

Oracle® Communications Essentials Guide



Release S-Cz9.3.0

F92221-06

April 2025

ORACLE®

Copyright © 2024, 2025, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

About this guide

Revision History

1	Introduction to Subscriber Aware Load Balancer S-Cz9.3.0	
	Platform and Public Cloud Support	1-1
	Virtual Machine Platform Requirements	1-4
	Image Files and Boot Files	1-6
	Upgrade Information	1-6
	Upgrade Checklist	1-7
	Coproduct Support	1-7
	Patches Included in This Release	1-8
	New Features	1-8
	Starting your SLB	1-9
2	Overview	
	Functional Overview	2-1
	Balancing and Rebalancing	2-2
	Balancing	2-2
	OCSBC Overload Conditions Affecting Load Balancing	2-3
	Rebalancing	2-4
	IPv4 IPv6 Dual Stack	2-5
	OCSLB Cluster Member Graceful Shutdown	2-6
	High-level Procedure for Graceful OCSBC Shutdown	2-7
	Detailed Description of Graceful Shutdowns with Active SIP Calls or Registrations	2-7
	Georedundant High Availability (HA)	2-8
3	Subscriber-Aware Load Balancer Configuration	
	SLB Configuration	3-1
	Core Configuration on the OCSLB	3-1

SLB Tunnel Configuration	3-3
Sample SLB Tunnel Configuration	3-4
Cluster Configuration	3-4
Sample Cluster Configuration	3-9
Service Ports Configuration	3-9
Sample Service Port Configuration	3-10
Traffic Policy Configuration	3-11
Sample Traffic Policy Configuration	3-12
Load Balancer Policy Configuration	3-12
Sample Load Balancer Policy Configuration	3-15
Distribution Policy Configuration	3-16
Sample Distribution Rule Configurations	3-20
Forced Rebalance	3-21
OCSBC Configuration	3-22
OCSBC Tunnel Configuration	3-22
Sample OCSBC Tunnel Configuration	3-24
SIP Configuration	3-25
Online Offline Configuration	3-27
IMS-AKA and TLS Support	3-27
OCSLB Configuration for IMS-AKA and TLS Traffic	3-28
OCSBC Configuration for IMS-AKA Traffic	3-28
Displaying Encrypted Traffic Detail on the OCSBC	3-29
SLB VLAN Configuration	3-30
OCSBC Configuration for VLANs	3-32
OCSLB Configuration for VLANs	3-34
Sample VLAN Configuration	3-35

4 OCSLB/Cluster Management & Diagnostics

OCSLB Statistics	4-1
Display Behavior for Show Commands with the Size Argument	4-1
show balancer	4-1
show balancer endpoints	4-2
show balancer members	4-3
show balancer metrics	4-4
show balancer realms	4-5
show balancer statistics	4-5
show balancer tunnels	4-7
Cluster Control Protocol Statistics	4-10
show ccd	4-10
show ccd ccp	4-10
show ccd sds	4-13

show ccd stats	4-16
show ccd rebalance	4-18
OCSBC Cluster Member Statistics	4-19
show sip lb-endpoints	4-19
show sip ccp	4-21

5 Subscriber-Aware Load Balancer SNMP Reference

Overview	5-1
Enterprise Traps	5-1
License MIB (ap-license.mib)	5-1
Subscriber-Aware Load Balancer MIB (ap-slb.mib)	5-1

6 Known Issues and Caveats

Known Issues for Release S-Cz9.3.0	6-1
Caveats for Release S-Cz9.3.0	6-2

About this guide

This guide is written for network administrators and architects, and provides information about the SLB configuration. For information on configuration and operation of the Oracle Communications Session Border Controller (OCSBC) and Oracle Enterprise Session Border Controller (OESBC) as SLB cluster members, refer to the Release Notes for those products' release versions.

**Note:**

This document refers to the OCSBC as cluster members throughout. Nevertheless, you use the same functionality, operation, and configuration to run the OCSLB with the OESBC as you use with the OCSBC.

Related Documentation

The following table describes the documentation set for this release.

Document Name	Document Description
Release Notes	Contains information about the current documentation set release, including new features and management changes.
Configuration Guide	Contains information about the administration and software configuration of the Service Provider Oracle Communications Subscriber-Aware Load Balancer.
ACLI Reference Guide	Contains explanations of how to use the ACLI, as an alphabetical listings and descriptions of all ACLI commands and configuration parameters.
Maintenance and Troubleshooting Guide	Contains information about Oracle Communications Subscriber-Aware Load Balancer logs, performance announcements, system management, inventory management, upgrades, working with configurations, and managing backups and archives.
MIB Guide	Contains information about Management Information Base (MIBs), Oracle Communication's enterprise MIBs, general trap information, including specific details about standard traps and enterprise traps, Simple Network Management Protocol (SNMP) GET query information (including standard and enterprise SNMP GET query names, object identifier names and numbers, and descriptions), examples of scalar and table objects.
Accounting Guide	Contains information about the Oracle Communications Subscriber-Aware Load Balancer's accounting support, including details about RADIUS and Diameter accounting.

Document Name	Document Description
Admin Security Guide	Contains information about the Oracle Communications Subscriber-Aware Load Balancer's support for its Administrative Security license.
Security Guide	Contains information about security considerations and best practices from a network and application security perspective for the Oracle Communications Subscriber-Aware Load Balancer family of products.
Platform Preparation and Installation Guide	Contains information about upgrading system images and any pre-boot system provisioning.
Call Traffic Monitoring Guide	Contains information about traffic monitoring and packet traces as collected on the system. This guide also includes WebGUI configuration used for the SIP Monitor and Trace application.
HMR Guide	Contains information about configuring and using Header Manipulation Rules to manage service traffic.

Revision History

This section provides a revision history for this document.

Date	Description
March 2024	Initial release.
October 2024	Corrects value description for throttle-rate in traffic policy.
February 2025	Adds features for S-Cz9.3.0p5.
April 2025	Adds caveat on check-upgrade-readiness command.

1

Introduction to Subscriber Aware Load Balancer S-Cz9.3.0

This Oracle Communications Subscriber-Aware Load Balancer (SLB) release introduction chapter provides the following information about this product:

- Supported platforms and hardware requirements
- Version detail
- Upgrade information

A summary of known issues and caveats is presented in the Known Issues Appendix in this SLB Essentials Guide.

Platform and Public Cloud Support

The Oracle Communications Subscriber-Aware Load Balancer (SLB) can be run on the platforms listed in this section.



Note:

The SLB does not support automatic, dynamic disk resizing.



Note:

Virtual SLBs do not support media interfaces when media interfaces of different NIC models are attached. Media Interfaces are supported only when all media interfaces are of the same model, belong to the same Ethernet Controller, and have the same PCI Vendor ID and Device ID.

Supported Hypervisors

The SLB supports the following hypervisors for version S-Cz9.3.0:

- KVM (the following versions or later)
 - Linux kernel version: 3.10.0-1127
 - Library: libvirt 4.5.0
 - API: QEMU 4.5.0
 - Hypervisor: QEMU 1.5.3
- VMware: vSphere ESXi (Version 7.0 or later)

Supported Public Cloud Platforms

You can run the S-Cz9.3.0 SLB over the following public cloud platforms.

- Oracle Cloud Infrastructure (OCI)
OCI Cloud Shapes and options supported by this release are listed below. After deployment, you can change the shape of your machine by, for example, adding disks and interfaces.

Shape	OCPUs/ VCPUs	vNICs	Tx/Rx Queues	Max Forwarding Cores	DoS Protection	Memory
VM.Standard2.4	4/8	4	2	2	Y	60
VM.Standard2.8	8/16	8	2	2	Y	120
VM.Optimized3. Flex-Small	4/8	4	8	6 ¹	Y	16
VM.Optimized3. Flex-Medium	8/16	8	15	14 ²	Y	32
VM.Optimized3. Flex-Large	16/32	16	15	15	Y	64

¹ This maximum is 5 when using DoS Protection

² This maximum is 13 when using DoS Protection

Networking using image mode [SR-IOV mode - Native] is supported on OCI. PV and Emulated modes are not currently supported.

 Note:

Although the VM.Optimized3.Flex OCI shape is flexible, allowing you to choose from 1-18 OCPUs and 1-256GB of memory, the vSBC requires a minimum of 4 OCPUs and 16GB of memory per instance on these Flex shapes.

- Amazon Web Services (EC2)
This table lists the AWS instance sizes that apply to the SLB.

Instance Type	vCPUs	Memory (GB)	Max NICs
c5.xlarge*	4	8	4
c5.2xlarge	8	16	4
c5.4xlarge	16	32	8
c5.9xlarge	36	72	8
c5.12xlarge	48	96	8
c5.18xlarge	72	144	15
c5n.xlarge	4	10.5	4
c5n.2xlarge	8	21	4
c5n.4xlarge	16	42	8
c5n.9xlarge	36	96	8
c5n.18xlarge	72	192	15

* — Hyperthreading must be enabled for this shape.

 Note:

C5 instances use the Nitro hypervisor.

 Note:

ENA is supported on the C5/C5n family.

- Google Cloud Platform
The following table lists the GCP instance sizes that you can use for the SLB.

Table 1-1 GCP Machine Types

Machine Type	vCPUs	Memory (GB)	vNICs	Egress Bandwidth (Gbps)	Max Tx/Rx queues per VM
n2-standard-4	4	16	4	10	4
n2-standard-8	8	32	8	16	8
n2-standard-16	16	64	8	32	16

Use the n2-standard-4 machine type if you're deploying an SLB that requires one management interface and only two or three media interfaces. Otherwise, use the n2-standard-8 or n2-standard-16 machine types for an SLB that requires one management interface and four media interfaces. Also use the n2-standard-8 or n2-standard-16 machine types if deploying the SLB in HA mode.

Before deploying your SLB, check the [Available regions and zones](#) to confirm that your region and zone support N2 shapes.

On GCP the SLB must use the **virtio** network interface card. The SLB will not work with the GVNIC

OpenStack Compatibility

Oracle distributes Heat templates for the Newton and Pike versions of OpenStack. Download the source, [nnSCZ930_HOT.tar.gz](#), and follow the [OpenStack Heat Template](#) instructions.

The nnSCZ930_HOT.tar.gz file contains two files:

- nnSCZ930_HOT_pike.tar
- nnSCZ930_HOT_newton.tar

Use the Newton template when running either the Newton or Ocata versions of OpenStack. Use the Pike template when running Pike or a later version of OpenStack.

Platform Hyperthreading Support

Of the supported hypervisors, only VMware does not expose SMT capability to the SLB. OCI, AWS and GCP enable SMT by default and expose it to the SLB.

DPDK Reference

The SLB relies on DPDK for packet processing and related functions. You may reference the Tested Platforms section of the DPDK release notes available at <https://doc.dpdk.org>. This information can be used in conjunction with this Release Notes document for you to set a baseline of:

- CPU
- Host OS and version
- NIC driver and version

- NIC firmware version



Note:

Oracle only qualifies a specific subset of platforms. Not all the hardware listed as supported by DPDK is enabled and supported in this software.

The DPDK version used in this release is:

- 22.11

Virtual Machine Platform Requirements

A Virtual Network Function (VNF) requires the CPU core, memory, disk size, and network interfaces specified for operation. Deployment details, such as the use of distributed DoS protection, dictate resource utilization beyond the defaults. .

Default vSBC Resources

The default compute for the SLB image files is as follows:

- 8 vCPU Cores
- 32 GB RAM
- 40 GB hard disk (pre-formatted)
- 8 interfaces as follows:
 - 1 for management (wancom0)
 - 2 for HA (wancom1 and 2)
 - 1 spare
 - 4 for media

Minimum VNF Resources

VM resource configuration defaults to the following:

- 4 vCPU Cores
- 8 GB RAM
- 20 GB hard disk (pre-formatted)
- 4 interfaces as follows:
 - 1 for management (wancom0)
 - 1 for HA (wancom1)
 - 2 for media

Interface Host Mode

The SLB S-Cz9.3.0 VNF supports interface architectures using Hardware Virtualization Mode - Paravirtualized (HVM-PV):

- ESXi - No manual configuration required.

- KVM - HVM mode is enabled by default. Specifying PV as the interface type results in HVM plus PV.

Supported Interface Input-Output Modes

- Para-virtualized
- SR-IOV
- PCI Passthrough

Supported Ethernet Controller, Driver, and Traffic Type based on Input-Output Modes

The following table lists supported Ethernet Controllers (chipset families) and their supported driver that Oracle supports for Virtual Machine deployments. Reference the host hardware specifications, where you run your hypervisor, to learn the Ethernet controller in use. The second table provides parallel information for virtual interface support. Refer to the separate platform benchmark report for example system-as-qualified performance data.



Note:

Virtual SBCs do not support media interfaces when media interfaces of different NIC models are attached. Media Interfaces are supported only when all media interfaces are of the same model, belong to the same Ethernet Controller, and have the same PCI Vendor ID and Device ID.

For KVM and VMware, accelerated media/signaling using SR-IOV and PCI-pt modes are supported for the following card types.

Ethernet Controller	Driver	SR-IOV	PCI Passthrough
Intel 82599 / X520 / X540	ixgbe	M	M
Intel i210 / i350	igb	M	M
Intel X710 / XL710	i40e	M	M
Intel X710 / XL710 / XXC710	i40en ¹	M	M
Mellanox Connect X-4	mlx5	M	M

¹ This driver is not supported on KVM.

For PV mode (default, all supported hypervisors), the following virtual network interface types are supported. You can use any make/model NIC card on the host as long as the hypervisor presents it to the VM as one of these vNIC types.

Virtual Network Interface	Driver	W/M
KVM (PV)	virtio	W/M
VMware (PV)	VMXNET3	W/M

- W - wancom (management) interface
- M - media interface

**Note:**

Accelerated media/signaling using SR-IOV (VF) or PCI-pt (DDA) modes are not currently supported for Hyper-V when running on Private Virtual Infrastructures.

CPU Core Resources

The SLB S-Cz9.3.0 VNF requires an Intel Core7 processor or higher, or a fully emulated equivalent including 64-bit SSSE3 and SSE4.2 support .

If the hypervisor uses CPU emulation (for example, qemu), Oracle recommends that you set the deployment to pass the full set of host CPU features to the VM.

Image Files and Boot Files

For Virtual Machines

The SLB S-Cz9.3.0 version includes distributions suited for deployment over hypervisors. Download packages contain virtual machine templates for a range of virtual architectures. Use the following distributions to deploy the SLB as a virtual machine:

- `nnSCZ930-img-vm_kvm.tgz`—Compressed image file including SLB VNF for KVM virtual machines, Oracle Cloud Infrastructure (OCI), and AWS EC2 and C5 instances.
- `nnSCZ930-img-vm_vmware.ova`—Open Virtualization Archive (.ova) distribution of the SLB VNF for ESXi virtual machines.

The KVM, and ESXi packages include:

- **Product software**—Bootable image of the product allowing startup and operation as a virtual machine. This disk image is in either the vmdk (for VMware) or qcow2 (for KVM) format.
- **OVF File**—XML descriptor information containing metadata for the overall package, including identification, and default virtual machine resource requirements. The .ovf file format is specific to the supported hypervisor.
- `legal.txt` (KVM only)—Licensing information, including the Oracle End-User license agreement (EULA) terms covering the use of this software, and third-party license notifications.

Upgrade Information

This section provides key information about upgrading to this software version.

Supported Upgrade Paths

The following in-service (hitless) upgrade/rollback paths are supported by the SLB. Always start the upgrade process with the latest patch version of your current release:

- S-Cz9.1.0p10 (or higher) to S-Cz9.3.0
- S-Cz9.2.0p4 (or higher) to S-Cz9.3.0

Endpoint Capacity Configuration during Upgrade

If you are upgrading from S-Cz8.1.0, S-Cz8.3.0 or S-Cz8.4.0 to S-Cz9.0.0, S-Cz9.1.0 or S-Cz9.3.0, ensure, as a step prior to upgrading, that the endpoint capacity configuration in **setup entitlements, LB Endpoint Capacity** is not greater than 5 million before upgrade. 5 million endpoints is the maximum for this entitlement. The upgrade may fail if this number is greater than 5 million.

In addition, during upgrade, the number of endpoints in the entitlement should be readjusted if memory is less than 32GB. Specifically:

- If memory is between 16GB to 32GB, then the total number of endpoints in entitlements should be less than or equal to 2.5 Million endpoints.
- If memory is less than 16GB, then the total number of endpoints in entitlements should be less than or equal to 1 Million endpoints

Upgrade Checklist

Before upgrading the Oracle Communications Subscriber-Aware Load Balancer software:

1. Obtain the name and location of the target software image file from either Oracle Software Delivery Cloud, <https://edelivery.oracle.com/>, or My Oracle Support, <https://support.oracle.com>, as applicable.
2. Provision platforms with the Oracle Communications Subscriber-Aware Load Balancer image file in the boot parameters.
3. Run the **check-upgrade-readiness** command and examine its output for any recommendations or requirements prior to upgrade.
4. Verify the integrity of your configuration using the ACLI **verify-config** command.
5. Back up a well-working configuration. Name the file descriptively so you can fall back to this configuration easily.
6. Refer to the Oracle Communications Subscriber-Aware Load Balancer Release Notes for any caveats involving software upgrades.

Coproduct Support

The products/features listed in this section run in concert with the Oracle Communications Subscriber-Aware Load Balancer (SLB) for their respective solutions.

Oracle Communications Subscriber-Aware Load Balancer

This release of the SLB supports both the OCSBC and the OESBC as follows:

- The S-Cz9.1.0 release on the Acme Packet 4600, 6100, 6300, and 6350 and VNF platforms as cluster members.
- The S-Cz9.2.0 release on the Acme Packet 4600, 6100, 6300, and 6350 and VNF platforms as cluster members.
- The S-Cz9.3.0 release on the Acme Packet 4600, 6100, 6300, and 6350 and VNF platforms as cluster members.

Oracle Communications Session Delivery Manager

This S-Cz9.3.0 SLB release can interoperate with the following versions of the Oracle Communications Session Delivery Manager:

- 9.0 and higher



Note:

To manage S-Cz9.3.0 patches in conjunction with Oracle's Session Delivery Manager, review the build notes to determine if an XSD file is required and review the readme file in the XSD file. XSD files may work with older SDM releases, though it is not guaranteed.

Oracle Communications Session Delivery Manager Cloud

This S-Cz9.3.0 SLB release can interoperate with the following versions of the Oracle Communications Session Delivery Manager Cloud:

- 24.1

Oracle Communications Operations Manager

This S-Cz9.3.0 SLB release is not supported by OCOM.

Patches Included in This Release

The following information assures you that when upgrading, the S-Cz9.3.0 release includes defect fixes from neighboring patch releases.

Neighboring Patches Included

- S-Cz900p10
- S-Cz910p9
- S-Cz920p4

New Features

The Oracle Communications Subscriber-Aware Load Balancer (SLB) S-Cz9.3.0 release provides system and performance enhancements over prior release versions. This release also supports the following new features.

VLANs on the SLB

You can configure the SLB to use overlapping IP addressing between the access network and SBC clusters, supported by VLANs. When configured, the SLB includes VLANs within its ingress and egress traffic matching processes for policy-based load balancing. This allows it to operate within an access network that uses VLANs. These deployments also require applicable configuration on SBC cluster members. This feature is supported for both IPv4 and IPv6 addressing.

This support begins with S-Cz9.3.0p5.

See the VLAN Support section of this *Oracle Communications Subscriber-Aware Load Balancer* for information confirming this feature.

Starting your SLB

Oracle Communications Subscriber-Aware Load Balancer (SLB) is supported on the platforms listed for this version. Find installation information about those platforms in this Session Delivery Product version's *Platform Preparation and Installation Guide*. The SLB also requires that you take several steps after installation before you can configure the product.

Critical steps you cannot bypass before configuration include:

- Setting system passwords

Note:

For the OCI platform, system passwords are unique, and generated by OCI for VMs. These passwords are a combination of the initial acme/packet passwords, plus a unique OCI ID. See the *Set the User and Administrative Passwords on the OCSBC* section under the *Create and Deploy on OCI* documentation in the *Platform Preparation and Installation Guide* for further information about these passwords. That information applies equally to the SLB.

- Setup product
- Setup entitlements (Reboot required after all entitlement changes)

Prior to configuring your SLB for service, we recommend that you review the information and procedures in the Getting Started chapter of this software version's *ACLI Configuration Guide*. This chapter offers information that help you:

- Review hardware installation procedures
- Connect to your SLB using a console connection or SSH (secure shell)
- Become familiar with the SLB's boot parameters and how to change them if needed
- Set up product-type, features, and functionality
- Load and activate a SLB software image
- Choose a configuration mechanism: ACLI, Oracle Communications Session Element Manager or ACP/XML
- Enable RADIUS authentication
- Customize your login banner

2

Overview

As service providers deploy larger and larger SIP access networks, scalability problems are presenting unique challenges, particularly from an operational standpoint. Deployments that scale beyond the number of users serviceable by a single Oracle Communications Session Border Controller (OCSBC) – as well as deployments that use a geographically redundant OCSBC for catastrophic fail over purposes – encounter edge reachability problems. In general there are two coarse techniques that carriers use today to support end-point populations that exceed one Oracle Communications Session Border Controller capacity: they either use a DNS-based distribution mechanism, or they will pre-provision endpoint to point to specific OCSBCs (manually load balancing them). Each of these solutions has its drawbacks. End users – many of them familiar with load balancing equipment deployed to scale protocols such as HTTP or SMTP – have expressed interest in a device that will perform dedicated load balancing for their SIP endpoint.

The SLB addresses the need for scaling a network edge to millions of endpoint. Designed as a standalone system, the network architect can deploy a physical or virtual SLB capable of supporting an extremely large number of endpoints, where an endpoint is defined as a unique source and destination IP address. The SLB aggregates signaling from large endpoint populations to reduce the edge reachability problems.

The network architect reduces this problem by deploying clusters of OCSBCs supported by the SLB. These OCSBCs can be operating as either Physical Network Functions (PNFs) and Virtual Network Functions (VNFs). The SLB supports clusters of homogenous or heterogenous groups of PNFs and/or VNFs.

Functional Overview

The Oracle Communications Subscriber-Aware Load Balancer (SLB) is a discrete network element that processes all SIP end-point signaling traffic entering the service provider network. The SLB is not necessarily the first network device to receive signaling traffic, as, depending on network topology, additional network components (for example, routers, network address translators, and so on) can lie between the end-point and the SLB.

Upon receipt of a SIP packet from an unknown source, the SLB uses a provisioned policy to select an appropriate next-hop Oracle Communications Session Border Controller (OCSBC) for traffic originated by that end-point. Subsequent packets from the same end-point are forwarded to the same OCSBC. The first packet, the one used to make the route decision, and all subsequent packets sent through the SLB to the next-hop OCSBC are encapsulated within an IP-in-IP format as defined in RFC 2003, IP Encapsulation within IP.

SBCs that participate in the load balancing-enabled deployment are enhanced by several capabilities. First, the OCSBC supports RFC 2003 tunnel for both packet transmission and reception. Second, the OCSBC periodically transmits health and performance data to the SLB; such information is evaluated and entered into the SLB's route determination algorithm. Lastly, the OCSBC participates in any SLB-initiated rebalance operation, as described in the Rebalancing section. A group of OCSBCs, with the above-listed capabilities, that receive signaling traffic from the SLB, is referred to as a cluster.

The IP-in-IP encapsulation technique provides SLB transparency to the terminating OCSBC. That is, when an OCSBC receives an encapsulated packet via the SLB, it can discard the

outer encapsulation leaving behind an identical packet as transmitted originally by the end-point. Visibility into the actual packet transmitted by the end-point is necessary to provide certain services in the OCSBC (for example, hosted NAT traversal, session-agent matching, and so on). A secondary goal achieved by using this encapsulation technique is that it provides a disassociation function between an OCSBC's connected network and its SIP reachability. That is, an OCSBC can be assigned any IP address it wants from a network topology standpoint, yet still process SIP packets as though it were logically situated elsewhere at Layer 5. In a larger sense, the physicality of the OCSBC is no longer important; like-configured, logically identical OCSBCs can be spread all over the globe.

Balancing and Rebalancing

The Oracle Communications Subscriber-Aware Load Balancer (SLB) performs two primary functions as the front-end to a Oracle Communications Session Border Controller (OCSBC) cluster: balancing traffic and rebalancing traffic. There are several key distinctions, which are described in the following two sections.

Balancing

Balancing is the distribution of new endpoints, identified by the combination of unique source and destination address pairs, among the members of the SLB cluster, and sending SIP traffic to the most appropriate member, based on your configuration. You define this balancing by configuring policies to apply to address pair and global balancing configuration. Distribution policies (**lb-policy**) define and direct traffic specifying target realms by address pairs. After identifying a target realm, or for packets that do not match a distribution policy, the SLB calculates the best cluster member for this traffic from the pool of candidates (**lbp-config**). The simplest configuration can skip realm participation, instead distributing traffic in a round-robin fashion across the entire pool of cluster members. OCSBCs can also report themselves as overloaded, which excludes them as candidate targets until they report themselves as available for service again.

The section [Distribution Policy Configuration](#) expands upon address pair and target realm and, ultimately, candidate pool identification. The load balance config (**lbp-config**) specifies how to identify the best OCSBC in the candidate pool. Regardless of the distribution method, policy-based or round-robin, the SLB, by default, weighs each candidate OCSBC, and identifies the one with the lowest current utilization. In addition, the **lbp-config** allows you to specify traffic constraints that control state timing and the amount of traffic managed by each config.

Within the **lbp-config**, you can select from two strategies for weighing target OCSBC utilization:

Least Occupied—This strategy distributes to the OCSBC with the least current endpoint count.

Capacity Proportional—This strategy causes the SLB to maintain a calculation using the maximum occupancy and the current endpoint count on each applicable OCSBC. A higher maximum capacity indicates the ability to support a higher number of endpoints.

The SLB balances traffic distribution by checking status detail on each OCSBC in the cluster, and forwarding to the best target, as follows:

1. If the OCSBC reports itself as overloaded, the SLB skips that OCSBC from consideration.
2. If the traffic matches a policy, the SLB iterates through the OCSBCs that match the policy using either round robin or capacity proportional (referring to the **strategy** configuration).
3. If there is no policy configured, the SLB uses round-robin or capacity-proportional to identify OCSBCs.

- For least-occupied (default) the SLB selects the OCSBC with the fewest endpoints.
 - For capacity-proportional, the SLB selects the OCSBC with the lowest endpoint to maximum capacity ratio.
4. The SLB skips over any OCSBC that has exceeded its maximum SLB-assigned endpoints.
 5. The SLB identifies and forward to the target OCSBC.

Contrasting Least Occupied and Capacity Proportional Strategies

When configured for the least occupied strategy, the SLB effectively balances using round-robin distribution. As shown below, each OCSBC gets an equal number of endpoints.

```
slb-1# show balancer metrics
```

SBC	Name	local epts	remote epts	max reg	CPU	max CPU	max Mem%	Over Mem	Load
1021	SBC-1b	501	501	1000	0.2	90.0	21.0	95.0	no
1022	SBC-1c	501	501	1000	0.1	90.0	21.0	95.0	no
1023	SBC-1a	501	501	1200	0.0	90.0	21.0	95.0	no

In the above scenario, each OCSBC hosts 501 of the total 1503 endpoints based on 'round-robin' policy.

SBC-1a, however, has a set to a higher "max reg" value (1200) than the other OCSBCs. This can occur for a variety of reasons, including maximum capacity and available computing resources. The OCSBC reports this value in every CCP message. When using the capacity proportional approach, the SLB allocates more endpoints to the OCSBCs that have higher maximum registration values. The formula used to weigh the OCSBC is endpoint count (maintained locally), divided by the maximum registration value.

The output below shows how this configuration change would affect the endpoint count.

```
slb-1# show balancer metrics
```

SBC	Name	local epts	remote epts	max reg	CPU	max CPU	max Mem%	Over Mem	Load
1021	SBC-1b	470	470	1000	0.2	90.0	21.0	95.0	no
1022	SBC-1c	470	470	1000	0.1	90.0	21.0	95.0	no
1023	SBC-1a	563	563	1200	0.0	90.0	21.0	95.0	no

In the above scenario, The SLB allocates each OCSBC at 46% of its total limit configured. SBC-1a hosts a larger number endpoints because it supports a higher maximum registration value. Both rebalancing procedures and the exclusion of overloaded OCSBCs are maintained in both least-occupied and capacity-proportional balancing.

OCSBC Overload Conditions Affecting Load Balancing

Each cluster OCSBC will report an overload status to the OCSLB. Factors which contribute to OCSBC overload determination include SIP thread CPU utilization and overall OCSBC memory threshold.

Even though each clustered OCSBC regularly reports CPU data to the SLB, the OCSBC's CPU utilization is not factored into the preference of one OCSBC over another. Rather, an OCSBC whose CPU utilization rate, determined using a per-thread CPU load check of the busiest call-related threads (SIP and MBCD), exceeds its load limit threshold (by default, 90%)

is excluded from the list of candidates. For example, assuming that both OCSBCs are licensed for the same number of sessions, an OCSBC with a CPU load of 89% and a current occupancy of 10,000 endpoint will have equal footing with an OCSBC with a CPU load of 10% and a current occupancy of 10,000 endpoint. But an OCSBC with a CPU load of 90% and an occupancy of 0 endpoint will never receive new assignments from the SLB, until its CPU utilization rate falls below the 90% threshold.

When load-balancing traffic, the Oracle Communications Subscriber-Aware Load Balancer (SLB) skips Oracle Communications Session Border Controllers (OCSBCs) that report overloaded memory or CPU. CPU utilization is measured on a per-thread basis, referring to each SIP and MBCD thread for their resource utilization. Configuring the applicable OCSBC memory utilization threshold requires that the user consider multiple OCSBC settings, explained below.

An OCSBC's memory utilization threshold is the percentage of overall system memory utilization that, when exceeded, triggers the OCSBC to set its overload flag. The OCSBC then tells the SLB it is overloaded via the standard update process. The OCSBC sets this same flag when CPU utilization exceeds its overload threshold. When memory and CPU utilization fall below their thresholds, the OCSBC clears the overload flag.

The **memory-utilization-threshold** in the **system-config** allows the user to explicitly set the memory utilization threshold used for load balancing. During operation, the OCSBC refers to this and two other settings to determine when it notifies the SLB that it is in a memory overload condition. These settings include any user-configured critical memory alarm value and the **system-config, heap-threshold** option setting. The operational process, which effectively determines the lowest of these settings, is as follows:

1. The OCSBC refers to its **memory-utilization-threshold** setting. If set, use that value for the steps below.
2. The OCSBC refers to its alarm configuration. If there is a critical memory alarm value lower than the **memory-utilization-threshold**, the system sets the **memory-utilization-threshold** to that alarm's setting.
3. The OCSBC refers to its **heap-threshold** setting. If lower than the **memory-utilization-threshold** and the alarm setting, the system sets the **memory-utilization-threshold** to the **heap-threshold**'s setting.
4. If the **memory-utilization-threshold** value is lower than the alarm and **heap-threshold**, the OCSBC uses its value.

If none of these values are set explicitly, the OCSBC uses the **heap-threshold** default of 90%.

Values for the system-config's **memory-utilization-threshold** option range from 1% to 95%. The syntax below shows the option set to 75%.

```
ORACLE(system-config)#options +memory-utilization-threshold=75
```

The user can display the OCSBC's running configuration to see these settings.

Rebalancing

Rebalancing, as opposed to balancing, is taking some number of existing endpoints from functioning OCSBCs and redistributing these existing endpoints between current cluster members. Rebalancing can be automatically scheduled when a new OCSBC joins an existing cluster, or immediately invoked with the Acme Packet Command Line Interface (ACLI). When an OCSBC exits a cluster, whatever the reason, all of its endpoints are invalidated on the

Oracle Communications Subscriber-Aware Load Balancer (SLB) and those endpoints are essentially balanced when they revisit the SLB.

A new OCSBC joins an existing cluster by initiating the establishment of an IP-in-IP tunnel between itself and the SLB. During an initial handshake the OCSBC designates which SLB service port or ports it is prepared to support. If there are existing OCSBCs supporting these designated service ports, the SLB instructs some or all of these OCSBCs to divest themselves of a specified number of endpoints. The SLB calculates the number of divested endpoints based upon the overall occupancy of that service relative to the SLB's contribution to that occupancy. Existing cluster members not advertising support for service ports designated by the new cluster member are excluded from the rebalance queue.

The SLB sequences through eligible cluster members one at a time, using a proprietary protocol to request nomination and removal of eligible endpoints. The OCSBC replies with a CCP response that lists candidate endpoints. The SLB removes existing forwarding rules associated with those endpoints, and repeats the CCP request/response process until the cluster member divests itself of the specified number of endpoints.

When the divested endpoints re-engage with the SLB (upon their next scheduled registration refresh, for example), the SLB lacks a forwarding rule that maps them to a specific OCSBC. Consequently, the message is passed up to the software processes running on the SLB's host, which chooses a new OCSBC destination for that endpoint – presumably, the new cluster member that has the most available capacity.

The cluster member, after being requested to nominate endpoints for rebalancing, uses several criteria for choosing the most attractive candidates. As part of its standard SIP processing performed by SBCs, the cluster member is aware of the expiry times for all of the entries in its SIP registration cache. Therefore, the cluster member can predict with a high degree of accuracy when any given endpoint will be signaling back into the cluster. As the forwarding rules on the cluster member are triggered by endpoint messages, the cluster member considers an endpoint whose registration entry is due to expire shortly an attractive candidate for rebalance. Note, however, that in many cases it is not prudent to nominate endpoints whose SIP registration cache entries are due to expire immediately, as this can cause a race condition between the CCP response and the SIP REGISTER message from the endpoint to the SIP registration function. To avoid this potential dilemma, cluster members are equipped with the ability to skip ahead to candidates whose expiry is not immediate.

Further, each cluster member categorizes the endpoints stored in its cache based upon a priority value that is determined via the SLB's distribution policy (see Distribution Policy Configuration for more details). It nominates endpoints from its lowest priority buckets first.

Finally, the SLB does not rebalance an active SIP endpoint — an endpoint engaged in a phone conversation.

After removing endpoints from the first cluster member, the SLB moves to the next cluster member in the rebalance queue and uses the same CCP request/response exchange to remove additional endpoints. The same procedure repeats for additional cluster members until the SLB attains the target number of divested endpoints.

IPv4 IPv6 Dual Stack

While major carriers are proceeding toward a pure IPv6 network for next generation services, current practicalities require the continued support of IPv4 handsets and other devices. As a result, the Oracle Communications Subscriber-Aware Load Balancer provides support for single non-channelized physical interfaces that support both IPv4 and IPv6 ingress and egress on the same network interface.

Support for the dual stack interface requires no new additional configuration elements, and is provided by the proper configuration of the following elements in the *ACLI Configuration Guide*:

Configuration Element	Section containing configuration element in the <i>ACLI Configuration Guide</i> documentation
Physical Interfaces	(Platform) Physical Interfaces: SLB in the System Configuration chapter.
IPv4 Network Interfaces	IPv4 Address Configuration in the System Configuration chapter.
IPv6 Network Interfaces	Configuring Network Interfaces: SLB, Licensing, Globally Enabling IPv6, IPv6 Address Configuration, IPv6 Default Gateway, and Network Interfaces and IPv6 in the System Configuration chapter. 0)
IPv4 & IPv6 SIP Interfaces	SIP Interfaces in the SIP Signaling Services chapter.
IPv4 & IPv6 Realms	Realm Configuration in the Realms and Nested Realms chapter.

OCSLB Cluster Member Graceful Shutdown

When it becomes necessary to temporarily remove an Oracle Communications Session Border Controller (OCSBC) from active service, and make it available only for administrative purposes, the user issues a **set-system-state offline** ACLI command. The OCSBC begins a graceful shutdown. The shutdown is graceful in that active calls and registrations are not affected, but new calls and registrations are rejected except as discussed below. When the user issues the command, the OCSBC goes into **becoming offline** mode. Once there are no active SIP sessions and no active SIP registrations in the system, the OCSBC transitions to **offline** mode. If the OCSBC is a member of an Oracle Communications Subscriber-Aware Load Balancer (SLB) Cluster, the offline status is communicated to the SLB when the user issues the **set-system-state offline** command, and the SLB excludes the offline OCSBC in future endpoint (re)balancing algorithms.

A version of this OCSBC graceful shutdown procedure exists in OCSBC releases previous to S-CZ7.3.0, but the procedure is enhanced for this and future releases. Previous versions only looked at active SIP sessions (calls), without monitoring active SIP registrations, and did not attempt to manipulate the period of time that active calls and registrations lingered on the OCSBC. The problem with this approach was that in the interval between setting the OCSBC to **offline** mode, and the subscriber registrations expiring, any inactive subscriber was essentially unreachable. With some carriers setting registration expiry timers to an hour or more (or 30 minutes in between registration refresh), this may have resulted in significant periods of unreachability. With this release, the new **sip-config** parameter **retry-after-upon-offline** is used to minimize the amount of time active calls and registrations keep the OCSBC from going completely offline.

The OCSBC side of this graceful shutdown procedure is followed with or without the OCSBC being a member of an SLB cluster. The graceful shutdown procedure is limited only to SIP calls and registrations.

High-level Procedure for Graceful OCSBC Shutdown

In its simplest form, this is the graceful shutdown procedure. Details and exceptions to this procedure when there are active calls or registrations are discussed in later paragraphs. The first six actions are performed whether or not the OCSBC is part of an Oracle Communications Subscriber-Aware Load Balancer (SLB) Cluster

- The OCSBC receives the **set-system-state offline** command.
- The OCSBC transitions to **becoming offline** mode.
- The OCSBC accepts calls and subscribes from registered endpoints.
- The OCSBC rejects calls from non-registered endpoints.
- The OCSBC rejects new registrations with a **503 Service Unavailable** error message.
- The OCSBC checks the number SIP INVITE based sessions and number of SIP registrations. When both counts are 0, the OCSBC transitions to the **offline** state.



Note:

Previous versions only looked at active SIP sessions (calls), without monitoring active SIP registrations.

If the OCSBC is part of an SLB Cluster:

- The SLB client on the OCSBC changes its cluster status to **shutdown** state.
- The OCSBC informs the SLB that it is offline.
- The SLB ceases to forward new end-points to the OCSBC and puts the OCSBC in a shutdown state.
- SLB continues to forward all messages for existing registered endpoints to the offline OCSBC.
- The OCSBC continues to send heartbeat updates the SLB as before.

Detailed Description of Graceful Shutdowns with Active SIP Calls or Registrations

This is the procedure when active SIP calls or registrations are on an OCSBC.

When the system receives the **set-system-state offline** command, it transitions to **becoming offline** mode. It begins checking the number of SIP-INVITE-based sessions and the number of SIP registrations, and continues to check them when sessions complete or registrations expire while it is in **becoming offline** mode. When both counts reach zero, the system transitions to **offline mode**. If the system is a member of a Oracle Communications Subscriber-Aware Load Balancer (SLB) Cluster, the SLB client on the OCSBC changes its cluster status to the **shutdown** state, and informs the SLB that it is **offline**. The SLB ceases to forward new end-points to the OCSBC and lists the OCSBC in a **shutdown** state on the SLB. The OCSBC continues to send heartbeat updates to the SLB as before.

Active calls continue normally when the OCSBC is in **becoming offline** mode. If SIP refresh registrations arrive for endpoints that have active calls, they are accepted. However, the expiry of these endpoints is reduced to the configurable **retry-after-upon-offline** timer value (in

seconds) defined under **sip-config** on the OCSBC. This timer should be configured to be a much lower time interval than originally requested by the refresh registrations, so that endpoints refresh sooner and thus the registrations expire as closely as possible to when the active call ends. If the new timer value configured in **retry-after-upon-offline** is greater than the existing registration requested refresh value, or if its value is '0' (unconfigured), the original registration refresh request is honored.

Refresh registrations for endpoints that do not have any active calls are rejected with a configurable response code defined in the **sip-config reg-reject-response-upon-offline** parameter. The default for this parameter is the **503 Service Unavailable** message. It includes a **Retry-After** header with a configurable timer set in **retry-after-upon-offline**. If the value of the configuration is 0 (unconfigured), the header is not included in the rejection message. Once these refreshes are rejected, OCSBC immediately removes such endpoints from its registration cache. It is a force remove. De-registrations are forwarded to the core. There is no local response. Removals are communicated to the SLB.

Any new calls that arrive for endpoints that currently have registration entries are not rejected. The same **retry-after-upon-offline** action is performed.

Any other SIP methods (like SUBSCRIBE or MESSAGE) intended for this endpoint is handled normally and are not rejected. Priority calls are processed as usual by the OCSBC, regardless of whether an active registration is present in the OCSBC as long as the OCSBC is in **becoming offline** state. When the OCSBC transitions to the **offline** state, even priority calls are rejected. If the priority calls cannot be forwarded to the endpoint, a **380 Alternative Service** response may be sent, depending on the OCSBC's configuration. However, when the OCSBC achieves offline mode, even priority calls are rejected. New non-priority calls coming for endpoints that are not currently registered are rejected with the **503 Service Unavailable** error message, as has always been done.

The OCSBC sends the endpoint removal requests to the SLB so that the SLB removes them from its endpoint table. If a REGISTER message comes in with multiple contacts, it's possible that one of the contacts has an active call while others do not. In that scenario, the contact without active call has the Expires value in the Contact header changed to 0 and is forwarded to the core. When the response arrives from the core, the Contact with active call has its Expires parameter modified to the **retry-after-upon-offline** value or the UA expires value, whichever is lower. Any contact with no active calls is removed from the cache.

Eventually, all SIP calls end, and all registrations expire. The OCSBC transitions to the **offline** system state. The OCSBC continues to send heartbeat updates to the SLB.

At any time after the issuance of the **set-system-state offline** command, a **set-system-state online** command may be issued. If the OCSBC is in **becoming offline** mode, the process is aborted and the OCSBC again becomes **online**. The OCSBC state is forwarded to the SLB, and the OCSBC once again participates in the SLB's (re)balancing process.

Georedundant High Availability (HA)

You can locate the two nodes that make up an HA pair in different locations from one another. This is known as georedundancy, which increases fault tolerance. A georedundant pair must adhere to rigid network operating conditions to ensure that all state and call data is shared between the systems, and that failovers happen quickly without losing calls.

The following network constraints are required for georedundant operation:

- A pair of dedicated fiber routes between sites is required. Each route must have non-blocking bandwidth sufficient to connect wancom1 and wancom2 ports (i.e., 1Gbps per port)

- Inter-site round-trip time (RTT) must be less than 10 ms. 5 ms or less is ideal. Georedundant operation must be built upon a properly engineered layer-2 WAN (eg. MPLS or Metro Ethernet) that connects active and standby HA pair members.
- Simultaneous packet loss across the inter-site link pair must be 0%. Loss of consecutive heartbeats could potentially result in split-brain behaviors.
- Security (privacy and data-integrity) must be provided by the network itself.

As with local HA nodes, management traffic (e.g. SSH, SFTP, SNMP, etc.) must be confined to the wancom0 management interface. HA node peers must have their wancom0 IP addresses on the same subnet. All Oracle Communications Subscriber-Aware Load Balancer configuration, including host routes and the system-config's **default-gateway**, is shared between the HA pair so it is not possible to have two different management interface default-gateways. This implies the requirement of an L2-switched connection between the 2 wancom0 management interfaces.

Troubleshooting Georedundant Deployments

The Oracle Communications Subscriber-Aware Load Balancer provides rich statistics and status information on HA operation, documented in the *ACLI Reference* and *Maintenance and Troubleshooting Guides*. Some of this information is especially suited for troubleshooting the latency and packet-loss requirements for georedundant deployments, including:

- Details within the **show redundancy** command output, including:
 - Request-response round-trip time measurements (**show redundancy <task-name>**)
 - Request-response loss measurements (**show redundancy <task-name>**)
 - journal statistics (**show redundancy <task-name> journals**)
 - journal latency (**show redundancy <task-name> journals**)
 - protocol-specific redundancy actions (**show redundancy <task-name> actions**)
 - protocol-specific redundancy objects (**show redundancy <task-name> objects**)
- Details within the **show queues** command output, including:
 - sipd command queue statistics (**show queue <task-name> commands**)
- Protocol-specific log messaging

3

Subscriber-Aware Load Balancer Configuration

SLB Configuration

This section explains how to configure functionality specific to the Oracle Communications Subscriber-Aware Load Balancer (SLB); it does not include configuration steps for functions that it shares in common with its corresponding Oracle Communications Session Border Controllers (OCSBCs) (for example, system-config, phy-interface, network-interface, and so on). For information about general SLB configuration, refer to the appropriate documentation as listed in About This Guide.

SLB configuration is quite simple; aside from basic network connectivity, the service interfaces, and the distribution policy, much of the configuration is learned dynamically from the OCSBCs that comprise the cluster.

Core Configuration on the OCSLB

When deploying a VNF, you configure CPU core settings using **system-config** parameters. This configuration is based on the specific needs of individual implementations. These parameters allow you to set and change the number of cores they want to assign to forwarding, DoS, and transcoding functionality. The system determines which cores perform those functions automatically.

You determine and manage your core configuration based on the services you need. The system allocates cores to signaling upon installation. You add forwarding cores to match their needs for handling media. You also add DoS and/or transcoding cores if you need those functions in your deployment. Reboot the system for core configuration changes to take effect.

Note the following:

- By default, core 0 is always set to signaling.
- The system selects cores based on function; users do not assign cores.
- The system sets unassigned cores to signaling, with a maximum of 24.

When you make core assignments, the Oracle Communications Subscriber-Aware Load Balancer (SLB) provides an error message if the system detects an issue. In addition, the system performs a check when the user issues the **verify-config** command to ensure that the total number of forwarding, plus DOS, plus transcoding cores does not exceed the maximum number of physical cores. After you save and activate a configuration that includes a change to the core configuration, the system displays a prompt to remind you that a reboot is required for the changes to take place.

You can verify core configuration from the ACLI, using the **show datapath-config** command or after core configuration changes during the save and activation processes. When using hyperthreading, which divides cores into a single physical (primary) and a single logical (secondary) core, this display may differ. Bear in mind that the SLB binds functions to primary or secondary cores using its own criteria, with secondary cores performing signaling functions

only. Hypervisors that provide a view into the type of core assigned to a function allow **show datapath-config** to display primary cores in upper-case letters and secondary cores in lower-case letters. Other hypervisors show all cores as physical.

The SLB uses the following lettering (upper- and lower-case) in the ACLI to show core assignments:

- S - Signaling
- D - DoS
- F - Forwarding

The **system-config** element includes the following parameters for core assignment:

- **dos-cores**— Sets the number of cores the system must allocate for DOS functionality. A maximum of one core is allowed. It is recommended that the virtual SLB has a DoS core configured.
- **forwarding-cores**—Sets the number of cores the system must allocate for the forwarding engine.

**Note:**

Transcoding cores, which apply to the OCSBC, are not relevant or configurable on the SLB.

To change core assignments, access the **system-config**, as follows.

```
ORACLE# configure terminal
ORACLE(configure)# system
ORACLE(system)# system-config
ORACLE(system-config)#
```

Change existing core assignment settings using the **system-config** parameters listed above. For example, to reserve a core for DoS processing:

```
ORACLE#(system-config) dos-cores 1
```

The SLB VNF has no system-based maximum number of cores, other than the range of the **system-config** parameters.

**Note:**

Refer to the *New in this Release* section at the beginning of this Essentials Guide for any release-specific core configuration constraints.

The system checks CPU core resources before every boot, as configuration can affect resource requirements. Examples of such resource requirement variations include:

- There is at least 1 CPU assigned to signaling (by the system).
- If DoS is required, then there are at least 1 CPU assigned to forwarding and 1 to DoS.
- If DoS is not required, then there is at least 1 CPU assigned to forwarding.

The system performs resource utilization checks every time it boots for CPU, memory and hard-disk to avoid configuration/resource conflicts.

Core configuration is supported by HA. For HA systems, resource utilization on the backup must be the same as the primary.

 Note:

The hypervisor always reports the datapath CPU usage fully utilized. This isolates a physical CPU to this work load. However, this may cause the hypervisor to generate a persistent alarm indicating that the VM is using an excessive amount of CPU, possibly triggering throttling. The user should configure their hypervisor monitoring appropriately, to avoid such throttling.

SLB Tunnel Configuration

The Oracle Communications Subscriber-Aware Load Balancer (SLB) sends and receives signaling messages to and from clustered OCSBCs through an IP-in-IP tunnel. The SLB requires one tunnel per interface.

Use the following procedure to perform required SLB-side tunnel configuration. Completion of tunnel configuration is accomplished on the clustered OCSBCs as described in OCSBC Tunnel Configuration.

1. From superuser mode, use the following ACLI command sequence to access tunnel-config configuration mode. While in this mode, you partially configure the tunnel-config configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# system
ORACLE(system)# network-interface
ORACLE(network-interface)# tunnel-config
ORACLE(tunnel-config)# ?
local-ip-address      tunnel local IP address
port                  local & remote control ports
protocol              tunnel control transport protocol
tls-profile           tunnel control TLS profile
select               select tunnel to edit
no                    delete tunnel
show                  show tunnel
done                  write tunnel information
quit                  quit out of configuration mode
exit                  return to previous menu
ORACLE(tunnel-config)#
```

2. Use the **local-ip-address** parameter to specify the IP address at the SLB end of the tunnel.

As the terminus for all tunnels from the clustered OCSBCs — and never the tunnel originator — only the local address is configured on the SLB.

 Note:

This address also supports the exchange of CCP messages.

```
ORACLE(tunnel-config)# local-ip-address 182.16.204.210
ORACLE(tunnel-config)#
```

3. Use the **port** parameter to specify the port used to send and receive CCP messages.

```
ORACLE(tunnel-config)# port 4444
ORACLE(tunnel-config)#
```

4. Use the **protocol** parameter to specify the transport protocol used in support of cluster control messages.

The only protocol supported for this release is UDP.

```
ORACLE(tunnel-config)# protocol UDP
ORACLE(tunnel-config)#
```

5. Use **done**, **exit**, and **verify-config** to complete configuration of this tunnel-config configuration element.
6. Repeat Steps 1 through 6 to configure additional tunnel-config configuration elements.

Sample SLB Tunnel Configuration

The following formatted extract from **show running-config** ACLI output shows a sample tunnel configuration.

```
tunnel-config
local-ip-address      182.16.204.210
port                  4444
protocol              UDP
tls-profile
last-modified-by      admin@console
last-modified-date    2013-11-07 18:49:04
```

Cluster Configuration

The cluster-config configuration element manages basic SLB interaction with clustered OCSBCs — it contains a set of global parameters that define the management of the RFC 2003 IP-in-IP tunnels that connect the SLB to clustered OCSBCs, and the details of rebalance operations. In addition, cluster-config provides for the creation of a list of service interfaces (signaling addresses) that are advertised to endpoints comprising the user access population.

Use the following procedure to perform required cluster-config configuration.

1. From superuser mode, use the following ACLI command sequence to access cluster-config configuration mode. While in this mode, you configure the cluster-config configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
```

```
ORACLE(session-router)# cluster-config
ORACLE(cluster-config)# ?
```

state	cluster control state
log-level	configure log level
auto-rebalance	Auto-rebalance cluster on new SD availability
source-rebalance-threshold	Percentage of advertised registration
capacity	
dest-rebalance-threshold	Percentage of advertised registration
capacity	
dest-rebalance-max	Percentage of advertised registration
capacity	
tunnel-check-interval	How often an SD's tunnels are checked
tunnel-fail-interval	Time for which no messages have been received
rebalance-request-delay	Delay between subsequent rebalance requests
session-multiplier	ratio of users (endpoints to sessions)
atom-limit-divisor	ratio of atoms (e.g. contacts to endpoints)
rebalance-skip-ahead	Skip endpoints refreshing sooner than
rebalance-max-refresh	Skip endpoints refreshing later than
ignore-tgt-svcs-on-rebalance	When selecting source SDs during rebalancing
rebalance-del-app-entries	Delete Application endpoint Data
inactive-sd-limit	Duration no SD control messages received (seconds)
red-port	redundant mgcp sync port
red-max-trans	max redundant transactions to keep
red-sync-start-time	redundant sync start timeout
red-sync-comp-time	redundant sync complete timeout
service-ports	configure service ports
select	select cluster config
no	delete cluster config
show	show cluster config
done	save cluster config information
exit	return to previous menu

2. Use the **state** parameter to enable or disable the SLB software.

The default setting, enabled, enables SLB functionality; disabled renders the SLB inoperable.

```
ORACLE(cluster-config)# state enabled
ORACLE(cluster-config)#
```

3. Use the **log-level** parameter to specify the contents of the SLB log.

Log messages are listed below in descending order of severity.

- emergency — the most severe
- critical
- major (error)
- minor (error)
- warning
- notice
- info — (default) the least severe
- trace — (test/debug, not used in production environments)

- **debug** — (test/debug, not used in production environments)
- **detail** — (test/debug, not used in production environments)

In the absence of an explicitly configured value, **log-level** defaults to **critical**, meaning that log messages with a severity of critical or greater (emergency) are written to the SLB log.

```
ORACLE(cluster-config)# log-level critical
ORACLE(cluster-config)#
```

4. Use the **auto-rebalance** parameter to specify SLB behavior when a new OCSBC joins an existing cluster.

With this parameter enabled, the default setting, the SLB redistributes endpoints among cluster members when a new member joins the cluster. Refer to the Rebalancing section for operational details.

With this parameter disabled, the alternate setting, pre-existing OCSBCs retain their endpoint populations, and the SLB directs all new endpoints to the newly active OCSBC until that OCSBC reaches maximum occupancy.

```
ORACLE(cluster-config)# auto-rebalance enabled
ORACLE(cluster-config)#
```

5. If **auto-rebalance** is set to enabled, use the **source-rebalance-threshold** and **dest-rebalance-threshold** parameters to specify threshold settings that identify existing cluster SBCs as either endpoint sources or endpoint destinations during the rebalance operation. Use the **dest-rebalance-max** parameter to specify the occupancy for the new cluster member. Refer to the Balancing section for details on occupancy and its calculation.

If **auto-rebalance** is set to disabled, these three parameters can be ignored.

Parameter values are numeric percentages within the range 0 through 100.

source-rebalance-threshold specifies the minimum occupancy percent that identifies a clustered OCSBC as a source of endpoints during a rebalance operation. For example, using the default value of 50 (percent), any clustered SBC with an occupancy rate of 50% or more sheds endpoints during a rebalance. The SLB assigns these endpoints to the new cluster member.

dest-rebalance-threshold specifies the maximum occupancy percent that identifies a clustered OCSBC as a destination for endpoints during a rebalance operation. Note that the default setting of 0 (percent), ensures that no pre-existing OCSBC gains endpoints during a rebalance.

dest-rebalance-max specifies the maximum occupancy percent that the SLB transfers to the new cluster member during a rebalance operation. The default setting is 80 (percent). Should this threshold value be attained, the SLB distributes remaining endpoints to those OCSBCs identified as endpoint destinations by their **dest-rebalance-threshold** settings.

```
ORACLE(cluster-config)# source-rebalance-threshold 50
ORACLE(cluster-config)# dest-rebalance-threshold 40
ORACLE(cluster-config)# dest-rebalance-max 75
```

6. If **auto-rebalance** is set to enabled, you can optionally use four additional parameters to fine-tune rebalance operational details.

If **auto-rebalance** is set to disabled, these four parameters can be ignored.

rebalance-request-delay specifies the interval (in milliseconds) between endpoint request messages sent from the SLB to a clustered OCSBC. As explained in the Rebalancing

section, these messages request a list of endpoints that will be redistributed from the OCSBC to a new cluster member.

By default, this parameter is set to 500 milliseconds.

Setting this parameter to a higher value results in longer times for the completion of rebalancing; however longer durations provide more time for cluster member processing of SIP traffic.

rebalance-skip-ahead restricts the target set of OCSBC endpoints registration eligible for rebalancing to those whose re-registration is not imminent — that is, the registration is not scheduled within the number of milliseconds specified by the parameter setting. Setting this parameter to a non-zero value mitigates against the possibility of a race condition precipitated by a simultaneous endpoint removal generated by the OCSBC and the arrival of endpoint signalling on an SLB service port. The default setting (0 milliseconds) effectively makes the entire OCSBC endpoint set eligible for rebalancing.

rebalance-max-refresh restricts the target set of OCSBC endpoints eligible for rebalancing to those whose re-registration is no further in the future than the time period (milliseconds) specified by this parameter— for example, assuming a parameter value of 6000, the target endpoint set is restricted to those whose re-registration is scheduled within the next 6 seconds.

Because a re-balancing operation necessarily introduces a small window of unreachability for re-balanced endpoints, this parameter provides users with some degree of control over the period of time that a re-balanced endpoint may be unreachable.

The default setting (0 milliseconds) effectively makes the entire SBC endpoint set eligible for rebalancing.

rebalance-del-app-entries specifies when cached SIP entries for rebalanced endpoints are removed from the clustered OCSBC. The default setting (enabled) specifies that the SBC removes cached registration entries at the completion of the rebalance operation. When set to disabled, this parameter specifies that cached entries are retained after a rebalance operation, and subsequently removed from the cache by standard time-out procedures.

```
ORACLE(cluster-config)# rebalance-request-delay 750
ORACLE(cluster-config)# rebalance-skip-ahead 100
ORACLE(cluster-config)# rebalance-max-refresh 1000
ORACLE(cluster-config)# rebalance-del-app-entries enabled
```

7. Three parameters, **tunnel-fail-interval**, **tunnel-check-interval**, and **inactive-sd-limit** maintain and monitor the IP-in-IP tunnels established between the SLB and clustered SBCs.

tunnel-fail-interval specifies the interval (in milliseconds) between periodic keepalive messages sent from a clustered OCSBC to the SLB. If the SLB fails to receive a keepalive message within the specified period, it flags the tunnel as dead. By default, this parameter is set to 10000 milliseconds.

tunnel-check-interval specifies the interval (in milliseconds) between SLB tunnel audits. During a tunnel audit, the SLB checks the status of each tunnel and removes all tunnels flagged as dead. By default, this parameter is set to 15000 milliseconds.

If you change default settings for either parameter, ensure that the setting for **tunnel-check-interval** is greater than the **tunnel-fail-interval** setting.

inactive-sd-limit specifies the maximum silent interval (defined as the absence of heartbeat traffic from any tunnel) seconds) before the SLB flags a cluster member as dead, and removes that OCSBC from the cluster. By default, this parameter is set to 1800

seconds (30 minutes). supported values are integers within the range 0 through 31556926 (365 days).

```
ORACLE(cluster-config)# tunnel-fail-interval 10000
ORACLE(cluster-config)# tunnel-check-interval 15000
ORACLE(cluster-config)# inactive-sd-limit 900
```

8. Use the **session-multiplier** and **atom-limit-divisor** parameters to specify optional, user-configurable numeric factors used in occupancy and occupancy rate calculations.

session-multiplier provides a factor that when multiplied by an OCSBC's licensed session limit, determines the maximum number of endpoints that the OCSBC can support (that is, its maximum occupancy).

The default setting is 10; valid settings include any integer values within the range 1 through 100.

Using the default setting, an OCSBC licensed for 32,000 concurrent sessions has a maximum theoretical occupancy of 320,000 endpoints.

atom-limit-divisor provides another factor that can be used in occupancy and occupancy percent calculations. By default, occupancy calculations are based on endpoints (IP addresses), and do not take into account the fact that the same IP address can represent multiple users.

The default setting is 1, which assumes a conservative 1-to-1 correlation between endpoints and users; valid settings include any integer values within the range 1 through 1000.

Note:

The SLB initially calculates a tentative maximum occupancy value, expressed as a number of endpoint addresses, for each clustered OCSBC. SLB calculations are based upon the licensed capacity of each cluster member, and the values assigned to the session-multiplier and atom-limit-divisor parameters. After calculating the tentative maximum occupancy value, the SLB compares this value to the value of the registration-cache-limit parameter as defined on the clustered OCSBC. If the value of registration-cache-limit is either 0, or greater than the tentative maximum occupancy value, the calculated value is retained as the occupancy ceiling. However, if the registration-cache-limit value is greater than 0, but less than the tentative calculation, the value of registration-cache-limit is used as the occupancy ceiling.

Once an OCSBC has reached its maximum number of endpoints, the SLB removes it from the load balancing algorithm. These parameter settings should be changed only after careful examination of network conditions and behavior.

```
ORACLE(cluster-config)# session-multiplier 10
ORACLE(cluster-config)# atom-limit-divisor 1
```

9. The **ignore-tgt-svc-on-rebalance** parameter is not currently supported, and can be safely ignored.
10. Retain default settings for the **red-port**, **red-max-trans**, **red-sync-start-time**, and **red-sync-comp-time** parameters.
11. Use **done**, **exit**, and **verify-config** to complete cluster configuration.

Sample Cluster Configuration

The following formatted extract from **show running-config** CLI output shows a sample cluster configuration.

```
cluster-config
state                               enabled
log-level                           CRITICAL
auto-rebalance                      enabled
source-rebalance-threshold          50
dest-rebalance-threshold             40
dest-rebalance-max                   75
tunnel-check-interval               750
tunnel-fail-interval                10000
rebalance-request-delay              500
session-multiplier                   4
rebalance-skip-ahead                 0
rebalance-max-refresh                0
ignore-tgt-svcs-on-rebalance         disabled
atom-limit-divisor                   1000
rebalance-del-app-entries            enabled
inactive-sd-limit                   1800
red-port                             2001
red-max-trans                        10000
red-sync-start-time                  5000
red-sync-comp-time                   1000
service-port
last-modified-by                     admin@console
last-modified-date                   2013-11-07 18:49:04
```

Service Ports Configuration

A service-port is essentially a SIP port monitored by the SLB for incoming signaling from the user population. For virtually all network topologies, multiple service ports are expected on a typical SLB configuration. A service-port is a multiple instance configuration element; for each service-port advertised to the access network(s), at least one service-port configuration element must be configured.

Configuration changes to service-ports cause a reset to the flow-IDs associated with that port. This reset causes a wide variety of data changes, including endpoint re-assignments, data counter discrepancies and so forth. Although these changes are allowed, the user must allow for a significant amount of time to pass before expecting up-to-date show commands and endpoint assignments.

Use the following procedure to perform required service-ports configuration.

1. From superuser mode, use the following CLI command sequence to access service-port configuration mode. While in this mode, you configure one or more service-port configuration elements.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# cluster-config
ORACLE(cluster-config)# service-ports
ORACLE(service-port)# ?
```

address	IP address
port	port (default: 5060)
protocol	transport protocol
network-interface	network interface for service port
select	select cluster config
no	delete cluster config
show	show cluster config
done	save cluster config information
exit	return to previous menu

```
ORACLE(service-port) #
```

2. Use the required **address** parameter to specify the IPv4 or IPv6 address of this service port.

```
ORACLE(service-port) # address 10.0.0.1
ORACLE(service-port) #
```

3. Use the **port** parameter to specify the port monitored by the SLB for incoming signaling messages.

In the absence of an explicitly configured port, the SLB provides a default value of 5060 (the registered SIP port).

Allowable values are integers within the range 0 through 65535.

```
ORACLE(service-port) # port 5060
ORACLE(service-port) #
```

4. Use the **protocol** parameter to choose the transport protocol.
The supported setting is UDP (the recommended default).

```
ORACLE(service-port) # protocol udp
ORACLE(service-port) #
```

5. Use the required **network-interface** parameter to identify the SLB network interface that supports this service port. With this parameter, you have the option of specifying IPv4 or IPv6 (.4 or .6).

```
ORACLE(service-port) # network-interface M00:0.4
ORACLE(service-port) #
```

6. Use **done**, **exit**, and **verify-config** to complete configuration of this service-port configuration element.
7. Repeat Steps 1 through 6 to configure additional service-port configuration elements.

Sample Service Port Configuration

The following formatted extract from **show running-config** ACLI output shows a sample service port configuration.

```
service-port
address          192.169.203.83
port             5060
protocol         UDP
network-interface M00:0.4
```

last-modified-by	admin@console
last-modified-date	2013-11-07 18:49:04

Traffic Policy Configuration

This configuration record will enable management of tunnel bandwidth on a per-cluster member basis. The **name** of this policy will be entered into the OCSBC Tunnel Configurations.

Note:

If you do not need to change any of the implicit defaults for the traffic policy, you do not need to configure this policy at all. The implicit default configuration for this policy is as below. If you must change any of the parameters from the implicit default, you must name the resulting traffic policy **default**.

Use the following procedure to perform traffic policy configuration if required.

1. Access the **traffic-policy-config** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# traffic-policy-config
ORACLE(traffic-policy-config)#
```

2. Enter a **name** for this traffic policy configuration. This is the string identifier for this policy.

```
ORACLE(traffic-policy-config)# name tp1
ORACLE(traffic-policy-config)#
```

3. Enter a **throttle-rate** for this traffic policy configuration. This is the per tunnel throttle rate in packets per second.

```
ORACLE(traffic-policy-config)# throttle-rate 800
ORACLE(traffic-policy-config)#
```

4. Enter a **max-signaling-rate** for this traffic policy configuration. This is the maximum signaling rate to a cluster member in packets per second.

```
ORACLE(traffic-policy-config)# max-signaling-rate 32000
ORACLE(traffic-policy-config)#
```

5. Enter a **min-untrusted-pct** for this traffic policy configuration. This is the minimum percentage of signaling rate allocated to untrusted traffic.

```
ORACLE(traffic-policy-config)# min-untrusted-pct 33
ORACLE(traffic-policy-config)#
```

6. Enter a **max-untrusted-pct** for this traffic policy configuration. This is the maximum percentage of signaling rate allocated to untrusted traffic.

```
ORACLE(traffic-policy-config)# max-untrusted-pct 66
ORACLE(traffic-policy-config)#
```

7. Use **done**, **exit**, and **verify-config** to complete configuration of this traffic policy configuration element.
8. Repeat steps 1 through 7 to configure additional traffic policy configuration elements.

Sample Traffic Policy Configuration

The following formatted extract from **traffic-policy-config** shows the default policy configuration.



Note:

For this initial release of software, if you do not need to change any of the implicit defaults for the traffic policy, you do not need to configure this policy at all. The implicit default configuration for this policy is as below. If you need to change any of the parameters from the implicit default, you must name the resulting traffic policy **default**

```
SLB1(traffic-policy-config)# show
traffic-policy-config
      name                      default
  throttle-rate                800
max-signaling-rate             32000
min-untrusted-pct              10
max-untrusted-pct              30
options
```

Load Balancer Policy Configuration

The lbp-config configuration element manages the SLB endpoint table. It also creates and manages a list of service interfaces (signaling addresses) that are advertised to endpoints comprising the user access population.

Use the following procedure to perform required lbp-config configuration.

1. From superuser mode, use the following ACLI command sequence to access lbp-config configuration mode. While in this mode, you configure the lbp-config configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# lbp-config
ORACLE(lbp-config)#?
state                      lbp state
log-level                  configure log level
untrusted-grace-period     Untrusted grace period
max-untrusted-percentage   Maximum untrusted endpoints percentage
max-untrusted-upper-threshold
                           Maximum untrusted endpoints upper
                           threshold
max-untrusted-lower-threshold
                           Maximum untrusted endpoints upper
                           threshold
endpoint-capacity-upper-threshold
                           endpoint capacity upper threshold
endpoint-capacity-lower-threshold
                           endpoint capacity lower threshold
red-port                   lbp redundant sync port: 0 to disable
```

red-max-trans	and 2000 to enable maximum redundancy transactions to keep on active
red-sync-start-time	timeout for transitioning from standby to active
red-sync-comp-time	sync request timeout after initial sync completion
port-aware-balancing	Include endpoint source port, in addition
used	to the source IP address if NAT is
options	optional features/parameters
strategy	Configure the balancing strategy
select	select lbp config
no	delete lbp config
show	show lbp config
done	save lbp config information
exit	return to previous menu
ORACLE(lbp-config)#	

2. Use the **state** parameter to enable or disable the SLB software.

The default setting, enabled, enables SLB functionality; disabled renders the SLB inoperable.

```
ORACLE(lbp-config)# state enabled
ORACLE(lbp-config)#
```

3. Use the **log-level** parameter to specify the contents of the SLB log.

Log messages are listed below in descending order of severity.

- emergency — the most severe
- critical
- major (error)
- minor (error)
- warning
- notice
- info — (default) the least severe
- trace — (test/debug, not used in production environments)
- debug — (test/debug, not used in production environments)
- detail — (test/debug, not used in production environments)

In the absence of an explicitly configured value, **log-level** defaults to critical, meaning that log messages with a severity of critical or greater (emergency) are written to the LBP log.

```
ORACLE(lbp-config)# log-level critical
ORACLE(lbp-config)#
```

4. Use the **untrusted-grace-period** parameter to specify the maximum time, in seconds, that a forwarding rule is retained by the SLB before it is confirmed with a promotion message from the OCSBC that received the untrusted endpoint. Refer to the Balancing section for message details

In the absence of an explicitly assigned value, the SLB provides a default setting of 32 (seconds). If this time period elapses without a promotion message arriving to confirm this user, the SLB deletes the entry.

Setting this parameter to 0 allows untrusted/unconfirmed entries to exist indefinitely without aging out.

```
ORACLE(lbp-config) # untrusted-grace-period 32
```

5. Use the **max-untrusted-percentage** parameter to specify the percentage of the overall endpoint population that is reserved for untrusted users.

The default setting is 20 (percent); supported values are integers within the range 1 through 100.

This percentage is applied to the overall remaining occupancy of the SLB after trusted (confirmed) users are accounted for. For example, when empty, the SLB holds two million forwarding rules; assuming the default setting, at most 400,000 rules are reserved for untrusted rules. By the time one million users have been promoted, 20% of the remaining space means that up to 200,000 entries can be used for untrusted users.

```
ORACLE(lbp-config) # max-untrusted-percentage 20
```

6. Use the **max-untrusted-upper-threshold** parameter to specify a threshold level at which the SLB (1) raises an alarm, and (2) issues an SNMP trap reporting an excessive number of untrusted endpoints within the entire endpoint population.

This parameter, which has a default setting of 80 (percent), is calculated as a percent of **max-untrusted-percentage**. For example, assuming default settings for both parameters, the SLB raises an alarm and issues an SNMP trap when the percentage of untrusted endpoints attains 16%.

```
ORACLE(lbp-config) # max-untrusted-upper-threshold 80
```

7. Use the **max-untrusted-lower-threshold** parameter to specify a threshold level at which the SLB (1) clears the existing untrusted endpoint alarm, and (2) issues an SNMP trap reporting alarm clearance.

This parameter, which has a default setting of 70 (percent), is calculated as a percent of **max-untrusted-percentage**. For example, assuming default settings for both parameters, the SLB clears an alarm and issues an SNMP trap when the percentage of untrusted endpoints falls to 14%.

```
ORACLE(lbp-config) # max-untrusted-lower-threshold 70
```

8. Use the **endpoint-capacity-upper-threshold** and **endpoint-capacity-lower-threshold** parameters to implement license-based management and monitoring of the SLB endpoint counts.

endpoint-capacity-upper-threshold specifies a threshold level at which the SLB (1) raises an alarm, and (2) issues an SNMP trap reporting an excessive number of active endpoints.

This parameter, which has a default setting of 80 (percent), is calculated as a percentage of the endpoints allowed by the installed SLB license.

endpoint-capacity-lower-threshold specifies a threshold level at which the SLB (1) clears the existing endpoint alarm, and (2) issues an SNMP trap reporting alarm clearance.

This parameter, which has a default setting of 70 (percent), is calculated as a percentage of the endpoints allowed by the installed SLB license.

```
ORACLE(lbp-config) # endpoint-capacity-upper-threshold 80
ORACLE(lbp-config) # endpoint-capacity-lower-threshold 70
ORACLE(lbp-config) #
```

9. Enable **port-aware-balancing** to include endpoint source port, in addition to the source IP and destination service representation when looking up a unique EPT prior to forwarding towards the OCSBC cluster. Choices are enabled and disabled. Default is disabled.

Reboot all SLBs and OCSBCs when enabling or disabling this parameter.

```
ORACLE(lbp-config) # port-aware-balancing enabled
ORACLE(lbp-config) #
```

WARNING:

The user must reset the deployment's endpoint tables upon any change to this parameter to establish entry consistency. Reboot or, in the case of devices operating in HA mode, dual reboot all systems affected by changes to this parameter.

10. Set your preferred **strategy** to determine how the SLB distributes new end-points to the OCSBCs. Choices include capacity-proportional and least-occupied. The default is least-occupied.

```
ORACLE(lbp-config) # strategy least-occupied
ORACLE(lbp-config) #
```

11. Use **done**, **exit**, and **verify-config** to complete configuration of this load-balancer-policy configuration element.

Sample Load Balancer Policy Configuration

The following formatted extract from **show running-config** ACLI output shows a sample load balancer policy configuration with port-aware-balancing enabled.

```
lbp-config
state                               enabled
log-level                           NOTICE
untrusted-grace-period              32
max-untrusted-percentage            20
max-untrusted-upper-threshold       80
max-untrusted-lower-threshold       70
end-point-capacity-upper-threshold  80
end-point-capacity-lower-threshold  70
red-port                            0
red-max-trans                        500000
red-sync-start-time                  5000
red-sync-comp-time                   1000
port-aware-balancing                 enabled
options
strategy                             least-occupied
```

last-modified-by
last-modified-date

admin@console
2015-11-07 18:49:04

Distribution Policy Configuration

Distributing endpoints equitably among the cluster members is the primary function of the SLB. The lb-policy configuration element allows you to control the method of the SLB's distribution based on matching criteria. Using inbound packet matching criteria, you can control the assignment of users to OCSBCs. Matching is done by data available up to and including the transport layer of the packet: source IP address and port, destination IP address and port, and transport protocol. The IP addresses and ports may or may not include bit masks as well.

Conceptually, the load balancer policy table, with sample data, looks akin to the following.

Source IP/Mask	Source Port/ Mask	Destination IP/Mask	Destination Port/Mask	Transport Protocol Requirements (list)	Realm Identifiers (list)
192.168.7.22/32	0/0	10.0.0.1/32	5060/16		West
192.168.1.0/24	0/0	10.0.0.1/32	5060/16	UDP, TCP	North, South, West
192.168.0.0/16	0/0	10.0.0.1/32	5060/16	UDP, TCP	East, West
0.0.0.0/0	0/0	0.0.0.0/0	0/0		

Policies are matched using a longest prefix match algorithm; the most specific policy is selected when comparing policies to received packets. One and only one policy is chosen per packet; if the next hops in that route are all unavailable, the next best route is not consulted (instead, the default policy may be consulted – see below). This is different than the local-policy behavior on the OCSBC.

Within each policy you may configure multiple next hops, where each next hop is a named group of OCSBCs. In the sample policy table, this is indicated in the second policy with a source IP range of 192.168.1.0/24. The realm identifier list for this policy indicates North, South, West. Each of these realm identifiers represents a collection of zero or more OCSBCs, in OCSBC parlance these are roughly analogous to session-agent groups. Each of these realm identifiers is also assigned a priority (a value between 1 and 31, with 31 representing the highest priority) in the configuration, and the SLB sorts the possible destinations with the highest priority first. Upon receipt of a packet matching a policy with multiple configured realm identifiers, the SLB gives preference to OCSBCs from the realm identifier with the highest priority. Should no OCSBCs be available in that priority level (due to saturation, unavailability, and so on.) the SLB moves on to investigate the next priority level, and so on. Should no OCSBCs become available after traversing the entire list of all OCSBCs within each priority level, the OCSBC either drops the packet or attempt to use the default policy.

The bottom row of the sample table shows this implicit, last resort default policy. When enabled, the SLB reverts to the default policy when all of the potential next hop realms referenced in the endpoint's distribution rule are unavailable. In that event, the default policy attempts to locate a clustered OCSBC that advertises support for the service-interface that the packet arrived on. The realm is not considered when matching to the default policy. If such an OCSBC is found, the SLB forwards the packet to that DBC; if such an OCSBC is not found, the SLB drops the packet.

It is not necessary to configure the default policy — it is simply intended as a catchall policy, and may be used when all that is required is a simple round-robin balancing scheme based on simple metrics (for example, CPU utilization and number of registrations currently hosted by an

OCSBC). If no policies are configured on the SLB, the default policy is used. The default realm is implied in the above table as * and is enabled by default for policy records.

Use the following procedure to perform required lb-policy configuration.

1. From superuser mode, use the following ACLI command sequence to access lb-policy configuration mode. While in this mode, you configure the distribution rules used to implement policy-based load balancing on the SLB.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# lb-policy
ORACLE(lb-policy)# ?
state                        lb policy state
default-realm                use default realm
description                  load balancer policy description
protocols                    list of protocols
lb-realms                    list of realms
                             name
                             priority
source-addr                  source ip address
destination-addr             destination ip address
select                       select lb policy
no                            delete lb policy
show                         show lb policy
done                         save lb policy information
quit                         quit out of configuration mode
exit                         return to previous menu
ORACLE(lb-policy)#
```

2. Use the **state** parameter to enable or disable this distribution rule.

The default setting, enabled, enables the distribution rule; disabled disables the rule.

```
ORACLE(lb-policy)# state enabled
ORACLE(lb-policy)#
```

3. Use the **default-realm** parameter to enable or disable the default distribution policy.

The default setting, enabled, enables the default policy; disabled disables the policy.

With **default-realm** enabled, the SLB provides a best-effort delivery model if the next-hop realms listed in this distribution rule are unavailable. With **default-realm** disabled, the orphaned packet is dropped.

```
ORACLE(lb-policy)# default-realm enabled
ORACLE(lb-policy)#
```

4. Optionally use the **description** parameter to provide a description of this distribution rule.

```
ORACLE(lb-policy)# description Local traffic to Los Angeles site
ORACLE(lb-policy)#
```

5. Use the **protocols** parameter to construct a list of protocols that must be supported by this distribution rule.

```
ORACLE(lb-policy)# protocols udp  
ORACLE(lb-policy)#
```

6. Use either the **source-addr** parameter or the **destination-address** parameter to specify matching criteria for this distribution rule.

Use the **source-addr** parameter to specify source-address-based matching criteria.

Packets whose source IP addresses match the criteria specified by this parameter are subject to this distribution rule.

```
ORACLE(lb-policy)# source-addr 10.0.0.1  
ORACLE(lb-policy)#
```

matches any port on the specified IP source address

```
ORACLE(lb-policy)# source-addr 10.0.0.1:5060  
ORACLE(lb-policy)#
```

matches the specified IP source address:port pair

```
ORACLE(lb-policy)# source-addr 10.0.0.1/24  
ORACLE(lb-policy)#
```

matches any IP source address, any port on the 10.0.0.x subnet

```
ORACLE(lb-policy)# source-addr 10.0.0.240/28:5060  
ORACLE(lb-policy)#
```

matches IP source addresses 10.0.0.240:5060 through 10.0.0.255:5060

Use the **destination-addr** parameter to specify destination-address-based matching criteria.

Packets whose destination IP addresses match the criteria specified by this parameter are subject to this distribution rule.

```
ORACLE(lb-policy)# destination-addr 10.0.0.1  
ORACLE(lb-policy)#
```

matches any port on the specified IP destination address

```
ORACLE(lb-policy)# destination-addr 10.0.0.1:5060  
ORACLE(lb-policy)#
```

matches the specified IP destination address:port pair

```
ORACLE(lb-policy)# destination-addr 10.0.0.1/24  
ORACLE(lb-policy)#
```

matches any IP destination address, any port on the 10.0.0.x subnet

```
ORACLE(lb-policy)# destination-addr 10.0.0.240/28:5060
ORACLE(lb-policy)#
```

matches destination IP addresses 10.0.0.240:5060 through 10.0.0.255:5060

7. Use the `lb-realms` parameter to access lb-realm configuration mode.

While in lb-realm configuration mode you identify one or more SLBs eligible to receive traffic that matches this distribution rule.

```
ORACLE(lb-policy)# lb-realms
ORACLE(lb-realm)#
name                realm name (string identifier)
priority            priority (range 1-31)
select              select a lb realm to edit
no                  delete selected lb realm
show                show lb realm information
done                write lb realm information
exit                return to previous menu
ORACLE(lb-realm)#
```

8. Use the `name` parameter to identify the realm.

As previously discussed, the name field is roughly analogous to an OCSBC session-agent group. OCSBCs configured to communicate within a cluster hosted by an SLB advertise offered services to the SLB. These services (for example, SIP support) exist in realms, whose names are sent to the SLB as part of the OCSBC advertisement. The SLB, upon receipt of these advertisements, joins each OCSBC into one or more realm identifier groups based upon the realm name(s) the OCSBC has offered up. The **name** command of the lb-realm configuration element matches this distribution rule to a supporting OCSBC that has offered that realm name for cluster membership.

```
ORACLE(lb-realm)# name LosAngeles
ORACLE(lb-realm)#
```

9. Use the `priority` parameter to specify the realm priority.

Priority is expressed as an integer value within the range 1 to 31 — the higher the integer, the greater the priority.

The default value, 1, specifies use of the default routing policy, and should not be used when policy-based distribution is enabled.

Priority values are considered when multiple OCSBCs offer the same service to matched packets.

```
ORACLE(lb-realm)# priority 31
ORACLE(lb-realm)#
```

10. Use `done`, `exit`, and `verify-config` to complete configuration of this lb-realm configuration element.

11. To specify other eligible SLBs, repeat Steps 7 through 10. For example,

```
ORACLE(lb-policy)# lb-realms
ORACLE(lb-realm)# name LasVegas
```

```
ORACLE(lb-realm)# priority 25
ORACLE(lb-realm)# done
ORACLE(lb-realm)# exit
ORACLE(lb-realm)# verify-config
```

12. Use **done**, **exit**, and **verify-config** to complete configuration of this distribution rule.
13. To specify additional distribution rules, repeat Steps 1 through 12 as often as necessary.

Sample Distribution Rule Configurations

The following formatted extract from **show running-config** ACLI output shows sample distribution rule configurations.

```
lb-policy
state                enabled
default-realm        enabled
description
protocols            TCP
    lb-realm
        name          Realm192p1
        priority      10
source-addr          1.1.0.0/16
destination-addr      0.0.0.0/0
last-modified-by      admin@console
last-modified-date    2013-11-07 18:58:10
lb-policy
state                enabled
default-realm        enabled
description
protocols            TCP
    lb-realm
        name          Realm192p1
        priority      7
source-addr          1.20.0.0/16
destination-addr      0.0.0.0/0
last-modified-by      admin@console
last-modified-date    2013-11-07 19:01:01
lb-policy
state                enabled
default-realm        enabled
description
protocols            TCP
    lb-realm
        name          Realm192p1
        priority      5
source-addr          1.120.0.0/16
destination-addr      0.0.0.0/0
last-modified-by      admin@console
last-modified-date    2013-11-07 19:00:49
lb-policy
state                enabled
default-realm        enabled
description
protocols            TCP
    lb-realm
```

```
name      Realm192p1
priority  3
```

Forced Rebalance

The **notify ccd rebalance** ACLI command initiates an immediate forced rebalance operation. A forced rebalance operation is identical to the one described in the Rebalancing section.

notify ccd rebalance [cancel [sd-name]]

```
ORACLE# notify ccd rebalance
```

This command string initiates the forced rebalance by calculating drop counts for each eligible cluster member, and then requesting drops from the first cluster member in the rebalance queue.

```
ORACLE# notify ccd rebalance cancel
```

This command string terminates the forced rebalance.

```
ORACLE# notify ccd rebalance cancel ~sam
```

This command string terminates the forced rebalance for a specified cluster member. Note the use of tilde special character, which forces the SLB to do a substring match of the following string against all cluster member names. Assuming a cluster member samadams — that cluster member removes itself from the rebalance queue, if it has not yet removed endpoints, or ceases endpoint removal and exits the queue if it is currently doing so.

Notify CCD Drop

The **notify ccd drop** ACLI command instructs the target cluster member to drop a specific number of endpoints from a specific realm, from all realms, or without regard for realm.

notify ccd drop <sd-name> (<realm> <number> | <number>)

```
ORACLE# notify ccd drop ~sam boston 100
```

This command string instructs the target cluster member to drop 100 endpoints from the boston realm.

```
ORACLE# notify ccd drop ~sam * 100
```

This command string, using the * special character, instructs the target cluster member to drop 100 endpoints from all realms.

```
ORACLE# notify ccd drop ~sam 100
```

This command string instructs the target cluster member to drop 100 endpoints without regard for realm.

You can also use the argument **all** after the **<sd-name>** or **<realm>** arguments to instruct the system to drop all endpoints from the specified SBC or realm.

```
ORACLE# notify ccd drop ~sam all
```

This command string instructs the target cluster member to drop all endpoints without regard for realm.

```
ORACLE# notify ccd drop ~sam boston all
```

This command string instructs the target cluster member to drop all endpoints from the boston realm.

OCSBC Configuration

This section describes the configuration necessary to allow an Oracle Communications Session Border Controller (OCSBC) to join a cluster. Configuration is simplified to allow for an easy and seamless migration from a deployed standalone OCSBC to a deployed clustered OCSBC. There are only two places where new configuration is required: in the network-interface configuration element, where tunnel information is defined; and in the signaling application's interface, (the sip-interface configuration element).

OCSBC Tunnel Configuration

Configuring the properties of the IP-in-IP tunnel on the Oracle Communications Session Border Controller (OCSBC) is a matter of configuring the local IP address, remote IP address, and specifying transport layer and application layer protocol support.

The following example uses a tunnel named sipSignaling, which was initially and partially configured on the Oracle Communications Subscriber-Aware Load Balancer (SLB). Note in the following configuration that the value of **remote-ip-address** parameter must agree with the value which was previously set with the **local-ip-address** parameter on the SLB. The complementary configuration performed on the SLB enables tunnel establishment between the OCSBC and the SLB.

1. From superuser mode, use the following ACLI command sequence to access tunnel-config configuration mode. While in this mode, you perform required SLB tunnel configuration.

```
ORACLE# configure terminal
ORACLE(configure)# system
ORACLE(system)# network-interface
ORACLE(network-interface)# tunnel-config
ORACLE(tunnel-config)# ?
name                tunnel name
local-ip-address     tunnel local IP address
remote-mac-address   tunnel remote mac address
remote-ip-address    tunnel remote IP address
application          application protocol for this tunnel
port                tunnel local & remote control ports
protocol             tunnel control transport protocol
tls-profile          tunnel control TLS profile
traffic-policy       Name of traffic policy that
                     applies to this tunnel
```



```
select          select tunnel to edit
no              delete tunnel
show            show tunnel
done            write tunnel information
exit            return to previous menu
ORACLE(tunnel-config) #
```

2. Use the **name** command to provide a unique identifier for this tunnel instance.

```
ORACLE(tunnel-config) # name sipSignaling
ORACLE(tunnel-config) #
```

3. Use the **local-ip-address** parameter to specify the IP address at the OCSBC end of the tunnel.

 Note:

This address also supports the exchange of CCP messages.

```
ORACLE(tunnel-config) # local-ip-address 1.1.1.100
ORACLE(tunnel-config) #
```

4. Ignore the **remote-mac-address** parameter which is not required for tunnel configuration.
5. Use the **remote-ip-address** parameter to specify the IP address at the SLB end of the tunnel.

 Note:

This address also supports the exchange of CCP messages.

```
ORACLE(tunnel-config) # remote-ip-address 182.16.204.210
ORACLE(tunnel-config) #
```

6. Use the **port** parameter to specify the port used to send and receive cluster control messages.

```
ORACLE(tunnel-config) # port 4444
ORACLE(tunnel-config) #
```

7. Use the **protocol** parameter to specify the transport protocol used in support of cluster control messages.

Supported transport protocol is UDP (the recommended default).

```
ORACLE(tunnel-config) # protocol UDP
ORACLE(tunnel-config) #
```

8. Use the **application** parameter to specify the application protocol supported by this tunnel.

Specify the SIP protocol.

```
ORACLE(tunnel-config)# application SIP
ORACLE(tunnel-config)#
```

9. Use **traffic-policy** to enter the name of the traffic policy that applies to this tunnel (1-128 characters long) as configured on the SLB.

This configuration is a per-tunnel configuration. Once configured, it will be passed on via the CCP protocol to SLB in Heartbeat messages.

The CCD task running on the SLB will extract the traffic policy name and will find the matching traffic-policy configuration on the SLB.

```
ORACLE(tunnel-config)# traffic-policy <pattern>
ORACLE(tunnel-config)#
```

10. Use **done**, **exit**, and **verify-config** to complete configuration of this tunnel-config configuration element.
11. Repeat Steps 1 through 9 to complete tunnel configuration on other SIP interfaces as required.

Sample OCSBC Tunnel Configuration

The following formatted extract from **show running-config** ACLI output shows a sample OCSBC (cluster member) configuration.

```
tunnel-config
name                               one
local-ip-address                   1.1.1.100
remote-mac-address
remote-ip-address                  182.16.204.210
port                               4444
protocol                           UDP
tls-
profile                            TLS-LB
traffic-
policy                             tp1
application                         SIP
last-modified-by                    admin@console
last-modified-date                  2013-11-10 23:24:15
```

Note:

This configuration is a per-tunnel configuration. Once configured, it will be passed on via the CCP protocol to the OCSLB in Heartbeat messages.

SIP Configuration

You must ensure there are both **sip-interface** and **sip-config** SIP configurations operating on the Oracle Communications Session Border Controller (OCSBC) for it to participate as a member of an Oracle Communications Subscriber-Aware Load Balancer SLB cluster.

With respect to the **sip-config**, the SLB uses your **hostname** setting as the identification of the OCSBC, for example, within its show commands. This allows you to segregate individual OCSBC behavior on the SLB. In many cases, administrators have already configured an OCSBC **hostname** for other purposes. This setting must be unique on each OCSBC.

```
westy# configure terminal
westy(configure)# session-router
westy(session-router)# sip-config
westy(sip-interface)# select
westy(sip-interface)# sip-config
westy(sip-interface)# hostname SLB-member1
```

In a traditional OCSBC configuration the IP address assigned to a sip-port configuration element is contained within the address space defined by the network interface netmask. This is not the case for clustered OCSBCs. Rather, the IP address assigned to the sip-port is identical to the address of an SLB service-port advertised on the access network. The process of encapsulating the packets between the SLB and OCSBC masks the fact that the IP address the OCSBC expects to receive IP packets on is different than the Layer 5 address the OCSBC expects the SIP address on.

Consistency of realm identification is vital to successful and predictable policy-based load balancing. Take particular care to ensure that the **realm-id** of the sip-interface configuration element mirrors the **lb-realm** assignments made while configuring distribution rules. See the Distribution Policy Configuration section.

In the following configuration example, the **realm-id** is LosAngeles. This OCSBC, when booted, will detect that it is a member of an SLB cluster and register the service port 10.0.0.1:5060/UDP as the realm LosAngeles with the SLB. The SLB will automatically create the OCSBC group LosAngeles (if it doesn't exist) or join the OCSBC to the group LosAngeles (if it is not the first to advertise LosAngeles). Policy statements that direct packets to LosAngeles now consider this OCSBC as a potential destination, assuming the address:port/protocol also are consistent with the policy's matching criteria.

This technique allows you to configure the same IP:port/protocol on multiple OCSBCs, with different realm-id labels, to indicate priority of one OCSBC or group of OCSBCs over another. As an example, consider several OCSBCs geographically situated together with the label LosAngeles, and several other OCSBCs geographically situated elsewhere with the label NewYork, all with the identical SIP interface and SIP port configuration. A policy can be easily defined to give preference to a source subnet of users in California to the LosAngeles member OCSBCs, with NewYork as a second priority. This provides flexibility in network design without undue burden in the configuration: OCSBCs' tagged with the same realm name are joined in dynamically created OCSBC groups by the SLB, with no explicit configuration required on the SLB whatsoever.

1. From superuser mode, use the following ACLI command sequence to access sip-interface configuration mode. While in this mode, you verify the **realm-id** and assign the newly created IP-in-IP tunnel to a SIP interface.

```
westy# configure terminal
westy(configure)# session-router
westy(session-router)# sip-interface
westy(sip-interface)# select
<realm-id>: LosAngeles
1: LosAngeles 172.192.1.15:5060
selection: 1
westy(sip-interface)# show
sip-interface
      state          enabled
      realm-id       LosAngeles
      ...
      ...
      ...
westy(sip-interface)#
```

2. Use the **tunnel-name** parameter to assign the IP-in-IP tunnel to the current SIP interface.

```
westy(sip-interface)# tunnel-name sipSignaling
westy(sip-interface)# ?
```

3. Use the **sip-port** command to move to sip-port configuration mode.

```
westy(sip-interface)# sip-port
westy(sip-port)# ?
address          IP Address
port             port (default: 5060)
transport-protocol transport protocol
tls-profile       the profile name
allow-anonymous  allowed requests from SIP realm
ims-aka-profile  ims-aka profile name
select           select a sip port to edit
no              delete a selected sip port
show            show sip port information
done            write sip port information
exit            return to previous menu
westy(sip-port)#
```

4. Use the **address**, **port**, and **transport-protocol** parameters to mirror the address of an existing SLB service port.

```
westy(sip-port)# address 10.0.0.1
westy(sip-port)# port 5060
westy(sip-port)# transport-protocol udp
westy(sip-port)#
```

5. Use **done**, **exit**, and **verify-config** to complete configuration of this sip-port configuration element.
6. Repeat Steps 1 through 5 as necessary to verify **realm-ids**, assign IP-in-IP tunnels, and create mirrored service ports on additional SIP interfaces.

Online Offline Configuration

The **set-system-state** CLI command provides the ability to temporarily place a clustered OCSBC in the offline state. The offline setting puts the OCSBC into a state where it is powered on and available only for administrative purposes.

The transition to the offline state is graceful in that existing calls are not affected by the state transition. The OCSBC informs the Oracle Communications Subscriber-Aware Load Balancer (SLB) of the impending status change via a CCP message. Upon receiving such a message, the SLB ceases to forward new endpoints to the OCSBC, and places the OCSBC in the Shutdown state. The OCSBC, for its part, enters a state that results in the rejection of any incoming out-of-dialog SIP requests. Eventually all calls complete, registrations expire and are removed by the SLB, and returning endpoints are allotted to active OCSBCs.

Use the **set-system-state offline** CLI command to place an OCSBC in the offline state.

```
ORACLE# set-system-state offline
Are you sure you want to bring the system offline? [y/n]?: y
Setting system state to going-offline, process will complete when all current
calls have completed
ORACLE#
```



Note:

An OCSBC in the offline state plays no role in a balance or rebalance operation.

In a similar fashion use the **set-system-state online** CLI command to place an OCSBC in the online state.

```
ORACLE# set-system-state online
Are you sure you want to bring the system online? [y/n]?: y
Setting system state to online
ORACLE#
```

IMS-AKA and TLS Support

The Oracle Communications Subscriber-Aware Load Balancer (SLB) supports IMS-AKA and TLS traffic, forwarding it to and from the Session Border Controllers (SBCs) within IP-over-IP tunnels. Both IPv4 and IPv6 are supported.

IMS-AKA and TLS traffic support requires configuration on the SLB and the OCSBC:

- For IMS-AKA:
 - The SLB requires a **service-port** configured with **port 0** and **protocol ALL**.
 - The OCSBC requires a dedicated range of client ports and a dedicated server port configured in the IMS-AKA config to accommodate encrypted traffic.
- For TLS:
 - The SLB requires the applicable **service-port** configured with either:

- * The applicable **service-port** configured with the **protocol** value of **ALL** and the **port** configured with **0**, or
- * The applicable **service-port** configured with the **protocol** value of **TCP** and the **port** configured with the correct number.
- The OCSBC requires normal TLS configuration, as described in the *ACLI Configuration Guide*.

OCSLB Configuration for IMS-AKA and TLS Traffic

The user makes the settings below to the applicable **cluster-config** element on the Oracle Communications Subscriber-Aware Load Balancer (SLB) to support IMS-AKA and TLS traffic. These setting allow this support by preventing the SLB from restricting the type of traffic supported by the cluster.

1. From superuser mode, use the following ACLI command sequence to access the **cluster-config** element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# cluster-config
ORACLE(cluster-config)#service-ports
ORACLE(service-port)#
```

2. Use the **port** parameter to specify the port monitored by the SLB for incoming signaling messages.

The required setting for IMS-AKA or TLS is **0**.

```
ORACLE(service-port)# port 0
ORACLE(service-port)#
```

3. Use the **protocol** parameter to choose the transport protocol.

The required setting for IMS-AKA or TLS is **ALL**.

```
ORACLE(service-port)# protocol ALL
ORACLE(service-port)#
```

4. Use **done**, **exit**, and **verify-config** to complete configuration of this tunnel-config configuration element.
5. Use **done**, **exit**, and **verify-config** to complete configuration of the OCSBC **ims-aka-profile**.

OCSBC Configuration for IMS-AKA Traffic

The user makes the settings below on the Oracle Communications Session Border Controller (OCSBC) to support IMS-AKA traffic while operating with the Oracle Communications Subscriber-Aware Load Balancer.

1. On each OCSBC in the cluster, access the **ims-aka-profile**.

The required setting for IMS-AKA or TLS is **ALL**.

```
ORACLE# configure terminal
ORACLE(configure)#security
```

```
ORACLE (security) #ims-aka-profile
ORACLE (ims-aka-profile) #
```

2. Use the **start-protected-client-port** parameter to set the starting port number for the range of ports needed for IMS-AKA.

The example below show the start port as **4061**.

```
ORACLE (ims-aka-profile) #start-protected-client-port 4061
ORACLE (ims-aka-profile) #
```

3. Use the **end-protected-client-port** parameter to set the end port number for the range of ports needed for IMS-AKA.

The example below show the end port as **4063**.

```
ORACLE (ims-aka-profile) #start-protected-client-port 4063
ORACLE (ims-aka-profile) #
```

4. Use the **protected-server-port** parameter to set the server port needed for IMS-AKA.

The example below show the end port as **4060**.

```
ORACLE (ims-aka-profile) #start-protected-client-port 4060
ORACLE (ims-aka-profile) #
```

5. Use **done**, **exit**, and **verify-config** to complete this configuration.

Displaying Encrypted Traffic Detail on the OCSBC

The syntax of the **show sipd** CLI command allows for the **tunnel** parameter, which displays Oracle Communications Subscriber-Aware Load Balancer (SLB) tunnel statistics on the Oracle Communications Session Border Controller (OCSBC). Sample output, which shows static operational information at the top and specific port statistics at the bottom, is shown below.

```
ORACLE# show sipd tunnel
```

```
|Tunnel          : 11|182.16.209.48|182.16.209.1
-----
|Conf Name       : M01:11/one
|State           : InService      Cur Ping TMOs   : 0/5
|LB              : hermes         Total Ping TMOs : 0
|SBC Atoms       : 0              Service Ports   : 20/20
|SP Atoms        : 0              Application      : SIP
|Last Event      : srcAddrAware   Purge Timer     : 
|Timer Event     : Heartbeat      Timer           : 1604ms
|IPT Handle      : 0x3f           Network Intf    : 0|1.11
|ActiveLbId      : 3              Waiting On      : 
|LostCtlCount    : 0              Last Lost Cntrl : Never
|NextCfgCheck    :                CCP Version      : 7/7
|Source Key      : src-addr
|
|Service Port                               Prev| Cur|Next lbStat Handle Atoms
|-----|-----|-----|-----|-----|-----
|Realm192p1:192.168.209.1:4060<6>          CRng IS  -   200   513   0
|Realm192p1:192.168.209.1:4060<17>         CRng IS  -   200   518   0
```

Realm192p1:192.168.209.1:4061<6>	CRng IS	-	200	514	0
Realm192p1:192.168.209.1:4061<17>	CRng IS	-	200	519	0
Realm192p1:192.168.209.1:4062<6>	CRng IS	-	200	515	0
Realm192p1:192.168.209.1:4062<17>	CRng IS	-	200	520	0
Realm192p1:192.168.209.1:4063<6>	CRng IS	-	200	516	0
Realm192p1:192.168.209.1:4063<17>	CRng IS	-	200	521	0
Realm192p1:192.168.209.1:5060<6>	CRng IS	-	200	517	0
Realm192p1:192.168.209.1:5060<17>	CRng IS	-	200	522	0

To understand the command's service port output, consider the scenario where the user configures the OCSBC as shown in the section titled *SBC Configuration for IMS-AKA Traffic*. This configuration defines the protected port range over which IMS-AKA traffic moves between the OCSBC and the SLB.

The user also typically configures two **sip-ports** on the OCSBC to accommodate IMS-AKA. (This is true regardless of whether the OCSBC supports IMS-AKA behind an SLB or directly.) When configured in conjunction with the protected port range configured in *SBC Configuration for IMS-AKA Traffic*, the OCSBC creates SLB service ports for IMS-AKA in addition to the two ports listening on 5060. The **show sipd tunnel**, therefore, displays statistics for all TCP and UDP ports. In the command output below, <6> indicates a TCP port and <17> indicates a UDP port. Ports using IPv6 can exist simultaneously, and would also be displayed by the command.

SLB VLAN Configuration

You can configure the SLB to use overlapping IP addressing between the access network and SBC clusters, supported by VLANs. When configured, the SLB includes VLANs within its ingress and egress traffic matching processes for policy-based load balancing. This allows it to operate within an access network that uses VLANs. These deployments also require applicable configuration on SBC cluster members. This feature is supported for both IPv4 and IPv6 addressing.

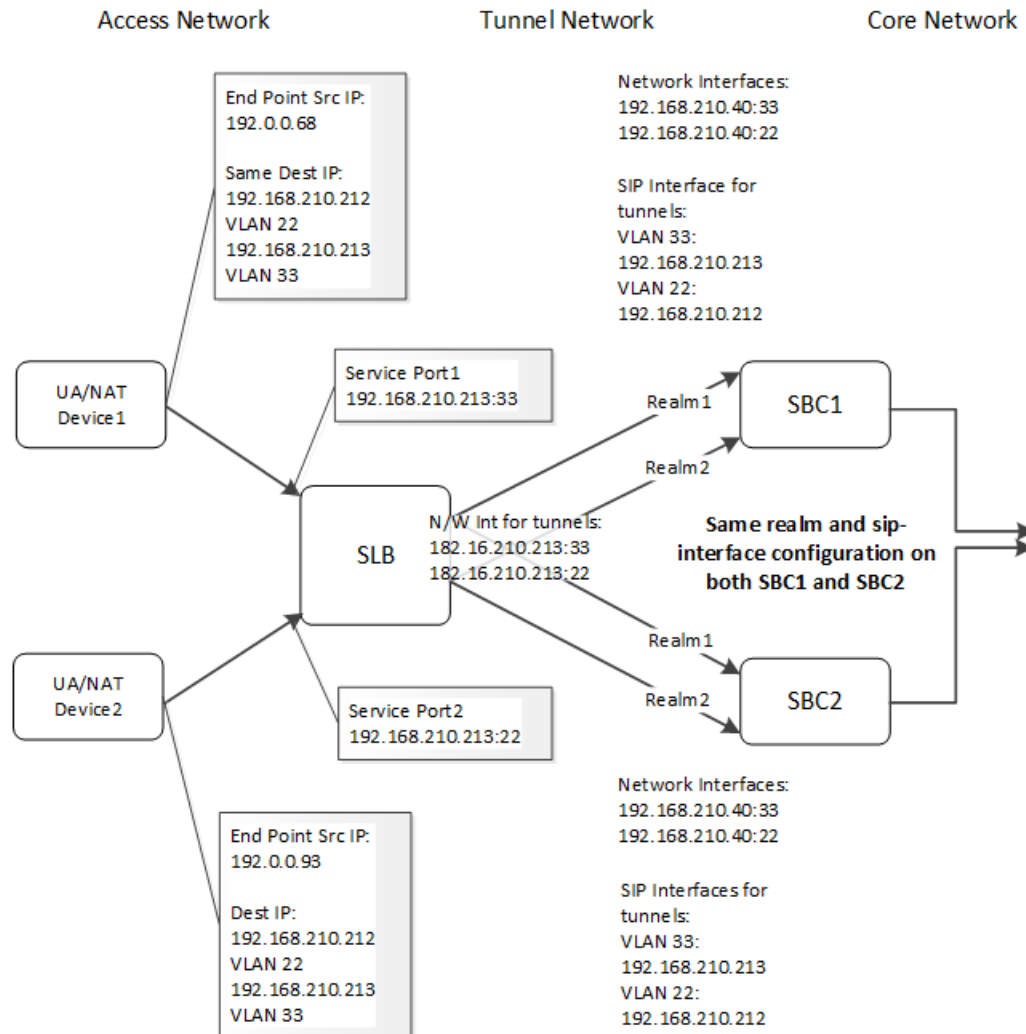
When you deploy an SLB cluster to support VLANs on the access network, you configure the SLB to recognize and retain VLAN association on its ingress (access) interfaces. The use of tunneling between the SLB and the SBC cluster members encapsulates this traffic, allowing the SLB/SBC load balancing function to operate exactly as it does within environments that do not use VLANs on their access networks. In addition, this architecture supports SLB control traffic operation that is independent of the service traffic.

Configurations that vary from deployments that do not use VLANs includes:

- Access Network—Configure the SLB **service-port** elements with the same IP/port/protocol, but with different VLANs.
- Tunnel Network:
 - SBC—Configure **sip-interface** elements facing the SLB to separate realms using the **realm-id** parameters. This results in having one tunnel per **sip-interface**. This design scales to 250 tunnels.
 - SLB—Configure global parameters for **vlan-based-routing** in the **lbp-config** and **sip-config**, and specify the applicable VLANs in the **network-interface** parameters of the **service-ports**. This design scales to 1250 tunnels.
- Core Network—No variations.

By using tunneling, you can configure the **sip-interface** elements that face the SLB identically on all SBCs in the cluster for each VLAN. You maintain unique IP addressing for the tunnels' outer packets on each **network-interface**. Each **sip-interface** can be thought of as having a

“local instance” at the SLB. The connections to both realms on the SBCs represent the load-balancing mechanism driven by the configured load-balancing policy.



Service and CCP traffic is supported as follows:

- When it receives a SIP message at its service port, the SLB selects an SBC, encapsulates the IP packet, and sends it using the correct **network-interface** IP address. Upon reception, the SBC de-encapsulates the packet and processes the message based on the inner IP packet.
The SLB directs incoming SIP messages to different SBCs based on their layer-3 source IP address. The SLB does not look at any IP addresses in the SIP message itself.
- Cluster Control Protocol (CCP) messages are not tunneled. These messages go between the SLB **network-interface** IP address and the **network-interface** IP addresses for each SBC in the cluster. The system only sends SIP traffic through tunnels.

Configuration

To deploy SLB in front of SBCs with OLIP, VLAN tagging on the service interfaces exposed to customer is required. To support VLANs, the SLB supports the same address, port and protocol with different VLAN. In addition, you must also configure cluster SBCs with the same VLAN that is configured on the SLB service port.

To enable this feature on the SLB, you first enable the parameter shown below, in the **lbp-config**, and reboot.

```
ORACLE(lbp-config)#vlan-based-routing enabled
```

Next, enable the option shown below, in the **sip-config**, and reboot.

```
ORACLE(sip-config)#options +vlan-based-routing
```

Beyond these global configurations, you configure **sip-interface**, **network-interface** and **service-port** as conceptually described above. Task steps and a sample configuration are provided below.

**Note:**

When operating the SBC as a PNF (ACME Packet platform), be aware that the default value for the **arp-msg-bandwidth** parameter is 32000. To handle ARP DOS thresholds within the context of this feature, configure **arp-msg-bandwidth** to 50000. To establish the same support when operating as a VNF (vSBC), set the **max-arp-rate** parameter to 30.

OCSBC Configuration for VLANs

The user makes the settings below on the Oracle Communications Session Border Controller (OCSBC) to support VLANs on the SLB access side while operating with the Oracle Communications Subscriber-Aware Load Balancer.

1. On each OCSBC in the cluster, access the **sip-config**.

```
ORACLE# configure terminal
ORACLE(configure)#session-router
ORACLE(session-router)#sip-config
ORACLE(sip-config)#
```

2. To enable this feature on the SLB, you must enable the parameter shown below, in the **sip-config**. This parameter requires that you reboot. This procedure performs the reboot after additional configuration.

```
ORACLE(sip-config)# vlan-based-routing enabled 0
ORACLE(sip-config)#
```

3. Use **done**, **exit** to return to the **session-router** prompt.
4. On each OCSBC in the cluster, access your **network-interface** elements.

```
ORACLE(session-router)#network-interface
ORACLE(network-interface)#
```

5. Use **done**, **exit**, and **exit** to return to the configure prompt.
6. Use the **network-interface** element to specify the tunnels to be used for service traffic.

The example below shows a single **network-interface** objects supporting two tunnels.

```
network-interface
  name M10
  sub-port-id 33
  ip-address 186.16.210.218
  netmask 255.255.0.0
  tunnel-config
    name T100
    local-ip-address 182.16.210.218
    remote-ip-address 182.16.210.213
    application SIP
  name M10
  sub-port-id 22
  ip-address 186.16.210.218
  netmask 255.255.0.0
  tunnel-config
    name T99
    local-ip-address 192.168.210.218
    remote-ip-address 192.168.210.213
    application SIP
```

7. On each OCSBC in the cluster, access the **network-interface**.

```
ORACLE(configure)#system-config
ORACLE(session-router)#sip-interface
ORACLE(sip-interface)#
```

8. Use the **sip-interface** element to configure your tunnels.

The example below shows two sip-ports configured for tunnels using the same IP address over two different tunnels.

```
sip-interface
  realm-id Realm1
  description
  sip-port
    address 192.168.210.213
    port 5063
    transport-protocol TCP
    allow-anonymous registered
  registration-caching enabled
  secured-network enabled
  tunnel-name T99
sip-interface
  realm-id Realm2
  description
  sip-port
    address 192.168.210.213
    port 5063
    transport-protocol TCP
    allow-anonymous registered
  registration-caching enabled
  secured-network enabled
  tunnel-name T100
```

9. Use **done**, **exit**, and **verify-config** to complete this configuration.
10. Reboot you SBC.

OCSLB Configuration for VLANs

The user makes the settings below to the **lbp-config** element and the **lbp-config** element on the SLB to support VLANs on its access side.

1. From superuser mode, use the following ACLI command sequence to access the **lbp-config** element.

```
ORACLE# configure terminal
ORACLE(configure)# session-router
ORACLE(session-router)# lbp-config
ORACLE(lbp-config)#
```

2. To enable this feature on the SLB, you must enable the parameter shown below, in the **lbp-config**. This parameter requires that you reboot. This procedure performs the reboot after additional configuration.

```
ORACLE(lbp-config)# vlan-based-routing enabled 0
ORACLE(lbp-config)#
```

3. To enable this feature on the SBC, you must enable the option shown below, in the **sip-config**, and reboot.

```
ORACLE(sip-config)# options +vlan-based-routing
ORACLE(sip-config)#
```

This parameter requires that you reboot. This procedure performs the reboot after additional configuration.

4. Use the **protocol** parameter to choose the transport protocol.

The required setting for IMS-AKA or TLS is **ALL**.

```
ORACLE(service-port)# protocol ALL
ORACLE(service-port)#
```

5. Use **done**, **exit**, and **verify-config** to complete configuration of this tunnel-config configuration element.

```
cluster-config
  log-level          NOTICE
  source-rebalance-threshold 0
  rebalance-skip-ahead 2000
  ...
  service-port
    address          192.168.210.213
    port              5060
    network-interface M00:33
  service-port
    address          192.168.210.213
    port              5060
    network-interface M00:22
```

6. Use **done**, **exit**, and **verify-config** to complete VLAN configuration of the SLB.
7. Reboot you SLB.

Sample VLAN Configuration

The following formatted extract from **show running-config** CLI output shows a sample tunnel configuration.

On the SLB ...

```
cluster-config
  log-level                                NOTICE
  source-rebalance-threshold              0
  rebalance-skip-ahead                    2000
...
  service-port
    address                               192.168.180.1
    port                                  5060
    network-interface                      M00:100
  service-port
    address                               192.168.180.1
    port                                  5060
    network-interface                      M00:102
```

On the SBC

```
sip-interface
  realm-id                                Realm192p98
  description
  sip-port
    address                               192.168.248.52
    port                                  5063
    transport-protocol                    TCP
    allow-anonymous                       registered
  sip-port
    address                               192.168.248.52
    port                                  5063
    transport-protocol                    TCP
    allow-anonymous                       registered
  registration-caching                    enabled
  secured-network                          enabled
  tunnel-name                             T99
sip-interface
  realm-id                                Realm192p99
  description
  sip-port
    address                               192.168.248.52
    port                                  5063
    transport-protocol                    TCP
    allow-anonymous                       registered
  sip-port
    address                               192.168.248.52
    port                                  5063
    transport-protocol                    TCP
```

allow-anonymous	registered
registration-caching	enabled
secured-network	enabled
tunnel-name	T100

As mentioned earlier, customers use unique IPs for each **sip-interface** and the same **realm-id** on the **network-interface** that we configured for your tunnel. This configuration is the same for each **sip-interface**. This means a single tunnel is sufficient when you are using unique IPs.

To achieve OLIP, however, you use different realm-ids across sip-interfaces, and configure tunnels on each corresponding **network-interface**. This means that OLIP on the access side requires multiple tunnels, as shown below.

```
network-interface
  name M10
  sub-port-id 348
  ip-address 192.168.248.70
  netmask 255.255.0.0
  hip-address 192.168.248.70
  icmp-address 192.168.248.70
  tunnel-config
    name T429
    local-ip-address 192.168.248.70
    remote-ip-address 102.168.248.40
    application SIP
  name M10
  sub-port-id 349
  ip-address 192.168.248.70
  netmask 255.255.0.0
  hip-address 192.168.248.70
  icmp-address 192.168.248.70
  tunnel-config
    name T250
    local-ip-address 192.168.248.70
    remote-ip-address 102.168.248.40
    application SIP
```

4

OCSLB/Cluster Management & Diagnostics

OCSLB Statistics

The Oracle Communications Subscriber-Aware Load Balancer provides the operator with a full set of statistical data for troubleshooting and diagnostic purposes. This section describes current statistical outputs and defines displayed values. It is important to become familiar with the data and the collection process when opening trouble tickets as service personnel will rely upon this information to assist you in diagnosing hardware, software, and/or network issues.

Display Behavior for Show Commands with the Size Argument

Some SLB show commands display very large output for each entry. These commands display the output of first 5 entries if you do not specify the **size** parameter. After displaying the first 5 entries, the system prompts the user to display the next 5 entries. These commands include:

- **show sipd tunnels <size>**
- **show sipd ccp <size>**
- **show ccd ccp <size>**

The system considers what you set for the **size** argument when you include it in your command string. When it does this, however, it first checks whether the size is less than 10. If the number of entries to be displayed is less than 10, the system uses your argument. If the number is greater than 10, however, the system ignores your entry and limits the size to 5 entries.

You can exit the command between output windows by typing **n**.

show balancer

The **show balancer** command is the root of all statistical data pertinent to SLB operation. Below is a list of valid arguments, which are described in further detail in the following sections:

```
ORACLE# show balancer ?
endpoints  show session load balancer endpoints
members    show session load balancer cluster member summary
metrics     show load balancer metrics
realms      show load balancer realms
tunnels     show session load balancer statistics
statistics  show session load balancer IP-in-IP tunnel info
ORACLE#
```

The show balancer Command using the More Argument

The **show balancer** commands listed below enable the **cli-more** command temporarily. This command fills your screen with output, stopping and allowing you to exit or continue showing new pages of information. This process continues until the system has displayed all of the

command's output. After the command has completed, the system disables the **cli-more** function.

The applicable **show balancer** commands include:

- show balancer members
- show balancer realms
- show balancer tunnels
- show balancer tunnels errors
- show balancer tunnels fragments
- show balancer tunnels statistics
- show balancer tunnels fragments reset
- show balancer tunnels statistics reset

show balancer endpoints

The **show balancer endpoints** command displays a full list of all IP-to-OCSBC mappings resident in the Oracle Communications Subscriber-Aware Load Balancer (SLB). As the SLB can hold up to ten million entries, the output of this command can and will grow very large, and extreme caution should be exercised when executing this command on a heavily trafficked SLB system.

```
ORACLE# show balancer endpoints
IP address Port Access Core Flags SBC Handle
-----
15.0.0.24 5060 00134324 10134324 c0000000 1023 [wigglytuff]
15.0.0.22 5060 00134323 10134323 c0000000 1022 [jigglypuff]
15.0.0.20 5060 00134322 10134322 c0000000 1021 [tuono]
15.0.0.18 5060 00134321 10134321 c0000000 1020 [superduke]
15.0.0.16 5060 00134320 10134320 c0000000 1023 [wigglytuff]
15.0.0.14 5060 00134319 10134319 c0000000 1022 [jigglypuff]
15.0.0.12 5060 00134318 10134318 c0000000 1021 [tuono]
15.0.0.10 5060 00134317 10134317 c0000000 1020 [superduke]
15.0.0.8 5060 00134316 10134316 c0000000 1023 [wigglytuff]
15.0.0.6 5060 00134315 10134315 c0000000 1022 [jigglypuff]
15.0.0.4 5060 00134314 10134314 c0000000 1021 [tuono]
15.0.0.2 5060 00134313 10134313 c0000000 1020 [superduke]
ORACLE#
```

The table provided by **show balancer endpoints** displays every endpoint mapping. In the above example, note that IP addresses in the 15.0.0.0/24 space are being distributed among a number of SLBs. The IP address and Port columns pinpoint a specific endpoint. The Index, Address, and Flags columns contain SLB internal reference identifiers for locating that specific endpoint in memory. The SLB Handle column identifies which SLB serves that endpoint; use the **show balancer members** command to display a mapping of SLB names to SLB handles.

You can use optional command arguments to filter/restrict command output.

show balancer endpoints address <ip-address> restricts the display to one endpoint.

For example:

```
show balancer endpoints address 15.0.0.232
```

displays data for the specified IP endpoint.

show balancer endpoints address <ip-address>/<port_num> restricts the display to a specific port on a specific IP address.

For example:

```
show balancer endpoints address 15.0.0.232:5060
```

Displays data for port 5060 on the specified endpoint.

show balancer endpoints address <ip-address>/<bit-mask-len> restricts the display to a contiguous range of endpoint addresses.

For example:

```
show balancer endpoints address 15.0.0.0/24
```

Displays data for the 15.0.0.0 subnet.

```
show balancer endpoints address 15.0.0.240/28
```

Displays data for endpoint addresses 15.0.0.240 through 15.0.0.255.

```
show balancer endpoints <ip-address>/<bit-mask-len><:port_num>
```

Displays data for a specific port on a contiguous range of endpoint addresses.

show balancer members

The **show balancer members** command provides a list of all OCSBCs that have registered with the Oracle Communications Subscriber-Aware Load Balancer (SLB).

```
ORACLE# show balancer members
SBC Name                Source Address  Destination Address  S/P/VLAN
Endpoints
-----
-----
1020 superduke          68.68.68.100    68.68.68.5          0/0/0
3
1021 tuono              68.68.68.100    68.68.68.4          0/0/0
3
1022 jigglypuff         68.68.68.100    68.68.68.1          0/0/0
3
1023 wigglytuff         68.68.68.100    68.68.68.2          0/0/0
3

          max endpoints: 12
        max untrusted endpoints: 200
          current endpoints: 12
```

```

current untrusted endpoints: 0
                        current SBCs: 4
ORACLE#

```

The OCSBC column contains the OCSBC handle, an internal shorthand that identifies a specific OCSBC. The **show balancer members** command provides a handle-to-hostname mapping.

Name contains the SBC hostname.

Source IP contains the local (SLB) tunnel address.

Destination IP contains the remote (SBC) tunnel address.

Slot, Port, and Vlan identify the local interface that supports the SLB-to-OCSBC tunnel.

endpoints contains the number of endpoint-SBC associations that the SLB created for each specific OCSBC,

max endpoints contains the licensed capacity of the OCSBC.

max untrusted endpoints contains the maximum allowed number of untrusted endpoints.

current endpoints contains the current number of endpoints, trusted and untrusted

current untrusted endpoints contains the current number of untrusted endpoints.

show balancer metrics

The Oracle Communications Subscriber-Aware Load Balancer (SLB)'s **show balancer metrics** command displays a comparison between the number of local endpoints (that is, the associations between source addresses and each OCSBC) and the number of remote endpoints (that is, what the OCSBC reports to the SLB as the number of endpoints it has received via the tunneled interface). In the following output, those two numbers are the same; this is true if and only if there are no users in the access network that have multiple phone lines sourced from the same IP address. Were that the case, the number of remote endpoints would be higher than the number of local endpoints.

This table is populated with the data received in the periodic heartbeats from the OCSBC to the SLB. As these heartbeats are somewhat infrequent (every two seconds by default), the data in this table should only be considered accurate within two seconds.

```

ORACLE# show balancer metrics

```

SBC Name	local epts	remote epts	max reg	CPU	max CPU	Mem%	max Mem	Over Load
93 magichat	0	0	480000	2.7	90.0	0.9	95.0	no
94 westy	0	0	480000	2.7	90.0	0.8	95.0	no
95 samadams	0	0	480000	2.8	90.0	0.7	95.0	no
96 bass	0	0	480000	4.3	90.0	2.8	95.0	no
97 sixtus	0	0	480000	2.9	90.0	12.8	95.0	no
98 newcastle	0	0	480000	2.9	90.0	1.8	95.0	no
99 guinness	0	0	480000	3.6	90.0	0.8	95.0	no

```

ORACLE#

```

Fields descriptions include:

- SBC contains the OCSBC handle.

- Name contains the OCSBC hostname.
- max reg contains the maximum number of endpoints the SLB will send to this specific OCSBC. Its value is derived from the product of the **session-multiplier** parameter in the cluster-config configuration element and the OCSBC's licensed session capacity. The OCSBC passes this value to the SLB during the OCSBC's registration process into the cluster.
- CPU contains the last received information on the CPU percentage from this OCSBC.
- Max CPU contains the threshold percentage at which the OCSBC defines itself as overloaded.
- Mem contains the last received information on the Mem percentage from this OCSBC.
- Max Mem contains the threshold percentage at which the OCSBC defines itself as overloaded.
- Overload displays whether or not the OCSBC is reporting itself as overloaded to the SLB.

show balancer realms

The **show balancer realms** command displays a composite list of realms that all member OCSBCs have registered with the OCSLB.

```
ORACLE# show balancer realms
Realm          SBC Tunnel Name          ref count endpoints
-----
access         99  4092 newcastle          1      53535
access         98  4091 magichat           1      53535
access         97  4090 augustiner          1      53535
access         94  4086 bass                1      53535
access         93  4085 westy               1      53535
access         92  4084 sixtus              1      53535
access         96  4089 guinness            1      53535
net-13         99  4088 samadams            1      62550
net-13         91  4087 stbernie            1      62550
ORACLE#
```

In this example, seven of the nine OCSBCs have registered the realm access and two have registered the realm net-13. The total number of endpoints for each of these services is indicated in the rightmost column. The **ref count** column is reserved for future use.

show balancer statistics

The **show balancer statistics** command displays statistical output pertinent to low-level events on the OCSLB. The contents and output of this command are subject to change, and will be documented in a subsequent document release.

```
ORACLE# show balancer statistics
Balance stats:
-----
LBP agent not found          0
packets dropped by standby    0
max capacity reached drops    0
total packets processed       3
packets dropped in balance     0
group not found               0
```

service not found count	0
untrusted dropped	0
invalid endpoint	0
duplicate ept packet drops	0
insert error	0
Tx packet failed count	0
throttle drops	0
sbc not found drops	0
forwarded duplicates	0
policy miss	0
realm miss	3
throttle skips	0
throttle policy skips	0
EPT mgmt stats:	
untrusted age outs	3
Total Endpoint add reqs	3
Endpoints added	3
ept added as trusted	0
ept add invalid	0
ept added unknown state	0
ept added success	3
ept insert not valid	0
ept insert not pend	0
EPT add errors	0
no endpoint handles	0
already added	0
ept add db fail	0
flow add err	0
datapath add err	0
Total Endpoint update reqs	0
Endpoints updated	0
endpoint not valid	0
endpoint being deleted	0
endpoint already trusted	0
endpoint not found	0
unknown trust state	0
not untrusted pending	0
not trusted pending	0
trusted not valid	0
cbk unknown state	0
bad callback	0
ept update success	0
EPT update errors	0
update param err	0
find group err	0
invalid index	0
flow update err	0
ept upd cbk invalid	0
datapath upd err	0
marked invalid	0
not pending	0
not inserted	0
Total Endpoint remove reqs	3
CCD/RED Endpoint remove reqs	0
Endpoints removed	3
unknown state	0

```

del not wait          0
del not pend          0
ept delete success    3
EPT delete errors     0
find group err        0
endpoint not found    0
invalid index         0
delete in progress    0
endpoint SBC mismatch 0
ept del invalid       0
ept del db fail       0
datapath del err      0
insert wait           0
insert not pend       0
-----
trusted endpoints (EPT db) 0
untrusted endpoints (EPT db) 0
-----
total trusted endpoints    0
total untrusted endpoints  0
total endpoints            0
available endpoint handles 5000000

```

Map / queue sizing:

```

insert size 0
trust  size 0
remove size 0

```

Max requests, etc.

```

g_lbpMaxRequests: 0
g_lbpMaxRequests_highwater: 1
g_lbpMsg_highwater: 0
g_lbp_setEndpointTrustLevel 0
g_lbp_removeEndpoint 0
g_lbp_max_untrusted 10000
g_pendingHAListHighWater 0
g_pendingHADelListHighWater 0

```

show balancer tunnels

When implemented on the OCSLB, the **show balancer tunnels** command generates a list of data for each tunnel between the Oracle Communications Subscriber-Aware Load Balancer (SLB) and its clustered OCSBCs. It includes the tunnel source and destination addresses, as well as an internal switch ID (swid) for this tunnel.

```

ORACLE# show balancer tunnels
1020(1025/1026)::
outer src addr = 68.68.68.100
outer dst addr = 68.68.68.5
slot/port/vlan = 0/0/0
traffic policy selected: "" ; traffic policy configured: implicit defaults.
service: 172.16.2.3:5060 [access] protocols: 17/21588

```

```
1021(1025/1026)::
outer src addr = 68.68.68.100
outer dst addr = 68.68.68.4
slot/port/vlan = 0/0/0
traffic policy selected: "" ; traffic policy configured: implicit defaults.
service: 172.16.2.3:5060 [access] protocols: 17/21588

1022(1025/1026)::
outer src addr = 68.68.68.100
outer dst addr = 68.68.68.1
slot/port/vlan = 0/0/0
traffic policy selected: "" ; traffic policy configured: implicit defaults.
service: 172.16.2.3:5060 [access] protocols: 17/21588

1023(1025/1026)::
outer src addr = 68.68.68.100
outer dst addr = 68.68.68.2
slot/port/vlan = 0/0/0
traffic policy selected: "" ; traffic policy configured: implicit defaults.
service: 172.16.2.3:5060 [access] protocols: 17/21588

ACMEPACKET# show balancer tunnels
errors          fragments      statistics
```

Use the **errors** argument for error reporting and troubleshooting.

```
ORACLE# show balancer tunnels errors
src addr 68.68.68.100 / dst addr 68.68.68.5 / slot 0 / port 0 / vlan 0:
Proto Encaps Errors Decaps Errors
-----
17              0          0

src addr 68.68.68.100 / dst addr 68.68.68.4 / slot 0 / port 0 / vlan 0:
Proto Encaps Errors Decaps Errors
-----
17              0          0

src addr 68.68.68.100 / dst addr 68.68.68.1 / slot 0 / port 0 / vlan 0:
Proto Encaps Errors Decaps Errors
-----
17              0          0

src addr 68.68.68.100 / dst addr 68.68.68.2 / slot 0 / port 0 / vlan 0:
Proto Encaps Errors Decaps Errors
-----
17              0          0

unknown protocol:      0
do not fragment drops: 0
no matching tunnel:    0
service lookup failed: 0
IP frag msg failure:   0
mblk alloc failues:    0
IP frame too large:    0
unknown errors:        0
```

```
unknown errors:          0
ORACLE#show balancer tunnels
errors          fragments    statistics
```

The **show balancer tunnels errors** command can also be executed on an OCSBC cluster member. In this usage, the displayed data is restricted to errors between the specific cluster member and the SLB.

Use the **fragments** argument for information related to packet fragmentation/reassembly details.

```
ORACLE# show balancer tunnels fragments
src addr 68.68.68.100 / dst addr 68.68.68.5 / slot 0 / port 0 / vlan 0:
IP:Port:      172.16.2.3:5060
Proto Encap Pkts Encap Octets Decap Pkts Decap Octets
-----
17          3      1239          0          0

src addr 68.68.68.100 / dst addr 68.68.68.4 / slot 0 / port 0 / vlan 0:
IP:Port:      172.16.2.3:5060
Proto Encap Pkts Encap Octets Decap Pkts Decap Octets
-----
17          3      1240          0          0

src addr 68.68.68.100 / dst addr 68.68.68.1 / slot 0 / port 0 / vlan 0:
IP:Port:      172.16.2.3:5060
Proto Encap Pkts Encap Octets Decap Pkts Decap Octets
-----
17          3      1243          0          0

src addr 68.68.68.100 / dst addr 68.68.68.2 / slot 0 / port 0 / vlan 0:
IP:Port:      172.16.2.3:5060
Proto Encap Pkts Encap Octets Decap Pkts Decap Octets
-----
17          3      1244          0          0
ORACLE#show balancer tunnels
errors          fragments    statistics
```

The **show balancer tunnels fragments** command can also be executed on an OCSBC cluster member. In this usage, the displayed data is restricted to fragmentation operations between the specific cluster member and the SLB.

Use the **statistics** argument for information related to packet counts.

```
ORACLE# show balancer tunnels statistics
src ip 182.16.203.83 / dst ip 182.16.203.87 / slot 0 / port 1 / vlan 0:
IP:Port: 192.169.203.83:5050
Proto Encap Pkts Encap Octets Decap Pkts Decap Octets
-----
6          0          0          0          0
IP:Port: 192.169.203.83:5060
Proto Encap Pkts Encap Octets Decap Pkts Decap Octets
```

```

-----
17          48011      24213914          0          0
src ip 182.16.203.83 / dst ip 182.16.203.86 / slot 0 / port 1 / vlan 0:
IP:Port:  192.169.203.83:5060
Proto Encap Pkts Encap Octets Decap Pkts Decap Octets
-----
17          48017      24217918          0          0
ORACLE#

```

The **show balancer tunnels statistics** command can also be executed on an OCSBC cluster member. In this usage, the displayed data is restricted to traffic counts between the specific cluster member and the SLB.

Cluster Control Protocol Statistics

The CCP provides the operator with a full set of statistical data for troubleshooting and diagnostic purposes.

show ccd

The **show ccd** command is the root of all statistical data pertinent to CCP operation. Below is a list of valid arguments, which are described in further detail in the following sections:

```

ORACLE# show ccd ?
ccp          Cluster Control Protocol Stats
rebalance    Display rebalance queue
reset        Reset Stats
sds          Controlled SDs
stats        Cluster Control Stats
ORACLE#

```

show ccd ccp

The **show ccd ccp** command displays aggregated data (that is, from all cluster members) about specific CCP operations.

Svc Add	Recent	Total	PerMax
=====	=====	=====	=====
Ops Recvd	2	68	2
Op Replies Sent	2	68	2

Status Code	Received			Sent		
	Recent	Total	PerMax	Recent	Total	PerMax
-----	-----	-----	-----	-----	-----	-----
200 OK	0	0	0	1	34	1
404 Not Found	0	0	0	1	34	1

Metrics	Recent	Total	PerMax
=====	=====	=====	=====
Ops Recvd	16	43661	16
Op Replies Sent	16	43662	16

Received			Sent		
-----	-----	-----	-----	-----	-----

Status Code	Recent	Total	PerMax	Recent	Total	PerMax
-----	-----	-----	-----	-----	-----	-----
200 OK	0	0	0	14	43595	15
410 Gone	0	0	0	2	67	2

OverloadMetrics	Recent	Total	PerMax
=====	=====	=====	=====
Ops Recvd	16	43661	16
Op Replies Sent	16	43662	16

Status Code	Recent	Total	PerMax	Recent	Total	PerMax
-----	-----	-----	-----	-----	-----	-----
200 OK	0	0	0	15	43629	15
410 Gone	0	0	0	1	33	1

Prov Done	Recent	Total	PerMax
=====	=====	=====	=====
Ops Recvd	1	34	1
Op Replies Sent	1	34	1

Status Code	Recent	Total	PerMax	Recent	Total	PerMax
-----	-----	-----	-----	-----	-----	-----
200 OK	0	0	0	1	34	1

Stop Down	Recent	Total	PerMax
=====	=====	=====	=====
Ops Recvd	3	98	3
Op Replies Sent	3	99	3

Status Code	Recent	Total	PerMax	Recent	Total	PerMax
-----	-----	-----	-----	-----	-----	-----
200 OK	0	0	0	3	98	3
410 Gone	0	0	0	0	1	1

Use the hostname argument to display data for a specific cluster member.

ORACLE# show ccd ccp westy

westy

Svc Add	Recent	Total	PerMax
=====	=====	=====	=====
Ops Recvd	0	20	20
Op Replies Sent	0	20	20

Status Code	Recent	Total	PerMax	Recent	Total	PerMax
-----	-----	-----	-----	-----	-----	-----
200 OK	0	0	0	0	20	20

EP Promo	Recent	Total	PerMax
----------	--------	-------	--------

```

=====
Ops Recvd          0      1000    1000
Duplicate Ops      0        1        1
Op Replies Sent    0      1000    1000

----- Received -----
Status Code      Recent      Total    PerMax
-----
200 OK           0        0        0

----- Sent -----
Recent      Total    PerMax
-----
0          1000    1000

Metrics
=====
Recent      Total    PerMax
=====
Ops Recvd      24     17517     15
Op Replies Sent 24     17517     15

----- Received -----
Status Code      Recent      Total    PerMax
-----
200 OK           0        0        0
410 Gone         0        0        0

----- Sent -----
Recent      Total    PerMax
-----
24         17516     15
0           1         1

OverloadMetrics
=====
Recent      Total    PerMax
=====
Ops Recvd      24     17517     15
Op Replies Sent 24     17517     15

----- Received -----
Status Code      Recent      Total    PerMax
-----
200 OK           0        0        0

Prov Done
=====
Recent      Total    PerMax
=====
Ops Recvd      0        1        1
Op Replies Sent 0        1        1

----- Received -----
Status Code      Recent      Total    PerMax
-----
200 OK           0        0        0

Stop Down
=====
Recent      Total    PerMax
=====
Ops Recvd      0        2        2
Op Replies Sent 0        2        2

----- Received -----
Status Code      Recent      Total    PerMax
-----
200 OK           0        0        0

ORACLE#

```

show ccd sds

The **show ccd sds** command displays a table containing an overview of all of the data gleaned from the CCP from each OCSBC. The syntax for this command extends to allow individual SBC IDs or the **all** and **size** arguments. The **size** argument is explained in the Display Behavior for Show Commands with the Size Argument topic.

The full syntax of the command includes:

An example of the basic command, **show ccd sds**, is shown immediately below.

```
ORACLE# show ccd sds
Session Director      Hdl  State      Tunnel Svcs Version      HW      LastPing
-----
augustiner            95  InService  1/1      2  6.2.0.30b8  VM      1966ms
bass                  94  InService  1/1      2  6.2.0.30b8  VM      1966ms
guinness              96  InService  1/1      2  6.2.0.30b8  VM      1966ms
magichat              97  InService  1/1      2  6.2.0.30b8  VM      1966ms
newcastel             98  InService  1/1      2  6.2.0.30b8  VM      1966ms
samadams              99  InService  1/1      2  6.2.0.30b8  VM      1966ms
sixtus                92  InService  1/1      2  6.2.0.30b8  VM      1966ms
stbernie              91  InService  1/1      2  6.2.0.30b8  VM      1966ms
westy                 93  InService  1/1      2  6.2.0.30b8  VM      1966ms
ORACLE#
```

Field descriptions include:

- Session Director contains the hostname of the cluster OCSBCs that are connected to the SLB.
- Hdl contains the clustered OCSBC handle, an internal shorthand that identifies a specific cluster member. The **show balancer members** command provides a handle to hostname mapping.
- State contains the current OCSBC state. Valid states are:
 - Init — during initial handshaking with the SLB
 - InService — healthy and operating normally
 - Rebalance — during a cluster expansion/contraction operation
 - LostControl — no longer communicating with the SLB
- Tunnel displays (the number of tunnels in service)/(the number of tunnels configured on the OCSBC).
- Svcs contains the number of advertised services (protocols) that the OCSBC has negotiated with the SLB.
- Version contains the software version running on that OCSBC.
- HW identifies the hardware platform (in this case, **VM** identifies virtual machines).
- LastPing is not currently used.

When issued with an optional hostname argument, the **show ccd sds <id>** command provides a detailed report for the target hostname.

```
ORACLE# show ccd sds bass
Session Director: bass
```

```
+-----+
|State      : InService      Handle      : 0x3ff
|Tunnels    : 1              ServicePorts : 20
|HW Type    : VM             SW Version   : 8.1.0(53)
|Last Ping  : 312ms          Remote State : Online
|App Count  : 1              Active Id   : 4
|
|Service:      App  SvcPorts Tunnels Endpoints DropCount
+-----+-----+-----+-----+-----+-----+
|Realm192p1    SIP    10      1        0          0
|Realm192p1_v66 SIP    10      1       1000        0
|
| Tunnel#: 0
+-----+
| ID: (11|182.16.209.1|182.16.209.56)
| App: SIP
| Handle: 0x3ff
| Svcs: 20
| LastHB: 312ms
| Traffic Policy: Implicit Defaults
|
|# CPU    MAX  CurReg RegLimit CurSes  MaxSess State  CtlVer Mem% Max  OverLoad
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|0 0.0% 90.0% 1000  0          0      80000  InSer  7/7   38.0 95.0  no
| 0.0% 90.0% 1000 800000  0      80000
|
|Overloads Reported      : 0
|Causes: Memory Threshold Exceeded (0); Thread Overload- SIP (0),
|MBCD (0); Other (0)
|
|Service Port                                App  Handle  TunNdx Avail
+-----+-----+-----+-----+-----+-----+
|Realm192p1:192.168.218.7:4060<6>          SIP  513(1)   0    yes
|Realm192p1:192.168.218.7:4060<17>         SIP  514(2)   0    yes
```

```
ORACLE#
```

State

- State — the current OCSBC state
- Handle — the OCSBC handle
- Tunnels — the current number of OCSBC tunnels
- ServicePorts — the current number of OCSBC service ports
- HW Type — the hardware platform (in this case, SD3 identifies an Acme Packet 4500 OCSBC)
- SW Version — the installed software revision level

- Last Ping — the number of elapsed milliseconds, since a ping/keepalive was received from this OCSBC
- App Count — the number of applications supported by the OCSBC

Services State

- Service — the realm advertised by the OCSBC in the Service Port ID
- App — the supported protocol: SIP
- SVCPorts — the current number of service ports
- Tunnels — the current number of tunnels
- endpoints — the cumulative number of endpoints for this service
- DropCount — the number of elements to drop when rebalancing this OCSBC

Tunnel State

- # — the tunnel index (0 or 1)
- Tunnel — the SLB and OCSBC tunnel IP address
- App — the supported protocol: SIP
- Handle — the handle for the tunnel
- Svcs — the number if service ports supporting the tunnel
- LastHB — the number of elapsed milliseconds since a heartbeat was received from the remote end of this tunnel

Service Port Data

- Service Port — the service path (the concatenation of realm, IP address, port number, and IP Level 4 protocol number — 17 for UDP, 6 for TCP)
- App — the supported protocol: SIP
- Handle — the handle for the service port
- TunNdx — the tunnel the service port is registered for
- Avail — current availability (yes or no) determined by the presence of heartbeats

When issued with an optional hostname argument and the optional tunnel metrics argument, the **show ccd sds <id> <tunnel metrics>** command provides a detailed report on specific system capacities, as reported by the target hostname via CCP.

Tunnel Metrics Data

- # — the tunnel number (0 or 1)
- CPU — the current CPU utilization rate
- Max — the maximum supported CPU utilization rate, if this value is exceeded, the tunnel implements a load limit algorithm
- CurReg — the current number of registrations supported by the OCSBC
- RegLimit — the maximum number of registrations supported by the OCSBC
- CurSess — the current call count reported by the OCSBC
- MaxSess — the maximum sessions for which the OCSBC is licensed
- State — whether or not the tunnel is in service
- Mem% — the current memory utilization

- **Max** — the maximum supported memory utilization rate, if this value is exceeded, the tunnel implements a load limit algorithm
- **OverLoad** — whether or not the OCSBC is reporting itself overloaded, and therefore out of contention for accepting new traffic

show ccd stats

The **show ccd stats** command displays endpoint statistics for the OCSBC members of the cluster.

```
ORACLE# show ccd stats
17:10:09-54
```

SD	Active	----- Period -----			---- LifeTime ----		
		Rate	High	Total	Total	PerMax	High
bass	I285714	0.0	285714	0	285.71K	13.76K	285.71K
guinness	I285714	0.0	285714	0	285.71K	13.76K	285.71K
magichat	I285714	0.0	285714	0	285.71K	13.76K	285.71K
newcastel	I285714	0.0	285714	0	285.71K	13.76K	285.71K
samadams	I285714	0.0	285714	0	285.71K	13.76K	285.71K
sixtus	I285714	0.0	285714	0	285.71K	13.76K	285.71K
westy	I285714	0.0	285714	0	285.71K	13.76K	285.71K

```
Total endpoints: 153908
Total rate      : 0.0
Total SDs      : 9
ORACLE#
```

The Period stats provided represent an accumulation of data for the amount of time specified after the dash separator in the timestamp printed in the first line of output (in this example, the period represents 54 seconds).

The single ASCII character between the SD column and the Active column is the state of that OCSBC; the letter I represents InService.

The Rate column displays the transmission rate of new endpoint associations to that particular OCSBC. (In the sample, no new endpoints are arriving in the cluster, so all of the OCSBCs show a rate of 0.0.) The High field indicates the highest number of active endpoint associations for the current period.

When issued with an optional hostname argument, the **show ccd stats** command provides a detailed report for the target hostname.

```
ORACLE# show ccd stats bass
15:09:25-59
```

SD bass		[InService]				
State	Active	High	Total	Total	Permax	High
Tunnels	1	1	0	2	1	1
Service Ports	2	2	0	2	1	2
endpoints	53571	53571	0	53571	14399	53571
Contacts	53571	53571	0	53571	14399	53571
Sessions	0	0	0	0	0	0

```
----- Lifetime -----
Recent          Total          PerMax
```

Heartbeats rcvd	30	27426	15
Heartbeats Missed	0	1	1
Tunnel Adds	0	2	1
Tunnel Removes	0	1	1
Service Adds	0	2	1
Service Removes	0	0	0
endpoint Removes	0	0	0
endpoint Promotes	0	53571	13561
endpoints Skipped	0	0	0
Rebalance Source	0	0	0
Rebalance Targe	0	0	0
Rebalance Request	0	0	0
Rebalance Replies	0	0	0
CPU Above Limit	0	0	0
CPU Above Threshold	0	0	0
Online Transitions	0	0	0
Offline transitions	0	0	0
Tunnel Add Fails	0	0	0
CCD Tunnel Add Fails	0	0	0
Tunnel Remove Fails	0	0	0
CCD Tunnel Remove Fails	0	0	0
Service Add Fails	0	0	0
CCD Svc Add Fails	0	0	0
Service Remove Fails	0	0	0
CCD Svc Remove Fails	0	0	0
Service Adds No Cfg	0	0	0
Bad Service Handle	0	0	0
endpoint Remove Fails	0	0	0
endpoint Prom Fail	0	0	0
ORACLE#			

The **Period** stats provided represent an accumulation of data for the amount of time specified after the dash separator in the timestamp printed in the first line of output (in this example, the period represents 59 seconds).

Tunnels contains the number of tunnels between the SLB and the target OCSBC, in this case, bass.

Service Ports contains the number of Service Ports advertised by the target OCSBC when it joined the cluster.

endpoints and **Contacts** contain the number of endpoint associations the SLB has assigned to the target OCSBC. If there is only one registering device at a given endpoint, a one-to-one correlation between endpoints and contacts is expected. However, if the **atom-limit-divisor** parameter has been set to a non-default value, the number of contacts exceeds the number of endpoints.

Sessions contains the number of active calls.

The **Lifetime** stats provided represent an accumulation of data since the last reboot.

HeartBeats rcvd contains the number of heartbeat/keepalive messages received from the target OCSBC. Heartbeats are sent every two seconds by the OCSBC.

HeartBeats Missed contains the number of scheduled heartbeat/keepalive messages not received from the target OCSBC.

The **Tunnel Adds** and **Tunnel Removes** counters are incremented when an OCSBC joins the cluster and leaves the cluster, respectively.

The **Service Adds** and **Service Removes** counters are incremented when an OCSBC advertises support for a service and withdraws support for a service, respectively. This generally happens only when an OCSBC first joins the cluster, or if the configuration on a clustered OCSBC is changed, saved, and activated.

The **endpoint Removes** counter tracks the number of OCSBC-originated Cluster Control messages that request the SLB to delete a forwarding rule. Such a request can be the result of (1) a rebalance operation (when the SLB asks for the OCSBC to nominate candidates for rebalancing), (2) an endpoint de-registration with the OCSBC, or (3) an endpoint is power down. Generally, whenever a registration cache entry on a clustered endpoint is removed by the OCSBC, it notifies the SLB to remove that binding.

The **endpoint Promotes** counter tracks the number of promotion messages the OCSBC sends to the SLB to validate an untrusted forwarding rule. When the SLB first creates a forwarding rule for a new endpoint, it treats it as untrusted. When the OCSBC receives a 200 OK for a REGISTER message from that endpoint's registrar, the OCSBC sends a Promote Cluster Control message to the SLB. At this point, the SLB modifies the particular forwarding rule and assigns it trusted status. If this Promote message is not received within the time configured as the untrusted-grace-period in the lbp-config, the SLB deletes the untrusted entry.

endpoints Skipped contains the number of endpoints in its registration cache that the OCSBC has skipped over during a rebalance request. Skipping may be done for one of two reasons: either the most appropriate user for rebalancing was in an active phone call (and **rebalance-skip-calls** was enabled in cluster-config), or the **rebalance-skip-ahead** value in cluster-config was set to a nonzero value. In this case, when the OCSBC is asked to nominate users for rebalance, it will skip over any users whose registration cache entry is due to expire within the number of milliseconds set as the **rebalance-skip-ahead** value.

Rebalance Source contains the number of times the target OCSBC was used as a source of endpoints during a rebalance operation (that is, it supplied endpoints to a cluster member that was added to the cluster after itself).

Rebalance Target contains the opposite: the number of times that OCSBC was the recipient of endpoints from other sources during a rebalance operation.

The **Rebalance Requests** and **Rebalance Replies** counters increment upon receipt of a Cluster Control message from the SLB to the OCSBC asking it to divest itself of endpoints, and the responsive Cluster Control message from the OCSBC that indicates the endpoints the OCSBC has chosen.

The **CPU Above Limit** and **CPU Above Threshold** counters increment whenever an OCSBC has reported a high CPU value, and has been taken out of consideration for new endpoint assignments. Generally, the CPU limit and threshold are the same value (90%). However, it is possible to configure the threshold to be lower using the sip-config **option load-limit**.

show ccd rebalance

The **show ccd rebalance** command displays aggregated data (that is, from all cluster members) about specific CCP operations.

The **show ccd rebalance stats** command displays aggregated data from all or from specific cluster members about rebalancing operations.

show ccd rebalance (<stats> <sd-name>)

```
ORACLE# show ccd rebalance
```

This command string instructs the SLB to display the number of requests and drops for each cluster member SBC.

```
ORACLEshow ccd rebalance stats
Session Director          ReqCount      DropCount
-----
X92-vmw-sr-md-vsbc-002    0              0
cranberry                 0              0
dewberry                  0              0
X92-vmw-sr-md-vsbc-001    0              0
X92-vmw-sr-mlx-md-001     0              0
```

This command string instructs the SLB to display the number of requests and drops for all realms on the cluster member SBC named cranberry.

show ccd rebalance stats <sbcs_name>

```
ORACLEshow ccd rebalance stats cranberry
Session Director: cranberry
-----
Seervice                  ReqCount      DropCount
-----
Cran192                   0              0
Cran192p1                 0              0
Cran192p10                0              0
Cran192p100               0              0
Cran192p101               0              0
```

This command string instructs the SLB to display the number of requests and drops for Cran192p1 realm on the cluster member SBC named cranberry.

show ccd rebalance stats <sbcs_name> <realm_name>

```
ORACLEshow ccd rebalance stats cranberry Cran192p1
Session Director: cranberry
-----
Service                  ReqCount      DropCount
-----
Cran192p1                0              0
```

OCSBC Cluster Member Statistics

The OCSBC cluster member also provides the operator with summary statistical data for active endpoints

show sip lb-endpoints

The **show sip lb-endpoints** command displays OCSBC endpoint stats by realm or tunneled service ports, by sip-interface since each SIP interface is uniquely identified by its realm name.

While this command was not changed for the addition of source port keys, there are some important items to note. When all endpoints are behind a NAT and source ports are used in endpoint keys, the number of endpoints should match the number of atoms. Were all endpoints are behind a single NAT, and source address keys in use, There would be many atoms and only one endpoint. Obviously in mixed environments this will be less clear and thus this command less useful. However, in lab environments this can be useful.

```
ORACLE# sho sipd lb-endpoints
```

```
-----  
Realm Endpoint Stats  
-----
```

```
10:57:29-35
```

```
Service Realm192p1
```

	-----	Period	-----	-----	Lifetime	-----
	Active	High	Total		Total	PerMax High
Endpoints	2				2	
2		2			2	2
Atoms	2				2	
2		2			2	2

	-----	Lifetime	-----
	Recent	Total	PerMax
Refreshes	0	0	0
Adds	2	2	2
Low Skips	0	0	0
High Skips	0	0	0
Auth Promo Tries	0	0	0
noTrust Promo Tries	0	0	0
Promo Tries	0	0	0
Remove Conflicts	0	0	0
Remote Deletes	0	0	0
SP Removes	0	0	0
Expiry Deletes		0	0 0
Session Deletes	0	0	0
Session Adds	0	0	0
Move Deletes	0	0	0
Move Del No Tells	0	0	0
SvcMove Deletes	0	0	0
SvcMove Del NoTell	0	0	0
Auth Promotes	0		
0	0		
Auth Deletes	0	0	0
Add Errors	0	0	0
Delete Deny Sess	0	0	0
Delete Deny Reg	0	0	0
Delete Deny Purge	0	0	0
Delete Missing	0	0	0
Delete Errors	0	0	0
Update Deny Purge	0	0	0
Auth Deny Purge	0	0	0
Remote Sess Skips	0	0	0
Remote Del Fails	0	0	0
SP Remove Fails	0	0	0
Expiry Del Fails	0	0	0
Sess Del Fails	0	0	0
Sess Add Fails	0	0	0

Move Del Fails	0	0	0
Move No Tell Fails	0	0	0
SvcMove Del Fails	0	0	0
SvcMv NoTell Fails	0	0	0
Auth Promo Fails	0	0	0
Auth Del Fails	0	0	0
App Cache Dels	0	0	0

show sip ccp

The **show sip ccp** command displays a cluster-member-specific summary of CCP operations.

```
westy# show sip ccp
```

```
-----  
M00:0.4:T2  
-----
```

EP Del	Recent	Total	PerMax
=====	=====	=====	=====
Ops Sent	0	1	1
Op Replies Recvd	0	1	1

			Received			Sent		
Status Code	Recent	Total	PerMax	Recent	Total	PerMax	Recent	Total
-----	-----	-----	-----	-----	-----	-----	-----	-----
200 OK	0	1	1	0	0	0	0	0

EP Promo	Recent	Total	PerMax
=====	=====	=====	=====
Ops Sent	0	992	538
Op Replies Recvd	0	992	538

			Received			Sent		
Status Code	Recent	Total	PerMax	Recent	Total	PerMax	Recent	Total
-----	-----	-----	-----	-----	-----	-----	-----	-----
200 OK	0	992	538	0	0	0	0	0

Metrics	Recent	Total	PerMax
=====	=====	=====	=====
Ops Sent	25	207	15
Op Replies Recvd	25	207	15

			Received			Sent		
Status Code	Recent	Total	PerMax	Recent	Total	PerMax	Recent	Total
-----	-----	-----	-----	-----	-----	-----	-----	-----
200 OK	25	207	15	0	0	0	0	0

Stop Down	Recent	Total	PerMax
=====	=====	=====	=====
Ops Sent	0	2	2
Op Replies Recvd	0	2	2

			Received			Sent		
Status Code	Recent	Total	PerMax	Recent	Total	PerMax	Recent	Total
-----	-----	-----	-----	-----	-----	-----	-----	-----

-----	-----	-----	-----	-----	-----
200 OK	0	2	2 0	0	0
westy#					

5

Subscriber-Aware Load Balancer SNMP Reference

Overview

This chapter provides an overview of SNMP support for Oracle Communications Subscriber-Aware Load Balancer (SLB) features.

Enterprise Traps

The following table identifies the SLB proprietary traps supported by the SLB.

Trap Name	Description
apSLBEndpointCapacityThresholdTrap	Generated when the number of endpoints on the SLB exceeds the configured threshold.
apSLBEndpointCapacityThresholdClearTrap	Generated when the number of endpoints on the SLB falls below the configured threshold.
apSLBUntrustedEndpointCapacityThresholdTrap	Generated when the number of untrusted endpoints on the SLB exceeds the configured threshold.
apSLBUntrustedEndpointCapacityThresholdClearTrap	Generated when the number of untrusted endpoints on the SLB falls below the configured threshold.

License MIB (ap-license.mib)

MIB Object	Object ID 1.3.6.1.4.1.9148.3.5.1.1.1+	Description
apLicenseSLBEndpointCap	.23	SLB endpoint capacity (leaf)

Subscriber-Aware Load Balancer MIB (ap-slb.mib)

MIB Object	Object ID 1.3.6.1.4.1.9148.3.11.1.1+	Description
apSLBStatsEndpointsCurrent	.1	Number of endpoints currently on the SLB.
apSLBStatsEndpointsDenied	.2	Number of endpoints denied by the SLB because the system has reached the maximum endpoint capacity.
apSLBEndpointCapacity	.3	Maximum number of endpoints allowed on the SLB. This value is based on the installed SLB license(s).

MIB Object	Object ID 1.3.6.1.4.1.9148.3.11.1.1+	Description
apSLBEndpointCapacityUpperThresh	.4	Percentage of endpoints relative to maximum threshold capacity.
apSLBEndpointCapacityLowerThresh	.5	Percentage of endpoints relative to minimum threshold capacity.
apSLBStatsUntrustedEndpointsCurrent	.6	Number of untrusted endpoints currently on the SLB.
apSLBStatsTrustedEndpointsCurrent	.7	Number of trusted endpoints currently on the SLB.
apSLBStatsUntrustedEndpointsDenied	.8	The number of untrusted endpoints denied by the SLB due to the total number of untrusted endpoints exceeding the configured maximum threshold.
apSLBStatsUntrustedEndpointsAgedOut	.9	The number of untrusted endpoints aged out of the system because they were not authenticated within the configured grace period.
apSLBUntrustedEndpointCapacity	.10	Maximum number of untrusted endpoints allowed on the SLB. This value is a configured percentage of the maximum endpoint capacity of the system.
apSLBUntrustedEndpointCapacityUpperThresh	.11	Percentage of untrusted endpoint maximum threshold capacity in use.
apSLBUntrustedEndpointCapacityLowerThresh	.12	Percentage of untrusted endpoint minimum threshold capacity percentage.
apSLBStatsClusterMembersCurrent	.13	The number of cluster members assigned to this vSLB.

6

Known Issues and Caveats

Known Issues for Release S-Cz9.3.0

The Known Issues below apply to release S-Cz9.3.0 of the Subscriber Aware Load Balancer (SLB). For Known Issues on products that are Subscriber-Aware Load Balancer (SLB) cluster members, refer to the Oracle Communications Session Border Controllers (OCSBC) product's Release Notes.

ID	Description	Severity	Found In
N/A	When populated with 10 million registrations or more, the SLB may not synchronize all registrations before the default redundancy-config's becoming-standby-time timeout of 180000. Workaround: Set the SLB's becoming-standby-time timeout to 1800000 when handling 10 million registrations or more.	N/A	S-Cz7.3.10
N/A	The Oracle Communications Session Border Controller does not support overlapping IP addresses on Media interfaces between end points and SBCs in deployments that use an Oracle Communications Subscriber Aware Load Balancer for load balancing.	N/A	S-Cz7.3.10

Resolved Known Issues

The following table provides a list of previous Known Issues that are now resolved.

ID	Description	Severity	Found In	Fixed In
22066052	During a HA failover, IPT core miss errors are incremented, which results in retransmissions.	3	S-Cz7.2.10p3	S-Cz9.3.0
20577310	When approaching the maximum supported registrations per second, the user may find that only a small percentage of endpoint registrations correctly expire and must re-register.	4	S-Cz7.2.10	S-Cz9.3.0
26050436	When an SLB is operating in HA mode, the standby periodically sends SYNC messages to the Active. If, for any reason, the Active's response is delayed, the standby resends these SYNCs. If this "resend" happens after the active has responded, the active writes a "Stray response" log message to the log.lbp file. This issue may confuse the user, but does not impact service.	4	S-Cz7.3.10	S-Cz9.0.0
25814014	The capacity-proportional balancing strategy, introduced as a new feature in the OCSLB is not documented in the SLB Essentials Guide.	4	S-Cz8.3.0m1	S-Cz8.3.0m1p1

Caveats for Release S-Cz9.3.0

The caveats below apply to release S-Cz9.3.0 of the Subscriber-Aware Load Balancer (SLB). For Session Border Controller (SBC) Caveats, when they are (SLB) cluster members, refer to the appropriate documentation for the SBC.

Upgrade Readiness Command

Do not run the **check-upgrade-readiness** command on a vSLB running versions SCZ920p8, SCZ920p9, SCZ920p10. For these versions, this command can adversely impact vSLB operation.

OCI Region Deployment

Do not install the OCSLB and the associated OCSBC clusters in different OCI regions. These deployments result in problems, such as registration failures.

When deploying over OCI, install your OCSLB and OCSBC clusters in the same regions.

Out of Subnet GARP Parameter

- Although the **disable-garp-out-of-subnet** parameter is visible on the Subscriber Aware Load Balancer (SLB), it does not apply and has no effect on that product.

Cluster Membership

- Each SBC may be a member of only one cluster, and a cluster may be associated with only one SLB redundant pair.

Protocol Support

- The Oracle Communications Session Border Controller's FTP Server is deprecated. Only SFTP server services are supported.
- When handling TCP calls through load-balanced clusters, the SBC, in some scenarios, attempts to initiate the TCP handshake using the ephemeral port established for SIP services over the SBC-SLB tunnel instead of the end station's port. These calls fail because the SBC cannot utilize the tunnel properly. Example scenarios include:
 - After an HA SBC pair that is a member of a cluster fails over, the new active contains correct registrations, but not end station sockets. TCP calls to those end stations fail. When these end stations refresh their registrations, these calls can succeed.
 - TCP calls originating from the core to an SBC that is a member of a cluster, then the SLB, then toward end stations that are not registered at the SBC fail. The target end stations must register for these calls to succeed.

Fragmented Ping Support

- The Oracle Communications Session Border Controller does not respond to inbound fragmented ping packets.

Inactivity Age-out of Trusted Endpoints

- The inactivity age-out functionality for trusted endpoints is not yet implemented.

Physical Interface RTC Support

- After changing any Physical Interface configuration, a system reboot is required.

- Output from the packet trace local feature on hardware platforms running this software version may display invalid MAC addresses for signaling packets.

Command Line Interface Discrepancies

- The SLB's **tunnel-config** element includes a **tls-profile** parameter. This parameter is not functional.
- A platform running as a Subscriber-Aware Load Balancer (SLB) can still configure Session Border Controllers (SBC)-specific group-names.