ICOS 3.2 functional specification

ICOS 3.2 functional specification

Table of Contents

List of Figures

List of Tables

Chapter 1. About This Document

1.1. Purpose and Audience

This document provides a detailed functional description of ICOS software. It includes an overview of the software architecture and describes how networking protocols and management features are supported and implemented in the software. This information is intended for system administrators who are using ICOS software on their switch and router products.

1.2. Technical Support

Netberg provides customer access to a wide range of information, including software updates through its customer support support@netbergtw.com [mailto:support@netbergtw.com]

In addition, Netberg provides other product support through its Downloads and Support site http://netbergtw.com/top-support/

Chapter 2. ICOS overview

Netberg's industry-leading ICOS product solutions deliver production-ready networking software to manufacturers of Ethernet and IP systems. By leveraging the company's extensive experience base and worldwide partners, Netberg has assembled the three cornerstone attributes of a production-ready solution:

- Complete portfolio — ICOS is an extensive software suite of advanced networking features and protocols necessary to develop a variety of Ethernet and IP infrastructure systems for data center applications.
- Full integration — ICOS accelerates development activities by integrating software with industry-leading embedded operating systems, switching silicon, and CPUs.
- Certified solutions — Netberg conducts extensive quality assurance testing on production platforms to verify functional and system performance and to certify that the end product is ready for production shipment.

ICOS production-ready networking software provides a scalable, portable, and verified solutions set to accommodate the ever-increasing demands on network devices. ICOS software is an architecture suited for any network infrastructure device using leading-edge applications that require detailed packet inspection or dissection.

This section includes the following subsections:

- Section 2.1, “Architecture Overview”
- Section 2.2, “Control and Forwarding Planes”
- Section 2.3, “Layers”
- Section 2.4, “Open Network Install Environment Support”
- Section 2.5, “ICOS-as-a-Service on x86 Linux-based Systems”

2.1. Architecture Overview

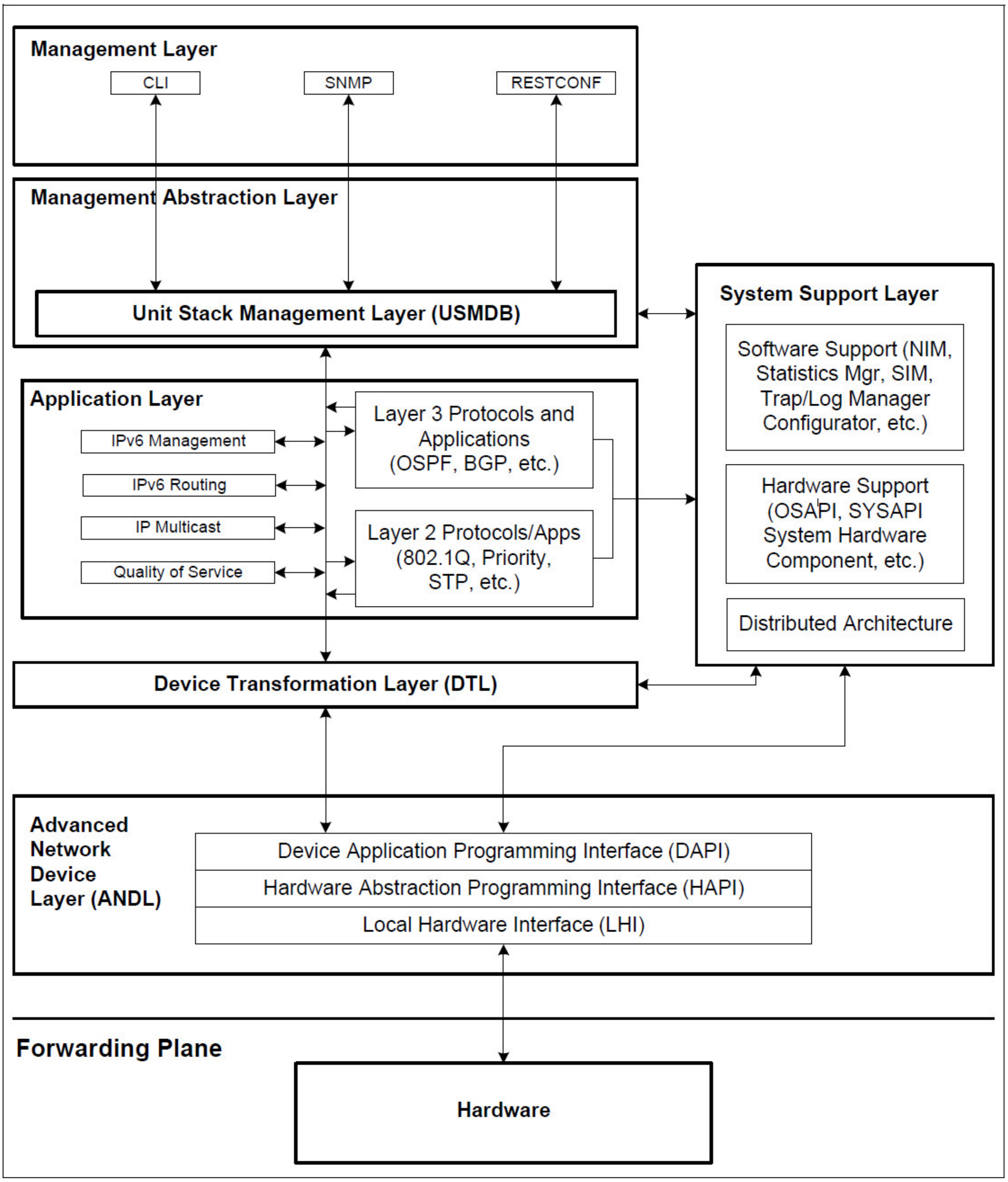
ICOS fills a growing demand for service-enabling functionality for today’s infrastructure market by allowing vendors to easily plug in advanced services that provide new revenue streams. Representing more than one million lines of productionquality code, ICOS 3.2 runs on Linux and x86 processors.

ICOS is designed to address the key functionality required in today’s networking devices. Support for the latest standards-based protocols ensures interoperability of an ICOS device with other systems within the network. By offering fully integrated device management through a command line interface and SNMP), the task of developing and deploying a network device is greatly minimized. The management abstraction layer includes the USMDB sublayer. The management interfaces use the management abstraction layer to configure the system.

ICOS software allows for rapid integration between network processing devices and other system components, thus reducing product development time and cost. The Application Programming Interfaces (APIs) are designed to be carefully abstracted without compromising performance. The layered architecture allows system designers to use varying combinations of modules within the ICOS software and to build customized applications to collaborate with the software at any level. Furthermore, the existing architectural components simplify the extension of the existing module components or for the addition of new functionality to the system.

The figure below illustrates the ICOS system layers and protocols. The ICOS system layers and protocols diagram is included in each of the sections to illustrate how each component fits within the overall architecture.

Figure 2.1. System Layers


Management Layer
CLI
SNMP
RESTCONF
Management Abstraction Layer
Unit Stack Management Layer (USMDB)
System Support Layer
Software Support (NIM, Statistics Mgr, SIM, Trap/Log Manager Configurator, etc.)
Hardware Support (OSAPI, SYSAPI System Hardware Component, etc.)
Distributed Architecture
Application Layer
IPv6 Management
IPv6 Routing
IP Multicast
Quality of Service
Layer 3 Protocols and Applications (OSPF, BGP, etc.)
Layer 2 Protocols/Apps (802.1Q, Priority, STP, etc.)
Device Transformation Layer (DTL)
Advanced Network Device Layer (ANDL)
Device Application Programming Interface (DAPI)
Hardware Abstraction Programming Interface (HAPI)
Local Hardware Interface (LHI)
Forwarding Plane
Hardware

2.1.1. ICOS Process Architecture

ICOS is implemented using multiple Linux processes. The process architecture improves fault isolation, allows certain parts of the system to be restarted in case of failure, and simplifies integration with third-party applications.

2.2. Control and Forwarding Planes

The figure above depicts a high-level view of the control and forwarding planes. The control plane disseminates device control commands and messages to the appropriate components. The forwarding plane services data packets and forwards them to the proper destination. The forwarding plane is not described because it is hardware-specific and outside the scope of this document.

2.3. Layers

This section describes the components within each layer.

2.3.1. Management Layer

The Management layer contains user interface components.

Table 2.1. Management Layer Components

Component	Description
Command Line Interface (CLI)	Textual interface where commands are typed at a line item prompt
Simple Network Management Protocol (SNMP)	Simple UDP-based network management protocol application accessed through a management information base (MIB) browser
RESTCONF	An HTTP-based network management protocol that allows user to monitor, read status, and configure a switch programmatically. It makes use of schema described by YANG models to describe the data exposed by the device.
User Manager	Data structure of user login information
Configurable Management VLAN	Feature used to manage a networking device
Management Security	Protocols that provide network security

See Chapter 3, *Management Layer* for details.

2.3.2. System Support Layer

The System Support layer provides critical systems management by abstracting operating systems, code management, and other utilities.

Table 2.2. System Support Layer Components

Component	Description
Software Support	
Infrastructure component: • Configurator • System Interface Manager (SIM) • Network Interface Manager (NIM)	• Initiates most applications within the ICOS software. • Stores all system-specific information such as the system IP address, system MAC address and its type, system and port configuration information, system name, location, and so on. The SIM is also responsible for defining the modes of operation of the system with respect to port monitoring, data transfer, flow control, broadcast storm recovery, and so on. • Facilitates all of the application components with device interface information, such as port number and card number addresses, and also provides interface parameters (frame size limits, and so on).

Component	Description
Service component: - Statistics Manager - Trap and Log Managers - Utilities	- Collects, evaluates, and presents statistical data essential to the functioning of the system. This is a software package. - Works in conjunction with the SIM to handle and log exceptions. - Provides services to the entire ICOS software.
Hardware Support	
Operating System API (OSAPI)	Provides a set of interfaces to OS support functions. System API (SYSAPI) Provides system-wide routines for network and buffer support and provides the interface into the system registry.
System Hardware Component	Provides an interface to all system hardware components except those components handled by the advanced network device layer, so that the underlying hardware details are hidden from the caller.

See Chapter 4, *System Support Layer* for details.

2.3.3. Unit Stack Manager [Database] (USMDB) Layer

The USMDB facilitates communication between the Management layer and all other layers. The USMDB layer consists of a large library of pass-through APIs that direct and translate commands to the proper modules. The USMDB layer enables threads within the ICOS process as well as external processes to communicate with the ICOS application layer.

See Chapter 5, *Management Abstraction Layer* for details.

2.3.4. Application Layer

The application layer contains protocol managing, data control, and tracking applications.

Table 2.3. Application Layer Components

Component	Description
Base/Security Features	Basic NOS services and security
Layer 2 — Switching Module	Implements layer 2 protocols such as 802.1Q and STP. The Switching module is the foundational package.
Data Center Module	Implements the data center components.
Layer 3 — Routing Module	Implements Layer 3 routing protocols such as BGP and OSPF.
IPv6 Routing	Implements the ability for the product to be an IPv6 Forum compliant IPv6 Router, including OSPFv3 for dynamic routing.
IPv6 Management Module	Implements the ability to fully manage the box through IPv6 management interfaces.

Component	Description
Quality of Service Module	Provides better services to selected network traffic. The Quality of Service module includes the Access Control List component and the Differentiated Services component.
IP Multicast Module	The Multicast component is best suited for video and audio traffic requiring multicast packet control for optimal operation.
BGP Module	Implements the BGP-4 Exterior Gateway routing protocol.

See Chapter 6, *Application Layer* for details.

2.3.5. Device Transformation Layer

The Device Transformation Layer (DTL) passes data and control information between the Application layer and the Advanced Network Device layer.

See Chapter 7, *Device Transformation Layer* for details.

2.3.6. Advanced Network Device Layer

The Advanced Network Device Layer (ANDL) controls and manages network processing devices.

Table 2.4. ANDL Components

Component	Description
Device Application Programming Interface (DAPI)	Passes data and control information between the Application layer and HAPI.
Hardware Abstraction Programming Interface (HAPI)	Contains the Network Processor-specific software that interacts with the hardware.
Local Hardware Interface (LHI)	Communicates directly with the Network Processing layer to receive, transmit, and process information.

See Chapter 8, *Advanced Network Device Layer* for details.

2.4. Open Network Install Environment Support

ICOS includes support for the Open Network Install Environment (ONIE), which allows customers to install their choice of network operating system (NOS) onto an ICOS platform. When the switch boots, ONIE enables the switch to fetch a NOS stored on a remote server. The remote server can hold multiple NOS images, and the administrator can specify which NOS to load and run on the switch. ONIE support in ICOS facilitates automated data center provisioning by enabling a bare-metal network switch ecosystem. ONIE is a small operating system. It is pre-installed as firmware and requires an ONIE-compliant boot loader (u-boot/BusyBox), kernel (Linux) and ONIE discovery and execution application provided by the ODM.

2.5. ICOS-as-a-Service on x86 Linux-based Systems

On x86 platforms running Linux, ICOS operates as a Linux application running as a service. This document refers to this feature as ICOS-as-a-Service. The main difference between ICOS-as-a-Service and ICOS as embedded system software is that ICOS-as-a-Service does not control the entire system. For example, when operating as a Linux application, an ICOS code update updates the ICOS application only — not the kernel or the root file system. The ICOS code package is used to control a Broadcom network switch chip that is connected to the system CPU through a PCIe interface. User authentication is generally done via Linux applications such as LDAP and switch provisioning is generally done via Linux Applications such as Puppet. ICOS-as-a-Service interacts with existing Linux services that typically run on Linux servers, such as NTP, Syslog, and NetSNMP.

2.5.1. Linux Integration of Base/Security Features

The following Base/Security package features are available for ICOS-as-a-service operation:

- The switch host name is obtained from Linux in addition to being configurable using the ICOS hostname command.
- Syslog events are sent to the Linux syslog in addition to the ICOS syslog.
- The text configuration files, such as startup-config and the configuration scripts are stored in uncompressed format, enabling the system administrator to view and edit these files.
- The ICOS SNTP feature and the clock management commands are removed. ICOS relies on the Linux clock management and the Linux NTP service to set the clock.
- The SDM template can be set via a text file. This makes it easier to configure the switch when using Zero-Touch Provisioning by writing a text file in the file system instead of starting ICOS and using the CLI commands to modify the SDM template.
- The Linux LLDP application has been tested with ICOS and verified to work as long as other ICOS features that depend on LLDP, such as DCBX and VoIP, are disabled.
- The kernel dump feature allows the system to perform a warm reboot into a new kernel in reserved memory, which enables capturing the current state of the operating kernel for postmortem analysis. This feature is implemented as a set of bash scripts in a DEB package that can be used with or without ICOS application running. The feature provides a convenient method to invoke the “crash” console kernel debugging utility without requiring complex configuration by the user. This provides the necessary handling to allow debugging of the ICOS customized Linux kernel. This feature is available only on Ubuntu Linux distributions.
- Multiple Consoles for CLI: When ICOS is started as a service, the switchdrvr listens on a Unix domain socket in the background. Once the user is authenticated and logged into the Linux system, the user can open a switchdrvr CLI session by invoking the **icos-cli** application (a *C*-based program). The **icos-cli** application can only be executed by sudo/root users. The root/sudo users always get maximum privilege level access while communicating with ICOS. The x86 platforms support maximum of four concurrent sessions through Unix domain sockets.

- The CLI **show** command is implemented as the **icos-show** Linux command on ICOS platforms. When ICOS is started as service, the switchdrvr listens on a feature-specific Unix domain socket in the background. After the user is authenticated and logged into the Linux system, the user can execute the **icos-show** application (a *C*-based program). The **icos-show** application can only be executed by sudo/root users. Any user other than root/sudo attempting to run ICOS using **icos-show** is denied access with an error message.

2.5.2. Linux Integration of Switching Features

The following Switching package features are available for ICOS-as-a-service operation:

- The physical port statistics and status can be monitored via the Linux ethtool utility. For example, the command *ethtool -S fpti1_0_1* reports the counters for port 0/1. • The physical ports can be managed to some extent via Linux commands. For example, the ports can be shut down via the command *ifconfig <intf> down*. The port speed and auto-negotiation status can also be changed.

- The LAGs can be managed to some extent via Linux utilities. The LAG membership can be changed and the LAG status/statistics can be viewed.

2.5.3. Linux Integration of Routing Features

The following Routing features are available for ICOS-as-a-service operation.

- The ICOS routing interfaces for the default router can be managed via Linux utilities to some extent. To enable the management via Linux, the routing interfaces reported via the *ifconfig* command have names that can be easily correlated to the ICOS port-based and VLAN-based routing interfaces.

- The IP addresses for the routing interfaces can be assigned using the Linux *ip* utility.

- The IP addresses for routing interfaces can be configured persistently using the */etc/network/interfaces* file.

- ICOS can push routes added by third-party applications into the hardware.

2.5.4. Net-SNMP Proxy

SNMP management on x86 platforms is normally done using Net-SNMP software. ICOS provides a Net-SNMP proxy capability that enables allows system managers to use Net-SNMP to manage the ICOS system. This feature enables proxy-forwarding of SNMPv3 requests for select MIBs to the SNMP engine built into ICOS. Traps and notifications generated by ICOS are handled by the Net-SNMP trap server (snmptrapd) and are proxy-forwarded to configured external trap receivers. The system administrator configures SNMP functionality on the Linux system using familiar means, with minimal configuration of ICOS required. The proxy-forwarding feature is supported for SNMPv3 using a context-to-community mapping. Additionally, limited proxy capability using SNMPv2c is supported.

The Net-SNMP Proxy feature acts as an “override” for the built-in ICOS SNMP operation. When Net-SNMP Proxy is enabled using the ICOS CLI, the existing ICOS SNMP configuration is ignored but preserved. The administrator may continue configuring such items as the SNMP server UDP port, SNMP communities, or an SNMPv3 user and view; however this configuration not applied during proxy operation. The configuration is retained in the running configuration so that when Net-SNMP Proxy is disabled, the settings can be reapplied.

Chapter 3. Management Layer

The Management layer consists of the components table below describes.

Table 3.1. Management Layer Components

Component	Description
CLI	Allows the end user to configure the network device and view device settings and statistics using a serial interface connected directly from a PC to the serial port of the network device. The CLI details are provided in the *CLI Command Reference*.
Telnet	Allows the end user to configure the network device and view device settings and statistics using a terminal emulator.
SNMP	Allows the user to configure the network device and view device settings and statistics using a Simple Network Management Protocol agent that supports SNMPv1, SNMPv2c, or SNMPv3. SNMP details are provided in the *ICOS user manual* Guide.
SSH	Allows the end user to configure the network device and view device settings and statistics using a secure shell.
RESTCONF	An HTTP-based network management protocol that allows user to monitor, read status, and configure a switch programmatically. It makes use of schema described by YANG models to describe the data exposed by the device.

To view a diagram of the Management layer within the ICOS architecture, see Figure 2.1, “System Layers”.

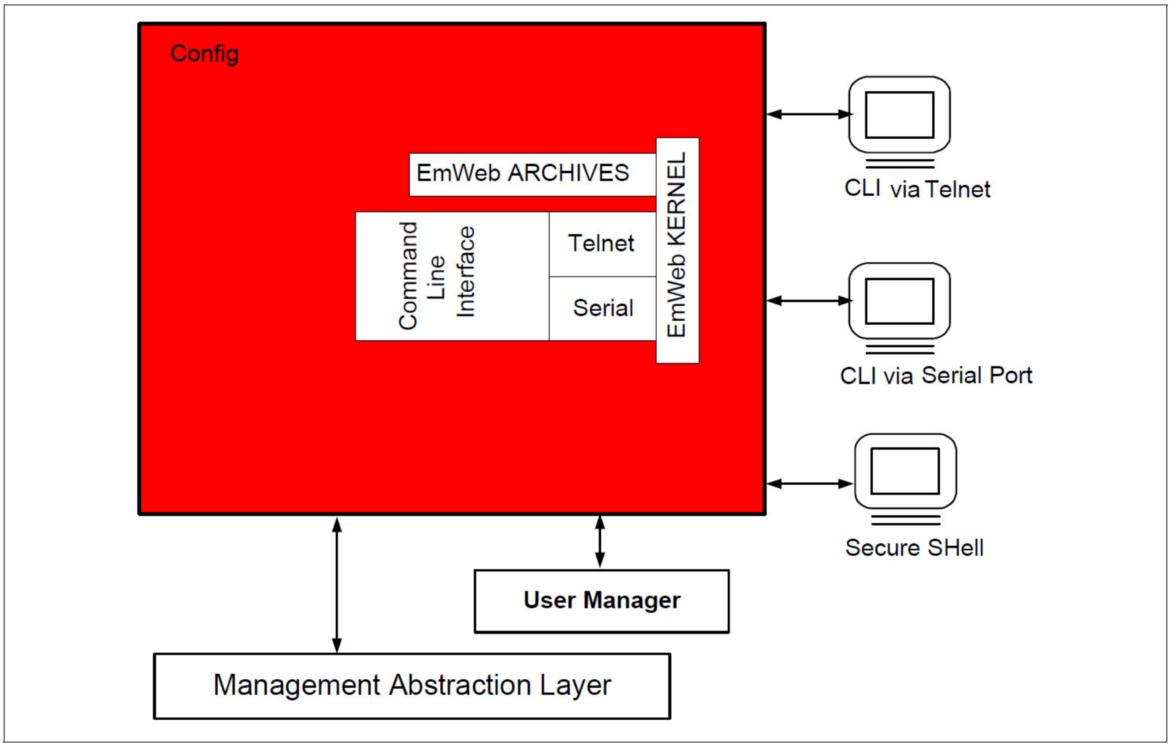
This section contains the following subsections:

- Section 3.1, “Config Architecture”
- Section 3.2, “Simple Network Management Protocol (SNMP)”
- Section 3.3, “RESTCONF”
- Section 3.4, “User Manager”
- Section 3.5, “Device Management Interfaces”
- Section 3.6, “Configurable Management VLAN”
- Section 3.7, “Management Security”

3.1. Config Architecture

The Config component provides the user interface support. It includes the CLI, which provides the ability to configure the network device, view settings and statistics, and upload or download code or configuration images. The Config component includes a command-line interface that is accessible through the device’s serial interface or by using a Telnet connection.

Figure 3.1. ICOS config architecture



The core Config architecture supports a CLI interface driven by the EmWeb kernel. Users can access the CLI using either a serial connection or a Telnet connection.

3.1.1. Text-based Configuration

The nonvolatile system configuration is stored in a CLI text format to simplify viewing the configuration. When the user saves the configuration, the running configuration is produced for the entire system. This output is saved to the file system in a text-based config file. When the system is restarted, it first configures the system defaults, and then the saved config file is run to restore the previous nonvolatile configuration. This config file may be copied on and off the system, and may be edited offline and reapplied either to the original system or to a different system. To save the size of the configuration file size and to increase the user experience, range commands are supported for VLAN creation/deletion, participation, and tagging in the CLI interface.

For compatibility with previous ICOS releases, until a text-based config file is created and exists in the file system, the previous *fastpath.cfg* binary format file is used to provide system configuration data. When a textbased file is created, it takes precedence.

3.1.2. Command-Line Interface Scripting

The configuration scripting feature allows the user to save the current ICOS configuration in text format.

To modify the configuration script file, follow these procedures:

1. Upload the file to a personal computer.

2. Edit the file.

3. Download the file to a networking device.

4. Apply it to the ICOS system.

With this feature in place, the ICOS user has the flexibility of creating configuration scripts and then applying the scripts to several devices.

3.1.3. Functional Description

The configuration scripting feature performs the following tasks:

- Saves the running configuration in a text formatted script file. The user provides the name for the configuration script.

- Allows saving as many scripts as possible within a given storage space limit. Multiple configuration scripts per networking device are possible. The user can create a configuration script or transfer a configuration script to the networking device using the copy command.

- Lists the available configuration scripts.

- Deletes one or all configuration scripts.

- Validates the configuration scripts before applying the script. The validation functions check for command syntactical errors but do not check command sequence errors.

- Applies the configuration script to the networking device as follows:

  - Prompts the user for confirmation.

  - Prompts to save the active configuration on the networking device.

  - Executes the command and displays success/failure messages on screen. If a command fails, the line number and command details are displayed. A warning is also issued.

- Uploads/downloads the configuration script to/from the networking device. Configuration scripts can be transferred using the **copy** command. To prevent file duplication, the download function prompts the user if the file name exists. When downloading a file, the file is validated and copied to the networking device. File execution does not occur at this time. If the file validation fails, the system indicates the line number and the failed command and prompts the user before saving the file.

3.1.4. Script File Format

The configuration script file contains one formatted CLI command per line and does not store the carriage return character. A configuration script file should use **exit** to change the CLI mode. The script file details are displayed with the **show running-config** command.

3.1.5. Configuration Scripts and Image Downgrades

x86 version of ICOS keep all settings in the text format. Simply copy the data.

3.1.6. Multiscreen Terminal Pagination

The CLI allows users to advance multiscreen output as a single line when return key is entered, and the output advances a whole page (terminal length) when space bar is pressed.

3.1.7. CLI Logged to Local File and System Log Server

The ICOS Command Logging component logs all command line interface commands issued on the system. The command log messages are stored with the other system logs and provide the system operators with a detailed log of the commands executed.

CLI command logging is configured through all the ICOS interfaces. When the feature is enabled, all CLI commands are logged using the existing logging subsystems. When this administrative mode is disabled, the CLI commands are not logged. By default, this mode is disabled.

The logging severity is set to *SEVERITY_NOTICE*. The logging severity is stored as a parameter of the configuration and only modifiable at compile time.

The Command Logging subsystem provides a CLI logging API through the usmDb layer. The use of this API, *usmDbCmdLoggerEntryAdd*, varies based upon the specific CLI implementation. For the ICOS CLI, the command logger is invoked from the CLI parser for most commands. There are exceptions where the logger must be invoked within the command itself due to the nature of the command. An example is the logout command. The log entry must be added before the user is logged out so the user name and connection can be determined.

The *usmDbCmdLoggerEntryAdd* API accepts one string argument. The CLI implementation determines the contents of the string. The ICOS implementation builds a string containing the user name, connection or IP address, and the command string. The ICOS uses a utility function, *cliWebCmdLoggerEntryAdd*, which builds the string to be passed to the command logger.

For example, the CLI log message for the user admin is:

```
  JAN 01 00:01:35 0.0.0.0-1 UNKN[54373024]: cmd_logger_api.c(93) 20 %
CLI:<connectionID>:<userID>:show vlan-assist-mac-learn all
```

If enabled, the CLI command logger subsystem begins to log commands immediately after the user is authenticated. After authentication, the CLI generates an explicit message and invokes the command logger.

The format of the message at login is:

```
  JAN 01 00:01:35 0.0.0.0-1 UNKN[54373024]: cmd_logger_api.c(93) 20 %
```

```
CLI:<connectionID>:<userID>:
User <userID> logged in
```

The CLI command log subsystem also logs all user log out instances. The format of the log message is:

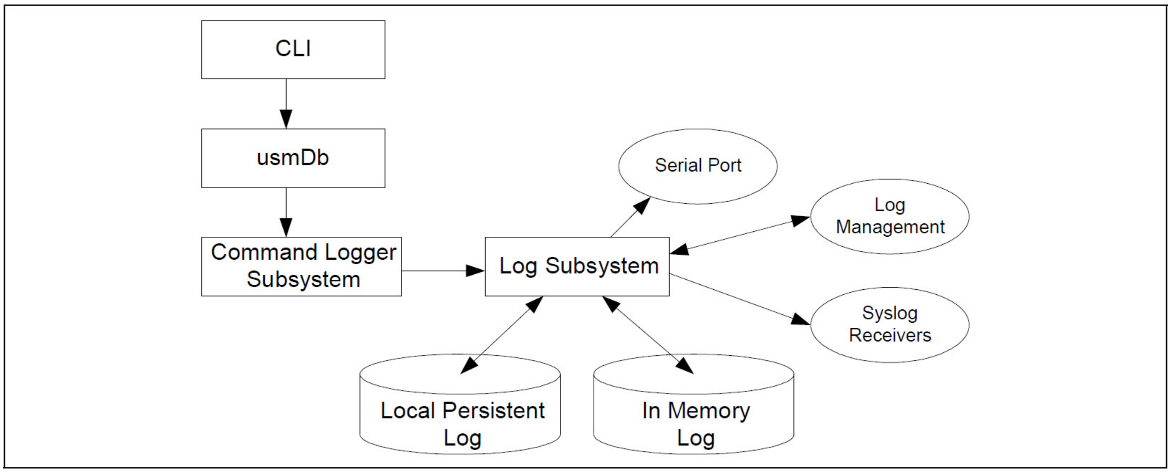
```
  JAN 01 00:01:35 0.0.0.0-1 UNKN[54373024]: cmd_logger_api.c(93) 20 %
CLI:<connectionID>:<userID>:logout
```

The CLI also logs messages to indicate disconnections caused by idle timeouts. The format of the message is as follows:

```
  JAN 01 00:01:35 0.0.0.0-1 UNKN[54373024]: cmd_logger_api.c(93) 20 %
CLI:<connectionID>:<userID>:User Disconnected due to Idle Timeout
```

The command logger messages are redirected to the logging subsystem, which logs the CLI command using any one of the existing logging subsystems. A generalized view of the CLI command logger subsystem is shown in the figure below.

Figure 3.2. Command Logger Subsystem



The existing log subsystem uses the passed parameters and makes calls to other system components to obtain the other required parameters such as:

- The timestamp
- The host IP address
- The file name
- The line number

The message is then sent to the persistent log function, the in-memory log function, the syslog function, and the serial port log function. Each of these functions examines the configuration and decides whether to log the message.

Log management is responsible for configuring the log subsystem and for retrieving and displaying the contents of various logs.

3.1.8. Command-Line Interface Login Banner

The ICOS software provides a pre-login banner utility that displays a user-configurable banner before the user prompt.

3.1.8.1. Functional Overview

The login banner is stored in an editable text file (*banner.txt*). When the system boots up, the content of the text file is displayed before the login prompt. If this file is not present, a default banner is shown.

3.1.8.2. Banner File Management

An administrator may upload or download the banner.txt file to nonvolatile storage using the user interface (CLI or SNMP). An administrator may also delete the file from nonvolatile storage.

The size of the *banner.txt* file must be no greater than *BANNER_MAX_FILE_SIZE*. Attempts to download a file that exceeds this limit fail. The default banner is used instead.

3.1.8.3. Banner Restrictions

The banner file is governed by the following restrictions:

- If the banner file is not present or the file size is greater than the MAX_FILE_SIZE (2K), the default banner is shown.
- If the row contains more than MAX_ROW_SIZE (80), the row is truncated.
- If the number of lines in the banner file is greater than MAX_ROW_COUNT (20), the default banner is shown.

3.1.9. Outbound Telnet

The Telnet protocol (outlined in RFC 854) allows users (clients) to connect to multiuser computers (servers) on the network. Telnet is often employed when a user communicates with a remote login service.

Telnet is the terminal emulation protocol in the TCP/IP suite. Telnet uses TCP as the transport protocol to initiate a connection between server and client. After connecting, the telnet server and client enter a period of option negotiation that determines the options each side is capable of supporting for the connection. The connected systems can negotiate new options or renegotiate old options at any time. In general, each end of the Telnet connection attempts to implement all options that maximize performance for the systems involved.

When a Telnet connection is initiated, each side of the connection is assumed to originate and terminate at a Network Virtual Terminal, or NVT. Therefore, the server and user hosts do not maintain information about the characteristics of each other’s terminals and terminal-handling conventions.

The idea of negotiated options recognizes that many hosts want to supply services in addition to those services available within an NVT. Also, many users have sophisticated terminals and would like to have elegant, rather than minimal, services.

Option requests may flurry back and forth when a Telnet connection is initiated, since each party tries to get the best possible service from the other party. Beyond that, however, options are used to dynamically modify the characteristics of the connection to suit varying local conditions.

3.1.10. Multiprocess Feature Support

The multiprocess support allows the ability to add new applications at run-time. This feature enables customers to load scripts and executable programs on the switch to allow certain applications to be implemented as processes. The multiprocess feature provides the infrastructure to support third party applications.

3.2. Simple Network Management Protocol (SNMP)

The SNMP component provides a machine-to-machine interface for the ICOS software product family. This includes the ability to configure the network device, view settings and statistics, and upload or download code or configuration images. The agent includes a get-bulk command to reduce network management traffic when retrieving a sequence of Management Information Base (MIB) variables and an elaborate set of error codes for improved reporting to the network control station.

The extensible and advanced design of the ICOS SNMP component makes adding remote manageability to networked devices undemanding. The agent allows a network control station to retrieve reports from the networked device. These reports are based upon the defined objects in the MIB. The agent queries, reports, and sets MIB variables based upon directions from the network control station or upon preset conditions.

3.2.1. Remote Monitoring (RMON) Architecture

The ICOS SNMP component includes an RMON (remote monitoring) agent. RMON is a base technology used by network management applications to manage a network. Troubleshooting and network planning can be accomplished through the network management applications.

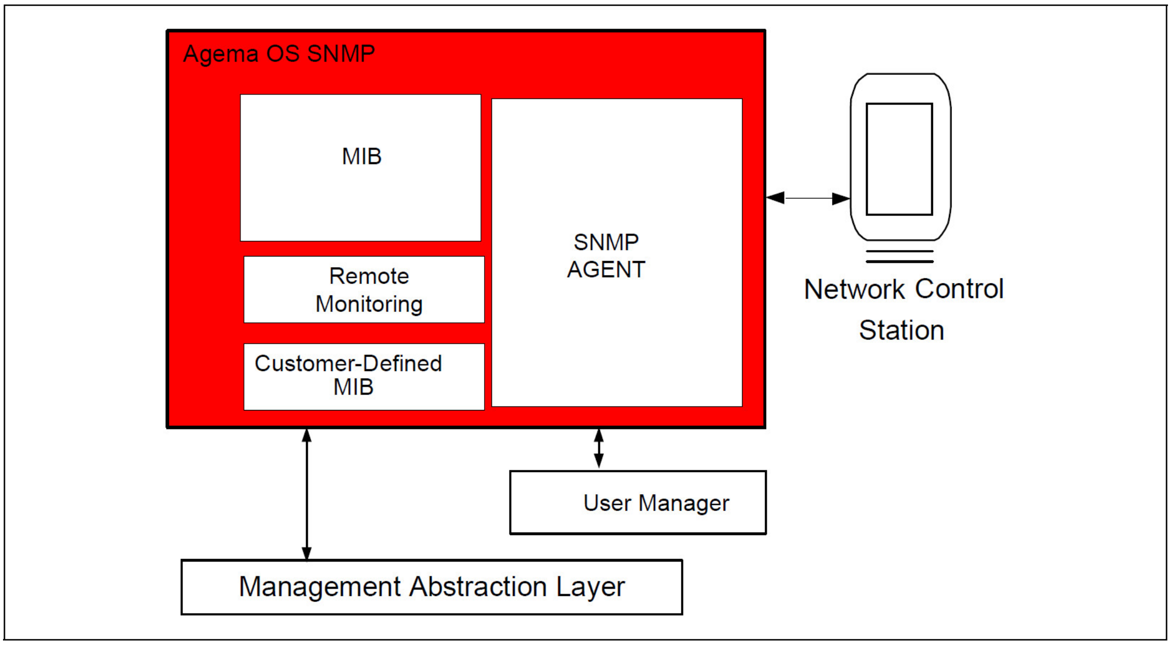
The network monitor monitors traffic on a network and records selected portions of the network traffic and statistics. The collected traffic and statistics are retrieved using SNMP and CLI. The data collected is defined in the RMON MIB, RFC 2819.

A device that supports gathering and reporting the RMON data is referred to as an RMON probe or RMON Agent. An RMON probe provides RMON data to an RMON Manager for analysis and presentation to the user. An RMON probe may be embedded in an existing network device or stand-alone.

The architecture of the SNMP with Remote Monitoring is shown in the figure below.

All communication between ICOS SNMP and the Application layer is handled through the Unit Stack Manager Database (USMDB) layer.

Figure 3.3. ICOS SNMP Architecture



3.2.2. Supported Versions

The SNMP agent supports the SNMPv1, SNMPv2c, or SNMPv3 protocols. Each protocol has a make file in the SNMP agent code base. The SNMP agent versions are back-portable. The software supports configuring SNMPv3 servers, users, and traps.

3.2.3. Private Management Information Base (MIB)

The SNMP agent supports adding any standard SMIv1 or SMIv2 MIB to the agent. This includes standard RFC MIBs as well as customer-defined MIBs. For the ICOS products, we include a private MIB containing configuration and statistics reporting objects for features supported by the ICOS products that are not covered by standard RFCs.

3.2.4. USMDB Interface Considerations

The USMDB layer is designed to provide a common interface to other components, and therefore returns information using L7 #defines. Because of the possibility that the indices used, and/or values returned, do not correspond to SNMP values, a conversion occurs between the two sets of values. For each MIB that requires a value or index translation, there is a file called *k_mib_<MIB name>_api.h*, which is included by the *k_mib_<mib name>.c* file. When GET or SET operations are performed, these functions are called instead of the USMDB functions to handle the conversion between the interval value and the correct SNMP value.

3.2.5. Handling Interfaces to System Code

The interfaces to system code are handled through Get functions. Get functions first perform the correct USMDB function call to obtain the information. During a successful call, the USMDB translates between the L7 #defines to the SNMP #defines located in the file *privatedefs.h*. If the translation is successful, the SNMP value is returned.

A *Set* function is similar to the *Get* functions, with the difference being the order of the translation. In this call, the value passed to the function must first be translated from the SNMP #defines to the L7 #defines before being passed to the USMDB Set function.

The third function type handles custom indexing for components that do not operate on an index. In this case, *<mib_object>EntryGet* and GetNext functions are created to handle correctly checking for the validity of index values and searching for the next valid index during a GET NEXT request. Since the components do not handle indexing, Get and Set functions must also be written to handle obtaining and setting the object values.

3.3. RESTCONF

RESTCONF is an HTTP-based network management protocol that allows the user to monitor, read status, and configure a switch programmatically. It makes use of schema described by YANG models to describe the data exposed by the device. It allows web-based applications to configure a switch, create a back-up of its running configuration, and replicate its configuration to other switches.

Monitoring and notification features are not implemented. These features may be added in future releases.

3.3.1. Functional Description

RESTCONF uses HTTP operations to provide Create, Retrieve, Update, Delete (CRUD) operations on a NETCONF datastore containing YANG defined data. The YANG language defines the syntax and semantics of datastore content, operational data, protocol operations, and notification events.

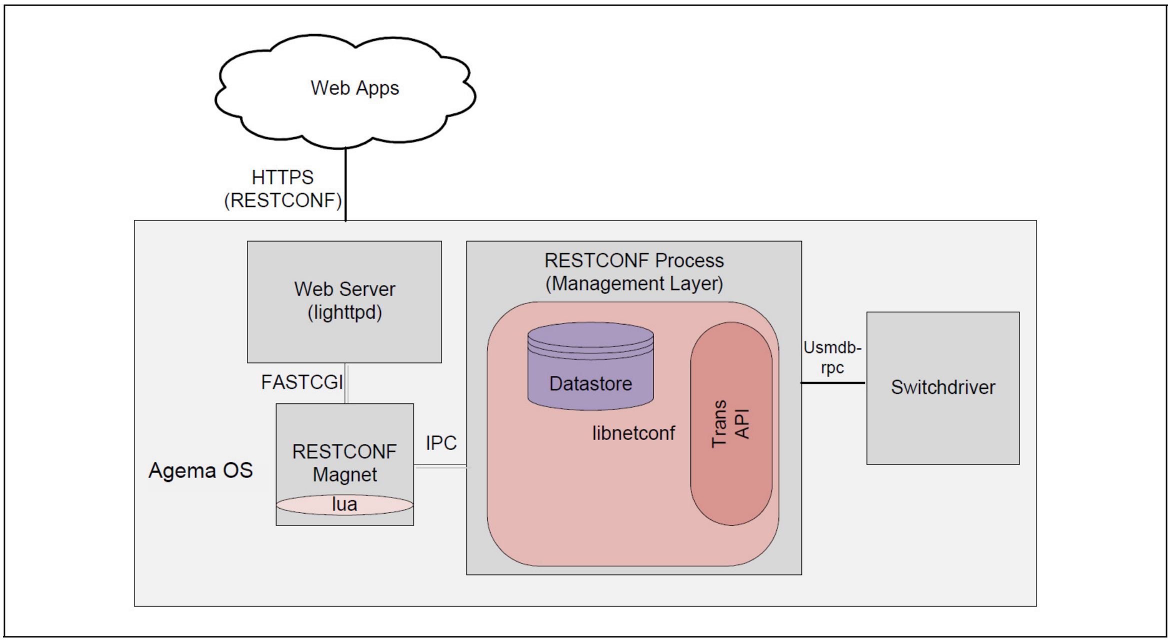
Configuration data and state data are exposed as resources that can be retrieved with the GET method. Resources representing configuration data can be modified with the DELETE, PATCH, POST, and PUT methods. Data is encoded with XML (JSON not supported).

RESTCONF architecture uses ICOS webserver (lighttpd) to handle incoming HTTPS (RESTCONF) requests. Web server forwards any request with root url path "/restconf" and "/yangmodules" to RESTCONF magnet. The web server and associated magnet infrastructure takes care of handling TLS (HTTPS) transport infrastructure and client authentication.

3.3.2. Architecture

The RESTCONF process is an ICOS multiprocess management application that adheres to the guidelines of ICOS multiprocess architecture. It makes use of the open source “libnetconf” library to support the datastore concept. The same library also provides infrastructure to auto-generate a “C” stub function to fetch configuration and state information and to apply the configuration on the switchdriver. The code inside the generated stub code makes usmDb RPC calls to achieve its purpose. The figure below shows the RESTCONF architecture.

Figure 3.4. ICOS RESTCONF Interface Architecture



3.4. User Manager

The User Manager component maintains the data structure containing user login information. The User Manager component authenticates locally configured users and users who are configured on a Remote Authentication Server (RAS).

3.4.1. Authorization

Authorization function determines if user is authorized to perform a given activity. ICOS allows per-command authorization using TACACS+ and RADIUS.

ICOS provides support for exec authorization and command authorization. The valid methods for each type of authorization are shown in the following table:

Method	Commands	Exec
local	No	Yes
none	Yes	Yes
radius	yes	Yes
tacacs	Yes	Yes

3.4.2. RADIUS Change of Authorization

The RADIUS Dynamic Authorization feature implements part of the Dynamic Authorization Server (DAS) functionality defined in RFC 5176 (Dynamic Authorization Extensions to Remote Authentication Dial In User Services). This feature enables a RADIUS server or any other external server to send messages to a Network Access Server (NAS) to terminate a user's session. This is desirable when a device or user session is causing problems in normal network operation.

RFC 5176 defines the DAS and Dynamic Authorization Client (DAC) and the following types of messages:

- Disconnect messages—This message from the DAC may result in terminating a user's session.
- Change of Authorization messages—This message from a DAC results in changing authorization status of the session.

As of release the current release, ICOS the DAS implementation that handles the Disconnect message only.

When ICOS DAS receives Disconnect Message from DAC, it looks for NAS identification and User Identity attributes available in the Disconnect Message. If the match for the NAS attribute and user's identify is found then it disconnect matching sessions and when successful, sends an ACK to DAC. The DAS sends a NAK with "Acct-Terminate-Cause" attribute (49) with value set to 6 if the user's session is not available or one or more sessions could not be disconnected by DAS.

3.4.3. Command Authorization

The RADIUS protocol does not support command authorization. ICOS uses the following mechanism to achieve command authorization with RADIUS:

ICOS uses RADIUS vendor specific Attribute-Value pair for downloading list of permitted/denied commands from RADIUS server. RADIUS server should be configured to return vendor specific Attribute-Value pair. ICOS downloads the list of commands permitted/denied from the RADIUS server while authenticating the user and cache list of commands per user with it. User executed command(s) are validated against the downloaded command lists for the authenticated user. If the command is permitted, then the return value is PASS. If the command is not allowed, then the return value is FAIL. If the user does not have any permit/deny command list, the default value is to permit all commands. Any changes in the user command authorization access list take effect upon user logoff and login. The user can specify multiple vendor-specific attribute-value pair in RADIUS server.

ICOS supports two permission modes for command authorization:

- Permit: The commands that are listed in the permit commands list are allowed to execute by the user.
- Deny: The commands that are listed in the deny are not allowed by the user.

In ICOS software, the default permission for all the commands is permit.

ICOS allows configuring both a TACACS+ server and RADIUS server. If the first method of command authorization returns an error, then the second method is used for command authorization. ICOS supports either permit or deny in the RADIUS vendor-specific attribute-value pair for a user. If both permit and deny are configured on the RADIUS server, then ICOS allows or denies the commands based on the value present in the AV-pair.

3.4.4. Accounting

The administrator may choose to monitor user activity on the switch. The following user activities can be tracked:

- **User exec sessions**: User login and logout times are noted and conveyed to an external AAA server.
- **User executed commands**: Commands executed by the user and the time of execution are accounted and conveyed to an external AAA server.
- **Dot1x**: Sends accounting records for network access.

The valid methods for each type of accounting are as follows:

Method	**Commands**	**Dot1x**	**Exec**
radius	No	yes	yes
tacacs	Yes	no	yes

User activity can be monitored at the end and/or at the beginning of the activity. For this purpose, the following record-types are available:

- Start-stop: Accounting notifications are sent at the beginning and at the end of an exec session or a userexecuted command. User activity does not wait for the accounting notification to be recorded at the AAA server.

- Stop-only: Accounting notifications are sent at the end of an exec session or a user-executed command.

3.4.5. Accounting Method Lists

An Accounting Method List is an ordered list of Accounting Methods that can be applied to the Accounting Types (Exec/Commands). Accounting Method Lists are identified by the ‘default’ keyword or by a user defined name. The supported accounting methods are TACACS+ and RADIUS. TACACS+ accounts both Command and Exec sessions, and RADIUS accounts user exec sessions and 802.1X sessions.

Accounting Method Lists can be applied to the following access line modes for accounting purposes:

- Console: This mode is used when a user logs in to the switch using a serial console.
- Telnet: This mode is used when a user logs in through Telnet.
- SSH: This mode is used when a user logs in through SSH.

3.4.6. Authentication

Users can be authenticated based on:

- Login mode (login)
- Domain name
- Switch access method (dot1x)
- Access to Privileged EXEC mode (Enable)
- Two levels of access
  - 1 = Read-only
  - 15 = Read-write

A future implementation will provide up to 15 levels of access.

The valid methods for each type of authentication are as follows:

Method	**Login**	**Enable**	**dot1x**
enable	yes	yes	no
ias	no	no	yes
line	yes	yes	no
local	yes	no	yes
none	yes	yes	Yes

Method	Login	Enable	dot1x
radius	yes	yes	yes
tacacs	yes	yes	no
deny	no	yes	no

3.4.7. Domain Authentication

ICOS supports authentication based on domain name in addition to the user name and password.

ICOS allows the switch to be configured in a domain. The user can enable or disable domain functionality:

- Domain enabled: In this case, when only the user name is entered by user, then ICOS sends the user name as domain-name (configured on switch)\username to the RADIUS server. If the user enters the domain name and user name, then ICOS sends the user name input as domain-name (entered by user)\username to the RADIUS server.
- Domain disabled: In this case, the domain name is not included when the user name is sent to the RADIUS server.

If a user domain is already provided by the user/supplicant, ICOS assumes that domain name reaches the ICOS application along with the user name in the format Domainname\username.

3.4.8. Authentication Preference Lists

To authenticate a user, the authentication methods in the APL for the access line are attempted in order until an authentication attempt returns a success or failure. If a method times out or returns an error, the next method in the list is attempted. A method may return an error if the credentials are unavailable with authentication source. The component requesting authentication is unaware of the ultimate authentication source. The following table summarizes the various methods:

Method	Username?	Password?	Can Error?
enable	No	Yes	yes
ias	Yes	Yes	no
line	No	Yes	yes
local	Yes	Yes	yes
none	No	No	no
radius	Yes	Yes	yes
tacacs	Yes	Yes	yes

Once an APL is created, a reference to that APL can be stored in the access line configuration to determine how specific components should authenticate users. The APL and associated component ID are stored together. A single APL can be referenced by multiple users and components.

The following APLs can be defined:

- Login APLs - this APL is used when a user logs into a switch using Console, Telnet or SSH. You can define multiple APLs. However, only one APL is active for each access line.
- Enable APLs - this APL is used when a user tries to enter Privilege Exec mode using Console, Telnet or SSH. You can define multiple APLs. However, only one APL is active for each access line.
- A single APL defined for Dot1X

The following default APLs are provided:

- DefaultList = Default Login APL for Console Access.
- NetworkList = Default Login APL for Telnet and SSH Accesses.
- EnableList = Default Enable APL for Console.
- networkEnableList = Default enable APL for telnet and SSH.

The methods defined for the default APLs are configurable.

3.4.9. Access Lines

The access line modes are as follows:

- Console: This mode is used when user logs in to the switch through Serial Console.
- Telnet: This mode is used when user logs in through Telnet.
- SSH: This mode is used when user logs in through SSH.
- Dot1X: This mode is used when a port is in Dot1X mode.

3.5. Device Management Interfaces

An ICOS device can be managed through a command-line interface or SNMP. The following interfaces may be used to access the management application, depending on the hardware platform and software build options:

- A service port. This port is available on some hardware platforms. This requires a dedicated Ethernet connection to a management station.
- A network port. This interface may be included as a software build option. The network port is a logical interface that uses one of the front-panel Ethernet ports. The network port can be assigned an IP address, statically or through DHCP. The address assigned to the Network port is always associated with a management VLAN.

On builds that do not support the network port option:

- The management VLAN is defaults to 0.
- A host interface is created on VLAN 1 by default.

On builds that support the network port option:

- The management VLAN defaults to 1.
  - Any IP interface. Any IP interface can act as a host interface. A host interface is configured in the same way as a routing interface: it is assigned one or more IP addresses, and the interface is configured as an IP—not a switching—interface. The IP address can be assigned manually by the administrator or leased dynamically through a DHCP server.

When routing is disabled globally, an IP interface acts as a host interface. Packets received on the interface cannot be forwarded on to other interfaces.

When routing is enabled globally, an IP interface acts as a routing interface. Traffic received on a routing interface can be forwarded on to other interfaces.

3.6. Configurable Management VLAN

A management VLAN is optionally used to manage a networking device. The device management is accomplished by means of the IP address associated with the network device interface. The network interface is the virtual interface through which the switch is manageable. The term manageable applies to IP management of the networking device by means of CLI (Telnet) or SNMP.

A particular networking device is allowed one management VLAN. The value of the management VLAN defaults to 1. The ICOS software provides a feature to configure the management VLAN to any valid value. The configurable Management VLAN feature applies only to the network interface (that is, the virtual interface associated with the network address). The configurable Management VLAN does not apply to the service port or any routing interfaces, including physical and VLAN interfaces. When a VLAN is configured as a host or routing interface, the networking device is also manageable through that VLAN using the IP address associated with the VLAN interface.

3.7. Management Security

ICOS software includes secure shell (SSH) and secure sockets layer (SSL) to help ensure the security of network transactions.

3.7.1. Secure Shell (SSH)

The ICOS software SSH feature is detailed in Table below.

Table 3.2. Secure Shell Feature Details

SSH Feature	**Component Type**	
Connection Type	Interactive Login	
Authentication Method	Password	
Ciphers	SSH Version 1 • DES • 3DES • Blowfish	SSH Version 2 • 3DES-CBC • AES128-CBC, AES192-CBC, AES256-CBC • AES128-CTR, AES192-CTR, AES256-CTR • ARCFOUR, ARCFOUR128, ARCFOUR256 • CAST128-CBC • Blowfish-CBC
Hash Algorithms	SSH Version 1 • MD5 • CRC-32	SSH Version 2 • SHA-1 • RIPEMD-160 • MD5
Key Exchange Methods	Diffie-Hellman	
Compression Algorithms	zlib None (no compression)	
Public Key Algorithms	SSH Version 1 • RSA	SSH Version 2 • DSA • DH
SSH Protocol	Versions SSH 2.0 SSH 1.5	

Keys and certificates can be generated externally (that is, offline) and downloaded to the target or generated directly by the ICOS software.

3.7.2. Secure Sockets Layer

SSL provides a means of abstracting an encrypted connection between two stations. Once established, such a connection is virtually no different to use than an unsecured connection. This allows an established protocol to operate in a secure manner on an open network.

The ICOS software SSL feature is detailed in Table below.

Table 3.3. Secure Sockets Layer Details

SSL Feature	**Component Type**
Ciphers	RC4 DES 3DES
Hash algorithms	MD5 SHA-1
Key Exchange methods	Diffie-Hellman RSA
SSL protocol versions	TLS 1.0 SSL 3.0

Keys and certificates can be generated externally (that is, offline) and downloaded to the target or generated directly by the ICOS software.

3.7.3. Password Management

The password management component includes the following features:

- Configurable Minimum Password Length

The administrator has the option of requiring user passwords to be a minimum length. The administrator can choose to have the switch enforce a minimum length between 8 and 64 characters. The default minimum length is 8 although there is no default password (zero length string).

- Password History

Keeping a history of previous passwords ensures that users cannot reuse passwords often. The administrator can configure the switch to store up to 10 of the last passwords for each user. The default operation is that no history is stored.

- Password Aging

The switch can implement an aging process on passwords and require users to change them when they expire. The administrator can configure the switch to force a password change between 1 and 365 days. By default, password aging is disabled. When a password expires, the user must enter a new password before continuing.

- User Lockout

The administrator may choose to strengthen the security of the switch by enabling the user lockout feature. A lockout count between 1 and 5 attempts can be configured. When a lockout count is configured, then a user that is logging in must enter the correct password within that count. Otherwise, that user is locked out form further remote switch access. Only an administrator with read/write access can reactivate that user. The user lockout feature is disabled by default. The user lockout feature only applies to remote users. That is, a session on the serial port cannot result in a user being locked out. This ensures that if a hacker tries to log in as *admin* and causes the account to be locked out, then the administrator with physical access to the switch can still log in and reactivate the *admin* account.

3.7.4. Strong Passwords

The software supports configuration of the characteristics of a strong password. Password strength is a measure of the effectiveness of a password in resisting guessing and brute-force attacks. The strength of a password is a function of length, complexity and randomness. Using strong passwords lowers overall risk of a security breach. This feature can be used to enforce a baseline password strength for all locally administered users.

The following parameters can be configured to determine the characteristics of a strong password:

- The minimum number of uppercase letters.
- The minimum number of lowercase letters.
- The minimum number of numeric characters.
- The minimum number of special characters from the set (`! $ % ^ & * ( ) _ - + = { [}] : ; @ ' ~ # | \ < , > . /).
- Whether the password can contain the associated login name.
- The maximum number of consecutive characters (such as “abcd” or “1234”).
- The maximum number of repeated characters or numbers (such as “1111” or “aaaa”).
- The minimum number of character classes (upper case, lower case, number, or special characters).
- Whether the password contains particular strings that have been configured to be excluded.

Chapter 4. System Support Layer

This section describes the architectural design and implementation of System Support and is organized as follows:

- Section 4.2, “Application Functionality” outlines the architecture of the System Support applications.
- Section 4.3, “Software Support Components” provides an overview of the functionality, components, and collaborators of the System Support layer.

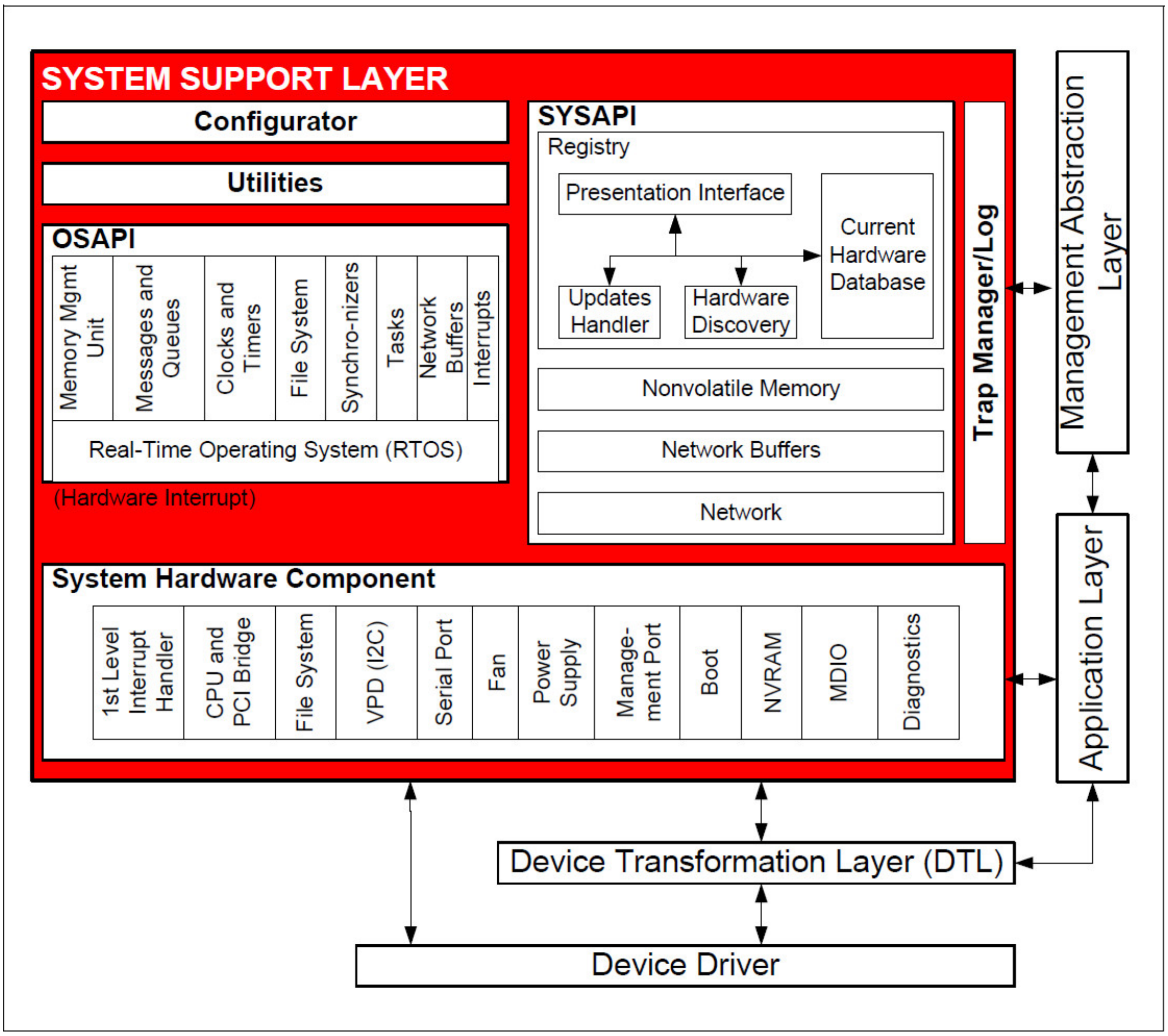
To view the Systems Support component in relation to the ICOS architecture, see Figure 2.1, “System Layers”. System Support is a collection of components that provide functionality to the overall ICOS software by abstracting operating systems, hardware, code management, and utilities.

4.1. System Support Architecture

The architecture of System Support is displayed in separate diagrams in the hardware and software support component sections that follow. System Support interacts with the following components and layers:

- Unit Stack Manager Database (USMDB)
- Application Layer
- Device Transformation Layer (DTL)
- Advanced Network Device Layer

Figure 4.1. System Support Architecture



4.2. Application Functionality

The components of the System Support layer are actually independent applications that have been conceptually grouped together as system-supporting applications. The following sections outline the functionality and components. The first section outlines the Software Support components, and the next section outlines the components.

4.2.1. Application Characteristics

The primary characteristics of the System Support layer are:

- Flexible, independent modules: Each module operates independently of the others so that any changes made to one do not affect any other. This allows a set of individual modules that are easy to extend, change, and reshape.
- Easy to customize: System Support functions with any combination of modules depending on the requirements.
- Easy to maintain both architecturally and within the code.
- OSAPI and SYSAPI provide flexible OS and system independence to the entire software system.

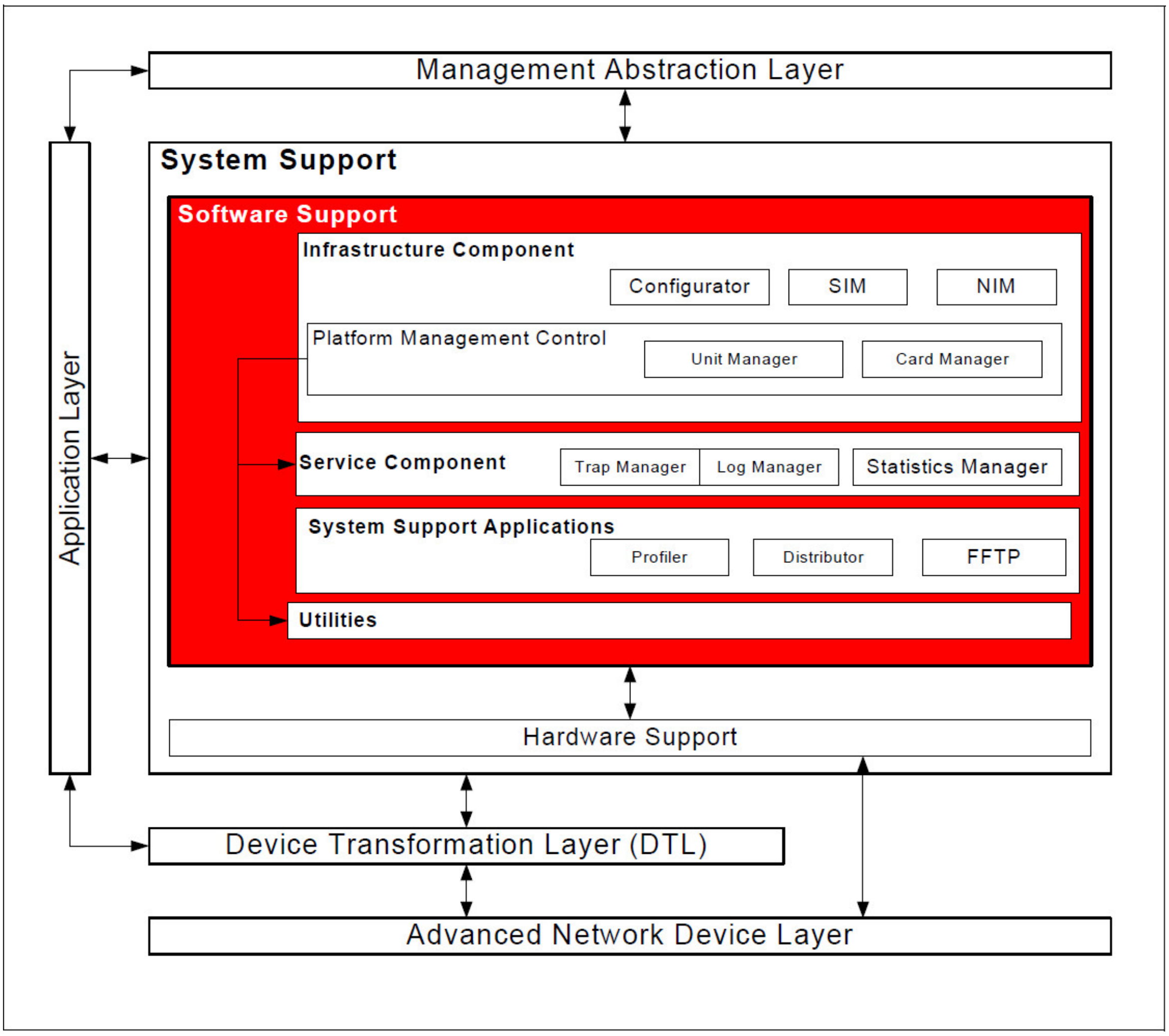
4.3. Software Support Components

The figure below illustrates the ICOS system in relation to the Software Support Components, which includes the following components:

- Infrastructure Component
- Service Component
- Utilities

These components are outlined in the following sections.

Figure 4.2. System Support—Software Support Components



4.3.1. Infrastructure Component

The Infrastructure component controls and configures the operation of other applications, as well as storing general information critical to the operation of key components with the ICOS software.

Table 4.1. Configurator

Functionality	The Configurator initializes all ICOS components. The system initialization is performed in multiple phases. During each phase the Configurator invokes the initialization function for each component. The component initialization functions invoked by the configurator are listed in the global array *cnfgrComponentList[] in file cnfgr_sid.c*.
Components	None
Interacts with	Most modules within the ICOS software.

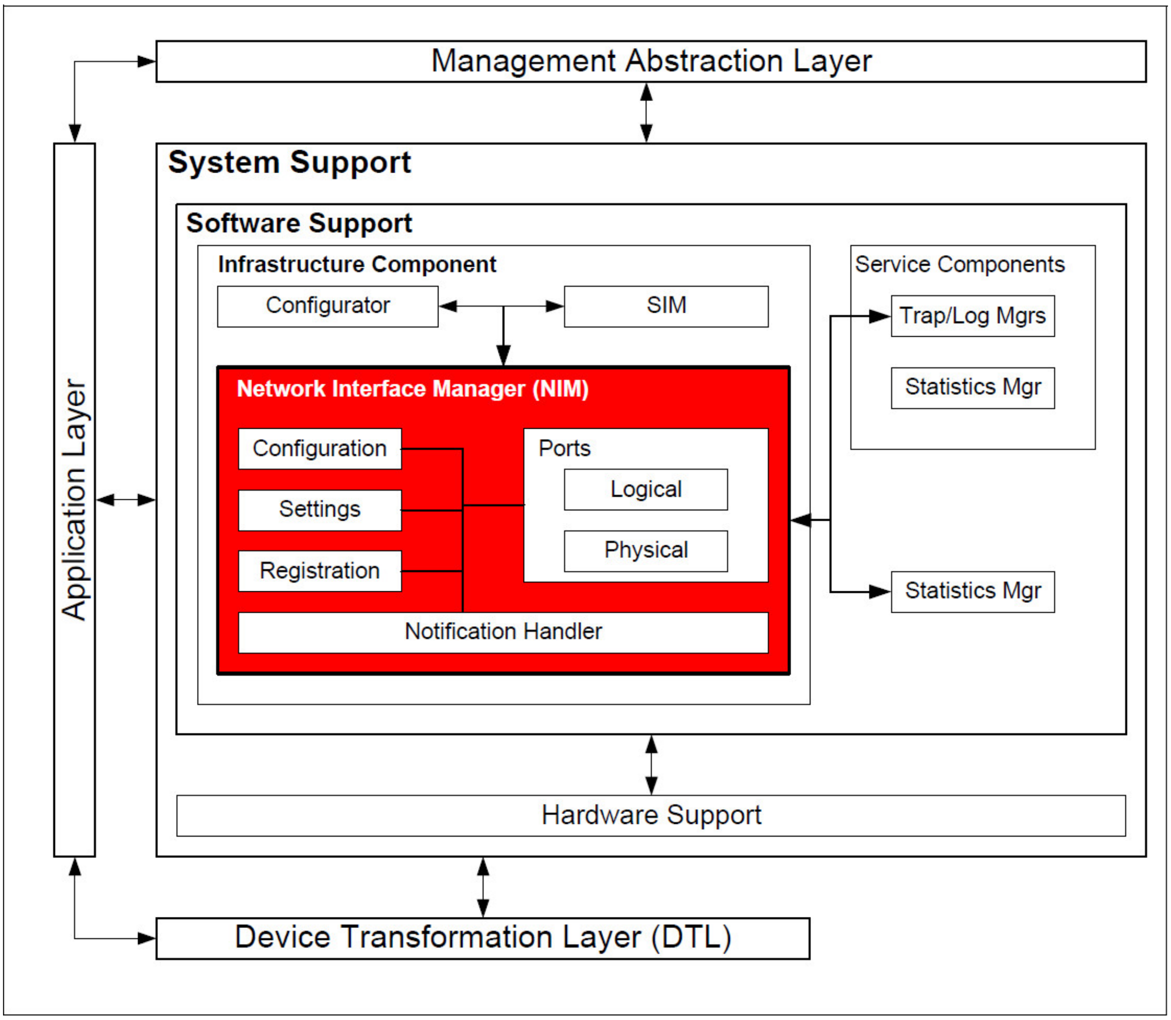
Table 4.2. System Interface Manager (SIM)

Functionality	The SIM stores all system-specific information such as the system IP address, system MAC address and its type, system and port configuration information, system name, location, and so on. The SIM is also responsible for defining the modes of operation of the system with respect to port monitoring, data transfer, flow control, broadcast storm recovery, and so on.
Components	None
Interacts with	Most modules within the ICOS software.

Table 4.3. Network Interface Manager (NIM)

Functionality	The NIM provides for configuration and management of network interfaces. The NIM also provides for management of interface state changes. The figure below displays the NIM architecture.
Components	The major components of the NIM are: • Configuration: Configures new ports and/or restores the configuration for existing ports. • Ports: Allocates sufficient memory for interfaces to be created. This component assigns an internal interface number that is a logical assignment or address for a port. The NIM creates interface counters that track incoming and outgoing port activity, which are in turn used by the statistics manager. The types of ports are: • Physical: Manages ports that exist physically on a network device. • Logical: Manages logically interconnected physical ports. • Settings: Allows runtime configuration of port parameters. This component allows the software to Set or Get port parameters during runtime. • Registration: Tracks API information for callback purposes when event notification is needed. • Notification Handler: Notifies all registered subsystems with any changes in port states settings or configurations. The events that are handled in this component are shown in the *L7_PORT_EVENTS_t* structure.
Interacts with	The NIM interacts with most components of the Application and System Support Layers.

Figure 4.3. NIM Architecture



4.3.2. Service Components

The Service component provides information on the state of the overall ICOS software, such as statistics, trace manager, dumping, log manager, and others.

Table 4.4. Statistics Manager

Functionality	The Statistics Manager is a software package designed to collect, evaluate, and present statistical data essential to the functioning of the software. The Statistics Manager creates metrics based on counters throughout the software and is responsible for keeping the client or administration informed of the nature of the data flow associated with the software. This package was created with generic components to permit its reuse irrespective of the type or position of the system. Located in the application layer, the Statistics Manager draws required statistical information from components such as the NIM at the request of the

	client. The Statistics Manager maintains a pool of current counters that may be updated or reset depending upon the client's requirements. The components of the Statistics Manager, outlined in the following sections, communicate with each other internally to provide accurate data without making the client aware of the internal manipulations involved. For performance reasons, counters are created dynamically as the client requires them, even though the corresponding counter may be calculated at lower layers. The Statistics Manager begins keeping track of a particular counter when a client requests the counter. Therefore, the software design assumes that the client provides the Statistics Manager with the required information about individual counters that need to be maintained.

Table 4.5. Utilities

Functionality	A set of common programming utilities is provided for developers.
Components	• Adelson-Velskii and Landis (AVL) tree manager: An information and storage management application. • Double Linked List (DLL): An abstract double-linked list data type. • Buffer pool manager: A memory management utility used to create private fixed size buffer pools. • Message Digest (MD5): A utility that takes a message (arbitrarily sized input data) and generates a fixed-size output. • Password Scrambler: A utility used to rearrange the bits in a password. • Type-Length-Value (TLV) Utility: A utility for parsing (reading) existing TLVs. The utility relies on a caller-supplied external function for interpreting the TLV entry definition and its contents. TLV type codes from 0x0000 to 0x03FF are reserved for functional categories having meaning throughout an ICOS system. • type: Command-dependent, identifies the content of the value field. • Length: Specifies the number of bytes in the value field, which must be a multiple of four. • Value: Parameter data, format depends on the type. The value field may consist of one or more additional type-length-value constructs. • Random Number Generator (RNG): A utility used to output numbers having no specific order. • zlib: A compression utility. • Keying for Advanced Features: ICOS Keying for Advanced Features blocks access to unlicensed features. Blocking prohibits the configuration and management of features the user has not licensed. This allows a software platform to contain components that a user is prevented from accessing without a valid license key.

4.3.3. Hardware Support Component

The figure below illustrates the Hardware Support Component in relation to the ICOS system, and the following sections detail the component’s functionality. The Hardware Support Components are:

- OSAPI
- SYSAPI
- System Hardware Component

Figure 4.4. System Support—Hardware Support Components

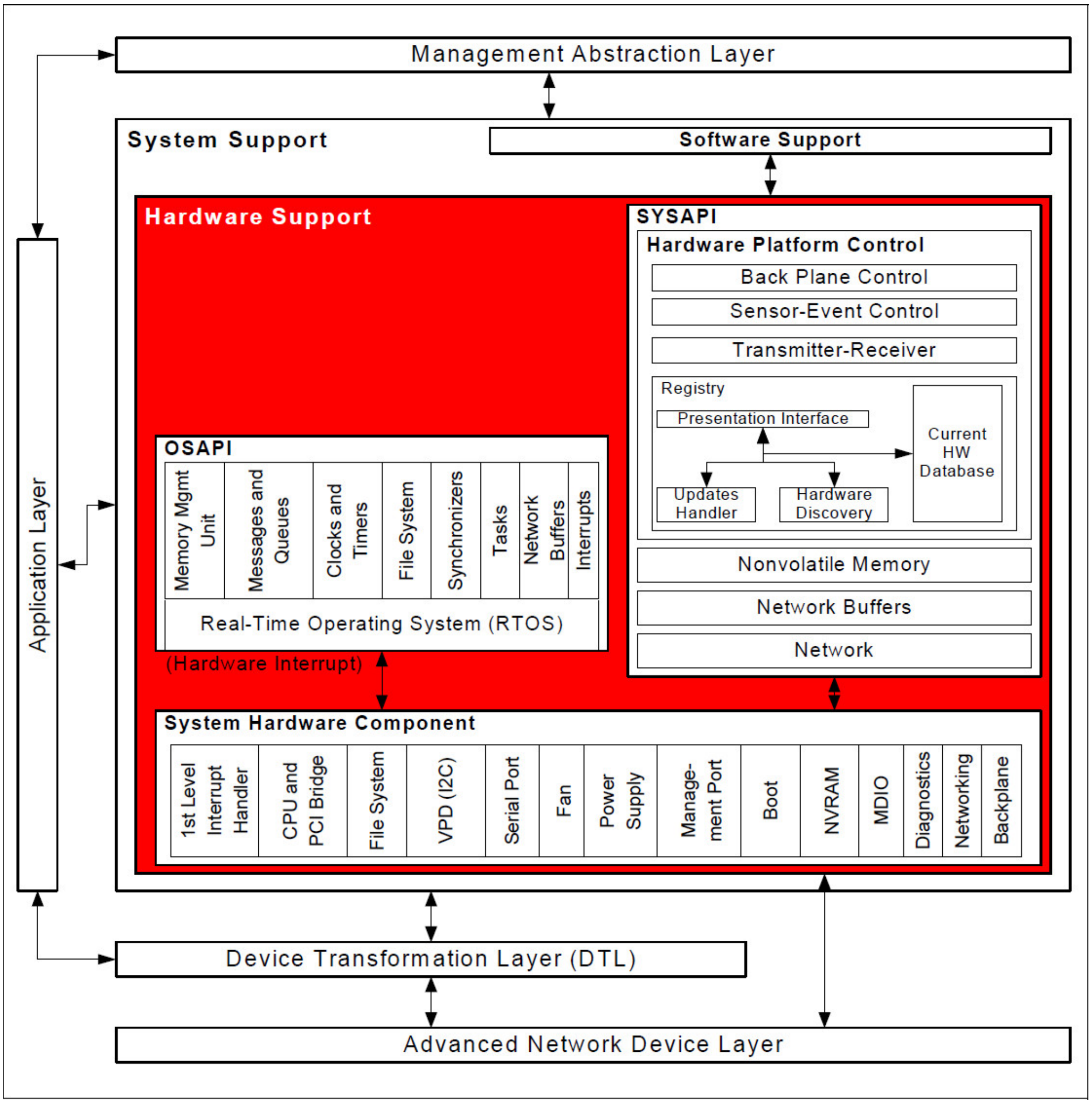


Table 4.6. Operating System Application Programming Interface (OSAPI)

Functionality	The OSAPI component provides a set of interfaces for abstracting various OS support functions.
Components	• Memory Management Unit: Abstracts allocation and freeing of memory blocks. • Messages and Queues: Abstracts for creating and deleting of message queues and sending and receiving messages to and from message queues. • Clocks and Timers: Abstracts adding and freeing timers and presenting current time. • File System: Abstracts file creating, deleting, reading, and writing. • Synchronizers: Abstracts creating, deleting, and passing of semaphores. • Tasks: Abstracts task creation, deletion, delay, locking and unlocking of task switching. • Interrupts: Abstracts disabling and enabling external system interrupts. • Network: Abstracts various RTOS network functions. • Real-Time Operating System (RTOS): Abstracts operating systems with which other systems can interface.
Interacts with	• System Hardware Component • System Support • OS

Table 4.7. System Application Programming Interface (SYSAPI)

Functionality	The SYSAPI component provides systemwide routines for network buffer support and provides the interface into the system registry.
Components	• Nonvolatile Memory: Abstracts reading and writing nonvolatile memory. • Network Buffers: Abstracts managing network buffers. • Network: Abstracts various network parameter settings and network utilities. • Registry: Discovers current hardware platform and loads registry to reflect current hardware platform, presents the registry to applications, and maintains and updates registry database to reflect hardware updates. • Presentation Interface: Presents the registry to the user API interface. • Updates Handler: Responsible for updating the hardware database when hardware configurations change during runtime.

	• Hardware Discovery: Populates the registry at boot time describing the hardware platform. • Current Hardware Database: Manages the allocation from the network database on the hardware.
Interacts with	• System Hardware Component • OSAPI

Table 4.8. System Hardware Component

Functionality	The System Hardware Component is responsible for providing an interface to all system hardware components except those handled by the Advanced Network Device Layer. Therefore, the underlying hardware details are hidden from the caller.
Components	• 1st-Level Interrupt Handler: Provides interrupt service routines that are called when an interrupt is detected by the system processor. • CPU and PCI Bridge: Provides support for accessing the system’s Bridge. • File System: Provides support for reading and writing to the File System. • VPD: Provides support for reading and writing the system’s Vital Product Data areas. • Serial Port: Provides support for reading and writing the system’s serial management port. • Fan: Provides support for detecting and reporting proper fan operation. • Power Supply: Provides support for detecting and reporting proper power supply operation. • Management Port: Provides support for reading and writing to system’s management port. • Boot: Provides support for initial reset code and verifying and starting operational code. • NVRAM: Provides support for reading and writing to the system’s non-volatile memory regions. • MDIO: Provides support for reading and writing to the system’s MDIO areas. • Diagnostics: Provides support for the execution and reporting of diagnostics.
Interacts with	• System Hardware Component • OSAPI

4.4. Instrumentation System Support

The BroadView™ Instrumentation Agent has been integrated with ICOS to provide buffer statistics tracking (BST) on ICOS. The BroadView Instrumentation Agent provides a way to access the instrumentation features of the underlying silicon.

The BroadView™ Instrumentation Agent establishes communication with underlying silicon via the ICOS network operating system. It collects various instrumentation statistics, processes and packages the data appropriately, and provides the data to an interested Collector. Similarly, the Agent configures the silicon based on the configuration requests from the Collector.

The Agent communicates with the Collector using REST-style communication, with the data exchange in the JSON-RPC (2.0) format. The Agent supports both the pull model of operation (where the Collector requests data and obtains it) as well as push model of operation (where the Agent sends periodic reports, asynchronously).

Chapter 5. Management Abstraction Layer

This section describes the design and implementation of the Management Abstraction Layer.

5.1. Unit Stack Management Database (USMDB)

The USMDB is the software interface that facilitates communication between the management layer and the rest of the software. The USMDB interacts with management layer components such as the CLI, most components of the application layer, and system support. To view a diagram of the position of the USMDB in relation to the ICOS architecture, see Figure 2.1, “System Layers”.

5.1.1. USMDB Architecture

The Overall Architecture diagram in Figure 2.1, “System Layers” illustrates the interaction between the USMDB and the following components:

- User Interface applications such as CLI
- Protocol modules (Layer 2, Layer 3, and so on)
- Network Interface Module (NIM)
- System Support
- Statistics Manager

5.1.2. Application Functionality and Features

The Unit Stack Manager Database (USMDB) Layer is a software interface that facilitates communication between the Management layer (CLI and SNMP) and the rest of the ICOS software. This layer provides isolation between the user interface and the ICOS applications, allowing a third-party UI to be written for ICOS. This is accomplished through a large library of pass-through APIs that direct and translate commands to the proper modules. USMDB is designed as a thin layer, with most of the work performed by the underlying components.

The USMDB APIs provide the interfaces to layer 2 protocols and applications, such as 802.1Q, LAG, and Forwarding Database. The APIs also provide the interfaces to Layer 3 protocols, such as BGP and OSPF, and support for various MIB.

The primary features of the USMDB are:

- Flexible and independent: Each module operates independently of the others so that any changes made to one do not affect any other. This allows a set of individual modules that are easy to extend, change, and reshape.
- Customizable: The USMDB functions with any combination of modules depending on the requirements. The USMDB provides the flexibility to add or remove various application or software modules with ease.
- Maintainable: The software is easy to maintain, both architecturally and within the code.
- Isolated: USMDB isolates the management layer from the application layer APIs; changes to application layer APIs can be contained within the USMDB layer and the Config/USM code is unaffected.

Chapter 6. Application Layer

The application layer section describes the application layer modules. The modules are listed below and shown in Figure 2.1, “System Layers”:

- Section 6.1, “Base/Security Features”
- Section 6.2, “Layer 2—Switching Module”
- Section 6.3, “Data Center Module”
- Section 6.4, “Layer 3—Routing Module”
- Section 6.5, “IPv6 Routing and Management”
- Section 6.6, “Quality of Service Module”
- Section 6.7, “IP Multicast Module”

Each of the above modules is described in the sections that follow.

6.1. Base/Security Features

The Base features and Security features provide support for system capabilities such as configuration management, logging, DHCP, and system utilities. The following features are described in this section:

- Section 6.1.1, “Malicious Code Detection”
- Section 6.1.2, “Flow Control”
- Section 6.1.3, “Asymmetric Flow Control”
- Section 6.1.4, “File Download”
- Section 6.1.5, “Non-Disruptive Configuration Management”
- Section 6.1.6, “Bootstrap Protocol (BOOTP)”
- Section 6.1.7, “DHCP Client”
- Section 6.1.8, “XMODEM Protocol”
- Section 6.1.9, “AutoInstall Support”
- Section 6.1.10, “Auto Image Upgrade”
- Section 6.1.11, “Cable Testing”
- Section 6.1.12, “sFlow”
- Section 6.1.13, “DNS Client”
- Section 6.1.14, “Traceroute”
- Section 6.1.15, “Simple Network Time Protocol”
- Section 6.1.16, “Time Ranges Component”
- Section 6.1.17, “Port Description”
- Section 6.1.18, “Flash Robustness”
- Section 6.1.19, “Serviceability”
- Section 6.1.20, “DoS Protection”
- Section 6.1.21, “Industry Standard Discovery Protocol (ISDP)”
- Section 6.1.22, “RADIUS”
- Section 6.1.23, “TACACS+”
- Section 6.1.24, “IP Address Conflict Notification”
- Section 6.1.25, “Warm Reload”
- Section 6.1.26, “Port Mode Change”

- Section 6.1.27, “Switch Database Management Templates”
- Section 6.1.28, “Statistics Application”
- Section 6.1.29, “Dynamic Warpcore™”
- Section 6.1.30, “Syslog Support”
- Section 6.1.31, “Source IP Address Configuration”
- Section 6.1.32, “Factory Default Configuration File”
- Section 6.1.33, “Chef Client Integration”
- Section 6.1.34, “Puppet Client Integration”
- Section 6.1.35, “MPLS”
- Section 6.1.36, “Interface Error Disable and Auto Recovery”
- Section 6.1.37, “Watchdog Services”
- Section 6.1.38, “Packet Trace”
- Section 6.1.39, “RESTful APIs”

6.1.1. Malicious Code Detection

ICOS provides a mechanism to detect the integrity of the image if the software binary is corrupted or tampered while the administrator attempts to download the software image to the switch. ICOS addresses this problem by using digital signatures to verify the integrity of the binary image. It also provides flexibility to download a digitally signed configuration script and verify the digital signature to ensure the integrity of the downloaded configuration file.

6.1.2. Flow Control

Flow control is a mechanism or protocol used to temporarily suspend transmission of data to a device to avoid overloading the device receive path. The ICOS software implements the flow control mechanism defined in IEEE 802.3 Annexes 31A and 31B (formerly IEEE 802.3x). The ICOS software is able to transmit a MAC Control frame containing the PAUSE opcode to halt transmission by the device receiving the PAUSE frame for a specified time period. Since the protocol is only implemented on full-duplex links, there is no ambiguity as to the identity of the devices involved. Flow control may be enabled for all full-duplex ports on the box.

6.1.3. Asymmetric Flow Control

Asymmetric Flow Control can only be configured globally for all ports on StrataXGS® silicon-based switches. When in asymmetric flow control mode, the switch responds to PAUSE frames received from peers by stopping packet transmission, but the switch does not initiate MAC control PAUSE frames. When the switch is configured in asymmetric flow control (or no flow control mode), the device is placed in egress drop mode. Egress drop mode maximizes the throughput of the system at the expense of packet loss in a heavily congested system, and this mode avoids head of line blocking. Asymmetric flow control is not supported on Fast Ethernet platforms, as the support was introduced to the physical layer with the Gigabit PHY specifications.

In asymmetric flow control mode, the switch advertises the symmetric flow control capability, but forces the Tx Pause to OFF in the MAC layer. At PHY level, Pause bit = 1, and ASM_DIR =1 have to be advertised to peer. At Driver level, Tx Pause = 0, and Rx Pause = 1. The operational state (MAC layer) of receive Flow Control (Rx) is based on the pause resolution table 5. The operational state (MAC layer) of Flow Control on Send side (Tx) is always Off.

6.1.4. File Download

ICOS supports download of following file types to the switch using FTP/TFTP/SFTP/SCP protocols:

- Code
- Configuration
- Text configuration
- SSH keys and certificates
- SSL keys and certificates
- CLI banner file

6.1.4.1. FTP and TFTP

FTP and TFTP are standard network protocols used to transfer files. These protocols can read and write files to and from a remote server. FASPATH includes both FTP and TFTP clients that communicate with a server. FTP is defined in RFC 959. Upon failure of a FTP transfer operation, a log message is sent to the logging component, the initiating application is notified of the failure, and any partial or temporary files for the transfer are removed from persistent memory.

TFTP is a simple protocol and lacks many of the features supported by FTP. The TFTP transfer begins with a request to a server to read or write a file. If the server grants the request, the connection is opened and the file is transferred in 512-byte blocks of data. Each packet is acknowledged separately before the next packet is sent. The acknowledgement of a data packet of less than 512 bytes indicates the end of the transfer. TFTP interacts with BOOTP to load the boot file into the system.

FTP/TFTP can be used to transfer multiple file types such as configuration, error log, trap log, and system trace files. The FTP/TFTP component running on a Linux device calls external applications.

6.1.4.2. SCP and SFTP

ICOS supports Secure Copy (SCP) and Secure FTP (SFTP) as methods of file transfer that allow secure file transfer to/from an ICOS switch.

6.1.5. Non-Disruptive Configuration Management

In the data center network, where the network administrator may manage thousands of switches, when the switch configuration is changed by uploading a new configuration file to it, the switch can gracefully resolve any differences between the running configuration and the new configuration. For example if the switch has VLANs 10, 20, and 30 configured, and the new configuration has VLANs 10, 20, and 40, the switch deletes VLAN 30 and creates VLAN 40 without disturbing traffic forwarding on VLANs 10 and 20.

Without this feature, to upgrade to a new configuration, the administrator must either provide a new configuration file and restart the switch or upload a ‘delta’ configuration. Restarting the switch is disruptive, and managing delta configurations is very difficult on the large scale.

6.1.6. Bootstrap Protocol (BOOTP)

The Bootstrap Protocol (BOOTP) is a user Datagram Protocol/Internet Protocol (UDP/IP) that allows devices to obtain IP information from a BOOTP server. BOOTP facilitates the configuration of the ICOS software in remote locations. When it is used to get the switch IP information, the ICOS software must be configured to use the appropriate network configuration protocol. When BOOTP is selected, the switch periodically sends out requests until a response is received from the BOOTP server.

6.1.7. DHCP Client

The DHCP client is a UDP protocol that allows devices to obtain stateful DHCP information such as the IP address, network mask, and gateway, and stateless DHCP information such as DNS servers, TFTP servers, etc. from the DHCP server. The ICOS software allows the user to enable the client protocol on the DHCPcapable interfaces, which include the network port (if this feature is enabled on the device), service port (or out-of-band interface), as well as host and routing interfaces.

DHCP client is implemented based on the RFC 2131 standard, “Dynamic Host Configuration Protocol”. Refer to RFC 2131 for more protocol details.

DHCP client-Identifier option (Option 61) is used by DHCP clients to specify their unique identity. The client identifier is a string with a minimum length of 2 bytes and a maximum length of 128 bytes. It is interpreted by DHCP servers or relay agents. The current specification deals with the DHCP client only. This feature is configurable on the Network port, Service port, and In-band interfaces. It is enabled by default.

6.1.8. XMODEM Protocol

ICOS software uses the XMODEM protocol to transfer operational code, configuration files, and logs using the serial port. ICOS software supports both the XMODEM standard mode and the XMODEM-1K mode. The XMODEM utility is bundled with the Linux kernel.

6.1.9. AutoInstall Support

The AutoInstall feature allows the network manager to load the configuration on a switch automatically when the device is initialized and no valid configuration file is found on the switch. AutoInstall begins whenever an ICOS device is turned on and no valid configuration file is detected. In ICOS, the AutoInstall process requires that the DHCP client is enabled on its network interface by default. This allows the switch to initiate the AutoInstall procedure when it is powered up. While obtaining the network IP address from the configured BOOTP/DHCP server, the DHCP client requests the

TFTP server address, DNS server address, and the configuration file name using DHCP/BOOTP options 6, 66, 67, 150 and other DHCP fields. The AutoInstall feature applies the configuration fetched from the TFTP server and automatically saves the configuration into nonvolatile memory based on the configuration option. This feature is useful when the administrator has only remote access to the device.

6.1.10. Auto Image Upgrade

Auto-Image Upgrade allows the device to upgrade to newer a software image or bootfile automatically during initialization, with limited administrative configuration on the device. A DHCP server provides the IP address and other information necessary for the switch to upgrade to newer image. The process may also include autoconfiguration of the device. This feature works in conjunction with AutoInstall feature, where the device is autoconfigured during initialization (see “AutoInstall Support” on page 64). This feature is primarily intended for use to upgrade multiple switches in the network to the same software version, and to synchronize the newly deployed switch in a network with the images in the existing switches.

6.1.11. Cable Testing

ICOS administrators can run cable tests on a copper cable connected to on any physical port on the device. The cable test feature is not supported for optical fiber cables. The cable test can be run at all the following port speeds and duplex modes:

- 10 Mbps Full/Half duplex
- 100 Mbps Full/Half duplex
- 1000 Mbps Full/Half duplex

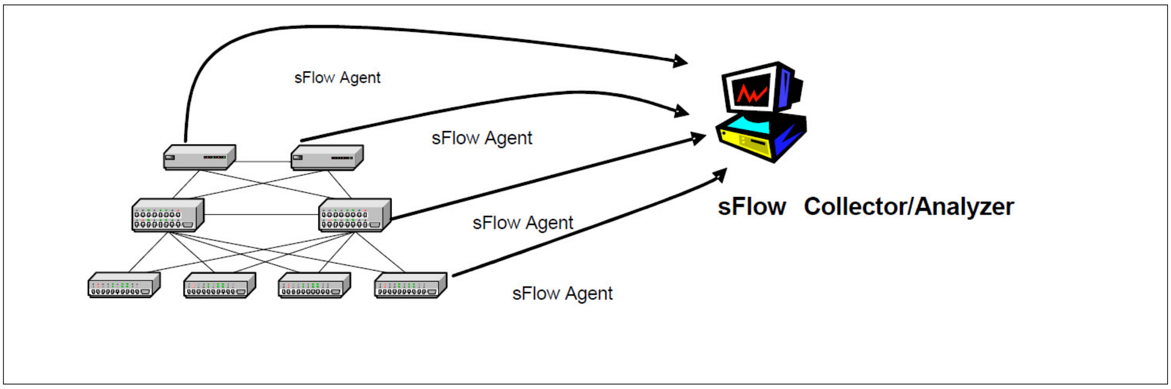
If the port has an active link while cable test is run, the link can go down for the duration of the test. It may take several seconds to run the test. Following result is presented to the user after the cable test is executed:

- Cable Status: The cable status is reported as Normal, Open or Short.
  - Normal: Indicates the cable is working correctly.
  - Open: Indicates the cable is disconnected or there is a faulty connector.
  - Short: Indicates there is an electrical short in the cable.
- Cable length: Shows the estimated length of the cable in meters. The length is calculated as a range between the shortest estimated length and the longest estimated length.
- Failure location: Indicates the estimated distance in meters from end of the cable to the failure location. The failure location is valid only if the cable status is Open or Short.

6.1.12. sFlow

sFlow is the standard for monitoring high-speed switched and routed networks. sFlow technology is built into network equipment and gives complete visibility into network activity, enabling effective management and control of network resources.

Figure 6.1. sFlow in a Network



The sFlow monitoring system consists of an sFlow Agent (embedded in a switch or router or in a stand-alone probe) and a central sFlow Collector. The sFlow Agent uses sampling technology to capture traffic statistics from the device it is monitoring. sFlow datagrams are used to forward the sampled traffic statistics immediately to an sFlow Collector for analysis.

The sFlow Agent uses two forms of sampling: statistical packet-based sampling of switched or routed Packet Flows and time-based sampling of counters.

The sFlow agent can export LAG counters for a port that is part of a LAG, along with Ethernet counters. The LAG counters exported to the sFlow receiver in the counter sample includes the following information:

- Actor and Partner ports system IDs.
- Actor and Partner ports admin and operational states.
- Statistics about different types of packets exchanged between actor and partner ports such as number of LACPDUs, Marker PDUs, Marker Response PDUs etc.,

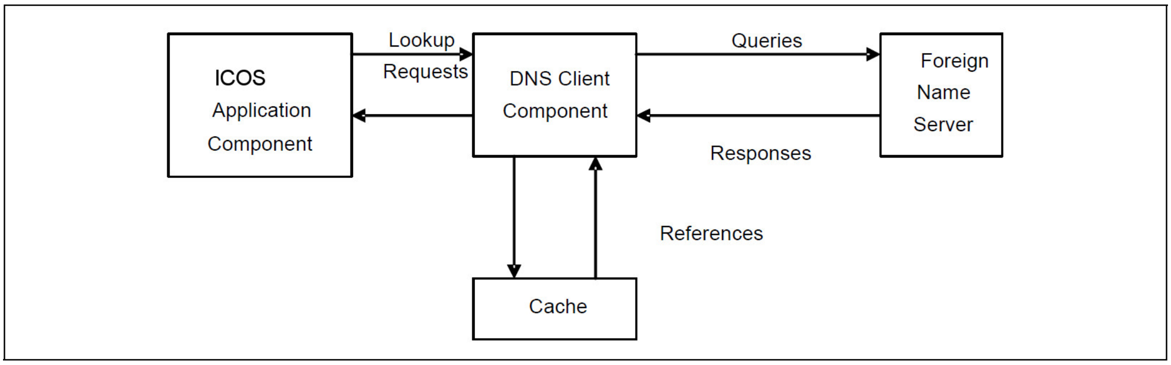
This additional information (LAG counters) is exported in the sFlow counter sample only if sFlow is supported on LAG members.

ICOS supports packet sampling in hardware on Netberg higher grade platforms. Packet sampling in hardware does not require the sampled packet to be copied to the CPU for processing and is, therefore, less CPU-intensive (However, the counter sampling mechanism is performed in software.)

6.1.13. DNS Client

The Domain Name System (DNS) is an Internet directory service. DNS is how host names are translated into IP addresses. The DNS Client component, when enabled, provides a host name lookup service to other components of ICOS. DNS client service can be globally enabled or disabled. The figure below shows the operation of the DNS client component.

Figure 6.2. DNS Client Operation in the Network



The ICOS DNS client contacts one or more configured DNS servers to resolve a host name to an IP address. The list of servers is configured by providing an IP address for each DNS server. When more than one DNS server is configured in the system, server precedence is determined by the order in which the servers are added. A DNS server can be configured using an IPv4 address. DNS server address can be static or dynamic. A static DNS server refers to the one configured by the user. A dynamic DNS server is the one that is dynamically obtained by the DHCP client.

A default domain name can be configured, which defines the domain to use when performing a lookup on an unqualified host name. A default domain-name list can be configured. If there is no domain list, the default domain name configured is used. If there is a domain list, the default domain name is not used. An entry in domain-name list can be static as well as dynamic. A static domain name entry refers to the one configured by user as part of domain-name list configuration. A dynamic domain-name entry refers to the one which is dynamically obtained by DHCP client running in the system.

The DNS client in ICOS operates in recursive mode, which means the DNS client sets the recursion desired bit and the server contacts any other name server for host name resolution. With recursive mode, the server returns a response to the client and never refers the client to any other server for name resolution.

DNS names accept spaces in the Host Names, but consecutive spaces are not supported.

6.1.14. Traceroute

Traceroute is used to discover the routes that packets actually take when traveling to their destination through the network on a hop-by-hop basis. The *traceroute* command output displays all network layer (Layer 3) devices, such as routers, that the packet passes through on the way to the destination. This utility can be used to get the route to the destination host, detect issues on the network and to obtain round-trip time (RTT); such as remote routing problems can cause the no answer error message, as well as the *network unreachable* error message. But the *network unreachable* message does not always signify a routing problem. It can mean that the remote network cannot be reached because something is down between the local host and the remote destination. Traceroute can help to locate these problems.

The user can specify the initial and maximum time to live (TTL) in probe packets, the maximum number of failures before termination, the number of probes sent for each TTL, and the size of each probe. A traceroute can also be initiated via SNMP.

6.1.15. Simple Network Time Protocol

The Simple Network Time Protocol (SNTP) is widely used for synchronizing network resources. SNTP Version 4 is described in RFC 2030. SNTP is an adaptation of the Network Time Protocol (RFC 1305) useful for situations where the full performance of NTP is not justified. SNTP can operate in unicast mode (point-to-point) or broadcast mode (point-to-multipoint). Various NTP implementations can operate as either a client or a server. To an NTP or SNTP server, NTP and SNTP clients are indistinguishable. Likewise, to an NTP or SNTP client, NTP and SNTP servers are indistinguishable. Furthermore, any version of NTP is compatible with any other version of NTP. ICOS SNTP implements the client side of SNTP. ICOS software can communicate with SNTP time servers over IPv4 or IPv6 networks.

The SNTP and time configuration feature includes the ability to configure the time zone and daylight savings time functionality.

6.1.16. Time Ranges Component

The Time Ranges component, which is an infrastructure component, supports time-based ACLs by providing a notion of time to other ICOS components. It can be used by other components to allow the network administrators to apply and remove configuration based on time of the day. To do so, a time range is created that defines specific times of the day and week. The time range is identified by a name and then referenced by some other configuration parameters, for example, an ACL rule, so that those time restrictions are imposed on them.

6.1.17. Port Description

This feature allows user access to assign a text string to describe the port to the interface. The text is used only for reference.

6.1.18. Flash Robustness

The Boot Loader solutions are designed to maximize flash robustness in ICOS products. The goal is to minimize the opportunity for failures that leave a device in a state that requires nontrivial user intervention or, in particular, a field return.

This can be achieved by techniques such as:

- Creation of failover mechanisms in the platforms
- Preventive processes to avoid accidental corruptions and errors
- Auto recovery and recovery mechanisms that require minimal human intervention

6.1.19. Serviceability

ICOS serviceability enables easy debugging and provide the debug functions in normal commands. The following section describes several debug commands and shows how they help in debugging and diagnosing the system.

6.1.19.1. Support Mode

ICOS software contains a support mode that is hidden and available only when the *techsupport enable* command is run. Support mode is disabled by default. When the user enters support mode, help for all the commands in support mode is available. Commands in support mode can be executed like normal commands and do not have to be prefixed with support key words. Configurations related to support mode are shown in the show tech-support command. They can be persisted by using the command save in support mode. Support configurations are stored in a separate binary configuration file, which cannot be uploaded or downloaded.

⚠ Support commands should only be used by the manufacturer’s technical support personnel as improper use could cause unexpected system behavior and/or invalidate product warranty.

6.1.19.2. CPU Utilization

CPU Utilization provides information on the usage of the CPU by specific tasks.

CPU Utilization is not concerned with the traffic to the CPU, but tasks that keep the CPU busy.

The percentage of CPU utilization by task is reported for the following time frames:

- Five seconds
- One minute
- Five minute

The CLI command to access CPU utilization is *show process cpu*.

For more information on the command, refer to the *ICOS CLI Command Reference*.

The administrator can configure a threshold values of CPU utilization at which log/trap events are generated.

The log/trap displays the top usage tasks.

6.1.19.3. Memory Utilization

The administrator can configure a free memory threshold value at which log/trap events are generated. The log/ trap displays the top memory usage tasks.

6.1.19.4. Debug Commands for Protocols

Debug commands cause the output of the enabled trace to display on a serial port or telnet console. Note that the output resulting from enabling a debug trace always displays on the serial port. The output resulting from enabling a debug trace displays on all login sessions for which any debug trace has been enabled. The configuration of a debug command remains in effect the whole login session.

The output of a debug command is always submitted to the syslog utility at a DEBUG severity level. As such, it can be forwarded to a syslog server, stored in the buffer log, or otherwise processed in accordance with the configuration of the syslog utility. Configuration of console logging in the syslog utility is not required in order to view the output of debug traces.

Debug commands are provided in the normal CLI tree. Debug settings are not persistent and are not visible in the running configuration. To view the current debug settings, use the **show debug** command.

The output of debug commands can be large and may adversely affect system performance.

The following commands are added to the debug framework.

Table 6.1. Debug Commands

Debug Command	Description
arp	Configure ARP debug flags
bgp	Configure BGP debug flags.
clear	Clear all debug flags.
console	Enable/disable session display of debug trace output.
dot1x	Configure Dot1x debug flags.
igmpsnooping	Configure IGMP snooping debug flags.
ip	Configure IP debug flags.
ipv6	Configure IPv6 debug flags.
isdp	Debug ISDP packets.
lacp	Configure lacp debug flags.
ospf	Configure OSPF debug flags.
ping	Configure ping debug flags.
sflow	Configure sFlow debug flags.
spanning-tree	Configure spanning-tree debug flags.

Enabling debug for all IP packets can cause a serious impact on the system performance; therefore, it is limited by ACLs. This means debug can be enabled for IP packets that conform to the configured ACL. This also limits the feature availability to only when the QoS component is available. Debug for VRRP and ARP are available on routing builds.

For more information on the above commands, refer to the *ICOS CLI Command Reference*.

6.1.19.5. Persistent CLI Command History

CLI Commands are logged to the existing Persistent log. When persistent logging is enabled, logging still happens to a file in the RAM and is written to Flash memory when any of the following events occur.

- Logging buffer is full and about to wrap

- Resetting of system
- Crashing of system
- Elapsing of Timer

A timer is started after receiving the first message. If any of the above events do not occur till the timer expires, and the timer elapses, messages from RAM are written to Flash memory. This ensures periodic saving of the messages.

So, for CLI command logging to work, the user must enable persistent logging and command logging features. Persistent logging is disabled by default. The default severity for persistent logging is configured for alert messages.

Persistent logs can be uploaded to a FTP/TFTP/SCP/SFTP server. If the system reboots, the current persistent log files are renamed and saved to persistent memory. Previous logs can also be uploaded to a remote server.

6.1.19.6. Password Recovery

To make it easier to reconfigure the admin password without losing the configuration, a Password Recovery Mode option is available in the boot menu. Password Recovery Mode is available only for the admin user. If the user chooses this option and starts the application, the password and login-type for the admin are skipped while applying the start-up config. This allows a user to log in to the device as admin with the default password.

This option works only for the session when a user has started the device with Password recovery mode option. If the device is restarted, the device expects the start-up configuration password. The user must configure the password and save it after starting the device in the Password recovery mode.

Passwords and login-types of all other users remain as per the start-up configuration and are not changed while logging into the device even in Password Recovery Mode.

6.1.19.7. Packet Trace

Debug commands can be used to perform packet traces for the VRRP, IP, ARP, and L2, protocol packets. Packet trace for the IP protocol is mandated to be limited by the ACL. Only traces of IP packets that match the ACL are printed.

6.1.19.8. Debug Capability for Memory Dumps

The software includes a compile-time option to enable routing heap file/line tracing. The option is turned off by default to save memory. By enabling this option, the increase in routing heap is about 30%.

6.1.19.9. Configurable Break-in String

A configurable break-in string can be used to enter into the break-in console.

6.1.19.10. Email Alerts

Email Alerting is an extension of the logging system. The ICOS logging system allows the user to configure a variety of destinations for log messages. This feature adds the email configuration, whereby log messages are sent to a configured SMTP server such that an administrator receives the logs in a specified email account.

6.1.19.11. System Log Sorting

System logs are displayed in reverse chronological order (newest first).

6.1.19.12. Redirecting Debug Data over Telnet

A telnet session and port number can be configured for sending debug data in a serialized manner.

6.1.19.13. Crash Dump Information

Crash dumps provide system information and have the following characteristics:

- Modules can register a contiguous block of memory to be collected when the system crashes.
- If the file system is still up after a crash dump, all blocks of memory are collected, compressed and saved to a local file.
- A signature for the block of memory is also stored to identify it.
- This is an alternative to saving all data sections of memory where there is a concern it will not compress enough to fit in flash.
- Crash dumps are saved to contiguous blocks of memory to prevent the save from failing due to corrupted ICOS data structures.
- Crash dump files can be copied to a remote system via a USB port (if supported), by retrieving the file via NFS (if supported), or by file transfer using TFTP or FTP. On systems that include sufficient available flash memory (several gigabytes or more), the Crash dump file can also be written to flash.

6.1.19.14. Support for Compile-Time Enable of Heap File/Line Tracing

In ICOS, there is an option to enable/disable file/line tracking in the routing heap. This option is controlled by setting *DO_MEMCHK in ip_exports.h*. This option is turned off to save memory. ICOS enables the option for tracking the routing heap. When enabled, the routing heap is increased by about 30%.

6.1.19.15. Execution of Show Commands in Any Command Mode

The CLI supports the ability to view the output of any show command from any command mode.

6.1.19.16. Retrieving Configuration Information for Technical Support

The **show tech-support** CLI command can be used to display system and configuration information. This information may assist technical support in troubleshooting customer problems. The command combines the output from various other show commands, which can be configured. It also includes message queue information. The output can be written to a file for uploading to a server.

6.1.19.17. Additional Serviceability Features

ICOS software includes the following additional serviceability features:

- **MBUF Debug Scheme**: provides ability to configure MBUF threshold limits and generating notification when MBUF limits have reached.
- **Full Memory Dump (core dump)**: provides the ability to retrieve the state from a crashed system such that it can be then loaded into a debugger, in which the state can be recreated.

In addition to generating core dump file for crashes that occur during normal operation, a core dump file can be generated during system initialization and when the system is in an unconfigured state. The core dump feature can be preconfigured by providing configuration parameters in a boot menu option.

The core dump feature includes a tracing mechanism in a shell script that writes the various traces in a file in the file */mnt/fastpath/coredump_log.txt*. If the file already exists then it is backed up to a file */mnt/fastpath/coredump_log.old*.

For systems that include a hardware Reset button, the full memory dump feature includes an option to be generated a non-maskable interrupt event (NMI).

- LOG_ERROR: provides the run-time registration mechanism with LOG_ERROR. Additionally, an application/component exists to provide support for the crash recorder infrastructure. This is a compile-time option.
- Task Execution Dump: provides the snapshot of osapiDebugStackTrace to get to know what other tasks are doing when the crash happens.

6.1.20. DoS Protection

Several Denial of Service (DoS) attacks have been documented and characterized. The Netberg hardware platforms based on Broadcom Tomahawk has support for classifying and blocking multiple types of attacks. If the DoS function is enabled, the silicon classifies packets as they ingress the switch and if one of the enabled attacks is detected, the packets are dropped. This protects other devices in the network from receiving the

Denial of Service attacks and potentially causing harm. The following list shows the DoS attack detection that ICOS supports. Some platforms do not support detection for all of the DoS attack types in the list.

- SIP=DIP:
  - Source IP address = Destination IP address.
- First Fragment:
  - TCP Header size smaller then configured value.
- TCP Fragment:

  - IP Fragment Offset = 1.
- TCP Flag:
  - TCP Flag SYN set and Source Port < 1024 or TCP Control Flags = 0 and
  - TCP Sequence Number = 0 or TCP Flags FIN, URG, and PSH set and
  - TCP Sequence Number = 0 or TCP Flags SYN and FIN set.
- L4 Port:
  - Source TCP/UDP Port = Destination TCP/UDP Port.
- ICMP:
  - Limiting the size of ICMP Ping packets.
- SMAC=DMAC:
  - Source MAC address = Destination MAC address.
- TCP Port:
  - Source TCP Port = Destination TCP Port.
- UDP Port:
  - Source UDP Port = Destination UDP Port.
- TCP Flag & Sequence:
  - TCP Flag SYN set and Source Port < 1024 or TCP Control Flags = 0 and
  - TCP Sequence Number = 0 or TCP Flags FIN, URG, and PSH set and
  - TCP Sequence Number = 0 or TCP Flags SYN and FIN set.
- TCP Offset:
  - Checks for TCP header offset =1.
- TCP SYN:
  - TCP Flag SYN set.
- TCP SYN & FIN:
  - TCP Flags SYN and FIN set.
- TCP FIN & URG & PSH:
  - TCP Flags FIN and URG and PSH set and TCP Sequence Number = 0.
- ICMPv6:

  - Limiting the size of ICMPv6 Ping packets.
- ICMP Fragment:
  - Checks for fragmented ICMP packets.

6.1.21. Industry Standard Discovery Protocol (ISDP)

Industry Standard Discovery Protocol (ISDP) is a proprietary layer 2 network protocol which interoperates with Cisco network equipment and is used to share information between neighboring devices. ICOS participates in the ISDP protocol and is able to both discover and be discovered by devices that support the Cisco Discovery Protocol (CDP).

6.1.22. RADIUS

Managing and determining the validity of users in a large network can be significantly simplified by making use of a single database of accessible information as in an Authentication Server. Remote Authentication Dial-In User Service (RADIUS) servers commonly support the Remote Authentication Dial-In User Service (RADIUS) protocol as defined by RFC 2865. RADIUS permits access to a users authentication and configuration information contained on the server only when requests are received from a client that shares an encrypted secret with the server. This secret is never transmitted over the network in an attempt to maintain a secure environment. Any requests from clients that are not appropriately configured with the secret or access from unauthorized devices are silently discarded by the server.

Silently discarded packets are abandoned without any further processing, but the ICOS software RADIUS Client services generates logs and increments status counters to record these occurrences.

RADIUS conforms to a client/server model with secure communications using UDP as a transport protocol. It is extremely flexible, supporting a variety of methods to authenticate and statistically track users. It is very extensible, allowing for new methods of authentication to be added without disrupting existing functionality. The RADIUS client supports up to 32 named authentication and accounting servers.

6.1.23. TACACS+

TACACS+ provides access control for networked devices using one or more centralized servers, similar to RADIUS this protocol simplifies authentication by making use of a single database that can be shared by many clients on a large network. TACACS+ is based on the TACACS protocol (described in RFC1492) but additionally provides for separate authentication, authorization and accounting services. The original protocol was UDP based with messages passed in clear text over the network; TACACS+ uses TCP to ensure reliable delivery and a shared key configured on the client and daemon server to encrypt all messages.

6.1.24. IP Address Conflict Notification

This feature detects and reports an IP address conflict detected for an active IP address on any IP interface in the system, including service port, network port, routing, or host interfaces. A message is logged and an SNMP trap is reported when a conflict occurs. A conflict is detected when an ARP packet is received on an interface with the sender IP address matching any of the interface IP address.

6.1.25. Warm Reload

The Warm Reload feature reduces the time it takes to reboot a Linux switch, thereby reducing the traffic disruption in the network during a switch reboot. For a typical Linux switch, the traffic disruption is reduced from about 2 minutes for a cold reboot to about 10 seconds for a warm reboot.

A warm reload restarts only the application process; it does not restart the boot code, the Linux kernel, and the root file system. Since the warm reload does not restart all components, some code upgrades require that customers perform a cold reboot.

A warm reload can only be initiated by the administrator; it cannot happen automatically. The resets caused by the LOG_ERROR macro or process exceptions are always cold resets.

6.1.26. Port Mode Change

The ports on the Netberg hardware platform based on Broadcom Tomahawk family of devices are configurable in either 100/40 Gbps mode or 25/10 Gbps mode. ICOS software provides a configuration option for changing a port from one mode to the other without requiring the user to recompile the ICOS software. When the port is configured in 100 Gbps mode, the four 25 Gbps ports at the same physical interface are disabled. A switch reset is required after a port mode change.

6.1.27. Switch Database Management Templates

An SDM template provides a description of the maximum resources a switch or router can use for various features. Different SDM templates allow different combinations of scaling factors, enabling different allocations of resources depending on how the device is used.

ICOS supports the following SDM templates. The end user has the ability to configure a switch or router in one of these modes:

- (Default) IPv4 Data Center Plus SDM Default template—Sets the IPv4 scaling factors to those used in IPv4-only builds, sets the IPv6 scaling factors to 0, and increases the maximum number of ECMP next hops.

- IPv4-only SDM template—Sets the IPv6 scaling factors to 0, even though the build supports IPv6, and sets the IPv4 scaling factors to those used in IPv4-only builds, without the Data Center package.

- IPv4 Data Center SDM template—Sets the IPv6 scaling factors to 0, sets the IPv4 unicast scaling factors to those used in builds with IPv6, and increases the maximum number of ECMP next hops.

- Dual IPv4/IPv6 SDM template—Uses the IPv4 scaling factors when IPv6 routing is included in the build.

- Dual IPv4/IPv6 Data Center SDM template—Uses the IPv4 scaling factors when IPv6 routing is included in the build and increases the maximum number of ECMP next hops.

Applications that enforce scaling limits specified in an SDM template enforce the template values rather than platform or product constants. For example, when the routing table manager adds a new IPv4 route, it compares the number of existing routes to the template limit to determine whether the routing table is already full.

In ICOS software, SDM templates define the following parameters:

- ARP Entries
- IPv4 Unicast Routes
- IPv6 NDP Entries
- IPv6 Unicast Routes
- ECMP Next Hops
- IPv4 multicast routes
- IPv6 multicast routes

The operational SDM template can be selected using the CLI management interface. When a user changes the SDM template via the management interface, the switch must be rebooted before it will take affect.

6.1.28. Statistics Application

The statistics application collects the statistics at a configurable time interval. The user can specify the port number(s) or a range of ports for statistics to be displayed. The configured time interval applies to all ports. Detailed statistics are collected between the specified time range in date and time format. The time range can be defined as having an absolute time entry and/or a periodic time. For example, a user can specify the statistics to be collected and displayed between 9:00 12 NOV 2011 (START) and 21:00 12 NOV 2011 (END) or schedule it on every MON, WED and FRI 9:00 (START) to 21:00 (END).

The user receives these statistics in a number of ways:

- Using the CLI to request a set of counters.
- Configuring the device to display statistics using syslog or email alert. The syslog or email alert messages are sent by statistics application at the configured END time.

The statistics are presented on the console at the configured END time.

6.1.29. Dynamic Warpcore™

The Warpcore feature is a SerDes IP core integrated into the Netberg hardware platforms based on Broadcom Tomahawk family of devices. The Warpcore implementation on Netberg hardware platforms based on Broadcom Tomahawk devices contains four independent SerDes lanes that can operate from 10 Mbps to 25 Gbps, and each 100G Warpcore port can be *expanded* to up to four ports capable of independently operating at max 25G speeds. To take advantage of this functionality, ICOS presents the *Expandable Ports* feature to the user. This feature allows the user to configure the Warpcore modes of the port using the CLI, web, or SNMP.

ICOS supports two Warpcore versions: static and dynamic. Due to Netberg hardware restrictions, the feature is available in a *static* version on Netberg hardware boards. In this version, any user configuration is stored as part of HPC persistent data and is applied only on the next reboot. In both Warpcore versions, the user configuration will also be visible as part of the running configuration of the original interface.

In both static and dynamic Warpcore versions, when the operating mode of the port is changed, ICOS not only changes the speeds but also attaches a different interface representing the port. The original port is moved to the detached state. All interfaces are displayed to the user with their corresponding *attached* and *detached* states representing the port mode that is active for the Warpcore port.

6.1.30. Syslog Support

Components running on an ICOS enabled platform may generate messages of interest to system administrators in understanding the state of the system and diagnosing operation issues. Messages may be generated in response to events, faults, or errors occurring on the platform, as well as changes in configuration or other occurrences. These messages can be stored locally on the platform or forwarded to one or more centralized points of collection for monitoring and long-term archival. The Local and remote configuration of the logging capability is desirable, including filtering of messages logged or forwarded based on severity and generating component.

Syslog is supported over IPv4 and IPv6 management interfaces, and IPv4 or IPv6 server hosts can be configured as Syslog destinations.

6.1.31. Source IP Address Configuration

Syslog, TACACS, SNTP, sFlow, SNMP Trap, RADIUS, and DNS Client allows the IP Stack to pick the source IP address while generating the packet. This feature provides an option for the user to select an interface for the source IP address while the management protocol transmits packets to the management stations. The source address is used for filling the IP header of management protocol packets.

The feature is available with the Routing package.

6.1.32. Factory Default Configuration File

ICOS includes a *factory-default* text configuration file that contains CLI commands and has the same format as the startup-config file. When the switch is restored to factory default settings (using the command *clear config* or some other mechanism), the factory default configuration is copied to the startup configuration.

6.1.33. Chef Client Integration

Chef is a tool that helps facilitate and automate device configuration networks that involve a large number of devices, such as data centers and cloud environments. Chef has three major components:

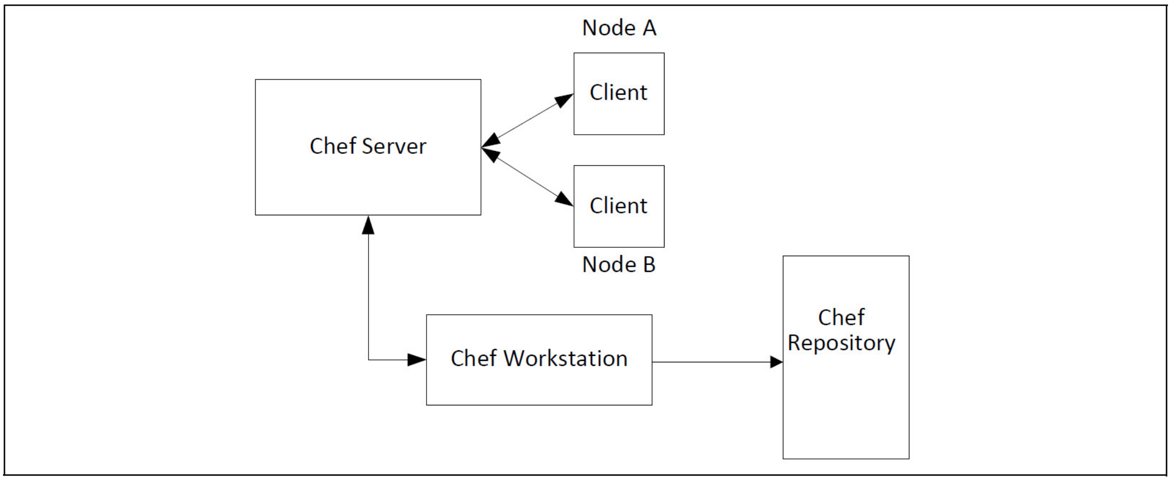
- Chef workstation — The Chef component on which the network configuration is designed. An administrator prepares configuration information on the Chef workstation and sends it to the Chef server to be further sent to the Chef client.

- Chef server — The Chef component that the Chef client interacts with to get the switch configuration information.
- Chef node — Any server or virtual server that is configured to be maintained by a Chef client. A node can be physical or cloud-based.

The Chef features in ICOS allow ICOS-based switches to be configured using Chef.

Each Chef configuration consists of one or more workstations, a single server and all the nodes that need to be configured by Chef, as the figure below shows. The Chef client is installed on all the nodes that are to be configured using Chef framework. Cookbooks and Recipes, which are written in Ruby scripting language, are used to tell the Chef client how to configure each node. The Chef workstation is where cookbooks are written to define the network and to specify how each node in the network needs to be configured. These cookbooks are maintained in a repository maintained by the workstation. To perform the actual configuration, the recipes and cookbooks are pushed from the workstation to the Chef server. At this point the recipes and roles can then be applied to specific Chef nodes. The Chef client, which runs in each node, performs the actual configuration on the node. Chef is implemented in Ruby. The recipes are also written in Ruby but can call out to other languages such as Bash, C and Python.

Figure 6.3. Chef Topology



For more information about Chef, see http://www.opscode.com/chef/

6.1.34. Puppet Client Integration

Like Chef, Puppet is a tool that makes it easier to automatically configure a large number of devices in a network environment. Puppet helps system administrators manage infrastructure throughout its lifecycle, from provisioning and configuration to patch management and compliance. Using Puppet, an administrator is able to easily automate repetitive tasks, quickly deploy critical applications, and proactively manage change, scaling from tens of servers to thousands, on-premise or in the cloud.

Puppet is a product designed to deploy system configurations. Puppet includes the following features:

- Is open-source software, based on Ruby

- Is policy based
- Runs every 30 minutes
- Is an abstraction layer between the system administrator and the system
- Can run on any UNIX operating system

The Puppet agent integrated into ICOS allows ICOS-based switches to be configured using Puppet.

Puppet uses a declarative, model-based mechanism for automation of repetitive tasks. Puppet mechanisms fall into the four stages mentioned as below:

- **Define**: The desired state of the node (agent) is defined using Puppet’s declarative configuration language.
- **Simulate**: The configuration changes are simulated before actually enforcing them.
- **Enforce**: The deployed desired state is enforced automatically, correcting any configuration drift.
- **Report**: The differences between actual and desired states and any changes made enforcing the desired state are reported.

For more information about Puppet, see the Puppet Labs website: https://puppetlabs.com/.

6.1.35. MPLS

The Multiprotocol Label Switching (MPLS) feature is targeted towards data center customers deploying ICOS-based switches in leaf-and-spine or other popular data center network topologies. These types of switches with MPLS capabilities are typically known as Provider (“P”) switches that perform the Label Switch Router (LSR) functionality. These switches support MPLS-tagged packet reception and MPLS-tagged packet transmission. The switches do not convert between MPLS and non-MPLS traffic except for stripping the last MPLS tag on transmitted packets when the label action is “last-pop”.

The MPLS label distribution is done by adding support for RFC3107 to the BGP protocol.

6.1.36. Interface Error Disable and Auto Recovery

When there is an error condition for an interface, the interface is shut down and placed in diagnostic disabled state. The error-disabled interface does not allow any traffic until it is re-enabled. The administrator can manually enable an error-disabled interface, or the administrator can enable the Auto Recovery feature. Auto Recovery re-enables the interface after the expiry of configured time interval.

6.1.37. Watchdog Services

The watchdog service provides an ICOS switch the ability recover when it is no longer executing properly. When a recovery is attempted, valuable debug information is gathered and saved before the switch is reset. This feature takes advantage of the watchdog feature available in the Linux kernel and the existing watchdog support already present in ICOS software.

6.1.38. Packet Trace

6.1.38.1. Overview

When a packet enters a system, it is processed and forwarded out on one or more destination ports. Typically, the results of intermediate packet processing steps are not visible. If the packet is dropped or not forwarded as expected, drop reason or error event counters may be incremented. However, since these are typically coarse counters, they may not provide sufficient information to debug why the packet was dropped or forwarded incorrectly.

The packet trace feature provides detailed information on how a specific packet is processed through the ingress pipeline. The feature allows the user to send a special visibility loopback packet into the Ingress Packet Processing Pipeline that is then processed as if it were received on one of the front-panel ports, so that internal forwarding and packet processing states can be logged. The internal forwarding and packet processing data retrieved for the packet as a part of the packet trace feature is called a trace profile. The trace profile contains data such as the lookup resolution results, lookup status, state of the ingress port, hashing info for the packet (i.e., LAG hash resolution, and ECMP route resolution). This information can be useful for detecting/diagnosing potential network problems.

This feature is supported on Netberg hardware platform based on Broadcom Tomahawk family of devices platforms. The egress LAG member port and ECMP route taken by the packet can be found using packet trace.

6.1.38.2. Operation

Given a specific packet, the packet trace feature can identify the egress LAG member port or the ECMP route the packet would take.

The trace packet to be injected can be specified in two ways:

- Using CLI commands that take user input for important packet fields that enable ICOS to create and inject the packet.
- Using a PCAP file that can be copied onto the system using the copy command.

Using the CLI

The Following types of packets can be specified for getting the trace profile:

- L2—Ethernet.
- L3—IPv4/IPv6.
- L4—TCP/UDP

For retrieving the trace profile for an Ethernet packet, the L2 packet fields are used to create a packet. Similarly, when an IPv4 packet’s trace profile is required, the L2 and IPv4 fields are used to create the required packet headers. The TCP/UDP port numbers can be used to create a TCP/UDP header. The ethertype and IP protocol fields are filled in appropriately for each packet type.

The hardware can trace a VLAN tagged packet only.

Using a PCAP File

PCAP is a basic format used to save captured packets from the network. PCAP file format for a pcap file with two captured packets is given below:

Global PCAP Header	Packet 1 PCAP Header	Packet 1 Data	Packet 2 PCAP Header	Packet 2 Data

A pcap file containing multiple packets can be downloaded onto the switch to be used for packet tracing. The downloaded pcap file is stored in */mnt/application* (EXEC_PATH) and, therefore, is not persistent across a reload. The pcap file must be < 1 MB in size.

The packet trace feature provides support for the following types of information:

- Summary — For all complete relevant packets parsed in the given pcap file (i.e., seen to egress on the given LAG or ECMP route), a summary of the LAG/ECMP link/route utilization is provided as a percentage. For example, out of 100 packets in the pcap file, if 20 are seen to egress on LAG member port A, 50 on B and 30 on C, in terms of percentage, link utilization for the member ports is specified as:

A: 20%

B: 50%

C: 30%

- Detailed — Detailed trace profile information (egress LAG member port, ECMP NH information) is provided for a maximum of the first 20 complete packets parsed in the pcap file. PCAP version 2.4 is supported by packet trace. Additionally, any incomplete packets, where the captured packet length < original packet length on wire (rpcapFilePktHdr.capLen < rpcapFilePktHdr.len), are skipped over. No packet validation is done for packets in a PCAP file. Each packet in the PCAP file is extracted and provided to the SDK to be injected into the silicon. A valid trace profile is provided by the SDK/hardware for a packet that went through the pipeline processing successfully. Global PCAP Header Packet 1 PCAP Header Packet 1 Data Packet 2 PCAP Header Packet 2 Data

NOTE:

- All packets in the PCAP file must be VLAN tagged, since the hardware can only trace VLAN tagged packets. No valid trace profile/data is retrieved for a non-VLAN tagged packet.
- For detailed packet info in a PCAP file, the CLI shows the following basic packet fields only:
  - Src MAC, Dst MAC, and VLAN tag.
  - Src IP/IP6, Dst IP/IP6.

6.1.39. RESTful APIs

This feature enables the OpEN API to be accessed across the network by a device that is remote to the switch. A subset of the OpEN APIs is made available and client code is provided that allows applications to be written in a similar way, independent of whether they will be executed locally (by a switch agent) or remotely.

The RESTful API feature includes support for Authentication, Authorization, and Security (AAS) access to its API library. The HTTP or HTTPS client must log in by authenticating a user ID and password. Authentication is performed using the existing User Interface Session Manager that is available to the Lighttpd web server. As such, the user id and password must match an existing entry within the ICOS user database for which a Session Identifier (SID) is generated and returned to the client. This SID must then accompany all HTTP or HTTPS client requests in the header for which server-side verification is performed.

In addition to AAS, additional supported APIs include Talac/Scaleblaze and OpEN Overlay.

RESTful API support is available for following modules:

- VLAN
- Interface
- System
- Bridge
- FDB
- LAG
- STP
- Storm Control
- IEEE 802.1ad (QinQ)
- L2 Multicast
- Mirroring
- IP (ifTable, ipAddrTable, MIB-II RFC-1213 – tables only)

6.2. Layer 2—Switching Module

This section describes the following layer 2 components:

- Section 6.2.1, “Layer 2 Architecture”
- Section 6.2.2, “Link Aggregation (IEEE 802.3ad)”
- Section 6.2.4, “Virtual Port Channels”
- Section 6.2.5, “Provisioning (IEEE 802.1p)”
- Section 6.2.6, “Virtual LAN (IEEE 802.1Q)”
- Section 6.2.7, “Switchport Modes”
- Section 6.2.8, “Double VLAN (DVLAN) Tunneling”
- Section 6.2.9, “RADIUS-Based Dynamic VLAN Assignment”
- Section 6.2.10, “Private VLANs”
- Section 6.2.11, “Forwarding, Aging, Learning”
- Section 6.2.12, “Independent VLAN Learning”
- Section 6.2.13, “Static MAC Filtering”
- Section 6.2.14, “Jumbo Frames Technology”
- Section 6.2.15, “IGMP Snooping”
- Section 6.2.16, “MLD Snooping”
- Section 6.2.17, “IGMP and MLD Snooping Querier”
- Section 6.2.18, “Protected Ports”
- Section 6.2.19, “Multicast VLAN Registration”
- Section 6.2.20, “Multiple Spanning Tree Protocol”
- Section 6.2.21, “Per-VLAN Spanning Tree Protocol and Per-VLAN Rapid Spanning Tree Protocol”
- Section 6.2.22, “Port-Based Network Access Control (IEEE 802.1X)”
- Section 6.2.23, “Port Mirroring”
- Section 6.2.24, “Internal 802.1X Authentication Server”
- Section 6.2.25, “MAC-Based 802.1X”

- Section 6.2.26, “802.1X Monitor Mode”
- Section 6.2.27, “Dot1X Supplicant”
- Section 6.2.28, “Guest VLAN”
- Section 6.2.29, “Unauthenticated VLAN”
- Section 6.2.30, “Multiuser VLAN Assignment and Filter ID Support”
- Section 6.2.31, “LLDP and LLDP-MED”
- Section 6.2.32, “DHCP Snooping and DHCPv6 Snooping”
- Section 6.2.33, “IP Source Guard”
- Section 6.2.34, “Dynamic ARP Inspection”
- Section 6.2.35, “DHCP L2 Relay”
- Section 6.2.36, “Storm Control”
- Section 6.2.37, “Port Security”
- Section 6.2.38, “Unidirectional Link Detection (UDLD)”
- Section 6.2.39, “Link Dependency”
- Section 6.2.40, “IPv6 RA Guard”
- Section 6.2.41, “Management and Control Plane ACLs”
- Section 6.2.42, “OpenFlow”

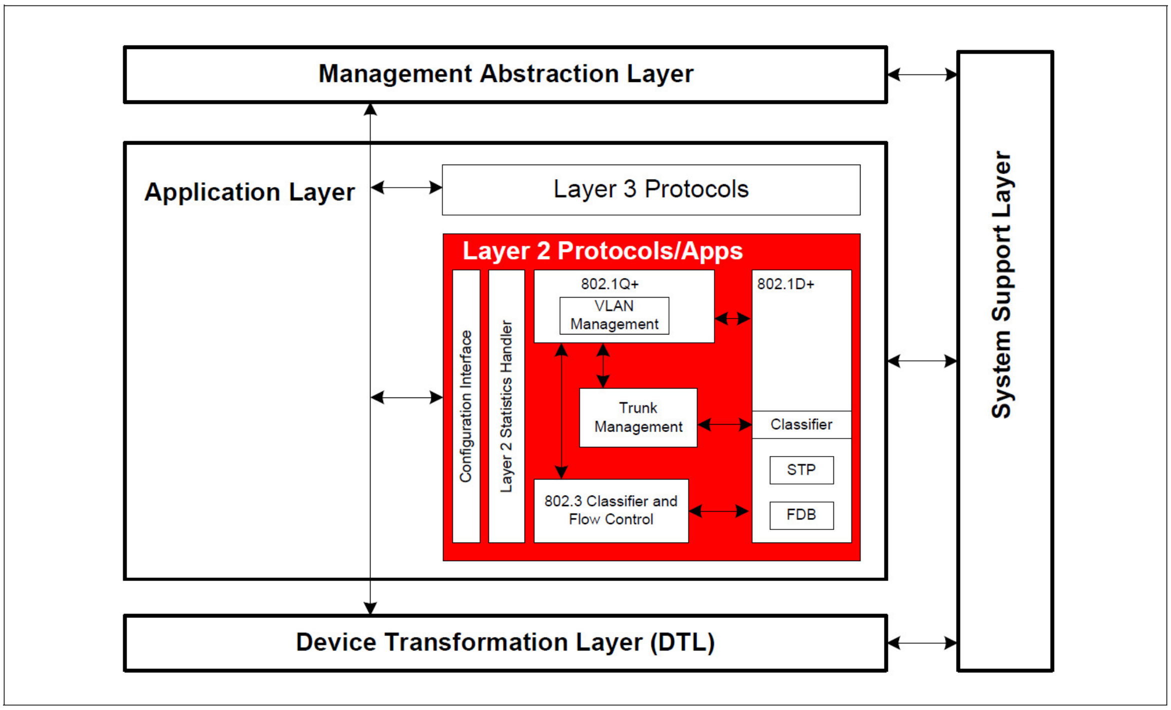
6.2.1. Layer 2 Architecture

The switching module provides the base layer 2 support for LAN (Local Area Network) and WAN (Wide Area Network) environments and is the foundation of the ICOS software family. The ICOS software family allows you to build a complete Ethernet layer 2 product with advanced functionality.

To accomplish its layer 2 base role, the switching module integrates a set of customizable components that enable programmers to define networks to match specific requirements and to manage them securely and efficiently. The switching module consists of Ethernet layer 2 applications and protocols such as Port Mirroring, LAG, 802.1Q VLAN, 802.1p, and others as described in this document. The switching module also consists of a number of management protocols, including BOOTP and XMODEM.

Some of the components within the switching module architecture are shown in the figure below.

Figure 6.4. ICOS Layer 2 Architecture



This section presents an overview of the Switching protocols/applications.

6.2.2. Link Aggregation (IEEE 802.3ad)

Trunking, which is also called Port Channels or Link Aggregation, is initiated and maintained by the periodic exchanges of Link Aggregation Control PDUs (LACPDUs). When LACP is enabled for a physical interface, LACPDUs must not be dropped for any reason. Conversely, when LACP is disabled for the physical interface LACPDUs must be dropped.

From a system perspective, a LAG is treated as a physical port. A LAG and a physical port use the same configuration parameters for administrative enable/disable, port priority, and path cost.

A LAG failure of one or more of the links does not stop traffic in any manner. Upon failure, the flows mapped to a link are dynamically reassigned to the remaining links of the LAG. Similarly when links are added to a LAG, the conversations may need to be shifted to a new link member. Before any relocation of a conversation, the system ensures reordered frames do not exist.

The LAG component supports a feature whereby a LAG acquires ports upon create. The ports are placed in the discarding state in hardware and transmit only LACPDUs. Acquired ports are not available for configuration. The number of ports that can be members of a single LAG is platform-dependant. Some platforms can support up to 32 ports in a LAG.

6.2.2.1. Static LAGs

A static LAG is no different from a dynamically configured LAG. All the requirements for the member ports hold true. [member ports must be physical, same speed, and so on] The only difference is this LAG has an additional parameter *static* which makes this LAG not require a partner system to be able to aggregate it's member ports.

A static LAG does not transmit or process received LACPDUs, that is, the member ports do not transmit LACPDUs and all the LACPDUs it may receive are dropped. A dropped counter is maintained to count the number of such PDUs.

Configured members are added to the LAG (active participation) immediately if the LAG is configured to be static. There is no wait time before we add the port to the LAG.

A LAG can be either static or dynamic not both. It cannot have some members participate in the protocol while other members not participate.

6.2.2.2. LAG Member Flap Counters

ICOS supports counters to track LAG member flaps and LAG flaps. The counter for a LAG member is incremented when the physical port is either shut down by the administrator or when its link state is down. The member flap counters are not incremented when the LAG is administratively shut down. When all active member ports are INACTIVE (i.e. either admin down or link down), then the counter that tracks LAG flaps is incremented.

The LAG member flap counter is incremented in the following cases

- When a LAG member is admin shutdown
- When a LAG members link is down

The member flap counters are reset to zero when a port is removed and re-added as LAG member ports.

The LAG interface flap counter is incremented in the following cases

- When the LAG interface is admin shutdown
- When the LAG interface has no ACTIVE members sufficient to meet the minimum links criteria.

Counters are not incremented if the LAG interface is previously shutdown. The LAG flap counters are maintained across a management failover. They are not retained across reboots.

6.2.2.3. LAG Interaction with Other Features

From a system perspective, a LAG is treated just as a physical port, with the same configuration parameters for administrative enable/disable, spanning tree port priority, and path cost as any other physical port.

LAGs and VLANs

When members are added to a LAG, they are removed from all existing VLAN membership. When members are removed from a LAG, the members rejoin the VLANs that they were previously members of as per the configuration file.

The LAG interface can be a member of a VLAN complying with IEEE 802.1Q.

LAGs and STP

Spanning tree does not maintain state for members of a LAG. Though the Spanning Tree does maintain state for the LAG interface. As far as STP is concerned members of a LAG do not exist. (Internally, the STP state of the LAG interface is replicated for the member links.)

When members are deleted from a LAG they become normal links and spanning tree maintains their state information.

Statistics

Statistics are maintained for all LAG interfaces.

6.2.2.4. LAG Hashing Algorithm

The purpose of link aggregation is to increase bandwidth between two switches. It is achieved by aggregating multiple ports in one logical group. A common problem of port channels is the possibility of changing packets order in particular TCP session. The resolution of this problem is correct selection of physical port within port channel for transmitting the packet to keep original packets order.

To perform this selection, a hash function is used. This function can take several packet fields as input, and produce a specific destination port number within a port channel as output. It is possible to use several hashing functions to improve bandwidth usage efficiently.

The hashing algorithm is configurable for each LAG. The types of LAG algorithms available may vary depending upon platform capabilities. Typically, an administrator is able to choose from hash algorithms utilizing the following attributes of a packet to determine the outgoing port:

- Source MAC, VLAN, EtherType, and incoming port associated with the packet.
- Source IP and Source TCP/UDP fields of the packet.
- Destination MAC, VLAN, EtherType, and incoming port associated with the packet.
- Source MAC, Destination MAC, VLAN, EtherType, and incoming port associated with the packet.
- Destination IP and Destination TCP/UDP Port fields of the packet.
- Source/Destination MAC, VLAN, EtherType, and incoming port associated with the packet.
- Source/Destination IP and Source/Destination TCP/UDP Port fields of the packet.

On Netberg hardware platforms based on Broadcom Tomahawk, the LAG hashing support is extended to Enhanced hashing mode, which provides improved load-balancing performance. LAG hashing on these platforms has the following features:

- MODULO-N operation based on the number of ports in the LAG, where N is the number of active link members in the LAG.
- Packet attributes selection based on the packet type. For L2 packets, Source and Destination MAC addresses are used for hash computation. For IP packets, Source IP, Destination IP address, and TCP/ UDP ports are used.

- Non-Unicast traffic and Unicast traffic is hashed using a common hash algorithm.

6.2.2.5. Hash Prediction with ECMP and LAG

The Hash Prediction feature provides a utility to predict how packets will be forwarded over a Link Aggregation Group (LAG) or to the next-hop device when Equal-Cost Multipath (ECMP) is the destination. Given the link aggregation method, ingress physical port, and values of various packet fields, the utility predicts an egress physical port for the packet.

An ECMP group is identified by the IP address of one of its members. By entering the IP address in the form <prefix/prefix-length>, the utility predicts the packet's physical egress port based on the destination ECMP group. To predict the an egress physical port when the egress objects are VLAN routing interfaces with LAG or port interfaces as members of the VLANs, the utility requires the PVID to be configured on the interfaces and the next hops to be fully installed in hardware.

To make correct prediction when LAGs are used as egress interfaces, the utility requires the enhanced hashing mode to be set on the LAGs.

Hash prediction is supported only for unicast packets on Netberg hardware platform based on Broadcom Tomahawk family of devices-based platforms.

6.2.2.6. Manual Aggregation

ICOS software supports manual addition and deletion of links to aggregates.

In manual configuration of aggregates, the ports send their Actor Information (LACPDUs) to the partner system in order to find a suitable Partner to form an aggregation. When the Partner System neglects to respond using LACPDUs, the ICOS software aggregates manually. The ICOS software uses the currently configured default Partner Values for Partner Information.

Care must be taken while enabling this type of configuration. If the Partner System is not 802.3ad-compliant or the Link Aggregation Control protocol is not enabled, there may be network instability. Network instability occurs when one side assumes that the members in an aggregation are one single link, while the other side is oblivious to this aggregation and continues to treat the *members* as individual links.

In the ICOS software system, the Actor System waits for 3 seconds before aggregating manually. The 3-second wait time is specified by the protocol.

If a manual LAG member sees an LACPDU that contains information different from the currently configured default partner values, that particular member drops out of the LAG. This configured member does not aggregate with the LAG until all the other active members see the new information. When each of the other active members sees the new information, they continue to drop out of the LAG. When all the members have dropped out of the LAG, they form an aggregate with the new information.

Manual Configuration on a Runtime Basis

ICOS supports the Manual configuration of LAGs on a runtime basis. This setting does not affect the dynamic working of the protocol; therefore, if a partner exists, it aggregates dynamically. This is a per system configuration option, applicable on all LAG interfaces. The default setting is Disable. Therefore, by default, the LAG does not aggregate until a suitable partner is found to form an aggregation. The following are two scenarios with a manual setting.

CASE 1: Manual Disabled to Enabled

All the LAGs that have configured members but have no active members now proceed to aggregate statically (on linked up interfaces only). There is no effect on dynamic LAGs.

CASE 2: Manual Enabled to Disabled

Active members of LAGs that are statically maintained drop out of the LAG, and the LAG goes down with no active members. There is no effect on dynamic LAGs.

6.2.3. Dynamic LAG Limits

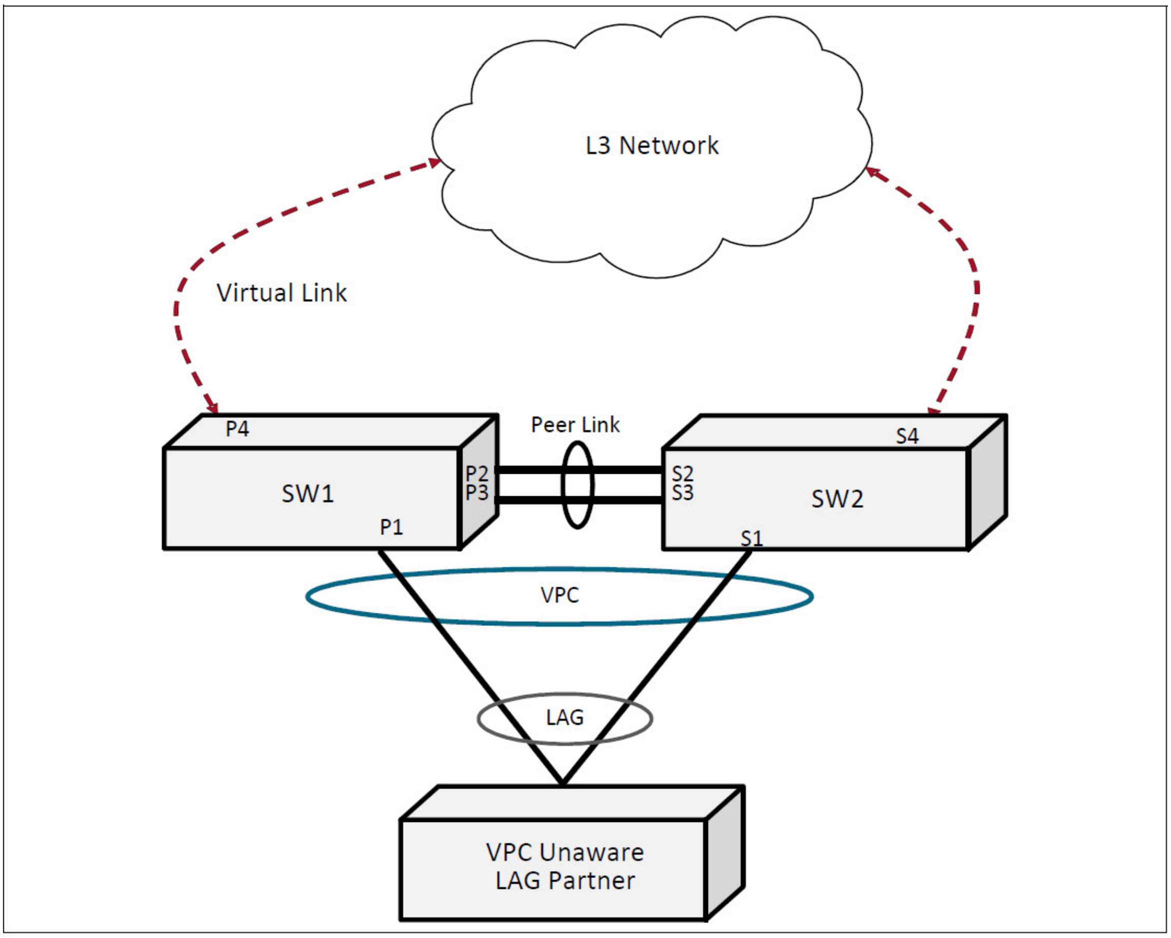
The Dynamic LAG Limits feature allows you to make the number of dynamic ports the limiting factor, rather than number of dynamic LAG interfaces. The maximum number of dynamic ports per LAG is as many LAGs as the user wants to create. In other words, instead of having a limit for 8 dynamic LAGs with eight ports each, there is a restriction for (8 × 8 = 64) ports that can be configured as members of a dynamic LAG, and the user can choose to configure 8 × 8, 32 × 2, or any other combination.

6.2.4. Virtual Port Channels

Like standard LAGs, virtual port channels (VPCs) allow one or more Ethernet links to be aggregated together to increase speed and provide redundancy. VPCs are also known as multichassis or multiswitch link aggregation groups (MLAGs). With standard LAGs, the aggregated links must be on the same physical device, but VPCs do not share that requirement. The VPC feature allows links on two different switches to pair with links on a partner device. The partner device is unaware that it is pairing with two different devices to form a port channel.

The figure below shows an example of a network topology that uses VPCs. SW1 and SW2 are VPC switches. Together, they form one end of the LAG. The VPC unaware device is the LAG partner that forms the other end of the LAG.

Figure 6.5. VPC Topology



The components in the figure above are as follows:

- **VPC switches**: VPC-aware switches running ICOS. No more than two VPC-aware switches can pair to form one end of the LAG. In the figure 14, SW1 and SW2 are VPC peer switches. These two switches form a logical end point for the VPC.

- **VPC interfaces**: Port channels (LAGs) on VPC-aware switches that are configured as VPCs. VPC functionality is a property of port channels. The administrator can configure multiple instances of VPC interfaces on the two VPC switches. ICOS supports 64 instances of VPC interfaces. Port-channel functionality like min-links and the maximum number of ports supported per LAG applies to VPC interfaces too.

- **VPC member ports**: Ports on both VPC switches that are part of the VPC interface (P1 on SW1 and S1 on SW2)

- **Non redundant ports**: (Non-VPC) ports on either switches that are not part of the VPC (ports P4 and S4). VLANs cannot be shared between VPC interfaces and non-redundant ports.

- **VPC peer link**: A link between the two VPC peer switches (ports P2,P3,S2,S3). This link is used to carry:

- Keep-alive packets between the two peer switches. The keep-alive protocol is used to elect the Primary and the Secondary switch.
- PDUs and LACPDUs between the Secondary and Primary VPC devices.
- Control messages such as VPC member port related events, FDB/MFDB entries, configuration details.

In addition to supporting best-effort delivery of these messages, ICOS supports both Synchronous Reliable Control Message Delivery and Asynchronous Reliable Control Message Delivery mechanisms. If the message is received by the peer, it sends an ACK back to the calling VPC component to acknowledge receipt of the message. The VPC application will try three times every second until an ACK is received. If an ACK is not received at the end of all the attempts, an error is reported back to the calling component.

- Data traffic over the peer's VPC member ports, when the member ports of the VPC interface are all down on this device.

ICOS supports peer failure fast detection, which enables detecting a peer failure in 1 second when DCPDP is enabled.

Only one peer-link can be configured per device. The peer-link is crucial for the operation of the VPC component. A port channel must be configured as the peer-link. VLANs configured on VPC interfaces must be configured on the peer-link as well.

- **VPC Dual Control Plane Detection link**: A virtual link that is used to advertise the dual control plane detection protocol (DCPDP) packets between the two VPC switches (ports P4, S4). This protocol is optional. The protocol indicates the presence of the peer switch in the network. The DCPDP protocol should not be configured on VPC interfaces. The DCPDP packet transmission interval and reception timeout duration are configurable.

The two switches that form the VPC pair (primary/secondary) must support the same set of features and have equal capacity (for example, number of FDB entries supported and number of ports). Additionally, the peer switches that form the VPC pair must run the same firmware version.

6.2.4.1. Fast-Failover

ICOS supports configuration of a unique VPC MAC address and VPC system priority, which enable fastfailover in the event a primary switch failure. The secondary switch advertises these VPC values—instead of the switch MAC address and system priority— in LACP PDUs sent on VPC member ports. The secondary switch also uses the VPC MAC address in the designated bridge ID field in STP BPDUs sent on VPC member ports. For nonredundant ports, the secondary switch uses the switch MAC address and system priority values in LACPDUs and STP BPDUs.

In the case of a primary switch failure, traffic disruption is limited to the time required for the partner devices that are dual-attached to the MLAG domain to detect the link down on the primary device and redistribute the traffic using the links connected to the secondary device.

6.2.4.2. VPC Limitations

This ICOS implementation of the VPC feature is proprietary and is not based on any standards. The following constraints apply:

- ICOS VPC does not interoperate with other vendor implementations.
- Peer switches must run the same version of ICOS.
- The upgrade scenario is minimally disruptive (not hitless). The reconvergence equivalent to spanning-tree.
- Link failover has a momentary packet loss due to a brief LAG flap on VPC partners.
- If a primary switch failure occurs, the reconvergence is equivalent to spanning-tree.
- The VPC port channel controlled on primary switch only.
- VRRP is supported only at the edge.
- The administrator must configure the VPC (VPC) peers identically.

6.2.4.3. VPC Interaction with Other Components

Table below shows the list of switching components that are supported on an VPC interface. Protocols such as UDLD and LLDP that are enabled only on LAG member ports and not on the LAG interface can be enabled on VPC member ports. The administrator must ensure that the configuration is the same on both the devices for consistency in handling traffic.

Table 6.2. Switching Features Supported on VPC Interfaces

Component	Supported	Component	Supported
DOT1Q	Yes	MRP	No
802.1p	No	MMRP	No
Unauthenticated VLAN	No	ACL	N/A
Guest VLAN	No	DiffServ	N/A
MAC Authentication Bypass	No	CoS	N/A
Broadcast Storm Recovery	No	ACL Logging	N/A
802.3ad	Yes	Flow-based port mirroring	N/A
LAG Hashing	Yes	iSCSI	No
Port Mirroring	N/A	DOT1AD	No
MAC Filtering	N/A	DCBX	No
MFDB	No	ETS	No
IGMP/MLD Snooping	No	QCN	No
802.1Qbb	No	FIP Snooping	No
802.1s (STP, RSTP, MSTP, PVSTP, PVRSTP)	Yes	MVRP	No
Loop Guard	No	Management ACL	No
FDB	Yes	UDLD	N/A
Port MAC locking	No	Private VLAN	No
DVLAN	No	LLPF	No
802.1AB	No	Port Aggregator	No

Component	Supported	Component	Supported
Protected Port	No	MVR	No
DHCP Snooping	No	Class-Based VLAN	No
IP Source Guard	No	DHCP Filtering	No
Dynamic ARP Inspection	No	EASY_ACL	No
Auto-negotiation	N/A	Media VLAN	No
L2 Relay	No	VLAN-Rate Limit	No

6.2.5. Provisioning (IEEE 802.1p)

The 802.1p classification is carried within the 802.1Q VLAN tag. This tag, when present, is part of the layer 2 header carried within frames. The 802.1p protocol defines a total of eight priority levels and class of service queues. The actual number of class of service queues available is device-dependent.

6.2.5.1. Configurable Priority Queues

The ICOS software administrator can change the default 802.1p priority assigned for untagged packets on a per-port basis. The administrator is able to reassign 802.1p Priority Mapping to any of the eight class of service queues in the device.

6.2.6. Virtual LAN (IEEE 802.1Q)

802.1Q VLAN is an implementation of the Virtual Local Area Network, specification 802.1Q. Operating at layer 2 of the OSI model, the VLAN is a means of parsing a single network into logical user groups or organizations as if they physically resided on a dedicated LAN segment of their own. In reality, this virtually defined community may have individual members scattered across a large, extended LAN. The VLAN identifier is part of the 802.1Q tag, which is added to an Ethernet frame by an 802.1Q-compliant switch. Devices recognizing 802.1Q-tagged frames maintain appropriate tables to track VLANs. The first 3 bits of the 802.1Q tag are used by 802.1p to establish priority for the packet.

ICOS supports 802.1Q VLANs. As such, ports may simultaneously belong to multiple VLANs. VLANs allow a network to be logically segmented without regard to the physical locations of devices in the network. The 802.1Q VLAN bridging functionality is incorporated in the ICOS software.

ICOS switching supports up to 4094 VLANs for forwarding. It supports 4096 if the NULL VLAN (VLAN ID 0) and the Administrative VLAN (VLAN ID 4095), both specified in IEEE 802.1Q, are counted.

VLANs can be allocated by subnet and netmask pairs, thus allowing overlapping subnets. For example, subnet 10.10.128.0 with Mask 255.255.128.0 and subnet 10.10.0.0 with Mask 255.255.0.0 can have different VLAN associations.

6.2.7. Switchport Modes

In addition to configuring port VLAN memberships and tagging behavior using legacy CLI commands, the administrator can configure a switch port as a trunk port or access port. The switchport mode feature helps to minimize the potential for configuration errors and makes VLAN configuration easier by reducing the amount of commands needed for port configuration. For example, to configure a port connected to an end user, the administrator can configure the port in Access mode. Ports connected to other switches can be configured in trunk mode. VLAN assignments and tagging behavior are automatically configured as appropriate for the connection type.

A third switchport mode, general mode, provides no configuration restrictions and allows the administrator to configure the port with legacy VLAN commands.

6.2.8. Double VLAN (DVLAN) Tunneling

The use of an additional tag on VLAN traffic differentiates between customers in a Metropolitan Area Network (MAN) while preserving the VLAN identification of the customer's domain. With the tunneling feature, service providers can use a single VLAN to support customers who have multiple VLANs. The customer's domain VLAN IDs are preserved and traffic from various customers is segregated within the infrastructure, even though they appear to be members of one VLAN. The tunneling feature expands VLAN space by using a VLAN-within-a-VLAN scheme and adding a tag to the tagged packets.

Double VLAN Tunneling is a feature developed for service providers who transport traffic of multiple customers and are required to maintain the VLAN and layer 2 protocol configurations of each customer. The tunneling feature provides connectivity on a shared infrastructure with the same characteristics of a private network.

A second tag, which increases the packet size by another 4 bytes, is added to differentiate customers in the MAN core. The second tag format duplicates the format of the 802.1Q VLAN tag, hence the name Double VLAN Tagging (Nested/Stacked VLAN Tagging).

The 4-byte tag consists of the following:

- 2 bytes for the EtherType
- 4 bits for the priority and CFI bit
- 12 bits for a customer ID

The configuration of the particular parts is platform dependent. The ICOS software provides per-port configuration options for the EtherType and the customer ID. The DVLAN Tag priority and CFI bit(s) are not configurable using the ICOS software management component.

An incoming frame is identified as tagged or untagged based on Tag Protocol Identifier (TPID) value it contains. The 802.1Q standard specifies a TPID value (0x8100) to recognize an incoming frame as tagged or untagged. Any valid Ethernet frame with a value 0x8100 in the 12th and 13th bytes is recognized as tagged frame. 802.1Q switches check the 12th and 13th bytes to decide the tag status of incoming frame.

ICOS software can be configured to enable the port in double-VLAN (DVLAN) mode. In this mode switch looks for 12th, 13th, 16th, and 17th bytes for the tag status in the incoming frame. The outer tag (S-TAG) TPID is identified with the 12th and 13th bytes values. The inner tag (C-TAG) TPID is identified with 16th and 17th bytes values. These two TPID values can be different or the same. For Metro Ethernet frames, the default S-TAG TPID is 0x88a8 and the default C-TAG TPID is 0x8100. Big-endian byte order is used. VLAN normalization, source MAC learning, and forwarding are based on the S-TAG value in a received frame.

ICOS supports configuring one outer VLAN TPID value per switch. The ICOS DVLAN solution supports configuring multiple TPID values per switch and mapping them to the ports as needed by the Administrator. This allows administrators to configure same or different TPIDs for different ports. This feature depends on the underlying switching ASIC. The global and port default TPID is 0x8100.

6.2.9. RADIUS-Based Dynamic VLAN Assignment

The software can dynamically create VLANs in the system when the VLANs assigned by RADIUS servers for 802.1X-authenticated clients do not exist in the system. This is supported only for VLAN IDs (numbers)—not for RADIUS server VLAN names.

6.2.10. Private VLANs

The Private VLANs feature separates a regular VLAN domain into two or more subdomains. Each subdomain is defined (or represented) by a primary VLAN and a secondary VLAN. A private VLAN can have multiple VLAN pairs with each pair identifying a subdomain.

6.2.11. Forwarding, Aging, Learning

Forwarding, Aging, and Learning are considered to be one component with three related functions. Those functions are summarized as follows:

6.2.11.1. Forwarding

Forwarding occurs when a frame is processed completely by either the bridge function or the routing function. At layer 2, frames are forwarded according to their MAC address type, which is either unicast or multicast. A unicast frame is forwarded in accordance with the address entry in the networking device’s address tables or filtering database. The frame is forwarded to the port associated with the unicast address in the address entry. Multicast frames are forwarded in accordance with their address entry in the switch filtering database. These entries are created as a result of permanent configuration, static destination filters, or MMRP registration. If no entry exists, the frame is forwarded to all ports in the associated VLAN.

6.2.11.2. Aging

An address aging time-out parameter based on the 802.1D specification is included in the networking device. This parameter is a persistent input and output parameter and represents time in seconds. Aging is controlled by microcode. An address aging time-out parameter is user configurable. The parameter minimum and maximum values may be bounded by networking device implementation.

6.2.11.3. Learning

Learning and managing MAC addresses is according to the IEEE 802-1D-1998 and 802-1Q-1998 standards. The ICOS software supports Independent VLAN Learning (IVL). All learning takes place in the underlying device. The forwarding database is populated from messages received from the platform.

6.2.12. Independent VLAN Learning

Independent VLAN Learning (IVL) allows unicast address-to-port mappings to be created based on a MAC Address in conjunction with a VLAN ID.

This arrangement associates the MAC Address only with the VLAN on which the frame was received. Therefore, frames are forwarded based on their unicast destination address as well as their VLAN membership. This configuration affords multiple occurrences of an address in the forwarding database. Each address associates with a unique VLAN. Care must be taken in the administration of networks, as multiple instances of a MAC address, each on a different VLAN, can quickly eat up address entries.

Each VLAN is associated with its own forwarding database. Hence the number of forwarding databases equals the number of VLANs supported.

The MAC address stored is supplemented by a 2-byte VLAN ID. The first 2 bytes of a forwarding database entry contain the VLAN ID associated, and the next 6 bytes contain the MAC address. There is a one-to-one relationship between VLAN ID and FID (forwarding database ID).

6.2.13. Static MAC Filtering

Static MAC Filtering allows you to add a number of unicast or multicast MAC addresses directly to the forwarding database. This is typically a small number relative to the total size of the database. Associated with each static MAC address is a set of source ports, a set of destination ports and VLAN information.

Any packet with a particular static MAC address in a particular VLAN is admitted only if the ingress port is in the set of source ports; otherwise, the packet is dropped. On the egress side, the packet, if admitted, is sent out of all the ports that are in the set of destination ports.

Upon ingress, each packet’s destination MAC address is compared against the forwarding database. If the address is not in the table, the code behaves as currently designed. If the address is in the table, then it is checked to see if it has been defined as a filter. If the MAC address is not defined as a filter, then the code continues as currently designed.

If the specific destination MAC address is defined as a filter, then the ingress port number is compared to the set of source ports listed for the address. If the port of ingress is not in the set of source ports, then the packet is immediately discarded. If the ingress port is a member of the set of source ports, then the packet is admitted.

For packets admitted because of a MAC filter match only, the following additional steps are performed. Note that all other egress processing remains unchanged.

At the egress port, if the destination port number is in the set of destination ports, the packet is transmitted. If the destination port is not in the set of destination ports, then the packet is discarded.

Static entries are never aged and can only be removed by user command.

6.2.14. Jumbo Frames Technology

The Jumbo Frames technology is employed in certain situations to reduce the task load on a server CPU and to transmit large amounts of data efficiently. The Jumbo Frames Technology predominantly appears where certain applications would benefit from using a larger frame size (for example, Network File System (NFS).

The larger frame size eliminates some of the need for fragmentation, leading to greater throughput. The increase in throughput is particularly valuable on data center servers where the larger frame size increases efficiency of the system and allows processing of more requests.

The ICOS software Jumbo Frames feature extends the standard ethernet MTU (Max Frame Size) from 1518 (1522 with VLAN header) bytes to 12288 bytes. The maximum Ethernet Frame Size is a platform dependent value. This feature is configurable through the user interface.

ICOS software assumes that all packets are in Ethernet format.

The Jumbo Frames technology is currently not defined by a standard. However, any device connecting to the same broadcast domain should support the same MTU.

6.2.15. IGMP Snooping

Internet Group Management Protocol (IGMP) Snooping is a feature that allows a switch to forward multicast traffic intelligently on the switch. Multicast IP traffic is traffic that is destined to a host group. Host groups are identified by class D IP addresses, which range from 224.0.0.0 to 239.255.255.255. Based on the IGMP query and report messages, the switch forwards traffic only to the ports that request the multicast traffic. This prevents the switch from broadcasting the traffic to all ports and possibly affecting network performance.

Special attention should be brought to the IP address range 224.0.0.1 through 224.0.0.255, which is reserved for routing protocols and other low-level topology discovery or maintenance protocols. For example, the address 224.0.0.1 is the all hosts address, and 224.0.0.2 indicates all routers on this subnet.

A traditional Ethernet network may be separated into different network segments to prevent overloading of the shared media. Bridges and switches connect these segments. When a packet with a broadcast or multicast destination address is received, the switch forwards a copy into each of the remaining network segments in accordance with the IEEE MAC Bridge standard. Eventually, the packet is made accessible to all nodes connected to the network.

This approach works well for broadcast packets that are intended to be seen or processed by all connected nodes. In the case of multicast packets, however, this approach could lead to less efficient use of network bandwidth, particularly when the packet is intended for only a small number of nodes. Packets are flooded into network segments where no node has any interest in receiving the packet. While nodes rarely incur any processing overhead to filter packets addressed to unrequested group addresses, they are unable to transmit new packets onto the shared media for the period of time that the multicast packet is flooded. The problem of wasting bandwidth is even worse when the LAN segment is not shared, for example, in full-duplex links.

Allowing switches to snoop IGMP packets is a creative effort to solve this problem. The switch uses the information in the IGMP packets as they are being forwarded throughout the network to determine which segments should receive packets directed to the group address.

6.2.15.1. Source Specific Multicasting (SSM)

The information in this section applies to both IGMP snooping and MLD snooping.

In addition to the standard IGMP/MLD mechanisms to solicit multicast streams from a multicast router, IGMP Version 3 (MLD Version 2) adds support for *source filtering*. This mechanism provides the ability for a host to report interest in receiving a particular multicast stream only from among a set of specific source addresses, or its interest in receiving a multicast stream from any source other than a set of specific source addresses.

ICOS supports the ability to parse the IGMPv3/MLDv2 packet for Source Specific Multicasting information and honor the host’s requests for SSM.

6.2.15.2. Control Packet Flooding

The information in this section applies to both IGMP snooping and MLD snooping.

RFC 4541 Section 2.1.2 mandates that all non-IGMP packets destined to the 224.0.0.x Multicast IP Address **must** be forwarded to *all* the VLAN members.

ICOS MGMD snooping floods multicast packets with DIP=224.0.0.x to *all* members of the incoming VLAN irrespective of the configured filtering behavior.

Support for this feature to flood packets with DIP=224.0.0.x irrespective of the entries in the L2 Multicast Forwarding Tables is platform-dependent. In platforms that do not have the required hardware capability, two ACLs (one for IPv4 and another for IPv6) are consumed in the switching silicon to accomplish the flooding using software. The software flooding workaround in such systems may result in degraded performance due to heavy CPU usage or DoS attack scenarios leading to some higher layer protocols not functioning properly (due to missed/dropped control packets).

The feature is applicable to IPv6 as well when MLD Snooping is enabled, where the corresponding IPv6 Destination IP Address is FF0X:0:0:0:0:0:0:0.

6.2.15.3. Flooding to mRouter Ports

The information in this section applies to both IGMP snooping and MLD snooping.

RFC 4541 Section 2.1.2 mandates that all unregistered multicast data streams **must** be forwarded on *all* mRouter ports for the particular VLAN.

ICOS supports the flooding of unregistered multicast streams to all mRouter ports in the VLAN irrespective of the configured filtering behavior.

The mRouter flooding feature depends on the ability of the underlying switching silicon to flood packets to specific ports in the incoming VLAN when there are no entries in the L2 Multicast Forwarding Tables for the specific stream. In platforms that do not have the required hardware capability, incoming multicast streams will always be flooded in the ingress VLAN when there is a L2MC-MISS in the switching silicon. In other words, the functionality to filter unregistered multicast packets cannot be enabled in unsupported hardware.

6.2.15.4. IGMP Snooping Operation in the Network—Multicast Forwarding Table

IGMP snooping switches build forwarding lists by monitoring for, and in some cases intercepting, IGMP messages. Although the software processing the IGMP messages could maintain state information based on the full IP group addresses, the forwarding tables in ICOS are mapped to link layer addresses.

The Multicast Forwarding Database (MFDB) manages the forwarding address table for layer 2 multicast protocols, such as IGMP Snooping.

The IGMP Snooping code in the CPU ages out IGMP entries in the MFDB. If a report for a particular group on a particular interface is not received within a certain time interval (query interval), the IGMP Snooping code deletes that interface from the group. The value for query interval time is configurable using management.

If an IGMP Leave Group message is received on an interface, the IGMP Snooping code sends a query on that interface and wait a specified length of time (maximum response time). If no response is received within that time, that interface is removed from the group. The value for maximum response time is configurable using management.

Because only the least significant 23 bits of the IP address is mapped to Ethernet addresses [RFC1112], there is a loss of information when forwarding solely on the destination MAC address. This means that, for example, 225.0.0.123 and 239.128.0.123 and similar IP multicast addresses all map to MAC address 01-00-5e-00-00-7b (for Ethernet). As a consequence, IGMP snooping switches may collapse IP multicast group memberships into a single Ethernet multicast membership group.

In addition to building and maintaining lists of multicast group memberships, the snooping switch also maintains a list of multicast routers. When forwarding multicast packets, they should be forwarded on ports that have joined using IGMP and also on ports on which multicast routers are attached. The reason for this is that in IGMP there is only one active query mechanism. This means that all other routers on the network are suppressed and thus not detectable by the switch. If a query is not received on an interface within a specified length of time (multicast router present expiration time), that interface is removed from the list of interfaces with multicast routers attached. The multicast router present expiration time is configurable using management. The default value for the multicast router expiration time is zero, which indicates an infinite time-out (that is, no expiration).

The IGMP application supports the following:

- Global configuration or per interface configuration. Per VLAN configuration is not supported in the IGMP snooping application.
- Validation of the IP header checksum (as well as the IGMP header checksum) and discarding of the frame upon checksum error.
- Maintenance of the forwarding table entries based on the MAC address versus the IP address.
- Flooding of unregistered multicast data packets to all ports in the VLAN.

6.2.16. MLD Snooping

In IPv6, Multicast Listener Discover (MLD) snooping performs functions similar to IGMP snooping in IPv4. With MLD snooping, IPv6 multicast data is selectively forwarded to a list of ports that want to receive the data, instead of being flooded to all ports in a VLAN. This list is constructed by snooping IPv6 multicast control packets.

MLD is a protocol used by IPv6 multicast routers to discover the presence of multicast listeners (nodes wishing to receive IPv6 multicast packets) on its directly attached links and to discover which multicast packets are of interest to neighboring nodes. MLD is derived from IGMP. MLD version 1 (MLDv1) is equivalent to IGMPv2. MLD version 2 (MLDv2) is equivalent to IGMPv3. MLD is a subprotocol of Internet Control Message Protocol version 6 (ICMPv6), and MLD messages are a subset of ICMPv6 messages, identified in IPv6 packets by a preceding Next Header value of 58.

The switch can snoop on both MLDv1 and MLDv2 protocol packets and bridge IPv6 multicast data based on destination IPv6 Multicast MAC Addresses. The switch can be configured to perform MLD Snooping and IGMP Snooping simultaneously.

The implementation is compliant to RFC 4541.

6.2.16.1. Operation in the Network

Routers and hosts use MLD messaging to start, manage, and stop multicast services. A host sends an MLD report (join-group) message to request multicast services. If the group is active, the router forwards the desired multicast group packets to the interface that the join request was detected on. The complementary message to join is the MLD Done message. A host sends a message to leave a group when it no longer desires to receive the multicast services of that specific group. A multicast service may not be localized to the router. The router maintains multicast route tables of groups it has learned about and learns about groups from routing protocols.

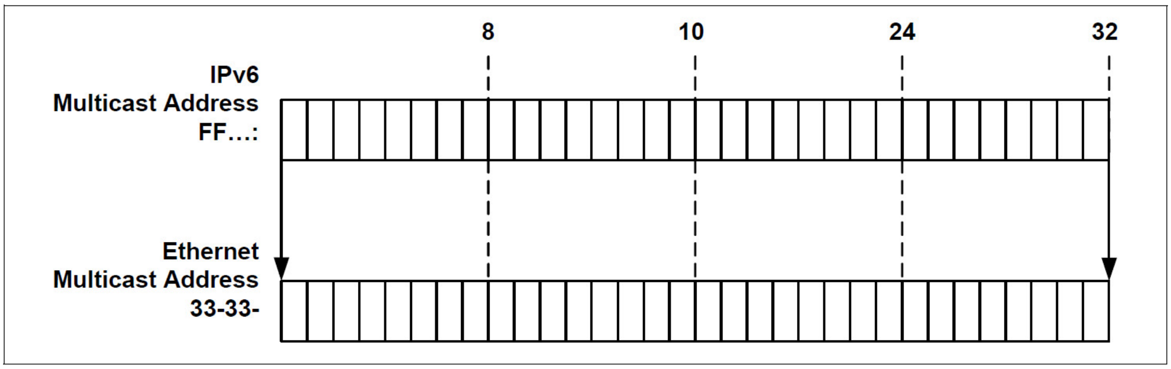
MLD Snooping can be operational even while IGMP Snooping is operational. They are independent and their configurations are also independent. However, MLD Snooping and IGMP Snooping share the same MFDB table. No preference is given to entries of either protocol in the MFDB table and entries are added in the order the control packets are received.

6.2.16.2. Multicast Forwarding Database (MFDB)

MLD Snooping switches build forwarding lists by intercepting MLD messages. Although the software processing the MLD messages could maintain state information based on the full IP group addresses, the forwarding tables in ICOS are mapped to link layer addresses. The Multicast Forwarding Database (MFDB) manages the forwarding address table for layer 2 multicast protocols, including IGMP and MLD Snooping. If a report for a particular group on a particular interface is not received within a certain time interval (group membership interval), the IGMP/MLD snooping component deletes the interface from the list of interfaces for that group entry. The value for group membership interval time is configurable using management. If an IGMP/MLD Leave/Done Group message is received on an interface, the IGMP/MLD Snooping code sends a general query on that interface and waits a specified length of time (maximum response time). If no response is received within that time, that interface is removed from the group. The value for maximum response time is configurable using management.

Only the least significant 32 bits of the IPv6 address are mapped to Ethernet addresses. All the IPv6 Multicast MAC addresses have their first two most significant bytes as 33 (33:33:XX:XX:XX:XX). The latter 32 bits are extracted from the four least significant bytes of the IPv6 multicast address.

Figure 6.6. IPv6-Ethernet Address Mapping in MLD



MLD Snooping uses the MFDB component to create a VID:MAC entry in the hardware for the groups it has learned.

In addition to building and maintaining lists of multicast group memberships, the Snooping switch also maintains a list of multicast routers. Multicast packets should be forwarded on ports that have joined using MLD/IGMP membership report packets and also on ports on which multicast routers are attached. The reason for this is that in MLD/IGMP there is only one active querier. This means that all other routers on the network are suppressed and thus not detectable by the switch. If a query is not received on an interface within a specified length of time (multicast router present expiration time), that interface is removed from the list of interfaces with multicast routers attached. The multicast router present expiration time is configurable using management. The default value for the multicast router expiration time is zero, which indicates an infinite time-out; that is, no expiration.

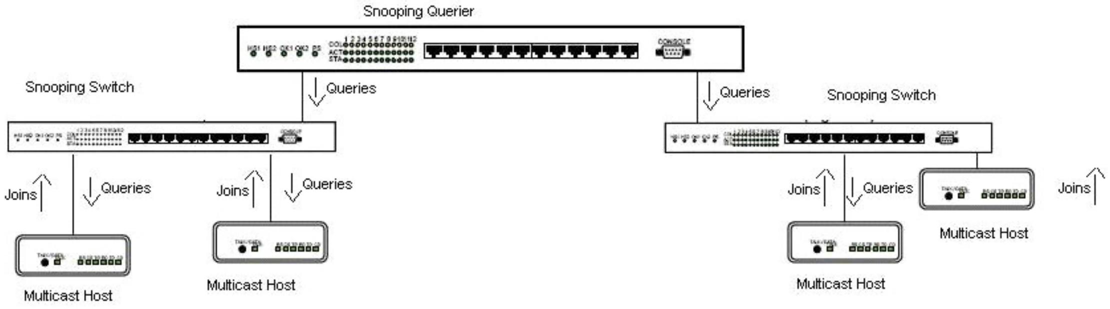
6.2.17. IGMP and MLD Snooping Querier

The IGMP/MLD Snooping Querier is an extension to the IGMP/MLD Snooping feature. It enhances the switch capability to simulate a IGMP/MLD router in a Layer 2 network, thus removing the need to have an IGMP/MLD Router in a Layer 2 network to collect the Multicast group membership information. The querier functionality is a small subset of the IGMP/MLD router functionality

IGMP Snooping Querier and MLD Snooping Querier are interoperable and can be enabled simultaneously. This section refers generally to both as the Snooping Querier.

Figure below illustrates Snooping Querier functionality in a network.

Figure 6.7. Snooping Querier



Snooping Querier should be used to support IGMP/MLD snooping in a VLAN where Multicast Routers are not configured because the multicast traffic does not need to be routed. In a network with IP multicast routing, the IP multicast router acts as the IGMP/MLD querier. If the IP-multicast traffic in a VLAN must be Layer 2-switchedonly, an IP-multicast router is not required; but without an IP-multicast router on a VLAN, the switch could be configured as the IGMP/MLD querier so that it can send queries. When IGMP/MLD Snooping Querier is enabled, the Querier sends out periodic IGMP/MLD General Queries that trigger the Multicast listeners/member to send their joins so as to receive the Multicast data traffic. IGMP/MLD snooping listens to these reports to establish appropriate forwarding.

6.2.17.1. Query Handling/Querier Election

When the switch receives an IGMP/MLD query with nonzero IP address, it means that:

- There is a Multicast Router in the same VLAN.
- There is a Snooping Querier in the same VLAN.

It cannot be concluded which of these is correct. To overcome this, Querier Election Mode is provided as a configuration parameter that the Administrator can use to toggle the switch to act as Querier or not.

If Querier Election mode is enabled, the Snooping Querier tries to compare its Snooping Querier Address (SQA) IP Address (R3) with the incoming Querier’s IP address Last Querier Address (LQA).

Table 6.3. Querier Election

Condition	Result
LQA < SQA	The last querier wins the election. Snooping Querier moves into operational nonquerier state. It starts the other querier time-out counter. Whenever it finds that the last querier is better than itself, it tops up the counter.
LQA = SQA	This condition arises because of a misconfiguration. The Snooping Querier remains in the Querier operational mode.
LQA > SQA	The Snooping Querier moves into Operational Mode = Querier.

Table 6.4. Version Conflict

Condition	Result
LQV > SQV	Snooping Querier is still operationally in Querier mode. Snooping Querier waits for the LQ to downgrade to the version equal to itself.
LQV = SQV	Querier election as described above.
LQV < SQV	Snooping Querier downgrades its operating version to the LQV.

LQV = Received Querier Version

SQV = Switch operation Querier Version

When Querier Election mode is disabled, Snooping Querier does not participate in the querier election (it moves to Operational State—Nonquerier). Upon receiving a membership query with a nonzero source address, it waits for the last querier to time out. When the Snooping Querier is in the Nonquerier state, the software maintains the multicast router attached ports list. When the Snooping Querier moves to the Querier state, the router port list is cleared for the ports in that VLAN.

The Querier Election Mode parameter is used to provide flexibility to the user to configure the Snooping Querier's behavior upon seeing a querier in the VLAN, as described in the following scenarios.

- Scenario 1: The user knows that there is no other Snooping Querier, and if any querier is there, it is the multicast router. In such a scenario, the user would want the Snooping Querier to give up and not participate in the querier election.
- Scenario 2: The user wants to configure multiple Snooping Queriers in the network for the sake of redundancy.

When the Snooping Querier receives a Query message with a zero IP address (0.0.0.0 / ::), it forwards a membership query on all interfaces on this VLAN, except the incoming interface.

6.2.17.2. Sending Queries

When the Snooping Querier is in Operational State = Querier on a VLAN, it sends out periodic General Queries with the operational version as the Querier Version and the IP source address as the Snooping Querier Address. This Query is sent on all the member ports that are forwarding. For MLDv1/IGMPv2,3 queries, the max response time (configured as snooping configuration parameter) is to be filled in the query packet being sent.

A global Query Interval timer is maintained for all the Querier-enabled VLANs; thus, it tries to send out the periodic query for all the Querier-enabled VLANs. If all 4K VLANs are supported as Querier-Enabled, performance could suffer. For this reason, a limit is placed on the number of VLANs that can be enabled for Snooping Querier. The other option is to implement the timers independent of each VLAN, but, again, this might also result in performance issues. Due to resource constraints, the Querier feature can be enabled up to a maximum of 256 different VLANs, which is a platform-specific constraint.

ICOS also supports the following IGMP and Multicast Group Membership Discovery (MGMD) Snooping solutions:

- **Configurable flooding behavior for unregistered multicast streams**

Added a configuration options for to filtering or forwarding unregistered multicast streams. This setting is configurable on a per-VLAN basis. The following CLI commands were added to accomplish this:

mac address-table multicast forbidden-unregistered vlan vlan-id

Forbids forwarding of unregistered multicast streams in the specified VLAN ID.

mac address-table multicast forward-unregistered vlan vlan-id

Forwards unregistered multicast streams in the specified VLAN ID.

mac address-table multicast forward-all vlan vlan-id

Forwards all multicast streams (both registered and unregistered) in the specified VLAN ID.

- **IGMP Report Suppression Mechanism**

The switch uses IGMP report suppression to limit the membership report traffic sent to multicast-capable routers. The switch forwards only one IGMP report per multicast router query to multicast devices. When IGMP report suppression is enabled, the switch sends the first IGMP report from all hosts for a group to all multicast routers. By default, the feature is disabled on ICOS.

The feature is supported only when the multicast query has IGMPv1 and IGMPv2 reports. This feature is not supported when the query includes IGMPv3 reports.

- **Leave Report handling Enhancement**

If the Querier is operating in IGMPv2/MLDv1 mode only, a Group-Specific query is sent on the interface that received the Leave message in response to an incoming Leave report.

- **Forwarding in Ingress VLAN with Routing package present**

The StrataXGS IV family of silicon has the capability to schedule L2 Multicast Table lookups if there are no IPMC entries, thereby achieving both forwarding in the Ingress VLAN and routing across VLANs using the same two tables. ICOS enhances the MGMD Snooping solution to capitalize on this silicon capability.

6.2.18. Protected Ports

The protected ports feature can be used to prevent ports from forwarding traffic to each other, even if they are on the same VLAN. Ports are designated as either protected or unprotected. Ports are unprotected by default.

6.2.19. Multicast VLAN Registration

Multicast VLAN registration (MVR), like IGMP snooping, allows the Layer 2 switch to listen to the IGMP frames. MVR and IGMP snooping operate independently from each other, and both protocols can be enabled on the same switch interfaces. When both MVR and IGMP snooping are enabled on an interface, MVR listens to the join and report messages only for groups configured statically; all other groups are managed by IGMP snooping.

There are two types of MVR ports:

- A *source* port is a port to which the multicast traffic is flowing. It must be a member of the multicast VLAN.
- A *receiver* port is a port to which the listening host is connected. It can be a member of any VLAN except the multicast VLAN.

The MVR port type is configured by the administrator.

The multicast VLAN is the VLAN that is used in the network for MVR purposes. Only one multicast VLAN can be configured per switch. Multicast data sent on the multicast VLAN is forwarded to all MVR receiver ports that are in different VLANs.

6.2.20. Multiple Spanning Tree Protocol

The Multiple Spanning Tree Protocol (MSTP) component complies with IEEE 802.1s by efficiently navigating VLAN traffic over separate interfaces for multiple instances of Spanning Tree. IEEE 802.1w, Rapid Spanning Tree, is supported through the IEEE 802.1s implementation. The difference between the RSTP and STP (IEEE 802.1D) is the ability to configure and recognize full-duplex connectivity and ports that are connected to end stations. The difference enables RSTP to rapidly transition to the *Forwarding* state and to suppress the Topology Change Notification PDUs, where possible.

A VLAN ID does not have to be preconfigured before mapping it to an MST instance.

6.2.20.1. STP Loop Guard

The Loop Guard feature is an enhancement of the Multiple Spanning Tree Protocol. Loop guard protects a network from forwarding loops induced by BPDU packet loss. It can be configured to prevent a blocked port from transitioning to the forwarding state when the port stops receiving BPDUs for some reason (such as a unidirectional link failure).

6.2.20.2. STP BPDU Guard

The STP BPDU guard allows network administrator to enforce the STP domain borders and keep the active topology be consistent and predictable. The switches behind the edge ports that have STP BPDU guard enabled are not able to influence the overall STP topology. At the reception of BPDUs, the BPDU guard operation disables the port that is configured with this option and transitions the port into disable state. This would lead to administrative disable of the port.

6.2.20.3. STP Root Guard

The root guard ensures that the port on which root guard is enabled is the designated port. In a root bridge ports are all designated ports, unless two or more ports of the root bridge are connected together. If the bridge receives superior STP BPDUs on a root guard enabled port, root guard moves this port to a root inconsistent STP state. This root inconsistent state is effectively equal to a listening state. No traffic is forwarded across this port. In this way, the root guard enforces the position of the root bridge. In MSTP scenario the port may be designated in one of the instances while being alternate in the CIST, and so on. Root guard is a per port (not a per port per instance command) configuration so all the MSTP instances this port participates in should not be in root role.

6.2.20.4. STP BPDU Filtering

STP BPDU filtering applies to all operational edge ports. Edge Port in an operational state is supposed to be connected to hosts that typically drop BPDUs. If an operational edge port receives a BPDU, it immediately loses its operational status. In that case, if BPDU filtering is enabled on this port then it drops the BPDUs received on this port.

6.2.20.5. STP BPDU Flooding

STP BPDU flooding feature applies to the STP disabled switch. To enable BPDU flooding on a port, STP should be disabled on the switch administratively. When this feature is enabled on the switch, it floods all the ports with the BPDU flood feature enabled on it.

6.2.21. Per-VLAN Spanning Tree Protocol and Per-VLAN Rapid Spanning Tree Protocol

Per-VLAN Spanning Tree Protocol and Per-VLAN Rapid Spanning Tree Protocol (PVSTP/ PVRSTP) are per- VLAN versions of STP (IEEE 802.1d) and RSTP (IEEE 802.1w), respectively. In other words, with PVSTP/ PVRSTP, each configured VLAN runs an independent instance of PVST/PRVST. Each PVSTP/PVRSTP instance elects a root bridge independent of the other. This means there are as many root bridges in the region as there are VLANs configured for PVSTP/ PVRSTP.

The difference between STP and PVSTP/PVRSTP is primarily in the way the protocol maps spanning tree instances to VLANs: PVSTP/PVRSTP creates a spanning tree instance for every VLAN, whereas STP maps all VLANs to one instance.

PVSTP is equivalent to the Cisco PVST+ protocol, and the two protocols can interoperate. Similarly, PVRSTP is equivalent to Cisco’s RPVST+, and these two protocols can interoperate.

Enabling PVSTP or PVRSTP on a switch disables other spanning tree modes on the switch.

The switch running PVSTP/PVRSTP transmits IEEE spanning tree BPDUs along with SSTP BPDUs. The SSTP BPDUs are transmitted as untagged packets on an access or native VLAN and transmitted as tagged packets on other VLANs.

If the switch running PVSTP/PVRSTP receives an IEEE spanning tree BPDU, then the switch will include the BPDU in an access VLAN or native VLAN instance.

PVSTP/PVRSTP behavior is as follows:

- Access port: If the port is configured as access port, then the port sends IEEE spanning tree BPDUs.
- Trunk port: If the port is configured as a trunk port, then the port sends IEEE spanning tree BPDUs and SSTP BPDUs on the native VLAN. For other VLANs, the port transmits SSTP BPDUs

as tagged packets with the respective VLANs. If the trunk port receives IEEE spanning tree BPDUs, then the received BPDUs are consumed and processed by the instance that is mapped to the native VLAN. The SSTP BPDUs are processed by instances to which the respective VLANs are mapped.

If the switch that is running a standard IEEE spanning tree protocol (STP, RSTP, or MSTP), and it receives the SSTP format BPDUs, the switch does not treat them as standard BPDUs. Hence, the received SSTP-formatted BPDUs are flooded on all the ports of the corresponding VLAN. The SSTP BPDUs are multicast over the region.

The interoperation between a switch that runs a standard IEEE spanning tree protocol and a switch that runs PVSTP/PVRSTP is achieved using CIST. In other words, to communicate with each other, a switch running the standard IEEE spanning tree protocol uses its CIST, and a switch running PVSTP/PVRSTP uses an access VLAN or native VLAN instance.

The PVSTP/PVRSTP FastUplink feature is used for quick selection of a port with the lowest cost when the root port fails. In other words, the FastUplink feature is used to reduce convergence time when a link fails. This feature is similar to Cisco’s UplinkFast feature. When the primary link fails, FastUplink creates an alternate path immediately. Since the ports that apply FastUplink do not have to wait for normal convergence time, this will speed up the transition from the failed primary link to the backup link (the port in a blocking state).

FastBackbone is a PVSTP feature that allows for faster convergence time when an indirect link to root fails. When a root port or blocked port receives an inferior BPDU from the designated switch on that port, the switch determines that an indirect link to the root has failed. To speed up convergence, the max age timer is immediately expired, and the port is put through the Listening and Learning states.

Both FastBackbone and FastUplink work only with PVSTP.

The timer speed for normal timers versus timers with FastBackbone are as follows:

- Normal Timers = Max Age (20 sec by default) + Listening (15 sec) + Learning (15 sec)
- With FastBackbone = Max Age (0 Expired) + Listening (15 sec) + Learning (15 sec)

6.2.22. Port-Based Network Access Control (IEEE 802.1X)

Local Area Networks (LANs) are often deployed in environments that permit the attachment of unauthorized devices. The networks also permit unauthorized users to attempt to access the LAN through existing equipment. In such environments, the administrator may desire to restrict access to the services offered by the LAN.

Port-based network access control makes use of the physical characteristics of LAN infrastructures to provide a means of authenticating and authorizing devices attached to a LAN port. Port-based network access control prevents access to the port in cases in which the authentication and authorization process fails. A port is defined as a single point of attachment to the LAN.

The software also supports VLAN assignment clients based on the RADIUS server authentication.

6.2.23. Port Mirroring

Port mirroring is used to monitor the network traffic that one or more ports or the ports within a VLAN send and receive. The Port Mirroring feature creates a copy of the traffic that the source interface handles and sends it to a destination port or a Remote Switched Port Analyzer (RSPAN) VLAN. All traffic from the source can be mirrored and sent toward the destination, or you can specify that only traffic flows that match the criteria in an ACL are mirrored.

The source is the port or VLAN that is being monitored. The destination is where the packets from the source port are sent. When the destination is a port on the local device, a network protocol analyzer is typically connected to the Ethernet port to analyze the traffic patterns of source ports.

The port mirroring feature allows the user to configure multiple sessions. One session consists of one destination port and multiple source interfaces (physical port or VLAN member ports). When a particular session is enabled, any traffic entering or leaving the source interfaces of that session is copied (mirrored) onto the destination port or RSPAN VLAN.

ICOS supports up to four monitor sessions and up to four RSPAN VLANs.

A session is operationally active only if both a destination port and at least one source port are configured. If neither is true, the session is inactive.

A port configured as a destination port acts as a mirroring port when the session is operationally active. If it is not, the port acts as a normal port and participates in all normal operation with respect to transmitting traffic.

The exact behavior of the mirroring feature may vary with the platform. Typical behavior is described as follows:

- Source: Any Ethernet port, LAG port, or VLAN. Platforms may behave unpredictably if an attempt is made to mirror a port of greater speed than the monitoring port.
- Traffic monitored: Ingress, Egress or both.
- Reflector port: a trunk port that carries the mirrored traffic towards the destination device.
- Probe port is not network connected: Once configured, there is no network connectivity on the probe port.

The probe port does not forward any traffic and does not receive any traffic. The probe tool attached to the probe port is generally unable to ping the networking device or ping through the networking device, and nobody is able to ping the probe tool.

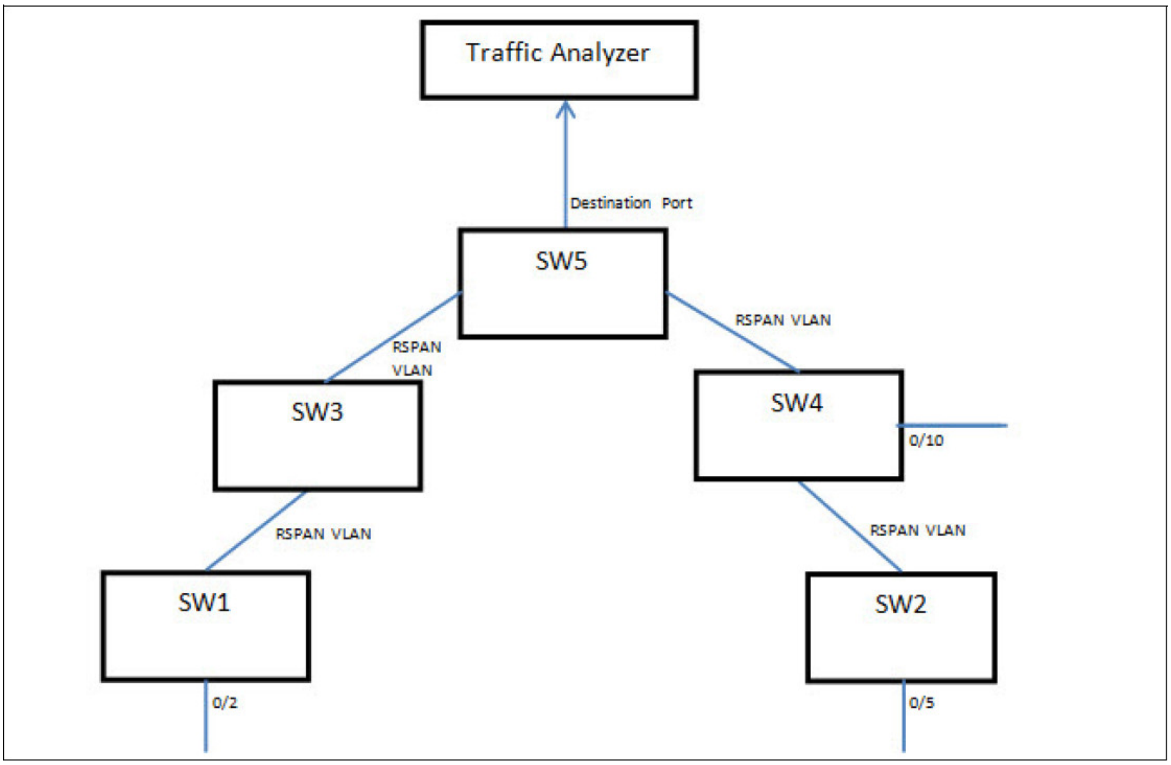
6.2.23.1. Remote Switched Port Analyzer

The figure below illustrates an example of a network that uses RSPAN. The example shows five switches. Switches SW1, SW2, and SW4 are the source switches. Switch SW3 acts as an intermediate switch. Switch SW5 is the destination switch.

Switch SW4 acts as source switch as well as the intermediate switch for network traffic from switch /SW1.

The ports connected towards the destination switch (SW5) must be configured with tagging, and with the VLAN ID as the RSPAN VLAN. These ports are configured with the RSPAN VLAN participation as well.

Figure 6.8. RSPAN Example



On the source switches, the traffic received/transmitted on source ports (0/2 on SW1, 0/5 on SW2 and 0/10 on SW4) is tagged with the RSPAN VLAN and transmitted on the configured reflector port. The reflector port is the physical interface that carries the mirrored traffic towards the destination switch (SW5).

The intermediate switch (SW3) just forwards the incoming tagged traffic towards the destination switch (SW5). The destination switch (SW5) accepts all the tagged (with RSPAN VLAN) packets and mirrors them on the destination port (to which the traffic analyzer is connected).

At the source switch (SW1, SW2 and SW4) the below parameters are configured:

- Source ports (i.e. the traffic on this port is mirrored)
- RSPAN VLAN (as destination)
- Reflector port
- Tx/Rx

At the destination switch (SW5) the below parameters are configured:

- RSPAN VLAN (as source)

- Probe port

Access mode ports configured with RSPAN VLAN membership are inactive

6.2.23.2. Flow-Based Mirroring

ACLs are attached to the mirroring session. The network traffic that matches the ACL is only sent to the destination port. This feature is supported for remote monitoring also. An IP/MAC access-list can be attached to the mirroring session.

6.2.23.3. CPU Traffic Filters

When mirroring traffic to and from the CPU, filters can be created that match only certain packets for inclusion in the packet capture. Filters can be based on the protocol along with IP address, MAC address, and TCP and UPD port numbers. In lieu of a named protocol, a custom option can be used to specify the offset and data to match.

The match condition for the filter can be one or more of the following: STP, LACPDU, ARP, UDLD, BCAST, MCAST, UCAST, LLDP, IP, OSPF, BGP, DHCP, SRCIP, DSTIP, SRCMAC, DSTMAC, SRCTCP, DSTTCP, SRCUDP, DSTUDP, and custom data with offset.

ICOS supports using two software filters (one filter for Tx and one for Rx), and can configure the filter to match one, multiple, or all of the supported protocols in the Tx or Rx direction, or both directions. CPU traffic either in the Rx or Tx direction is compared with the defined user-level filters. Filter statistics are updated for the packet matching the filter.

Statistics counters are available for each filter option per interface and direction. For example, if a filter is defined for STP and LACPDU packets on port-1 for Rx and Tx direction, then each STP or LACPDU packet received on port-1 increments STP and LACP counter statistics. Similarly, STP or LACPDU packets sent by the switch from port-1 also increment the counter statistics. The counter statistics for an interface are associated with the last updated timestamp to determine when the counter on an interface was most recently updated.

6.2.24. Internal 802.1X Authentication Server

The User Manager component includes an internal 802.1X authentication server feature that enables the administrator to separately create, maintain, and authenticate users for network (802.1X) access.

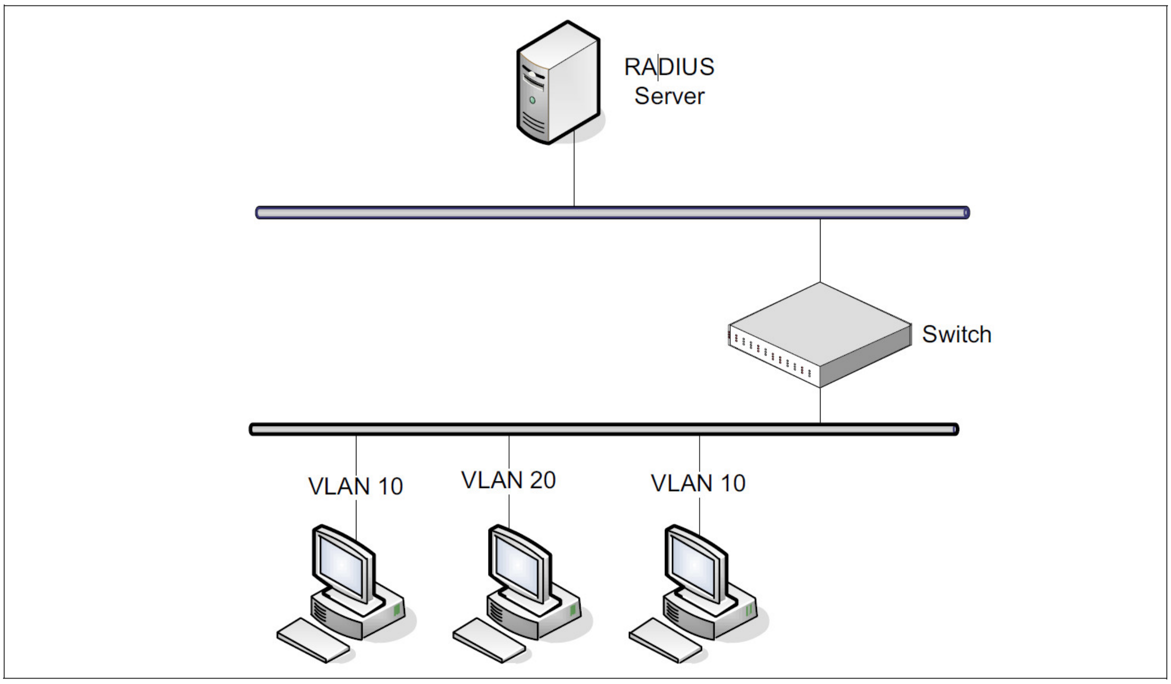
6.2.25. MAC-Based 802.1X

The MAC-Based Authentication is an extension to the 802.1X IEEE standard. This feature focuses on supporting authentication of multiple clients per port; that is, though a port is authorized by one of the clients connected to the port, the other clients that are connected to the same port of the switch do not have access to the port. Instead every client must authenticate itself before the client can get access to the port.

When a client authenticates itself initially on the network, the switch acts as the authenticator to the clients on the network and forwards the authentication request to the RADIUS server. If the authentication succeeds, the port is placed in authorized state and the client is able to forward or receive traffic through the port.

In a standard Dot1X scenario, all subsequent clients in the network that are connected to the same port need not authenticate to use the port on the switch. When MAC-based Dot1X authentication is enabled, all the subsequent clients in the network that are connected to the same port need to authenticate themselves to use the port on the switch.

Figure 6.9. MAC-Based Authentication



6.2.26. 802.1X Monitor Mode

The 802.1X monitor mode is a special mode that can be enabled in conjunction with Dot1X authentication. It allows network access even in the event of an authentication failure. The results of the authentication process are logged for diagnostic purposes. This mode provides a mechanism for the administrator to identify shortcomings in the configuration of a Dot1X authentication on the switch without affecting the network access to the users.

6.2.27. Dot1X Supplicant

IEEE 802.1X supplicant capability allows the system to authenticate itself with the network prior to being allowed to join it. The supplicant initiates communication with the authenticator by sending a start packet on port initialization. On reception of requests from the authenticator, the supplicant sends back the appropriate responses according to the 802.1X standard. On successful authentication, the supplicant port moves to the authenticated state.

The ICOS implementation and framework is based on the IEEE standard 802.1X 2004 and supports RFC 3748 for MD5 EAP challenges and RFC 2289 for OTP-plaintext password implementations.

6.2.28. Guest VLAN

The Guest VLAN feature allows a switch to provide a distinguished service to unauthenticated users (not rogue users who fail authentication). This feature provides a mechanism to allow visitors and contractors to have network access to reach external network with no ability to surf internal LAN.

When a client that does not support 802.1X is connected to an unauthorized port that is 802.1X-enabled, the client does not respond to the 802.1X requests from the switch. Therefore, the port remains in the unauthorized state, and the client is not granted access to the network. If a guest VLAN is configured for that port, then the port is placed in the configured guest VLAN, and the port is moved to the authorized state, allowing access to the client.

Client devices that are 802.1X-supplicant-enabled authenticate with the switch when they are plugged into the 802.1X-enabled switch port. The switch verifies the credentials of the client by communicating with an authentication server. If the credentials are verified, the authentication server informs the switch to *unblock* the switch port and allows the client unrestricted access to the network; that is, the client is a member of an internal VLAN.

Guest VLAN Supplicant mode is a global configuration for all the ports on the switch. When a port is configured for Guest VLAN in this mode, if a client fails authentication on the port, the client is assigned to the guest VLAN configured on that port. The port is assigned a Guest VLAN ID and is moved to the authorized status. Disabling the supplicant mode does not clear the ports that are already authorized and assigned Guest VLAN IDs.

6.2.29. Unauthenticated VLAN

Unauthenticated VLAN feature allows a switch to provide a distinguished service to unauthenticated users. This feature allows the authorization failed clients to have limited access to the network.

When an unauthenticated VLAN is configured on a port, and Dot1x is enabled on the port with port-based authentication, the port is placed in an unauthenticated VLAN when the supplicant on the port fails authentication. If a Guest VLAN is also enabled on the port, then the Guest VLAN is used only for ports that have Dot1x supplicants as unaware clients.

The reauthentication timer is still running when the unauthenticated VLAN is assigned to the port; if the port needs to be reauthenticated based on the presence of supplicants, it may be placed in the Guest VLAN (if one is configured) if no supplicants are available at that time.

When the unauthenticated VLAN is configured on a port and 802.1X is enabled on the port with MAC-based authentication, the client’s MAC address is associated with the unauthenticated VLAN when the supplicant fails authentication. If the Guest VLAN is also enabled on the port, the Guest VLAN is used only for ports that have 802.1X supplicant-unaware clients.

6.2.30. Multiuser VLAN Assignment and Filter ID Support

This feature allows RADIUS specified attributes, such as VLAN ID and Filter ID, to be associated with an authenticated client. When a client authenticates successfully, if a VLAN attribute is sent by the RADIUS server, the client is associated with the RADIUS assigned VLAN, that is, traffic from and to that client flows through the RADIUS assigned VLAN. Similarly, when the client authenticates successfully, if a Filter ID attribute is specified by the RADIUS server, this Filter ID is applied to the traffic from that client. The Filter ID identifies a DiffServ policy on the switch.

6.2.31. LLDP and LLDP-MED

6.2.31.1. Link Layer Discovery Protocol

The IEEE 802.1AB standard defines the Link Layer Discovery Protocol (LLDP). The protocol allows stations residing on an 802 LAN to advertise major capabilities, physical descriptions, and management information to physically adjacent devices, allowing a network management system (NMS) to access and display this information.

The standard is designed to be extensible, providing for the optional exchange of organizational specific information and data related to other IEEE standards. The initial ICOS implementation supports only the required basic management set of type length values (TLVs).

LLDP is a one-way protocol; there are no request/response sequences. Information is advertised by stations implementing the transmit function. The information is received and processed by stations implementing the receive function. Devices are not required to implement both transmit and receive functions and each function can be enabled or disabled separately by the network manager. ICOS supports both the transmit and receive functions in order to support device discovery.

The LLDP component transmit and receive functions can be enabled/disabled separately per physical port. By default, both transmit and receive functions are disabled on all ports. The application starts each transmit and receive state machine appropriately based on the configured status and operational state of the port.

The transmit function is configurable with respect to packet construction and timing parameters. The required Chassis ID, Port ID, and Time to Live (TTL) TLVs are always included in the Link Layer Discovery Protocol Data Unit (LLDPDU). However, inclusion of the optional TLVs in the management set is configurable by the administrator. By default, they are not included. The transmit function extracts the local system information and builds the LLDPDU based on the specified configuration for the port. In addition, the administrator has control over timing parameters affecting the TTL of LLDPDUs and the interval in which they are transmitted.

The receive function accepts incoming LLDPDU frames and stores information about the remote stations. Both local and remote data may be displayed by the user interface and retrieved using SNMP as defined in the LLDP MIB definitions. The component maintains one remote entry per physical network connection.

The LLDP component manages a number of statistical parameters representing the operation of each transmit and receive function on a per-port basis. These statistics may be displayed by the user interface and retrieved using SNMP as defined in the MIB definitions.

6.2.31.2. Media Endpoint Discovery Extensions (LLDP-MED)

LLDP-MED uses LLDP’s organizationally-specific TLV extensions and defines new TLVs that facilitate deployment of VoIP in a wired or wireless LAN/MAN environment. It also makes mandatory a few optional TLVs from LLDP and recommends not transmitting some TLVs.

The TLVs only communicate information; they do not automatically translate into configuration. An external application may query the MED MIB and take management actions in configuring functionality.

Since LLDP-MED uses the framework of LLDP, it is bound by the same requirements of the original specification. The frame format, restrictions, and implications are all preserved.

The standard defines four types of LLDP-MED devices. Three of them represent the actual endpoints, classified as Class I Generic (IP Communication Controller, and so on), Class II Media (Conference Bridge, and so on), and Class III Communication (IP Telephone and so on). The fourth device is network connectivity device, which is typically a LAN switch/router, IEEE 802.1 bridge, or IEEE 802.11 wireless access point, and so on.

The ICOS implementation of LLDP-MED focuses on network connectivity devices. The TLVs that are to be transmitted from a MED perspective can be grouped into the following categories:

- Mandatory 802.1AB TLVs (with modifications)
  - Chassis ID TLV (subtype shall default to MAC Address)
  - Port ID TLV (subtype shall default to MAC address
  - TTL TLV
  - MAC/PHY configuration/status TLV
  - End of lldpdu
- Optional 802.1AB TLV
  - Systems Capabilities TLV. Optional for inclusion in the transmission of LLDPDU.
  - Power using MDI TLV: This TLV is NOT recommended for transmission in order to conserve LLDPDU space.
- Mandatory LLDP-MED TLVs
  - LLDP-MED Capabilities
- Conditionally Required LLDP-MED TLVs
  - Network Policy
  - Location Identification
  - Extended Power-via-MDI
  - Optional LLDP-MED TLVs
  - Inventory Management TLVs

For further information about the TLVs, refer to 802.1AB.

LLDP can have multiple LLDP neighbors per interface. The number of such neighbors supported is limited by the memory constraints. A product specific constant defines the maximum number of neighbors supported by the switch. There is no restriction on the number of neighbors supported on a per LLDP port. If all the remote entries on the switch are filled up, the new neighbors are ignored.

6.2.32. DHCP Snooping and DHCPv6 Snooping

DHCP Snooping is a security feature that monitors DHCP messages between DHCP clients and DHCP servers to filter harmful DHCP messages and build a bindings database of {MAC address, IP address, VLAN ID, interface} tuples that are considered authorized.

DHCPv6 snooping works only with DHCPv6 stateful server

The DHCP snooping application processes incoming DHCP messages. For DHCPRELEASE and DHCPDECLINE messages from the DHCPv4 client and for RELEASE and DECLINE messages from the DHCPv6 client and RECONFIGURE messages from the DHCPv6 server, the application compares the receive interface and VLAN with the client's interface and VLAN in the bindings database. If the interfaces do not match, the application logs the event and drops the message. For valid client messages, DHCP snooping compares the source MAC address to the DHCP client hardware address. When there is a mismatch, DHCP snooping logs and drops the packet. The network administrator can disable this feature using no ip dhcp snooping verify mac-address for DHCPv4 and no ipv6 dhcp snooping verify mac-address for DHCPv6. DHCP Snooping forwards valid client messages on trusted members within the VLAN. If DHCP Relay and/or DHCP Server coexist with DHCP Snooping, the DHCP client message is sent to the DHCP Relay or/and DHCP Server for further processing.

The DHCP Snooping application uses DHCP messages to build and maintain the binding's database. The binding's database only includes data for clients on untrusted ports. DHCP Snooping creates a tentative binding from DHCP DISCOVER (DHCPv4), SOLICIT (DHCPv6), and REQUEST messages. Tentative bindings tie a client to a port (the port where the DHCP client message was received). Tentative bindings are completed when DHCP Snooping learns the client's IP address from a DHCP ACK message from a DHCPv4 server or a REPLY message from a DHCPv6 server on a trusted port. DHCP snooping removes bindings in response to DECLINE and RELEASE from a DHCPv4 client and NACK from a DHCPv4 server. Similarly, for RELEASE and DECLINE from a DHCPv6 client and RECONFIGURE message received from a DHCPv6 client, the snooping component removes the bindings from the database. The DHCP Snooping application ignores the ACK messages sent as a replies to DHCP Inform messages received on trusted ports from DHCPv4 servers. Likewise, the DHCP snooping application ignores REPLY messages that are sent in response to CONFIRM messages received on trusted ports from DHCPv6 servers. The network administrator can enter static bindings into the binding database.

IP Source Guard and Dynamic ARP Inspection use the DHCP Snooping bindings database for the validation of IP and ARP packets.

6.2.33. IP Source Guard

IP Source Guard (IPSG) is a security feature that filters IP packets based on source ID. The source ID may either be source IP address or a {source IP address, source MAC address} pair. The network administrator configures whether enforcement includes the source MAC address. The network administrator can configure static authorized source IDs. The DHCP Snooping bindings database and static IPSG entries identify authorized source IDs. IPSG is enabled on physical and LAG ports. IPSG is disabled by default.

If the network administrator enables IPSG on a port where DHCP snooping is disabled or where DHCP snooping is enabled but the port is trusted, all IP traffic received on that port is dropped depending upon the admin configured IPSG entries. IPSG cannot be enabled on a port-based routing interface.

IPSG uses two enforcement mechanisms: the L2FDB to enforce the source MAC address and ingress VLAN and an ingress classifier to enforce the source IP address or {source IP, source MAC} pair.

6.2.33.1. IPv6 Source Guard

IPv6 source guard (IPv6SG) is a security feature that filters IPv6 packets based on source ID. The source ID is either a source IPv6 address or {source IPv6 address and source MAC address pair}. The network administrator configures whether enforcement includes the source MAC address.

The DHCPv6 snooping binding database and static IPv6SG entries configured by administrator are identified as authorized source IDs.

Initially, all IPv6 traffic on the IPv6SG enabled port is blocked except for DHCPv6 packets. After a client receives an IP address from the DHCPv6 server, or after a static IPv6 source binding is configured by the administrator, all traffic with that IPv6 source address is permitted from that client. Traffic from other hosts is denied.

For each source ID in the binding database and for all manual IPv6SG entries, IPv6SG notifies the driver to install an ingress classifier rule permitting matching packets. If source MAC checking is configured, the classifier verifies that the {source IPv6 address, source MAC address} pair matches a DHCP binding. The hardware drops unauthorized packets. If the number of stations on a port exceeds the available number of classifier rules, then the hardware installs rules for the number of source IDs that fit. Traffic from other sources is dropped.

IPv6SG is enabled on physical and LAG ports. IPSG is disabled by default. Zero, multicast and loopback IPv6 addresses are not allowed to configure as IPv6 static source guard entry.

6.2.34. Dynamic ARP Inspection

Dynamic ARP Inspection (DAI) is a security feature that rejects invalid and malicious ARP packets. The feature prevents a class of man-in-the-middle attacks, where an unfriendly station intercepts traffic for other stations by poisoning the ARP caches of its neighbors. The miscreant sends ARP requests or responses mapping another station IP address to its own MAC address.

DAI drops ARP packets whose sender MAC address and sender IP address do not match an entry in the DHCP Snooping bindings database.

6.2.35. DHCP L2 Relay

The DHCP Relay agent is a network device used for communication between DHCP client and server when they are placed in different subnets and to complete the DHCP protocol operation. Such a device, a L3 Relay agent, is generally a router that has IP interfaces on both the client and server subnets and can route between them. However in a L2 Switched networks there may be one or more infrastructure devices (such as a switch) in between the client and the L3 Relay agent; and so some of the client device information required by the L3 Relay agent may not be visible to it. In this case, a L2 Relay agent can be used to add the further information that the L3 Relay Agent and DHCP Server may need to complete the task required by the network administrator.

The DHCP relay agent’s role was expanded by RFC 3046, which specifies the DHCP relay agent Information Option, more commonly referred to as DHCP option-82. The relay agent may add this option to packets it relays from clients to servers, and must remove this option from packets it relays from servers to clients. The relay agent information option defines two suboptions with defined meanings, circuit-id and remote-id, to carry information that the relay agent may know about the circuit or attachment point of the client, and about the client’s identity. The format of these suboptions is implementation-specific. These suboptions may be used by the DHCP server to affect how it treats the client, and also may be used by the relay agent to limit broadcast replies to the specific circuit or attachment point of the client. The L2 Relay component covers this functionality for ICOS. The circuit-id added is the port number for the client requests. The configuration of the options is based on the service VLAN-IDs and service subscriptions.

6.2.36. Storm Control

Storm control allows for rate limiting of specific types of packets through the forwarding plane. The administrator can configure the absolute rate in packets-per-second for the Storm control threshold. Each classified packet type (broadcast, multicast, or unicast) can be enabled/disabled per port, and the threshold level at which Storm- Control is active is also configurable per-port and per-type (as a percentage of interface speed).

Upon enabling Storm control on an interface, if the ingress rate of that type of packet (L2 broadcast, multicast, or unicast) is greater than the configured threshold level (as a percentage of port speed or as an absolute packets-per-second rate), the forwarding-plane discards the excess traffic.

Per-port and per-storm control type (broadcast, multicast, or unicast), the storm control feature can be configured to automatically shut down a port when a storm condition is detected on the port; or to send a trap. When configured to shut down, the port is put into a diag-disabled state. The user must manually re-enable the interface for it to be operational. When configured to send a trap, the trap is sent once in every 30 seconds.

This feature may not be available on all platforms.

6.2.37. Port Security

Port Security allows a network administrator to secure the network by locking down allowable MAC addresses on a given port. Packets with a matching source MAC address (secure packets) are forwarded. All other packets (unsecure packets) are restricted.

ICOS Port Security implements two traffic filtering methods:

- Dynamic Locking: The user specifies the maximum number of dynamic MAC addresses that can be learned on a port. The Maximum number of MAC addresses is platform-dependent and is given in the software release notes. After the limit is reached, additional MAC addresses are not learned. Only frames with an allowable source MAC address are forwarded.

- Static Locking: The user manually specifies a list of static MAC addresses for a port. Dynamically locked addresses can be converted to statically locked addresses.

The traffic filtering methods are used concurrently.

Dynamic locking implements a *first arrival* mechanism for Port Security. The user specifies how many addresses can be learned on the locked port. If the limit has not been reached, then a packet with an unknown source MAC address is learned and forwarded normally. Once the limit is reached, no more addresses are learned on the port. Any packets with source MAC addresses that were not already learned are discarded. Note that the user can effectively disable dynamic locking by setting the number of allowable dynamic entries to zero.

Static locking allows the user to specify a list of MAC addresses that are allowed on a port. The behavior of packets is the same as for dynamic locking: only packets with an allowable source MAC address can be forwarded. Note that the user is allowed to take all of the dynamically locked MAC addresses on a port and move them to a static state.

Port Security helps secure the network by preventing unknown devices from forwarding packets into the network. The user can lock down a port, and only a specified number of addresses can be learned on that port. For instance, if the users want to ensure that only a single device can be active on a port, they can set the number of allowable dynamic addresses to one. After the MAC address of the first device is learned, no other devices are allowed to forward frames into the network.

When link goes down on a port, all of the dynamically locked addresses are *freed*. That is, when link is restored, that port can once again learn addresses up to the user specified limit.

If the users know the specific MAC address (or addresses) that exists off of a particular port, they can specify those addresses as allowable. By setting the number of allowable dynamic entries to zero, only packets with a source MAC address matching a MAC address in the static list can be forwarded.

A dynamically locked MAC address is eligible to be aged out if another packet with that MAC address is not seen within the age-out time. Dynamically locked MAC addresses are also eligible to be relearned on another port if a station movement occurs. Statically locked MAC addresses are not eligible for aging. If a packet comes in on a port with a MAC address that is statically locked on another port, then that packet is discarded.

Dynamically locked addresses can be converted to statically locked addresses. If the limit of statically locked MAC addresses is less than the number of dynamically locked MAC addresses, then the addresses that are converted are done so on a first arrival basis (that is, the first *X* addresses are converted, where *X* is the number of remaining statically locked MAC addresses).

Traps can be generated when a packet is received on a locked port with a MAC address that is not allowable.

6.2.38. Unidirectional Link Detection (UDLD)

The UDLD feature detects unidirectional links to physical ports. UDLD must be enabled on the both sides of the link in order to detect an unidirectional link. The UDLD protocol operates by exchanging packets containing information about neighboring devices. The purpose of the UDLD feature is to detect and avoid unidirectional links. A unidirectional link is a forwarding anomaly in a Layer 2 communication channel in which a bidirectional link stops passing traffic in one direction.

6.2.39. Link Dependency

The link dependency feature allows specified ports to be enabled or disabled based on the link state of other ports. In other words, this feature allows the link state of certain ports to be dependent on the link state of other ports. In the simplest form, for example, if port A is dependent on port B, and the switch detects a link loss on port B, the switch automatically brings down the link on port A. When the link is restored to port B, the switch automatically restores the link to port A. The link action command option determines whether link A will come up or go down, depending upon the state of link B.

6.2.40. IPv6 RA Guard

The IPv6 RA guard feature allows the network administrator to block or reject rogue Router Advertisement (RA) or redirect messages on host ports. The ICOS RA guard implementation supports host mode configuration. In this mode, RA guard is enabled on a port that is connected to an end-host, so all router advertisement and redirect messages from this user are dropped by the switch.

ICOS supports stateless RA guard.

6.2.41. Management and Control Plane ACLs

ACLs on the CPU interface can be used to control management access to the switch. The administrator can define the IP address, MAC address, or protocol through which management access to the switch is allowed.

Control Plane ACLs work only for CPU Inbound traffic and cannot be used for CPU outbound traffic.

Control plane ACLs are only available on qualifying hardware with an egress field processor.

The Control-plane ACL feature is not applicable to traffic from out-of-band ports, also known as the service port. It is applicable only to packets coming to the CPU through the switching silicon.

- Layer 2, IPv4, and IPv6 ACLs are supported on the CPU port.

6.2.42. OpenFlow

The OpenFlow feature enables the switch to be managed by a centralized OpenFlow Controller using the OpenFlow protocol. ICOS supports the OpenFlow 1.0 and OpenFlow 1.3 standards.

The OpenFlow 1.0 standard supports a single-table data forwarding path. However ICOS supports Open Vswitch proprietary extensions to enable the OpenFlow controller to access multiple forwarding tables.

The OpenFlow 1.3 standard enables a multi-table data forwarding path. However, ICOS supports a singletable OpenFlow 1.3 data forwarding path. Support for additional hardware tables in the OpenFlow 1.3 data path may be added in future releases.

The OpenFlow feature has the following major functions:

- Enable and disable OpenFlow

- Deploy OpenFlow configuration
- Interact with the OpenFlow controllers
- Deploy OpenFlow controller flows
- Collect port and queue status and statistics
- Support OpenFlow controller group tables
- Support hardware network address translation

6.2.42.1. Interoperation with OpenFlow Controllers

ICOS implements certain enhancements to the OpenFlow protocol to optimize it for the Data Center environment and to make it compatible with Open vSwitch. ICOS interacts with any OpenFlow controller that supports OpenFlow 1.0 and OpenFlow 1.3 standards. Interoperability is verified, however, only with the Open Daylight Controller.

6.3. Data Center Module

The Data Center Module section describes the data center components. The following data center features are described in this section:

- Section 6.3.1, “Priority Flow Control”
- Section 6.3.2, “DCBX”
- Section 6.3.3, “802.1Qaz – Enhanced Transmission Selection”
- Section 6.3.4, “802.1Qau – Congestion Notification”
- Section 6.3.5, “Resilient Hashing”
- Section 6.3.6, “DCVPN Gateway”
- Section 6.3.7, “Overlay API”
- Section 6.3.8, “Zero-Touch Provisioning”
- Section 6.3.9, “OpenStack Plug-in”
- Section 6.3.10, “OpenStack Gateway (VTEP)”
- Section 6.3.11, “FIP Snooping”
- Section 6.3.12, “OpenDaylight Controller”
- Section 6.3.13, “Dynamic Topology Map and Prescriptive Topology Mapping”

6.3.1. Priority Flow Control

Priority Flow Control (PFC) provides a means of pausing individual priorities within a single physical link. By pausing the congested priority or priorities independently, protocols that are highly loss-sensitive can share the same link with traffic that has different loss tolerances. The priorities are differentiated by the 802.1p priority field of the 802.1Q VLAN header. PFC is available only on certain silicon.

PFC is intended to eliminate frame loss due to congestion on a link. This is achieved by a mechanism that is similar to the IEEE 802.3x pause mechanism, but operates on individual priorities. This mechanism, in conjunction with other data center bridging (DCB) technologies, enables support for higher layer protocols that are highly loss-sensitive while not affecting the operation of traditional LAN protocols that utilize other priorities.

PFC uses a unique control packet defined in IEEE 802.1Qbb; therefore, PFC is not compatible with 802.3x Flow Control (FC). An interface that is configured for PFC automatically disables FC. When PFC is disabled on an interface, the FC configuration for the interface becomes active. Any FC frames received on a PFC-configured interface are ignored.

Each 802.1p priority is configured as either drop or no-drop on a given front panel port. If an 802.1p priority that is designated as no-drop is congested, the priority is paused. Drop priorities do not participate in pause. By default, there are no 802.1p priority classifications configured and PFC is not enabled.

While several no-drop priorities may be configured on a supporting system, the actual number of lossless priorities supported on a given switch chip within the system is a function of the switch chip's packet buffer, the maximum supported MTU size, the pause delay, and the total number of ports. To guarantee lossless behavior, the switch chip must send a pause message prior to exhausting its available packet buffer and must have sufficient buffer to absorb the delay.

To guarantee lossless behavior, all interfaces must change to ingress-based congestion control when PFC is enabled on an interface. If egress congestion control is used with Head Of Line (HOL) prevention, the egress interface may drop packets prior to the ingress interface reaching its pause threshold. As a result, any interface that is not enabled for PFC or Flow Control may suffer HOL drop-like behavior when the headroom is exceeded and congestion exists. This behavior also exists when FC is enabled.

The effective behavior on an interface enabled for PFC without a no-drop priority is no-pause-enabled. If the user enables PFC but does not create any no-drop priorities, then the interface will not be lossless. Interfaces that are not enabled for PFC do not act on PFC control frames.

6.3.2. DCBX

Data Center Bridging Exchange Protocol (DCBX) is used by DCB devices to exchange configuration information with directly connected peers. DCBX is used in L2 only environments. The protocol is also used to configure and detect the configuration mismatch of the peer DCB devices. All of the data center applications (ETS, PFC and application priority) are used together in the data center to achieve the high availability, low latency, and loss less (enhanced Ethernet) service on the existing native Ethernet networks.

- DCBX is the interface for the following applications to propagate configuration information.
- Enhanced Transmission Selection (ETS)
- Priority-based Flow Control (PFC)
- Application Priorities (such as FCoE)

6.3.3. 802.1Qaz – Enhanced Transmission Selection

Enhanced Transmission Selection (ETS) is defined in the 802.1Qaz standard. ETS introduces a new 4-bit field called the Priority Group ID (PGID). There are 16 PGID values with 15 being a special “No Bandwidth Limit” value and 8–14 being reserved values. ETS allows one or more priorities to be assigned to a PGID. Each PGID is allocated a percentage of available bandwidth on the link. Available bandwidth refers to the maximum percentage of available link bandwidth after priorities within PGID 15 are serviced. Once allocated, a PGID may only use the available bandwidth up to the maximum percentage allocated. ETS may be used standalone with DCBX. However, all ETS/PFC/CN would be deployed in the data center domains. DCBX protocol advertises the ETS configuration to the peers and receives the ETS configuration from the peers. Based on the implementation, the switches can act upon the received ETS data from the peers.

6.3.4. 802.1Qau – Congestion Notification

Congestion Notification (CN) is defined in the 802.1Qau standard. A consequence of link level pausing (i.e. 802.1Qbb) is “congestion spreading”. This is the domino effect of buffer congestion propagating upstream causing secondary bottlenecks. A layer two congestion control algorithm allows a primary bottleneck to directly reduce the rates of those sources whose packets pass through it, thereby preventing secondary bottlenecks. Congestion notification is broken up into two algorithms: CP and RP. CP, Switch or Congestion Point Dynamics is the mechanism in which a switch buffer samples incoming packets and generates a feedback message addressed to the source of the sampled packets with the extent of the congestion. RP, Rate Limiter or Reaction Point Dynamics is the mechanism by which a Rate Limiter (RL) decreases its sending rate based on feedback and increases its rate voluntarily to recover lost bandwidth and probe for available bandwidth. Congestion Notification may be used standalone with DCBX. However, all ETS/PFC/CN would be deployed in the data center domains. DCBX protocol advertises the CN configuration to the peers and receives the CN configuration from the peers. Based on this implementation, the switches can act upon the received CN data from the peers.

6.3.5. Resilient Hashing

Resilient Hashing (RH) is an Netberg hardware platform based on Broadcom Tomahawk family of devices feature that introduces an extra level of indirection between the hash value and the selected output port for ECMP/LAG. In a typical non-RH configuration, the output port can be thought of as being selected by output_port = hash MOD #ports. Thus, if the number of ports changes, the output_port will likely change for all flows, even if the flow was on a port that was not affected. This can cause degraded performance due to frame reordering.

With RH, the output port is selected as output_port = table[hash]; i.e., the hash value is used to index into a table of ports. At start-of-day, that table is configured such that each output port is equally distributed, therefore causing traffic to be statistically equally distributed. If a port goes down, then only the entries that use that port need to be rewritten. Other ports are left untouched and, therefore, do not suffer degraded performance.

6.3.6. DCVPN Gateway

The Data Center Virtual Private Network (DCVPN) gateway feature enables network virtualization technologies (VXLAN/NVGRE) to communicate with another network, particularly a virtual LAN (VLAN). It offers VXLAN Tunnel End Point (VTEP) functionality for VXLAN and Network Visualization Edge (NVE) functionality for NVGRE tunnels on the switch.

Both VXLAN and NVGRE are layer-3, IP-based technologies that prepend an existing Layer-2 frame with a new IP header, providing layer-3-based tunneling capabilities for Layer 2-frames. Essentially, it extends a Layer-2 domain across a Layer-3 boundary.

For the traffic from a VXLAN/NVGRE to use services on physical devices in a distant network, the traffic must pass through a DCVPN Gateway.

This feature is configurable through CLI. It also offers an Overlay API to facilitate programming from external agents, as described in the following section.

6.3.7. Overlay API

The Overlay (L2-over-L3 Tunnel) OpEN API feature is a set of OpEN APIs that can be used to manage DCVPN Gateway functionality on the switch. It allows processes outside of the ICOS main process (switchdrvr) to create and manage L2 tunnels over an underlying L3 network (L2-over-L3 overlay). This facilitates the management of overlay tunnels on the switch by external controllers in the network or data centers. The tunnels can be of type VXLAN or NVGRE.

Overlay API is part of ICOS OpEN ADK. It also provides a sample OpEN application that demonstrates the use of the Overlay APIs. This application (tunnel_example) can be built using OpEN ADK and loaded on to the switch, and then started from the command line. It exercises all supported Overlay OpEN APIs with appropriate arguments to create/manipulate the tenant network and tunnels in the ICOS main process.

6.3.8. Zero-Touch Provisioning

The Zero-Touch Provisioning (ZTP) feature is an enhancement to the existing AutoInstall feature. It supports installation of Chef Client or Puppet Agent at the time of device boot-up. In previous releases, the user would install Chef Client/Puppet Agent manually by logging on to the device and executing series of commands or scripts. In ICOS 3.0, support was added for automatic installation of Chef Client/Puppet Agent. ZTP uses DHCP option 125 to download an .ini file from TFTP server, and then Chef Client/Puppet Agent is installed as defined in the .ini file.

Automatic installation of Chef Client or Puppet Agent occurs in the following situations:

- When the device boots with no saved configuration found in the designated storage areas.
- When the device boots with a saved configuration that has AutoInstall enabled.

6.3.9. OpenStack Plug-in

A Neutron MS plug-in allows basic configuration of the switch from within the OpenStack orchestration software.

The following can be configured:

- Interface settings
- VLAN creation, port membership, PVID

6.3.10. OpenStack Gateway (VTEP)

OpenStack VXLAN Tunnel End Point (VTEP) is designed to provide tenants in an Open-vSwitch (OVS)-based OpenStack cluster access to physical devices connected to specific Netberg switching hardware. This access comes via overlay networks that are tenant-specific (e.g., mapped to a tenant and thus providing tenant isolation). The feature provides the components of a side cluster that works with OpenStack and extends the tenant network to include hardware devices plugged into an Netberg switch. The product initially supports VXLAN but is actually designed to be trivially modifiable to support NVGRE or any other flavor of overlay network that is supported by both OpenStack/OVS and by the switch silicon/APIs.

Using Netberg hardware platform based on Broadcom Tomahawk family of devices, this feature enables the TOR to act as a VTEP for VXLAN tunnels in OpenStack. Little to no configuration is required, and the TOR automatically configures itself into the mesh of tunnels that OpenStack creates for the compute nodes.

6.3.11. FIP Snooping

ICOS switches with the FIP Snooping feature enabled are deployed as transit switches in the FCoE domain in order to provide the enhanced Ethernet transport service over the native Ethernet cloud in combination with other data center technologies like PFC, ETS, CN and DCBX.

Only transit switch functionality is supported. FCoE Forwarders (FCF) functionality is not supported.

FIP snooping is a frame inspection method that can be used by transit switches to monitor FIP frames and apply policies based on the information in those frames. This allows for:

- Enhanced FCoE security by preventing FCoE MAC spoofing
- Creates FC point-to-point links within the Ethernet LAN
- Allows auto-configuration of ACLs based on information in the FIP frames.

The following are the minimum third-party hardware interoperability requirements:

- Cisco Nexus 5020 FCF switch with firmware 4.1(3)N2(1x) or later, should have minimum of one 4G native FC module
- EMC CX4-120 or VMXe Storage unit with 4 Gig FC I/O Module

6.3.12. OpenDaylight Controller

OpenDaylight (ODL) is an open platform for network programmability to enable Software Defined Networking (SDN) and Network Functions Virtualization (NFV) for networks at any size and scale. ICOS supports interoperation with the OpenDaylight Controller version Helium-SR1.1.

6.3.13. Dynamic Topology Map and Prescriptive Topology Mapping

To easily identify ports where a network cabling error and/or other cabling complication (miswiring) has occurred, a CLI command can be used to light the LED for a single port or multiple ports and turn off all other port LEDs. The port-locator enable command is executed on individual interfaces.

In the case where a port has two LEDs, one for link and a second for activity, only the link LED is used for the port locator function. The activity LED will be turned off while the port locator is active. If a port has link and activity combined on a single LED, the LED will not blink if activity is present on the port, regardless of whether port-locator is enabled or disabled on the port.

The out-of-band port LED is not affected by this feature.

Prescriptive Topology Mapping (PTM) uses a topology file to verify the cabling on a switch. The topology file can be distributed either by Chef or Puppet, or can be provided manually to all the switches in the network to verify the entire topology. PTM relies on an open-source LLDP demon (LLDPD) to gather information about the partner switches and their links.

6.4. Layer 3—Routing Module

ICOS supports IPv4 unicast static and dynamic routing. The major layer 3 features are as follows:

- Section 6.4.1, “Routing Interfaces and Addressing”
- Section 6.4.2, “Address Resolution Protocol (ARP)”
- Section 6.4.3, “Unicast Routing Table”
- Section 6.4.4, “Policy-Based Routing”
- Section 6.4.5, “Software Forwarding”
- Section 6.4.6, “ICMP”
- Section 6.4.7, “Router Discovery Protocol”
- Section 6.4.8, “IP Helper”
- Section 6.4.9, “Interface Flap Dampening”
- Section 6.4.10, “Default Routes on Management Interfaces”
- Section 6.4.11, “Support for RFC 3021 Subnets”
- Section 6.4.12, “Virtual Router Redundancy Protocol (VRRP)”
- Section 6.4.13, “Static Routes”
- Section 6.4.14, “Dynamic routing protocols”:
  - Section 6.4.15, “Open Shortest Path First (OSPF)”
  - Section 6.4.16, “Border Gateway Protocol 4 (BGP4) Module”
- Section 6.4.17, “VRF Lite”
- Section 6.4.18, “BFD”
- Section 6.4.19, “MPLS Support”
- Section 6.4.20, “Local AS Support (Hide ASN)”
- Section 6.4.21, “RFC 5549”
- Section 6.4.22, “Algorithmic Longest Prefix Match (ALPM)”

6.4.1. Routing Interfaces and Addressing

ICOS supports three types of routing interfaces: port-based routing interfaces, VLAN routing interfaces, and loopback interfaces.

6.4.1.1. Port-based Routing Interfaces

A port-based routing interface is created by enabling routing on a physical port. That physical port is the only member of the routing interface. The status of the routing interface depends on whether the physical port is up or down. Port-based routing interfaces are often used for point-to-point connections between two routers.

A VLAN ID is internally assigned to each port-based routing interface. That VLAN ID cannot be assigned to other interfaces. Configuration options control which VLAN IDs are assigned to port-based routing interfaces. No VLAN IDs are reserved for port-based routing interfaces. If no port-based routing interfaces are created, the entire range of VLAN IDs is available for other uses.

6.4.1.2. VLAN Routing Interfaces

A VLAN routing interface is created by enabling routing on a VLAN. A VLAN routing interface includes all physical ports that are members of the VLAN. The routing interface is up if at least one of the member interfaces is up. A VLAN interface may include a LAG (or “port-channel”) as one of its members. Routing cannot be enabled directly on a LAG. A new slot/port ID is created for each VLAN routing interface. The port ID for each VLAN interface can be specified as part of the configuration. VLAN routing interfaces are often used to connect to LANs with multiple hosts.

6.4.1.3. Loopback Interfaces

A loopback interface is a layer 3 interface that does not depend on any physical interface. Because it does not depend on any physical interface, a loopback interface is always considered to be up. The IP addresses configured on loopback interfaces are commonly used as the target IP address for management or other remote connections to the router.

6.4.1.4. IP Addresses

A routing interface requires at least one IP address. Additional addresses, called secondary addresses, may also be configured. IP addresses may be assigned through manual configuration or using DHCP. Each IP address must be a valid unicast IP address. ICOS does not support network masks with length 31.

6.4.1.5. IP Maximum Transmission Unit (MTU)

The IP MTU is the maximum length of an IP packet that can be transmitted on a routing interface. The IP MTU on each routing interface defaults to the layer 2 (or “link”) MTU, minus the length of the Ethernet header, and must always be smaller than the link MTU. If the link MTU is changed, the IP MTU changes with it, unless the interface’s IP MTU is explicitly configured. ICOS supports jumbo frames; the IP MTU may be set as large as the largest jumbo frame, minus the size of the Ethernet header. Hardware typically enforces the IP MTU when routing a packet and enforces the layer 2 MTU when switching a packet.

6.4.1.6. Network Directed Broadcast

ICOS supports forwarding of network directed broadcast packets. A network-directed broadcast packet has a destination IP address whose host bits are all set to 1. The router adjacent to the subnet to which the packet is directed broadcasts the packet onto the subnet so that all stations on the subnet receive the packet. Network directed broadcasts are disabled by default but may be enabled on a per-interface basis.

6.4.1.7. Host Interfaces

When routing is globally disabled, the router does not forward packets between routing interfaces. Each routing interface acts as a host interface, able to receive packets whose final destination is the router itself, and able to send locally originated packets.

6.4.1.8. Unnumbered Interface Support for IPv4

ICOS supports the option to leave certain IP interfaces unnumbered for IPv4. Unnumbered interfaces conserve IPv4 address space and are useful in situations where adjacencies are transient, and adjacent interfaces cannot be easily configured with IPv4 addresses in the same subnet. OSPF adjacency can be formed over unnumbered interfaces by configuring OSPF point-to-point operation on the interfaces. Running OSPF in unnumbered point-to-point mode eliminates the need to configure neighbors to be in the same IPv4 subnet. Both iBGP and eBGP are supported for IPv4 over unnumbered interfaces.

The unnumbered IP interfaces are intended for router-to-router links and not for router-to-host links.

The ICOS implementation of unnumbered interfaces follows the standard industry practice and guidance described in RFC 5309.

6.4.2. Address Resolution Protocol (ARP)

ICOS uses ARP (RFC 826) to map IPv4 addresses to Ethernet MAC addresses. The IP stack implements ARP for out-of-band management interfaces, the network port and service port. ICOS application code implements ARP for routing interfaces. ICOS provides the following global configuration knobs for ARP on routing interfaces:

- The maximum age of an ARP entry, at which time it is either renewed or removed from the ARP cache
- The amount of time to wait for a response to an ARP Request
- The maximum number of ARP Requests to send when resolving an IP address
- The maximum number of entries that may be stored in the ARP cache
- Whether to unconditionally send an ARP Request to attempt to renew an entry when it reaches the maximum age
- Static ARP entries

ICOS sends a gratuitous ARP for each IP address on a routing interface when the interface comes up. The gratuitous ARP informs neighbors of the IP address-to-MAC mapping for the interface.

The ARP cache contains five types of entries: local, static, dynamic, gateway, and negative. A local entry is an entry for an IP address configured on the router. A static entry is a manually configured entry. Static entries do not age. A dynamic entry maps the IP address of a neighbor to its MAC address. A gateway entry is a dynamic entry whose IP address is the next hop address of one or more routes in the routing table. ICOS tries to retain gateway entries since the loss of a gateway entry may affect forwarding of IP packets to many destinations. A negative entry specifies an IP address but no MAC address and indicates that address resolution is in progress for the IP address. Negative entries are installed in the hardware like any other ARP cache entry. Hardware drops packets that match a negative entry to avoid flooding the CPU with data packets whose next hop cannot be resolved.

When the router forwards a data packet and the forwarding table specifies a next hop IP address that is not yet resolved to a MAC address, the hardware traps the data packet to the CPU, where the packet is forwarded in software. The packet triggers ARP resolution (i.e., the router sends an ARP Request for the packet's next hop IP address). ARP stores up to three data packets for each IP address being resolved, and forwards the packets if the next hop address is successfully resolved. The packets are discarded if the next hop address cannot be resolved.

Each ARP cache entry has an age. The age is the length of time since the router last confirmed resolution of the IP address. When an entry's age reaches the configured maximum age time (the default is 20 minutes), the ARP application may either delete the entry or try to renew it. If the application decides to try to renew the entry, it sends an ARP Request and waits for a response. If a response is received, the age is reset to 0. Otherwise, the entry is deleted. Local and static entries do not age. ARP always tries to renew gateway entries. Negative entries do not age in the same way, but are either converted to a dynamic or gateway entry, if resolution succeeds, or are deleted when the resolution attempt fails. Dynamic entries are not renewed by default, but a configuration option can be enabled to always renew them. If the dynamic renew configuration option is not enabled, then whether they are renewed depends on whether the entry has been used to forward data packets since its age was last reset to 0. If so, the ARP application attempts to renew the entry.

6.4.3. Unicast Routing Table

ICOS maintains a single unicast routing table. ICOS uses the routing table to forward unicast IP packets and respond to reverse path forwarding queries from IP multicast protocols. ICOS follows the typical industry practice of assigning each route a route preference, sometimes called administrative distance. Preference values are from 1 to 255. Default preferences for different types of routes can be configured. When there are multiple routes to the same destination, the routing table selects the route with the lowest preference value as the best route to the destination. A route with a preference of 255 cannot be selected as the best route. The routing table only installs best routes in the hardware forwarding table and the IP stack's routing table. The routing table has a fixed limit to the number of routes that can be installed. This limit should never be exceeded. Doing so risks routing loops and black holes.

The unicast routing table only contains routes associated with the routing interfaces. Local and default routes for the out-of-band management interfaces are not installed in this routing table.

Each route may have up to a fixed number of ECMP next hops. For each best route, all ECMP next hops are installed in the hardware forwarding table. The hardware uses a hash algorithm based on the packet's IP header to select a next hop when forwarding. Packets belonging to a given stream always hash to the same next hop to avoid delivering packets to the final destination out of order. Only a single next hop from each ECMP route is installed in the IP stack's routing table.

Applications within ICOS, such as the unicast routing protocols, can register to have the routing table notify them when the routing table changes or when the best route to a specific destination IP address changes. These facilities are used for route redistribution and next hop resolution (for example, when BGP has to resolve a BGP NEXT HOP to a local next hop).

6.4.4. Policy-Based Routing

To forward packets to destination addresses, routers typically make forwarding decision based on routing tables, which are populated by information given by dynamic routing protocols and/or static routing. Policy-based routing enables the network administrator to define forwarding behavior based on the contents of a packet. In brief, Policy Based Routing overrides traditional destination-based routing behavior. ICOS supports policybased routing for IPv4 only.

The ICOS policy based routing feature allows the administrator to define packet matches based on the following IPv4 packet attributes:

- Packet size
- Protocol of the payload (protocol field in IP header)
- Source MAC address
- Source IP address
- Destination IP address
- VLAN ID
- Priority (802.1P priority)

When a match is found, policy-based routing overrides the traditional forwarding behavior accomplished through destination-based routing.

6.4.5. Software Forwarding

Most unicast IP data packets are forwarded in hardware; however, there are exception cases that require forwarding in software. A common exception case is when a packet’s next hop IP address is not resolved to a MAC address. In these cases, the hardware traps the packet to the CPU. ICOS application software, not the IP stack, forwards these IPv4 packets. ICOS looks up the packet’s destination IP address in the unicast routing table, selects a next hop if the best route is an ECMP route, and looks up the next hop IP address in the ARP cache. If the next hop is not yet resolved, ARP sends an ARP Request and queues up to 3 packets to that destination until the ARP Reply is received. The ICOS software forwarding path does not fragment IPv4 packets larger than the IP MTU on the outgoing interface. ICOS drops such packets and sends an ICMP Destination Unreachable message to the source of the data packet.

IPv4 packets that originate on the router follow a different path. Applications typically send these packets on a socket through the IP stack. The IP stack uses its own routing table, which includes routes in the ICOS routing table, to forward these packets. Routing interfaces have the NO_ARP flag set; so the IP stack leaves the destination MAC address set to all zeros. ICOS fills in the destination MAC address using the ICOS ARP cache before transmitting the packet.

The IP stack’s routing table includes all routes from the ICOS unicast routing table and routes associated with the out-of-band management interfaces. The management routes include local routes for the IP addresses assigned to the management interfaces and possibly a default route. ICOS never installs more than one default route in the IP stack's routing table. If the ICOS unicast routing table has a default route and there is a default route on a management interface, the routing table default is installed in the IP stack.

6.4.6. ICMP

ICOS supports the Internet Control Message Protocol (ICMP) (RFC 792).

6.4.6.1. Echo Request/Reply

ICOS replies to Echo Request messages by default. ICOS offers a global configuration option to prevent the router from replying to Echo Requests.

6.4.6.2. Destination Unreachable

When the router cannot forward an IP packet, it sends an ICMP Destination Unreachable message to the sender. To protect the network from certain types of denial of service attack that trigger a large number of Destination Unreachable messages, the router limits the rate at which it sends Destination Unreachable messages. The rate limit is configurable and applies to the total number of messages sent on all routing interfaces. ICOS offers a configuration option to prevent the router from sending any Destination Unreachable messages on a specific routing interface.

6.4.6.3. Redirects

When the hardware forwards a packet on the routing interface where it was received, the hardware copies the data packet to the CPU. The software forwarding path sends an ICMP Redirect packet to the sender to tell the sender to forward future packets to a different router. ICOS offers configuration options to disable sending ICMP Redirect messages on a specific interface or on all interfaces.

6.4.7. Router Discovery Protocol

ICOS can be configured to send ICMP router discovery messages (RFC 1256). When this feature is enabled, the router periodically sends router advertisements messages and responds to router solicitation messages. Hosts receiving these router advertisements may use the router as a default gateway.

6.4.8. IP Helper

The IP Helper feature provides a mechanism that allows a router to forward certain configured UDP broadcast packets to a particular IP address. This allows various applications to reach servers on nonlocal subnets, even if the application was designed to assume a server is always on a local subnet and uses broadcast packets (with either the limited broadcast address 255.255.255.255, or a network directed broadcast address) to reach the server. IP helper provides the DHCP relay feature.

The network administrator can configure relay entries both globally and on routing interfaces. Each relay entry maps an ingress interface and destination UDP port number to a single IPv4 address (the helper address). The network administrator may configure multiple relay entries for the same interface and UDP port, in which case the relay agent relays matching packets to each server address. Interface configuration takes priority over global configuration. That is, if a packet's destination UDP port matches any entry on the ingress interface, the packet is handled according to the interface configuration. If the packet does not match any entry on the ingress interface, the packet is handled according to the global IP helper configuration.

The network administrator can configure discard relay entries, which direct the system to discard matching packets. Discard entries are used to discard packets received on a specific interface when those packets would otherwise be relayed according to a global relay entry. Discard relay entries may be configured on interfaces, but are not configured globally.

In addition to configuring the server addresses, the network administrator configures which UDP ports are forwarded. Certain UDP port numbers can be specified by name as a convenience, but the network administrator can configure a relay entry with any UDP port number. The network administrator may configure relay entries that do not specify a destination UDP port. The relay agent relays assumes these entries match packets with the UDP destination ports listed in Table below. This is the list of default ports.

Table 6.5. Default Ports—UDP Port Numbers Implied by Wildcard

Protocol	**UDP Port Number**
IEN-116 Name Service	42
DNS	53
NetBIOS Name Server	137
NetBIOS Datagram Server	138
TACACS Server	49
Time Service	37
DHCP	67
Trivial File Transfer Protocol	69

The relay agent relays DHCP packets in both directions. It relays broadcast packets from the client to one or more DHCP servers, and relays to the client packets that the DHCP server unicasts back to the relay agent. For other protocols, the relay agent only relays broadcast packets from the client to the server. Packets from the server back to the client are assumed to be unicast directly to the client. Because there is no relay in the return direction for protocols other than DHCP, the relay agent retains the source IP address from the original client packet. The relay agent uses a local IP address as the source IP address of relayed DHCP client packets.

When the relay agent receives a broadcast UDP packet on a routing interface, it checks if the interface is configured to relay the destination UDP port. If so, the relay agent unicasts the packet to the configured server IP addresses. Otherwise, the relay agent checks if there is a global configuration for the destination UDP port. If so, the relay agent unicasts the packet to the configured server IP addresses. Otherwise the packet is not relayed. Note that if the packet matches a discard relay entry on the ingress interface, then the packet is not forwarded, regardless of the global configuration.

The relay agent only relays packets that meet the following conditions:

- The destination MAC address must be the all-ones broadcast address (FF:FF:FF:FF:FF:FF).

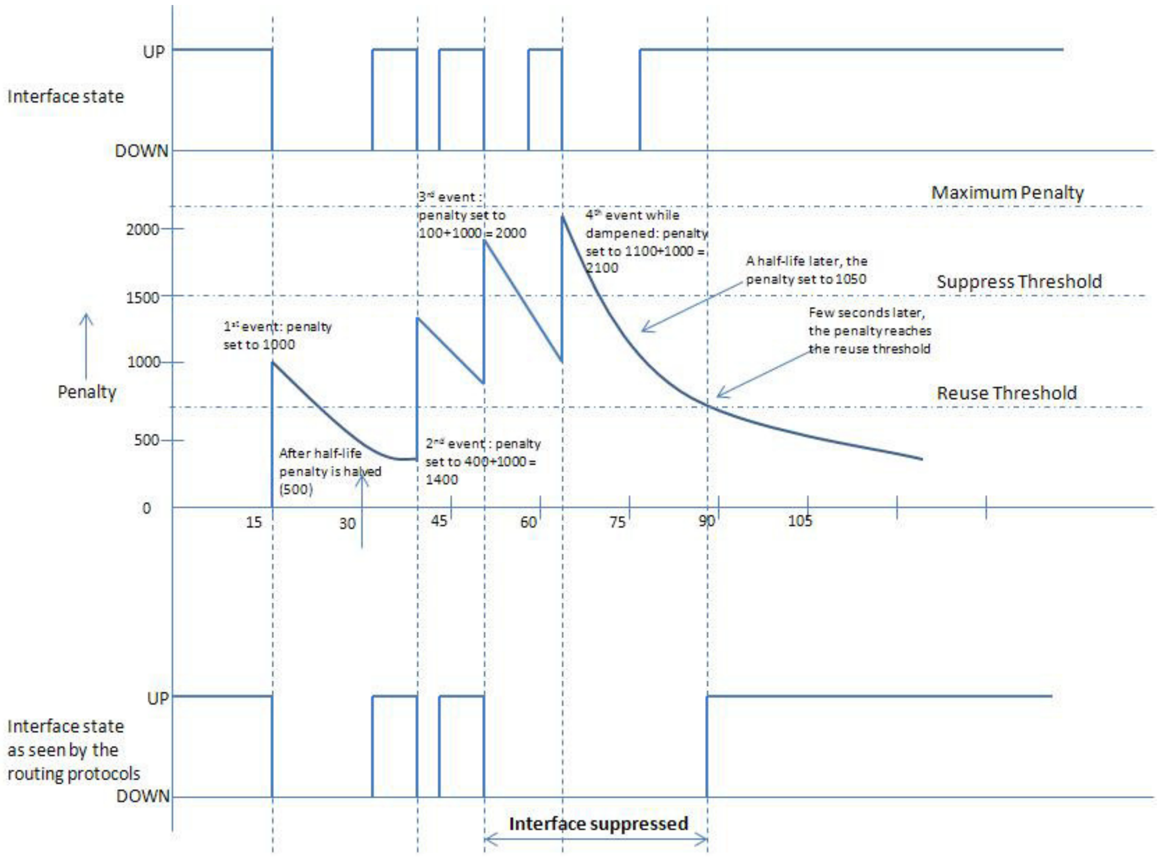
- The destination IP address must be the limited broadcast address (255.255.255.255) or a directed broadcast address for the receive interface.
- The IP time-to-live (TTL) must be greater than 1.
- The protocol field in the IP header must be UDP (17).
- The destination UDP port must match a configured relay entry.

6.4.9. Interface Flap Dampening

When an interface bounces rapidly, it can cause instability on the local system. The instability can propagate to other parts of the network through rapid advertisement of the interface status change. This feature rate limits up/down reports for an interface.

The IP Event Dampening project reduces the effect of interface flaps on routing protocols. The routing protocols temporarily disable their processing (on the unstable interface) until the interface becomes stable, thereby increasing the overall stability of the network.

Figure 6.10. Interface Dampening Example



The figure above displays the status of an interface when it flaps regularly. If dampening is enabled on an interface, a penalty (default 1000) is assigned to the interface when the link status of an interface changes from UP to DOWN. The penalty decays exponentially based on the following formula:

```
P(t) = P(0) * pow(2, –t/H)
* P(t): Penalty of the interface after t seconds
* P(0): Initial Penalty
* H: Half-life period
```

As time passes without another event occurring, the penalty is decreased, based on the half-life as shown in the formula. H is the half-life period—this is the amount of time it takes for the penalty to decrease by half.

The figure above starts at time 0, with a penalty of 0. When the first event (link status UP to DOWN) occurs, a fixed, nonconfigurable penalty of 1000 is assigned to the interface, making the total penalty 1000. As time passes without another event occurring, the penalty is decreased, based on the half-life period. Each time the half-life period (in this case 15 seconds) passes, the current penalty is halved. Therefore, after 15 seconds, the penalty is 500.

A few seconds later, while the penalty is still decreasing, a second event occurs: 1000 is added to the current penalty, making the total penalty 1400. Again, as time passes, the penalty decays exponentially, reaching 1000 before the third event occurs. When the third event occurs, 1000 is again added to the total penalty. It reaches 2000, which is above the suppress threshold (in this case 1500); therefore, future events are dampened by leaving the interface in the suppressed state. At this point, the routing protocols see the interface status as DOWN even if the NIM has reported the interface active (as it is in suppressed state).

Again, as time passes, the penalty is cut in half for each passing half-life period, reaching 1100 before the fourth event occurs. When the fourth event occurs, 1000 is again added, making the penalty 2100 and leaving the interface in the suppressed state. Over time, the penalty finally drops to 750 (at around 90 seconds in the example), which is the reuse threshold. At this point, IP MAP sends a routing UP event to the routing protocols.

6.4.10. Default Routes on Management Interfaces

If the router learns the default route on a routing interface, the IP stack cannot route a packet to an IP address normally reached using the default route on either the service port or network port because only the default route on the routing interface is installed in the IP stack routing table.

The default route learned on the routing interface is preferred over the default route learned on the management interface.

ICOS supports multiple routing tables to allow the installation of multiple default routes. ICOS uses the Linux policy routing feature to overcome the management issue. In addition to the default routing table in the Linux IP stack, separate routing tables are created for the network port (if included in the build) and the service port (if included in the build). These tables are used for outgoing packets whose source IP address is the same as the IP address of either the service port or the network port.

The source IP address of the packet can be used to select the routing table.

- If the packet is sent on a TCP connection, and the TCP connection was initiated remotely, then the packet's source IP address is always the IP address that was the target of the TCP connection. For example, if a user access the router via SSH at 10.1.1.1, then packets sent back to the SSH client will have a source IP address of 10.1.1.1.

- For some management traffic (such as ping and traceroute), ICOS gives administrator the ability to configure the source IP address. When the administrator configures the source address, ICOS binds the socket used to transmit packets to that source address. The bind operation sets the source address prior to the route lookup.

Even though there are separate routing tables for the default route on the network and service ports, the default route selected by ICOS is still installed in the default Linux routing table, even if the default route is associated with the network or service port. This is important for maintaining compatibility with behavior in previous releases, even when packets are sent without consciously setting the source IP address.

With the new routing tables created for the service port and the network port, ICOS enforces the logic mentioned below:

- if the outgoing packet’s source IP address is an address on the service port:
  - Use the routing table associated with the service port.
- if the outgoing packet’s source IP address is an address on the network port:
  - Use the routing table associated with the network port.
- otherwise
  - Use the default routing table.

ICOS sends one netlink message to the kernel to create the network port routing table. ICOS sends another netlink message to the kernel to add the default route into the network port’s routing table. The network port route table is deleted when the network port’s default route associated with the network port is either unconfigured or expired (DHCP).

Similar to the network port routing table, a routing table is created for the service port.

6.4.11. Support for RFC 3021 Subnets

ICOS supports RFC 3021, Using 31-Bit Prefixes on IPv4 Point-to-Point Links. Using 31-bit prefixes reduces the IP address space required to number a network with point-to-point links.

6.4.11.1. Impact on Routing Protocols

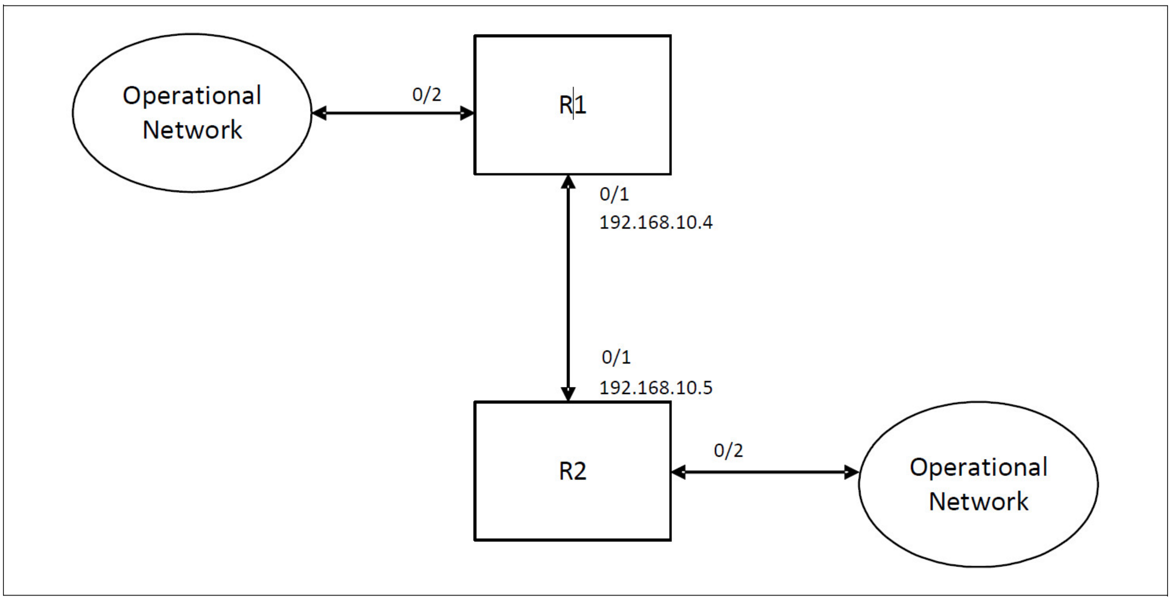
The 31-bit prefixes have no impact on (unicast and multicast) routing protocols and other Layer-3-based protocols supported by ICOS (viz., BGP, OSPF, RIP, PIM etc.,) as all of them are classless routing protocols. Furthermore, only unicast and multicast are used for the communication between them. Directed broadcast is not used.

6.4.11.2. Usage Scenarios

Configuring a link with a /31 subnet is helpful in the following situations:

In the figure below, routers R1 and R2 have two routing interfaces, 0/1 and 0/2. Routers R1 and R2 are connected (on interface 0/1) via a point-to-point link with a 31-bit subnet mask.

Figure 6.11. Example Network RFC3021 Subnets



The following shows the configuration of router R1 with a 31-bit subnet mask:

```
configure
  interface 0/1
    routing
    ip address 192.168.10.4 255.255.255.254
  exit
exit
```

The following shows the configuration of router R2 with a 31-bit subnet mask:

```
configure
  interface 0/1
    routing
    ip address 192.168.10.5 255.255.255.254
  exit
exit
```

6.4.11.3. Usage Considerations

1. Configuration of a 31-bit prefix on non point-to-point links should be used cautiously and may not provide the desired functionality when the non-point-to-point links have more than two hosts.

2. Use of 31-bit prefixes may not operate correctly on a point-to-point link in which only one end supports 31-bit prefixes.

3. Configuration of a 31-bit prefix is allowed for both primary and any secondary addresses of an interface and also for a loopback interface.

Another popular implementation prints a warning message when the user configures a /31 address:

```
Switch(config-if)#ip address 172.20.101.1 255.255.255.254
```

```
% Warning: use /31 mask on non point-to-point interface cautiously
```

ICOS does not print a warning message, as this feature is now mature and the message is unnecessary.

6.4.12. Virtual Router Redundancy Protocol (VRRP)

ICOS provides an implementation of VRRP (RFC 3768). ICOS supports multiple virtual routers on a routing interface.

6.4.12.1. Ping

RFC 3768 specifies that a router may only accept IP packets sent to the virtual router's IP address if the router is the address owner (that is, if the IP address used as the virtual router's IP address is an address configured on the local routing interface). In practice, this restriction makes it more difficult to troubleshoot network connectivity problems. When a host cannot communicate, it is common to ping (send an ICMP Echo Request) the host's default gateway to determine whether the problem is in the first hop of the path to the destination. When the default gateway is a virtual router that does not respond to pings, the administrator cannot use this troubleshooting technique. Because of this, it has been common for VRRP implementations to respond to pings, in spite of the prohibition in the RFC.

In ICOS, a VRRP virtual router can be configured to respond to Echo Requests. The VRRP Master responds to Echo Requests sent to the virtual router's primary address or any of its secondary addresses. Ping to a VRRP IP address only works from the host side (where the virtual router is configured). When the VRRP master responds with an Echo Reply, the source IPv4 address is the VRRP address and source MAC address is the virtual router's MAC address.

6.4.12.2. Route and Interface Tracking

Because the VRRP master serves as the default gateway for hosts on its LAN, it is important that the master router is connected to the rest of the network. By default, VRRP selects a master based on configured priorities, but this selection method does not take into account whether the router can reach the rest of the network.

The ICOS implementation of VRRP can track specific routes in the unicast routing table, or track specific routing interfaces. The presence of a tracked route or the state of a tracked routing interface adjusts the router's priority to become the VRRP master.

6.4.13. Static Routes

The network administrator can configure static unicast routes. Static routes with the same destination prefix and route preference are combined into an ECMP route. ICOS only installs a static route in the routing table if the next hop IP address is on a local subnet and the routing interface to that subnet is up. ICOS allows configuration of static reject routes. Packets that match a reject route are discarded.

6.4.14. Dynamic routing protocols

6.4.15. Open Shortest Path First (OSPF)

The ICOS implementation of OSPF supports the following:

- OSPF version 2 (RFC 2328) with the exception of nonbroadcast multi-access (NBMA) interfaces or pointto- multipoint interfaces.
- Not-so-stubby areas (RFC 3101) and external LSA overflow (RFC 1765).
- The flooding of opaque LSAs (RFC 2370), but provides no facilities for originating or parsing them.
- Stub router advertisement (RFC 3137). In addition to automatically becoming a stub router when a resource limitation prevents OSPF from computing a complete routing table, the network administrator may manually configure OSPF to act as a stub router. OSPF may also be configured to start up in stub router mode, only advertising normal metrics after a configurable start-up period.
- OSPF passive interfaces and point-to-point operation over Ethernet. Area ranges may be configured with a static cost. Configurable LSA transmit pacing reduces the chance of overrunning the receive buffers of neighbors. Configurable LSA pacing groups spread self-originated LSAs over the refresh interval to avoid periodic flooding spikes. LSA flooding can be disabled on specific interfaces to reduce flooding in highlymeshed networks. OSPF supports ECMP with a configurable maximum number of next hops. Delay and hold timers may be configured to prevent the shortest path algorithm from running so often that OSPF consumes too much CPU time.

OSPF can redistribute connected, static, and BGP routes. A distribute list can be configured to filter redistributed routes. A router can be configured to originate a default route. Default origination is optionally conditioned on the presence of a non-OSPF default route in the unicast routing table.

- The standard OSPF MIB as specified in RFC 1850 as well as additional proprietary MIB variables.
- RFC 6860 for hiding the transit-only networks in OSPF. A transit-only network is defined as a network connecting only routers. Hiding transit-only networks can speed up network convergence and reduce vulnerability to remote attacks to the routers in the transit-only network.

6.4.15.1. Automatic Exiting of Stub Router Mode

OSPF enters stub router mode in order to inform other routers that a router should not be used as a transit point. It does this by setting the metric for transit links in the router LSA to the maximum and re-originating the router LSA as specified in RFC 3137. Stub router mode may be entered either automatically (due to a lack of resources) or by configuration. Stub router mode can be configured to always be in effect or to be in effect only for a set period after startup. When stub router mode is entered due to lack of resources, OSPF tries to automatically exit stub router mode. Once every 60 seconds, OSPF checks various resource constraints, and if sufficient resources are available (i.e., they are all below the thresholds), OSPF exits stub router mode by reoriginating the router LSA with proper metric values on transit links.

6.4.15.2. OSPF Equal Cost Multipath (ECMP)

A device running the IP routing protocol OSPF maintains multiple equal-cost routes to all destinations. The multiple routes are of the same type (intra-area, inter-area, type 1 external or type 2 external), cost, and have the same associated area. However, each route is defined by a separate advertising router and next hop.

With ECMP, a device forwards traffic to a specified destination through multiple paths thereby taking advantage of the bandwidth of both links.

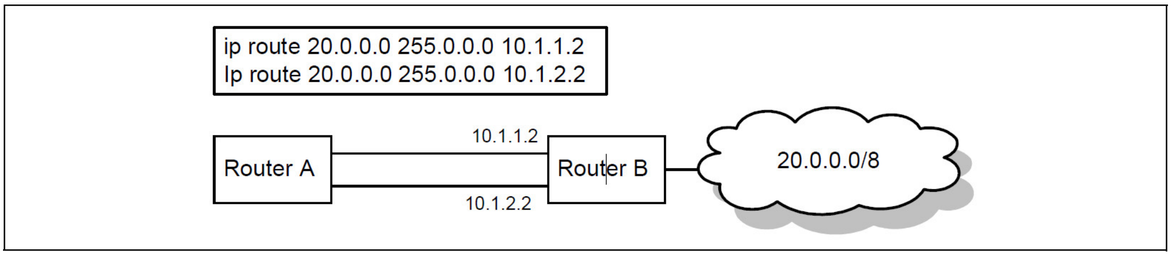
ECMP routes are configured statically or learned dynamically as follows:

- Configured Statically: If an operator configures multiple static routes to the exact same destination but with different next hops, those routes are treated as a single route with two next hops.
- Learned Dynamically: Routing protocols can learn ECMP routes. For example, in the figure below, OSPF is configured on both links connecting Router A and Router B. Router B advertises its connection to 20.0.0.0/8, then Router A computes an OSPF route to 20.0.0.0/8 with next hops of 10.1.1.2 and 10.1.2.2.

ICOS software stores static and dynamic routes in a single combined routing table. In ICOS software, this routing table is referred to as RTO. RTO accepts ECMP routes, but it is important to understand that RTO does not combine routes from different sources to create ECMP routes. Referring to the figure below, assume OSPF is only configured on one of the links between Router A and Router B. Then on Router A, OSPF reports to RTO a route to 20.0.0.0/8 with a next hop of 10.1.1.2. If the user configures a static route to 20.0.0.0/ 8 with a single next hop of 10.1.2.2, RTO does NOT combine the OSPF and static route into a single route to 20.0.0.0/8 with two next hops. All next hops within an ECMP route must be provided by the same source.

The maximum number of next hops reported by OSPF is configurable. Global enable or disable of ECMP for all route sources is unsupported.

Figure 6.12. A Static Route with Two Next Hops



```
ip route 20.0.0.0 255.0.0.0 10.1.1.2
Ip route 20.0.0.0 255.0.0.0 10.1.2.2
```

On Netberg hardware platforms based on Broadcom Tomahawk, the ECMP hashing support is extended to Enhanced hashing mode, which provides improved load-balancing performance. ECMP hashing on these platforms has the following features:

- MODULO-N operation based on the number N of next hops in the route.
- Packet attributes selection based on the packet type. For IP packets, the following fields are used: Source IP address, Destination IP address, TCP/UDP port, Pv4 Protocol, IPv6 Next Header.

6.4.15.3. Optimization to Support ECMP Hops in Large-Scale Networks

Data center platforms are required to support an increasing number of ECMP next hops in a route, which has a considerable scaling impact in terms of memory and CPU processing. To address this, the ICOS OSPFv2 implementation optimizes memory usage and processing time in large-scale networks to enable a large number of next hops.

To support a large number of next hops, the OSPF implementation has been optimized as follows:

- Next hop information is stored in an OSPF route in an optimized format so that the memory usage from the routing heap is unaffected by the number of ECMP next hops supported per route.
- Transmit pacing is implemented on flooded LSAs, which means that LSAs can be flooded in an optimized manner to the OSPFv2 neighbors. Before the transmit pacing timer fires, the software keeps track of the neighbors from which the same LSA instance has been received and suppress flooding to all those neighbors.

6.4.15.4. ECMP Hash Selection

Users can choose the load balancing/sharing algorithm used for selecting the final ECMP route. The CLI enables choosing various combinations of IP header fields, including the inner or outer IP headers in tunneled packets. Both IPv4 and IPv6 are supported. The field selectors remain the same for all packet types. The following is a list of available hash field selection algorithms. The list may vary depending upon platform.

- Source IP address of the packet.
- Destination IP address of the packet.
- Source and Destination IP address of the packet.
- Source IP address and Source TCP/UDP Port field associated with the packet.
- Destination IP address and Destination TCP/UDP Port field associated with the packet.
- Source, Destination IP address and Source, Destination TCP/UDP Port field associated with the packet.

For tunneled packets, the user also must select whether the inner or the outer IP header should be used.

6.4.16. Border Gateway Protocol 4 (BGP4) Module

The Border Gateway Protocol (BGP) is an inter-autonomous system (AS) routing protocol. The primary function of a BGP system is to exchange network reachability information with other BGP systems. This information is sufficient to construct a graph of AS connectivity from which routing loops may be pruned.

BGP can be configured only through the CLI. SNMP support is limited to the standard MIB, which primarily provides status reporting, and a proprietary MIB which provides additional status variables. Configuration through SNMP or the web interface is not supported.

BGP may learn the maximum number of routes supported by each platform, up to 16k routes.

See RFC 4271 section 3 for a general overview of BGP. ICOS does not support any version of BGP other than version 4.

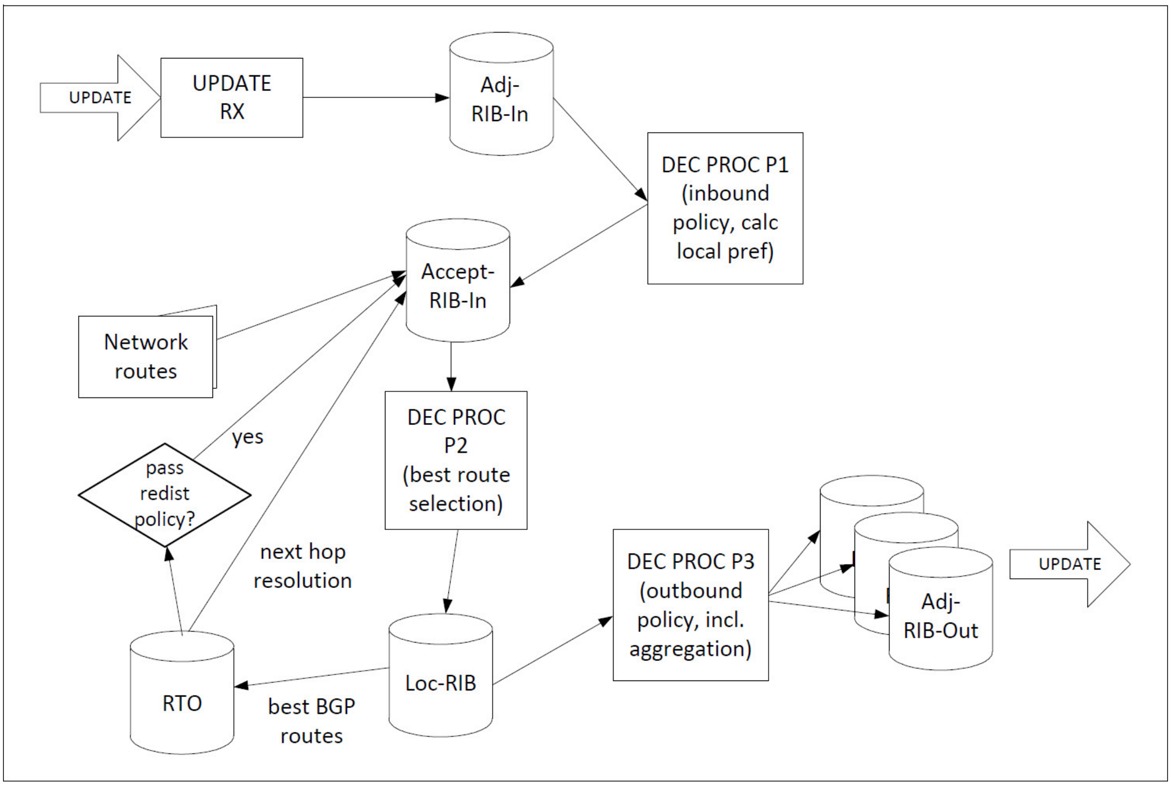
ICOS BGP supports only IPv4 and IPv6 unicast routes. Both IPv4 and IPv6 peering are supported. IPv4 routes may be carried over IPv4 peering sessions. IPv6 routes may be carried over IPv4 or IPv6 peering sessions. The only optional parameters recognized in an OPEN message are the Capabilities option (RFC 5492) and the multiprotocol capabilities option (RFC 4760). If a neighbor includes the deprecated authentication parameter in its OPEN, ICOS BGP rejects the OPEN and will not form an adjacency.

The network operator can configure a maximum number of prefixes to accept from a peer. The limit defaults to the maximum number of routes that can be installed in the hardware forwarding table. When the limit is reached, the behavior is a configurable option; by default, BGP shuts down the peer. BGP may be configured to instead discard new address prefixes but not terminate the peer (RFC 4271 section 6.7).

6.4.16.1. Decision Process Overview

The BGP decision process is logic that applies inbound policy to routing information from peers, computes routes, and advertises routes to peers. The figure below shows an overview of the decision process. BGP parses incoming UPDATE messages, storing routing information in the routing information base (RIB) named Adj-RIB-In. Phase 1 of the decision process applies inbound policy to routes in Adj-RIB-In. Routes that pass inbound policy are copied to Accept-RIB-In and LOCAL_PREF is set. BGP uses the routing table object (RTO) to resolve the BGP next hop to a local next hop. Locally originated routes (those configured with the network command or redistributed from another protocol) go directly to Accept-RIB-In. Phase 2 of the decision process selects the best route to each destination in Accept-RIB-In. Each best route is stored in the local RIB and given to RTO. Phase 3 of the decision process applies outbound policy to routes in the local RIB and determines the status of aggregate routes. Active aggregates and individual routes that pass outbound policy are placed in an Adj-RIB-Out specific to each update group, and UPDATE messages are sent to communicate the routes to neighbors.

Figure 6.13. Decision Process



Route Selection ICOS BGP uses the following route selection rules (see Section 6.4.16.2, “Path Attributes” for a descriptions of the items mentioned here):

1. Prefer the route with the higher local preference.

2. Prefer a locally originated route over a non-locally originated route.

3. Prefer the route with the shorter AS Path.

4. Prefer the route with the lower ORIGIN. (IGP) is better than (EGP), which is better than INCOMPLETE.

5. Prefer the route with the lower Multi-Exit Discriminator (MED) attribute. By default, MEDs are compared only for routes from the same AS, but a configuration option allows comparison of MEDs from different ASs. A route with no MED is considered to have a MED of 0.

6. Prefer an eBGP route to an iBGP route.

7. Prefer the route with the lower IGP cost to the BGP NEXT HOP.

8. Prefer the route learned from the peer with the lower router ID.

9. Prefer the route learned from the peer with the lower peer IP address.

6.4.16.2. Path Attributes

ICOS supports all path attributes described in RFC 4271.

ICOS BGP sets the ORIGIN path attribute to IGP for routes originated through the network command, and to INCOMPLETE for routes originated through route redistribution. ICOS BGP never sets the ORIGIN path attribute to EGP.

ICOS BGP sets the AS_PATH path attribute in compliance with RFC 4271. ICOS BGP does require that paths from external peers include the configured AS number of the peer as the first AS in the path. ICOS BGP enforces a configurable limit to the length of the AS_PATH attribute in received paths. Paths that exceed the limit are discarded.

ICOS BGP offers a configuration option (neighbor next-hop-self) to set the NEXT_HOP attribute to a local IP address when sending an UPDATE message to an internal peer. Otherwise, ICOS BGP follows the guidance in RFC 4271 when sending to internal peers. When sending an UPDATE message to an external peer, ICOS BGP retains the NEXT_HOP address if it is an address on the subnet used to connect the peers but is not the peer’s IP address. Otherwise, ICOS BGP sets the NEXT_HOP path attribute to the local IP address on the interface to the peer. ICOS BGP does not support “first party” next hop. ICOS does not allow the network operator to disable third party next hop. Additionally, ICOS does not support multihop EBGP (RFC 4271 section 5.1.3).

The Multi-Exit Discriminator (MED) attribute is sent to external peers when a router advertises routes that originate within the local AS. The MED value may be configured for redistributed routes, either using the metric option of the redistribution command or by configuring a default-metric. If a MED attribute is not configured for a redistributed route, the route is advertised without a MED attribute. Routes originated through the network command set the MED attribute to the metric of the IGP route to the same network. The MED may also be set on locally originated routes using a route map. The MED for non-locally originated routes is propagated to internal peers. By default, MEDs are compared only when two routes are received from external peers in the same AS. There is a configuration option to force BGP to compare MEDs for paths received from different ASs.

When BGP receives an UPDATE message from an external peer, it assigns a local preference value during phase 1 of the decision process. Local preference is set to a fixed, configured value, which is the same for all paths received from all neighbors. This value is attached to the path in the LOCAL_PREF path attribute when the path is advertised to internal peers. The configured default local preference is assigned to all locally originated routes and to the paths for all active aggregate addresses. LOCAL_PREF can be configured to different values on different routers to influence the exit point from the AS that other routers select for each destination. An inbound route map can override the default local preference. LOCAL_PREF is never included in paths sent to external peers. If the user changes the default local preference while BGP is running, BGP automatically initiates an immediate soft inbound reset for all external peers, updates the local preference for all locally originated routes, and recomputes routes.

For each aggregate address configured, the network administrator may specify whether to advertise an AS_SET of the AS numbers in the paths from which the aggregate was formed. When the aggregate is advertised with an empty AS Path, the ATOMIC_AGGREGATE path attribute is attached to the path. In either case, the AGGREGATOR path attribute is attached.

6.4.16.3. BGP Finite State Machine (FSM)

ICOS BGP supports all mandatory FSM session attributes and the following optional session attributes (RFC 4271 section 8):

- **AllowAutomaticStart**: Connections are automatically restarted after an error closes a connection. An adjacency to an external peer in the IDLE state is automatically started if the routing interface to that peer comes up. An adjacency to an internal peer in the IDLE state is automatically started when the peer’s IP address becomes reachable.

- **AllowAutomaticStop**: When a neighbor sends more prefixes than the configured limit, the connection may be automatically shut down, depending on configuration. AutoStop is also used for fast failover. When the routing interface to an external peer goes down, the peering session is automatically stopped. Similarly, if an internal peer becomes unreachable, the peering session is automatically stopped.

- **CollisionDetectEstablishedState**: When an OPEN message is received on a TCP connection and the adjacency using that connection has already reached the ESTABLISHED state, the adjacency is cleared. If an OPEN message is received on a different TCP connection than the one used to reach the ESTABLISHED state, the new TCP connection is cleared and the adjacency remains up.

- **DampPeerOscillations**: An idle hold time is enforced between automatic restarts. The length of the idle time depends on the reason the adjacency entered the idle state.

- **IdleHoldTime/IdleHoldTimer**: After an error clears a connection or a TCP connection fails, ICOS BGP waits before attempting to reestablish the adjacency. The waiting time varies depending on the event. When a TCP connection fails, BGP waits 30 to 60 seconds. When a NOTIFICATION is received, BGP waits 1 to 2 seconds. Other events trigger a wait of 10 to 20 seconds. The delay time is not configurable.

- **SendNOTIFICATIONWithoutOPEN**: ICOS will accept a NOTIFICATION packet from a peer that has not first sent an OPEN packet. ICOS will not send a NOTIFICATION without first sending an OPEN.

None of the optional session attributes are configurable.

ICOS supports the manual start and stop events. A manual start event occurs when the user first configures a peer (*neighbor remote-as*) or administratively enables a peer (*no neighbor shutdown*). A manual stop event occurs when the user administratively disables a neighbor (*neighbor shutdown*).

Of the optional events in RFC 4271 section 8.1.2 - 8.1.5, only the following events are supported:

- AutomaticStart_with_DampPeerOscillations (Event 6)

- AutomaticStop (Event 8)

- IdleHoldTimer_Expires (Event 13)

ICOS BGP allows multiple BGP sessions between the same two routers. Each session must be established on a different pair of IP addresses.

ICOS BGP includes two capabilities in every OPEN message it sends. The first is the Route Refresh capability described in RFC 2918. The second is the multiprotocol capability described in RFC 4760. ICOS always advertises the IPv4/unicast AFI/SAFI pair. If the user has activated IPv6 for the peer, the OPEN message also includes the IPv6/unicast pair. Although ICOS BGP does not support any AFI/SAFI pairs other than IPv4/unicast when IPv6 is not enabled, ICOS advertises the multiprotocol capability with IPv4/unicast because some other implementations appear to require this to establish an adjacency.

6.4.16.4. Detecting Loss of Adjacency

ICOS optionally drops an adjacency with an external peer when the routing interface to that peer goes down. This behavior can be enabled globally or on specific interfaces using the *bgp fast-external-failover* and *ip bgp fast-external-failover* commands. BGP accomplishes this behavior by listening to router events. When BGP gets a routing interface down event, BGP drops the adjacency with all external peers whose IPv4 address is in one of the subnets on the failed interface.

ICOS also offers an option to quickly detect loss of reachability to internal peers and drop the BGP adjacency when such a loss occurs. Because internal peers are often not on a local subnet (and an internal peer can be reached through multiple local interfaces), BGP cannot determine internal peer reachability based on the local link state. Instead, when this feature is enabled, BGP registers for address resolution changes for the IP address of each internal peer. When a peer’s address becomes unreachable (i.e., the route table manager deletes the route to the peer and no non-default route to the peer remains), BGP drops the adjacency to the peer. BGP considers an internal peer to be unreachable if the only route to the peer is a default route. This feature is enabled or disabled globally for all internal peers using *bgp fast-internal-failover*. Because internal peers are not associated with a single interface, there is no interface configuration option.

When fast failover is not enabled for a peer, the adjacency remains in the ESTABLISHED state until the hold timer expires. When connectivity to the peer is lost, the BGP Next Hop for routes learned from affected peers becomes unreachable. This change makes the routes unusable, and

BGP immediately removes them from the routing table. So even without the fast failover behavior enabled, the routing table reacts quickly to changes in local interface state. However, when the adjacency remains in ESTABLISHED state even though the neighbor is unreachable, BGP cannot send UPDATE messages to the neighbor. If the link is restored before the dead interval expires, there is no event to cause BGP to resend the failed UPDATEs. Because BGP does not periodically refresh routing state, the loss is permanent. To avoid this situation, when an UPDATE message fails to be sent to any member of an outbound update group, BGP reschedules the update send process to resend the data. Thus, having a neighbor in an ESTABLISHED but unreachable state causes duplicate data to be sent to other members of the update group. With fast failover enabled, the adjacency to the unreachable neighbor is no longer ESTABLISHED, and if an UPDATE is sent to the neighbor's update group, BGP does not try to send to the failed neighbor. When the failed adjacency is reestablished, BGP resends all routing information to the neighbor.

Both internal and external failover should happen within a second of the loss of reachability. Enabling fast failover should eliminate the need to set a short hold time and send KEEPALIVE messages rapidly.

Fast failover is enabled by default.

6.4.16.5. Authentication

RFC 4271 requires support for TCP MD5 authentication as specified in RFC 2385. ICOS supports TCP MD5 authentication. The network administrator may optionally enable TCP MD5 for a specific peering session by configuring a password on each end of the connection.

6.4.16.6. Outbound Update Groups

To reduce the memory required for the Adj-RIB-Out and to reduce the processing required by the phase 3 decision process, BGP sorts peers into update groups. Every peer in an update group has the same configured (or default) value for minRouteAdvertisementInterval and the same set of outbound policies. Each update group contains only internal or external peers. Thus, the same information is advertised to every peer in the update group and may be advertised at the same time. A single advertised path list (Adj-RIB-Out) is retained for each update group. A single UPDATE message is constructed and a copy sent to each peer in the update group. When a peer in the ESTABLISHED state moves from one update group to another because of a configuration change, BGP withdraws all prefixes previously advertised to the peer and advertises to the peer the Adj-RIBOut of the new update group.

BGP maintains separate update groups for IPv4 and IPv6. If IPv6 is active for a peer with an IPv4 address, the peer is in both an IPv4 update group and in an IPv6 update group. A neighbor may be in an IPv6 update group for an IPv4 peer session (if the network administrator activates IPv6 on the peer session) and in an IPv6 update group for an IPv6 peer session. Such a configuration is probably a misconfiguration. BGP will send IPv6 Network Layer Reachability Information (NLRI) to the neighbor twice.

BGP assigns peers to update groups automatically. The ICOS CLI has no configuration associated with update groups; however, the user interface reports update group membership.

6.4.16.7. Removing Private AS Numbers

BGP can be configured to remove private AS numbers from the AS_PATH attribute of paths advertised to external peers. An organization can use private AS numbers internally. Private AS numbers must be removed from routes to destinations within private ASs before the routes are advertised in the Internet.

Two-byte ASNs in the range 64,512 to 65,535 are removed when this option is enabled. The administrator can optionally configure BGP to replace private ASNs with the local AS number. The replace option maintains the original length of an AS path, which can be important when the AS path length is used in route selection. The option to remove or replace private ASNs can be configured independently for each address family.

6.4.16.8. Templates

ICOS supports configuration of neighbor parameters in named peer templates. A template defines a set of peer parameters. Multiple peers can inherit parameters from a template, eliminating the need to repeat common configuration for every peer. A neighbor can inherit from a single template. BGP accepts configuration of up to 32 templates.

Neighbor configuration parameters can be divided into two groups, session parameters and policy parameters. Session parameters apply to the peering session. Session parameters include configuration options such as keepalive and hold timers. Policy parameters are specific to the routes for an address family (e.g., IPv4 and IPv6), such as the maximum number of routes accepted from a peer or prefix lists used to filter routes received from or sent to a peer. Peer templates allow both session parameters and policies to be configured within the same template. With a template, policy parameters are configured for a specific address family.

Session parameters that may be configured in a template are as follows:

- Neighbor connect retry interval
- Neighbor description
- Neighbor TCP MD5 password
- Neighbor administrative state (active or shutdown)
- Keepalive and hold timers
- Source IP address in UPDATE messages

Policy parameters that may be configured per address family within a template are as follows:

- Advertisement interval
- Neighbor-specific default route origination
- Neighbor-specific prefix filters for send and receive
- Neighbor-specific AS path filters for send and receive
- Neighbor-specific route map for send and receive
- Maximum prefixes accepted from a neighbor
- Next-hop-self option
- Whether or not to include the communities attribute in UPDATEs send to the neighbor

6.4.16.9. Resolving Interface Routes

In ICOS software, the next hop of a route is always a set of next hop IP addresses. ICOS does not support routes whose next hop is simply an interface. Thus, the second route resolvability condition in RFC 4271 section 9.1.2.1 does not apply.

6.4.16.10. Originating BGP Routes

A router running ICOS BGP can originate a BGP route through route redistribution and through configuration (the network command). Attributes of locally originated routes may be set through a route map. Locally originated BGP routes are sent to both internal and external peers unless filtered by outbound policy.

Locally originated routes are added to Accept-RIB-In. Phase 2 of the decision process considers locally originated routes along with routes received from peers when selecting the best BGP route to each destination.

The user may configure BGP to originate the same prefix through a network command and through redistribution. ICOS BGP creates a different path for each if the path attributes differ. BGP only advertises the prefix with the preferred path.

RFC 4271 section 9.2.1.2 specifies “a minimum amount of time that must elapse between successive advertisements of UPDATE messages that report changes within the advertising BGP speaker’s own autonomous systems” and refers to this as minASOriginationInterval. RFC 4271 section 10 suggests a default of 15 seconds. ICOS BGP does not enforce minASOriginationInterval, but relies on minRouteAdvertisementInterval, which is applied to all advertisements, to dampen flaps of locally originated routes. Delay and hold timers limit how often phase 2 of the decision process runs. This phase 2 dampening limits route origination, as does IP event dampening when interface flaps would otherwise cause rapid origination.

BGP originates a default route to all neighbors if the *default-information originate* command is entered and the default route is among the routes BGP redistributes. Because this default origination depends on redistribution, BGP normally originates a default only if a default is in the routing table. The *always* option can be configured to eliminate this requirement. If the routing table does not contain a default route, but the network administrator wants BGP to originate a default route, the administrator can configure a static default route. To prevent the static default route from affecting the local router’s forwarding, the default route can be given a preference of 255 (*ip route 0.0.0.0 0.0.0.0 next-hop 255*) or it can be configured as a reject route (*ip route 0.0.0.0 0.0.0.0 Null0*).

BGP can also originate a default to a specific neighbor using *neighbor default-originate*. This form of default origination does not install a default route in the BGP routing table (it will not appear in *show ip bgp*), nor does it install a default route in the Adj-RIB-Out for the update group of peers so configured (it will not appear in *show ip bgp neighbor advertised-routes*). A neighbor specific default has no MED and the Origin is IGP. A neighbor specific default is only advertised if the Adj-RIB-Out does not include a default learned by other means, either from *default-information originate* or a default learned from a peer. This type of default origination is not conditioned on the presence of a default route in the routing table.

6.4.16.11. Equal Cost Multipath (ECMP)

By default, ICOS BGP selects a single next hop for each BGP route. ICOS BGP can be configured to install BGP routes with up to 32 next hops in the common routing table (RTO). (Some hardware platforms may limit the number of next hops to fewer than 32.) The network administrator can independently configure the maximum number of next hops for routes through internal and external peers.

Paths can be used to form an ECMP route when they are both internal or both external, the resolved next hop is different, and the following attributes are the same:

- Local preference
- AS path length
- Origin
- MED
- IGP distance to the BGP next hop

ICOS BGP does not require ECMP next hops to be in a common AS. This behavior is the same as Cisco ECMP route selection when the hidden Cisco command bgp bestpath as-path multipath-relax is configured.

When advertising to neighbors, BGP always advertises the single best path to each destination prefix, even if BGP has an ECMP route to a destination.

6.4.16.12. BGP Next Hop Resolution

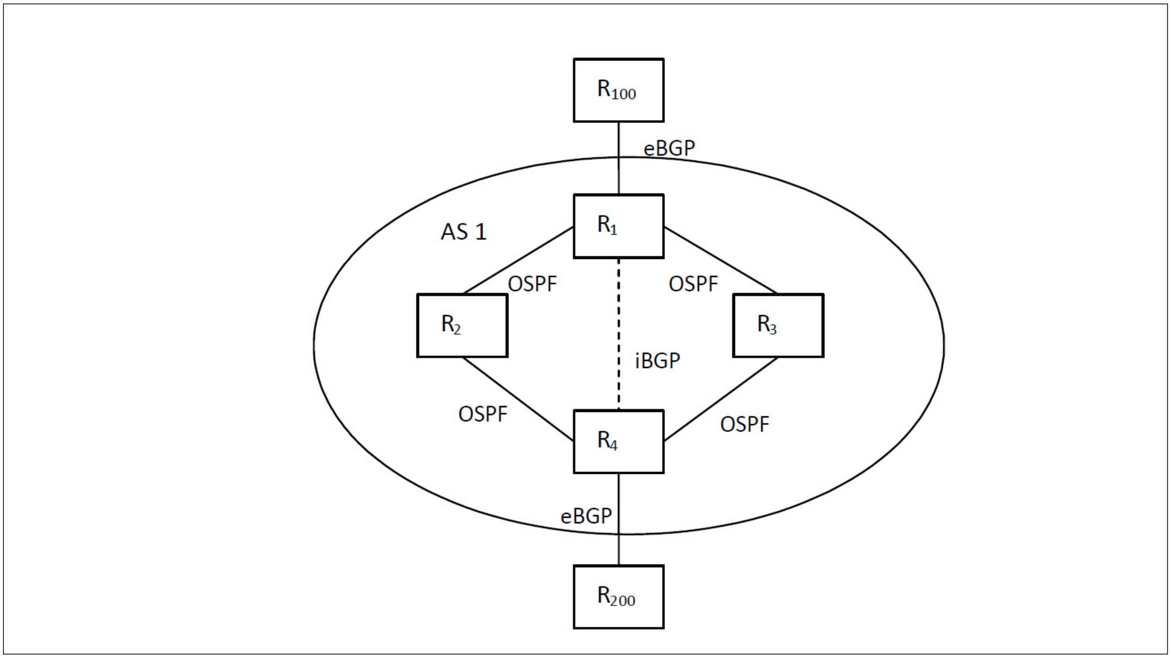
BGP UPDATE messages specify a NEXT_HOP attribute for each prefix. The NEXT_HOP attribute is always on an attached subnet for the receiver when the UPDATE is received from an external peer (since ICOS BGP does not support eBGP multihop). But the NEXT_HOP on routes from internal peers is not always on a local subnet. Thus, BGP has to resolve the BGP NEXT_HOP to one or more local next hops (similar to how a router resolves a tunnel endpoint to a local next hop). BGP resolves a remote NEXT_HOP by asking RTO for the longest prefix match. As the routing table changes, the resolution for a NEXT_HOP may change. BGP registers each remote BGP NEXT_HOP with RTO for next hop resolution changes.

When RTO notifies BGP of a next hop resolution change, BGP finds all the paths whose BGP NEXT_HOP is the IP address whose resolution changed and updates the immediate next hops for each path. A next hop resolution change triggers phase 2 of the decision process for the affected prefixes.

ICOS allows up to 1024 addresses to be registered for next hop resolution changes. This should be sufficient for BGP. The number of addresses BGP needs to track is limited to the number of external peers to the router’s autonomous system (not just the external peers for the router itself), or, if routers are configured to advertise themselves as the next hop (next-hop-self), the number of internal peers.

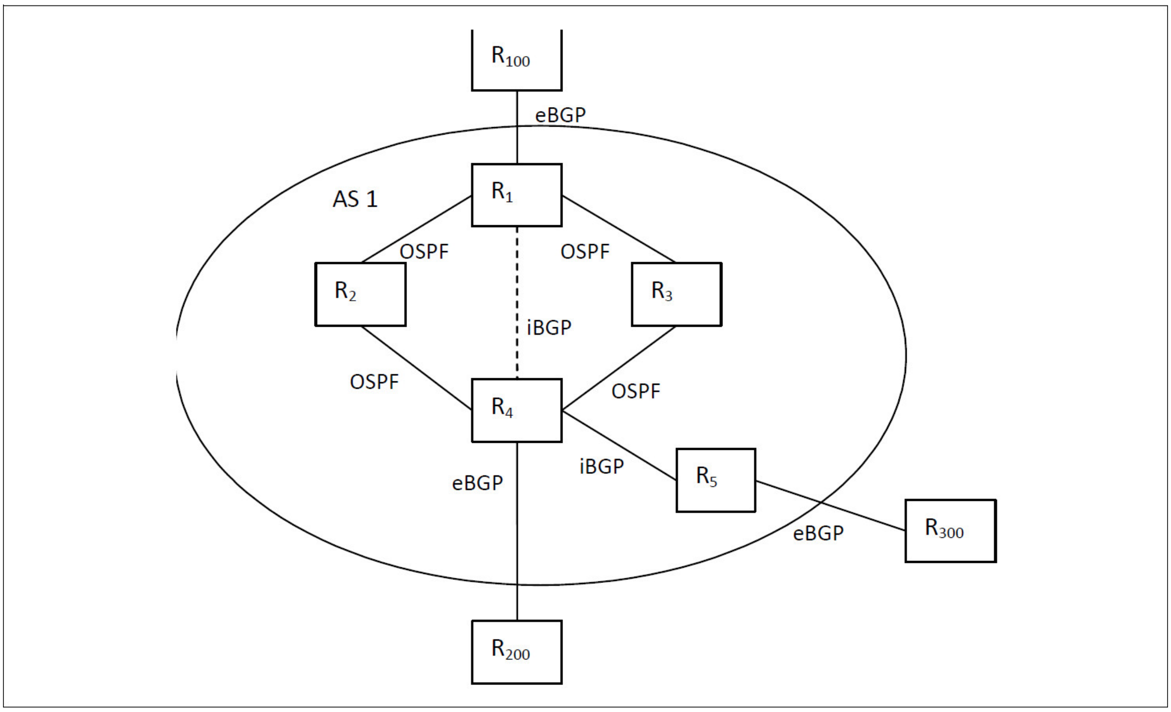
A BGP NEXT_HOP can resolve to an ECMP IGP route. When BGP is configured to allow ECMP iBGP routes, the BGP NEXT_HOP resolves to multiple next hops. BGP retains up to the number of resolved next hops allowed for an iBGP route. For example, in the figure below, R4 receives an iBGP route from internal peer R1. The BGP NEXT_HOP of this path resolves to an ECMP OSPF route through R2 and R3. If BGP is configured on R4 to allow ECMP iBGP routes, then R4 will resolve the path’s BGP NEXT_HOP to a pair of next hops through R2 and R3.

Figure 6.14. ECMP NEXT_HOP Resolution



When BGP paths are combined into an ECMP route, their next hop sets are merged to form the set of next hops for the route. For example, in the figure below, if R4 learns another route via R5 and R300 with the same destination as the route in the previous example, and the path from R300 is equivalent to the path through R100, then R4 will install a route using R2, R3, and R5 as next hops.

Figure 6.15. Combining iBGP Routes



6.4.16.13. Address Aggregation

ICOS BGP supports address aggregation. The network administrator can configure up to 128 aggregate addresses. BGP compares active prefixes in the local RIB to the set of aggregate addresses. To be considered a match for an aggregate address, a prefix must be more specific (i.e., have a longer prefix length) than the aggregate address. A prefix whose prefix length equals the length of the aggregate address is not considered a match. If one or more prefixes fall within an aggregate, the aggregate is considered active. A prefix must be used for forwarding to be considered for inclusion in an aggregate address (unless it is a locally originated prefix). Aggregate addresses may overlap (for example, 10.1.0.0/16 and 10.0.0.0/8). A prefix that matches overlapping aggregates is considered to match only the aggregate with the longest mask. When an aggregate address becomes active (that is, when the first contained route is matched to the aggregate), BGP adds a discard route to RTO with prefix and network mask equal to those defined for the aggregate address. Aggregate addresses apply to both locally originated routes and routes learned from peers.

Address aggregation is done prior to application of outbound policy. Thus, an active aggregate may be advertised to a neighbor, even if the outbound policy to the neighbor filters all of the aggregate’s more specific routes (but permits the aggregate itself).

An aggregate address is advertised with a set of path attributes derived from the best paths for each NLRI included in the aggregate. Path attributes of the aggregate are formed as follows:

- If one or more aggregated routes have ORIGIN set to INCOMPLETE, the aggregate path sets ORIGIN to INCOMPLETE. Otherwise, if one or more routes has ORIGIN set to EGP, the aggregate path sets ORIGIN to EGP. Otherwise, ORIGIN is set to IGP in the aggregate path.

- Local preference is set to the default local preference configured on the router that creates the aggregate. (Of course, if the aggregate is advertised to an external peer, local preference is not included.)

- NEXT_HOP is not imported from the aggregated routes. It is always set to the local IPv4 address on the TCP connection to the peer.

- If the as-set option is configured for an aggregate, then the aggregate is advertised with a non-empty AS_PATH. If the AS_PATH of all contained routes is the same, then the AS_PATH of the aggregate is the AS_PATH of the contained routes. Otherwise, if the contained routes have different AS_PATHs, the AS_PATH attribute includes an AS_SET with each of the AS numbers listed in the AS PATHs of the aggregated routes. If the as-set option is not configured, the aggregate is advertised with an empty AS_PATH.

- If BGP is configured to aggregate routes with different MEDs, no MED is included in the path for the aggregate. Otherwise, if the as-set option is not configured, the aggregate MED is set to the MED for the aggregated routes. If the as-set option is configured and the first segment in the AS Path is an AS SET, then no MED is advertised.

- If the as-set option is configured, the aggregate’s path does not include the ATOMIC_AGGREGATE attribute. Otherwise, it does.

- The AGGREGATOR attribute is always included.

- If the individual routes have communities and the aggregate does not have the ATOMIC_AGGREGATE attribute set, the aggregate is advertised with the union of the communities from the individual routes. If the aggregate carries the ATOMIC_AGGREGATE attribute, the aggregate is advertised with no communities.

ICOS BGP never aggregates paths with unknown attributes. By default, ICOS BGP does not aggregate paths with different MEDs, but there is a configuration option to allow this.

6.4.16.14. Inbound Policy

An inbound policy is a policy applied to UPDATE messages received from peers. ICOS BGP supports the following inbound policies:

- A prefix filter that applies to all neighbors (*distribute-list in*)
- A prefix filter that applies to a specific neighbor (*neighbor prefix-list in*)
- A per-neighbor AS path filter (*neighbor filter-list in*)
- A per-neighbor route map (*neighbor route-map in*)

These policy mechanisms determine whether to accept or reject routes received from neighbors. A route map may change the attributes of received routes.

6.4.16.15. Outbound Policy

An outbound policy is a policy applied to BGP’s best routes (those in the local RIB and active aggregates) and determines which routes are advertised to each peer. The route map option may change the attributes advertised to a peer. ICOS BGP supports the following outbound policies:

- A prefix filter that applies to all neighbors (*distribute-list out*)
- A prefix filter that applies to a specific neighbor (*neighbor prefix-list out*)
- A prefix filter that applies to redistributed routes (*redistribute route-map* with *match ip-address or distribute-list out protocol*)
- A per-neighbor AS path filter (*neighbor filter-list out*)
- A per-neighbor route map (*neighbor route-map out*)

6.4.16.16. Routing Policy Changes

When the user makes a routing policy configuration change, ICOS BGP automatically applies the new policy. Like any other configuration change, routing policy changes are immediately saved in the running configuration, as soon as the user enters the command.

Even though policy configuration changes are committed to the running configuration immediately, they do not take operational effect until three minutes after the last configuration change. The delay allows the user time to make other configuration changes or correct any mistakes before the change takes effect. If another event, such as receipt of an UPDATE message or a neighbor established event, triggers the decision process while waiting for the three minutes to expire, then the decision process runs at the time of the event using the old policy configuration. If the user wants to apply policy changes immediately, *clear ip bgp* can be entered to trigger an immediate soft reset.

In response to a change to an outbound policy, BGP recomputes update group membership and advertises updates to the affected peer to reflect the change in policy.

In response to a change to an inbound policy, BGP schedules phase 1 of the decision process. If the policy change is neighbor-specific, phase 1 only reevaluates routes received from that neighbor. If the change is global, phase 1 reevaluates all routes. If an affected neighbor supports Route Refresh, BGP sends a ROUTE REFRESH message to the neighbor and applies the new policy to the UPDATE messages received in response. If a neighbor does not support Route Refresh, BGP applies the new policy to path information previously received from the neighbor. As with outbound policy, inbound policy changes are immediately committed to the running configuration but do not take effect for three minutes. The soft reset is deferred for three minutes to allow configuration changes to be finalized before they are applied. The command *clear ip bgp* can be entered to trigger an immediate soft reset, if desired.

At start-up, when the saved configuration is applied, there could potentially be a lot of churn to outbound update groups and filtering of routing information. This start-up churn is avoided by keeping BGP globally disabled until after the entire configuration is applied and the status of all routing interfaces is known.

6.4.16.17. BGP Timers

ICOS BGP supports the five mandatory timers described in RFC 4271 section 10. ICOS BGP employs the optional IdleHoldTimer, but does not have a DelayOpenTimer.

When ICOS BGP initiates a TCP connection to a peer, it starts a retry timer (called the ConnectRetryTimer in RFC 4271). If the connection is not established before the retry timer expires, BGP initiates a new TCP connection attempt. Up to three retries are attempted with exponential backoff of the retry time. The initial retry time is configurable per neighbor.

The IDLE hold timer runs when a peer has automatically transitioned to the IDLE state. When the IDLE hold timer expires, BGP attempts to form an adjacency. The IDLE hold time is a jittered to avoid synchronization of retries. The idle hold time varies depending on the event that triggered the transition to IDLE.

ICOS BGP starts hold and keepalive timers for each peer. When BGP establishes an adjacency, the neighbors agree to use the minimum hold time configured on either neighbor. BGP sends KEEPALIVE messages at either 1/3 of the negotiated hold time or the configured keepalive interval, whichever is more frequent. Keepalive times are jittered.

RFC 4271 section 9.2.1.1 specifies a “minimum amount of time that must elapse between an advertisement and/ or withdrawal of routes to a particular destination by a BGP speaker to a peer.” In ICOS BGP, this advertisement interval is configurable independently for each peer, defaulting to 30 seconds for external peers and 5 seconds for internal peers. The advertisement interval may be configured to 0. ICOS BGP enforces the advertisement interval by limiting how often phase 3 of the decision process can run for each outbound update group. The advertisement interval is applied to withdrawals and active advertisements.

6.4.16.18. Communities

ICOS BGP supports BGP standard communities as defined in RFC 1997. (ICOS BGP also supports extended communities; see Section 6.4.16.27, “BGP Extended Communities”)

ICOS supports community lists for matching routes based on the community, and supports matching and setting communities in route maps.

ICOS BGP recognizes and honors the following well-known communities:

- Standard:NO_EXPORT—A route carrying this community is not advertised to external peers.
- NO_ADVERTISE—A route carrying this community is not advertised to any peer.
- NO_EXPORT_SUBCONFED—A route carrying this community is not advertised to external peers.

If ICOS receives an UPDATE message with more than 512 communities, a NOTIFICATION message is returned to the sender with error UPDATE message/attribute length error.

6.4.16.19. Routing Table Overflow

This section describes the routing table BGP routing table and Routing Table Overflow (RTO)

BGP Routing Table

Device configuration errors and other network transients can cause temporary or sustained spikes in the BGP routing table size. To protect the router from allocating too much memory in these scenarios, ICOS BGP limits the BGP routing table size. The limit is set to the number of routes that the device can install in its hardware forwarding table. BGP imposes separate limits for each address family it supports. Once the BGP routing table is full, new routes computed in phase 2 of the decision process cannot be added. These routes are not added to RTO, are therefore not used for forwarding, and are not advertised to neighbors. When the BGP routing table becomes full, a log message is written. While BGP remains in this state, it periodically writes a log message that states the number of NLRI that could not be added to the routing table.

BGP automatically recovers from a temporary spike in BGP routes above this limit. When BGP cannot add a route to the BGP routing table, it sets the phase 2 pending flag on that NLRI in the Accept RIB. While there are NLRI in this state, BGP periodically checks if there is space available in the BGP routing table, and if so, runs phase 2. When space becomes available in the BGP routing table, these routes are added.

RTO

If BGP computes a new route but the common routing table, RTO, does not accept the route because RTO is full, BGP flags the route as one not added to RTO. BGP periodically tries to add these routes to RTO.

BGP forwards a route to its neighbors even when the route failed to add to RTO. The only necessary condition for forwarding a route to the neighbor is that the route should be the best route in the BGP database. When used in conjunction with VRF, this rule may make routing black holes likely unless the network capacity is planned correctly.

BGP AS Path Ignore

In some scenarios, it is important for the administrator to ignore the AS Path length of different paths in the bestroute selection so as to form ECMP routes.

The AS Path length of a PATH is one of the parameters used in the BGP best-route selection algorithm. The user may choose to ignore this parameter during the best-route selection using the command *bgp bestpath aspath ignore*.

6.4.16.20. Route Reflection

ICOS BGP can be configured as a route reflector as described in RFC 4456. Like any BGP implementation, ICOS BGP can also act as a route reflector client. Route reflection eliminates the need to configure a full mesh of iBGP peering sessions. As its name implies, this feature allows a router to reflect a route received from an internal peer to another internal peer. Under conventional BGP rules, a router can only send an internal peer routes learned from an external peer or routes locally originated.

The administrator can configure an internal BGP peer to be a route reflector client. Alternatively, the administrator can configure a peer template to make any inheriting peers route reflector clients. The client status of a peer can be configured independently for IPv4 and IPv6.

A cluster may have multiple route reflectors. Route reflectors within the same cluster are configured with a cluster ID. When a route reflector reflects a route, it prepends its cluster ID to a list of cluster IDs in the CLUSTER_LIST attribute.

RFC 4456 notes that “when a RR reflects a route, it SHOULD NOT modify the following path attributes: NEXT_HOP, AS_PATH, LOCAL_PREF, and MED. Their modification could potentially result in routing loops.” For this reason, if a route reflector client has an outbound neighbor route-map configured, the set statements in the route map are ignored.

6.4.16.21. IPv6 and BGP

ICOS supports both IPv4 and IPv6 peering sessions. IPv4 routes are advertised on IPv4 peer sessions. ICOS does not support advertisement of IPv4 routes over IPv6 peer sessions. IPv6 routes can be advertised over either type of peer session as described in RFC 4760 and RFC 2545. The user must explicitly activate IPv6 route advertisement on either type of peer session. When IPv6 is enabled, the OPEN message is delivered.

IPv6 prefixes can be originated through route redistribution or a network command. Both can be configured with a route map to set path attributes. BGP can also originate an IPv6 default route. Default-origination can be neighbor-specific. IPv6 routes can be filtered using prefix lists, route maps with community lists, and using AS path access lists. BGP can compute IPv6 routes with up to 32 ECMP next hops (if not limited by user configuration or hardware limitations).

IPv6 Peering Using Link Local Address

ICOS allows an IPv6 link local address to be configured as a BGP peer address. When a link local peer address is used, the administrator must also specify the routing interface used to reach the peer.

Network Address of Next Hop

When advertising IPv6 routes, the Network Address of Next Hop field in MP_REACH_NLRI is set according to RFC 2545. Under conditions specified in this RFC, both a global and a link local next hop address may be included. The primary purpose of the global address is an address that can be re-advertised to internal peers. The primary purpose of the link local address is for use as the next hop of routes.

Interfaces to external peers will normally have both a link local and a global IPv6 address. Both addresses are included in the Network Address of Next Hop field when sending MP_REACH_NLRI to the peer. Normally, internal peers are not on a common subnet (even when they are, peering is normally to addresses on loopback interfaces) and the Network Address of Next Hop field includes only a global address. Even when the peer address of an internal peer is on a local link, ICOS BGP only advertises a global next hop IPv6 address.

- **Source Address Selection**: When BGP initiates a TCP connection to a peer, it selects a source IPv6 address. When the user has configured a source interface (using neighbor update-source), the source address is taken from this interface. When the peer's IPv6 address is a link local address, the local interface used to reach the peer is configured (in neighbor remote-as) and the source address is taken from this interface.

If the neighbor address is a link local address, BGP selects a link local address as the source address. Otherwise, BGP selects a local address in the same subnet as the neighbor's address. If no such address is found, BGP selects the first active global IPv6 address on the source interface.

- **Using Policy to Specify Next Hop**: The network administrator can override the normal rules for selecting a next hop address by configuring IPv6 next hops with outbound policy (a neighbor-specific route map with a *set ipv6 next-hop* term). When configuring an IPv6 next hop, the network administrator should ensure the neighbor can reach the next hop address. For example, a link local next hop address should not be configured to an internal peer not on a local link. Using per-neighbor outbound policy to set the IPv6 next hop has the disadvantage of putting each neighbor in a different outbound update group, thus losing the efficiency advantages of sharing an Adj-RIBOut and of building an UPDATE message once and sending it to many peers.

Alternatively, the network administrator can configure inbound policy on the receiver to set IPv6 next hops.

6.4.16.22. IPv6 Link Local Address Auto Detect

When the BGP protocol is deployed in an IPv6 data center network, it is desirable to IPv6 link-local addresses as BGP neighbors. Using link local addresses avoids the need to assign and manage global IPv6 addresses on interconnect links.

ICOS already supports BGP neighbors with link-local IPv6 addresses, but it requires that the link-local IPv6 address of the neighbor be configured using the *BGP neighbor* command. Since the link-local address is derived from the switch MAC address, this means that the network administrator needs to know the MAC addresses of all the switches deployed in the network and, if one switch fails and is replaced with a different switch, then all the BGP neighbor switches must be reconfigured to change the link-local address specified in their neighbor commands.

The IPv6 Link-Local Address Auto-Detect feature eliminates the need for the network administrator to configure the link-local IPv6 address of every neighbor. Instead of specifying the link-local IPv6 address, the network administrator can use the special keyword autodetect to refer to the link-local IPv6 address of the neighbor. For example: *neighbor autodetect interface 0/21 remote-as 10000*.

6.4.16.23. BGP ebgp-multihop

This feature allows EBGP sessions with non-directly connected neighbors.

By default, EBGP neighbors must be directly-connected peers. This is imposed by limiting the TTL value to 1. By configuring *ebgp-multihop <ttl>*, the EBGP neighbors relationship can be extended to non-directlyconnected peers. The multihop command is relevant only for eBGP and not for iBGP.

6.4.16.24. BGP allowas-in

This feature allows BGP to accept prefixes even if my ASN is part of the AS path. By default, the prefixes if received with my ASN in the AS path are rejected as part of loop prevention mechanism. This behavior can be overridden by configuring *allowas-in <count>*.

6.4.16.25. BGP Support for VRF

ICOS BGP is Virtual Routing and Forwarding (VRF) aware.

In a typical enterprise network that has multiple virtual forwarding environments (such as multi-tenant offices or different departments), they are connected either via multiple CPEs or by a single multi-VPN CPE to the PE on the service provider side.

In the case of multiple CPEs, each CPE is VRF-unaware and connects each VRF to the PE equipment via an IGP.

In the case of multi-VPN CPEs, the CPE is VRF-aware, which means the following:

- The CPE exchanges routes with enterprise routers within each VRF instance independently using IGP (OSPF or BGP or static)
- The CPE propagates these VRF route tables to the PE via MP-BGP or via an independent BGP session per VRF. ICOS BGP is enabled to run independent sessions to peers in each VRF instance and redistribute the routes from the VRF routing tables.

6.4.16.26. BGP Dynamic Neighbors

BGP neighbors can be dynamically created whenever connection requests from peers are received from a configured IP address range. Creating neighbors dynamically avoids explicit configuration by the administrator when forming peering with neighbors, irrespective of the subnet to which the IP addresses belong.

The administrator specifies the address range to listen on, and the neighbors properties are inherited from a peer template. As a result, all dynamically created neighbors inherit the properties from the template.

The number of configurable listen address ranges in the system is limited to 10. The number of dynamic peers created as a result of this feature are also limited by the total number of peers allowed in the system.

6.4.16.27. BGP Extended Communities

ICOS BGP supports standard extended communities as defined in RFC 4360. ICOS supports extended community lists for matching routes based on the extended community, and supports matching and setting of extended communities in route maps.

The extended community attribute provides a mechanism for labelling routes carried in BGP-4. These labels are then used to control the distribution of the routes among VRFs.

A BGP route can carry both standard and extended communities attributes. It can also carry multiple community attributes by using the additive keyword (for standard communities) and by using route-maps when exporting the VRF routes (for extended communities).

Routing policies that are applied inbound or outbound on a BGP neighbor can match on the Extended communities or modify the Extended communities associated with PATH attributes or prefixes.

Extended Community Structure

Each Extended Community attribute has a defined community type code of 16 and is encoded into a 8 Octet Value. The first two octets are the attribute type and the remaining six octet hold the value of attribute. The Value from 0 through 0x7FFF is assigned by IANA and Value from 0x8000 through 0xFFFF are inclusive vendorspecific.

The extended community attribute is represented in one or more ways but ICOS only supports the following formats:

- 2-octet AS-specific extended community
- IPv4 address-specific extended community

The format specifics of the extended community values based on their types can be found in the RFC 4360.

Types of Extended Communities BGP recognizes and honors the following well-known extended community attributes (RFC 4360):

- Route Target Community—Identifies one or more routers that may receive a set of routes (attached with this community) carried by BGP. The value of the high-order octet of the Type field for the Route Target Community can be 0x00, 0x01 or 0x02. The value of the low-order octet (Sub-type) of the Type field for this community is 0x02 (if represented in Two-octet AS specific format) and 0x102 (if represented in IPv4 address specific format).
- Route Origin Community—Identifies one or more routers that inject a set of routes (attached with this community) carried by BGP. It is used to prevent routing loops when a site is multi-homed to the MPLS/ VPN backbone and the site uses the AS-Override feature. It identifies the site where the routes are learned, based on its Origin, so that is not re-advertised back to that site from a PE-Router somewhere else in the MPLS/VPN backbone.

These communities are transitive across the Autonomous System boundary. The following sections provide more information on the possible uses of these communities.

VPNv4/VRF Route Distribution via BGP

PE routers use BGP to distribute VPN routes to each other. Each VRF has its own address space, meaning that the same address can be used in any number of VRFs where, in each VRF, the address specifies a different system. But a BGP speaker can install and distribute only one route to a given address prefix. ICOS allows BGP to install and distribute multiple routes to a single IP address prefix. It is recommended that administrators use a policy to determine which sites can use which routes; several such routes are installed by BGP but only one must appear in any particular per-site VRF route table. This is achieved by the use of a new address family.

VPNv4 Address Family

MP-BGP allows BGP to carry routes from different address families. To allow BGP to carry and distribute overlapping address routes, each address/route is made unique. To achieve this, a new

VPNv4 address family is introduced. A VPN-IPv4 address is a 12-byte quantity, beginning with a 8-byte Route Distinguisher (RD) followed by a 4-byte IPv4 address.

If two VRFs use the same IPv4 address prefix, the PE translates these into unique VPN-IPv4 address prefixes by prepending the RD (configured per VRF) to the address. The purpose of the RD is only to allow one to create unique routes to a common IPv4 address prefix. The structuring of RD provides no semantics. When BGP compares two such addresses, it ignores the RD structure completely and just compares it as a 12-byte entity.

A PE is configured to associate routes that belong to a particular CE instance (a VRF) with a particular RD. When BGP redistributes these routes, the PE router prepends the configured RD value (for that CE) to the route and carries them to the other PE as VPNv4 routes. The PE router that receives these VPNv4 routes installs them in the global BGP table along with the RD. If two routes have the same address prefix but different RD values, only the first route is installed to the RTO table of the CE that imports the route and the rest are ignored.

ICOS BGP sends traditional (IPv4) NLRI in addition to MP-BGP NLRI when a neighbor is activated in VPNv4 address family mode.

Controlling Route Distribution

This section describes the method in which the VPNv4 route distribution is controlled.

- **The Route Target Attribute (RT)**: A Route Target attribute identifies a set of sites. Associating a particular Route Target attribute with a route allows the route to be placed in the per-site (CE) VRF tables. Every per-site (CE) VRF is associated with one or more Route Target attributes.

When a VPNv4 route is created by a PE router, it is associated with one or more Route Target attributes. These are carried in BGP as attributes of that route.

Any route associated with Route Target attribute RT1 must be distributed to every PE router that has a VRF associated with Route Target RT1. When such a route is received by a PE router, (depending on the BGP decision process) it is installed in each of the PE’s VRF tables that are associated with Route Target RT1.

When a PE router receives a route from one of its CE routers, it attaches to the route one or more Export Route Target attributes (as configured for that CE VRF). The route is then carried via MP-BGP to the other PE router. The PE router that receives the route compares it with the Import Route Target attributes configured for one or multiple VRFs and, depending on the match, installs the route in the matching VRF table.

The Export Route Target attributes and the Import Route Target attributes are two distinct sets and may or may not be the same. If they are same, only then the route is allowed to be installed in that particular VRF table.

A BGP route can have only one RD but multiple Route Targets.

A PE may be configured to associate all the routes that belong to a VRF with a particular Route Target attribute. In a way, this limits the administrative control to selectively associate a particular Route Target attribute only to some routes. ICOS overcomes this limitation with the use of an Export and Import Map Policy. Export and Import maps provides greater flexibility to the administrator wherein the administrator can associate some routes of a VRF with a particular Route Target attribute and other routes with another Route Target attribute.

The Route Target attribute helps in route leaking among multiple VRFs in a PE. Essentially the route leaking between VRFs can be achieved without any BGP adjacencies in the VRF instances, but with just the import and export Route Target statements.

ICOS allows configuring Route Target attributes in VRF mode, using IP Extended Community lists in association with inbound/outbound Route maps.

- **The Site of Origin Attribute (SoO)**: A VPNv4 route may optionally carry a Origin attribute that uniquely identifies a set of sites. This attribute identifies the corresponding route as having come from one of the sites.

The SoO attribute is used to identify the specific site from which the PE learns the route and is used in the identification and prevention of routing loops. The SoO extended community is a BGP extended community attribute used to identify routes that have originated from a site to prevent the re-advertisement of that prefix back to the source site, thus preventing routing loops.

SoO enables filtering of traffic based on the site from which it was originated. SoO filtering manages traffic and prevents routing loops from occurring in complex and mixed-network topologies in which the customer sites might possess backdoor links between sites.

SoO is one of the attributes a PE router assigns to a prefix prior to redistributing any VPNv4 prefixes. All prefixes learned from a particular site must be assigned the same site of origin attribute, even if the site is multiple-connected to a single PE, or is connected to multiple PEs.

ICOS allows configuration of the SoO attribute using IP Extended Community lists in association with inbound/outbound route maps.

How VPNv4 NLRI is carried in BGP

The BGP Multiprotocol Extensions are used to encode the NLRI. If the Address Family Identifier (AFI) field is set to 1 and the Subsequent Address Family Identifier (SAFI) field is set to 128, the NLRI is an VPNv4 address. AFI 1 is used since the network layer protocol associated with the NLRI is still IP.

For two BGP speakers to exchange labeled VPN-IPv4 NLRI, they must use BGP Capabilities Advertisement (in an OPEN message) to ensure that they both are capable of properly processing such NLRI. This is done by using capability code 1 (multiprotocol BGP), with an AFI of 1 and an SAFI of 128.

The VPNv4 NLRI is encoded as specified in the above sections, where the prefix consists of an 8-byte RD followed by an IPv4 prefix.

6.4.16.28. Scaling Parameters

ICOS allows BGP to scale in a number of directions, including number of neighbors, number of ECMP next hops, and number of BGP routes. The table below lists some BGP scaling parameters. BGP imposes no limit on the number of routes other than the limit imposed by the size of the hardware forwarding table. However, it is likely that platforms will not have enough resources (typically memory and CPU) to simultaneously scale every parameter to its maximum. It is possible to compute the maximum number of BGP routes while the number of neighbors is small and the number of ECMP next hops is small. Alternatively, it is possible to form the maximum number of adjacencies if the number of routes is small.

Table 6.6. BGP Scaling Parameters

Parameter Name	Parameter Description	Standard Value	File
L7_BGP_MAX_ROUTES	Maximum number of BGP routes.	platRtr-RouteMax EntriesGet()	bgp_exports.h
L7_BGP_MAX_NUMBER_OF_ PEERS	Maximum number of internal and external peers.	256	bgp_exports.h
L7_BGP_MAX_NUMBER_OF_ VPNV4_PEERS	Maximum number of peers allowed to be activated for VPNv4 address family mode.	16	bgp_exports.h
L7_BGP_MAX_NETWORK_ STMTS	Max number of network commands for each address family.	64	bgp_exports.h
L7_BGP4_MAXPATH_MAX	Maximum number of next hops in a BGP route	32 May be limited by platform	l3_bgp_commdefs.h
L7_BGP_MAX_ADDR_AGG_ENTRIES	Maximum number of aggregation entries	128	bgp_exports.h
L7_IP_NHRES_MAX	Maximum number of next hop resolution registrations	1024	l3_commdefs.h
L7_BGP_MAX_PEER_ TEMPLATES	The maximum number of peer templates	32	bgp_exports.h

6.4.16.29. Usage Scenarios

ICOS BGP is suitable for Enterprise or data center use. It does not scale to hold a full Internet routing table.

6.4.17. VRF Lite

The Virtual Routing and Forwarding (VRF) Lite feature enables a router to function as multiple routers. Each virtual router manages its own routing domain. Specifically, each virtual router maintains its own IP routes, routing interfaces, and host entries, which enables each virtual router to make its own routing decisions, independent of other virtual routers. More than one virtual routing table may contain a route to a given destination. The network administrator can associate a subset of the router’s interfaces with each virtual router. The router routes packets according to the virtual routing table associated with the packet’s ingress interface. Each interface can be associated with at most one virtual router.

The OSPF, Ping, Traceroute, and BGP applications are VR-aware.

6.4.18. BFD

In a network device, Bidirectional Forwarding Detection (BFD) is presented as a service to its user applications, which provides them options to create and destroy a session with a peer device and reports upon the session status. BFD uses a simple hello mechanism that is similar to the neighbor detection components of some well-known protocols. It establishes an operational session between a pair of network devices to detect a two-way communication path between them and serves information regarding it to the user applications. The pair of devices transmits BFD packets between them periodically and, if one stops receiving peer packets within the detection time limit, it considers the bidirectional path to have failed. It then notifies the application protocol using its services. BFD allows each device to estimate how quickly it can send and receive BFD packets to agree with its neighbor on how fast detection of failure could be done. BFD can operate between two devices on top of any underlying data protocol (network layer, link layer, tunnels, etc.) as the payload of any encapsulating protocol appropriate for the transmission medium. The ICOS BFD is designed to work with IPv4 networks and supports IPv4 address-based encapsulations. BFD is available with both the Border Gateway Protocol and with OSPF. These protocols use BFD to identify fast detection of connectivity status with adjacent routers. MPLS Support The ICOS BGP protocol supports RFC 3107 to distribute MPLS labels. When the label distribution capability is enabled, BGP carries the SAFI value of 4 along with the corresponding address family (AFI for IPv4 or IPv6) and the interface prefix or loopback global prefix associated with that label in the advertised NLRI. ICOS supports label distribution only over eBGP connections. There is no specific user interface restriction to prevent the network administrator from enabling MPLS label distribution with iBGP sessions, but the iBGP sessions are not tested and, therefore, enabling MPLS with iBGP is a network configuration error.

6.4.19. MPLS Support

The ICOS BGP protocol supports RFC 3107 to distribute MPLS labels. When the label distribution capability is enabled, BGP carries the SAFI value of 4 along with the corresponding address family (AFI for IPv4 or IPv6) and the interface prefix or loopback global prefix associated with that label in the advertised NLRI. ICOS supports label distribution only over eBGP connections. There is no specific user interface restriction to prevent the network administrator from enabling MPLS label distribution with iBGP sessions, but the iBGP sessions are not tested and, therefore, enabling MPLS with iBGP is a network configuration error.

6.4.20. Local AS Support (Hide ASN)

In typical data center deployments using Clos networks, the peering is all external BGP between the BGP devices, requiring an unique ASN for each router. Normally, the private BGP networks are expected to use private AS numbers. But there are only 1024 private AS numbers in the standard 2-byte ASN. Hence, customers are forced to use public ASNs in their private networks.

When such private networks are interconnected to each other, there needs to be a way to manipulate the public ASNs in the route advertisements so that the private networks with the public ASNs do not experience ASN conflicts. ICOS provides a command to achieve this:

```
neighbor <ipv4 or ipv6 address> local-as no-prepend replace-as
```

The options no-prepend and replace-as:

- The router replaces the global AS of the router with the configured *local-as* when advertising the routes to the peer on which this command is configured.
- The *local-as* is not prepended to the routes received from the neighbor on which this command is configured.

NOTE:

- When the *local-as* is configured on a peer, the BGP peer adjacency is reset.
- This feature is applicable to external BGP peers only.
- The *local-as* configuration is allowed in the peer group template.

6.4.21. RFC 5549

The goal of RFC 5549, “Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop,” is to enable users to deploy a mixed IPv4/IPv6 network without having to assign IPv4 addresses to transit links between switches. Instead, IPv6 interfaces are used for forwarding the IPv4 traffic.

The basic concept is to enable IPv4 routes to use IPv6 NDPs to determine the next hop. For example the *show ip route* command shows some IPv4 routes with IPv6 addresses as next hops.

There is no IPv6 tunneling involved in this solution. Rather the IPv4 packets are routed as normal, except they use next hops determined by the IPv6 protocol.

RFC 5549 also adds BGP extensions to insert these IPv4 routes with IPv6 next hops into the routing table.

6.4.22. Algorithmic Longest Prefix Match (ALPM)

ALPM is a protocol used by routers to select an entry from a forwarding table. When an exact match is not found in the forwarding table, the match with the longest subnet mask, also called longest prefix match, is chosen. It is called the longest prefix match because it is also the entry where the largest number of leading address bits of the destination address match those in the table entry.

ALPM is primarily a switch silicon feature and the algorithm for this is implemented in the SDK on the chip. ALPM enables supporting for large number of routes (for BGP, 32k IPv4 routes and 24k IPv6 are supported).

Support for ALPM is platform-dependent. For platforms that support ALPM, two SDM templates, “dual-ipv4-andipv6 alpm-data-center” and “dual-ipv4-and-ipv6 alpm-mpls-data-center”, are made available to accommodate the larger number of routes.

6.5. IPv6 Routing and Management

IPv6 is the next generation of the Internet Protocol. With 128-bit addresses, IPv6 solves the address depletion issues seen with IPv4 and removes the requirement for Network Address Translators (NATs). With IPv6, security is more integrated, and network configuration is simplified, yet more flexible. ICOS supports both EUI-64 interface identifiers and manually configured interface IDs.

ICOS supports management access via IPv6 and unicast static routing; however, it does not support full IPv6 routing functionality. In ICOS, IPv6 coexists with IPv4. As with IPv4, IPv6 routing can be enabled on physical and VLAN interfaces. Each layer 3 routing interface can be used for IPv4, IPv6, or both. Higher layer protocols, such as UDP and TCP, do not change with IPv6. For this reason, a single dual IP stack, provided by the Linux operating system, is used for transport of both IPv4 and IPv6. A single sockets interface provides access to both IPv4 and IPv6 services in the IP stack. ICOS supports static IPv6 routes and supports OSPFv3 as a dynamic unicast routing protocol for IPv6.

The major layer-3 features are as follows:

- Section 6.5.1, “OSPFv3”
- Section 6.5.2, “DHCPv6”
- Section 6.5.3, “IPv4 to IPv6 Transition”
- Section 6.5.4, “IPv6 Routing Support”
- Section 6.5.5, “IPv6 Management Features”

6.5.1. OSPFv3

OSPFv3 is the Open Shortest Path First (OSPF) routing protocol for IPv6. It is similar to OSPFv2 in its concept of a link state database, intra/inter area and AS external routes and virtual links. It also differs from its IPv4 counterpoint in a number of respects: peering is done using link-local addresses, the protocol is link rather than network centric and addressing semantics have been moved to leaf LSAs, which eventually allow its use for both IPv4 and IPv6. Point to point links is supported in order enable operation over tunnels. OSPFv3 views 6over4 tunnels as a point-to-point interface with link-local address (and possibly a global address). OSPFv3 uses the reported MTU for tunnel interfaces.

OSPFv3 supports ECMP routes. OSPFv3 includes NSSA and AS-external LSA overflow limit support. RFC 1583 compatibility does not apply to OSPFv3. OSPFv3 authentication utilizes the IPv6 stack IPSEC mechanisms. Because the initial IPv6 release provides no IPSEC, OSPFv3 supports no authentication mechanisms. OSPFv3 does not support MOSPF.

LSA formats are changed, and the type 3 and 4 summary LSAs are renamed inter-area-prefix and inter-arearouter LSAs. Also note that OSPFv3 LSA identifiers contain no addressing semantics. LSA scope is generalized to link, area, and AS scope. OSPFv3 specifies the processing of unsupported LSAs. Unsupported LSAs are maintained in the database and flooded according to scope. In OSPFv3, Routers with 100 or more interfaces generate more than one router LSA. A new link LSA has been created. Addresses in LSAs are specified as [prefix, prefix length].

Area ID and Router ID remain 32-bit identifiers. OSPFv3 identifies Neighbors by router ID instead of the interface address used in OSPFv2.

OSPFv3 supports RFC 6860 for hiding the transit-only networks. A transit-only network is defined as a network connecting only routers. Hiding transit-only networks can speed up network convergence and reduce vulnerability to remote attacks to the routers in the transit-only network.

6.5.2. DHCPv6

DHCP is generally used between clients (such as hosts) and servers (such as routers) for the purpose of assigning IP addresses, gateways, and other networking definitions such as DNS, NTP, and/or SIP parameters. However, IPv6 natively provides for auto-configuration of IP addresses through IPv6 Neighbor Discovery Protocol (NDP) and the use of Router Advertisement messages. Thus the role of DHCPv6 within the network is different than that of DHCPv4 in that it is less relied upon for IP address assignment.

ICOS uses DHCPv6 client to dynamically assign a global IPv6 address on the management interface (which can be network port, service port or host interface). A link-local IPv6 address is automatically assigned on the management interface when it comes up. To be able to manage the DUT globally, a global IPv6 address must be assigned on a management interface using Stateless Address Autoconfiguration (using Router Advertisements) or manual configuration.

ICOS provides the following DHCPv6 features:

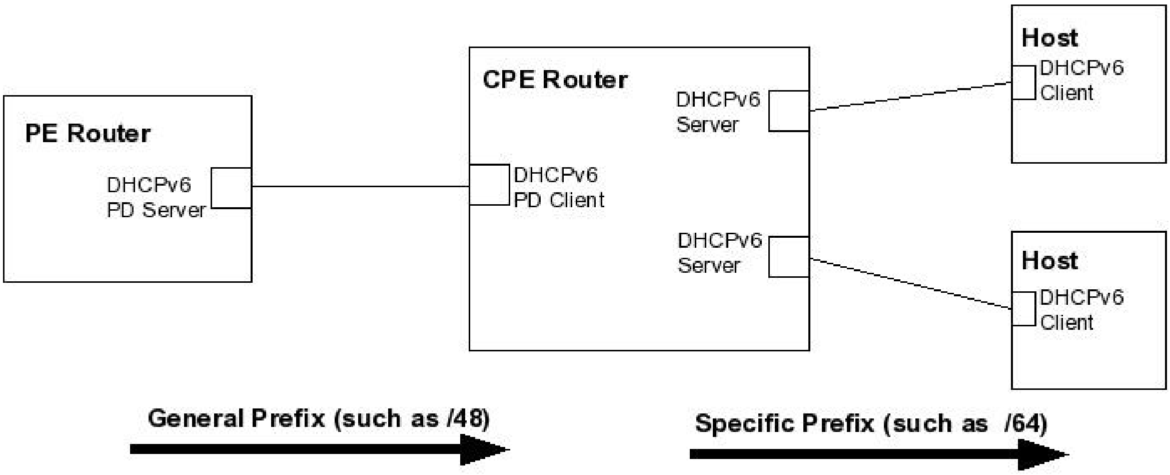
- Configuration of IPv6 global address autoconfiguration on management interfaces.
- Enabling/disabling of the stateful DHCPv6 client on management interfaces to dynamically acquire a global IPv6 address. Assigning the address dynamically using the Stateful DHCPv6 Server avoids address conflicts that can happen due to manual configuration.
- Stateful server functionality to assign IPv6 addresses dynamically to DHCPv6 stateful clients (prefix delegation clients and end host clients) based on the Client DUID in compliance with RFC 3315.
- Prefix delegation client support to receive a general prefix from the prefix delegation server for further assignment over local router interfaces to end host clients.

DHCPv6 server and client interactions are described by RFC 3315. There are many similarities between DHCPv6 and DHCPv4 interactions and options, but the messages and option definitions are sufficiently different to the extent that there is no DHCPv4 to DHCPv6 migration or interoperability.

DHCPv6 incorporates the notion of the stateless server, where DHCPv6 is not used for IP address assignment to a client; rather it only provides other networking information such as DNS, NTP, and/or SIP information. The stateless server behavior is described by RFC 3736, which simply contains descriptions of the portions of RFC 3315 that are necessary for stateless server behavior. In order for a router to drive a DHCPv6 client to utilize stateless DHCPv6, the *other stateful configuration* option must be configured for neighbor discovery on the corresponding IPv6 router interface. This in turn causes DHCPv6 clients to send the DHCPv6 *Information Request* message in response. A DHCPv6 server then responds by providing only networking definitions such as DNS domain name and server definitions, NTP server definitions, and/or SIP definitions.

With the larger address space inherent to IPv6, addresses within a network can be allocated more effectively in a hierarchical fashion. DHCPv6 introduces the notion of *prefix delegation* as described in RFC 3633 as a way for routers to centralize and delegate IP address assignment. The figure below depicts a typical network scenario where prefix delegation is used.

Figure 6.16. DHCPv6 Prefix Delegation Scenario



In the figure above, the PE router acts as Prefix Delegation server and defines one or more general prefixes to delegate to a CPE router acting as a Prefix Delegation client. The CPE router that can then allocate more specific addresses within the given general prefix range to assign to its local router interfaces. The CPE router can in turn use the given general prefix in allocating and assigning addresses to host machines that may be utilizing IPv6 auto-address configuration or acting as DHCPv6 clients.

6.5.3. IPv4 to IPv6 Transition

To support IPv4 to IPv6 transition, ICOS supports configured tunnels (RFC 4213) and automatic 6to4 tunnels (RFC 3056). 6to4 tunnels are automatically formed IPv4 tunnels carrying IPv6 traffic. The automatic tunnel's IPv4 destination address is derived from the 6to4 IPv6 address of the tunnel's next hop. ICOS can act as a 6to4 border router that connects a 6to4 site to a 6to4 domain. The border router sends and receives tunneled traffic from routers in the 6to4 domain that include other 6to4 border routers and 6to4 relay routers. Only 1 6to4 tunnel can be created on an ICOS switch.

6.5.4. IPv6 Routing Support

In addition to the other IPv6 routing features this section describes, ICOS IPv6 routing includes support for the following features:

- Neighbor Discovery Protocol (NDP).
  - Neighbor advertisement and solicitation.
  - Duplicate address detection.
  - Unreachability detection.
  - Router advertisement and solicitation.
- Stateless autoconfiguration of end nodes.

- EUI-64 interface identifiers and manually configured interface IDs.
- Ethernet and tunnel interface types.
- IPv6 prefix lists, which are used to specify a range of IPv6 prefixes that must be matched before a permit or deny statement can be applied. These prefix lists are used as routing policy constructs to filter/manage routing updates sent and received between BGP peers.
- Manual configuration of IPv6 neighbors on the service port, network ports, and routing/host interfaces.
- Mitigation for the DoS effect caused due to unresolved IPv6 packets copied to CPU for Neighbor resolution (described in RFC 6583).
  - Configuration of a rate limit value in packets per second for the number of unresolved IPv6 packets received by the CPU.
  - Maximum number of multicast neighbor solicitations sent during IPv6 neighbor resolution or during Neighbor Unreachability Detection (NUD).
  - Maximum number of unicast neighbor solicitations sent during NUD.
  - The exponential backoff multiple value used during the computation of the next timeout value for transmission of unicast or multicast neighbor solicitation during NUD.
- Configuration of 127-bit IPv6 prefixes on the routing/host IPv6 interfaces.
- Configurable option to enable or disable the generation of IPv6 redirect packets to the source node in a redirect scenario.
- Negative IPv6 neighbor entries, which solves the problem of CPU churn due to the continuous CPU bound IPv6 data traffic caused by the failure or unreachability of a host or next hop to which the data is destined.
- Configurable option to send router advertisements on an IPv6 routing interface with an unspecified (0) current hop limit value. This setting tells the hosts on that link to ignore the hop limit from this router.
- Dynamic renewal of dynamic IPv6 neighbor entries, with implementation details as follows:
  - When a neighbor entry is deleted either during Neighbor Unreachability Detection (NUD) timeout or during STALE neighbor entry timeout (1200 seconds), one of the following two actions are taken:
    - If dynamic renew is enabled, the neighbor is not checked for its hardware usage hit-bit and is retried for resolution. This initially creates an INCOMPLETE entry that transitions later to REACHABLE if the retry is successful or to FAILED if the retry is unsuccessful.
    - If dynamic renew is disabled, the neighbor is retried for resolution only if the hardware usage hit-bit is set. If the hit-bit is set, the routing feature initially creates an INCOMPLETE entry that later transitions to REACHABLE or FAILED.
    - If dynamic renew is enabled, the STALE neighbor cache entries are periodically subjected to NUD at the rate of 300 neighbors every 40 seconds. This means with a fully populated table of size of 2560, for example, the same neighbor is subjected to NUD every 320 sec-

onds. Whenever NUD is triggered, the exponential backoff-based NUD is triggered in the OS stack.

6.5.5. IPv6 Management Features

The networking device can be configured through the CLI and via SNMP. The following management protocols and applications can be run over IPv6 transport:

- Pingv6
- Traceroutev6
- TFTP
- SSH
- SSL
- TELNET
- SNMP

For ICMPv6, error PDU generation is supported, as are path MTU, echo request/reply, and redirect. For SNMP, ICOS supports the IPv6 MIB, ICMPv6 MIB, and private MIB extensions.

Router Advertisement is an integral part of IPv6 and is supported. Numerous options are available including stateless/stateful address configuration, router and address lifetimes, and neighbor discovery timer control.

Ethernet and tunnel interface types are supported. For Ethernet, link-local address mapping and multicast address mapping are supported. The tunnel interface type supports link-local address mapping but not general neighbor discovery since the interface is not considered to have a link layer address. The network operator can configure multiple global addresses per interface.

Applications using IPv6 can use both the basic socket extensions for IPv6 (RFC 3493) as well as the advanced sockets (RFC 3542) extensions. Netlink socket (RFC 3549) support is included.

The network port, service port, and host interfaces are logical management interfaces. The IP stack’s routing table contains IPv6 routes associated with these management interfaces.

6.6. Quality of Service Module

The Quality of Service Module (QoS) section describes the QoS components. The QoS Module contains the access control list (ACL) and differentiated services (DiffServ) components. The module is within the application layer, as shown in Figure 2.1, “System Layers”.

The following features are described in this section:

- Section 6.6.2, “Access Control Lists”
- Section 6.6.3, “Differentiated Services (DiffServ)”
- Section 6.6.4, “Class of Service (CoS)”

6.6.1. QoS Overview

Quality of Service (QoS) technologies are intended to provide guaranteed timely delivery of specific application data to a particular destination. In contrast, standard IP-based networks are designed to provide best effort data delivery service. Best effort service implies that the network deliver the data in a timely fashion, although there is no guarantee. During times of congestion, packets may be delayed, sent sporadically, or dropped. For typical Internet applications, such as electronic mail and file transfer, a slight degradation in service is acceptable and in many cases unnoticeable. Conversely, any degradation of service has undesirable effects on applications with strict timing requirements, such as voice or multimedia.

QoS is a means of providing consistent, predictable data delivery by distinguishing between packets that have strict timing requirements from those that are more tolerant of delay. Packets with strict timing requirements are given special treatment in a QoS-capable network. To accomplish this, all elements of the network must be QoScapable. If one node is unable to meet the necessary timing requirements, this creates a deficiency in the network path and the performance of the entire packet flow is compromised.

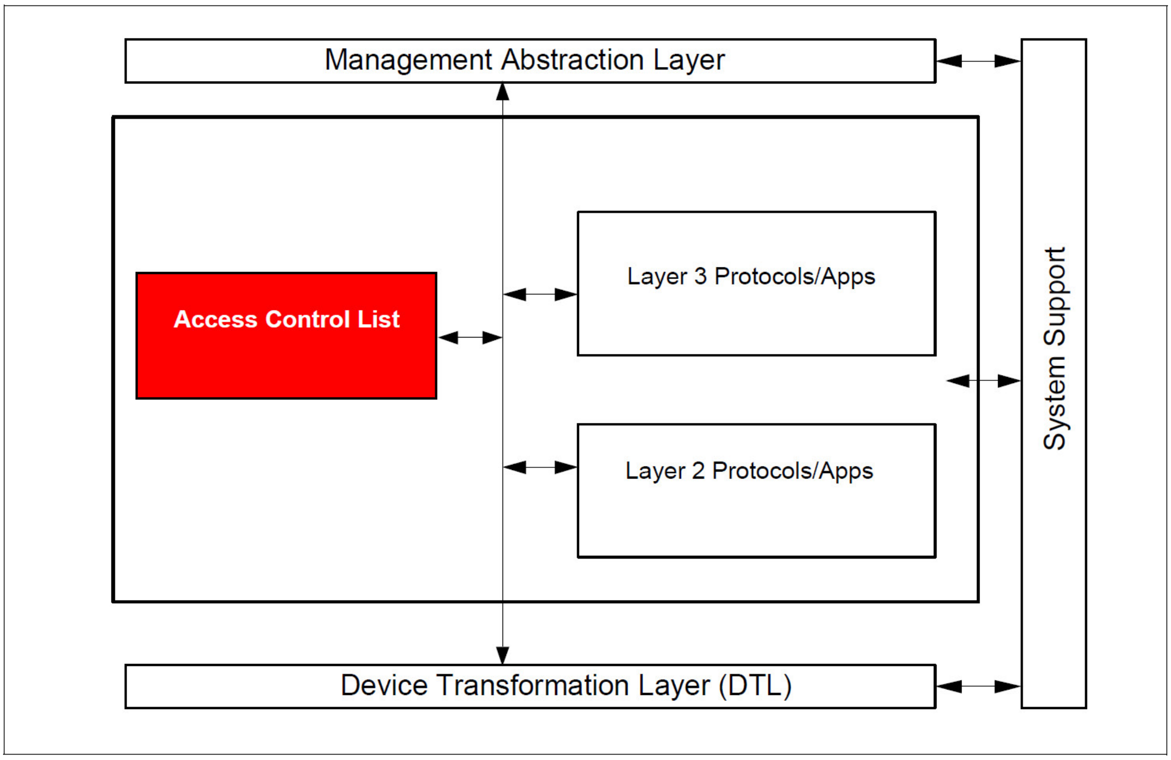
6.6.2. Access Control Lists

Access Control Lists (ACL) ensure that only authorized users have access to specific resources while blocking out any unwarranted attempts to reach network resources.

ACLs are used to restrict contents of routing updates, decide which types of traffic are forwarded or blocked, and, above all, provide security for the network. ACLs are normally used in firewall routers that are positioned between the internal network and an external network, such as the Internet. They can also be used on a router positioned between two parts of the network to control the traffic entering or exiting a specific part of the internal network.

The ICOS ACL feature allows classification of packets based upon layer 2 through Layer 4 header information. An Ethernet IPv6 packet is distinguished from an IPv4 packet by its unique Ethertype value; thus, all IPv4 and IPv6 classifiers include the EtherType field.

Figure 6.17. Access Control List Architecture



6.6.2.1. Layer 2 ACLs

The layer 2 ACL feature provides access list capability by allowing classification on the layer 2 header of an Ethernet frame, including the 802.1Q VLAN tag(s). In addition, the rule action set is enhanced to designate which (egress) CoS queue should handle the traffic, and whether the traffic flow is to be redirected to a specific outgoing interface.

Characteristics of the layer 2 ACL feature are explained in the sections that follow.

Layer 2 MAC Named Access Lists

MAC access lists are identified by a user-specified name instead of a number.

Layer 2 ACL Classification Fields

The following packet fields are defined for layer 2 ACLs:

- CoS (802.1p) user priority
- Destination MAC address (with wildcard), or *any*
- Ethertype (keyword or custom value)
- Secondary CoS (802.1p) user priority
- Secondary VLAN identifier range

- VLAN identifier range
- Source MAC address (with wildcard), or *any*

The CoS and VLAN ID range fields belong to the first 802.1Q tag in the Ethernet frame. This is the only tag in a frame containing a single VLAN tag, and is the outer tag in a double VLAN tagged frame. The Secondary CoS and Secondary VLAN ID range fields, therefore, are only valid for a double VLAN tagged Ethernet frame and are contained in the inner tag.

A Rule Definition to Permit/Deny a Match on Every Packet

Similar to the IP extended access lists:

- deny any any
- permit any any

Ethertype Field

Specified by using a keyword from the following list:

- appletalk
- ipv6
- netbios
- arp
- ipx
- novell
- ibmsna
- mplsmcast
- pppoe
- ipv4
- mplsucast
- rarp

In addition, a custom Ethertype value from 0x0600 to 0xFFFF may be specified by the user. If an Ethertype is not specified in a layer 2 ACL rule definition, any Ethertype is considered a match.

Assign (CoS) Queue Designation

The user specifies which CoS queue handles packets conforming to an ACL *permit* rule. This action is ignored for any *deny* rule, since, by definition, matching packets are dropped. Packets matching an ACL rule for which the CoS queue assignment is designated are directed to the corresponding queue number (0 to n – 1, where n is the maximum number of supported traffic classes) of the egress interface.

ACL Counters

For the following ACL types, ICOS provides a counter for every ACL rule applied on physical interface, LAG, and VLAN, with no additional configuration:

- IP standard ACLs
- IP extended ACLs
- IPv4 named ACLs
- IPv6 named ACLs
- MAC ACLs

The counters tally the hit-count for each ACL rule in both the ingress and egress directions. These counter values can be viewed and reset using CLI show and clear commands for ACLs.

6.6.2.2. Layer 3/4 IPv4 ACLs

The Layer 3/4 ACL feature supports IP access lists, both standard and extended. These lists check the Layer 3 portion of a packet, looking specifically at information contained in the IP header and, in certain cases, the TCP or UDP header. An Ethertype of 0x0800 is assumed in the case of IP access lists. Permit and deny actions are supported for each ACL rule.

Standard Layer 3/4 ACLs can be classified based on the source IP address and netmask. Extended Layer 3/4 ACLs can be classified based on one or more of the following:

- Protocol Keyword
  - ICMP
  - IGMP
  - IP
  - TCP
  - UDP
  - Protocol number (1–255)
- Source IP Address
- Source IP Mask
- Source L4 Port
  - Domain
  - Echo
  - FTP

  - FTPDATA
  - HTTP
  - SMTP
  - SNMP
  - Telnet
  - TFTP
  - WWW
- Destination IP Address
- Destination IP Mask
- Destination L4 Port
  - Domain
  - Echo
  - FTP
  - FTPDATA
  - HTTP
  - SMTP
  - SNMP
  - Telnet
  - TFTP
  - WWW
- Service Type
  - IP DSCP
  - IP precedence
  - IP TOS

The IPv4 ACL feature supports IP fragment filtering to help block two types of attacks that involve IP fragments of TCP packets:

- Tiny fragment attacks
- Overlapping fragment attacks.

The attacks and security considerations are described in RFC 1858.

Named IP Access Lists

This feature allows the user to create and manipulate IPv4 ACLs identified by a name in addition to numbered ACLs. Named IPv4 ACLs are extended ACLs rather than *standard*.

6.6.2.3. IPv6 ACLs

ICOS supports the following match criteria for IPv6 ACLs:

- Protocol
- Source prefix/prefix length
- Source L4 Port
- Destination prefix/prefix length
- Destination L4 Port
- Flow label
- IPv6 DSCP service

6.6.2.4. MAC and IP ACLs

Redirect Interface

A user configures an ACL *permit* rule to force its matching traffic stream to a specific egress interface, bypassing any forwarding decision normally performed by the device. The interface can be a physical port or a LAG. The redirect interface rule action is independent of, but compatible with, the assign queue rule action.

Flow-Based Mirroring

IP and MAC ACL can mirror traffic that matches a permit rule to a specific physical port or LAG. This is similar to the redirect attribute, except that a copy of the permitted traffic is delivered to the mirror interface while the packet is forwarded normally through the device. Note that the mirror and redirect attributes are mutually exclusive for a given ACL rule.

Using ACLs to mirror traffic is called flow-based mirroring since the traffic flow is defined by the ACL classification rules. This is in contrast to port mirroring, where all traffic encountered on a specific interface is replicated on another interface.

Multiple ACLs per Interface

Multiple ACLs per interface are supported. The ACLs can be combination of layer 2 and/or Layer 3/4 ACLs.

LAG Support

ACL assignment is appropriate for both physical ports and LAGs.

VLAN Support

ACLs can be configured to apply to a VLAN instead of an interface. Traffic tagged with a VLAN ID (either receivetagged or tagged by ingress process such as PVID) is evaluated for a match regardless of the interface on which it is received.

6.6.2.5. Time-Based ACLs

Access to a switch or router can be made more secure through the use of ACLs to control the type of traffic allowed into or out of specific ports. An ACL consists of a series of rules, each of which describes the type of traffic to be processed and the actions to take for packets that meet the classification criteria. Rules within an ACL are evaluated sequentially until a match is found, if any. Every ACL in ICOS is terminated by an implicit “deny all” rule, which covers any packet not matching a preceding explicit rule.

This time-based ACLs feature extends the ICOS ACL capability to dynamically apply an explicit ACL rule defined within an ACL for a predefined time interval by specifying a time range on a per-rule basis within an ACL, so that the time restrictions are imposed on the ACL rule.

The time-based ACLs feature enables network administrators to define when and for what duration an individual rule of an ACL is in effect. A time range is created that defines specific time interval and is referenced by an individual ACL rule so that it is operational only during the specified time range. With time-based ACLs, network administrators have more control over permitting or denying a user access to network resources.

See Section 6.1.16, “Time Ranges Component” for a description of the software component that enables this feature.

6.6.2.6. Policy Dynamic Slice Sizing

This feature provides better utilization of the field processors managed by the ICOS policy manager software. Specifically, there are two primary aspects:

- Utilize the “Virtual Slice Grouping” mechanism when possible. This hardware feature allows field processor slices to be bound together serially, resulting in an increased number of rules possible in a given policy.
- Add intelligence to the policy manager code to enable dynamic slice configuration. This technique adds flexibility to the method by which slices are managed by the policy manager. Newly added policies use the least amount of field processor resources as possible. When field processor resources become exhausted, the policy manager attempts to shuffle policies as appropriate in order to free additional field processor resources, thereby increasing overall rule capacity.

6.6.2.7. ACL Logging

Access list rules are monitored in hardware to either permit or deny traffic matching a particular classification pattern, but the network administrator currently has no insight as to which rules are being *hit*. Some hardware platforms have the ability to count the number of hits for a particular classifier rule. The ACL logging feature allows these hardware hit counts to be collected on a per-rule basis and reported periodically to the network administrator using the system logging facility and an SNMP trap.

The ICOS ACL permit/deny rule specification is augmented with a *log* parameter that enables hardware hit count collection and reporting. Depending on platform capabilities, logging can be specified for deny rules, permit rules, or both. A five minute logging interval is used, at which time trap log entries are written for each ACL logging rule that accumulated a nonzero hit count during that interval. The logging interval is not user configurable.

The hardware platform may support some finite number of counter resources, so it may not be possible to log every ACL rule. An ACL may be defined with any number of logging rules, but the number of rules that are actually logged cannot be determined until the ACL is applied to an interface. Furthermore, hardware counters that become available after an ACL is applied are not retroactively assigned to rules that were unable to be logged (the ACL must be unapplied then reapplied). Rules that are unable to be logged are still active in the ACL for purposes of permitting or denying a matching packet.

6.6.2.8. ACL and Rate Limiting

The user can specify a simple rate limiting rule attribute to apply to inbound and outbound traffic. Traffic that conforming to the specified rate limit is allowed to transmit and nonconforming traffic is dropped. Rate limiting is supported on all QoS-capable interfaces (physical, LAG, and control-plane). Rate limiting in the outbound direction is supported only on platforms that include an Egress Field Processor (EFP).

6.6.2.9. IPv4 and IPv6 Qualifiers

IPv4 and IPv6 ACL commands include keywords and operators that help make the ACL rule more granular and allow the ACL to qualify:

- Packets based on the operators *not equal to, less than, and greater than* for destination L4 port numbers.
- Packets based on the operators *not equal to, less than, and greater than* for source L4 port numbers.
- Fragmented IPv6 packets (packets that have next header field set to 44).
- Routed IPv6 packets (packets that are routed in the switch).
- Packets based on ICMP type, ICMP code, and ICMP message.
- Packets with any of the following TCP flags:
  - ack: Acknowledgment bit set.
  - established: An established connection. A match occurs if the TCP datagram has the ACK or RST bits set.
  - fin: Finished bit set; no more data from sender.
  - psh: Push function bit set.
  - rst: Reset bit set.
  - syn: Synchronize bit set.
  - urg: Urgent pointer bit set.

6.6.2.10. ACL Remarks

Users can use ACL remarks to include comments for ACL rule entries in any MAC ACL. Remarks assist the user in understanding ACL rules easily.

6.6.2.11. ACL Rule Priority

This feature allows user to add sequence numbers to ACL rule entries and re-sequence them. When a new ACL rule entry is added, the sequence number can be specified so that the new ACL rule entry is placed in the desired position in the access list.

6.6.3. Differentiated Services (DiffServ)

With DiffServ, network resources are apportioned based on traffic classification and priority, giving preferential treatment to data with strict timing requirements according to network management policy.

DiffServ is a method of offering quality-of-service treatment for network traffic without the need for a resource reservation protocol. An administrator specifically provisions the network equipment (switches, routers) to identify the following:

- The classes of traffic in the network
- The QoS treatment that the classes of traffic receive

DiffServ controls the traffic acceptance throughout the DiffServ domain, the traffic transmission throughout the DiffServ domain and the bandwidth guarantee within the network nodes. By controlling the acceptance, the transmission and the bandwidth, a policy-based range of services is established.

When the Layer 3 protocol is IPv4, traffic can be classified based on the following criteria:

- Class of Service
- Secondary Class of Service
- Destination IP Address
- Destination Layer 4 Port
- Destination MAC Address
- EtherType
- IP DSCP
- IP Precedence
- IP ToS
- Protocol
- Reference Class
- Source IP Address

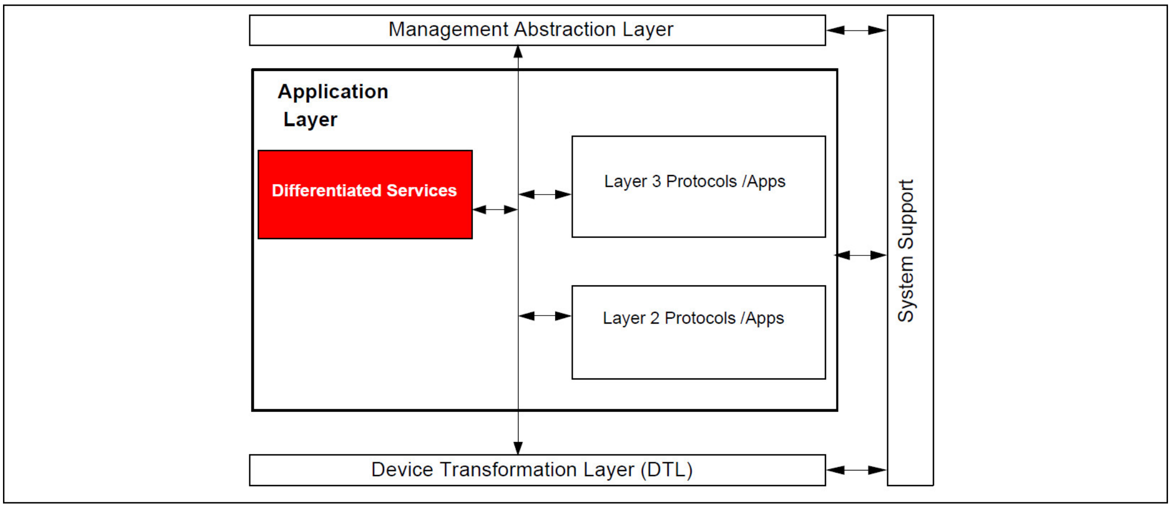
- Source Layer 4 Port
- Source MAC Address
- Secondary VLAN
- VLAN

When the layer 3 protocol is IPv6, traffic can be classified based on the following criteria:

- Destination IPv6 Prefix
- Destination Layer 4 Port
- Flow Label
- IP DSCP
- Protocol
- Reference Class
- Source IPv6 Prefix
- Source Layer 4 Port

The following figure shows the DiffServ architecture.

Figure 6.18. Differentiated Services Architecture



6.6.3.1. ECN Support

Explicit Congestion Notification (ECN) is defined in RFC 3168. Conventional TCP networks signal congestion by dropping packets. A Random Early Discard scheme provides earlier notification than tail drop by dropping packets already queued for transmission. ECN marks congested packets that would otherwise have been dropped and expects an ECN capable receiver to signal congestion back to the transmitter without the need to retransmit the packet that would have been dropped. For TCP, this means that the TCP receiver signals a reduced window size to the transmitter but does not request retransmission of the CE marked packet.

ICOS implements ECN capability as part of the WRED configuration process. Eligible packets are marked by hardware based upon the WRED configuration. The network operator can configure any CoS queue to operate in ECN marking mode and can configure different discard thresholds for each color.

6.6.4. Class of Service (CoS)

The ICOS software CoS Queueing feature allows the user to directly configure device queueing and, therefore, provide the desired QoS behavior without the complexities of DiffServ. The CoS feature allows the user to determine the following queue behavior:

- Queue Mapping
  - Trusted Port Queue Mapping
  - Untrusted Port Default Priority
- Queue Configuration

This enables the software to support a wide variety of delay sensitive video and audio multicast applications.

CoS mapping tables, port default priority, and hardware queue parameters may be configured on LAG interfaces as well as physical port interfaces.

In most networking devices, each physical port consists of one or more queues for transmitting packets. Multiple queues per port are often provided to give preference to certain packets over others based on user-defined criteria. When a packet is queued for transmission, the servicing rate depends upon queue configuration and traffic quantity. If a delay is necessary, packets get held in the queue until the scheduler authorizes the transmission. As queues become full, packets are dropped.

The packet drop precedence indicates the packets sensitivity to being dropped during times of queue congestion. The packet drop precedence is often referred to as packet coloring. The packet coloring types are:

- Green — A low drop precedence. Allows the packet to be transmitted under most circumstances
- Yellow — A higher drop precedence. Subjects the packet to dropping when bursts become excessive.
- Red — The highest drop precedence. Discards the packet whenever the queue is congested.

In some hardware implementations, the queue depth can be managed using tail dropping or a weighted random early discard (WRED) technique. These methods often use customizable threshold parameters that are specified on a per-drop-precedence basis.

6.6.4.1. Queue Mapping

The priority of a packet arriving at an interface is used to steer the packet to the appropriate outbound CoS queue through a mapping table. Network packets arriving at an ingress port are directed to one of n queues in an egress port(s) based on the translation of packet priority to CoS queue. The CoS mapping tables define the queue used to handle each enumerated type of user priority designated in either the 802.1p, IP precedence, or IP DSCP contents of a packet. If none of these fields are trusted to contain a meaningful COS queue designation, the ingress port can be configured to use its default priority to specify the CoS queue.

CoS queue mappings use the concept of trusted and untrusted ports.

Trusted Port Queue Mappings

A trusted port is one that takes at face value a certain priority designation within arriving packets. Specifically, a port may be configured to trust one of the following packet fields:

- 802.1p User Priority
- IP Precedence
- IP DSCP

Packets arriving at the port ingress are inspected and their trusted field value is used to designate the COS queue that the packet is placed when forwarded to the appropriate egress port. A mapping table associates the trusted field value with the desired COS queue.

Untrusted Port Default Priority

Alternatively, a port may be configured as untrusted, whereby it does not trust any incoming packet priority designation and uses the port default priority value instead. All packets arriving at the ingress of an untrusted port are directed to a specific COS queue on the appropriate egress port(s) in accordance with the configured default priority of the ingress port. This process is also used for cases where a trusted port mapping is unable to be honored, such as when a non-IP packet arrives at a port configured to trust the IP precedence or IP DSCP value.

Queue Configuration

Flexible CoS queues per port assure the lowest latency to high priority traffic. CoS queue characteristics such as minimum guaranteed bandwidth and transmission rate shaping are configurable at the queue (or port) level. The CoS queue configuration is global or per-interface. The ICOS CoS application translates the user configuration information into a DAPI data structure that is passed to the HAPI component for processing. The specific HAPI implementation sets up the hardware CoS queues using the appropriate device interactions.

Queue configuration involves setting the following hardware port egress queue configuration parameters:

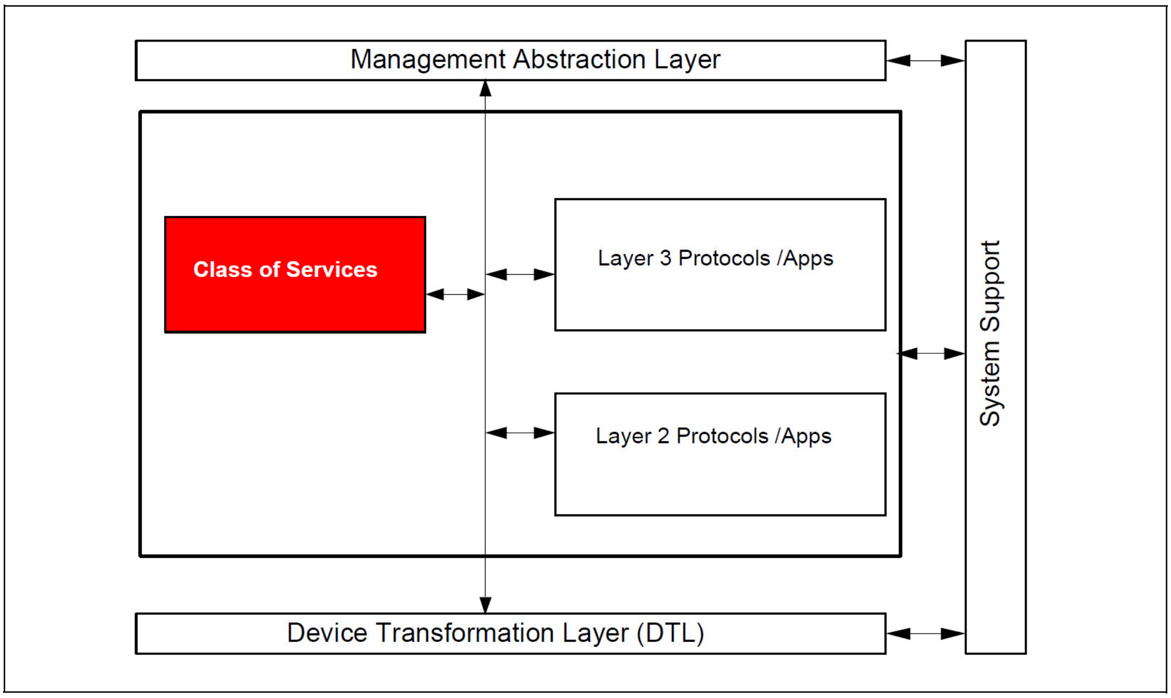
- Scheduler type: strict versus weighted
- Minimum guaranteed bandwidth
- Maximum allowed bandwidth (that is, shaping)
- Queue management type: tail drop versus WRED
- Tail drop parameters: threshold
- WRED parameters: minimum threshold, maximum threshold, drop probability

Defining these on a per-queue basis allows the user to create the desired service characteristics for different types of traffic. The tail drop and WRED parameters are specified individually for each supported drop precedence level.

In addition, the following are specified on a per-interface basis:

- Queue management type: tail drop versus WRED (only if per-queue config not supported)
- WRED decay exponent

Figure 6.19. Class of Services Architecture



6.7. IP Multicast Module

Installation of the ICOS software Routing component is a prerequisite for Multicast. The Multicast component is best suited for video and audio traffic requiring multicast packet control for optimal operation. The Multicast component includes support for IGMPv2, IGMPv3, PIM-DM, PIM-SM, and DVMRP.

Communication from point to multipoint is called Multicasting. The source host (point) transmits a message to a group of zero or more hosts (multipoint) that are identified by a single IP destination address. Although the task may be accomplished by sending unicast (point-to-point) messages to each of the destination hosts, multicasting is the more desirable method for this type of transmission.

A multicast message is delivered to all members of its destination host group with the same best-efforts reliability as regular unicast IP messages. The message is not guaranteed to arrive intact at all members of the destination group or in the same order relative to other messages.

The advantages of multicasting are explained below:

- Network Load Decrease: A number of applications are required to transmit packets to hundreds of stations. The packets transmitted to these stations share a group of links on their paths to their destinations. Multicast transmission can conserve much needed network bandwidth, since multicasting transmission requires the transmission of only a single packet by the source and replicates this packet only if it is necessary (at forks of the multicast delivery tree).

- Discovery of resources: A number of applications require a host to find out whether a certain type of service is available. Internet protocols such as Bootstrap Protocol (BOOTP) and Open Shortest Path First (OSPF) protocol are among these applications. Using multicast messages and sending the query to those hosts which are potentially capable of providing this service speeds the gathering of this information considerably. Although a group of hosts residing on the same network are the intended target for the majority of multicast packets, this limitation is not mandatory. Discovering the local domain-name server is the intended use of multicast messages on remote networks when there is less than one server per network.

- Applications used for datacasting: Since multimedia transmission has become increasingly popular, multicast transmission use has increased. Multicast transmission may be used to efficiently accommodate this type of communication. For instance, the audio and video signals are captured, compressed and transmitted to a group of receiving stations. Instead of using a set of point-to-point connections between the participating nodes, multicasting can be used for distribution of the multimedia data to the receivers. The participating stations are free to join or leave an audio-cast or a video-cast as needed. The variable membership maintenance is managed efficiently through multicasting.

This section describes the following multicast features:

- Section 6.7.1, “Internet Group Management Protocol”

- Section 6.7.2, “Multicast Listener Discovery”

- Section 6.7.3, “Distance Vector Multicast Routing Protocol”

- Section 6.7.4, “Protocol Independent Multicast—Dense Mode (PIM-DM)”

- Section 6.7.5, “Protocol Independent Multicast—Sparse Mode (PIM-SM)”
- Section 6.7.6, “General Restrictions for IP Multicast Configuration”
- Section 6.7.7, “Multicast Static Routes (MRoutes)”
- Section 6.7.8, “Serviceability for Multicast”

6.7.1. Internet Group Management Protocol

Internet Group Management Protocol (IGMP) is the multicast group membership discovery protocol used for IPv4 multicast groups. Three versions of IGMP exist. Versions one and two are widely deployed. Since IGMP is used between end systems (often desktops) and the multicast router, the version of IGMP required depends on the end-user operating system being supported. Any implementation of IGMP must support all earlier versions.

The following list describes the basic operation of IGMP, common to all versions. A multicast router can act as both an IGMP host and an IGMP router and as a result can respond to its own IGMP messages. The ICOS implementation of IGMPv3 supports the multicast router portion of the protocol (that is, not the host portion).

It is backward compatible with IGMPv1 and IGMPv2.

- One router periodically broadcasts IGMP Query messages onto the network.
- Hosts respond to the Query messages by sending IGMP Report messages indicating their group memberships.
- All routers receive the Report messages and note the memberships of hosts on the network.
- If a router does not receive a Report message for a particular group for a period of time, the router assumes there are no more members of the group on the network.

All IGMP messages are raw IP data grams and are sent to multicast group addresses, with a time to leave (TTL) of 1. Since raw IP does not provide reliable transport, some messages are sent multiple times to aid reliability.

IGMPv3 is a major revision of the protocol and provides improved group membership latency. When a host joins a new multicast group on an interface, it immediately sends an unsolicited IGMP Report message for that group. IGMPv2 introduced a Leave Group message, which is sent by a host when it leaves a multicast group for which it was the last host to send an IGMP Report message. Receipt of this message causes the Querier possibly to reduce the remaining lifetime of its state for the group, and to send a group-specific IGMP Query message to the multicast group. The Leave Group message is not used with IGMPv3, since the source address filtering mechanism provides the same functionality.

IGMPv3 also allows hosts to specify the list of hosts from which they want to receive traffic. Traffic from other hosts is blocked inside the network. It also allows hosts to block packets for all sources sending unwanted traffic.

IGMPv3 adds the capability for a multicast router to learn which sources are of interest to neighboring systems for packets sent to any particular multicast address. This information gathered by IGMP is provided to the multicast routing protocol (that is, DVMRP, PIM-DM, and PIM-SM) that is currently active on the router in order to ensure multicast packets are delivered to all networks where there are interested receivers.

The ICOS implementation of IGMP supports only the multicast router portion of the protocol and not the listener portion.

6.7.2. Multicast Listener Discovery

Multicast Listener Discovery (MLD) protocol enables IPv6 routers to discover the presence of multicast listeners, the nodes who wish to receive the multicast data packets on its directly attached interfaces. On IPv6 multicast routers, MLD replaces the functionality performed by IGMP on IPv4 networks.

MLD discovers which multicast addresses are of interest to its neighboring nodes and provides this information to the active multicast routing protocol that makes decisions on the flow of multicast data packets.

The ICOS implementation of MLD v2 supports only the multicast router portion of the protocol and not the listener portion. It is backward-compatible with MLD v1.

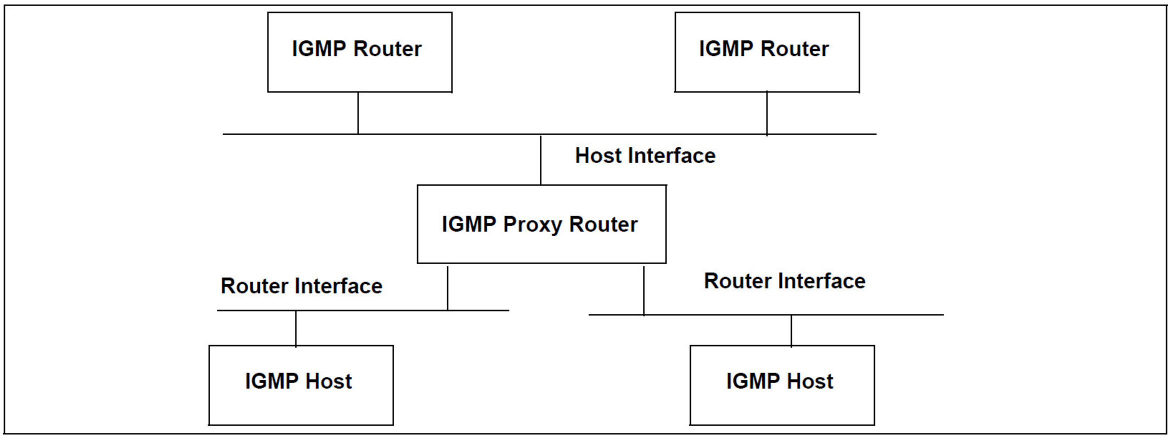
6.7.2.1. IGMP and MLD Proxy

IGMP Proxy is used by the router on IPv4 systems to enable the system to issue IGMP host messages on behalf of hosts that the system discovered through standard IGMP router interfaces, thus acting as proxy to all its hosts residing on its router interfaces.

MLD proxy is used by the router on IPv6 systems to enable the system to issue MLD host messages on behalf of hosts that the system discovered through standard MLD router interfaces, thus acting as proxy to all its hosts residing on its router interfaces.

The following diagram depicts a typical IGMP Proxy router implementation that caters the IGMP proxy functionality to the multicast hosts present on the network associated to the downstream interfaces and has the IGMP router functionality operational on the downstream interfaces. While the upstream interface of the IGMP Proxy reports the consolidated group membership information collected from the downstream interfaces to the upstream IGMP routers.

Figure 6.20. IGMP Proxy Implementation



ICOS supports Router implementation of IGMP Version 3, Version 2, and Version 1. Version 3 adds support for source filtering [SSM] and needs to be interoperable with Versions 1 and 2. Version 2 supports the group membership terminations to be quickly reported to overcome leave latency and is designed to be interoperable with Version 1.

6.7.3. Distance Vector Multicast Routing Protocol

Distance Vector Multicast Routing Protocol (DVMRP) is a dense mode multicast protocol and is most appropriate for use in networks where bandwidth is relatively plentiful and there is at least one multicast group member in each subnet. DVMRP assumes that all hosts are part of a multicast group until it is informed of multicast group changes. When the dense-mode multicast router is informed of a group membership change, the multicast delivery tree is pruned. DVMRP uses a distributed routing algorithm to build per-source-group multicast trees. It is also called Broadcast and Prune Multicasting protocol. It dynamically generates per-sourcegroup multicast trees using Reverse Path Multicasting. Trees are calculated and updated dynamically to track membership of individual groups.

The DVMRP protocol operates as follows:

1. The first message for any (source, group) pair is forwarded to the entire multicast network, with respect to the packet’s time-to-live (TTL).

2. TTL restricts the area to be flooded by the message.

3. Any leaf router that does not have members on directly attached subnetworks sends back prune messages to the upstream router.

4. The branch that transmitted a prune message is deleted from the delivery tree.

DVMRP transmits two types of messages:

- Protocol messages: These include Probe messages, Prune messages, Graft and Graft Acknowledgements, and so on, which are sent as multicast packets. Most protocol messages are addressed to ALL-DVMRP-ROUTERS. If a router in the path does not support multicasting, it encapsulates these packets in unicast IP packets. Encapsulated packets are tunneled through nonmulticast routers and decapsulated by the multicast router at the end of the tunnel.

- Routing messages: These are unicast messages used to exchange routing information. They are typically sent by a neighboring router when it is restarted, has lost its routing information, or has come up from a down state. They are also transmitted when a graft is sent in response to a new member’s subscription to an existing group, or when a new group is subscribed.

When messages arrive, the reverse path to the source of the message is discovered by examining the routing table. If the message arrived on the interface that would be used to transmit the message back to the source, the message is transmitted to downstream routers. Otherwise, the message is not on the optimal delivery tree, so the packet is dropped. In this way, loops and duplicates are filtered. DVMRP discovers its neighbors by periodically sending probe messages with a TTL of one (1). These probe messages contain the list of neighboring DVMRP routers from which a probe has been received. With this, a two-way neighbor relationship is established.

When two different routers are connected to the same multiaccess network, there is a possibility of getting duplicate packets. To avoid getting duplicate packets, nominate a Designated Forwarder (DF) for each network. The router with the least metric is the designated forwarder. In case of a tie, the router with the lower IP address is the designated forwarder. Once a designated forwarder is elected, the multicast trees are built. If the destination group of a packet exists in the local database and the router on which the packet arrived is the designated forwarder, then that interface is added to the downstream interface list.

The edge routers remove interfaces with no group members from their multicast trees. If all the downstream interfaces are removed, the router sends a prune message to its upstream neighbors. Every prune message has a lifetime, after which the interface is joined back onto the delivery tree. If the unwanted datagrams still appear, the prune is initiated again. If a host joins a previously pruned branch of a tree, the DVMRP routers use graft messages to cancel the previous prunes.

6.7.3.1. Limitations

- ICOS DVMRP does not support DVMRP tunnels.
- ICOS supports only the IPv4 version of DVMRP.

6.7.4. Protocol Independent Multicast—Dense Mode (PIM-DM)

Protocol Independent Multicast (PIM) protocols are not dependent on any particular unicast routing protocols to construct forwarding information for multicast packets, although unicast information is needed for forwarding packets. The Dense Mode (DM) version of PIM is most appropriate for networks with relatively plentiful bandwidth and with at least one multicast member in each subnet.

PIM-DM assumes that all hosts are part of a multicast group and forwards packets to hosts until informed that group membership has changed. A group membership change results in the multicast delivery tree being pruned.

The PIM-DM protocol operates as follows:

1. The first message for any (source, group) pair is forwarded to the entire multicast network, with respect to the time-to-live (TTL) value in the packet.
2. TTL restricts the area flooded by the packet.
3. All leaf routers with no members in a directly attached subnet send prune messages to the upstream router.
4. Any branch for which a prune message is received is deleted from the delivery tree.

PIM-DM uses Reverse Path Forwarding (RPF), which is the fundamental concept in multicast routing that enables routers to correctly forward multicast messages down the distribution tree. RPF makes use of the existing unicast routing table to determine the upstream and downstream neighbors and build a source-based shortest-path distribution tree. A router forwards a multicast message only if the multicast message is received on the upstream interface. This RPF check helps to guarantee that the distribution tree is loop-free.

The multicast messages contain the source and group information so that downstream routers can build up their multicast forwarding tables. If the source goes inactive, the tree is torn down. Multicast messages arriving at a router over the proper receiving interface (that is, the interface that provides the shortest path back to the source) are forwarded on all downstream interfaces until unnecessary branches of the tree are explicitly pruned. In addition to the prune messages, PIM-DM uses graft messages and assert messages. Graft messages are used when a new host wants to join a group, and assert messages are used to shut off duplicate flows.

To minimize the repeated flooding of datagrams and the subsequent pruning associated with a particular (S,G) pair, PIM-DM uses a State Refresh message. This message is sent by the router(s) directly connected to the Source and is propagated throughout the network. When received by a router on its RPF interface, the State Refresh message causes an existing prune state to be refreshed. These State Refresh messages are generated periodically by the router directly attached to the Source.

Routing must be enabled on the switch and the applicable interfaces before PIM-DM becomes enabled and operational. The switch supports PIM-DM Version 2. The PIM-DM feature supports distributing both IPv4 and IPv6 routes.

6.7.5. Protocol Independent Multicast—Sparse Mode (PIM-SM)

As with PIM-DM, PIM-SM is not dependent on any particular unicast routing protocols to construct the forwarding information it uses for multicast packets, although unicast information is needed for actual forwarding. The Sparse Mode (SM) version of PIM is most appropriate for networks with relatively limited bandwidth and where group membership is widely distributed across regions.

Sparse mode protocols begin with the assumption that few routers in the network is involved in any given multicast path. Sparse mode routers minimize network traffic by adding branches to the tree only when explicitly requested to do so. Therefore, sparse mode protocols such as PIM-SM are better suited to WANs than are dense mode protocols.

PIM-SM uses the following concepts:

- Rendezvous Point (RP): The root of a shared distribution tree down which all multicast traffic flows.
- Designated Router (DR): Responsible for sending join messages to the RP for group members and for sending register messages to the RP for sources.

PIM-SM uses shared trees by default and implements source-based trees for efficiency. PIM-SM assumes that none of the hosts want multicast traffic unless they specifically ask for it. It creates a shared distribution tree centered on a defined RP from which source traffic is relayed to the receivers. Senders first send the multicast data to the RP, which, in turn, sends the data down the shared tree to the receivers. Shared trees centered on an RP do not necessarily provide the shortest or optimal path. In such cases, PIM-SM provides a means to switch to more efficient source-specific trees.

As soon as an active source sends a packet to its DR, the DR is responsible for registering the source with the RP and requesting that the RP build a tree back to that DR. The DR encapsulates the multicast data from the source in a Register message and unicasts that data to the RP.

PIM-SM uses the explicit join model whereby hosts wishing to receive multicast packets send PIM join messages to the RP using the DR. When a DR gets a membership indication for a new group, it looks up the RP associated with the group and sends a join message to the RP.

Upon reception of the first data packet, the DR at the receiver's end automatically switches to the source's shortest path tree by initiating a PIM Join towards the source. After sending the PIM Join (as said above), the DR also prunes the RP tree by initiating a PIM Prune message towards the RP. This ensures that the multicast data flows only on the source's shortest path tree and does not flow down the shared RP tree.

The PIM-SM feature supports distributing both IPv4 and IPv6 routes. For information about the PIM-SM Bootstrap Router (BSR) mechanism, see *draft-ietf-pim-sm-bsr-05*.

6.7.5.1. RP Failover

RP Fast Failover

An RP that is configured as a Candidate (C-RP) for a particular multicast group range sends periodic C-RP advertisements to the BSR. The default interval for the PIM C-RP advertisements to be sent is 60 seconds. The C-RP message format also includes a Holdtime field, which is 2.5 times the C-RP Advertisement interval. The BSR considers that a C-RP is down if it does not hear from the C-RP for this Holdtime period.

Multicast traffic recovery with PIM-SM in the event of an RP failover takes around 150 seconds (for the BSR to detect that the elected RP is down). This is because of the default time interval (60 seconds) that ICOS uses for transmitting Candidate-RP advertisements to the BSR. In general, interval does not work well with multicast applications that are interested in high availability.

To achieve RP fast failover, whereby the BSR quickly detects that a C-RP is down and reelects a new RP from among the available C-RPs, the Candidate-RP advertisement interval is made to be configurable.

For example, if the user configures the Candidate-RP advertisement interval to 1 second, in the event of the elected RP failure, the BSR detects that the C-RP is down in 2.5 seconds.

Due to queue lengths and processing delays on the switch platforms, lower intervals have been known to cause problems. Also, keep in mind that a router with 30 LAN segments and a C-RP message interval of 1 second will need to send out 30 PIM C-RP messages every second. The processing involved by the receiving router to process the received message and subsequently to forward this message should also be considered.

The goal is not to set the C-RP advertisement interval too low so that it leads to unnecessary flapping. It is recommended that the administrator configures this value to lower intervals only depending on the topological and the end-application requirements, as it may be taxing on the CPU.

BSR Fast Failover

As the RP and BSR go hand-in-hand, effort is also made to support BSR fast failover.

A BSR configured as a Candidate (C-BSR) periodically multicasts C-BSR advertisements to the network. Upon receiving the C-BSR messages, the routers run an algorithm and elect one C-BSR as the BSR for the network. The default interval for the PIM C-BSR advertisements to be sent is 60 seconds. The C-BSRs will time-out a BSR if they do not hear from that BSR for 2.2 times the C-BSR advertisement interval.

To achieve BSR fast failover, whereby the C-BSRs to quickly detect that a BSR is down and re-elect a new BSR from among the available C-BSRs, the Candidate-BSR advertisement interval is made to be configurable.

For example, if the user configures the Candidate-BSR advertisement interval to 5 seconds, the interval after which a BSR is considered down (if no BSR message is received from this BSR) is 11 seconds.

The concerns highlighted in Section 6.7.5.1, “RP Failover” are applicable to this section too.

The improvements described in Section 6.7.5.1, “RP Failover” and this section are for RP and BSR failover cases alone, and these guarantee only that the RP or BSR failover is achieved as per the formula/algorithm specified in these sections.

RPs and BSRs generally sit in the core network so that they are easily accessible. It is recommended that the administrator do not lower the C-RP message interval or the C-BSR message interval in the access network, as doing so uses more network bandwidth and CPU cycles.

To improve the multicast traffic recovery times in the network, the following recommended best practices may also be followed. It is recommended that administrators follow these practices, subject to the topological requirements and as the particular demands of the network.

PIM Hello Interval Tuning

The *ip pim hello-interval* command controls the interval that a PIM hello packet is transmitted out each PIMenabled interface.

The PIM hello packets are used to discover PIM neighbors and to determine the Designated Router (DR) on each network segment. The default interval for the PIM hello packets to be sent is 30 seconds. A PIM neighbor is considered down after three consecutive missed messages. Therefore, it could take 90 seconds for the DR to failover. If the query interval is lowered to 1 second, then the DR failover time is reduced to 3 seconds. A toolow value for the hello message interval is not recommended, as it would be taxing on the CPU.

Register Rate Limits

When a new source starts transmitting in a PIM Sparse Mode network, the packets are encapsulated and unicasted to the Rendezvous Point (RP). This process can be taxing on the CPU of the Designated Router (DR) and the RP if the source is running at a high data rate or there are many new sources starting at the same time. This scenario can potentially occur immediately after a network failover.

To protect both the edge routers and the RP, it is recommended to set the *ip pim register-rate limit* to a relatively low value. Normally, there is no limit to the number of packets that can be encapsulated and sent to the RP. The limit will depend on the number of potential sources registering at the same time and their data rate. A typical setting in a PIM Sparse Mode (PIM-SM) network is between four and 10 messages per second.

Unicast Routing Convergence

For a shared tree, the RPF interface provides the best reverse metric to the RP. The RPF interface details are provided by the unicast routing protocol. The unicast routing protocol convergence time has a direct impact on the multicast traffic recovery in router failure scenarios. Therefore, efforts have to be made to ensure that the unicast routing protocol converges before the PIMSM converges, namely, lowering the duration of the heartbeat messages, such as Hello timers and Dead intervals in the case of OSPF. Keep in mind that this helps in faster recovery but comes at a CPU cost due to the control processing load.

PIM Join/Prune Interval Tuning

PIM-SM sends periodic Join/Prune messages to its upstream (RPF) routers to keep the multicast channel alive. The default period is 60 seconds. This value may need to be turned down (using the *ip pim join-prune interval* command) in scenarios to help faster PIMSM convergence. Keep in mind that this would be taxing on the CPU.

Multicast Traffic Recovery Time

In general, the multicast traffic recovery time is dominated by the time required to reconstruct the unicast routing table. If possible, the unicast routing protocol parameters should be tuned to allow rapid detection of topology changes and prompt updating of the routing table. PIM generally generates a high control load during the recovery time, and this is directly proportional to the number of multicast route entries (mostly receivers) that need to be recovered.

6.7.5.2. PIM Register Mechanism Optimization and Supported Topologies

With PIMSM, a user can configure a router as a PIM RP even if it is multiple hops away from the multicast source.

PIM Register Performance

When a multicast data source starts sending data destined for a multicast group, the FHR takes those data packets, unicast-encapsulates them, and sends them directly to the RP. The RP receives these encapsulated data packets, decapsulates them, and forwards them onto the shared tree. For PIM, ICOS software supports the following behavior:

- ICOS behavior as a First hop router

When a multicast data source (S) starts sending data destined for a multicast group (G), the FHR receives these packets and traps them to the CPU. The FHR immediately installs a negative entry in the hardware to prevent further data packets reaching the CPU. The FHR then unicast-encapsulates the first received data packet in the form of a PIM Register message, and software forwards it to the RP. The FHR sends only two such Register messages to the RP.

After the FHR receives the PIM Join from the RP, the negative entry is replaced with a native forwarding entry, so that subsequent data packets are forwarded in the hardware.

If the initial Register messages (two of them) do not reach the RP or the PIM Join sent in response does not reach the FHR, the data stream never gets forwarded. So, the negative entry is timed out and removed after

3 seconds so that the data packets are trapped to the CPU, and this entire process initiates again.

- ICOS behavior as a RP

Upon the reception of the first Register message from the FHR, RP decapsulates it and forwards the packet onto the shared tree. RP then immediately switches to the shortest path tree (SPT) by initiating a PIM (S,G) Join towards the FHR. It follows up the PIM (S,G) Join by also sending a Register Stop message to the FHR (not waiting for the duplicate data packets from the FHR as RFC 4601 says).

ICOS software utilizes the (* ,G) based forwarding feature of the hardware. RP installs a (* ,G) forwarding entry in the hardware when a shared tree is built initially, and updates it whenever the receivers join or leave. After switch-over, when the RP receives native multicast data, all the data packets are now switched in the hardware, thereby minimizing the packet loss.

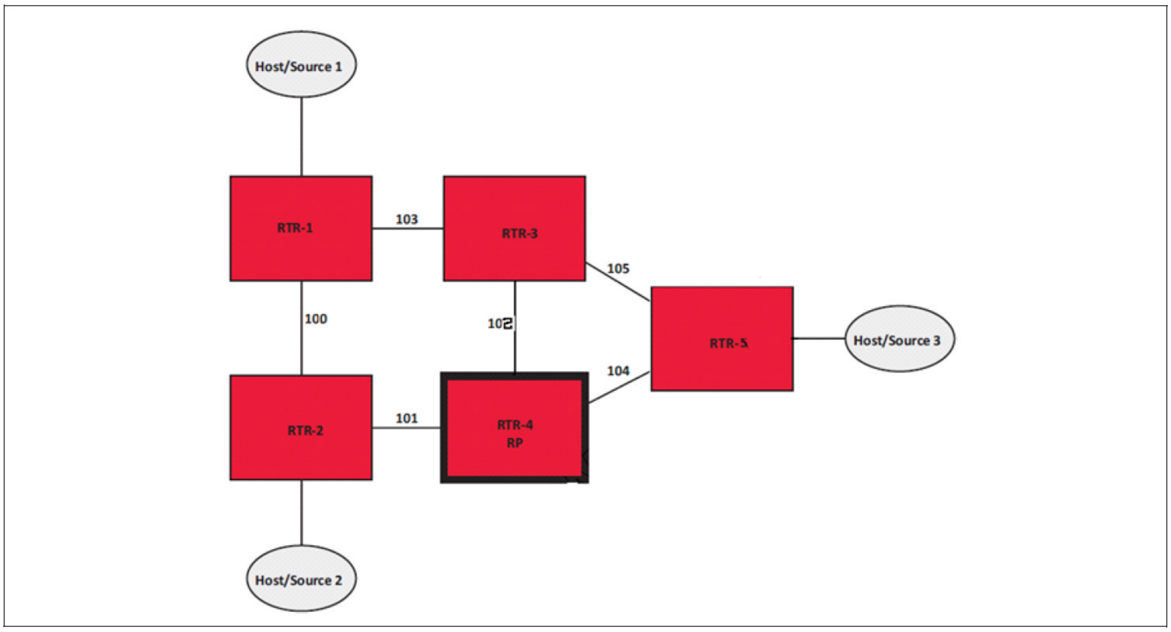
Network Failover Scenarios

ICOS addresses the possible network failover scenarios, as explained in the following example scenario.

Example Scenario — FHR Not Being the RP

This example shows a scenario where an intermediate router on the shortest path tree fails. It uses the network topology shown in the figure below:

Figure 6.21. Network Failover Example



In the topology shown in the figure above, RTR-4 is the RP. Source-1 (S) is the multicast source transmitting data for a multicast group (G).

Host 3 announces interest in receiving data for the multicast group G by sending a IGMPv2 Join to RTR-5. RTR-5 initiates creation of the shared tree by sending a PIM (* ,G) Join towards the RP, and creates a (* ,G) entry in their route table.

After Source-1 starts sending data, RTR-1 (the DR adjacent to the source) encapsulates the multicast data in PIM Register messages and unicasts it to RP (RTR-4). RTR-4 decapsulates the data packets and forwards them natively to RTR-5. RTR-4 (RP) then initiates a switch-over to the source tree by sending an (S,G) Join towards the source, and following it up with a Register-stop message to stop RTR-1 from sending further encapsulated Register messages.

The unicast route table at RTR-5 shows that the best path towards Source-1 is via RTR-3, whereas RTR-5 is receiving data via RTR-4 on a suboptimal (longer) path. At the Last hop router, ICOS, on receiving the first data packet, initiates a switch-over to the source tree by sending an (S,G) Join towards the source. It prunes the RP tree by sending a (*,G) Prune towards the RP. This enables Host-3 to receive multicast traffic from Source- 1 on the optimal path.

Now, consider a case where the network fails for a while. Assume that RTR-3 goes down and, as a result, the multicast source tree that was built before (RTR-5 to RTR-1) is broken and Host-3 stops receiving multicast data. Note that the shared tree (RTR-5 to RTR-4) is intact, as the network did not fail in this path.

6.7.5.3. PIM (S,G,rpt) Upstream and Downstream State Machine Transitions

The ICOS PIM-SM implementation supports the state machine transitions and events for the PIM (S,G,rpt) upstream and downstream state machines as described in RFC 4601, including the following two state machines to handle certain failover scenarios in large multicast topologies:

- PruneDesired(S,G,rpt) → TRUE: After the PIM (last-hop) router switches to the shortest path, it might want to send a *prune* towards the RP. Triggering this event enables the (S,G,rpt) FSM to transition to the *pruned* state.
- PruneDesired(S,G,rpt) → FALSE: After the shortest path fails over, the PIM (last-hop) router might again want to receive the multicast stream from the RP by sending a (S,G,rpt) *join* towards the RP. Triggering this event enables the (S,G,rpt) FSM to transition to the *NotPruned* state.

6.7.6. General Restrictions for IP Multicast Configuration

This section describes the general restrictions related to the IP Multicast configuration in ICOS.

- Enabling IPv6 Multicast Routing1

To globally enable IPv6 multicast routing, you must use the *ip multicast* command. There is no separate command to enable IPv6 multicast.

In Global mode:

```
(router) (Config) #ip multicast
```

- Only one multicast routing protocol for IPv4 and one multicast routing protocol for IPv6 can be enabled on the router. Multiple routing protocols cannot run simultaneously.

In Global mode:

```
(router) (Config) #ip pim sparse
```

With the above configuration, the IPv4 version of PIM sparse mode is enabled on the router. When an IPv4 version of a multicast routing protocol is enabled, the user cannot enable the IPv4 version of another multicast routing protocol. If the user tries to do so, an error message displays, as follows:

```
(router) (Config) #ip pim dense
“Failed: Multicast Routing Protocol (PIMSM) is already configured.”
```

A similar restriction applies to the IPv6 version of the multicast routing protocols.

- ICOS does not support multicast on loopback interfaces.

6.7.7. Multicast Static Routes (MRoutes)

The IP Multicast Static Route (MRoute) feature allows multicast paths to diverge from the unicast paths. When using PIM, the router expects to receive packets on the same interface it uses to send Unicast packets back to the source. Static MRoute allows in Reverse Path Forwarding (RPF) selection only. It describes the RPF that the Multicast traffic should take.

Mroutes are most frequently used when the Unicast network does not have the same topology as the multicast network. There are two main reasons why this might be the case:

- Tunnels that bypass the nonmulticast sections of the network.
- Forcing multicast traffic to take a different path than Unicast traffic.

Mroutes are similar to unicast static routes but differ in the following ways:

- Static Mroutes are used to calculate RPF information (not to forward traffic).
- Static Mroutes cannot be redistributed.

6.7.8. Serviceability for Multicast

The multicast feature supports several CLI commands to help troubleshoot and resolve issues that involve multicast traffic.

Debug commands are available to allow packet tracing for each of the multicast components, namely DVMR, Multicast Forwarder (MFC) (IPv4 and IPv6), and PIMSM (IPv4 and IPv6), for both reception and transmission. The output of these debug commands is displayed on the console of the session on which the command is executed. The output is also submitted to Syslog with a severity level of Debug. The traces include protocol packet information for the respective address family for which the trace is enabled.

The debug commands for multicast support control and data packet exchange. These debug commands allow tracing either the received packets, transmitted packets, or both RX and TX for a particular multicast protocol based on the debug configuration for that protocol. The specific IPv4 and IPv6 commands allow debugging packets for the specified address family.

In addition to debug commands, a CLI command is available to allow an administrator to manually remove IPv4 and IPv6 multicast route entries from the MFDB that have become stale or stuck.

Chapter 7. Device Transformation Layer

This section describes the architectural design and implementation of the device transformation layer (DTL). The DTL interacts with three major components of the software: the application layer, the advanced network device layer, and system support. The DTL is the middleware that facilitates communication of both data and control information between the application layer and the advanced network device layer. Figure 2.1, “System Layers” displays the position of the DTL in relation to the ICOS architecture.

This section contains the following subsections:

- Section 7.1, “DTL Architecture”

7.1. DTL Architecture

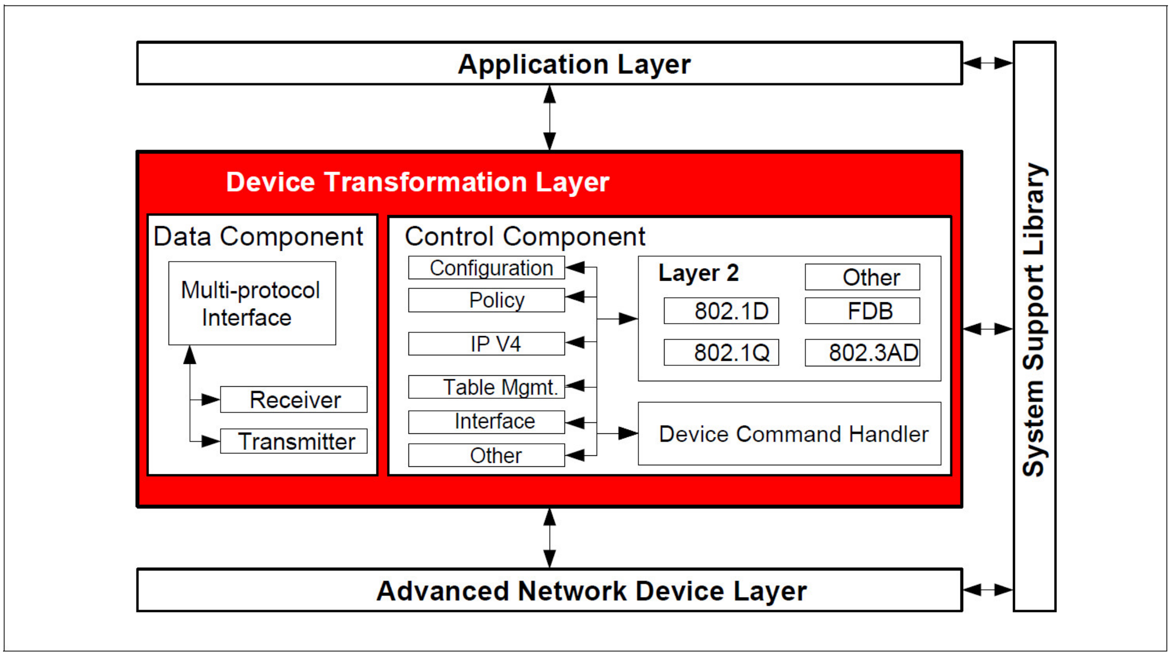
The architecture of the DTL is shown in the figure below. The figure depicts the interaction between the DTL and the following components:

- System Support
- Application Layer
- Advanced Network Device Layer

The internal architecture of this layer consists of the two major components shown in the figure below:

- DTL Data Component: Administers the data flow between the application layer and the advanced network device layer.
- DTL Control Component: Administers the control (settings, parameters, and so on) flow between the application layer and the advanced network device layer.

Figure 7.1. Device Transformation Layer Block Diagram



Chapter 8. Advanced Network Device Layer

This section provides an overview of the advanced network device layer (ANDL), in which the network driver operates, and explains its major architectural components.

The overall architecture figure, Figure 2.1, “System Layers”, shows the relationship between an Advanced Network Driver and the rest of an ICOS system. The driver interacts with two major system components: the Device Transformation Layer (DTL) and System Support. All communication with the networking device takes place through the network driver. The communication between the network driver and the hardware, using the Local Hardware Interface, is hardware-specific and varies with the implementation.

This section contains the following subsections:

- Section 8.1, “Driver Structure”
- Section 8.2, “Directory Structure”
- Section 8.3, “Packages”
- Section 8.4, “Hardware Support APIs”
- Section 8.5, “Memory Management”
- Section 8.6, “Packet Flow”
- Section 8.7, “Slot-Port Numbering”
- Section 8.8, “Initialization”
- Section 8.9, “Command Families”

8.1. Driver Structure

The network driver consists of the Device Application Programming Interface (DAPI) and the Hardware Abstraction Programming Interface (HAPI). The DAPI code provides a hardware-independent interface for the application code, and does not need to be changed when an ICOS implementation is ported to new hardware. The DAPI code has been used unchanged for several different implementations. The HAPI code contains all of the hardware-specific code needed to interface directly to a specific hardware implementation. If a network processor is part of an implementation, the microcode is part of the HAPI module.

The HAPI module in the sample device driver is a skeleton driver that contains all of the data structures and function prototypes required for an actual driver implementation.

The DAPI code interfaces to the HAPI code using function pointers. These pointers are hard-coded in the various initialization routines and must be changed to point to the new functions when the HAPI code is changed. See Section 8.8, “Initialization” for more information on initialization.

8.2. Directory Structure

ICOS code uses one main directory tree. The subdirectories for the driver are *\bsp* and *\andl*.

- The *\bsp* directory contains the code and data structures needed to provide support for a given board:
  - The *\bsp\cpu* directory provides support for the CPU complex which includes file system support, interrupt assignment, PCI configuration, and optionally CPLD support.
  - The *\bsp\platform* directory supports the board configuration. This includes *hpc_card_db.h* and *hpc_unit_db.h* which describe the card layout of the system. Constants specifying the maximum number of VLANs, layer 2 MAC addresses, and so on, are defined based on the switch chip family in the files *bcm562xx.h*, *robo.h*, *xgs3.h*, or *xgs4.h*.
- The *\andl* directory contains the code and structures for the network driver. The *\andl\dapi* directory contains the code for the Device Application Programming Interface (DAPI) and the hardware independent API. The \andl\hapi directory contains the code for the Hardware Abstraction Programming Interface (HAPI), the hardware dependent code.

8.3. Packages

ICOS code is grouped into independent packages containing specific functionality. Base and Switching functionality are always included. The Routing package is created by using initialization functions and a directory structure. There are #ifdefs in the initialization code that are used to control the inclusion of other functionality, in flex packages.

The Routing package is created by putting *hapiXXXXL3Init* into a new directory. In the layer 2 only package, the file and function exist but are simply stubs (returning success). In the Routing package, *hapiXXXXL3Init* initializes the Layer 3 data and function table. The function table is preinitialized with pointers to error functions; therefore, if the function table is not updated with pointers to actual functions, and the application attempts to call those functions, an error condition occurs and a message describing which function was invoked is printed to the screen by *dapiCtl*.

Inclusion of flex packages is controlled by:

\src\system_support\base\infrastructure\cnfgr\base\cnfgr_flex_packages.c.

This module includes the component api.h files that set the #ifdef variables defining package support. For example,

\src\l7public\common\flex.h

includes the code:

```
#define L7_ROUTING_PACKAGE 1
```

This code is used by the rest of the code to determine whether to include ROUTING package structures and functions.

8.4. Hardware Support APIs

The network driver makes use of the ICOS system APIs to access Operating System services and common system services.

8.4.1. Operating System Abstraction

ICOS code does not make direct operating system calls. An Operating System Abstraction layer (OSAPI) is used to maximize portability between operating systems. Developers should use the appropriate OSAPI call to obtain operating system services such as memory management, timer support, semaphores, or interrupt handling. The OSAPI function definitions are in *\src\l7public\api \osapi.h*, and the functions themselves are in *\os\xxx\osapi\osapi.c*, where xxx names the relevant operating system.

8.4.2. System Support

Common system functions are provided by the System Support API (SYSAPI). These functions include support for handling network buffers, known as mbufs. SYSAPI also includes the initialization functions that populate the slot and port registries. The SYSAPI function definitions are in *\src\l7public\api\sysapi.h* and the functions themselves are in *\src\system_support\base\system\sysapi.c*.

8.5. Memory Management

8.5.1. Memory Allocation

During initialization memory is allocated for the code and static variables, the heap, and global variables. Memory for address tables and buffer pools is taken from the heap, which is managed by the OSAPI code. To avoid fragmentation as much memory allocation as possible should be done at initialization time, using maximum values based on the constants set in *platform.h*. For example, the mbuf pool is allocated as follows:

```
pMbufPool = osapiMalloc(L7_MAX_PORTS_PER_SLOT * L7_MAX_PHYSICAL_SLOTS_PER_BOX
* L7_MAX_MBUFS_PER_PORT * (mtu_size + phy_size) );
```

To maintain operating system independence, the following OSAPI functions should be used when allocating and freeing memory:

8.5.1.1. osapiMalloc

```
void *osapiMalloc(L7_uint32 size)
```

Returns a pointer to a block of memory of the requested size or L7_NULLPTR if the request could not be filled. The memory is initialized to zero. If the operating system in use is Linux, a tag, set to 0xabcd1234, is added at the end of the block.

8.5.1.2. osapiFree

```
void osapiFree(void * memory)
```

Frees a block of memory.

8.5.2. Buffer Management

Up to three different buffer types may exist in the system: buffers used by the hardware, network buffers used for frames handled by the driver, and operating system specific buffers used to pass data between the driver and the application code.

Management of any hardware buffers and the transfer of data between network buffers and hardware buffers or the media, are the responsibility of the hardware-specific code operating below the driver.

Management of network buffers is handled by SYSAPI code; management of application buffers is handled by OSAPI code. The ICOS system has its own buffer management utilities. The buffer management API is located in *\src\l7public\api\sysapi.h*.

The network buffer pool is part of the heap. Network buffers should be referenced using pointers of type *L7_netBufHandle* and have the following header format:

8.5.2.1. SYSAPI_NET_MBUF_HEADER_t

```
typedef struct
```

```
{
L7_uint32 applSpecVar;
L7_uchar8 *bufStart;
L7_uint32 bufLength;
void *osBuffer;
} SYSAPI_NET_MBUF_HEADER_t;
```

- *applSpecVar* — for use by the application code
- *bufStart* — pointer to start of the layer 2 header
- *bufLength* — length of the packet
- *osBuffer* — if not NULL, pointer to the first or only application buffer

The following macros and functions are used for buffer management:

8.5.2.2. DAPI_NET_MBUF_GET

Usage: *netMbufHandle = DAPI_NET_MBUF_GET*

Description: This macro calls *sysapiNetMbufGet* to allocate a buffer from the buffer pool and is used when the driver receives a frame from the hardware.

8.5.2.3. DAPI_NET_MBUF_FREE

Usage: *DAPI_NET_MBUF_FREE(netMbufHandle)*

Description: This macro calls *sysapiNetMbufFree* to return a buffer to the buffer pool.

8.5.2.4. DAPI_NET_MBUF_GET_DATASTART

Usage: *dataStart = DAPI_NET_MBUF_GET_DATASTART(netMbufHandle)*

Description: This macro calls *sysapiNetMbufGetDataStart* to get the pointer (*bufStart) to the start of the layer 2 header in the buffer.

8.5.2.5. DAPI_NET_MBUF_SET_DATASTART

Usage: *DAPI_NET_MBUF_SET_DATASTART(netMbufHandle, dataStart)*

Description: This macro calls *sysapiNetMbufSetDataStart* to set the pointer (*bufStart) to the beginning of the layer 2 header in the buffer. This is especially helpful if there is another header before the layer 2 header that needs to be bypassed before the packet is passed to the application code.

8.5.2.6. DAPI_NET_MBUF_GET_DATA_LENGTH

Usage: *size = DAPI_NET_MBUF_GET_DATA_LENGTH(netMbufHandle)*

Description: This macro calls *sysapiNetMbufGetDataLength* to get the length of the data in the buffer. The macro returns *bufLength* from the buffer header.

8.5.2.7. DAPI_NET_MBUF_SET_DATA_LENGTH

Usage: *DAPI_NET_MBUF_SET_DATALENGTH(netMbufHandle, size)*

Description: This macro calls *sysapiNetMbufSetDataLength* to set the length of the data in the buffer. The macro sets *bufLength* from the buffer header. The macro is used by hapiBroadSend to pad the frame to the minimum Ethernet packet size.

8.5.2.8. DAPI_NET_MBUF_GET_FRAME_LENGTH

Usage: *size = DAPI_NET_MBUF_GET_FRAME_LENGTH(netMbufHandle)*

Description: This macro calls *sysapiNetMbufGetFrameLength* to obtain the total length of the frame; that is, the length of data in the network buffer, as returned by *DAPI_NET_MBUF_SET_DATA_LENGTH*, plus the length of the operating system buffer pointed to by *osBuffer*, unless *osBuffer* is null.

8.5.2.9. DAPI_NET_MBUF_GET_NEXT_BUFFER

Usage: *netMbufHandle = DAPI_NET_MBUF_GET_NEXT_BUFFER(netMbufHandle, blockHandle, bufData, bufSize)*

Description: This macro calls *sysapiNetMbufGetNextBuffer* to get another buffer. The buffer may be an OSAPI or a SYSAPI buffer, depending on the parameters (see Section 8.5.3.1, “Transmit” and Section 8.5.3.2, “Receive” for more details).

8.5.3. Frame Handling

Frame transmission and reception are heavily dependent on the hardware being used. The application code calls *dapiCtl* with *DAPI_CMD_t set to DAPI_CMD_FRAME_SEND* to transmit data and expects to receive frames using the *DAPI_EVENT_FRAME_RX* callback. The implementation of the code to support these functions is hardwaredependent, but some guidelines are given below.

8.5.3.1. Transmit

On transmit, the driver code handles any device-specific items. For example, if a header needs to be prepended to the packet for a particular processor, either the HAPI transmit function prepends the header or the code or hardware below the driver.

The application code passes a SYSAPI buffer to the driver. The *osbuffer* field in the SYSAPI buffer header may point to one or more OSAPI buffers. The OSAPI buffers do not themselves contain data, but their *bufstart* fields point to the location of the data. The driver obtains a SYSAPI buffer (using API_’NET_MBUF_GET_NEXT_BUFFER') and copies the packet data into that buffer before passing the buffer to the hardware. The *hapiXXXSend* frees all buffers using the *DAPI_NET_MBUF_FREE* macro when transmission is completed.

8.5.3.2. Receive

Packets are passed to the application code using the *DAPI_EVENT_FRAME_RX* callback. The driver code is responsible for extracting the VLAN ID and priority data from the frame’s VLAN tag, or returning the default values for the port if no tag is present.

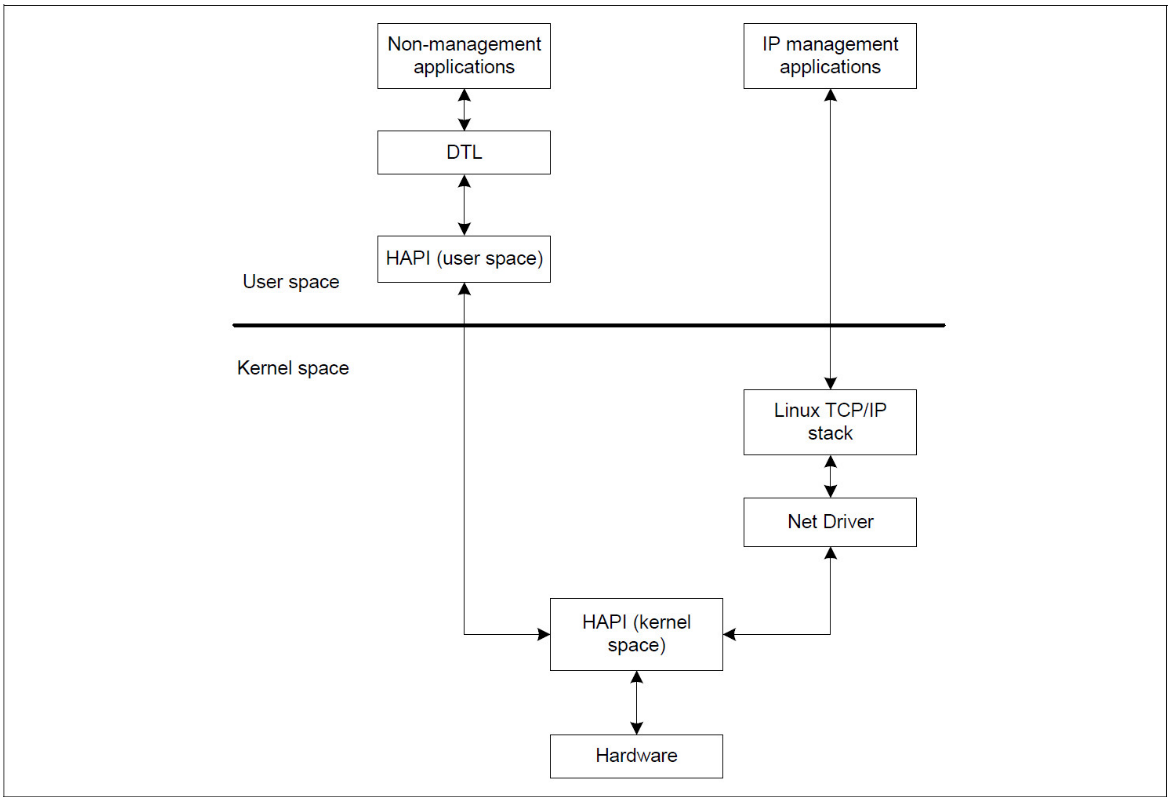
When a packet is received by the hardware, the driver obtains a SYSAPI buffer and copies the packet data into the buffer (all SYSAPI buffers are the same length and are long enough to contain a maximum length Ethernet frame).

8.6. Packet Flow

ICOS has been ported to Linux.

The following illustration shows the basic flow for packets going to and from the CPU:

Figure 8.1. Packet Flow



Consider the following issues regarding the data path in Linux:

General: Using the TAP driver (the Linux TUN/TAP interface), a virtual network interface is created for each IP address that receives packets from the ASIC. This includes the network port interface (the device name used is dtl0).

The Ethernet service port traffic is handled outside the ICOS software platform. Linux directly handles ethernet service port packets.

ICOS software applications that only send and receive IP packets (SNMP, OSPF, and so on) use the normal sockets interface, as any Unix program would. Their packets get sent to these virtual TAP interfaces and received on these virtual TAP interfaces.

ICOS software applications that work at layer 2 (Spanning Tree, MVR, and so on) send their packets to *dtlPduTransmit* and expect to see packets at *dtlPduReceive*, which are tied to *hapiBroadSend* and *hapiBroadReceive* by DAPI.

Receive: When a packet (destined for the CPU) is received by the ASIC, the user-space HAPI code calls *hapiBroadReceive*, which sends the packet to *dtlPduReceive* using *dapiCallback*. DTL determines if the frame is destined for an ICOS software platform layer 2 application. If it is destined for an layer 2 application, DTL sends the packet to the layer 2 application for handling. If the packet is not destined for a layer 2 application, the packet is sent to the appropriate virtual Linux network interface using the TAP driver. The Linux kernel sends the packet to the appropriate socket.

Transmit: When transmitting a packet, a layer 2 application calls *dtlPduTransmit*, which calls *hapiBroadSend*.

If an application sends packets to a socket, the Linux kernel makes them available at the TAP device. The TAP device is monitored by a dedicated task in the ICOS software platform and any packets are passed to *dtlPduTransmit*, which passes the packets to *hapiBroadSend*. The *hapiBroadSend* function sends the packets to the ASIC to be transmitted onto the network.

Summary: When performing a Linux-based MasterDriver® port, a customer must do the following:

- Verify the target’s Linux kernel supports the TUN/TAP driver.
- Verify */dev/tap* is available.
- Use *hapiBroadSend* for sending packets.
- Use *hapiBroadReceive* for receiving packets.
- Verify *hapiBroadReceive* sets up the arguments correctly for calling *dtlPduReceive* using *dapiCallback*.

The ICOS software platform handles the remaining intricacies of packet handling between the ASIC and the CPU.

8.7. Slot-Port Numbering

Both DAPI and HAPI reference physical entities such as cards and ports using a Slot-Port (SP) naming convention. This convention is also used to identify certain logical entities such as Link Aggregation (LAG) interfaces.

The slot number has two uses. In the case of physical ports, the slot number identifies the card containing the ports. In the case of logical and CPU ports, it also identifies the type of interface or port.

8.7.1. Physical Slot Numbers

Physical slot numbers begin with zero and are allocated up to the maximum number of physical slots *L7_MAX_PHYSICAL_SLOTS_PER_BOX* defined in *platform.h*.

8.7.2. Logical Slot Numbers

Logical slots immediately follow physical slots and identify LAG or router interfaces. The maximum number of such slots is defined by *L7_MAX_LOGICAL_SLOTS_PER_BOX* in *platform.h*. The values are as follows:

- *L7_LAG_SLOT_NUM (L7_MAX_PHYSICAL_SLOTS_PER_BOX + 0)*
- *L7_UNUSED_SLOT_NUM (L7_LAG_SLOT_NUM + 1)*
- *L7_ROUTER_SLOT_NUM (L7_UNUSED_SLOT_NUM + 1)*

8.7.3. CPU Slot Numbers

The CPU slots immediately follow the logical slots. The maximum number of CPU slots is defined by *L7_MAX_CPU_SLOTS_PER_BOX* in *platform.h*, which, with the current code packages, should be set to 1. The slot number definition is:

- *L7_CPU_SLOT_NUM (L7_ROUTER_SLOT_NUM + 1)*

The port identifies the specific physical port or logical interface being managed on a given slot.

8.7.4. Physical Ports

The physical ports for each slot are numbered sequentially starting from zero. The maximum number of ports per slot is defined by *L7_MAX_PHYSICAL_PORTS_PER_SLOT* in *platform.h*.

8.7.5. Logical Interfaces

There are two types of logical interfaces: LAG and VLAN routing interfaces.

- LAG interfaces are only used for bridging functions. The number of interfaces that may be defined for a slot is determined by *L7_MAX_LOGICAL_PORTS_PER_SLOT* in *platform.h*. Each LAG interface consists of a set of up to eight physical ports identified by their own USP structure.

- VLAN routing interfaces are only used for routing functions.

8.7.6. CPU Ports

CPU ports are handled by the driver as one or more physical entities located on physical slots. However, only one CPU port is identified to the application, and it is identified by the USP convention as a CPU port on a logical CPU slot.

The USP is defined in the following structure:

```
DAPI_USP_t
typedef struct
{
 L7_ushort16 unit;
 L7_ushort16 slot;
 L7_ushort16 port;
} DAPI_USP_t;
```

8.8. Initialization

Hardware initialization is handled by the boot and bring-up code. This code is hardware and operating system specific, and its operation is outside the scope of this document. Of particular interest with respect to the driver code are:

8.8.1. sysapiSystemInit

This function allocates and initializes the network buffer pool, queues, and semaphores, and calls the *sysapiHardwareDiscovery* and *populateSlotRegistry* functions in *\bsp\broadcom\bcm\xxx\ipl \hwutils.c*, where xxx names the relevant operating system.

- *sysapiHardwareDiscovery*: This function reads system configuration data such as machine type and global MAC address from the CPU board’s serial EEPROM and stores it in the registry structure, and then sets the MTU size and flash memory type.
- *populateSlotRegistry*: This function allocates memory for, and then initializes, both the physical and logical slot and port registry structures. The physical slot and port registry initialization code need to be updated when the driver is ported to a new hardware implementation.

8.8.2. cnfgrInit

This function calls the initialization routines for the individual components based on the settings of the component IDs. The component IDs are defined in *L7_COMPONENT_IDS_t* in *\src\l7public \common\commdefs.h* and are set to either *L7_ENABLE* or *L7_DISABLE* in *FD_cnfgrList in \src \l7public\common\default_cnfgr.h*.

FD_cnfgrList controls the sequence of initialization for those components that require it. A package consists of one or more components.

8.8.3. dtlNetInit

This function loads *dtlEndLoad*, which allocates and initializes global memory.

When the application code gets control, it initializes the driver by issuing one call to a general initialization function, *dapiInit*, followed by a call to *dapiCardInit* for each physical and logical card in the system. *dapiInit* calls *cardDbInit* to have the *CARD_INFO_t* structures initialized. *cardDbInit* calls an initialization routine for each card type. These routines must be updated to set the structure members *hapiInit* and *hapiCardInit* to the correct pointers to the hardware-specific initialization functions.

8.9. Command Families

After initialization, all calls from the application code to the driver are made to the function *dapiCtl*. The parameters include a command code, identifying the action required. Commands are grouped into function families: each family has an associated data structure used for passing additional parameters, which consist of a union of structures. Each structure defines the parameter data for a specific command. If a command does not complete synchronously, the application is notified by a callback when it completes. Callbacks are also used for unsolicited notifications (received frame, port state change, and so on) and for the driver to solicit information from the application. One callback can be registered for each family.

Valid command codes are defined in the enumerated variable *DAPI_CMD_t*: valid callback codes are defined in the enumerated variable *DAPI_EVENT_t*: both variables are in *\src\l7public\api \dapi.h*.

The families and their associated command and event codes are listed in the DAPI Commands section. The families present for a given system depend on which packages are included: All systems include the System, Interface Management, and Frame families, and the Switching package, which includes Address Management, QVLAN Management, MRP Management, and Link Aggregation Management.

Chapter 9. Supported RFCs and Standards

This section lists supported RFCs and other standards.

9.1. Functional Summary and Supported Standards and RFCs

9.1.1. Management

9.1.1.1. Core Features

- RFC 854 — Telnet
- RFC 855 — Telnet option specifications
- RFC 1155 — SMI v1
- RFC 1157 — SNMP
- RFC 1212 — Concise MIB definitions
- RFC 1901 — Community-based SNMP v2
- RFC 1908 — Coexistence between SNMP v1 and SNMP v2
- RFC 2271 — SNMP framework MIB
- RFC 2295 — Transparent content negotiation
- RFC 2576 — Coexistence between SNMP v1, v2, and v3
- RFC 2578 — SMI v2
- RFC 2579 — Textual conventions for SMI v2
- RFC 2580 — Conformance statements for SMI v2
- RFC 3410 — Introduction and Applicability Statements for Internet Standard Management Framework
- RFC 3411 — An Architecture for Describing SNMP Management Frameworks
- RFC 3412 — Message Processing & Dispatching
- RFC 3413 — SNMP Applications
- RFC 3414 — User-Based Security Model
- RFC 3415 — View-based Access Control Model
- RFC 3416 — Version 2 of SNMP Protocol Operations
- RFC 3417 — Transport Mappings
- RFC 3418 — Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)

- Configurable management VLAN
- SSL 3.0 and TLS 1.0
  - RFC 2246 — The TLS protocol, version 1.0
  - RFC 3268 — AES cipher suites for Transport layer security
- SSH 1.5 and 2.0
  - RFC 4251 — SSH protocol architecture
  - RFC 4252 — SSH authentication protocol
  - RFC 4253 — SSH transport layer protocol
  - RFC 4254 — SSH connection protocol
  - RFC 4716 — SECSH public key file format
  - RFC 4419 — Diffie-Hellman group exchange for the SSH transport layer protocol
- RESTCONF — https://tools.ietf.org/html/draft-ietf-netconf-restconf-04
  - RFC 6020 — A Data Modeling Language for NETCONF
  - RFC 6415 — Web Host Metadata
  - RFC 6536 — NETCONF Access Control Model
  - RFC 7223 — YANG Data Model for Interface Management
  - RFC 7277 — YANG Data Model for IP Management
  - RFC 7317 — YANG Data Model for System Management
  - draft-ietf-netmod-syslog-model-03
  - draft-ietf-netconf-yang-library-00
  - draft-ietf-httpauth-basicauth-update-03

9.1.1.2. Advanced Management Features

- Industry-standard CLI with the following features:
  - Scripting capability
  - Command completion
  - Context-sensitive help
- Optional user password encryption
- Multisession Telnet server

- Auto Image Upgrade

9.1.2. Switching

9.1.2.1. Core Features

- IEEE 802.1AB — Link level discovery protocol
- IEEE 802.1D — Spanning tree
- IEEE 802.1p — Ethernet priority with user provisioning and mapping
- IEEE 802.1Q — Virtual LANs w/ port-based VLANs
- IEEE 802.1s — Multiple spanning tree compatibility
- IEEE 802.1w — Rapid spanning tree
- IEEE 802.1X — Port-based authentication
- IEEE 802.3 — 10Base-T
- IEEE 802.3u — 100Base-T
- IEEE 802.3ab — 1000Base-T
- IEEE 802.3ac — VLAN tagging
- IEEE 802.3ad — Link aggregation
- IEEE 802.3ae — 10 GbE
- IEEE 802.3x — Flow control
- ANSI/TIA-1057 — LLDP-MED
- RFC 4541 — IGMP snooping
- RFC 5171 — Unidirectional Link Detection (UDLD) Protocol

9.1.2.2. Additional Layer 2 Functionality

- Broadcast storm recovery
- Double VLAN/VMAN tagging
- DHCP Snooping
- Dynamic ARP inspection
- Independent VLAN Learning (IVL) support
- Jumbo Ethernet frames
- Port mirroring

- Static MAC filtering
- IGMP snooping querier
- Port MAC locking
- IP source guard
- Protected ports

9.1.2.3. System Facilities

- Event and error logging facility
- Runtime and configuration download capability
- PING utility
- XMODEM
- RFC 768 — UDP
- RFC 783 — TFTP
- RFC 791 — IP
- RFC 792 — ICMP
- RFC 793 — TCP
- RFC 826 — Ethernet ARP
- RFC 951 — BOOTP
- RFC 1321 — Message digest algorithm
- RFC 1534 — Interoperability between BOOTP and DHCP
- RFC 2030 — Simple Network Time Protocol (SNTP) V4 for IPv4, IPv6, and OSI
- RFC 2131 — DHCP Client
- RFC 2132 — DHCP options and BOOTP vendor extensions
- RFC 2865 — RADIUS client
- RFC 2866 — RADIUS accounting
- RFC 2868 — RADIUS attributes for tunnel protocol support
- RFC 2869 — RADIUS extensions
- RFC 28869bis - RADIUS support for Extensible Authentication Protocol (EAP)
- RFC 5424 — The Syslog protocol
- RFC 3580 — 802.1X RADIUS usage guidelines

- RFC 5176 - Dynamic Authorization Extensions to RADIUS
- sFlow Version 5 — Industry standard for sFlow implementation (http://www.sflow.org/sflow_version_5.txt)
- sFlow LAG Counters Structure — Standard to export LACP counters in the sFlow counter sample for a port that is a member of a LAG (http://sflow.org/draft_sflow_lag.txt)

9.1.3. Routing

9.1.3.1. Core Features

- RFC 1256 — ICMP router discovery messages
- RFC 1321 — Message digest algorithm
- RFC 1519 — CIDR
- RFC 1765 — OSPF database overflow
- RFC 1812 — Requirements for IPv4 routers
- RFC 2131 — DHCP relay
- RFC 2328 — OSPFv2
- RFC 2370 — The OSPF Opaque LSA Option
- RFC 3021 — Using 31-Bit Prefixes on IPv4 Point-to-Point Links
- RFC 3046 — DHCP/BOOTP relay
- RFC 3101 — The OSPF “Not So Stubby Area” (NSSA) option
- RFC 3107 — Carrying label information in BGP-4
- RFC 3137 — OSPF Stub Router Advertisement
- RFC 3623 — Graceful OSPF Restart
- RFC 3768 — Virtual Router Redundancy Protocol (VRRP)
- RFC 5187 — OSPFv3 Graceful Restart
- RFC 5309 — Point-to-Point Operation over LAN in Link State Routing Protocols
- RFC 5340 — OSPF for IPv6
- RFC 5549 — Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop
- RFC 5880 — Bidirectional Forwarding Detection (BFD)
- RFC 5881 — Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)
- RFC 6860 — Hiding Transit-Only networks in OSPF

- Route redistribution across BGP and OSPF
- VLAN routing

9.1.4. IPv6 Routing

- RFC 1981 — Path MTU for IPv6
- RFC 2460 — IPv6 Protocol Specification
- RFC 2464 — IPv6 over Ethernet
- RFC 2711 — IPv6 Router Alert
- RFC 3056 — Connection of IPv6 Domains via IPv4 Clouds
- RFC 3315 — Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
- RFC 3513 — Addressing Architecture for IPv6
- RFC 3484 — Default Address Selection for IPv6
- RFC 3493 — Basic Socket Interface for IPv6
- RFC 3542 — Advanced Sockets API for IPv6
- RFC 3587 — IPv6 Global Unicast Address Format
- RFC 3633 — IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6
- RFC 3736 — Stateless DHCPv6
- RFC 4213 — Basic Transition Mechanisms for IPv6
- RFC 4291 — Addressing Architecture for IPv6
- RFC 4443 — ICMPv6
- RFC 4861 — Neighbor discovery for IPv6
- RFC 4862 — IPv6 stateless address autoconfiguration
- RFC 6164 — Using 127-Bit IPv6 Prefixes on Inter-Router Links
- RFC 6583 — Operational Neighbor Discovery Problems

9.1.5. BGP4

9.1.5.1. Core Features

- RFC 1997 — BGP Communities Attribute
- RFC 2385 — Protection of BGP Sessions via the TCP MD5 Signature Option
- RFC 2545 — BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing

- RFC 2918 — Route Refresh Capability for BGP-4
- RFC 3107 — Carrying Label Information in BGP-4
- RFC 4271 — A Border Gateway Protocol 4 (BGP-4)
- RFC 4360 — BGP Extended Communities Attribute
- RFC 4456 — BGP Route Reflectors
- RFC 4486 — Subcodes for BGP Cease Notification Message
- RFC 4760 — Multiprotocol Extensions for BGP-4
- RFC 5492 — Capabilities Advertisement with BGP-4

9.1.6. Multicast

9.1.6.1. Core Features

- RFC 1112 — Host extensions for IP multicasting
- RFC 2236 — IGMP v2
- RFC 2365 — Administratively scoped boundaries
- RFC 2710 — MLDv1
- RFC 3376 — IGMPv3
- RFC3810 — MLDv2
- RFC3973 — PIM-DM
- RFC4601 — PIM-SM
- draft-ietf-idmr-dvmrp-v3-10 - DVMRP
- draft-ietf-magma-igmp-proxy-06.txt — IGMP/MLD-based multicast forwarding (IGMP/MLD proxying)
- draft-ietf-magma-igmpv3-and-routing-05.txt — IGMPv3 and multicast routing protocol interaction
- draft-ietf-pim-sm-bsr-05 — Bootstrap Router (BSR) Mechanism for PIM
- Static RP configuration

9.1.7. Quality of Service

9.1.7.1. DiffServ

- RFC 2474 — Definition of the differentiated services field (DS Field) in the IPv4 and IPv6 headers

- RFC 2475 — An architecture for differentiated services
- RFC 2597 — Assured forwarding PHB group
- RFC 2697 — Single-rate policing
- RFC 3246 — An expedited forwarding PHB (Per-Hop Behavior)
- RFC 3260 — New terminology and clarifications for DiffServ

9.1.7.2. Access Control Lists (ACL)

- Permit/deny actions for inbound or outbound IP traffic classification based on:
  - Type of service (ToS) or differentiated services (DS) DSCP field
  - Source IP address
  - Destination IP address
  - TCP/UDP source port
  - TCP/UDP destination port
  - IP protocol number
  - IPv6 flow Label
- Permit/deny actions for inbound or outbound Layer 2 traffic classification based on:
  - Source MAC address
  - Destination MAC address
  - EtherType
  - VLAN identifier value or range (outer and/or inner VLAN tag)
  - 802.1p user priority (outer and/or inner VLAN tag)
- Optional rule attributes:
  - Assign matching traffic flow to a specific queue
  - Redirect or mirror (flow-based mirroring) matching traffic flow to a specific port
  - Generate trap log entries containing rule hit counts
- RFC 1858—Security Considerations for IP Fragment Filtering

9.1.7.3. Class of Service (CoS)

- Direct user configuration of the following:
  - IP DSCP to traffic class mapping

- IP precedence to traffic class mapping
- Interface trust mode: 802.1p, IP Precedence, IP DSCP, or untrusted
- Interface traffic shaping rate
- Minimum and maximum bandwidth per queue
- Strict priority versus weighted (WRR/WDRR/WFQ) scheduling per queue
- Tail drop versus Weighted Random Early Detection (WRED) queue depth management

9.1.8. ICOS OpenFlow

- OpenFlow Switch Specification, Version 1.0.0 (Wire Protocol 0x01) and version 1.3.0

9.1.9. ICOS Data Center

- IEEE 802.1Qau Draft 2.4 — QCN (Quantized Congestion Notification)
- IEEE 802.1Qaz Draft 2.4 — ETS (Enhanced Transmission Selection)
- ANSI/INCITS Fibre Channel Backbone-5 (FC-BB-5) REV 2.0.0—FIP Snooping Bridge
- draft-sridharan-virtualization-nvgre-02.txt — Network Virtualization using Generic Routing Encapsulation (NVGRE)
- draft-mahalingam-dutt-dcops-vxlan-04.txt — A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks (VXLAN)
- RFC 7047 — Open vSwitch Database Management Protocol
- RFC 3032 — MPLS Label Stack Encoding

Chapter 10. ICOS Externally Licensed Software

10.1. Overview

ICOS contains a number of third-party externally licensed software components. This appendix contains information regarding these components, the license for each of these components, and where these components are used in ICOS.

10.2. Copyright Adherence

In all cases, ICOS adheres to the terms and conditions as cited in the Table 10.1, “Externally Licensed Components”.

10.3. Externally Licensed Components

ICOS contains a number of 3rd party externally licensed software components. This section contains information regarding these components, the license for each of these components, and where these components are used in ICOS.

Table 10.1. Externally Licensed Components

Name of Component	Source of Component	Source Version	Used in ICOS Module	Has Netberg Modified Original Code?
DHCP Client	Red Hat eCos http://ecos.sourceware.org/oldlicense.html	eCos 1.1	Switching	Yes
Routing	The FreeBSD Project www.freebsd.org	1.130.2.21	Routing, IP Multicast	Yes
Base code download and decompression	Jean-loup Gailly and Mark Adler http://sourceforge.net/projects/libpng/files/zlib/1.1.4/	1.1.4	Switching	Yes
SSH	The OpenSSH Project www.openssh.org	6.8p1	Management	Yes
SSL	The OpenSSL Project www.openssl.org	1.0.2	Management	No
IP Multicast (IGMP, DVMRP)	The Regents of the University of Michigan and Merit Network, Inc. http://www.regents.umich.edu/	2.2.0	IP Multicast	Yes
MD5	RSA Security, Inc http://www.ietf.org/rfc/rfc1321.txt	RFC 1321	Switching	Yes
HMAC	RSA Security, Inc http://www.ietf.org/rfc/rfc2104.txt	RFC 2104	Switching	Yes
EMANATE/Lite	SNMP Research International, Inc. http://www.snmp.com/products/emlite.shtml	16.2.0.4	Management	Yes
Linux kernel	The Linux Kernel Organization http://git.kernel.org/?p=linux/kernel/git/stable/linux-stable.git	3.16.0.29	Linux BSP	No
Busybox	www.busybox.nett/downloads/	1.18.5	Linux BSP	Yes
lrzsz	Uwe Ohse http://ohse.de/uwe/software/lrzsz.html	0.12.20	Linux BSP	No

Name of Component	**Source of Component**	**Source Version**	**Used in ICOS Module**	**Has Netberg Modified Original Code?**
utelnetd	Pengutronix http://www.pengutronix.com/software/utelnetd/download/	0.1.4	Linux BSP	No
Mtd	Arcom Control System, Ltd. http://git.infradead.org/mtd-utils.git	1.0	Linux BSP	No
7-zip	Igor Pavlov http://www.7-zip.org/download.html	4.48	Bootloader (CFE)	No
YUI Compressor	Yahoo! Inc. http://yuilibrary.com/downloads/#yuicompressor	2.4.2	Tools/Build	No
strace	http://sourceforge.net/projects/strace/files/strace/4.6	4.6	Linux BSP	No
DHCPv6	KAME Project (http://sourceforge.net/projects/wide-dhcpv6/files/)	20080615 tar ball	IPv6	Yes
ONIE	GITHUB https://github.com/opencomputeproject/onie	2015.11	ONIE	Yes
Ubuntu Utopic kernel	http://kernel.ubuntu.com/git/ubuntu/ubuntu-trusty.git/tag/?id=Ubuntults-3.16.0-29.39_14.04.1	Ubuntu-lts-3.16.0-29.39_14.04.1	Linux BSP	No
OVS 2.3.0	http://openvswitch.org/releases/openvswitch-2.3.0.tar.gz	2.3.0	Base	No
Cjson	http://sourceforge.net/projects/cjson/	N/A	Base	Yes
Libnetconf	https://github.com/cesnet/libnetconf	0.9	Management	Yes
Libxml2	https://git.gnome.org/browse/libxml2/	2.9.2	Management	No
Libxslt	https://git.gnome.org/browse/libxslt/	1.1.27	Management	No