



Cisco Content Services Switch Global Server Load-Balancing Configuration Guide

Software Version 7.40
August 2004

Corporate Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Text Part Number: OL-5652-01



THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCSP, the Cisco Square Bridge logo, Cisco Unity, Follow Me Browsing, FormShare, and StackWise are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn, and iQuick Study are service marks of Cisco Systems, Inc.; and Aironet, ASIST, BPX, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Empowering the Internet Generation, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, GigaDrive, GigaStack, HomeLink, Internet Quotient, IOS, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, LightStream, Linksys, MeetingPlace, MGX, the Networkers logo, Networking Academy, Network Registrar, *Packet*, PIX, Post-Routing, Pre-Routing, ProConnect, RateMUX, Registrar, ScriptShare, SlideCast, SMARTnet, StrataView Plus, SwitchProbe, TeleRouter, The Fastest Way to Increase Your Internet Quotient, TransPath, and VCO are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0406R)



Preface xv

Audience xvi

How to Use This Guide xvi

Related Documentation xvii

Symbols and Conventions xx

Obtaining Documentation xxi

 Cisco.com xxi

 Ordering Documentation xxi

Documentation Feedback xxii

Obtaining Technical Assistance xxii

 Cisco Technical Support Website xxiii

 Submitting a Service Request xxiii

 Definitions of Service Request Severity xxiv

Obtaining Additional Publications and Information xxiv

CHAPTER 1

Configuring the CSS as a Domain Name System Server 1-1

Overview of the CSS DNS Feature 1-2

 Zone-Based DNS 1-3

 Content Rule-Based DNS 1-4

Overview of the CSS Application Peering Protocol 1-4

 APP and Zone-Based DNS 1-4

 APP and Rule-Based DNS 1-6

Configuring the Application Peering Protocol 1-9

 Enabling APP 1-9

 Configuring the APP Frame Size 1-9

- Configuring the APP Port **1-10**
- Configuring an APP Session **1-10**
- Using the rcmd Command **1-12**
- Displaying APP Configurations **1-13**
- Configuring Zone-Based DNS on a CSS **1-16**
 - Zone-Based DNS Quick Start **1-17**
 - Configuring a DNS Server **1-19**
 - Enabling a DNS Server **1-20**
 - Configuring dns-server bufferCount **1-20**
 - Configuring a DNS Forwarder **1-21**
 - Configuring the Number of DNS Responder Tasks **1-22**
 - Configuring DNS Server Zones **1-22**
 - Configuring dns-server zone load **1-25**
 - Configuring Domain Records **1-26**
 - Removing a Domain Record **1-33**
 - Resetting the DNS Record Statistics **1-33**
 - Adding a DNS Name to a Content Rule **1-34**
 - Removing a DNS Name from a Content Rule **1-34**
 - Configuring DNS Records with a Zero Weight **1-35**
- Configuring kal-ap-vip **1-36**
 - Overview **1-36**
 - Configuration Requirements **1-37**
 - Configuring a kal-ap-vip Client **1-37**
- Configuring Content Rule-Based DNS on a CSS **1-38**
 - Content Rule-Based DNS Quick Start **1-39**
 - Configuring the DNS Exchange Policy for an Owner **1-41**
 - Configuring CSS DNS Peering **1-42**
 - Configuring the DNS Peer Interval **1-42**
 - Configuring DNS Peer Receive Slots **1-42**

Configuring DNS Peer Send Slots	1-43
Configuring DNS Peer Load Variance	1-43
Configuring a DNS Server	1-43
Enabling a DNS Server	1-44
Configuring dns-server bufferCount	1-44
Configuring a DNS Forwarder	1-45
Configuring dns-server respTasks	1-46
Adding a DNS Name to a Content Rule	1-46
Removing a DNS Name from a Content Rule	1-47
Displaying CSS DNS Information	1-47
Displaying DNS Server Information	1-48
Displaying DNS Server Configuration Information	1-48
Displaying DNS Server Database Statistics	1-50
Displaying DNS Server Domain Statistics	1-51
Displaying DNS Forwarder Statistics	1-51
Displaying DNS Server Zones	1-53
Displaying DNS Record Information	1-54
Displaying DNS-Record Statistics	1-54
Displaying DNS Record Keepalive Information	1-56
Displaying the DNS Record Weight	1-57
Displaying DNS Peer Information	1-58
Displaying Domain Summary Information	1-59

CHAPTER 2**Configuring the DNS Sticky Feature** 2-1

Overview of DNS Sticky	2-2
DNS Sticky Without a GSDB	2-3
DNS Sticky with a GSDB	2-3
DNS Sticky in a Network Proximity Environment	2-4

- DNS Sticky Quick-Start Procedures **2-5**
 - Configuring DNS Sticky without a GSDB **2-5**
 - Configuring DNS Sticky with a GSDB **2-7**
 - Global Sticky Database Configuration Quick Start **2-7**
 - DNS Server Configuration Quick Start **2-9**
 - Configuring DNS Sticky with Network Proximity **2-11**
 - Global Sticky Database Configuration Quick Start **2-11**
 - DNS Server Configuration Quickstart **2-12**
- Converting Content Rule-Based DNS to Zone-Based DNS **2-13**
- Configuring DNS Sticky Parameters **2-14**
 - Enabling the Global Sticky Database **2-14**
 - Resetting the Global Sticky Database Statistics **2-15**
 - Configuring the Global Sticky Database Interface **2-15**
 - Resetting the Global Sticky Database Interface Statistics **2-16**
 - Configuring the Time to Live for Global Sticky Database Entries **2-17**
 - Configuring Sticky Domain Records **2-17**
 - Configuring Server Zones for DNS Sticky **2-18**
- Displaying DNS Sticky Statistics **2-18**
 - Displaying Global Sticky Database Statistics **2-18**
 - Displaying GSDB Interface Statistics **2-19**
 - Displaying DNS Sticky Domain Record Statistics **2-20**
 - Displaying Domain Load Statistics **2-21**
 - Displaying DNS Record Statistics **2-22**
 - Displaying DNS Record Keepalives **2-22**
 - Displaying Proximity and GSDB Metrics **2-22**
 - Displaying Server Zones for DNS Sticky **2-24**

CHAPTER 3**Configuring a CSS as a Content Routing Agent 3-1**

- Overview of the CRA Feature 3-2
- CRA Quick Start 3-4
- Configuring CRA Parameters 3-5
 - Enabling the CRA 3-5
 - Configuring the CPU Load Threshold 3-5
 - Configuring CRA Domain Records 3-6
 - Configuring an Alias for an Existing Client Domain 3-8
 - Clearing Domain Statistics 3-9
- Displaying CRA Statistics 3-10

CHAPTER 4**Configuring a Client-Side Accelerator 4-1**

- Overview of the Client Side Accelerator Feature 4-2
- Configuration Examples of CSA 4-3
 - Single-POP CSA Configuration 4-3
 - Multiple-POP CSA Configuration 4-5
- CSA Quick Start 4-7
- Configuring CSA Parameters 4-12
 - Enabling the CSA Feature 4-12
 - Configuring the Domain Cache 4-13
 - Configuring a DNS Server Forwarder 4-14
 - Configuring Accelerated Domains 4-15
 - Resetting the DNS Record Statistics 4-16
 - Configuring the CSA DNS Server Zones 4-16
- Displaying CSA Information 4-16
 - Displaying a CSA Configuration 4-16
 - Displaying DNS Server Domain Cache Statistics 4-18
 - Displaying DNS Server Zones 4-18

Displaying DNS Record Keepalive Information 4-18
 Displaying Domain Acceleration Records Statistics 4-19

CHAPTER 5

Configuring Network Proximity 5-1

Entering Your Proximity License Keys 5-2
 Entering the Enhanced Feature Set License Key 5-2
 Entering the Proximity Database License Key 5-3
 Overview of Network Proximity 5-3
 Proximity Database 5-5
 Proximity Domain Name Server 5-6
 Proximity Zones 5-7
 Peer Mesh 5-8
 Example of Network Proximity 5-9
 Network Proximity Configuration Quick Start 5-12
 PDB Configuration Quick Start 5-12
 PDNS Configuration Quick Start 5-13
 Configuring a Proximity Database 5-15
 Configuring APP-UDP and APP 5-16
 Enabling APP-UDP 5-17
 Securing APP-UDP Datagrams 5-17
 Specifying APP-UDP Options 5-18
 Removing an APP-UDP Options Record 5-19
 Specifying the APP-UDP Port 5-19
 Showing APP-UDP Configurations 5-20
 Enabling the PDB 5-21
 Assigning Proximity Metrics 5-22
 Flushing Proximity Assignments 5-23
 Configuring Proximity Time to Live 5-24
 Storing the PDB 5-25
 Retrieving the PDB 5-26

Refining Proximity Metrics	5-27
Using Proximity Reprobe	5-28
Clearing the PDB	5-28
Configuring the Proximity Probe Module	5-29
Configuring the Proximity Probe Module Method	5-29
Specifying the Proximity Probe Module Samples	5-30
Configuring the Proximity Probe Module Metric Weighting	5-30
Configuring the Proximity Probe Module Interval	5-31
Specifying Proximity Probe Module TCP-ports	5-32
Using Network Proximity Tiers	5-33
Proximity Tiers	5-33
Example of Tiered Network Proximity	5-33
Displaying PDB Configurations	5-37
Displaying the PDB	5-37
Displaying Proximity Metrics	5-38
Displaying Proximity Statistics	5-40
Displaying Proximity Refinement	5-41
Displaying Proximity Assignments	5-42
Displaying Proximity Zones	5-43
Displaying Proximity Zone Statistics	5-44
Displaying Proximity Probe Module Statistics	5-45
Configuring a PDNS	5-46
Configuring APP-UDP and APP	5-47
Enabling the PDNS	5-47
Configuring Domain Records	5-49
Disabling the PDNS	5-49
Clearing the DNS Server Statistics	5-50
Enabling the Proximity Lookup Cache	5-50
Removing Entries from the Proximity Lookup Cache	5-51

- Displaying PDNS Configurations 5-52
 - Displaying the Proximity Cache 5-52
 - Displaying DNS Record Statistics 5-54
 - Displaying DNS Record Keepalives 5-54
 - Displaying DNS Server Zones 5-54
 - Displaying DNS Record Proximity 5-54
 - Displaying DNS Server Information 5-55

INDEX



<i>Figure 1-1</i>	Example of GSLB Using Two Zones	1-6
<i>Figure 1-2</i>	Example of GSLB (Static Proximity) Using Two CSSs Configured as Authoritative DNS Servers	1-8
<i>Figure 3-1</i>	Example of Boomerang Content Routing Process - Direct Mode	3-3
<i>Figure 4-1</i>	Example of a Client Side Accelerator Configuration Example	4-4
<i>Figure 4-2</i>	Example of a Client Side Accelerator APP Mesh Configuration	4-6
<i>Figure 5-1</i>	Simplified Example of Network Proximity	5-4
<i>Figure 5-2</i>	Example of Network Proximity Zones	5-8
<i>Figure 5-3</i>	Two-Zone Network Proximity Example	5-10
<i>Figure 5-4</i>	Tiered Network Proximity Configuration	5-35



<i>Table 1-1</i>	Field Descriptions for the show app Command	1-14
<i>Table 1-2</i>	Field Descriptions for the show app session Command	1-14
<i>Table 1-3</i>	Zone-Based DNS Configuration Quick Start	1-17
<i>Table 1-4</i>	Content Rule-Based DNS Configuration Quick Start	1-39
<i>Table 1-5</i>	Field Descriptions for the show dns-server Command	1-48
<i>Table 1-6</i>	Field Descriptions for the show dns-server dbase Command	1-50
<i>Table 1-7</i>	Field Descriptions for the show dns-server stats Command	1-51
<i>Table 1-8</i>	Field Descriptions for the show dns-server forwarder Command	1-52
<i>Table 1-9</i>	Field Descriptions for the show zone Command	1-53
<i>Table 1-10</i>	Field Descriptions for the show dns-record statistics Command	1-55
<i>Table 1-11</i>	Field Descriptions for the show dns-record keepalive Command	1-56
<i>Table 1-12</i>	Field Descriptions for the show dns-record weight Command	1-57
<i>Table 1-13</i>	Field Descriptions for the show dns-peer Command	1-58
<i>Table 1-14</i>	Field Descriptions for the show domain Command	1-59
<i>Table 2-1</i>	DNS Sticky Without a GSDB Configuration Quick Start	2-5
<i>Table 2-2</i>	Global Sticky Database Configuration Quick Start	2-7
<i>Table 2-3</i>	DNS Server Configuration Quick Start	2-9
<i>Table 2-4</i>	Global Sticky Database Configuration Quick Start	2-11
<i>Table 2-5</i>	DNS Server Configuration Quick Start	2-12
<i>Table 2-6</i>	Field Descriptions for the show gsdb Command	2-18
<i>Table 2-7</i>	Field Descriptions for the show gsdb-interface Command	2-19
<i>Table 2-8</i>	Field Descriptions for the show dns-record sticky Command	2-20
<i>Table 2-9</i>	Field Descriptions for the show dns-record load Command	2-21

<i>Table 2-10</i>	Field Descriptions for the show proximity metric Command	2-23
<i>Table 3-1</i>	Content Routing Agent Configuration Quick Start	3-4
<i>Table 3-2</i>	Configuring a Password on a CSS (CRA) Versus a Content Router	3-7
<i>Table 3-3</i>	Field Descriptions for the show dns-boomerang client Command	3-10
<i>Table 4-1</i>	Client Side Accelerator Configuration Quick Start	4-7
<i>Table 4-2</i>	Field Descriptions for the show dns-server accelerate domains Command	4-17
<i>Table 4-3</i>	Field Descriptions for the show dns-server domain-cache Command	4-18
<i>Table 4-4</i>	Field Descriptions for the show dns-record accel Command	4-19
<i>Table 5-1</i>	PDB Configuration Quick Start	5-12
<i>Table 5-2</i>	PDNS Configuration Quick Start	5-13
<i>Table 5-3</i>	Field Descriptions for the show app-udp global Command	5-20
<i>Table 5-4</i>	Field Descriptions for the show app-up secure Command	5-21
<i>Table 5-5</i>	Field Descriptions for the show proximity Command	5-37
<i>Table 5-6</i>	Field Descriptions for the show proximity metric Command	5-39
<i>Table 5-7</i>	Field Descriptions for the show proximity statistics Command	5-40
<i>Table 5-8</i>	Field Descriptions for the show proximity refine Command	5-41
<i>Table 5-9</i>	Field Descriptions for the show proximity assign Command	5-42
<i>Table 5-10</i>	Field Descriptions for the show proximity zone Command	5-43
<i>Table 5-11</i>	Show Proximity Zone Statistics Display Fields	5-44
<i>Table 5-12</i>	Field Descriptions for the show proximity probe rtt statistics Command	5-45
<i>Table 5-13</i>	Show Proximity Cache Display Fields	5-53
<i>Table 5-14</i>	Show Proximity Cache All Display Fields	5-53
<i>Table 5-15</i>	Field Descriptions for the show dns-record proximity Command	5-55



Preface

This guide provides instructions for configuring the global load-balancing features of the Cisco 11500 Series Content Services Switches (CSS). Information in this guide applies to all CSS models except where noted.

The CSS software is available in a Standard or optional Enhanced feature set. The Enhanced feature set contains all of the Standard feature set and also includes Network Address Translation (NAT) Peering, Domain Name Service (DNS), Demand-Based Content Replication (Dynamic Hot Content Overflow), Content Staging and Replication, and Network Proximity DNS. Proximity Database and Secure Management, which includes Secure Shell Host and SSL strong encryption for the Device Management software, are optional features.

This preface contains the following major sections:

- [Audience](#)
- [How to Use This Guide](#)
- [Related Documentation](#)
- [Symbols and Conventions](#)
- [Obtaining Documentation](#)
- [Documentation Feedback](#)
- [Obtaining Technical Assistance](#)
- [Obtaining Additional Publications and Information](#)

Audience

This guide is intended for the following trained and qualified service personnel who are responsible for configuring the CSS:

- Web master
- System administrator
- System operator

How to Use This Guide

This guide is organized as follows:

Chapter	Description
Chapter 1, Configuring the CSS as a Domain Name System Server	Configure the Domain Name Service (DNS) on a CSS to translate domain names into IP addresses.
Chapter 2, Configuring the DNS Sticky Feature	Configure DNS Sticky on a CSS to maintain persistence on a server for e-commerce clients.
Chapter 3, Configuring a CSS as a Content Routing Agent	Configure a CSS as a content routing agent (CRA) to enhance a user's browser experience.
Chapter 4, Configuring a Client-Side Accelerator	Configure a CSS as a Client Side Accelerator (CSA) to accelerate the retrieval of domain content.
Chapter 5, Configuring Network Proximity	Configure Network Proximity on a CSS to improve network performance.

Related Documentation

In addition to this document, the Content Services Switch documentation includes the following publications.

Document Title	Description
<i>Release Note for the Cisco 11500 Series Content Services Switch</i>	This release note provides information on operating considerations, caveats, and command line interface (CLI) commands for the Cisco 11500 series CSS.
<i>Cisco 11500 Series Content Services Switch Hardware Installation Guide</i>	This guide provides information for installing, cabling, and powering the Cisco 11500 series CSS. In addition, this guide provides information about CSS specifications, cable pinouts, and hardware troubleshooting.
<i>Cisco Content Services Switch Getting Started Guide</i>	This guide describes how to perform initial administration and configuration tasks on the CSS, including: <ul style="list-style-type: none">• Booting the CSS for the first time and a routine basis, and logging in to the CSS• Configuring the username and password, Ethernet management port, static IP routes, and the date and time• Configuring DNS server for hostname resolution• Configuring sticky cookies with a sticky overview and advanced load-balancing method using cookies• A task list to help you find information in the CSS documentation• Troubleshooting the boot process

Document Title	Description
<i>Cisco Content Services Switch Administration Guide</i>	<p>This guide describes how to perform administrative tasks on the CSS, including upgrading your CSS software and configuring the following:</p> <ul style="list-style-type: none"> • Logging, including displaying log messages and interpreting sys.log messages • User profile and CSS parameters • SNMP • RMON • XML documents to configure the CSS • CSS scripting language • Offline Diagnostic Monitor (Offline DM) menu
<i>Cisco Content Services Switch Routing and Bridging Configuration Guide</i>	<p>This guide describes how to perform routing and bridging configuration tasks on the CSS, including:</p> <ul style="list-style-type: none"> • Management ports, interfaces, and circuits • Spanning-tree bridging • Address Resolution Protocol (ARP) • Routing Information Protocol (RIP) • Internet Protocol (IP) • Open Shortest Path First (OSPF) protocol • Cisco Discovery Protocol (CDP) • Dynamic Host Configuration Protocol (DHCP) relay agent

Document Title	Description
<i>Cisco Content Services Switch Content Load-Balancing Configuration Guide</i>	This guide describes how to perform CSS content load-balancing configuration tasks, including: <ul style="list-style-type: none"> • Flow and port mapping • Services • Service, global, and script keepalives • Source groups • Loads for services • Dynamic Feedback Protocol (DFP) • Owners • Content rules • Sticky parameters • HTTP header load balancing • Content caching • Content replication
<i>Cisco Content Services Switch Redundancy Configuration Guide</i>	This guide describes how to perform CSS redundancy configuration tasks, including: <ul style="list-style-type: none"> • VIP and virtual interface redundancy • Adaptive session redundancy • Box-to-box redundancy
<i>Cisco Content Services Switch Security Configuration Guide</i>	This guide describes how to perform CSS security configuration tasks, including: <ul style="list-style-type: none"> • Controlling access to the CSS • Secure Shell Daemon protocol • Radius • TACACS+ • Firewall load balancing

Document Title	Description
<i>Cisco Content Services Switch SSL Configuration Guide</i>	This guide describes how to perform CSS SSL configuration tasks, including: <ul style="list-style-type: none"> • SSL certificate and keys • SSL termination • Backend SSL • SSL initiation
<i>Cisco Content Services Switch Command Reference</i>	This reference provides an alphabetical list of all CLI commands including syntax, options, and related commands.
<i>Cisco Content Services Switch Device Management User's Guide</i>	This guide describes how to use the Device Management user interface, an HTML-based Web-based application that you use to configure and manage your CSS.

Symbols and Conventions

This guide uses the following symbols and conventions to identify different types of information.



Caution

A caution means that a specific action you take could cause a loss of data or adversely impact use of the equipment.



Warning

A warning describes an action that could cause you physical harm or damage the equipment.



Note

A note provides important related information, reminders, and recommendations.

Bold text indicates a command in a paragraph.

`Courier` text indicates text that appears on a command line, including the CLI prompt.

Courier bold text indicates commands and text you enter in a command line.

Italics text indicates the first occurrence of a new term, book title, emphasized text, and variables for which you supply values.

1. A numbered list indicates that the order of the list items is important.
 - a. An alphabetical list indicates that the order of the secondary list items is important.
- A bulleted list indicates that the order of the list topics is unimportant.
 - An indented list indicates that the order of the list subtopics is unimportant.

Obtaining Documentation

Cisco documentation and additional literature are available on Cisco.com. Cisco also provides several ways to obtain technical assistance and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation at this URL:

<http://www.cisco.com/univercd/home/home.htm>

You can access the Cisco website at this URL:

<http://www.cisco.com>

You can access international Cisco websites at this URL:

http://www.cisco.com/public/countries_languages.shtml

Ordering Documentation

You can find instructions for ordering documentation at this URL:

http://www.cisco.com/univercd/cc/td/doc/es_inpk/pdi.htm

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Ordering tool:

<http://www.cisco.com/en/US/partner/ordering/index.shtml>

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, USA) at 408 526-7208 or, elsewhere in North America, by calling 800 553-NETS (6387).

Documentation Feedback

You can send comments about technical documentation to bug-doc@cisco.com.

You can submit comments by using the response card (if present) behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

For all customers, partners, resellers, and distributors who hold valid Cisco service contracts, Cisco Technical Support provides 24-hour-a-day, award-winning technical assistance. The Cisco Technical Support Website on Cisco.com features extensive online support resources. In addition, Cisco Technical Assistance Center (TAC) engineers provide telephone support. If you do not hold a valid Cisco service contract, contact your reseller.

Cisco Technical Support Website

The Cisco Technical Support Website provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The website is available 24 hours a day, 365 days a year at this URL:

<http://www.cisco.com/techsupport>

Access to all tools on the Cisco Technical Support Website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register at this URL:

<http://tools.cisco.com/RPF/register/register.do>

Submitting a Service Request

Using the online TAC Service Request Tool is the fastest way to open S3 and S4 service requests. (S3 and S4 service requests are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Service Request Tool automatically provides recommended solutions. If your issue is not resolved using the recommended resources, your service request will be assigned to a Cisco TAC engineer. The TAC Service Request Tool is located at this URL:

<http://www.cisco.com/techsupport/servicerequest>

For S1 or S2 service requests or if you do not have Internet access, contact the Cisco TAC by telephone. (S1 or S2 service requests are those in which your production network is down or severely degraded.) Cisco TAC engineers are assigned immediately to S1 and S2 service requests to help keep your business operations running smoothly.

To open a service request by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411 (Australia: 1 800 805 227)

EMEA: +32 2 704 55 55

USA: 1 800 553 2447

For a complete list of Cisco TAC contacts, go to this URL:

<http://www.cisco.com/techsupport/contacts>

Definitions of Service Request Severity

To ensure that all service requests are reported in a standard format, Cisco has established severity definitions.

Severity 1 (S1)—Your network is “down,” or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Severity 2 (S2)—Operation of an existing network is severely degraded, or significant aspects of your business operation are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Severity 3 (S3)—Operational performance of your network is impaired, but most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Severity 4 (S4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- Cisco Marketplace provides a variety of Cisco books, reference guides, and logo merchandise. Visit Cisco Marketplace, the company store, at this URL:
<http://www.cisco.com/go/marketplace/>
- The Cisco *Product Catalog* describes the networking products offered by Cisco Systems, as well as ordering and customer support services. Access the Cisco Product Catalog at this URL:
<http://cisco.com/univercd/cc/td/doc/pcat/>
- *Cisco Press* publishes a wide range of general networking, training and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to Cisco Press at this URL:
<http://www.ciscopress.com>

- *Packet* magazine is the Cisco Systems technical user magazine for maximizing Internet and networking investments. Each quarter, Packet delivers coverage of the latest industry trends, technology breakthroughs, and Cisco products and solutions, as well as network deployment and troubleshooting tips, configuration examples, customer case studies, certification and training information, and links to scores of in-depth online resources. You can access Packet magazine at this URL:

<http://www.cisco.com/packet>

- *iQ Magazine* is the quarterly publication from Cisco Systems designed to help growing companies learn how they can use technology to increase revenue, streamline their business, and expand services. The publication identifies the challenges facing these companies and the technologies to help solve them, using real-world case studies and business strategies to help readers make sound technology investment decisions. You can access iQ Magazine at this URL:

<http://www.cisco.com/go/iqmagazine>

- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the Internet Protocol Journal at this URL:

<http://www.cisco.com/ipj>

- World-class networking training is available from Cisco. You can view current offerings at this URL:

<http://www.cisco.com/en/US/learning/index.html>



Configuring the CSS as a Domain Name System Server

This chapter describes how to configure a CSS as a Domain Name System (DNS) server to respond to DNS requests by resolving a domain name to an IP address. Information in this chapter applies to all CSS models, except where noted.



Note

The DNS feature requires the CSS Enhanced feature set license.

This chapter provides the following major sections:

- [Overview of the CSS DNS Feature](#)
- [Overview of the CSS Application Peering Protocol](#)
- [Configuring the Application Peering Protocol](#)
- [Displaying APP Configurations](#)
- [Configuring Zone-Based DNS on a CSS](#)
- [Configuring Content Rule-Based DNS on a CSS](#)
- [Displaying CSS DNS Information](#)

Overview of the CSS DNS Feature

The CSS DNS feature enables you to configure one or more CSSs to construct highly available, distributed, and load-sensitive websites. Groups of CSSs may host many distributed websites concurrently. These groups make decisions that can be configured independently for each distributed website using local and remote load-balancing and domain information.

CSSs that are configured together for DNS form a *content domain*. Within the content domain, CSSs are known as peers. You can configure peers to exchange domain records (zone-based DNS only), content rules (content rule-based DNS only), and service availability and load (both zone-based and rule-based DNS).

Each CSS becomes aware of all the locations for the content associated with a domain name and the operational state and load of the location. The CSS can then intelligently direct clients to a site where they can best obtain the desired content. In addition, a CSS never sends a client to a location that is overburdened or out of service.

You can configure a CSS as an authoritative DNS server. An authoritative DNS server does not depend on lower-level name servers for the answer to a DNS request. Instead, the CSS responds directly to the request by sending a locally configured or learned address record (A-record) of the subdomain to the resolver (the requestor of the DNS resolution, which could be a client, the client local DNS server, or another DNS server). A domain is a subdomain of another domain if it is contained within that other domain. For example, `www.cisco.com` is a subdomain of `cisco.com`.

You can also configure a CSS to query lower-level name servers to help resolve a DNS request. You accomplish this by configuring name server records (NS-records), which point to the lower-level DNS servers, on the CSS.

For example, when a user clicks a URL on a Web page:

1. The client asks the locally configured DNS server for a translation of a domain name to an IP address.
2. The local DNS server learns the interface address of the CSS through normal DNS processing.
3. The local client DNS server requests address resolution from the CSS authoritative DNS server.

4. The CSS authoritative DNS server returns the VIP address of the best location (based on server availability and load) where the client can retrieve the content.
5. The local DNS server responds to the client with the VIP.
6. The client uses the VIP to access the content.

**Note**

The CSS implementation of DNS server functionality is a streamlined, endnode-only approach. The CSS does not support zone transfer among other DNS servers. However, each CSS configured in a content domain can act as the authoritative DNS server for the subdomain.

You can configure DNS on a CSS in either of the following two ways:

- Zone-based DNS
- Content rule-based DNS

Zone-Based DNS

Zone-based DNS creates content domains in geographical areas or zones using the **dns-server zone** command. For example, each zone may be a country, a state, or a city located in different parts of the world. Each CSS peer acts as an authoritative DNS server for the subdomain or subdomains that it represents in a zone. A CSS uses its locally configured or learned A-records or name server records (NS-records) to resolve DNS requests for configured subdomains. This is the recommended method of configuring global server load balancing (GSLB) on a CSS.

Like content rule-based DNS (see the next section), zone-based DNS uses the Application Peering Protocol (APP) for communication and information exchange between CSS peers. However, in this case, the CSS peers exchange subdomain address records (A-records) or name-server records (NS-records), with load and status information.

When a CSS receives a DNS request from a client local DNS server, the CSS attempts to resolve the domain name based on the local domains that are directly connected to it or its APP peers. If the CSS can resolve the domain name locally, it uses its A-records that were locally configured or learned through APP to resolve the domain name to an IP address. The CSS then responds with the IP address associated with the domain name to the client local DNS server.

If the CSS cannot resolve the domain name using its A-records, it forwards the request to a lower-level name server (may be a CSS) using its locally configured and learned NS-records. The lower-level DNS server returns an A-record to the CSS, which uses the A-record to resolve the domain name and then forwards the IP address to the local DNS server.

For details about configuring zone-based DNS, see to the [“Configuring Zone-Based DNS on a CSS”](#) section.

Content Rule-Based DNS

Content rule-based DNS uses domain names configured in content rules to resolve DNS requests. Like zone-based DNS, rule-based DNS uses APP for communication and information exchange between CSS peers. Unlike zone-based CSS peers, rule-based DNS peers exchange owner names and content rules according to their configured exchange policies as well as service load and status information.

For details about configuring rule-based DNS, see to the [“Configuring Content Rule-Based DNS on a CSS”](#) section.

Overview of the CSS Application Peering Protocol

CSSs configured within the same zone or content domain initiate communication using Application Peering Protocol (APP) sessions with their peers upon system bootup or when peers first become connected through an APP session. Thereafter, changes in local configurations are relayed to the peers automatically as they occur. When the APP session is up, the peers exchange service load and domain information. APP provides a guaranteed and private communications channel for this exchange. APP is used by both zone-based and rule-based CSS DNS servers.

APP and Zone-Based DNS

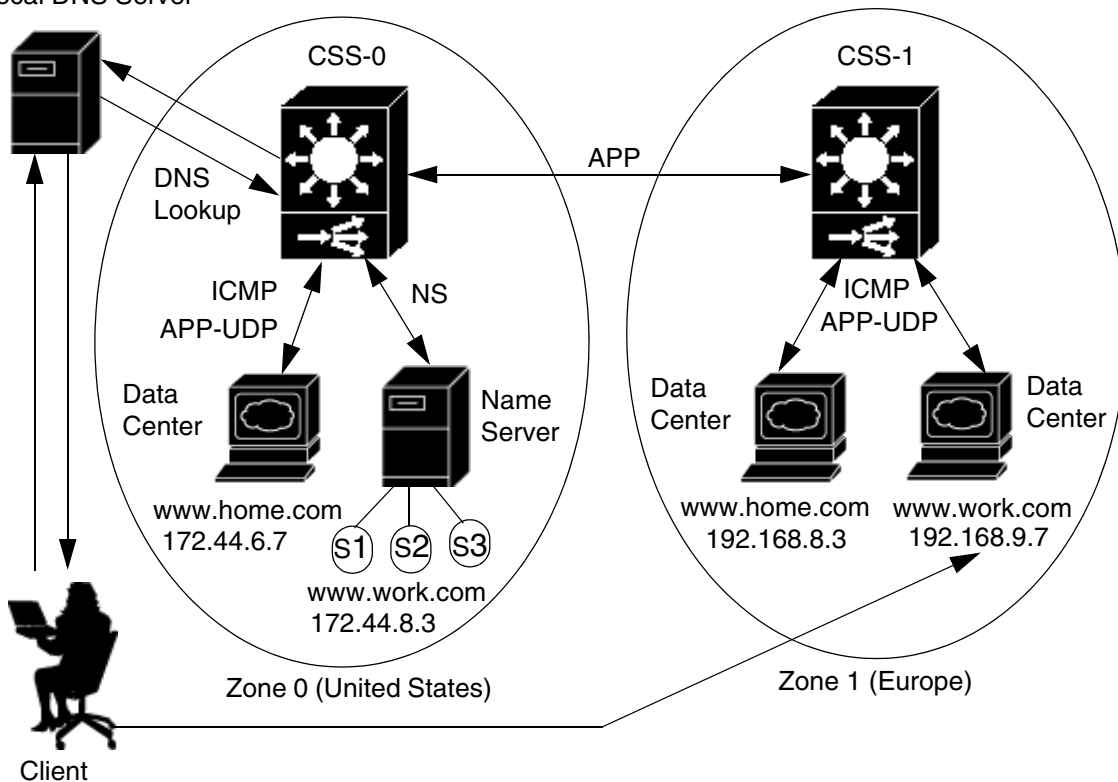
In addition to exchanging service load and status using APP, CSS peers in a zone-based DNS configuration exchange A-records for directly connected domains and NS-records for lower-level data centers. This domain record exchange allows remote peers to learn about other peer CSS domains.

Figure 1-1 illustrates how zone-based DNS operates with two CSS peers in different zones. For example, when a client requests *www.work.com*:

1. The client browser asks the locally configured DNS server for a translation to an IP address.
2. The DNS server round-robins an address resolution request to one of the CSSs.
3. The selected CSS authoritative DNS server (in this example, CSS-0) determines server availability based on the DNS balance type configured for the zone or the domain record as follows:
 - If CSS-0 determines that the best selection is a name server (NS) record, the CSS begins a recursive query of the name server to determine an authoritative response.
 - If CSS-0 finds that the best selection is an address record (A-record), it formulates an authoritative response immediately. In this example, CSS-0 decides that the best selection is an A-record (learned through the peer mesh with CSS-1) for a data center in Zone 1.
4. CSS-0 sends an authoritative response that contains the resolved IP address of *www.work.com* to the client local DNS server.
5. The local DNS server notifies the client that sufficient domain name resolution information is available to establish a data connection to *www.work.com*.
6. Lastly, the client uses the local DNS server response information (IP address) to connect to a service and starts receiving content. In this example, the best service is located in Zone 1 at IP address 192.168.9.7.

Figure 1-1 Example of GSLB Using Two Zones

Local DNS Server



APP and Rule-Based DNS

For rule-based DNS, CSS peers exchange owner names according to the DNS exchange policies configured for each owner. For each owner that a CSS is configured to share with its peers, the CSS sends the locally configured content rules and DNS name information. Upon receiving a peer's content rule information, the CSS compares each DNS name and content rule to its local configuration.

Content rules that:

- Match a locally configured content rule cause a *dynamic service* to be added automatically to the local content rule. The local content rule points to the peer for an alternate location for the content.
- Do not have a corresponding local entry cause the CSS to automatically create a *dynamic content rule* containing a dynamic service that points to the peer that has the content rule configured.

The determination of whether or not a content rule matches is based strictly on content rule name. Peers having matching content rule names must have exact copies of rule definitions with the exception of VIP addresses. DNS names do not need to be identical.

**Note**

CSSs do not include dynamic services or dynamic content rules in their running-config or startup-config files. Dynamic services and dynamic content rules are temporary and are removed when the peer connection terminates.

For example, when a client requests *www.arrowpoint.com*:

1. The client browser asks the locally configured DNS server for a translation to an IP address.
2. The DNS server round-robins an address resolution request to one of the CSSs.
3. The selected CSS authoritative DNS server determines server availability based on the DNS balance type.

If the CSS is configured as DNS balance type **dnsbalance preferlocal** and is:

- Able to locally handle the request for this DNS name, it returns the local VIP to the DNS server.
- Not able to handle the request for this DNS name (the server has reached a defined load threshold or is unavailable), the CSS returns the dynamic content rule VIP to the DNS server.

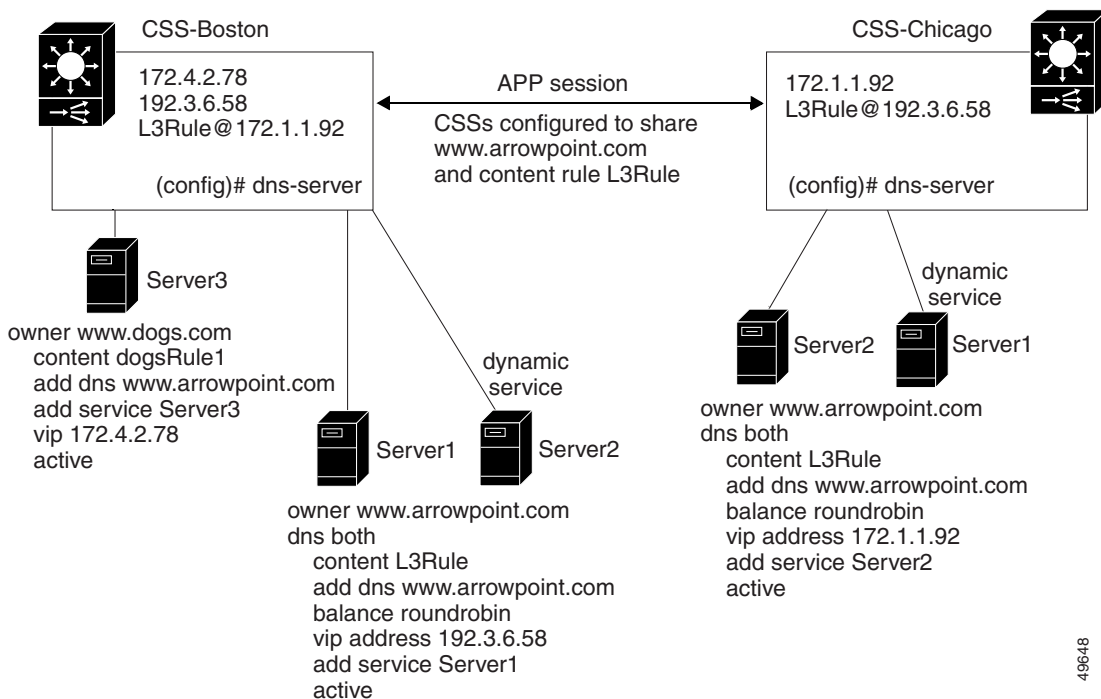
If the CSS is configured as DNS balance type **dnsbalance roundrobin**, the CSS resolves requests by evenly distributing the load to resolve domain names among local and remote content domain sites.

For information on configuring DNS balance types, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

4. The DNS server forwards the resolved VIP to the client.
5. The client uses the VIP to access the content.

Figure 1-2 illustrates two peer CSSs configured as authoritative DNS servers using rule-based DNS. Each CSS knows its local content rule VIPs and dynamic content rule VIPs. The @ sign within a content rule VIP indicates a dynamic content rule. Owner *www.arrowpoint.com* is configured for **dns both** (push and accept owner *www.arrowpoint.com* and its content rule *L3Rule*). Even though CSS-Boston contains owners *www.arrowpoint.com* and *www.dogs.com*, only owner *www.arrowpoint.com* and content rule *L3Rule* are shared between the CSSs.

Figure 1-2 Example of GSLB (Static Proximity) Using Two CSSs Configured as Authoritative DNS Servers



49648

Configuring the Application Peering Protocol

The Application Peering Protocol feature is part of the CSS Enhanced feature set. You need to configure APP prior to configuring the DNS server for both zone-based and rule-based DNS.

To configure APP, use the **app** command. The options for this global configuration mode command are:

- **app** - Enables all APP sessions
- **app framesz** - Sets the maximum frame size allowed by APP
- **app port** - Sets the TCP port that listens for APP connections
- **app session** - Creates an APP session

Enabling APP

To enable APP, use the **app** command on each CSS peer. For example:

```
(config)# app
```

To disable all APP sessions, enter:

```
(config)# no app
```

Configuring the APP Frame Size

To set the maximum size allowed by the APP, use the **app framesz** command. Enter the maximum APP frame size from 10240 to 65535. The default is 10240. Upon session establishment, peers select the smallest configured frame size to use for session communication. For example, CSS-A is configured for frame size 15000 and CSS-B is configured for frame size 20000. Once the session is established, CSS-B will use frame size 15000.

For example:

```
(config)# app framesz 15000
```

To restore the default frame size to 10240, enter:

```
(config)# no app framesz
```

Configuring the APP Port

To set the TCP port number, use the **app port** command. This port listens for APP connections. Enter a port number from 1025 to 65535. The default TCP port is 5001.

For example:

```
(config)# app port 1500
```

To restore the default port number to 5001, enter:

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



Note

APP frame failures occur in some networks as the result of routing configurations. Suppose the local CSS peer (CSS A) sends an APP frame to a remote CSS peer (CSS B) on a particular interface (B1). CSS B responds from an interface (B2) whose IP address is different from the original destination address. CSS A drops the APP response frame causing APP to fail. This issue is compounded when you use equal-cost multi-path (ECMP) routing. In this case, configure a static route for the APP session.

Configuring an APP Session

To create a content domain between two or more CSSs, the CSSs use APP sessions. An APP session allows the CSSs to share the same content rule, load, and domain information, which are used to determine the best site to resolve a DNS request. To create an APP session between two CSSs, use the **app session** command.

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

```
app session ip_address {keepalive frequency {authChallenge|authNone  
session_secret {encryptMd5hash|encryptNone  
{rcmdEnable|rcmdDisable}}}}
```

**Note**

The **authChallengeauthNone** and **encryptMd5hashencryptNone** APP command options must be identical for both CSSs in an APP session or the session will not come up. The **keepalive** and **rcmd** command options do not have to be identical between CSS peers.

The variables and options are:

- *ip_address* - IP address for the peer CSS.

**Note**

Do not configure an APP session peer with a local CSS IP address (for example, a circuit IP address or Management port IP address). If you do, the following error message appears: `Illegal IP address for APP.`

- *keepalive frequency* - Optional time in seconds between sending keepalive messages to this peer CSS. Enter an integer from 14 to 255. The default is 14.
- **authChallengeauthNone** - Optional authentication method for the session. Enter either **authChallenge** for Challenge Handshake Authentication Protocol (CHAP) method or **authNone** for no authentication method. The default is no authentication.
- *session_secret* - Secret used with **authChallenge** to authenticate a peer or used with **encryptMd5hash** to provide an MD5hash encryption scheme for the session. Enter an unquoted text string with a maximum of 32 characters and no spaces.
- **encryptMd5hashencryptNone** - Optional encryption method for the packets. Enter either **encryptMd5hash** for MD5 base hashing method or **encryptNone** for no encryption method. The default is no encryption.
- **rcmdEnable!rcmdDisable** - Optional setting for sending remote CLI commands to the peer through the **rcmd** command. Enter either **rcmdEnable** to allow the sending of CLI commands or **rcmdDisable** to disallow the sending of CLI commands. The default setting is enabled.

To terminate an APP session, enter the **no app session** command and an IP address:

```
(config)# no app session 192.2.2.2
```

For example, to configure a CSS in Boston (IP address 172.1.1.1) to be a peer of a CSS in Chicago (IP address 192.2.2.2), use the **app** command to configure:

```
CSS-Boston(config)# app session 192.2.2.2
CSS-Chicago(config)# app session 172.1.1.1
```

Using the rcmd Command

To issue CLI commands (including playing scripts) to remote CSS peers over an APP session, use the **rcmd** command. Before you use this command, ensure that:

- Circuit addresses are set up on both the local and the remote CSSs
- Routing is set up properly on both CSSs
- APP is configured on both CSSs (see the [“Configuring the Application Peering Protocol”](#) section)

The **rcmd** command is available in SuperUser mode.

The syntax for this command is:

```
rcmd ip_address or host “CLI command {;CLI command...}”
      {timeout_response}
```

The variables are:

- *ip_address* or *host* - The IP address or host name for the peer.
- *CLI command* - One or more CLI commands you want to issue to the peer. Enter the command, its options, and variables exactly. Enclose the command text string in quotes (“”). If the CLI command itself requires quotes, use single quotes around that part of the command string. For example:

```
rcmd 192.168.12.23 "script play test `argument1 argument2`"
```

When entering multiple CLI commands, insert a semicolon (;) character to separate each command.



Note You cannot issue **grep**, **grep** within a script command, or **redirect** commands.

- *timeout_response* - The optional amount of time, in seconds, to wait for the output command response from the peer. Enter an integer from 3 to 300 (5 minutes). The default is 3 seconds.

For example:

```
# rcmd 192.2.2.2 "show domain" 10
```

Displaying APP Configurations

To display the APP configuration or session information, use the **show app** command. APP is the method in which you configure private communications links between CSSs in the same content domain. A content domain consists of two or more CSSs configured to exchange content information.

The syntax and options for this command are:

- **show app** - Displays whether APP is enabled, its port number, and its frame size setting. For example:

```
(config)# show app
```

- **show app session** - Displays all IP session information including the session ID, IP address, and state. For example:

```
(config)# show app session
```

- **show app session ip_address** - Displays the IP session information including the session ID, IP address, and state. For example:

```
(config)# show app session 192.168.10.10
```

- **show app session verbose** - Displays the IP session information. In addition, the **verbose** keyword displays detailed information about the IP configuration parameters for the session, including the local address, keepalive frequency, authorization and encryption type, frame size, and packet activity. For example:

```
(config)# show app session verbose
```

- **show app session ip_address verbose** - Displays the same information as the **show app session verbose** command except that it displays information only for the specified IP address. For example:

```
(config)# show app session 192.168.10.10 verbose
```

To display a list of IP addresses, enter **show app session ?**.

Table 1-1 describes the fields in the **show app** output.

Table 1-1 Field Descriptions for the show app Command

Field	Description
Enabled or Disabled	All APP sessions are either enabled or disabled.
PortNumber	The TCP port number that listens for APP connections. The port can be a number from 1 to 65535. The default is 5001.
MaxFrameSize	The maximum frame size allowed on an APP channel between CSSs. The maximum frame size is a number from 10240 to 65535. The default is 10240.

Table 1-2 describes the fields in the **show app session** output.

Table 1-2 Field Descriptions for the show app session Command

Field	Description
App Session Information	DNS-resolved host name as defined through the host command.
Session ID	The unique identifier for the session.
IP Address	The IP address for the peer CSS.

Table 1-2 Field Descriptions for the *show app session* Command (continued)

Field	Description
State	<p>The current state of the session. The possible states include:</p> <ul style="list-style-type: none"> • APP_SESSION_STOP - Indicates that the session is about to be deleted • APP_SESSION_INIT - Indicates that the session is initializing • APP_SESSION_OPEN - Indicates that the connection to the peer has been made • APP_SESSION_AUTH - Indicates that authentication is occurring • APP_SESSION_UP - Indicates that the session is up • APP_SESSION_DOWN - Indicates that the session is down
Local Address	The local interface address. If the session is down, no address is displayed.
rcmdEnable	The setting for the sending of remote CLI commands to the peer through the rcmd command. The Enabled setting allows the sending of CLI commands. The Disabled setting disallows the sending of CLI commands. The default setting is enabled.
KalFreq	The time in seconds between sending keepalive messages to this peer CSS. The time can be from 14 to 255 seconds (15 minutes). The default is 14.
Auth Type	The authentication method for the session. The method is either authChallenge for Challenge Handshake Authentication Protocol (CHAP) method or none for no authentication method. The default is no authentication.
Encrypt Type	The encryption method for the packets. The method is either encryptMd5hash for MD5 base hashing method or none for no encryption method. The default is no encryption.

Table 1-2 Field Descriptions for the `show app session` Command (continued)

Field	Description
MaxFrameSz	The maximum frame size allowed on an APP channel between CSSs. The frame size is a number from 10240 to 65535. The default is 10240.
Pkts Tx	The number of packets sent during the session.
Pkts Rx	The number of packets received during the session.
Pkts Rej	The number of packets rejected during the session.
Last UP event	The day and time of the most recent UP event.
Last DOWN event	The day and time of the most recent DOWN event.
FSM Events	Finite State Machine events as related to the state field.
STOP	The number of APP_SESSION_STOP events. This field is always zero.
INIT	The number of APP_SESSION_INIT events.
OPEN	The number of APP_SESSION_OPEN events.
AUTH	The number of APP_SESSION_AUTH events.
UP	The number of APP_SESSION_UP events.
DOWN	The number of APP_SESSION_DOWN events.
Attached Apl	The application identifier.

Configuring Zone-Based DNS on a CSS

Zone-based DNS is the recommended method for configuring global server load balancing (GSLB) on a CSS. GSLB refers to a configuration where two or more geographically distributed CSSs represent one or more domains. Each CSS:

- Acts as an authoritative DNS server for the domain it represents
- Shares domain records and load information with other CSS peers using APP

Each zone is represented by one CSS or a redundant pair of CSSs in the peer mesh. Additional CSSs can exist in the same zone as lower-level DNS servers or content servers. A zone can be a geographic area such as a country, a region, or a city. On each CSS, you configure domain records that the CSS uses to resolve DNS requests. These records can be one of the following types:

- 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).

This section contains the following topics:

- [Zone-Based DNS Quick Start](#)
- [Configuring a DNS Server](#)
- [Configuring Domain Records](#)
- [Configuring DNS Records with a Zero Weight](#)

Zone-Based DNS Quick Start

[Table 1-3](#) provides a quick overview of the steps required to configure zone-based DNS. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI command, see the sections following [Table 1-3](#).

Table 1-3 Zone-Based DNS Configuration Quick Start

Task and Command Example

1. Enter config mode.

```
# config
(config)#
```

2. Enable the Application Peering Protocol-User Datagram Protocol (APP-UDP) only if you intend to use the DNS record keepalive type of **kal-ap** or **kal-ap-vip**. See the [“Configuring Domain Records”](#) section.

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

Table 1-3 Zone-Based DNS Configuration Quick Start (continued)

Task and Command Example
<p>3. Enable APP to allow the local CSS to communicate with remote CSS peers. See the “Configuring the Application Peering Protocol” section.</p> <pre>(config)# app</pre>
<p>4. Configure the local CSS zone. Specify a zone number, tier level, optional text description, and optional default load-balancing method and weight. See the “Configuring DNS Server Zones” section.</p> <pre>(config)# dns-server zone 0 tier1 "usa" weightedrr</pre>
<p>5. Configure an APP session with the remote CSS that is participating in the peer mesh with the local CSS. The IP address you enter is a local interface address on the remote CSS. See the “Configuring the Application Peering Protocol” section.</p> <pre>(config)# app session 172.16.2.5</pre>
<p>6. Create A-records for domains in the local zone. 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. See the “Configuring Domain Records” section.</p> <pre>(config)# dns-record a www.home.com 192.168.6.7 0 single kal-ap</pre>
<p>7. Create NS-records for domains on other DNS servers within the local 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. See the “Configuring Domain Records” section.</p> <pre>(config)# dns-record ns www.work.com 172.16.6.8 0 single kal-ap</pre>
<p>8. (Optional) 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 <i>Cisco Content Services Switch Content Load-Balancing Configuration Guide</i>.</p>
<p>9. Configure the CSS to act as a DNS server. See the “Enabling a DNS Server” section.</p> <pre>(config)# dns-server</pre>
<p>10. (Optional) Verify the configuration using the show commands described in the “Displaying CSS DNS Information” section.</p>

The following running-config example shows the results of entering the commands described in [Table 1-3](#).

```
! ***** GLOBAL *****  
  
app-udp  
  
dns-server zone 0 tier1 "usa" weightedrr  
dns-record a www.home.com 192.168.6.7 0 single kal-ap  
dns-record ns www.work.com 172.16.6.8 0 single kal-ap  
dns-server  
  
app  
app session 172.16.2.5
```

Configuring a DNS Server

As a DNS server, a CSS resolves domain names to IP addresses when it receives DNS requests from clients. To configure a CSS as a DNS server, use the **dns-server** command and its options. The options for this global configuration mode command are:

- **dns-server bufferCount** - Modifies the DNS response buffer count.
- **dns-server forwarder** - Enables a DNS forwarder (a CSS or a fully functional BIND server), which resolves DNS requests that a CSS cannot resolve.
- **dns-server respTasks** - Modifies the DNS responder task count.
- **dns-server zone** - Enables zone-based DNS in a non-proximity configuration or enables a Proximity Domain Name Server (PDNS) in a Network Proximity configuration. For details about Network Proximity, see [Chapter 5, Configuring Network Proximity](#).
- **dns-server zone load** - Specifies how the CSS handles the least-loaded load-balancing method.



Note

You need to configure APP before you configure the DNS server as shown in [Table 1-3](#).

Enabling a DNS Server

To enable a CSS to resolve domain names to IP addresses, use the **dns-server** command.

**Note**

The **dns-server** command is part of the CSS Enhanced feature set and requires a separate license key.

For example:

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

To disable DNS server functionality on a CSS, enter:

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

Configuring dns-server bufferCount

To change the DNS response buffer count on the CSS, use the **dns-server bufferCount** command. Enter the number of buffers allocated for query response from 2 to 1000. The default is 50.

Use this command with the **show dns-server** command (see [Table 1-5](#)) to tune the CSS only if the CSS experiences buffer depletion during normal operation. If the number of available name server buffers (NS Buffers) in the Current Free Count field drops below 2, use the **dns-server bufferCount** command to increase the buffer count. You can also use the Minimum Free Count (a low-water mark indicator) and the Reclaimed Buffer Count fields as indications of buffer depletion. When the supply of available buffers is depleted, the CSS reclaims used buffers.

For example:

```
(config)# dns-server bufferCount 100
```

To set the DNS response buffer count to its default value of 50, enter:

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

Configuring a DNS Forwarder

If the CSS cannot resolve a DNS request, it sends the request to another DNS server to obtain a suitable response. This server, called a DNS forwarder, can be any DNS server or a CSS configured for DNS. The CSS sends to the forwarder DNS requests that:

- Are not resolvable by the CSS
- Contain an unsupported request or record type

The forwarder resolves the DNS requests and sends DNS responses to the client transparently through the CSS. To monitor forwarder health, a keepalive mechanism (internal to the CSS) sends queries periodically to the forwarder to validate its state.



Note

You must configure at least one local DNS server zone before configuring a DNS forwarder. For details on DNS server zones, see the [“Configuring DNS Server Zones”](#) section earlier in this chapter.

To configure a DNS forwarder on a CSS, use the **dns-server forwarder** command. For Client Side Accelerator (CSA) configurations, the forwarder must be a full-featured DNS server. For details on CSA, see [Chapter 4, Configuring a Client-Side Accelerator](#).

The syntax for this global configuration mode command is:

```
dns-server forwarder [primary ip_address | secondary ip_address | zero]
```

The variables and options are:

- **primary** - Specifies a DNS server as the first choice forwarder. The CSS sends unresolvable requests to the primary forwarder unless it is unavailable, in which case, it uses the secondary forwarder. When the primary forwarder is available again, the CSS resumes sending requests to the primary forwarder.
- **secondary** - Specifies a DNS server as the second choice forwarder.
- *ip_address* - Specifies the IP address of the forwarder. Enter the address in dotted-decimal notation (for example, 192.168.11.1).
- **zero** - Resets the statistics of both forwarders on a CSS.

For example:

```
(config)# dns-server forwarder primary 192.168.11.1 secondary  
192.168.11.2
```

To delete the primary forwarder on a CSS, enter:

```
(config)# no dns-server forwarder primary
```

Configuring the Number of DNS Responder Tasks

The responder task is the part of the CSS DNS server that responds to DNS requests from clients. Typically, the default number of 2 responder tasks is sufficient to handle the volume of DNS requests received by the CSS. To change the DNS responder task count, use the **dns-server respTasks** command. Enter the number of tasks to handle DNS responses as an integer from 1 to 250. The default is 2.

For example:

```
(config)# dns-server respTasks 3
```

To set the DNS responder task count to its default value of 2, enter:

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

Configuring DNS Server Zones

To enable zone-based DNS on a CSS in a global server load-balancing (GSLB) environment, use the **dns-server zone** command and its options. In a Network Proximity configuration, use this command to enable a PDNS. For more information on Network Proximity, see [Chapter 5, Configuring Network Proximity](#).



Note

Before you enable a Proximity Domain Name Server (PDNS) or Zone-based DNS, you must configure APP. If you are using Network Proximity or the **kal-ap** keepalive type, you must also configure APP-UDP. For details on configuring APP, see the “[Configuring the Application Peering Protocol](#)” section earlier in this chapter. For details on configuring APP-UDP, see the “[Configuring APP-UDP and APP](#)” section in [Chapter 5, Configuring Network Proximity](#).”

The syntax for this global configuration mode command is:

```
dns-server zone zone_index {tier1|tier2 {"description"  
  {weightedrr|srcip|leastloaded|preferlocal|roundrobin|ip_address  
  {weightedrr|srcip|leastloaded|preferlocal|roundrobin} {weight}}}}
```

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

- *zone_index* - The numerical identifier of the DNS server zone. The *zone_index* value must be a unique zone number on the network. In a Network Proximity configuration, this number must match the zone index configured on the Proximity Database (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** - The optional maximum number of zones (peers) that may participate in the CSS peer mesh. The tier you select must be the same as the tier for the other CSSs participating in the peer mesh. Enter **tier1** for a maximum of 6 zones. Enter **tier2** for a maximum of 16 zones. The default is tier1.
- *description* - Optional quoted text description of the DNS server zone. Enter a quoted text string with a maximum of 20 characters.
- **weightedrr**|**srcip**|**leastloaded**|**preferlocal**|**roundrobin** - The optional default load-balancing method that the DNS server uses to select returned records when a PDB is unavailable or not configured. The load-balancing method you select here can be overridden by the load-balancing method that you assign to individual DNS records.
 - **weightedrr** - Use this load-balancing method with the *weight* variable to define the default weight applied to all DNS records in the local zone. The CSS gives a zone priority over other zones in a peer mesh according to the assigned domain weights. Each CSS in a mesh maintains an internal list of zones ordered from highest to lowest according to weight. The heaviest zone (the zone with the highest *weight* value for a particular domain) receives DNS requests until it reaches its maximum number of requests, then the next heaviest zone receives DNS requests until it reaches its maximum, and so on. When all the zones have reached their maximum number of requests, the CSS resets the counters and the cycle starts over again.

When you add a new DNS zone, each CSS adds the new zone to its list by weight. In this case, the CSSs do not reset their hit counters. This process prevents flooding of the heaviest zone every time you add or remove a zone.

For example, a zone with a domain that has a weight of 10 receives twice as many hits as a zone with the same domain configured with a weight of 5. Use the **dns-record** command to assign domain weights. See the “Configuring Domain Records” section later in this chapter.

- **srcip** - The CSS uses a source IP address hash to select the zone index to return to the client.
- **leastloaded** - The CSS reports loads and selects a record from the zone that has the least traffic.
- **preferlocal** - The CSS returns a record from the local zone whenever possible, using roundrobin when it is not possible.
- **roundrobin** - The CSS cycles among records available at the different zones. This load-balancing method is the default.
- *ip_address* - The IP address of the PDB. In a proximity configuration, enter the address in dotted-decimal notation (for example, 172.16.2.2). If you choose the zone capabilities (peer mesh) of a DNS server in a non-proximity environment, do not use this variable.
- *weight* - If you do not configure a weight for individual records using the **dns-record** command, then use this variable with the **weightedrr** load-balancing option to define the default weight applied to all DNS records in the local zone. Enter an integer from 0 to 10. The default is 1. To display the weight that you configured on a record using either the **dns-server zone** command or the **dns-record** command, enter the **show dns-record weight** command.

For example:

```
(config)# dns-server zone 0 tier1 "pdns-usa" weightedrr 5
```

To disable the local DNS zone, enter:

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

**Note**

If you need to modify a **dns-server zone** value, you must first disable the DNS server using the **no dns-server** command and then remove the zone using the **no dns-server zone** command. Restore the DNS server zone with the value change, and then reenables the DNS server.

Configuring dns-server zone load

Use the **dns-server zone load** command to configure how the CSS handles the least-loaded balance method.

The syntax for this global configuration mode command is:

```
dns-server zone load [reporting|frequency seconds|variance number]
```

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

- **reporting** - Enables the processing of local DNS server zone load information and the sharing of it with peers. The default is enabled.
- **frequency** - Specifies the period of time between the processing of local DNS server load information and the subsequent delivery of load information to peers.
- **seconds** - Specifies the frequency time (in seconds). Enter an integer between 5 and 300 seconds (5 minutes). The default is 30 seconds.
- **variance** - Specifies the range of load numbers between zones that are considered similar for the least-loaded algorithm. If the load numbers of all zones are within the specified range, the CSS uses minimum response times to identify the least-loaded site.

**Note**

For GSLB, we recommend that you set the same load variance value on all CSSs in a peer mesh. If you configure the absolute load calculation method, we recommend that you configure a load variance of 0. For information about the absolute load calculation method, refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

- **number** - Specifies the variance value. Enter an integer between 0 and 255. The default is 255.

For example, to set the DNS server zone load frequency to 120, enter:

```
(config)# dns-server zone load frequency 120
```

To disable DNS server zone load reporting, enter:

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


Note

To enable or disable **dns-server zone load reporting**, you must first disable the DNS server using the **no dns-server** command. Then issue the **dns-server zone load reporting** or the **no dns-server zone load reporting** command.

Configuring Domain Records

Peer CSS DNS servers that participate in a zone mesh share domain record information using APP (see the “[Configuring the Application Peering Protocol](#)” section earlier in this chapter). The DNS servers use the resulting database of domain names and their zone index information to make zone-based DNS decisions.

The CSS uses the following types of domain records to map a domain name to an IP address or to another DNS server, or to accelerate a domain:

- address record (A-record) - Domain record mapped to an IP address.
- name server record (NS-record) - Domain record mapped to a DNS server IP address.
- accelerated record - Domain record associated with a Client Side Accelerator. See [Chapter 4, Configuring a Client-Side Accelerator](#).

To create an address record (A-record) on a CSS that maps the domain name to an IP address or to create a name server record (NS-record) that maps the domain name to the IP address of a lower-level DNS server, use the **dns-record** command. Use the **no** form of this command to delete an A-record or an NS-record. This command is not available on a CSS configured as a PDB.

The syntax for this global configuration mode command is:

```
dns-record a dns_name ip_address {ttl_value {single|multiple
{ kal-ap-vip|kal-ap|kal-icmp|kal-none {ip_address2 {threshold
{sticky-enabled|sticky-disabled
{usedefault|weightedrr|scip|leastloaded|preferlocal|roundrobin|
proximity {weight}}}}}}}
```

```

dns-record ns dns_name ip_address {ttl_value {single|multiple
{kal-ap-vip|kal-ap|kal-icmpl|kal-none {ip_address2 {threshold
{default|forwarder {sticky-enabled|sticky-disabled
{usedefault|weightedrr|srcip|leastloaded|preferlocal|roundrobin|
proximity {weight}}}}}}}
```

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

- **a** - Specifies that the CSS generates the DNS response based on local information. This option creates an A-record on the CSS that maps the domain name to an IP address.
- **ns** - Specifies that the CSS passes the request to another DNS server whose IP address is contained in a NS-record. This option creates a name server record (NS-record) on a CSS that maps the domain name to the IP address of a lower-level DNS server. The CSS exhibits a pass-through behavior with respect to NS-records. That is, the CSS that receives the original DNS request queries the device specified by the IP address in the NS-record.
- *dns_name* - The fully qualified domain name mapped to the address record. Enter the name as a lowercase unquoted text string with no spaces and a maximum length of 63 characters.
- *ip_address* - IP address bound to the domain name within the DNS server zone. Enter the address in dotted-decimal notation (for example, 172.16.6.7).
- *ttl_value* - The optional Time to Live (TTL) value in seconds. This value determines how long the DNS client remembers the IP address response to the query. Enter a value between 0 to 65535. The default is 0.
- **single|multiple** - Optional number of records to return in a DNS response message. By default, the DNS server returns a single A-record. Specifying **single** returns one A-record. Specifying **multiple** returns two A-records.
- **kal-ap-vip** - Optional Cisco CSS keepalive message type keyword used by a CSS client to request load information for the VIP specified in the *ip_address* value from the CSS agent specified in the *ip_address2* value. Use this option to allow a CSS client to query a local or remote CSS agent for load information for a VIP configured on one or more content rules. The keepalive frequency is set to 6 seconds and is not configurable. For details on configuring **kal-ap-vip**, see the “[Configuring kal-ap-vip](#)” section.



Note To generate **kal-ap** or **kal-ap-vip** keepalive messages to query agents for load information, CSSs acting as clients must be running the Enhanced feature set. Lower-level CSSs acting as **kal-ap** or **kal-ap-vip** agents (data centers or DNS servers) do not require the Enhanced feature set. When the Proximity Domain Name Server (PDNS) is directly attached to a server farm, an internal keepalive is used.

- **kal-ap** - Optional keepalive message type keyword that specifies the CSS keepalive message. Use this option to obtain load information from remote as well as local services based on domains configured on a single content rule. If you configure **kal-ap** on a CSS acting as a client, you must also configure the **add dns** command in a content rule with the appropriate domain names on the CSS acting as an agent. The agent responds with the load information for the configured domain names. The keepalive frequency is set to 6 seconds and is not configurable. For details about the **add dns** command, see the [“Adding a DNS Name to a Content Rule”](#) section.
- **kal-icmp** (Default keepalive) - The optional keepalive message type keyword that specifies ICMP echo (ping). The keepalive frequency is set to 6 seconds and is not configurable.
- **kal-none** - The optional keepalive message type keyword that specifies no keepalive messaging.

For example:

```
(config)# dns-record a www.home.com 172.16.6.7 15 single kal-icmp
```

- *ip_address2* - IP address of the local interface receiving CSS keepalive messages. If you omit this address while the keepalive type is specified, the CSS uses the DNS IP address to complete keepalive messaging. Generally, this is required for **kal-ap** or **kal-ap-vip**.
- *threshold* - The load threshold is used only with the **kal-ap** CSS keepalive. Typically, the CSS keepalive reports 255 when a service is unavailable. This threshold allows the CSS to interpret lower reported numbers as unavailable. For example, if this parameter has a value of 100, all received load numbers greater than or equal to 100 cause the domain record to become unavailable for DNS decisions. Enter a value from 2 to 254. The default is 254.

For example:

```
(config)# dns-record a www.home.com 172.16.6.7 15 single kal-ap  
123.45.6.12 100
```

- **default** - For NS-records only. In a Network Proximity configuration, the CSS uses PDB information to return the next most proximate location. When a PDB is not available or not configured, the CSS uses the roundrobin load-balancing method. There is no failover scenario.
- **forwarder** - For NS-records only. Use this option to eliminate a potential single point of failure by providing up to two alternative DNS servers called forwarders. A forwarder can be any DNS server including a CSS configured as a DNS server. If the lower-level DNS server indicated in the NS-record is Down, the CSS sends the DNS request to the primary or secondary forwarder if configured. For information on configuring a DNS forwarder, see the [“Configuring a DNS Forwarder”](#) section.

In a Network Proximity configuration, if an optimal miss occurs (the lower-level DNS server that was indicated in the NS-record is Down), the PDNS sends the DNS request to the primary or secondary forwarder, depending on forwarder health and configuration. An optimal miss occurs when the PDNS cannot return the NS-record for the zone that the PDB indicated was most proximate.

For this failover to occur, the local NS-record must be in the Down state, and the PDB has indicated the local zone to be the zone most proximate to the client.

- **sticky-enabled** - Causes a CSS DNS server to attempt to send a sticky response to the client for the specified domain. For details on configuring DNS Sticky, see [Chapter 2, Configuring the DNS Sticky Feature](#). The CSS makes a decision based on one of the following three scenarios:
 - In a global server load-balancing (GSLB) environment without a global sticky database (GSDB), the CSS selects a server based on the srcip hash (regardless of the default load-balancing method) and the availability of the domain in the zone mesh. The use of the srcip hash ensures that the CSS selects a consistent zone for a given source IP address.

- In a GSLB environment with a GSDB, the CSS sends a lookup request to the Global Sticky Database for the requesting client’s local DNS server. If the GSDB has an entry in its sticky database for the client’s local DNS server IP address, it returns the appropriate zone index to the CSS. The CSS then returns the associated IP address to the client. Otherwise, the CSS selects a zone based on the default load-balancing method and informs the GSDB about the selected zone.
- In a Network Proximity environment, the CSS configured as a Proximity Domain Name Server (PDNS) first consults the GSDB. If a sticky database entry exists for the client’s local DNS server IP address, the PDNS sends the appropriate IP address to the client based on the zone index returned by the GSDB. If the GSDB does not contain an entry for the client’s local DNS server IP address, the PDNS consults the Proximity Database (PDB).

If the PDB contains an entry for the client’s local DNS server IP address, the PDNS formulates a response to the client based on the ordered zone index returned by the PDB and keepalive information. The PDNS informs the GSDB about the selected zone (performs a “set” function). If the PDB does not have an entry for the client’s local DNS server IP address or the sticky zone is unavailable, the CSS selects a zone based on its default load-balancing method and informs the GSDB about the selected zone.



Caution

If you configure any sticky domains in a particular zone, you must configure all sticky domains participating in the peer mesh in that same zone. Otherwise, the thrashing of the sticky zone index causes DNS Sticky to fail.

- **sticky-disabled** - Disables DNS Sticky for the specified domain on a CSS. This is the default. For details on configuring DNS Sticky, see [Chapter 2, Configuring the DNS Sticky Feature](#).

For example:

```
(config)# dns-record a www.home.com 123.45.6.7 15 single kal-ap
172.16.6.12 100 sticky-enabled
```

- **usedefault** - Returns domain records using the default DNS load-balancing method configured for the zone. See the “[Configuring DNS Server Zones](#)” section earlier in this chapter.

- **weightedrr** - Returns domain records based on the weighted roundrobin load-balancing method. This method uses the *weight* value to determine the zone from which the record should be requested.
- **srcip** - Returns domain records using a source IP address hash. For sticky-enabled domains without a GSDB, the CSS uses the *srcip* method regardless of the configured balance method.
- **leastloaded** - Returns domain records from the zone with the smallest load. Requires the *kal-ap* keepalive for remote sites or ICMP keepalives for local content rules.
- **preferlocal** - Returns local domain records whenever possible. If no local record exists, the CSS uses the balance method configured for the zone with the lowest zone index.
- **roundrobin** - Returns domain records by cycling among records available at the different zones to evenly distribute the load.
- **proximity** (the default) - Returns domain records based on proximity information. If a PDB is not configured or is unavailable in a zone, the CSS applies the default balance method for the selected zone for DNS resolution.



Note The above domain record load-balancing methods override the load-balancing options configured with the **dns-server zone** command. We recommend that you do not mix domain record load-balancing methods for the same domain.

For example:

```
(config)# dns-record a www.home.com 172.16.6.7 15 single kal-ap  
172.16.6.12 100 sticky-enabled leastloaded
```



Note For sticky-enabled domains without a GSDB, a CSS uses the *srcip* method regardless of the configured balance method. For sticky-enabled domains with a GSDB, a CSS uses the configured balance method when the GSDB does not contain an entry for the requested domain.

- *weight* - Value assigned to a domain in the local zone to determine how many hits the local zone receives for the specified domain compared with other zones in a peer mesh. For example, a zone with a domain weight of 10 receives twice as many hits as another zone with the same domain configured with a weight of 5.

Use this parameter with the weighted roundrobin DNS load-balancing method. (See the “[Configuring DNS Server Zones](#)” section earlier in this chapter.) CSSs configured as authoritative DNS servers in a peer mesh share domain weights with each other through APP. Enter an integer from 0 to 10. The default is 1. The weight value that you assign to an individual domain record takes precedence over the default record weight that you configure for the local zone.

The CSS uses the following guidelines when selecting a DNS load-balancing method on a domain basis:

- If a local record exists, the CSS uses the configured domain balance method to determine local DNS resolutions. This rule applies regardless of the keepalive state of the local record.
- If no local record exists, the CSS uses the balance method configured for the zone with the lowest zone index.

For example, consider the following configuration.

Zone	Domain Record	Balance Method
0	www.test.com	leastloaded
1	www.test.com	roundrobin
2	no local record configured for www.test.com	none configured

With this configuration, you can expect the following behavior:

- DNS resolutions occurring on the Zone 0 and Zone 2 DNS servers use the least-loaded balance method.
- DNS resolutions occurring on the Zone 1 DNS server use the roundrobin balance method.

**Note**

If you need to modify an existing A-record parameter, you must first remove the record using the **no dns-record a** *domain_name* command. Then recreate the A-record with the parameter change using the **dns-record a** command.

Removing a Domain Record

To remove a domain record from the running-config, use the **no dns-record** command.

The syntax for this global configuration mode command is:

```
no dns-record dns_name
```

The *dns_name* variable maps the DNS name to the address record. Enter the name as a case-sensitive unquoted text string with no spaces and a maximum length of 63 characters.

For example:

```
(config)# no dns-record www.home.com
```

Resetting the DNS Record Statistics

To reset the DNS record statistics displayed by the **show dns-record** command, use the **dns-record zero** command.

The syntax for this global configuration mode command is:

```
dns-record zero [a/ns {dns_name}|accel {dns_name}]
```

The options and variables for this command are:

- **a/ns** - Resets the statistics to zero for the domain records that are displayed by the **show dns-record statistics** command (see the “[Displaying DNS-Record Statistics](#)” section later in this chapter) and the **show dns-record proximity** command (see [Chapter 5, Configuring Network Proximity](#)).
- *dns_name* - Resets the statistics for the specified domain name mapped to the DNS record. To view a list of domain names, enter:

```
dns-record zero [a/ns|accel] ?
```

- **accel** - Resets the counters to zero for the accelerated records that are displayed by the **show dns-record accel** command in a Client Side Accelerator configuration. See the “[Displaying Domain Acceleration Records Statistics](#)” section in [Chapter 4, Configuring a Client-Side Accelerator](#).

Adding a DNS Name to a Content Rule

To specify a DNS name that maps to a content rule, use the **add dns** command. For zone-based DNS, use this command to configure domain names on a CSS acting as a **kal-ap** agent (for example, a CSS acting as a data center or a lower-level DNS server for an upper-level CSS configured with **kal-ap**, the CSS keepalive type). For details on **kal-ap**, see the “[Configuring Domain Records](#)” section.

Also, use the **add dns** command to specify domains on a CSS configured as a Content Routing Agent (CRA) VIP. For details on configuring a Content Routing Agent, see [Chapter 3, Configuring a CSS as a Content Routing Agent](#).

Enter the DNS name as a lowercase unquoted text string with no spaces and a length of 1 to 31 characters.



Note

The **add dns** command is part of the CSS Standard feature set.

For example:

```
(config-owner-content[arrowpoint-rule1])# add dns www.arrowpoint.com
```

Removing a DNS Name from a Content Rule

To remove a DNS name from a content rule, use the **remove dns** command with the DNS name you wish to remove. Enter the DNS name as a case-sensitive unquoted text string with no spaces and a maximum of 31 characters.



Note

The **remove dns** command is part of the CSS Standard feature set.

For example:

```
(config-owner-content[arrowpoint-rule1])# remove dns  
www.arrowpoint.com
```

To display a list of DNS names, enter:

```
(config-owner-content[arrowpoint-rule1])# remove dns ?
```

Configuring DNS Records with a Zero Weight

To provide backup sites in a DNS weighted roundrobin configuration when all domain records with weights from 1 to 10 are unavailable, configure DNS records with a weight of zero. When a DNS record has a weight of zero, a CSS does not consider that record for selection when using the weighted roundrobin algorithm unless all the other records, with weights from 1 to 10, are unavailable. This feature is intended especially for use in disaster recovery sites.

In a DNS peer mesh, it is possible to have multiple DNS records that have a weight of zero. If all the DNS records weighted from 1 to 10 are unavailable, then the CSS cycles through the zero-weighted DNS records using the roundrobin balance method.

You can configure the same DNS record with different balance methods on two sites. In this case, all decisions are based on the balance method that is configured on the CSS making the DNS decision. Even if a DNS record is not configured with weighted roundrobin, it still broadcasts a default weight of 1 to all its peers.

If you configure a default weight for all records in a zone, a CSS advertises that value for records that are part of the zone regardless of their balance method. You can override the default record weight for a zone by configuring a record with weighted roundrobin and assigning a weight value to the record using the **dns-record** command. For details on configuring a default record weight for a zone, see the “[Configuring DNS Server Zones](#)” section.



Note

If DNS Sticky is enabled, a CSS could stick a user to a zero-weighted site. Even if other non-zero-weighted sites return to an active state, the CSS will not attempt to reroute the user to a non-zero site.

For details on the **dns-record** command including syntax, variables, and options, see the “[Configuring Domain Records](#)” section.

Configuring kal-ap-vip

The **kal-ap-vip** option of the **dns-record** command extends the functionality of **kal-ap** (the CSS keepalive that uses domain names configured on a single content rule) by providing load and status responses to queries for virtual IP (VIP) addresses configured on one or more content rules. This feature allows greater flexibility and accuracy of load and status reports for multiple content rules that are configured with the same VIP. This feature also eliminates the need for configuring domain names on a CSS that is responding to **kal-ap-vip** queries only and is not running a local DNS server.

Overview

In a manner similar to **kal-ap**, **kal-ap-vip** has two main components:

- Client
- Agent

A client is a CSS that requests load and status information for a VIP from an agent. You configure a client to generate queries using the **dns-record** command. For details, see the “[Configuring a kal-ap-vip Client](#)” section later in this section.

An agent is a CSS that responds to client queries with load and status reports for the requested VIPs. A **kal-ap-vip** agent can handle and respond to queries from local or remote CSSs (including itself) and other supported devices. No additional configuration is required for the agent.

To best service requests for a domain when a CSS makes GSLB decisions, a CSS may need to consider the keepalive status and load information of all content rules sharing the same VIP. Often, a **kal-ap-vip** configuration has at least two content rules to handle domain traffic: one for port 80 (TCP) and one for port 443 (SSL). The load reported by the agent is the average load of all the content rules that share the same VIP, unless a content rule is suspended.

In order for a **kal-ap-vip** agent to return a load value from 2 to 254 (indicating an Alive status) for a requested VIP, at least one service must be Up on each content rule sharing the requested VIP. For a requested VIP, if all services configured on one content rule are Down, or if one content rule is suspended, the agent reports a load of 255, indicating that the VIP is unavailable.

Configuration Requirements

Kal-ap-vip requires that you configure the following:

- Application Peering Protocol-User Datagram Protocol (APP-UDP) - Used to transmit **kal-ap-vip** datagrams. (For information on configuring APP-UDP, see the “[Configuring APP-UDP and APP](#)” section in [Chapter 5, Configuring Network Proximity](#).) The datagrams can contain a mix of both kal-ap (by domain or tag) and **kal-ap-vip** requests.
- **dns-record** command with the **kal-ap-vip** option - Used to configure a **kal-ap-vip** client. See the “[Configuring a kal-ap-vip Client](#)” section later in this chapter.



Note

You can configure **kal-ap-vip** and **kal-ap** on the same CSS. If you configure **kal-ap** on a CSS, you must also configure the **add dns** command with the appropriate domain names on the CSS acting as an agent. The agent responds with the load information for a VIP and/or a domain, as appropriate. For information on the **add dns** command, see the “[Adding a DNS Name to a Content Rule](#)” section.

Configuring a kal-ap-vip Client

To configure a **kal-ap-vip** client on a CSS to allow the CSS to query a **kal-ap-vip** agent for keepalive information on multiple content rules, use the **kal-ap-vip** option of the **dns-record** command.

The syntax for this global configuration command is:

```
dns-record alns dns_name ip_address {ttl_value {single|multiple  
{kal-ap-vip {ip_address2}}}}
```

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

- **alns** - Indicates a request for an address record (**a**) or a name server record (**ns**).
- *dns_name* - Domain name mapped to the address record or name server record. Enter the name as a lowercase unquoted text string with no spaces and a maximum of 63 characters.

- *ip_address* - IP address bound to the domain name within the DNS server zone. Enter the address in dotted-decimal notation (for example, 172.16.6.7). This is the VIP for which a CSS client sends a **kal-ap-vip** request to itself or another CSS agent for load information.
- *tll_value* - Optional Time to Live (TTL) value, in seconds. This value determines how long the DNS client remembers the IP address response to the query. Enter a value between 0 to 65535. The default is 0.
- **single|multiple** - Optional number of records to return in a DNS response message. By default, the DNS server returns a single A-record. Specifying **single** returns one A- or NS-record. Specifying **multiple** returns two A- or NS-records.
- **kal-ap-vip** - Optional CSS keepalive message type keyword used by a CSS client to request load information for the VIP specified in the *ip_address* value from the CSS agent specified in the *ip_address2* value. Use this option to allow a CSS client to query a local or remote CSS agent for load information for a VIP configured on multiple content rules.
- *ip_address2* - IP address of the local or remote CSS agent interface receiving CSS keepalive messages. If you omit this address while the keepalive type is specified, the CSS uses the DNS IP address to complete keepalive messaging.

For example:

```
(config)# dns-record a www.work.com 192.168.12.7 10 single
kal-ap-vip 172.16.25.3
```

For details on the other **dns-record** command options and variables, see the [“Configuring Domain Records”](#) section.

Configuring Content Rule-Based DNS on a CSS

Content rule-based DNS uses domain names configured on content rules to resolve DNS requests to IP addresses. Such a configuration is sometimes referred to as *static proximity*. Each CSS:

- Acts as an authoritative DNS server for the domain it represents
- Shares owner information, content rules, and load information with other CSS peers using APP

This section contains the following topics:

- [Content Rule-Based DNS Quick Start](#)
- [Configuring the DNS Exchange Policy for an Owner](#)
- [Configuring CSS DNS Peering](#)
- [Configuring a DNS Server](#)
- [Adding a DNS Name to a Content Rule](#)
- [Removing a DNS Name from a Content Rule](#)

**Note**

The recommended method for configuring DNS in a global server load balancing environment on a CSS is zone-based DNS. For details on configuring zone-based DNS server functionality on a CSS, see the “[Configuring Zone-Based DNS on a CSS](#)” section.

Content Rule-Based DNS Quick Start

[Table 1-4](#) provides a quick overview of the steps required to configure content rule-based DNS. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI command, see the sections following [Table 1-4](#).

Table 1-4 *Content Rule-Based DNS Configuration Quick Start*

Task and Command Example

1. Enter config mode.

```
# config
(config)#
```

2. Enable APP to allow the local CSS to communicate with remote CSSs. See the “[Configuring the Application Peering Protocol](#)” section.

```
(config)# app
```

Table 1-4 Content Rule-Based DNS Configuration Quick Start (continued)

Task and Command Example
<p>3. Configure an APP session with any remote CSS that is participating in the peer mesh with the local CSS. The IP address you enter is a local interface address on the remote CSS. See the “Configuring the Application Peering Protocol” section. Be sure to configure APP sessions on the remote CSSs also.</p> <pre>(config)# app session 172.16.2.5</pre>
<p>4. Configure the CSS to act as a DNS server. See the “Enabling a DNS Server” section.</p> <pre>(config)# dns-server</pre>
<p>5. (Optional) Configure the dns-peer command and its options to set:</p> <ul style="list-style-type: none"> • Interval between load reports • Receive-slots • Send slots <p>See the “Configuring CSS DNS Peering” section.</p> <pre>(config)# dns-peer interval 60</pre>
<p>6. (Optional) Set the DNS exchange policy for the owner. See the “Configuring the DNS Exchange Policy for an Owner” section.</p> <pre>(config-owner[arrowpoint])# dns both</pre>
<p>7. Use the add dns command to configure:</p> <ul style="list-style-type: none"> • Domain that maps to a content rule • TTL for the domain name <p>See the “Adding a DNS Name to a Content Rule” section.</p> <pre>(config-owner-content[arrowpoint-rule1])# add dns www.arrowpoint.com 36</pre>
<p>8. (Optional) Verify the configuration using the show commands described in the sections that follow this table.</p>

The following running-config example shows the results of entering the commands described in [Table 1-4](#).

```
! ***** GLOBAL *****
  dns-server
  dns-peer interval 60

  app
  app session 172.16.2.5

! ***** OWNER *****
owner arrowpoint
  dns both

  content rule1
  add dns www.arrowpoint.com 36
```

Configuring the DNS Exchange Policy for an Owner

To set the DNS exchange policy for an owner, use the **dns** command. This command specifies how a CSS exchanges owner information and content rules with DNS peers. This functionality is disabled by default.

The syntax and options for this owner mode command are:

- **no dns** - Sets no DNS exchange policy for this owner (default). This owner is hidden from the CSS peer.
- **dns accept** - Accepts all content rules for this owner proposed by the CSS peer.
- **dns push** - Advertises the owner and pushes all its content rules to the CSS peer.
- **dns both** - Advertises the owner and pushes all its content rules to the CSS peer, and accepts all this owner's content rules proposed by the CSS peer.

For example, enter:

```
(config-owner[arrowpoint])# dns both
```

To reset the default CSS behavior of no exchange policy, enter:

```
(config-owner[arrowpoint])# no dns
```

Configuring CSS DNS Peering

To configure DNS peer functionality on a CSS, use the **dns-peer** command and its options. Peering specifies how CSSs share domain names and load information. This functionality is automatically enabled when you configure APP. See the “[Configuring the Application Peering Protocol](#)” section.

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

- **dns-peer interval** - Sets the time between the load reports that a CSS sends to the CSS DNS peers.
- **dns-peer receive-slots** - Sets the maximum number of DNS names that a CSS can receive from each CSS DNS peer.
- **dns-peer send-slots** - Sets the maximum number of DNS names that a CSS can send to each CSS DNS peer.

Configuring the DNS Peer Interval

To set the time between sending load reports to the CSS DNS peers, use the **dns-peer interval** command. Enter the peer interval time from 5 to 120 seconds. The default is 5.

For example:

```
(config)# dns-peer interval 60
```

To reset the DNS peer interval to its default value of 5 seconds, enter:

```
(config)# no dns-peer interval
```

Configuring DNS Peer Receive Slots

To set the maximum number of DNS names that a CSS can *receive* from each CSS DNS peer, use the **dns-peer receive-slots** command. Enter a number from 128 to 1024. The default is 128. Use this command to tune a heavily accessed CSS that is resolving more than 128 DNS names.

For example:

```
(config)# dns-peer receive-slots 200
```

To reset the DNS peer receive slots number to its default of 128, enter:

```
(config)# no dns-peer receive-slots
```

Configuring DNS Peer Send Slots

To set the maximum number of DNS names that a CSS can *send* to each CSS DNS peer, use the **dns-peer send-slots** command. Enter a number from 128 to 1024. The default is 128. Use this command to tune a CSS that is reporting more than 128 DNS names to each peer.

For example:

```
(config)# dns-peer send-slots 200
```

To reset the DNS peer send slots number to its default of 128, enter:

```
(config)# no dns-peer send-slots
```

Configuring DNS Peer Load Variance

To set the deterministic difference in peer load numbers that a CSS considers to be similar for the leastloaded algorithm in a DNS load-balancing decision., use the **dns-peer load variance** command.

The syntax for this global configuration mode command is:

```
dns-peer load variance number
```

For the *number* variable, enter an integer from 0 to 254. The default is 50. A value of zero disables the two-stage comparison used to determine the least-loaded site.

For example:

```
(config)# dns-peer load variance 200
```

To reset the DNS peer load variance to its default value of 50, enter:

```
(config)# no dns-peer variance
```

Configuring a DNS Server

As a DNS server, a CSS resolves domain names to IP addresses when it receives DNS requests from clients. To configure a CSS as a DNS server, use the **dns-server** command and its options. The options for this global configuration mode command are:

- **dns-server** - Enables the DNS server functionality on a CSS.
- **dns-server bufferCount** - Modifies the DNS response buffer count.

- **dns-server respTasks** - Modifies the DNS responder task count.
- **dns-server forwarder** - Enables a DNS forwarder (a CSS or a fully-functional BIND server), which resolves DNS requests that a CSS cannot resolve.

**Note**

You need to configure APP before you configure the DNS server as shown in [Table 1-4](#).

Enabling a DNS Server

To enable the DNS server functionality on a CSS, use the **dns-server** command.

**Note**

The **dns-server** command is part of the CSS Enhanced feature set and requires a separate license key.

For example:

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

To disable DNS server functionality on a CSS, enter:

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

Configuring dns-server bufferCount

To change the DNS response buffer count on the CSS, use the **dns-server bufferCount** command. Enter the number of buffers allocated for query response from 2 to 1000. The default is 50.

Use this command with the **show dns-server** command to tune the CSS only if the CSS experiences buffer depletion during normal operation. If the number of available name server buffers (NS Buffers) displayed by the **show dns-server** command drops below 2, use the **dns-server bufferCount** to increase the buffer count. You can also use the reclaimed buffer count as an indication of buffer depletion. When the supply of available buffers is depleted, the CSS reclaims used buffers and the oldest requests are dropped.

For example:

```
(config)# dns-server bufferCount 100
```

To reset the DNS response buffer count to its default value of 50, enter:

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

Configuring a DNS Forwarder

If the CSS cannot resolve a DNS request, it sends the request to another DNS server to obtain a suitable response. This server, called a DNS forwarder, can be any DNS server or a CSS configured as a DNS server. The CSS sends to the forwarder DNS requests that:

- Cannot be resolved by the CSS
- Contain an unsupported request or record type



Note

For Client Side Accelerator (CSA) configurations, the forwarder must be a full-featured DNS server. For details about CSA, see [Chapter 4, Configuring a Client-Side Accelerator](#).

The forwarder resolves the DNS requests and sends DNS responses to the client transparently through the CSS. To monitor forwarder health, a keepalive mechanism (internal to the CSS) sends queries periodically to the forwarder to validate its state.

To configure a DNS forwarder on a CSS, use the **dns-server forwarder** command. The syntax for this global configuration mode command is:

```
dns-server forwarder [primary ip_address | secondary ip_address | zero]
```

The variables and options are:

- **primary** - Specifies a DNS server as the first choice forwarder. The CSS sends requests that it cannot resolve to the primary forwarder unless it is unavailable, in which case, it uses the secondary forwarder. When the primary forwarder is available again, the CSS resumes sending requests to the primary forwarder.
- **secondary** - Specifies a DNS server as the second choice forwarder.
- *ip_address* - Specifies the IP address of the forwarder. Enter the address in dotted-decimal notation (for example, 192.168.11.1).
- **zero** - Resets the statistics of both forwarders on a CSS.

For example:

```
(config)# dns-server forwarder primary 192.168.11.1 secondary  
192.168.11.2
```

To delete the primary forwarder on a CSS, enter:

```
(config)# no dns-server forwarder primary
```

Configuring dns-server respTasks

To change the DNS responder task count, use the **dns-server respTasks** command. Enter the number of tasks to handle DNS responses as an integer from 1 to 250. The default is 2.

For example:

```
(config)# dns-server respTasks 3
```

To set the DNS responder task count to its default value of 2, enter:

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

Adding a DNS Name to a Content Rule

To specify a DNS name that maps to a content rule, use the **add dns** command. Enter the DNS name as a lowercase unquoted text string with no spaces and a length of 1 to 31 characters.



Note

The **add dns** command is part of the CSS Standard feature set.

When you add the DNS name to the content rule, you may also enter an optional Time to Live (TTL) value in seconds. This value specifies how long the DNS client remembers the IP address response to the query. Enter a value from 0 to 255. The default is 0.



Note

You must configure the TTL when you add the DNS name to the content rule. To add a TTL to an existing rule, use the **remove dns** command to remove the dns name. Then use the **add dns** command to reconfigure the DNS name with a TTL value.

For example:

```
(config-owner-content [arrowpoint-rule1])# add dns www.arrowpoint.com
36
```

Removing a DNS Name from a Content Rule

To remove a DNS name from a content rule, use the **remove dns** command with the DNS name you wish to remove. Enter the DNS name as a case-sensitive unquoted text string with no spaces and a maximum of 31 characters.



Note

The **remove dns** command is part of the CSS Standard feature set.

For example:

```
(config-owner-content [arrowpoint-rule1])# remove dns
www.arrowpoint.com
```

To display a list of DNS names, enter:

```
(config-owner-content [arrowpoint-rule1])# remove dns ?
```

Displaying CSS DNS Information

Use the **show** commands in the following sections to display DNS information for a CSS configured as a DNS server. This section contains the following topics:

- [Displaying DNS Server Information](#)
- [Displaying DNS Server Zones](#)
- [Displaying DNS Record Information](#)
- [Displaying DNS Peer Information](#)
- [Displaying Domain Summary Information](#)

Displaying DNS Server Information

To display DNS server configuration and database information, use the **show dns-server** commands for both zone-based and rule-based DNS configurations. These commands provide the following options and information:

- **show dns-server** - Displays DNS server configuration information
- **show dns-server dbase** - Displays DNS database information
- **show dns-server stats** - Displays DNS database statistics
- **show dns-server forwarder** - Displays DNS server forwarder statistics

Displaying DNS Server Configuration Information

Use the **show dns-server** command to display information about your DNS server configuration. The syntax for this global configuration mode command is:

```
show dns-server
```

[Table 1-5](#) describes the fields in the **show dns-server** output.

Table 1-5 *Field Descriptions for the show dns-server Command*

Field	Description
DNS Server Configuration	The enable or disable state of the DNS server function on the CSS. When enabled, the CSS acts as the authoritative name server for the content domain.
ACL Index	The ACL index number applied to the DNS server. If this field is 0, no ACL has been applied.
Responder Task Count	The configured DNS server responder task count. These tasks handle responses to incoming DNS query requests. The default is 2. The range is from 1 to 250.
Name Server Buffers	
Total Count	The configured DNS server buffer count. The responder tasks share the buffers to handle incoming queries. The default is 50.
Current Free Count	The number of buffers currently available.

Table 1-5 *Field Descriptions for the show dns-server Command (continued)*

Field	Description
Minimum Free Count	The smallest number of buffers that were available at any one time.
Reclaimed Count	The number of buffers forcibly reclaimed by the DNS server software.
Requests Accepted	The number of DNS queries accepted.
Responses Sent	The number of DNS responses sent.
No Error	The number of queries that the DNS server successfully answered.
Format Error	The number of queries received that had a packet format error.
Server Failure	The number of times that a referenced name server did not reply to a query.
Name Error	The number of queries received that the DNS server was not able to answer because the domain was not configured or no resources were online.
Not Implemented	The number of queries received requesting an operation that has not been implemented in the DNS server.
Operation Refused	The number of queries the DNS server received that it refused to answer.
Internal Resolver	
Requests Sent	The number of queries sent to another name server for resolution.
Responses Accepted	The number of replies received from another name server.
Proximity Lookups	
Requests Sent	The number of proximity lookups sent to the PDB.
Responses Accepted	The number of proximity responses received from the PDB.

**Note**

Proximity lookup information is displayed only when you configure a PDB IP address. For information on configuring a PDB, see the “[Configuring a Proximity Database](#)” section in [Chapter 5, Configuring Network Proximity](#).

Displaying DNS Server Database Statistics

Use the **dns-server dbase** command to display DNS server database statistics. The DNS server database contains DNS names that are configured locally or learned from peers and Time to Live (TTL) information for each DNS name. The syntax for this global configuration mode command is:

```
show dns-server dbase
```

[Table 1-6](#) describes the fields in the **show dns-server dbase** output.

Table 1-6 *Field Descriptions for the show dns-server dbase Command*

Field	Description
DN	The domain name of the entry.
DNSCB	The address of the DNS control block structure to return a DNS query response for the entry. This address is the location best suited to handle the request.
PROX	The address for the proximity record.

**Note**

When DNSCB and PROX have null values (0x0), these values indicate a host table mapping. For details, refer to the **host** command in the *Cisco Content Services Switch Administration Guide*.

Displaying DNS Server Domain Statistics

Use the **show dns-server stats** command to display DNS server domain statistics. The syntax for this global configuration mode command is:

```
show dns-server stats
```

[Table 1-7](#) describes the fields in the **show dns-server stats** output.

Table 1-7 *Field Descriptions for the show dns-server stats Command*

Field	Description
DNS Name	The domain name entry
Content Name	Where the domain entry is mapped (A-record, NS-record, or host table), or a content rule name
Location	The IP address associated with the entry
Resolve Local	The number of local resolutions performed for the entry
Remote	The number of remote resolutions performed for the entry

Displaying DNS Forwarder Statistics

Use the **show dns-server forwarder** command to display statistics on the CSS for the DNS forwarders. The syntax for this global configuration mode command is:

```
show dns-server forwarder
```

Table 1-8 describes the fields in the **show dns-server forwarder** output.

Table 1-8 *Field Descriptions for the show dns-server forwarder Command*

Field	Description
DNS Server Forwarder Primary	The state of the primary forwarder. The states are: <ul style="list-style-type: none"> • Not Configured • Up • Down
DNS Server Forwarder Secondary	The state of the secondary forwarder. The states are: <ul style="list-style-type: none"> • Not Configured • Up • Down
State Changes	The number of times that the forwarder's state changed.
Requests Sent	The total number of requests sent to a particular forwarder.
Responses Accepted	The total number of responses received from a particular forwarder.
Totals:	
Request Sent	The total number of requests sent to forwarders (primary and secondary).
Responses Accepted	The total number of responses received from forwarders (primary and secondary).

Displaying DNS Server Zones

Use the **show zone** command to display information about communication and the state of the specified DNS server zone or proximity zone, or all zones in a peer mesh.

The syntax for this global configuration command is:

```
show zone {zone {verbose} | local | verbose}
```

The variable and options for this command are:

- **zone** - Displays the zone index of a peer. If you omit this variable, this command displays the states of all proximity zones.
- **local** - Displays local zone information. This information includes a count of transmitted and received client packet types, the count of client packets, and a count of transmit errors.
- **verbose** - Displays extra information per APP negotiation. This information includes a count of transmitted and received client packet types, the count of client packets, and a count of APP transmit errors.

For example:

```
(config)# show zone
```

To display proximity zones, including a count of transmitted and received client packet types, the count of client packets, and a count of APP transmit errors, enter:

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

[Table 1-9](#) describes the fields in the **show zone** output.

Table 1-9 Field Descriptions for the show zone Command

Field	Description
Index	The zone index of the peer. The initial value is 255. Once peer communications are established using APP, the value changes to the zone index of the peer. If peer communications cannot be negotiated, the value remains at 255.
Description	Zone description as supplied by the peer from the dns-server zone command.

Table 1-9 *Field Descriptions for the show zone Command (continued)*

Field	Description
IP Address	The IP address of the peer. It corresponds to a locally configured APP session.
State	The state of the peer negotiation, which includes: <ul style="list-style-type: none"> • INIT - Initializing. Waiting for local configuration to complete. • SREQ - A connection request message has been sent to the peer. • RACK - An acknowledgment message has been received from the peer. • SACK - An acknowledgment request has been sent to the peer. • OPEN - Negotiations with the peer have completed successfully and the connection is open. • CLOSED - Negotiations with the peer have failed and the connection is closed.
State Chgs	The number of times the state has transitioned to OPEN and CLOSED.
UpTime	The amount of time that APP has been in the OPEN state.

Displaying DNS Record Information

Use the **show dns-record** command to display statistics about the DNS records that were manually configured or learned from peers in a zone-based DNS configuration.

Displaying DNS-Record Statistics

Use the **show dns-record statistics** command to display statistics associated with the address records (A-records), name server records (NS-records), or accelerated domain records (accel) configured locally and learned by the CSS from its peers. For information about accelerated domain records, see [Chapter 4, Configuring a Client-Side Accelerator](#).

The syntax for this global configuration mode command is:

```
show dns-record statistics {dns_name}
```

You may enter an optional domain name target to display content. If you omit the domain name, all domains appear.

For example:

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

Table 1-10 describes the fields in the **show dns-record statistics** output.

Table 1-10 Field Descriptions for the show dns-record statistics Command

Field	Description
<Domain name>	Domain name for the record.
Local	State of the local entry for the record. Up indicates that the entry is configured. A “-” character indicates that the entry is learned and not configured. Down indicates that the keepalive failed.
Zone Count	Number of zones where this record is configured.
Zone	Index number for the zone. A “*” character prepending the zone number indicates that the zone is a local entry.
Description	Zone description.
Type	DNS record type: <ul style="list-style-type: none"> • A - Address record • NS - Name-server record • Accel - An accelerated domain associated with a Client Side Accelerator (CSA)
IP Address	Configured IP address for the zone.

Table 1-10 Field Descriptions for the `show dns-record statistics` Command (continued)

Field (continued)	Description
TTL	Time to Live, which indicates how long the receiver of a DNS reply for the given domain should cache the address information. By default, the TTL value is 0, indicating that the name server receiving the response should not cache the information.
Hits	Total number of DNS hits.

Displaying DNS Record Keepalive Information

Use the `show dns-record keepalive` command to display DNS record keepalive information. The syntax for this global configuration mode command is:

```
show dns-record keepalive {dns-name}
```

The variable for this command is *dns-name*, the domain name associated with the DNS record. You can enter an optional domain name target to display content. If you omit this variable, all DNS records appear.

[Table 1-11](#) describes the fields in the `show dns-record keepalive` output.

Table 1-11 Field Descriptions for the `show dns-record keepalive` Command

Field	Description
Name	Domain name for the record.
Type	Keepalive message type for the record: Accel, ICMP, kal-ap, or none.
IP	Destination IP address of the keepalive message.
State	State of the record, either UP or DOWN.
Transitions	Number of state transitions.

Table 1-11 Field Descriptions for the *show dns-record keepalive* Command (continued)

Field	Description
Load	Load for the record, which applies only to a kal-ap record type. All other types always have a load of “-”, indicating an undetermined load (load reports are not being received). If the load value exceeds the threshold value, the DNS server removes the DNS record from eligibility.
Threshold	Configured load threshold for the record. This threshold applies only to a kal-ap record type. Record types of ICMP and none do not use the threshold value.

Displaying the DNS Record Weight

Use the **show dns-record weight** command to display the configured weight and the number of hits for all domains or the specified domain. The syntax for this global configuration command is:

```
show dns-record weight {dns_name}
```

The *dns-name* variable for this command is the domain name associated with the DNS record. You can enter an optional domain name target to display information for the specified domain record. If you omit this variable, all DNS records appear.

[Table 1-12](#) describes the fields in the **show dns-record weight** output.

Table 1-12 Field Descriptions for the *show dns-record weight* Command

Field	Description
Name	Domain name for the record.
Total Hits	Total number of hits in all zones for the specified domain name.
Zone	Zone index for each zone where the domain record resides. An asterisk indicates the local zone.
Description	Text description of the zone.
IP Address	IP address of the DNS server within the DNS server zone.

Table 1-12 *Field Descriptions for the show dns-record weight Command (continued)*

Field	Description
Weight	Configured weight value for the record.
Current Hits	Current number of hits for the domain record in the zone.
Total Hits	Total number of hits for the domain record in the zone.

Displaying DNS Peer Information

To display the DNS peering configuration, use the **show dns-peer** command in a rule-based DNS configuration.

For example:

```
(config)# show dns-peer
```

[Table 1-13](#) describes the fields in the **show dns-peer** output.

Table 1-13 *Field Descriptions for the show dns-peer Command*

Field	Description
CSD Peer Rcv Slots	The configured maximum number of DNS names that the CSS can receive from each CSS DNS peer over an APP connection. The default is 128. The range is from 128 to 1024.
CSD Peer Snd Slots	The configured maximum DNS names that the CSS can send to each CSS DNS peer. The default is 128. The range is from 128 to 1024.
Peer Report Interval	The configured time in seconds between sending load reports to CSS DNS peers over an APP connection. The default is 5. The range is from 5 to 120.

Displaying Domain Summary Information

To display content domain summary information, use the **show domain** command. The syntax and options are listed below. For options that require an IP address, specify the IP address for the peer.

- **show domain** - Displays content domain summary information including the number of domain peers and information about each peer.
- **show domain ip_address send|receive** - Displays content domain summary information including the number of domain peers and information for the specified peer IP address. To see a list of addresses, enter **show domain ?**.
 - Include the **send** option to display only the send load reports and transmit message statistics.
 - Include the **receive** option to display only the receive load reports and receive message statistics.
- **show domain hotlist** - Displays configuration information about domain hot lists.
- **show domain owners** - Displays shared owner names.
- **show domain owners ip_address** - Displays shared owner names for the specified peer IP address.
- **show domain rules** - Displays locally created or negotiated names.
- **show domain rules ip_address** - Displays locally created or negotiated names for the specified peer IP address.

Table 1-14 describes the fields in the **show domain** output.

Table 1-14 Field Descriptions for the show domain Command

Field	Description
Content Domain Summary	The number of domain peers.
Peer	The address for the peer.
CCC State	The state of the master FSM (finite state machine) that negotiates the APP (CCC) link.
OWN State	The state of the owner policy negotiation FSM that determines the owners about whom the peers share domain name and rule information.

Table 1-14 Field Descriptions for the show domain Command (continued)

Field	Description
Rule State	The state of the rule policy negotiation FSM that exchanges individual domain name and rule matching criteria and load report information.
SendSlots	The number of individual domain name rules on which the CSS sends load reports to the peer.
ReceiveSlots	The number of individual domain name rules on which the CSS receives load reports from the peer.
Interval	The time interval in seconds that load reports are sent to the peer.
MinRespTime	The minimum local flow response time. This number is shared with the peer to be used in conjunction with load numbers to normalize the load numbers shared between peers.
MaxRespTime	The maximum local flow response time. This number is shared with the peer to be used in conjunction with load numbers to normalize the load numbers shared between peers.
Policy	The negotiated load report send and receive policies.
Sending Load Reports for	The list of domain names for which the CSS sends load reports to the peer.
Receiving Load Reports for	The list of domain names for which the CSS receives load reports from the peer.
CCC Msg stats	The number of times each of the message types used in the CCC/OWN/Rule FSM negotiations with the peer has been sent or received.



Configuring the DNS Sticky Feature

This chapter provides an overview of the CSS Domain Name System (DNS) Sticky feature and describes how to configure it for operation. Information in this chapter applies to all CSS models, except where noted.

**Note**

The DNS Sticky feature requires the CSS Enhanced feature set license.

This chapter provides the following major sections:

- [Overview of DNS Sticky](#)
- [DNS Sticky Quick-Start Procedures](#)
- [Converting Content Rule-Based DNS to Zone-Based DNS](#)
- [Configuring DNS Sticky Parameters](#)
- [Displaying DNS Sticky Statistics](#)

Overview of DNS Sticky

Configure DNS Sticky on a CSS to ensure that e-commerce clients remain connected to a particular server for the duration of a transaction even when the client's browser refreshes the DNS mapping. While some browsers allow client connections to remain for the lifetime of the browser instance or for several hours, other browsers impose a connection limit of 30 minutes before requiring a DNS re-resolution. This may not be long enough for a client to complete an e-commerce transaction. A new DNS resolution could cause the client to connect to a server different from the original server and interrupt the transaction. DNS Sticky ensures that a client can complete a transaction if a DNS re-resolution occurs.

DNS Sticky extends the functionality of global server load balancing (GSLB) and Network Proximity by providing:

- **Stickiness on a per domain basis** - Allows you to configure DNS Sticky only on the domains you want.
- **Zone-based DNS** - Provides service for configured domains in a maximum of 256 zones using the roundrobin, preferlocal, least-loaded, or srcip (source IP address) load-balancing method.
- **Global Sticky Database (GSDB)** - Maintains a database of sticky mappings and provides appropriate responses to DNS Sticky queries from CSSs configured as authoritative DNS servers. The GSDB is a dedicated CSS 11150 with 256 MB of RAM configured as a sticky database. You configure a GSDB on a CSS configured as a Proximity Database (PDB) in each GSLB zone.

You can configure DNS Sticky in three different environments, depending on your current configuration and business needs as follows:

- [DNS Sticky Without a GSDB](#)
- [DNS Sticky with a GSDB](#)
- [DNS Sticky in a Network Proximity Environment](#)

DNS Sticky Without a GSDB

DNS Sticky without a GSDB in a GSLB environment provides a static, simple, and cost-effective solution to the DNS sticky problem. This solution:

- Allows you to configure DNS Sticky on the domains you want
- Uses the srcip load-balancing method to keep clients connected to a particular zone based on a srcip hash
- Provides services for domains in a maximum of 256 zones (using two tier2 levels)
- Does not require the configuration of a dedicated GSDB

In a GSLB sticky configuration without a GSDB, the CSS configured as an authoritative DNS server selects a server for a sticky domain request based on the srcip hash (regardless of the default load-balancing method) and the availability of the domain in the zone mesh. The use of the srcip hash ensures that the CSS selects a consistent zone for a given source IP address.

DNS Sticky with a GSDB

DNS Sticky with a GSDB in a GSLB environment provides a more robust sticky load-balancing solution than one without a GSDB. This solution includes all of the benefits of DNS Sticky without a GSDB, plus:

- A GSDB to keep track of sticky mappings and provide responses to requests for sticky-enabled domains
- Configuration of up to two GSDB interfaces on the authoritative CSS DNS server for redundancy purposes
- More effective sticky load balancing across all domain sites using the least-loaded load-balancing method

**Note**

If you configure a GSDB and any sticky domains in a particular zone, you must configure all sticky domains participating in the peer mesh in that same zone. Otherwise, the thrashing of the sticky zone index could cause DNS Sticky to fail. For details on configuring sticky domains, see the “[Configuring Domain Records](#)” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

In a GSLB sticky configuration with a GSDB, a CSS configured as an authoritative DNS server sends a lookup request to the GSDB for a sticky domain requested by a client. If the GSDB finds an entry for the client's local DNS server IP address in its sticky database, it returns the sticky zone index to the CSS. The CSS uses the sticky zone index and keepalive information to send the appropriate IP address to the client. If the GSDB does not have an entry for the client's local DNS server IP address or the zone in the sticky zone index returned by the GSDB is unavailable, the CSS selects a zone in the mesh based on the configured load-balancing method and informs the GSDB about the selected zone.

**Note**

Configuring a GSDB requires the prior configuration of a Proximity Database (PDB) on the same CSS. For details on configuring a PDB, see the “[Configuring a Proximity Database](#)” section in [Chapter 5, Configuring Network Proximity](#).

DNS Sticky in a Network Proximity Environment

Configure DNS Sticky in a Network Proximity environment to add stickiness to your network. This solution adds all the benefits of DNS Sticky with a GSDB to your existing proximity configuration. In this case, you can specify critical e-commerce sites as sticky domains and use proximity for your other domains.

In a Network Proximity environment, you configure a GSDB on the CSS configured as a Proximity Database (PDB) and at least one GSDB interface on the Proximity Domain Name Server (PDNS) in each zone. The IP address of the primary **GSDB interface** is typically the same as the PDB IP address. In addition, you configure sticky domain records using the **dns-record** command.

When a CSS configured as a PDNS receives a client request for a sticky domain, the PDNS first consults the GSDB. If a sticky database entry exists for the client's local DNS server IP address, the PDNS sends the appropriate IP address to the client based on the zone index returned by the GSDB. If a sticky database entry does not exist for the client's local DNS server IP address, the PDNS consults the PDB for a Proximity-based answer. The PDNS formulates a response to the client based on the ordered zone index returned by the PDB and keepalive information. The PDNS informs the GSDB about the selected zone.

If neither the GSDB nor the PDB returns a suitable response, the PDNS selects a zone based on its configured default load-balancing method to formulate an appropriate response to the client and informs the GSDB about the selected zone.

For details on configuring Network Proximity, see [Chapter 5, Configuring Network Proximity](#).

DNS Sticky Quick-Start Procedures

The following sections provide the procedures required to configure DNS Sticky on a CSS.

Configuring DNS Sticky without a GSDB

[Table 2-1](#) provides a quick overview of the steps required to configure DNS Sticky on a CSS without a GSDB. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI commands, see the DNS Sticky commands later in this chapter.

Table 2-1 *DNS Sticky Without a GSDB Configuration Quick Start*

Task and Command Example
<ol style="list-style-type: none">1. On a CSS that you want to configure DNS Sticky without a GSDB, enter config mode. <pre># config (config)#</pre>
<ol style="list-style-type: none">2. Enable Application Peering Protocol (APP). See the “Configuring the Application Peering Protocol” section in Chapter 1, Configuring the CSS as a Domain Name System Server. <pre>(config)# app</pre>
<ol style="list-style-type: none">3. Configure the DNS server zone. Specify the zone, tier number, and an optional text description. Do not enter a Proximity Database (PDB) IP address. See the “Configuring DNS Server Zones” section in Chapter 1, Configuring the CSS as a Domain Name System Server. <pre>(config)# dns-server zone 0 tier1 “usa”</pre>

Table 2-1 DNS Sticky Without a GSDB Configuration Quick Start (continued)**Task and Command Example**

4. Configure APP sessions with other DNS servers (if any) that are participating in the peer mesh. The IP address you enter is a local interface address (circuit address) on the DNS server in another zone. See the [“Configuring the Application Peering Protocol”](#) section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

```
(config)# app session 172.27.16.5
```

5. Configure sticky domain records. See the [“Configuring Sticky Domain Records”](#) section later in this chapter.

```
(config)# dns-record a www.home.com 192.168.1.5 15 single kal-ap
172.27.25.4 50 sticky-enabled
(config)# dns-record ns www.work.com 192.168.12.7 15 single
kal-ap 172.27.33.3 100 default sticky-enabled
```

6. Configure the CSS to act as a DNS server. See the [“Configuring a DNS Server”](#) section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

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

The following running-config example shows the results of entering the commands described in [Table 2-2](#).

```
!***** GLOBAL *****
dns-server zone 0 tier1 "usa"
dns-record a www.home.com 192.168.1.5 15 single kal-ap 172.27.25.4
50 sticky-enabled
dns-record ns www.work.com 192.168.12.7 15 single kal-ap 172.27.33.3
100 default sticky-enabled
dns-server

app
app session 172.27.16.5
```

Configuring DNS Sticky with a GSDB

The following sections describe the steps required to configure DNS Sticky with a GSDB. You can configure the GSDB and the DNS server in any order.

Global Sticky Database Configuration Quick Start

[Table 2-2](#) provides a quick overview of the steps required to configure the GSDB on a CSS. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI commands, see the DNS Sticky commands later in this chapter.

Table 2-2 Global Sticky Database Configuration Quick Start

Task and Command Example

1. On a dedicated CSS 11150 with 256 MB of RAM that you want to configure as a Global Sticky Database (GSDB), enter config mode.

```
# config
(config)#
```

2. Enable the Application Peering Protocol-User Datagram Protocol (APP-UDP) to allow the GSDB to communicate with the CSS authoritative DNS server in the same zone. See the “[Configuring APP-UDP and APP](#)” section in [Chapter 5, Configuring Network Proximity](#).

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

3. Enable the Application Peering Protocol (APP) to allow the GSDB to communicate with other GSDBs. See the “[Configuring the Application Peering Protocol](#)” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

```
(config)# app
```

4. Configure APP sessions with other GSDBs that are participating in the peer mesh with this GSDB. The IP address you enter is a local interface address on another GSDB. See the “[Configuring the Application Peering Protocol](#)” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

```
(config)# app session 172.27.16.3
```

Table 2-2 Global Sticky Database Configuration Quick Start (continued)**Task and Command Example**

5. Configure a Proximity Database (PDB) if not already configured. (To configure a GSDB, you must configure a PDB first.) For details on configuring a PDB, see the “[Configuring a Proximity Database](#)” section in [Chapter 5, Configuring Network Proximity](#).

```
(config)# proximity db 0 tier1 "usa"
```

6. Enable the GSDB.

```
(config)# gsdb
```

7. Optionally, configure the time-to-live (TTL) in seconds for the GSDB sticky entries. Enter an integer between 300 and 1000000. The default is 7200 seconds (2 hours).

```
(config)# gsdb ttl 14400
```

The following running-config example shows the results of entering the commands described in [Table 2-2](#).

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

proximity db 0 tier1 "usa"
gsdb
gsdb ttl 14400

app
app session 172.27.16.3
```

DNS Server Configuration Quick Start

Table 2-3 provides a quick overview of the steps required to configure the DNS Sticky feature on a CSS acting as an authoritative DNS server and using a GSDB. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI commands, see the DNS Sticky commands later in this chapter.

Table 2-3 DNS Server Configuration Quick Start

Task and Command Example

1. On a CSS different from the GSDB, but in the same zone, enter config mode.

```
# config
(config)#
```

2. Enable APP-UDP to allow the CSS to communicate with the GSDB. See the “Configuring APP-UDP and APP” section in Chapter 5, Configuring Network Proximity.

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

3. Enable Application Peering Protocol (APP). See the “Configuring the Application Peering Protocol” section in Chapter 1, Configuring the CSS as a Domain Name System Server.

```
(config)# app
```

4. Configure up to two interfaces on the CSS to communicate with the GSDB. See the “Configuring the Global Sticky Database Interface” section later in this chapter.

```
(config)# gsdb-interface primary 192.168.68.12
(config)# gsdb-interface secondary 192.168.68.13
```

5. Configure the DNS server zone for zone-based DNS. Specify the zone, tier number, and an optional text description. Do *not* enter a PDB IP address. See the “Configuring DNS Server Zones” section in Chapter 1, Configuring the CSS as a Domain Name System Server.

```
(config)# dns-server zone 0 tier1 "usa"
```

Table 2-3 DNS Server Configuration Quick Start (continued)**Task and Command Example**

6. Configure the CSS to act as a DNS server. See the “Configuring a DNS Server” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

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

7. Configure APP sessions with other DNS servers (if any) that are participating in the peer mesh with this zone. The IP address you enter is a local interface address on the DNS server in another zone. See the “Configuring the Application Peering Protocol” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

```
(config)# app session 172.27.16.5
```

8. Configure A-records and NS-records on the CSS. Use the **sticky-enabled** option for those domains that clients will use for e-commerce applications and any other applications requiring stickiness. See the “Configuring Sticky Domain Records” section later in this chapter.

```
(config)# dns-record a www.home.com 192.168.1.1 15 single kal-ap
172.27.25.4 50 sticky-enabled
(config)# dns-record ns www.work.com 192.172.12.1 15 single
kal-ap 172.27.33.3 100 default sticky-enabled
```

The following running-config example shows the results of entering the commands described in [Table 2-3](#).

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

dns-server zone 0 tier1 "usa"
dns-record a www.home.com 192.168.1.1 15 single kal-ap 172.27.25.4
50 sticky-enabled
dns-record ns www.work.com 192.168.12.1 15 single kal-ap 172.27.33.3
100 default sticky-enabled
gsdb-interface primary 192.168.68.12
gsdb-interface secondary 192.168.68.13
dns-server

app
app session 172.27.16.5
```

Configuring DNS Sticky with Network Proximity

The following sections describe the steps required to configure DNS Sticky in an existing Network Proximity configuration. For details on configuring Network Proximity, see [Chapter 5, Configuring Network Proximity](#).

Global Sticky Database Configuration Quick Start

[Table 2-4](#) provides a quick overview of the steps required to configure the GSDB on a PDB. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI commands, see the DNS Sticky commands later in this chapter.

Table 2-4 Global Sticky Database Configuration Quick Start

Task and Command Example

1. On a PDB (CSS 11150 with 256 MB of RAM configured as a Proximity Database) that you want to configure as a Global Sticky Database (GSDB), enter config mode.

```
# config
(config)#
```

2. Enable the GSDB.

```
(config)# gsdb
```

3. Optionally, configure the time-to-live (TTL) in seconds for the GSDB sticky entries. Enter an integer between 300 and 1000000. The default is 7200 (2 hours).

```
(config)# gsdb ttl 14400
```

DNS Server Configuration Quickstart

Table 2-5 provides a quick overview of the steps required to configure the DNS Sticky feature on a PDNS. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI commands, see the DNS Sticky commands later in this chapter.

Table 2-5 DNS Server Configuration Quick Start

Task and Command Example

1. On a CSS that you want to configure DNS Sticky, enter config mode.

```
# config
(config)#
```

2. Configure up to two interfaces on the CSS to communicate with the GSDB. See “[Configuring the Global Sticky Database Interface](#)” later in this chapter.

```
(config)# gsdb-interface primary 192.168.68.1
(config)# gsdb-interface secondary 192.168.68.2
```

3. Configure A-records and NS-records on the CSS. Use the **sticky-enabled** option for those domains that clients will use for e-commerce applications and any other applications requiring stickiness. See “[Configuring Sticky Domain Records](#)” later in this chapter.

```
(config)# dns-record a www.home.com 192.168.1.1 15 single kal-ap
172.68.25.1 50 sticky-enabled
(config)# dns-record ns www.work.com 192.168.12.1 15 single
kal-ap 172.92.33.1 100 default sticky-enabled
```

Converting Content Rule-Based DNS to Zone-Based DNS

DNS Sticky requires a zone-based DNS configuration. If you currently have a content rule-based DNS configuration, use the following procedure to convert your DNS configuration to a zone-based DNS configuration.

1. Remove all rule-based DNS commands from the existing configuration by issuing the “no” form of the commands. For example:

```
(config)# no dns-peer interval
(config)# no dns-peer receive-slots
(config)# no dns-peer send-slots

(config-owner)# no dns

(config-owner-content)# remove dns
(config-owner-content)# no dns-balance
```

2. Use the **dns-server zone** command to create zone information for each network location. See the “[Configuring DNS Server Zones](#)” section in [Chapter 1, Configuring the DNS Sticky Feature](#).

Note the following:

- The *zone_index* value must be different for each zone.
 - You can select tier2 for up to 16 different zones (tier1 allows 6 zones).
 - Select the load-balancing method of your choice.
3. Create DNS records that point to VIPs that are currently associated with DNS names. See the “[Configuring Domain Records](#)” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

For example, suppose you have the following owner configuration.

```
!***** OWNER *****
owner GLB

content rule1
  add service s1
  vip address 5.5.5.5
  add dns www.work.com
  active
```

You would need to add a record similar to the following.

```
(config)# dns-record a www.work.com 5.5.5.5 0 single kal-ap  
1.1.1.1
```

For details on configuring zone-based DNS, see the “Configuring a DNS Server” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

For details on configuring content rule-based DNS, see the “Configuring Content Rule-Based DNS on a CSS” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

Configuring DNS Sticky Parameters

The following sections describe the commands and their options and variables that you use to configure DNS Sticky.

Enabling the Global Sticky Database

The global sticky database comprises:

- Records of clients’ local DNS servers
- Sticky entry TTL values
- Sticky zone index

The sticky zone index contains a listing of configured sticky zones and IP addresses. If the GSDB has an entry for the requesting client’s local DNS server, the GSDB sends the sticky zone index to the CSS DNS server that made the GSDB lookup request. The CSS uses the sticky zone index and keepalive information to select the appropriate sticky zone and sends the IP address of the zone to the client’s local DNS server.



Note

Because the Global Sticky Database (GSDB) requires the configuration of a Proximity Database (PDB), you must configure a PDB before you enable the GSDB on the same CSS. For details on configuring a PDB, see the “Configuring a Proximity Database” section in [Chapter 5, Configuring Network Proximity](#).

To enable a GSDB on a dedicated CSS 11150 with 256 MB of RAM when you are configuring GSLB with a GSDB (see “[DNS Sticky with a GSDB](#)” earlier in this chapter) or when you are using DNS Sticky in a Network Proximity configuration (see “[DNS Sticky in a Network Proximity Environment](#)” earlier in this chapter), use the **gsdb** command.

**Note**

The **gsdb** command and its **show** and **no** versions are part of the PDB feature set and require the PDB license key.

**Note**

You do not need to configure a GSDB to use the basic DNS Sticky feature in a GSLB environment. However, a GSDB provides a more robust DNS Sticky and load-balancing configuration. For details on the available types of DNS Sticky configurations, see “[Overview of DNS Sticky](#)” earlier in this chapter.

The syntax for this global configuration mode command is:

```
gsdb
```

To disable a GSDB, enter:

```
(config)# no gsdb
```

Resetting the Global Sticky Database Statistics

Use the **gsdb zero** command to reset the Sticky Lookups and Sticky Sets statistics that are displayed by the **show gsdb** command. The syntax for this global configuration mode command is:

```
gsdb zero
```

Configuring the Global Sticky Database Interface

Use the **gsdb-interface** command on the CSS DNS server to create an interface for the CSS to communicate with a GSDB. A GSDB responds with a zone index to sticky queries from CSS DNS servers. All GSDBs participating in a peer mesh share sticky TTL and sticky zone information over APP.



Note The **gsdb-interface** command and its **no** version are part of the Enhanced feature set.

The syntax for this global configuration mode command is:

```
gsdb-interface [primary|secondary] ip_address
```

The variables and options are:

- **primary|secondary** - Specifies an interface for the primary or secondary GSDB. The CSS uses the primary GSDB for sticky requests unless it is unavailable, in which case it uses the secondary GSDB.
- *ip_address* - IP address of the GSDB. Enter the address in dotted-decimal notation (for example, 192.168.11.1).



Note In a Network Proximity configuration, the IP address of the primary GSDB interface is typically the same as the IP address of the PDB.

For example:

```
(config)# gsdb-interface primary 192.168.11.1
```

To delete a primary GSDB interface, enter:

```
(config)# no gsdb-interface primary
```

Resetting the Global Sticky Database Interface Statistics

Use the **gsdb-interface zero** command to reset the GSDB interface statistics that are displayed by the **show gsdb-interface** command. The syntax for this global configuration mode command is:

```
gsdb-interface zero
```

Configuring the Time to Live for Global Sticky Database Entries

Issue the **gsdb ttl** command on the GSDB to specify a time to live (TTL) for the GSDB sticky domain entries. The value you enter determines the length of time in seconds that GSDB entries are valid. Any new request from a D-proxy for a sticky domain that arrives before the timer expires, resets the timer.

The syntax for this global configuration mode command is:

```
gsdb ttl ttl_value
```

The variable is *ttl_value*, which specifies the length of time in seconds that GSDB entries are valid. Enter an integer between 300 and 1000000. The default is 7200 seconds (2 hours).

For example:

```
(config)# gsdb ttl 7200
```

Configuring Sticky Domain Records

Use the **dns-record** command to configure sticky domain records on the CSS configured as a DNS server. Domain records labeled as **sticky-enabled** indicate to the CSS that it should attempt to provide a sticky response before it answers the DNS query from the client.



Note

If you configure a GSDB and any sticky domains in a particular zone, you must configure all sticky domains participating in the peer mesh in that same zone. Otherwise, the thrashing of the sticky zone index could cause DNS Sticky to fail.

For details on configuring domain records, see the [“Configuring Domain Records”](#) section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

Configuring Server Zones for DNS Sticky

Use the **dns-server zone** command to configure DNS server zones in the CSS. This feature allows the CSS to respond to DNS requests based upon different balance criteria and domain availability within zones or locations. For details on configuring zones, see the “[Configuring DNS Server Zones](#)” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

Displaying DNS Sticky Statistics

To display DNS Sticky statistics for the GSDB, GSDB interface, domain records, and DNS server zones, use the **show** commands in this section.

Displaying Global Sticky Database Statistics

Use the **show gsdb** command to display GSDB statistics. This command is part of the PDB feature set and is available in all modes. The syntax is:

```
show gsdb
```

[Table 2-6](#) describes the fields in the **show gsdb** output.

Table 2-6 *Field Descriptions for the show gsdb Command*

Field	Description
Sticky Lookups	The number of sticky requests received from a CSS DNS server.
Sticky Sets	The number of times the DNS server selected the sticky zone and informed the GSDB because the GSDB did not have the zone information in its database.
Sticky BLKs Present	The number of sticky blocks that are currently in the GSDB. The sticky blocks contain the zone and TTL information for sticky-enabled domains.
Sticky TTL	The time to live (in seconds) of a sticky entry in the GSDB. Values range from 300 to 1000000 seconds. The default is 7200 seconds (2 hours).

Displaying GSDB Interface Statistics

Use the **show gsdb-interface** command to display statistics for the GSDB interface on the DNS server CSS. This command is part of the Enhanced feature set and is available in all modes.



Note

This command is not available on a PDB or a GSDB.

The syntax is:

```
show gsdb-interface
```

[Table 2-7](#) describes the fields in the **show gsdb-interface** output.

Table 2-7 *Field Descriptions for the show gsdb-interface Command*

Field	Description
Active GSDB	The GSDB that is currently being used: Primary or Secondary.
Primary Trans	The number of times the primary GSDB transitioned state between Up and Down.
Primary Req	The number of requests received by the primary GSDB from DNS servers.
Primary Rsp	The number of responses sent to DNS servers by the primary GSDB.
Secondary Trans	The number of times the secondary GSDB transitioned state between Up and Down.
Secondary Req	The number of requests received by the secondary GSDB from DNS servers.
Secondary Rsp	The number of responses sent to DNS servers by the secondary GSDB.
Total Req	The total number of requests sent by the DNS server to the GSDB.
Total Rsp	The total number of responses received by the DNS server from the GSDB.

Displaying DNS Sticky Domain Record Statistics

Use the **show dns-record sticky** command to view statistics associated with sticky domain records. This command is part of the Enhanced feature set and is available in all modes. The syntax is:

```
show dns-record sticky {dns_name}
```

The variable is *dns_name*, which is the DNS name mapped to a domain record for which you want to display sticky domain statistics. Enter the name as a lower case unquoted text string with no spaces and a maximum of 63 characters.

[Table 2-8](#) describes the fields in the **show dns-record sticky** output.

Table 2-8 Field Descriptions for the show dns-record sticky Command

Field	Description
Name	The name of the sticky domain associated with the record.
Last Zone	The zone index of the last zone that was selected either by the GSDB or by the DNS server's load-balancing method.
Last IP Used	The last source (D-proxy) IP address used as a key to make a sticky decision.
Sets	The number of times the DNS server selected the sticky zone and informed the GSDB because the GSDB did not have the zone information in its database.
GSDB Lookups	The number of times a DNS server sent a sticky lookup request to the GSDB for the specified domain.
GSDB Responses	The number of times the GSDB responded to GSDB Lookup requests from a DNS server for the specified domain.

Displaying Domain Load Statistics

Use the **show dns-record load** command to display load information associated with domains. The syntax for this all configuration mode command is:

```
show dns-record load {dns_name}
```

The variable is *dns_name*, which is the DNS name mapped to a domain record for which you want to display load statistics. Enter the name as a lower case unquoted text string with no spaces and a maximum of 63 characters.

[Table 2-9](#) describes the fields in the **show dns-record load** output.

Table 2-9 Field Descriptions for the show dns-record load Command

Field	Description
Name	The name of the domain associated with the record.
LeastLoaded	The zone index of the current least-loaded zone in the peer mesh.
Zone	The zone index of the zone or zones in which the record exists. An asterisk (*) indicates the zone index of the local zone.
Description	A text description of the zone.
Type	The record type: <ul style="list-style-type: none"> • A - Address record • NS - Name server record
IP Address	The IP address associated with the record for the returned zone.
Load	The load number, an integer from 2 to 255 indicating the zone's current burden for the specified domain. A load of 255 indicates that the service is offline. A dash (-) indicates an undefined load, that is, load reports are not being received.
MinRespTime	The response time of the fastest server associated with the zone. This parameter value is used to break ties when load numbers are similar. A dash indicates an undefined MinRespTime.

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 on displaying DNS record statistics, see the “[Displaying DNS Record Information](#)” 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 on displaying DNS record keepalives, see the “[Displaying DNS Record Keepalive Information](#)” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

Displaying Proximity and GSDB Metrics

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

The syntax for this global configuration mode command is:

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

The variables and options are:

- *ip_address* - IP address of the client’s local DNS server for which you want to display proximity metrics. Enter the address in dotted-decimal notation (for example, 192.168.11.1).
- *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 keyword 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.

In the GSDB, the metrics are sorted by sticky zone index. In the PDB, the round-trip time (RTT) metrics are sorted by proximity zone. In the PDNS, the metrics are sorted by RTT. An asterisk next to a zone indicates the local 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.

For example, to view the PDB and /or GSDB 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
```

[Table 2-10](#) describes the fields in the **show proximity metric** output.

Table 2-10 Field Descriptions for the show proximity metric Command

Field	Description
IP Address	The IP address of the client's local DNS server for which you want to display metrics.
IP Prefix	The IP prefix length or subnet mask associated with the IP address.
Index	The zone index number associated with the DNS zone. An asterisk (*) indicates the local zone where you issued the command.
Description	A logical name or text description of the zone.
Metric	The round-trip time (RTT) between the PDB and a referral-DNS server. The DNS server uses the RTT as the proximity metric for load-balancing decisions.
Sticky Value	The sticky zone index stored in the GSDB and returned to the PDNS after a sticky lookup.
TTL	For DNS Sticky configurations, the remaining time-to-live (TTL) in seconds for this GSDB entry.

Displaying Server Zones for DNS Sticky

Use the **show zone command** to display information about DNS server zones communicating in a zone mesh. For details on displaying DNS server zones, see the “[Displaying DNS Server Zones](#)” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

Configuring a CSS as a Content Routing Agent

This chapter provides an overview of the CSS Content Routing Agent (CRA) feature and describes how to configure it for operation. Information in this chapter applies to all CSS models, except where noted.

This chapter provides the following major sections:

- [Overview of the CRA Feature](#)
- [CRA Quick Start](#)
- [Configuring CRA Parameters](#)
- [Displaying CRA Statistics](#)

Overview of the CRA Feature

To improve a client's overall browser experience by decreasing the response times for content requests, configure a CSS as a Content Routing Agent (CRA). A Cisco Content Router 4430-B (Content Router) running software version 1.1 redirects a client to the closest (best) replicated-content site represented by a CRA, based on network delay using a software process called boomerang. For details on the Cisco Content Router software and boomerang, refer to the *Cisco Content Routing Software Configuration Guide and Command Reference*, Release 1.1.

Configure a CRA at each content site within each domain that you want to support. This configuration also requires a Content Router.

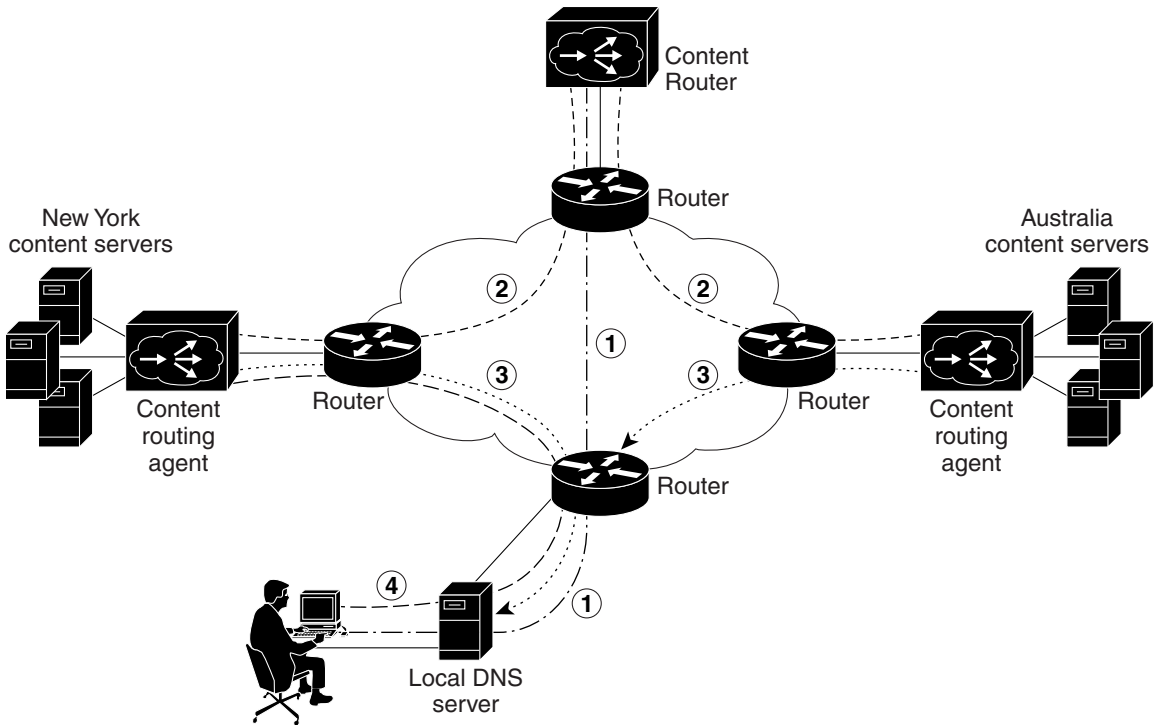
The Content Router intercepts DNS requests from a client, then routes them to a CRA. For example, to route `www.foo.com`, the address record (A-record) in the primary DNS server for `www.foo.com` is changed to a name server record (NS-record) pointing to the Content Router. The Content Router and its CRAs handle all requests for the IP address of `www.foo.com`. When the CRAs receive a DNS request from the Content Router, the CRAs respond to the client's local name server at the same time. The first response through the network is used and the local name server discards all other responses. The CRA with the winning response is the site to which the client connects.

[Figure 3-1](#) shows an example of the boomerang process in direct mode. A CSS configured as a CRA also works with a Content Router operating in (WCCP) mode. For more information on Content Router modes, refer to the *Cisco Content Routing Software Configuration Guide and Command Reference*, Release 1.1.

**Note**

The Content Routing Agent feature is part of the CSS Standard feature set.

Figure 3-1 Example of Boomerang Content Routing Process - Direct Mode



- ① DNS request is sent to Content Router.
- ② Content Router forwards request to content routing agents.
- ③ Agents simultaneously send responses back to local DNS server.
First response through the network contains the IP address of the best site.
- ④ User connects to best site.

46626

CRA Quick Start

Table 3-1 provides a quick overview of the steps required to configure the Content Routing Agent feature on a CSS. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI commands, see the sections following the table.

Table 3-1 Content Routing Agent Configuration Quick Start

Task and Command Example

1. Configure a Cisco Content Router 4430-B. Configure Content Routing Agents (CRAs) and the domains associated with them on the Content Router. For details, refer to the *Cisco Content Routing Software Configuration Guide and Command Reference*, Release 1.1.

2. On a CSS that you want to configure as a CRA, enter config mode.


```
# config
(config)#
```

3. Enable the CRA feature on a CSS.


```
(config)# dns-boomerang client enable
```

4. Create a domain record in the CRA for each domain with which you associated the CRA when you configured the Content Router.


```
(config)# dns-boomerang client domain www.sample.com 192.168.11.3
```

5. (Optional) Configure an alias for each configured domain to reduce administrative overhead.


```
(config)# dns-boomerang client domain www.sample.com alias
gif.www.sample.com
```

6. Display CRA statistics.


```
(config)# show dns-boomerang client
```

The following running-config example shows the results of entering the commands described in [Table 3-1](#).

```
!***** GLOBAL *****  
dns-boomerang client enable  
dns-boomerang client domain www.sample.com 192.168.11.3  
dns-boomerang client domain www.sample.com alias gif.www.sample.com
```

Configuring CRA Parameters

The following sections describe the CLI commands and their options and variables that you use to configure the CSS as a CRA.

Enabling the CRA

To enable the CRA functionality on the CSS, use the **dns-boomerang client enable** command. There are no options for this global configuration mode command.

For example:

```
(config)# dns-boomerang client enable
```

To disable the CRA, enter:

```
(config)# no dns-boomerang client enable
```

Configuring the CPU Load Threshold

To set the CSS CPU load threshold for domains configured to use or return a local virtual IP address (VIP), use the **dns-boomerang client cpu-threshold** command. If the CPU load exceeds the configured threshold value, then the CSS drops subsequent incoming DNS requests from the Content Router until the load is lower than the threshold.

The syntax for this global configuration mode command is:

```
dns-boomerang client cpu-threshold number
```

The *number* variable specifies the load threshold value. Enter an integer from 1 to 99. The default is 99.

For example:

```
(config)# dns-boomerang client cpu-threshold 50
```

To reset the CSS CPU threshold to the default value, enter:

```
(config)# no dns-boomerang client cpu-threshold
```



Note

To display the CPU load, use the **show system-resources** command.

Configuring CRA Domain Records

To create a domain record in the Content Routing Agent DNS server for each of the domains you associated the agent with when you configured domains on the Content Router, use the **dns-boomerang client domain** command. If the matching domain record keepalive messaging succeeds, the CSS uses this record for DNS resolutions. There is no Content Routing Agent configuration mode. Unlike other **dns-record** commands on the CSS, this command requires keywords for specifying options. (For details on configuring DNS domain records for other DNS applications, see the “[Configuring Domain Records](#)” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).)

The syntax for this global configuration mode command is:

```
dns-boomerang client domain dns_name ip_or_host [{"uri"}] {key
  [{"secret"}|des-encrypted encrypted_key|"encrypt_key"]}
  {dns-ttl number1} {ip-ttl number2} {threshold number3}]
```

The variables and options for this command are:

- *dns_name* - The domain name mapped to the client record. Enter the name as a case-sensitive, unquoted text string with no spaces and a maximum length of 72 characters. For example, www.sample.com.
- *ip_or_host* - The IP address or the host name of the content server or web cache bound to the domain name on the CSS. This IP address can be a local VIP. Enter the address in dotted-decimal notation (for example, 192.168.11.3).

- “*uri*” - The optional URI that the CSS uses for the keepalive probe to the Content Router for a domain. Enter a quoted text string with a maximum of 255 characters. If you do not prepend the URI with a slash (/) character, the CSS prepends it.
- **key** - The optional keyword that defines the clear-text shared RC4 secret or DES encryption key on the Content Router. See [Table 3-2](#) for a comparison of how you configure a password on a CSS (configured as a CRA) and on a Content Router.
- “*secret*” - The optional clear-text Content Router secret for encrypting packets sent between a Content Router and a CRA. The secret you specify here must match the secret configured on the Content Router. Enter the secret as a case-sensitive quoted text string with no spaces and a maximum of 64 characters (not including the quotes). For example, if MySecret is the secret configured on the Content Router for this domain, then enter “**MySecret**”.
- **des-encrypted** - The optional keyword that specifies that a Data Encryption Standard (DES)-encrypted password follows.
- *encrypted_key* - The DES encryption key that the CSS had previously encrypted. The CSS does not re-encrypt this key and saves it in the running-config as you entered it. Enter an unquoted case-sensitive text string with no spaces and a maximum of 64 characters.
- “*encrypt_key*” - The DES encryption key that you want the CSS to encrypt. The CSS saves the encrypted key in the running-config as you entered it. Enter a quoted case-sensitive text string with no spaces and a maximum of 16 characters.

Table 3-2 Configuring a Password on a CSS (CRA) Versus a Content Router

CSS Password Command	Content Router Password Command
key “ <i>secret</i> ”	no equivalent
key des-encrypt “ <i>password</i> ”	key word or key 0 word
key des-encrypt <i>password</i>	key 7 word



Note

The DES encryption algorithm on the CSS is different from the Cisco Type 7 encryption algorithm on the Content Router. Therefore, encrypted passwords are displayed differently on the CSS and on the CR.

- **dns-ttl** *number* - The optional DNS time-to-live keyword and value in seconds returned in the CRA's DNS responses. This option determines the length of time a DNS server caches the returned information for reuse. Enter an integer from 10 to 2147483647 seconds for a CSS. The default is the **dns-ttl** value configured on the Content Router.
- **ip-ttl** *number* - The optional IP routing time-to-live keyword and value in router hops returned in the CRA's DNS responses. This option determines how many router hops a response packet traverses enroute to the D-Proxy before it is discarded. This helps to eliminate the CRA from longer response routes. Enter an integer from 1 to 255. The default is the ip-ttl value configured on the Content Router.
- **threshold** *number* - The optional load threshold for testing the keepalive state of a local VIP. If the load on the dns-record associated with the content rule is greater than the threshold value, then the CSS drops subsequent Content Router requests for that record until the load is lower than the threshold. Enter an integer from 2 to 254. The default value is 254.



Note You must also configure the **add dns** command in the VIP's content rule to add domain names. Refer to the *Cisco Content Services Switch Global Server Load-Balancing Configuration Guide*.

For example:

```
(config)# dns-boomerang client domain www.foo.com 192.168.11.1 key  
"MySecret" dns-ttl 240 ip-ttl 5 threshold 175
```

To remove a CRA domain, enter:

```
(config)# no dns-boomerang client domain www.foo.com
```

Configuring an Alias for an Existing Client Domain

You can create an alias for each configured client domain. An alias reduces administrative overhead by allowing you to use the shorter alias name instead of the domain name, IP address, and all the other options and variables associated with the configured record. The alias behaves exactly the same as the configured domain name.

To create an alias for an existing client domain, use the **dns-boomerang client domain alias** command. The syntax for this global configuration mode command is:

```
dns-boomerang client domain dns_name alias alias_name
```

The variables and options are:

- *dns_name* - The domain name of a configured client record. Enter the name as a case-sensitive, unquoted text string with no spaces and a maximum of 72 characters.
- **alias** - The keyword required to create an alias name.
- *alias_name* - The domain name that the CSS treats exactly the same as the associated *dns_name*. Enter the name as a case-sensitive, unquoted text string with no spaces and a maximum of 72 characters.

For example:

```
(config)# dns-boomerang client domain www.sample.com alias  
gif.www.sample.com
```

To remove the alias, enter:

```
(config)# no dns-boomerang client domain alias www.sample.com
```

Clearing Domain Statistics

To clear the statistics for one or all configured domains, use the **dns-boomerang client zero** command. If you do not specify a domain name, the CSS clears the statistics for all configured domains. This command is available in SuperUser and all configuration modes.

The syntax for this global configuration mode command is:

```
dns-boomerang client zero dns-name
```

The variable for this command is *dns_name*, the domain name mapped to the client record statistics that you want to clear. Enter the name as a case-sensitive, unquoted text string with no spaces and a maximum of 72 characters.

For example:

```
(config)# dns-boomerang client zero www.sample.com
```

Displaying CRA Statistics

To display information for all configured CRA domains, use the **show dns-boomerang client** command. This command is available in SuperUser and all configuration modes.

The syntax for this global configuration mode command is:

```
show dns-boomerang client {all|global|domain {domain_name}}
```

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

- **all** - Displays all information (global and domain-related) for all domains mapped to a client record. Same as the **show dns-boomerang client command**.
- **global** - Displays global information only for all domains mapped to a client record.
- **domain** - Displays domain-related information for all domains mapped to a client record.
- *domain_name* - Displays domain-related information for the specified domain.

For example:

```
(config)# show dns-boomerang client global
```

[Table 3-3](#) describes the fields in the **show dns-boomerang client** output.

Table 3-3 *Field Descriptions for the show dns-boomerang client Command*

Field	Description
Total DNS A-record requests	The total number of address record requests from the Content Server.
Total packets dropped	
Unknown domain	The number of DNS packets with domains not configured on this CSS (for Content Routing).
Invalid source address	The number of packets with invalid source addresses.

Table 3-3 *Field Descriptions for the show dns-boomerang client Command (continued)*

Field	Description
Excess length	The number of packets that had lengths longer than what the CR could send.
CPU threshold exceeded	The number of packets dropped because the CPU threshold was exceeded.
Configured CPU threshold	The configured threshold value above which the CSS drops requests from the Content Router.
Rule load threshold exceeded	The number of requests from the Content Router that were dropped because the load on a local rule exceeded the configured threshold.
Keepalive state Down	The number of packets dropped because the keepalive failed.
Security failure	The number of requests for this domain that were dropped due to security errors (key/secret failure or mismatch).
Domain	The DNS name mapped to the client record.
Content server	The address of the content server bound to the domain.
Origin server	The address for the most recently used origin server that was passed from the Content Router.
Bad probes	The number of times (in percent) that the keepalive message failed to find the service Up.
DNS A-record requests	The number of DNS address record requests for this domain from the Content Router.
Dropped (server down)	The number of requests for this domain that were dropped because the server was down.
Dropped (CPU busy)	The number of requests for this domain that were dropped because the CPU threshold was exceeded.
Dropped (rule load exceeded)	The number of requests from the Content Router that were dropped because the load on a local rule exceeded the configured threshold.

Table 3-3 *Field Descriptions for the show dns-boomerang client Command (continued)*

Field	Description
Configured threshold	The load threshold value you configured with the dns-boomerang client domain command to test the keepalive state of a local VIP.
Security failures	The number of requests for this domain that were dropped due to security errors (key/secret failure or mismatch).
Alias	The alias that maps to the configured domain name.
DNS A-record requests	The number of DNS address record requests for this alias from the Content Router.



Configuring a Client-Side Accelerator

This chapter provides an overview of the CSS Client Side Accelerator (CSA) feature and describes how to configure it for operation. Information in this chapter applies to all CSS models, except where noted.



Note

The Client Side Accelerator feature requires the CSS Enhanced feature set license.

This chapter provides the following major sections:

- [Overview of the Client Side Accelerator Feature](#)
- [Configuration Examples of CSA](#)
- [CSA Quick Start](#)
- [Configuring CSA Parameters](#)
- [Displaying CSA Information](#)

Overview of the Client Side Accelerator Feature

To accelerate the retrieval of domain content, configure a CSS as a CSA, using transparent caches (TCs) to store content locally. This feature improves a user's experience by reducing the time for content to arrive in a browser.

A CSA resides on the client side of the Internet and is the first DNS server to which clients send a DNS request. When a CSA receives a DNS request for content located in a domain configured for acceleration and the number of requests exceeds the configured threshold, the CSA returns an address record (A-record) of the local virtual IP address (VIP) to the client. The client uses the IP address in the A-record to connect to the service in a local TC farm.

For non-accelerated content or unresolvable DNS requests, the CSA sends the DNS request to a DNS server forwarder. The forwarder, which is not a CSS, is a fully-functional Berkeley Internet Name Domain (BIND) DNS server. The forwarder returns a DNS response to the client transparently through the CSA.

You can configure a peer mesh of multiple CSAs that belong to one service provider but are located at various points of presence (POPs). Using Application Peering Protocol (APP), the CSAs in a peer mesh share accelerated domain records. This allows you to leverage content available at a cache farm in one POP to provide content to clients located at another POP. Once the same candidate domain has been accelerated at two POPs, cache backup can occur if a cache at one of those POPs fails.

Service providers can use CSAs to bill back domain acceleration to content providers. You can configure certain domains for acceleration, then bill back content providers based on the number of hits on the accelerated domains.

Configuration Examples of CSA

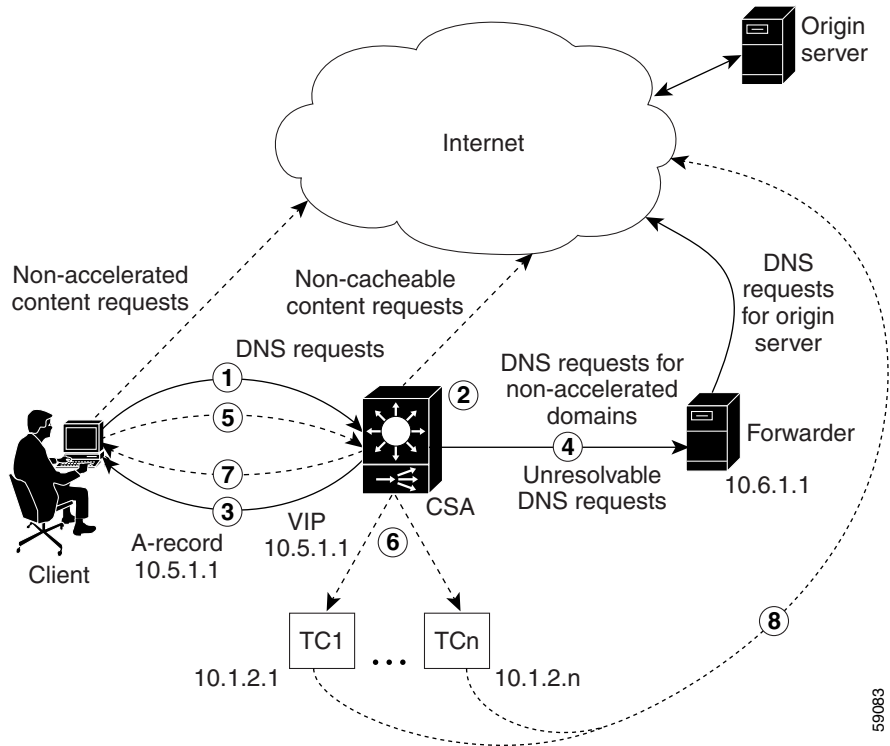
This section describes two possible CSA configurations: the first, a single POP (Figure 4-1) and the second, two POPs using an APP peer mesh (Figure 4-2).

Single-POP CSA Configuration

The following sequence describes the steps depicted in Figure 4-1:

1. The Client population sends DNS requests to the CSA for DNS resolution of the domain name `www.acme.com`.
2. The CSA is configured to accelerate the domain `www.acme.com`.
3. When the number of requests for `www.acme.com` exceed the configured threshold (or the threshold has already been exceeded), the CSA returns the accelerated VIP in an A-record to the clients.
4. For all other requests, the CSS forwards the queries to the DNS server forwarder for resolution.
5. Clients initiate a connection with the CSA for `www.acme.com` using the VIP in the A-record.
6. The CSA matches the request on a Layer 5 content rule that has transparent caches configured as the services. The CSA performs destination NATing based on the host tag and performs MAC forwarding.
7. If the cache contains the content, the CSA returns it to the clients.
8. If the cache does not contain the content, the cache fetches the content from the origin server.

Figure 4-1 Example of a Client Side Accelerator Configuration Example



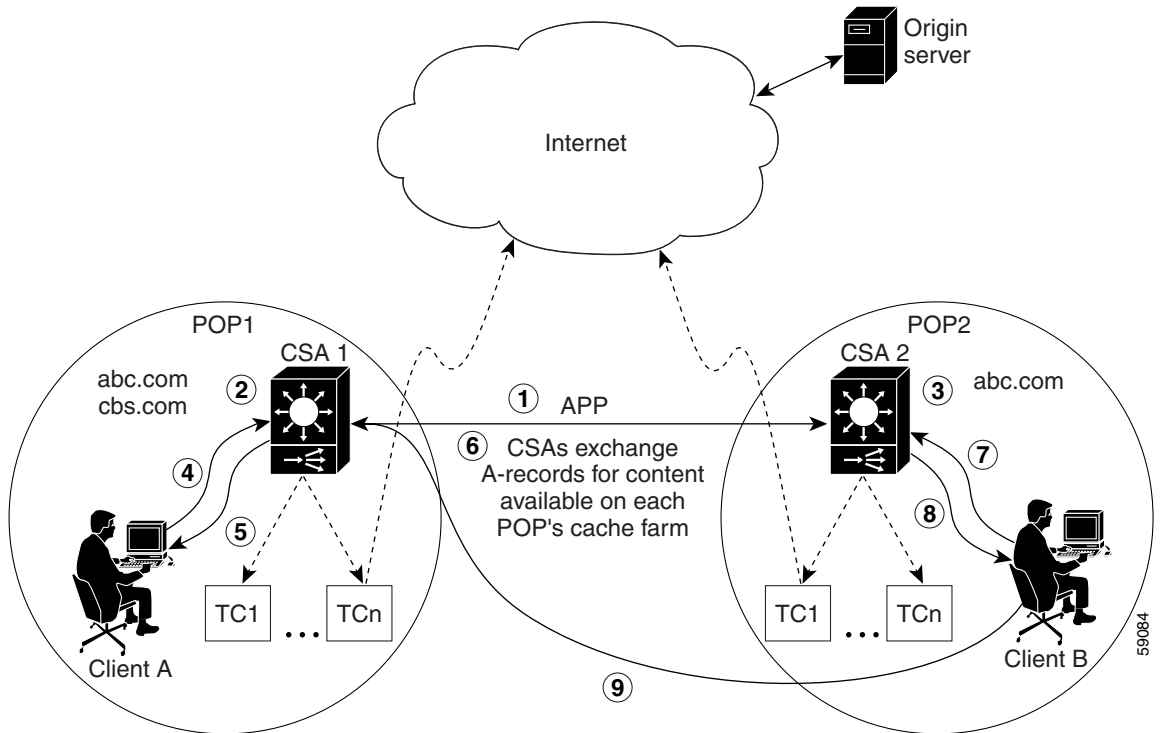
59083

Multiple-POP CSA Configuration

The following sequence describes the steps depicted in [Figure 4-2](#).

1. An APP mesh is configured for the CSAs in POP1 and POP2.
2. CSA1 in POP1 is configured to accelerate the domains abc.com and cbs.com.
3. CSA2 in POP2 is configured to accelerate the domain abc.com.
4. Client A population sends DNS requests to CSA1 in POP1 for abc.com.
5. When the number of requests for abc.com exceeds the configured threshold, CSA1 creates an A-record for abc.com and returns it to the clients. Clients in POP1 initiate a connection with CSA1 using the VIP in the A-record.
6. CSA1 also sends the A-record for abc.com out on the APP mesh.
7. Client B population sends DNS requests for abc.com to CSA2 in POP2. If CSA2 is configured to accelerate abc.com in multiple locations and if the domain becomes hot (requests exceed configured threshold), repeat Steps 5 and 6 for CSA2 in POP2.
8. If abc.com is not yet hot in POP2 or if CSA2 is configured to accelerate the domain in a single location, CSA2 sends the A-record in its domain database (learned in Step 6) to the Client B population.
9. The clients in POP2 request content in abc.com from CSA1 in POP1.

Figure 4-2 Example of a Client Side Accelerator APP Mesh Configuration



CSA Quick Start

Table 4-1 provides a quick overview of the steps required to configure the Client Side Accelerator feature on a CSS. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI command, see the sections following the table.

Table 4-1 Client Side Accelerator Configuration Quick Start

Task and Command Example

1. Enter config mode.

```
# config
(config)#
```

2. Enable Application Peering Protocol (APP).

```
(config)# app
```

3. Configure an APP session with each remote CSA peer to create a peer mesh. See the “Configuring an APP Session” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

```
(config)# app session 172.27.16.2
```

4. Configure back-end remapping as the preferred connection reset method. Refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*. This step is recommended, but not required.

```
(config)# persistence reset remap
(config)# bypass persistence disable
```

5. Configure candidate domains for acceleration. See “Configuring Accelerated Domains” later in this chapter.

```
(config)# dns-record accel abc.com 192.168.1.3
(config)# dns-record accel cbs.com 192.168.1.3
```

6. Configure CSA and enable sharing of content between peer CSAs. See “Enabling the CSA Feature” later in this chapter.

```
(config)# dns-server accelerate domains 50 30 256 single-location
```

7. Configure the DNS server and the number of CSA peers. See the “Configuring DNS Server Zones” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

```
(config)# dns-server zone 1 tier2
```

Table 4-1 Client Side Accelerator Configuration Quick Start (continued)

Task and Command Example
<p>8. Configure the DNS server forwarder. See the “Configuring a DNS Forwarder” section in Chapter 1, Configuring the CSS as a Domain Name System Server.</p> <pre>(config)# dns-server forwarder primary 192.168.1.10</pre>
<p>9. Enable the DNS server.</p> <pre>(config)# dns-server</pre>
<p>10. Configure the transparent caches as services. Refer to the <i>Cisco Content Services Switch Content Load-Balancing Configuration Guide</i>.</p> <pre>(config)# service transHosttag1 (config-service[transHosttag])# ip address 10.1.2.1 (config-service[transHosttag])# protocol tcp (config-service[transHosttag])# port 80 (config-service[transHosttag])# type transparent (config-service[transHosttag])# transparent-hosttag (config-service[transHosttag])# active</pre> <p>Repeat for each additional cache.</p>
<p>11. Configure an EQL.</p> <pre>(config)# eql cacheable (config-eql[cacheable])# extension .jpg</pre>

Table 4-1 Client Side Accelerator Configuration Quick Start (continued)**Task and Command Example**

12. Configure a content rule with the same VIP address as that used for the accelerated records. Add the transparent caches as services. Refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

```
(config)# owner accelerate
(config-owner[accelerate])# content 15-accel
(config-owner-content[accelerate-15-accel])# vip address
192.168.1.3
(config-owner-content[accelerate-15-accel])# protocol TCP
(config-owner-content[accelerate-15-accel])# port 80
(config-owner-content[accelerate-15-accel])# url "/" eq1
cacheable
(config-owner-content[accelerate-15-accel])# add service
transHosttag1
(config-owner-content[accelerate-15-accel])# add service
transHosttagn
(config-owner-content[accelerate-15-accel])# no persistent
(config-owner-content[accelerate-15-accel])# balance url
(config-owner-content[accelerate-15-accel])# active
```

13. Configure a service to bypass the cache farm for noncacheable content. Refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

```
(config)# service bypassCache
(config-service[bypassCache])# ip address 0.0.0.0
(config-service[bypassCache])# protocol tcp
(config-service[bypassCache])# port 80
(config-service[bypassCache])# keepalive type none
(config-service[bypassCache])# bypass-hosttag
(config-service[bypassCache])# active
```

Table 4-1 Client Side Accelerator Configuration Quick Start (continued)**Task and Command Example**

14. Configure a content rule for non-cacheable content on domains that you want to accelerate. Add the bypass cache as the only service. Refer to the *Cisco Content Services Switch Content Load-Balancing Configuration Guide*.

```
(config)# owner accelerate
(config-owner[accelerate])# content nonCacheable
(config-owner-content[accelerate-nonCacheable])# vip address
192.168.1.3
(config-owner-content[accelerate-nonCacheable])# protocol TCP
(config-owner-content[accelerate-nonCacheable])# port 80
(config-owner-content[accelerate-nonCacheable])# url "/"
(config-owner-content[accelerate-nonCacheable])# add service
bypassCache
(config-owner-content[accelerate-nonCacheable])# no persistent
(config-owner-content[accelerate-nonCacheable])# active
```

The following running-config example shows the results of entering the commands described in [Table 4-1](#).

```
!***** GLOBAL *****
persistence reset remap
bypass persistence disable

dns-record accel abc.com 192.168.1.2
dns-record accel cbs.com 192.168.1.2
dns-server accelerate domains 50 30 256 single-location
dns-server zone 1 tier2
dns-server forwarder primary 192.168.1.10
dns-server

app
app session 172.27.16.2
!***** SERVICE *****
service bypassCache
ip address 0.0.0.0
protocol tcp
port 80
keepalive type none
bypass-hosttag
active
```

```
service transHosttag1
  ip address 10.1.2.1
  protocol tcp
  port 80
  type transparent-cache
  transparent-hosttag
  active

!***** EQL *****
eql cacheable
  extension .jpg

!***** OWNER *****
owner accelerate

content l5-accel
  vip address 192.168.1.3
  protocol tcp
  port 80
  url "/" eql cacheable
  add service transHosttag1
  no persistent
  balance url
  active

content nonCacheable
  vip address 192.168.1.3
  protocol tcp
  port 80
  url "/"
  add service bypassCache
  no persistent
  active
```

Configuring CSA Parameters

The following sections describe the CLI commands and their options and variables that you use to configure the CSA feature on a CSS.

Enabling the CSA Feature

To enable the CSA functionality on a CSS, use the **dns-server accelerate domains** command. This command enables the acceleration of domains that have been or will be configured for acceleration using the **dns-record accel** command. The CSA uses the *threshold* variable to determine if it should accelerate a candidate domain. You can also configure whether or not the CSA shares content with peer CSAs.

The syntax for this global configuration mode command is:

```
dns-server accelerate domains threshold interval max_number  
[single_location/multi_location]
```

The variables and options for this command are:

- *threshold* - The number of hits per *interval* below which the CSA does not accelerate a domain. When the number of hits equals or exceeds the threshold during the configured *interval*, the CSA accelerates the domain. Enter an integer from 0 to 65535. The default is 0, indicating immediate acceleration.
- *interval* - The time period in minutes over which the CSA samples the hits on a domain and compares the number of hits with the configured *threshold* value to determine hot content and domain acceleration. Enter an integer from 1 to 60 minutes. The default is 5 minutes.
- *max_number* - The maximum number of domains that the CSA can accelerate. Enter an integer from 0 to 4096. The default is 1024.
- **single_location** - Allows the acceleration of a domain at one cache farm (POP) at a time. This is the default behavior.

- **multi_location** - Allows multiple CSAs to accelerate the same domain, possibly resulting in multiple cache farms maintaining the same content. This can occur when two or more CSAs (located in different POPs) are configured for `multi_location` and accelerate the same domain. Each cache farm will maintain the same content after:
 - The CSAs accelerate the same domain
 - A cache in each POP retrieves the same content from the origin server

The following command example:

- Accelerates domains that are accessed at least at the rate of 50 hits per minute.
- Accelerates a maximum of 100 candidate domains at any given time.
- Forces this CSA to allow acceleration of a given domain to occur in only one cache farm at a time.

```
(config)# dns-server accelerate domains 50 1 100 single_location
```

To disable CSA functionality on a CSS, enter:

```
(config)# no dns-server accelerate domains
```

Configuring the Domain Cache

To enable the tracking of DNS request counts and to configure the domain cache on the CSA, use the **dns-server domain-cache** command. As requests arrive at the CSA, entries are created or updated in the domain cache with the hit counts. You can use this command with the **show dns-server domain-cache** command to determine which domains you want to accelerate based on their hit counts.



Note

Do not use this command during normal CSA operation. It causes unnecessary overhead on the CSA.

The syntax for this global configuration command is:

```
dns-server domain-cache {cache_size ageout} purge {dns_name}|zero  
  {dns_name}}
```

The variables and options for this command are:

- *cache_size* - Specifies the number of domains that the CSA can cache. Enter an integer from 1 to 4096. The default is 1024.
- *ageout* - The maximum number of seconds that the domain entry remains in the cache. Enter an integer from 0 to 60. The default is 10 seconds. A value of zero causes the domain entry to never age out.
- **purge** - Removes all entries or the specified entries from the domain cache.
- *dns_name* - The DNS entry that you want to remove from the domain cache. To see a list of DNS entries, enter:

```
(config)# dns-server domain-cache purge ?
```
- **zero** - Resets all counters for all entries or the specified entry in the domain cache to zero.
- *dns_name* - The DNS entry for which you want to reset counters.

For example:

```
(config)# dns-server domain-cache 512
```

The above command creates a domain cache that can hold up to 512 most recently requested domain entries. The entries will age out and be removed from the domain cache after 10 seconds (the default).



Note

The operation of the domain cache can impact the DNS request/response rate performance. Use the domain cache only when you need to identify potential acceleration candidates.

Configuring a DNS Server Forwarder

To configure a DNS server forwarder on a CSS configured as a CSA, use the **dns-server forwarder** command. If the CSA cannot resolve a DNS request, it sends the request to the forwarder to obtain a suitable response. For details on configuring a DNS server forwarder, see the “[Configuring a DNS Forwarder](#)” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

Configuring Accelerated Domains

To specify the domains you want to accelerate, use the **dns-record accel** command. Map the domain name to a content rule using an IP address.

**Note**

If the content rule bound to the acceleration candidate domain is suspended or cannot provide service for content requests, the CSA does not accelerate the domain.

The syntax for this global configuration mode command is:

```
dns-record accel dns_name ip_address {ageout}
```

**Note**

You cannot configure a domain name as two different DNS record types on the same CSS. For example, if you have configured abc.com as **dns-record accel**, you cannot configure it as **dns-record a** or **dns-record ns** on the same CSS. For information on configuring other types of DNS records, see the “[Configuring Domain Records](#)” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

The variables and options for this command are:

- *dns_name* - The domain name that you want to map to the acceleration record. Enter the name as a case-sensitive unquoted text string with no spaces and a maximum of 63 characters.
- *ip_address* - The IP address of the local content rule that will handle content requests for the DNS name during content acceleration.
- *ageout* - The optional number of minutes that the domain remains accelerated. Enter an integer from 0 to 525600. The default is 180 minutes. If you enter 0, the accelerated domain record never ages out.

The following command example creates an acceleration record for the domain abc.com. When the number of requests for the domain exceeds the threshold specified in the **dns-server accelerate domains** command, the CSA accelerates the domain for six minutes. Clients can access the domain’s content based on the content rule with the IP address 192.168.11.1.

```
(config)# dns-record accel abc.com 192.168.11.1 6
```

To delete the domain acceleration record, enter:

```
(config)# no dns-record accel abc.com
```

Resetting the DNS Record Statistics

To reset the DNS record statistics displayed by the **show dns-record** command, use the **dns-record zero** command. For details on resetting DNS record statistics, see the “[Resetting the DNS Record Statistics](#)” and “[Displaying DNS-Record Statistics](#)” sections in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

Configuring the CSA DNS Server Zones

To configure the number of DNS server zones on a CSA, use the **dns-server zone** command. For details on DNS server zones, see the “[Configuring DNS Server Zones](#)” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

Displaying CSA Information

To display information and statistics for your CSA configuration, use the **show** commands in this section.

Displaying a CSA Configuration

To display the CSA configuration on a CSS, use the **show dns-server accelerate domains** command. The syntax for this global configuration mode command is:

```
show dns-server accelerate domains
```

Table 4-2 describes the fields in the `show dns-server accelerate domains` output.

Table 4-2 *Field Descriptions for the show dns-server accelerate domains Command*

Field	Description
Current CSA Config	The state of the CSA configuration: disabled or enabled.
Threshold	The configured hits threshold used to determine whether or not a domain is accelerated. When the hits on the domain are greater than or equal to the threshold, the CSA accelerates the domain. The range is from 0 to 65535. The default is 0, indicating that the candidate domains are always accelerated.
Interval	The configured time interval, in minutes, over which the CSS samples the hits on the domain and compares it with the threshold. The range is from 1 to 60 minutes. The default is 5 minutes.
Expirations	The number of times that the configured interval expired. You can use this value to determine whether domains are being accelerated or decelerated as expected. The CSA decelerates an accelerated domain only after the interval expires.
Max. to Accel	The maximum number of domains that can be accelerated. The range is 0 to 4096. The default is 1024.
Location	Indicates whether a single or multiple CSAs maintain the same content. <ul style="list-style-type: none"> • Single-location, the default setting, allows the acceleration of a domain at one cache farm (POP) at a time. • Multi-location allows multiple CSAs to accelerate the same domain resulting in multiple cache farms maintaining the same content.
Candidates Total	The total number of candidate domains that are configured on the CSA.
Candidates Accelerated	The total number of domains that are currently accelerated.

Displaying DNS Server Domain Cache Statistics

To display statistics for the DNS server domain cache, use the **show dns-server domain-cache** command. Use this command with the **dns-server domain-cache** command to help you determine which domains you want to accelerate based on their hit counts. The syntax for this global configuration mode command is:

```
show dns-server domain-cache {summary}
```

Table 4-3 describes the fields in the **show dns-server domain-cache** output.

Table 4-3 *Field Descriptions for the show dns-server domain-cache Command*

Field	Description
Domain	The domain name of the entry
Counter	The number of DNS requests



Note

If you include the **summary** option, the command output displays the domain cache configuration and state only.

Displaying DNS Server Zones

To display information about communication with, and the state of, the specified DNS zone or all zones in a peer mesh, use the **show zone command**. For details on displaying DNS server zones, see the “[Displaying DNS Server Zones](#)” section [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

Displaying DNS Record Keepalive Information

To display DNS record keepalive information, use the **show dns-record keepalive** command. For details on displaying DNS record keepalives, see the “[Displaying DNS Record Keepalive Information](#)” section in [Chapter 1, Configuring the CSS as a Domain Name System Server](#).

Displaying Domain Acceleration Records Statistics

To view statistics associated with domain acceleration records, use the **show dns-record accel** command. The syntax for this global configuration mode command is:

```
show dns-record accel dns-name
```

The variable for this command is *dns_name*, the domain name mapped to the acceleration record that you want to display. Enter the name as a case-sensitive unquoted text string with no spaces and a maximum of 63 characters.

[Table 4-4](#) describes the fields in the **show dns-record accel** output.

Table 4-4 Field Descriptions for the show dns-record accel Command

Field	Description
<Name>	The domain name for the acceleration record.
State	The state of the acceleration record, either ACCEL or NOT ACCEL. <ul style="list-style-type: none"> ACCEL - Indicates that the record is currently accelerated NOT ACCEL - Indicates the record is currently not accelerated
Vip Address	The virtual IP address of the local content rule that handles the content requests for the domain name during content acceleration.
Secs til Ageout	The number of seconds remaining until the CSA decelerates the domain record. The range is from 0 to 525600. The default is 180 seconds.
Interval Hits	The number of content hits that occurred during the interval set with the dns-server domain-cache command.
Total Hits	The total number of DNS hits for this record.
AccelCount	The number of times that content was accelerated.



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:

- [Entering Your Proximity License Keys](#)
- [Overview of Network Proximity](#)
- [Network Proximity Configuration Quick Start](#)
- [Configuring a Proximity Database](#)
- [Using Network Proximity Tiers](#)
- [Displaying PDB Configurations](#)
- [Configuring a PDNS](#)
- [Displaying PDNS Configurations](#)



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.

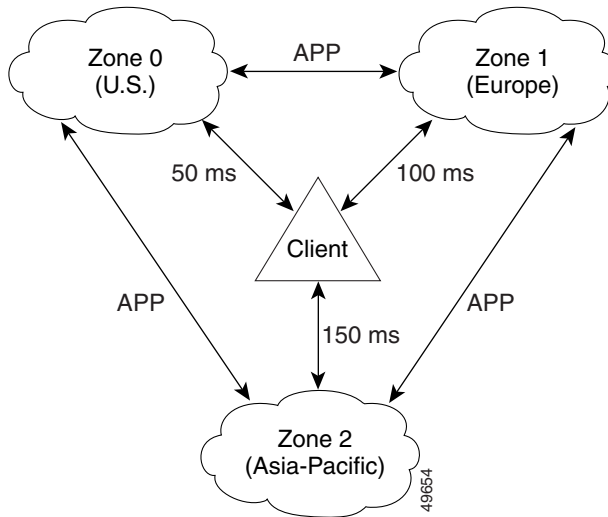
```
Enter the Software License Key (q to quit): nnnnnnnnnnnnnn
```

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:

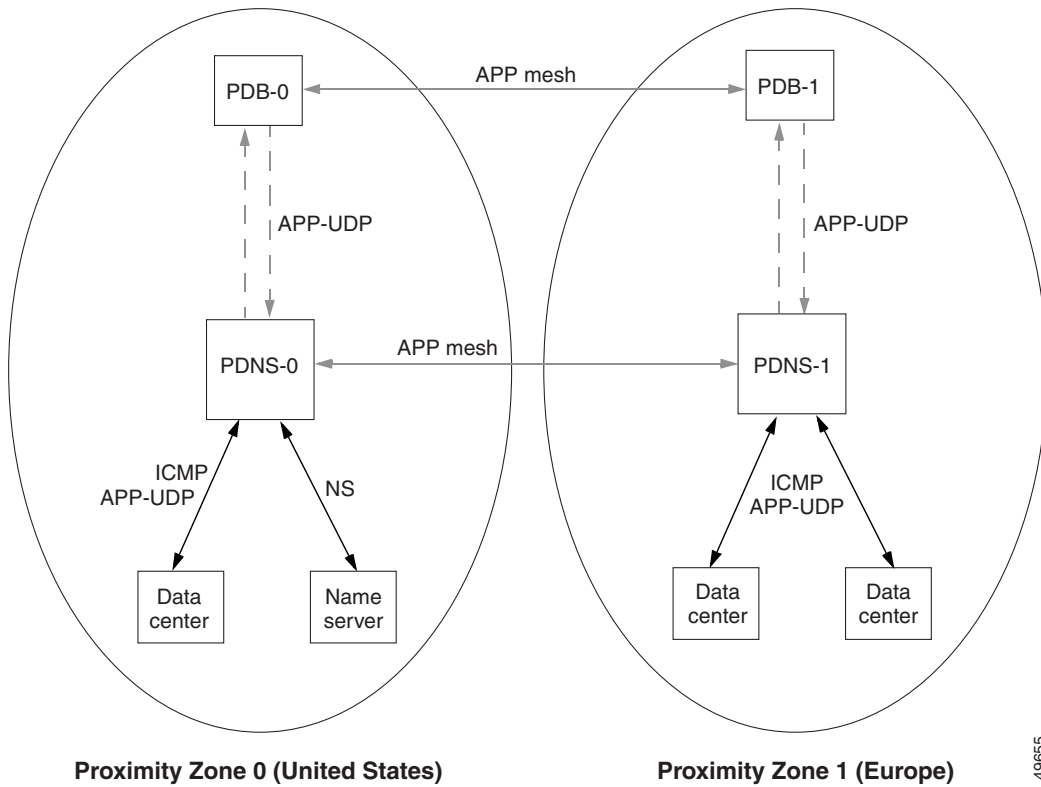
- The average load of the content rule to which the domain is attached
- The load of a locally configured A-record or NS-record

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



49655

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.

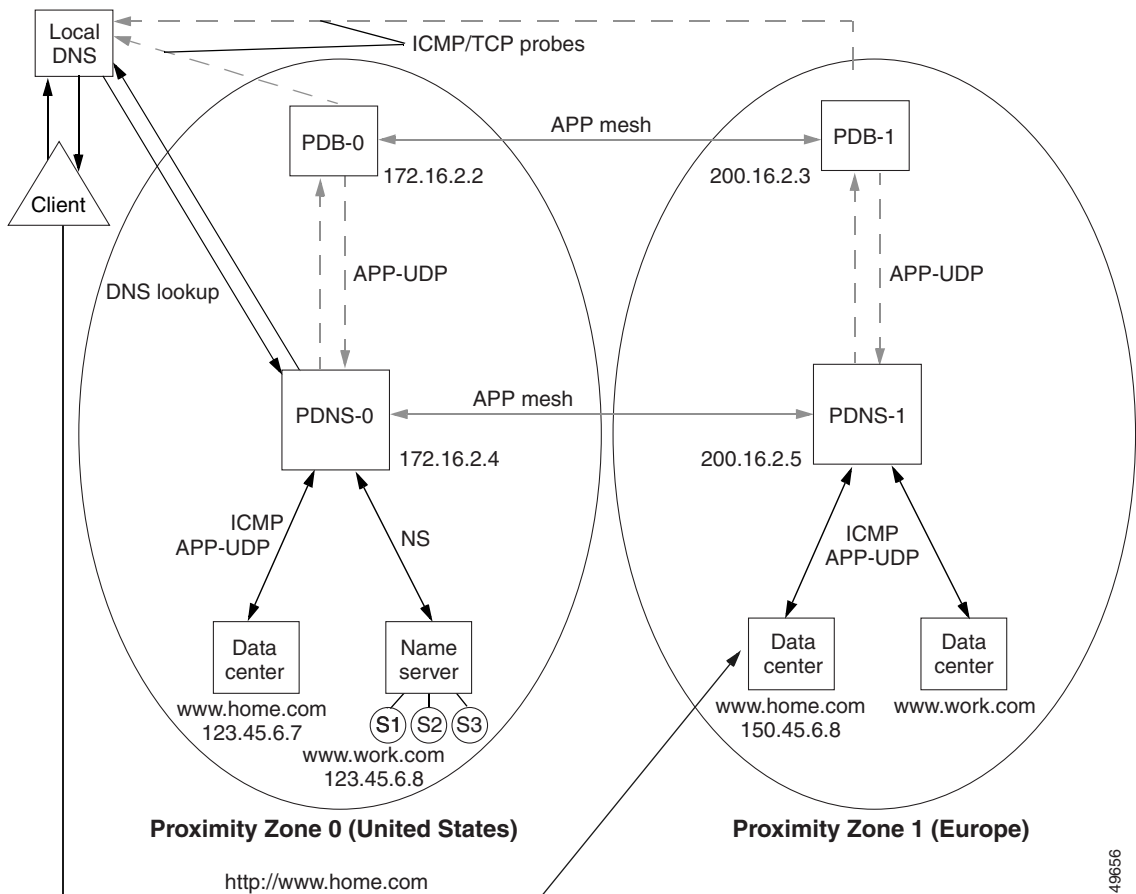
The following example illustrates a single point-in-time request from a client. See [Figure 5-3](#) for an illustration of the following steps.

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



49656

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.	<p>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.</p> <pre>(config)# app</pre>
4.	<p>Configure PDNS-0. Specify the proximity zone and tier number, an optional text description, and the IP address associated with PDB-0.</p> <pre>(config)# dns-server zone 0 tier1 "usa" 172.16.2.2</pre>
5.	<p>Configure the CSS to act as a DNS server.</p> <pre>(config)# dns-server</pre>
6.	<p>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.</p> <pre>(config)# app session 200.16.2.5</pre>
7.	<p>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.</p> <pre>(config)# dns-record a www.home.com 123.45.6.7 0 single kal-icmp</pre>
8.	<p>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.</p> <pre>(config)# dns-record ns www.work.com 123.45.6.8 0 single kal-icmp</pre>
9.	<p>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 <i>Cisco Content Services Switch Content Load-Balancing Configuration Guide</i>.</p>

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 ip_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 reprobng.

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 (0.10×40) to 90% of the current metric value (0.90×50) 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.

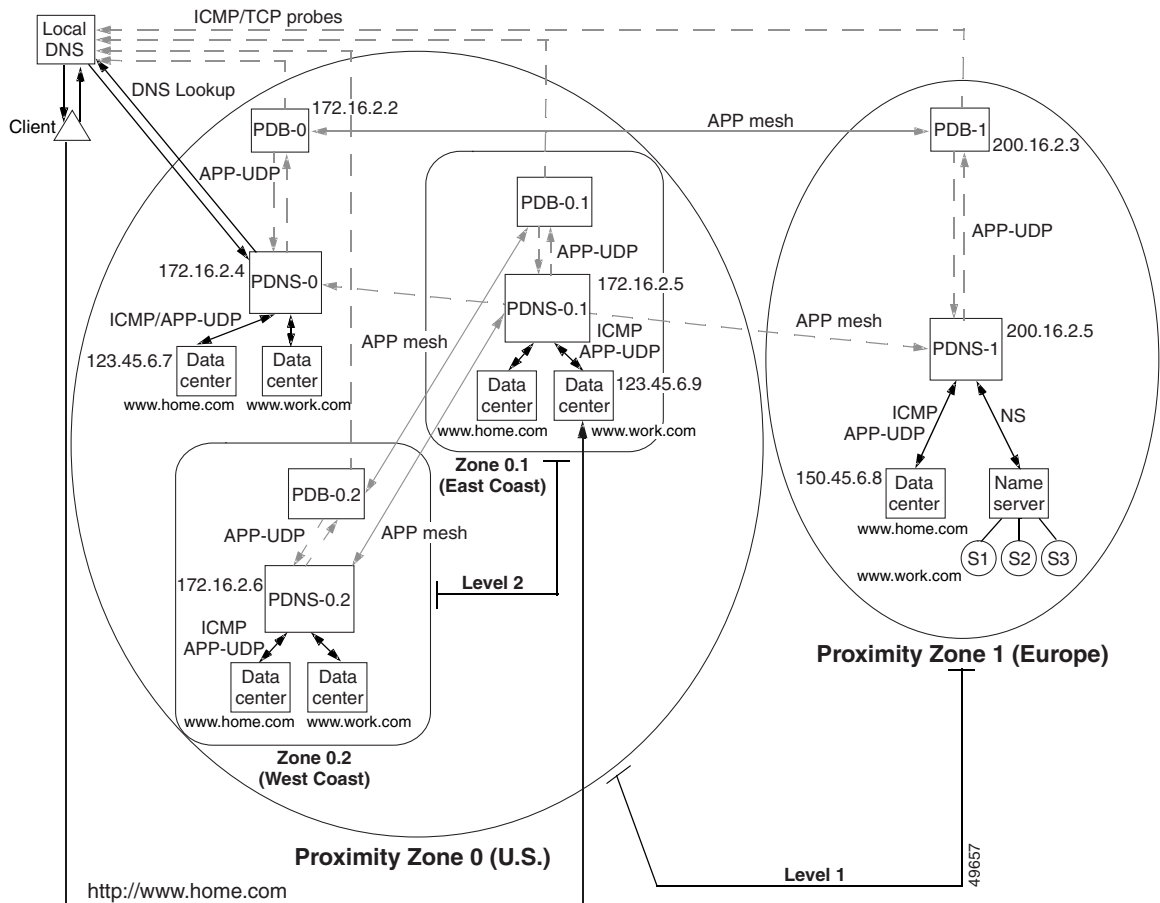
The following steps describe how Network Proximity determines the most proximate service for a client requesting the domain *www.work.com*. See [Figure 5-4](#).

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.

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](#)
- [Displaying Proximity Metrics](#)
- [Displaying Proximity Statistics](#)
- [Displaying Proximity Refinement](#)
- [Displaying Proximity Assignments](#)
- [Displaying Proximity Zones](#)
- [Displaying Proximity Zone Statistics](#)
- [Displaying Proximity Probe Module Statistics](#)

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: <ul style="list-style-type: none"> • 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_prefixall
```



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](#)
- [Displaying DNS Record Statistics](#)
- [Displaying DNS Record Keepalives](#)
- [Displaying DNS Server Zones](#)
- [Displaying DNS Record Proximity](#)
- [Displaying DNS Server Information](#)

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 {allip_address ip_prefix}
```

The **show proximity cache** 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](#).



A

accelerated domain [4-15](#)

address records. See A-records

APP

configurations, displaying [1-13](#)

configuring [1-9](#)

frame size [1-9](#)

overview [1-4](#)

port [1-10](#)

Proximity Database [5-21](#)

Proximity Domain Name Server [5-47](#)

session between two CSSs [1-10](#)

using with Network Proximity [5-16](#)

Application Peering Protocol. See APP

Application Peering Protocol-User Datagram Protocol. See APP-UDP

APP-UDP

configurations, displaying [5-20](#)

configuring [5-16](#)

enabling [5-17](#)

options, configuring [5-18](#)

options, removing [5-19](#)

port [5-19](#)

Proximity Database [5-21](#)

Proximity Domain Name Server [5-47](#)

security [5-17](#)

A-records [1-26](#)

audience [xvi](#)

B

boomerang [3-2](#)

buffer count, DNS server [1-20, 1-44](#)

C

cache

domain, for Client Side Accelerator [4-13](#)

PDNS lookup [5-50, 5-51, 5-52](#)

Client Side Accelerator

configuration, displaying [4-16](#)

configuration quick start [4-7](#)

configuring [4-1, 4-12](#)

disabling [4-13](#)

DNS server forwarder [4-14](#)

DNS server zones [4-16](#)

domain cache [4-13](#)

domain cache statistics, displaying [4-18](#)

enabling [4-12](#)

- information, displaying [4-16](#)
- overview [4-2](#)
- running-config example [4-10](#)
- configuration quick start
 - Client Side Accelerator [4-7](#)
 - Content Routing Agent [3-4](#)
 - DNS, content rule-based [1-39](#)
 - DNS, zone-based [1-17](#)
 - DNS Sticky [2-5](#)
 - Network Proximity [5-12](#)
 - Proximity Database [5-12](#)
 - Proximity Domain Name Server [5-13](#)
- content
 - domain, creating using APP session [1-10](#)
 - router [3-2](#)
- Content Routing Agent
 - configuration quick start [3-4](#)
 - configuring [3-5](#)
 - CPU load threshold [3-5](#)
 - disabling [3-5](#)
 - displaying statistics [3-10](#)
 - domain alias [3-9](#)
 - domain records [1-34, 3-6](#)
 - domain statistics, clearing [3-9](#)
 - enabling [3-5](#)
 - example [3-3](#)
 - overview [3-2](#)
 - running-config example [3-5](#)

- CRA. See Content Routing Agent
- CSA. See Client Side Accelerator

D

- database
 - global sticky [2-7, 2-11, 2-14](#)
 - proximity [2-4, 2-14, 5-5, 5-12, 5-15, 5-21, 5-37](#)
- disaster recovery [1-35](#)
- DNS
 - Client Side Accelerator [4-2](#)
 - configuration quick start, content rule-based [1-39](#)
 - configuration quick start, zone-based [1-17](#)
 - content domain [1-2](#)
 - Content Routing Agent [3-2](#)
 - content rule-based [1-38, 2-13](#)
 - converting content rule-based to zone-based [2-13](#)
 - owner [1-41](#)
 - peer interval [1-42](#)
 - peer receive slots [1-42](#)
 - peer send slots [1-43](#)
 - proximity record statistics, displaying [5-54](#)
 - records, configuring [1-26](#)
 - record statistics, resetting [4-16](#)
 - removing from content rule [1-34, 1-47](#)
 - running config example, content rule-based [1-41](#)
 - running-config example, zone-based [1-19](#)

- server forwarder [1-21, 1-45, 4-14](#)
 - server zones [1-22, 4-16](#)
 - service, adding to content rule [1-34, 1-46](#)
 - sticky [2-1](#)
 - weighted roundrobin [1-23, 1-26, 1-31, 1-32, 1-57](#)
 - zone-based [1-22, 2-13](#)
- DNS peer
- CSS, configuring as [1-42](#)
 - information, displaying [1-58](#)
- DNS records
- address (A) [1-26](#)
 - displaying information [1-54](#)
 - keepalive [1-28](#)
 - name server (NS) [1-26](#)
 - removing [1-33](#)
 - resetting statistics [1-33](#)
 - zero weight [1-35](#)
- DNS server
- authoritative [1-16, 1-19](#)
 - buffer count [1-20, 1-44](#)
 - configuration, displaying [1-48](#)
 - database information, displaying [1-50](#)
 - domain records [1-33, 1-54](#)
 - domain statistics, displaying [1-51](#)
 - forwarder [1-21, 1-29, 1-45](#)
 - forwarder statistics, displaying [1-51](#)
 - peer interval [1-42](#)
 - responder task count [1-22, 1-46](#)
 - server and zone information, displaying [1-47, 1-48](#)
 - zone [1-22, 1-25, 1-53, 4-16](#)
- DNS Sticky
- configuration quick start [2-5](#)
 - converting content rule-based DNS to zone-based [2-13](#)
 - displaying statistics [2-18](#)
 - domain load statistics [2-21](#)
 - domain records [1-29, 2-17](#)
 - domain record statistics, displaying [2-20](#)
 - Global Sticky Database [2-14](#)
 - interface for GSDB [2-15](#)
 - overview [2-2](#)
 - running-config example for DNS server [2-10](#)
 - running-config example for Global Sticky Database [2-8](#)
 - running-config example without a GSDB [2-6](#)
 - TTL for GSDB [2-17](#)
 - with a GSDB [2-3](#)
 - with Network Proximity [2-4](#)
 - without GSDB [2-3](#)
- documentation
- audience [xvi](#)
 - chapter contents [xvi](#)
 - set [xvii](#)
 - symbols and conventions [xx](#)
- domain
- accelerated [4-15](#)
 - cache [4-13, 4-18](#)
 - content [1-10](#)

load statistics [2-21](#)
 name system, overview [1-2](#)
 records [1-26, 1-33, 1-54, 5-49](#)
 statistics, displaying [1-51](#)
 summary information, displaying [1-59](#)

Domain Name System. See DNS

domain records

address (A) [1-26](#)
 displaying information [1-54](#)
 keepalive [1-28](#)
 name server (NS) [1-26](#)
 removing [1-33](#)
 resetting statistics [1-33](#)
 zero weight [1-35](#)

E

example

Network Proximity, operation [5-9](#)
 Network Proximity tiers [5-33](#)
 running-config for Client Side
 Accelerator [4-10](#)
 running-config for Content Routing
 Agent [3-5](#)
 running-config for content rule-based
 DNS [1-41](#)
 running-config for DNS Sticky server [2-10](#)
 running-config for DNS Sticky without a
 GSDB [2-6](#)

running-config for Global Sticky
 Database [2-8](#)
 running-config for zone-based DNS [1-19](#)

F

forwarder

DNS server [1-21, 1-29, 1-45, 4-14](#)
 statistics, displaying [1-51](#)
 frame size, configuring for APP [1-9](#)

G

Global Sticky Database

configuration quick start [2-7](#)
 enabling [2-14](#)
 interface, configuring [2-15](#)
 interface statistics, displaying [2-19](#)
 interface statistics, resetting [2-16](#)
 metrics [2-22](#)
 statistics, displaying [2-18](#)
 statistics, resetting [2-15](#)
 TTL for entries [2-17](#)

GSDB. See Global Sticky Database

K

keepalive

kal-ap [1-28](#)
 kal-icmp [1-28](#)

L

license key

- Enhanced feature set [5-2](#)
- Proximity Database [5-2](#)

load balancing

- DNS records [1-23](#)
- weighted roundrobin [1-23, 1-26, 1-31, 1-32, 1-57](#)

lookup cache

- displaying statistics [5-52](#)
- enabling [5-50](#)
- PDNS [5-52](#)
- removing entries [5-51](#)

Mmesh, peer [5-8](#)metrics, assigning proximity [5-22](#)**N**

name server records. see NS-records

Network Proximity

- APP [5-16](#)
- APP-UDP [5-16](#)
- configuration quick start [5-12](#)
- example [5-9, 5-33](#)
- license keys [5-2](#)
- overview [5-1, 5-3](#)

peer mesh [5-8](#)Proximity Database [5-5, 5-12, 5-15](#)Proximity Domain Name Server [5-6, 5-13](#)tiers [5-33](#)zones [5-7, 5-47, 5-54](#)NS-records [1-26](#)**O**owner, DNS exchange policy [1-41](#)**P**

PDB. See Proximity Database

PDNS. See Proximity Domain Name Server

peer

- interval, configuring for DNS [1-42](#)
- mesh [5-8](#)
- receive slots, configuring for DNS [1-42](#)
- send slots, configuring for DNS [1-43](#)

peering protocol, overview [1-4](#)

probe module

- ICMP delay interval [5-31](#)
- ICMP requests [5-30](#)
- methods [5-29](#)
- metric weighting [5-30](#)
- statistics [5-45](#)
- TCP ports [5-32](#)

probes, resending proximity [5-28](#)

proximity. See Network Proximity

Proximity Database

- activity, displaying [5-37](#)
- archiving [5-25](#)
- assignments, displaying [5-42](#)
- assignments, flushing [5-23](#)
- clearing [5-28](#)
- configuration quick start [5-12](#)
- configuring [5-15](#)
- DNS Sticky [2-4, 2-14](#)
- enabling [5-21](#)
- IP address [1-24](#)
- metrics, assigning [5-22](#)
- metrics, displaying [5-38](#)
- metrics, refining [5-27](#)
- overview [5-5](#)
- probe module [5-29](#)
- probe module statistics, displaying [5-45](#)
- refinement, displaying [5-41](#)
- reprobing [5-28](#)
- retrieving [5-26](#)
- statistics, displaying [5-40](#)
- TTL, configuring [5-24](#)
- zone statistics [5-43, 5-44](#)

Proximity Domain Name Server

- APP [5-47](#)
- APP-UDP [5-47](#)
- A-records [1-26](#)
- cache [5-24](#)

- configuration overview [5-46](#)
- configuration quick start [5-13](#)
- configurations, displaying [5-52](#)
- disabling [5-49](#)
- DNS-record keepalives, displaying [5-54](#)
- DNS-record proximity statistics, displaying [5-54](#)
- DNS-record statistics, displaying [5-54](#)
- DNS server information, displaying [5-55](#)
- DNS server statistics, clearing [5-50](#)
- DNS Sticky [2-4](#)
- domain records [1-33, 1-54, 5-49](#)
- enabling [1-22, 1-44, 5-47](#)
- lookup cache [5-50, 5-51, 5-52](#)
- NS-records [1-26](#)
- overview [5-6](#)
- zones, displaying [5-54](#)

Q

quick start

- Client Side Accelerator [4-7](#)
- Content Routing Agent [3-4](#)
- DNS, content rule-based [1-39](#)
- DNS, zone-based [1-17](#)
- DNS Sticky [2-5](#)
- Network Proximity [5-12](#)
- Proximity Database [5-12](#)
- Proximity Domain Name Server [5-13](#)

R

RCMD command [1-12](#)

records

address (A) [1-26](#)

DNS Sticky [1-29](#)

keepalive [1-28](#)

name server (NS) [1-26](#)

removing [1-33](#)

statistics [1-54](#)

statistics, resetting [1-33, 4-16](#)

weight, configuring [1-32](#)

weight, displaying [1-57](#)

roundrobin, DNS weighted [1-23, 1-26, 1-31, 1-32, 1-57](#)

round-trip time. See RTT

RTT [5-3, 5-45](#)

running-config example

Client Side Accelerator [4-10](#)

Content Routing Agent [3-5](#)

DNS, content rule-based [1-41](#)

DNS, zone-based [1-19](#)

DNS Sticky server [2-10](#)

DNS Sticky without a GSDB [2-6](#)

Global Sticky Database [2-8](#)

S

sticky

DNS [2-1](#)

domain records [1-29](#)

T

TCP port number, configuring for APP [1-10](#)

tiers

example [5-33](#)

Network Proximity [5-33, 5-47](#)

TTL, proximity [5-24](#)

W

weight

configuring DNS record [1-32](#)

configuring zone default [1-24](#)

displaying DNS record [1-57](#)

weighted roundrobin, DNS [1-23, 1-26, 1-31, 1-32, 1-57](#)

Z

zero-weighted domain records [1-35](#)

zones

- Client Side Accelerator [4-16](#)

- displaying data [5-43](#)

- DNS server [1-22](#)

- DNS server load [1-25](#)

- information, displaying [1-53](#)

- Network Proximity [5-7, 5-47, 5-54](#)

- proximity statistics, displaying [5-44](#)

- zone transfer, unsupported among DNS servers [1-3](#)