CHAPTER 5

Configuring Network Proximity

Network Proximity provides a content delivery solution that significantly improves network performance. This optional software feature uses a CSS configured as a database that is populated by actively probing the network to determine the proximity of clients and services. Additional CSSs (any model) perform database lookup requests and domain name resolution to determine the most proximate service for a client.

Note The Network Proximity feature requires the CSS Enhanced feature set license.

This chapter describes the Network Proximity feature and provides related configuration information in the following major sections:

Caution If you configure your CSS as a Proximity Database, you cannot use the CSS for load balancing.

Entering Your Proximity License Keys

Before you can configure Network Proximity on CSSs, you must purchase:

- An Enhanced feature set for a Proximity Domain Name Server (PDNS)
- The Proximity Database (PDB) option

If you purchased the Enhanced feature set or the Proximity Database option:

- During the initial CSS order placement, a Claim Certificate is included in the accessory kit.
- After receiving the CSS, Cisco Systems sends the Claim Certificate to you by mail.

Note If you cannot locate the Enhanced feature set Claim Certificate or the Proximity Database Claim Certificate in the accessory kit, call the Cisco Technical Assistance Center (TAC) toll free, 24 hours a day, 7 days a week at 1-800-553-2447 or 1-408-526-7209. You can also e-mail the TAC at tac@cisco.com.

Follow the instructions on the Claim Certificate to obtain the software license key for each feature.

Entering the Enhanced Feature Set License Key

Enter the license key for the Enhanced feature set, which includes the Proximity Domain Name Server (PDNS) feature, on each CSS (any model) that you want to use exclusively as a PDNS. To install the Enhanced feature set license key:

1. Log in to the CSS and enter the **license** command.

   ```
   # license
   ```

2. Enter the 12-digit Enhanced feature set software license key. For example:

   ```
   Enter the Software License Key (q to quit): nnnnnnnnnnnn
   ```

The Enhanced feature set license key is now properly installed and the feature set is activated.

Entering the Proximity Database License Key

Caution If you configure your CSS as a Proximity Database, you cannot use the CSS for load balancing.

Enter the license key for the Proximity Database (PDB) software option on each CSS 11150 with 256 MB of memory that you want to use exclusively as a PDB.

To install the PDB software license key:

1. Log in to the CSS and enter the **license** command.

   ```
   # license
   ```

2. Enter the 12-digit Proximity Database license key.

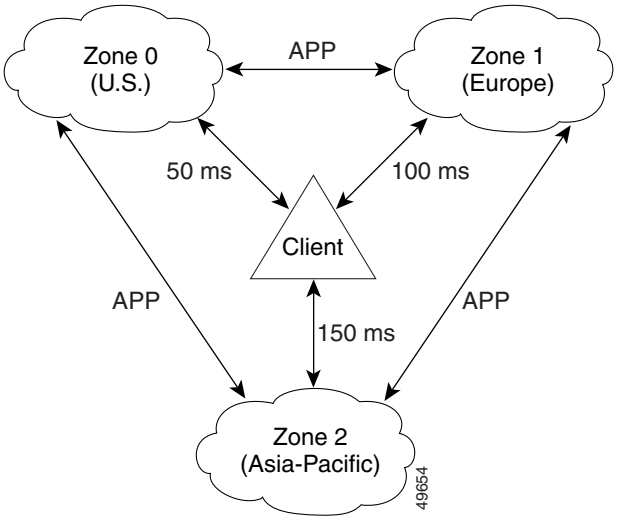
   ```
   Enter the Software License Key (q to quit): nnnnnnnnnnnn
   ```

3. Reboot the CSS for the software license key to activate the PDB feature.

Overview of Network Proximity

Proximity represents a topological relationship between a client and content services. In a network topology perspective, as used in this chapter, proximity refers to connecting a client to the most proximate service based on a measurement of the round-trip time (RTT) between the client's local DNS server and a proximity zone (see Figure 5-1).

Figure 5-1 Simplified Example of Network Proximity



In Figure 5-1, the lowest RTT value is returned from Zone 0. Therefore, Network Proximity would link the client to a service located in Zone 0, regardless of the physical location of the client. The three zones communicate with each other using the Application Peering Protocol (APP). For details on APP, see the "Configuring the Application Peering Protocol" section in Chapter 1, Configuring the CSS as a Domain Name System Server.

The major components and concepts in Network Proximity are:

- Proximity Database
- Proximity Domain Name Server
- Proximity Zones
- Peer Mesh

Proximity Database

A Proximity Database (PDB) is a dedicated CSS 11150 with 256 MB of memory *and* is configured as a PDB using the optional Proximity Database software feature. (For details on configuring a CSS 11150 as a PDB, see "Configuring a Proximity Database" later in this chapter.) One PDB and one or more Proximity Domain Name Servers (PDNSs) and data centers (server farms or lower-level DNS servers) define a subset of the Internet address space called a *proximity zone*. (For details on proximity zones, see "Proximity Zones" later in this chapter.)

Note A PDB can service up to four PDNSs generating their maximum request rates per zone. If the PDNSs are not fully loaded, you can configure additional PDNSs per zone.

Network Proximity, as implemented on a CSS, uses a topology-testing technique that actively probes clients to determine the relative location of clients and services. To accomplish this, a PDB uses ICMP and TCP requests to actively probe a client's local DNS server for proximity information. The PDB analyzes the probe responses, then stores the resulting network RTT metrics (in milliseconds) in its database.

When a PDNS sends the PDB a proximity lookup request for a client using APP-UDP, the PDB compares the RTT metrics for that client and responds immediately with an ordered *zone index*, a list of proximity zones in preferred order by RTT. The PDNS then uses the ordered zone index, along with domain name records and keepalive information, to determine the most proximate service for the client.

Note Probing conducted by the PDB is asynchronous with lookups conducted by the PDNS. Therefore, a PDB will never block a lookup request from a PDNS.

A PDB communicates with PDBs in other zones using a *peer mesh*, implemented with APP. (For details on APP, see the "Configuring the Application Peering Protocol" section in Chapter 1, Configuring the CSS as a Domain Name System Server.) This enables PDBs to periodically "learn" the latest RTT metrics information from the other PDBs in the peer mesh to ensure that a client is connected to the most proximate service. Each PDB contains a metric for every source block of interest in each proximity zone. A source block of interest is a CIDR block that contains a client's local DNS server.

Proximity Domain Name Server

A Proximity Domain Name Server (PDNS) is any CSS that is running the Enhanced feature set *and* is configured as a PDNS using the Enhanced software feature set. (For details on configuring a CSS as a PDNS, see "Configuring a PDNS" later in this chapter.) A PDNS performs PDB lookup requests, using Application Peering Protocol-User Datagram Protocol (APP-UDP), in response to DNS requests that the PDNS receives from a client's local DNS server. The PDB responds to these lookup requests immediately with the ordered zone index. The PDNS uses the ordered zone index along with domain name records and keepalive information to make an authoritative DNS response to the client's local DNS server.

The primary task of a PDNS is to respond to DNS requests based on proximity and domain availability. However, the CSS is not excluded from supporting local content rules and services, as well as non-Proximity-based DNS load balancing. These non-PDNS activities will affect the CSS's performance as a PDNS and the PDNS activities will affect the CSS's performance as a content services switch, depending on the PDNS's load.

Every proximity zone contains one or more PDNSs, up to a maximum of four generating their maximum request rates per zone. If the PDNSs are not fully loaded, you can configure additional PDNSs in a zone. Each PDNS within a zone acts as an authoritative DNS server for domains representing data centers. A data center can be a server farm attached directly to a CSS or can be a lower-level DNS server (which may or may not be a CSS) representing a server farm. You configure the domains statically on each PDNS.

Each PDNS maintains the following records for the domains configured on it:

- **Address record (A-record)** - Any domain that represents a data center, that is not front-ended by another DNS server, and that can be translated to an IP address.
- **Name server record (NS-record)** - Any domain that is front-ended by a lower-level DNS server (not necessarily a CSS).

A PDNS updates its domain records continually through keepalive messages (using ICMP or APP-UDP) that it sends to its locally configured virtual IP addresses (VIPs) and data centers. The PDNS uses the keepalive responses to track the load (kal-ap keepalive only, see below) and availability of locally configured domains. Each PDNS in a proximity zone shares its domain information with other PDNSs in each zone using an APP *peer mesh* (see "Peer Mesh" later in this chapter). There is no communication between PDNSs within the same zone, and each PDNS communicates with one PDNS per zone.

For the optional CSS keepalive type (kal-ap), the keepalive client resides on the PDNS, while the keepalive daemon resides on any CSS-based data center that is the configured recipient of A-records or NS-records as configured on the controlling PDNS. The keepalive daemon extracts the load information of the specified domain names and returns them to the PDNS. This load information originates from:

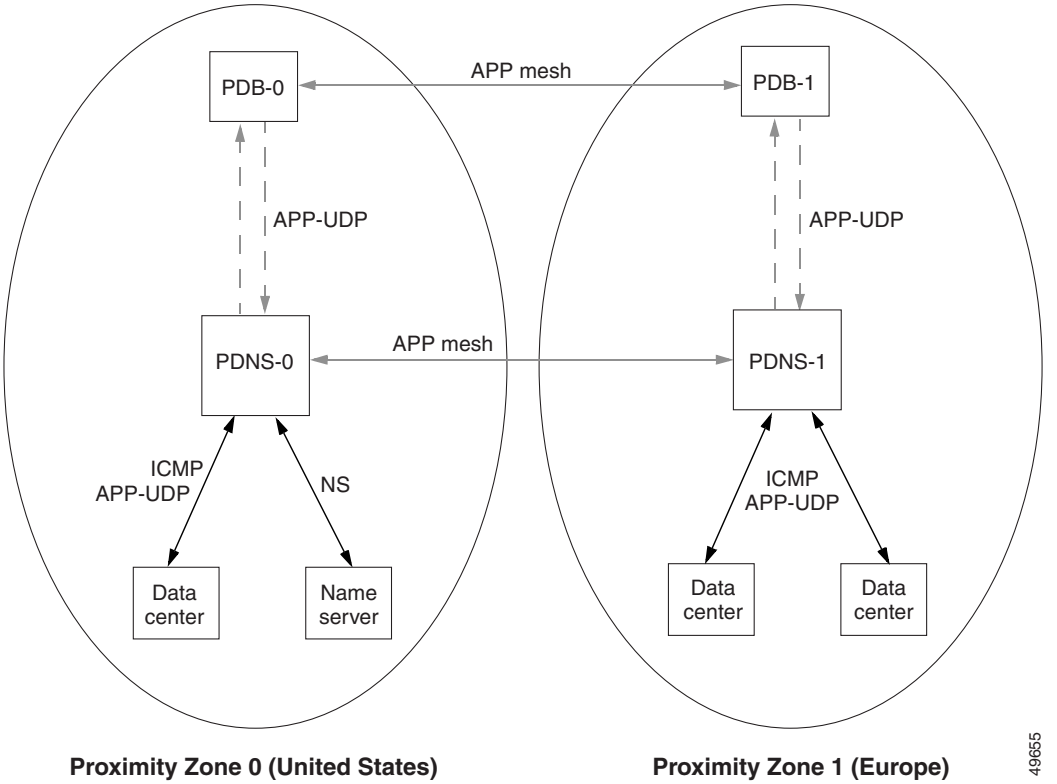
- The average load of the content rule to which the domain is attached
- The load of a locally configured A-record or NS-record

Proximity Zones

A proximity zone is a logical grouping of network devices that consists of one PDB, one or more PDNSs, and services. Although a zone is really a logical subset of the Ipv4 address space, a zone can also be geographically related to a continent, a country, or a major city (Figure 5-2).

For example, you can create proximity zones to group geographically distinct network devices. A proximity zone containing data centers in the United States logically groups nodes within a distinct geographical area. Another proximity zone may logically group nodes and data centers in Europe, for example. Zones are numbered beginning with zero.

Figure 5-2 Example of Network Proximity Zones



Peer Mesh

To communicate proximity information between proximity zones, Network Proximity uses APP to create a *peer mesh*. A peer mesh is an abstraction layer that uses APP to provide common functions (for example, zone configuration information) between Network Proximity devices. A *PDB mesh* allows PDBs to communicate with one another across proximity zones to share proximity metrics. A *PDNS mesh* allows a PDNS in one zone to communicate with one other PDNS in each proximity zone to share domain records and keepalive information. For details on APP, see the "Configuring the Application Peering Protocol" section in Chapter 1, Configuring the CSS as a Domain Name System Server.

Note You can use the concept of zones with a peer mesh to share domain record information between CSSs acting as DNS servers without the use of a PDB. This configuration allows a scalable method of domain name sharing and the use of NS-records in a non-Proximity-based CSS DNS server environment. For more information, see Chapter 1, Configuring the CSS as a Domain Name System Server.

Example of Network Proximity

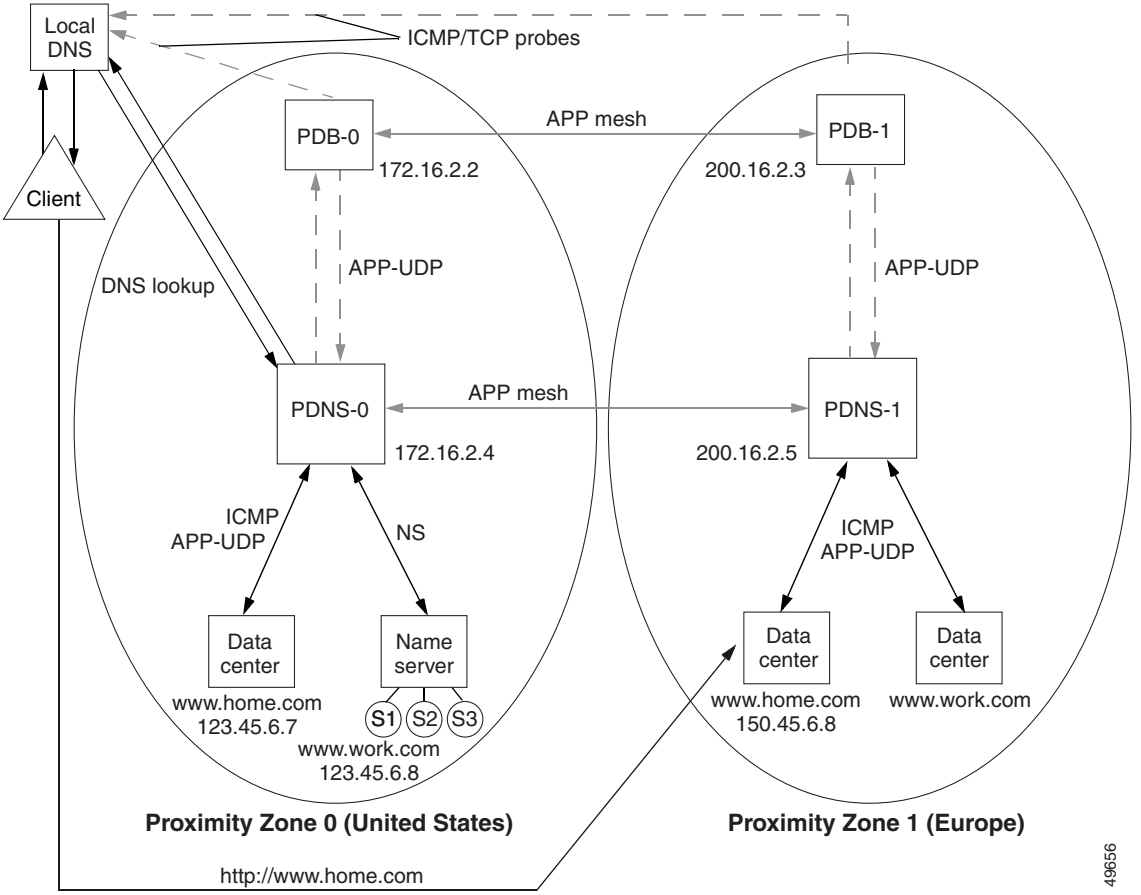
Note A PDB sends ICMP and TCP probes to a client's local DNS server the first time the PDB receives a lookup request for that client from a PDNS. If you configure refinement (see "Refining Proximity Metrics" later in this chapter), a PDB will continue to probe that client periodically. Based on the responses it receives from the probes and the information it receives through its peer mesh, a PDB builds and maintains a database of RTT metrics for clients throughout the network. This process is independent of, and asynchronous with respect to, client requests.

The following example illustrates a single point-in-time request from a client. See Figure 5-3 for an illustration of the following steps.

1. A client performs an HTTP request for the domain name *www.home.com*.
2. The local DNS server performs iterative DNS requests to the root server and to the .com server to resolve the domain name into an IP address or refers the client to an authoritative DNS server. (This step is not shown in Figure 5-3.)
3. The .com server, which is an authoritative DNS server for *home.com*, has the IP addresses of PDNS-0 and PDNS-1 in its configuration. Both PDNSs are authoritative DNS server for *www.work.com*. In this example, the .com server refers the local DNS server to PDNS-0 in Zone 0. (Typically, the .com server uses a roundrobin or other load-balancing method to refer local DNS servers to a PDNS. This step is not shown in Figure 5-3.)

Note Your configuration may include an enterprise DNS server that is positioned between the .com server and the PDNS. The enterprise DNS server would be an authoritative DNS server for *home.com*. The enterprise DNS server contains the IP addresses of the PDNSs and refers the local DNS server to the appropriate PDNS. In either configuration, the PDNS is authoritative for *www.home.com*.

Figure 5-3 Two-Zone Network Proximity Example



4. The local DNS server forwards the client's request for *www.home.com* to PDNS-0 in Zone 0.

5. PDNS-0 determines the most proximate zone to send the client to using one of the following scenarios:

   a. PDNS-0 first searches its cache for a previously saved ordered zone index, a preferred order of zones closest to the client as determined by PDB-0 and based on information from probes and the PDB's peer mesh.

      If PDNS-0 finds the ordered zone index in its cache, it uses that data along with keepalive information and domain records (locally configured and learned through its peer mesh) to determine the most proximate zone to service the client.

   b. If the ordered zone index is not cached, PDNS-0 sends to PDB-0 (using APP-UDP) a lookup request that contains the IP address of the client. PDB-0 calculates the preferred order of zones for the client and returns the ordered zone index to PDNS-0 immediately. PDNS-0 uses the zone order along with keepalive information and domain records to determine the most proximate zone to service the client.

   c. If the ordered zone index is not cached and PDB-0 is not available, PDNS-0 uses its keepalive information, domain records, and a roundrobin method to select a service to handle the request.

6. If the PDNS determines that the best selection is a name server (NS) record, the PDNS begins a recursive query of the name server to determine an authoritative response. If the PDNS finds that the best selection is an address record (A-record), it formulates an authoritative response immediately. In this example, PDNS-0 decides that the best selection is an A-record (learned through the peer mesh with PDNS-1) for a data center in Zone 1.

7. The PDNS sends an authoritative response that contains the resolved IP address of *www.home.com* to the client's local DNS server.

8. The local DNS server notifies the client that sufficient domain name resolution information is available to establish a data connection to *www.home.com*.

9. Lastly, the client uses the local DNS server response information (IP address) to connect to a service in the most proximate zone and starts receiving content. In this example, the most proximate service is located in Proximity Zone 1 at IP address 150.45.6.8.

Note For details on advanced Network Proximity topics, including tiers and nested zones, see "Using Network Proximity Tiers" later in this chapter.

Network Proximity Configuration Quick Start

Table 5-1 and Table 5-2 provide a quick overview of the steps required to configure the PDB and PDNS, respectively. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the command options, see the sections following Table 5-1 and Table 5-2.

PDB Configuration Quick Start

Table 5-1 provides an overview of the steps required to configure a PDB on a dedicated CSS 11150 with 256 MB of RAM. Follow these steps to configure PDB-0 located in Proximity Zone 0 in Figure 5-3. Use the CLI commands outlined in the table to configure basic PDB settings.

Table 5-1 PDB Configuration Quick Start

Task and Command Example
1. Enter config mode by typing **config**. `(config)#`
2. Enable the Application Peering Protocol-User Datagram Protocol (APP-UDP) to allow PDB-0 to communicate with PDNS-0. `(config)# app-udp`
3. Enable the Application Peering Protocol (APP) to allow PDB-0 to communicate with PDB-1. See the "Configuring the Application Peering Protocol" section in Chapter 1, Configuring the CSS as a Domain Name System Server. `(config)# app`

Table 5-1 PDB Configuration Quick Start (continued)

Task and Command Example
4. Configure the **app session** with PDB-1, which is participating in the peer mesh with PDB-0. The IP address you enter is a local interface address on PDB-1. See the "Configuring the Application Peering Protocol" section in Chapter 1, Configuring the CSS as a Domain Name System Server. `(config)# app session 200.16.2.3`
5. Configure PDB-0 in Proximity Zone 0. `(config)# proximity db 0 tier1 "pdb-usa"`

The following running-config example shows the results of entering the commands described in Table 5-1.

```
!************************** GLOBAL ***************************
  app-udp

  proximity db 0 tier1 "pdb-usa"

  app
  app session 200.16.2.3
```

PDNS Configuration Quick Start

Table 5-2 provides an overview of the steps required to configure PDNS-0 located in Proximity Zone 0 in Figure 5-3. Use the CLI commands outlined in the table to configure basic PDNS settings.

Table 5-2 PDNS Configuration Quick Start

Task and Command Example
1. Enter config mode by typing **config**. `(config)#`
2. Enable APP-UDP to allow PDNS-0 to communicate with PDB-0. `(config)# app-udp`

Table 5-2 PDNS Configuration Quick Start (continued)

Task and Command Example
3. Enable APP to allow PDNS-0 to communicate with PDNS-1. See the "Configuring the Application Peering Protocol" section in Chapter 1, Configuring the CSS as a Domain Name System Server. `(config)# app`
4. Configure PDNS-0. Specify the proximity zone and tier number, an optional text description, and the IP address associated with PDB-0. `(config)# dns-server zone 0 tier1 "usa" 172.16.2.2`
5. Configure the CSS to act as a DNS server. `(config)# dns-server`
6. Configure the app session with PDNS-1 that is participating in the mesh with PDNS-0. The IP address you enter is a local interface address on PDNS-1. See the "Configuring the Application Peering Protocol" section in Chapter 1, Configuring the CSS as a Domain Name System Server. `(config)# app session 200.16.2.5`
7. Create A-records for domains in Zone 0. Specify the domain name mapped to the address record and the IP address bound to the domain name. Include an optional time to live (TTL) value, the number of records to return in a DNS response message, and the keepalive message type. `(config)# dns-record a www.home.com 123.45.6.7 0 single kal-icmp`
8. Create NS-records for domains on other DNS servers within the proximity zone. Specify the domain name mapped to a domain IP address. Include an optional TTL value, the number of records to return in a DNS response message, and the keepalive message type. `(config)# dns-record ns www.work.com 123.45.6.8 0 single kal-icmp`
9. Optionally, create content rules for local A-records. In some configurations, there may not be any local content rules or services. For details on creating content rules, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

The following running-config example shows the results of entering the commands described in Table 5-2.

```
!************************** GLOBAL ***************************
  app-udp

  dns-server zone 0 tier1 "usa" 172.16.2.2
  dns-server
  dns-record a www.home.com 123.45.6.7 0 single kal-icmp
  dns-record ns www.work.com 123.45.6.8 0 single kal-icmp

  app
  app session 200.16.2.3
```

Configuring a Proximity Database

Caution If you configure your CSS as a Proximity Database, you cannot use the CSS for load balancing.

A PDB is a dedicated CSS 11150 with 256 MB of RAM and configured as a Proximity Database. Configure one PDB in each Network Proximity zone you want to create. Once configured, a PDB stores network topology information used to determine the relationship between proximity zones and a client that requests a service. The PDB populates its database through active probing of clients (local DNS servers) and sharing information with PDBs in other zones using an APP mesh. The PDB also responds to lookup requests from each PDNS configured in a zone using APP-UDP.

Note You must connect a PDB to a PDNS over a reliable link because of the requirements of the APP-UDP-based proximity lookup mechanism.

Configuring a PDB requires the following two tasks:

- Configuring APP-UDP and APP
- Enabling the PDB

Optionally, you can configure additional PDB parameters as follows:

- Assigning Proximity Metrics
- Flushing Proximity Assignments
- Configuring Proximity Time to Live
- Storing the PDB
- Retrieving the PDB
- Refining Proximity Metrics
- Using Proximity Reprobe
- Clearing the PDB
- Configuring the Proximity Probe Module

Configuring APP-UDP and APP

Network Proximity uses the Application Peering Protocol-User Datagram Protocol (APP-UDP) to exchange proximity information between a PDB and a PDNS, and between a PDNS and services. APP-UDP is a connectionless form of APP.

Note After you configure APP-UDP, you need to configure APP. APP enables a PDB to exchange zone index information with other PDBs in a peer mesh and a PDNS to exchange address records and keepalive information with other PDNSs in a peer mesh. For information on configuring APP, see the "Configuring the Application Peering Protocol" section in Chapter 1, Configuring the CSS as a Domain Name System Server.

Configuring APP-UDP for proximity requires you to enable APP-UDP.

Optionally, you can configure additional APP-UDP parameters as follows:

- Securing APP-UDP Datagrams
- Specifying APP-UDP Options
- Removing an APP-UDP Options Record
- Specifying the APP-UDP Port
- Showing APP-UDP Configurations

Enabling APP-UDP

To configure APP-UDP datagram messaging on the PDB and all PDNSs in each zone, use the **app-udp** command. This command is available in global configuration mode.

The **app-udp** command supports the following options:

- **app-udp** - Enables APP-UDP datagram messaging
- **app-udp secure** - Specifies that all incoming APP-UDP datagrams must be encrypted
- **app-udp options** - Configures APP-UDP options used when communicating with a CSS peer
- **app-udp port** - Sets the UDP port that listens for APP-UDP datagrams

For example:

```
(config)# app-udp
```

To disable APP-UDP messaging, enter:

```
(config)# no app-udp
```

Securing APP-UDP Datagrams

Encryption prevents unauthorized messages from entering the CSS. To require that all incoming APP-UDP datagrams be encrypted, use the **app-udp secure** command. This command is used in conjunction with the **app-udp options** command that specifies secure messages that the CSS accepts.

Caution Using this command without the **app-udp options** command results in all incoming data being dropped.

The syntax for this global configuration mode command is:

app-udp secure

The following example illustrates the use of the **app-udp secure** command. In this example, this configuration allows only incoming traffic from IP address 200.16.2.3 encrypted with the password *mysecret*. The password is an unquoted text string with a maximum of 31 characters. There is no default.

For example:

```
(config)# app-udp
(config)# app-udp secure
(config)# app-udp options 200.16.2.3 encrypt-md5hash mysecret
```

To restore the default behavior of the CSS to accept all APP-UDP datagrams, enter:

```
(config)# no app-udp secure
```

Specifying APP-UDP Options

The **app-udp** command allows you to specify the encryption method and secret for datagrams sent to or received from an IP address. The CSS applies the options to packets sent to the destination address or applies them when the CSS receives datagrams with a matching source IP address. You can configure the IP address to 0.0.0.0 to apply a set of security options to all inbound and outbound datagrams that are not more specifically configured. Using the IP address 0.0.0.0 allows you to set a global security configuration that can be applied to an arbitrary number of peers.

To associate APP-UDP options with an IP address, use the **app-udp options** command.

The syntax for this global configuration mode command is:

app-udp options ***i****p_address* **encrypt-md5hash** *secret*

The **app-udp options** command contains optional fields that allow you to encrypt datagrams. This encryption method applies to datagrams sent and received over an IP address. Encryption options include:

- *ip_address* - The IP address associated with this group of options. Enter the address in dotted-decimal notation (for example, 200.16.2.3).
- **encrypt-md5hash** - The MD5 hashing method used for datagram encryption.
- *secret* - The string used in the encryption and decryption of the MD5 hashing method. Enter an unquoted text string with a maximum of 31 characters. There is no default.

The following example configures the IP address with the **encrypt-md5hash** global option. Datagrams sent to or received from 200.16.2.3 are encrypted with the password *mysecret*. All other datagrams received or transmitted, are subjected to the default encryption secret *anothersecret*.

For example:

```
(config)# app-udp
(config)# app-udp options 200.16.2.3 encrypt-md5hash mysecret
(config)# app-udp options 0.0.0.0 encrypt-md5hash anothersecret
```

Removing an APP-UDP Options Record

To remove an options record, use the **no app-udp options** command. This command includes an *ip_address* option to enable the CSS to disassociate an IP address from a group of options. Enter the address in dotted-decimal notation (for example, 200.16.2.3).

The syntax for this global configuration mode command is:

no app-udp options *ip_address*

For example:

```
(config)# no app-udp options 200.16.2.3
```

Specifying the APP-UDP Port

To set the UDP port number that listens for APP-UDP datagrams, use the **app-udp port** command. The **app-udp port** command includes the *port_number* variable, which specifies the UDP port number. Enter a value from 1025 to 65535. The default is 5002.

The syntax for this global configuration mode command is:

app-udp port *port_number*

For example:

```
(config)# app-udp port 2
```

To restore the UDP port number to its default value of 5002, enter:

```
(config)# no app-udp port
```

Note Now that you have configured APP-UDP, you must configure APP as described in Chapter 1, Configuring the CSS as a Domain Name System Server, in the "Configuring the Application Peering Protocol" section to enable PDB and PDNS peer meshes.

Showing APP-UDP Configurations

To display APP-UDP global statistical information and security configuration settings, use the **show app-udp** command.

The options for the **show app-udp** command are:

- **show app-udp global** - Displays global statistical information about the operation of APP-UDP
- **show app-udp secure** - Displays the current security configuration settings for APP-UDP

For example, to display statistical information about the operation of APP-UDP, enter:

```
(config)# show app-udp global
```

Table 5-3 describes the fields in **show app-udp global** output.

Table 5-3 Field Descriptions for the show app-udp global Command

Field	Description
Transmit Frames	The number of frames transmitted through APP-UDP
Transmit Bytes	The number of bytes transmitted through APP-UDP
Transmit Errors	The number of frames dropped because of transmits resource errors
Receive Frames	The number of frames received through APP-UDP
Receive Bytes	The number of bytes received through APP-UDP
Receive Errors	The number of frames dropped because of security mismatches

For example, to display the current security configuration settings for APP-UDP, enter:

```
(config)# show app-udp secure
```

Table 5-4 describes the fields in the **show app-udp secure** output.

Table 5-4 Field Descriptions for the show app-up secure Command

Field	Description
Allow non-secure	The setting for whether or not encryption is a requirement for all inbound APP datagrams. Yes indicates that the CSS will accept all datagrams (default). No indicates that encryption is required.
IP Address	The IP address associated with this group of APP-UDP options.
Type	The encryption method. Currently, the only method is MD5 hashing.
Secret	The string used in encryption and decryption of the MD5 hashing method.

Enabling the PDB

Note Before you enable the PDB, you must configure APP-UDP and APP. For details on configuring APP-UDP, see "Configuring APP-UDP and APP" earlier in this chapter. For details on configuring APP, see the "Configuring the Application Peering Protocol" section in Chapter 1, Configuring the CSS as a Domain Name System Server.

To enable a PDB on a dedicated CSS 11150 with 256 MB of RAM, use the **proximity db** command. For a detailed description of a PDB, see the "Proximity Database" section.

Once you have enabled APP-UDP and APP, **proximity db** is the only command that is required to use the PDB. Other PDB commands are optional, but recommended, depending on your application. For details, see each command description in the following sections.

The syntax for this global configuration mode command is:

proximity db *zoneIndex* {**tier1**|**tier2**} {**"***description***"**}}

The **proximity db** command supports the following variables and options:

- *zone_index* - Numerical identifier of the proximity zone of a CSS. This number should match the zone index you configured on the PDNS. For tier1, enter an integer from 0 to 5. For tier2, enter an integer from 0 to 15. There is no default.
- **tier1 | tier2** - Specification of the tier in which a CSS participates. The tier dictates the maximum number of proximity zones that may participate in the mesh. Enter **tier1** for a maximum of six proximity zones. Enter **tier2** for a maximum of 16 proximity zones. The default is **tier1**.
- **"***description***"** - Optional quoted text description of a CSS proximity zone. Enter a quoted text string with a maximum of 32 characters.

For example:

```
(config)# proximity db 1 tier1 "pdb-usa"
```

To disable the Proximity Database, enter:

```
(config)# no proximity db
```

Assigning Proximity Metrics

Use the **proximity assign** command to provide a local metric or to provide metrics (in milliseconds) for all proximity zones. The **proximity assign** command overrides the default metric determination processes. This command allows you to turn off probe traffic to Classless Inter-Domain Routing (CIDR) blocks.

When you use this command, Network Proximity does not perform active probing of the assigned block. Assigned information is shared with all PDBs in the PDB mesh. You can use this Network Proximity command only on a PDB.

Note The **proximity assign** command is not added to the running-config.

The syntax for this SuperUser configuration mode command is:

proximity assign *ip_address ip_prefix* [**"***local_metric***"**|**"***metric_list***"**]

The **proximity assign** command supports the following variables:

- *ip_address* - Assigns metric information to the IP address.
- *ip_prefix* - Assigns metric information to the IP prefix length. Enter the prefix as either:
  - A prefix length in CIDR bitcount notation (for example, /24).
  - A subnet mask in dotted-decimal notation (for example, 255.255.255.0).
- **"***local_metric***"** - A single quoted metric (in milliseconds) used to represent the proximity zone where the command is issued. Enter a value between 1 and 255.
- **"***metric_list***"** - A list of quoted metrics (in milliseconds), in ascending proximity zone order, for the zones where you want to apply the **proximity assign** command. Enter a value between 0 and 255. A value of zero indicates no assignment for a zone, and is only a placeholder in a list of assigned metrics.

For example, the following command uses the *local_metric* variable to assign a value of 200 to all client DNS addresses included in the range **172.23.5.7/24.**

```
# proximity assign 172.23.5.7/24 "200"
```

This command uses the *metric_list* variable to assign a value of 200 ms to proximity zone 0, does not configure zone 1 (specified by a value of zero), and assigns a value of 50 ms to zone 2.

```
# proximity assign 172.23.5.7/24 "200 0 50"
```

Flushing Proximity Assignments

Use the **proximity assign flush** command to remove existing proximity assignments configured with the **proximity assign** command. You can use this Network Proximity command only on a PDB.

Note Using the **proximity assign flush** command without additional syntax removes all proximity assignments.

The syntax for this SuperUser configuration mode command is:

proximity assign flush {*ip_address ip_prefix*}

The **proximity assign flush** command supports the following variables:

- *ip_address* - The IP address of previous proximity assignments you wish to remove.
- *ip_prefix* - IP prefix of previous proximity assignments you wish to remove. Enter the prefix as either:
  - A prefix length in CIDR bitcount notation (for example, /24).
  - A subnet mask in dotted-decimal notation (for example, 255.255.255.0).

For example:

```
# proximity assign flush 172.23.5.7/24
```

Configuring Proximity Time to Live

Use the **proximity ttl** command to set the TTL value, in minutes, for each PDB response. This value tells the PDNS how long to cache the PDB response. You can use this Network Proximity command only on a PDB.

The syntax for this global configuration mode command is:

proximity ttl assigned | probe *minutes*

The options for this global configuration mode command are:

- **proximity ttl assigned** *minutes* - Sets the TTL value for client addresses that are assigned at the PDB. Enter a value from 0 to 255. The default is 60.
- **proximity ttl probe** *minutes* - Sets the TTL value for client addresses that are being probed by the PDB. Enter a value from 0 to 255. The default is 0, which disables the caching of responses.

For example:

```
(config)# proximity ttl assigned 25
```

To reset the TTL value to its default, enter:

```
(config)# no proximity ttl probe
```

Note A TTL value of 255 never ages out the entries.

Storing the PDB

Use the **proximity commit** command to write a portion or all of the proximity database to a file in the log directory on the CSS disk or to a file on an FTP server. This command is useful for exporting the database so that you can view, modify, or recover information in the PDB. The database output contains metrics for all proximity zones, the current advertisement state, and hit counts.

To retrieve the database log file, use the **proximity retrieve** command. You can use this Network Proximity command only on a PDB.

By default, when you enter this command without any of its options, it writes the entire database to an XML-formatted file named proximity.db in the log directory on the CSS disk. You can optionally specify that the database be encoded using compact binary encoding. You can also specify that the database be written to a file on an FTP server.

The syntax for this SuperUser command is:

proximity commit {*ip_address ip_prefix*|**entire-db** {**ftp** *ftp_record ftp_filename* {**bin**}|**log** *filename* {**bin**}}}

The **proximity commit** command supports the following variables and options:

- *ip_address* - The starting IP address in the database that you want to write to the CSS disk or FTP server. Enter the IP address in dotted-decimal notation (for example, 175.23.5.7).
- *ip_prefix* - The IP prefix length of the database that you want to write to the CSS disk or FTP server. Enter the prefix as either:
  - A prefix length in CIDR bitcount notation (for example, /24).
  - A subnet mask in dotted-decimal notation (for example, 255.255.255.0).
- **entire-db** - An optional keyword to commit the entire PDB. By default, the entire database is written to disk.
- **ftp** - An optional keyword to write a specified file to an FTP server.
- *ftp_record* - The name for the FTP record file. Enter an unquoted text string with no spaces and a maximum of 16 characters. You must create an FTP record using the global config **ftp-record** command. For information on configuring an FTP record, refer to the *Cisco Content Services Switch Administration Guide*.

- *ftp_filename* - The filename to use when copying the database to an FTP server.
- **log** - An optional keyword to write a specified file to the log directory on the CSS disk.
- *filename* - The filename to use when storing the PDB in the log directory on the CSS disk. Enter a filename with a maximum of 32 characters. The default filename is *proximity.db*.
- **bin** - Specifies binary output for the PDB. A binary encoded database requires approximately 32 bytes per entry.

Note An XML database occupies approximately three times the space a binary-encoded database occupies. However, a binary encoded database cannot be viewed.

For example:

```
# proximity commit 172.23.5.7/24 xml
```

Retrieving the PDB

Use the **proximity retrieve** command to load a PDB from disk or an FTP server. Proximity metrics loaded from the database file replace any overlapping entries existing in the database and supplement any non-overlapping database entries. You can use this Network Proximity command only on a PDB.

Note If you enter the **proximity retrieve** command without any of its options, the CSS loads the file *proximity.db* from disk by default.

The **proximity retrieve** command distinguishes between XML encoded and binary database formats automatically.

The syntax for this SuperUser command is:

proximity retrieve {**ftp** *ftp_recordname* **ftp** *ftp_filename*|**log** *filename*}

The **proximity retrieve** command supports the following variables:

- **ftp -** The optional keyword to retrieve a specified file from an FTP server.
- *ftp_recordname* - The name of an existing FTP record for an FTP server. The FTP record contains the FTP server IP address, username, and password. To create an FTP record, use the **(config) ftp-record** command.
- *ftp_filename* - The PDB filename located on the FTP server.
- **log** - The optional keyword to retrieve a specified file (other than the proximity.db file) from the log directory on the CSS disk.
- *filename* - The PDB filename located in the log directory on the CSS disk.

For example:

```
# proximity retrieve ftp proxconfig proxconfignew
```

Refining Proximity Metrics

Use the **proximity refine** and the **proximity refine once** commands to initiate the continuous or single refinement, respectively, of metric entries in the PDB. Refinement updates the metric entries for all clients in the database to reflect conditions that exist at a particular point in time. You can use these Network Proximity commands only on a PDB.

When you issue the **proximity refine** command, the PDB probes all existing clients in the database periodically based on the size of the database and the database hit counts for the clients. The PDB organizes clients into three groups by hit count: N1, N2, and N3. The PDB probes N1 more frequently than N2, and N2 more frequently than N3. The percentage of time spent probing N1, N2, and N3 is approximately 45%, 35%, and 20%, respectively.

When you issue the **proximity refine once** command, the PDB probes all existing clients in the database only once.

The syntax for these SuperUser configuration mode commands are:

proximity refine

proximity refine once

To stop a refinement in progress, enter:

```
# no proximity refine
```

Using Proximity Reprobe

Use the **proximity reprobe** command to send additional probes to existing IP addresses in the proximity database once. You can use this Network Proximity command only on a PDB. You can use the **proximity reprobe** command with the **proximity refine** commands.

Note IP addresses configured with the **proximity assign** command are not eligible for reprobing.

The syntax for this SuperUser configuration mode command is:

proximity reprobe *ip_address* {*ip_prefix*}

The **proximity reprobe** command supports the following variables:

- *ip_address* - The IP address to probe.
- *ip_prefix* - The optional IP prefix to associate with the IP address that performs probing for a source block of addresses. Enter the prefix as either:
  - A prefix length in CIDR bitcount notation (for example, /24).
  - A subnet mask in dotted-decimal notation (for example, 255.255.255.0).

For example:

```
# proximity reprobe 172.23.5.7/24
```

Clearing the PDB

Use the **proximity clear** command to remove entries from the proximity database.

Caution Be sure you want to permanently delete entries in the PDB before you use this command. Using the **proximity clear** command without optional variables permanently removes all entries in the proximity database.

The syntax for this SuperUser command is:

proximity clear *ip_address ip_prefix*

The **proximity clear** command supports the following variables:

- *ip_address* - The IP address for the entries you want to remove. Enter the address in dotted-decimal format (for example, 172.23.5.7).
- *ip_prefix* - The IP prefix length used in conjunction with the IP address. Enter the prefix as either:
  - A prefix length in CIDR bitcount notation (for example, /24).
  - A subnet mask in dotted-decimal notation (for example, 255.255.255.0)

Configuring the Proximity Probe Module

The Proximity Probe Module is responsible for sending ICMP and TCP probes to clients based on PDNS lookup requests to the PDB and refinement settings. See the following sections to configure the Proximity Probe Module:

- Configuring the Proximity Probe Module Method
- Specifying the Proximity Probe Module Samples
- Configuring the Proximity Probe Module Metric Weighting
- Configuring the Proximity Probe Module Interval
- Specifying Proximity Probe Module TCP-ports

Configuring the Proximity Probe Module Method

Use the **proximity probe rtt method** command to configure the primary and secondary methods used for proximity metric discovery. You can use this Network Proximity command only on a PDB.

The syntax for this global configuration mode command is:

proximity probe rtt method [**icmp** {**tcp**}|**tcp** {**icmp**}]

The **proximity probe rtt method** command supports the following options:

- **icmp** - Use ICMP Echo requests as the primary method. The default is **icmp**.
- **tcp** - Use a TCP SYN/SYN ACK approach to the configured TCP ports as the primary RTT discovery method.

For example:

```
(config)# proximity rtt method icmp
```

Specifying the Proximity Probe Module Samples

Use the **proximity probe rtt samples** command to configure the number of ICMP requests to send for each client probe. You can use this Network Proximity command only on a PDB.

Note Because only one TCP SYN request is sent, you cannot configure this command for TCP probes.

The syntax for this global configuration mode command is:

proximity probe rtt samples *number*

The *number* variable specifies the default number of ICMP echo requests used for averaging during an initial probe. Enter a number from 1 to 30. The default is 2.

For example:

```
(config)# proximity probe rtt samples 5
```

To reset the number of ICMP echo requests to its default value of 2, enter:

```
(config)# no proximity probe rtt samples
```

Configuring the Proximity Probe Module Metric Weighting

Use the **proximity probe rtt metric-weighting** command to configure the percentage of the previously stored metric value in the database that is used to determine the new metric value. This command allows the PDB to smooth network metric variation caused by network congestion, flash crowds, and so on. You can use this Network Proximity command only on a PDB.

The syntax for this global configuration mode command is:

proximity probe rtt metric-weighting *number*

The *number* variable specifies the percentage of the previous metric value used. Enter a number from 0 to 99. The default is 0.

For example:

```
(config)# proximity probe rtt metric-weighting 10
```

For this example, suppose the previously stored metric value for a client's local DNS server is 40 and the current metric value is 50. If you issue the command above, the PDB adds 10% of the previous metric value (0.10 x 40) to 90% of the current metric value (0.90 x 50) to determine the new metric value. So, the new metric value would be 49. A *number* value of 50 causes the PDB to average the previous and current metric values.

To reset this command to its default value of 0, enter:

```
(config)# no proximity probe rtt metric-weighting
```

Configuring the Proximity Probe Module Interval

Use the **proximity probe rtt interval** command to configure the delay in seconds between ICMP samples. You can use this Network Proximity command only on a PDB.

The syntax for this global configuration mode command is:

proximity probe rtt interval *seconds*

The *seconds* variable identifies the length of time (in seconds) to wait between ICMP samples. Use a range between 1 to 10. The default is 1.

For example:

```
(config)# proximity probe rtt interval 5
```

To reset the delay between samples to its default value of 1 second, enter:

```
(config)# no proximity probe rtt interval
```

Specifying Proximity Probe Module TCP-ports

Use the **proximity probe rtt tcp-ports** command to configure the probe ports for TCP proximity metric discovery. You can use this Network Proximity command only on a PDB.

The syntax for this global configuration mode command is:

proximity probe rtt tcp-ports *port_number1* {*port_number2* {*port_number3* {*port_number4*}}}

Define the port number to be tried, in order of precedence. Enter a number from 1 to 65535 to enable a port. The defaults for the various port numbers include:

- *port_number1* is 23, Telnet port
- *port_number2* is 21, FTP port
- *port_number3* is 80, HTTP port
- *port_number4* is 0, this port is not tried

Note Ports that you do not specify default to 0.

To reset the probe ports to their default values, enter:

```
(config)# no proximity probe rtt tcp-ports
```

Using Network Proximity Tiers

The following sections describe the advanced Network Proximity concept of *tiers*. Network Proximity uses tiers to further expand the proximity architecture by allowing you to create more distinct network zones and subzones.

Proximity Tiers

Sharing information among multiple PDBs may result in the management of a very large data set. As you add more proximity zones to the network, Network Proximity scales to provide more distinct network zones, allowing zones or subzones to exist within other zones. Network Proximity treats these zones as:

- Level 1 zones (first level)
- Level 2 zones (nested levels)

Note You can configure six Level 1 proximity zones and 16 Level 2 proximity zones. A Level 1 tier supports up to 2 million unique local DNS server addresses. A Level 2 tier supports slightly less than one million unique local DNS server addresses.

In a tiered Network Proximity model, the owner of the name server record is a nested PDNS that is communicating with a nested PDB located within the Level 2 proximity subzone.

Example of Tiered Network Proximity

In Figure 5-4, a two-tiered configuration example illustrates how you can use tiers to group more specific network proximity zones. The proximity zone that encompasses all network devices within the United States is broken down further to include an additional tier that comprises the more specific geographical proximity zones, East Coast and West Coast.

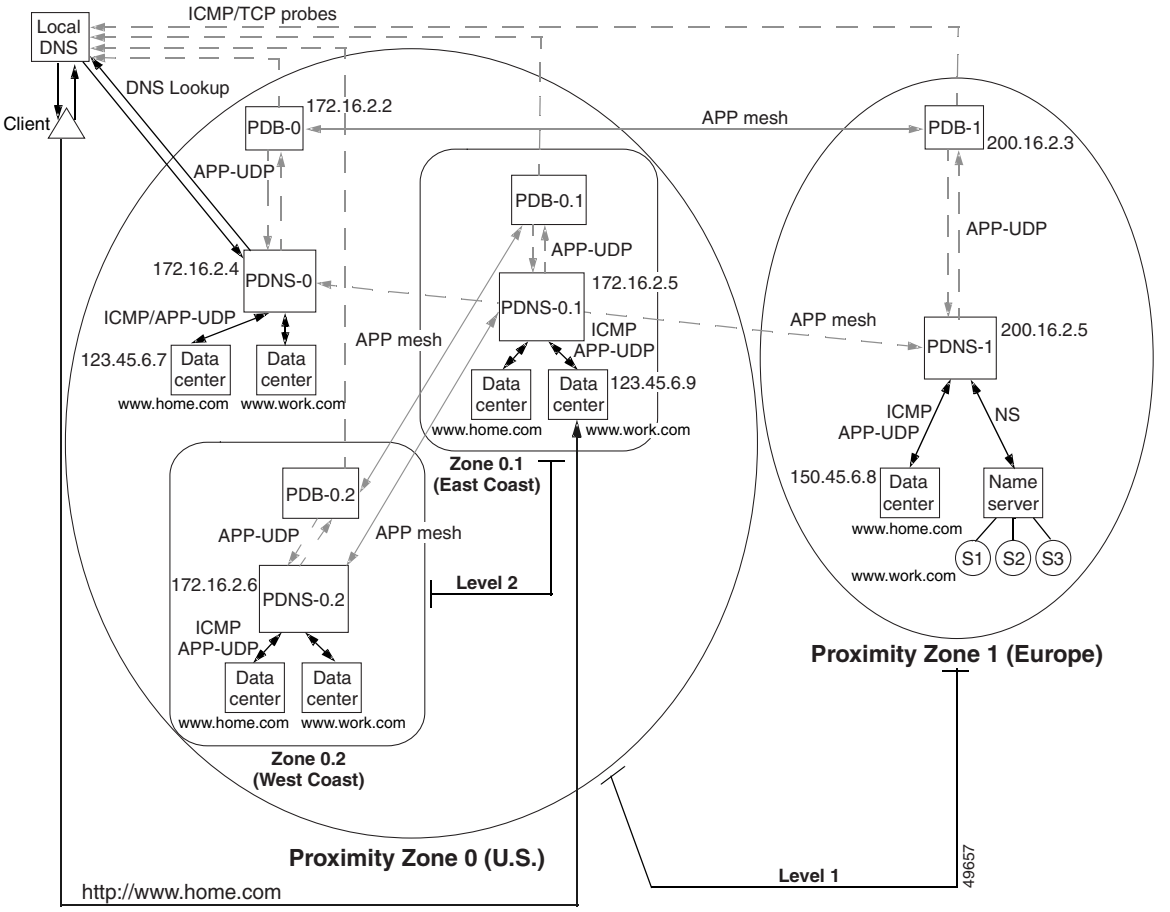
By adding a tier to this configuration, the capacity of Network Proximity is extended by creating two subzones (Zones 0.1 and 0.2) that include additional PDBs, PDNSs, and data centers. In this way, you can scale Network Proximity to meet your users' needs with increased proximity specificity and thereby increase network performance.

The following steps describe how Network Proximity determines the most proximate service for a client requesting the domain *www.work.com*. See Figure 5-4.

1. The client performs an HTTP request for the domain name *www.work.com*.
2. The client's DNS server performs iterative DNS requests to the root server and to the .com server to start resolving the domain name into an IP address. (This step is not shown in Figure 5-4.)
3. The .com server, which is an authoritative DNS for *work.com*, has the IP addresses of PDNS-0 and PDNS-1 in its configuration. The Level 1 PDNSs are authoritative DNSs for *www.work.com*. In this example, the .com server refers the client's DNS server to PDNS-0 in Zone 0. (Typically, the .com server uses a roundrobin or other load-balancing method to refer local DNS servers to a PDNS. This step is not shown in Figure 5-4.)

Note Your configuration may include an enterprise DNS server that is positioned between the .com server and the PDNSs. The enterprise DNS server would be an authoritative DNS server for *work.com*. In this case, the enterprise DNS server contains the IP addresses of the PDNSs in its configuration and refers the local DNS server to the appropriate PDNS. In either configuration, the PDNS is authoritative for *www.work.com*.

Figure 5-4 Tiered Network Proximity Configuration



4. The local DNS server forwards the client's request for *www.work.com* to PDNS-0 in Zone 0.
5. PDNS-0 determines the most proximate zone to send the client to using one of the following scenarios:
   a. PDNS-0 first searches its cache for a previously saved ordered zone index, a preferred order of zones closest to the client as determined by PDB-0 and based on information from probes and the PDB's peer mesh.

If PDNS-0 finds the ordered zone index in its cache, it uses that data along with keepalive information and domain records (locally configured and learned through its peer mesh) to determine the most proximate zone to service the client.

b. If the ordered zone index is not cached, PDNS-0 sends to PDB-0 (using APP-UDP) a lookup request that contains the IP address of the client. PDB-0 calculates the preferred order of zones for the client and returns the ordered zone index to PDNS-0 immediately. PDNS-0 uses the zone order along with keepalive information and domain records to determine the most proximate zone to service the client.

c. If the ordered zone index is not cached and PDB-0 is not available, PDNS-0 uses its keepalive information, domain records, and a roundrobin method to select a service to handle the request.

6. If the PDNS determines that the best selection is a name server (NS) record, the PDNS begins a recursive query of the name server to determine an authoritative response. If the PDNS finds that the best selection is an address record (A-record), it formulates an authoritative response immediately. In this example, PDNS-0 decides that the best selection is a name server (NS) record for PDNS-0.1 in Zone 0.1.

7. PDNS-0.1 uses the same logic outlined in steps 5 and 6 above to determine the most proximate service for the client. In this example, PDNS-0.1 decides that the best selection is an address record (A-record) for one of its attached data centers. PDNS-0.1 then makes an authoritative response to PDNS-0 with the A-record for the data center that contains *www.work.com*.

8. PDNS-0 sends an authoritative response that contains the resolved IP address of *www.work.com* to the client's local DNS server.

9. The client's DNS server notifies the client that sufficient domain name resolution information is available to establish a data connection to *www.work.com*.

10. Lastly, the client uses the resolved IP address to connect to a service in the most proximate zone and starts receiving content. In this example, the most proximate service is located in Proximity Zone 0.1 (East Coast) at IP address 123.45.6.9.

Displaying PDB Configurations

The CSS provides a comprehensive set of Network Proximity **show** commands that display information about the PDB. Use the **show proximity** command to display PDB configuration or session information. See the following sections for information on using Proximity Database show commands:

Displaying the PDB

Use the **show proximity** command to display an activity summary of the proximity database. This command functions only on a PDB.

For example:

```
# show proximity
```

Table 5-5 describes the fields in the **show proximity** output.

Table 5-5 Field Descriptions for the show proximity Command

Field	Description
Lookups	The total number of resolved proximity requests
Lookup Rate	The number of resolved proximity requests per second
Probe TTL	The configured time-to-live value for client addresses that are probed
Assigned TTL	The configured time-to-live value for client addresses that are assigned to the Proximity Database

Table 5-5 Field Descriptions for the show proximity Command (continued)

Field (continued)	Description
Assigned Blocks	Blocks in the PDB that are assigned
Probed Blocks	Blocks in the PDB that are probed
Total Blocks	Total number of blocks in the PDB
Last Retrieve	The last time that a proximity retrieve was executed
Last Commit	The last time that a proximity commit was executed
DB Utilization	Percentage of the PDB used
Refinement	Whether or not refinement is activated
Total Peers	The total number of peers in the PDB mesh

Note All database values are cleared when you reboot the CSS or you issue the **no proximity db** command.

Displaying Proximity Metrics

Use the **show proximity metric** command to display metrics (in milliseconds) associated with a client's local DNS server IP address. This command is available on a PDNS and a PDB.

The syntax and options for this global configuration mode command are:

show proximity metric *ip_address* {*ip_prefix* {**aggregate**}}

- *ip_address* - The IP address of a client's local DNS server for metric display. Enter the address in dotted-decimal notation (for example 172.23.5.7).
- *ip_prefix* - This optional parameter is used to map an IP prefix to an IP address allowing you to view metrics over a range of IP addresses. Enter the prefix as either:
  - A prefix length in CIDR bitcount notation (for example, /24).
  - A subnet mask in dotted-decimal notation (for example, 255.255.255.0).

- **aggregate** - This optional parameter allows you to view aggregated metrics that are available in both /16 and /8 subnet masks.

Note Probed metrics are statistically aggregated at the /8 and /16 prefix levels.

For example, to view the proximity metrics associated with the client IP address of 172.23.5.7 and an IP prefix of 24, enter:

```
(config)# show proximity metric 172.23.5.7/24
```

In the PDB, the RTT metrics are sorted by proximity zone. In the PDNS, the metrics are sorted by RTT. An asterisk next to a zone indicates the zone where the command was issued.

Note The maximum value of an RTT metric is 3968 ms. A value of 4095 ms indicates that a client's local name server was unreachable or had an RTT value of more than 4 seconds.

Table 5-6 describes fields in the **show proximity metric** output.

Table 5-6 Field Descriptions for the show proximity metric Command

Field	Description
Index	The zone index number associated with the PDNS zone. An asterisk (*) indicates the local zone where you issued this command.
Description	A logical name or description to the zone.
Metric	Round-Trip Time (RTT) between the PDB and a Referral-DNS as the proximity metric for load balancing decisions.

Displaying Proximity Statistics

Use the **show proximity statistics** command to display statistics associated with client IP addresses. This Network Proximity command is only available on the PDB.

The syntax for this global configuration mode command is:

show proximity statistics *ip_address* {*ip_prefix* {**aggregate**}}

The variables and options for this command are:

- *ip_address* - The IP address for statistics display. Enter the address in dotted-decimal notation (for example 172.23.5.7).
- *ip_prefix* - This optional parameter is used to map an IP prefix to an IP address. This allows you to view metrics over a range of IP addresses, indicated by the prefix. Enter the prefix as either:
  - A prefix length in CIDR bitcount notation (for example, /24).
  - A subnet mask in dotted-decimal notation (for example, 255.255.255.0).
- **aggregate** - This optional parameter allows you to view aggregated statistics that are available in both /16 and /8 subnet masks.

For example, to view the proximity statistics associated with the client IP address of 10.1.0.0 and an IP prefix of 16, enter:

```
(config)# show proximity statistics 10.1.0.0/16
```

Table 5-7 describes fields in the **show proximity statistics** output.

Table 5-7 Field Descriptions for the show proximity statistics Command

Field	Description
IP/Prefix	The IP address and prefix associated with the statistics display.
Lookup Count	The number of resolved proximity requests per second.

Displaying Proximity Refinement

Use the **show proximity refine** command to display information pertaining to entries being refined in the PDB. This Network Proximity command is only available on a PDB. For an explanation of the N1, N2, and N3 groups mentioned below, see "Refining Proximity Metrics" earlier in this chapter.

For example:

```
(config)# show proximity refine
```

Table 5-8 describes fields in the **show proximity refine** output.

Table 5-8 Field Descriptions for the show proximity refine Command

Field	Description
N1 - N3 Count	The number of entries in each N class
N1 - N3 Percent	Of all entries, the percentage of entries in the N class
N1 - N3 Rate	The number of probes per second
N1 - N3 Probed	The total number of probes since the **proximity refine** command was invoked
N1 - N3 Cycle Time	The amount of time to cycle through the N count
Aggregate Count	The total count for N1 through N3
Aggregate Probed	The probed total for N1 through N3
Aggregate Rate	The rate total for N1 through N3

Displaying Proximity Assignments

Use the **show proximity assign** command to display the user-assigned metric values (in milliseconds) of all proximity zones or for a configured IP address range.

The syntax and variables for this global configuration mode command are:

show proximity assign {*ip_address ip_prefix*}

- *ip_address* - The optional IP address to display metrics over a range of IP addresses. Enter the IP address in dotted-decimal format (for example, 172.23.5.7).
- *ip_prefix* - The optional IP prefix to associate with the IP address that performs probing for a source block of addresses. Enter the prefix as either:
  - A prefix length in CIDR bitcount notation (for example, /24).
  - A subnet mask in dotted-decimal notation (for example, 255.255.255.0).

For example, to view the metric assignments for all IP addresses within the range of 200.16.0.0 to 200.16.255.255, enter:

```
(config)# show proximity assign 172.23.5.7/16
```

Table 5-9 describes the fields in the **show proximity assign** output.

Table 5-9 Field Descriptions for the show proximity assign Command

Field	Description
IP/Prefix	The IP address to search for in the cache and the IP prefix associated with the IP address for cache searching
Hits	The total number of hits
Zone Metrics	The list of metrics in ascending order to represent all zones

Displaying Proximity Zones

Use the **show proximity zone** command to view the state information for each proximity zone, excluding the local proximity zone. This command is similar to the **show zone** command for the PDNS; however, the **show proximity zone** command provides information from the perspective of the PDB. This Network Proximity command is available only on a PDB.

The syntax for this global configuration mode command is:

show proximity zone {*number*}

Use the *number* variable to display the state information for a specific proximity zone. Enter a zone number from 0 to 15.

For example, to display the state information for proximity zone 1, enter:

```
(config)# show proximity zone 1
```

Table 5-10 describes the fields in the **show proximity zone** output.

Table 5-10 Field Descriptions for the show proximity zone Command

Field	Description
Index	The local index number associated with the PDNS zone. The * indicates the local zone where you issued this command.
Description	A text description of the zone that associates a logical name description to the zone.
IP Address	The IP address used for PDB communication with the zone peer.
State	The state of the PDB connection with the peer, which includes: • **Initializing** - The PDB state connection is initializing • **Sync** - The PDB state connection is synchronizing with the peer • **Normal** - The PDB state connection is normal • **Illegal** - The PDB state is an illegal connection
UpTime	Elapsed time since the **proximity db** command was executed locally, or since the peer entered the PDB mesh.

Displaying Proximity Zone Statistics

The **show proximity zone statistics** command displays information about the APP peer mesh blocks sent and received for a peer for all proximity zones.

The syntax for this global configuration mode command is:

show proximity zone statistics {*number*}

Use the *number* variable to display statistics for a specific proximity zone. Enter a zone number from 0 to 15.

For example, to display the peer block information for zone 1, enter:

```
(config)# show proximity zone statistics 1
```

Table 5-11 describes the fields in the **show proximity zone statistics** display.

Table 5-11 Show Proximity Zone Statistics Display Fields

Field	Description
Index	The local index number associated with the proximity zone.
Description	A text description of the proximity zone that associates a logical name description with the proximity zone as entered with the **proximity db** command.
Sent	The number of blocks sent to the peer.
Received	The number of blocks received from the peer.
Last Update	The last time information was exchanged between the local PDB and the peer in either direction.

Displaying Proximity Probe Module Statistics

Use the **show proximity probe rtt statistics** command to view the Round Trip Time (RTT) probe module statistics.

The syntax for this global configuration mode command is:

show proximity probe rtt statistics

For example:

```
(config)# show proximity probe rtt statistics
```

Table 5-12 describes the fields in the **show proximity probe rtt statistics** output.

Table 5-12 Field Descriptions for the show proximity probe rtt statistics Command

Field	Description
Total Client Probes	The total number of times that the PDB has probed a client to measure the RTT value. This may be more than the total number of unique clients and may be less than the actual number of ICMP or TCP requests.
Average Probes/minute	The cumulative average number of probes per minute since the PDB was last reset.
ICMP requests sent	Specifies the number of ICMP probe requests used to calculate the RTT value.
ICMP responses	The total number of ICMP responses that the PDB has received. Valid ICMP responses are used to measure the RTT.
ICMP failures	The total number of ICMP requests that were successfully sent but did not receive a reply. The ICMP requests that do not receive a response are not used to measure the RTT value.
Average ICMP requests/minute	Specifies the time delay in seconds between consecutive ICMP requests to an individual client.
ICMP send failures	The total number of ICMP requests that the PDB tried to send but failed internally due to a missing route or other problem.

Table 5-12 Field Descriptions for the show proximity probe rtt statistics Command (continued)

Field (continued)	Description
TCP requests	The total number of TCP requests that have been successfully sent from the PDB in order to measure the RTT value.
TCP responses	The total number of TCP responses that the PDB has received. Valid TCP responses are used to measure the RTT value.
TCP failures	The total number of failed TCP requests destined for the port on the client's local name server.
Average TCP requests/minute	The cumulative average of TCP requests per minute that were successfully sent during the time period since the PDB was last reset.
TCP send failures	The total number of TCP requests that the PDB tried to send but failed internally due to a missing route or other problem.

Configuring a PDNS

The Proximity Domain Name Server (PDNS) is an authoritative DNS server that uses information from the Proximity Database (PDB) to resolve DNS requests based on an ordered zone index. As an authoritative DNS server, the PDNS uses domain records to map a given domain to an IP address or to a lower-level DNS server. You can configure a total of 1024 unique domain names for all PDNSs in a proximity mesh per proximity level. The same domain names can appear in all zones and on multiple PDNSs within a zone.

Note You must connect a PDNS to a PDB over a reliable link because of the requirements of the APP-UDP-based proximity lookup mechanism.

Configuring a PDNS involves the following required tasks:

- Configuring APP-UDP and APP
- Enabling the PDNS
- Configuring Domain Records

Optionally, you can perform the following PDNS-related tasks:

- Disabling the PDNS
- Clearing the DNS Server Statistics
- Enabling the Proximity Lookup Cache
- Removing Entries from the Proximity Lookup Cache

Configuring APP-UDP and APP

Network Proximity uses the Application Peering Protocol-User Datagram Protocol (APP-UDP) to exchange proximity information between a PDB and a PDNS, and between a PDNS and services. APP-UDP is a connectionless form of the Application Peering Protocol (APP). For details, see "Configuring APP-UDP and APP" earlier in this chapter.

Note In addition to configuring APP-UDP, you need to configure APP. APP enables a PDB and a PDNS to exchange proximity information with their peers. For information on configuring APP, see the "Configuring the Application Peering Protocol" section in Chapter 1, Configuring the CSS as a Domain Name System Server.

Enabling the PDNS

Note Before you enable the PDNS, you must configure APP-UDP and APP. For details on configuring APP-UDP, see "Configuring APP-UDP and APP" earlier in this chapter. For details on configuring APP, see the "Configuring the Application Peering Protocol" section in Chapter 1, Configuring the CSS as a Domain Name System Server.

Use the **dns-server zone** and **dns-server** commands to enable the PDNS. The syntax for this global configuration mode command is:

dns-server zone *zone_index* {**tier1|tier2** {**"***description***"** {*ip_address* {**round-robin|prefer-local**}}}}

The **dns-server zone** command supports the following variables and options:

- *zone_index* - Numerical identifier of the proximity zone of the CSS. This number should match the zone index configured on the PDB. Enter an integer from 0 to 15. Valid entries are 0 to 5 for tier 1 and 0 to 15 for tier 2. There is no default.
- **tier1 | tier2** - Specify the tier in which the CSS participates. The tier dictates the maximum number of proximity zones that may participate in the mesh. If you choose **tier1**, a maximum of six proximity zones may participate in the mesh. If you choose **tier2**, a maximum of 16 proximity zones may participate in the mesh. The default is **tier1.**
- *description* - Optional quoted text description of the CSS proximity zone. Enter a quoted text string with a maximum of 20 characters.
- *ip_address* - The IP address of the PDB. Enter the address in dotted-decimal notation (for example: 172.16.2.2). If you choose the zone capabilities (peer mesh) of a PDNS in a non-proximity environment, this variable is optional.
- **roundrobin|preferlocal -** The optional load-balancing method that the DNS server uses to select returned records when a Proximity Database is not configured or is unavailable.
  - **roundrobin** - The server cycles among records available at the different zones. This is the default method.
  - **preferlocal** - The server returns a record from the local zone whenever possible, using round-robin when it is not possible.

For example:

```
(config)# dns-server zone 1 tier1 "pdns-usa" 172.16.2.2
```

Configuring Domain Records

Use the **dns-record** command and its options to create a domain record on the PDNS. The PDNS uses two types of domain records to map a domain name to an IP address or to another DNS server:

- **A-record** - A domain record mapped to an IP address
- **NS-record** - A domain record mapped to a DNS server IP address

For details on configuring the **dns-record** command, see the "Configuring Domain Records" section in Chapter 1, Configuring the CSS as a Domain Name System Server.

Disabling the PDNS

Use the **no dns-server zone** command to disable the PDNS Proximity functions by removing the **dns-server zone** command configuration.

Disabling the PDNS:

- Prevents it from submitting proximity metric lookup requests to the PDB
- Stops the peer mesh communications and record keepalive processing

After issuing the **no dns-server zone** command, you can still use the PDNS as a DNS server.

For example:

```
(config)# no dns-server zone
```

Note Before you can issue this command, you must issue the **no dns-server** command. The **no dns-server** command also disables the Network Proximity functions and DNS server functions on the PDNS. Because this command does not delete the **dns-server zone** command configuration, you may want to use the **no dns-server** command to disable a PDNS temporarily.

Clearing the DNS Server Statistics

Use the **dns-server zero** command in global configuration mode to set the DNS server request and response statistics displayed by the **show dns-server** command to zero.

For example:

```
(config)# dns-server zero
```

Enabling the Proximity Lookup Cache

Use the **proximity cache-size** command to modify the size of the proximity lookup cache. The PDNS uses the proximity lookup cache to store PDB responses. The proximity lookup cache allows the PDNS to resolve proximity decisions faster by allowing a local lookup.

Note The proximity cache is limited to 48,000 entries.

The syntax for this global configuration mode command is:

proximity cache-size *cache_size*

The **proximity cache-size** command includes a *cache size* variable that specifies the size of the proximity lookup cache. Enter a value between 0 and 48,000. Entering a value of 0 disables the proximity lookup cache. Modifying the cache size results in flushing the existing entries. The default cache size is 16,000.

For example:

```
(config)# proximity cache-size 30000
```

To restore the default cache size (16,000 entries), enter:

```
(config)# no proximity cache-size
```

Removing Entries from the Proximity Lookup Cache

Use the **proximity cache-remove** command to remove entries from the proximity lookup cache. The prefix length parameter allows you to remove multiple entries in a single operation. This Network Proximity command can be used only on a PDNS.

The syntax for this SuperUser configuration mode command is:

proximity cache-remove *ip_address ip_prefix*|**all**

Note If you specify **all,** you cannot specify an *ip_address* or *ip_prefix* value.

The **proximity cache-remove** command supports the following variables and option:

- *ip_address* - The IP address to remove from the proximity cache.
- *ip_prefix* - The IP prefix length associated with the IP address removed from the proximity cache. Enter the prefix as either:
  - A prefix length in CIDR bitcount notation (for example, /24).
  - A subnet mask in dotted-decimal notation (for example, 255.255.255.0).
- **all** - This keyword removes all entries in the proximity lookup cache.

For example:

```
# proximity cache-remove 150.45.6.10 /24
```

Displaying PDNS Configurations

The CSS CLI provides a comprehensive set of Network Proximity **show** commands that display proximity configurations. See the following sections for information on using PDNS **show** commands:

Displaying the Proximity Cache

Use the **show proximity cache** command to display the current state of the proximity lookup cache. This display provides information about the current cache configuration, entries present, number of hits, and so on. This command is available only on the PDNS.

The syntax for this global configuration command is:

show proximity cache {**all**|*ip_address ip_prefix*}

The **show proximity cach** command supports the following variables and option:

- **all** - Display all addresses in the cache.
- *ip_address* - The IP address to search for in the cache.
- *ip_prefix* - The IP prefix to associate with the IP address for cache searching. Enter the prefix as either:
  - A prefix length in CIDR bitcount notation (for example, /24).
  - A subnet mask in dotted-decimal notation (for example, 255.255.255.0).

For example:

```
(config)# show proximity cache
```

Table 5-13 describes the fields in the show proximity cache screen.

Table 5-13 Show Proximity Cache Display Fields

Field	Description
Maximum Entries	The maximum number of entries the cache supports
Used Entries	The number of entries used by the cache
Free Entries	The number of free entries in the caches
Percent Available	The available percentage of unused cache
Hits	The number of cache lookup hits
Misses	The number of cache lookup misses
Percent Hits	The percentage of cache lookup hits

To display all information pertaining to the proximity cache, enter:

```
(config)# show proximity cache all
```

Table 5-14 describes the fields in the **show proximity cache all** screen.

Table 5-14 Show Proximity Cache All Display Fields

Field	Description
IP/Prefix	The IP address in the cache and the IP prefix associated with the IP address.
Hits	The total number of hits the cache received.
Descending Zone Proximity	Indices of desirable zones ordered by proximity to the client.
TTL	The TTL value associated with the cache entry. The “N” in the second row tells the PDNS to never age out the entries in the cache and is enabled by a TTL value of 255.

Displaying DNS Record Statistics

Use the **show dns-record statistics** command to display statistics associated with the domain records configured locally and learned by the CSS from its peers. For details, see the "Displaying DNS-Record Statistics" section in Chapter 1, Configuring the CSS as a Domain Name System Server.

Displaying DNS Record Keepalives

Use the **show dns-record keepalive** command to display information about keepalives associated with DNS records. For details, see the "Displaying DNS Record Information" section in Chapter 1, Configuring the CSS as a Domain Name System Server.

Displaying DNS Server Zones

Use the **show zone** command to display information about proximity zones communicating with a CSS Network Proximity service. For details, see the "Displaying DNS Server Zones" section in Chapter 1, Configuring the CSS as a Domain Name System Server. To display PDB-related zone information, see "Displaying Proximity Zones" earlier in this chapter.

Displaying DNS Record Proximity

Use the **show dns-record proximity** command to display dns-record proximity statistics.

The syntax for this global configuration mode command is:

show dns-record proximity

For example:

```
(config)# show dns-record proximity
```

Table 5-15 describes the fields in the **show dns-record proximity** output.

Table 5-15 Field Descriptions for the show dns-record proximity Command

Field	Description
<Domain name>	The domain name for the record.
Zone	The index number for the zone. A "*" character preceding the zone number indicates that the zone is a local entry. A value of 255 indicates that the record never came up.
Description	The proximity zone description.
Hits Optimal	This entry increments when the DNS server returns the index that the PDB indicated was most proximate.
Hits SubOptimal	This entry increments when the DNS server returns an index that is different from the first one that the PDB indicated was most proximate.
Misses Optimal	This field increments when the PDNS must send a client to a zone that is not indicated by the first zone index returned by the PDB.
Misses SubOptimal	This field increments when the PDNS must send a client to a zone that is not indicated by either the first or second zone index returned by the PDB.

Displaying DNS Server Information

Use the **show dns-server** command to display DNS server configuration and database information. Although this command is not specifically a PDNS command, it is nonetheless useful for displaying DNS server information. For details on using the **show dns-server** command, see the "Displaying CSS DNS Information" section in Chapter 1, Configuring the CSS as a Domain Name System Server.