COMPUTING

FlowPilot™

Configuration Guide

P/N: 6806800R03H

December 2016

ARTESYN™

EMBEDDED TECHNOLOGIES

© Copyright 2016 Artesyn Embedded Technologies, Inc.

All rights reserved.

Trademarks

Artesyn Embedded Technologies, Artesyn and the Artesyn Embedded Technologies logo are trademarks and service marks of Artesyn Embedded Technologies, Inc. All other names and logos referred to are trade names, trademarks, or registered trademarks of their respective owners. © 2016 Artesyn Embedded Technologies, Inc. All rights reserved. For full legal terms and conditions, please visit www.artesyn.com/legal.

Notice

While reasonable efforts have been made to assure the accuracy of this document, Artesyn assumes no liability resulting from any omissions in this document, or from the use of the information obtained therein. Artesyn reserves the right to revise this document and to make changes from time to time in the content hereof without obligation of Artesyn to notify any person of such revision or changes.

Electronic versions of this material may be read online, downloaded for personal use, or referenced in another document as a URL to an Artesyn website. The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of Artesyn.

It is possible that this publication may contain reference to or information about Artesyn products (machines and programs), programming, or services that are not available in your country. Such references or information must not be construed to mean that Artesyn intends to announce such Artesyn products, programming, or services in your country.

Limited and Restricted Rights Legend

If the documentation contained herein is supplied, directly or indirectly, to the U.S. Government, the following notice shall apply unless otherwise agreed to in writing by Artesyn.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data clause at DFARS 252.227-7013 (Nov. 1995) and of the Rights in Noncommercial Computer Software and Documentation clause at DFARS 252.227-7014 (Jun. 1995).

Contact Address

Artesyn Embedded Technologies
Marketing Communications
2900 S. Diablo Way, Suite 190
Tempe, Arizona 85282

Artesyn Embedded Technologies
Lilienthalstr. 17-19
85579 Neubiberg/Munich
Germany

Contents

List of Tables

List of Tables

List of Figures

Overview of Contents

This document explains about FlowPilot™. The user should have the prior knowledge about ATCA architecture, ATCA terminology like hub blade, payload blade, backplane, and ATCA chassis, and the SRstackware basic usage.

This manual is divided into the following chapters and appendices:

- Chapter 1, *Introduction, on page 15* gives a brief overview of the FlowPilot and its Licensing information.
- Chapter 2, *FlowPilot Stateless Load Balancer, on page 17* provides information about FlowPilot Stateless Load Balancer features, functional description, and configuration.
- Chapter 3, *FlowPilot ECMP Load Balancer, on page 49* provides information about FlowPilot ECMP Load Balancer overview, configuration and command description.
- Appendix A, *FlowPilot Stateless Load Balancer Default Configuration, on page 59* contains configuration information and also steps on how to customize the configuration.
- Appendix B, *FlowPilot ECMP Load Balancer Sample Configuration, on page 65* includes procedure on how to configure, modify, and remove ECMP Load Balancer configuration.
- Appendix C, *Related Documentation, on page 73* provides a listing of related product documentation.

Abbreviations

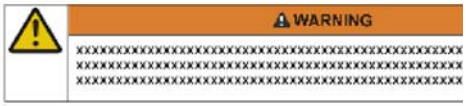
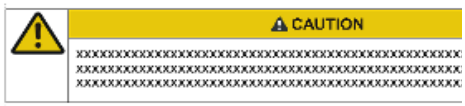
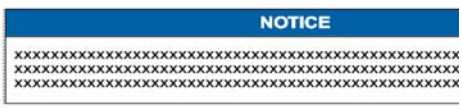
This document uses the following abbreviations:

Abbreviation	Definition
AdvancedTCA, ATCA	Advanced Telecommunications Computing Architecture
ARP	Address Resolution Protocol
ECMP	Equal Cost Multi Path
GARP	Gratuitous Address Resolution Protocol
MAC	Media Access Control

Conventions

The following table describes the conventions used throughout this manual.

Notation	Description
0x00000000	Typical notation for hexadecimal numbers (digits are 0 through F), for example used for addresses and offsets
0b0000	Same for binary numbers (digits are 0 and 1)
bold	Used to emphasize a word
`Screen`	Used for on-screen output and code related elements or commands in body text
`Courier + Bold`	Used to characterize user input and to separate it from system output
Reference	Used for references and for table and figure descriptions
File > Exit	Notation for selecting a submenu
<text>	Notation for variables and keys
[text]	Notation for software buttons to click on the screen and parameter description
...	Repeated item for example node 1, node 2, ..., node 12
. . .	Omission of information from example/command that is not necessary at the time being
..	Ranges, for example: 0..4 means one of the integers 0,1,2,3, and 4 (used in registers)
|	Logical OR

Notation	Description
	Indicates a hazardous situation which, if not avoided, could result in death or serious injury
	Indicates a hazardous situation which, if not avoided, may result in minor or moderate injury
	Indicates a property damage message
	No danger encountered. Pay attention to important information

Summary of Changes

The summary of changes is provided in the following table:

Part Number	Publication Date	Description
6806800R03H	December 2016	Updated *Target prediction* on page 43.
6806800R03G	August 2015	Added a new section *ARP Generator Sample Utility* on page 70.
6806800R03F	May 2015	Added a new chapter *FlowPilot ECMP Load Balancer* on page 49 and *FlowPilot ECMP Load Balancer Sample Configuration* on page 65. Updated the chapter *Introduction* on page 15 and *FlowPilot Stateless Load Balancer Default Configuration* on page 59.
6806800R03E	June 2014	Re-branded to Artesyn template.

Part Number	Publication Date	Description
6806800R03D	March 2014	Added Port Failover on page 18, Target prediction on page 18, Port weight on page 18, Hash-Algorithm Selection on page 19, Port Failover Configuration on page 43, Target prediction on page 44, Port Weight Configuration on page 46, Hash-Algorithm Selection on page 47, and Pass-through exceptions on page 49. Updated Health check on page 19, Static Channel Configuration on page 31, Health Check Functionality on page 32.
6806800R03C	April 2013	Updated Chapter 3, Overview, on page 27, Default Configuration on page 57, and How to Customize the Configuration on page 59.
6806800R03B	December 2012	Updated Chapter 3, Overview, on page 27, Command Syntax on page 32, Health Check Functionality on page 32, MAC learning on page 49, Licensing on page 49, Default Configuration on page 57, and How to Customize the Configuration on page 59.
6806800R03A	October 2012	Initial Release

FlowPilot™ uses packet field hashing mechanism to provide load balancing solutions. FlowPilot is used to balance the load of ingress traffic on hub blade (ATCA-F140) towards payload blades. It provides two independent load balancing solutions and they are as follows:

- FlowPilot Stateless Load Balancer - This solution is used to load balance the traffic to trunk member ports. In the previous releases, FlowPilot Stateless Load Balancer was referred as FlowPilot. For detailed information, refer to Chapter 2, *FlowPilot Stateless Load Balancer, on page 17*.
- FlowPilot Equal Cost Multi Path (ECMP) Load Balancer - This solution is used to load balance the traffic to Next Hops. For detailed information, refer to Chapter 3, *FlowPilot ECMP Load Balancer, on page 49*.

1.1 Licensing

FlowPilot needs a license to enable the functionality. Refer to the *ATCA-F140 Basic Blade Services Programming Reference Guide* for the procedure to procure and use FlowPilot license.

FlowPilot Stateless Load Balancer

FlowPilot Stateless Load Balancer is a load balancing software which load balances the incoming traffic towards the payload blades. It has the following features:

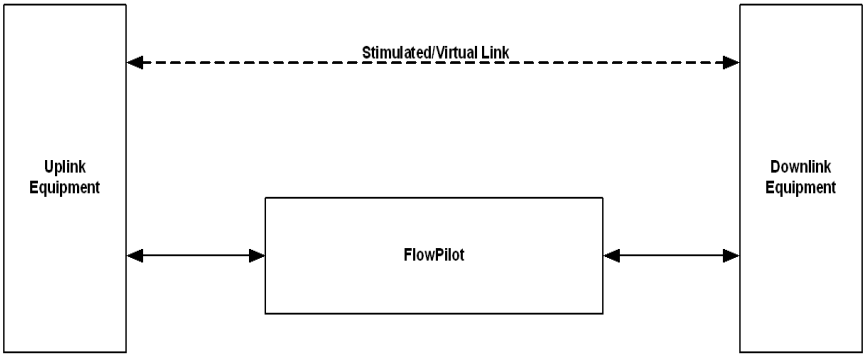
- Filtering the traffic based on packet fields
- Switching the traffic based on blade failures
- Port Failover
- Dynamic Load Balancing
- Target Prediction
- Health check of the blades
- Link transparency
- Pass-through
- Manipulation of forwarding table

In this chapter, the term FlowPilot refers to FlowPilot Stateless Load Balancer. For detailed information about FlowPilot Stateless Load Balancer features, refer the section *Functional Description* on page 19.

FlowPilot Stateless Load Balancer is supported on Artesyn ATCA-F140 blade and runs on the Ethernet fabric chipset. FlowPilot Stateless Load Balancer is a licensed software and it is part of SRstackware module present on the ATCA switch blades.

The typical hardware configuration includes a pair of redundant ATCA-F140 blades on an ATCA chassis, which has payload blades in other slots. The ATCA-F140 can take 80G traffic from each direction (uplink to downlink and vice-versa) and load balance the traffic towards the payload blades. The payload blades typically run customer applications like bump-in-the-wire. The bump-in-the-wire application processes the traffic and forwards to external equipment connected to uplink side and downlink side without the external equipment knowing about the bump-in-wire network element. FlowPilot Stateless Load Balancer offers the features that are needed to accomplish this task.

Figure 2-1 FlowPilot Stateless Load Balancer



The traffic flow in both the directions of FlowPilot Stateless Load Balancer is as follows:

- Uplink -> Downlink direction
  - The uplink equipment sends the traffic to the uplink side port of FlowPilot Stateless Load Balancer.
  - FlowPilot Stateless Load Balancer receives and load balances the traffic based on the Source IP address from uplink side port to the payload blades.
  - The applications (for example, bump-in-the-wire) on the payload blades process the traffic and sends the traffic back to the hub blade
  - FlowPilot Stateless Load Balancer sends this traffic to the downlink equipment through downlink port.
- Downlink -> Uplink direction
  - The downlink equipment sends the traffic to the downlink side port of FlowPilot Stateless Load Balancer.
  - FlowPilot Stateless Load Balancer receives and load balances the traffic based on the destination IP address from downlink side port to the payload blades.

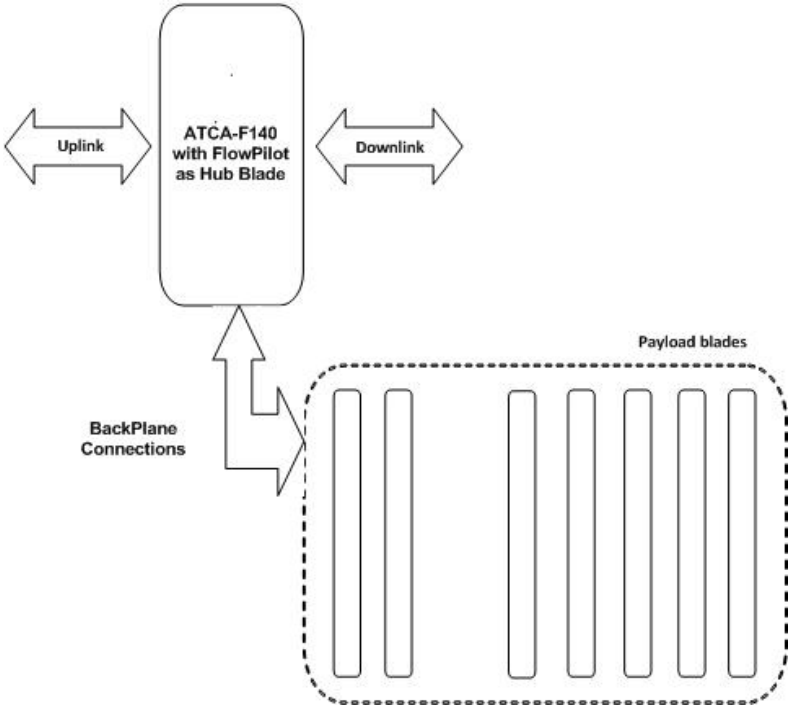
– The applications (for example, bump-in-the-wire) on the payload blades process the traffic and sends the traffic back to the hub blade.

– FlowPilot Stateless Load Balancer sends this traffic to the uplink equipment through uplink port.

2.1 Functional Description

This section provides an overview of FlowPilot Stateless Load Balancer features. The figure below illustrates the traffic flow in both the directions of FlowPilot Stateless Load Balancer.

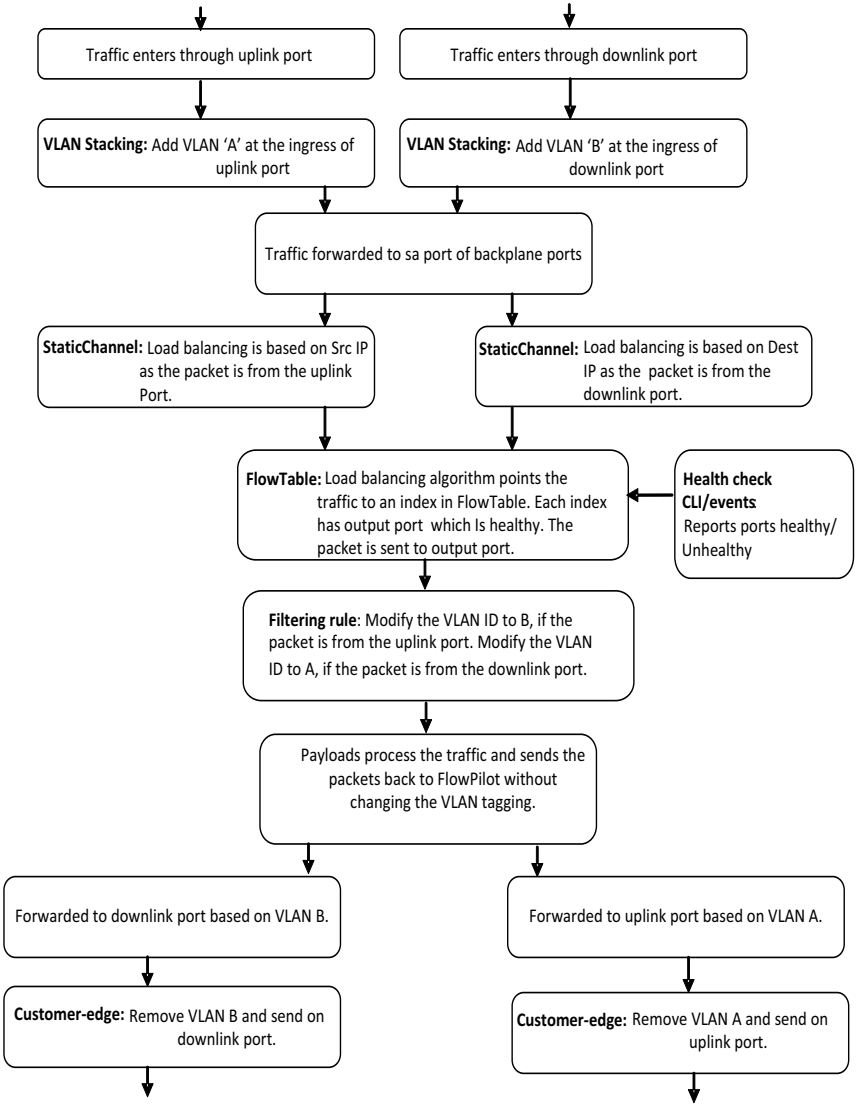
Figure 2-2 FlowPilot Stateless Load Balancer Traffic Flow



2.1.1 FlowPilot Stateless Load Balancer Packet Processing Logic

The following figure depicts the packet processing logic in FlowPilot Stateless Load Balancer.

Figure 2-3 FlowPilot Stateless Load Balancer Packet Processing Logic



FlowPilot Stateless Load Balancer forwards the packets transparently between the external equipment. Any modification to the packet is typically done by the application (for example, bump-in-the-wire) running on the payload blades. However, FlowPilot Stateless Load Balancer tags the packet with different VLANs for the applications to distinguish between the flow directions. The packet VLAN tag is added at FlowPilot Stateless Load Balancer ingress port and removed at the FlowPilot Stateless Load Balancer egress port, providing transparency.

2.1.1.1 FlowTable Modifications

In default FlowPilot configuration, load balance uses the source IP address for uplink port and the destination IP address for downlink port, a typical flow or a session would always be forwarded to the same payload. FlowPilot Stateless Load Balancer uses the hash algorithm to maintain the mapping of session/flow to payload. The algorithm computes a bucket location based on the value in the packet IP address. FlowPilot Stateless Load Balancer has 1024 bucket locations. The outgoing ports are configured in these bucket locations known as FlowTable. The user can modify the outgoing ports in the FlowTable to change the course of the traffic or can configure the FlowPilot Stateless Load Balancer to modify the FlowTable automatically.

2.1.1.2 Application Switchover

The applications typically run on the payloads of the system. The applications maintain the state information of the traffic sessions they receive from the hub blade. To maintain redundancy, the application may sync its session's state from payload-A to payload-B in the system. In order to support this redundancy, FlowPilot Stateless Load Balancer can be configured to switchover the traffic to payload-B in case of a failover of payload-A. If the application can sync to more than one payload blades (for example, to payload-B and payload-C and payload-D) then the FlowPilot Stateless Load Balancer can be configured to switchover (load balance) the failed payload-A traffic equally to more than one payload blade.

2.1.1.3 Port Failover

FlowPilot Stateless Load Balancer supports (n+m) failover policy at application group level. The payloads that are part of the same application group can be assigned with active or backup role. For every 'n' active payloads, user can choose 'm' backup payloads. Typically, the backup payloads won't be receiving the application traffic as long as the active payloads are operational. When an active payload fails, then its first backup payload will be made active and the traffic of failed payload will be directed to the newly become active payload.

NOTICE

After multiple failovers, in intermediate state, when some ports become active (but not all), SA forwarding table will have uneven distribution, which means during this there will be uneven load-balancing.

2.1.1.4 Target prediction

FlowPilot Stateless Load Balancer provides an interface through which a user can predict the target port/payload to which a particular traffic flow will be load balanced. The interface expects a bit pattern of the header fields of the flow to return the bucket index and port information of the aggregator in the FlowTable to which this flow is destined.

2.1.1.5 Port weight

The capabilities of processing entities behind a port may vary significantly. The processing entity could be a single payload blade or a chassis with multiple payloads. Thus, a port with multiple payloads, could process more traffic than a port with single payload. By distributing the traffic uniformly to all these ports, one may not be fully utilizing their processing capabilities. Using port-weight configuration, a processing entity with better capabilities can be assigned with more weight than others. Thus, a port with more weight will receive more traffic than the others.

This configuration can also be used at runtime to achieve dynamic load-balancing. One can run a thread to monitor the load of the processing entities behind a port through health-check logic and identify the load experienced by those entities.

Upon identifying Over-load or Under-load situation, one can change the port weights to adjust the load. As soon as, the weight of an over-loaded port is reduced, some of its traffic will be load-balanced to other ports.

2.1.1.6 Hash-Algorithm Selection

FlowPilot Stateless Load Balancer provides an interface through which a user can select a hash algorithm of provided two algorithms globally in RTAG7 (enhanced) engine. In enhanced load-balancing, by default, FlowPilot Stateless Load Balancer uses the hash algorithm 1 of hash A (normal) and hash B (extended). Both hash A and hash B support symmetric hashing. User can select hash algorithm 2 for better load-balancing, which can be verified using target-predict command explained in *Target prediction* on page 43. In hash algorithm, 2 symmetric hashing will only work with hash A (enhanced).

2.1.1.7 Health check

A blade failover may not always mean a link failure. A link failure is very easy to detect and FlowPilot Stateless Load Balancer takes the switchover actions based on link failover by default. FlowPilot Stateless Load Balancer can also do health check of the payload blades. The health check is typically a special Layer 2 packet sent to the payload and the payload bounces the same packet back to FlowPilot Stateless Load Balancer within certain time.

Apart from simply bouncing the received health-check packet, a payload blade may wish to share the state of the processing entities present on it. This processor state/load information can be provided in the data section of the health-check packet and thus passing it to the customer applications.

The health check failure will be considered as blade failover and if configured, FlowPilot Stateless Load Balancer triggers FlowTable modifications.

2.1.1.8 Link Transparency

The uplink equipment and the downlink equipment assume that they are connected to each other. FlowPilot Stateless Load Balancer simulates this behavior by monitoring the uplink and downlink ports connected to each other. If the uplink port is link down, then the downlink port is shutdown and vice versa.

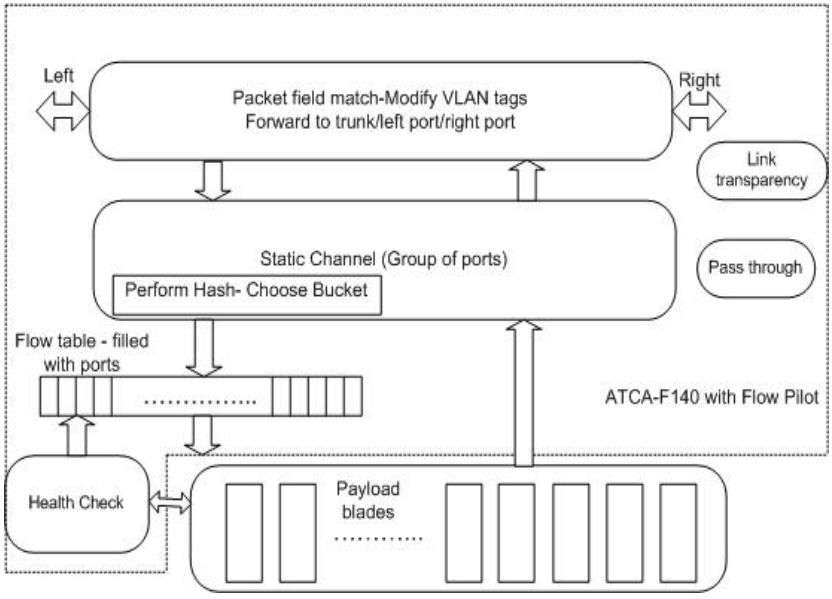
2.1.1.9 Content Aware Filtering

The traffic can be filtered based on the packet fields and actions performed on them. An example is to configure an action to assign a different VLAN ID to the packet, if the IP address is in between a certain range. This VLAN ID can be used in the payload blades to take intelligent actions.

2.1.2 Block Diagram

The following figure illustrates the FlowPilot Stateless Load Balancer block diagram.

Figure 2-4 FlowPilot Stateless Load Balancer Block Diagram



The above diagram shows the ATCA-F140 blade with FlowPilot Stateless Load Balancer functionality. The uplink and downlink ports receive the traffic and direct the traffic to static channel. The static channel performs the hash based load balance and puts the packet to bucket index of FlowTable. The FlowTable is filled with the port information and the port information is filled with healthy ports information from Health check module. The packets are sent to these ports. The payload blades process the packets and send them back to FlowPilot Stateless Load Balancer.

2.2 Detailed Information

SRstackware is delivered with FlowPilot Stateless Load Balancer default configuration file. Replace the SRstackware default configuration file with FlowPilot Stateless Load Balancer default configuration file and then reboot the blade. The complete configuration details are provided in *FlowPilot Stateless Load Balancer Default Configuration* on page 59.

You can customize the FlowPilot Stateless Load Balancer configuration easily using configuration wizard script.

`/opt/srstackware/scripts/generate_fp_scripts.sh`

- It is an interactive script that poses a series of questions to the user for customizing FlowPilot Stateless Load Balancer configuration. For more information, see *How to Customize the Configuration* on page 61.
- This script simply relies on the user inputs and generates customized FlowPilot Stateless Load Balancer configuration scripts.
- Based on valid user inputs, this script generates the below customized FlowPilot Stateless Load Balancer configuration scripts, at `/etc/opt/srstackware/config/flowpilot/` directory:
  - `enable_flowpilot.custom`
  - `disable_flowpilot.custom`

  You can invoke these scripts in the imish CLI, using `imish -f <config script>`.

It is user's responsibility to provide valid inputs to the wizard script to get valid customization scripts.

Every time this script is executed, it generates a fresh set of configuration scripts replacing the previously generated files.

To use these scripts, follow the steps mentioned below:

1. Run the command:
   `trid-cfg –a -y -i /opt/bladeservices/cfg/trid_1440_def_fp` for AXP-1440 chassis
   OR

`trid-cfg –a -y -i /opt/bladeservices/cfg/trid_640_def_fp` for AXP-640 chassis.

2. Boot the ATCA-F140 with default FlowPilot Stateless Load Balancer configuration and save the running configuration using imish CLI command, `#write file`.
3. Run the corresponding disable config script without any modifications:
   `imish -f disable_<chassi-ID>_flowpilot<license-ID>_config`
   For example, if the blade is booted with `srs.1440.2.fp40.cfg`, the corresponding disable config script is `disable_1440_flowpilot40_config`.
4. Run the Configuration wizard script:
   `/opt/srstackware/scripts/generate_fp_scripts.sh`

   The `enable_flowpilot.custom` and `disable_flowpilot.custom` files will be generated.
5. Run the customized enable config script
   `imish -f enable_flowpilot.custom`
   Save this custom configuration as the default configuration.
6. To disable the custom configuration, the corresponding disable script, that is, `disable_flowpilot.custom` should be loaded using imish.

FlowPilot Stateless Load Balancer needs configuration of several features of SRstackware. FlowPilot Stateless Load Balancer also offers a simple FlowPilot config script which can be used to configure all these features in a proper sequence. The sequence is explained in Appendix A, *Default Configuration, on page 59*.

2.3 Features Used for Configuration

This section helps the user to understand the features specific to FlowPilot Stateless Load Balancer and helps in customizing the configuration.

The following are the features required for configuring FlowPilot Stateless Load Balancer:

1. VLAN stacking customer-edge configuration - to add VLAN tags
2. Hybrid port configuration - General VLAN configuration
3. Matchlists - To configure the filtering rules
4. Static Channel - To aggregate the ports and apply load balancing algorithm

5. Link transparency - To configure link transparency between uplink and downlink ports.
6. Pass-through - To let all the traffic pass through without copying to management processor.
7. Health check - To check the payload health and trigger FlowTable manipulations.

2.4 Application Integration

The SRS API library provides application to configure programmatically and also to receive events. The application can run remotely or on the same switch blade to configure FlowPilot Stateless Load Balancer commands. The SRS API library takes the CLI commands as the input for configuration. The SRS API returns asynchronous events like link up/down and health check events. The application can trigger FlowTable modifications based on these events. For more information, refer to the *SRstackware API Developer Guide*.

2.5 Uplink Port and Downlink Port Configuration

Choose the uplink port and the downlink port. These are typically the external ports either on the front panel of ATCA-F140 or on the ATCA-F140 RTM.

2.5.1 VLAN Stacking

These ports are configured to perform VLAN stacking at customer edge. This means an outer VLAN tag will be added to all the incoming packets and the outer VLAN tag is deleted for all the outgoing packets. As a result, the packet content is transparent to FlowPilot Stateless Load Balancer.

2.5.2 Link Transparency

The uplink and the downlink ports can be configured to maintain link transparency. When the uplink port's operational link is down, then the downlink port is administratively shutdown automatically. FlowPilot Stateless Load Balancer monitors the uplink port's operational link to be up, to configure the downlink port administratively up. It does the similar operation to uplink port when the downlink port is operationally down. The CLI command below can be used to configure link transparency.

2.5.2.1 Command Syntax

```
link-associate <left-port> <right-port>

no link-associate <left-port> <right-port>
```

2.5.2.2 Command mode

Configure mode

2.5.3 Selection of Load Balancing Per Port

The load balancing requires configuration of normal and extended field-select on the external ports. The following configuration should be done on the external ports.

2.5.3.1 Command Syntax

On uplink port, use the command `load-balance field-select normal`

On downlink port, use the command `load-balance field-select extended`

2.5.3.2 Command mode

Interface mode

2.6 Filtering and Assigning VLAN Tags

In SRstackware, the filtering rules are configured through matchlists. A matchlist configuration is a two-step process. Firstly, the match fields are chosen and the matching values are configured to these fields. These are called as matchlists. These matchlists are assigned to actions using rule command.

The packets coming from uplink and downlink ports are assigned to uplink VLAN ID and downlink VLAN ID, respectively. The payloads when sending the packets back to hub blade, the payload blades retain these VLAN IDs of the packets.

FlowPilot Stateless Load Balancer requires at least two mandatory matchlists be configured. The first matchlist is to match the left in port and modify the VLAN to the right port. The second matchlist is to match the right in port and modify the VLAN to left port.

However, you can configure more matchlists to modify the VLAN IDs based on the packet fields before sending to payloads. The payloads can make use of these VLAN tags to take intelligent actions, for example, further load-balancing of the packets and send these VLAN tags back to hub blade.

For example, using `action modify-vlan`, the uplink VLAN range <1000-1500> can be associated to uplink port and downlink VLAN range <2000-2500> can be associated to downlink port, with a match criteria of IP address ranges. In the return path, a matchlist can be configured for uplink VLAN range <1000-1500> to redirect to downlink port and downlink VLAN range <2000-2500> to redirect to uplink port. Use the mask parameter to reduce the number of matchlists.

2.7 Static Channel Configuration

SRstackware supports static channel configuration. The static channel configuration bundles group of ports into an aggregation port, but does not run the LACP protocol. FlowPilot Stateless Load Balancer requires aggregation of all the ports connected to payload blades into one static channel group. The uplink and downlink ports can also be static channel towards the external equipment.

2.7.1 Load Balancing

The load balancing algorithm should be configured on a static channel.

FlowPilot Stateless Load Balancer supports load balancing using the below header fields:

- IPv4/IPv6 source IP and destination IP addresses
- Source and destination MAC addresses
- MPLS (Outer Header)

FlowPilot Stateless Load Balancer uses an efficient hash-based load balanced algorithm and it should be configured on the static channel configured towards the backplane/payload ports.

The load balancing algorithm should choose source IP for the packets arrived from the uplink port and destination IP for the packets arrived from the downlink port. The load balancing algorithm acquires this information from the normal and extended load balance commands configured on the uplink and downlink ports.

Use the commands below to configure the load balancing algorithm on the static channel.

2.7.1.1 Command Syntax

```
port-channel load-balance <src-ip/dst-ip/src-dst-ip/src-mac/dst-
mac/src-dstmac> [enhanced .....] [extended-hash] [mpls] [use-
inner/use-outer]
```

2.7.1.2 Command mode

Interface mode (on sa# only)

The source/destination IP address can be inside tunneled traffic/single VLAN tagged packets/ up to 3 MPLS labels. The GRE and IP tunnels are supported. You can configure load balancing based on either the inner or the outer header for tunnels.

2.8 Health Check Functionality

FlowPilot Stateless Load Balancer can check the status of payload health. The health check configuration is optional and it allows configuration on a specific port and can be disabled on other ports. In case health check is disabled, FlowPilot Stateless Load Balancer only uses link states for FlowTable modifications.

FlowPilot Stateless Load Balancer sends special packets to the payloads on the backplane port. These special packets also known as Health-check request packets, have FlowPilot-specific source MAC address and an internally computed destination MAC address. The payloads bounce these special packets back to FlowPilot Stateless Load Balancer with or without any modification.

FlowPilot sends three packets within a duration to a port and expects at least one packet to be received back before timeout. The timeout can be configured.

Health check optionally provides a mechanism to carry the processor state information of the processing entities behind a port. The processor state information typically contains the state of the processing-entities/cores and the load experienced by each of them.

So, a payload blade that wishes to share its processor state information should provide that information in the health-check response packet. It is the payload blade's responsibility to form the health-check response packet in the prescribed format given by health-check module.

Sharing of processor state information is optional, so a payload blade may or may not choose to share that information.

Health-check request packet format

Source MAC	Destination MAC	Type	VLAN Tag			
			Priority	CFI	ID	EtherType
02 00 00 00 xx 00 Where xx denotes the logical-slot number of the FlowPilot Stateless Load Balancer.	02 00 00 00 00 xx Where xx denotes the port number of the port on which health-check is enabled.	0x8100 VLAN	0	0	4094	0x8809

Health-check response packet format

Source MAC	Destination MAC	Type	VLAN Tag				Processor TLV	End TLV
			Priority	CFI	ID	EtherType		
02 00 00 00 xx 00 Where xx denotes the logical-slot number of the FlowPilot.	02 00 00 00 00 xx Where xx denotes the port number of the port on which health-check is enabled.	0x8100 VLAN	0	0	4094	0x8809	Type: 1 Length: <0-255> Value: <0-255> Octets	Type: 0 Length: 0 Value: 0

Type = 1 (Processor TLV)
Type = 0 (End TLV)

Processor TLV structure:

Type (8-bits)	Length (8-bits)	Value (0-255 Octets)
1	<0-255>	

Inactive: Core for which state information is not available or doesn't want to share.

Active: Core for which state information is available.

Processor TLV's Value Octet (8-bits):

Bits 7:1	Bit 0
Load percent: 0-100 %	Inactive: 0
	Active: 1

Inactive: Core for which state information is not available or does not want to share.

Active: Core for which state information is available.

Examples for Processor TLV fillings in decimal numbered format:

1. Type=1; Length=2; Value=61 75;
   Two Active cores with 30% and 37% load respectively.
2. Type=1; Length=10; Value=43 51 63 75 83 33 49 91 93 97;
   Ten Active cores with respective loads of 21%, 25%, 31%, 37%, 41%, 16%, 24%, 45%, 46% and 48%.
3. Type=1; Length=0; Value=0
   Zero number of cores.

Processor TLV structure should be used to represent the processor state information of maximum 256 cores behind a port. Each core's state information is stored in a unique octet of the 'Value' field, which is an array of Octets and the array size is limited by 'Length' field.

Health-check packets will be tagged with a reserved VLAN ID 4094. The payload blades may use this vlan-tag to identify the health-check packets and treat them separately from rest of the data traffic.

Port states are either "Active/Inactive".

- The port state is Active if the port operational link is up. The health check is performed only on the active ports.
- The port is Inactive if the link is down or the health state is Unhealthy.

The health states are either Healthy, Unhealthy, or Initial State.

- The healthy state means that the port link is up and the payload is healthy.
- The unhealthy state means the payload that does not bounce back the packet within the timeout period. The health check ceases to monitor the payload on this port.

- The Initial State means that the health check mechanism is in-progress on a port whose link is up and the time-out period is not yet expired. It is also the default health state.
- The user/application checks the payload for the unhealthy reason and configures the port back to Healthy state.
- During initialization, the health check monitoring is done till the payload resends the special packet, at least once back to FlowPilot, to avoid user/application intervention. In this situation, the port health state continues to be in Initial State.

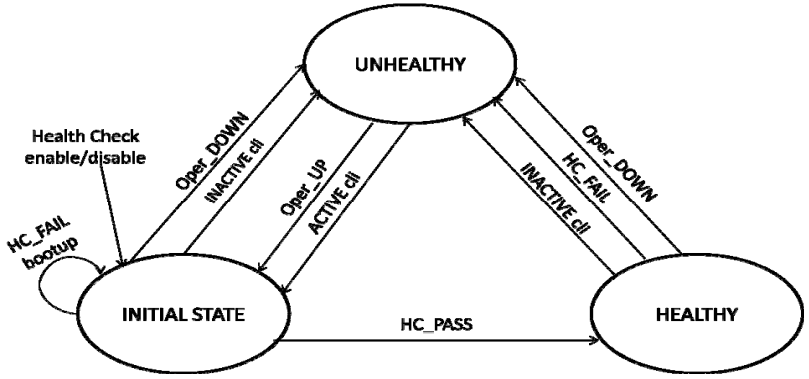
The port states, health states and processor state information are available through SRS API events. The application can register for NSM_MSG_HEALTH_CHECK_STATUS event. The structure below is sent to the application along with the event.

```
struct nsm_health_check_event_info
{
  int event;
#define SRS_MSG_HC_HEALTH_EVENT         1
#define SRS_MSG_HC_CPU_OVER_LOAD_EVENT  2
#define SRS_MSG_HC_CPU_UNDER_LOAD_EVENT 3
  char ifname[21];
  u_int32_t ifindex;
  u_int8_t port_state;
#define SRS_MSG_HC_PORT_ACTIVE   1
#define SRS_MSG_HC_PORT_INACTIVE 0
  union
  {
    struct
    {
      u_int8_t health_state;
#define SRS_MSG_HC_HEALTHY 1
#define SRS_MSG_HC_UNHEALTHY 0
    } health;
    struct
    {
       u_int8_t core_state;
#define SRS_MSG_HC_CORE_ACTIVE   1
```

```
#define SRS_MSG_HC_CORE_INACTIVE 0
       u_int8_t core_load;
    } cpu_core[256];
  } u;
```

The events are generated whenever a port becomes Healthy or Unhealthy.

Figure 2-5 Healthcheck State Transition Diagram



Oper_DOWN	:	The port becomes Operationally DOWN
Oper_UP	:	The port becomes Operationally UP
HC_PASS	:	Health Check procedure succeeds
HC_FAIL	:	Health Check procedure fails
HC_FAIL bootup	:	Health Check procedure fails, during board bootup
ACTIVE cli	:	'change-port-state ACTIVE' cli command
INACTIVE cli	:	'change-port-state INACTIVE' cli command

The user/application/FlowPilot Stateless Load Balancer FlowTable manipulation module (if health check is enabled) can use the "health states" to take necessary actions.

A processor-over-load event will be notified whenever the load experienced by a port exceeds a configured load-threshold. A processor-under-load event will be notified when the load subsides below the configured load-threshold value with a hysteresis of 10%. Always, an over-load event will precede an under-load event.

The CLI commands below can be used to configure a health check.

2.8.1 Configuring Health Check

2.8.1.1 Command Syntax

```
health-check IFNAME health-timeout <50-5000> [load-threshold <10-
100>]
```

```
no health-check IFNAME
```

2.8.1.2 Command mode

Config mode

2.8.1.3 Arguments

`IFNAME` - Interface name

`health-timeout` - Timeout interval in milliseconds.

`load-threshold` - Percentage threshold for Load experienced by payload blade.

2.8.1.4 Description

This command enables or disables the health-check feature on the selected set of interfaces.

Using the optional argument `load-threshold`, one may set the threshold for informing management interface about the load experienced by a port.

`no health-check` disables health check feature on the selected ports.

Example:

```
health-check xe5 health-timeout 5000 load-threshold 80
health-check xe13 health-timeout 1000
```

2.8.2 Changing the Port State

2.8.2.1 Command Syntax

```
change-port-state <ACTIVE/INACTIVE>
```

2.8.2.2 Command mode

Interface mode

2.8.2.3 Description

This command can be used by the user/application to change the port state when the payload/application port reports an unhealthy event or on any other event such as link up and link down.

2.8.3 Displaying the Health Check Status

2.8.3.1 Command Syntax

```
show health-check (load-status|)
```

2.8.3.2 Command mode

Exec mode

2.8.3.3 Description

This command displays the health-check status of the health-check enabled ports. The optional argument, `load-status`, will display the load experienced by the CPU cores behind the healthy ports.

Example:

```
#show health-check

IFNAME Port-State Health-State

xe5     Active     HEALTHY

xe6     Active     HEALTHY

xe7     Inactive   UNHEALTHY

xe8     Inactive   UNHEALTHY

#show health-check load-status
```

```
IFNAME  Port-State Health-State Threshold Cores %Loads(core<0-n>)

xe5     Active     HEALTHY       80%       10    0%,0%,100%,
0%,100%,0%,100%,0%,100%,0

xe6     Active     HEALTHY       95%        4    80%,90%,70%,100%

xe7     Inactive   UNHEALTHY

xe8     Inactive   UNHEALTHY
```

2.9 FlowTable Operations

The outgoing ports configured in the static channel and connected to the payload blades are listed in a table (FlowTable). Each index of this table is called a bucket. FlowPilot Stateless Load Balancer uses the hash-based load balancing algorithm and the output is a bucket location.

FlowTable has a default configuration with all the ports configured in a round robin fashion. If there are several modifications of FlowTable, the user can always revert back to the default configuration of the FlowTable.

There are 1024 entries in the FlowTable. Every entry is configured with a specific port present in the static channel group. The FlowTable configures the table in round-robin fashion in the order of the ports configured in the static channel group. The user/application can modify the table using the CLI commands listed below.

2.9.1 Modifying the FlowTable

2.9.1.1 Command Syntax

```
port-channel sa-fw-table assign-port <ge#/xe#> <cr> index-range <a>
to <b> every <n>

port-channel sa-fw-table replace-port <ge#/xe#> with <ge#/xe#> <cr>
every <n>

port-channel sa-fw-table default
```

2.9.1.2 Command mode

Interface mode (sa# only)

2.9.1.3 Description

The `port-channel sa-fw-table` commands enable you to modify the flow table as following:

The `port-channel sa-fw-table assign-port` command assigns/configures the port to the FlowTable Index.

- If index is not provided, then port is assigned to all the entries in the flow table.
- If only <a> is provided, then port is assigned to that entry only.
- If <a> and <b> are provided, then port is assigned to range <a> `to` <b>.
- If <every> <n> is provided, then port is assigned to every nth entry of flow table between index <a> to <b>. Index <a> is assigned with that port.

The `port-channel sa-fw-table replace-port <ge#/xe#> with <ge#/xe#>` command replaces the ports in the FlowTable.

- If <every> <n> is not provided, then all the occurrences of port given first are replaced with latter port.
- If <every> <n> is provided, then <n> occurrences are skipped and next occurrence will be replaced with latter port.

The "port-channel sa-fw-table default" CLI command is used to bring back default configuration for a static channel aggregator.

Example, for command `port-channel sa-fw-table replace-port`

FlowTable

xe1 indices

1 5 9 13 17 21 25 29 etc.

xe2 indices

2 6 10 14 18 22 26 30 etc.

For command,

```
port-channel sa-fw-table replace-port xe1 with xe2 skip 2
```

FlowTable

xe1 indices

1 5 13 17 25 29

xe2 indices

2 6 9 10 14 18 21 22 26 30

2.9.2 Displaying the FlowTable

2.9.2.1 Command Syntax

```
show static-channel-group <sa#> sa-fw-table port <xe#>|all

show static-channel-group <sa#> sa-fw-table index <a>-<b>

show static-channel-group <sa#> sa-fw-table

show static-channel-group <sa#> sa-fw-table last-modified port
<xe#>|all

show static-channel-group <sa#> sa-fw-table last-modified index
<a>-<b>

show static-channel-group <sa#> sa-fw-table last-modified
```

2.9.2.2 Command mode

Exec mode

2.9.2.3 Description

The show commands will be used to show current entries of FlowTable. There are four different commands which can be used to view the entries.

Examples:

```
show static-channel-group sa1 sa-fw-table port xe1
```

xe1 : 1, 5, 9, 13, 17, 21 .....

```
show static-channel-group sa1 sa-fw-table port all
```

if xe1,xe2,xe3 & xe4 are part of sa1

Xe1 : 1, 5, 9, 13, 17, 21, ...

Xe2 : 2, 6, 10, 14, 18, 22, ...

Xe3 : 3, 7, 11, 15, 19, 23, ...

Xe4 : 4, 8, 12, 16, 20, 24 .....

```
show static-channel-group sa1 sa-fw-table index 1-100
```

```
1    2    3    4    5    6    7    8    9

xe1 xe2  xe3   xe4  xe1   xe2   xe3  xe4  xe1

10   11   12   13   14   15   16   17    18

xe2  xe3  xe4  xe1  xe2  xe3  xe4   xe1   xe2
```

```
show static-channel-group sa1 sa-fw-table last-modified port xe1

show static-channel-group sa1 sa-fw-table last-modified index 0-
1023
```

Using the above commands, you can view the last modified entries of flow table. Outputs are similar to the previous show commands. However they show only entries that are modified since last configuration.

2.10 Switchover Configuration

The ports configured in the FlowTable are the ones that are reported healthy (incase health check is enabled) or the ones which have their link up. The user can configure the switchover ports which can be used when the health check application is reporting that the port is unhealthy or the link is down. When the user configures switch over list, then the ports in the switchover list are replaced with the port that is reporting failure in a round robin fashion. All the ports including switchover list should be part of the trunk. The below command can be used to configure switchover list.

2.10.1 Command Syntax

```
port-channel sa-fw-table on-inactive <ge#/xe#> switchover-list
<xe#1,xe#2,xe#3>|all
```

2.10.2 Command mode

interface mode (sa#)

2.10.3 Example

```
port-channel sa-fw-table on-inactive xe1 switchover-list
xe2,xe3,xe4
```

If xe1 becomes inactive due to either link down or unhealthy condition, the command replaces entries of xe1 in FlowTable with xe2, xe3, and xe4 in round robin fashion.

2.11 Port Failover Configuration

The member ports of the trunk can be assigned with active or backup role. The ports configured in the FlowTable are the ones that are in active role. Even though the backup ports are also part of the trunk, they will not be configured in the FlowTable and thus will not receive any traffic.

The user can configure the backup ports which can be used when the health check application is reporting that the active port is unhealthy or the link is down. When the user configures backup list, then the first port in the backup list will be chosen to replace with the port that is reporting failure in a round robin fashion. The below command can be used to configure backup-list.

2.11.1 Command Syntax

```
port-channel sa-fw-table backup-list <xe#1,xe#2,xe#3>
no port-channel sa-fw-table backup-list
```

2.11.2 Command mode

interface mode (sa#)

2.11.3 Example

```
port-channel sa-fw-table backup-list xe2,xe4
```

If xe1, xe2, xe3, and xe4 are part of static-aggregator and xe1 becomes inactive due to either link down or unhealthy condition, replace entries of xe1 in flow table with xe2.

Then if xe3 becomes inactive due to either link down or unhealthy condition, replace entries of xe3 with xe4.

Then if xe2 goes down, replace with xe4.

```
show command:
#show static-channel-group
  Static Aggregator: sa1
  Member:
   xe1
   xe2<BACKUP PORT,none>    /* backup inactive port */
   xe3
   xe4 <BACKUP PORT,xe1>    /* backup active port came in place of
xe1(inactive) */

no port-channel sa-fw-table backup-list
```

To remove configuration.

2.12 Target prediction

It is possible to predict the target where a specific packet will go when load-balanced by FlowPilot Stateless Load Balancer. This depends on number of parameters configured; some parameters can be configured by user, whereas some are internal configurations.

User need to provide bit pattern of exactly the same number of header fields, which are used in load-balancing configurationi.e., `port-channel load-balance` ... CLI command.

2.12.1 Command Syntax

```
#show target-predict ?
  interface  interface name of the sa ex:sa1

#show target-predict interface sa# ?
  extended  Source MAC address
  normal    Select extended hash field for load balancing

#show target-predict interface sa# normal/extended
  dipv4        IP address (e.g. 10.0.0.1)
  dipv6        IPv6 address in A:B:C:D:E:F:G:H format (e.g: 3ffe:506::1)
  dstmac       Enter the MAC address in HHHH.HHHH.HHHH format
  dstport      port number
  protocol-id  protocol identification number
  sipv4        IP address (e.g. 10.0.0.1)
  sipv6        IPv6 address in A:B:C:D:E:F:G:H format (e.g: 3ffe:506::1)
  srcmac       Enter the MAC address in HHHH.HHHH.HHHH format
  srcport      port number
```

2.12.2 Command mode

Exec mode and Privileged Exec mode

2.12.3 Example

Below are typical examples for this command, assuming required configuration executed prior to this.

```
1)For interface sa1
#show target-predict interface sa1 normal sipv4 1.2.3.4 dipv4 1.2.3.1
Bucket/port : 773/xe1

2)For interface sa2
#show target-predict interface sa2 extended sipv4 1.2.3.4
Bucket/port: 772/xe5
```

2.13 Port Weight Configuration

For every member port of an aggregator, user can assign weights. By default, equal weights (highest) are assigned to all the ports.

When user assigns a different weight to a port, buckets will be assigned to ports based on configured weight. If less weight is assigned, then extra buckets will be distributed to other ports. If more weight is assigned, then buckets assigned to other ports will be replaced by this port.

As soon as weights are assigned to the port, internally, average will be calculated every time keeping all port's weights in calculation, and buckets will be assigned to all the ports based on the port's weight.

If the ingress traffic is evenly distributed, it can be observed that traffic getting load-balanced based on weights.

NOTICE

- Port-weight feature is recommended to use only in preemptive mode.
- Port weight is single CLI, user should configure all ports' weights in one go.

2.13.1 Command Syntax

```
port-channel sa-fw-table port-weight-mode
[nonpreemptive|preemptive]
port-channel sa-fw-table port-weight ?
IFNAME  interface names of the with weights by comma(xe1-<1-10>,xe2-
<1-10>)

          ex: xe1-1,xe-2
```

2.13.2 Arguments

`<1-10>` - Port's weight

`IFNAME` - List of interface names separated by commas.

2.13.3 Command mode

interface mode (sa#)

2.13.4 Description

`port-weight-mode` command can be used to configure preemptive/nonpreemptive mode. This will decide behavior when there is failover. In preemptive mode, on failover, weight will be recalculated. In non-preemptive mode, active port will be simply replaced by backup port without taking weight into account.

`port-weight` command used to configure port's weight. User can assign same weights to multiple ports using single command.

By default, all the ports will be assigned with weight of 10.

The weight can be assigned in the range of 1 to 10, where 1 signifies least weight, least number of buckets will be assigned and 10 signifies highest weight, that is maximum number of buckets are assigned to the port considering weights of other ports.

To assign default buckets to port mapping, user can call `set default` command. It will again equally distribute buckets in forwarding tables to ports.

`set default` command can used to remove weight configuration.

2.13.5 Example

```
port-channel sa-fw-table port-weight-mode nonpreemptive
port-channel sa-fw-table port-weight 1 xe57,xe58,xe59
port-channel sa-fw-table port-weight 2 xe60
```

2.14 Hash-Algorithm Selection

In FlowPilot Stateless Load Balancer, user can select a hash algorithm among the two algorithms provided, based on the user load-balancing requirement. The algorithm one (default algorithm) will support symmetric hashing in both hash A and hash B of RTAG7 hash engine and the algorithm two can be selected for better load-balancing. But in algorithm two, only hash A will support symmetric hashing. Target prediction can be used to verify the difference between the two algorithms.

2.14.1 CLI Commands

```
set-hash-algorithm ?
  extended  Select extended hash field for load balancing
  normal    Select normal hash field for load balancing

set-hash-algorithm normal ?
  algorithm  hash algorithm
set-hash-algorithm normal algorithm ?
  <1-2>  hash algorithm number
```

2.14.2 Arguments

```
extended Select extended hash field for load balancing
normal    Select normal hash field for load balancing
 <1-2> hash algorithm number
```

2.14.3 Command mode

Configuration mode

2.14.4 Description

`set-hash-algorithm` command effects load-balancing globally and can be used to select hashing algorithm. By default, hash algorithm one is configured. User can also select algorithm two for better load-balancing. In hash algorithm two, only hash A (enhanced) will support symmetric hashing.

Variation in load-balancing of algorithm one and two can be observed by target-prediction command explained in *Target prediction* on page 43.

2.14.5 Example

```
set-hash-algorithm normal algorithm 2
```

This command will select hash algorithm two of Hash A for selecting the same in hash B. User may need to use extended option instead of normal. The current hash algorithm can be known using `show running-config` command.

2.15 Pass-through Functionality

FlowPilot Stateless Load Balancer needs all the traffic to be forwarded and not to be copied to the management processor. This is known as Pass-through and needs to be configured explicitly. The command below can be used for the same.

2.15.1 Command Syntax

```
pass-through
```

```
no pass-through
```

2.15.2 Command mode

Config mode

2.16 Pass-through exceptions

2.16.1 Command Syntax

```
no pass-through lacp
```

2.17 MAC learning

MAC learning should be disabled on all the ports on Fabric chipset for FlowPilot Stateless Load Balancer to work.

SRstackware server load-balancing solution is based on L3 routing principles, which load balances the incoming traffic received at hub blade towards the target hosts residing on one or more payload/processing blades. In L3 terms, this solution in principle equals to Equal Cost Multi path (ECMP) load-balancing.

This solution is supported on Artesyn ATCA-F140 blade and runs on the Ethernet fabric chipset. This software is part of SRstackware module available on the ATCA switch blades.

The typical hardware configuration includes a pair of redundant ATCA-F140 blades on an ATCA chassis, which has payload blades in other slots. The ATCA-F140 can receive traffic from RTM ports and load balance the traffic towards the processing blades/payload blades. The payload blades typically run customer applications.

Equal Cost Multi Path (ECMP) is a technique for routing packets among multiple paths of equal cost. If multiple equal cost paths (next-hops) to the same destination exist, then this solution can be used to provide load-balancing among the redundant paths. In other words, to load balance the traffic which have same destination IP/subnet to a group of payload blades or ports or next-hops (NH), this solution can be used.

This solution supports four load-balancer groups (ECMP group) with 1024 members in each group.

A load-balancer group is a collection of pool members (next-hops), which have the same cost metric. Each ECMP group member will have an entry in ECMP table. While routing traffic, if ECMP enabled for a route, an ECMP group is selected for routing, and one of the ECMP group members from ECMP table will be selected as final next-hop. ECMP member selection will be done through a hash algorithm applied on load balancing criterion based on source IP.

SRstackware server load-balancing solution provides interface to configure ECMP groups and members. This solution will support complete L3 functionality.

3.1 ECMP Group Configuration

ECMP group is termed as pool, includes properties described below along with group members.

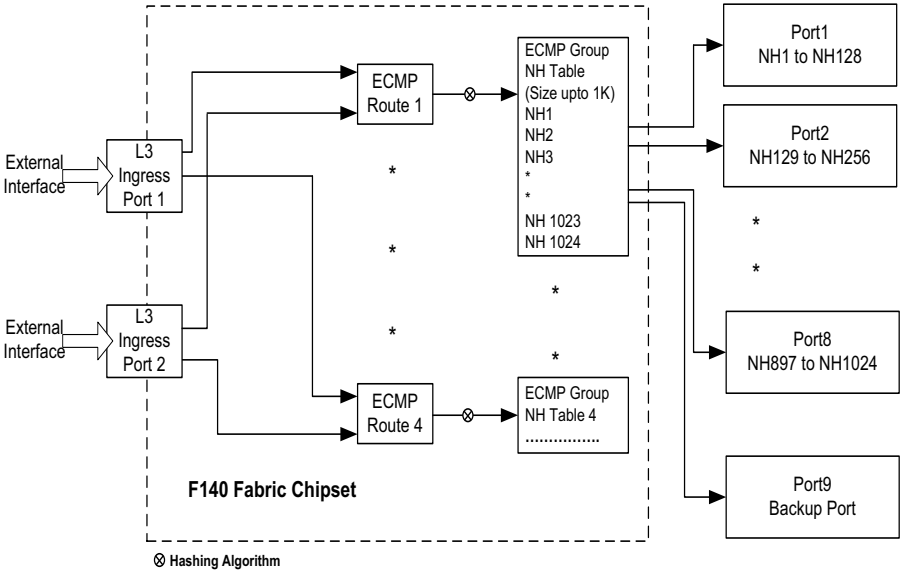
- Destination prefix: Destination IP/Subnet of traffic to be associated to an ECMP group.
- Load-balancing criteria: The algorithm used to distribute load among the members of the pool. As of now, packet source IP is the supported criteria for load-balancing and it is the default method.

- ECMP Group (pool) state: ECMP Group state is a software controlled state; a group can be in Active/Inactive state. If pool is in Inactive state, all routed traffic through this ECMP group will be dropped. During pool Active state, traffic will be routed through the selected ECMP members, provided the next-hop MAC and interface information is populated for member.
- ECMP Member: ECMP members are configured with next-hop IP address. Next-hop MAC and interface information are required to route traffic through an ECMP group member. This information can be populated through Gratuitous Address Resolution Protocol (GARP)/ARP from the payloads/Next-hop corresponding to ECMP group member.

Static configuration of next-hop MAC address and interface information is also supported.

A typical ECMP group configuration is shown in following figure.

Figure 3-1 ECMP Group Configuration



With this configuration, user can configure up to four pools (ECMP routes/groups) with associated destination prefix/IP and then add members with associated NH IP to the pool. Number of members depend on the connected next-hops where user wants to load-balance the traffic.

Until pool is not activated, all routed traffic through this ECMP group will be dropped. Once pool is activated, software will be ready for receiving GARP/ARP packets. Once member's next-hop information is populated either statically or through GARP/ARP, based on the load-balancing, traffic will be routed through selected next-hop.

After activating the pool, when traffic ingresses from one of the L3 ingress ports, based on the destination IP/prefix the traffic will be routed. If the selected route is ECMP, based on the hashing algorithm, it will select one of the entry in ECMP group table and corresponding next-hop. Switch logic will route the packet to the next-hop from the corresponding port.

There can be up to 4 ECMP groups (pools) and each pool can have maximum 1024 members (Next-hops). If there are eight egress ports, each can host 128 next-hops.

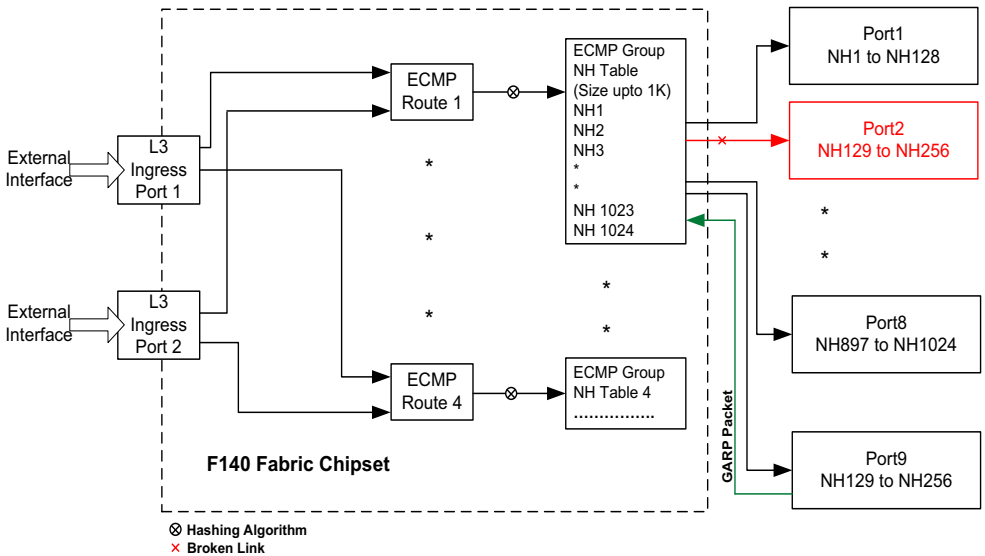
The following are the features of FlowPilot ECMP Load Balancer:

- Four ECMP routes (pools) are supported with 1024 next-hops (pool members) in each pool.
- ECMP load-balancing is based on packet source IP. As of now, only source IP based load-balancing is provided.
- Pool configuration includes pool create, destroy, activate and deactivate.
- Pool member configuration includes add/delete member to/from pool.
- Update of pool members (next-hops) is supported based on ARP/GARP from connected ports or through static configuration.
- More preference for ECMP route is given over RIP, OSPF, and Static route for the same destination network.

3.2 Failover Scenario

As shown in Figure 3-2, if Port 2 goes down, entries related to the next-hops hosted by Port 2 will be invalid and packets hashed to these entries will be dropped. User needs to implement a mechanism to detect the failure and initiate GARP packets for these next-hops from the backup port/blade. Once GARP packets are received for these next-hops and entries are populated by the software, packets hashed to these entries will be sent to the new active port.

Figure 3-2 Failover Scenario



3.3 User Interface

The following are the CLI commands of ECMP Load-balancing.

3.3.1 lb-ecmp-pool <pool_id>

Description

This command creates a pool with ID <pool_id> and moves to the "Pool mode". If pool is already created, it will move to "Pool mode". pool_id: 1-4 Unique identifier for the pool

Synopsis

```
lb-ecmp-pool <pool_id>
```

pool_id: 1-4 Unique identifier for the pool

Command mode

Config mode

3.3.2 no lb-ecmp-pool <pool_id>

Description

This command removes a pool with ID <pool_id>.

Synopsis

```
no lb-ecmp-pool <pool_id>
```

pool_id: 1-4 Unique identifier for the pool

Command mode

Config mode

3.3.3 pool dst-prefix <destination prefix>/<mask>

Description

This command associates destination IP/prefix to a pool for load-balancing. Prefix must be configured prior to other pool attributes or pool member configuration.

Synopsis

```
pool dst-prefix <destination prefix>/<mask>
```

prefix: Prefix Destination prefix
mask: 0-32 Prefix mask length

Command mode

Pool mode

3.3.4 pool lb-method <lb_method>

Description

This command associates load-balancing method to a pool. As of now, source IP based load balance is supported. By default `lb_method` is "src_ip".

Synopsis

```
pool lb-method <lb_method>
```

lb_method: src_ip Method used to distribute load among pool members.

Command mode

Pool mode

3.3.5 pool subnet-id <nexthop subnet_id>

Description

This command allows to associate subnet to the pool when all pool members belong to same subnet.

Synopsis

```
pool  subnet-id <nexthop subnet_id>
```

nexthop subnet_id: <prefix> /<mask> Subnet to which pool members belong

Command mode

Pool mode

3.3.6 pool description <description>

Description

This command associates some description to a pool.

Synopsis

```
pool description <description>
```

Command mode

Pool mode

3.3.7 pool activate

Description

This command activates the pool corresponding to current mode. Pool and member configuration will be activated (added) at hardware.

ARP is required for populating member entries. A sample utility for ARP generation is packaged with SRStackware FlowPilot RPM. For more details, refer the section Appendix B, *ARP Generator Sample Utility, on page 70*.

Synopsis

```
pool activate
```

Command mode

Pool mode

3.3.8 no pool activate

Description

This command deactivates the pool corresponding to current mode. Pool and member configuration will be deactivated (removed) at hardware.

Synopsis

```
no pool activate
```

Command mode

Pool mode

3.3.9 member add <member_id> ip-address <ip_address>

Description

This command creates a pool member and adds member to the pool corresponding to the current mode. Pool ID will be associated to the member automatically based to current mode. Pool members can be added only when pool is in "Inactive" state.

Before adding a pool member, user need to create an intervlan/L3 interface, assign IP address to the interface in same subnet as next hops and then add all processing blade connected ports in the same VLAN.

Synopsis

```
member add <member_id> ip-address <ip_address>
```

member_id: 1-1024 Unique identifier for the pool member
ip_address: IP address IP address of pool member

Command mode

Pool mode

3.3.10 member del <member_id>

Description

This command deletes a pool member from the pool corresponding to current mode. Pool members can be deleted only when pool is in "Inactive" state.

Synopsis

```
member del <member_id>
```

member_id: 1-1024 Unique identifier for the pool member

Command mode

Pool mode

3.3.11 member update <member_id> mac-address <mac_address> interface <interface>

Description

This command allows static configuration of next-hop MAC address and interface of a pool member. When next hop MAC address and interface of pool members are configured statically, pool member state will be "Valid", irrespective of pool state (Active or Inactive).

Synopsis

```
member  update <member_id> mac-address <mac_address> interface
<interface>
```

member_id: 1-1024 Unique identifier for the pool member
mac_address: MAC address MAC address of nexthop
interface: Interface name Interface of nexthop

Command mode

Pool mode

3.3.12 Show lb-ecmp-pool [<pool_id>]

Description

This command shows information of all configured pools of specified pool ID.

Synopsis

```
Show lb-ecmp-pool [<pool_id>]
```

Pool_id: 1-4 Unique identifier for the pool

Command mode

Privileged Exec mode

3.3.13 Show lb-ecmp-pool <pool_id> member [<member_id>]

Description

This command shows all pool members' information pertaining to a pool ID when specific `member_id` is not provided. If `member_id` is provided, information of pool member with `member_id` in `pool_id` will be shown.

Synopsis

```
Show lb-ecmp-pool <pool_id> member [<member_id>]
```

Pool_id: 1-4 Unique identifier for the pool
Member_id 1-1024 Unique identifier for the pool

Command mode

Privileged Exec mode

A.1 Default Configuration

SRstackware is delivered with the following default configuration files, at `/etc/opt/srstackware/config/` directory:

- `srs.1.cfg` for ATCA-F140 in logical slot1
- `srs.2.cfg` for ATCA-F140 in logical slot2

FlowPilot Stateless Load Balancer is also delivered with separate chassis and slot specific default configuration files at `/etc/opt/srstackware/config/flowpilot/` directory:

- `srs.<chassis-ID>.<logical-slot>.fp40.cfg`

For example:

- ATCA-F140 on AXP-1440 chassis with FlowPilot Stateless Load Balancer logical slot-1 config file:
  - `srs.1440.1.fp40.cfg`
- ATCA-F140 on AXP-640 chassis with FlowPilot Stateless Load Balancer logical slot-2 config file:
  - `srs.640.2.fp40.cfg`

In addition to the default FlowPilot Stateless Load Balancer configuration files, chassis and license specific default Enable and Disable FlowPilot Stateless Load Balancer configuration scripts are also packaged at `/etc/opt/srstackware/config/flowpilot/` directory:

- `enable_<chassis-ID>_flowpilot40_config`
- `disable_<chassis-ID>_flowpilot40_config`

For example:

- `enable_1440_flowpilot40_config`
- `disable_640_flowpilot40_config`

Before using the default FlowPilot Stateless Load Balancer configuration files, it is required to configure ATCA-F140 as per the chassis setup. Run the command below:

`trid-cfg –a -y -i /opt/bladeservices/cfg/trid_1440_def_fp` for AXP-1440 chassis

OR

`trid-cfg –a -y -i /opt/bladeservices/cfg/trid_640_def_fp` for AXP-640 chassis.

For more information, refer to the *ATCA-F140 Basic Blade Services Programming Reference Guide* for more information.

FlowPilot Stateless Load Balancer's default configuration file has the configurations required to enable the features. You should copy/rename the default FlowPilot Stateless Load Balancer configuration file as `srs.1.cfg/srs.2.cfg` and reboot the blade. It is always recommended to take a copy of `srs.1.cfg/srs.2.cfg` files before copying default FlowPilot Stateless Load Balancer configuration files.

For example:

- `cp /etc/opt/srstackware/config/flowpilot/srs.1440.1.fp40.cfg /etc/opt/srstackware/config/srs.1.cfg`

It is recommended to save the current configuration, once the blade boots up with default FlowPilot Stateless Load Balancer configuration file.

`#write file` (`imish` CLI command)

The following explains the default configuration file of FlowPilot Stateless Load Balancer:

- Disables MAC learning on the Fabric chipset
- VLAN Configuration
  - VLAN 2000 and 3000 is created specifically to be used by FlowPilot Stateless Load Balancer. These VLANs are added as outer VLAN, when traffic enters from left and right port respectively using vlan-stacking.
  - Left port is configured with VLAN 2000 as a customer edge port and right port is configured with VLAN 3000 as a customer edge port.
  - All backplane ports are configured with both VLAN2000 & 3000 as a provider-ports.

- Configuration to put uplink and downlink ports to STP BLOCK state.
  - This is present automatically in the running config file, if the link associate commands are given on the ports. During boot time, the uplink and downlink ports are initially kept in block state and when the link associate commands are executed, the ports are made to forward state.
- Creates a static channel with all back-plane ports as its members.
  - Single static channel will be created with all backplane ports
- Configures load-balance configurations on static-channel. Default configuration file has the following load-balance configuration:
  - port-channel load-balance src-ip enhanced use-outer-header
  - port-channel load-balance dst-ip enhanced extended-hash use-outer-header
- Configures health-check on all payload ports. (Adds match-list to detect health-check special packets.)
- Enables pass-through configuration to stop traffic reaching management processor.
- Associates uplink port with downlink port with link association command.
- Adds match-list commands to redirect the packets from uplink->backplane ports->Downlink and vice-versa.
- Finally, the uplink and downlink ports are pushed to STP FORWARD state to start FlowPilot Stateless Load Balancer operation.

A.2 Port Assumptions

The default configuration assumes the left and right ports as Front panel QSFP+ ports (Group mode). All the backplane ports towards payloads are configured as one static channel.

A.3 How to Customize the Configuration

The default FlowPilot Stateless Load Balancer configuration can be customized easily using the configuration wizard script,
`/opt/srstackware/scripts/generate_fp_scripts.sh`

This script generates the below customized FlowPilot Stateless Load Balancer configuration scripts:

- `enable_flowpilot.custom`
- `disable_flowpilot.custom`

These scripts can be invoked using imish, `imish -f enable_flowpilot.custom` and `imish -f disable_flowpilot.custom`

To use these scripts:

1. Run the command.
   `trid-cfg –a -y -i /opt/bladeservices/cfg/trid_1440_def_fp` for AXP-1440 chassis.
   OR
   `trid-cfg –a -y -i /opt/bladeservices/cfg/trid_640_def_fp` for AXP-640 chassis.
2. Boot the ATCA-F140 with Default FlowPilot Stateless Load Balancer configuration and save the running configuration using imish CLI command, `#write file`. To boot up with default FlowPilot Stateless Load Balancer configuration, see Appendix A, *Default Configuration*.
3. Run the corresponding disable config script without any modifications:
   `imish -f disable_<chassi-ID>_flowpilot40_config`

   For example, if the blade is booted with `srs.640.1.fp40.cfg`, the corresponding disable config script is `disable_640_flowpilot40_config`.
4. Run the Configuration wizard script:
   `/opt/srstackware/scripts/generate_fp_scripts.sh`

   The `enable_flowpilot.custom` and `disable_flowpilot.custom` files will be generated.
5. Run the customized enable config script
   `imish -f enable_flowpilot.custom`
   Save this custom configuration as the default configuration.
6. To disable the custom configuration, the corresponding disable script, that is, `disable_flowpilot.custom` should be loaded using imish.

A.4 Matchlist Commands

FlowPilot Stateless Load Balancer relevant commands for matchlists are shown below. Refer to the SRstackware documentation in Appendix C, *Related Documentation, on page 73*, for the complete list of matchlists.

A.4.1 Command Syntax

```
match-list <matchlist-ID> <base/fabric>

match port inports <port-name>

match l3param srcip <IP address> <mask>

match l3param dstip <IP address> <mask>

rule match-list <matchlist-ID> action modify-vlanid <VLAN-ID>

rule match-list <matchlist-ID> action redirect-port <port-name>
```

A.4.2 Command mode

match commands in Matchlist mode

rule command in Configure mode

A.5 VLAN Stacking command

The VLAN stacking command is as given below.

A.5.1 Command Syntax

```
switchport vlan-stacking customer-edge

switchport vlan-stacking provider-port
```

A.5.2 Command mode

Interface mode

A.6 MAC Learning

The command below can be used to disable MAC learning on the fabric chipset.

This means that the payloads which are part of static channel group cannot have a dedicated IP address and have connection terminations from external equipment.

A.6.1 Command Syntax

```
no bridge <bridge #> acquire
```

A.6.2 Command mode

Config mode

A.7 Static Channel configuration

The static channel can be configured using the below command.

A.7.1 Command Syntax

```
static-channel-group <channel number>
```

A.7.2 Command mode

Interface mode

Prerequisite:

Create an intervlan interface and assign IP address in same subnet as target hosts. Add all processing blade connected ports in the same VLAN.

The following sections provide information about ECMP configuration, using a sample configuration. The ECMP configuration includes create, remove, activate and deactivate a pool and the pool member configuration includes add/delete member to/from pool.

B.1 Creating FlowPilot ECMP Load Balancer

The following are the steps for creating FlowPilot ECMP Load Balancer:

1. Create an ECMP pool.
   Configuration commands:
   #**configure terminal**
   (config)#**lb-ecmp-pool 1**

   Validation commands:
   #**show lb-ecmp-pool 1**

```
Pool ID: 1
Description       :
Destination prefix:
Subnet            :
Load balance method: Source IP address
State             : INACTIVE
Member count      : 0
```

2. Configure pool parameter destination prefix and pool members.

   Configuration commands: These commands should be executed only in pool mode.
   #**configure terminal**
   (config)#**lb-ecmp-pool 1**

   To provide pool description (This is an optional command):
   (config-ecmp-pool)#**pool description DEMO POOL**

   To set the destination IP of a packet to be routed:

```
(config-ecmp-pool)#pool dst-prefix 10.10.10.0/24
```

To add a pool member with member ID 1 with address 192.168.8.1:

```
(config-ecmp-pool)#member add 1 address 192.168.8.1
```

To add a pool member with member ID 2 with address 192.168.8.2:

```
(config-ecmp-pool)#member add 2 address 192.168.8.2
.
.
.
```

Pool member addresses mentioned above are the IP addresses of applications running on the payload blades.

NOTICE

Prior to member configuration, make sure that an interface within this subnet exists; otherwise "member" configuration will be failed.

Validation commands:

```
#show lb-ecmp-pool 1

Pool ID: 1
Description       : DEMO POOL
Destination prefix: 10.10.10.0/24
Subnet            :
Load balance method: Source IP address
State             : INACTIVE
Member count      : 2

#show lb-ecmp-pool 1 member

Member-ID  IP-Address        MAC-Address    Port  State
1          192.168.8.1     0000.0000.0000 vlan2.2 Invalid
2          192.168.8.2     0000.0000.0000 vlan2.2 Invalid
```

3. Activate the pool.
   Configuration commands:

```
#configure terminal
```

```
(config)#lb-ecmp-pool 1
(config-ecmp-pool)#pool activate
```

Validation commands:

```
#show lb-ecmp-pool 1

Pool ID: 1
Description       : DEMO POOL
Destination prefix: 10.10.10.0/24
Subnet            :
Load balance method: Source IP address
State             : ACTIVE
Member count      : 2

#show lb-ecmp-pool 1 member

Member-ID  IP-Address        MAC-Address    Port   State
1          192.168.8.1     0000.0000.0000 vlan2.2 Invalid
2          192.168.8.2     0000.0000.0000 vlan2.2 Invalid
```

4. Pool is active now, but pool members are in "Invalid" state. By sending ARP/GARP packets you can make them valid. In this demo, GARP is sent to 192.168.8.1 pool member. After sending GARP to 192.168.8.1 pool member, the member status is as follows:

```
#show lb-ecmp-pool 1 member

Member-ID  IP-Address        MAC-Address    Port   State
1          192.168.8.1     1234.1234.1234 vlan2.2 Valid
2          192.168.8.2     0000.0000.0000 vlan2.2 Invalid
```

B.2 Modifying FlowPilot ECMP Load Balancer

The following are the steps for modifying an existing FlowPilot ECMP Load Balancer:

1. Deactivate the required pool.

NOTICE

Deactivation will remove all configurations from hardware.

Configuration commands:

```
#configure terminal
(config)#lb-ecmp-pool 1
(config-ecmp-pool)#no pool activate
```

Validation commands:

```
#show lb-ecmp-pool 1

Pool ID: 1
Description       : DEMO POOL
Destination prefix: 10.10.10.0/24
Subnet            :
Load balance method: Source IP address
State             : INACTIVE
Member count      : 2

#show lb-ecmp-pool 1 member

Member-ID  IP-Address        MAC-Address    Port  State
1          192.168.8.1     0000.0000.0000 vlan2.2 Invalid
2          192.168.8.2     0000.0000.0000 vlan2.2 Invalid
```

NOTICE

You can add new member (s) or delete existing member(s) from pool in its deactivated state.

2. Modifying an existing pool.
   In this step, we are deleting existing member with member id 2 and adding a new member with member id 3.
   Configuration commands:

```
(config-ecmp-pool)#member del 2

(config-ecmp-pool)#member add 3 address 192.168.8.50
```

   Validation commands:

```
#show lb-ecmp-pool 1

Pool ID: 1
Description       : DEMO POOL
  Destination prefix: 10.10.10.0/24
  Subnet            :
  Load balance method: Source IP address
  State             : INACTIVE
  Member count      : 2

#show lb-ecmp-pool 1 member

Member-ID  IP-Address        MAC-Address    Port   State
1          192.168.8.1     0000.0000.0000 vlan2.2 Invalid
3          192.168.8.50    0000.0000.0000 vlan2.2 Invalid
```

3. Activate the pool (deactivated in step 1) and send ARP/GARP packets to make pool members valid.
   Configuration commands:

```
#configure terminal
(config)#lb-ecmp-pool 1
(config-ecmp-pool)#pool activate
```

   Validation commands:

```
#show lb-ecmp-pool 1

Pool ID: 1
Description       : DEMO POOL
Destination prefix: 10.10.10.0/24
Subnet            :
```

```
Load balance method: Source IP address
State              : ACTIVE
Member count       : 2

#show lb-ecmp-pool 1 member

Member-ID  IP-Address        MAC-Address    Port  State
1          192.168.8.1     1234.1234.1234 vlan2.2 Valid
3          192.168.8.50    0000.0000.0000 vlan2.2 Invalid
```

B.3 Removing FlowPilot ECMP Load Balancer

The following are the steps for removing FlowPilot ECMP Load Balancer:

1. Deactivate the pool.
   Configuration commands:
   ```
   #configure terminal
   (config)#lb-ecmp-pool 1
   (config-ecmp-pool)#no pool activate
   ```
2. Remove the pool (this command to be used in CONFIG mode).
   Configuration commands:
   ```
   #configure terminal
   (config)#no lb-ecmp-pool 1
   ```

B.4 ARP Generator Sample Utility

Before pumping the data traffic for load balancing, it is required to update the MAC of ECMP Pool Members or Next Hops. This is achieved through exchange of ARP or GARP protocol packets between System Controllers and each pool member. ARP/GARP packets can be initiated either by pool members or payloads or by system controller (ATCA-F140).

A sample utility is packaged with SRStackware FlowPilot RPM which consists of a shell script `/opt/srstackware/ecmp/arp_pool_member.sh` and a binary `/opt/srstackware/ecmp/get_pool_member` to send ARP packets to pool members.

`get_pool_member` is an SRS_API based application, which logs all ECMP group pool members IP in */tmp/pool_member* file. (For more details on writing SRS_API based application, refer to *SRstackwareAPI Developer Guide*). The source file of this application is placed at */usr/share/srstackware/srs_get_ecmp_member.c* for reference; this can be customized based on the requirement.

`arp_pool_member.sh` script executes `get_pool_member` binary and initiates arp to all the pool members. As ARP entries could time out, it is required to send ARP packets in regular intervals. In this script `arping` is done for every 5 seconds. Based on the requirement, the time interval can be adjusted. This script needs to be called after the ECMP pool is created and made active.

Usage:

```
/opt/srstackware/ecmp/arp_pool_member.sh
```

If `arping` does not pick the proper interface automatically, you need to provide the interface name manually in the script as shown below.

```
/sbin/arping -c 1 $i -q -I <Interface name>
```

Example:

If next hops are connected with vlan2.101 interface then you need to update the script as mentioned below.

```
/sbin/arping -c 1 $i -q -I vlan2.101
```

C.1 Artesyn Embedded Technologies - Embedded Computing Documentation

The publications listed below are referenced in this manual. You can obtain electronic copies of Artesyn Embedded Technologies - Embedded Computing publications by contacting your local Artesyn sales office. For released products, you can also visit our Web site for the latest copies of our product documentation.

1. Go to www.artesyn.com/computing/support/product/technical-documentation.php.
2. Under FILTER OPTIONS, click the **Document types** drop-down list box to select the type of document you are looking for.
3. In the Search text box, type the product or document name and click Filter.

Table C-1 Artesyn Embedded Technologies - Embedded Computing Publications

Document Title and Source	Publication Number
SRstackware ATCA-F140 Document Collection	6806800N94
BBS on ATCA-F140 with SRstackware Programmer's Reference Guide	6806800U52

ARTESYN™
EMBEDDED TECHNOLOGIES

Artesyn Embedded Technologies, Artesyn and the Artesyn Embedded Technologies logo are trademarks and service marks of Artesyn Embedded Technologies, Inc.
All other product or service names are the property of their respective owners.
© 2016 Artesyn Embedded Technologies, Inc.