|  |
| --- |
| Mellanox Data Center  Certification Test Plan |
| <Customer Name> |

|  |
| --- |
| Rev 1.0 |
| For |
| Release Number |
| (Switch Type SNxx) |
| **Mellanox Contact Information**  Name:  Email:  Phone number: |
| **Customer Contact Information**  Name:  Email:  Phone number: |



Table of Contents

[Document Revision History 8](#_Toc493355908)

[About this Manual 9](#_Toc493355909)

[1. Introduction and Objectives 10](#_Toc493355910)

[1.1 POC Objectives 10](#_Toc493355911)

[1.2 Key Deliverables 11](#_Toc493355912)

[1.3 POC Process 11](#_Toc493355913)

[2. Requirements 12](#_Toc493355914)

[2.1 Business Requirements 12](#_Toc493355915)

[2.2 Technical Requirements 12](#_Toc493355916)

[2.3 Business Constraints (if applicable) 13](#_Toc493355917)

[3. Design Proposal 14](#_Toc493355918)

[3.1 Design Diagram 14](#_Toc493355919)

[3.2 Design Description 14](#_Toc493355920)

[4. Bill of Materials 15](#_Toc493355921)

[5. Tolly Tests 17](#_Toc493355922)

[5.1 Per Port Fairness Test 17](#_Toc493355923)

[5.2 Latency Test with Servers 18](#_Toc493355924)

[5.3 Latency Test using Ixia 19](#_Toc493355925)

[5.4 Line Rate Forwarding Performance 20](#_Toc493355926)

[5.5 Microburst Absorption Test 20](#_Toc493355927)

[6. Hardware Performance 22](#_Toc493355928)

[6.1 Buffers 22](#_Toc493355929)

[6.2 VoQ 22](#_Toc493355930)

[6.3 Buffer test for 4x25GE-100GE 23](#_Toc493355931)

[6.4 Traffic congestion 24](#_Toc493355932)

[6.5 Mirror the packets marked with TCP port number 25](#_Toc493355933)

[6.6 Tests on Whole System Reboot 26](#_Toc493355934)

[6.7 Tests on Power Supply Redundancy 27](#_Toc493355935)

[6.8 Tests on Fan Redundancy 28](#_Toc493355936)

[6.9 Power Consumption Test 28](#_Toc493355937)

[7. Basic Function Test 30](#_Toc493355938)

[7.1 Telnet 30](#_Toc493355939)

[7.2 SSH 30](#_Toc493355940)

[7.3 Tacacs+ 31](#_Toc493355941)

[7.4 SNMPv2/v3 31](#_Toc493355942)

[7.5 LLDP 32](#_Toc493355943)

[7.6 Syslog 32](#_Toc493355944)

[7.7 NTP 32](#_Toc493355945)

[7.8 Port Mirroring 33](#_Toc493355946)

[7.9 Out-of-Band Management VRF 33](#_Toc493355947)

[8. sFlow 35](#_Toc493355948)

[8.1 sFlow 35](#_Toc493355949)

[9. Layer 2 Basic Test 36](#_Toc493355950)

[9.1 MAC-Table Scaling 36](#_Toc493355951)

[9.2 MAC Address Learning Rate 36](#_Toc493355952)

[9.3 LAG with Max Member 38](#_Toc493355953)

[9.4 STP 38](#_Toc493355954)

[9.5 Storm Suppression 39](#_Toc493355955)

[9.6 Max number of LAGs 39](#_Toc493355956)

[9.7 Mac flapping 40](#_Toc493355957)

[9.8 STP edge port 41](#_Toc493355958)

[9.9 STP bpdu-guard 42](#_Toc493355959)

[9.10 Hierarchical HASH 42](#_Toc493355960)

[9.11 ACL Test 43](#_Toc493355961)

[10. MLAG 46](#_Toc493355962)

[10.1 MLAG Basic Test 46](#_Toc493355963)

[10.2 MLAG Basic Test with IGMP 47](#_Toc493355964)

[10.3 MLAG Split 48](#_Toc493355965)

[10.4 MLAG with MAGP 49](#_Toc493355966)

[10.5 MLAG Convergence – MLAG Link Failover 50](#_Toc493355967)

[10.6 MLAG Convergence – Reboot 51](#_Toc493355968)

[10.7 MLAG Convergence – IPL 52](#_Toc493355969)

[11. Layer 3 & Routing 53](#_Toc493355970)

[11.1 ARP-Table Scaling 53](#_Toc493355971)

[11.2 ARP Learning Rate 53](#_Toc493355972)

[11.3 ECMP 55](#_Toc493355973)

[11.4 ND-table scaling 55](#_Toc493355974)

[11.5 ND learning rate 56](#_Toc493355975)

[11.6 Arp migration 57](#_Toc493355976)

[11.7 Aging of MAC in ARP table 58](#_Toc493355977)

[11.8 Max number of secondary IP addresses under a single VLAN IP interface 58](#_Toc493355978)

[11.9 ECMP Hash 59](#_Toc493355979)

[11.10 L3 ACL Functionality 60](#_Toc493355980)

[11.11 OSPF Convergence 61](#_Toc493355981)

[11.12 Number of ospf neighbor 61](#_Toc493355982)

[11.13 Route metric test 62](#_Toc493355983)

[11.14 OSPF scaling in a single area and convergence 63](#_Toc493355984)

[11.15 Multiple OSPF neighbor’s flapping 64](#_Toc493355985)

[11.16 Performance for OSPF running in Multi-VRFs 65](#_Toc493355986)

[11.17 OSPF routing flapping 65](#_Toc493355987)

[11.18 OSPF convergence under ECMP 66](#_Toc493355988)

[11.19 BGP/BGPv6 RIB/FIB size 67](#_Toc493355989)

[11.20 BGP/BGPv6 routes learning-rate 68](#_Toc493355990)

[11.21 BGP update-interval test 69](#_Toc493355991)

[11.22 BGP/BGPv6 routing policy 70](#_Toc493355992)

[11.23 BGP/BGPv6 Route Aggregation 71](#_Toc493355993)

[11.24 BGP Route Flapping 72](#_Toc493355994)

[11.25 BGP Convergence 74](#_Toc493355995)

[11.26 Number of BGP Neighbors 75](#_Toc493355996)

[11.27 BGP Unnumbered 75](#_Toc493355997)

[12. IGMP & PIM-SM 77](#_Toc493355998)

[13. QOS 78](#_Toc493355999)

[13.1 Services Classified by VLAN 78](#_Toc493356000)

[13.2 Services Classified with L4 Port Number 78](#_Toc493356001)

[13.3 Strict Priority Queue 79](#_Toc493356002)

[13.4 Strict Priority and WRR Mixed 80](#_Toc493356003)

[13.5 Ingress Rate-Limit 81](#_Toc493356004)

[13.6 Egress Rate-Limit 81](#_Toc493356005)

[13.7 Granularity and Accuracy of Rate-Limit 82](#_Toc493356006)

[14. VXLAN 83](#_Toc493356007)

[14.1 HW VTEP - VXLAN with NSX 83](#_Toc493356008)

[14.2 VXLAN with LNV 84](#_Toc493356009)

[14.3 VXLAN and EVPN 85](#_Toc493356010)

[14.4 VXLAN Scaling 86](#_Toc493356011)

[14.5 VTEP Scaling 87](#_Toc493356012)

[14.6 VXLAN Asymmetric Routing 88](#_Toc493356013)

[14.7 VXLAN Symmetrical Routing 89](#_Toc493356014)

[15. Solutions – Leaf & Spine 90](#_Toc493356015)

[15.1 Leaf & Spine with ECMP with BGP & MLAG 90](#_Toc493356016)

[15.2 Leaf & Spine with ECMP Resilient Hashing 90](#_Toc493356017)

[15.3 Leaf & Spine – ECMP Convergence 91](#_Toc493356018)

[16. Solutions – TAP Aggregation (OpenFlow) MLNX-OS 93](#_Toc493356019)

[17. Solutions – Routing on Host Cumulus 94](#_Toc493356020)

[17.1 Routing on Host with BGP 94](#_Toc493356021)

[17.2 Routing on host with OSPF 95](#_Toc493356022)

[18. SOLUTIONS – Storage Solution (with & without RoCE) 98](#_Toc493356023)

[18.1 Non-RoCE 98](#_Toc493356024)

[18.2 Lossless RoCE w/ QoS 98](#_Toc493356025)

[18.3 Resilient RoCE 99](#_Toc493356026)

[19. Solutions – HFT PIM-SM 101](#_Toc493356027)

[20. Appendix 102](#_Toc493356028)

[20.1 About Mellanox Switches 102](#_Toc493356029)

[20.1.1 Designed for Open Networking 102](#_Toc493356030)

[20.1.2 Differentiated ASIC 102](#_Toc493356031)

[20.1.3 End-2-End Solution 102](#_Toc493356032)

[20.2 Mellanox Spectrum Advantages 103](#_Toc493356033)

[20.2.1 Latency 103](#_Toc493356034)

[20.2.2 Performance 103](#_Toc493356035)

[20.2.3 Buffering 103](#_Toc493356036)

[20.2.4 Low Power Consumption 103](#_Toc493356037)

[20.2.5 Traffic Visibility / Telemetry 103](#_Toc493356038)

[20.2.6 Best Fit for RoCE (if applicable) 104](#_Toc493356039)

[20.3 Sample Configurations 105](#_Toc493356040)

[21. Heading 1 [Just a sample of style usage – you can delete] 106](#_Toc493356041)

[21.1 Heading 2 106](#_Toc493356042)

[21.1.1 Heading 3 106](#_Toc493356043)

List of Figures

[Figure 1: Caption 73](#_Toc492284014)

List of Tables

[Table 1: Document Revision History 7](#_Toc492284015)

[Table 2. Sample Bill of Materials 14](#_Toc492284016)

[Table 3: Table Caption 72](#_Toc492284017)

[Table 4: Table Two Columns 73](#_Toc492284018)

Document Revision History

Table 1: Document Revision History

| Revision | Date | Description |
| --- | --- | --- |
| 1.0 | TBD | Initial release. |

About this Manual

This document is intended to offer Mellanox customers ready-made POC test cases for all the Mellanox switch features and solutions supported.

Document Conventions

The following lists conventions used in this document.

|  |  |
| --- | --- |
|  | NOTE: Identifies important information that contains helpful suggestions. |

|  |  |
| --- | --- |
|  | CAUTION: Alerts you to the risk of personal injury, system damage, or loss of data. |

|  |  |
| --- | --- |
|  | WARNING: Warns you that failure to take or avoid a specific action might result in personal injury or a malfunction of the hardware or software. Be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents before you work on any equipment. |

Terms and Abbreviations

The following terms and abbreviations are used in this document.

| Definition/Abbreviation | Description |
| --- | --- |
| DUT | Device Under Test |
| Ixia | Tester | Analyzer | Traffic generator test equipment or person operating it. |
| POC | Proof of concept |

# Introduction and Objectives

As your business grows and changes, so do the demands on your network infrastructure. Having to plan for a data center which is highly scalable and cost-effective is key to ensuring your competitive edge in the landscape. Mellanox can help address these challenges with its hardware and software product portfolio. A Proof of Concept (POC) of the Mellanox portfolio can accurately assess how your data center needs can be met by:

* Helping validate key features and functionality of a preliminary or conceptual solution.
* Reducing the risk to your project by providing engineering assistance for the POC.
* Speed up your testing time, by eliminating the potentially costly and time-consuming process of defining test cases, processes, etc.
* Help educate and expand the knowledgebase of your staff, helping them to deploy, operate, and scale your infrastructure going forward.

## POC Objectives

The objective of this document is to define:

* Timeline for a successful POC.
* Detailed network architecture and design of customer requirements in form of a lab-based POC.
* POC test-bed hardware equipment to be deployed within a lab environment at the customer site or at Mellanox labs.
* POC test-bed software which will be installed on the test-bed hardware equipment.
* Key features and functionality required for the defined network architecture with inter-operability testing, performance testing, HW tests & application specific tests as needed.
* High level plan which includes timelines and milestones for developing, testing, and deploying the POC test-bed components.
* Support for the process in the event the customer should require troubleshooting assistance.
* A review of the customer’s test goals (including pass/fail criteria) in each test case of the POC.
* Final pass/fail criteria and explanations of any workarounds if needed.

The end goal is to execute the test plan and to create a report of the test results.

## Key Deliverables

The key deliverables of POC testing consists of:

* Details of the physical setup and software versions of the lab environment
* Configuration details of the setup
* POC testing results mapped against the pass/fail criteria
* Proposed network architecture
* Workarounds – if needed
* Roadmap dates and details – if needed
* Ansible playbooks – if needed

## POC Process

In order to meet the above deliverables, the following process is proposed:

**Step – 1: Gathering requirements.**

A detailed technical meeting between Mellanox and Customer Architects to define the following:

* High level network architecture/design along with application usage.
* Defining required protocols, port speeds, number of ports, etc.
* Choosing the operating system.
* Closing the test plan & corresponding pass/fail criteria.
* Defining the minimum POC design.
* Determining key dates:
* When can the POC start?
* By when does the customer need the results?
* Any other important dates to be noted.
* Decide on the location of POC, along with assigned owners from both the customer and Mellanox sides.
* Assign the POC owner (who will lead the testing)

**Step – 2: Test Execution.**

It is expected that during the POC testing phase, the customer and Mellanox contacts will meet at regular intervals to ensure that the overall objective is being achieved.

**Step – 3: Test Report.**

On execution completion of the test plan, a test report will be generated along with any workarounds for failed cases or roadmaps if applicable.

**Step – 4: Final Review.**

On completion of the test report, the account team will meet with the customer to review the goals against the expected results, exceptions, work arounds, etc.

# Requirements

## Business Requirements

[Sample]

Company-A will be conducting a SDN Proof of Concept to select a macro-segmentation and software orchestration solution for its San Jose and Hong Kong data centers. These will be new software defined data centers, which will need the latest Layer-2 and Layer-3 protocols along with software based service chaining. The objective is to segment mission-critical applications, provide better visibility, and automatically provision new applications. Company-A needs to test the required protocols and solutions from various vendors for its final vendor selection.

The POC needs to follow the dates below:

* Requirement lock: 08/15/2017
* Procurement of HW: 08/30/2017
* Test execution and final report: 9/30/2017
* Discussion with vendors: 10/15/2017
* Final selection: 10/30/2017

Vendors will need to run the POC in their own labs and work with Company-A to ensure proper execution of tests. Once the tests are completed, a final closure meeting will review the results in detail and explicitly write down any work arounds or any roadmap items with dates.

## Technical Requirements

[Sample]

This document outlines a test plan that will help confirm which vendor platform satisfies Company-A’s requirements to provide advanced, feature-rich, high availability, and resilient security policies and service protection for Company-A’s network and application infrastructure. Successful completion of this test plan implies that the reviewers are comfortable that the products were tested to their full extent against the outlined test criteria and scoring matrix specified. At the high level, the following protocols and solutions will need to be tested:

* Port speed tests – to ensure line rate performance
* Hardware system tests – to ensure HW durability and stability
* MLAG protocol with failover numbers
* VXLAN with EVPN working with MLAG or Type – 1 EVPN support
* EVPN routing
* Controller based service chaining
* BGP routing tests – basic, failovers and performance tests
* Ability to show load balancing in the fabric
* Ability to support multi-vendor OS – open networking
* Various scale matrix (for L3)
* Interoperability with Juniper (with L2 and L3 protocols)

A detailed test plan should be made (and reviewed with Company-A) based on above requirements and each test case should have pass/fail or reporting criteria. Test diagrams and configurations should be attached in the appendix.

Any third party software should also be mentioned in the appendix.

## Business Constraints (if applicable)

[Sample]

The test time lines are rigid and the POC should be completed on time. Along with time constraints, there are also the following constraints:

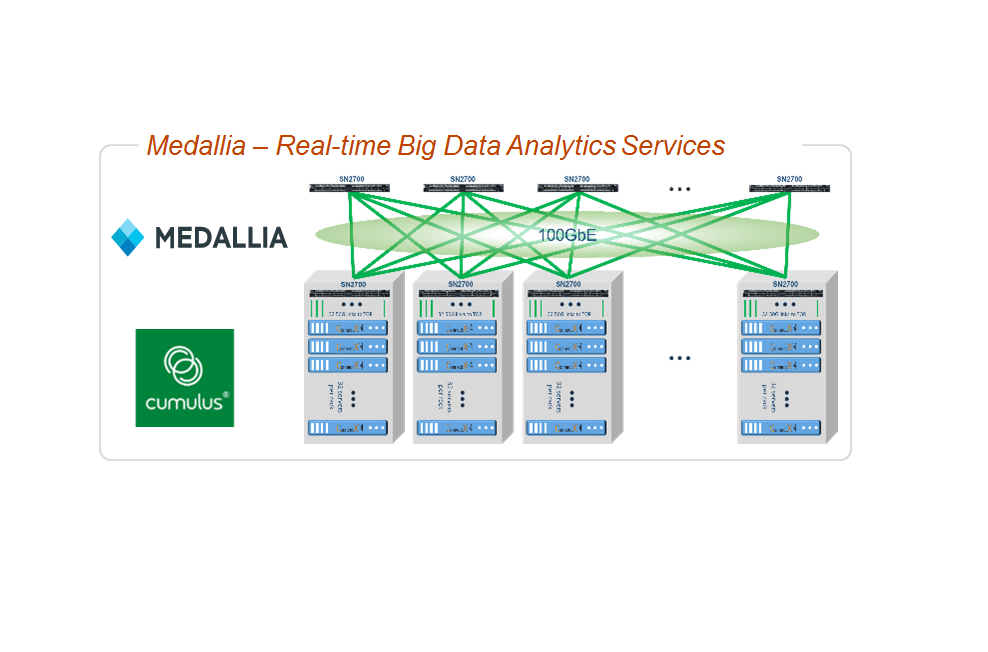
* Company-A will not provide any detailed inputs on network requirements, vendors will have to come up with their own designs to prove their functionality.
* All tests should be executed at the vendor’s lab and Company-A will not pay for any POC equipment.
* Once a vendor is selected for final bidding, there may be requirements for onsite test resources at Company-A’s Palo Alto location for a month duration.

# Design Proposal

## Design Diagram

[Sample]

The High Level Design (HLD) gives an overview of the proposed network topology.



## Design Description

[Sample]

The design is based on ROH (Routing on Host) technology and is capable of scaling up to x racks, with a 2:1 oversubscription ratio. This design will need to have kernel level access on hosts, so that BGP can run on the hosts. IP unnumbered is used in this design, so there will be no IP address in P2P links. For configuration, Ansible playbooks are recommended.

<Start documenting the design description here>

# Bill of Materials

Following quantities per product is needed for this POC (or for this deployment).

<Add what is applicable, below is a sample>

Table 2. Sample Bill of Materials

| Product Sample Photo | Qty | Mellanox  Part Number | Description | Technical Description | Details |
| --- | --- | --- | --- | --- | --- |
|  |  | MSN2100-CB2F | SN2100 Switch | Spectrum based half switch | [Product Spec](http://www.mellanox.com/related-docs/prod_eth_switches/PB_SN2100.pdf) |
|  |  | MSN2700-BS2F | SN2700 Switch | Spectrum based full switch | [Product Spec](http://www.mellanox.com/related-docs/prod_eth_switches/PB_SN2700.pdf) |
|  |  | AS4610 | Edgecore AS4610 | Out of band mgmt switch |  |
|  |  | MC2210411-SR4L | Optical module | 40Gb/s QSFP+ | [Product Spec](http://www.mellanox.com/related-docs/prod_cables/PB_40GbE_QSFP_Lite_Optical_Transceiver.pdf) |
|  |  | MC2609130-003 | Splitter cable | 40Gb/s to 4x 10Gb/s QSFP+ to 4x SFP+ | [Product Spec](http://www.mellanox.com/related-docs/prod_cables/PB_Hybrid_Passive_QSFP_to_4X_SFP+_Cables.pdf) |
|  |  | MCP1600-C001 | 100 GbE cable | 100Gb/s QSFP28 | [Product Spec](http://www.mellanox.com/related-docs/prod_cables/PB_100Gbs_Passive_Copper_Cables.pdf) |
|  |  | MFA1A00-C030 | 100 GbE Optics Cable | 100Gb/s QSFP28 | [Product Spec](http://www.mellanox.com/related-docs/prod_cables/PB_100Gbs_VCSEL-Based_Active_Optical_Cables.pdf) |
|  |  | MC2210130-001 | 40 GbE Cable | 40Gb/s QSFP+ | [Product Spec](http://www.mellanox.com/related-docs/prod_cables/PB_40GbE_Passive_Copper_Cables.pdf) |

# Tolly Tests

The Tolly Group ([www.tolly.com](http://www.tolly.com)) is a leading global provider of testing and third-party validation and certification services to the Information Technology industry. This chapter outlines specific tests based on the Tolly Group specifications.

## Per Port Fairness Test

|  |  |
| --- | --- |
| Objectives of Tests | Test the fairness for traffic |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Connect the ports of 100GbE on the DUTs as shown in above figure. 2. Start sending traffic from Tport2, 3 and 4, destined towards Tport1. 3. Analyze on Tport1 that we get 33.3% of traffic of each type (destined from Tport2, 3 and 4). 4. Try the test with different sets of ports. 5. Test same setup with more ports – if available. |
| Expected Result | 1. Verify that the traffic received at Tport1 is always equal from Tport2 and 3, irrespective of the sets of ports are used. 2. Plot the graph (in pie chart format) to show fair distribution of traffic. |
| Descriptions |  |
| Results |  |

## Latency Test with Servers

|  |  |
| --- | --- |
| Objectives of Tests | Test the latency |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Setup the servers with following steps on Intel NICs:   **# tar zxf parallel\_studio\_xe\_2016\_update2.tgz**  **# sudo -s**  **# cd parallel\_studio\_xe\_2016\_update2/**  **# ./install.sh**  **Install procedures:**   1. Press "Enter" key to continue or "q" to quit: 2. 1. Skip missing optional prerequisites [default] 3. Type 'accept' to continue or 'decline' to go back to the previous menu: accept 4. 2. I want to evaluate Intel(R) Parallel Studio XE 2016 Update 2 Cluster Edition for Linux\* or activate later 5. 2. No, I don't want to participate in the Intel® Software Improvement Program at this time. 6. 1. Finish configuring installation target [default] 7. Please type a selection or press "Enter" to accept default choice [1]: 8. Would you like to install to this directory? ( Yes/No ) [ Yes ] : 9. 1. Continue the installation [default] 10. 1. Skip missing optional prerequisites [default] 11. Press "Enter" key to quit:   **Finally, the Intel MPI is installed.**   1. Setup the servers with following steps on Intel NICs:   **# ifconfig eno33557248 up**  **# ifconfig eno33557248 2.2.2.5 netmask 255.255.255.0**  **# ifconfig eno33557248 mtu 9000**  **# vi /root/mpd.hosts**  mttw5.mlnx  mttw6.mlnx  **# sudo vi /etc/hosts**  2.2.2.5 mttw5.mlnx  2.2.2.6 mttw6.mlnx  **# ssh-ketygen**  **…**  **# ssh-copy-id root@mttw6.mlnx**  Now you can ssh login to remote server with typing password  **# systemctl stop firewalld.service**  **# tuned-adm profile latency-performance**  **# ifconfig eno33557248 up**  **# ifconfig eno33557248 2.2.2.6 netmask 255.255.255.0**  **# ifconfig eno33557248 mtu 9000**  **# vi /root/mpd.hosts**  mttw5.mlnx  mttw6.mlnx  **# sudo vi /etc/hosts**  2.2.2.5 mttw5.mlnx  2.2.2.6 mttw6.mlnx  **# ssh-ketygen**  **…**  **# ssh-copy-id root@mttw5.mlnx**  Now you can ssh login to remote server with typing password  **# systemctl stop firewalld.service**  **# tuned-adm profile latency-performance**   1. Run TCP pinp latency test with following scripts and command:   **# vi runTCPintelMPI.sh** |
| Expected Result | 1. Verify that the traffic received at Tport1 is always equal from Tport2 and 3, irrespective of the sets of ports are used. |
| Descriptions |  |
| Results |  |

## Latency Test using Ixia

|  |  |
| --- | --- |
| Objectives of Tests | Test latency using IxScriptmate |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Configure RFC 2544 test with Ixia. 2. Run the test for various packet sizes and record the latency. |
| Expected Results | 1. Record the latency numbers. 2. Plot the graph for various packet sizes. |
| Descriptions |  |
| Results |  |

## Line Rate Forwarding Performance

|  |  |
| --- | --- |
| Objectives of Tests | Test the fully populated switch and calculate the total number of packets that the switch can forward |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Connect the pairs of ports to Tx and Rx pairs of Ixia. 2. Configure Ixia to send bi-directional traffic per port pair for various packet sizes. 3. Record the performance in terms of total number of packets that the switch can forward. |
| Expected Results | 1. Traffic can be forwarded normally for line rate for every packet size. 2. Plot the graph for various packet sizes. |
| Descriptions |  |
| Results |  |

## Microburst Absorption Test

|  |  |
| --- | --- |
| Objectives of Tests | Microburst test |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Send 100% line-rate traffic through DUT from Tport1 to Tport3. 2. Send burst traffic 100,000 packets through DUT from Tport2 to Tport3, which makes Tport3 congested. 3. Configure the required buffers per port, and based on that observe the behavior. Various RoCE profiles can be tested. 4. Test 512bytes/1518bytes/9216bytes, buffer size=(total number of burst packets - number of packet lost)x packet size, record the packet received and calculate the drops. |
| Expected Result | 1. Record the size of buffer, along with real traffic recorded per port. Plot a graph to show microburst absorption. |
| Descriptions |  |
| Results |  |

# Hardware Performance

## Buffers

|  |  |
| --- | --- |
| Objectives of Tests | buffer |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | Senario1   1. Tester sends 100% line-rate traffic through DUT from Tport1 to Tport3 2. tester sends burst traffic 100000 packets through DUT from Tport2 to Tport3, which makes port3 congested。 3. Test 512bytes/1518bytes/9216bytes，buffer size=(total number of burst packets- number of packet lost)x packet size, record as Result 1. |
| Expected Result | 1. Record the size of buffer |
| Descriptions |  |
| Results |  |

## VoQ

|  |  |
| --- | --- |
| Objectives of Tests | Virtual Output Queue |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | **Scenario 1**   1. Tester sends 100% line-rate traffic through DUT from Tport1 to Tport3, and sends 50% line-rate traffic through DUT from Tport2 to Tport3, sends 50% line-rate traffic through DUT from Tport2 and Tport4, which makes port 3 of DUT congested, check at Tport4 to see if there is packet loss, record as Result 1.   **Scenario 2**   1. Tester sends 50% line-rate traffic through DUT from Tport1 and Tport3, sends 50% line-rate traffic through DUT from Tport1 and Tport4; and tester sends 60% line-rate traffic through DUT from Tport2 to Tport3, sends 40% line-rate traffic through DUT from Tport2 to Tport4, which makes port3 of DUT congested, check at Tport4 to see if there is packet loss, record as Result 2. |
| Expected Result | 1. No packet loss at designated port 2. No packet loss at designated port |
| Descriptions |  |
| Results |  |

## Buffer test for 4x25GE-100GE

|  |  |
| --- | --- |
| Objectives of Tests | Test 25G，100G interface buffer |
| Test configurations and connections (as shown in the figure) | * + - 1. 4 25GE Interface port1-4 send line rate traffic to 100G interface 5, and the fifth 25G interface send burst traffic to interface 5, write down latency, packet loss and buffer       2. One 100G interface send traffic to 4 25G interface, write down latency, packet loss, buffer, and traffic bandwidth allocation between 25G interfaces |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure |  |
| Expected Result |  |
| Descriptions |  |
| Results |  |

## Traffic congestion

|  |  |
| --- | --- |
| Objectives of Tests | Test Traffic Congestion |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Configurationn DUT ip address at the interface connected with traffic generator,TC1 is the 100G interface, TC2 and TC3 is the 25G interface 2. Send 25G traffic from TC1 to TC2, with expected results 1 3. Send 25G traffic from TC1 to TC3, with expected results 2 4. Increase 2G traafic to TC2, with the expected results 3 |
| Expected Result | 1. TC2 received traffic forwarding does not lose packets 2. TC3 receives traffic forwarding without packet loss TC2 received traffic forwarding due to congestion and packet loss, TC3 received traffic forwarding packet loss |
| Descriptions |  |
| Results |  |

## Mirror the packets marked with TCP port number

|  |  |
| --- | --- |
| Objectives of Tests | Packet mirror based on TCP port number |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Port2 is mirrored port, Port3 is port for analyzer, port1 and Port2 are in same vlan 2. Apply ACL to match source TCP port and set up the port mirroring at ingress of port 1 3. Tester send traffic from Tport1 to Tport2, record as Result1. |
| Expected Result | 1. Tport2 receives traffic normally, Tport3 receives the mirrored traffic normally. |
| Descriptions |  |
| Results |  |

## Tests on Whole System Reboot

|  |  |
| --- | --- |
| Objectives of Tests | Test the time for rebooting the whole system with full ports forwarding performance |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. With above setup, start bi-directional traffic with Ixia. 2. Make sure traffic is passing end to end. 3. Cut off the power to one of the DUTs then re-power it. 4. Measure the time it took to resume the traffic – by measuring the number of packets lost [by total packet lost / rate of traffic]. |
| Expected Results | 1. Record the time. |
| Descriptions |  |
| Results |  |

## Tests on Power Supply Redundancy

|  |  |
| --- | --- |
| Objectives of Tests | Tests on Power Supply Redundancy *\*\*Test not applicable on SN2100* |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. With above setup, start bi-directional traffic with Ixia. 2. Make sure traffic is passing end to end. 3. Pull one of the power supplies from the switch. |
| Expected Result | 1. Traffic can be forwarded normally without packet loss. |
| Descriptions |  |
| Results |  |

## Tests on Fan Redundancy

|  |  |
| --- | --- |
| Objectives of Tests | Tests on fan redundancy. *\*\*Test not applicable on SN2100* |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. With above setup, start bi-directional traffic with Ixia. 2. Make sure traffic is passing end to end. 3. Pull one of the fan units. |
| Expected Result | 1. Traffic can be forwarded normally without packet loss. |
| Descriptions |  |
| Results |  |

## Power Consumption Test

|  |  |
| --- | --- |
| Objectives of Tests | Power consumption test – fully loaded |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. In above setup – shut down all 2. Make sure traffic is passing end to end. 3. Pull one of the fan units. |
| Expected Result | 1. Traffic can be forwarded normally without packet loss. |
| Descriptions |  |
| Results | |  |  | | --- | --- | | Traffic [%] | PDU Power | | | 0 | 129 | | 10 | 130 | | 20 | 132 | | 30 | 136 | | 40 | 138 | | 50 | 141 | | 60 | 144 | | 70 | 148 | | 80 | 150 | | 90 | 151 | | 100 | 155 | |

# Basic Function Test

## Telnet

|  |  |
| --- | --- |
| Objectives of Tests | Telnet functional test |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Enable telnet on DUT (if required). 2. Initiate a telnet request from PC to DUT. 3. Try multiple simultaneous sessions and record the number. |
| Expected Result | 1. PC can telnet to DUT normally. 2. Record on how many telnet sessions can be running in parallel. |
| Descriptions |  |
| Results |  |

## SSH

|  |  |
| --- | --- |
| Objectives of Tests | SSH functional test |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Enable SSH on DUT. 2. Initiate a SSH request from PC to DUT. 3. Try SCP to and from PC/DUT. |
| Expected Result | 1. SSH connections are possible. 2. DUT can upload/download files from/to PC through SCP. |
| Descriptions |  |
| Results |  |

## Tacacs+

|  |  |
| --- | --- |
| Objective s of Tests | Tacacs+ functional test |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. On DUT, enable telnet and SSH, deleting all local accounts and then enabling authentication as TACAS+. 2. Create an account in TACAS Server for telnet and SSH, and set the privilege for telnet/SSH DUT. 3. Try running telnet/SSH from PC with configured and unconfigured accounts. |
| Expected Result | 1. After checking the Tacacs+ authentication, we can login into the device from PC and be authorized correctly. 2. Verify that the correct user roles are assigned to logged in users (who are authenticated by Tacacs+). |
| Descriptions |  |
| Results |  |

## SNMPv2/v3

|  |  |
| --- | --- |
| Objectives of Tests | SNMP functional test |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. On the DUT, enable snmpv2. 2. On the PC, get the sysObjectID information in a form of attachment in MIB management software. 3. On the DUT, change to enable snmpv3, then configure the authentication; repeat step 1. |
| Expected Result | 1. Verify that DUT supports reading OID information from attachments and confirm that SNMPv2 can retrieve the version and patch information at the same time. 2. Verify that DUT supports reading OID information from attachments and confirm that SNMPv3 can retrieve the version and patch information at the same time. |
| Descriptions |  |
| Results |  |

## LLDP

|  |  |
| --- | --- |
| Objectives of Tests | LLDP functional test |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Setup two links between DUTs, then enable LLDP. 2. Execute show LLDP commands and record the result. |
| Expected Result | 1. Verify that DUT can read LLDP neighbor information and record the display formats for the neighbor. |
| Descriptions |  |
| Results |  |

## Syslog

|  |  |
| --- | --- |
| Objectives of Tests | Syslog functional test |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. On the DUT, enable syslog, then specify the syslog server; establish an OSPF neighborhood relationship between two DUTs. 2. On the DUT, flip the link between two switches. 3. Record the syslog on the syslog server. |
| Expected Result | 1. Verify that interface Up/Down log and OSPF neighbor Up/Down log can be received on the Syslog server. |
| Descriptions |  |
| Results |  |

## NTP

|  |  |
| --- | --- |
| Objectives of Tests | NTP functional test |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. On DUT1, enable NTP server; while on DUT2, enable NTP client. 2. Change the system time on DUT1. Record the time on DUT2. 3. Change DUT1 into NTP client and specify DUT2 as server. Record the time on DUT1. |
| Expected Result | 1. Verify that the time on DUT2 can be synchronized with DUT1 time. 2. Verify that with changed settings, DUT1 can be synchronized with DUT2 time. |
| Descriptions |  |
| Results |  |

## Port Mirroring

|  |  |
| --- | --- |
| Objectives of Tests | Port mirroring functional test |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Port2 is a mirrored port, Port3 is port for analyzer, Port1 and Port2 are in the same VLAN. 2. Ixia sends traffic from Tport1 to Tport2. 3. Check the ingress on Ixia on Tport3. |
| Expected Result | 1. Verify that Tport2 receives traffic normally, while Tport3 receives the mirrored traffic normally. |
| Descriptions |  |
| Results |  |

## Out-of-Band Management VRF

|  |  |
| --- | --- |
| Objectives of Tests | Out-of-band management VRF functional test |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Setup VRF, configure an IP address on management port and combine management port with this VRF. 2. Enable telnet on DUT. 3. Try doing enables telnet from PC to DUT. |
| Expected Result | 1. Verify that a telnet based login can be done from PC into DUT normally. |
| Descriptions |  |
| Results |  |

# sFlow

## sFlow

|  |  |
| --- | --- |
| Objectives of Tests | sFlow functional test |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Enable sFlow on DUT, set collector ID as 1, sampling rate as 4096, polling interval as 120 2. Configure Port 3 as sFlow collector port. 3. Send Ixia traffic from Tport1 to Tport2, Port3 is used for sFlow collector. |
| Expected Result | 1. Verify sFlow works. 2. sFlow receives traffic normally on Tport3. |
| Descriptions |  |
| Results |  |

# Layer 2 Basic Test

## MAC-Table Scaling

|  |  |
| --- | --- |
| Objectives of Tests | Test the size of MAC table |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. In above setup, send various src-mac from Tport1, the number of src-mac is the value DUT claims support. Check the DUT for how many MACs have been learned. 2. Send traffic from Tport2 destined to the exact dst-mac addresses (sent in step-1). |
| Expected Result | 1. Verify that DUT can learn all the MACs sent from Ixia from Tport1. 2. When the traffic sent from Tport2, verify that only Tport1 can receive traffic, Tport3 can’t receive any traffic. 3. Specify the MAC scale learned in step-1 of the test procedure. |
| Descriptions |  |
| Results |  |

## MAC Address Learning Rate

|  |  |
| --- | --- |
| Objectives of Tests | MAC address learning rate |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. In above setup, Ixia sends data traffic with various dst-mac from Tport1, the number of dst-mac is the value DUT claims support. 2. Ixia sends traffic from Tport2 with src-mac same as the dst-mac of traffic arriving from Tport1 (in step-1). |
| Expected Result | 1. Verify that initially both Tport2 and Tport3 can receive traffic (as initially traffic will flood, being unknown unicast). 2. Verify that traffic to Tport3 decreases by and by, and becomes 0 finally (as the MAC is learned). 3. Record the time it takes from traffic deceasing to 0 as t1. MAC-learning rate=MAC table size/t1. |
| Descriptions |  |
| Results | root@r-csi-sn2410-02:~# net show bridge macs dynamic | wc -l  88004    76813/2.852= 26933 Macs/sec |

## LAG with Max Member

|  |  |
| --- | --- |
| Objectives of Tests | Test the max number of LAG members |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Create LAG with max number of member links supported. 2. Make Tport1 and Tport2 within the same VLAN. 3. Sends traffic with various SIPs from Tport1 to Tport2. |
| Expected Result | 1. Verify that the traffic is forwarding normally. 2. Check the counters for LAG ports and verify that traffic is load balanced evenly among the member links. |
| Descriptions |  |
| Results |  |

## STP

|  |  |
| --- | --- |
| Objectives of Tests | Test the functionality of STP (RSTP, PVST or regular STP) |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Running STP among 3 DUTs, make DUT1 as a root bridge. 2. Observe if STP works well to break the loop, by selecting right ports to block based on right priority (port MAC address based). 3. Unplug cable between DUT1 and DUT3, watch the state change of the ports. |
| Expected Result | 1. Verify that DUT supports STP and breaks the loop as expected – verify various/needed flavors of STP. 2. Verify that the STP state change as expected. |
| Descriptions |  |
| Results |  |

## Storm Suppression

|  |  |
| --- | --- |
| Objectives of Tests | Test the storm suppression |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Make the ports of DUT within the same VLAN, configure broadcast suppression that allows 500Mbps broadcast passed 2. Tester sends traffic with various src-mac from Tport1, then stop traffic after DUT learns them. 3. Tester sends 10% line-rate broadcast traffic, 10% line-rate unicast traffic destined to the MAC learnt by DUT. 4. Observe the rate of broadcast traffic and Unicast traffic at Tport2, record as Result 1. |
| Expected Result | 1. Broadcast traffic is suppressed well, but unicast traffic is not affected. |
| Descriptions |  |
| Results |  |

## Max number of LAGs

|  |  |
| --- | --- |
| Objectives of Tests | Test the max number of LAG |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Create max number of LAGs as system claims to support, each LAG contains one port. |
| Expected Result | The max number LAGs are same as the number system claims to support. |
| Descriptions |  |
| Results |  |

## Mac flapping

|  |  |
| --- | --- |
| Objectives of Tests | Verify function of MAC flapping |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. TC1 connects port1 of DUT;TC2 connects to port2 of DUT; TC3 connects to port3 of DUT 2. Tester injects traffic with dmac=mac1 from TC1 into DUT, record as result-1。 3. Tester injects traffic with smac=mac1 from bothTC2&TC3, record as result-2。 4. DUT enable mac flapping, record as result-3 |
| Expected Result | 1. Both TC2&TC3 can receive traffic 2. mac1 moves from port2 to port 3 back and forth 3. Verify if DUT can shutdown the ports where mac flapping over , verify if DUT can designate some a port that won’t be shutdown even mac-flapping happens. |
| Descriptions |  |
| Results |  |

## STP edge port

|  |  |
| --- | --- |
| Objectives of Tests | Verify the function of STP Edge port |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. TC1 connects port1 of DUT; TC2 connects to port2 of DUT; 2. DUT doesn’t enable stp edge, Inject any kind of L2 traffic from TC 1, change port 2 of DUT from shutdown to “no-shutdown”, record as result-1 3. DUT enables stp edge, Inject any kind of L2 traffic from TC 1, change port 2 of DUT from shutdown to “no-shutdown”, record as result-2 |
| Expected Result | 1. Port 2 of DUT becomes forwarding until it takes as long as stp delay timer, then TC2 receives packets, record the delay time 2. Port 2 of DUT becomes forwarding at once, TC2 receives packets |
| Descriptions |  |
| Results |  |

## STP bpdu-guard

|  |  |
| --- | --- |
| Objectives of Tests | Verify the function of DUT with STP bpdu-guard |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. TC1 connects port1 of DUT; TC2 connects to port2 of DUT; 2. DUT doesn’t enable bpdu-guard，set port1 as stp edge port, TC1 sends BPDU packet,record as result-1 3. DUT enables bpdu-guard set port1 as stp edge port, TC1 sends BPDU packet,record as result-2 |
| Expected Result | 1. Port1 is reset to non-edge port, DUT recalculates STP. 2. Port1 is shutdown, and no shutdown after some a while. |
| Descriptions |  |
| Results |  |

## Hierarchical HASH

|  |  |
| --- | --- |
| Objectives of Tests | Test if traffic can be load balanced evenly on tier1 ECMP and tier2 ecmp |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Inject 4 flows from TC3 2. Traffic will be hashed and load balance over the 2 links to 2 switches, traffic arrived at the left switch will be hashed and load balanced over the 2 links to TC1 and TC2. 3. Record the result. |
| Expected Result | The traffic from tier1 switch to tier2 switch is even.  The traffic from tier2 switch to TC2 and TC3 is even. |
| Descriptions | Hash factor can be tuned to avoid traffic polarized at tier2. |
| Results |  |

## ACL Test

|  |  |
| --- | --- |
| Objectives of Tests | Test ACL spec |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Configure the number of acl rules on the DUT. The interface connected to TC is a Layer 3 interface 2. The DUT is connected to TC1 and the outgoing ACL is configured. The action is deny, TC1 sends the matching acl traffic to TC2, with expected result 1 3. In step 2, issue the same ACL to the incoming direction of the TC2 interface with expected result 2 4. In step 2, issue the same ACL to the outbound direction of the TC1 interface. TC2 sends the matching ACL traffic to TC1 with the expected result 3 5. Configure the ACL on the DUT to match the TCP tag bits, respectively, to the outgoing direction and the inbound direction of TC1. The actions are deny, TC1 and TC2 respectively enter the traffic matching the TCP tag bit, and the expected result 4 6. Change the interface to TC to Layer 2 and repeat steps 2 to 5 |
| Expected Result | 1. DUT can be issued on the declared specifications of the acl number, and support to check resource usage, test traffic is discarded by ACL 2. Record acl resource usage 3. Record acl resource usage, test traffic is discarded by ACL 4. The ACL can be correctly configured to the interface. The traffic can be correctly matched and discarded |
| Descriptions |  |
| Results |  |

# MLAG

## MLAG Basic Test

|  |  |
| --- | --- |
| Objectives of Tests | Multi-chassis LAG functional test |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Setup MLAG between DUT1 and DUT2. 2. Setup LAG on access switch, and connect to DUT1 and DUT2. 3. Test sends traffic with various src-mac/dst-mac and src-ip/dst/ip from Port-A to Port-B end to end. 4. Try same settings with and without STP. 5. Check MAC address tables on DUT1 and DUT2 |
| Expected Result | 1. Verify that traffic forwarding is normal without packet loss and load balanced evenly on uplinks with or without STP. 2. Verify that the MAC addresses are synced. |
| Descriptions |  |
| Results |  |

## MLAG Basic Test with IGMP

|  |  |
| --- | --- |
| Objectives of Tests | Multi-chassis LAG functional test with IGMP |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Setup MLAG between DUT1 and DUT2. 2. Setup LAG on access switch, and connect to DUT1 and DUT2. 3. Send IGMP reports from Port-A and send L2 MCAST traffic from Port-B. 4. Try same settings with and without STP. 5. Check IGMP table on DUT1 and DUT2. |
| Expected Result | 1. Verify that traffic forwarding is normal without packet loss and load balanced evenly on uplinks with or without STP. 2. Verify that the IGMP table is synced. |
| Descriptions |  |
| Results |  |

## MLAG Split

|  |  |
| --- | --- |
| Objectives of Tests | MLAG split functional test. *\*\*Mellanox-OS only* |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Setup MLAG between DUT1 and DUT2. 2. Setup a KeepAlive link between DUT1 and DUT2. *In CL the backup link is optional.* 3. Setup LAG on access switch, and connect to DUT1 and DUT2. 4. Send traffic with various src-mac/dst-mac and src-ip/dst/ip from Port-A to Port-B. 5. Unplug keep. |
| Expected Result | 1. Forwarding normally without packet loss and load balanced evenly. 2. When MLAG splits, the MLAG port in standby unit status changes, traffic converges well. |
| Descriptions |  |
| Results |  |

## MLAG with MAGP

|  |  |
| --- | --- |
| Objectives of Tests | Test MLAG with MAGP with failover |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Setup MLAG between DUT1 and DUT2. 2. Setup LAG on access switch, and connect to DUT1 and DUT2. 3. Configure MAGP on DUT1 and DUT2 towards uplink (spine). 4. Send traffic with various src-mac/dst-mac and src-ip/dst/ip from Port-A to Port-B, record as results for following failover timings. 5. Unplug the link between access switch and DUT1 and spine, see that the flow which got impacted due to link down has migrated to a second uplink. Test this for bi-directional traffic. |
| Expected Result | Record the packet drops in ms. |
| Descriptions |  |
| Results |  |

## MLAG Convergence – MLAG Link Failover

|  |  |
| --- | --- |
| Objectives of Tests | Test MLAG failover numbers |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Setup MLAG between DUT1 and DUT2. 2. Setup LAG on access switch, and connect to DUT1 and DUT2. 3. Send traffic with various src-mac/dst-mac and src-ip/dst/ip from Port-A to Port-B, record as Results for following failover timings:    1. Unplug the link between access switch and DUT – 1 and measure the packet loss for the stream to switch from Access – DUT1 link to Access – DUT2 link.    2. Plug the link back and measure the timing again. 4. Unplug cables of keepalive, record as Result 2. |
| Expected Result | 1. Record the packet drops in ms. |
| Descriptions |  |
| Results |  |

## MLAG Convergence – Reboot

|  |  |
| --- | --- |
| Objectives of Tests | Test MLAG failover numbers |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Setup MLAG between DUT1 and DUT2. 2. Setup LAG on access switch, and connect to DUT1 and DUT2. 3. Test sends traffic with various src-mac/dst-mac and src-ip/dst/ip from Port-A to Port-B, record as Results for following failover timings:    1. Power down DUT1 and measure the packet loss for the stream to switch from Access – DUT1 link to Access – DUT2 link.    2. Bring up the DUT1 back and measure the timing again. 4. Unplug cables of keepalive, record as Result 2. |
| Expected Result | 1. Record the packet drops in ms. |
| Descriptions |  |
| Results |  |

## MLAG Convergence – IPL

|  |  |
| --- | --- |
| Objectives of Tests | Test MLAG failover numbers |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Setup MLAG between DUT1 and DUT2. 2. Setup LAG on access switch, and connect to DUT1 and DUT2. 3. Send traffic with various src-mac/dst-mac and src-ip/dst/ip from Port-A to Port-B, make sure there is no traffic on IPL: 4. Kill IPL link and measure the packet loss if any. |
| Expected Result | 1. Verify no packet loss. |
| Descriptions |  |
| Results |  |

# Layer 3 & Routing

## ARP-Table Scaling

|  |  |
| --- | --- |
| Objectives of Tests | Test the size of ARP table. |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. In above setup, Ixia sends max number of 128byte ARP-reply packets that advertise ARP entries that DUT claims support from Tport1. 2. Ixia sends traffic from Tport2 destined to the exact ARP entries. 3. Measure the size of ARP table. |
| Expected Result | 1. Verify that DUT learns all the ARPs. 2. Verify that traffic forwarding works well for all ARP entries. 3. Measure the ARP table size. |
| Descriptions |  |
| Results |  |

## ARP Learning Rate

|  |  |
| --- | --- |
| Objectives of Tests | Test the learning rate of ARP |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. In above setup, Ixia sends traffic destined to max number of ARP the DUT claims support from Tport2. 2. Ixia sends 128byte line-rate traffic of ARP reply packets from Tport1. |
| Expected Result | 1. Traffic is dropped 100% without ARP entries. 2. Traffic can be received as the DUT starts learning ARPs, record the time it takes since the traffic can be received to fully received, record as t1. 3. ARP learning-rate= Number of ARPs/t1. |
| Descriptions |  |
| Results | root@r-csi-sn2410-02:~# cl-resource-query  IPv4 host entries: 24001, 97% of maximum value 24576  IPv6 host entries: 0, 0% of maximum value 20480  IPv4 neighbors: 24001  IPv6 neighbors: 0    24000 ARPs/10 sec = 2400 ARPs/sec |

## ECMP

|  |  |
| --- | --- |
| Objectives of Tests | Test ECMP |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Connects max number of ECMP links that DUT claims support. 2. Establish EBGP neighbor between DUT1 and DUT2 for each ECMP link. 3. Verify the IXIA networks are learned via multiple ECMP next hops. 4. Use Ixia to send L3 traffic with various SIPs from Tport1 to Tport2. |
| Expected Result | 1. Traffic forwarding works well and load balanced evenly among links. |
| Descriptions |  |
| Results |  |

## ND-table scaling

|  |  |
| --- | --- |
| Objectives of Tests | Test the capacity of IPv6 Neighbor capacity |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. TC1 connects port1 of DUT; TC2 connects to port2 of DUT; 2. Configure IPv4& IPv6 addressess over port1 and port2. 3. TC1 injects line-rate traffic of 128byte ARP reply, record as result-1 4. TC2 injects traffic destined to the IP addresses same as ARP replied, record as result-2 5. TC1 injects line-rate traffic of 128byte ND RA, record as result-3 6. TC2 injects traffic destined to the IPv6 addresses same as ND RA, record as result-4 |
| Expected Result | 1. DUT can learn ARP entries, and the capacity of ARP table is same as the number that system claims to support. 2. Packet forwarding works well for all entries. 3. DUT can learn IPv6 neighbor entries, and the capacity of ND table is same as the number that system claims to support. 4. Packet forwarding works well for all entries. |
| Descriptions |  |
| Results |  |

## ND learning rate

|  |  |
| --- | --- |
| Objectives of Tests | Test ND learning rate |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. TC1 connects port1 of DUT; TC2 connects to port2 of DUT; 2. Configure IPv4& IPv6 addressess over port1 and port2. 3. TC1 injects line-rate traffic of 128byte ARP reply, record as result-1 4. TC2 injects traffic destined to the IP addresses same as ARP replied, record as result-2 5. TC1 injects line-rate traffic of 128byte ND RA, record as result-3 6. TC2 injects traffic destined to the IPv6 addresses same as ND RA, record as result-4 |
| Expected Result | 1. Packet dropped due to no ARP at the beginning. 2. Traffic forwarding starts by and by along with the ARP learning, record the interval as t1 since traffic starts forwarding to all ARP are learnt , arp learning-rate= number of arp/t1 3. Packet dropped due to no ND at the beginning. 4. Traffic forwarding starts by and by along with the ND learning, record the interval as t2 since traffic starts forwarding to all ARP are learnt , arp learning-rate= number of ND/t2 |
| Descriptions |  |
| Results |  |

## Arp migration

|  |  |
| --- | --- |
| Objectives of Tests | Test how DUT deal with ARP migration |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. TC2 injects ARP reply traffic with 100 different src-ip and mac , record as result-1 2. TC1 injects traffic destined to these IP address that ARP advertised, record as result-2 3. TC3 injects ARP reply traffic with 100 different src-ip and different mac , record as result-3 4. TC1 injects traffic destined to these IP address that ARP advertised, record as result-4。 |
| Expected Result | 1. DUT learns ARP entries 2. Traffic can be forwarded to TC2, mac and IP are correct by capturing packet at TC2. 3. DUT updates ARP entries correctly. 4. Traffic can be forwarded to TC3, mac and IP are correct by capturing packet at TC3. |
| Descriptions |  |
| Results |  |

## Aging of MAC in ARP table

|  |  |
| --- | --- |
| Objectives of Tests | Test how DUT deal with aging-out of MAC in ARP table. |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. TC1 connects port1 of DUT;TC2 connects to port2 of DUT; TC3 connects to port3 of DUT 2. Set arp age-out timer as 120s，mac age-out timer as 30 , assign port 1/2/3 into a same vlan 3. TesterTC3 injects arp replay，record as result-1。 4. TC1 injects traffic destined to these IP addresses that ARP advertised, record as result-2 5. Wait for 30s，check ARP entries in DUT when MAC is already aged out, record as result-3 |
| Expected Result | 1. DUT learns ARP entreis correctly. 2. TC2 can receives traffic, TC3 can’t 3. Observe the packet received by TC2 and T3C to conclude how DUT works under this scenario. |
| Descriptions |  |
| Results |  |

## Max number of secondary IP addresses under a single VLAN IP interface

|  |  |
| --- | --- |
| Objectives of Tests | Verify the max number of secondary IP addresses in a vlan interface |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. TC1 connects port1 of DUT; TC2 connects to port2 of DUT; 2. Assign port 1 in DUT into some a vlan 3. Create SVI interface for this vlan, configure max number of secondary IP addresses in this SVI 4. Create hosts under these IP subnets over TC1 of Tester, record as result-1。 5. Tester sends bidirectional traffic between TC1&TC2, record as result-2。 |
| Expected Result | 1. Hosts over TC1 can resolve ARP correctly 2. No packet loss for bidirectional traffic |
| Descriptions |  |
| Results |  |

## ECMP Hash

|  |  |
| --- | --- |
| Objectives of Tests | Test hash function |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Connects max number of ECMP links that DUT claims support. 2. Establish EBGP neighbor between DUT1 and DUT2 for each ECMP link. 3. Verify the IXIA networks are learned via multiple ECMP next hops. 4. Use Ixia to send L3 traffic with various SIPs from Tport1 to Tport2. 5. Tune the factor of hash in DUT, repeat step2, record as Result. |
| Expected Result | 1. Traffic forwarding works well without packet loss. 2. Hash factor can make load balanced evenly among the links. |
| Descriptions |  |
| Results |  |

## L3 ACL Functionality

|  |  |
| --- | --- |
| Objectives of Tests | ACL test |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Make sure Port1 and Port2 are in different VLAN and have different IP gateways, tester sends L3 traffic from Tport1 to Tport2. 2. Configure ACL in DUT, match SIP and action deny, apply it at ingress of Port1, and check the traffic at Tport2. |
| Expected Result | 1. Verify traffic forwarding works well without packet loss in step-1. 2. Verify Tport2 can’t see traffic in step-2. |
| Descriptions |  |
| Results |  |

## OSPF Convergence

|  |  |
| --- | --- |
| Objectives of Tests | Test convergence time of OSPF |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. In above setup, have Ixia advertise the same 10000 OSPF routes to DUT from Tport2 and Tport3. 2. Send traffic 100,000 frames/s from Tport1, destined to the exact 10,000 routes. 3. Count the number of received packets at Tport2 and Tport3. 4. Unplug the link to Port2 of DUT, wait for OSPF convergence. 5. Stop sending traffic, the total number of packet sent=N1, the number of packets received at Tport1= N2, the number of packets received at Tport2= N3, time of OSPF convergence=[N1-(N2+N3)]/100000/10000, record as Result 1. |
| Expected Result | 1. Measure the OSPF convergence number. |
| Descriptions |  |
| Results |  |

## Number of ospf neighbor

|  |  |
| --- | --- |
| Objectives of Tests | Test the number of ospf neighbor |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Establish the max number of OSPF adjacency that DUT claims support. 2. Send 2 way traffic between Tport1 and port2. |
| Expected Result | 1. Ospf adjacencies are established. Measure the number of neighbor. 2. Traffic is forwarded without packet loss. |
| Descriptions |  |
| Results |  |

## Route metric test

|  |  |
| --- | --- |
| Objectives of Tests | Test the preference among policy based route, host route, ARP and regular routes. |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. TC1 connects port1 of DUT;TC2 connects to port2 of DUT; TC3 connects to port3 of DUT; TC4 connects port4 of DUT;TC5 connects to port5 of DUT 2. Configure IP addresses in tester as below   TC1 10.1.1.2/24，GW 10.1.1.1  TC2 20.1.1.2/24，GW 20.1.1.1  TC3 30.1.1.2/24，GW 30.1.1.1  TC4 40.1.1.2/24，GW 40.1.1.1  TC5 50.1.1.2/24，GW 50.1.1.1   1. TC1 sends traffic destined to 40.1.1.100, record as result-1 2. Set nexthop as 20.1.1.2 for host route for 40.1.1.100/30, record as result-2 3. Create a host with IP 40.1.1.100 at TC4, and make DUT learns the ARP, record as result-3 4. Set nexthop as 30.1.1.2 for host route for 40.1.1.100/30, record as result-4 5. Set nexthop as 50.1.1.2 for policy-based route for 40.1.1.100/30, record as result-5 6. Change MTU to 1546，repeat step 3-4 |
| Expected Result | 1. All packets are lost 2. TC2 receives traffic 3. TC4 receives traffic 4. TC3 receives traffic 5. TC5 receives traffic |
| Descriptions |  |
| Results |  |

## OSPF scaling in a single area and convergence

|  |  |
| --- | --- |
| Objectives of Tests | OSPF scalaing in a single area and local convergence |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Establish max adjacecnies between DUT and tester, tester advertises 100 routes per an ajacendy.  2. OSPF network mode is set to P2P, and enable MD5 autentication at interfaces, Cost 10，SPF initial timer as 1s, delay 5s, SPF hold timer as 5s  3. Inject 4000 bidirectionaly TCP traffic of 256byte for 30mins  4. Tester establishes a new adjacncey and advertise a route by another port.  5. Tester sends two-way traffic between the new one route to the old 4000 routes  6. Checking CPU usages, memeory usage, and packet loss. |
| Expected Result | Port configurations  Ospf configuration  The max number routes learnt by DUT  No packet loss, cpu usage and memory usage are reasonable. |
| Descriptions |  |
| Results |  |

## Multiple OSPF neighbor’s flapping

|  |  |
| --- | --- |
| Objectives of Tests | Test cpu usage when adjacencies flapping |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Connect multiple links between DUT and tester 2. Tester simulates 60 OSPF routers per port and form a non-0 area, so as the make the adjacencies reach the max number 3. Tester sends max number of routes that DUT claims to support 4. Shutdown/no shutdown links between DUT and tester back and forth, observe the adjacencies converge and the routes learnt ,and the cpu usage. |
| Expected Result |  |
| Descriptions |  |
| Results |  |

## Performance for OSPF running in Multi-VRFs

|  |  |
| --- | --- |
| Objectives of Tests | Performance for OSPF running in Multi-VRFs |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Each VRF establishes adjacency between TC and DUT, inject routes into DUT. 2. Record how many adjacencies can be establish and the number of routes DUT can learn. |
| Expected Result |  |
| Descriptions |  |
| Results |  |

## OSPF routing flapping

|  |  |
| --- | --- |
| Objectives of Tests | Check the cpu and memory usage when OSPF route flapping |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. P-1 advertises the max number of routes to DUT 2. P-2 sends line rate traffic destined to the routes 3. P-1 frequently add/remove part of rotues(type3), observer the cpu/memory usage and traffic, record as result-1. |
| Expected Result |  |
| Descriptions |  |
| Results |  |

## OSPF convergence under ECMP

|  |  |
| --- | --- |
| Objectives of Tests | OSPF convergence under ECMP |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. P-1 advertises the max number of routes to DUT 2. P-2 sends traffic destined to the routes, 100000 frames/s 3. Unplug one link between the 2 DUT, traffic is re-routed to another link. 4. Stop traffic , the number of Tx packet of PORT2 is N1，the number rx of PORT1is N2，convergence time= [N1-N2]/100000 |
| Expected Result |  |
| Descriptions |  |
| Results |  |

## BGP/BGPv6 RIB/FIB size

|  |  |
| --- | --- |
| Objectives of Tests | Test BGP/BGPv6 RIB/FIB Size |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | * + - 1. Establish EBGP neighbors between DUT and TC1       2. TC1 send device to claimed FIB specification 32-bit BGP route, with expected results 1       3. TC2 sends the destination address to the traffic of the published route, with expected result 2       4. When the route is not revoked, TC1 continues to send the 32-bit BGP route until the device reach to claimed RIB specification, with expected result 3       5. Change the sending route to /26 and repeat steps 2 to 4       6. Establish EBGPv6 neighbors between DUT and TC1, and repeat steps 2 to 5 using IPv6 routes       7. Repeat steps 2 through 6 using Discrete Routing |
| Expected Result | 1. DUT can correctly learn the published route 2. All traffic can be forwarded normally without packet loss. 3. DUT can correctly learn the number of claimed RIB. |
| Descriptions |  |
| Results |  |

## BGP/BGPv6 routes learning-rate

|  |  |
| --- | --- |
| Objectives of Tests | Test BGP/BGPv6 Routes Learning rate |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Establish EBGP neighbors between DUT and TC1 2. TC1 send the test traffic of the 32-bit route whose destination address reach to the claimed FIB specification, with expected result 1 3. TC2 publishes the route corresponding to the destination address of the test flow, with expected result 2 4. Change the sending route to 26 and repeat steps 2 to 3 5. Repeat steps 2 through 4 using BGPv6 |
| Expected Result | 1. Because there is no route, the test traffic cannot be forwarded to TC2 2. The DUT learns the routing by step, and the test traffic is forwarded to TC2, records the time t1 at which TC2 begins to receive traffic to receive all traffic, route learning speed = declared FIB number / t1. |
| Descriptions |  |
| Results |  |

## BGP update-interval test

|  |  |
| --- | --- |
| Objectives of Tests | Test BGP update-interval configuration |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Establish EBGP neighbors between DUT and TC1 and TC2 2. The DUT use network to advertise the connected interface to TC1. The TC1 advertises the BGP route to the DUT with the expected result 1 3. Shutdown / undo shutdown interface connectted to TC1 with expected result 2 4. Modify the value of setting update-intervel, repeat step 3, with expected result 3 5. Set the value of update-intervel to 0, repeat step 3, with expected result 4 |
| Expected Result | 1. DUT and TC2 can learn the route advertised by TC1 and TC1 interface segment route 2. On the DUT and TC2 check the route update time 3. The DUT sends update to TC2 via the update-intervel interval 4. DUT immediately send update to TC2 |
| Descriptions |  |
| Results |  |

## BGP/BGPv6 routing policy

|  |  |
| --- | --- |
| Objectives of Tests | Test BGP/BGPv6 Routing Policy |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Establish EBGP neighbors between DUT and TC1 and TC2, and do not configure any BGP policies 2. TC1 advertised two routes of network1 and network2, with expected result 1 3. Configure the routing policy for the incoming direction of TC1 on the DUT, reject network1, accept network2, and have expected result 2 4. Configure the routing policy for the outbound direction of TC2 on the DUT, reject the network2, expect the result 3 5. Repeat steps 2 through 4 using BGPv6 |
| Expected Result | 1. DUT can learn the route of network1 and network2, TC2 can also learn the network1 and network2 route 2. DUT can only learn the route of network2, TC2 can only learn to route2 route. 3. DUT can only learn the route of the network2, TC2 did not learn the route. |
| Descriptions |  |
| Results |  |

## BGP/BGPv6 Route Aggregation

|  |  |
| --- | --- |
| Objectives of Tests | Test BGP/BGPv6 Route Aggregation |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Establish EBGP neighbors between DUT and TC1 and TC2, and do not configure any aggregation 2. TC1 advertised three segment network:   100.1.1.0/30  100.1.1.4/30  100.1.1.8/30，  Configure Three loopback interface in DUT, and network into BGP  110.1.1.1  110.1.1.2  110.1.1.3， with expected result 1   1. Configure route aggregation 100.1.1.0/24 and 110.1.1.0/26 on DUT, and do not configure specific route suppression, with expected result 2 2. Configure route aggregation 100.1.1.0/24 and 110.1.1.0/26 on the DUT, and configure the specific route suppression, with expected result 3 3. Repeat steps 2 through 4 using BGPv6 |
| Expected Result | 1. DUT can learn the / 30-bit and / 32-bit specific routes, TC2 can learn to / 30-bit and / 32-bit 6 specific routes 2. DUT can learn the / 30-bit and / 32-bit specific routes, TC2 can learn to 6 30 and / 32-bit specific routes and / 24-bit and / 26-bit two aggregated routes 3. DUT can learn the / 30-bit and / 32-bit specific routes, TC2 can only learn to / 24-bit and / 26-bit two aggregated routes |
| Descriptions |  |
| Results |  |

## BGP Route Flapping

|  |  |
| --- | --- |
| Objectives of Tests | Test equipment for the BGP Route Flapping |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Establish EBGP neighbors between DUT and TC1 2. The TC1 advertised the route of the declared specification with the expected result 3. TC2 sends the test traffic at 1/3 of the issued route as the destination address, with expected result 2 4. TC1 for the remaining 2/3 routes flapping, TC2 also for this 2/3 of the routes to send test traffic, with the expected results 3 5. Stop the route flapping on TC1 and have the expected result |
| Expected Result | 1. DUT can correctly learn the advertised route 2. Traffic can be forwarded correctly without packet loss. 3. Flapping routes traffic to the test flow have packet loss, the original 1/3 of the test flow without packet loss, after a long time the equipment is running well 4. DUT can re-learn all the advertised routes, all test traffic is correctly forwarded without packet loss |
| Descriptions |  |
| Results |  |

## BGP Convergence

|  |  |
| --- | --- |
| Objectives of Tests | Test time of BGP convergence |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Ixia advertises the same 10000 BGP routes to DUT from Tport2 and Tport3. 2. Ixia sends traffic 100,000 frames/s from Tport1, destined to the exact 10,000 routes. 3. Count the number of received packets at Tport2 and Tport3. 4. Unplug the link to port2 of DUT, wait for BGP convergence. 5. Stop sending traffic, the total number of packet sent=N1, the number of packets received at Tport1= N2, the number of packets received at Tport2= N3, time of BGP convergence=[N1-(N2+N3)]/100000/10000, record as Result 1. |
| Expected Result | 1. Measure the BGP convergence number. |
| Descriptions |  |
| Results |  |

## Number of BGP Neighbors

|  |  |
| --- | --- |
| Objectives of Tests | Test the number of BGP neighbors |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Establish the max number of BGP adjacencies that DUT claims support. 2. Send 2-way traffic between Tport1 and Port2. |
| Expected Result | 1. BGP adjacencies are established. Measure the number of neighbors. 2. Traffic is forwarded without packet loss. |
| Descriptions |  |
| Results |  |

## BGP Unnumbered

|  |  |
| --- | --- |
| Objectives of Tests | Test the number of BGP neighbors with unnumbered |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Configure the system for BGP unnumbered as given in **Mellanox Community.** 2. Send end to end traffic from Ixia (10.10.10.1 to 20.20.20.1). |
| Expected Result | 1. Verify the ping and that data traffic works fine. |
| Descriptions |  |
| Results |  |

# IGMP & PIM-SM

# QOS

## Services Classified by VLAN

|  |  |
| --- | --- |
| Objectives of Tests | Services classified by VLAN |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Ixia sends traffic with different VLAN ID from Tport1. 2. Configure policy to classify traffic at Port 1 of DUT, traffic of VLAN1’s 802.1p=7, VLAN2’s 802.1p=5, the traffic of the rest VLAN’s 802.1p=0. 3. Check if the 802.1p modification works as expected at Tport2, record as Result 1. |
| Expected Result | 1. DUT can classify traffic according to VLAN ID and can modify value of 802.1p. |
| Descriptions |  |
| Results |  |

## Services Classified with L4 Port Number

|  |  |
| --- | --- |
| Objectives of Tests | Services classified by L4 port number |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Ixia sends traffic with various L4 dst-ports from Tport1. 2. Classify traffic at port1 of DUT, set 802.1p=7 for traffic with the dst-port A, set 802.1p=5 for traffic with the dst-port B, keep the rest traffic 802.1p=0. 3. Check if the modification of 802.1p value is correct as expected, record as Result 1. |
| Expected Result | 1. DUT can classify traffic according to L4 port and change 802.1p value. |
| Descriptions |  |
| Results |  |

## Strict Priority Queue

|  |  |
| --- | --- |
| Objectives of Tests | Test strict priority queue |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Set up strict priority queue at Port 3 of DUT. 2. Guarantee traffic with priorities 6-7, and treat the rest of traffic as default. 3. Ixia sends traffic from Tport1 and Tport2 to Tport3 with different priorities, traffic with priorities 6-7 is 15%, traffic with other priorities is 85%. 4. Verify the traffic received at Tport3, record latency and counter of traffic with strict priority record as Result 1. |
| Expected Result | 1. DUT should guarantee the latency and bandwidth for traffic with strict priority. |
| Descriptions |  |
| Results |  |

## Strict Priority and WRR Mixed

|  |  |
| --- | --- |
| Objectives of Tests | WRR |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Set up strict priority queue at Port 3 of DUT. 2. Treat traffic with priority 6-7 as strict priority, treat traffic with priorities 1-5 as weight round robin, allocate 30% bandwidth for traffic with priorities 4-5, allocate 20% bandwidth for traffic with priorities 2-3, allocate 10% bandwidth for traffic with priority 1, best effort for traffic with priority 0. 3. Ixia sends traffic from Tport1 and Tport2 to Tport3 with different priorities, traffic with priorities 6-7 is 20%, traffic with priorities 4-5 is 20%, traffic with priorities 2-3 is 20%, traffic with priority 1 is 20%, traffic with priority 0 is 20 %. 4. Verify the traffic received at Tport3, record latency and counter of traffic with strict priority and other priorities record as Result 1. |
| Expected Result | 1. DUT should guarantee the latency and bandwidth for traffic with strict priority and WRR allocate the corresponding resources to other priorities. |
| Descriptions |  |
| Results |  |

## Ingress Rate-Limit

|  |  |
| --- | --- |
| Objectives of Tests | Ingress rate-limit |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Configure the ingress rate limit function based on physical port, MAC address, VLAN ID, IP address, limit the rate to 100Mbps. 2. Ixia sends 64byte, 1Gbps traffic from Tport1 to Tport2, check the receiving rate at Tport2. |
| Expected Result | 1. DUT support ingress rate limit. |
| Descriptions |  |
| Results |  |

## Egress Rate-Limit

|  |  |
| --- | --- |
| Objectives of Tests | Egress rate-limit |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Configure the egress rate limit function based on physical port, MAC address, VLAN ID, IP address, limit the rate to 100Mbps. 2. Ixia sends 64byte, 1Gbps traffic from Tport1 to Tport2, check the receiving rate at Tport2. |
| Expected Result | 1. DUT support egress rate limit. |
| Descriptions |  |
| Results |  |

## Granularity and Accuracy of Rate-Limit

|  |  |
| --- | --- |
| Objectives of Tests | Granularity and accuracy of rate-limit |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Configure the rate-limit policy to limit traffic from src-ip=IP1 as 100Mbps. 2. Ixia sends traffic of 64bytes and 1518bytes, check the accuracy of rate-limit, record as Result 1. 3. Configure the minimum bandwidth the rate-limit claims support. 4. Ixia sends traffic of 64byte, check the accuracy of rate-limit, record as Result 2. 5. Increase the bandwidth as minimum granularity. 6. Ixia sends traffic of 64byte, check the accuracy of rate-limit, record as Result 3. |
| Expected Result | 1. Both minimum bandwidth and minimum granularity is not bigger than 64Kbps. 2. Accuracy falls within the acceptable value. |
| Descriptions |  |
| Results |  |

# VXLAN

## HW VTEP - VXLAN with NSX

|  |  |
| --- | --- |
| Objectives of Tests | Test VXLAN with NSX |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Configure NSX systems and OVSDB in switch as per following document [(link).](https://community.mellanox.com/docs/DOC-2861) 2. Confirm end to end ping from VMs to bare metal servers. |
| Expected Result | 1. Ping should be fine between bare metal and virtual servers. |
| Descriptions |  |
| Results |  |

## VXLAN with LNV

|  |  |
| --- | --- |
| Objectives of Tests | Testing VXLAN with LNV (static VXLAN) |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Configure LNV based on the [document (link)](https://community.mellanox.com/docs/DOC-2762) 2. Confirm end to end ping between two hosts. |
| Expected Result | 1. End to end ping works fine. |
| Descriptions |  |
| Results |  |

## VXLAN and EVPN

|  |  |
| --- | --- |
| Objectives of Tests | Test VXLAN and EVPN |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Configure EVPN based VXLAN based on this [document link](https://community.mellanox.com/docs/DOC-2798). 2. Test end to end ping and data traffic using Ixia. 3. Tester sends ping traffic between 2 devices, record as Result 1. |
| Expected Result | 1. Ping connectivity is ok & traffic is fine. |
| Descriptions |  |
| Results |  |

## VXLAN Scaling

|  |  |
| --- | --- |
| Objectives of Tests | Test maximum number of VNIs |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Configure maximum number of VNIs between DUT1 and DUT3 (by mapping 1:1 VLAN to VNI). 2. Configure test subnet from Ixia for each VNI. 3. Send data traffic for each VNI from Ixia. |
| Expected Result | 1. Ping connectivity is ok, traffic is fine. 2. Get maximum number of VNIs. |
| Descriptions |  |
| Results |  |

## VTEP Scaling

|  |  |
| --- | --- |
| Objectives of Tests | Test maximum number of remote VTEPs |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Run protocol emulator in Tport2, and make 750 VNIs between DUT port and Tport2. This will emulate 750 VTEPs from the switch’s point of view. 2. Send traffic for a single MAC address from Tport1 (BUM traffic), the traffic should be replicated at DUT and should reach to 750 VTEPs (sitting in Ixia). 3. Run the test for max supported VTEPs. |
| Expected Result | 1. Record the maximum supported VTEPs. |
| Descriptions |  |
| Results |  |

## VXLAN Asymmetric Routing

|  |  |
| --- | --- |
| Objectives of Tests | Test asymmetric VXLAN routing |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Configure the system based on topology with asymmetric routing configurations. 2. Using Ixia ping from 10.10.10.11 to 20.20.20.33 and send end to end traffic. |
| Expected Result | 1. Ping connectivity is ok, traffic can be passed. |
| Descriptions |  |
| Results |  |

## VXLAN Symmetrical Routing

|  |  |
| --- | --- |
| Objectives of Tests | Symmetrical routing |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. TBD |
| Expected Result | 1. TBD |
| Descriptions | *TBD* |
| Results | *TBD* |

# Solutions – Leaf & Spine

## Leaf & Spine with ECMP with BGP & MLAG

|  |  |
| --- | --- |
| Objectives of Tests | Verify that ECMP based load balancing works in leaf/spine setup |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Setup BGP neighbor establish between the leaf and spine. 2. On one side, directly connect the Ixia (emulating L3 on TOR), and on the other side configure MLAG and connect Ixia to the access switch (emulating L2/K3 boundary). 3. Check routing table on leaf to see multiple paths for destination using ECMP. 4. Using Ixia send mixed traffic for SIP/DIP and measure the link counters to see the traffic is getting load balanced. |
| Expected Result | 1. Verify the ECMP load balancing. 2. Verify that MLAG works in conjunction with ECMP/BGP. |
| Descriptions |  |
| Results |  |

## Leaf & Spine with ECMP Resilient Hashing

|  |  |
| --- | --- |
| Objectives of Tests | Verify that ECMP resilient hashing works in leaf/spine setup |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Setup BGP neighborship between the leaf and spine. 2. On one side, directly connect the Ixia (emulating L3 on TOR), and on the other side configure MLAG and connect Ixia to the access switch (emulating L2/L3 boundary). 3. Check routing table on leaf to see multiple paths for destination using ECMP. 4. Using Ixia send mixed traffic for SIP/DIP and measure the link counters to see the traffic is getting load balanced. 5. Once stable – flip the link between the leaf and spine, and see that the flows are transferred to a second spine. 6. Bring the spine link up, and see that flow does not come back. 7. Add one more flow, and see that the existing flow does not get re-hashed by a new flow. |
| Expected Result | 1. Verify the ECMP load balancing works and resilient hashing works with link flap and new flow introduction. |
| Descriptions |  |
| Results |  |

## Leaf & Spine – ECMP Convergence

|  |  |
| --- | --- |
| Objectives of Tests | Measure ECMP convergence |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Setup BGP neighborship between the leaf and spine. 2. On one side, directly connect the Ixia (emulating L3 on TOR), and on the other side configure MLAG and connect Ixia to the access switch (emulating L2/L3 boundary). 3. Check routing table on leaf to see multiple paths for destination using ECMP. 4. Using Ixia send mixed traffic for SIP/DIP and measure the link counters to see the traffic is getting load balanced. 5. Once the flow is settled, kill a link between leaf and spine. 6. Measure the number packets dropped when the flow is forced to change from one ECMP link to another. |
| Expected Result | 1. Measure the time it took for converging to next ECMP next hop. |
| Descriptions |  |
| Results |  |

# Solutions – TAP Aggregation (OpenFlow) MLNX-OS

# Solutions – Routing on Host Cumulus

## Routing on Host with BGP

|  |  |
| --- | --- |
| Objectives of Tests | Verify that ping between hosts work and further verify that BGP ECMP based load balancing works in Leaf/Spine setup with ROH |
| Test configurations and connections (as shown in the figure) | C:\Users\narayana\Pictures\Picture1.png |
| Test Procedure | 1. Setup BGP neighbor ship between leaf and spine 2. Enable Quagga on Host1 and Host2 3. On one side of Leaf1 and Leaf2, directly connect Hosts (emulating L3 on Server for ROH), and other side configure BGP from leafs to spines switches (Can try both BGP unnumbered or with IP addresses depending on customer need). 4. Check routing table on leaf1 and leaf2 to see multiple path for destination using ECMP 5. Ping each server IP from the other server to ensure connectivity, and check interface counters on the switches to make sure traffic is taking the expected path through leaf and spine switches. 6. Using any available traffic gen tool, generate traffic (IMIX) from 10.10.10.1 to 10.10.11.1 and reverse direction and measure the link counters to see the traffic is getting load balanced. |
| Expected Result | 1. Verify Ping works from one host to another  2. Verify the ECMP load balancing  3. Verify show ip route command shows multiple paths to destination |
| Descriptions |  |
| Results |  |

## 17.2 Routing on host with OSPF

|  |  |
| --- | --- |
| Objectives of Tests | Verify that OSPF ECMP based load balancing works in Leaf/Spine setup with ROH |
| Test configurations and connections (as shown in the figure) | C:\Users\narayana\Pictures\Picture1.png |
| Test Procedure | 1. Setup OSPF neighbor ship between leaf and spine 2. Enable Quagga on Host1 and Host2 3. On one side of Leaf1 and Leaf2, directly connect Hosts (emulating L3 on Server for ROH), and other side configure OSPF from leafs to spines switches (Can try both OSPF unnumbered or with IP addresses depending on customer need). 4. Check routing table on leaf1 and leaf2 to see multiple path for destination using ECMP 5. Ping each server IP from the other server to ensure connectivity, and check interface counters on the switches to make sure traffic is taking the expected path through leaf and spine switches. 6. Using any available traffic gen tool, generate traffic (IMIX) from 10.10.10.1 to 10.10.11.1 and reverse direction and measure the link counters to see the traffic is getting load balanced. |
| Expected Result | 1. Verify ping works from one host to another  2. Verify ECMP load balancing  2. Verify show ip route command shows multiple paths to destination |
| Descriptions |  |
| Results |  |

# SOLUTIONS – Storage Solution (with & without RoCE)

## Non-RoCE

|  |  |
| --- | --- |
| Objectives of Tests | Measure storage throughput – traditional / non-RoCE storage cluster |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Connect storage cluster and client device to switch on same IP subnet. 2. Configure switch ports with same untagged VLAN for storage cluster and client connections. 3. Configure ECN on all storage servers and clients. 4. Configure ECN on all connected ports for traffic class 0. 5. Run read test (whichever is applicable) –    1. Block storage w/ FIO – 1/2/4/8/16/32/64KB IO    2. Object storage w/ COSBench – 64/128/256/512KB & 1/4/8MB objects 6. Run write test (whichever is applicable) –    1. Block storage w/ FIO – 1/2/4/8/16/32/64KB IO    2. Object storage w/ COSBench – 64/128/256/512KB & 1/4/8MB objects |
| Expected Result | 1. Storage read and write throughput meet or exceed specification for storage cluster. |
| Descriptions |  |
| Results |  |

## Lossless RoCE w/ QoS

|  |  |
| --- | --- |
| Objectives of Tests | Measure storage throughput – Resilient RoCE storage array |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Connect storage node and client devices to switch on same IP subnet. 2. Configure switch ports with same untagged VLAN for storage cluster and client connections. 3. Configure ECN on all storage servers and clients. 4. Configure RoCEv2 on all storage servers and clients – including DSCP trust, CNP priority 48, and CM priority 26. 5. Configure ECN on all connected ports for traffic class 3. 6. Configure buffer pools – one for background traffic and one for RoCE. 7. Bind switch priorities 3 and 6 to priority groups 3 and 6. 8. Map interfaces with connected clients and storage nodes to shared pool created for RoCE traffic. 9. Configure traffic class 6 (CNP traffic) to strict priority on interfaces with clients and storage nodes connected. 10. Enable QoS Trust for L3 on the switch. 11. Run read test – Block storage w/ FIO – 1/2/4/8/16/32/64KB IO 12. Run write Block storage w/ FIO – 1/2/4/8/16/32/64KB IO |
| Expected Result | 1. Storage read and write throughput meet or exceed specification for storage cluster. |
| Descriptions |  |
| Results |  |

## Resilient RoCE

|  |  |
| --- | --- |
| Objectives of Tests | Measure storage throughput – Lossless RoCE storage array |
| Test configurations and connections (as shown in the figure) |  |
| Test Procedure | 1. Connect storage node and client devices to switch on same IP subnet. 2. Configure switch ports with same untagged VLAN for storage cluster and client connections. 3. Configure ECN on all storage servers and clients. 4. Configure RoCEv2 on all storage servers and clients – including DSCP trust, CNP priority 48, and CM priority 26. 5. Configure PFC on priority 3 on all storage servers and clients. 6. Configure ECN on all connected ports for traffic class 3. 7. Configure buffer pools – one for background traffic and one for RoCE. 8. Bind switch priorities 3 and 6 to priority groups 3 and 6. 9. Map interfaces with connected clients and storage nodes to shared pools – priority 3 is mapped to a lossless pool, priority 6 is mapped to a lossy pool. 10. Configure traffic class 6 (CNP traffic) to strict priority on interfaces with clients and storage nodes connected. 11. Enable QoS Trust for L3 on the switch. 12. Enable PFC for priority 3 on all ports with client and storage servers connected. 13. Run read test – Block storage w/ FIO – 1/2/4/8/16/32/64KB IO 14. Run write Block storage w/ FIO – 1/2/4/8/16/32/64KB IO |
| Expected Result | 1. Storage read and write throughput meet or exceed specification for storage cluster. |
| Descriptions |  |
| Results |  |

# Solutions – HFT PIM-SM

# Appendix

## About Mellanox Switches

### Designed for Open Networking

With Spectrum®, you can choose the OS that best fits your need; MLNX-OS or Cumulus Linux. Mellanox Spectrum is an Open Ethernet System and Open Source Network operating systems can run on top the Spectrum ASIC and switch.

For Customer\_name, the proposed network OS is **Cumulus Linux** or **Mellanox OS** due to extensive features support, fast technology innovation, and Overlay-SDN integration scalability and flexibility.

### Differentiated ASIC

Mellanox Ethernet switches, based on the homebred Spectrum ASIC offer the highest packet rate, lowest latency and lowest power consumption with the 100GbE Spectrum product line.

Mellanox Spectrum® is the only switch capable of handling superior microburst absorption with fair and predictable performance.

Mellanox leverages high performance expertise to design the only flawless, high performing Ethernet switch ASIC.

### End-2-End Solution

Mellanox is the only vendor in the industry providing a complete top of the line DC networking solution including NICs, cables/transceivers and switches based on in-house designed components (ASICs). An E2E solution guarantees application awareness and complete solution-level testing by Mellanox. In addition, it helps getting faster and more predictive support, when Mellanox acts as a single point of access for the investigation of any potential issue. Since early 2016, Mellanox Support is fully certified by Cumulus to cover 1st and 2nd level support, so that the E2E story expands even more to cover the Net-OS as well.

The proposed switch the Customer\_Name DC RFP is Mellanox SN2700, SN2100, for both ToR and aggregation.



## Mellanox Spectrum Advantages

### Latency

300ns regardless of the packet size + “Zero Jitter”

Other than being important for High Frequency Trading and In-memory DB applications, low latency is extremely important for next-gen NVMe based storage systems. This is because the disk access latency is so low, that cutting the network latency from >10us to 1.5us can affect the overall performance – IO/s, BW and latency of a SDS system today. Zero jitter, or keeping the same latency regardless of the packet size and the load on the switch, can help solve synchronization issues and make applications more predictable.

### Performance

4.77Bpps switch performance at 6.4Tb/s (3.2Tb/s for both Rx and Tx) and zero packet loss guarantee. Full line-rate on any packet size from 64B – 9KB as proved by RFC 2544 testing.

Any switch can move Jumbo frames around, but small packets are essential in today’s data center as well and losing them without congestion is just unacceptable.

For example, most Software-Defined Storage systems use small packets to signal redundant copy completion between two or more redundant disk systems. Losing this packet would mean that the transaction did not go through and a full copy needs to be sent again. In this case the network will have additional load that could have been prevented and more importantly, the overall performance of the storage system will be degraded – the user will keep on waiting for the copy operation to complete.

### Buffering

The Spectrum buffer design offers a 16MB **fully shared** buffer, that means all the switch ports share that same buffer pool with dynamic buffer allocation.

In addition, supreme microburst absorption helps preventing packet drops.

Scheduling fairness: upon congestion at a single recipient, all transmitting flows will get exactly the same amount of bandwidth, which guarantees traffic flow fairness regardless of the ports in use. This enables keeping SLA levels and gives ease of mind when designing the data center’s QoS scheme.

### Low Power Consumption

Typical power consumption of Mellanox SN2700 is 150W (+ any optical components). Spectrum is the lowest power-consuming switch on the market.

In addition to allowing OPEX savings and fitting into the trend of green DC designs, in some cases, it may allow an additional server to be installed into the same rack on behalf of the saved power, compared to the competition.

### Traffic Visibility / Telemetry

As it is becoming increasingly complex to manage networks, and network administrators need more tools to understand network behavior, it is necessary to provide basic information about network performance, identify network bottlenecks, and provide information for the purposes of network optimization and future planning. Therefore, network administrators are required to constantly review network port behavior, record port buffer consumption, identify shortage in buffer resources, and record flows, which lead to excessive buffer consumption.

Mellanox Spectrum provides the ability to enable Histogram Sampling of the port buffer occupancy at a <1us granularity comparing to the standard 1/second sampling rate.

Mellanox Spectrum advantages:

* Record occupancy changes over time
* Provide information for different levels of buffer occupancy
* Provide the amount of time the buffer has been occupied during the observation period

### Best Fit for RoCE (if applicable)

In addition to standard Ethernet implementations, the Spectrum ASIC was designed with the vision to provide the best support for evolving technologies that can greatly increase the performance of data center applications while lowering the CPU overhead on the servers – called RDMA or RoCE (RDMA over Converged Ethernet). Mellanox is a leader in this technology and Spectrum is equipped with mechanisms that allow best in class support for RoCE in the DC network:

* Applications over RoCE (including Storage applications like: iSER/SMB-Direct) work faster and the transport is more predictable – Spectrum provides the lowest latency and jitter.
* Fair distribution of BW between all clients with the same QoS profile – provided by Spectrum’s fully shared buffer and scheduling architecture.
* Detects congestion faster – Spectrum has a unique feature called “Fast ECN” sending the notification up to 8us faster than the competition.
* Resiliency – Unique technologies in Spectrum provide better survivability of the RoCE network (i.e. different lossy/lossless configuration schemes and PFC deadlock prevention).
* Future-ready – Supports various Adaptive-Routing schemes and improvements making this future technology more ready for RoCE traffic.
* RoCE aware switch – Use of RDMA header for ACL and RDMA flow (QP) for ECN Hash.
* Technology leadership – Mellanox is a leading contributor to RoCE and, hence, this technology is best tested and tuned on Mellanox E2E solutions.

## Sample Configurations

* MLAG
* OSPF
* BGP
* BGP Unnumbered
* Routing on host
* VXLAN with NSX
* VXLAN with LNV
* VXLAN with EVPN
* VXLAN Asymmetric
* Buffer configurations

# Heading 1 [Just a sample of style usage – you can delete]

|  |  |
| --- | --- |
|  | PLEASE USE THE STYLES LISTED IN THIS TEMPLATE AND NOT WORD’S DEFAULT STYLES WHEN PREPARING YOUR DOCUMENT.  You can display the correct styles by left-clicking the small box of the Change Styles group. See red circle in the figure below.  Click |

## Heading 2

### Heading 3

Body Text

* ***Procedure Heading***

Code Block

Code Block 2

Code Block 3

* List Bullet
* List Bullet 2
* List Bullet 3

1. List Number
2. List Number 2
3. List Number 3

Table 3: Table Caption

* To insert a caption

1. Highlight the table
2. Right click
3. Insert Caption…
4. Set the following:

* Label: Table
* Position: Above selected item

| Table Heading |  |  |
| --- | --- | --- |
| Table Body Text   * Table List Bullet * Table List Bullet 2 * Table List Bullet 3   Table List Continue  Table List Continue 2  Table List Continue 3   1. Table List Number 2. Table List Number 2 3. Table List Number 3   Table Code |  |  |

Table 4: Table Two Columns

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Figure 1: Caption

* To insert a caption

1. Highlight the figure
2. Right click
3. Insert Caption…
4. Set the following:

* Label: Figure
* Position: Above selected item