# IMPLEMENTING JUNIPER NETWORKS EX SERIES ETHERNET SWITCHES WITH WAN ACCELERATION SOLUTIONS

Although Juniper Networks has attempted to provide accurate information in this guide, Juniper Networks does not warrant or guarantee the accuracy of the information provided herein. Third party product descriptions and related technical details provided in this document are for information purposes only and such products are not supported by Juniper Networks. All information provided in this guide is provided "as is", with all faults, and without warranty of any kind, either expressed or implied or statutory. Juniper Networks and its suppliers hereby disclaim all warranties related to this guide and the information contained herein, whether expressed or implied of statutory including, without limitation, those of merchantability, fitness for a particular purpose and noninfringement, or arising from a course of dealing, usage, or trade practice.

## Table of Contents

## Table of Figures

## Introduction

WAN acceleration solutions that accelerate a broad range of applications and protocols over the wide area network are a requirement for today's distributed enterprise network. Juniper Networks® EX Series Ethernet Switches, running Juniper Networks Junos® operating system, provide complete end-to-end network infrastructure solutions for enterprise campus, branch, and data center deployments. This document will describe how to integrate EX Series Ethernet Switches with Juniper Networks WXC Series Application Acceleration Platforms for several scenarios, including inline and off-path/packet interception modes. This document will discuss in depth how one of the off-path deployments, called "external mode," uses Junos OS features such as filter-based forwarding on the EX Series switches to redirect the appropriate packets to the WXC Series device for optimization.

There is one common concern with external mode deployments. When the WAN acceleration device is connected to the EX Series switch through an intermediary Layer 2 switch, the EX Series switch cannot detect the loss of the WXC Series device, since the two are not directly connected. This causes the EX Series switch to redirect packets into a black hole. However, the Junos OS software's real-time performance monitoring (RPM) feature and Junos Script on the EX Series Ethernet switch provide a dynamic solution for tracking the health of the next hop, which in this scenario is the WAN acceleration device, so that proper actions can be taken to avoid traffic "black holing," should a WXC Series failure be detected by the RPM probes.

## Scope

This document is targeted at system engineers and other technical audiences interested in designing and implementing Juniper Networks EX Series Ethernet Switches with Juniper Networks WAN acceleration solutions in distributed enterprise networks.

## Design Considerations

This implementation guide describes how to deploy Juniper Networks WAN acceleration solutions with the EX Series Ethernet Switches in a variety of scenarios, including inline and off-path/packet interception modes.

## Protocol Operation

The topologies and protocols are specific to the type of deployment and will be discussed in detail in the following sections.

## Implementation

### Inline Mode Deployment

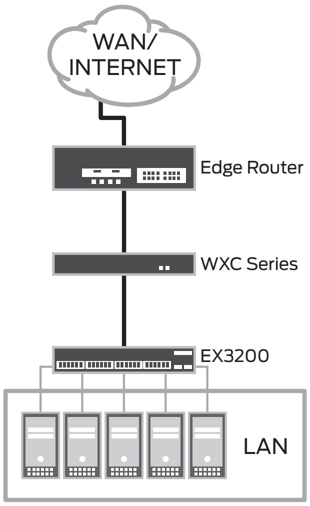

Figure 1: Inline mode deployment

In most simple network deployments, the WAN acceleration platform is deployed in the "inline" mode. In a Juniper network, this means that all packets to and from the WAN must pass through the WXC Series device to receive WAN optimization (acceleration, reduction, quality of service, etc.) as needed. All decompression and de-encapsulation will be performed on the remote WXC Series device (see Figure 1). The WXC Series device is transparent to the routed network, and inline mode is ideal for small networks due to its simplicity. The minimal configurations required on the EX Series Ethernet switch are as simple as if the switch were connecting directly to the WAN edge router. The WXC Series platform also offers hardware bypass support, so that if a problem such as a power failure occurs, the device will automatically "switch-to-wire," in effect becoming an Ethernet cable so that the network operations can continue without interruption.

## External Mode in Off-Path Deployment

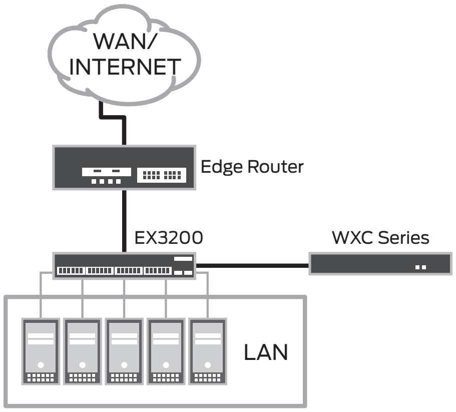

Figure 2: Off-Path mode deployment

Inline mode may not be ideal for some network topologies such as those where applications are latency sensitive, and it may not be optimal for traffic to pass through the WAN acceleration device. In those cases, network administrators like to control which traffic is redirected to the WXC Series platform and which traffic is switched or routed directly to the WAN edge router. In these situations, a WXC Series device can be installed in "off-path mode," also called "packet interception mode." In off-path mode, the WXC Series platform is connected to the EX Series switch in a one-armed fashion as shown in Figure 2. In an off-path deployment, one of the first design considerations is to determine which traffic needs to be redirected by the EX Series switches to the WXC Series, with the goal of only forwarding traffic that requires WAN optimization services. However, this is not a strict requirement since the WXC Series will "pass through" traffic that has no WAN optimization policy configured. There are three ways that traffic can be redirected to an off-path WXC Series device:  Route Injection Mode via RIPv2; Web Cache Communication Protocol (WCCP) via WCCPv2; and External Mode via policy routing. In this document, the third method will be discussed in detail. External Mode via policy routing can be achieved on EX Series switches using the filter-based forwarding feature available with Junos OS.

Filter-based forwarding (FBF) is a Junos OS feature that allows network administrators to control the next-hop selection for customer traffic by defining the input packet filters that examine packet header fields such as IP source address, IP destination address, IP protocol field, and source and destination TCP/UDP port numbers. If a packet satisfies the match conditions of the firewall filter, the packet is forwarded to the routing instance specified in the filter action statement. Once the routing instance is identified, traditional destination-based routing occurs using the routing table within that routing instance. Filter-based forwarding provides a very simple yet powerful policy-based routing table selector and can be used to redirect certain traffic to the WXC Series device in the off-path external mode deployment. The FBF feature, supported on Juniper Networks EX3200 Ethernet Switches and EX4200 Ethernet Switches since Junos OS 9.4, and on the Juniper Networks EX8200 line of Ethernet switches since Junos OS 9.6, can only be configured as an input filter on the Layer 3 interface or routed VLAN interface (RVI) on the EX Series switches. Outbound filter-based forwarding is not supported.

Figure 3 shows an example of off-path deployment using external mode. The WAN acceleration device "WXC_Local" is connected to the Juniper Networks EX8208 Ethernet Switch in one-armed fashion in an enterprise campus network. The EX8208 is the Layer 3 switch in the network core layer and interconnects the EX4200 Layer 2 access switch and the Juniper Networks M10i Multiservice Edge Router. Traffic from the servers in the local LAN 20.20.20.0/24 to the remote LAN 30.30.30.0/24 will be redirected through the WXC Series device "WXC_Local" and will be optimized.

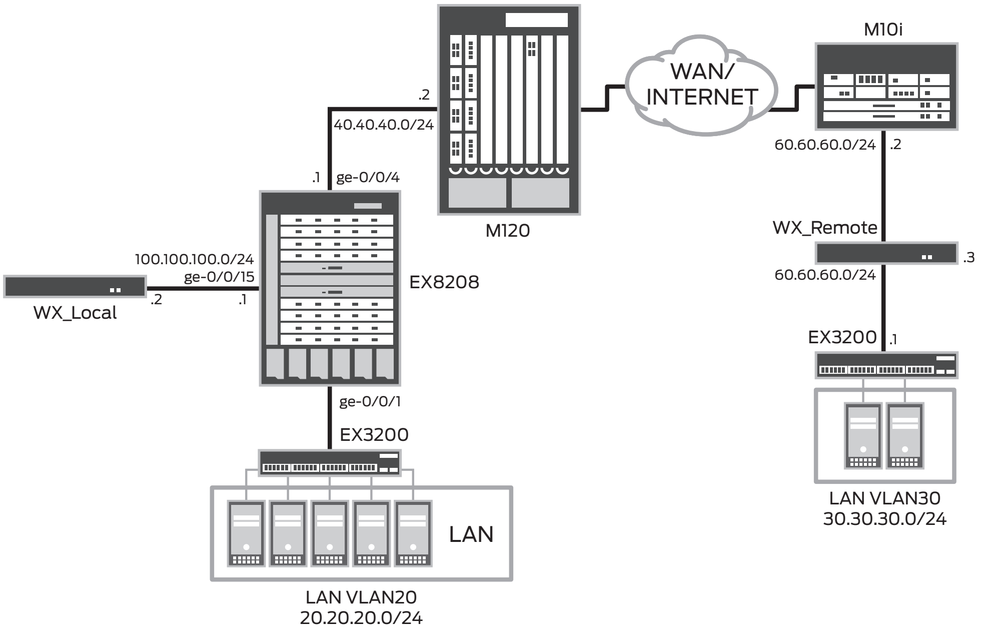

Figure 3: Example of external mode deployment

In this example, the WXC Series platform is directly connected to the Layer 3 interface GE-0/0/15 on the EX8208 switch and assigned the IP address 100.100.100.2/24, which is in the same subnet as the interface GE-0/0/15. The WXC Series device is enabled with packet intercept “external” mode and with a default gateway pointing to the IP address of interface GE-0/0/15 on the EX8208 switch. The local LAN subnet 20.20.20.0/24 will be enabled as the WAN optimization reduction subnets on the WX_Local. After the tunnel with the remote WXC Series device “WX_Remote” is established, WX_Local will receive the WAN optimization remote reduction subnets as 30.30.30.0/24.

To implement filter-based forwarding on an EX Series switch, a routing instance needs to be created and referenced by a firewall filter that matches either the source or the destination IP address or both. The goal of the firewall filter in this example is to match the traffic from the server in the local LAN segment with a source IP address of “20.20.20.0/24” with the server in the remote LAN segment with a destination IP address of “30.30.30.0/24.”

The action on the firewall filter references a routing instance “WXC-VRF” where the packets matching this filter will be redirected for routing. The routing table that will be used to route those matched packets is associated with the routing instance WXC-VRF and named “WXC-VRF.inet.0” in Junos OS software on the EX8208 switch. Any packets not matching this filter will be placed in the default routing table inet.0 for normal Layer 3 routing according to the results of next-hop lookup.  A more complicated firewall filter match criteria can also be implemented if needed—for example, creating a filter with TCP source/destination ports pertaining to certain applications and allowing only that traffic to be forwarded to the WXC Series device.

```
root@EX8208# show firewall
family inet {
    filter wxc-fbf {
        term t1 {
            from {
                source-address {
                    20.20.20.0/25;
                }
                destination-address {
                    30.30.30.0/25;
                }
            }
            then {
                routing-instance WXC-VRF;
            }
        }
        term default {
            then accept;
        }
    }
}
```

A routing instance that is referenced by the firewall filter above is needed, as this will be the routing table where the matched packet will be placed to look for its next hop. In the example below, two default routes with different metrics inside the routing instance are configured. During normal operation, the default route with the smaller metric 5 that points towards the WXC Series takes effect, since anything that matches that filter should be directed to the WXC Series platform for optimization. If anything goes wrong with the WXC Series device such as a power or interface failure, a backup default route with the bigger metric 20 will take over. This default route points matched packets to the next hop of the interface on the edge M120 router so that traffic will bypass the WXC Series device without impacting the existing traffic flows. Implementing an extra backup default route helps improve fault self recovery in certain failure scenarios.

```
root@EX8208# show routing-instances
WXC-VRF {
    instance-type virtual-router;
    routing-options {
        static {
            route 0.0.0.0/0 {
                qualified-next-hop 100.100.100.2 {
                    metric 5;
                }
                qualified-next-hop 40.40.40.2 {
                    metric 20;
                }
            }
        }
    }
}
```

The directed connected interface routes on the EX8208 switch, including the interface connecting to the WXC Series device, need to be integrated into the WXC-VRF routing instance, so that the next hop can be resolved and reachable within this routing instance for those filter-matched and redirected packets. This can be achieved by creating a “WXC-Forwarding” rib group, which contains both routing tables from the global routing instance inet.0 and the WXC-VRF routing instance WXC-VRF.inet.0. “WXC-Forwarding” should then be added within the interface routes under the routing-options stanza.

```
root@EX8208# show routing-options
interface-routes {
    rib-group inet WXC-Forwarding;
}
rib-groups {
    WXC-Forwarding {
        import-rib [ inet.0 WXC-VRF.inet.0 ];
    }
}
```

Checking the routes in the WXC-VRF routing table on the EX8208 switch, the interface ge-0/0/15 connecting to the WXC Series device and its subnet (100.100.100.0/24) are in the routing table of the routing instance WXC-VRF, as well as the default route pointing the matched traffic towards the WXC Series device.

```
root@EX8208# run show route table WXC-VRF
WXC-VRF.inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 2w4d 00:16:53, metric 5
                    > to 100.100.100.2 via vlan.100
20.20.20.0/24      *[Direct/0] 2w4d 00:16:53
                    > via vlan.20
20.20.20.1/32      *[Local/0] 18:00:03
                      Local via vlan.20
40.40.40.0/24      *[Direct /0] 2w4d 00:16:53
                        > via ge-0/0/4.0
40.40.40.1/32      *[Local/0] 18:00:03
                      Local via ge-0/0/4.0
100.100.100.0/24   *[Direct/0] 2w4d 00:16:53
                    > via ge-0/0/15.0
100.100.100.1/32   *[Local/0] 18:00:03
                      Local via ge-0/0/15.0
```

Finally, the firewall filter needs to be applied in the inbound direction on the Layer 3 RVI interface VLAN.20. The physical port ge-0/0/1, which is connected to the access LAN EX4200 switch, is a trunk port in which VLAN 20 resides.

```
root@EX8208# show interfaces
ge-0/0/1 {
    description “To the Access LAN Switch “
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
        }
    }
}
ge-0/0/4 {
    description “To the edge WAN Router”
    unit 0 {
        family inet {
            address 40.40.40.1/24;
        }
    }
}
ge-0/0/15 {
    description “To Off-Path WX”
    unit 0 {
        family inet {
            address 100.100.100.1/24;
        }
    }
}
vlan {
    unit 20 {
        family inet {
            filter {
                input wxc-fbf;
            }
            address 20.20.20.2/24
        }
    }
}
root@EX8208# show vlans
vlan20 {
    vlan-id 20;
    interface {
        ge-0/0/1.0;
    }
    l3-interface vlan.20;
}
```

## Next-Hop Health Tracking in the External Mode Deployment

In some external mode off-path deployment scenarios, the WAN acceleration device is not directly connected to the EX8200 switch but instead through an intermediary Layer 2 switch as shown in Figure 4. This deployment scenario raises concerns, since the EX8200 switch doesn't have the ability to check Layer 3 connectivity to its next-hop WXC Series device under a normal configuration. For example, if the interface on the WXC Series device goes down, the EX8200 switch is not able to detect the loss, since there is a Layer 2 switch between them which is indicating that the next hop in the EX8200 routing table is still healthy and the routes are still valid. Therefore, the EX8200 switch will keep redirecting filter-matched packets to the WXC Series device even though the WXC Series is not accessible, causing traffic black holing.

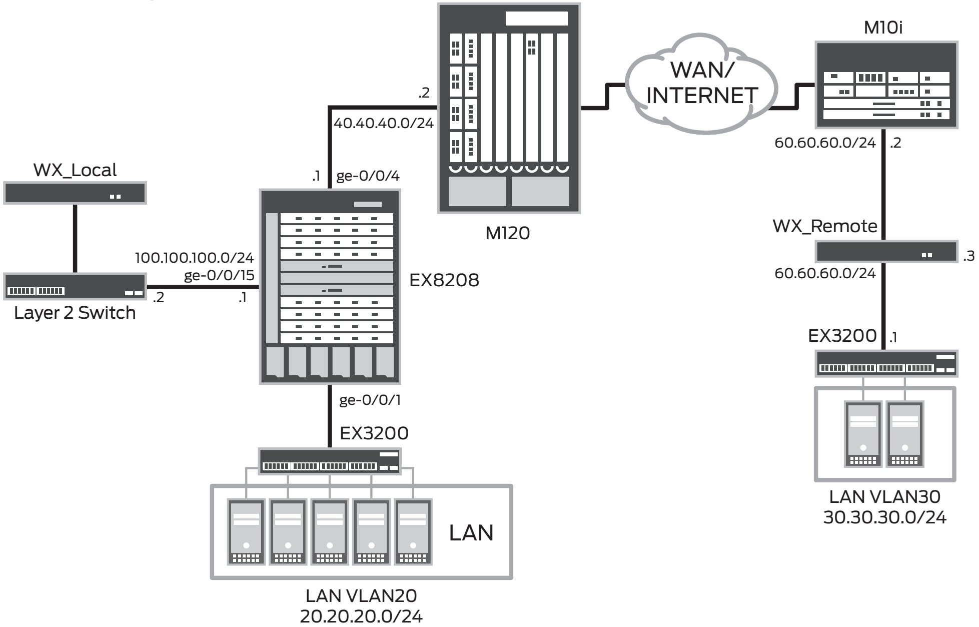

Figure 4: Example of external mode deployment with a Layer 2 switch between WXC Series device and EX8200 switch

The Junos OS real-time performance monitoring (RPM) feature on EX Series Ethernet Switches, along with Junos Script, provides a dynamic solution for checking the health of the next hop (WXC Series device) using RPM Internet Control Message Protocol ping (ICMP-Ping) probes, and modifying the filter-based forwarding configuration on the EX8200 switch so that it will redirect packets properly once it detects that the WXC Series device is not responding to the Layer 3 ICMP echo.

RPM enables network administrators to monitor and assess network performance in real time based on the jitter, delay, and packet loss experienced on the network. RPM can also be used to measure metrics such as round-trip delays and unanswered echo requests. To achieve this, RPM exchanges a set of probes with other IP hosts in the network for monitoring and connectivity tracking purposes. These probes are sent from a source node (an EX8208 switch in the following example) to other destination devices (a WXC Series platform in this case) that require tracking. Data such as transit delay and jitter can be collected from these probes, and can be used to provide an approximation of the delay and jitter experienced by live traffic in the network as well as the end-to-end connectivity health between these two RPM end hosts. RPM has been supported on the EX3200 and EX4200 switches since the Junos OS 9.3 release, and on the EX8200 switches starting with Junos OS 10.1.

RPM on the EX8208 switch enables the probe ICMP-Ping to track Layer 3 connectivity to the WXC Series device with the host IP address 100.100.100.2. If ICMP-echo-replies are not received from the WXC Series device, a system log message called "ping_test_failed" will be generated on the EX8208. That in turn will trigger Junos Script to modify the filter-based forwarding feature to deactivate the redirect function in FBF so that all packets will go through the EX8208 normal global routing table without passing through the WXC Series device to avoid traffic black holing. Once the network administrator restores Layer 3 connectivity between the WXC Series device and the EX8208 switch by replacing the broken cables, and the ICMP-echo-replies from the WXC Series are received by the EX8208 switch, a "ping_test_completed" system log message will be generated and Junos Script will automatically restore the original FBF configuration so that filter-matched packets will again be redirected to the WXC Series device for optimization.

The following configuration shows how the RPM feature is configured on the EX8208. The RPM test is named “wxc-ping” and the probe type is “icmp-ping.” The target IP host is the WXC Series device which has IP address 100.100.100.2. In a single RPM test, three probes will be sent out at one-second intervals, with the interval between each RPM test being 10 seconds. All parameters are user configurable.

```
root@EX8208# show services
rpm {
    probe wxc {
        test wxc-ping {
            probe-type icmp-ping;
            target address 100.100.100.2;
            probe-count 3;
            probe-interval 1;
            test-interval 10;
        }
    }
}
```

An example script named “WXC-Healthcheck.slax” is shown below. This script will deactivate the traffic redirecting term in the firewall filter once it is triggered by the “ping_test_failed” system log, and will reactivate the traffic redirecting term in the firewall filter if it is triggered by the “ping_test_completed” system log later.

The script “WXC-Healthcheck.slax” needs to be FTPed to the location “/config/db/scripts/event” on the EX8208 switch via logging as root.

```
/*
*    NAME: WXC-Healthcheck.slax
*    PURPOSE: Created to detect the state of an off-path stream link
*               When DOWN is detected the traffic redirecting term in
*               the Firewall Filter is deactivated.
*               When UP is detected the traffic redirecting term in
*               the Firewall Filter is activated.
*
*
*    CREATED: 10/04/09
*    BY: Ting Zou
*/

version 1.0;
ns junos = “http://xml.juniper.net/junos/*/junos”;
ns xnm = “http://xml.juniper.net/xnm/1.1/xnm”;
ns jcs = “http://xml.juniper.net/junos/commit-scripts/1.0”;
import “../import/junos.xsl”;

/*
*
*   Parms that are passed in from the configuration or cli
*
*/
var $arguments = {
    <argument> {
        <name> “filter”;
        <description> “Firewall Filter Name”;
    }
    <argument> {
        <name> “term”;
        <description> “Firewall Filter Term Name”;
    }
    <argument> {
        <name> “action”;
        <description> “What to do (active/inactive)”;
    }
}
param $filter;
param $term;
param $action;

match / {


    var $check-route = jcs:invoke (“get-configuration”);
    var $tt = $check-route/firewall/family/inet/filter[name=$filter]/
term[name=$term];


/*
*
*   conditional statement to check if config is different from action. *   ie.
config is inactive and action active
*
*/

    if ( (not($tt[@inactive]) and $action == “inactive”) || ($tt[@inactive] and
$action == “active”)) {

    if ($filter and $term and $action) {

/*
*
*   Open a connection with mgd. use config private
*
*/
        var $con = jcs:open();
        var $open = <open-configuration> {
                       <private>;
        }
        var $result = jcs:execute( $con, $open );
        if (not($con)) {
            call write-output($level = “user.error”, $id = “ERROR: “, $type =
“WXC-Healthcheck.slax[Error]: “, $where = “b”, $message = “Not able to connect to
local mgd”);
        }
/*
*
*   XML structure for activating/deactivating the primary firewall
*   filter
*
*/
        var $xml = {
            <configuration> {
                <firewall> {
                    <family> {
                        <inet> {
                            <filter> {
                                <name> $filter;
                                if ($action == “active”) {
                                    <term active=”active”> {
                                        <name> $term;
                                    }
                                }
                               if ($action == “inactive”) {
                                    <term inactive=”inactive”> {
                                        <name> $term;
                                    }

                               }
                            }
                        }
                    }
                 }
             }
        }

/*
*
*   Use load-configuration template defined in junos.xsl to load and
*   commit the configuration.
*
*/

var $private-results = jcs:execute( $con, $open );
var $load-configuration = <load-configuration> {
                            copy-of $xml;
                          }
var $results = jcs:execute( $con, $load-configuration );
var $commit-configuration = <commit-configuration>;
var $commit-results = jcs:execute( $con, $commit-configuration );

/*
*
*   Use load-configuration template defined in junos.xsl to load and
*   commit the configuration
*
*/

/*
*
*   Close the mgd connection
*
*/

    var $close-private = <close-configuration>;
    var $close-configuration-results = jcs:execute( $con, $close-private );
    var $close-results = jcs:close( $con);

/*
*
*   Check the results and process them
*
*/

        for-each ($results//xnm:warning) {
            if (not(contains(message,”statement not found:”))) {
            call write-output($level = “user.warning”, $id = “WARNING: “, $type =
“WXC-Healthcheck.slax[Warning]: “, $where = “b”, $message = message);
            }
        }
        if ($results//xnm:error) {
            for-each ($results//xnm:error) {
                call write-output($level = “user.error”, $id = “ERROR: “, $type =
“WXC-Healthcheck.slax[Error]: “, $where = “b”, $message = message);
            }
        }
        else {
            call write-output($level = “user.info”, $id = “INFO: “, $type = “WXC-
Healthcheck.slax[Success]: “, $where = “b”, $message = “Successfully processed
the FF”);
        }

    }
    else {
        call write-output($where = “s”, $message = “Missing Script Arguments”);
    }
  }
}

/*
*   NAME: WRITE-OUTPUT
*   PURPOSE: Writes a message to standard and/or the syslog depending * *
on the parms
*   passed to it.
*   CALLED: Called after the XML changes have been submitted to MGD * * *
process.
*
*
*   PARMS PASSED:
*        $level = The syslog message level
*        $id = Standout message ID
*        $type = The string identifying the type of message
*        $message = The string to print out
*        $where = Used to specify if “b”oth STDOUT and SYSLOG
*                  should be written to or only “S”TDOUT
*
*/
template write-output($level = “user.info”, $id = “INFO: “, $type = “ “, $where =
“b”, $message) {
     var $stdmessage = $id _ $message;

     if ($where == “s”) {
        expr jcs:output($stdmessage);
     }
     else {
         expr jcs:syslog($level, $type, $message);

         expr jcs:output($stdmessage);
     }

}
```

Custom event policies need to be configured on the EX8208 so that a Junos OS process called event process (eventd) will be listening for specific events such as system logging messages and creating log files, invoking Junos OS commands, or invoking event scripts in response. The event policy “rpm_down” listens for the system logging message “PING_TEST_FAILED” with attributes matched to the RPM probe test owner “wxc” and test name “wxc_ping” in 10 second intervals and triggers the Junos Script “WXC-Healthcheck.slax” once it sees the “PING_TEST_FAILED” message indicating that the EX8208 has lost Layer 3 connectivity to the WXC Series device. With the additional configuration “set system syslog file messages any any” on the EX Series switch, those system logging messages can be found in the file /var/log/messages.

```
root@EX8208# show event-options
policy rpm_down {
    events PING_TEST_FAILED;
    within 10 {
        trigger on 1;
    }
attributes-match {
                      PING_TEST_FAILED.test-owner matches “^wxc$”;
     PING_TEST_FAILED.test-name matches “^wxc_ping$”;
}
    then {
        event-script WXC-Healthcheck.slax {
            arguments {
                filter wxc-fbf;
                term t1;
                action inactive;
            }
        }
    }
}
event-script {
    file WXC-Healthcheck.slax;
}
```

The Junos Script needs to parse several parameters. The argument “filter” shows the targeted firewall filter that the script needs to modify. The argument “term” identifies which term within the filter needs to be edited. The argument “action” tells the script to deactivate the term in the firewall filter or reactivate the term. The following shows the running firewall filter configuration on the EX Series switch after the Junos Script deactivates the traffic redirecting term in the FBF so that packets won’t pass through the WXC Series device anymore.

```
root@EX8208# show firewall
family inet {
    filter wxc-fbf {
        inactive: term t1 {
            from {
                source-address {
                    20.20.20.0/25;
                }
                destination-address {
                    30.30.30.0/25;
                }
            }
            then {
                routing-instance WXC-VRF;
            }
        }
        term default {
            then accept;
        }
    }
}
```

The event policy “rpm_up” listens for the system logging message “PING_TEST_COMPLETED” in 20 second intervals, and it triggers the Junos Script “WXC-Healthcheck.slax” once it sees the “PING_TEST_COMPLETED” message indicating that Layer 3 connectivity between the EX8208 and WXC Series device has been restored. With the parsed parameters, the Junos Script will reactivate the traffic redirecting term in the FBF so that the packets will again pass through the WXC Series device for optimization.

```
root@EX8208# show event-options
policy rpm_up {
    events PING_TEST_COMPLETED;
    within 20 {
        trigger on 1;
}
attributes-match {
                      PING_TEST_COMPLETED.test-owner matches “^wxc$”;
     PING_TEST_COMPLETED.test-name matches “^wxc_ping$”;
}
    then {
        event-script WXC-Healthcheck.slax {
            arguments {
                filter wxc-fbf;
                term t1;
                action active;
            }
        }
    }
}
```

## Summary

Juniper Networks EX Series Ethernet Switches provide a high-performance network infrastructure solution that can be integrated with the WXC Series WAN application acceleration devices. With the functionality and features of Junos OS and the powerful capabilities of Junos Script, this solution enables dynamic self recovery for the most common failure scenarios.

## About Juniper Networks

Juniper Networks, Inc. is the leader in high-performance networking. Juniper offers a high-performance network infrastructure that creates a responsive and trusted environment for accelerating the deployment of services and applications over a single network. This fuels high-performance businesses. Additional information can be found at **www.juniper.net**.

**Corporate and Sales Headquarters**
Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA
Phone: 888.JUNIPER (888.586.4737)
or 408.745.2000
Fax: 408.745.2100
www.juniper.net

**APAC Headquarters**
Juniper Networks (Hong Kong)
26/F, Cityplaza One
1111 King's Road
Taikoo Shing, Hong Kong
Phone: 852.2332.3636
Fax: 852.2574.7803

**EMEA Headquarters**
Juniper Networks Ireland
Airside Business Park
Swords, County Dublin, Ireland
Phone: 35.31.8903.600
EMEA Sales: 00800.4586.4737
Fax: 35.31.8903.601

To purchase Juniper Networks solutions, please contact your Juniper Networks representative at 1-866-298-6428 or authorized reseller.

8010072-001-EN Mar 2010

♻ Printed on recycled paper