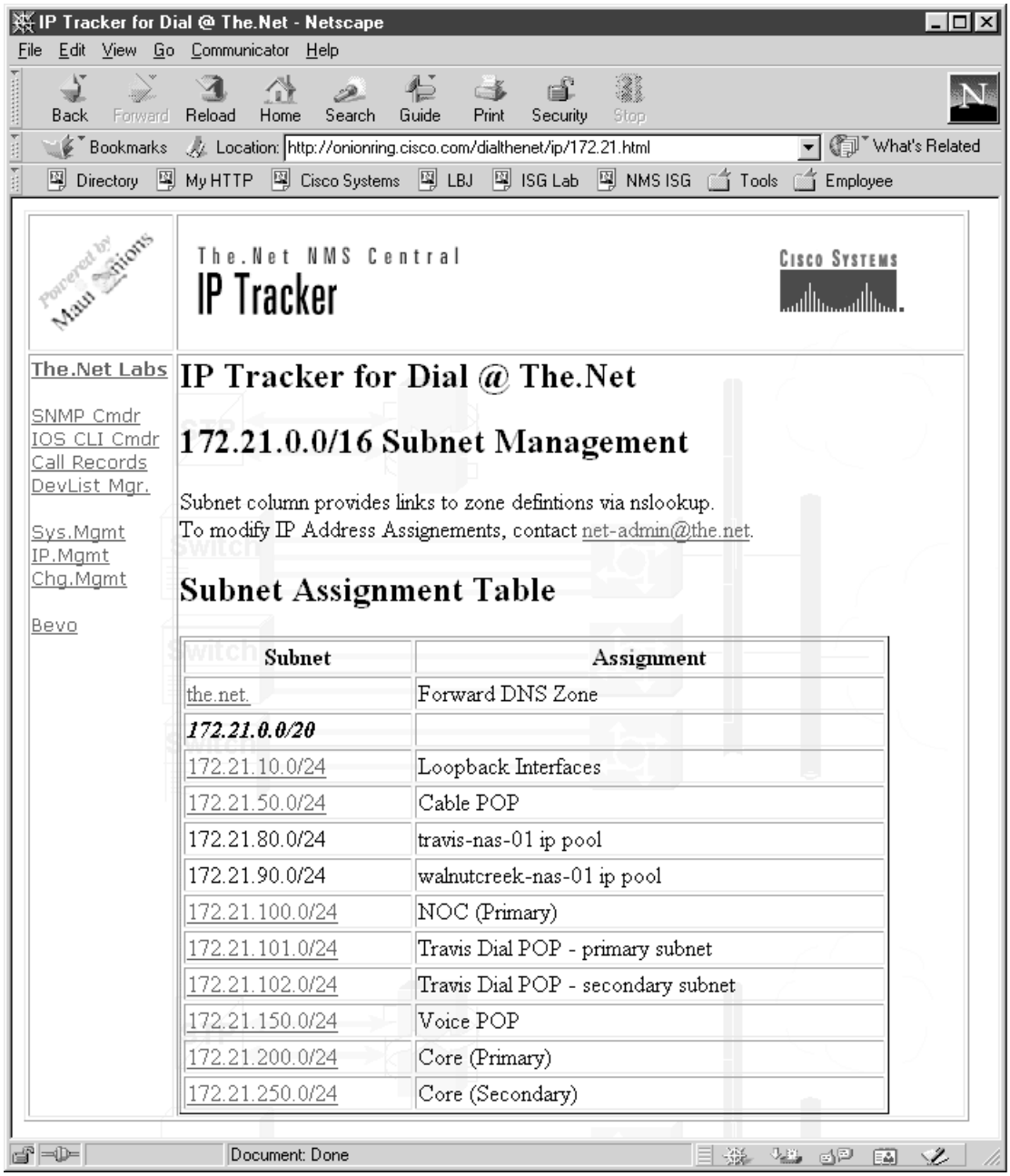
An IP tracker web page:

- Provides web access to the IP database that is managed by Cisco Network Registrar.
- Retrieves current IP address block assignments from a DNS server.
- Uses two CGI scripts to provide a web-enabled look into DNS for each zone.

To create an IP tracker web page, follow these steps:

Step 1 Become familiar with the layout of an IP tracker web page. In Figure 24, the subnet column shows a list of all managed zones. The assignment column describes the purpose of each zone.

Figure 24 IP Tracker Web Page



Step 2 Understand how the CGI scripts function.

There are two scripts that work together to return an NSLOOKUP list query (ls) for a specified zone in a CGI link.

- *dnszone.pl*—Runs the CGI process. In the subnet column in Figure 24, the entry 172.21.10.0/24 is an active link that calls the dnszone.pl script.

  The active link is coded as:

  <td>172.21.10.0/24</td>

  Once invoked, dnszone.pl calls the second script, *dnszone_dump*.

- *dnszone_dump*—An expect script that steps through the NSLOOKUP interactive mode and returns the output of a "ls [ZONE]" command to the dnszone.pl script. The zone list, returned to the requesting web-based management browser, appears:

```
ls 10.21.172.in-addr.arpa.

[www.the.net]
 0                              host = broadcast-0.the.net
 1                              host = unused-1.the.net
 2                              host = unused-2.the.net
 3                              host = unused-3.the.net
 4                              host = unused-4.the.net
 5                              host = doc-core-01.the.net
 6                              host = doc-core-02.the.net
 7                              host = doc-core-03.the.net
 8                              host = doc-ls1010-01.the.net
 9                              host = doc-switch-01.the.net
 10                             host = doc-pix-01.the.net
 10.21.172.in-addr.arpa.        server = onionring.the.net
 11                             host = doc-AS5800-01.the.net
 12                             host = doc-oob-03.the.net
 13                             host = doc-2610-01.the.net
 14                             host = doc-3810a-01.the.net
 15                             host = doc-3810d-01.the.net
 16                             host = doc-ubr7246-01.the.net
 17                             host = doc-switch-02.the.net
```

Download the source code for the scripts and customize them for your environment. The following is an example of a return using http, which shows the list of hosts and their address numbers for a specified reverse DNS zone:

```
##########################       DNSZONE.PL      ##########################

#!/usr/local/bin/perl -w
use CGI_Lite;
#---------------------------------------------------------------------
# Script Name:          dnszone.pl
# Version:              1.0
# Last modified by:     xxx July 11, 2000
# Requirements:         dnszone_dump (Expect script)
#                       CGI_Lite.pm
# Description:          Returns via http the list of hosts and their address numbers
#                       of a specified reverse DNS zone.
# Created by:           xxx
# Date:                 May 15, 2000
# Contact:              coe-iae@cisco.com
#---------------------------------------------------------------------

# POST method to extract parameter strings.
# Single value extracted
# Create an instance of CGI_Lite
my $query = new CGI_Lite;
# Send an appropriate MIME header to the browser
print "Content-type: text/html\n\n";
# Send the beginning HTML
print "<HTML><Head><Title>Hello!</Title>\n";
print "</Head></Body>\n";
# Get the form data into a hash
my %FORM = $query->parse_form_data;
# Get the user's name from the hash
my $zone = $FORM{"zone"};
# Now that we have retrieved the zone, we get the dump from expect
$data=`/opt/CSCOlwt/bin/dnszone_dump $zone`;
# we print it out
print "<pre>";
print "$data";
print "</pre>";
print "</Body></HTML>\n";


##########################       DNSZONE_DUMP    ##########################

#!/usr/local/bin/expect --
#---------------------------------------------------------------------
# Script Name:          dnszone_dump
# Version:              1.0
# Last modified by:     xxx July 11, 2000
# Requirements:         dnszone.pl (calls this script)
# Description:          Uses nslookup to produce a zone list of the
#                       zone name passed to it by dnszone.pl
# Created by:           xxx
# Date:                 May 15, 2000
# Contact:              coe-iae@cisco.com
#---------------------------------------------------------------------

if {$argc != 1} {
        puts "USAGE: zonedump <reversezone>"
        puts "e.g. zonedump 61.32.172.in-addr.arpa."
        exit
}
```

```
set zone [lindex $argv 0]


log_user 0
spawn /usr/sbin/nslookup

expect ">"
sleep 1
log_user 1
send "ls $zone\r\n"
expect ">"
log_user 0
sleep 1
send "exit\r\n"
exit
```

How to Create a Reverse DNS Zone

By creating reverse lookup zones for each IP subnet, you gain a robust database that can be used to track assignments within an IP address space. Reverse lookups can determine the allocation status of any address from any DNS client.

Network operators must account for used and unused IP addresses. It is recommended that each IP address be given a DNS PTR Resource Record, even if the address is unused. For example, you can look up and resolve an IP address as "unused-XXX.the.net."

See the following example to create a zone from a BIND file by entering the **zone** command:

```
nrcmd> zone 101.21.172.in-addr.arpa. create primary file=the.net_rev_zone.txt
```

The following edited BIND definition file is for "the.net_rev_zone.txt."

```
@                       IN      SOA     onionring.the.net
esupport-austin.the.net. (
                                2000071600      ; serial number
                                3600            ; Refresh 1 hours
                                1800            ; Retry 30 minutes
                                86400           ; Expire 24 hours
                                86400           ; TTL 24 hours
                                )
;
                        IN      NS      onionring.the.net.
;
0                       IN      PTR     broadcast-0.the.net.
1                       IN      PTR     unused-1.the.net.
2                       IN      PTR     unused-2.the.net.
3                       IN      PTR     unused-3.the.net.
4                       IN      PTR     unused-4.the.net.
5                       IN      PTR     unused-5.the.net.
6                       IN      PTR     unused-6.the.net.
7                       IN      PTR     unused-7.the.net.
8                       IN      PTR     unused-8.the.net.
9                       IN      PTR     unused-9.the.net.
10                      IN      PTR     unused-10.the.net.
11                      IN      PTR     unused-11.the.net.
12                      IN      PTR     unused-12.the.net.
13                      IN      PTR     unused-13.the.net.
14                      IN      PTR     unused-14.the.net.
15                      IN      PTR     unused-15.the.net.
16                      IN      PTR     unused-16.the.net.
17                      IN      PTR     unused-17.the.net.
```

```
18                      IN      PTR     unused-18.the.net.
19                      IN      PTR     unused-19.the.net.
20                      IN      PTR     doc-rtr58-01.the.net.
21                      IN      PTR     doc-rtr54-01.the.net.
22                      IN      PTR     doc-rtr53-01.the.net.
23                      IN      PTR     doc-rtr53-01.the.net.

(Truncated for brevity..)

253                     IN      PTR     unused-253.the.net.
254                     IN      PTR     unused-254.the.net.
255                     IN      PTR     broadcast-255.the.net.
```

The following are sample BIND files that can be used as a template and edited for your environment.

Note This field is 'informational' and can be queried using NSLOOKUP. It should contain the e-mail address of the relevant DNS administrator in dotted notation (the @ replaced with a dot). Many customers use 'hostmaster.FQDN' and then set an e-mail alias or forward to point to the relevant administrator.

```
city.business.slb.com.                      IN          SOA
hostmaster.city.business.slb.com. 1997121600 serial number 3600
Refresh 1 hours 1800
Retry 30 minutes 86400
Expire 24 hours 86400
TTL 24 hours
host.city.business.slb.com.                 IN          NS
```

Note In the reverse table, you should have one entry for each host address in the class C network. This table is the authoritative source of information on hostnames and can then be used as an address allocation table. Just change 'unused-X' to 'hostname' when you

assign an address and reverse the process to de-assign an address. This way, if someone connects a computer to your network you will see 'unused-x' in the system logs and will know which IP address is in use without being allocated.

```
0.city.business.slb.com. 1          IN          PTR broadcast-
1.city.business.slb.com. 2          IN          PTR unused-
2.city.business.slb.com. 3          IN          PTR unused-
1.city.business.slb.com. 2          IN          PTR unused-
2.city.business.slb.com. 3          IN          PTR unused-
3.city.business.slb.com. 4          IN          PTR unused-
4.city.business.slb.com. 5          IN          PTR unused-
5.city.business.slb.com. 6          IN          PTR unused-
6.city.business.slb.com. 7          IN          PTR unused-
7.city.business.slb.com. 8          IN          PTR unused-
8.city.business.slb.com. 9          IN          PTR unused-
9.city.business.slb.com. 10         IN          PTR unused-
10.city.business.slb.com. 11        IN          PTR unused-
11.city.business.slb.com. 12        IN          PTR unused-
12.city.business.slb.com. 13        IN          PTR unused-
13.city.business.slb.com. 14        IN          PTR unused-
14.city.business.slb.com. 15        IN          PTR unused-
15.city.business.slb.com. 16        IN          PTR unused-
16.city.business.slb.com. 17        IN          PTR unused-
17.city.business.slb.com. 18        IN          PTR unused-
18.city.business.slb.com. 19        IN          PTR unused-
19.city.business.slb.com. 20        IN          PTR unused-
20.city.business.slb.com. 21        IN          PTR unused-
21.city.business.slb.com. 22        IN          PTR unused-
22.city.business.slb.com. 23        IN          PTR unused-
23.city.business.slb.com. 24        IN          PTR unused-
24.city.business.slb.com. 25        IN          PTR unused-
25.city.business.slb.com. 26        IN          PTR unused-
26.city.business.slb.com. 27        IN          PTR unused-
27.city.business.slb.com. 28        IN          PTR unused-
28.city.business.slb.com. 29        IN          PTR unused-
29.city.business.slb.com. 30        IN          PTR unused-
30.city.business.slb.com. 31        IN          PTR unused-
31.city.business.slb.com. 32        IN          PTR unused-
32.city.business.slb.com. 33        IN          PTR unused-
33.city.business.slb.com. 34        IN          PTR unused-
34.city.business.slb.com. 35        IN          PTR unused-
35.city.business.slb.com. 36        IN          PTR unused-
36.city.business.slb.com. 37        IN          PTR unused-
37.city.business.slb.com. 38        IN          PTR unused-
38.city.business.slb.com. 39        IN          PTR unused-
39.city.business.slb.com. 40        IN          PTR unused-
40.city.business.slb.com. 41        IN          PTR unused-
41.city.business.slb.com. 42        IN          PTR unused-

(Truncated for brevity..)

234.city.business.slb.com. 235       IN          PTR unused-
235.city.business.slb.com. 236       IN          PTR unused-
236.city.business.slb.com. 237       IN          PTR unused-
253.city.business.slb.com. 254       IN          PTR unused-
254.city.business.slb.com. 255       IN          PTR unused-
255.city.business.slb.com. 255       IN          PTR broadcast-
```